webdyn

Flexitron group



MTX-Tunnel V11.19

Software User Manual

Index

General Notes 9

Important Information 9

Revision Information 9

1. Introduction 10

2. MTX-Tunnel Features 11

3. FAQ. MTX-Tunnel Basic Concepts 22

4. MTX-Tunnel Configuration 31

5. Webserver Service 40

6. Telnet Service 44

7. AT Commands. User API 47

7.1 AT Commands MTX-Tunnel User Procedure 47

7.2 Standard AT Commands Supported by MTX-Tunnel 52

7.3 Special MTX-Tunnel AT Commands (API) 53

7.4 AT Commands Batch Files 66

8. MTX-Tunnel Configuration Parameters 68

8.1 General Configuration Parameters “MTX_” 68

8.2 Configuration Parameters Related with COM1: “COMM” 127

8.3 Configuration Parameters Related with COM2: “COMM2_” 133

8.4 Telnet Configuration Parameters: “TELNET_” 139

8.5 WebServer Configuration Parameters: “WEBSERVER_” 150

8.6 Wakeup Configuration Parameter: “WAKEUP_” 158

8.7 GPRS connection configuration parameters: “GPRS_” 160

8.8 TCP Configuration parameters: “TCP_” 168

8.9 UDP configuration parameter: “UDP_” 172

8.10 ALARM Configuration parameter: “ALARM_” 175

8.11 FIREWALL configuration parameters: “FIREWALL_” 180

8.12 SMS Configuration parameter: “SMS_” 182

8.13 DynDNS configuration parameter: “DYNDNS_” 196

8.14 DNS Parameter configuration: “DNS_” 202

8.15 ULP Configuration parameter: “ULP_” 220

8.16 LOGGER configuration parameters: “LOGGER_” 228
8.17 MODBUS configuration parameter: “MODBUS_” 245
8.18 MODBUSTCP Configuration parameters: “MODBUSTCP_” 258
8.19 CSD: “CSD_” Configuration Parameters 263
8.20 Digital I/O and relay configuration parameters: “GPIO_” 267
8.21 Configuration parameters related to analog inputs: “ADC_” 276
8.22 Configuration parameters related to: “LINK_” 281
8.23 TEMPERATURE configuration parameters: “TEMPERATURE _” 288
8.24 GPS Configuration parameters: “GPS_” 292
8.25 SNMP configuration parameters: “SNMP_” 296
8.26 MQTT Related configuration parameters: “MQTT _” 301
8.27 TACACS+ related configuration parameters: “TACACS_” 317
8.28 Configuration parameters related to WAVENIS: “WAVENIS_” 320
8.29 Configuration parameters related to IEC870-5-102: “IEC102_” 322
8.30 Configuration parameters related to DUAL SIM: “DUALSIM_” 327
8.31 Configuration parameters related to WIRELESS M-BUS: “WMBUS_” 330

CONFIGURATION EXAMPLES 334

1. ANNEX: Basic Scenarios, Configuration Examples 334

1.1 EXAMPLE: LTE-4G/3G/2G-Serial tunnel. MTX-Tunnel is configured as a TCP/IP server and permanently connected to Internet using a SIM provided by the operator with a fixed IP address. 334

1.2 EXAMPLE: MTXTunnel configured as TCP/IP server and occasionally connected to 3G. Network operator dynamic IP addresses are used. 336

1.3 EXAMPLE: MTXTunnel configured as TCP/IP server and occasionally connected to 3G. Dynamic IP addressing. Firewall and authorized phone numbers active 338

1.4 EXAMPLE: MTXTunnel configured as TCP/IP client and connected permanently to 3G. Network operator dynamic IP addresses are used. 341

2. ANNEX: Advanced Scenarios, Configuration Examples 343

2.1 EXAMPLE: Advanced 3G-RS232 Tunnel – Setting up the MTX Tunnel as permanently connected and TCP/IP Server configured to send the IP by socket to a private DNS server. 343

2.2 EXAMPLE: Advanced 3G-RS232 Tunnel – Setting up the MTX Tunnel as a permanently connected TCP/IP Server and configured to send the IP address to a web server by HTTP 346

2.3 EXAMPLE: Tunnel 3G-RS232 – Set up the modem as a permanently connected TCP/IP server, with DynDNS and Client Temporal socket enabled. 350

2.4 EXAMPLE: Advanced Tunnel 3G-RS232. Set up the modem as a TCP/IP Server connected

occasionally when a GPIO and ADC level changes. 353

2.5 EXAMPLE: Dual Tunnel 4G-RS232. Controlling 2 RS232 devices with 1 modem and SIM card. Socket connection server type 356

2.6 EXAMPLE: Double 3G-RS232 tunnel. Two RS232 devices controlled by a single modem and a single SIM card. Client socket connection type. 358

2.7 EXAMPLE: Serial Cable replacer RS232-4G-RS232. 360

2.8 EXAMPLE: Serial Cable replacer RS232-4G-RS232 via MQTT. 363

2.9 EXAMPLE: Serial 3G-UDP Tunnel. MTX configured as Client/Server UDP connected permanently to 3G and using a SIM card with a fixed IP address. 367

2.10 EXAMPLE: 3G-Serial Tunnel with SSL security. MTX-Tunnel configured as TCP/IP Client connected to GPRS permanently using a SIM card with a dynamic IP address 369

2.11 EXAMPLE: Advanced tunnel 4G/3G/2G-Serial - Settings for MTX-Tunnel as TCP/IP Client with a connection 3 times an hour. 371

2.12 EXAMPLE: Advanced tunnel 4G-serial - Receiving and sending data to servers with UDP data packets. 374

2.13 EXAMPLE: A 3G-serial gateway, broadcasting data that begins with a chosen header to a server. 376

2.14 EXAMPLE: Conversion of a GSM communications system into an IP (3G/2G) communication system. 378

2.15 EXAMPLE: Use of MTX-Tunnel as 4G/3G/2G-RS232 gateway and for time synchronization of an external device connected to the modem. 381

2.16 EXAMPLE: Use of MTX-Tunnel as a serial datalogger 384

2.17 EXAMPLE: Use of MTX-Tunnel as a datalogger. Sending customized and periodic serial data for proprietary protocols. 387

2.18 EXAMPLE: 3G-Serial transparent gateway with SNMP service activated 391

2.19 EXAMPLE: Sending GPS positions in real time to server using TCP socket or HTTP. 395

2.20 EXAMPLE: Sending GPS positions, temperature and trailer opening detection for cold chain monitoring. 400

2.21 EXAMPLE: Alarms sent by SMS and VOICE call upon detection of 220V power loss and upon return of power. 406

2.22 EXAMPLE: Access RS232/RS485 port of a device via MQTT (for serial RAW data management). Transparent gateway “RS232/485 - MQTT”. 408

2.23 EXAMPLE: Data transmission by RS232 via SMS for the control of electronic equipment with proprietary protocols. Use of ALIAS to send non-printable characters. 412

2.24 EXAMPLE: Alarms sent by SMS and MQTT message before detection of 220V power loss and before the return of power. Modem with supercap of 1 minute of autonomy. 414

3. ANNEX: ULP Configuration and Example Scenarios 418

3.1 EXAMPLE: Sending an SMS alarm when the digital input is activated with ultra low power

consumption. 418

3.2 EXAMPLE: Periodic sending of digital and analog input value telemetry by 4G/3G/2G using ULP modem. Sending to MQTT broker. 420

3.3 EXAMPLE: Sending digital and analog input value telemetry by 4G/3G/2G according to an event (not periodically) using ultra low power modem. Sending to MQTT broker..... 424

3.4 EXAMPLE: Timing alarm activation of IP-RS232 tunnel using MTX-ULP..... 429

3.5 EXAMPLE: IP-RS232/RS485 Tunnel using an ULP consumption modem. Programmed timing activation. 431

3.6 EXAMPLE: Periodic monitoring via 4G/3G/2G of a 4-20mA sensor with ultra-low consumption modem. Activating sensor supply via relay. Sending to broker MQTT. 434

4. ANNEX: API Configurations and Example Scenarios 438

4.1 EXAMPLE: API used for reading MTX-Tunnel GPIOs and ADCs from 3rd party site. 438

4.2 EXAMPLE: Using API to send data coming from a 3rd party webpage form and output serial port MTX-Tunnel. 442

4.3 EXAMPLE: Using API to switch a relay from a third-party or end user webpage. 446

4.4 EXAMPLE: Using API to send SMS to external PLC devices connected to COM1 at the same time that the 4G-RS232 tunnel is active. 449

5. ANNEX: Scenario and Configuration Examples for Wavenis 452

5.1 EXAMPLE: Remote reading of pulse counters with radio communications (868 MHz) from PLC through modbus RTU protocol. 452

6. ANNEX: Examples and Scenarios for Automatic Readings of Modbus Devices 457

6.1 EXAMPLE: PLC Modbus RTU periodically registers readings, sends them as a JSON object to webserver. 457

6.2 EXAMPLE: Periodically obtaining readings from 5 electronic meters and sending them to a webserver using the ModBus RTU protocol. 461

6.3 EXAMPLE: Reading and periodic sending of records of 5 Electric meters with ModBus RTU protocol and memory map for different records to a webserver. 465

6.4 EXAMPLE: Regular reading of 3 Modbus temperature sensors and then sending to a webserver(Model MTX-Temp-RS485-IP65 of MTX accesories)..... 469

6.5 EXAMPLE: Regular reading and delivery of Modbus records of a PLC + IP gateway for remote update of the PLC firmware to a webserver. 472

6.6 EXAMPLE: ADVANCED example for modbus devices reading and broadcasting to a web platform via a JSON object. 476

6.7 EXAMPLE: Reading modbus Word and Binary registries, sending data to webplaform using JSON object. 480

6.8 EXAMPLE: Reading and writing Modbus registers via SMS. 484

6.9 EXAMPLE: Reading and sending Modbus registers from a PLC to a webserver. Remote and occasional access to PLC registers in real time and for remote update of PLC's firmware..... 488

6.10 EXAMPLE: Periodical sending of vehicle's GPS locations and the registers of Modbus RTU temperature sensor RS485 to a Web server.Occasional 3G-RS232 gateway for PLC's firmware actualization. 492

6.11 EXAMPLE: Relays control, MTX modem digital and analog inputs from a SCADA system using Modbus TCP protocol. 497

6.12 EXAMPLE: Providing IP to Master Modbus RTU and Modbus RTU slaves. 500

6.13 EXAMPLE: Reading and sending Modbus registries periodically from a PLC to an FTP server. 504

6.14 EXAMPLE: Modbus TCP/Modbus RTU 4G/3G/2G gateway. 508

6.15 EXAMPLE: Reading/writing MODBUS records on RS232 device using MQTT communication. 510

6.16 EXAMPLE: About how to provide MQTT communications to master Modbus RTU and Modbus RTU slaves. 513

6.17 EXAMPLE: Reading and periodic sending to an MQTT broker of the MODBUS registers of a PLC. DUAL SIM configuration due to loss of connectivity of an operator. 518

7. ANNEX: Metering Scenario Examples. Meter Reading Scenarios via GSM, GPRS, 3G & 4G..... 522

7.1 EXAMPLE: Basic example for meter reading with simultaneous GPRS connections and GSM calls. 522

7.2 EXAMPLE: An advanced example for reading 2 serial devices using a single modem/SIM and simultaneously using GPRS connection and GSM calls. There is additional serial tunnelling when there are no GSM/ GPRS connections. 525

7.3 EXAMPLE: Example of meter reading via IP connection for real-time readings and for Energy operator. Preference for Energy operator's IP connection. 530

7.4 EXAMPLE: Example of reading of several pulse counters (dry contact or reed). Sending data to web platform. 533

7.5 EXAMPLE: Example of pulse meter reading (dry contact or reed) with sending data to the Web Platform. 536

7.6 EXAMPLE: Reading meter boxes via 3G in scenarios with a modem that can't be disconnected (the current modem must be kept). 540

7.7 EXAMPLE: Basic example for reading 2 meters with 1 modem (and a single SIM), combining GPRS connections and GSM calls. 544

7.8 EXAMPLE:. Reading IEC counters, DLMS/Cosem, Modbus with protocols implemented in an MQTT platform. 548

7.9 EXAMPLE: Meter reading via GSM call and/or IP communications (without SSL/TLS security). Incorporation of Device Manager (with SSL/TLS security) for modem management. 552

7.10 EXAMPLE: Meter reading via GSM data call (CSD) and IP communications (with SSL/TLS security). Incorporation of Device Manager (with SSL/TLS security) for modem management. 556

7.11 EXAMPLE: Reading of counters per CSD call and IP communications with SSL/TLS. Incorporation of Device Manager for modem management and connection between modems and the meter box reading platform. 561

7.12 EXAMPLE: Monitoring of 7 meters with pulse outputs via MQTT with alarm to detect the opening of the meter access door. 565

7.13 EXAMPLE: Monitoring of 1 IEC870-5-102 meter via RS232 in real time (every 5 minutes) with data sent to the HTTP platform. 571

7.14 EXAMPLE: Monitoring of 16 IEC870-5-102 counters via RS485 in real time (every 5 minutes) with data sent to the MQTT platform. 576

7.15 EXAMPLE: Basic example for meter reading via CSD call in environments with only 3G coverage (that is, where there is no 2G coverage). 581

7.16 EXAMPLE: Monitoring (instantaneous values and fiscal closure) of 1 IEC870-5-102 counter via RS232 in real time (every 5 minutes) with data sent to the MQTT platform. Support for CSD calls in environments with 2G coverage. 584

8. ANNEX: EXAMPLE SCENARIOS AND CONFIGURATIONS FOR THE USE OF DIGITAL OUTPUTS AND RELAYS 592

8.1 EXAMPLE: Switching of relays based on the MTX astronomical clock. 592

8.2 EXAMPLE: Switching of a relay based on the MTX astronomical clock and based on scheduled times for special days. 595

8.3 EXAMPLE: Switching of a relay for 5 seconds with an SMS message. 598

8.4 EXAMPLE: Example of using MQTT to activate/deactivate relays of a 3G modem from an Android/IO mobile phone. 601

8.5 EXAMPLE: Periodic monitoring by 4G/3G/2G of 0-10V sensor activating the power supply of the sensor through a relay output of the modem. Sending to web server using HTTP GET. 605

8.6 EXAMPLE: SMS alarms to change up to 8 digital inputs. 609

8.7 EXAMPLE: SMS alarms to change up to 2 analog inputs. 612

8.8 EXAMPLE: Voice call alarm to change up to 8 digital inputs. 615

8.9 EXAMPLE: Voice call alarm to change up to 2 analog inputs of 0-10V and 4-20mA. 618

8.10 EXAMPLE: Instant sending of telemetry via MQTT of up to 8 digital inputs. 620

8.11 EXAMPLE: Instant sending of telemetries via MQTT of up to 2 analog inputs 0-10V and 4-20mA. 624

8.12 EXAMPLE: Change by SMS of up to 9 digital outputs or 9 relays. 628

8.13 EXAMPLE: Activate a relay by voice call and by dry contact inputs to open the parking gate. 632

8.14 EXAMPLE: Remote management of up to 9 digital outputs and relays through Telnet and/or MQTT/S. 636

8.15 EXAMPLE: Automatic activation of relays according to the value of Modbus registers. Automatic sending of state changes to the MQTT/S platform. 641

8.16 EXAMPLE: Automatic activation of relay output depending on the value of analog 0-10V probe 646

8.17 EXAMPLE: DATALOGGER. Telemetry (digital and analog inputs) sent periodically to the web

server via HTTP (via JSON object) + transparent IP-RS232 gateway for remote access to the RS232 device. 651

8.18 EXAMPLE: DATALOGGER. Telemetry (digital and analog inputs) sent periodically to broker MQTT (via JSON object) + transparent IP-RS232 gateway for remote access to RS232 device. 656

8.19 EXAMPLE: JSON customization and sending topics in the data frames sent by MTX-Tunnel. 661

8.20 EXAMPLE: Control of the relays, digital outputs, digital inputs and 0-10V/4-20mA analog inputs of an MTX modem from a SCADA system using the MODBUS TCP protocol. 669

8.21 EXAMPLE: About writing to Modbus RTU devices based on the digital and analog inputs of the modem. 673

8.22 EXAMPLE: Remote management of up to 4 0-10V analog outputs via Telnet and/or MQTT/S. 678

9. ANNEX: MTX-TUNNEL CONNECTION WITH WEB PLATFORMS 682

9.1 EXAMPLE: MTX-Tunnel connection with Platform/MQTT broker 682

9.2 EXAMPLE: MTX-Tunnel connection with Control Platform via dedicated TCP Socket. 689

9.3 EXAMPLE: MTX-Tunnel configuration modification from a PHP, ASP... webserver 692

9.4 EXAMPLE: Automatic connection to the Cervello platform via SMS 696

10. ANNEX: EXAMPLE SCENARIOS OF READING AND SENDING DATA FROM W-MBUS DEVICES WITH MTX-TUNNEL 698

10.1 EXAMPLE: Reading data from W-Mbus water meters and sending it transparently to the MQTT platform. Configuration of time windows and filter by manufacturer. 698

10.2 EXAMPLE: Reading data from W-Mbus water meters and W-MBus temperature sensors. Sending data transparently to the MQTT platform. Configuration of time windows, period and filters by manufacturer and serial number of each device. 702

1. ANNEX A: MTX-TUNNEL I/O DESCRIPTION 706

1.1 MTX-IoT-S 706

1.2 MTX-IoT 710

1.3 MTX-T 713

1.4 MTX-T2 714

2. ANNEX B: HOW TO USE VIRTUAL SERIAL PORTS (VSPE) 715

2.1 EMULATION SOFTWARE FOR SERIAL PORTS: VSPE 715

3. ANNEX C: GENERATION AND USE OF SSL CERTIFICATES 720

3.1 Certificate usage documentation 720

3. ANNEX D: MANUFACTURER FILTERS W-MBUS 722

MTX-Tunnel Release Notes 755

General Notes

Product is deemed accepted by recipient and is provided without interface to recipient's products. The documentation and/or product are provided for testing, evaluation, integration and information purposes. The documentation and/or products are provided on an "as is" basis only and may contain deficiencies or inadequacies. The documentation and/or products are provided without warranty of any kind, express or implied. To the maximum extent permitted by applicable law, Matrix Electronica further disclaims all warranties; including without limitation any implied warranties of merchantability, completeness, fitness for a particular purpose and non-infringement of third-party rights. The entire risk arising out of the use or performance of the product and documentation remains with recipient. This product is not intended for use in life support appliances, devices or systems where the malfunction of the product can reasonably be expected to result in personal injury. Applications incorporating the described product must be designed to be in accordance with the technical specifications provided in these guidelines. Failure to comply with any of the required procedures can result in malfunctions or serious discrepancies in results.

Furthermore, all safety instructions regarding the use of mobile technical systems, including GSM products, which also apply to cellular phones, must be followed. Matrix Electronica or its suppliers shall, regardless of any legal theory upon which the claim is based, not be liable for any consequential, incidental, direct, indirect, punitive or other damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information or data, or other pecuniary loss) arising out the use of or inability to use the documentation and/or product, even if Matrix Electronica has been advised of the possibility of such damages. The foregoing limitations of liability shall not apply in case of mandatory liability, e.g. under the Spanish Product Liability Act, in case of intent, gross negligence, injury of life, body or health, or breach of a condition which goes to the root of the contract. However, claims for damages arising from a breach of a condition, which goes to the root of the contract, shall be limited to the foreseeable damage, which is intrinsic to the contract, unless caused by intent or gross negligence or based on liability for injury of life, body or health. The above provision does not imply a change on the burden of proof to the detriment of the recipient. It is subject to change without notice at any time. The interpretation of this general note shall be governed and construed according to Spanish law without reference to any other substantive law.

Important Information

This technical description contains important information for the start up and use of the MTX-Tunnel application. Read it carefully before you start working with MTX Terminal Java enabled & MTX-TUNNEL. The warranty will be void should damage occur due to non-compliance with these instructions for use. We cannot accept any responsibility for consequential loss.

Revision Information

VERSION. 10.15 Release: July 2019

1. Introduction

The MTX-Tunnel is mainly a serial-4G/3G/2G and serial-gsm Gateway (gateway) designed for tele-reading applications of serial devices and tele-maintenance, that is, to avoid displacements simply by connecting a RS232 / RS485 serial cable to a device. You can connect to your serial devices as if they were connected to the serial port of your computer (electrical meters, PLCs, weather stations, etc.).

The current version of the MTX-Tunnel has greatly increased its performance in relation to previous versions, allowing to solve many of the real scenarios related to tele-maintenance and telemetry (control of digital and analog inputs and outputs, control of relays by Telnet commands , MQTT, SMS... sending GPS positions for fleet controls, remote configuration through Telnet, MQTT, SMS, webserver, Device Manager, multiple TCP gateways Client / Server / UDP, Serial Datalogger, Modbus Datalogger...).

HOW TO USE THIS MANUAL

If you are not familiar with MTX-Tunnel solution we recommend that you start with these sections:

- Read section “FAQ. MTX-Tunnel Basic Concepts”: by quickly reading these 10 pages, you will understand what can be done with MTX-Tunnel and whether it will be useful or not
- Read section “MTX-Tunnel configuration”: you will quickly learn how to configure the MTX-Tunnel and a step-by-step basic first configuration
- Read annex “Scenarios examples”: dozens of examples of real-case scenarios are currently available that are currently working in the field. Chances are, the solution you’re looking for is already in one of the examples and doesn’t need much more effort on your part. Find the example closest to what you need and modify it to fit your needs completely
- Read annex “Configuration parameters”: once you locate in the Annex the example of the scenario most similar to the application you want to carry out, use the chapter where the configuration parameters are explained to understand them and adjust them to your needs

2. MTX-Tunnel Features

Hardware Multiplatform

The new MTX-Tunnel can be installed in the Java enabled MTX-TERMINAL modem family. It is important to know that by using specific terminals the MTX-Tunnel will have behave differently than using general ones. You can order MTX-Tunnel to your distributor based on the following models:

MTX-IoT-S [4-N-N] P/N: 199802407	**Modem 4G/3G/2G** 1x RS232 1x RS485 1x USB 1x latch relay 8x digital inputs* 8x digital outputs (open collector)* 8x pulse counter (dry contact)* 2x analog inputs (0-10Vdc/4-20mA) 4x analogue output (0-10Vdc) *configurable as inputs or outputs
MTX-IoT-S [4-N-GPS] P/N: 199802409	**Modem 4G/3G/2G** 1x RS232/RS485 1x USB 1x GNSS (GPS+GLONASS) 1x latch relay 8x digital inputs* 8x digital outputs (open collector)* 8x pulse counter (dry contact)* 2x analog inputs (0-10Vdc/4-20mA) 4x analogue output (0-10Vdc) *configurable as inputs or outputs
MTX-IoT-S [4-N-N] AUS P/N: 199802408	**Modem 4G/3G/2G** 1x RS232 1x RS485 1x USB 1x latch relay 8x digital inputs* 8x digital outputs (open collector)* 8x pulse counter (dry contact)* 2x analog inputs (0-10Vdc/4-20mA) 4x analogue output (0-10Vdc) *configurable as inputs or outputs

MTX-IoT-S [4-N-GPS] AUS P/N: 199802410	**Modem 4G/3G/2G** 1x RS232/RS485 1x USB 1x latch relay 8x digital inputs* 8x digital outputs (open collector)* 8x pulse counter (dry contact)* 2x analog inputs (0-10Vdc/4-20mA) 4x analogue output (0-10Vdc) *configurable as inputs or outputs
MTX-IoT [3-S-N-N] P/N: 199801393	**Modem 3G/2G** 1x RS232 1x RS232/RS485 1x USB 3x digital inputs 2x digital outputs (open collector) 2x pulse counter (dry contact)* 2x analog inputs (0-50V/4-20mA) *shared with digital inputs
MTX-IoT [3-S-A-N] P/N: 199801403	**Modem 3G/2G** 1x RS232 1x RS232/RS485 1x USB 3x digital inputs 2x digital outputs (open collector) 2x pulse counters (dry contact)* 2x analogue inputs (0-50V/4-20mA) Analogue audio *shared with digital inputs
MTX-IoT [3-S-A-GPS] P/N: 199801448	**Modem 3G/2G** 1x RS232/RS485 1x USB 1x GNSS (GPS+GLONASS) 3x digital inputs 2x digital outputs (open collector) 2x pulse counters (dry contact)* 2x analogue inputs (0-50V/4-20mA) Analogue audio *shared with digital inputs

MTX-IoT [3-S-N-GPS] P/N: 199801456	**Modem 3G/2G** 1x RS232/RS485 1x GNSS (GPS+GLONASS) 1x USB 3x digital inputs 2x digital outputs (open collector) 2x pulse counters (dry contact)* 2x analog inputs (0-50V/4-20mA) *shared with digital inputs
MTX-IoT [4-S-N-N] P/N: 199801436	**Modem 4G/3G/2G** 1x RS232 1x RS232/RS485: 1x USB 3x digital inputs 2x digital outputs (open collector) 2x pulse counters (dry contact)* 2x analog inputs (0-50V/4-20mA) *shared with digital inputs
MTX-IoT [4-S-N-N] AT&T P/N: 199801439	**Modem 4G/3G/2G** 1x RS232 1x RS232/RS485 1x USB 3x digital inputs 2x digital ouputs (open collector) 2x pulse counters (dry contact)* 2x analog inputs (0-50V/4-20mA) *shared with digital inputs
MTX-IoT [4-S-N-N] AUS P/N: 199801446	**Modem 4G/3G/2G** 1x RS232 1x RS232/RS485 1x USB 3x digital inputs 2x digital outputs (open collector) 2x pulse counters (dry contact)* 2x analog inputs (0-50V/4-20mA) *shared with digital inputs

MTX-IoT [4-S-N-W868] P/N: 199801404	**Modem 4G/3G/2G** 1x RS232/RS485 1x RF (868MHz internal telemetry wavecard) 1x USB 3x digital input 2x digital output (open collector) 2x pulse counters (dry contact)* 2x analog inputs (0-50V/4-20mA) *shared with digital inputs
MTX-IoT [4-S-N-GPS] P/N: 199801452	**Modem 4G/3G/2G** 1x RS232/RS485 1x GNSS (GPS+GLONASS) 1x USB 3x digital input 2x digital output (open collector) 2x pulse counters (dry contact)* 2x analog inputs (0-50V/4-20mA) *shared with digital inputs
MTX-IoT-XS [4-N] P/N: 199801473	**Modem 4G/3G/2G** 1x RS232 1x RS485 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-IoT-XS [4-D] P/N: 199801475	**Modem 4G/3G/2G** 2x RS232 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-IoT-XS [4-N] AUS P/N: 199801474	**Modem 4G/3G/2G** 1x RS232 1xRS485 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-IoT-XS [4-D] AUS P/N: 199801476	**Modem 4G/3G/2G** 2x RS232 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T [2-N] P/N: 199801421	**Modem 2G** 1x RS232 1x RS485 1x USB 1x digital input (0-50Vdc)/pulse counter

MTX-T [3-N] P/N: 199801422	**Modem 3G/2G** 1x RS232 1x RS485 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T [4-N] (4G/2G) P/N: 199801424	**Modem 2G/4G** 1x RS232 1x RS485 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T [4-N] P/N: 199801445	**Modem 4G/3G/2G** 1x RS232 1x RS485 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T [4-N] AT&T P/N: 199801432	**Modem 3G/4G** 1x RS232 1x RS485 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T [4-N] AUS P/N: 199801450	**Modem 3G/4G** 1x RS232 1x RS485 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T2 [2-N] P/N: 199801455	**Modem 2G** 2x RS232 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T2 [3-N] P/N: 199801406	**Modem 3G/2G** 2x RS232 1x USB 1x digital input (0-50Vdc)/pulse counter
MTX-T2 [4-N] P/N: 199801438	**Modem 4G/3G/2G** 2x RS232 1x USB 1x digital input (0-50Vdc)/pulse counter

Gateway TCP, UDP, GSM, MQTT

MTX-TUNNEL can use these different protocols:

4G/3G/2G - Serial Tunnel (TCP Server mode): MTX-Tunnel is waiting for incoming connection from a specific TCP port. When it's accepted, the transparent tunnel serial-IP is created

4G/3G/2G - Serial Tunnel (TCP Client mode): the transparent tunnel serial-IP is created when a connection to a specific remote server is made by the MTX-Tunnel

4G/3G/2G - Serial Tunnel (UDP mode): MTX-Tunnel is connected to 4G/3G/2G and can create the transparent tunnel serial-IP using UDP protocol instead of TCP

4G/3G/2G - Serial Tunnel (MQTT mode): MTX-Tunnel connected to 4G/3G/2G allows establishing a transparent 4G/3G/2G gateway - Serial but using the MQTT protocol

Tunnel GSM-Serial: MTX-Tunnel allows the reception of conventional GSM calls (for example to accept the typical calls from energy operators (Endesa, Iberdrola...) for meter reading. MTX-Tunnel also allows the simultaneity of GSM calls with 2G connections In other words, you can access both your device via GSM and 2G Remember that if you need to use GSM calls with a 3G modem, you must configure the modem to work in 2G with the parameter GPRS_mode: 2g

GPRS/3G/4G Connections: Permanent/On request

Unlike in previous versions of the MTX-Tunnel, from version 5.0 onwards any type of GPRS/3G series tunnel (TCP Server, TCP Client, UDP, etc.) can be activated upon request. As well as the possibility of maintaining the GPRS/3G session active 100% of the time (also called permanent connection), the GPRS/3G session can be activated via one of the following methods:

Incoming SMS (any or only authorized numbers)

Incoming missed call (any or only authorized identified numbers)

A change in the level of digital input

A voltage value (ADC) reaching programmed level

Incoming data that is present on the serial port (only in TCP server mode)

GSM-Series Gateways with GPRS-Series Priority

Since MTX-Tunnelv7.11 version, the reception of GSM data calls with priority to 2G is managed. Very important in Metering applications (meter reading). MTX-Tunnel can establish a 2G-Serial gateway to read a meter in real time, but when it receives a GSM call to read the meter (typically from an energy operator such as Endesa, Iberdrola...) the 2G IP gateways- Serials are interrupted to give way to the GSM call, reestablishing IP communication at the end of the operator's GSM call.

NOTE: 3G models must be configured in 2G mode to accept GSM calls (GPRS_mode parameter: 2G).

Webserver

New MTX-Tunnel includes an embedded WebServer that can be activated. WebServer enables remote access to MTX-Tunnel using a normal Internet browser connection. Main WebServer features are:

WebServer public access or restricted access (login & password will be required)

Showing the status of digital and analog inputs

Changing the digital output level (example relay activation in MTX-IND terminal) in one click

Changing MTX-Tunnel parameter configuration remotely

Executing AT commands remotely with a simple Web page (as an example, see network coverage using AT+CSQ command)

WebServer includes API (HTTP GET) to integrate and remotely control the MTX-Tunnel from third party Web pages; for example, to change a relay's status from an external Web page

Telnet

The new MTX-Tunnel includes a small Telnet embedded server that can be activated. If active, MTX-Tunnel can be remotely accessed using a PC with a Telnet application.

Features:

Telnet public or restricted access (login & password). Improved security with OTP (one time password) and SHA-256

Able to enquire analog and digital inputs value/statues

Able to change digital outputs (relays in MTX-IND) status

Able to change MTX-Tunnel configuration parameters remotely

Execute AT commands remotely (for instance, to read the coverage, reset the equipment, read modbus registries of an associated equipment connected in real time, etc. remotely)

Telnet can be integrated into end application to control MTX-Tunnel remotely

In short, with Telnet you will be able to access your modem remotely without the need for moving around to change the configuration, check network coverage, etc.

SMS Alarm

It is possible to configure the MTX-Tunnel to send alarm SMS with configurable text before the change of state of a digital input. The SMS message can be sent to up to 10 different phone numbers. Allows you to send a different text SMS, if desired, depending on the value of the digital input (for example, "alarm on" / "alarm off").

It is possible to configure the MTX-Tunnel to send alarm SMSs (with configurable texts) up to 10 telephone numbers, depending on certain conditions. For example, you can send an alert SMS when a digital input

changes, when an analog input exceeds a certain value, etc.

Voice Call Alarms

In the same way as SMS alarms, it is possible to configure the MTX-Tunnel to generate alarms by voice call when a digital input is changed or when an analog input (0-50V or 4-20mA) exceeds a certain threshold.

MQTT Alarms

In the same way that alarms by SMS or Voice calls, it is possible to configure the MTX-Tunnel to send MQTT / S messages to an MQTT / S broker when a change of a digital input is detected, or an analog input varies true threshold etc.

Pulse Counters

All the digital inputs of the MTX modem can be configured as pulse readers, specially designed for Metering applications for reading water meters. The MTX-IOT-S model supports up to 8 pulse counters simultaneously.

Management of Digital Oputputs and relays

The digital outputs and relays of MTX modems can be configured in different ways: “manual” (to change the status of the output / relay by Telnet, MQTT, SMS,...), or to activate them by time, by timing, depending on the value of a digital input, or depending on the value of an analog input, or depending on the value of a modbus register of a slave modbus rtu device connected to the MTX, or when receiving a voice call, by astronomical clock, etc.

Full Control or MTX-Tunnel Using SMS, Telnet, MQTT/S, Modbus TCP

MTX-Tunnel allows you full control via SMS. AT commands can be sent to the MTX-Tunnel from a mobile phone (from any number or only from authorized phone numbers). This allows reading or changing the status of a digital input or output, changing the status of a relay, knowing the coverage remotely, obtaining the GPS position by SMS, reading / changing the modem configuration, etc.

The creation of custom ALIASes by the user is allowed. For example, if the ALIAS is created: “RELE1ON> AT ^ MTXTUNNEL = SETIO, 0,1” implies that the sending of an SMS with the text “RELE1ON” is interpreted by the modem as the AT command “AT ^ MTXTUNNEL = SETIO 0.1 ”, which will switch the GPIO0 output.

It is also possible to control the modem through remote AT commands through Telnet, through MQTT / S and even through Modbus TCP.

DynDNS

DynDNS is offered by dyndns.org, and can easily assign a DNS name to an IP dynamic address for free.

A SIM card can connect to GPRS/3G in 2 ways: fixed IP address or dynamic IP address. Dynamic IP

addresses are more common and cheaper in service. The problem is that the network operator assigns a different IP address when MTX-Tunnel establishes a GPRS/3G connection.

Using DynDNS you can associate a DNS name (yourdevice1.dyndns.org) to the IP obtained from Tunnel.

It is also compatible with the No-Ip service.

Private DNS

One of the most important features of the new MTX-Tunnel is the fact that they send their current IP address when it's changed (after new connections). In a real scenario with thousands of remote MTX-Tunnel working with dynamic IP SIMs, using DynDNS is not feasible. So it is recommended a private server be used for this.

MTX-Tunnel is capable of informing with its IP every time it changes, or periodically, and it can do so against a server via TCP socket, it can also inform using HTTP GET / HTTP POST to a WEB server, or also sending information to an MQTT / MQTTS topic.

Besides sending the IP address, it is possible to send information about the modem coverage, FW version, digital/analog I/Os status, modem temperature, etc. in the same data frame.

Firewall

If enabled, all incoming connections on MTX-Tunnel are firewalled. This is a protection method for the Serial-GPRS tunnel, Web services and Telnet. Also any unauthorized IP addresses will be blocked. You can configure up to 10 authorized IP addresses.

Automatic Timing Synchronization

MTX-Tunnel does not need to set the time. It is done automatically. It uses two time servers, using GPRS. This is mandatory when using the Datalogger feature on MTX-Tunnel in timing schedules to automatically activate the gateway at a set time. It allows to use two protocolos: protocol TP (Time Protocol) and NTP (Network Time Protocol).

Serial (RS232/485)-HTTP Tunnel

MTX-Tunnel can create an HTTP-Serial tunnel to access serial devices connected using a web page.

Serial (RS232/485)-SMS Tunnel

MTX-Tunnel can create a SMS-Serial Tunnel. This way, all of the text sent via SMS can be redirected to the output on the serial port of MTX-Terminal.

Ultra Low Power ULP Tunnel

MTX-Tunnel can be used with the MTX-IoT [4-S-N-N]-STD-N-ULP terminals and they are ideal in any application where consumption is critical as the modem and 4G/3G/2G are only connected for short periods of time at specific times.

FEATURES

Switch on the modem. Activate the 4G/3G/2G connection and the service (serial tunnel, WebServer, telnet, SMS ...) periodically every X hours, where X can be configured. Switch on the modem and activate 4G/3G/2G session and services (serial tunnel, WebServer, telnet, SMS...) periodically at specific times/ dates, which can be configured.

Datalogger sending via HTTP/S, MQTT/S

MTX-Tunnel has the capacity of a datalogger, that is, to store data inside its flash memory and, in the event that there is 4G/3G/2G coverage, send it to a central server via HTTP/S or MQTT/S, avoiding data loss. All data is sent in JSON format along with the timestap of the moment of capture.

The MTX-Tunnel can act as a datalogger for:

Reading pulse counters

Reading of digital and analog inputs

GPS position reading

Reading of RS232/485 serial port data (that is, it is capable of storing and subsequently sending “generic” data captured by a serial port).

Modbus registers. The MTX-Tunnel is capable of acting as a Modbus hub for up to 20 Modbus RTU slave devices, periodically reading their registers, storing them inside, and sending them to a server via HTTP/S, MQTT/S

Wavenis sensor logs

Security

MTX-Tunnel is capable of establishing secure sockets with encryption enabled against an SSL / TLS server. Only possible for Socket TCP Client tunnel connections. It also allows the sending of telemetries by HTTPS and MQTTS.

As of MTX-Tunnel v10, if desired, it is possible to install / delete certificates from SSL Root CA servers. You can do it locally or remotely).

Also, starting with MTX-Tunnel v10, it is possible to install, if needed, a client SSL/TLS certificate on the modem itself. This ensures the authentication of the device.

MTX-Tunnel also allows encryption of the configuration file (see parameter MTX_encryptedConfig) as well as block the flash memory of the modem (see parameter MTX_mes)

Modbus Equipment Monitoring

MTX-Tunnel is capable of periodically reading memory tables from Modbus devices (232/485 RTU series) and automatically sending them to a web server using a JSON object (http / https / mqtt / mqtts / ftp). If there is no coverage, 4G/3G/2G is capable of storing up to 1500 reading records in flash memory to send them when there is coverage, so as not to lose any data.

Integration of MTX-Tunnel in end applications. API

MTX-Tunnel can be easily integrated in a customer application as it can receive AT commands in the following interfaces:

COM1 (GPRS-serial tunnel can be active or not)

COM2 serial port

SMS

HTTP

Telnet

Client GPRS TCP socket. When tunnel is created, AT commands can be sent

It is possible to execute both AT commands –standard and Cinterion proprietary ones- and MTX-Tunnel commands.

Read the API manual section for more information.

3. FAQ. MTX-Tunnel Basic Concepts

If still you have doubts about the new MTX-Tunnel please read the following FAQ section:

What is the MTX-Tunnel?

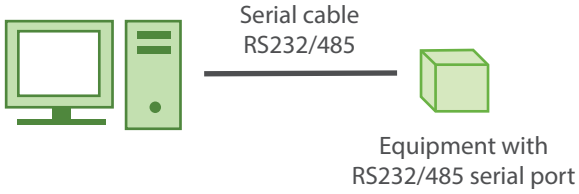
The MTX-Tunnel is an embedded software application that you can order from your distributor which is already installed into the MTX-Terminal modem family:



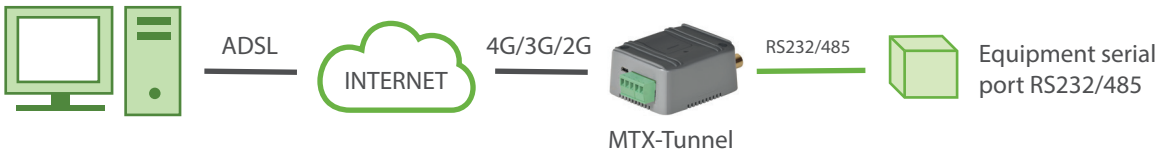
What is the MTX-Tunnel for?

The MTX-Tunnel can be mainly used to create a transparent serial Gateway (or tunnel) 4G/3G/2G/GSM (RS232/485). If you already have a machine or device with a serial port and want to control them remotely as if it were physically connected to your computer, MTX-Tunnel is the solution you need.

This is the scenario: serial equipment (RS232/RS485) connected to a PC to read/write data...



With GPRS-Serial MTX-Tunnel gateway, the above scenario is shown in the following example. Now your PC has to establish a TCP/IP connection using MTX-Tunnel. Then, EVERYTHING you send to this TCP/IP connection will be sent to the equipment's serial port by MTX-Tunnel. On the contrary, all of the information in the equipment's serial port is sent to your server using the GPRS network.



This feature (serial gateway 4G/3G/2G) is one of the many features that the MTX-Tunnel allows. Check the examples listed in this guide to grasp all the posibilities.

Is MTX-Tunnel needed at the PC server's side?

It depends, but in general, it is not needed at the PC server's side 99% of the time.

Not needed: If you already have your PC control software and have the option to connect via TCP/IP or UDP, a modem with MTX-Tunnel is not needed. Just configure the IP and the TCP port of the remote MTX-Tunnel and your PC will use the existing Internet connection to send and receive data remotely.

Not needed: If your PC control software does not have the option to connect using TCP/IP and the only option you have is to choose a COM port, the MTX-Tunnel modem is not needed. There are some freeware drivers for your operating system as Windows can emulate a COM port. Once this free driver is installed, a virtual COM (like COM100) will be installed in your PC and you must point to the IP and TCP port of the MTX-Tunnel remote. You must choose this virtual COM in your PC software. Please contact iotsupport@mtxm2m.com for more information about recommended emulation COM-TCP drivers.

Needed: If you need a "serial cable replacer" because you have to communicate two RS232 serial devices remotely and neither is a PC, i.e. you cannot install a virtual COM port because there is no operating system, you will need two MTX-Tunnels, one in each end. This is the scenario:



Who starts the connection?

MTX-Tunnel has the following modes TCP Server, TCP Client and UDP.

TCP Server mode. MTX-Tunnel is waiting to receive incoming connections. This means that the remote device (PC server) will start and establish the 4G/3G/2G - Serial Gateway. (In newer versions of MTX-Tunnel, a temporary client socket can be created when there is no connection available and MTX-Tunnel receives data via the series port

TCP Client mode. The MTX-Tunnel will start the TCP gateway. It will connect to the configured IP port of the server PC AND establish the GRPS-Serial Gateway automatically

UDP mode. UDP is not oriented to connection protocol. MTX-Tunnel just waits for the UDP packet and sends them to the serial port and vice versa. The data present at the serial port is sent to a PC via UDP

Check the examples in the Annex 1 and 2 for more information.

It is mandatory to be permanently connected to GPRS?

No, it's not.

If your application requires it (99% of the cases), the modem can always be connected to GPRS/3G. Remember that network operators will bill the data volume, not time.

If you do not need MTX-Tunnel to be connected 100% of the time and you want the connection to be sporadic, MTX-Tunnel can be activated in these ways:

Missed call from authorized phone number

Incoming SMS with the text “mtxtunnel on” from an authorized phone number

By a change on the level of an input

If the analog input is higher than the configured limits

With any date/hour scheduled timing

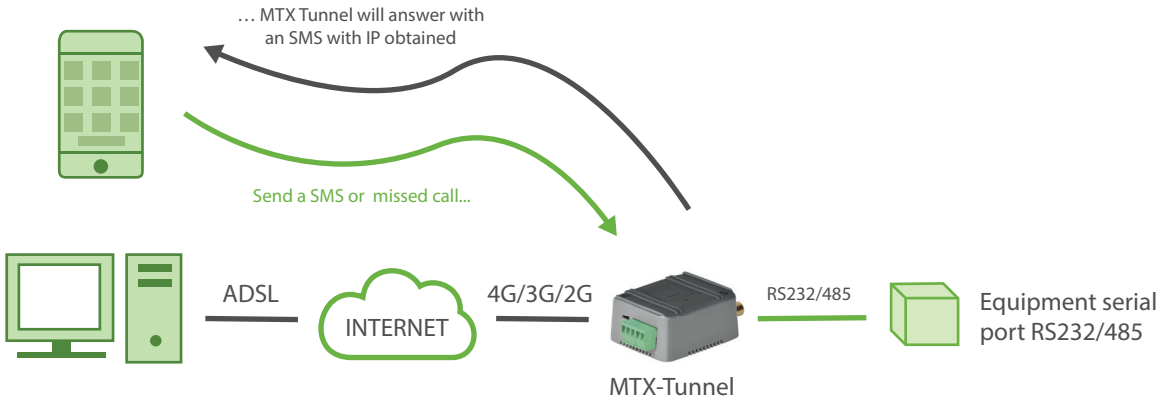
Just when data is present on RS232 port (only in TCP Server mode)

So... how long is MTX-Tunnel active for (GPRS/3G connected)?

It is configurable by the GPRS_timeout parameter. You can specify the time in minutes after which, if no GPRS connection is detected, MTX-Tunnel will close the session.

I want to use MTX-Tunnel as a TCP Server so I can connect to it periodically from my PC. Will I need a SIM card with a fixed IP?

It is not mandatory. There are various ways of finding out the remote IP if using SIMs with normal dynamic IP addresses. You can either make a missed call or send an SMS with the word “mtxtunnel on” to the remote MTX-Tunnel equipments. MTX-Tunnel will reply with an SMS including the IP obtained at this moment in time.



The new MTX-Tunnel is DynDNS featured. DynDNS is a service allowing you to associate a DNS name (like modem1.dyndns.org) to the IP obtained by MTX-Tunnel. For now, you can use this service for free: www.dyndns.org. It is also fully compatible with the free service offered by No-IP (www.no-ip.com).

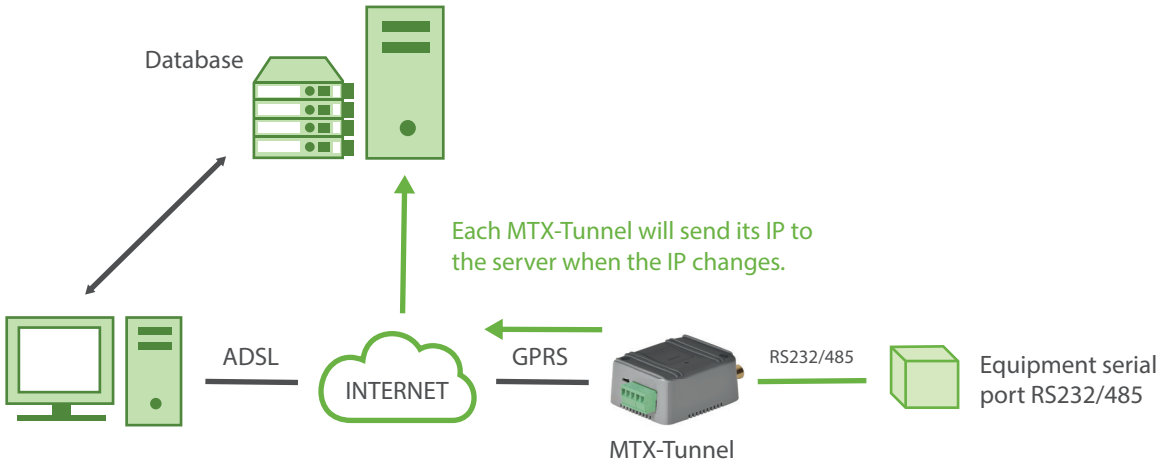
Keep in mind that these services are NOT offered by MTX-Tunnel; we do not offer support and we are not responsible for these third party services.

I found that in my case I’m going to use thousands of MTX-Tunnel devices. I cannot use missed calls or SMSs to work out IP addresses. I do not want to use DynDNS as it can be difficult to handle and even costly. What can I do?

The MTX-Tunnel can inform the server PC every time an IP address changes.

You just need to enable a configuration parameter so that each time MTX-Tunnel changes the IP it will

send frame data to a server PC with the following information: IMEI, the newly obtained IP and an optional user-configurable text.



Is it possible to send the new IP to a Web server? I'm more familiar with Web programming – ASP, PHP – rather than TCP-IP sockets. I'm thinking about using a database. Is this possible?

Yes, it does support this protocol conversion.

And via MQTT/MQTTS?

Yes, it can configure a TOPIC in the MTX-Tunnel to inform about the IP change (it can include other data like the digital inputs status, counters, analog inputs, etc.). Check the parameter DNS_mqttTopic.

MTX-Tunnel can be installed on several Cinterion-based modem terminals like MTX-65i, MTX-IND, MTX-IND-V1, MTX-65i-ULP, MTX-65i-GPS-V6, MTX-65i-RS485, as well as the 3G versions. What are the differences between them?

For the majority of applications, the MTX-65i is suitable.

If you require RS485 serial communication, you will need a MTX-65i-RS485 modem. If you require the remote activation of relays, or to be able to remotely read 4-20mA, the recommended modem is MTX-IND.

If power consumption is critical, use the MTX-65-ULP modem as it can be completely powered-off (consumption is around 2uA) except in the configured situations.

If you need some extra features like GPS positioning or GPRS series gateways, you should choose either the MTX-65i-GPS-V6 or MTX-65i-GPS-V7 series modem.

Obviously, if you require data at high velocity, the 3G versions of these models can be used.

You are talking about relays, analog inputs... Is MTX-Tunnel not a Serial-GPRS Gateway?

MTX-Tunnel is a serial-GPRS Gateway but can also simultaneouslycontrol digital inputs/outputs, analog inputs, relays, GPS receivers, and devices connected to SPI/I2C, as well as read modbus series devices and 868MHz radio devices.

For example,the MTX-Tunnel can remotely read a digital input or commute remotely with a relay. It can also send automatically, at a pre-configured time of every X seconds,

Also, it can send automatically, at a pre-configured time of every X seconds, the status of all inputs/outputs or the GPS position to a server PC or WebServer via HTTP/S GET or HTTP/S POST) or via MQTT/S, etc. SMS messages can also be used. It can also remotely read a sensor connected to SPI or I2C. See the example annex for more information.

I've read about sending AT commands using a TCP/IP connection. Can I use an AT command to change the value of a digital output?

Yes, you can send AT commands via TCP/IP; you can also send them from a normal modem, via serial ports or you can send them via SMS. You can also activate relays or read or write digital input/outputs on your Web page (see API section 7 and example scenarios)

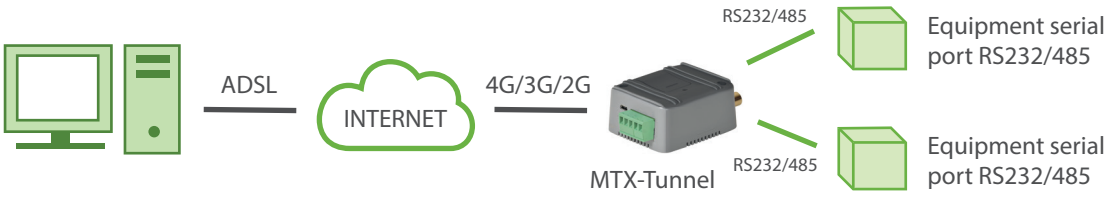
So I can activate a relay using a SMS, but... it is impractical because the AT commands are not intuitive nor easy to remember.

Yes. It is possible to send AT commands via SMS and therefore activate a relay; however, it is not necessary to send the exact AT command. MTX-Tunnel supports ALIAS which means you can configure the modem to understand the SMS with the text "RELE1ON" as the command "AT^MTXTUNNEL=SETIO,0,1" which would commute RELAY1 associated to GPIO0. Up to 10 ALIAS strings can be created.

Could I control two RS232 external devices with one terminal?

Yes, you can. MTX-Tunnel can control two RS232 external devices with just one SIM card. MTX-Tunnel will create two serial-GPRS tunnels running in parallel.

Just remember that the MTX-65i, and MTX-65i-ULP modems only have the TX and RX lines and therefore the secondary port cannot use flow control because the do not have the CTS and RTS lines; they only have the TX and RX lines.



What about using the MTX-Tunnel on a GPS modem? Can I use it for fleet management?

It is not intended for professional fleet management. The possibility of installing the MTX-Tunnel on a GPS modem exists so that the GPS position can be read in any given moment, i.e. the location of the modem

can always be known, be it through requesting it via IP, WebServer, Telnet or SMS.

MTX-Tunnel can be configured to automatically send the GPS position every X seconds. However, MTX-Tunnel doesn't internally store the position in the Flash memory, meaning that if there's no GPRS network and the GPS position can't be sent in the moment, the positions won't be stored for sending when the network coverage is restored. This is a typical application if GPRS coverage is lost. Therefor, it can only be used for basic fleet management as professional systems store the GPS position points like a data logger for future sending.

What is WebServer used for?

MTX-Tunnel WebServer which is included can be used to read the digital input/outputs or analog inputs and to change digital outputs easily on a PC (connected to the Internet) using a standard Internet browser.

Not only this, you can remotely see and modify the MTX-Tunnel's configuration parameters. Also you can execute AT remote commands, like network coverage (using the command AT+CSQ), check incoming SMS, etc.

What is Telnet used for?

You can basically do the same with the Telnet service in MTX-Tunnel as you can with WebServer but it is more commonly used for third party application integration. Please read the TELNET and API section.

If you are unsure about whether to use WebServer or Telnet, we recommend Telnet since the remote access is much quicker.

I'm worried about unauthorized access using WebServer or Telnet.

MTX-Tunnel has an internal firewall which can be activated. Then MTX-Tunnel will only accept connections from previously configured IP addresses. Any other IP addresses will be blocked. Check the parameters FIREWALL_ in this guide.

For maintenance purposes, I would like access to MTX-Tunnel at any time and in any location, from any IP address.

In this case Firewall WebServer can be disabled, but we recommend protecting the WebServer with a user Login and Password. MTX-Tunnel can work with or without a Login/Password (public WebServer). The same applies for Telnet.

And if I do not have a SIM with public IP, and I cannot connect via Telnet or Webserver to the modem, how do I manage it?

Via MQTT/S. With MQTT it is not necessary for the modem to have a fixed address, much less a public address. With MQTT, you can send AT commands to the modem remotely and get responses. See the MQTT_ parameter section for more information and Example 9.1 of this manual.

Could I or somebody else receive an SMS when an input changes, like alarm detection?

You can configure MTX-Tunnel to send special and configured SMS text strings to up to 10 configured different phone numbers. The SMS text can be configured in different text

strings related to digital or analog input. It can also send an MQTT message to the server.

MTX-Tunnel is a 4G/3G/2G-Serial tunnel Gateway. What is a SMS-Serial tunnel? Do you have some examples?

SMS-Serial Tunnel: You can define a key text, for example "MTX", so that when you send an SMS with that key at the beginning, such as an SMS with the text "MTX12345", the MTX-Tunnel forwards your serial port "12345", collect the response from the serial device connected to the MTX-Tunnel and forward it with another SMS.

I need a low power application terminal modem. What can MTX-Tunnel do?

It allows the modem to be switched off completely until an event happens. This way the power consumption is about 2 uA. The modem is off and cannot do anything. It cannot receive calls, SMSs or communicate at all. Therefore an event needs to happen.

This event can be a digital input level change or a periodic alarm. As an example, MTX-Tunnel can be configured to wake up every 24 hours, send the telemetry (all the inputs values, RS232...) and after five minutes of being awake in case it needed to communicate with a serial equipment connected to the MTX-Tunnel, the terminal will be automatically switched off for another 24 hours.

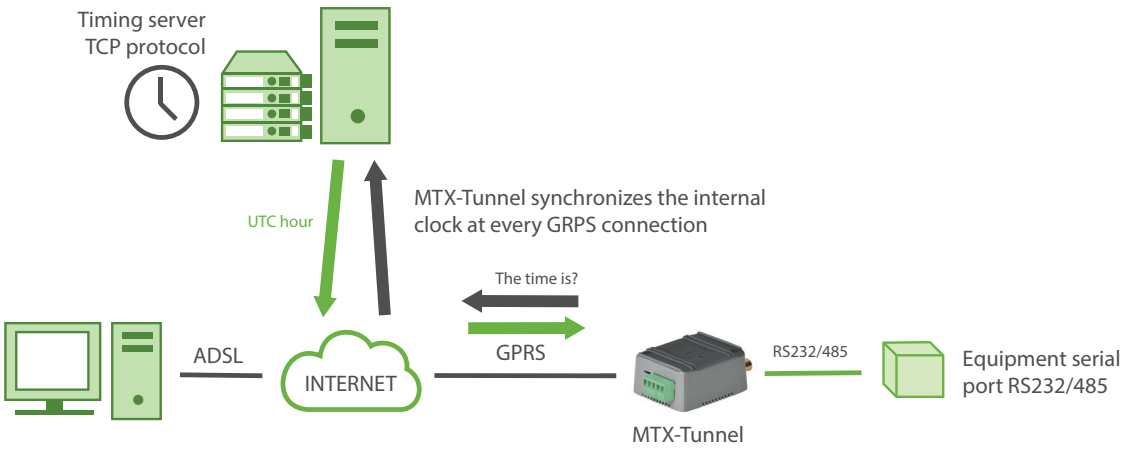
Is it also possible to define scheduled wake up tasks. For example, the modem can wake up for X minutes every day at 10.00 AM or only on the 1st and 15th days of the current month at 08.00am and 08.00pm, or as needed.

If MTX-Tunnel can be woken up in a configured time, does this mean that there is clock inside? But you can also say that it's powered off. Please explain.

ULP modems have their own Real Time Clock that allows to them to wake up at a configured/scheduled time/date.

Does the clock lose time? How can I assure that it always keeps time? that there is clock inside? But you can also say that it's powered off. Please explain.

MTX-Tunnel includes time synchronization via 4G/3G/2G; in fact, it is mandatory to use it if the RTC is to be used. This way, each time a connection to the 4G/3G/2G is made, another is also made to a time server to synchronize the time and ensure it is always accurate (to UTC time). It is also mandatory to use the time server when using the Datalogger function that is included in MTX-Tunnel.



For time synchronization it is possible to use TP or NTP servers. You must specify it in the MTX_TPProtocol parameter.

What about SSL security? How does it work?

For some sensitive applications you can use SSL / TLS communications if you want. Through SSL / TLS your data travels encrypted. It is only possible to use SSL / TLS when the modem is configured in TCP Client mode and the Server PC to which the MTX-Tunnel is connected is prepared to support SSL sockets under the specifications:

TLS Protocol Version 1.0 as RFC 2246.

SSL V3 as The SSL Protocol Version 3.0

WAP(TM) TLS Profile and Tunneling Specification as WAP-219-TLS-20010411-a

It is also possible to send data via HTTPS for sending telemetries over secure web servers. Same to send data via MQTTS.

From the MTX-Tunnel v10 on it is also possible to include in the modem up to 10 SSL Root CA certificates, as well as an SSL client certificate if needed.

What does “API” feature mean?

API is mainly a way to integrate MTX-Tunnel in end user application. Basically it is like a special AT command end user. It can be easily integrated in a web page and can also switch a relay (change an Output) for example.

API could also be used to remotely access the MTX configuration and send AT commands at the same time without knowing all of the configuration parameter syntax.

Please explain what MTX-Tunnel is able to do with a Modbus device connected to serial port on MTX-Terminal.

MTX-Tunnel is capable of periodically and autonomously interrogating a modbus rtu device connected to its serial port, it stores the variables in memory and will send them to its HTTP server or MQTT broker using a JSON object. If there are shipping issues, coverage issues, or whatever, you can store up to 1,500 readings for forwarding when modem connectivity issues are resolved.

Is it possible to access the advice via GPRS and GSM at the same time?

Indeed, since MTX-Tunnelv7.11 it has been possible to establish gateways with both at the same time, but GSM calls have priority. A typical is example is that of an energy operator (Endesa, Iberdrola...). They could want daily access to a meter to take a reading via GSM but you also want to be able to access the meter but via GPRS, MTX-Tunnel allows you to do this. Remember that to receive analog calls you must configure the modem to work in 2G mode with the parameter. 4G modems do not have the GSM calling feature.

How to configure a specific scenario?

The new MTX-Tunnel has a lot of configuration parameters, more than explained in the FAQ section. Take a look in the configuration section.

Next you can find step-by-step MTX-Tunnel examples of first configuration scenarios; they are very useful for you for first hands-on usage.

In the Annex there are lots of examples of scenarios with the appropriate configuration to get MTX-Tunnel working. Try to find the closest scenario to you and review the copy & paste selected configuration, it's nothing more special than that.

If you have specific questions contact the support line iotsupport@mtxm2m.com.

4. MTX-Tunnel Configuration

In this section you will create your first GPS-serial tunnel and we will guide you through every step of the procedure. You will create a TCP server tunnel, connect from your PC and transmit/receive data.

You Need

- PC with Windows 2000 or XP operating system possibly with an RS232 serial port. If you have a PC without an RS232 serial physical port, you can use a common USB-RS232 converter
- An RS232 serial cable to connect MTX-Terminal to the RS232 PC port. The cable must be pin to pin connected (pin1-pin1, pin2-pin2, pin9-pin9). Do not use a null-modem cable (crossed connection)
- Cinterion MES (Module Exchange Suite) application software which is free, please ask for the download link.

  https://www.dropbox.com/s/rw4tswmpmlbkkz4/mes_2.7.0.0.zip?dl=0
- If you use a USB cable instead of an RS232 cable, you will need the modem drivers. If you do not have them you can download them from the FTP or request them at iotsupport@mtxm2m.com

MES Installation & Configuration

MES application. What is needed for?

MES software will allow you to see your MTX-Terminal modem as a memory stick.

MES means “Module Exchange Suite” and it is a Cinterion application.

MTX-Tunnel application is stored inside the flash memory of the terminal. To access to the flash memory you need the MES application so you need to extract and edit the “config.txt” file.

-IMPORTANT NOTE. In the previous version of MTX-Tunnel, the configuration file was MTX-Tunnel.jad. On MTX-Tunnel do not use/edit this file.

The MTX-Tunnel configuration file is “config.txt” file. After being extracted from the internal terminal memory you can edit it with a normal text editor, like Windows Notepad.

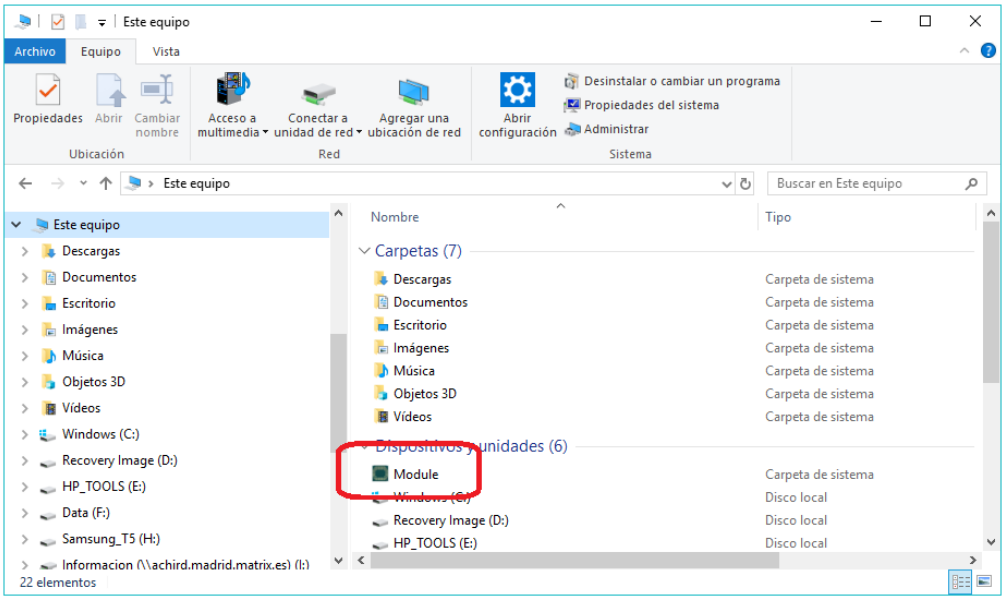
After editing this file you have to copy it again into the terminal’s internal memory using MES application software.

ATTENTION. MES application MUST be configured BEFORE using it for the FIRST time

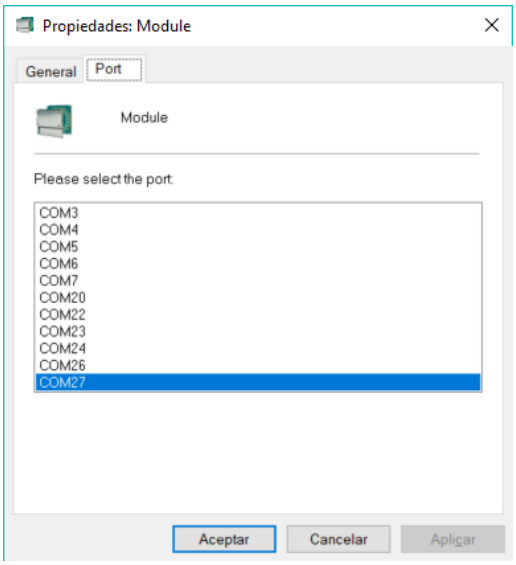
How is MES application configure?

After installing the MES application, you have to configure it. This MES application needs to know the COM serial port of your PC, which is connected to MTX Terminal modem.

Go to “My Computer” and look for a blue icon.



Right “Click” this icon and select “Properties”. Then, select COM port in “Port” section.



You can see the COM ports available in your computer. From this list select the COM port which the MTX terminal modem is connected to and then “Click” on “Apply” button. Now the MES application is configured. This step is not needed if you do not change the COM port to another available one.

If instead of connecting your computer using the modem DB9 connector you prefer to connect it via the microUSB connector, you will need USB drivers.

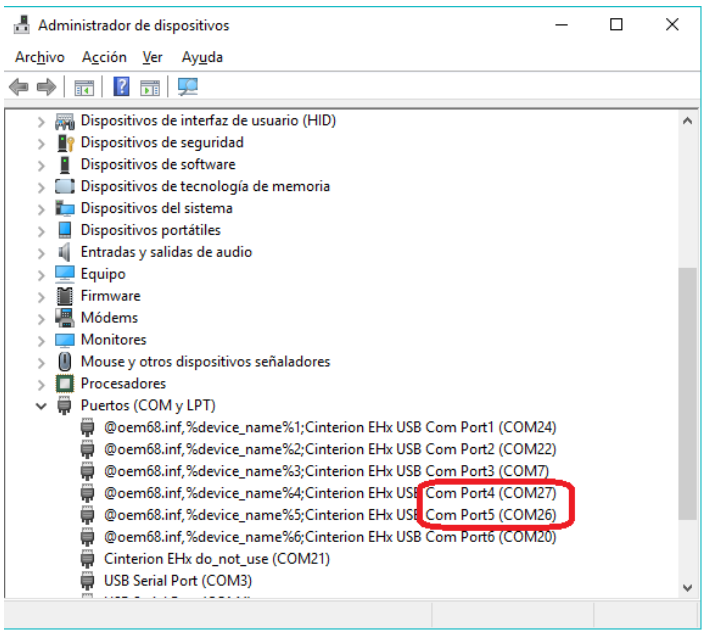
For MTX-3G versions:

https://www.dropbox.com/s/bdxcm1ec9qrbavb/EHSx%20Driver%20v1113.zip?dl=0

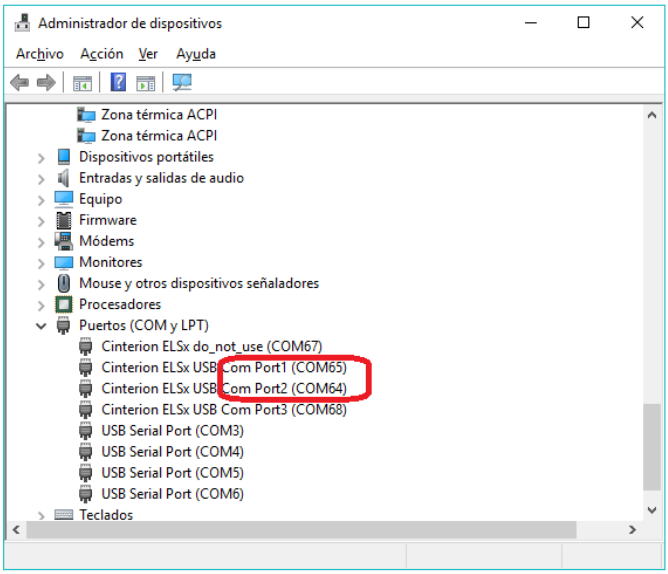
For MTX-4G versions:

https://www.dropbox.com/s/77ebcwmeqd8zmz2/ELS61.rar?dl=0

Once the drivers are installed, Windows will show several virtual COM ports. For MTX-3G versions, choose port 4 or 5 (in this case they are the ports COM27 and COM28).



In case of MTX-4G versions, choose port 1 or 2 (in this case they are the ports COM65 and COM64).

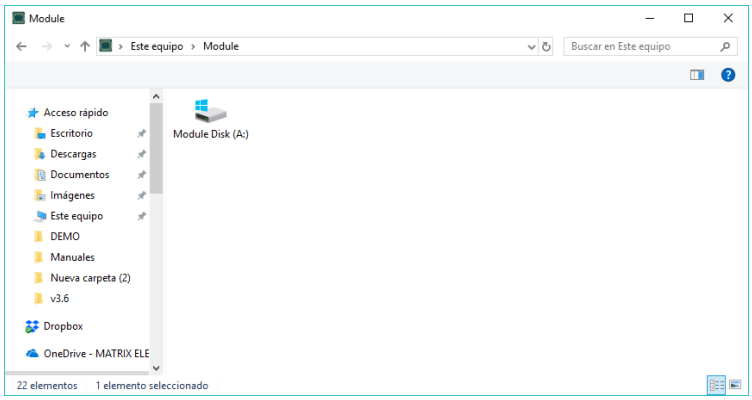


FIRST MTX-Tunnel Configuration

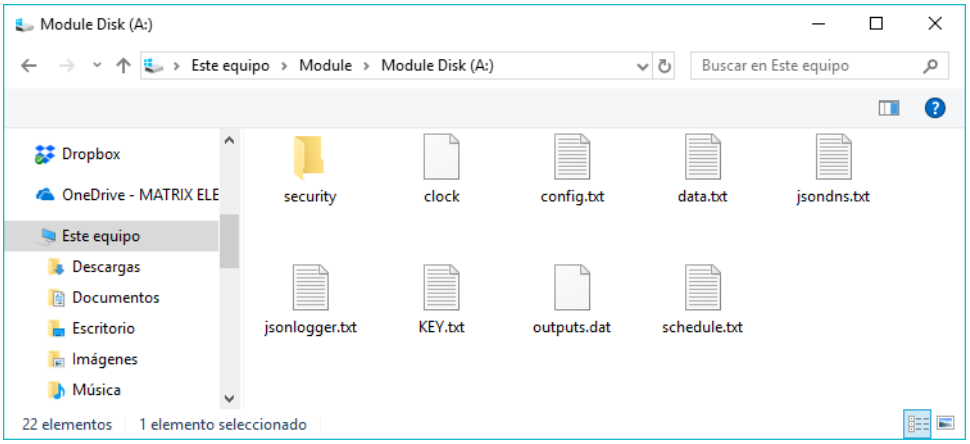
If you are not familiar with MTX-Tunnel, we recommend you do this test configuration to learn about the "MTX-Tunnel TCP-Server" scenario. Follow these steps:

- Do NOT apply power to MTX modem terminal. Do not insert a SIM card into MTX Modem. Open the SIM holder.
- Connect the terminal modem to the computer using a normal RS232 cable (not null modem).
- Apply power to the modem and wait at least 10 seconds.

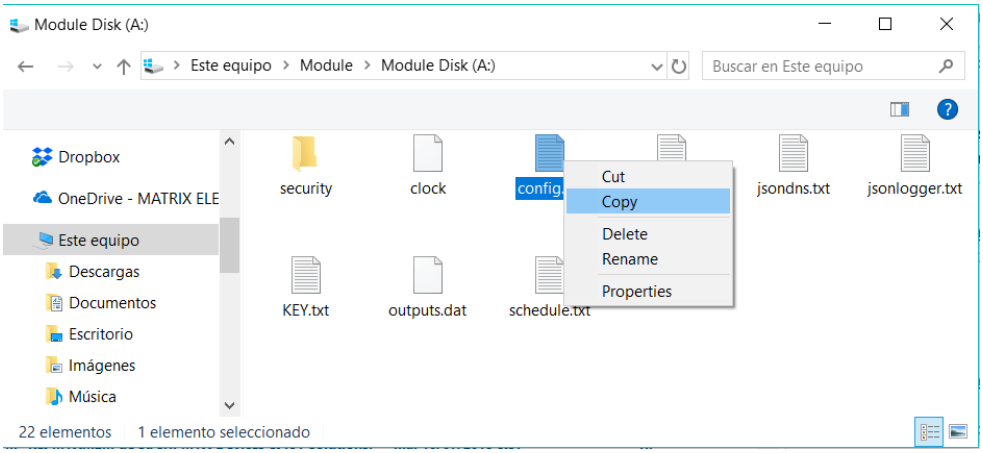
  Then, if the modem does not have a SIM card, MTX-Tunnel enters into CONFIGURATION mode.
- Please "double click" on MES icon in "My computer". This screen will appear:



- Please "double click" into "Module Disk A: /" (or press ENTER) to see files into MTX-Terminal modem memory.



- If you can see the files, copy/paste “config.txt” into a hard disk computer directory to edit it. This is because this file cannot be edited in this memory; you have to do in your computer.



- Open the “config.txt” file with Notepad. The “config.txt” configuration file contains the main information for running MTX-Tunnel application correctly. Write this configuration as follows:

  MTX_PIN: 0000

  MTX_mode: server

  MTX_urc: on

  MTX_model: 199801436

  GPRS_apn: movistar.es

  GPRS_login: MOVISTAR

  GPRS_password: MOVISTAR

  GPRS_timeout: 0

  GPRS_DNS: 8.8.8.8

  TCP_port: 20010

  COMM_baudrate: 115200

  COMM_bitsperchar: 8

  COMM_autorts: on

  COMM_autocts: on

  COMM_stopbits: 1

  COMM_parity: none

  FIREWALL_enabled: off

  SMS_allPhones: on

For the first time please just modify the blue marked fields.

MTX_PIN parameter: please write your SIM Card's PIN number to be used. If it doesn't have a PIN, write 0000.

In MTX_model please write which MTX-Terminal you're going to use (you will find the name on the sticker on the bottom of the modem). It is preferable to use the P/N (9 digit number) that you will find on the same sticker.

It is also IMPORTANT to modify and write the GPRS_apn, GPRS_login, GPRS_password parameters of the 4G/3G/2G SIM card network operator.

We recommend that you use your common or familiar/popular SIM card for this first "hands-on" use, like Movistar or Vodafone. We have noticed that other not well-known network operators use Proxies and block entry connections (in same way the ADSL router in your home/office does). Otherwise, you will have to use NAT.

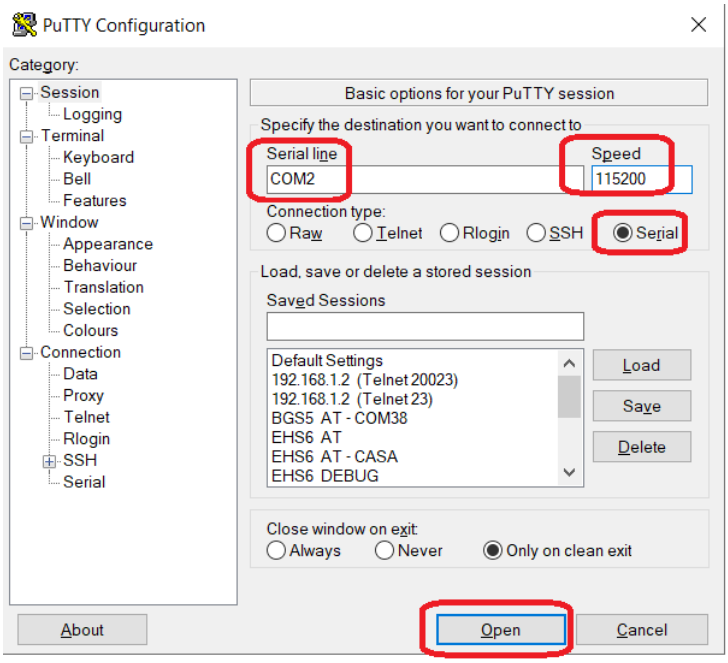
Low cost or virtual operators can be used for client connections/configurations (MTX_mode: client) but not for Server connections/configurations (MTX_mode: server) like we do in this example because incoming connections are blocked.

Please check with your operator if Proxy is used.

- Save the changes in the "config.txt" file and copy it into MTX terminal memory using MES to overwrite the old one.
- After this please switch off or remove the power from the modem. Insert the SIM card and close the SIM holder. MTX-Tunnel is now ready to use.

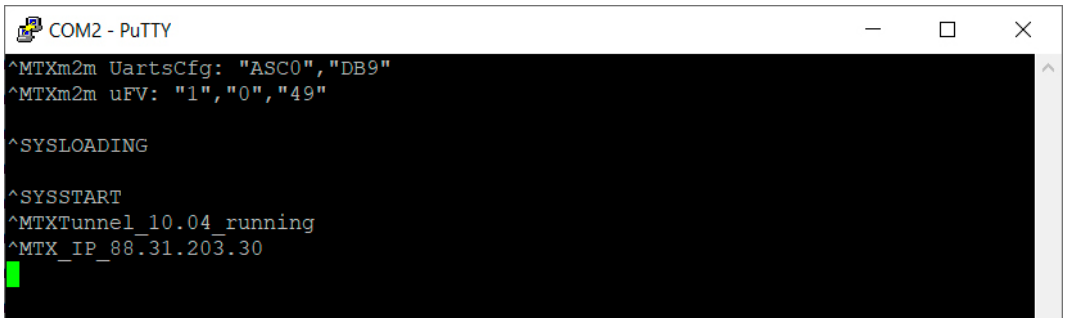
Testing MTX-Tunnel Configuration

- Open a HyperTerminal or another similar terminal application like putty. Configure PC R232 serial port with the same value MTX has; i.e. the same as the "config.txt" example (115200, 8, N, 1 and HW flow control).

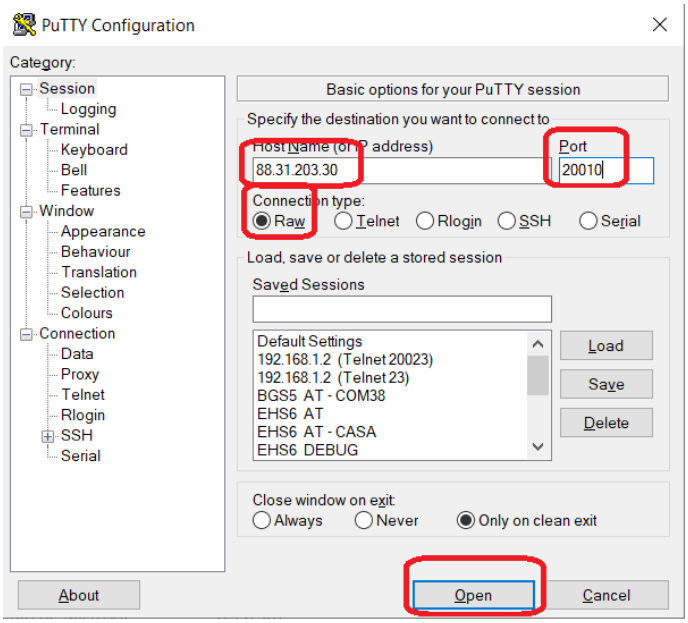


- Then, apply power to the modem connected to the computer. You will be able to see the green LED flickering slowly which indicates that the terminal is looking for GSM network registration. After a few seconds (not more than 60) you will see the LED flickering with a fast 1 flash every 4 seconds. This indicates that the terminal is registered with the GSM network. After a few more seconds you will see a blue LED turn on. This means that the modem is connected to the 4G/3G/2G network and has an IP from the operator (Note: the configuration of the LED behavior can change. Check the section of configuration parameters: MTX_blueLEed, etc.)

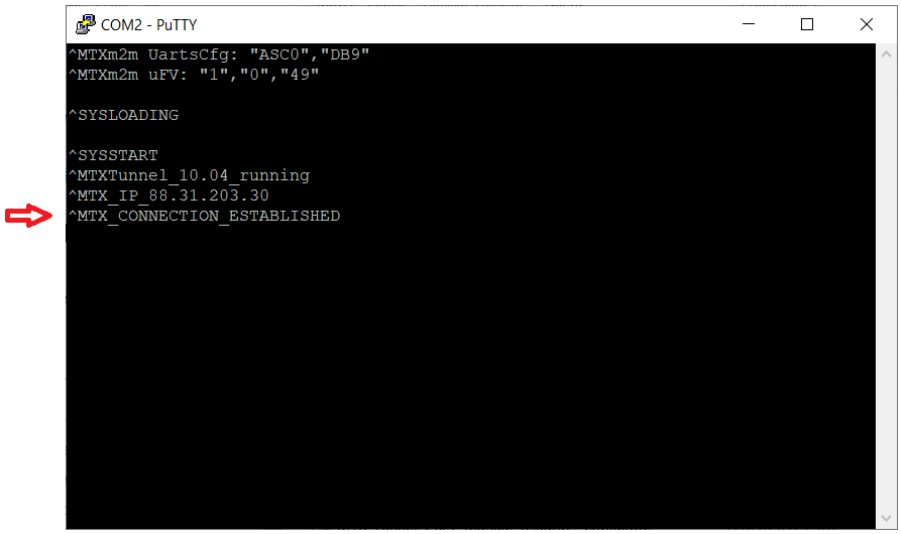
  You can also check in the HyperTerminal window application (MTX_urc is on) messages from MTX-Tunnel. You can check that the IP shown is the IP that your network operator has assigned. Now MTX-Tunnel is READY. It is waiting for incoming connections in the 20010 configured port.



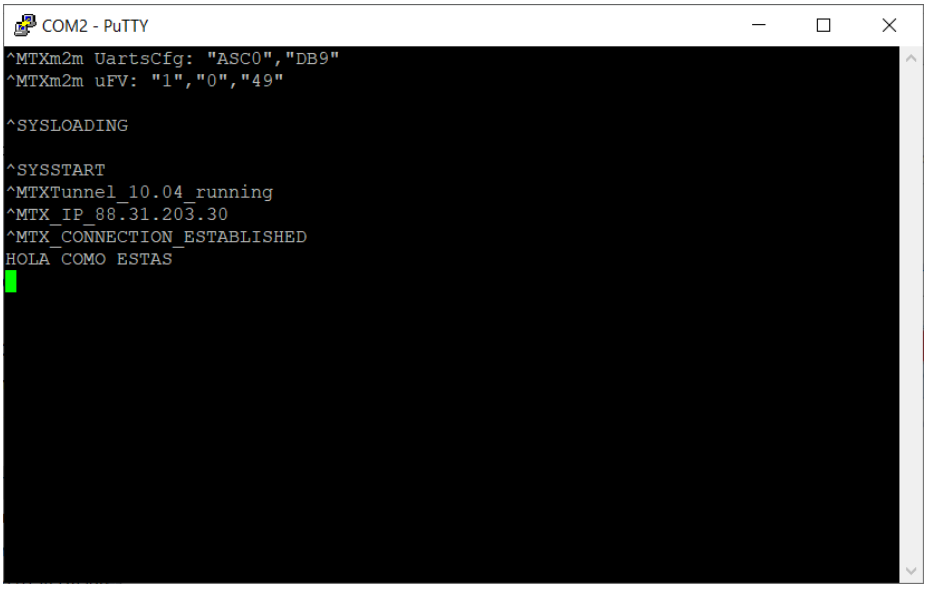
- Open ANOTHER HyperTerminal session (putty). Now, do not use the COM port option, choose the TCP/IP Winsock option and specify the IP address above and the 20010 TCP port.



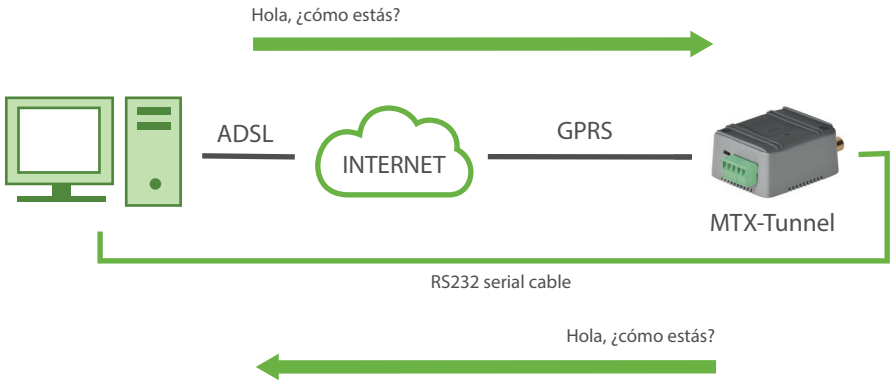
- After pressing the “Open” button you will see that you are connected to MTX-Tunnel using the serial port in the HyperTerminal window. The GPRS-serial gateway is done. You will see the message ^MTX_CONNECTION_ESTABLISHED indicating the entering TCP connection.



- Now, everything you write in the second HyperTerminal window is received by MTX-Tunnel and sent to the COM serial port. If you write (send) “HOLA, COMO ESTAS” from HyperTerminal TCP/IP WinSock connection, it is received in the other MTX-Tunnel HyperTerminal window.



In summary, now you have created your first GPRS/3G -- serial Gateway tunnel, like the following picture:

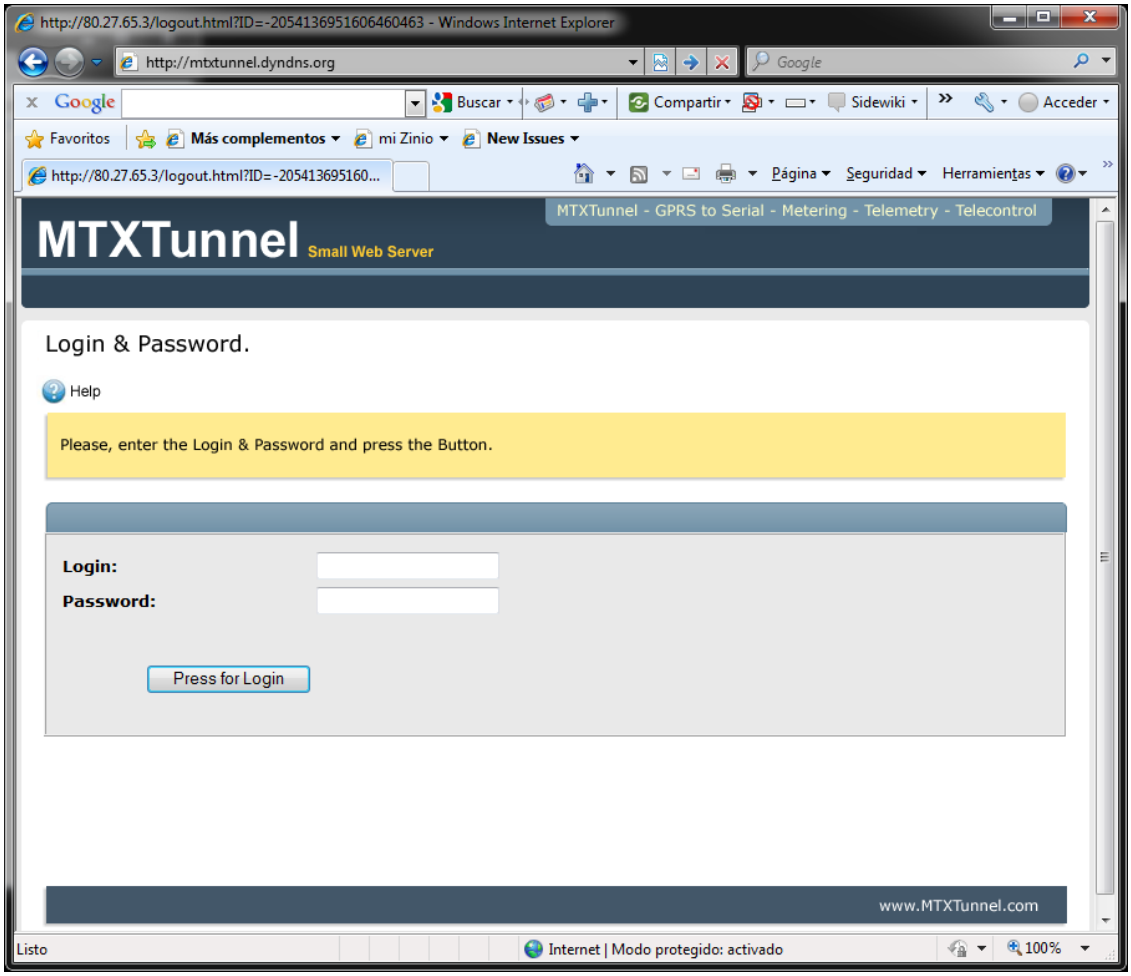


5. Webserver Service

With MTX-TUNNEL version 5.0 WebServer is included. If enabled, you will be able to remotely access the GSM modem terminal with MTX-TUNNEL if your PC connected to the Internet.

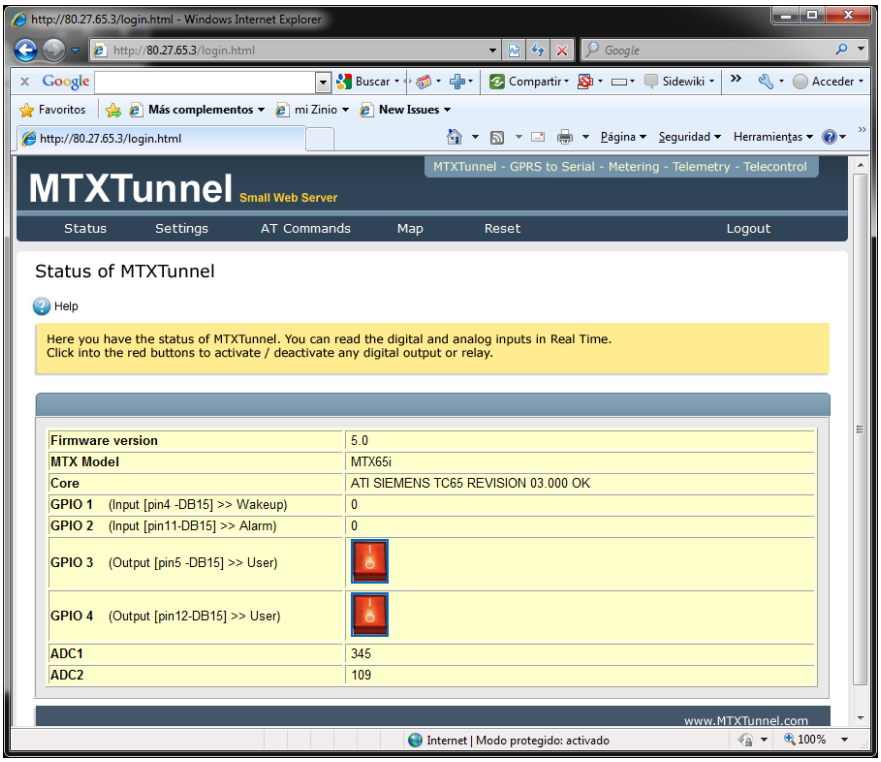
To access WebServer, open an Internet browser –we recommend Microsoft Internet Explorer-on any PC and introduce the MTX-Tunnel IP address or DNS if the DynDNS feature is activated. See next picture.

If you specified a Login and Password configuration parameter, this is what the WebServer will ask for at first.

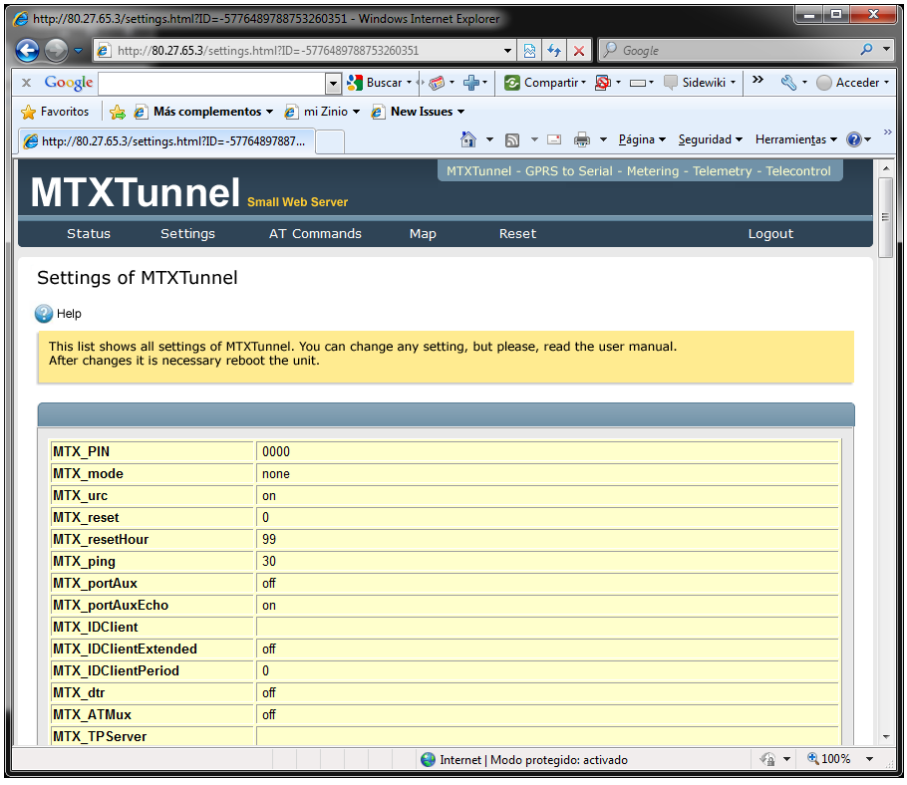


After the login and password are checked, the next page is WebServer Status. You can find MTX-Tunnel information such as the firmware version, Cinterion model information and GPIO/Analog status/value.

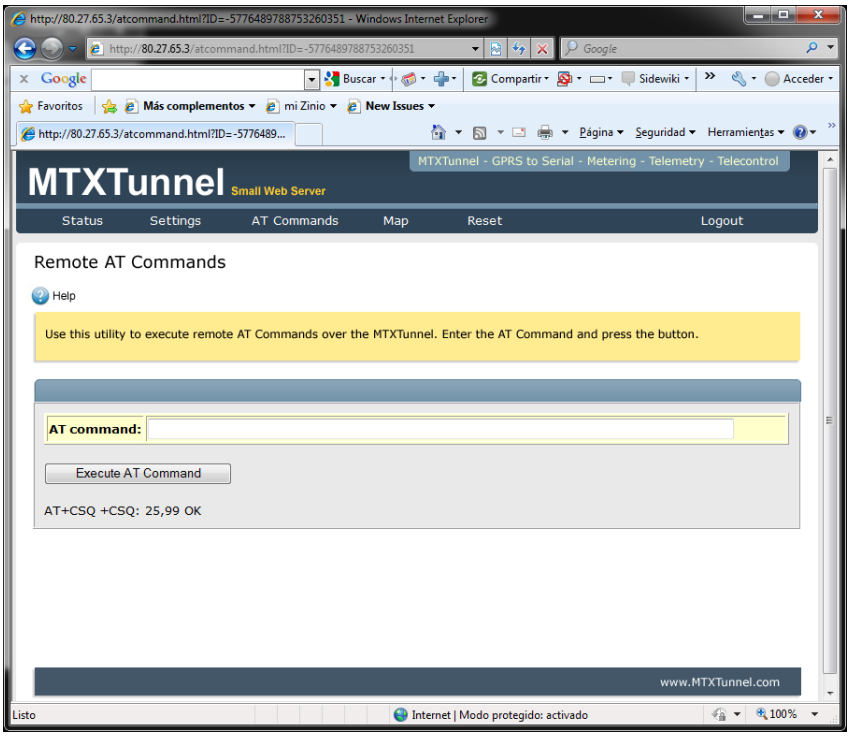
Please notice the next example. GPIO3 & GPIO4 are configured as output. In “value” label there is a link where you can remotely change the value just by clicking your mouse.



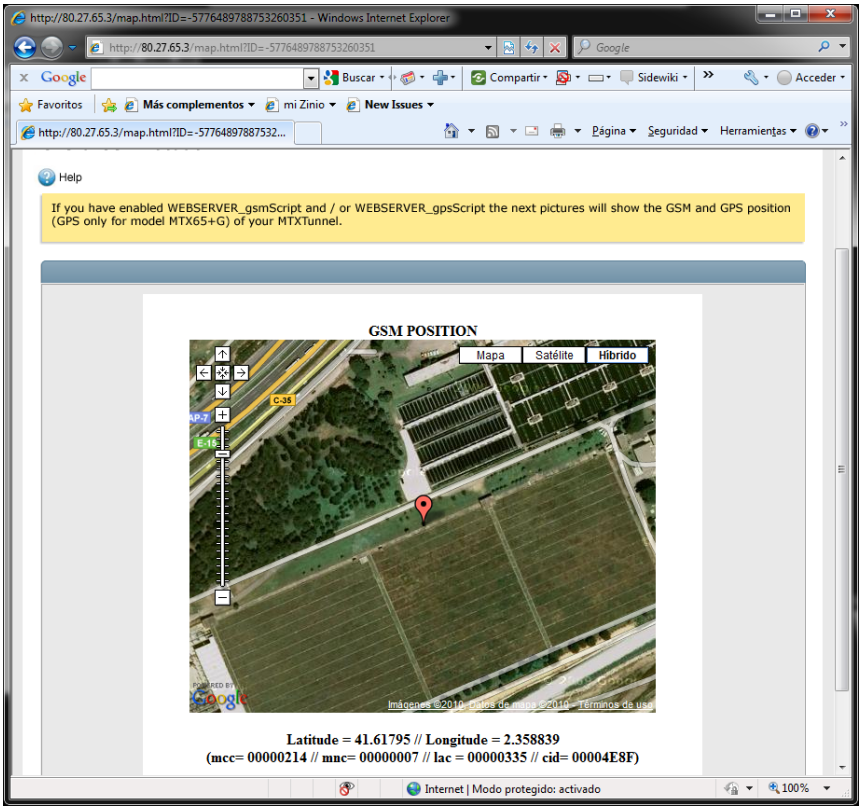
On the Setting menu label you can see and change any MTX-Tunnel configuration value. Any value must be done one by one because it is designed only for punctual changes. Click on Reset to reset the terminal modem and use the new configuration. It is recommended to use Telnet or MQTT/S instead of a Web Browser.



AT Commands page is intended for the remote execution of AT commands. As an example, AT+CSQ command value is shown below:



Maps. In this section you can see maps. MTX devices can use GPS or cell network positioning.



Reset menu will save configuration values and it resets MTX-Tunnel.

Logout will close the session.

There are 3 important configuration values in “Settings” related to the embedded MTX-Tunnel WebServer:

WEBSERVER_skin

WEBSERVER_gsmScript

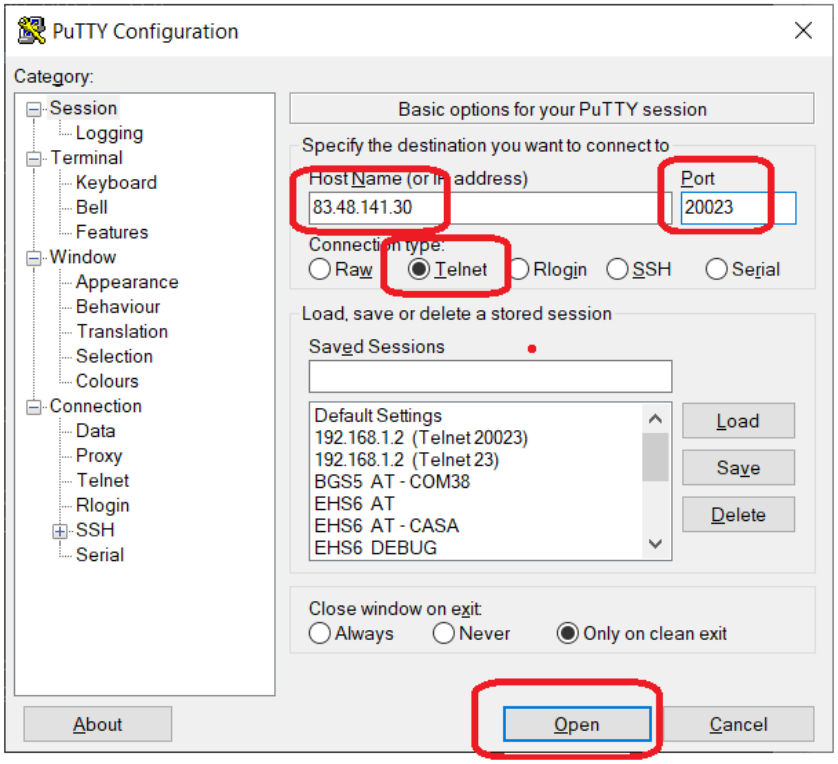
WEBSERVER_gpsScript

Read the description of configuration values in this section for more information.

6. Telnet Service

MTX-Tunnel features a small Telnet server. If enabled, you can remotely access MTX-Tunnel with any Telnet client. It is recommended to use Telnet to access MTX-Tunnel remotely; it is faster and effortless compared to WebServer.

To access MTX-Tunnel Telnet, open the MSDos console in your PC and specify the IP address obtained by MTX-Tunnel. DNS can be used instead if DynDNS has been activated. You can also use the freeware Putty to connect via Telnet.



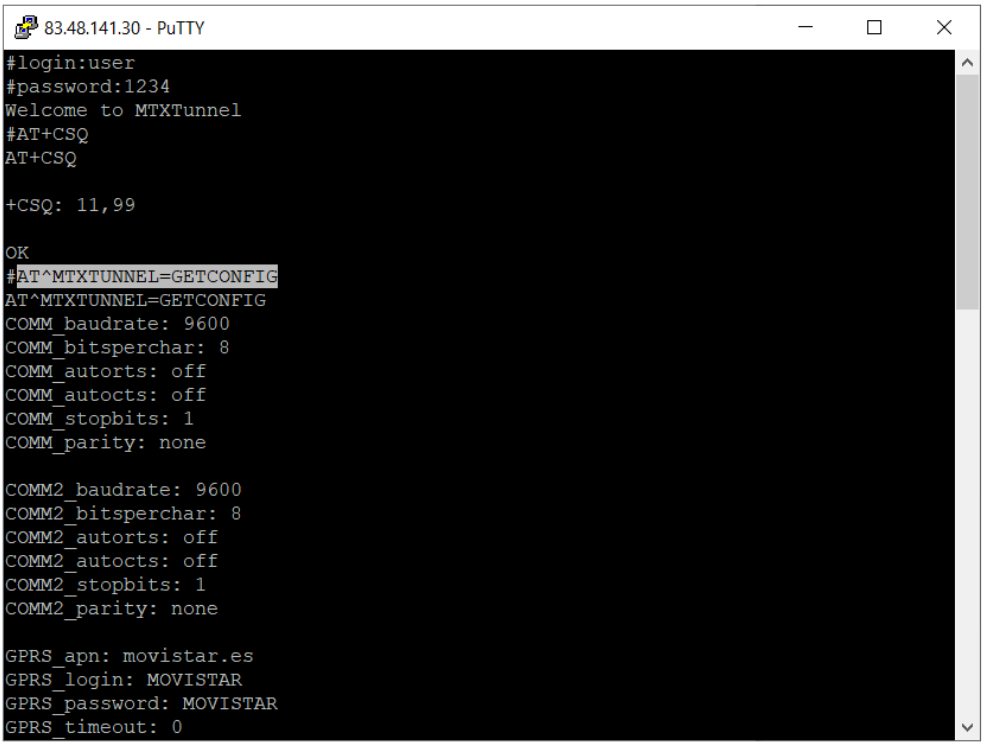
In the following example the default screen will be a LOG IN prompt.



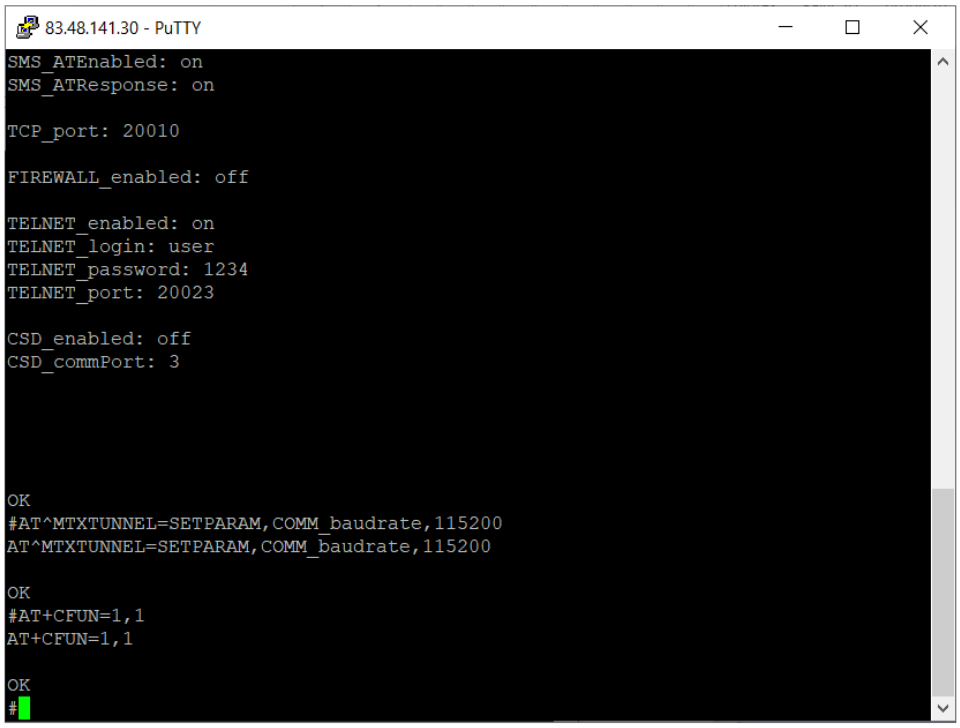
After the user and password are checked you can send AT commands to MTX-Tunnel. These commands can be modem standard ones or special ones, like to check network coverage with AT+CSQ command, etc.



Or to check the modem's configuration with the command AT^MTXTUNNEL=GETCONFIG



Or to change a remote configuration, like the speed of the serial port COM1 (specified in COMM_baudrate) to 115200 instead of 9600, for what we will use the command AT^MTXTUNNEL=SETPARAM,.... and the command AT+CFUN=1,1 for a remote reset.



There are many AT commands that can be executed remotely. Check the AT Commands section.

To use Telnet it is recommended to avoid the standard port 23 if you are using MTX-Tunnel on a public network (Internet). The reason is because of the dense traffic generated from other addresses to said port (to random IPs). It is recommended to use the port 20023 for Telnet, for instance.

Remember that the implemented Telnet has several posibilities, from the standard way (username and password) to OTP (One Time Password) that generates a different password for each session. Check the guide (TELNET_ parameters) for more information.

7. AT Commands. User API

In main applications, MTX-Tunnel is used as a simple GPRS-Serial gateway. But sometimes, MTX-Tunnel needs to be integrated into a third-party system application which needs to access the modem remotely to find out a digital input value, network coverage, to check incoming SMS or an important action like remotely or locally changing a configuration value without using OTAP or MES respectively.

User API consists of a set of special AT commands intended to integrate MTX-Tunnel in an end custom application. You can execute standard AT commands on MTX-Tunnel, such as the typical commands like “AT+CSQ” to check the network coverage, or “AT^SSIO” to change a digital output or commute a relay. However, MTX-Tunnel includes a series of special commands (API) which allows for the easy integration of MTX-Tunnel either in the system or that of a third party.

7.1 AT Commands MTX-Tunnel User Procedure

If you are not familiar with MTX-Tunnel, we recommend you do this test configuration to learn about the “MTX-Tunnel TCP-Server” scenario. Follow these steps:

Using main serial port COM1 DB9 (with encapsulation)

The AT command has a special syntax as the MTX-Tunnel application is running so tunnels can be created and the command is on the same channel.

Special syntax is used with AT commands between following TAGS <MTXTunnel> -</MTXTUNNEL>

For example, if you send this special command:

```
<MTXTUNNEL>AT</MTXTUNNEL>
```

You also will be receive the response by special syntax command:

```
<MTXTUNNEL>AT OK</MTXTUNNEL>
```

All these special syntax commands use the COM1 port and will be not resent by 4G/3G/2G as they are interpreted as an AT command by MTX-Tunnel. This way, the machine connected to COM1 can also have modem control.

To enable this feature the configuration parameter MTX_ATMux must be “on”. If you do not need this feature please do not enable the above command, you will save CPU resources.

NOTE: the encapsulated command should NOT end up in return (0x13 0x10). It must send exactly and only those characters so the command can be interpreted.

Using main serial port COM1 DB9 (without encapsulation)

It is possible to send the AT command without encapsulating using a special mode of the MTX-Tunnel, just for when the modem is used as a TCP Server or CSD (GSM call) gateway. In that situation it is possible to use the parameter MTX_ATMux: modem.

This way you will be able to send AT commands to the modem when there is not a TCP server connection established in that moment.

Using secondary serial port COM2

You do not need to use special syntax to send AT commands with COM2, just use normal syntax. Please be sure to configure the MTX_portAux parameter to “on”. If you do not need this feature please do not enable the above command, you will save CPU resources.

Please note that TELNET bypass (second GPRS-serial tunnel, to use 2 equipments) cannot be created if you need to use COM2 for auxiliary AT command use.

SMS messaging

You can send AT commands and receive their answer using SMS messages. Parameter configuration needed: SMS_ATEnabled “on” and SMS_ATResponse “on”.

No special syntax has to be used, MTX-Tunnel will use AT commands if the SMS message begins with “mtxtunnel” (or the specific SMS_header). For example, with a SMS with the text: “mtxtunnel at+csq” the modem executes the command and replies with another SMS with the signal level.

HTTP

It has been explained that Web Browser can be used to send AT commands. There is a more highly recommended way using “api.html” WebServer.

For example this way you read a digital input, change the level of the digital output and switch a relay or remotely read the network coverage as follows:

http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT%2BCSQ&LOGIN=user&PASS=1234

MTX-Tunnel will answer:

```
<MTXTUNNEL>AT+CSQ
+CSQ: 25, 99
OK
</MTXTUNNEL>
```

You need to extract the answer from the AT command between <MTXTunnel> tags.

Please note that URL syntax needs to be used and in this order: ATCOMMAND, LOGIN y PASSWORD.

Please also note that in URL special characters used on AT commands, like the “plus”, “+”, cannot be used and need to be decoded as follows:

%20	> blank space
+	> blank space
%21	> !
%22	> \
%23	> "
%24	> $
%25	> %
%26	> &
%27	> '
%28	> (
%29	>)
%2ª	> *
%2B	> +
%2C	> ,
%2D	> -
%2E	> .
%2F	> /
%3A	> :
%3B	> ;
%3C	> <
%3D	> =

%3E	> >
%3F	> ?
%40	> @
%5B	> [
%5C	> \
%5D	>]
%5E	> ^
%5F	> _

Telnet

In the previous chapter, we explained how to use TELNET to send AT commands. It is easy and convenient.

Using socket client connection

MTX-Tunnel is able to remotely send AT commands using the 4G/3G/2G -- Serial tunnel Gateway which is created when a TCP client socket is created. As explained before, this is useful when the network operator does not allow input connections or server mode.

To use this option please remember to configure MTX with parameter MTX_ATEmbedded to “on”.

To remotely send AT commands via a client socket you should encapsulate the command in between the following tags: <MTXTUNNELR> </ MTXTUNNELR>

For example, if you send the command:

```
<MTXTUNNELR> AT </ MTXTUNNELR>
```

Your response will be:

```
<MTXTUNNELR> AT OK </ MTXTUNNELR>
```

MQTT

As of MTX-Tunnelv9, it is possible to send remote AT commands through MQTT. It is very simple. Imagine a configuration like the following:

MQTT_enabled: on

MQTT_server: tcp://test.mosquitto.org:1883

MQTT_id: 12345678

MQTT_attopic1: AT1

MQTT_atrtopic: ATR

MQTT_qos: 0

MQTT_keepalive: 60

MQTT_persistent: off

To send remote AT commands to the modem via MQTT, all we have to do is send the AT commands to the topic specified in the MQTT_attopic1 parameter, since the modem subscribes to that topic after connecting to the broker. Any AT command received in that topic will be executed by the modem.

The response to the execution of the AT command, the modem sends it to the MQTT_atrtopic topic. That is, if you have a mobile phone with an MQTT client (for example the MyMQTT app) you can send AT commands to the modem very easily and see their answers.

Note that you have not only MQTT_attopic1, but also MQTT_attopic2 and MQTT_attopic3. Thought to be able to send AT commands simultaneously to modem groups. See these parameters in the manual for more information.

7.2 Standard AT Commands Supported by MTX-Tunnel

MTX-Tunnel runs on MTX-Terminal modems. Internally there is a Cinterion 4G/3G/2G core, EHS6 for 3G models and ELS61 for 4G models.

Those modules use the AT command set, some of them are standardized and others are Cinterion proprietary. MTX-Tunnel can run just a few of them. Those are:

- AT > OK
- ATI > Returns module firmware information
- AT+CSQ > Network GSM coverage
- AT+CCLK? > Current time output
- AT^SSIO=X,Y > Used to change the digital output X with the value Y
- AT^SGIO=X > Used to know the value of the digital input X
- AT^SRADC=X > Used to know the value of the analog input X
- AT^SGPSR=0 > Returns GPS position
- AT^MONI > Monitors information about BTS attached
- AT^SMONI > Monitors information about BTS attached
- AT+CFUN=1,1> Resets MTX-Tunnel
- AT^SMSO > Resets MTX-Tunnel (switches off and because MTX-Terminals is featured for it, it switches on automatically)

Please read the AT Command manual from Cinterion for more information about the use, syntax and answers of those commands. Please ask for a copy from your distributor or Matrix Electronica. Also check ftp.matrixelectronica.eu/MTX-Terminals for more information.

7.3 Special MTX-Tunnel AT Commands (API)

MTX-Tunnel needs special proprietary AT commands to allow MTX-Tunnel to be controlled by a third-party application, using a local serial port or 4G/3G/2G connection remotely. Please remember you can use one or two of the COM serial ports available. You can use them locally or you can use them remotely with IP (webserver, Telnet, MQTT, SNMP, gateway, etc.) and also SMS text messages. Please read carefully this AT special commands set features:

- AT^MTXTUNNEL=END

This command stops and ends MTX-Tunnel Java program execution. MTX-Terminal modem is now in normal mode.

This allows certain intelligent devices connected to MTX-Terminal to run or stop the MTX-Tunnel application and use the modem terminal for a voice call, CSD call, firmware upgrade... After this, MTX-Tunnel can be run again.

- AT^MTXTUNNEL=VERSION

String returned is version information. Only for MTX-Tunnel V7 and above.

- AT^MTXTUNNEL=SETBAUDRATE,port,speed

This command can change the modem serial port speed for a temporary amount of time, without the need for MTX-Tunnel to be reset.

EXAMPLE

```
AT^MTXTUNNEL=SETBAUDRATE, 0, 9600
```

ASC0 port baud rate speed is now 9600 bauds

```
AT^MTXTUNNEL=SETBAUDRATE, 1, 19200
```

ASC1 port speed is now 19200 bauds

Only for MTX-Tunnel V7 and above

- AT^MTXTUNNEL=GETPARAM,parameter_name

This command is intended to read or to find out a configuration parameter stored in a configuration file inside –non volatile memory-.

For example, to read APN value stored in configuration file, use:

```
AT^MTXTUNNEL=GETPARAM, GPRS_apn
```

- AT^MTXTUNNEL=GETCONFIG

This command is available for versions of MTX-Tunnel after v7.15. It returns the configuration file config. txt, avoiding the need for repeated use of AT^MTXTUNNEL=GETPARAM.

- AT^MTXTUNNEL= SETPARAM,parameter_name,ValueParametro

This command is used to change any MTX-Tunnel configuration parameter value.

For example, to change the baud rate of COM1 to 9600 bauds:

```
AT^MTXTUNNEL=SETPARAM, COMM_baudrate, 9600
```

Please be sure to reset the MTX-Tunnel application to get the new configuration.

The AT command to reset the terminal is AT+CFUN=1, 1

- AT^MTXTUNNEL= SETPARAM,parameter_name1:parameter_value1\r\nparameter_ name2:parameter_value2|r\n.....parameter_nameN:parameter_valueN}

This command is used to change several parameter values at the same time.

```
AT^MTXTUNNEL=SETPARAMS,{COMM_baudrate:9600\r\nCOMM_bitsperchar:8}
```

Please be sure to reset the MTX-Tunnel application to get the new configuration.

The AT command to reset the terminal is AT+CFUN=1, 1

- AT^MTXTUNNEL=GETIOS

This command is used to read at once the status of all the GPIOs (I / O value or pulse counters depending on how each GPIO is configured) and available ADCs of the modem. It is especially useful when sending AT commands by IP or by SMS, because this command is much faster than sending an AT command for each GPIO and ADC separately.

Starting with the MTX-Tunnelv11 version, the response format changes, returning the result in a JSON format. Example:

AT ^ MTXTUNNEL = GETIOS

{“IO0”: 0, “IO1”: 0, “IO2”: 0, “IO3”: 1, “IO4”: 1, “IO5”: 0, “AD0”: 3828 “ADM0”: “

voltage “,” AD1 “: 0,” ADM1 “:” voltage “}

OK

- AT^MTXTUNNEL=GETIO,numGPIO

This command returns the value of a digital input. NumGPIO indicates the GPIO number (1 to 10). Refer to GPIOs in the tables at the end of this manual for more information.

- AT^MTXTUNNEL=GETADC,numADC

This command returns the value of a particular analog input. NumADC indicates the number of ADC (1,2). Refer to ADCs in the tables at the end of this manual for more information.

- AT^MTXTUNNEL=RS232,mode,data

This command allows you to send and receive data directly from an MTX-Tunnel serial port.

This is useful for end applications using WEB page forms; you can get the form values from the web page, collect them in the serial port, and after using them, send a response to the web page.

You can see in Annex an example.

Mode values: 0, 1, 2 or 3.

0: data is forwarded through COM1 without waiting for a response from the device connected to COM1, so there is no value response to this AT command.

1: data is forwarded through COM1 and will wait for a response from the device connected to COM1. This response will be the return value of this AT command.

2: data is forwarded through COM2 without waiting for a response from the device connected to COM2, so there is no value response to this AT command.

3: data is forwarded through COM2 and will wait for a response from the device connected to COM2. This response will be the return value of this AT command.

NOTE

Mode value 1 and 3. Maximum device response is 160 characters long.

data parameter is a text string sent though serial port.

In data it is possible to specify hexadecimal (ie non-alphanumeric) values if they are included between the <HEX> </ HEX> tags. Example <HEX> 414243 </ HEX> will send the values 0x41 0x42 0x43 through the serial port.

- AT^MTXTUNNEL=SMS,phone_number,message

This proprietary AT command is intended to send SMS text messages.

Do not use the AT+CMGS command as it is intended to be used in end-party applications.

Parameters description:

“phone number”: End user’s phone number

“message” SMS text string

- AT^MTXTUNNEL=SETMODBUS,address;position1;data_1;data_2;...;data_n

When a MODBUS device is connected to secondary port on MTXTerminal, this proprietary AT command will allow some parameters stored in the memory table to change:

address: modbus device address (0 ... 255)

position1: first position to write in modbus memory table

data_1, data_2: data to be written in the Modbus on pointed position1 parameter

Important: address, position1 and data_n are separated by “;”

- AT^MTXTUNNEL=SETMODBUS2,address;position1;command;data_1;data_2;...;data_n

When a MODBUS device is connected to secondary port on MTX-Tunnel, this proprietary AT command will allow some parameters stored in the memory table to change:

address: modbus device address (0 ... 255)

position1: first position to write in modbus memory table

command: it indicates the command to be used. It can be the command 15 (coils) or 16 (registries)

data_1, data_2: data to be written in the Modbus on pointed position1 parameter

Important: address, position1 and data_n are separated by “;”. This command can use the modbus command 15 or 16.

- AT^MTXTUNNEL=GETMODBUS,address;position;numData;command

When a MODBUS device is connected to a secondary port on MTXTerminal, this proprietary AT command will allow you to read some parameters stored in the memory table:

address: modbus device address (0 ... 255)

position: first position to write in modbus memory table

numData: number of data to read from modbus table

command: you can use the modbus command 3 or 4

Important: address, position and numData are separated by ";"

- AT^MTXTUNNEL=GETMODBUSALL

In this guide you will find many examples of scenarios where MTX-Tunnel is configured to read periodically the registries of one or many modbus slave devices. For example, you could configure MTX-Tunnel to read X devices every hour. But if at some point you want to launch a reading of all the Modbus devices configured (for example to make a test without waiting for 1 hour until the event happens) you can execute this AT command. It will immediately launch the process of modbus reading and sending the data to the server.

- AT^MTXTUNNEL=TEMPORALCLIENT,IPaddress,TCPport,seconds

A new AT command that can create a temporal client TCP socket to a specific server IP address and port. Only use this when the MTX_mode parameter is "server" or "none".

Now you can send AT commands remotely, by simply using the special tags <MTXTUNNELR></MTXTUNNELR>

Parameters:

IPaddress: IP address (server) to be connected

TCPport: TCP to be connected

IMPORTANT. TEMPORAL client socket means it will be closed automatically if GPRS data is not sent or received within 60 seconds.

The "seconds" parameter is available for versions of MTX-Tunnel after v7.15. It allows you to specify the number of seconds during which the socket will be open. If the command is resent with the seconds parameter as "0" once the socket is established, the temporal socket is immediately closed, without waiting for it to finish.

- AT^MTXTUNNEL=DELETEBBDD

Use this command when you need to delete the datalogger internal memory on MTX-Tunnel.

After using it, the internal modem memory file "data.txt" will be deleted and the MTX-Terminal modem will be reset.

- AT^MTXTUNNEL=SETIO,X,Y

MTX terminals that use 3G technology do not allow the use of the command AT^SSIO which is used in GPRS terminals to change a digital input. Instead, AT^MTXTUNNEL=SETIO,X,Y is to be used, where X is the GPIO to be changed (0,...,9) and Y the value to be assigned (0, 1)

- AT^MTXTUNNEL=FTP, ftpServer, loginServerFTP, passwordServerFTP, directoriyFTP, remoteFilename, MTXFilename, NotificationURL

A FTP from a file saved on a remote server can be downloaded onto the MTX terminal, which can be used to change the configuration files “config.txt” and “operators.txt” or to download other times of files.

The server login and password must be inputted, along with the server directory where the file is located, its name in the server and the name to be used in the MTX device, and a URL that is notified when the download is carried out correctly.

EXAMPLE

```
at^mtxtunnel=FTP,ftp.mydomain.com,myUser,myPass,folder1/
folder2/,config.txt,config.txt,
```

- AT^MTXTUNNEL=DOWNLOAD,http://www.myDomain.es/myPath,myUsername, myPassword,myOriginFile,myDestinationPath,myDestinationFile,timeOut

It is possible to make an HTTP/HTTPS download of a file set on a web server in the MTX terminal. It is useful to change the configuration file “config.txt”, change the file “iperators.txt” or download other kind of file.

EXAMPLES

```
AT^MTXTUNNEL=DOWNLOAD,http://myDomain.com/myPath,myUser,myPass,config.
txt,,config.txt,30
```

```
AT^MTXTUNNEL=DOWNLOAD,http://myDomain.com/
myPath,myUser,myPass,ServerCertificate1.jar,security/certs/
servers/,ServerCertificate1.jar,30
```

Remember that if you change the configuration remotely, you must restart the MTX so it loads the new configuration (command AT+CFUN=1,1).

The time base for the timeOut parameter is seconds. This command will return OK or ERROR after downloading the file, not being able to send another command until the end of the same (download or timeout).

- AT^MTXTUNNEL=ADOWNLOAD,http://www.myDomain.es/myPath,myUsername, myPassword,myOriginFile,myDestinationPath,myDestinationFile,timeOut

It is possible to perform an asynchronous HTTP / HTTPS download of a file located on a web server in the MTX terminal. Useful to change the configuration file “config.txt”, change the file “operators.txt” or download another type of file.

EXAMPLES

```
AT^MTXTUNNEL=ADOWNLOAD,http://myDomain.com/
myPath,myUser,myPass,config.txt,,config.txt,30
```

```
AT^MTXTUNNEL=ADOWNLOAD,http://myDomain.com/
myPath,myUser,myPass,ServerCertificate1.jar,security/certs/
servers/,ServerCertificate1.jar,30
```

Remember that if you change the configuration remotely, you must restart the MTX for it to take the new configuration (command AT + CFUN = 1,1).

The time base for the timeOut parameter is seconds. This command will return OK or ERROR immediately while the file is downloaded in the background. Use the command AT ^ MTXTUNNEL = ISFILE to find out if the file has been downloaded.

- AT^MTXTUNNEL=ISFILE,path,fileName

It allows to find out if there is a file inside the modem. Useful to use in conjunction with the AT ^ MTXTUNNEL = ADOWNLOAD command.

EXAMPLES

```
AT^MTXTUNNEL=isfile,,config.txt
```

This command will return OK if the file exists or ERROR if it does not exist.

If the file is in the root directory, leave the path field blank.

- AT^MTXTUNNEL=SETCONFIGFILE,{ConfigRaw}

This command allows the entire content of MTX-Tunnel's configuration file to be established without having to change each parameter one-by-one. It is designed to be used only from a Web platform as a response to the sending of a JSON object. For example, when MTX sends a JSON object from its internal logger to a Web platform, it can receive this command as a response between the <MTXTUNNELR> and </MTXTUNNELR> to change the complete configuration.

MTX will restart with the new configuration once the AT command is processed. The configuration must be between the {} tags.

The parameters can be sent separated by the sign "\r\n".

EXAMPLE

```
{COMM_baudrate:9600\r\nCOMM_bitsperchar:8\r\n ……}
```

- AT^MTXTUNNEL=SETOUTPUTTIMER,numOutput,value

Allows to activate a digital or relay output for X seconds with a single AT command. The affected output must be previously configured with GPIO_modeX as “output” and GPIO_configX as “timer” (examples in chapter 8). numOutput indicates the ID of the digital output of the (0,1, ...) MTX. Value can take the values 0, 1.

- AT^MTXTUNNEL= GETPOWERSTATUS

Command to state whether a MTX modem is being powered by an external power supply or by an internal battery. Only for modems with internal battery. Possible responses:

-1 (ERROR), 0 (internal battery functioning), 1 (external power supply functioning).

- AT^MTXTUNNEL= GETCELLID

Returns the identificator of the telephone cell being used. Useful for gsm localization systems.

- AT^MTXTUNNEL= RESET,time

Resets the modem after the specific number of seconds indicated in “time” parameter (0, ... 86400).

- AT^MTXTUNNEL= getCounters

AT command not available from MTX-Tunnelv11. Use the command AT ^ MTXTUNNEL = GETIOS instead.

- AT^MTXTUNNEL= getCounter,numCounter

Returns the current value of the numCounter counter. The parameter “numCounter” indicates the ID of the GPIO configured as a pulse counter. The value returned by the command is a value between 0 and 4294967294. See Example 7.11 for more information.

- AT^MTXTUNNEL= setCounter,numCounter,value

Sets the current value of the numCounter counter. The parameter “numCounter” indicates the ID of the GPIO configured as a pulse counter. The “value” field can have a value between 0 and 4294967294.

- AT^MTXTUNNEL= setSchedule,ID:Day;Hour;Minute;ATCommand

Allows to configure an AT timing command for it to be executed on a certain day of the week (Monday, ... Sunday) at a certain hour and minute. For example, it may be of use if you want to switch a relay or a digital output at certain hours. Remember that the modem uses UTC time.

ID: 1 ... 200. Timing index. 200 timings maximum.

Day: 1...7. (1= Monday, 2=Tuesday, ... , 7 = Saturday)

Hour: 0...23

Minute: 0...59

ATCommand: execute AT Command

...

EXAMPLE

```
AT^MTXTUNNEL=setSchedule,1:2;22;0;AT^SSIO=0,1
```

(activates GPIO1 digital output every Tuesday at 22:00)

This command will write to a file named “schedule.txt” that is inside the modem. There is also the possibility of upload the file into the modem by using a FTP server and the command AT^MTXTUNNEL=FTP, ...

The above mentioned “schedule.txt” file has the following format:

1:day;hour;minute;ATcommand

2:day;hour;minute;ATcommand

3:day;hour;minute;ATcommand

...

If you want to delete a certain time setting, you can use “null” key in ATCommand field. For example, if you want to delete timing with ID=1:

```
AT^MTXTUNNEL=setSchedule,1:2;22;0;null
```

- AT^MTXTUNNEL=getSchedule,ID

Returns ID schedule of the modem, where ID = 1...200 indicates the schedule number.

- AT^MTXTUNNEL=getSchedules

Returns a list with all scheduled time settings.

- AT^MTXTUNNEL=delSchedules

Deletes all scheduled time settings.

- AT^MTXTUNNEL=getAstronomic,latitude,longitude,dd,mm,yyyy

Returns the Ortho and Sunset for a given latitude, longitude, day, month, and year. Example for the city of Madrid:

```
AT^MTXTUNNEL=getAstronomic,40.4893,-3.6827,12,03,2017
```

- AT^MTXTUNNEL=SETULPSECONDS,value

Command only for MTX models with ULP (Ultra Low Power). This command allows, once the modem is awake, to increase the time it stays awake or finish immediately. For example, you want to wake up the modem just 5 minutes every day to perform a task. The modem awakes and the task is completed in 2 minutes. Instead of waiting the remaining 3 minutes to enter ULP mode again, it can be shut down immediately specifying value 0, saving power.

value: 0 ... 86400 seconds

- AT^MTXTUNNEL=CERTIFICATE,LISTSERVERS

It lists user SSL Root CA certificates for SSL connections. It lists the files available in the folder “secutity/certs/servers,” that is up to 10 possible files ServerCertificate1.jar, ... ServerCertificate10.jar.

- AT^MTXTUNNEL=CERTIFICATE,INSTALLSERVERS

It installs all user SSL Root CA certificates for SSL connections. It installs the files available in the folder “secutity/certs/servers,” that is up to 10 possible files ServerCertificate1.jar, ... ServerCertificate10.jar.

- AT^MTXTUNNEL=CERTIFICATE,DELETESERVER,certificateFile

It eliminates the SSL Root CA certificate indicated. The parameter “certificateFile” indicates one of the 10 possible certificate files ServerCertificate1.jar, ... ServerCertificate10.jar that can be found in the directory “secutity/certs/servers” inside the modem. Once the command is executed the certificate is uninstalled and the file deleted from the directory.

- AT^MTXTUNNEL=CERTIFICATE,LISTCLIENTS

It lists client certificates for SSL connections (only needed if certificate client authentication is required from the server). It lists the files available in the folder “secutity/certs/servers,” that is up to 10 possible files ServerCertificate1.jar, ... ServerCertificate10.jar.

- AT^MTXTUNNEL=CERTIFICATE, INSTALLCLIENT,certificateFile

Unlike the certificates on the server, for which it is possible to install up to 10 at the same time, in the case of the client certificate, for obvious reasons it is only possible to install one certificate at a time. With this command it is specified the certificate to install.

- AT^MTXTUNNEL=CERTIFICATE,DELETECLIENT

It uninstalls the client certificate (but unlike the server certificate, it doesn’t delete any file from the directory “secutity/certs/servers”).

- AT^MTXTUNNEL=CERTIFICATE,DELETEFILECLIENT,certificateFile

It deletes the indicated certificate from the directory “secutity/certs/servers.”

- AT^MTXTUNNEL=setAstroSchedule,ID:Day;Month;SunriseHour;SunriseMinute; SunsetHour;SunsetMinute

Allows you to manually set the sunrise and sunset times for special days of the year (exceptions). In other words, if the astronomical clock is activated to act on a relay, it will switch based on the automatic calculations (sunrise / sunset) made by the modem, except for the times indicated by this command. This will allow you to set schedules for special days. See Example 8.7 for more information.

ID: 1 ... 100. Temporization index, 100 schedules max.

Day: 1...31

Month: 1...12

SunriseHour: 0...23

SunriseMinute: 0...59

SunriseHour: 0...23

SunriseMinute: 0...59

EXAMPLE

AT^MTXTUNNEL= setAstroSchedule,1:15;7;8;30;21;45

(for July 15 the ortho will be at 8:30 and the sunset at 21:45)

This command will write to a file named “astroschedule.txt” found inside the modem. The entire copy of said file from an FTP server is also allowed using the command AT ^ MTXTUNNEL = FTP,.... Ó AT ^ MTXTUNNEL = DOWNLOAD...

The format of this file “astroschedule.txt” is as follows:

1:day;month;SunriseHour;SunriseMinute;SunsetHour;SunSetMinute\r\n

2:day;month;SunriseHour;SunriseMinute;SunsetHour;SunSetMinute\r\n

3:day;month;SunriseHour;SunriseMinute;SunsetHour;SunSetMinute\r\n

...

- AT^MTXTUNNEL=getAstroSchedule,ID

Returns the special astronomical programming modem ID, where ID = 1 ... 100 indicates the exception number.

- AT^MTXTUNNEL=getAstroSchedules

Returns a list with all the exceptions of the astronomical clock.

- AT^MTXTUNNEL=delAstroSchedules

Removes all exceptions from the astronomical clock.

- AT^MTXTUNNEL=delAstroSchedule,ID

Removes all exceptions from the modem's astronomical clock ID, where ID = 1. 100 indicates the exception number.

- AT^MTXTUNNEL=DOWNLOAD,url,httpUsername,httpPassword,filename,modemPath, filenameDestination

Allows to download a file via http from a web server inside the modem.

EXAMPLES

AT^MTXTUNNEL=DOWNLOAD,http://www.miweb.com,,,config.txt,,config.txt

AT^MTXTUNNEL=DOWNLOAD,http://www.miweb.com/miPath,,,astroschedule.txt,, astroschedule.txt

AT^MTXTUNNEL=DOWNLOAD,http://www.miweb.com/mipath,miuser,miPassord, config.txt, security/,config.txt

- AT^MTXTUNNEL=IOEVENT

Starts a process of reading the modem's I / O (digital inputs and outputs, analog inputs and pulse counters) to store them in the internal datalogger for later sending to a Web server or MQTT broker, without waiting for them to be produce an event or read period.

- AT^MTXTUNNEL=GETIP

Returns the current IP address of the modem.

- AT^MTXTUNNEL=EXECUTE,file

Run an AT command batch file. Inside the flash memory of the modem is a folder named "atscripts". Batch files of AT commands can be incorporated into this folder to be executed using this command. In file the name of the file to be executed must be specified. See point 7.4 for more information.

- AT^MTXTUNNEL=FORCEDNS

The DNS_ configuration parameters allow the modem to be configured to periodically send the modem

status data (IP, coverage, GPIOs...). This command allows the immediate sending of the DNS frame without the need to end the configured period of time.

- AT^MTXTUNNEL=SETDAC,idDAC,valor

Command that allows changing the output value of a DAC for those modem models that have this interface.

idDAC: indica el identificador del DAC (0, ...)

valor: valor en milivoltios a aplicar en el DAC (0 ...10100)

- AT^MTXTUNNEL=GETDAC,idDAC

Command that allows reading the output value of a DAC of those modem models that have this interface. The returned value is in millivolts.

idDAC: indica el identificador del DAC (0, ...)

- AT^MTXTUNNEL=SETIOMAINTENANCE,idGPIO,mode

Command that allows configuring an output type GPIO in maintenance mode. This allows you to temporarily leave the configuration mode to go into manual mode and be able to be activated / deactivated with the command AT ^ MTXTUNNEL = SETIO.

For example, consider a GPIO output configured as an astronomical clock. The output will activate with sunset and deactivate with sunrise. Activating the maintenance mode the output will go into manual mode until the maintenance mode is exited or the MTX modem is restarted.

idGPIO: indicates the identifier of the GPIO output (0, ...)

mode: 0: normal working mode (the one configured in GPIO_config)

1: maintenance mode

- AT^MTXTUNNEL=SETWMBUSFILTERS,value

Command that allows activating or deactivating the configured manufacturer and device filters for reading W-MBUS sensors. Useful for commissioning of facilities where the MTX-Tunnel works as a W-MBus concentrator

value: 0: Filters disabled

1: Filters enabled

7.4 AT Commands Batch Files

Inside the flash memory of the modem is a folder called “atscripts”. Within this folder batch files of AT commands can be incorporated to be executed on certain occasions or by using the command AT ^ MTXTUNNEL = file.

For example, if a “miscommands.txt” file is found inside the “atscripts” folder with the content:

```
AT^MTXTUNNEL=SETIO,0,1
PAUSE 5
AT^MTXTUNNEL=SETIO,0,0
```

When you run the command AT ^ MTXTUNNEL = EXECUTE, mycommands.txt the modem will execute, in order, that list of commands (it will activate the output GPIO0, wait 5 seconds, and deactivate the output GPIO0.

Special Files

There are a series of special files for executing batch commands that will be automatically executed under certain circumstances, as detailed below. These files must be placed in the “atscripts” folder:

- “mtxtunnel_start.txt”

  After starting the modem, this batch file will be executed if it has content. Useful in case you need special actions to start the modem.

- “iologger_start.txt”

  This batch file is run just before LOGGER captures its digital / analog inputs. See Example 8.5 for more information.

- “iologger_end.txt”

  This batch file is executed just after LOGGER captures its digital / analog inputs. See Example 8.5 for more information.

- AT^MTXTUNNEL=SETIEC102, idMeter

Special command to perform a real-time reading of the instantaneous values (energy and power values) of a meter of the electric power distribution system. The protocol used follows the international standard IEC 870-5-102.

idMeter: indicates the identifier of the counter (ID001, ID002...)

- AT^MTXTUNNEL=GETIEC102, idMeter

Special command that returns in JSON format the last reading launched by the command “AT ^

MTXTUNNEL = SETIEC102, idMeter”.

idMeter: indicates the identifier of the counter (ID001, ID002...)

- AT^MTXTUNNEL=SETIEC102_CTAVM2, idMeter, horaIni, minutoIni, diaIni, mesIni, anoIni, horaFin, minutoFin, diaFin, mesFin, anoFin

Special command to perform a real-time reading of the integrated totals of a meter of the electric power distribution system. Said integrated totals correspond to the parameters referring to the memorized pricing information (Contract I) of the closing readings. The protocol used follows the international standard IEC 870-5-102.

idMeter: indicates the identifier of the counter (ID001, ID002, ...)

horaIni: initial hour (0... 23)

minutoIni: initial minute (0... 59)

diaIni: initial day (1... 31)

mesIni: initial month (1... 12)

anoIni: initial year (0... 99)

horaFin: final hour (0... 23)

minutoFin: final minute (0... 59)

diaFin: final day (1... 31)

mesFin: final month (1... 12)

anoIni: final year (0... 99)

- AT^MTXTUNNEL=GETIEC102_CTAVM2, idMeter

Special command that returns in JSON format the last reading launched by the command“AT^MTXTUNNEL=SETIEC102_CTAVM2, idMeter, horaIni, minutoIni, diaIni, mesIni, anoIni, horaFin, minutoFin, diaFin, mesFin, anoFin”.

idMeter: indicates the identifier of the counter (ID001, ID002...)

8. MTX-Tunnel Configuration Parameters

8.1 General Configuration Parameters "MTX_"

All general configuration parameter starts with "MTX_" prefix string.

MTX_PIN, MTX_PIN2

Description: PIN secure number of SIM inserted.

Possible values: 16 characters maximum

Default value: 0000

Additional notes:

- In case of using a SIM card with no PIN number, MTX_PIN security code does not need any value
- The MTX_PIN2 parameter is for exclusive use for models that have the "DUAL SIM" feature and refers to the Username used by the secondary SIM. In the case of the MTX-IOT-S family modems, the secondary SIM is the one inside the modem, accessible by opening the case

MTX_mode

Description: indicates the main mode of operation of the MTX-Tunnel software. This parameter will indicate whether the modem should behave as a TCP server socket (waiting for remote connections), as a TCP client socket (connecting the modem itself to a remote server) or as a UDP socket (to receive and send frames using the protocol UDP). It is also possible to use the mqtt value to implement an mqtt gateway - RS232

If it is not necessary to create any 4G/3G/2G-Series tunnel, the MTX_mode value must be set to “none”. For example, it can be useful when you only need an SMS alarm when changing a digital input.

Possible values: server, client, udp, mqtt, none

Default value: server

Additional notes:

- The reading of the special annex with the scenario examples is recommended for a better understanding of this parameter
- The use of “none” is useful for scenarios in which no 4G/3G/2G-Series tunnel is going to be used (see the example annexes)

MTX_urc

Description: MTX-Tunnel can output in COM1 (ASC0) the status of connections or working state of MTX-Tunnel as URC –Unsolicited Result Codes- messages.

URC messages can be:

```
^MTXTunnel_9.x_running
```

First message after powering up MTXTunnel. It means it is in running mode.

```
^MTX_IP_XXX.XXX.XXX.XXX
```

Message when MTXTunnel has got a new IP address in a GPRS connection.

```
^MTX_DTR_END_APPLICATION
```

This message is outputted when the MTX-Tunnel application has been stopped by a user request (special AT command or DTR serial line change level).

```
^MTX_CONNECTION_CLIENT_ESTABLISHED
```

Output message when MTX-Tunnel configured as client has connected successfully with the remote server.

```
^MTX_CONNECTION_CLIENT_END
```

Output message when a client configured MTX-Tunnel has closed the connection with the remote server because it has disconnected itself or because the socket has been closed remotely.

```
^MTX_CONNECTION_ESTABLISHED
```

Output message when MTX-Tunnel is server mode configured and accepts a remote socket connection...

```
^MTX_CONNECTION_END
```

Output message when MTX-Tunnel configured as server has closed the connection with the remote equipment because it has disconnected itself or because the socket has been closed remotely.

```
^MTX_SOCKET_UDP_ESTABLISHED
```

Output message when MTX-Tunnel configured as “udp” is ready to send and receive UDP data.

```
^MTX_SOCKET_UDP_END
```

URC message will be shown when MTX-Tunnel configured as “udp” closes the UDP socket due to a normal request (for example, time for GPRS connection has been expired).

```
^MTX_BITCOIN_INCOME_
```

This message is shown when MTX-Tunnel is configured to receive payments using bitcoin. See annex.

Possible values: on, off

Default value: off

Additional notes:

- We recommend you to not activate URC messages unless necessary. In a normal 4G/3G/2G-Serial RS232 tunnel Gateway these messages are in the same RS232 Serial port so there can be interference in the communication
- The first time when configuring and testing MTX-Tunnel it can be useful to see what MTX-Tunnel is doing or get information like the newly obtained IP address

MTX_reset

Description: time parameter (in minutes) so that MTX-Tunnel can be reset automatically.

A value of “0” indicates that the modem will never be automatically reset.

Possible values: 0... 43200 (43200 minutes=30 days)

Default value: 0

“0” disable this feature and MTX-TUNNEL will not be reset periodically.

Additional notes:

- It is not recommended to use this parameter unless you think it is necessary. MTX-Tunnel features many automatic procedures to ensure 4G/3G/2G connection will be stable and working 100% of the time.

MTX_resetHour

Description: this parameter can perform an automatic reset at a specific time X. Once reset, the MTX-Tunnel will restart automatically.

A value of “99” will cause the modem to never reset itself at any time.

Possible values: 0... 23-99

Default value: 99

Additional notes:

- “99” value disables this feature; modem will be never automatically reset
- It is not recommended to use this feature unless you think it is necessary; MTX-Tunnel has internal procedures allowing 4G/3G/2G connectivity to always be on...
- You need to use the MTX_TPServer parameter and use a timing server. The modem will synchronize internal time with server time TP or NTP (Network Time Protocol) based
- Modem time format is HOUR UTC
- If you enable this parameter, please use also MTX_reset parameter at 25 hours. This way reset will be performed and all services are restored even if timing synchronization fails

MTX_ping

Description: this is a very important configuration parameter to ensure GPRS connectivity. MTX-Tunnel will perform a PING to a configured IP address or DNS at configured periodic second timing.

If value is “0” PING will be never performed.

Possible values: 0... 1440 (1 day)

Default value: 30

Additional notes:

- We recommend using MTX_PING with a least 30 minutes value
- This parameter is more important if MTX-Tunnel is used in “server” mode. In this mode, MTX-Tunnel is waiting for incoming connections from remote equipments, and network operators can block the PPP connection without any notice. MTX-Tunnel cannot detect this block as there is no traffic so we use PING protocol to detect if the PPP connection is alive. PING traffic is almost insignificant and you could avoid the network operation PPP blocking when there is no traffic transmission

MTX_pingIP

Description: in the case of the above parameter MTX_ping > 0, i.e. when periodic ping is activated, this parameter value defines the PING IP address to be performed.

If you do not use this parameter, MTX-Tunnel will perform PING to its own IP address.

Be careful, some network operators do not allow performing PING to the newly obtained IP address, so we recommend you use a well known IP address. You can use your own server/office IP or the DNS Google one which is 8.8.8.8.

Possible values: xxx.xxx.xxx.xxx or you can use a URL like google.com

Default value: MTX-Tunnel obtained IP

Additional notes:

- We recommend that you use PING methods when using permanent connections

MTX_portAux

Description: most part of the MTX modem family have two COM serial ports, COM1 y COM2. If you activate this parameter with the value “on”, external equipment connected to COM2 could send AT commands.

Possible values: on, off, modbusmaster, gateway, bypass, wmbus

Default value: off

Additional notes:

- If you are not going to use this feature, disable this parameter with the value “off” as it will save internal CPU resources
- This parameter must be “off” when using MTX-Terminals that only have one COM1 port
- Please read the MTX_ATLimited parameter
- From MTX-Tunnel version 5.6, “wavenis” can be used as the parameter to control RF Wavenis protocol based devices connected to COM2
- From MTX-Tunnel version 7 “modbusmaster” can be used as the parameter to read RTU modbus devices connected to COM2. Read LOGGER_, MODBUS_ for more information
- From MTX-Tunel version 7.10 “gateway” can be used as the parameter to act as a gateway between the serial ports of the modem when there is no GPRS or GSM connection established. All data that enters the COM1 part is redirected to the COM2 port and vice versa
- From MTX-Tunnel version 9-20 it has a “bypass” functionality. This option is very similar to the option “gateway” but with preference for the serial gateway regarding the serial 3G gateway. You can find an example in the Annex 7.5
- As of version MTX-Tunnelv11.14 it has the option “wmabus”. This option configures the modem to be able to read wmbus devices, as long as the modem has an appropriate internal RF card

MTX_portAuxEcho

Description: enable MTX_portAux parameter “on” value when you need echo AT commands in COM2 port.

Possible values: on, off

Default value: on

Additional notes:

- This parameter is only valid when MTX_portAux is enabled (“on”). If not, this will not used

MTX_IDClient

Description: if configured in client mode, MTX-Tunnel sends an identification string when the connection with the server is done. This string is the first to be sent after the connection with the remote server.

This is intended to identify MTX-Tunnel with the server, and is useful when dynamic IP addressing is used.

Possible values: text string 255 characters max.

Default value: (empty, nothing is sent)

Additional notes:

- If you leave the value empty MTX-Tunnel will not send an identification string

MTX_IDClientExtended

Description: by enabling this parameter with the value “on” and using identification string with MTX_IDClient parameter, it is possible to send more information to a remote server.

When MTX_IDClientExtended is “on”, the extended string has the following format:

MTX_IDClient#IMEI#gpio1#gpio2# ... #gpio10#adc1#adc2#

MTX_IDClient is configuration string, IMEI is modem identifier, and gpioX is digital input/output and is analog input.

NOTE

With new models MTX-IoT [3-S-N-N] and MTX-IoT [3-S-N-N] the string will have the value of the pulse counter 1 and pulse counter 2

MTX_IDClient#IMEI#gpio20#gpio21#gpio22#gpio23#adc1#adc2#counter1# counter2#

If I/O are not needed and you set the parameter MTX_IDClientExtended “imei”:

MTX_IDClient#IMEI#

MTX_IDClient is configuration string, IMEI is modem identifier

Possible values: on, off, imei

Default value: off

Additional notes:

- If you leave the value empty MTX-Tunnel will not send an identification string

MTX_IDClientPeriod

Description: MTX_IDClient information string is only sent after the client connection to the remote server once after every new connection. With this parameter IDClient can send the information periodically every X seconds; for this, just use a value >0.

Possible values: 0 ... 2592000 (30 days)

Default value: 0 (only one string is sent at connection)

Additional notes:

- This can be useful if you need to remotely monitor the input/outputs and analog input because their statuses are sent periodically

MTX_dtr

Description: in some scenarios you may need to stop the MTX-Tunnel application. Then the modem would work with normal AT commands so you would be able to make a CSD call or voice call for example.

There are two ways to stop MTX-Tunnel:

- Send a proprietary AT command (AT^MTXTUNNEL=EXIT)
- Use modem DTR line on COM1 serial port. This feature must be enabled with the parameter value at “on”

Possible values: on, off

Default value: off

Additional notes:

- Once MTX-Tunnel is deactivated, it can be activated again restarting the modem

MTX_TPProtocol

Description: parameter available since MTX-Tunnel version v9.30. Previus versions can only use “tp” servers. It allows to choose the time sync protocol between “tp (Time Protocol)” and “ntp (Network Time Protocol).”

Possible values: Tp, ntp

Default value: tp (for compatibility reasons for previous versions)

Additional notes:

- Time servers give URG time, so when it is used by a modem it is also UTC time (please take into account what UTC time is your region on. For example, in Spain it’s UTC+1 or UTC+2 in the summer)
- It is compulsory to use a time server in case you use the internal datalogger (LOGGER_ parameters). For instance, if you want to use MTX-Tunnel to read modbus registries, you need to activate that time server, since in order to read modbus data the equipment needs to know the time to be able to save the data with the timestamp.

MTX_TPServer

Description: MTXTunnel can be timing synchronized using TP (Time protocol) or NTP (Network Time Protocol) since MTX-Tunnel v9.30. We can choose the protocol with the parameter MTX_TPProtocol. It connects to timing servers and fixes the RTC (Real Time Clock) deviation errors. Also it gets the time after power up.

It can be used on private or own time servers, but they are many free time servers and they can be used on MTX-Tunnel like this.

For example, if MTX_TPProtocol has “tp” value we can choose between these time servers:

time-a.timefreq.bldrdoc.gov
time-a.timefreq.bldrdoc.gov
time-b.timefreq.bldrdoc.gov
time-c.timefreq.bldrdoc.gov
utcnist.colorado.edu
time-nw.nist.gov
nist1.nyc.certifiedtime.com
nist1.dc.certifiedtime.com
nist1.sjc.certifiedtime.com
nist1.datum.com
ntp2.cmc.ec.gc.ca
ntps1-0.uni-erlangen.de
ntps1-1.uni-erlangen.de
ntps1-2.uni-erlangen.de
ntps1-0.cs.tu-berlin.de
time.ien.it
ptbtime1.ptb.de
ptbtime2.ptb.de > recommended as public free one
tp1.mtxm2m.com

If we select “ntp” in MTX_TPProtocol we can choose any NTP server (recommended). For example:
0.es.pool.ntp.org
1.es.pool.ntp.org

Possible values: Text string < 255 characters

Default value: none

Additional notes:

- Please note time server’s returns UTC time and there are a few hours difference in your country. As an example, in Spain the time is UTC+1 or UTC+2 in summer. UTC 09.00 time in July is 11.00 local time in Spain
- From MTX-Tunnel v7.15, this parameter can be specified as “null”. By doing this, the modem takes its current time as valid, without consulting an external server.This can be useful in situations where the WAKEUP_ parameter is used, whereby an activity must be carried out at certain intervals and the actual time is not really important

MTX_TPServer2

Description: backup timing server. If the previous main time server fails, MTX-Tunnel will take this second one as a security backup.

If MTX_TPProtocol is “tp” we can choose from these time servers:

time-b.timefreq.bldrdoc.gov
time-a.timefreq.bldrdoc.gov
time-b.timefreq.bldrdoc.gov
time-c.timefreq.bldrdoc.gov
utcnist.colorado.edu
time-nw.nist.gov
nist1.nyc.certifiedtime.com
nist1.dc.certifiedtime.com
nist1.sjc.certifiedtime.com
nist1.datum.com
ntp2.cmc.ec.gc.ca
ntps1-0.uni-erlangen.de
ntps1-1.uni-erlangen.de
ntps1-2.uni-erlangen.de
ntps1-0.cs.tu-berlin.de

time.ien.it
ptbtime1.ptb.de
ptbtime2.ptb.de
tp1.mtxm2m.com > recommended free public time server

If we select “ntp” in MTX_TPProtocol we can choose any NTP server. For example:
0.es.pool.ntp.org
1.es.pool.ntp.org

Possible values: Text string < 255 characters

Default value: None

Additional notes:

- You can use this backup timing server only if you have configured main in MTX_TPServer parameter

MTX_TPFormat

Description: allows you to specify the time format that the modem will use when sending data to a web server or mqtt that includes a timestamp.

Possible values: mtxtunnel, unix, epoch

Default value: mtxtunnel

Additional notes:

- In case of using the default format “mtxtunnel” the time format is as follows: dd/mm/yyyy HH/mm/ss
- In case of using the “unix” format, the time format is as follows: yyy-mm-dd HH:mm:ss
- In case of using the “epoch” format, the time format is as follows: 1583150549000 (milliseconds)
- Remember that the MTX modem always uses UTC time. If you need a time conversion, it should always be done on the server

MTX_ATMux

Description: this parameter, when enabled, activates the multiplexer on the COM1 serial port of MTX-Tunnel.

Multiplexer means it is possible to send AT commands to a modem when a 4G/3G/2G tunnel is active/connected. This way you can use AT commands to see network coverage/information, change or read a digital output/input, stop MTX-Tunnel or change a configuration parameter.

You need to write AT commands into special tags strings because MTX-Tunnel has to interpret it and not send it to the GPRS using following syntax:

<MTXTUNNEL> </MTXTUNNEL>

EXAMPLE

```
<MTXTUNNEL>AT+CSQ</MTXTUNNEL>
```

MTXTunnel will return:

```
<MTXTUNNEL>AT+CSQ +CSQ: 22, 99   OK</MTXTUNNEL>
```

Possible values: on, off

Default value: off

Additional notes:

- Special AT command must be sent with a pause of 1 second after last data sent on serial port but there can be no pauses of more than 50ms between characters
- Read chapter 7 to know more information about which AT commands are allowed in this multiplexer mode
- The option “modem” is available from the version 9.39 of the MTX-Tunnel. It allows sending AT commands to the modems without tags when the modem is being used as a TCP server gateway and/or CSD calls manager. It is useful for instance to configure the modem via AT commands without the need to load the configuration file “config.txt.”

MTX_WatchdogOnExit

Description: MTXTunnel features watchdog. If enabled, MTX-Tunnel must internally refresh the watchdog every 300 seconds (5 minutes). In case this refresh fails, the internal software watchdog will reset the MTX-Tunnel application.

If MTX-Tunnel is stopped (you can stop it using DTR or by AT commands), watchdog remains active, meaning that after 5 minutes, MTX-Tunnel will be reset and so, starts again.

If you disable this feature with an “off” value, watchdog will never reset MTX-Tunnel, even if it is ended.

Possible values: on, off

Default value: on

Additional notes:

- This is useful when the user needs to stop MTX-Tunnel temporarily to do certain classic modem communications (voice or data call, SMS….) and then if “on” ensure that MTX-Tunnel is running after 5 minutes

MTX_model

Description: specify the type of device on which the MTX-Tunnel application is running. Its use is completely mandatory and necessary. You must enter the Part Number.

Possible values:

MTX	PART NUMBER - MTX_model
MTX-T [2-N]	199801421
MTX-IoT [3-S-N-N]	199801393
MTX-IoT [3-S-N-N]-STD-G	199801415
MTX-IoT [3-S-N-GPS]	199801452
MTX-T [3-N]	199801422
MTX-T2 [3-N]	199801406
MTX-T [3-N]-G	199801423
MTX-IoT [4-S-N-N]	199801436
MTX-IoT [4-S-N-N] USA	199801439
MTX-IoT [4-S-N-N] AUS	199801446
MTX-T [4-N] (4G/2G)	199801424
MTX-T [4-N] (4G/3G/2G)	199801145
MTX-T [4-N] (USA-AT&T)	199801432
MTX-T [4-N] (AUS)	199801438
MTX-T [4-N]2	199801438
MTX-IoT [4-S-N-W868]	199801404
MTX-IoT [4-S-P-N]	199801437
MTX-IoT [4-S-R-N]	199801451

MTX-IoT [4-S-N-GPS]	199801452
MTX-4G-Java-IoT-STD-U	199801454
MTX-4G-Java-IoT-STD-UR	199801458
MTX-4G-Java-IoT-STD-UR-GPS	199801459
MTX-T [4-N]-S	199801464
MTX-IOT-S [4-S-N-N]	199802407
MTX-IOT-S [4-S-N-N] AUS	199802408
MTX-IOT-S [4-S-N-GPS]	199802409
MTX-IOT-S [4-S-N-GPS] AUS	199802410

Default value: none

Additional notes:

- It is mandatory to specify the correct MTX modem model on which the MTX-Tunnel application is running because each MTX modem model has a specific input/output configuration. An incorrect value of this parameter implies that the application may not work properly especially in relation to inputs/outputs (SMS alarms...)
- For xample, if you want to use the modem MTX-IoT [4-S-N-N] the value you must specify is MTX_model: 199801436
- The equipment leaves the factory with the MTX_model configured correctly. This table is only necessary for those cases in which it is required to copy the configuration used in one model over another

MTX_ATLimited

Description: this optional parameter can disable the limitation of AT command execution if the value is “off”. Remember you can use AT commands (multiplex on COM1, COM2, or via SMS or HTTP).

Possible values: on, off

Default value: on

Additional notes:

- We recommend that you set this parameter to “on”. Only use “off” when you need to use another command, but keep in mind that using AT commands without limitation could interfere with MTX-Tunnel behaviour. Please read AT command set of MTX-Terminals or ask iotsupport@mtxm2m.com for more information

MTX_clientSSL

Description: allows SSL secure socket communication (only client mode MTX_mode: client).

Remote server needs to support secure SSL socket connection.

Possible values: on, off

Default value: off

Additional notes:

- It is only possible to use an SSL security socket when MTX tunnel is used in client mode
- Do not use this if it is not necessary. Traffic data volume is increased and data communication speed will be slower
- The server must support any of following SSL standards
  - TLP protocol version 1.0 (RFC 2246)
  - SSL v3.0
  - WAP TLS Profile and Tunneling Specification

MTX_temporalClient

Description: this parameter allows you to establish a temporal client socket when MTX-Tunnel is in server mode (MTX_mode: server) and there is no connection established.

Example scenario: A series of MTX-Tunnel modems are available. Each MTX-Tunnel has a weather station on its COM1 serial port. All MTX-Tunnel are configured in “server” mode, since they are meant to establish a periodic connection from a central PC to collect the historical of temperatures of each meteorological station.

This allows to send critical alarm values without waiting for incoming server connections. After one minute (enough time to send the alarms), the socket is closed and MTX-Tunnel remains in normal server mode.

See scenario example in annex of this manual.

Possible values: on, off

Default value: off

Additional notes:

- This temporal tunnel connection takes just one minute, if there is no data traffic, the socket will be closed
- When the temporal socket is activated, the socket server services (server socket, WebServer....) are also activated. But if the temporary socket is closed after one minute, the associate services are closed after GPRS_timeout parameter
- The temporal client socket can be activated if the GPRS connection is always active (GPRS_timeout=0) or not (GPRS_timeout>0)
- If there is already a GPRS connection (socket connected to MTX-Tunnel) it is not possible to start the temporal client socket
- If the temporal client socket is running it is not possible to start server mode and any incoming connections will be not allowed
- It is MANDATORY that the MTX_ATMux parameter is disabled “off”, if not the temporal client socket connection will be not started
- From MTX-Tunnel v7 is it also possible to use a special AT remote commands to start the temporal client socket

MTX_temporalClientTimeout

Description: if a temporary TCP client socket is established, this parameter indicates how many seconds must pass without traffic on the socket (transmission or reception) for the temporary socket to close.

Possible values: 0... 3600 (seconds)

Default value: 60

Additional notes:

- Parameter available since version 10.18. In previous versions the value is always 60 seconds

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

MTX_msToSend

Description: a pause that indicates how many milliseconds must pass without receiving data through the serial port for the MTX-Tunnel to send data via GPRS.

Possible values: 0 ... 5000

Default value: 50

Additional notes:

- This is useful if the equipment connected to serial COM and MTX-Tunnel do not send data in concatenated way. Communication will be slower but all data is compacted

MTX_gatewayModBus

Description: this parameter will configure MTX-Tunnel as ModBus TCP/Modbus RTU tunnel gateway. MTX-Tunnel must be configured in server mode.

Possible values: on, off, comm, comm2

Default value: off

Additional notes:

- Remember MTX-Tunnel must be in server mode, waiting for TCP connection
- The comm, comm2 parameters are available from version 11.08 of the MTX-Tunnel and are useful when the modem has 2 IP-Serial gateways configured.

  An “on” value means that if the modem has two IP-Serial gateways configured, both act as modbus TCP/RTU gateways.

  A “comm” value makes only the main serial port (the one associated with COMM_ parameters) act as a modbus TCP/RTU gateway, with the secondary serial port (the one associated with COMM2_ parameters) acting as transparent gateway.

  A “comm2” value makes only the secondary serial port (the one associated with the COMM2_ parameters) act as a modbus TCP / RTU gateway, with the main serial port (the one associated with the COMM_ parameters) acting as a transparent gateway.

MTX_alwaysConnectedClient

Description: if MTX-Tunnel is configured in client mode (MTX_mode: Client), this parameter establishes a TCP socket connection once (value “off”) or in case the socket is closed, the connection is retried every 30 seconds (value “on”).

Possible values: on, off

Default value: on

Additional notes:

- “off” value is intended for the server to collect all data and close the socket. MTX-Tunnel will not retry in 30 seconds to open the socket, which will save resources in the server
- Parameter only valid if MTX-Tunnel is in client mode

MTX_init1, MTX_init2, MTX_init3, MTX_init4, MTX_init5

Description: these allow to specify up to 5 AT commands executed automatically each time MTX-Tunnel starts. As an example, one AT command could be sending an SMS when the modem is switched on.

Possible values: AT command text string

Default value: none

Additional notes:

- This can be used in many end applications and helps in a special start-up. Please check the AT commands manual or ask iotsupport@mtxm2m.com for further information

MTX_ATEmbedded

Description: this parameter allows the modem to interpret AT commands received by a client socket. That is, if this parameter is “on” from the server itself, AT commands can be sent to the MTX-Tunnel encapsulated between the tags: <MTXTUNNELR> and </MTXTUNNELR>. You can check the coverage, change settings... Very useful in the case of using socket type “client”.

Possible values: on, off, temporalclient

Default value: off

Additional notes:

- If you send an embedded AT command through a socket, you will also receive the response between the <MTXTUNNELR> </MTXTUNNELR> tags
- This mode of sending embedded AT commands allows you to bypass firewalls and proxies that many telephone operators use. If you cannot use Telnet to send remote AT commands to your MTX-Tunnel because your operator prevents it, use this route. Valid for both client, server and temporary client sockets
- The “temporaryclient” value option is only available from the MTX-Tunnel v10.18 version. If you set this value and the modem is set to (MTX_mode: server), only the <MTXTUNNELR> AT ^ MTXTUNNEL = DEFAULTTEMPORALCLIENT </MTXTUNNELR> command can be executed. Refer to the meterind example (meter reading) 7 of annex 6 for more information

MTX_radioBand

Description: this parameter can specify the preferred radio bands the modem will connect to. This is not really necessary in most cases, but in some countries in South America it is recommended to set this parameter.

Possible values: none, europe, america

Default value: none

Additional notes:

- If your modem is going to be used in Europe use “none” or “europe” value
- If your modem is going to be used in an American country, use “america” value
- Only available for MTX-65i modems family

MTX_invertedCom

Description: this parameter will invert COM values on MTX-Terminal modems with 2 serial COMS. As an example, MTX-IoT [3-S-N-N] has two RS232 serial ports, the main one with DB9 and the secondary one with DB15. If MTX_invertedCOM is enabled (value “on”) the secondary COM2 port now will act as COM1 and vice versa.

Possible values: on, off

Default value: off

Additional notes:

- This could be useful if you principally need the COM2 port of MTX-IoT [3-S-N-N]
- Please note that RS485 serial COM of MTX-IoT [3-S-N-N] modem is the secondary com. If you need to use it as primary (eg to attend a GSM call) must use MTX_invertedCom to “on”

MTX_flushSerialBuffers

Description: this parameter allows you to clean the serial buffers of any data to be sent before connecting to the TCP/IP socket. This means that if you have some outstanding serial data, it is removed by the modem's buffers before establishing the 4G/3G/2G-serial gateway.

Possible values: on, off

Default value: off

MTX_ATEmbeddedPass

Description: with the MTX_ATEmbeddedPass parameter set to “on” it is possible to send configuration AT commands in its own GPRS-serial gateway. With MTX_ATEmbeddedPass it is possible to set a password for embedded AT commands for enhanced security.

Possible values: String of up to 32 characters

Default value: none

Additional notes:

- If you set a password for the MTX_ATEmbeddedPass parameter, you will have to specify the password when you send an embedded AT command
- For example you need to send the command AT+CSQ. If you do not set a password you could send this <MTXTUNNELR>AT+CSQ</MTXTUNNELR>, but if you have set your password as XXX you will need to send <MTXTUNNELR XXX>AT+CSQ</MTXTUNNELR> which means you need to send: <MTXTUNNELR[space][password]> ATcommand</MTXTUNNELR>

MTX_clientReconnection

Description: this parameter is useful for configuration scenarios in which client connections are present (MTX_mode: client). In these scenarios this parameter specifies when the MTX-Tunnel will retry the connection after a shutdown by the remote server.

Possible values: 0 ... 86400 (seconds)

Default Value: 30

Additional notes:

- Note that if you set the value with a very low number (e.g. 0), MTX-Tunnel will retry the connection very quickly if there are constant problems or failures with the remote server and this will increase bandwidth

MTX_urcPort

Description: parameter available from MTX-Tunnel v7.15. Sets the output port of URC messages.

Possible values: asc0, asc1 y usb

Default value: asc0

Additional notes:

- The “asc0” value refers to main serial port (COMM_)
- The “asc1” value refers to the secondary serial port (COMM2_)
- The “usb” value refers to usb port

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

MTX_clientTimeout

Description: parameter available from version MTX-Tunnel v7.15. It allows you to specify the time, in seconds, that must be taken to close a client socket in the case of no 4G/3G/2G data exchange.

Possible values: 30... 86400

Default value: 1800 (30 minutes)

MTX_serverTimeout

Description: the parameter is available from MTX-Tunnel v9.18. It allows to specify the time in seconds. A server TCP Socket (for a 4G/3G/2G-serial gateway) will be closed in case there is no data transmission via 4G/3G/2G.

Possible values: 0... 86400

Default value: 0 (not activated)

Additional notes:

- This parameter is not necessary for the majority of scenarios. Its value can be 0 for most of them
- Where this parameter matters is in those scenarios where a serial port is used for two simultaneous tasks: autonomous reading of modbus registers + 4G/3G/2G-Serial gateway. That is to say, scenarios where the autonomous reading of modbus registers is suspended when a 4G/3G/2G-serial gateway is set up by the same serial port for a real-time action.

MTX_rssiLow

Description: parameter valid from version MTX-Tunnel v10.00. This parameter will start working if the parameter MTX_redLed or MTX_greenLed have selected the option “rssi.” If so, a coverage value under this range (MTX_rssiLow) will make the coverage LED to blink every 3 seconds indicating low signal level.

Possible values: 0... 31

Default value: 10

Additional notes:

- The values of a modem coverage level are standardized between 0 and 31, with 0 as the worst value and 31 as the greatest coverage
- The modem updates the status of its coverage LED every 10 seconds
- In case coverage is between the values MTX_rssiLow and MTX_rssiHigh, the coverage LED will indicate it with 2 blinks every 3 seconds
- For MTX-IOT-S modem models this parameter (MTX_rssiLow) indicates the value below which the yellow led, if configured to measure coverage (MTX_yellowLed: rssi) will flash 1 time. If the value is between MTX_rssiLow and MTX_rssiHigh it will flash 2 times and above MTX_rssiHigh it will flash 3 times

MTX_rssiHigh

Description: parameter valid from version MTX-Tunnel v10.00. This parameter will start working if the parameter MTX_redLed or MTX_greenLed have selected the option “rssi.” If so, a coverage value over this range (MTX_rssiLow) will make the coverage LED to blink 3 times every 3 seconds indicating high signal level.

Possible values: 0... 31

Default value: 20

Additional notes:

- The values of a modem coverage level are standardized between 0 and 31, with 0 as the worst value and 31 as the greatest coverage
- The modem updates the status of its coverage LED every 10 seconds
- In case coverage is between the values MTX_rssiLow and MTX_rssiHigh, the coverage LED will indicate it with 2 blinks every 3 seconds
- For MTX-IOT-S modem models this parameter (MTX_rssiHigh) indicates the value above which the yellow led, if configured to measure coverage (MTX_yellowLed: rssi) will flash 3 times. If the value is between MTX_rssiLow and MTX_rssiHigh it will blink 2 times and below MTX_rssiLow it will blink 1 time

MTX_greenLed

Description: parameter valid from version MTX-Tunnel v10.00. This parameter determins the behavior of the equipment green LED. Configured as “std” the behavior of the green LED is the standard Gemalto chipset (slow blink when the modem isn’t registered on the network, quick blink when it is). Configured like “rssi” the LED will blink every 3 seconds when the coverage level is low (coverage<MTX_lowRssi), 2 blinks when the coverage level is normal (MTX_rssiLow<=coverage<MTX_rssiHigh) and 3 blinks when the coverage level is high (coverage>= MTX_rssiHigh).

Possible values: std, rssi

Default value: std

Additional notes:

- Changing this parameter implies the equipment autoreset. That is, if you change the “std” value to “rssi” or viceversa, next time the modem starts it will autoreset once
- The modem updates the status of its coverage LED every 10 seconds
- The most interesting configuration is MTX_greenLed: rssi, MTX_blueLed: io2, MTX_redLed: sim
- This parameter is not available for MTX-IOT-S modems. Instead use the MTX_yellowLed parameter

MTX_blueLed

Description: parameter valid from version MTX-Tunnel v10.00. This parameter determins the behavior of the equipment blue LED. Configured as “off” the blue LED won’t light up. Configured as “ip” the blue LED will light continually while the modem has IP (connection to a 4G/3G/2G data network). Configured as “ip2” the blue LED will blink every 3 seconds while the modem has IP (connection to a 4G/3G/2G data network).

Possible values: off, ip, ip2

Default value: ip

Additional notes:

- If you use “ip2” you can also use MTX_greenLed ·rssi” since the blinking of the blue light happens moments after the coverage blinking, to ease the visualization
- The modem updates the status of its coverage LED every 10 seconds
- The most interesting configuration is MTX_greenLed: rssi, MTX_blueLed: io2, MTX_redLed: sim

MTX_redLed

Description: parameter valid from version MTX-Tunnel v10.00. This parameter determins the behavior of the equipment red LED. Configured as “off” the red LED won’t light up. Configured as “rssi” the red LED will behave like the MTX_greenLed when the value is “rssi.” Configured as “sim” the red LED will light up when: sim not inserted, incorrect sim pin, blocked sim (puk necessary).

Possible values: off, rssi, sim

Default value: off

Additional notes:

- Using the value “sim” is interesting because it allows to detect and resolve some connectivity problems quickly, detecting that the problem is in the sim
- The modem updates the status of its coverage LED every 10 seconds
- The most interesting configuration is MTX_greenLed: rssi, MTX_blueLed: io2, MTX_redLed: sim

MTX_yellowLed

Description: Parameter valid from version MTX-Tunnel v11.07 and valid only for MTX-IOT-S models.

This parameter determines the behavior of the yellow led of the equipment. Configured as “std” the behavior of the yellow led is the standard of the Gemalto chipset (slow flash when the modem is not registered on the network, fast flash when it is). Configured as “rssi”, the led will blink 1 every 3 seconds when the coverage level is low (coverage <MTX_lowRssi), 2 blinks when the coverage level is medium (MTX_rssiLow <= coverage <MTX_rssiHigh) and 3 blinks when the level coverage is high (coverage> = MTX_rssiHigh)

Possible values: std, rssi

Default value: rssi

Additional notes:

- Changing this parameter implies an Autoreset of the team. That is, if you change the value from “std” to “rssi” (or vice versa), the next time the modem is started, it will auto-reset once
- The modem updates the status of its coverage led every 10 seconds
- This parameter is available for MTX-IOT-S modems

MTX_fullDuplex

Description: parameter valid from version MTX-Tunnel v7.19. It allows you to improve the full-duplex capacity of the 4G/3G/2G-Serie gateways. It is especially designed for applications NOT based in question-answer comunications but with independent transmissions/receptions.

Possible values: on, off

Default value: off (disabled)

Additional notes:

- It is recommended to set it to “on” in the case of applications with independent asynchronous two-way communications. If you have a question/answer application (typical question from a server to whom the slave replies) do not activate this parameter
- Activating this parameter will slightly improve the asynchronous two-ways communications but will penalize with time other services (Telnet...)
- If you are not sure about if activating or not this parameter, we advice you not to include it in the config.txt setting file

MTX_filter

Description: parameter valid from v7.20. It allows you to use a filter in the 4G/3G/2G-Serie gateways (both in TCP server mode, as in TCP client and UDP modes). The use of a filter implies that, of the data frames received by the modem serial port, only those with a certain header will be sent via 4G/3G/2G.

Possible values: x,x,x,x... (bytes in header separated by a comma ",")

Default value: none (no headers used)

Additional notes:

- The header must be especified with bytes (decimal, no hexadecimal) separated by ","
- For example, if you only want to send the frames starting with "ABC", the MTX_filter parameter in the setting file should be: MTX_filter: 65,66,67 as A corresponds to ASCII 65, B to 66, C to 67
- Another example: if the MTX-Tunnel is connected to a red modbus and you are only interested in transmitting via GPRS the data frames to the MODBUS device with address 1, the MTX_filter parameter in the configuration file should be: MTX_filter: 1
- If you do not need to use filters, simply do not include this parameter in the configuration file
- Be aware you must take into account the MTX_msToSend parameter to use this parameter

MTX_latitude

Description: parameter valid from version MTX-Tunnel v7.27. Specifies the latitude (relative to the GPS position, in decimal format) where the MTX-Tunnel is installed. This parameter is necessary when using the MTX-Tunnel astronomical clock, for example to switch a relay or for a digital output automatically at sunset/sunrise time.

Possible values: -90.00000 to 90.00000

Default value: none

Additional notes:

- See scenario of Annex 8.4 for a better understanding of this parameter

MTX_longitude

Description: parameter valid from version MTX-Tunnel v7.27. Specifies the longitude (relative to the GPS position, in decimal format) where the MTX-Tunnel is installed. This parameter is necessary when using the MTX-Tunnel astronomical clock, for example to switch a relay or for a digital output automatically at sunset/sunrise time.

Possible values: -180.00000 to 180.00000

Default value: none

MTX_configMode

Description: parameter valid from version MTX-Tunnel v7.27. It allows you to choose if the “config” or “running” mode of the MTX-Tunnel is with or without the SIM inserted. That is, in “normal” mode (default mode) the MTX-Tunnel enters setting mode when the MTX is fed without a SIM card inserted and enters into “running” mode when it is fed with a SIM card inserted. In “reverse” mode the MTX enters “config” mode with a SIM card inserted and goes into“running” mode when there is no SIM card.

Possible values: -normal, reverse

Default value: normal

Additional notes:

- Do not use the reverse mode if you are not sure what this parameter is for. It is only possible to use it in very specific scenarios
- Use reverse mode only if you need the MTX-Tunnel for Logger without data delivery via 4G/3G/2G. That is, for example, to store the modbus records of a device during certain time in the modem internal memory. After this time, the modem is picked up and the “data.txt” file with the records stored is manually extracted
- From MTX-Tunnel 9.39 on it is possible to use the value “modem” with the parameter MTX_configMode. That allows, also using MTX_ATMux in “modem” mode, to send AT commands also in configuration mode “when the modem doesn’t have a SIM card.” In other words, you will be able to configure via AT commands (AT^MTXTUNNEL=SETPARAM, etc.) without the need to load the file config.txt.

MTX_interface

Description: parameter valid from version MTX-Tunnel v8.04. It allows you to choose the interface of communication between serial or USB.

Possible values: serial, usb

Default value: serial

Additional notes:

- Parameter only available for 3G models. It can't be used with GPRS models

V11.19 subject to changes | Webdyn © by Flexitron Group

MTX_encryptedConfig

Description: this parameter allows to encrypt the configuration file “config.txt”. If this parameter is set to “1”, the file “config.txt” will be encrypter after modem is power up. The “config.txt” file is different for each modem, so you can’t use the same encrypted “config.txt” file for any modem. If you need to use this option, it is very convenient to save the “config.txt” previously.

Possible values: 0, 1

Default value: 0

Additional notes:

- This parameter is only supported from MTX-Tunnel v9.39

MTX_mes

Description: it allows to activate/deactivate the modem MES. That is, it allows to activate/deactivate the access to the internal memory of the modem. For example, it can prevent non authorized access to the configuration file "config.txt." Remember you can also use the parameter MTX_encryptedConfig to encrypt the configuration. This MTX_mes parameter is a higher security level, that allows to block any access to the modem memory locally (USB, RS232).

Possible values: on, off

Default value: on (MES activated)

Additional notes:

- This parameter is only supported from MTX-Tunnel v10.04
- Be very careful with this command. If you set it off, after restarting the modem, the MES access will be blocked. The only way to unlock the access to the memory again will be sending the following commands:

```
AT^MTXTUNNEL=SETPARAM,MTX_mes,on
AT+CFUN=1,1
```

Then, have the precaution to test your access to the modem via Telnet, MQTT, SMS, etc. to the modem before blocking the memory

MTX_resetCond

Description: this parameter allows that, in the case the modem is configured like a serial IP gateway (TCP server mode), and it gets the daily autoreset condition in the MTX_reset or MTX_resetHour parameter, the reset won't happen when there's a socket established against the modem and MTX_resetCond is in "socket" mode.

Possible values: off, socket

Default value: off

Additional notes:

- This parameter can be very useful if the modem is configured to autoreset daily and is being used to make a serial IP gateway in server mode. Configuring this parameter like "socket" will avoid the modem resets while it's being used as a gateway. The reset will perform when the socket finishes

MTX_status

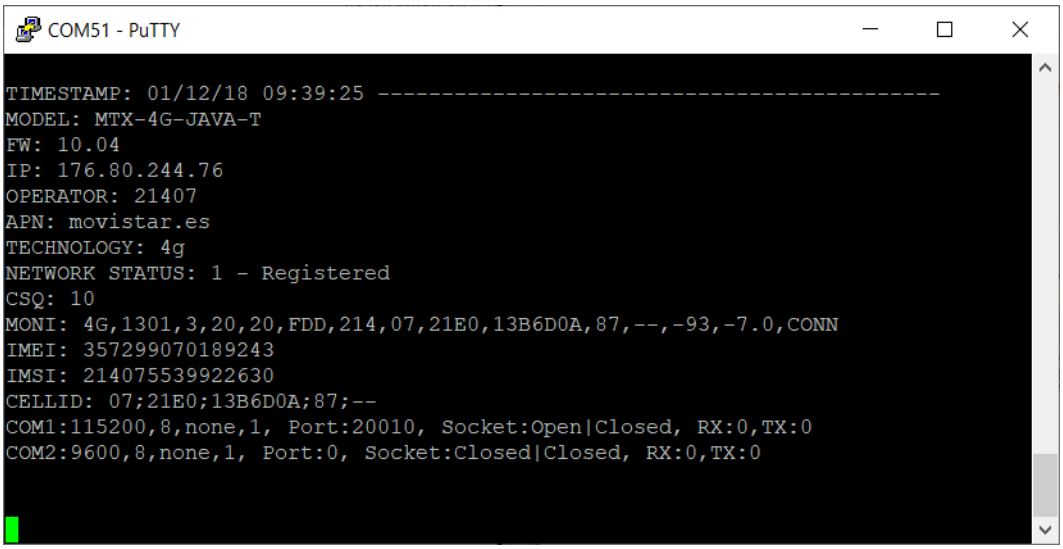
Description: this parameter allows, in the case of being activated (on), to extract certain status information via one of the USB ports created by the MTX modem in Windows (the port COM USB associated with the modem).

Possible values: on, off

Default value: off

Additional notes:

- This parameter can be useful to see the general status of the modem. It allows to see general aspects like the IP address obtained, the functioning network, the APN, the coverage, aspects of the BTS used, etc.



MTX_numGSMErrors

Description: this parameter allows, when there are multiple registry problems in the network, to autoreset the modem after X periods of registry tries.

Possible values: 0 ... 10000 (10 second blocks)

Default value: 0 (deactivated)

Additional notes:

- MTX-Tunnel tries to register in the network and checks that registry every 10 seconds. If the established value for that parameter is > 0, after re-trying, the modem will autoreset. For example, if you specify a value of 90, if the MTX modem can't register in the network in 90x10=900 seconds (15 minutes) the modem will autoreset. Although generally this parameter isn't necessary, it can be in some conflictive areas, where there are BTS problems normally. A value of 180 is recommended for security reasons.

MTX_defaultPrefix

Description: this parameter allows to assign a prefix to an entering call without a prefix.

Possible values: Up to 8 characters

Default value: none (deactivated)

Additional notes:

- For instance, if you call a modem located in Spain from a Spanish phone number located in Spain, the telephone number received by the modem won't have the +34 prefix. If you configure the entering phone number as an authorized number (SMS_validPhone1), you will need to configure the parameter MTX_defaultPrefix to "+34" so that prefix can be added to the incoming number

MTX_saveOutputState

Description: this parameter allows you to indicate whether the status of the outputs should be stored in the non-volatile memory of the modem. This will make, after a reset (or power recovery) the state of the outputs return to the state before the reset. In case of not storing the status of the outputs in non-volatile memory, the initial status of the outputs will be “deactivated”.

Possible values: on, off

Default value: on (activated)

MTX_api232Resp

Description: This parameter is useful with the AT^MTXTUNNEL = RS232,... command. It allows you to specify whether the response collected by said AT command should be returned in ASCII or HEX format.

For example, if the command AT^MTXTUNNEL = RS232, ... were used to send a command by SMS to the MTX modem which would be forwarded through the serial port to a device, and this would generate a response, said response would be forwarded literally when this parameter is in “ascii” mode or in hexadecimal when in “hex” mode. This is very useful when devices are outputting unrepresentable binary data in “ascii” format.

Possible values: ascii, hex

Default value: ascii

8.2 Configuration Parameters Related with COM1: "COMM"

Configuration parameters starting with the "COMM_" prefix references the main serial port on the MTX-Terminal modem. Check in the user guide which is the device's main port.

COMM_baudrate

Description: serial COM1 port speed baud rate.

Values (bps): 460800, 230400, 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200, 600, 300

Default value: 115200

Additional notes:

- You must use the same baud rate configuration with your device and MTX-Tunnel

COMM_autorts

Description: this parameter configures the COM1 serial port to have hardware flow control (RTS line). If enabled (value="on") then hardware flow control will be used, otherwise (value="off") means no hardware flow control will be used.

Possible values: on, off

Default value: on

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- Use the same value in COMM_autocts parameter

COMM_autocts

Description: this parameter configures the COM1 serial port to have hardware flow control (CTS line). If enabled (value="on") then, hardware flow control will be used, otherwise (value="off"), no hardware flow control will be used.

Possible values: on, off

Default value: on

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- Use the same value in COMM_autorts parameter

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

COMM_bitsperchar

Description: COM1 serial port number of bits per character.

Possible values: 7, 8

Default value: 8

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same.

COMM_stopbits

Description: number of stop bits per character in COM1 serial port.

Possible values: 1

Default value: 1

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- From version 7.27, 2 bits of data can be specified. 2 bit data communications can be slower, especially with large volume data communications. 1 stop bit is recommended

COMM_parity

Description: parity used in COM1 serial port communication.

Possible values: none, odd, even

Default value: none

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same

8.3 Configuration Parameters Related with COM2: "COMM2_"

Configuration parameters starting with the "COMM2_" prefix references the secondary serial port on MTX-Terminal modem. On the device it is the DB15 connector. Check in the user guide which is the secondary port.

This port is normally used to send AT commands to the terminal modem, so you can request network coverage or another MTX-Tunnel parameter configuration, etc.

This port can also be used to create another GPRS-serial tunnel gateway so you can control 2 devices with one terminal modem and one MTX-Tunnel application. See the Annex to see an example.

COMM2_baudrate

Description: serial COM2 port speed baud rate.

Possible values: 460800, 230400, 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200, 600, 300

Default value: 115200

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be same

COMM2_autorts

Description: this parameter configures the COM2 serial port to have hardware flow control (RTS line). If enabled (value="on") hardware flow control will be used; otherwise (value="off"), no hardware flow control will be used.

Possible values: on, off

Default value: off

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- Use the same value in the COMM_autorts parameter
- This parameter can be used only in the MTX-Java-IoT and MTX-Java-T2 terminal modems. Other MTX-Terminal modems do not have flow control in the second serial port

COMM2_autocts

Description: this parameter configures the COM2 serial port to have hardware flow control (CTS line). If enabled (value="on") hardware flow control will be used; otherwise (value="off"), no hardware flow control will be used.

Possible values: on, off

Default value: off

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- Use the same value in the COMM_autorts parameter
- This parameter can be used only in the MTX-Java-IoT and MTX-Java-T2 terminal modems. Other MTX-Terminal modems do not have flow control in the second serial port

COMM2_bitsperchar

Description: COM2 serial port number of bits per character.

Possible values: 7, 8

Default value: 8

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- This parameter can be used only in the MTX-Java-IoT and MTX-Java-T2 terminal modems. Other MTX-Terminal modems do not have flow control in the second serial port

COMM2_stopbits

Description: number of stop bits per character in COM2 serial port.

Possible values: 1

Default value: 1

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- This parameter can only be used in the MTX-IND and MTX-IND-V1 platforms. The rest of the models do not have flow control in the secondary port. From version 7.27, 2 bits of data can be specified. 2 bit data communications can be slower than those of 1 stop bit, especially with large volume data communications. 1 stop bit is recommended when possible

COMM2_parity

Description: parity used in COM2 serial port data communication.

Possible values: none, odd, even

Default value: none

Additional notes:

- Configuration of the device connected to the modem and MTX-Tunnel must be the same
- This parameter can be used only in the MTX-Java-IoT and MTX-Java-T2 terminal modems. Other MTX-Terminal modems do not have flow control in the second serial port

8.4 Telnet Configuration Parameters: "TELNET_"

Telnet configuration parameters start with a "TELNET" prefix. Telnet is recommended to remotely send AT commands, to change some configuration parameters or to find out the status of the remote modem with MTX-Tunnel. It is important to understand that Telnet will only work if the GPRS link has been established previously.

You can see TELNET in a MSDOS console or with specific software.

TELNET_enabled

Description: this command enables Telnet services in MTX-Tunnel.

Possible values: on, off

Default value: off

Additional notes:

- MTX-Tunnel only supports 2 socket servers simultaneously. This means that Telnet cannot be enabled if MTX-Tunnel is working in server mode (MTX_mode: server) and WebServer is active (WEBSERVER_enabled: on)

TELNET_login

Description: you can secure Telnet login with this command.

Possible values: Text string maximum 32 characters

Default value: (none)

Additional notes:

- If you write the login and password string, MTX-Tunnel will ask for credentials in the remote TELNET connection. If you do not use this feature, an anonymous connection will be created and there will be access without restriction after connection
- It is recommended to use login and password access security if TELNET_firewall is disabled ("off" value)

TELNET_password

Description: this command is used to specify a Telnet login password.

Possible values: Text string maximum 32 characters

Default value: (none)

Additional notes:

- If you enable login & password parameters, those who want to gain access will be asked for it when a remote Telnet connection is established. If you disable this feature (no value in login and password), you will have direct access to Telnet after connection
- It is recommended to use (enable) the login & password parameters if the TELNET_firewall parameter is “off”

TELNET_loginGuest

Description: it specifies the username in order to access the MTX-Tunnel telnet server as a guest.

Possible values: Text string maximum 32 characters

Default value: (none)

Additional notes:

- Under the Guest user it won't be possible to use commands like AT^MTXTUNNEL= so it won't be possible to read or change the MTX-Tunnel configuration. This user can execute supervision commands like AT+CSQ to check the coverage, AT+CFUN=1,1 to execute a remote reset, etc
- This user won't be available if we don't specify the username and password

TELNET_passwordGuest

Description: it specifies the password to be able to access the MTX-Tunnel telnet server as a guest.

Possible values: Text string maximum 32 characters

Default value: (none)

Additional notes:

- Under the Guest user it won't be possible to use commands like AT^MTXTUNNEL= so it won't be possible to read or change the MTX-Tunnel configuration. This user can execute supervision commands like AT+CSQ to check the coverage, AT+CFUN=1,1 to execute a remote reset, etc.
- This user won't be available if we don't specify the username and password

TELNET_firewall

Description: this command enables Telnet firewall.

If firewall is enabled it will only accept remote connections from specific authorized IP addresses -see FIREWALL IP1 command-.

If disabled, any IP address can remotely connect to MTX-Tunnel using Telnet.

Possible values: on, off

Default value: on

Additional notes:

- If firewall is disabled (value= “off”), it is recommended to use a login and password to avoid unauthorized access

TELNET_port

Description: TCP port used in MTX-Tunnel Telnet server.

Possible values: 1... 65535

Default value: 23

Additional notes:

- Do not use the same TCP port in command TCP_port and/or WEBSERVER_port
- If you are planning to use SIM card with public IP, TCP port 23 is not recommended. It is better to use another port like 20023. The reason is to avoid unnencessary data traffic

TELNET_bypass

Description: this parameter is only allowed for 2G models (MTX-65i family). All 3G models and MTX-IoT [3-S-N-N] can implement 2 gateway 3G/2G – Serial with parameters TCP_port and TCP_port2. Parameter NOT recommended. Only used for compatibility reasons.

This parameter enables the use of TELNET as a bypass connection to control the device connected to the MTX-Tunnel modem's COM serial port.

In other words, this can allow 2 simultaneous Serial-GPRS tunnels and control two connected pieces of equipment, one to COM1 and other to COM2.

Possible values: on, off

Default value: off

Additional notes:

- If enabled TELNET_login and TELNET_password parameters will be used

TELNET_instances

Description: this parameter allows you to have more than one simultaneous active Telnet sessions. Firmware versions prior to the version 9.12 do not allow it. You may dispose of up to two simultaneous active Telnet sessions.

Possible values: 1, 2

Default value: 1

Additional notes:

- It is not recommended to use this parameter unless it is absolutely necessary for the application. The reason is not to subtract memory and CPU resources from MTX-Tunnel unnecessarily

TELNET_auth

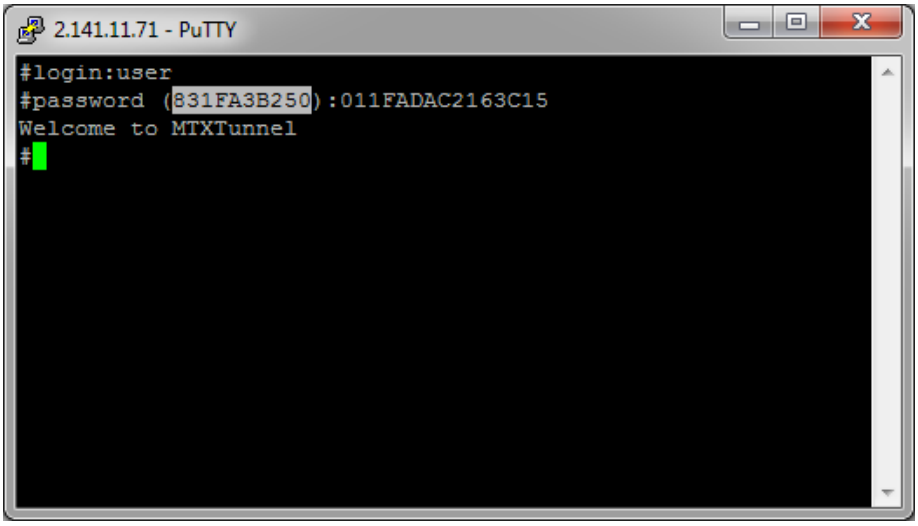
Description: this parameter allows you to increase security during the authentication phase of a Telnet session. When using the options “otp” (One Time Password) and “otpsms” (One Time Password SMS) you can enjoy a higher level of security compared to “std” (standard) provided by Telnet. The “otp” and “otpsms” options are available from MTX-Tunnel v9.20 onwards.

Possible values: std otp, otpsms

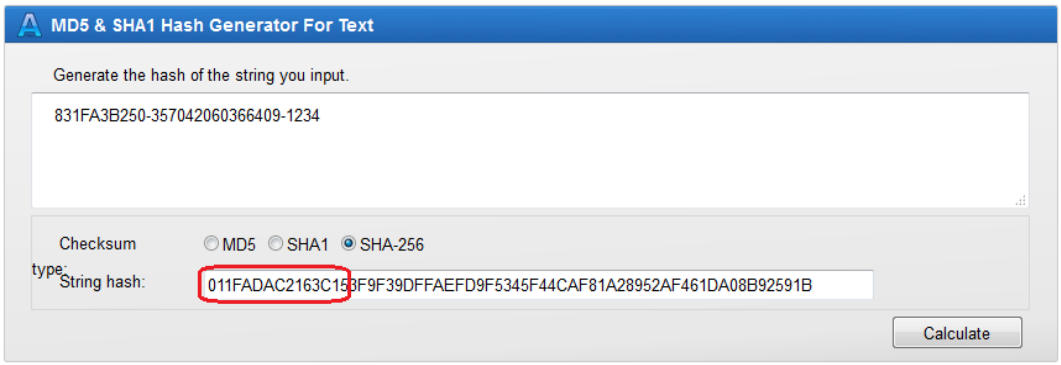
Default value: std

Additional notes:

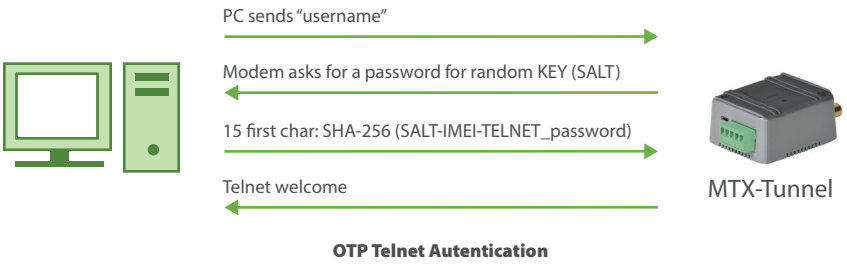
- When choosing the “std”option, you run the standard authentication process of a Telnet session, where the username (TELNET_login parameter) and a password (TELNET_password parameter) to be authenticated are sent to the remote modem
- When choosing the “otp” option,the password used in a remote modem is always variable, which makes it much less prone to be intercepted. The password can be used only once. The authentication process is as follows. After you have entered a username, the modem will require the password for a certain code, always randomly generated, indicated in brackets



- As a response to this password the first 15 HASH characters generated by SHA-256 and corresponding to random-IMEI-Telnet_password should be returned. That means, as it is shown in the example, that if the returned random password is 831FA3B250, the IMEI of the modem is 357042060366409, and if the Telnet Password is 1234, the HASH to be returned as a password is 011FADAC2163C15, just as it can be seen in the following example:



- This way the Password is never compromised, because the password is never sent through the Internet



- Finally, there is "otpsms" option. The process also consists of receiving an otp password (One Time Password), but it is requested via SMS. If you send a message with the text "mtxtunnel at^mtxtunnel=otp" (or its corresponding alias), you will receive a SMS message with a one-time login password. To use this method, it is recommended to use authorized phone numbers, for example:

  SMS_allPhones: off
  SMS_validPhone1: +34666123456
  ...
  This way only authorized phone numbers will be able to request their one-time password for Telnet.

- Remember that you can activate firewall (FIREWALL_enabled: on) to be able to access Telnet only from authorized IPs (FIREWALL_IPx)

- Finally, if you want, your can keep Telnet service inactive and activate it any moment via SMS messages sent from authorized phone numbers

8.5 WebServer Configuration Parameters: "WEBSERVER_"

WebServer configuration parameter starts with "WEBSERVER_". MTX-Tunnel features a small WebServer which you can easily access with Internet Explorer software and check any modem's state, network information, input state, or you could change a digital output, etc.

WEBSERVER_enabled

Description: specifies if the MTX-Tunnel webserver should be activated or not. Through the webserver you can visualized easily the modem digital and analog I/O status and to change the status of the digital outputs (or relays in case of a modem with relay). You can also visualize and modify the configuration of the equipment and send remote AT commands (for example to read the coverage remotely).

An API is available to integrate MTX-Tunnel WebServer in your own application for the following features:

- Create a HTTP-RS232 serial tunnel Gateway. The http web page can be a form. The data filled is sent using MTX-Tunnel to the serial com port of the attached equipment and vice versa
- Read, write or change any MTX-Tunnel configuration parameter in your own application. Your end customer will not realize that MTX-Tunnel is running so your application is completely customized
- The http web page can be programmed in a few minutes in order to show the input/output MTX-Tunnel values for example

Possible values: on, off

Default value: off

Additional notes:

- Please read WebServer example scenarios for more information

WEBSERVER_login

Description: you can secure WebServer login with this command.

Possible values: Text string maximum 32 characters

Default value: (none)

Additional notes:

- If you enable the login & password parameters, those who try to gain access will be asked for them when they connect to the web page. If this feature is disabled (no value in login and password), direct access to the web page will be available
- It is recommended to use (enable) the login & password parameters if the WEBSERVER_firewall parameter is “off”

WEBSERVER_password

Description: parameter used to specify password in order to access MTX-Tunnel WebServer.

Possible values: Text string maximum 32 characters

Default value: (none)

Additional notes:

- If you enable login & password parameters, those who try to gain access will be asked for them when they connect to the web page. If this feature is disabled (no value in login and password), direct access to the web page will be available
- It is recommended to use (enable) the login & password parameters if the WEBSERVER_firewall parameter is “off”

WEBSERVER_firewall

Description: this command enables the WebServer firewall.

If the firewall is enabled it will only accept remote connections from specific authorized IP address -see FIREWALL_IP1 command-

If disabled, any IP address can remotely connect to the MTX-Tunnel WebServer.

Possible values: on, off

Default value: on

Additional notes:

- If firewall is disabled (value= “off”), it is recommended to use the login (WEBSERVER_login) and password (WEBSERVER_password) parameters to avoid unauthorized access

WEBSERVER_port

Description: TCP port used in web connections MTX-Tunnel WebServer.

Possible values: 1... 65535

Default value: 80

Additional notes:

- Do not use the same TCP port value in the command TCP_port parameter and/or TELNET_port

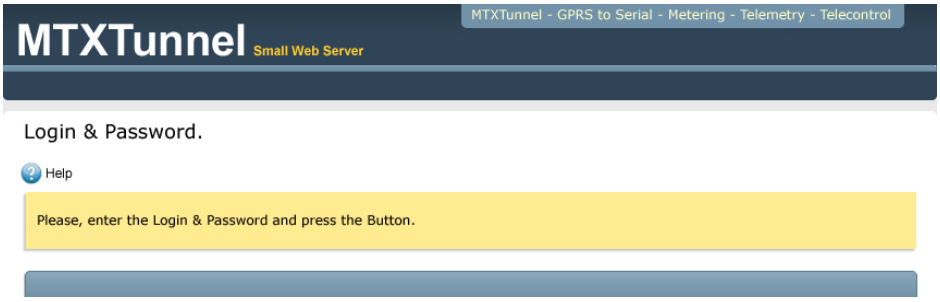
WEBSERVER_skin

Description:

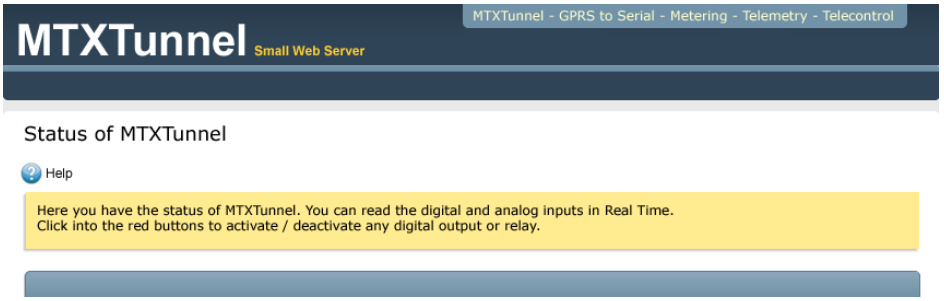
This indicates the URL link where you can find the MTX-Tunnel WebServer SKIN. You can customize the look of the MTX-Tunnel http web page with your company logo for example.

The URL must contain all the following images:

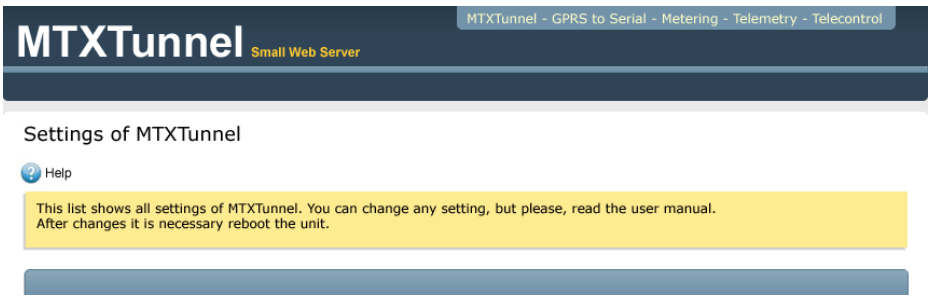
- “header0.gif” (858x268). Header in Login welcome page



- “header1.gif” (858x268). Header in “Status” section page



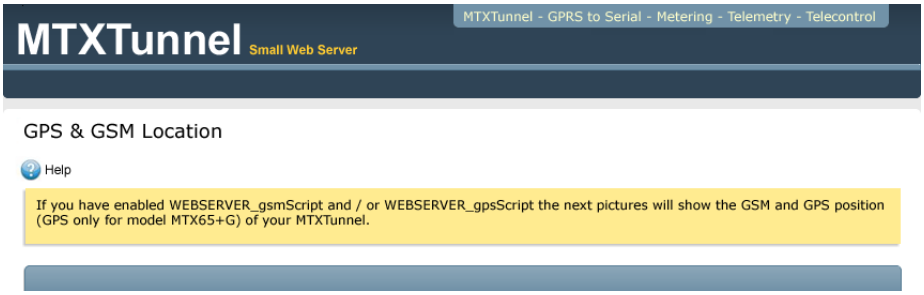
- “header2.gif” (858x268). Header in “Settings” page



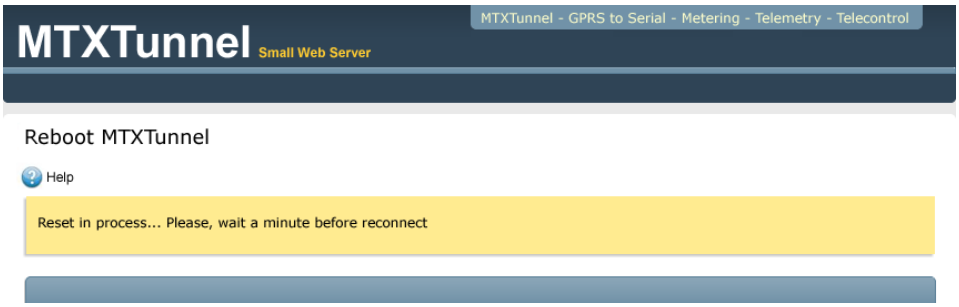
- “header3.gif” (858x268). Header in “AT Commands” page



- “header4.gif” (858x268). Header in “Map” page



- “header5.gif” (858x268). Header in “Reboot” page section



- “footer.gif” (858x55). WebServer page footer.



- “espera.gif” (32x32). Animated Gif showed when loading pages.



- “onButton.gif” (35x42). Small button indicating output level high (activated relay in MTX-IND modem terminal)



- “offButton.gif” (35x42). Small button indicating output level low (deactivated relay in MTX-IND modem terminal)



Values: URL maximum 100 character string

Default value: none

Additional notes:

- URL link must finish with “/”
- As an example, a URL not leading to your own Web server can be: http://www.mtxtunnel.com/webserverimg/

8.6 Wakeup Configuration Parameter: "WAKEUP_"

The previous MTX-Tunnel V5 version could only be used for 4G/3G/2G-serial tunnel Gateway in the following ways:

- 100% active after modem power-up
- X minutes active upon demand, after receiving an SMS with special text string "mtxtunnel on" or when receiving a missed call from an authorized configured phone number

As of version 5.0, the 4G/3G/2G session can be activated by time programming.

WAKEUP_timeEnabled

Description: 4G/3G/2G connections (Serial-GPRS tunnel, WebServer, Telnet services) can be based on time programming. You can specify a total of 10 different timing values.

Possible values: on, off

Default value: off

Additional notes:

- When the time condition is fulfilled, the 4G/3G/2G link is established and remains connected for the period specified in GPRS_timeout, except for when the 4G/3G/2G or serial data traffic is active or any other wakeup condition is active
- It is mandatory to use a time server (for internal RTC Synchronization). Please check the MTX_TPServer parameter for more information
- Internal modem timing is UTC (Universal Time Clock). Check the time difference in your country and check if the season affects the hour too. Spain is UTC+1 except in summer when it is UTC+2, so for example 09.00 UTC in July is 11.00 local time

WAKEUP_time1, WAKEUP_time2, WAKEUP_time10

Description: these parameters are used to declare the 10 different alarm triggers to activate 4G/3G/2G-Tunnel and all associated services: serial tunnel, WebServer, Telnet...

Possible values: DDHHMM (DD day, HH hour in 24h format and MM minutes)

Default value: none

Additional notes:

- DD and HH can take a value of XX. This means no day/time will be used in the timer
- Example 1:

  WAKEUP_time1: 012200

  (4G/3G/2G connection will be activated every 01 (1st) day of the month, at 22.00 h.)
- Example 2:

  WAKEUP_time1: 011030

  WAKEUP_time2: 151030

  (4G/3G/2G connection will be activated every 1st and 15th day of month, both at 10:30 h)
- Example 3:

  WAKEUP_time1: XX1800

  (4G/3G/2G connection will be activated every day of the month, at 18:00 h)
- Example 4:

  WAKEUP_time1: XX0200

  WAKEUP_time2: XX1400

  (4G/3G/2G connection will be activated every day of the month at 02:00 and at 14:00 h)
- Example 5:

  WAKEUP_time1: XXXX00

  (4G/3G/2G connection will be activated every day of the month, every hour on the hour)
- Modem timing format is UTC

8.7 GPRS connection configuration parameters: "GPRS_"

The following parameters are mandatory. They make reference to the MTX-Tunnel connection parameters related to the network operator. It depends on the SIM card and provisioned 4G/3G/2G service. Ask your network operator for further information.

GPRS_apn, GPRS_apn2

Description: 4G/3G/2G Access Point Name –APN-

Possible values: <100 characters text string

Default value: movistar.es

Additional notes:

- Some APN examples :

Movistar Spain (dynamic IP addressing):	movistar.es
Movistar Spain (fixed IP address):	internerestatico.movistar.es
Vodafone Spain (dynamic IP addressing):	airtelnet.es
Vodafone Spain (fixed IP address):	ipfija.vodafone.es
Orange Spain (dynamic IP address):	internet

- Not all network operators are recommended for MTX-Tunnel applications
- Some mobile network operators like Orange or other virtual network operators like Simyo or Yoigo use proxies and block incoming data traffic at TCP ports. Therefore MTX-Tunnel cannot be used in server mode (MTX_mode: server) and only in client mode, so MTX-Tunnel would be waiting for incoming connections
- We have found that other operators use fixed (or also dynamic IP) addressing in a private range. This means that you have lots of security but you have to create a VPN to allow communication
- Please do not hesitate to contact MTXterminals support line iotsupport@mtxm2m.com for further information
- The GPRS_apn2 parameter is for the exclusive use of models that have the "DUAL SIM" feature and refers to the APN used by the secondary SIM. In the case of the MTX-IOT-S family modems, the secondary SIM is the one inside the modem, accessible by opening the case

GPRS_login, GPRS_login2

Description: 4G/3G/2G connection LOGIN access parameter value. Your network operator will provide you with this value.

Possible values: Text string < 32 characters

Default value: Movistar

Additional notes:

- Login examples

Movistar Spain (dynamic IP addressing):	Movistar
Movistar Spain (fixed IP address):	Movistar
Vodafone Spain (dynamic IP addressing):	Vodafone
Vodafone Spain (fixed IP address):	Vodafone
Orange Spain (dynamic IP address):	Cliente

- Not all network operators are recommended for MTX-Tunnel applications.

  Some mobile network operators use proxies and block incoming data traffic at TCP ports, meaning MTX-Tunnel can only be used in server mode (MTX_mode: server). Contact your distributor for further information.

- The GPRS_login2 parameter is for exclusive use for models that have the “DUAL SIM” feature and refers to the Username used by the secondary SIM. In the case of MTX-IOT-S family modems, the secondary SIM is the one located inside the modem, accessible by opening its casing

GPRS_password, GPRS_password2

Description: PASSWORD for 4G/3G/2G setting. Ask your network operator to provide you with this value.

Possible values: Text string < 32 characters

Default value: Movistar

Additional notes:

- Some “password” value examples:

  Movistar Spain (dynamic IP addressing): Movistar

  Movistar Spain (fixed IP address): Movistar

  Vodafone Spain (dynamic IP addressing): Vodafone

  Vodafone Spain (fixed IP address): Vodafone

  Orange Spain (dynamic IP address): Amena

- Not all network operators are recommended for MTX-Tunnel applications.

  Some mobile network operators use proxies and block incoming data traffic at TCP ports, meaning MTX-Tunnel can only be used in server mode (MTX_mode: server). Contact your distributor for further information.

- The GPRS_password2 parameter is for the exclusive use of models that have the “DUAL SIM” feature and refers to the Password used by the secondary SIM. In the case of the MTX-IOT-S family modems, the secondary SIM is the one inside the modem, accessible by opening the case

GPRS_timeout

Description: this parameter sets the time period during which the 4G/3G/2G connection (and associated services – serial tunnel, WebServer, Telnet) will remain active when it is activated by an SMS, missed call, change in a digital or analog input, alarm timing, etc.

Possible values: 0... 100000 (minutes)

Default value: 0

Additional notes:

- Value=0 will establish a permanent GPRS connection; this means that the modem (and associated services –serial tunnel, WebServer, Telnet) will be connected to the 4G/3G/2G 100% of the time after power-up. MTX-Tunnel has intelligent procedures allowing persistent connectivity all the time, therefore it will reconnect after any failure in network signal coverage or power down
- Value>0 specifies the length of time that the 4G/3G/2G connection is active after activation (from SMS, missed call, change in a digital or analog input, alarm timing...)
- The time value is in minutes for MTX platforms, except the "ULP" models where the number represents the number of seconds, not minutes, that the GPRS connection will be active. This is to improve consumption

GPRS_autoTimeout

Description: parameter available from version MTX-Tunnel v7.15.

Only useful when the parameter GPRS_timeout > 0. It allows the 4G/3G/2G session timer to restart every time 4G/3G/2G data is sent/received.

Possible values: on, off

Default value: on

Additional notes:

- Example: if the parameter GPRS_timeout = 2, it implies that when the 4G/3G/2G session is activated (for example to set a 4G/3G/2G-Serie gateway against a server) it will last for 2 minutes. If the parameter GPRS_autoTimeout = on, every time that data is received via the 4G/3G/2G gateway, the 4G/3G/2G session timer will restart after 2 minutes, that is, 2 minutes with no traffic must pass for the session to close. If GPRS_autoTimeout = off, once the 4G/3G/2G session started, after 2 minutes, regardless of whether there was traffic or not, the session will close
- Very useful for low consumption scenarios

GPRS_dns, GPRS_dns2

Description: public DNS IP address.

Possible values: IP address in format XXX.XXX.XXX.XXX

Default value: 0.0.0.0 (0.0.0.0 = automatically provided by GSM)

Additional notes:

- Use DNS if you use names instead of IP addresses in parameters like TCP_IP, MTX_TPServer...
- The GPRS_dns2 parameter is for the exclusive use of models that have the “DUAL SIM” feature and refers to the DNS used by the secondary SIM. In the case of MTX-IOT-S family modems, the secondary SIM is the one located inside the modem, accessible by opening its casing

GPRS_auto

Description: this parameter will automatically use APN, login, password information from the SIM card.

Possible values: on, off

Default value: off

Additional notes:

- New parameter from version MTX-Tunnel v5.7
- When the GPRS_auto parameter has the value “on”, MTX-Tunnel will not take into account the values of the GPRS_apn, GPRS_login & GPRS_password parameters
- When the GPRS_auto parameter has the value “on”, it is necessary to add a new configuration file to MTX-Tunnel memory called “operators.txt”. This file has the following syntax:

  IDOperador1,apn,login,password,name[ENTER]

  IDOperador2,apn,login,password,name[ENTER]

  ...

  operators.txt file example

  21407:movistar.es,MOVISTAR,MOVISTAR,Movistar

  21401:airtelnet.es,vodafone,vodafone,Vodafone

  21403:internet,CLIENTE,AMENA,Orange

  21404:internet,,,Yoigo
- MTX-Tunnel gets the network operator (IDOperator variable) from “operator.txt” file and then gets the login and password information from it too
- This is useful if you do not know which SIM card operator the end customer will use and it also makes it possible to change the SIM card operator without any further configuration or operation

GPRS_mode

Description: parameter available from version MTX-Tunnel v7.15.

This parameter allows to choose the technology. It is possible to choose between “auto” (automatic change 4G/3G/2G), 2G only or 3G only or 4G only.

Possible values: auto, 2g, 3g, 4g

Default value: auto

Additional notes:

- This parameter is only allowed by modems with 3G and 4G technology Don’t use with 2G models

8.8 TCP Configuration parameters: "TCP_"

This important configuration parameter relates to TCP protocol communication in MTX-Tunnel. They are mandatory when using 4G/3G/2G TCP – Serial RS232/485 tunnel in client (MTX_mode: client) or server mode (MTX_mode: server).

TCP_IP

Description: IP server address that MTX-Tunnel will connect to if it is client configured (MTX_mode: client).

Possible values: IP address format XXX.XXX.XXX.XXX or DNS

Default value: none

Additional notes:

- This parameter is also used if MTX-Tunnel is configured in server mode (MTX_mode: server) and if MTX_temporalClient is enabled (value ="on"). The IP address or DNS name is used by the temporal client to establish a connection. Please read the information regarding the MTX_temporalClient parameter for more details

TCP_port

Description: TCP port value: MTX-Tunnel is used in both client (MTX_mode: client) and server mode (MTX_mode: server). In server mode, MTX-Tunnel waits for incoming connections at the specified port.

This specifies the TCP port that MTX-Tunnel will connect to when it is configured in client mode (MTX_mode: client), and in server mode (MTX_mode: server) it indicates the TCP port where incoming connections are made.

Possible values: 1... 65535

Default value: 20010

Additional notes:

- This parameter is also used if MTX-Tunnel is configured in server mode and MTX_temporalClient parameter is enabled (value =”on”). The IP address or DNS name is used by the temporal client to establish a connection. Please read the information regarding the MTX_temporalClient parameter for more details

TCP_IP2

Description: this parameter has only been available since the MTX-Tunnelv7.8. In the event that the modem is configured as a client (MTX_mode: client) and the connection is permanent (GPRS_timeout: 0), you can specify an IP address in this parameter which allows you to establish a second 4G/3G/2G-serial gateway.

This means that on one hand you have the first 4G/3G/2G-serial gateway associated with the modem's COM1 port and on the other hand you have the new gateway associated with the modem's COM2 port.

Possible values: An IP address XXX.XXX.XXX.XXX or a DNS

Default Value: none

Additional notes:

- This parameter can only be used with modems that have two serial ports.

TCP_port2

Description: this parameter has only been available since MTX-Tunnelv7.8. In the event that the modem is configured in client mode (MTX_mode: client), the connection is permanent (GPRS_timeout: 0) and you have specified an IP address in the parameter TCP_IP2, this parameter allows you to choose the port for the TCP connection.

This means that on the one hand you have the first 4G/3G/2G-serial gateway associated with the modem's COM1 port and on the other hand you have the new gateway associated with the modem's COM2 port.

From MTX-Tunnel v9 onwards, in case the modem is configured in Server mode (MTX_mode: server) and the connection is permanent (GPRS_timeout: 0) , this parameter allows you to specify the TCP listening port of the second gateway.

That means that on the one hand you have a 4G/3G/2G-serial gateway associated with the modem's COM1 port, and on the other hand you have the new gateway associated with the modem's COM2 serial port. So, two serial gateway running in parallel.

Possible values: 1... 65535

Default Value: 20010

Additional notes:

- This parameter can only be used with modems that have two serial ports.

8.9 UDP configuration parameter: "UDP_"

These parameters are related to MTX-Tunnel data communication using the UDP protocol. It is needed in the GPRS (UDP) – Serial (RS232/485) tunnel.

UDP_IP

Description: end IP address where MTX-Tunnel will send data to in UDP protocol. MTX-Tunnel needs be set in UDP mode (MTX_mode: udp).

Possible values: IP address format XXX.XXX.XXX.XXX. DNS text string can be also used

Default value: none

Additional notes:

- The use of the UDP communication protocol is only recommended in applications with a large amount of remote devices that need to connect to a server. It will save data traffic but keep in mind that it is not a connection-oriented protocol (no ACK)
- From MTX-Tunnel version 7.17 it is possible to specify up to 5 IP addresses to send/receive the UDP data; you just need to separate the IP addresses by a comma "," For example:

  UDP_IP: 100.101.102.1,200.201.202.2,200.201.202.10

UDP_localPort

Description: local UDP port used in UDP communication mode (MTX_mode: udp).

Possible values: 1... 65535

Default value: 20010

Additional notes:

- The use of UDP communication protocol is only recommended in applications with a large amount of remote devices that need to connect to a server. It will save data traffic but keep in mind that it is not a connection-oriented protocol (no ACK)

UDP_remotePort

Description: remote device's UDP port to where MTX-Tunnel sends serial data in UDP communication mode (MTX_mode: udp).

Possible values: 1... 65535

Default value: 20010

Additional notes:

- The use of UDP communication protocol is only recommended in applications with a large amount of remote devices that need to connect to a server. It will save data traffic but keep in mind that is not a connection-oriented protocol (no ACK)

8.10 ALARM Configuration parameter: "ALARM_"

MTX-Tunnel allows the sending of alarm SMS messages in case of changes to a digital input. The following configuration parameters refer to this feature.

MTX-Tunnel allows the sending of alarm messages under different circumstances, including SMS alarms. This section of configuration parameters refers to this feature.

ALARM_smsNumber1, ALARM_smsNumber2, ... , ALARM_smsNumber10

Description: to specify which phone numbers you want to send the SMS alarm to.

Possible values: Phone numbers, national or international numbering scheme

Default value: none

Additional notes:

- Valid characters are the numbers "0" ... "9" and the "+" character

ALARM_powerEnabled

Description: to specify if there will be an SMS alarm message when there is a power failure, or when the system goes back to normal.

Possible values: on, off

Default value: off

Additional notes:

- This parameter will only work properly in MTX devices with an internal battery

ALARM_powerMessageOn, ALARM_powerMessageOff

Description: it specifies the text of the SMS alarm message that will be sent when a change is detected in the power system. A message for power failure and a message for power back on can be specified.

Possible values: Text of less than 160 characters

Default value: “Power On” and “Power Off”

Additional notes:

- The text can be no longer than 160 characters, which is the maximum length of a SMS text message
- These parameters will only work properly with MTX devices with an internal battery

ALARM_ulpEnabled

Description: it allows to enable the SMS alarm for the equipment that has ULP. If this option is enabled, if an MTX is on ULP (asleep) and is awaken by a tamper input (digital input), when the MTX wakes up, it will send an alarm SMS.

Possible values: on, off

Default value: off

Additional notes:

- The text of the SMS message is established in the parameter ALARM_ulpMessage
- The telephone numbers you send the SMS to must be configured in the parameters ALARM_smsNumber1 ... ALARM_smsNumber10
- Find the example on this guide where the parameter ALARM_ulpEnabled is for mor information

ALARM_ulpMessage

Description: text of the SMS message that will be sent when an MTX with ULP awakes with a tamper input (digital input).

Possible values: A text containing less than 160 characters

Default value: MTX waking up

Additional notes:

- Remember to activate the parameter MTX_ulpEnabled if you want to use this feature
- The telephone numbers you send the SMS to must be configured in the parameters ALARM_smsNumber1 ... ALARM_smsNumber10
- Find the example on this guide where the parameter ALARM_ulpEnabled is for mor information

8.11 FIREWALL configuration parameters: "FIREWALL_"

MTXTunnel by default allows access from any IP address.

If you enable a firewall, your system will be secure as you will avoid unauthorized access. This way, only configured IP (or DNS) addresses can connect and access MTX-Tunnel and the other services (GPRS serial tunnel, WebServer, Telnet...).

FIREWALL_enabled

Description: parameter to enable firewall and to enable access only from the configured IP address.

Possible values: on, off

Default value: on

Additional notes:

- This firewall will avoid unauthorized connections with MTX-Tunnel services.If you need to use the same security access, it is mandatory to enable WEBSERVER_firewall and TELNET_firewall ("on" value)

FIREWALL_IP1, FIREWALL_IP2, ... , FIREWALL_IP10

Description: authorized IP addresses (FIREWALL_enabled: on).

Possible values: IP address XXX.XXX.XXX.XXX string format

Default value: none

Additional notes:

- Up to 10 IP addresses can be written

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

8.12 SMS Configuration parameter: "SMS_"

MTXTunnel has a lot of SMS features. SMSs can be used (sent and received) to configure, report, start/stop any service, etc.

You need to configure the SMS messaging feature with the following parameters:

SMS_sendIP

Description: MTXTunnel starts the GPRS connection and associated services (tunnel, WebServer, telnet ...) if it receives a missed call or an SMS with the text string "mtxtunnel on". This parameter means that MTX-Tunnel has to send the obtained IP address in another SMS.

Possible values: on, off

Default value: off

Additional notes:

- If the parameter was already enabled (value "on") and the data connection was already established, MTX-Tunnel sends the obtained IP address to the sender
- Only authorized phones will receive a response if the SMS_allPhones parameter is "off"

SMS_ATEnabled

Description: you can execute an AT command sent in an SMS format from a mobile phone terminal by enabling this parameter.

Possible values: on, off

Default value: off

Additional notes:

- Use a value of “on” to enable this feature. The SMS must start with the special string header “MTXTUNNEL AT”
- E.g. If you need to find out the remote modem network coverage, send an SMS with MTXTUNNEL AT+CSQ. Do not use “”
- Only authorized mobile phone numbers can execute AT commands using SMS if the SMS_allPhones parameter is disabled (“off” value)

SMS_ATResponse

Description: some SMSs can be sent to perform an operation and you do not need a response (an SMS to be sent back to you). However if you need an SMS response from MTX-Tunnel execution status, enable this parameter.

Possible values: on, off

Default value: off

Additional notes:

- If you send a command by SMS or RS232/485 (see SMS_tunnelString) to the end device, MTX-Tunnel waits for 5 seconds before replying and sending an SMS with the response or error

SMS_allPhones

Description: you can use this parameter to specify whether any phone number is authorized to start the data session by sending an SMS (text “mtxtunnel on”) or via a missed (voice) call.

Possible values: on, off

Default value: off

Additional notes:

- “on” value means all end phone numbers are authorized for SMS control
- “off” value will only allow this feature to work with configured phone numbers. See the next parameter to learn how to authorize phone numbers

SMS_validPhone1, SMS_validPhone2, ... , SMS_validPhone10

Description: this parameter is used to allow up to 10 phone numbers to have MTX-Tunnel SMS control.

SMS_allPhones parameter must be “off.”

Possible values: Numeric phone number, either local or international

Default value: none

Additional notes:

- Valid characters are the numbers “0”... “9” and the “+” character

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

SMS_alias1, SMS_alias2, ... , SMS_alias20

Description: previous configuration parameters show how to send AT commands via SMS so they are executed by MTX-Tunnel. However, sometimes it can be uncomfortable to send an SMS with the text AT^MTXTUNNEL=SETIO,3,1 to commute a digital output, specially if the MTX-Tunnel is part of a third-party system.

It is possible to establish up to 20 alias for the execution of AT commands (10 for versions before v11). Following the previous example, an alias could be established: Rele1on> AT^MTXTUNNEL=SETIO,3,1. With that, a received SMS by the MTX-Tunnel with the text “Rele1on” would be interpreted like the command “AT^MTXTUNNEL=SETIO,3,1.”

Possible values: 64 characters max. text string

Default value: none

Additional notes:

- The string must contain the special character “>” to separate the alias and the AT command

SMS_aliasOk

Description: the SMS_ATResponse parameter allows us to configure MTX-Tunnel to reply via SMS to an incoming SMS that contains an AT command with the corresponding response. This parameter (SMS_aliasOk) allows us to substitute the response given by the modem with user defined text when the AT command has been correctly executed.

Possible values: A text string with a maximum of 100 characters

Default value: none

Additional notes:

- If the SMS_aliasOk parameter contains text (i.e. not left blank), this text will be sent via SMS as a response to the AT command instead of the technical result of the AT command generated by the modem

SMS_aliasError

Description: the SMS_ATResponse parameter allows us to configure MTX-Tunnel to reply via SMS to an incoming SMS that contains an AT command with the corresponding response. This parameter (SMS_aliasOk) allows us to substitute the (technical) response given by the modem with user defined text when the AT command has NOT been correctly executed.

You can use this parameter if the execution of an AT command was unsuccessful and there was an error. It is possible to change the error with your own text response string.

Possible values: A text string with a maximum of100 characters

Default value: none

Additional notes:

- If the SMS_aliasOk parameter contains text (i.e. not left blank), this text will be sent via SMS as a response to the AT command instead of the technical result of the AT command generated by the modem

SMS_aliasResponse

Description: SMS_aliasResponse parameter allows to define the corresponding response to an alias. That means it allows to decide whether to return the full answer of the command to be run (i.e. including the executed command in the SMS) or to include the response only.

Possible values: full, result

Default value: full

Additional notes:

- Examples:

  In case the value of the parameter SMS_aliasResponse is “full”, the following will occur:

  SMS sent: TEMP

  SMS received: AT^MTXTUNNEL=GETMODBUS,1;10;1;3

  25

  OK

  In case the value of the parameter SMS_aliasResponse is “result”, the following will occur:

  SMS sent: TEMP

  SMS received: TEMP > 25

SMS_tunnelString

Description: MTXTunnel features SMS tunnelling to serial COM1. This parameter enables and configures text to be sent directly to the device connected to COM1.

A value “MTX” enables SMS tunnelling. Example: An SMS with the text “MTX+[space]123456789” will be interpreted and the text string “123456789” will be sent to the serial COM.

Possible values: A text string with a maximum of 16 characters

Default value: none

Additional notes:

- The space character is used as separator between the SMS_tunnelString parameter activation string and the text string to be sent to COM1
- If SMS_responseAT parameter is enabled (value=on), MTX-Tunnel will wait 5 seconds and an SMS will be sent back with the device’s response data present at the COM1 port
- This response must be truncated to 160 characters long so it does not exceed the string length

SMS_urc

Description: this parameter is available from version MTX-Tunnel v7.15.

If the value is set to “on”, every time an SMS messsage is received, the modem will send an URC via the serial port indicating the SMS has been received.

Possible values: on, off

Default value: off

Additional notes:

- The URC format is:

```
^MTX_SMS[space]PhoneNumber,Textmessage
```

  It can be very useful if, besides using a 4G/3G/2G serial gateway, you want to receive in RAW mode an SMS message sent from a mobile phone.

SMS_header

Description: parameter available from version MTX-Tunnel v7.17. In v7.16 MTX-Tunnel and previous versions, all AT commands sent via SMS had to be preceded by the key “mtxtunnel.” For example, to send the AT command to check remote coverage it was necessary to send a SMS message with the text “mtxtunnel at+csq.” It is now possible to personalize this key.

Possible values: Text of up to 6 characters

The value “none” will make it unnecessary to enter any header in the SMS

Default value: mtxtunnel (for compatibility with previous versions)

Additional notes:

- The use of an additional header is always recommended, especially if all telephone numbers are authorized to send commands via SMS

SMS_replaceText

Description: this parameter allows you to replace a string received in a SMS message with a different one. For example, if in an SMS message a text with the characters “XX” is included, these characters could be replaced by “@12345”.

Possible values: TextToReplace; TextReplacement

Default value: none

Additional notes:

- It can be useful to send special characters such as @
- An example of configuration can be:

  SMS_replaceText: XX,@12345 (texts separated with a semicolon)

SMS_defaultPrefix

Description: with this parameter it is possible to set the international prefix that will be considered for those SMS or missed GSM calls which are received in the modem with any prefix. That is to say, if modem receives a GSM call and it doesn't include the prefix, this prefix will be used.

Possible values: Maximum length 16 char

Default value: nothing

Additional notes:

- This parameter is only used by modem when the parameter SMS_allPhones has the value "off", i.e. when authorized telephone numbers are authorized
- Example. We need to make a missed call to the modem for getting the current IP address. Additionaly we want to send AT command by SMS from authorized telephone numbers

  SMS_allPhones: off

  SMS_sendIP: on

  SMS_ATEnabled: on

  SMS_ATResponse: on

  SMS_validPhone1: +34666123456

  SMS_validPhone2: +34666123457

  SMS_defaultPrefix: +34

8.13 DynDNS configuration parameter: "DYNDNS_"

MTXTunnel version 5.0 and above include the "DYNDNS_" parameter. They refer to the DynDNS service which allows you to assign a DNS name to a dynamic IP address, in such a way that a DNS with the format "mymodem.dyndns.org" will always take note of the IP address obtained by the network operator at all times.

MTXTunnel connects with DynDNS server so it can detect and follow any changes in IP address.

You can create a DynDNS account for free and obtain more information about this service by accessing the webpage (www.dyndns.org) of the provider of this service.

DYNDNS_enabled

Description: this parameter enables DynDNS in MTX-Tunnel.

If used, every data connection will refresh the IP address in the DynDNS server, so any change in IP address will be detected.

Possible values: on, off

Default value: off

Additional notes:

- You can open a free account with the DynDNS service at www.dyndns.org
- Only use the DynDNS feature when your SIM is dynamic IP provisioned. It makes no sense to use a dynamic IP provisioned SIM if your network operator has fixed IP addressing
- DynDNS must be used in incoming connection requests with MTX-Tunnel services using socket servers like:
    - GPRS-Serial RS232 server mode tunnel
    - WebServer
    - Telnet
    - If you are going to use MTX-Tunnel client connection services, meaning MTX-Tunnel starts out with a connection from a well known IP address, the DynDNS service is not needed and must be disabled (value "off")

DYNDNS_server

Description: DynDNS server information.

Possible values: A text string with a maximum of 128 characters

Default value: none

Additional notes:

- If you already have an account with www.dyndns.org, the DynDNS value is normally something like: members.dyndns.org. If you use www.no-ip.com the value is normally something like dynupdate.no-ip.com

DYNDNS_hostname

Description: DNS name of your account with your DynDNS service provider.

Possible values: A text string with a maximum of 128 characters

Default value: none

Additional notes:

- Example MyModem.dyndns.org

DYNDNS_login

Description: login of your DynDNS account.

Possible values: A text string with a maximum of 32 characters

Default value: none

DYNDNS_password

Description: your DynDNS account password.

Possible values: A text string with a maximum of 32 characters

Default value: none

DYNDNS_period

Description: a data string with the DNS information is sent if the MTX-Tunnel IP address changes, but you can also define it to send this information periodically. Therefore this gives you a more reliable indication as to whether the IP address has changed.

Possible values: 0, 30... 2592000 (in seconds)

Default value: 0

Additional notes:

- DynDNS will not be checked/upgraded if the value is"0", but it will still be aware of when the IP address changes
- Parameter value is in seconds
- It is recommended to use this parameter in some situations:
    - Let's suppose the DynDNS server fails or hangs up and the modem IP address changes. MTX-Tunnel will upgrade DynDNS even if the IP address has not changed

8.14 DNS Parameter configuration: "DNS_"

In the previous section about DynDNS we explained how and why we use it when we have a few remote modem SIMs that have dynamic IP addresses. If you plan to use and control a large number of devices, this is not practical. Also, DynDNS requires you to have an external server, which means that any possible drops on the server are harder to control.

The following parameters are useful for when you are required to use your own server to collect the changes detected by MTX-Tunnel in the IP addresses.

In other words, these parameters will make sure MTX-Tunnel informs your server of any changes in its IP address, giving you complete control of the system at all times.

DNS_enabled

Description: enable the DNS service using your own server. Every time the modem starts a data connection or changes the IP address, it is sent to your configured server.

Possible values: on, off

Default value: off

Additional notes:

- Only use the DNS feature when your SIM is dynamic IP provisioned. It makes no sense to use a dynamic IP provisioned SIM if your network operator has fixed IP addressing
- DNS must be used in incoming connection requests with MTX-Tunnel services using socket servers like:
    - GPRS-Serial RS232 server mode tunnel
    - WebServer
    - Telnet
    - If you are going to use MTX-Tunnel client connection services, meaning that the MTX-Tunnel starts with a connection from a well known IP address, the DNS service is not needed and must be disabled (value "off")

DNS_mode

Description: the MTXTunnel mode, in which MTX-Tunnel sends the change in IP address to a server. We can use a TCP socket or HTTP protocol for web server.

Possible values: socket, socketjson, http, mqtt

Default value: socket

Additional notes:

- If you are going to write your own code/application, use the “socket” option
- If you already have a web server and the code is written in ASP or PHP, we recommend that you use the “http” option
- If you are going to send data from sensors to a MQTT broker, it will be easier to use the “mqtt” method. From the MTX-Tunnel 9.25 on it is possible to configure the “mqtt” mode. If you select this method don’t forget to configure the MQTT parameters and the DNS_mqttTopic parameter

DNS_password

Description: you can secure this service by setting up a password. MTX-Tunnel will send the DNS_password in any new GPRS connection or if there are any IP address changes and your server will have to check the password.

Possible values: A text string of up to 64 characters

Default value: none

Additional notes:

- DNS_password is sent using either “socket” or “http” transmission mode

DNS_server

Description: socket mode: (DNS_mode: socket) The value is the DNS or IP server address where the new IP address is to be sent..

HTTP mode: (DNS_mode: http) The value is the URL (domain + web page) of the WebServer to which you will pass on information about the new IP address.

Possible values: Text string of less than 255 characters

Default value: none

Additional notes:

In HTTP mode “DNS_mode: http”, DNS_server URL does not include “http://” string.

DNS example URL: www.mtxtunnel.com/dns.asp

- In Socket mode, MTX-Tunnel sends the following string to the server:

```
#IMEI#DNS_password#IPPublica#
```

- In HTTP mode, MTX-Tunnel sends the following string to server:

```
URL?IMEI=<suIMEI>&PASS=<DNS_password>&IP=<IP Public>
```

If you program web pages in ASP, you can collect the information using this method as an example:

```
<%
IMEI=Request.QueryString(“IMEI”)
Password=Request.QueryString(“PASS”)
IP=Request.QueryString(“IP”)
%>
```

DNS_port

Description: TCP DNS server port being used in DNS socket mode (DNS_mode: socket).

Possible values: 1... 65535

Default value: 20011

Additional notes:

- Only use this parameter if “socket”mode is used. If you use “http” mode, the standard port is 80. However if you use another one, you need to include it in the DNS_server parameter
- E.g. if you are going to use port 20011, DNS_server value parameter is: www.mtxtunnel.com:20011/dns.asp

DNS_extended

Description: if enabled, and only if the DNS service is enabled too (DNS_enabled: on), MTX-Tunnel will send the password and IP address information together with the IMEI and more information like the GPIO status. This is useful in telemetry applications. To send additional information, DNS_extended must be configured to “on”.

Possible values: on, off

Default value: off

Additional notes:

- DNS_extended “on”, DNS_mode set as “socket”, the information sent to the server is:

```
#IMEI#DNS_password#IPPublica#gpio1#gpio2#gpio3#gpio4#gpio5
#gpio6#gpio7#gpio8#gpio9#gpio10#ADC1#ADC2#<gpsLocation>#
```

  gpioX is input/output X value (0 -1)

  ADCX is analog to digital converter X value.

  In the MTX terminal modems with GPS receiver, the GPS location information is also added.

- DNS_extended “on”, DNS_mode set as “http”, the information sent to the server is:

```
URL?IMEI=<suIMEI>&PASS=<DNS_password>&IP=<IP Public> &GPIO1=X
&GPIO2=X &GPIO3=X &GPIO4=X &GPIO5=X &GPIO6=X &GPIO7=X &GPIO8=X
&GPIO9=X  &GPIO10=X &ADC1=X&ADC2=X&GPS=<gpsLocation>
```

  gpioX is input/output X value (0 -1)

  ADCX is analog to digital converter X value.

  In the MTX terminal modems with GPS receiver, the GPS location information is also added.

DNS_gpios

Description: when this parameter is enabled it can send DNS information (which can include GPIO and ADC values if DNS_extended = “on”) when there is a change in a digital input.

Possible values: on, off

Default value: off

Additional notes:

- MTX-Tunnel sends just one data DNS information string if there has been one or several changes in digital inputs
- Parameter available from version MTX-Tunnel v5.3

DNS_adc1, DNS_adc2

Description: this parameter, if enabled, can send DNS information (which can include GPIO and ADC values if DNS_extended = “on”) only when the analog input is above or below trigger value.

Possible values: 0, 250 ... 47750

Default value: 0

Additional notes:

- 0 value disables this feature and a DNS string will be not sent
- DNS_adcX >=250 sets the trigger analog value above or below +-250mV, if it exceeds this value a DNS frame string will be sent
- DNS_adc1: 1200 means that when DNS_adc1 > 1200mV + 250mV, the DNS text string will be sent, as well as when DNS_adc1 < 1200mV – 250mV
- Parameter available from MTX-Tunnel v5.3

DNS_period

Description: a data string with the DNS information is sent whenever the MTX-Tunnel IP address changes, but you can also define it to send the IP address periodically. This is useful for a more secure indication of IP address changes and it establishes timing periods by sending telemetric data if the DNS_extended parameter is enabled (value “on”).

Possible values: 0, 30... 2592000

Default value: 0

Additional notes:

- Value=0 means DNS information is not sent periodically
- Value of time is in seconds

DNS_httpMode

Description: this parameter allows you to choose the type of communication that will be used when DNS_mode is set to “http” mode. You can choose between “get” and “json”

Possible values: get, getjson, postjson

Default value: get

Additional notes:

- The parameter is available from MTX-Tunnel version 8.10 onwards. Before this version the only possible option was “get”. It is strongly recommended not to use the option “get”, and use the option ‘’getjson” or “postjson”.
    - EXAMPLE of data transmission in “get” mode where DNS_extended parameter has the value “off”:

      http://www.mydomain.com?TYPE=DNS&IMEI=357973041110401&
      PASS=ID000001&IP=95.126.113.202&CSQ=24&VER=9.12&AUX=0&MOD=201

    - EXAMPLE of data transmission in “getjson” mode where DNS_extended parameter has the value “off”:

      {“TYPE”,”DNS”,”IMEI”: 357973041110401,“P”:”ID001”,”IP”:”95.126.113.202”,
      ”CSQ”:24,”VER”:”9.12”, “AUX”:”0”,”MOD”:201}

      Where:

      TYPE: DNS frame type

      IMEI: IMEI of the modem (unique device identification)

      P: user’s field in DNS_password parameter

      IP: MTX-Tunnel IP

      CSQ: GSM coverage of MTX (0... 31)

      VER: MTX-Tunnel firmware version

      AUX: Reserved for configuration version control

      MOD: MTX terminal model

    - EXAMPLE of data transmission in “getjson” mode where DNS_extended parameter has the value “on”:

      {“TYPE”,”DNS”,”IMEI”: 357973041110401, “P”:”ID001”,”IP”:”95.126.113.202”,
      ”CSQ”:24,”VER”:”9.12”, “AUX”:”0”,”MOD”:201,”IO1”:0,”IO2”:0,”IO3”:0,”IO4”:0,”IO5”:0,
      ”IO6”:0,“IO7”:0,”IO8”:0,”IO9”:0,”IO10”:0,”AD1”:100,”AD2”:2000,”CO1”:”0”,”CO3”:”0”,

”CID”:”214;07;0322;4E8F”}”}

Where:

TYPE: DNS frame type

IMEI: IMEI of the modem (unique device identification)

P: user’s field in DNS_password parameter

IP: MTX-Tunnel IP

CSQ: GSM coverage of MTX (0 ... 31)

VER: MTX-Tunnel firmware version

AUX: Reserved for configuration version control

MOD: MTX terminal model

IO1: Value of digital input/output 1 of the modem (if it is available)

IO2: Value of digital input/output 2 of the modem (if it is available)

IO3: Value of digital input/output 3 of the modem (if it is available)

IO4: Value of digital input/output 4 of the modem (if it is available)

IO5: Value of digital input/output 5 of the modem (if it is available)

IO6: Value of digital input/output 6 of the modem (if it is available)

IO7: Value of digital input/output 7 of the modem (if it is available)

IO8: Value of digital input/output 8 of the modem (if it is available)

IO9: Value of digital input/output 9 of the modem (if it is available)

IO10: Value of digital input/output 10 of the modem (if it is available)

AD1: Value of analog input 1 of the modem (if it is available)

AD2: Value of analog input 2 of the modem (if it is available)

CO1: Value of pulse counter input 1 (if it is available)

CO3: Value of pulse counter input 3 (if it is available)

CID: GSM cell information (for GSM localization)

- EXAMPLE of data transfer in “postjson” mode where DNS_extended parameter has the value “off”. When using “postjson” mode data is sent to JSON as follows:

  {“IMEI”:”357042060366409”,”TYPE”:”DNS”,“P”:”ID-12345678”,
  ”IP”:”95.126.113.202”,”CSQ”:24,”VER”:”9.12”,“AUX”:”4”,”MOD”:”201”,”VCC”:12100}

Where:

TYPE: DNS frame type

IMEI: MTX internal indentification

P: user's field in DNS_password parameter

IP: MTX-Tunnel IP

CSQ: GSM coverage of MTX (0 ... 31)

VER: MTX-Tunnel firmware version

AUX: Reserved for configuration version control

MOD: MTX terminal model

VCC: MTX supply voltage (in millivolts)

- EXAMPLE of data transmission in "postjson" mode where DNS_extended parameter has the value "on". When using "postjson" mode data is sent to JSON as follows:

  {"IMEI":"357042060366409","TYPE":"DNS","P":"ID-12345678",
  "IP":"95.126.113.202","CSQ":24,"VER":"9.12","AUX":"4","MOD":"201","IO1":0,"IO2":0,
  "IO3":0,"IO4":0,"IO5":0,"IO6":0,"IO7":0,"IO8":0,"IO9":0,"IO10":0,"AD1":0,"AD2":0,
  "CO1":"0","CO2":"0","CID":"214;07;0322;4E8F"}

  Where:

  TYPE: DNS frame type

  IMEI: MTX internal indentification

  P: user's field in DNS_password parameter

  IP: MTX-Tunnel IP

  CSQ: GSM coverage of MTX (0 ... 31)

  VER: MTX-Tunnel firmware version

  AUX: Reserved for configuration version control

  MOD: MTX terminal model

  IO1: Value of digital input/output 1 of the modem (if it is available)

  IO2: Value of digital input/output 2 of the modem (if it is available)

  IO3: Value of digital input/output 3 of the modem (if it is available)

  IO4: Value of digital input/output 4 of the modem (if it is available)

  IO5: Value of digital input/output 5 of the modem (if it is available)

  IO6: Value of digital input/output 6 of the modem (if it is available)

  IO7: Value of digital input/output 7 of the modem (if it is available)

IO8: Value of digital input/output 8 of the modem (if it is available)

IO9: Value of digital input/output 9 of the modem (if it is available)

IO10: Value of digital input/output 10 of the modem (if it is available)

AD1: Value of analog input 1 of the modem (if it is available)

AD2: Value of analog input 2 of the modem (if it is available)

CO1: Value of pulse counter input 1 (if it is available)

CO3: Value of pulse counter input 3 (if it is available)

CID: GSM cell information (for GSM localization)

DNS_serverLogin

Description: this parameter allows you to establish a login (username) that must be used by MTX-Tunnel to send data to a Web server containing a Login/Password authentication system.

Possible values: Text string of up to 64 characters

Default value: none

Additional notes:

- If your web server does not use a simple authentication system consisting of a Login and Password, this parameter is not necessary. It is necessary, however, in those web servers that use authentication systems

  This parameter is available in versions of MTX-Tunnel from v8.10.

DNS_serverPassword

Description: this parameter allows you to establish a password that must be used by MTX-Tunnel to send data to a Web server containing a Login/Password authentication system.

Possible values: Text string of up to 64 characters

Default value: none

Additional notes:

- If your web server does not use a simple authentication system consisting of a Login and Password, this parameter is not necessary. It is necessary, however, in those web servers that use authentication systems.

DNS_https

Description: allows to state whether information is sent via http or via https in encrypted format.

Possible values: on, off

Default value: off

Additional notes:

- Keep in mind that if you want to activate this option, you will need a web server with SSL support

DNS_header1, DNS_header2, DNS_header3

Description: allows to use personalized HTTP headers.Very useful for communication between MTX-Tunnel and third-party web platforms (like THINGWORKS).

Possible values: ASCII string up to 64 charaters

Default value: none

Additional notes:

- The headers have the following format: nameHeader;valueHeader. That is to say, fields are separated by ; (semicolon)
- You will find them necessary if you want your MTX-Tunnel to communicate with some generic platforms. For example, to communicate with Thingworks platform you will have to specify something similar to:

  DNS_header1: Content-Type;application/json

  DNS _header2: Accept;application/json

  DNS _header3: appKey;194f5476-7346-4638-ac30-bbca28595be1

DNS_mqttTopic

Description: configures the topic that MTX-Tunnel will use to publish all data in the DNS_ service.

Possible values: ASCII string up to 64 charaters

Default value: none

Additional notes:

- In the same way as the MQTT_attopicX parameters, you can use the [IMEI] tag in the parameter and the MTX-Tunnel will internally replace that tag with the appropriate value. For example, if your modem has the IMEI 354740050182909 you could specify in the config.txt configuration file the value:

  DNS_mqttTopic: [IMEI]/DNS

  And the MTX-Tunnel would send the DNS service data using the topic:

  354740050182909/DNS

- Data are sent to the MQTT topic in JSON format. Check the parameter DNS_httpMode to find information about sent JSON

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

8.15 ULP Configuration parameter: "ULP_"

The "ULP_" configuration parameter is only related to MTX-Tunnel installed on the modem terminal modem. ULP means Ultra Low Power mode in which power consumption is only 10uA. In this mode, the modem remains on standby waiting for a configured alarm or a change in the level of digital input. Then all 4G/3G/2G tunnels and services are established and after a different configured time, it will switch off and return to ULP mode.

If your application needs low power consumption please read the following configuration parameters:

ULP_enabled

Description: this parameter allows to activate the ULP service (low power) in case the MTX model has it.

Possible values: on, off

Default value: off

Additional notes:

In SMS mode (ULP_mode: sms) you must use the "ULP_sleepMode:minutes" configuration parameter.

It is not possible to set a time for the modem to wake up from the ULP mode; it is only possible to specify a time for it to sleep. This is because "ULP_sleepMode: date" needs a GPRS connection for time synchronization and it can only connect during normal wakeup mode.

- Check Annexes for more information. Contact iotsupport@mtxm2m.com for more help
- Check also parameters ALARM_ulpEnabled and ALARM_ulpMessage

ULP_sleepMode

Description: ULP sleep mode:

- "minutes" configuration value:

  MTXTunnel wakes up and connects to 4G/3G/2G enabling services (Serial tunnel, WebServer, Telnet...). MTX-Tunnel will remain active the time specified in ULP_secondsOn. Passed that time, MTX-Tunnel will go into ULP mode (~10uA), waking up again after X minutes specified in the parameter ULP_minutesOff.

- "date" configuration value:

  MTXTunnel wakes up and connects to 4G/3G/2G enabling services (Serial tunnel, WebServer, Telnet...). MTX-Tunnel will remain active the time specified in ULP_secondsOn. Passed that time, MTX-Tunnel will go into ULP mode (~10uA), waking up again at the time specified in the parameters ULP_time1, ULP_time2, ... ULP_time10.

Possible values: minutes, date

Default value: minutes

Additional notes:

- If you are going to use date mode you MUST synchronize the internal timing with the server timing using the MTX_TPServer parameter. The modem will connect with the server's time and update its internal timing every time it wakes up. MTX-Tunnel uses UTC timing

ULP_minutesOff

Description: this parameter specifies the number of minutes during which the modem is in ULP (Ultra Low Power). After this time, the modem switches to normal mode.

Possible values: 0... 43200

Default value: 0

Additional notes:

- Once the modem wakes up, it will remain on the time specified in ULP_secondsOn
- If you select a value ULP_minutesOff=0 and the parameter ULP_sleepMode has the value “minutes,” the modem will only awake with a tamper input (digital input)

ULP_secondsOn

Description: this parameter specifies the number of seconds during which the modem is in ULP (Ultra Low Power). After this time, the modem will go back to sleep.

Possible values: 0... 86400

Default value: 60

Additional notes:

- Once the modem goes to sleep, it will remain that way the time specified in ULP_minutesOff

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

ULP_time1, ULP_time2... ULP_time10

Description: up to 10 different specified alarm times to wake up MTX-Tunnel from ULP mode (2uA) and activate a GPRS session.

The GPRS session will be active during GPRS_timeout second time value.

Possible values: DDHHMM (DD: day, HH hour in 24h format & MM (minutes)

Default value: none

Additional notes:

- DD and HH values can be XX, meaning any value

  Examples:

  ULP_time1: 012200

  MTXTunnel wakes up (changes from ULP to normal mode) every 1st day of the month at 22:00)

  ULP_time1: 011030

  ULP_time2: 151030

  MTX-Tunnel wakes up every 1st and 15th day of the month, at 10:30

  ULP_time1: XX1800

  MTXTunnel wakes up every 1st day of the month, at 18:00

  ULP_time1: XX0200

  ULP_time2: XX1400

  MTXTunnel wakes up from ULP to normal mode every day of the month, at 02.00 and 14.00

  ULP_time1: XXXX00

  MTXTunnel wakes up from ULP mode, every day of the current month and every hour

  Modem time uses UTC HOUR. Check differences with local time

ULP_smsExtended

Description: if disabled (value “off”) and the ULP mode is SMS (ULP_mode: sms), an SMS will be sent with the text specified in the MTX_IDClient parameter.

If enabled (value “on”), the SMS will also contain extended telemetry as specified in the ULP_mode parameter.

Possible values: on, off

Default value: on

Additional notes:

- Sometimes you just need to send the “MTX_IDClient” value parameter and not the GPIOs values as this saves power
- For example when the digital tamper input changes, the SMS text will not include any value, just the “Input change”

ULP_smsNumber1, ULP_smsNumber2, ULP_smsNumber3

Description: this parameter sets the telephone numbers to which SMSs containing the GPIO and ADC values will be sent when ULP mode is SMS: (ULP_mode: SMS).

You can use up to 3 phone numbers.

Possible values: National or international phone numbers are valid

Default value: none

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

ULP_rtc

Description: this parameter configures the way to wake up the MTX-65-ULP modem.

Timing alarm: MTX-Tunnel wakes up when alarm timing occurs. Internal RTC is used.

Interruption digital input (tamper): MTX-65-ULP modem wakes up when a special digital trigger input is activated.

Possible values: on, off

Default value: on

8.16 LOGGER configuration parameters: "LOGGER_"

The parameters with prefix LOGGER_ are those related to the MTX-Tunnel configurations related to the internal MTX-Tunnel datalogger.

The activation of the internal datalogger allows the storage of data read from Wavenis remote sensors (radio temperature probes, pulse counters...) of the modem's own inputs and outputs and readings of a modbus device connected to the serial port of the modem.

The activation of the datalogger is mandatory when using Wavenis remote devices or modbus devices that must be read through the serial port. See the examples in the annex for more information.

LOGGER_enabled

Description: enable internal MTX-Tunnel datalogger. It will store RF cards –Wavenis cards- readings, input/output values/statuses and Modbus data in the memory and sends it to a http server using GET via a JSON object.

Possible values: on, off

Default value: off

Additional notes:

- See Annexes for more information and examples of scenarios. Please use the account support line at iotsupport@mtxm2m.com for further information

LOGGER_password

Description: this parameter can set a user password string that can be used in every JSON object frame that goes to the end server.

Possible values: A text string with a maximum of 64 characters

Default value: none

Additional notes:

- See annexes for more information and examples of scenarios. Please use the account support line at iotsupport@mtxm2m.com for further information

LOGGER_server

Description: this parameter sets the server's URL datalogger and sends its saved data using a JSON object character.

Possible values: A text string with a maximum of 256 characters

Default value: none

Additional notes:

- URL example:

  www.midominio.com/set.asp?V=

  V is the variable used to send JSON information to the server

- See annexes for more information and examples of scenarios. Please use the technical support line at iotsupport@mtxm2m.com for further information

LOGGER_registerSize

Description: this parameter sets the register size of the non volatile Flash internal datalogger for JSON objects.

E.g. If using a remote Wavetherm temperature sensor, the reading will be stored in the memory in a JSON format:

{“IMEI”:357973041110401,”TS”:”18/10/12
10:04:37”,”P”:”matrix”,”A”:0B19083000D3,”BAT”:0,”T1”:23.4375,”T2”:0.0 }

The parameter value should be at least the maximum frame length that a JSON frame has.

In the above example it would be be 110. If you do not know what it could be, use 300 as the value.

Possible values: 10... 1024

Default value: 100

Additional notes:

- See annexes for more information and examples of scenarios. Please use the technical support line at iotsupport@mtxm2m.com for further information

LOGGER_numRegistersFlash

Description: this parameter allows you to specify the maximum number of records that the MTX-Tunnel can store in its internal flash memory.

Possible values: 1... 10000

Default value: 1500

Additional notes:

- Remember that the flash memory is used in cases where the MTX-Tunnel doesn't have, at a given moment, 4G/3G/2G connectivity for the broadcasting of information. This memory is used to ensure records are not lost. When 4G/3G/2G connectivity is back those records will be sent again via GPRS.
- It is recommended that the data file (data.txt that will be automatically created inside the modem) doesn't occupy more than 1MByte. Remember that the size of the file is calculated by multiplying LOGGER_numRegistersFlas x Logger_registerSize.

LOGGER_ioPeriod

Description: period time datalogger takes readings from digital and analog inputs, stores them to memory and afterwards, they are sent to server using a JSON object.

Possible values: 0 (disabled), 30... 2592000

Default value: 0

Additional notes:

- Please use the technical support line at iotsupport@mtxm2m.com for any further questions

LOGGER_serverLogin

Description: this parameter allows you to specify the login (user name) that the MTX-Tunnel must use to send data to a Web platform that has an authentication system of Login and Password.

Possible values: String of up to 64 characters

Default value: none

Additional notes:

- If your web platform does not use a simple authentication system based on Login and Password then it is not necessary to include this parameter in the set up file. Its use is only compulsory in web platforms that use authentication mechanisms
- This parameter is available from version MTX-Tunnel 7.27

LOGGER_serverPassword

Description: it allows you to specify the password to be used by the MTX-Tunnel to send data to a Web Platform with an authentication system of Login and Password.

Possible values: String of up to 64 characters

Default value: none

Additional notes:

- If your web platform does not have a simple authentication system of Login and Password, then it is not necessary to include this parameter in the setting file. It is only compulsory for web platforms with authentication systems

LOGGER_serialFrequency

Description: this allows you to establish the frequency with which the internal logger must register the received serial data.

Possible values: 0... 1000 (seconds)

Default value: 0

Additional notes:

- This configuration parameter is valid in versions from MTX-Tunnel 8.10

  A value of 0 means that no data is registered in the modem's serial port.

  A value of 1 means all received data is registered in the serial port.

  A value of 6 means 1 in every 6 data strings is registered. This is useful in situations such as with a temperature sensor that automatically sends data (without the option to change this), and we need to change the frequency.

  The data should be sent consecutive, without pauses. The maximum size is 256 bytes.

LOGGER_serialData1... LOGGER_serialData10

Description: this allows you to configure up to 10 data strings to be sent by the modem; i.e. every X seconds (configured in the LOGGER_serialPeriod parameter), MTX will send this data via serial ports, logging the responses to send them to a platform at a later time.

Possible values: A5B78912... FE80916F (string in hexadecimal)

Default value: none

Additional notes:

- This configuration parameter is valid in versions from MTX-Tunnel 8.10
- The strings must be in hexadecimal format
- Data is sent according to the period indicated in the LOGGER_serialPeriod parameter. The minimum pause between each of the 10 strings being sent is 2 seconds
- The response from the device connected to the serial port can't be more than 256 bytes

LOGGER_serialPeriod

Description: this specifies how often the modem will send data contained in the LOGGER_serialData1 ... LOGGER_serialData10 parameters.

Possible values: 0 (disabled), 30... 2592000

Default value: 0

Additional notes:

- This configuration parameter is valid in versions from MTX-Tunnel 8.10

  Please consult the examples contained in the Annex for a clearer understanding of this parameter. If you require additional help, contact iotsupport@mtxm2m.com.

LOGGER_https

Description: allows to state if information is sent via http or via https in encrypted format.

Possible values: on, off

Default value: off

Additional notes:

- Keep in mind that if you want to activate this option, you will need a web server with SSL support

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

LOGGER_httpMode

Description: allows to choose the type of communication to be used with a Web Platform. It is possible to choose between “getjson” and “postjson.”

Possible values: getjson, postjson

Default value: getjson

Additional notes:

- Here are some examples of the format of data sent both by means of getjson and postjson methods. Remember that data frames sent by Logger can be very diverse, because Logger is engaged in different internal processes of MTX-Tunnel. The following examples show Logger frames where the I/O of MTX-Tunnel are sent every certain period of time (LOGGER_ioPeriod >0). They belong to ‘IOS” type, but it is possible to receive the following types: “IOS”, “SERIAL”, “TEMP”, “WMBUS”, WAVT”, “WAVSC”, “WAVSV”, “WAVF”, “WAVL”, “POWER”, “MBUS”

  Example of “getjson” mode:

  {“TYPE”:”IOS”,”IMEI”:357042060366409, “P”:”ID00001”, “TS”:”09/08/16 18:32:53”,”IO1” :0,”IO2”:0,”IO3”:0,”IO4”:0, “IO5”:0,”IO6”:0,”IO7”:0,”IO8”:0,”IO9”:0,”IO10”:0,”AD1”:0,”AD2”:0, “CO1”:”0”,”CO3”:”0”}

  Where:

  TYPE: IOS frame type

  IMEI: MTX internal indentification

  P: user’s field in LOGGER_password parameter

  TS: TimeStamp of when data are collected

  IO1: Value of digital input/output 1 of the modem (if it is available)

  IO2: Value of digital input/output 2 of the modem (if it is available)

  IO3: Value of digital input/output 3 of the modem (if it is available)

  IO4: Value of digital input/output 4 of the modem (if it is available)

  IO5: Value of digital input/output 5 of the modem (if it is available)

  IO6: Value of digital input/output 6 of the modem (if it is available)

  IO7: Value of digital input/output 7 of the modem (if it is available)

  IO8: Value of digital input/output 8 of the modem (if it is available)

  IO9: Value of digital input/output 9 of the modem (if it is available)

  IO10: Value of digital input/output 10 of the modem (if it is available)

  AD1: Value of analog input 1 of the modem (if it is available)

AD2: Value of analog input 2 of the modem (if it is available)

CO1: Value of pulse counter input 1 (if it is available)

CO3: Value of pulse counter input 3 (if it is available)

Example of “postjson” mode:

{“IMEI”:357042060366409,”TS”:”09/08/16 18:32:53”,”TYPE”:”IOS”,”IO1”:0,”IO2”:0,”IO3”:
0,”IO4”:0,”IO5”:0,”IO6”:0,”IO7”:0,”IO8”:0,”IO9”:0,”IO10”:0,”AD1”:0,
”AD2”:0,”CO1”:”0”,”CO3”:”0”}

TYPE: IOS frame type

IMEI: MTX internal indentification

P: user’s field in LOGGER_password parameter

TS: TimeStamp of when data are collected

IO1: Value of digital input/output 1 of the modem (if it is available)

IO2: Value of digital input/output 2 of the modem (if it is available)

IO3: Value of digital input/output 3 of the modem (if it is available)

IO4: Value of digital input/output 4 of the modem (if it is available)

IO5: Value of digital input/output 5 of the modem (if it is available)

IO6: Value of digital input/output 6 of the modem (if it is available)

IO7: Value of digital input/output 7 of the modem (if it is available)

IO8: Value of digital input/output 8 of the modem (if it is available)

IO9: Value of digital input/output 9 of the modem (if it is available)

IO10: Value of digital input/output 10 of the modem (if it is available)

AD1: Value of analog input 1 of the modem (if it is available)

AD2: Value of analog input 2 of the modem (if it is available)

CO1: Value of pulse counter input 1 (if it is available)

CO3: Value of pulse counter input 3 (if it is available)

LOGGER_header1, LOGGER_header2, LOGGER_header3

Description: allows to use personalized HTTP headers. Very useful for the communication between MTX-Tunnel and third-party web platforms (like THINGWORKS).

Possible values: Cadena ascii hasta 64 caracteres

Default value: none

Additional notes:

- The headers format is the following: nameHeader;valueHeader. Fields are separated by ";"
- You need them so your MTX-Tunnel to communicate with some generic platforms. For example, to communicate with Thingworks platform you will have to specify something similar to:

  LOGGER_header1: Content-Type;application/json

  LOGGER _header2: Accept;application/json

  LOGGER _header3: appKey;194f5476-7346-4638-ac30-bbca28595be1

LOGGER_mode

Description: it allows to select the mechanism of data sending registered in the internal datalogger.

Possible values: http, mqtt, ftp

Default value: http

Additional notes:

- If you select “mqtt”, remember to use the MQTT_ parameters indicated in this manual
- If you specify “mqtt”, you must also configure the LOGGER_mqttTopic parameter
- If you select “FTP,” you also need to specify the FTP server address, username and password in the following parameters respectively: LOGGER_server, LOGGER_serverLogin, and LOGGER_serverPassword

LOGGER_mqttTopic

Description: configure the topic that MTX-Tunnel will use to publish all data stored in the Logger.

Possible values: String up to 64 characters

Default value: none

Additional notes:

- In the same way as the MQTT_attopicX parameters, you can use the [IMEI] tag in the parameter and the MTX-Tunnel will internally replace that tag with the appropriate value. For example, if your modem has the IMEI 354740050182909 you could specify in the config.txt configuration file the value:

  LOGGER_mqttTopic: [IMEI]/LOGGER

  And the MTX-Tunnel would send logger data using the topic::

  354740050182909/LOGGER

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

8.17 MODBUS configuration parameter: "MODBUS_"

MODBUS_ parameters can configure MTX-Tunnel to take modbus RTU device readings. The Modbus RTU device is connected to a serial port on the modem of MTX Terminals and the internal registers are read.

It is also possible to read and write modbus variables with AT^MTXTUNNEL=GETMODBUS and AT^MTXTUNNEL=SETMODBUS commands.

MODBUS_address

Description: ModBus RTU address parameter specification.

Possible values: 1... 255

Default value: 1

Additional notes:

- Please use the technical support line at iotsupport@mtxm2m.com for further information

  From MTX-Tunnel version 7.12, it is possible to create several addresses in this parameter, meaning several Modbus devices can be read with just one modem. Each address must be separated by a semi-colon ";". For example, if you wish to read the memory table of each device with addresses 1, 2 and 3, this parameter would have the value 1;2;3.

MODBUS_start

Description: this parameter indicates the first position to read from in the Modbus register table.

Possible values: 1... 65535

Default value: 1

Additional notes:

- MTX-Tunnel uses 2 bytes (Word type) variable data. Different sized data need further operations to send JSON object data to your server

  From MTX-Tunnel version 7.12, it is possible to create several starting points in this parameter, meaning several ranges can be read. Each starting point must be separated by a semi-colon ";".

- Please use technical support line at iotsupport@mtxm2m.com for further information

MODBUS_numWords

Description: number of registers to be read from the ModBus device from the address indicated by the MODBUS_start parameter.

Possible values: 1... 20

Default value: 1

Additional notes:

- This parameter is linked together with LOGGER_registerSize, so take the time to adjust it too

  From MTX-Tunnel version 7.18, it is possible to establish several numbers in this parameter, meaning several ranges can be read. Each different number must be separated by a semi-colon ";".

- Please use technical support line at iotsupport@mtxm2m.com for further information

MODBUS_period

Description: this parameter shows after how many seconds the MTX-Tunnel must get the ModBus device to read its memory.

Possible values: 30... 2592000 seconds (from 30 seconds up to 30 days)

Default value: 900 (15 minutes)

Additional notes:

- Please use the technical support line at iotsupport@mtxm2m.com for further information

MODBUS_onlyChanges

Description: this parameter allows you to decide whether to send all readings from a Modbus device (set to “off”) or only send a message when there is a change in at least one of the readings (set to “on”).

Possible values: on, off

Default Value: off (all readings are sent)

Additional notes:

- This parameter is maintained for compatibility reasons. We ask you do not use it however. Instead, please use the parameter MODBUS_changeDiff

MODBUS_readCommand

Description: this parameter allows you to choose the Modbus read command for applications where the MTX-Tunnel acts as modbusmaster for the reading and delivery of modbus records. The command 0x03 is used in most devices, but in others it may be necessary to use the command 0x04.

With this parameter you can choose between 0x03 and 0x04. From MTX-Tunnel v8.07 you can use 0x01 and 0x02 for binary inputs.

Possible values: 1, 2 3, 4

Default value: 3

Additional notes:

- Please write to iotsupport@mtxm2m.com if you need further help
- If you do not know which one to choose, we recommend you specify value 3
- You can specify a different read command for each device to read. You must specify them separated by ";"

MODBUS_logFrequency

Description: this parameter allows you to choose the frequency of data recording in the Logger internal memory, to later deliver the records via GPRS. That is, it is possible to set, for example, a MODBUS_period at 60, which states that the reading period of the Modbus records in seconds will be 60. We can also set, for example, a MODBUS_logFrequency of 3, so one of every 3 records will be internally stored and broadcast via GPRS.

Possible values: 1... 65535

Default value: 1 (storage frequency = MODBUS_period)

Additional notes:

- Please write to iotsupport@mtxm2m.com if you need further help
- This parameter is very useful in scenarios where the MTX-Tunnel must poll several devices. For example, a device can be set with MODBUS_logFrequency "1" and another with MODBUS_logFrequency "5". If we wanted to set a MODBUS_period of 60 seconds, from one of the devices data would be stored and sent every 60 seconds, and every 300 (5x60=300 seconds) from the other one
- This parameter is very similar to the parameter MODBUS_logType. Please consult it in this manual; you will need it to use MODBUS_logFrequency properly

MODBUS_changeDiff

Description: by using this parameter, only the readings of Modbus records in which a change, as specified in MODBUS_changeDiff, has occurred are stored/sent. For example, if you specify a parameter MODBUS_changeDiff with a value of 10 and read the records from address 40 to 45, if one of the records 40 to 45 shows a difference (absolute value) >=10 compared to the last reading (which was stored/sent), then that reading (of records 40 to 45) will be stored in the logger to be later sent via GPRS.

Possible values: 0... 65535

Default value: 0

Additional notes:

- Please write to iotsupport@mtxm2m.com if you need further help
- This parameter is very useful in those scenarios where you only want to store/send data when there are significative changes in one of the variables. For example, imagine you have an application recording temperatures from a modbus sensor. The temperature must be read every 60 seconds, but you don't want to send the records via GPRS if the temperature doesn't change. Then, by setting a MODBUS_changeDiff at "1", for example, you would only send the temperature records when they differ in 1 degree
- This parameter is very similar to the parameter MODBUS_logType. Please consult it in this manual; you will need it to use MODBUS_changeDiff properly

MODBUS_logType

Description: with this parameter you can specify if you are using MODBUS_logFrequency, MODBUS_changeDiff or both. You can choose, for example, to store/send modbus readings from a device every MODBUS_logFrecuency, or to store/send modbus readings from the other device when one of the records changes more than specified in MODBUS_changeDiff v

P000000ossible values: 0 (MODBUS_logFrecuency), 1 (MODBUS_changeDiff), 2 (MODBUS_logFrecuency 000000MODBUS_changeDiff)

Default value: 0

Additional notes:

- Please write to iotsupport@mtxm2m.com if you need further help

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

MODBUS_custom

Description: this parameter adds a new value “C” in the JSON frame. This value is completely arbitrary for the user and can be used for whatever you want. For example, it could be interesting if you want to inform the server that incoming data is binary, word, long...

Possible values: User String

Default value: nothing

Additional notes:

- Please write to iotsupport@mtxm2m.com if you need further help

MODBUS_regType

Description: this parameter allows us to define the type of Modbus register that is to be read automatically. Either “Word” or “DoubleWord” can be chosen.

Possible values: 2 bytes (unsigned word), 4 bytes (unsigned doubleWord)

Default value: 2

Additional notes:

- This parameter is available in versions from MTX-Tunnel v8.08. Before this version, the value “2” (Word) was the only option
- Please write to iotsupport@mtxm2m.com if you need further help

MODBUS_format

Description: this parameter allows configuring the frame format of the modbus data sent by the MTX-Tunnel Datalogger.

Possible values: std, uintframe, hexframe

Default value: std (for reasons of compatibility with previous versions)

Additional notes:

- If the value of the MODBUS_format parameter is defined as “std”, the json frame format sent to the web server or mqtt will be analogous to that shown by the following example, where the modbus data is associated with the “Vx” variables in unsigned integer format

  {“MODB”:

  {“data”:

  {“IMEI”:”353085090011697”,

  “TYPE”:”MODB”,

  “TS”:”2020-03-19T07:21:04Z”,

  “A”:1,

  “ST”:20,

  “N”:4,

  “V1”:0,”V2”:0,”V3”:0,”V4”:0}

  }

  }

- If the value of the MODBUS_format parameter is defined as “uintarray”, the json frame format sent to the web server or mqtt will be analogous to that shown in the following example, where the modbus data is delivered in an array in unsigned integer format

  {“MODB”:

  {“data”:

  {“IMEI”:”353085090011697”,

  “TYPE”:”MODB”,

  “TS”:”2020-03-19T07:38:32Z”,

  “A”:1,

  “ST”:20,

  “N”:4,

  “V”:[10,11,12,13]}

```
        }
}
```

If the value of the MODBUS_format parameter is defined as “hexarray”, the json frame format sent to the web server or mqtt will be analogous to that shown by the following example, where the modbus data is delivered in an array in hexadecimal format.

```
{“MODB”:
        {“data”:
                {“IMEI”:”353085090011697”,
                “TYPE”:”MODB”,
                “TS”:”2020-03-19T07:38:32Z”,
                “A”:1,
                “ST”:20,
                “N”:4,
                “V”:[A,B,C,D]}
        }
}
```

8.18 MODBUSTCP Configuration parameters: "MODBUSTCP_"

From the MTX-Tunnel v9.22 version it is also possible to configure the MTX modems as a modbus slave device. By configuring the MTX modem as a modbus slave, you can write and read registries, read the status of the digital inputs, change the digital outputs and relays, read the analog digital converters, consult the coverage and send any command to the AT Modem (and to read the answer). This means you can perform any task you could do through Telnet, SMS, etc. with Modbus TCP. Only the Modbus commands 0x03 (reading) and 0x10 (writting) are implemented.

@Modbus	Descripción	R/W
1	GPIO0	RW
2	GPIO1	RW
3	GPIO2	RW
4	GPIO3	RW
5	GPIO4	RW
6	GPIO5	RW
7	GPIO6	RW
8	GPIO7	RW
9	GPIO8	RW
10	GPIO9	RW
11	ADC0	R
12	ADC1	R
13-14	COUNTER1	RW
15-16	COUNTER2	RW
17-18	COUNTER3	RW
19	CSQ	R
20	TECH	R

30	RESET	W
31	RESET COUNTER0	W
32	RESET COUNTER1	W
34	RESET COUNTER2	W
35	RESET COUNTER3	W
36	RESET COUNTER4	W
37	RESET COUNTER5	W
38	RESET COUNTER6	W
39	RESET COUNTER7	W
50-59	PASSWORD	RW
98	EXECUTE AT	W
99	STATUS AT EXECUTION	R
100-354	AT COMMAND	RW
500-754	AT COMMAND RESPONSE	R

MODBUSTCP_enabled

Description: by means of this parameter it is possible to activate/deactivate the Modbus TCP service of the MTX-Tunnel. When activated, it will cause the modem to behave like a Modbus Slave device, being able to control it from any Modbus TCP server (PC, SCADA, etc.).

Possible values: on, off

Default value: off

MODBUSTCP_port

Description: configure the TCP port the MTX tunnel will use to receive incoming connections to be used as a Modbus slave.

Possible values: 1... 65535

Default value: 502

Additional notes:

- If you are configuring the MTX-Tunnel as a Modbus TCP gateway to Modbus RTU, do not use the same port number in the TCP_port and MODBUSTCP_port parameters. For example, if for the Modbus TCP to Modbus RTU gateway you use TCP port 502, use TCP port 503 for this service
- Only a simultaneous connection is possible. If an existing connection exists and a new connection is entered, the connection will be accepted by closing the first connection

MODBUSTCP_password

Description: allows you to set a Password to be entered before you can perform a read or write operation.

Possible values: Maximum 10 alphanumeric characters

Default value: none

Additional notes:

- It would be advisable to specify a password to use the MTX-Tunnel as Modbus slave device with a SIM card with public IP address. This way, after making the connection to the port specified in MODBUSTCP_port, you must always enter the password in the registries between addresses 50 and 59
- You can also block non-authorized IP address connections if the firewall is enabled (FIREWALL_parameters)

8.19 CSD: "CSD_" Configuration Parameters

The parameters with the prefix "CSD_" are related to MTX-Tunnel configurations for telemetry scenarios for conventional GSM data calls.

MTXTunnel allows you to receive GSM data calls (CSD) to create a GSM-serial gateway. It also allows you to use both at the same time, giving priority to the GSM calls. It's ideal for meters where you must be able to connect to an electrical meter through a GSM call and via GPRS to obtain continuous readings or readings in real time. Examples of typical energy operators are Endesa, Iberdrola etc.

CSD_enabled

Description: this allows you to either enable or disable GSM data calls. If you set this parameter to "on", the modem will answer GSM data calls after the first ring, establishing a GSM-serial gateway. If at any point the modem answers a GSM data call when a GPRS-serial gateway is already established, the GPRS-serial gateway is "frozen" to make way for the GSM-serial gateway. Once the GSM call is over, the GPRS-serial gateway is re-connected and returns to its previous state.

Possible values: on, off

Default Value: off

Additional notes:

- Use the helpline iotsupport@mtxm2m.com if you need any extra help. Remember that it is only possible to accept CSD calls in GPRS mode, so if you need to accept a CSD call in a 3G modem, you need to configure it to work in 2G mode with GPRS_mode: 2g

CSD_commPort

Description: if the parameter CSD_enabled is set to "on", CSD_commPort allows you to specify if the serial port to use when receiving a GSM call is COM1 or COM2. For example, if we want to receive the call via the DB9 (COM1) in a MTX modem with two serial ports, the value of the CSD_commPort parameter will be "1". If we want to receive the GSM call via the DB15 (COM2) port, then the value to use will be 2.

Parameter valid from version MTX-Tunnel v7.29.

Possible values: 1, 2, 3

Default values: 1

Additional notes:

- It is true we could use the parameter MTX_invertedCom to choose the COM of a CSD call. However, this parameter is indispensable if we need to have a GPRS-Serie gateway via the COM1 and a GSM-Serie gateway via the COM2, for example, to control 2 different devices using a single modem
- Value 3 is special, since it redirects the CSD call to the MTX 2 serial ports. That meas it is possible, for instance, to connect 2 metering boxes to just one MTX modem

CSD_allPhones

Description: with this parameter it is possible to configure if any telephone number is authorized to call the modem for a GSM call (CSD) or only the authorized telephone number can make the call.

Possible values: on, off

Default value: on (all phone numbers are authorized)

Additional notes:

- This parameter with value “on” means that all telephone numbers are authorized. Typical behaviour of any other GSM modem in the market
- This patameter with value “off” means that MTX-Tunnel only allows calls made from authorized telephone numbers, increasing the security of Communications. The authorized telephone numbers must be configured with the parameters CSD_validPhone1 ... CSD_validPhone16
- If a call is received from a no-authorized telephone number, the modem simply hang up the call

CSD_validPhone1, CSD_validPhone2… CSD_validPhone16

Description: when the parameter CSD_allPhones is “off”, there parameters are for indicate the authorized telephone numbers for the reception of a GSM data call (CSD call). You can set up to 16 telephone numbers.

Possible values: A valid telephone number

Default value: nothing

Additional notes:

- The only valid characters for the telephone numbers are the character “+” and “0” … “9”

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

8.20 Digital I/O and relay configuration parameters: "GPIO_"

Most MTX modems have one or more digital inputs and outputs, and even relays. The GPIO_ parameters allow configuring the behavior of these inputs and outputs. For example, you can configure a digital input so that when it changes state, it will send alarm SMS, or send an MQTT message, or make a voice call, or execute an AT command automatically (or a batch of AT commands) or simply configure it as a pulse counting input.

The case of the digital outputs and relays is analogous. It is possible to configure the behavior to activate an output or relay manually by means of an AT command (sent by serial, SMS, modbus, Telnet, MQTT, ...) or to program a time for its activation, or to time the output, or to activate the output if A digital input is activated, or activated if an analog input is out of range, or through a voice call, or based on an astronomical clock, or even based on the value of a Modbus register from a Modbus RTU device that the modem connected to your serial port.

GPIO_mode0, GPIO_mode1, GPIO_mode2, ... , GPIO_mode9

Description: the GPIOs of the MTX modems (the digital inputs and outputs) can be of various types depending on the model of the modem used. On the one hand, a GPIO can only be of type "input", or it can only be of type "output", or it can be configured as "input" or as "output" (that is, it can be configured as input or output).

Through this parameter each GPIO is defined as "input" or as "output". Please refer to the characteristics tables in Annex A of this manual to know the possibilities of each GPIO depending on the MTX modem model.

Possible values: input, output

Default values: depends on the modem model

Additional notes:

- Set the parameter value as "input" if the GPIOx can only be configured as input or if the MTX model allows you to configure the GPIOx as input or output and for the application you want to configure the GPIOx as input
- Set the parameter value as "output" if the GPIOx can only be configured as digital output or if the MTX model allows you to configure the GPIOx as input or output and for the application you want to configure the GPIOx as output

GPIO_config0, GPIO_config1, GPIO_config2, ... , GPIO_config8

Description: using these parameters, you can configure the behavior of each of the modem's GPIOs (digital inputs/outputs). Certain values can be set in the event that the GPIO_modeX parameter is configured as "input" and others in case the GPIO_modeX is configured as "output".

Default values: normal

The values shown below are valid when the GPIOx is configured as "input". Note that many parameters actually consist of several parameters separated by period and as ";".

- normal

  With this value the GPIO is simply configured as a digital input without any special behavior assigned. This value is useful, for example, when you want to simply query the value of a digital input (for example with the command AT ^ MTXTUNNEL = GETIOS)

  Example:

  GPIO_mode3: input

  GPIO_config3: normal

  This example configures GPIO3 as normal input.

- sms;<value>;<timeout>;<mensajeOn>;<mensajeOff>

  sms: text indicating that the GPIO is associated with an SMS alarm

  <value>: 0,1,2

  A 0 indicates that the alarm SMS will be sent when the digital input is not activated

  A 1 indicates that the alarm SMS will be sent when the digital input is activated

  A 2 indicates that the alarm SMS will be sent when the digital input changes state (from enabled to disabled or vice versa)

  <timeout>: 0 ... 3600

  In the event of an SMS alarm condition, this value indicates the maximum time between shipments. This prevents continuous SMS sending if a digital input changes state continuously by accident.

  <mensajeOn>: max. 160 characters (without ;)

  In the event of an activated digital input SMS alarm condition, this is the text of the message to be sent.

  <mensajeOff>: max. 160 characters (without ;)

  In the event of a digital input deactivated SMS alarm condition, this is the text of the message to be sent.

Example:

GPIO_mode3: input

GPIO_config3: sms;2;15;Alarm ON;Alarm OFF

NOTE:

This example configures the GPIO3 as an SMS alarm input and will send an SMS message every time the GPIO3 changes state. It will not send more than one SMS every 15 seconds, and will use the texts “Alarm ON” and “Alarm OFF” depending on the value of the digital input.

The telephone numbers used as recipients of the SMS messages are those configured in the ALARM_smsNumberX parameters (being able to specify up to 10).

- counter

  With this value the GPIO is configured as a pulse counter. The value of the pulse count can be obtained with the command AT ^ MTXTUNNEL = GETCOUNTER, <IDCounter> or send it using the LOGGER_ parameters to a WEB or MQTT server (see example 7.11 for better understanding).

  Example:

  GPIO_mode7: input

  GPIO_config7: counter

  This example configures the GPIO7 as a pulse counter input.

- at;<atOn>;<atOff>

  With this value the GPIO is configured as trigger input of an AT command (or batch of AT commands).

  <atOn>: AT command to be executed when the digital input is activated

  <atOff>: AT command to be executed when the digital input is deactivated

  Example:

  GPIO_mode2: input

  GPIO_config2: at;AT^MTXTUNNEL=EXECUTE,fileon.txt;AT

  NOTE: This example configures GPIO2 as an AT command trigger input. When the input is activated, the command “AT ^ MTXTUNNEL = EXECUTE, fileon.txt” is executed (which in turn executes the AT commands contained in that file, see the description of this AT command for more information). If the digital input is deactivated, the command “AT” is simply executed. See Example 8.13 for a better understanding.

- mqtt;<value>;<timeout>

  With this value the GPIO is configured to send an MQTT message when the state of that GPIO changes.

  mqtt: text indicating that the GPIO is associated with sending MQTT messages

  <value>: 0,1,2

  0 indicates the MQTT message will be sent when the digital input is not activated

  1 indicates that the MQTT message will be sent when the digital input is activated

  A 2 indicates that the MQTT message will be sent when the digital input changes state (from enabled to disabled or vice versa)

  <timeout>: 0 ... 3600 (secs.)

  In the event of an MQTT shipping condition, this value indicates the maximum time between shipments. This prevents continuous MQTT message sending if a digital input changes state continuously by accident.

  Example:

  GPIO_mode3: input

  GPIO_config3: mqtt;2;0

  This example configures GPIO3 as an input associated with sending MQTT messages. A message will be sent every time the GPIO3 input changes state with no timeout between messages. See Example 8.10 for more information and to view the MQTT message delivery format.

  MQTT messages will be sent to the MQTT topic indicated in the MQTT_defaultIOTopic parameter with the Qos specified in MQTT_defaultIOQos.

- call;<value>;<timeout>

  With this value the GPIO is configured to make a voice call (without audio) when the state of said GPIO changes.

  call: text indicating that the GPIO is associated with making a voice call

  <value>: 0,1,2

  0 indicates that the call will be made when the digital input is not activated

  1 indicates that the call will be made when the digital input is activated

  2 indicates that the call will be made when the digital input changes state (from activated to deactivated or vice versa)

  <timeout>: 0 ... 3600 (secs.)

  In the event of a voice call condition, this value indicates the maximum time between calls. This

avoids that if a digital input changes state continuously by accident, continuous voice calls are made.

Example:

GPIO_mode3: input

GPIO_config3: call;1;300

This example configures GPIO3 as an input associated with making a voice call. It will be done every time the GPIO3 input is activated (1) and calls cannot be made more frequently within 5 minutes (300 seconds). See Example 8.8 for more information.

- wakeup;<value>

  The wakeup mode indicates that the GPIO, configured as input, will activate the 4G/3G/2G session for the configured time (in the parameter GPRS_timeout).

  call: text indicating that the GPIO, configured as input, is associated with the start of a 4G/3G/2G session

  <value>: 0,1,2

  0 means 4G/3G/2G session will take place when the digital input is not activated

  1 means 4G/3G/2G session will take place when the digital input is activated

  A 2 indicates that the 4G/3G/2G session will be performed when the digital input changes state (from enabled to disabled or vice versa)

  Example:

  GPRS_timeout: 10

  GPIO_mode0: input

  GPIO_config0: wakeup;1

  This example configures GPIO0 as an input associated with activating the 4G/3G/2G session. When GPIO0 is activated the 4G/3G/2G session will be activated for 10 minutes.

The values shown below are valid when the GPIOx is configured as “output”. Note that many parameters actually consist of several parameters separated by period and as “;”.

- normal

  With this value the GPIO is simply configured as a digital output without any special behavior assigned. This value is useful, for example, when you want to simply activate a digital or relay output (for example with the command AT ^ MTXTUNNEL = SETIO, X, Y).

Example:

GPIO_mode8: output

GPIO_config8: normal

This example configures GPIO8 as normal output. See Example 8.14 for further understanding.

- on

  With this value the GPIO, configured as output, is always activated.

  on: text indicating that the GPIO is an always-on output

  Example:

  GPIO_mode8: output

  GPIO_config8: on

  This example configures GPIO8 as always-on output.

- off

  With this value the GPIO, configured as output, is always disabled.

  off: text indicating that the GPIO is an output always disabled

  Example:

  GPIO_mode8: output

  GPIO_config8: off

  This example configures GPIO8 as always-off output.

- time;<HHon>;<NNon>;<HHoff>;<NNoff>

  Time mode allows you to specify an on/off time for a digital output/relay (remember that you can also use the “schedule.txt” file to enter more sophisticated schedules).

  time: text indicating that the GPIO is a scheduled exit

  <HHon>: time for activation of the output (0,..., 23) UTC

  <NNon>: minutes for output activation (0... 59) UTC

  <HHoff>: time to deactivate the output (0,..., 23) UTC

  <NNoff>: minutes for deactivation of the output (0... 59) UTC

Example:

GPIO_mode4: output

GPIO_config4: time;22;00;8;30

This example configures GPIO4 as a scheduled output to activate at 22:00 UTC and deactivate at 8:30 UTC

- timer

  The timer mode indicates that the output is timed. In other words, by setting the output as a timer, it is possible to use the command AT ^ MTXTUNNEL = SETOUTPUTTIMER, idOutput, Xseconds so that the GPIO idOutput is activated during Xseconds. See Example 8.3 for a better understanding.

  timer: text indicating that the GPIO is a timed exit

  Example:

  GPIO_mode4: output

  GPIO_config4: timer

  This example configures GPIO4 as a timed output. Configured in this way a command like AT ^ MTXTUNNEL = SETOUTPUTTIMER, 4.30 would activate the GPIO4 output for 30 seconds.

- digital;<idGPIO>;<inverted>

  Digital mode indicates that the status of an output will be related to the value of a digital input. That is, it is possible to make the digital output activate when a digital input is activated or vice versa.

  digital: text indicating that the GPIO is an output related to a digital input

  <idGPIO>: 0 ... 9. ID of the digital input to which it is related

  <inverted>: 0,1 0=no inverted, 1=invertied. Inverted implies that the output will activate when the digital input is deactivated, and vice versa

  Example:

  GPIO_mode0: input

  GPIO_config0: normal

  GPIO_mode3: output

  GPIO_config3: digital;0;0

  This example configures GPIO3 as output related to input GPIO0.

  When GPIO0 turns on, the GPIO3 output will turn on, and when GPIO0 turns off, the GPIO3 output will also turn off.

This task (output associated with digital input) can also be carried out with the “at” mode of the digital input. See Examples 8.13 (digital input related output) and 8.16 (analog input related output) for more information.

- modbus;<id>;<command>;<reg>;<minVal>;<maxVal>;<minAT>;<maxAT>

  The “modbus” mode indicates that the status of an output will be related to the value of a modbus register from an external device. That is, it is possible to make the digital output activate when a modbus register is above a value and deactivate below a certain value.

  modbus: text indicating that the GPIO is an output related to a modbus register from an external device

  <id>: 1 ... 254 modbus address of the modbus slave to interrogate

  <command>: 3, 4 modbus read command to use

  <reg>: 0 ... 65535 modbus register number to read

  <minVal>: 0 ... 65535 the output will be deactivated when the registry value is equal to or less than this value

  <maxVal>: 0 ... 65535 the output will be activated when the register value is equal to or greater than this value

  <minAT>: AT command to be executed when the registry value is equal to or below <minVal>

  <maxAT>: AT command to be executed when the registry value is equal to or above <maxVal>

  Example:

  GPIO_mode3: output

  GPIO_config3: modbus;10;3;1;250;300;;

  This example configures GPIO3 as output related to register 1 of the modbus RTU device with address 10 and which is read with the modbus 3 command. The GPIO3 output will deactivate when the register <= 250 and will activate when the register is> = 300. In this example AT commands are not executed.

- call;<at>

  The “call” mode indicates the output is related to a voice call. The output will activate 5 seconds when the modem receives a voice call.

  call: text indicating that the GPIO is an output related to an incoming voice call and will activate 5 seconds upon receiving the call

  <at>: optional AT command to be executed upon receiving the voice call

  Example:

  GPIO_mode8: output

GPIO_config8: call;AT+CSQ

This example configures the GPIO8 as an output which will activate for 5 seconds (fixed value) when a voice call is received in the modem. Additionally, the command “AT + CSQ” will be executed.

- astronomical;<MinutosOffsetOcaso>;<MinutosOffsetOrto>

  The “astronomical” mode indicates the output is related to the modem’s astronomical clock, activating and deactivating itself from it.

  astronomical: text indicating that the GPIO is an output related to the modem’s astronomical clock

  <minutosOffsetOcaso>: offset minutes for sunset

  <minutosOffsetOrto>: offset minutes for ortho

  Example:

  MTX_latitude: 41.6333

  MTX_longitude: 2.36667

  GPIO_mode8: output

  GPIO_config8: astronomical;-30;30

  This example configures the GPIO8 as an astronomical output, configuring an offset of -30 minutes for sunset (the output will be activated 30 minutes before reaching sunset) and 30 minutes for ortho (the output will be deactivated 30 minutes after ortho).

  Note that it is necessary to configure the MTX_latitude and MTX_longitude parameters so that the MTX modem knows how to configure the sunrise and sunset of the day.

8.21 Configuration parameters related to analog inputs: "ADC_"

Most MTX modems have one or more analog inputs (0-50V or 4-20mA). The ADC_ parameters allow you to configure the behavior of these analog inputs. For example, you can configure an analog input so that when it exceeds a level, it will send alarm SMS, or send an MQTT message, or make a voice call, or execute an AT command automatically (or a batch of AT commands) , etc.

We strongly recommend that you take a look at the examples in Annex 8 to better understand the operation.

ADC_mode0, ADC_mode1, ADC_mode2

Description: The ADCs of the MTX modems (analog inputs) can be of various types depending on the model of the modem used. They can be voltage type (0-50V) or current type (4-20mA).

This parameter indicates to the modem how each analog input will be used. See the characteristics tables in Annex A of this manual to find out the possibilities of each ADC depending on the MTX modem model.

Remember that to configure an analog input as voltage or current, in addition to indicating it in this parameter, you must appropriately configure the modem microswitches (see Annex A).

Possible values: voltage, current

Default values: voltage

Aditional notes:

- Set the parameter value as "voltage" if the ADCx wants to be used as voltage input (0-50V). Remember to also configure the modem microswitches for this purpose
- Set the parameter value to "current" if the ADCx wants to be used as current input (0-20mA). Remember to also configure the modem microswitches for this purpose

ADC_config0, ADC_config1, ADC_config2

Description: Using these parameters, you can configure the behavior of each of the modem's ADC inputs (analog inputs).

Default values: normal

- normal

  With this value the ADC input is simply configured as an analog input without any special behavior assigned. This value is useful, for example, when you want to simply query the value of an analog input (for example with the command AT ^ MTXTUNNEL = GETIOS, or with the periodic datalogger, etc.).

  Example:

  ADC_mode0: voltage

  ADC_config3: normal

  This example configures ADC0 as a normal voltage input.

- sms;<minValue>;<maxValue>;<hist>;<timeout>;<messageMin>;<messageMax>;<messageNormal>

  With this value the ADC input is configured to send an SMS alarm based on the ADC value.

  sms: text indicating that the ADC is associated with an SMS alarm

  <minValue>: 0 ... 50000. When the ADC value is equal to or less than this value, an SMS alarm will be generated with the text <messageMin>

  <maxValue>: 0 ... 50000.

  When the ADC value is equal to or greater than this value, an SMS alarm will be generated with the text <messageMax>

  <hist>: 0 ... 50000 hysteresis

  <timeout>: 0 ... 3600

  In the event of an SMS alarm condition, this value indicates the maximum time between shipments. This prevents continuous SMS sending if an analog input changes state continuously by accident.

  <messageMin>: max. 160 characters (without ;)

  In the event of a low value SMS alarm condition, this is the text of the SMS message to be sent.

  <messageMax>: max. 160 characters (without ;)

  In the event of a high value SMS alarm condition, this is the text of the SMS message to be sent.

<messageNormal>: max. 160 characters (without ;)

In the event that the ADC value returns to a normal state after an alarm state, this is the text of the SMS message to be sent.

Example:

ADC_mode0: voltage

ADC_config0: sms;2000;8000;100;60;Alarm ADC0 Low;Alarm ADC0 High;Alarm ADC0 normal

NOTE: This example configures the ADC0 as a voltage input and will send an SMS message whenever the value is below 2000mV or above 8000mV, with a hysteresis of 100mV and a timeout of 60 seconds.

NOTE 2: The recipient's phone numbers are configured in the ALARM_smsNumberX parameters.

- at;<minValue>;<maxValue>;<hist>;<atMin>;<atMax>;<atNormal>

  With this value the ADC input is configured to send an SMS alarm based on the ADC value.

  at: text indicating that the ADC is associated with the execution of an AT command

  <minValue>: 0 ... 50000. When the ADC value is equal to or less than this value, the <atMin> command will be executed

  <maxValue>: 0 ... 50000. When the ADC value is equal to or greater than this value, the <atMax> command will be executed

  <hist>: 0 ... 50000 hysteresis

  <atMin>: in the event of a low value alarm condition, this is the AT command that will be executedá

  <atMax>: in the event of a high value alarm condition, this is the AT command that will be executed

  <atNormal>: in the event that the ADC value returns to a normal state after an alarm state, this is the AT command that will be executed

  Example:

  ADC_mode0: voltage

  ADC_config0: at;2000;8000;0;AT^MTXTUNNEL=SETIO,8,1;AT^MTXTUNNEL= SETIO,8,0;AT

- mqtt;<changeValue>;<timeout>

  With this value the ADC is configured to send an MQTT message when the ADC changes a value to <changeValue>.

  mqtt: text indicating that the ADC is associated with sending MQTT messages

  <changevalue>: 0 .... 50000. When the analog input increases or decreases the configured

value, an MQTT message will be sent

<timeout>: 0 ... 3600 (secs.)

In the event of an MQTT shipping condition, this value indicates the maximum time between shipments. This prevents continuous MQTT message sending if an analog input changes state continuously by accident.

Example:

ADC_mode0: voltage

ADC_config0: mqtt;100;0

This example configures ADC0 as the voltage input associated with sending MQTT messages. A message will be sent every time the ADC0 input changes its 100mV value, with no timeout between messages. See Example 8.11 for more information and to view the MQTT message delivery format.

MQTT messages will be sent to the MQTT topic indicated in the MQTT_defaultIOTopic parameter with the Qos specified in MQTT_defaultIOQos.

- call;<minValue>;<maxValue>;<hist>;<timeout>

  With this value the ADC is configured to make a voice call (without audio) when the value of the ADC is below or above thresholds.

  call: text indicating that the ADC input is associated with making a voice call

  <minValue>: 0 ... 50000. When the ADC value is equal to or less than this value, a voice call will be made

  <maxValue>: 0 ... 50000. When the ADC value is equal to or greater than this value, a voice call will be made

  <hist>: 0 ... 50000 hysteresis

  <timeout>: 0 ... 3600 (secs.). Indicates the minimum time between 2 calls made by the modem

  Example:

  ADC_mode0: voltage

  ADC_config0: call;2000;8000;100;300

  This example configures the analog input ADC0 as associated with making a voice call. It will be carried out every time ADC0 has a value equal to or less than 2000mV or equal to or greater than 8000mV, with a hysteresis of 100mV and a timeout (minimum time between calls) of 300 seconds.

- wakeup;<minValue>;<maxValue>;<hist>

  The wakeup mode indicates that the ADC will activate the 4G/3G/2G session for the configured

time (in the GPRS_timeout parameter) when the analog value is pode below <minValue> or above <maxValue>.

call: text indicating that the GPIO, configured as input, is associated with the start of a 4G/3G/2G session

<minValue>: 0 ... 50000 Cuando el valor del ADC valga igual o sea inferior a este valor, se activará la sesión 4G/3G/2G

<maxValue>: 0 ... 50000. When the ADC value is equal to or greater than this value, the 4G/3G/2G session will be activated

<hist>: 0 ... 50000 hysteresis

Example:

GPRS_timeout: 10

ADC_mode0: voltage

ADC_config0: wakeup;1000;9000;100

This example configures ADC0 as the analog voltage input associated with 4G/3G/2G session activation. When ADC0 <1000mV or ADC0> 9000mV the 4G/3G/2G session will be activated for 10 minutes. A 100mV hysteresis is configured.

8.22 Configuration parameters related to: "LINK_"

The parameters with a LINK_ prefix are those related to the MTX-Tunnel configurations in scenarios where it is intended to monitor / configure the modem through a connection to a web platform in real time.

The MTX-Tunnel, as indicated throughout the manual, allows remote monitoring by AT commands. Either by Telnet, by SMS, Webserver, by a GPRS-RS232 gateway with embedded AT commands etc. With the new LINK_ feature, the MTX-Tunnel creates a permanent socket in TCP Client mode dedicated exclusively to this option. It is very useful, since being the TCP Client type connection, it does not depend on whether the modem has a SIM with Public IP.

See the examples in this manual for a better understanding. Remember also that MTX-Tunnel supports MQTT, so starting a web platform from scratch, it is recommended to use MQTT.

LINK_enabled

Description: It allows enabling or disabling the LINK socket for remote real-time supervision of a modem with MTX-Tunnel.

Possible values: on, off

Default values: off

Aditional notes:

- See the examples in the Annex for a better understanding of this parameter

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

LINK_IP

Description: If the LINK_enabled parameter is “on”, this parameter allows specifying the IP or DNS address to which the MTX-Tunnel modem will connect for Real Time monitoring.

Valid parameter from MTX-Tunnel version v9.06.

Possible values: IP or DNS (example: 1.2.3.4 at ip.mydomain.com)

Default values: none

Aditional notes:

- To use the LINK option, it is not necessary that the IP of the modem’s SIM is neither static nor public, since it will be the modem that initiates the connection. But it is recommended that the IP address to which the modem connects (the one specified in LINK_IP) is fixed

LINK_port

Description: If the LINK_enabled parameter is “on”, this parameter allows you to specify the TCP port to which the MTX-Tunnel modem will connect for Real Time monitoring.

Valid parameter from MTX-Tunnel version v9.06.

Possible values: 1... 65535

Default values: 20023

LINK_retryPeriod

Description: When connectivity to the IP address specified in LINK_IP is lost, this time indicates the time (in seconds) to pause before starting a new connection again.

Valid parameter from MTX-Tunnel version v9.06.

Possible values: 0... 3600

Default values: 30

Aditional notes:

- It is recommended to set a minimum time of 30 seconds at least. Otherwise, in the event of a remote server crash (the one corresponding to the LINK_IP IP), the modem will be continually retrying to establish a socket, with the consequent consumption of data. Take special care with this option if you do not have a SIM card with a flat data rate

LINK_timeout

Description: Indicates the seconds that must pass without data being received through the LINK socket to close the socket and reestablish itself.

Valid parameter from MTX-Tunnel version v9.06.

Possible values: 0... 3600

Default values: 1800

Aditional notes:

- This parameter allows you to resolve loss of connectivity problems against the remote control server. Sometimes a sudden shutdown of a connection server can occur (for example, a power loss, a blockage ...) in such cases a socket closure notice is not sent to the connected devices (in this case the MTX-Tunnel modems). If a closed socket indication is not received, the MTX-Tunnel will understand that the socket is established and will not establish it again. To resolve this situation, if no data is received in LINK_timeout seconds, the LINK socket closes and reopens, thus ensuring connectivity again

LINK_keyId

Description: Allows you to configure a text string to identify the equipment. This will be the first information sent by the MTX-Tunnel through the LINK socket after establishing communication with the maintenance server.

Parámetro válido desde la versión MTX-Tunnel v9.06.

Possible values: ASCII string up to 64 characters

Default values: none

Aditional notes:

- Use this parameter on your server as it will help you to identify the modem among all connected modems. It is interesting to use a KEYID instead of the IMEI of the modem, because in case of substitution of a faulty modem, ... you will only have to replace the faulty modem with a new one with the same configuration without having to make any changes to your server. maintenance, even though the IMEI of the modem is different

LINK_ssl

Description: Indicates if the LINK socket to establish with the maintenance server should be SSL or not.

Valid parameter from MTX-Tunnel version v9.06.

Possible values: 0, 1 (0=no, 1=yes)

Default values: 0

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

8.23 TEMPERATURE configuration parameters: "TEMPERATURE _"

Parameters with the prefix TEMPERATURE_ are related to MTX-Tunnel configurations in scenarios where a MTX-TEMP-RS232 temperature sensor is expected to be controlled. The sensor can be connected to a RS232 serial port of the modem to control the temperature indoors. It can send temperature information to a web server, send an SMS alarm when the temperature is beyond the limits, as well as send a notification to a web server.

TEMPERATURE_enabled

Description: allows to enable the control of the temperature sensor MTX-Temp-RS232.

Possible values: on, off

Default value: off

Additional notes:

- Consult iotsupport@mtxm2m.com for additional help.

TEMPERATURE_period

Description: when enabled, Logger allows you to set the interval in minutes in which you want to store the temperature value in the internal Logger of MTX-Tunnel, in order to send it later to your Web Server.

Possible values: 0... 1140 (minutes)

Default value: 10

Additional notes:

- The value “0” means that you do NOT want to store temperature information in the internal Logger in order to send them to a Web server. Useful only when it is necessary to send an alarm message either via SMS or to your Web server without recording/sending temperature data, if it is within the limits
- Parameter valid starting MTX-Tunnel v9.11

TEMPERATURE_max

Description: parameters to set a higher temperature limit at which, when exceeded, a HIGH TEMPERATURE alarm should go off.

Possible values: -25... 50

Default value: 50

Additional notes:

- When the temperature read by the sensor is higher than permitted, an alarm will be sent. If MTX-Tunnel Logger is configured (LOGGER_enabled: on) , an alarm notification in JSON format will be sent to your web server. You can find the JSON format at the end of the page
- Remember to configure the parameter TEMPERATURE_threshold.You can establish a hysteresis in order to avoid alarm notifications being sent continuously when the temperature is close to the alarm limit with small oscillations
- One alarm notification will be sent when the maximum temperature limit is overpassed and another one when the temperature is within the established limits
- In case of the sensor not being detected (for example when the sensor disconnects accidentally, intentionally, it is broken...) an alarm will be produced, too

{“IMEI”:353234028104337,”TS”:”08/06/14 13:39:33”,”P”:”ID-12345678”,”TYPE”:”TEMP”, ”TEM”:40.5,”TEMH”,1,”TEML”,0,”TEME”,0}

Where:

IMEI: the unique identification number of the modem

TS: TimeStamp of when data sent was generated

P: identification field in LOGGER_password

TYPE: “TEMP” -> indicates that JSON sent contains temperature data

TEM: temperature read

TEMH: 1= High temperature alarm ON; 0= High temperature alarm OFF

TEML: 1= Low temperature alarm ON; 0= Low temperature alarm OFF

TEME: 1= Sensor NOT detected alarm ON; 0= Sensor NOT detected OFF

TEMPERATURE_min

Description: parameters to establish a lower temperature limit at which, when exceeded, a LOW TEMPERATURE alarm should go off.

Possible values: -25... 50

Default value: 0

Additional notes:

- When the temperature read by the sensor is lower than permitted, an alarm will be sent. If MTX-Tunnel Logger is configured (LOGGER_enabled: on) , an alarm notification in JSON format will be sent to your web server. You can find the JSON format is at the end of the page
- Remember to configure the parameter TEMPERATURE_threshold.You can establish a histeresis in order to avoid alarm notifications being sent continuously when the temperature is close to the alarm limit with small oscillations
- One alarm notification will be sent when the minimum temperature limit is overpassed and another one when the temperature is within the established limits
- In case of the sensor not being detected (for example when the sensor disconnects accidentally, intentionally, it is broken...) an alarm will be produced, too

{“IMEI”:353234028104337,”TS”:”08/06/14 13:39:33”,”P”:”ID-12345678”,”TYPE”:”TEMP”, ”TEM”:40.5,”TEMH”,0,”TEML”,1,”TEME”,0}

Where:

IMEI: the unique identification number of the modem

TS: TimeStamp of when data sent was generated

P: identification field in LOGGER_password

TYPE: “TEMP” -> indicates that JSON sent contains temperature data

TEM: temperature read

TEMH: 1= High temperature alarm ON; 0= High temperature alarm OFF

TEML: 1= Low temperature alarm ON; 0= Low temperature alarm OFF

TEME: 1= Sensor NOT detected alarm ON; 0= Sensor NOT detected OFF

8.24 GPS Configuration parameters: "GPS_"

Parameters with the prefix GPS_ are related to MTX-Tunnel configurations to be used in MTX devices (only for those disposing of GPS).

By using an MTX with incorporated GPS module you can configure the modem to send the location to a WEB server periodically. Unlike the former versions of MTX-Tunnel firmware, from the version v9.18 onwards it is possible to store GPS locations in the internal Logger to avoid location data loss when there is no 4G/3G/2G coverage. The locations will be sent later when 4G/3G/2G coverage is there again.

GPS_period

Description: allows to set a time period in seconds in which GPS locations are collected in order to be stored in the Logger and later sent to a WEB server.That is to say, if you set the value 60, the modem will collect one GPS location per minute and send it in JSON format to your Web server.

Possible values: 0, 30... 2592000 (seconds)

Default value: 0 (not activated)

Additional notes:

- Despite the fact that it is possible to set a value permitting one reading every 30 seconds, the minimum value of 60 seconds is recommended, especially for areas with poor 4G/3G/2G coverage.

GPS_mode

Description: allows to specify if the data collected from GPS should be stored in the internal LOGGER_ for its further transmission to a server or they should be sent via a socket to specified address and TPC port.

Possible values: logger, socket, fasthttp

Default value: logger

Additional notes:

- By setting the value “logger” each GPS received (each “GPS_period” seconds) is stored in the internal LOGGER_. This way information is stored in flash memory to be sent in real time or later if there is not 4G/3G/2G coverante in a concrete moment. The “logger” option is the most properly way to implement a fleet control due to the GPS positions are not lost if the 4G/3G/2G network is missed for some time
- By setting the value “socket” MTX-Tunnel creates a TCP client socket to the configured IP in GPS_ip parameter and TCP port configured in GPS_port. This way every GPS_period seconds the current GPS location will be sent. This is the quickest way to send a location.Unlike the “logger” method locations are sent in real time,i.e. they are not stored in the internal LOGGER_. Useful for systems requiring real-time location
- Setting the value “fasthttp” is the quickest method of sending data through HTTP. You can establish one location transmission every 10 seconds approximately. In this mode LOGGER cannot be used for other tasks (GPIOs reading, Modbus values...)

GPS_ip

Description: in case of using “socket’ mode in GPS_mode, this parameter allows to specify the sending IP address of a GPS location.

Possible values: xxx.xxx.xxx.xxx (dirección IP o DNS)

Default value: none

Additional notes:

- This parameter is only taken into consideration if GPS_mode parameter is configured in “socket” mode. If “logger”or “fasthttp” mode is used, the sending IP address is specified in LOGGER_server, as usual

GPS_port

Description: in case of using “socket” mode in GPS_mode, this parameter allows to specify TCP port of GPS location transmission.

Possible values: 1... 65535

Default value: 20010

Additional notes:

- This parameter is only taken into consideration if GPS_mode parameter is configured in “socket” mode. If “logger”or “fasthttp” mode is used, the sending port is specified in LOGGER_server, as usual

8.25 SNMP configuration parameters: "SNMP_"

Configuration parameters with the prefix SNMP_ are related to MTX-Tunnel configurations where SNMP protocol is necessary. SNMP (Simple Network Management Protocol) is a very extended protocol serving for control of field-installed modems.

From the version MTX-Tunnel v9.16 onwards it is possible to use SNMP with all MTX modem models containing MTX-Tunnel firmware. SNMPv2 protocol is not fully included, but it does include the 2 most important functions GET and SET. By means of GET command it is possible to know the state of the modem installed in field (coverage, functioning time, operator...), as well as to read its configuration parameters. By means of SET command it is possible to change the modem's configurations remotely (serial port speed, apn...).

It is very important to know that via SNMP it is possible to execute any AT command in the modem in the same way we would it through Telnet, Webserver, SMS... This allows to carry out any remote task with SNMP, including reset, configuration changes, firmware update...

SNMP_enabled

Description: enable/disable is SNMP service in MTX-Tunnel.

Possible values: on, off

Default value: off

Additional notes:

- In case you do not need SNMP service, do not activate it in your application in order to avoid unnecessary CPU resources consumption

SNMP_port

Description: UDP listening port to receive GET/SET requests from SNMP protocol.

Possible values: 1... 65535

Default value: 161

Additional notes:

- It is recommended to maintain UDP 161 port, because this is the standard of SNMP protocol
- If you use MTX-Tunnel for a UDP-RS232 gateway, do not use the same UDP port for both services (for the gateway and for SNMP). Use different ports

SNMP_community , SNMP_communityW

Description: with these parameters you can configure passwords for reading (SNMP_community) and writing (SNMP_communityW) operations respectively.

Possible values: A text string up to 127 characters

Default value: public

Additional notes:

- Check SNMP_auth and SNMP_password parameters if you want additional security for SNMPv2

SNMP_auth

Description: this parameter allows you to increase the security of SNMP version 2. It basically allows you to enable or disable SNMP service by writing in an OID SNMP. This way it is possible to enable SNMP service only when it is necessary to conduct an operation.

Possible values: std, otp

Default value: std

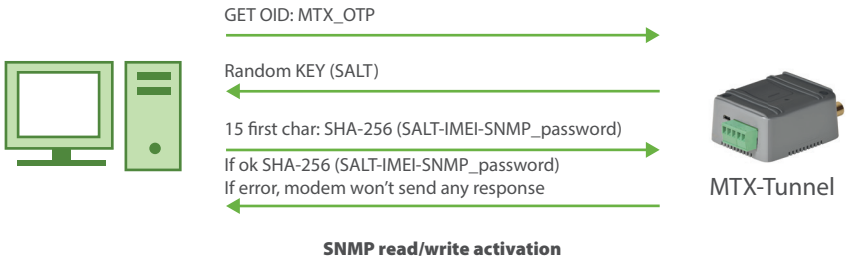
Additional notes:

- When choosing the option "std", the authentication process is a standard process of SNMPv2 communication, using SNMP_community and SNMP_communityW parameters
- When choosing the option "otp" SNMP can be enabled/disabled voluntarily. The process is as follows:

  1. OID MTX_OTP (.1.3.6.1.4.1.45711.2.1.10.3) is read. It returns a random value as a response (SALT)

  2. HASH is calculated, using SHA-256 from: SALT-IMEI-SNMP_password. The first 15 HASH characters are written in OID MTX_OTP. At this moment SNMP service is enabled for reading/writing options

  3. When finished, 0 should be written in OID MTX_OTP to suspend SNMP service



SNMP read/write activation

- Remember that you can also activate the firewall from the modem (FIREWALL_enabled: on) to be able to access SNMP only from authorized IPs (FIREWALL_IPx), thus providing a higher security level
- Finally, keep in mind that it is also possible to keep SNMP service deactivated and activate any moment via SMS from an authorized phone number

SNMP_password

Description: this parameter allows you to increase the security of SNMPv2. In case SNMP_auth parameter is configured as “otp”, it should be used to complete HASH SHA-256 that should be returned to the modem to enable SNMP service.

Possible values: 0... 32 chars

Default value: none

Additional notes:

- See SNMP_auth parameter for more information

8.26 MQTT Related configuration parameters: "MQTT _"

The MTX-Tunnel firmware has MQTT support as of version 9.25. It is possible to send data stored in the internal datalogger (LOGGER_mode: mqtt) using MQTT as well as the DNS data (DNS_mode: mqtt). It is also possible to send AT commands (to check modem status, make configuration changes, change relay status, etc.).

The main advantage is that being a standard MQTT protocol you can use a large number of applications that exist for your Android or IOS mobile. All regardless of whether the SIM card used has or does not have a public or private IP address.

MQTT_enabled

Description: enables the MQTT service of the MTX-Tunnel. It is imperative to enable it in case you want to use this service for Logger, DNS or AT commands.

Possible values: on, off

Default value: off

MQTT_server

Description: any MQTT device acting as a “client” must be connected to an MQTT broker. Here you must specify the connection URL or IP as well as the connecting TCP port.

Possible values: tcp://xxx.xxx.xxx.xxx:puerto or ssl://xxx.xxx.xxx.xxx:puerto

Default value: none

Additional notes:

- An example url for an MQTT broker can be tcp: //2.3.4.5: 1883
- On the Internet there are many free and paid MQTT broker services. For example you can test for your tests:

  tcp://test.mosquitto.org:1883

MQTT_id

Description: each MQTT device that communicates with an MQTT Broker should have a different identifier. By means of this property it is possible to establish said identifier.

Possible values: String up to 32 characters

Default value: none

Additional notes:

- If you use the same equipment identifier in two devices, the broker will disconnect one of them, because only one unique ID is allowed

MQTT_login

Description: some MQTT Broker services require that the client device be authenticated using a username and password system. By means of this parameter you can specify, if necessary, the username.

Possible values: String up to 128 characters

Default value: none

Additional notes:

- If your MQTT broker does not specify it, do not use this parameter

MQTT_password

Description: some MQTT Broker services require that the client device be authenticated using a username and password system. By means of this parameter you can specify, if necessary, the password.

Possible values: String up to 128 characters

Default value: none

Additional notes:

- If your MQTT broker does not specify it, do not use this parameter

MQTT_attopic1, MQTT_attopic2, MQTT_attopic3

Description: with these 3 parameters, you can specify the topics to which the MTX Tunnel will subscribe in order to receive AT commands from an application with MQTT (for example to be able to send AT commands from a mobile phone).

Possible values: String up to 128 characters

Default value: none

Additional notes:

- For example, an appropriate value could be:

  MQTT_attopic1: midispositivo/comandosAT/mtx1

  This would imply that all the AT commands that you send (for example from a mobile phone) to the topic device/commandAT/mtx1 of the Broker to which the modem is connected will arrive to the MTX-Tunnel and therefore will be executed.

- If you include the [IMEI] tag it will be replaced by the actual IMEI of the modem. It will be very useful in the case of having a significant number of MTX-Tunnel devices. For example, you could configure:

  MQTT_attopic1: [IMEI]/AT

  And the MTX-Tunnel would subscribe to the MQTT broker with the appropriate topic (eg 354740050182909/AT). This way you could put the same configuration file in all the devices.

- Up to 3 tracks available for AT commands, so you can specify groups. That is, if you wanted to change the configuration of all the MTX-Tunnel modems located in Barcelona, you could do it by configuring them with the following command:

  MQTT_attopic2: grupo/Barcelona

  So every time you send an ATcommand from your MQTT application (for example, from your mobile phone) to the MQTT broker with the topic “group/Barcelona,” any MTX-Tunnel subscribed to that topic would execute that same AT command.

  You could also configure the third topic (MQTT_attopic3) to execute an AT command on all MTX-Tunnel devices, as long as you set the MQTT_attopic3 parameter identically in all the config.txt files in all the MTX-Tunnel.

MQTT_atrtopic

Description: this parameter allows defining the topic that the MTX-Tunnel will use to send the responses to executed AT commands. That is, if the MTX-Tunnel receives an AT command through the topics MQTT_attopic1, MQTT_attopic2 or MQTT_attopic3, the answer will be returned in this topic.

Possible values: String up to 128 characters

Default value: none

Additional notes:

- If you are going to use MQTT_attopi1, MQTT_attopi2, MQTT_attopi3 to send AT commands to the MTX-Tunnel you must specify a topic in MQTT_atrtopic for the response. Obviously you must subscribe to this topic on the device that you use to send AT commands in order to receive such a response
- Analogously to the above parameters, it is possible to include the [IMEI] tag and the MTX-Tunnel will replace that tag with the corresponding actual value. For example:

  MQTT_atrtopic: [IMEI]/ATResponse
- If your MQTT application allows you to send a message (AT command) with a given ID (message identifier mqtt), the MTX-Tunnel will respond to the AT command with the same ID

MQTT_qos

Description: this parameter allows to establish the quality of service of the MQTT communications used.

Possible values: 0, 1, 2

Default value: 1

Additional notes:

- The value 0 implies that the message will be sent once and there will be no response message to check if the message arrived correctly or not to the Broker
- A value of 1 ensures that the message reaches the MQTT broker at least once (although it may arrive more often)
- The value 2 ensures that the message will reach the MQTT broker once and only once
- MQTT_qos is just used with LOGGER. DNS and answers to AT commands sent via MQTT use qos 0

MQTT_keepalive

Description: it is possible to define the seconds of keepalive with this parameter, that is, the seconds of waiting for the client to make a connection with the MQTT Broker to verify that the connection is still established.

Possible values: 10... 3600

Default value: 300 (5 minutes)

Additional notes:

- It is highly recommended to set a reasonable value for that value, such as 300. If you specify a maximum value of 1 hour you should keep in mind that in the event of a loss of connectivity with the mqtt broker (without socket closing signal) you could keep MTX-Tunnel disconnected from it during the same time period

MQTT_persistent

Description: this parameter allows to establish the persistence of the data for the data sent by MQTT from the LOGGER_ or by the DNS_ frames.

Possible values: on, off

Default value: off (non-persistent data)

Additional notes:

- Let's say you are sending a DNS_ frame via MQTT each time a digital input changes. To do this, you have configured the DNS_mqttTopic: [IMEI] / DNS parameter to send a JSON with the modem I / O data to that broker topic each time a digital input changes. If the persistence is "off" the data (actually the last data received) are NOT stored in the broker. That means, for example, if you are going to query the MTX-Tunnel status (I/O) from a mobile phone with an MQTT client, you would only know the current MTX-Tunnel I/O status if the MTX-Tunnel sent the data to the broker MQTT you had connected the mobile phone to the broker at that time
- If you want the MQTT broker, when you connect with the mobile phone or your application, to send you the last status received from the MTX-Tunnel, simply set this parameter on

MQTT_filetopic1

Description: this parameter allows to send a file to the modem via MQTT. The modem will subscribe to this broker TOPIC to receive files. This way, any file sent from this topic will be downloaded by the modem. It can be useful to change completely the configuration file config.txt for instance.

Possible values: Text up to 128 characters

Default value: none

Additional notes:

- The format we need to send the file to this TOPIC so it's received and processed by the modem is this:

  path/nombreFichero,<array de bytes>

- For instance, to change the configuration file "config.txt," it will be similar to:

  config.txt,COMM_baudrate: 9600[0x13][0x10]COMM_bitsperchar: 8[0x13][0x10]...

  (where [0x13] and [0x10] represent hexadecimal values of ENTER after each parameter in the configuration file)

- Check the parameter MQTT_filertopic to know the answer from the modem to this topic
- This parameter is only available from MTX-Tunnel 9.39 on

MQTT_filertopic

Description: this parameter allows to configure the mqtt TOPIC the modem will use to indicate if the reception of a file (via TOPIC MQTT_fileTopic1) was correct or not.

Possible values: Text up to 128 characters

Default value: none

Additional notes:

- The format the modem will send the answer after receiving a file using this TOPOC is this:

  nameFile,OK (in case the file is transferred correctly)

  nameFile,ERROR (in case there is an error)
- Check the parameter MQTT_filetopic1 to know how to send files to the modem
- This parameter is only available from MTX-Tunnel 9.39 on

MQTT_commrxtopic

Description: this parameter allows to configure the mqtt TOPIC to which the modem will subscribe to. All data the modem receives will be resent through the mail serial port (the one configured with the COMM_ parameters).

Possible values: Text up to 128 characters

Default value: none

Additional notes:

- To use this parameter we need to configure additionally the modem with the parameter MTX_ mode: mqtt
- All the bytes received on this topic will be resent through the main serial port of the MTX modem. If we need to resend them through the secondary port (RS485 in most MTX modems) we need to configure the parameter MTX_invertedCom: on
- This parameter is only available from MTX-Tunnel 10.09 on

MQTT_commtxtopic

Description: this parameter allows to configure the mqtt TOPIC to which the modem will send all data received through its main serial port (the one configured with COMM_ parameters).

Possible values: Text up to 128 characters

Default value: none

Additional notes:

- To use this parameter we need to configure additionally the modem with the parameter MTX_mode: mqtt
- All the bytes received on the main serial port of the MTX modem will be resent via MQTT to the topic configured on this parameter. If we need to use the secondary port (RS485 in most MTX modems) we need to configure the parameter MTX_invertedCom: on
- Remember we have the MTX_msToSend parameter. It may be necessary to use it if you need to receive the data in your MQTT platform in just one package (without any fragmentation)
- This parameter is only available from MTX-Tunnel 10.09 on

MQTT_defaultIOTopic

Description: this parameter allows the TOPIC mqtt to be configured for rapid change sending on digital and analog inputs/outputs, when the parameters "GPIO_configX" or "ADC_configX" have the value "mqtt".

Possible values: Text up to 128 characters

Default value: none

Additional notes:

- This parameter is only available from MTX-Tunnel 11 on

MQTT_defaultIOQos

Description: this parameter allows configuring the Qos mqtt for fast change sending on digital and analog inputs/outputs, when the parameters “GPIO_configX” or “ADC_configX” have the value “mqtt”.

Possible values: 0,1, 2

Default value: 0

Additional notes:

- This parameter is only available from MTX-Tunnel 11 on

8.27 TACACS+ related configuration parameters: "TACACS_"

The MTX-Tunnel firmware has TACACS + authentication support as of version 9.26. That is, it is possible to use a Tacacs + server to authenticate a user who wants to use the Telnet or SNMP service.

It will be necessary to configure the TELNET_auth or SNMP-auth parameter with the "tacacs +" option in order to use this functionality.

TACACS_server

Description: it will be the tacacs server + to use in case the Telnet or SNMP service authentication mode, is this method.

Possible values: IP or DNS

Default value: none

TACACS_port

Description: specifies the TCP port that the MTX-Tunnel will use for the Tacacs + authentication service.

Possible values: 1... 65535

Default value: 49

Additional notes:

- The same TCP port used in the TCP_port and / or WEBSERVER_port and / or TELNET_port parameters should not be used

TACACS_key

Description: KEY used for the Tacacs + authentication service. This KEY is shared by the Tacacs +.

Possible values: A maximum string of 32 characters

Default value: none

8.28 Configuration parameters related to WAVENIS: "WAVENIS_"

The parameters with the WAVENIS_ prefix are those related to the MTX-Tunnel configurations in telemetry scenarios via RF 868/915 MHz

The WAVENIS_ parameters are only to be used when the MTX-Tunnel is running on an MTX-4G-JAVA-IOT-STD-N-WC868 or MTX-4G-JAVA-IOT-STD-N-WC915 platform.

WAVENIS_mac1, WAVENIS_mac2, ... WAVENIS_mac32

Description: This parameter allows you to specify the MAC addresses of the Wavenis devices to read. Specify the MAC address of Waveflow devices (pulse counters with radio communications). The MAC addresses specified in this parameter will be the devices that will be periodically read by the MTX-Tunnel.

It also allows you to specify the repeaters that, if applicable, will be used.

Possible values: AABBCCDDEEFF;XX;YY;ZZ

AABBCCDDEEFF: MAC address wavenis valid

XX;YY;ZZ: 1,2, ... 10 indicating, in order, the repeaters to use

Default value: none

Additional notes:

- See the examples in Annex 5 for a better understanding of this parameter. Check with iotsupport@mtxm2m.com if you need additional help

WAVENIS_rep1, WAVENIS_rep2, ... WAVENIS_rep16

Description: Allows you to specify the MAC of up to 16 repeaters to be used in waveni communications.

Possible values: AABBCCDDEEFF MAC address wavenis valid

Default value: none

Additional notes:

- See the examples in Annex 5 for a better understanding of this parameter. Check with iotsupport@mtxm2m.com if you need additional help

8.29 Configuration parameters related to IEC870-5-102: "IEC102_"

The IEC102_ configuration parameters are only available in the "MTX-Tunnel IEC870", it is a product that is supplied separately.

The parameters with an IEC102_ prefix are those related to the MTX-Tunnel configurations in meter telemetry scenarios that use the IEC870-5-102 protocol.

The MTX-Tunnel has part of the IEC870-5-102 protocol implemented inside, which allows it to read certain parameters from one or more meters with the IEC870-5-102 protocol ... in real time, to be sent to an HTTP platform / MQTT and being an employer for energy efficiency companies. These parameters are the following:

- Instantaneous values of energy and power:

Active absolute energy (VabA)	Power Factor Phase 2 (thousandths)
Inductive Reactive Absolute Energy (VabRi)	Active Power Phase 3 (KW)
Capacitive Reactive Absolute Energy (VabRc)	Reactive Power Phase 3 (KVA)
Total Active Power (KW)	Power Factor Phase 3 (thousandths)
Total Reactive Power (KVA)	Current Phase 1 (tenths of an amp)
Total Power Factor (thousandths)	Voltage Phase 1 (tenths of a volt)
Active Power Phase 1 (KW)	Phase 2 current (tenths of an amp)
Reactive Power Phase 1 (KVA)	Phase 2 Voltage (tenths of a volt)
Power Factor Phase 1 (thousandths)	Current Phase 3 (tenths of an amp)
Active Power Phase 2 (KW)	Voltage Phase 3 (tenths of a volt)
Reactive Power Phase 2 (KVA)	

- Closing tax. Parameters referring to the memorized pricing information (Contract I) of the closing readings:

ELEMENT	TYPE	BYTES
Absolute energy A [A + / A-]	Whole without a sign	4
Incremental energy A [A + / A-]	Whole without a sign	4
Qualifier A	8 bit array	1
Absolute energy Ri [Ri + / Ri-]	Whole without a sign	4
Incremental energy Ri [Ri + / Ri-]	Whole without a sign	4
Qualifier Ri	8 bit array	1
Absolute energy Rc [Rc + / Rc-]	Whole without a sign	4
Incremental energy Rc [Rc + / Rc-]	Whole without a sign	4
Rc qualifier	8 bit array	1
Reserve 7	To define	4
Qualifier 7	8 bit array	1
Reserve 8	To define	4
Qualifier 8	8 bit array	1
Maximum power A [A + / A-]	Whole without a sign	4
Maximum power A date	Label time a	5
Maximum qualifier A	8 bit array	1
Power excesses A [A + / A-]	Whole without a sign	4
Excess qualifier	8 bit array	1
Start of period	Label time a	5
End of period	Label time a	5

IEC102_meter1, IEC102_meter2, ... IEC102_meter16

Description: This parameter allows to specify the basic configuration of each IEC870-5-102 meter to be read, specifying an identifier, the relay address, the measurement point address and the access password.

This parameter must be specified by separating the parameters by; (semicolon).

Possible values: <ID>;<link address>;<measured point direction>;<password>

<ID> : User counter identifier. Sring. Maximum 16 characters

<Link address>: 1 ... 65535

<Measured point direction>: 0 ... 65535

<Password>: 0 ... 4294967295

Default value: none

Additional notes:

- See the examples in Annex 7 for a better understanding of this parameter. Check with iotsupport@mtxm2m.com if you need additional help

IEC102_period

Description: This parameter allows you to specify the reading period of the counters. It is expressed in minutes. The specified number must be a divisor of 1440 (1440 minutes are the minutes of 1 day). For example, if you specify 15, the reading process will start at XX: 00, XX: 15; XX: 30, XX: 45.

Possible values: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 30, 32, 36, 40, 45, 48, 60, 72, 80, 90, 96, 120, 144, 160, 180, 240, 288, 360, 480, 720, 1440

Default value: 15

Additional notes:

- See the examples in Annex 7 for a better understanding of this parameter. Check with iotsupport@mtxm2m.com if you need additional help

IEC102_attempts

Description: This parameter allows you to specify the maximum number of attempts to read a counter in the event of an error.

Possible values: 1... 10

Default value: 3

Additional notes:

- See the examples in Annex 7 for a better understanding of this parameter. Check with iotsupport@mtxm2m.com if you need additional help

8.30 Configuration parameters related to DUAL SIM: "DUALSIM_"

The parameters with a DUALSIM_ prefix are those related to the MTX-Tunnel configurations installed in MTX modem models that have the DUALSIM feature. The fact of having dual SIM in the modem has the following advantages depending on the modem model:

- It is not mandatory to use two SIMs. Only a single SIM card slot can be used. In some modems, such as the MTX-IOT-S family, it allows choosing between an external SIM slot (easily accessible from outside the modem) or an internal SIM slot, where to access the SIM it is necessary to open the modem housing , making it difficult for third parties to access
- It is possible to use 2 simultaneous SIMs. The MTX-Tunnel in this case allows you to change from one to another in the event of a fall of one of these, which provides greater security against falls of a certain telephone operator or even due to the deterioration of a SIM card.

DUALSIM_select

Description: This parameter allows you to specify which SIMs to use. That is, if only the main SIM, the secondary SIM, and both SIMs must be used, in the latter case it is possible to specify which of the two SIMs is the first to be used after the modem has been reset.

Possible values:

sim: uses only the main SIM of the modem (associated with the parameters GPRS_apn, GPRS_login, GPRS_password y GPRS_dns)

sim2: use only the secondary SIM of the modem (associated with the parameters GPRS_apn2, GPRS_login2, GPRS_password2 y GPRS_dns2)

dual: it uses the primary SIM of the modem and the secondary SIM. The first to be used after modem startup is the main SIM. Afterwards, the secondary SIM is used in case of problems with the main SIM

dual2: it uses the primary SIM of the modem and the secondary SIM. The first to be used after the modem starts up is the secondary SIM. Afterwards, the main SIM is used in case of problems with the secondary SIM

Default value: sim

Additional notes:

- This parameter should only be used with DUALSIM models
- The modem enters configuration mode when it starts up without a SIM card inside:
  - If you use "sim" mode, the main SIM card mustn't be inserted to enter configuration mode
  - If you use "sim2" mode, the secondary SIM card must not be inserted to enter configuration mode.
  - If you use "dual" or "dual2" mode, none of the SIM cards must be inserted to enter configuration mode.
- Check with iotsupport@mtxm2m.com if you need additional help

DUALSIM_mode

Description: This parameter allows you to specify the operating mode when the DUALSIM_select parameter is set to “dual” or “dual2”. Specifies whether the SIM change occurs in the event of a loss of registration in the NETWORK for X configurable seconds or an event of IP loss for X configurable seconds.

Possible values: registration, ip

Default value: ip

Additional notes:

- The “registration” mode configures the modem to change the SIM in case of network registration problems for X seconds. It is the most recommended way to use
- The “ip” mode configures the modem to change the SIM in case of not being able to get an IP for X seconds. This method should only be used for configurations where the modem will have a permanent IP connection. Do not use this method if, for example, the modem will only handle SMS messages
- Check with iotsupport@mtxm2m.com if you need additional help

DUALSIM_timeout

Description: This parameter allows specifying the time, in seconds, from which a modem with the DUALSIM_select parameter configured as “dual” or “dual2” will change the SIM card if the event of loss of registration in the network or IP (configured in the DUALSIM_mode parameter) occurs during that time.

Possible values: 60 ... 3600

Default value: 120

Additional notes:

- Check with iotsupport@mtxm2m.com if you need additional help

8.31 Configuration parameters related to WIRELESS M-BUS: "WMBUS_"

The parameters with a WMBUS_ prefix are those related to the MTX-Tunnel configurations installed in MTX modem models that have W-MBUS support. Using these configuration parameters will allow the modem to be configured to read, store and send the W-MBUS frames collected from devices that have such technology (such as temperature sensors, water meters, electricity meters, etc.).

WMBUS_interval

Description: This parameter allows you to specify a time window in minutes. During these time windows, the modem will only accept readings from WMBus devices once based on its MAC, which will reduce data traffic in communications.

Possible values: Time windows are not used. All received frames are stored for sending. <1440 Minutes of the time window.

Default value: 0

Additional notes:

- A value of 0 will cause the device to store and send all the frames received from the same device (same serial number) as long as the configured filters (both serial number and manufacturer) are met. If there are no filters configured by serial number and manufacturer, a 0 will mean the storage and sending of all received WMBus frames

WMBus_mode

Description: This parameter allows you to specify the operating mode of the internal radio communication card W-MBus.

Possible values: 3, 7, 8, 9, 13, 14

Default value: 9

Additional notes:

- The mode must be chosen according to the W-MBus devices to be read

  3 = S mode

  7 = T-Meter mode

  8 = T-Other mode

  9 = T/C Other mode

  13 = C-Meter mode

  14 = C-Other mode
- Check with iotsupport@mtxm2m.com if you need additional help

WMBUS_filter

Description: This parameter allows you to specify a manufacturer filter. By specifying a manufacturer filter, the MTX-Tunnel will only store and send those W-MBus transmissions that correspond to a specific manufacturer, saving data from the SIM card.

Possible values: 3 character filter. See the following list of additional notes

Default value: none

Additional notes:

- See Annex D for the 3-character filter list for each W-MBus device manufacturer
- If you do not configure any manufacturer filter, no manufacturer filter will be performed, accepting the frames of any of them
- It is possible to specify a manufacturer filter individually for each device in the wmbus.txt file. (See the additional notes for Example 10.2 for more information.) If you want to use manufacturer filters specified in the wmbus.txt file, do not use this parameter WMBUS_filter

WMBUS_data

Description: This parameter allows you to specify the format of the W-MBus data sent within a JSON object to an HTTP platform, MQTT, etc.

Possible values: jsonrawhex, jsonrawbase64

Default value: jsonrawhex

Additional notes:

- The WMBus data collected by the MTX-Tunnel after the filtering process (by manufacturer, time window, serial numbers, is saved and sent to a remote data platform always in raw mode within a JSON object, where it must be decoded
- Example of jsonrawhex mode, where WMBus data is sent as WDATA parameter in hexadecimal mode

  {“IMEI”:”354033091777774”,”TYPE”:”WMBUS”,”TS”:”2020-12-11T11:37:47Z”,”WDATA”:”1d4446062a100020011b722a1000204606011bfd0000000466c1000000129b”}

- Example of jsonrawbase64 mode, where WMBus data is sent as WDATA parameter in base64 mode

  {“IMEI”:”354033091777774”,”TYPE”:”WMBUS”,”TS”:”2020-12-11T08:58:07Z”,”WDATA”:”HURGBioQACABG3IqEAAgRgYBGz8AAAAEZrcAAAASLw==”}

- Note that the data sent in base64 occupies less size in number of bytes transmitted, so with this method you will save bandwidth of your data plans

CONFIGURATION EXAMPLES

1. ANNEX: Basic Scenarios, Configuration Examples

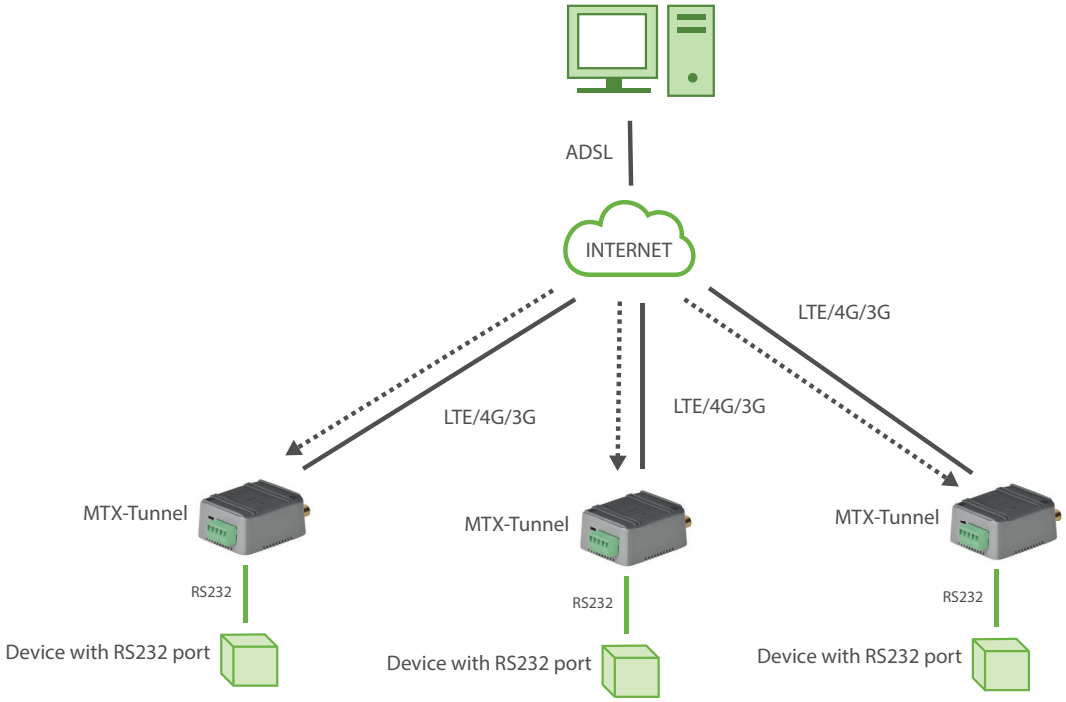
1.1 EXAMPLE: LTE-4G/3G/2G-Serial tunnel. MTX-Tunnel is configured as a TCP/IP server and permanently connected to Internet using a SIM provided by the operator with a fixed IP address.

Scenario details:

- There are 3 RS232 remote devices, configured as 115200, 8, N, 1 and hardware flow control. We need to control and monitor from a central server –PC with internet connection
- You can access them at any time. Therefore, the modem connected to the remote devices must be permanently connected to GPRS. So the modem is configured as a server waiting for an input connection request in TCP port number 20010
- We will use fixed IP address network operator SIM cards

Solution:

MTX-IoT [4-S-N-N] modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: internetestatico.movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Value 0 means MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card doesn't have PIN security, use 0000
MTX_model: 199801393	MTX-Terminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	We do not need the URC information messages
TCP_port: 20010	TCP port used
FIREWALL_enabled: off	Firewall disabled: Any incoming connection is allowed

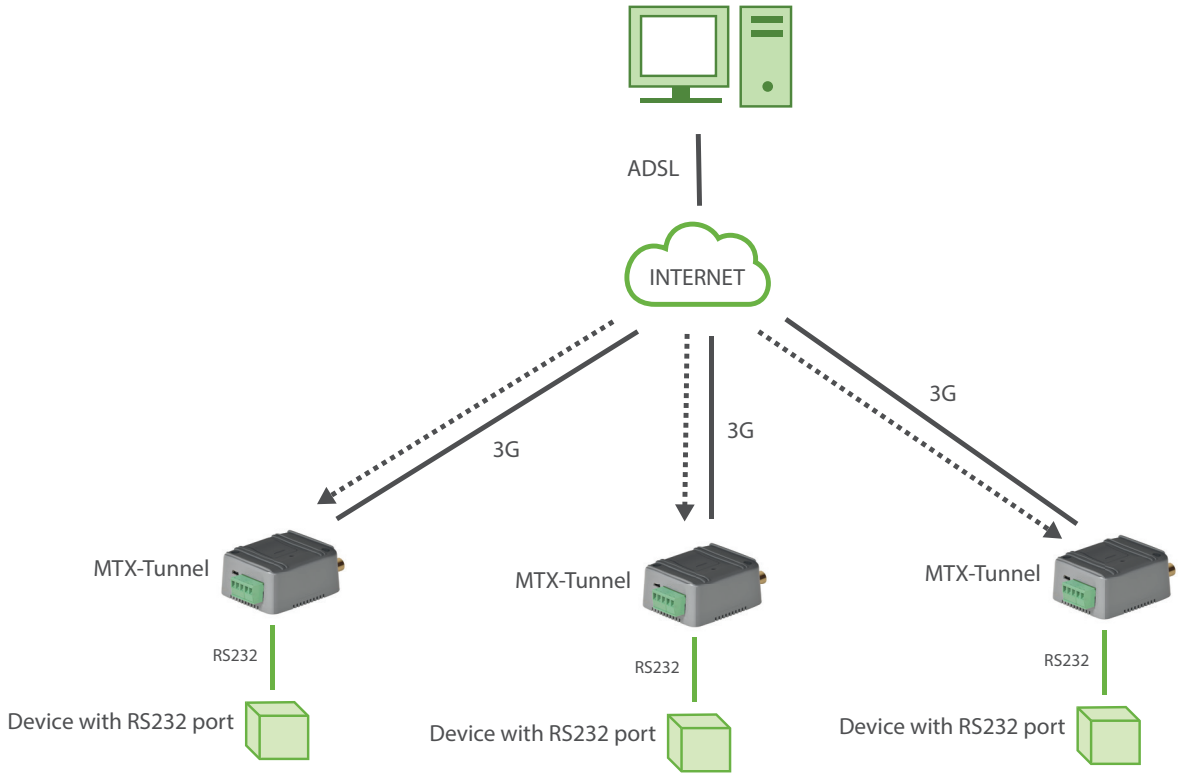
1.2 EXAMPLE: MTXTunnel configured as TCP/IP server and occasionally connected to 3G. Network operator dynamic IP addresses are used.

Scenario details:

- There are 3 RS232 remote devices, configured as a 9600, 8, N, 1 and no flow control. We need to control and monitor from a central server –PC with internet connection
- You need to access them occasionally with a remote control. Therefore the modem connected to remote devices does not need to be permanently connected to the 3G. The modem will only connect to the 3G when it receives a missed call or an SMS with the text string “on” from any phone number. Then, the modem will connect to the 3G waiting for an input connection request in TCP port number 20010
- We will use dynamic IP address network operator SIM cards

Solution:

MTX-IoT [4-S-N-N] modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	CTS hardware control disabled
COMM_autorts: off	RTS hardware control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 10	Connected 10 min if there's no serial data traffic
MTX_PIN: 0000	If SIM card doesn't have PIN security, use 0000
MTX_model: 199801393	MTX-Terminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will be not output
TCP_port: 20010	TCP port used
FIREWALL_enabled: off	Firewall disabled: Any incoming connection is allowed
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which made a call or "on" text SMS

Details:

- "GPRS_timeout: 10" makes the 4G/3G/2G connection activate itself after the missed call or incoming SMS with "on" text string but stop after 10 min. without GPRS/3G/4G data traffic

  As an example, if the data is transferred in 30 minutes, then the connection will be closed after another 10 minutes, giving a total of 40 minutes.

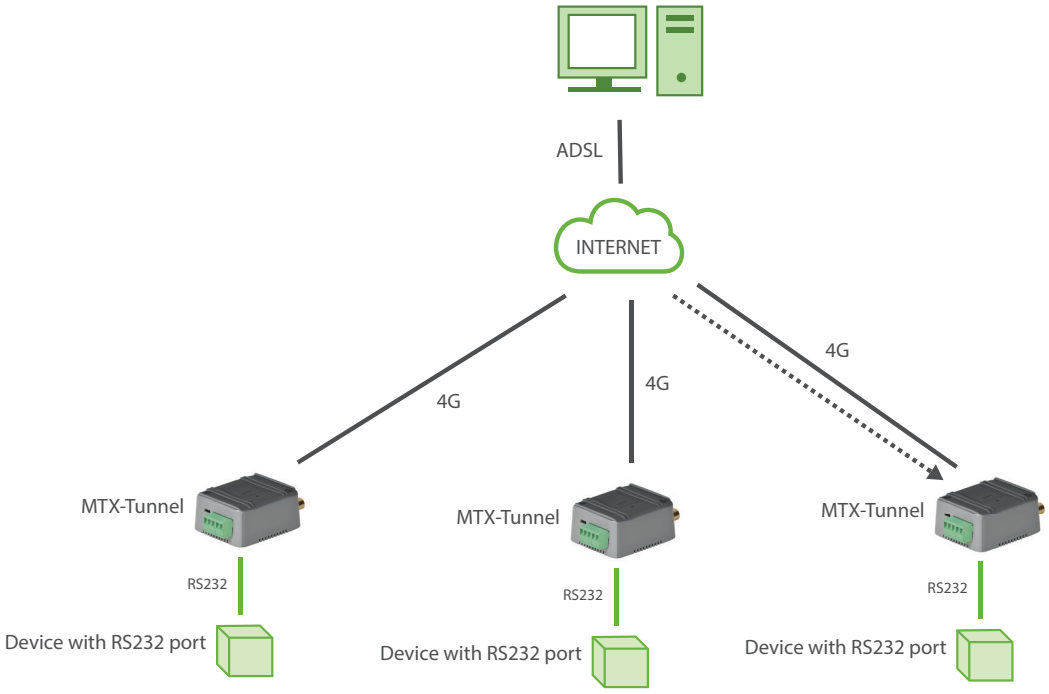
1.3 EXAMPLE: MTXTunnel configured as TCP/IP server and occasionally connected to 3G. Dynamic IP addressing. Firewall and authorized phone numbers active.

Scenario details:

- There are 3 RS232 devices configured as 9600, 8, N, 1 and no flow control. We need to control and monitor from a central server using the 3G-Serial tunnel gateway
- You need to access them occasionally with a remote control, so the modem connected to the remote devices does not need to be permanently connected to the 3G
- The modem will connect to the 3G for 10 minutes, but only after receiving a missed call or an SMS with the text string “on” from any phone number. The modem responds to same phone number with an SMS detailing the IP address obtained from the network
- Then, the modem will connect to the 4G waiting for an input connection request in TCP port number 20010
- We will use a dynamic IP address with network operator SIM cards.
- For more security, modems will only respond to incoming missed calls or special SMSs from authorized phones. Another security feature is that MTX-Tunnel’s modem will only accept an incoming connection from a specific IP address. For example in this case: 200.101.102.103

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	CTS flow control disabled
COMM_autorts: off	RTS flow control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 10	Connected 10 min if there's no serial data traffic
MTX_PIN: 0000	If SIM card doesn't have PIN security, use 0000
MTX_model: 199801393	MTX-Terminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will be not output
TCP_port: 20010	TCP port used
FIREWALL_enabled: on	Enabled to accept connections from authorized IPs
FIREWALL_IP1: 200.101.102.103	Authorized IP address to access MTX-Tunnel
SMS_allPhones: off	All phone numbers are forbidden
SMS_sendIP: on	IP to phone which made a missed call or "on" text SMS
SMS_validPhone1: +34666123456	Authorized phone number 1

SMS_validPhone2: +34666123457	Authorized phone number 2
SMS_dafaultPrefix: +34	Prefix for local incoming calls (doesn't include prefix)

Details:

- GPRS_timeout: 10" This parameter makes the 3G connection activate itself after the missed call or incoming SMS with "on" text string but stop after10 minutes without 4G-serial data traffic

  As an example, if the data is transferred in 30 minutes then the connection will be closed after another 10 minutes, giving a total of 40 minutes

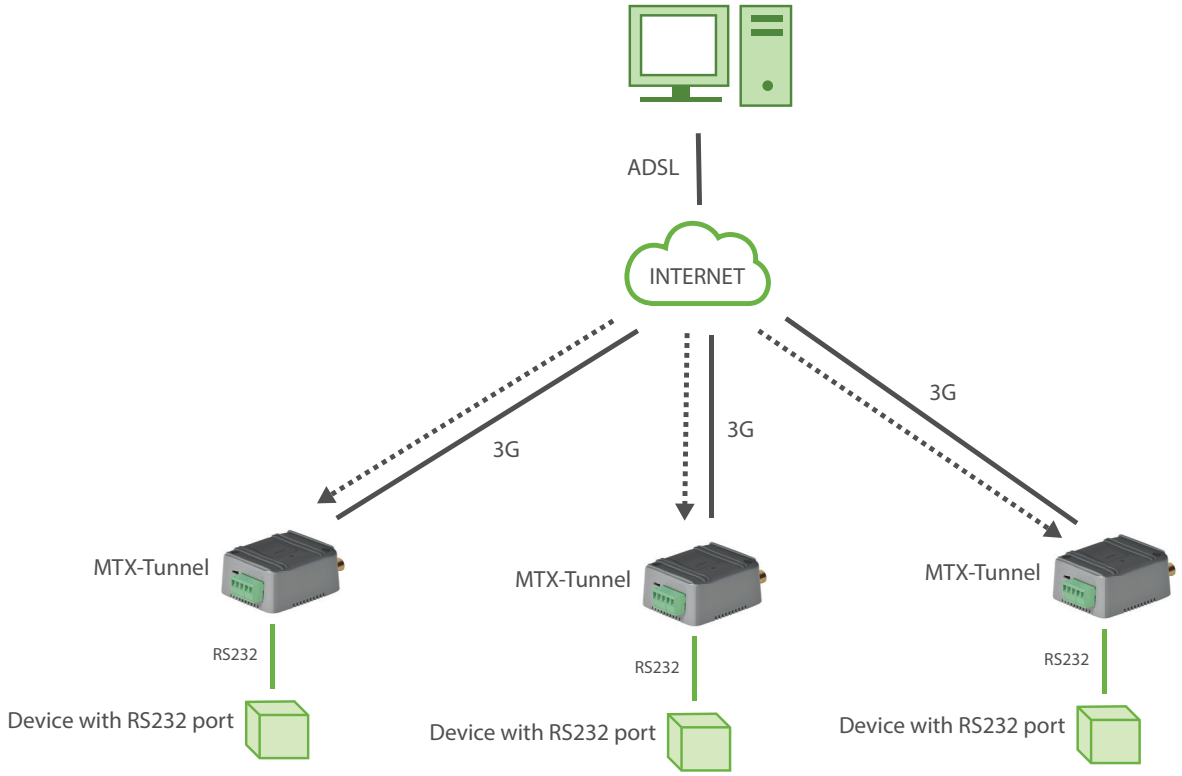
1.4 EXAMPLE: MTXTunnel configured as TCP/IP client and connected permanently to 3G. Network operator dynamic IP addresses are used.

Scenario details:

- There are 3 RS232 remote devices, configured as a 9600, 8, N, 1 and no flow control. We need to control and monitor periodically from a central server –PC with internet connection- using 3G-Serial tunnel gateway
- Server PC is waiting for incoming connections from remote MTX-Tunnel. This is the opposite to previous examples; MTX-Tunnel now connects to a known server IP address in the configured TCP/IP port. For example “server.mydomine.com” and TCP 20010 port
- MTXTunnel modems are permanently connected to the server, as the information must be in real time. As the modem connects to a well known central server IP address, the modems can use dynamic operator IP addressing

Solution:

MTX-IOT-3G modem+MTX-Tunnel



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	CTS hardware control disabled
COMM_autorts: off	RTS hardware control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	DNS server IP address
GPRS_timeout: 0	GPRS is permanently connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000
MTX_model: 199801393	MTX terminal modem model used
MTX_mode: client	TCP client mode
MTX_IDClient: equipoX	X value is different in each modem
MTX_urc: off	URC messages will be not sent
TCP_IP: oficina.midominio.com	Server IP address or DNS information
TCP_port: 20010	Server TCP port for connection

2. ANNEX: Advanced Scenarios, Configuration Examples

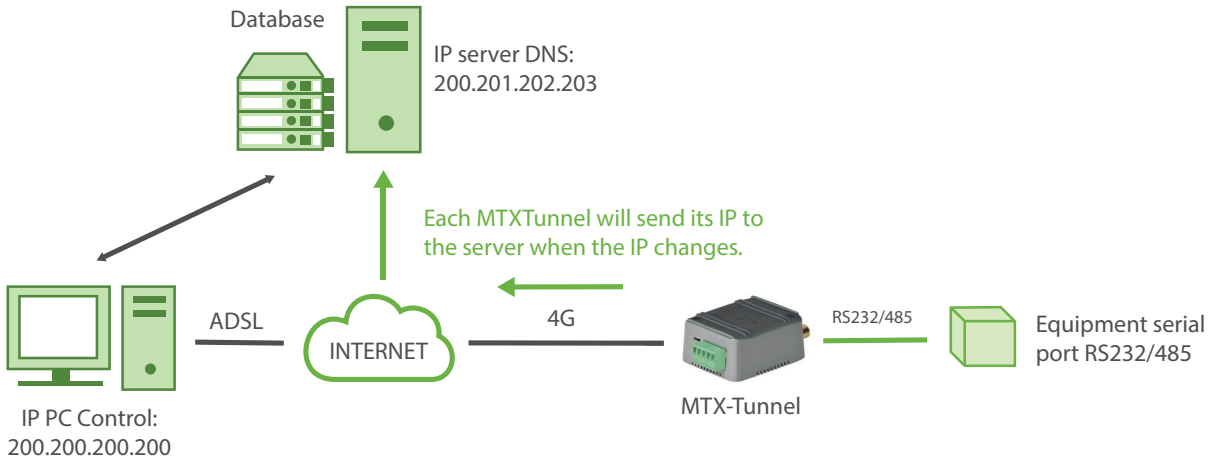
2.1 EXAMPLE: Advanced 3G-RS232 Tunnel – Setting up the MTX Tunnel as permanently connected and TCP/IP Server configured to send the IP by socket to a private DNS server.

Scenario details:

- We need to monitor 100 devices with the RS232 port (115200.8.N.1 and HW flow control) from a central control station using a 3G link. This will be done with the MTX-Tunnel to act as a transparent 3G-RS232 tunnel
- Access to the RS232 remote devices at any time is mandatory. So the modem connected to the serial port device (which needs to be controlled) must remain permanently connected to the 3G, waiting for a connection. Access to the modems must only be allowed for the IP coming from the PC Control (200.200.200.200) and also for a backup IP coming from the central office ADSL line, which is: 200.200.200.201
- In order to reduce costs, SIM cards use dynamic IP addressing. As we have a large number of modems, we will not use DynDNS. Instead, every time the MTX Tunnel sends the IP that it has been assigned (by the GSM operator) to a control centre. Every time a modem changes its IP address, the new IP must be sent to the control centre in order to report the change (IP 200.201.202.203 and port TCP 20000). This way, the PC control centre always knows the current IP of each modem

Solution:

MTX-IoT [4-S-N-N] modem+MTX-Tunnel firmware (GPRS-RS232 gateway)



Config.txt configuration file:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	Google DNS. Must be used if TCP_IP is set to DNS
GPRS_timeout: 0	Value 0 means MTX-Tunnel is GPRS always connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000
MTX_model: 199801393	MTX terminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will not be sent
TCP_port: 20010	TCP port used
FIREWALL_enabled: on	Accepts connections from authorized IPs
FIREWALL_IP1: 200.200.200.200	This IP address can only connect to remote modem
FIREWALL_IP2: 200.200.200.201	This IP address can only connect to remote modem
DNS_enabled: on	DNS service enabled
DNS_mode: socket	DNS server information sent via socket type

DNS_password: 12345678	Password string sent in DNS communication
DNS_server: 200.201.202.203	DNS server IP address
DNS_port: 20000	DNS server TCP port open
DNS_extended: off	GPIO and ADC values extra information is not sent

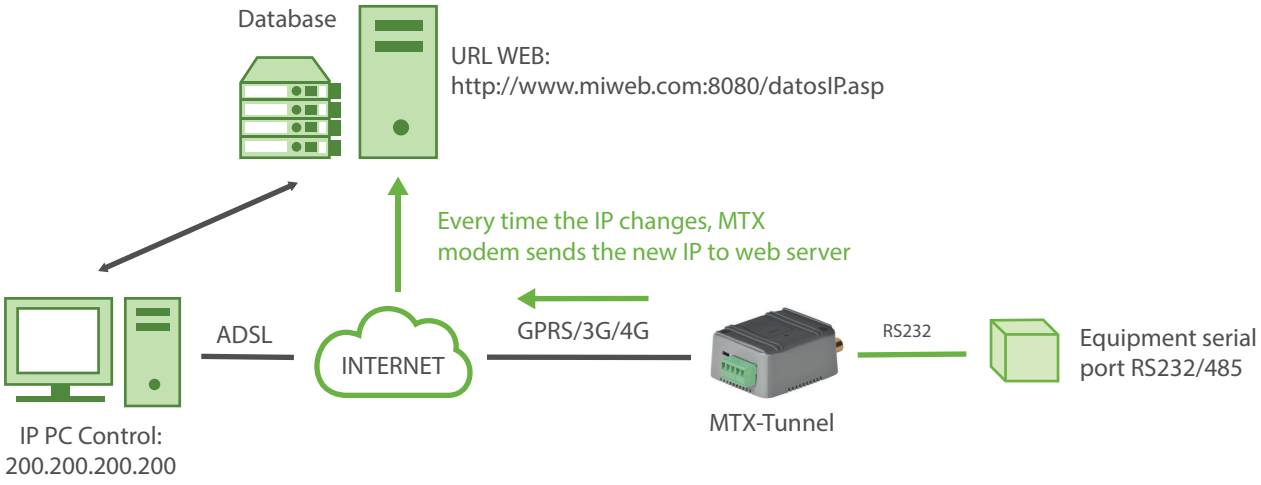
2.2 EXAMPLE: Advanced 3G-RS232 Tunnel – Setting up the MTX Tunnel as a permanently connected TCP/IP Server and configured to send the IP address to a web server by HTTP.

Scenario details:

- We need to monitor 100 devices with an RS232 port (115200.8.N.1 and HW flow control) from a central control station using a GPRS link. This will be done using the MTX-Tunnel acting as a transparent serial – 3G tunnel
- Access to the RS232 remote devices at any time is mandatory, so the modem connected to the serial port device (which needs to be controlled) must remain permanently connected to 3G waiting for a connection. Access to the modems must only be allowed for the IP coming from the PC Control (200.200.200.200) and also for a backup IP coming from the central office fiber line, whose IP is: 200.200.200.201
- In order to reduce costs, SIM cards will use dynamic IP addressing. As we have a large number of modems, we will not use DynDNS. So MTX Tunnel will send the IP every time that it the WEB server has been assigned a new one by HTTP (the IP address value assigned by the GSM operator). Every time a modem changes the IP address, the modem must send the new IP address value to the control centre (with URL http://www.miweb.com/datosIP.asp and port 8080). This way the external WebServer will store the IP in a SQL server database so the PC control centre has access to the devices at all times

Solution:

MTX-IoT [4-S-N-N] modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	Google DNS. Must be used if TCP_IP is set to DNS
GPRS_timeout: 0	Value 0 means MTX-Tunnel is always connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000
MTX_model: 199801393	MTX terminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will not be sent
TCP_port: 20010	TCP port used
FIREWALL_enabled: on	Enabled to accept authorized IPs
FIREWALL_IP1: 200.200.200.200	This IP can only connect to remote modem
FIREWALL_IP2: 200.200.200.201	This IP can only connect to remote modem
DNS_enabled: on	DNS service enabled
DNS_mode: http	GPRS IP address sent using HTTP GET

DNS_httpMode: get	It will be reported by HTTP GET
DNS_password: 12345678	Password string sent in DNS communication
DNS_server: miweb.com:8080/datosIP.asp	DNS server URL
DNS_extended: off	GPIO and ADC values extra information is not sent

Details:

- Bear in mind that when using HTTP, if you use a TCP port different to the standard HTTP (TCP 80), the port must be included in the parameter “DNS_server” and not in “DNS_port”
- Each time MTX-Tunnel changes the IP address, it will connect through and use the following URL:

  http://www.miweb.com:8080/datosIP.asp?IMEI=<suIMEI>&PASS=12345678&IP=<IP Public>

  The web server will receive the parameters sent by the MTX-Tunnel and store the IP address in a database. You can collect the parameters from the ASP page by inserting this example code in “datosIP.asp”.

```
<%
   IMEI=Request.QueryString (“IMEI”)
   Password=Request.QueryString (“PASS”)
   IP=Request.QueryString (“IP”)
%>
```

- If you prefer (recomended), you can send data to the server in JSON format via GET or POST. For example, if you wanted to send data in JSON format via HTTP GET, you would have to modify the configuration with something like this:

DNS_server: www.miweb.com:8080/datosIP.asp?data=	URL where the IP will be sent
DNS_httpMode: getjson	Data will be sent in JSON format

And to send data in JSON format via HTTP POST you would have to modify the configuration the following way:

DNS_server: www.miweb.com:8080/datosIP.asp	URL where the IP will be send
DNS_httpMode: postjson	Data will be sent in JSON format

When your modem is configured in getjson / postjson mode it will send a JSON with the following format:

{“IMEI”:357042060366409,”TYPE”:”DNS”,”P”:”12345678”,”IP”:”88.28.253.206”, “CSQ”:26,”VER”:”9.12”,”AUX”:””,”MOD”:”201”}

Where each parameter is:

IMEI: the unique identification number of the modem

TYPE: type of JSON sent (DNS in this case)

P: user’s field specified in DNS_password parameter

IP: modem’s current IP

CSQ: rssi of the modem (between 0 and 31)

VER: MTX-Tunnel version

MOD: MTX modem model

- Remember that you dispose of the configuration parameters DNS_header1, DNS_header2, DNS_header3 if you want to add headers to your HTTP requests

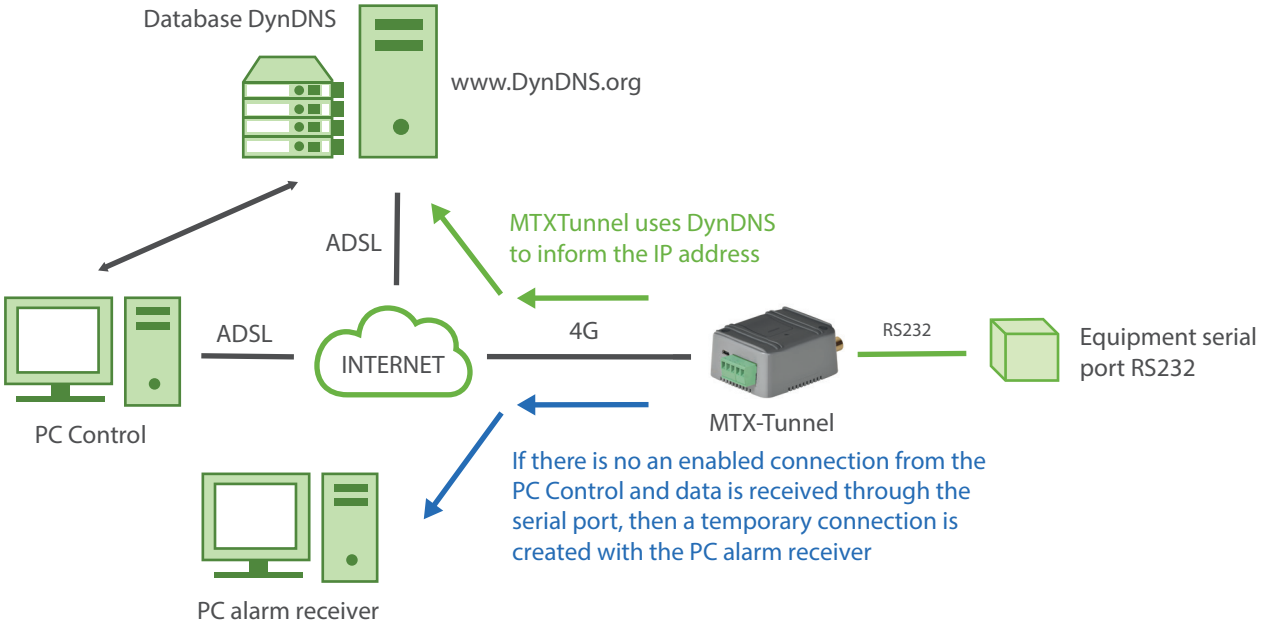
2.3 EXAMPLE: Tunnel 3G-RS232 – Set up the modem as a permanently connected TCP/IP server, with DynDNS and Client Temporal socket enabled.

Scenario details:

- There are several devices with an RS232 port which need to be monitored from a Control Centre via 3G
- You must be able to gain access to the RS232 devices. So the modem connected to the device's serial port must remain continuously connected to 3G, waiting for a connection
- Modems will only accept incoming connections from the IP central office: 200.200.200.200 and will reject any connection coming from different IP addresses
- In order to reduce the cost, SIM cards will use dynamic IP addresses. As there are only a few devices to be controlled, it is not necessary to configure a DNS private server (DNS_enabled: off) and you can just use the DynDNS service
- Central Control station connections will be periodic. Sometimes the devices may send an alarm through the port. In the case of not having a connection available between the PC central control station and the MTX-Tunnel, the MTX-Tunnel must create a temporary connection to another server (same address 200.200.200.200, and port 20011) to report the alarm

Solution:

MTX-IoT [4-S-N-N] modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	CTS hardware control disabled
COMM_autorts: off	RTS hardware control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	Google DNS. Must be used if TCP_IP is set to DNS
GPRS_timeout: 0	Value 0 means MTX-Tunnel is always connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000
MTX_model: 199801393	MTX terminal modem model used
MTX_mode: server	TCP server mode
MTX_temporalClient: on	Temporal client mode enabled
MTX_urc: off	URC messages will not be sent
TCP_IP: 200.200.200.200	Server IP address used for temporal client mode
TCP_port: 20010	Used in server and temporal client socket
FIREWALL_enabled: on	Firewall enabled
FIREWALL_IP1: 200.200.200.200	IP address authorized to connect to MTX-Tunnel
DYNDNS_enabled: on	DynDNS service enabled

DYNDNS_server: members.dyndns.org	DynDNS server URL
DYNDNS_hostname: mtxtunnel.dyndns.org	Your DNS name
DYNDNS_login: user	Your DynDNS account login
DYNDNS_password: myPassword	Your DynDNS password account

Details:

- Compatible with DynDNS & NO-IP. If using NO-IP (which is free) instead of DynDNS, the parameter where we have “members.dyndns.org” should be replaced by “dynupdate.no-ip.com”

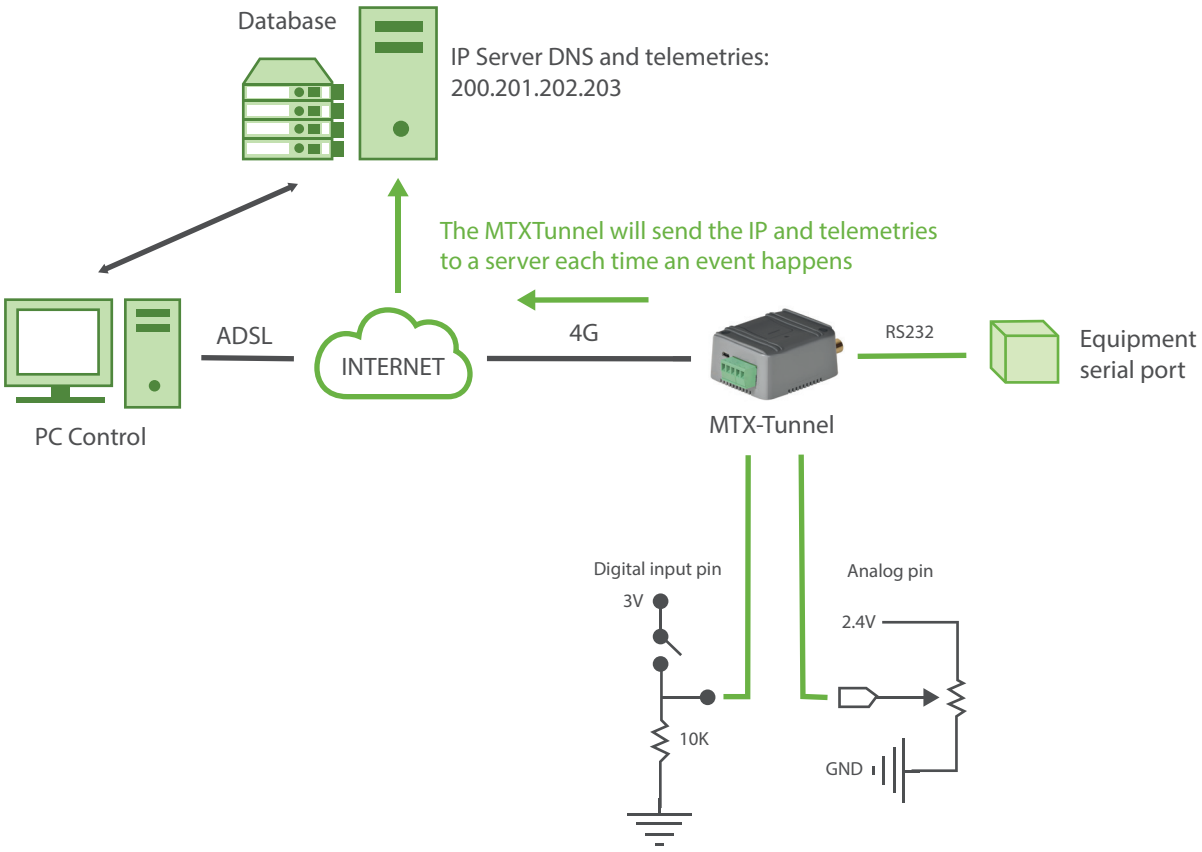
2.4 EXAMPLE: Advanced Tunnel 3G-RS232. Set up the modem as a TCP/IP Server connected occasionally when a GPIO and ADC level changes.

Scenario details:

- There are several devices with an RS232 port and they need to be monitored from a Central PC via a Tunnel 3G-RS232
- We will very rarely need to access the RS232 devices, only when there is a special incoming SMS/missed call, when there is a change from “0” to “1” on a digital input or when the level of the Analog/Digital converter of the modem is below 500mV or above 1500mV
- Therefore, the 3G-RS232 Tunnel should be activated for 10 minutes when it receives a missed call or a special SMS with the word “on” from any phone number or when the trigger is met for the Analog/Digital input. If there is a call or SMS, the MTX Tunnel will return an SMS with the IP address from the GSM operator because SIM cards with a dynamic IP will be used. The modem must also send its current IP address, the GPIOs and ADCs readings to a server and to a Control Centre (to report the event)

Solution:

MTX-IoT [4-S-N-N]-STD-N + MTX-Tunnel



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	Google DNS. Must be used if TCP_IP is set to DNS
GPRS_timeout: 10	Connected for 10 min if there's no serial data traffic
MTX_model: 199801393	MTXTerminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will not be sent
TCP_port: 20010	TCP port used
FIREWALL_enabled: off	Accepts incoming connections from any IP
SMS_sendIP: on	GPRS IP sent by SMS to mobile call or SMS text "on"
SMS_alPhones: on	All number phones are authorized
GPIO_mode0: input	GPIO0 configured as an input
GPIO_config0: wakeup;1	GPIO0 function
ADC_mode0: voltage	ADC0 configured as a voltage analog input

ADC_config0: wakeup;500;1500;10	Wakeup if ADC0<=500 or ADC0>=1500 with systeresis 10mV
DNS_enabled: on	Enables DNS service
DNS_server: 200.201.202.203	Server DNS IP address
DNS_mode: socket	DNS communication socket type
DNS_password: 12345678	Password string sent in DNS communication
DNS_port: 20000	DNS server port used
DNS_extended: on	Extended GPIO & analog values+IMEI information to server

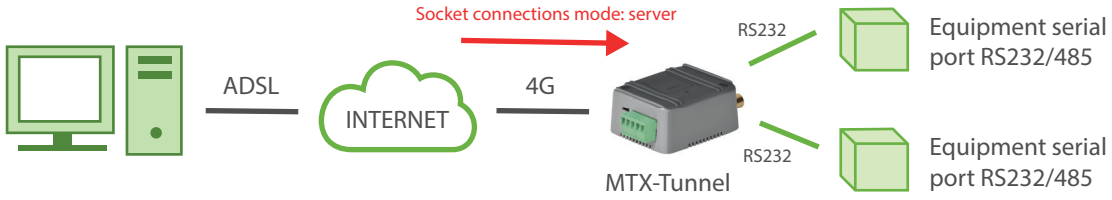
2.5 EXAMPLE: Dual Tunnel 4G-RS232. Controlling 2 RS232 devices with 1 modem and SIM card. Socket connection server type.

Scenario details:

- We want to control 2 serial RS232 devices that are near to each other. Therefore, as the modem MTX-T [4-N]2 has 2 serial ports, each serial port will be connected to one of two control devices which will economize the system by using a single modem and a single SIM card
- Simultaneous access to the 2 serial devices from a Control PC is require, so the modem should listen, configured as server, through two TCP / IP ports. It will be possible to access the serial device connected to the COM1 of the modem the port TCP 20010 and the serial device connected to the COM2 of the modem by the port TCP 20011
- The tunnel must remain permanently connected. There is no need to configure a DNS server since a SIM card with fixed IP address will be used
- Access will only be allowed from two IP addresses coming from the Control Centre. Any attempt to connect from other IP addresses (unauthorized access) must be aborted

Solution:

MTX-T [4-N]2 modem+firmware MTX-Tunnel



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity

COMM2_baudrate: 19200	Serial port baud rate 2
COMM2_bitsperchar: 8	8 bit data
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is connected to GPRS permanently
MTX_model: 199801438	MTXTerminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will not be sent
TCP_port: 20010	TCP port used for connection with serial device 1
TCP_port2: 20011	TCP port used for connection with serial device 2
FIREWALL_enabled: on	Accepts incoming connections from following IPs
FIREWALL_IP1:200.201.202.203	IP address enabled to access MTX-Tunnel
FIREWALL_IP2:200.201.202.204	IP address enabled to access MTX-Tunnel
TELNET_enabled: on	Telnet enable to control COM 2
TELNET_login: user	TELNET username
TELNET_password: 1234	TELNET password
TELNET_port: 20023	TCP port for telnet

Details:

- You can use 2 server sockets to create independent 4G/3G/2G-RS232/485 gateways

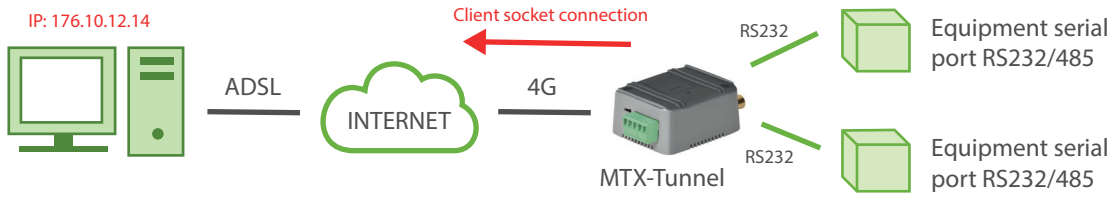
2.6 EXAMPLE: Double 3G-RS232 tunnel. Two RS232 devices controlled by a single modem and a single SIM card. Client socket connection type.

Scenario Details:

- We want to control 2 serial RS232 devices that are near to each other. Therefore, as the modem MTX-IoT [4-S-N-N] has 2 serial ports, each serial port will be connected to one of two control devices which will save the system by using a single modem and a single SIM card
- The modem is required to simultaneously create two 3G-RS232 gateways in client mode. The PC server's 20010 TCP port will connect to the gateway that has access to the modem's COM1 serial device and the 20011 TCP port will connect to the gateway that has access to the modem's COM2 serial device
- The 2 tunnels must be permanently connected

Solution:

MTX-T2 [3-N] modem+firmware MTX-Tunnel



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM2_baudrate: 19200	Serial port baud rate 2
COMM2_bitsperchar: 8	8 bit data

COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is connected to GPRS permanently
MTX_model: 199801406	MTXTerminal modem model used
MTX_mode: client	TCP client mode
MTX_urc: off	URC messages will not be sent
MTX_IDClient: ID1234ABCD	MTX identifier
MTX_rssiLevel: 10	Activation of signal level LED
TCP_IP: 176.10.12.14	The gateway associated with COM1 will connect to this IP
TCP_port: 20010	Gateway associated with COM1 will connect to this TCP
TCP_IP2: 176.10.12.14	The gateway associated with COM2 will connect to this IP
TCP_port2: 20011	Gateway associated with COM2 will connect to this TCP

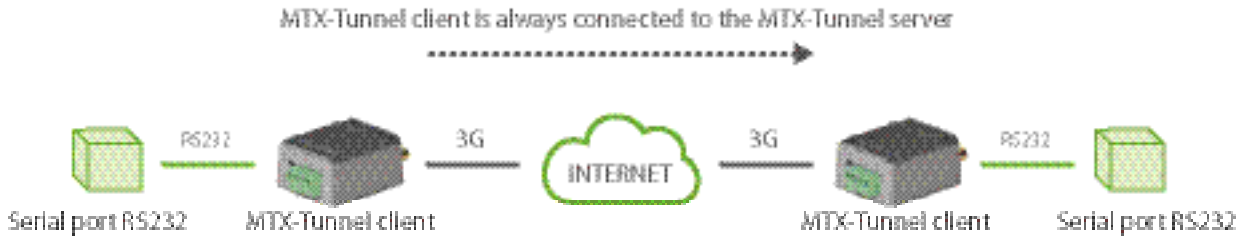
2.7 EXAMPLE: Serial Cable replacer RS232-4G-RS232.

Scenario details:

- We have 2 devices RS232 (115200.8, N, 1 flow control HW) linked by a serial cable through which data is sent. The intention is to replace this serial cable by a transparent RS232-4G-RS232 link
- The equipment does not contain a PC, nor is it intelligent, and the internal control programs cannot be changed. This can be seen as “black boxes” with an RS232 serial port with no possibility of changing anything. Therefore it will be necessary to use an MTX-Tunnel on each side of the communication, one acting as a “server “and the other one acting as a “client”
- One of them, the server, will use a SIM card with fixed IP address 200.1.2.3 (although it could be used in this scenario, we will not be using DynDNS in order to simplify the situation). The Client must be programmed to be connected to the MTX-Tunnel Server’s fixed IP address at all times so that the tunnel Serial-4G-Serial is always set

Solution:

2 MTX-T [4-N] modems+MTX-Tunnel firmware



Config.txt configuration file:

MTX-Tunnel SERVER:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity

GPRS_apn: internetestatico.movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Value 0 means MTX-Tunnel is always connected
MTX_model: 199801422	MTX terminal modem model used
MTX_mode: server	TCP server mode
TCP_port: 20010	TCP port used

MTX-Tunnel CLIENT:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Value 0 means MTX-Tunnel is always connected
MTX_model: 199801422	MTX terminal modem model used
MTX_mode: client	TCP client mode

TCP_IP: 200.1.2.3	Fixed IP if MTX-Tunnel is set as server mode
TCP_port: 20010	Remote TCP port waiting for connection

Details:

- A SIM card with a fixed IP address is being used because it is more convenient, but the MTX Tunnel server could use DynDNS if needed

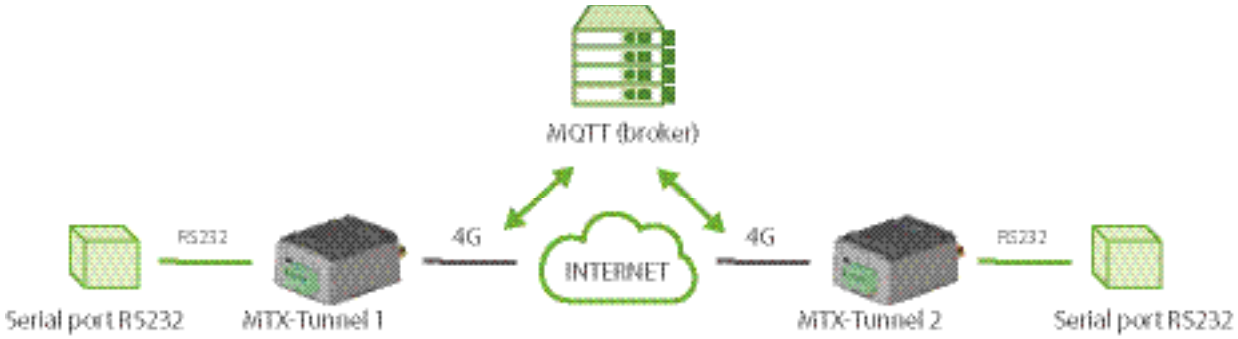
2.8 EXAMPLE: Serial Cable replacer RS232-4G-RS232 via MQTT.

Scenario details:

- There are 2 RS232 devices (115200,8, N, 1) connected by a serial cable through which data is sent. We want to replace the serial cable with a transparent link RS232-4G-RS232
- None of the devices is a PC. They are computers with hardly any intelligence and their internal control programs can't be modified. That is, they can be seen as "black boxes" with an RS232 serial port with no possibility of modifying anything, so it will be necessary to use one MTX-Tunnel on each side of the communication
- Both modems will use affordable SIM cards that don't have a public or fixed IP. Therefore, an MQTT broker will be used as an intermediary for communications

Solution:

2 MTX-T [4-N] modems+MTX-Tunnel firmware



Config.txt configuration file:

MTX-Tunnel 1:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autorts: off	RTS hardware control disabled
COMM_autocts: off	CTS hardware control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity

GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Value 0 means MTX-Tunnel is always connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_mode: mqtt	MQTT serial gateways will be used
MTX_model: 199801445	MTX terminal modem model used
MTX_ping: 35	One ping every 35 min. without communications
MTX_pingIP: 8.8.8.8	Ping address
MTX_invertedCom: off	Gateway port RS485
MTX_msToSend: 250	No fragmented networks
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds with its IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
FIREWALL_enabled: off	Any incoming connection form any IP is allowed
MQTT_enabled: on	MQTT service enabled
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to

MQTT_keepalive: 300	Connection keep alive (300 seconds)
MQTT_commrxtopic: mtx2	Data received will be retransmitted via serial
MQTT_commtxtopic: mtx1	Data received v/serial, retransmitted to this topic

MTX-Tunnel 2:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autorts: off	RTS hardware control disabled
COMM_autocts: off	CTS hardware control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Value 0 means MTX-Tunnel is always connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_mode: mqtt	MQTT serial gateways will be used
MTX_model: 199801445	MTX terminal modem model used
MTX_ping: 35	One ping every 35 min. without communications
MTX_pingIP: 8.8.8.8	Ping address
MTX_invertedCom: off	Gateway port RS485
MTX_msToSend: 250	No fragmented networks

SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds with its IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
FIREWALL_enabled: off	Any incoming connection form any IP is allowed
MQTT_enabled: on	MQTT service enabled
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
MQTT_commrxtopic: mtx1	Data received will be retransmitted via serial
MQTT_commtxtopic: mtx2	Data received v/serial, retransmitted to this topic

Details:

- The MTX-Tunnel 1 modem forwards the entire data stream it receives on its RS232 serial port to the MQTT broker to the “mtx1” topic. The MTX-Tunnel 2 modem, since it’s subscribed to the “mtx1” topic of the MQTT broker, automatically receives the data stream that it also forwards to its RS232 serial port. And viceversa
- If instead of using the RS232 port we need to perform an RS485 bridge, we only have to change the parameter MTX_invertedCom to “on”
- If we need to use secure communications (SSL) between both devices, we can use port 8883, specifying in the broker MQTT_server: ssl: //test.mosquitto.org: 8883. At the end of this manual we will find how to install SSL certificates on computers
- Keep in mind that communications latencies may be somewhat greater than direct communication latencies since there is an intermediary (the mqtt broker) and the speed of communications will depend on the power of the latter. Set the timeout if necessary

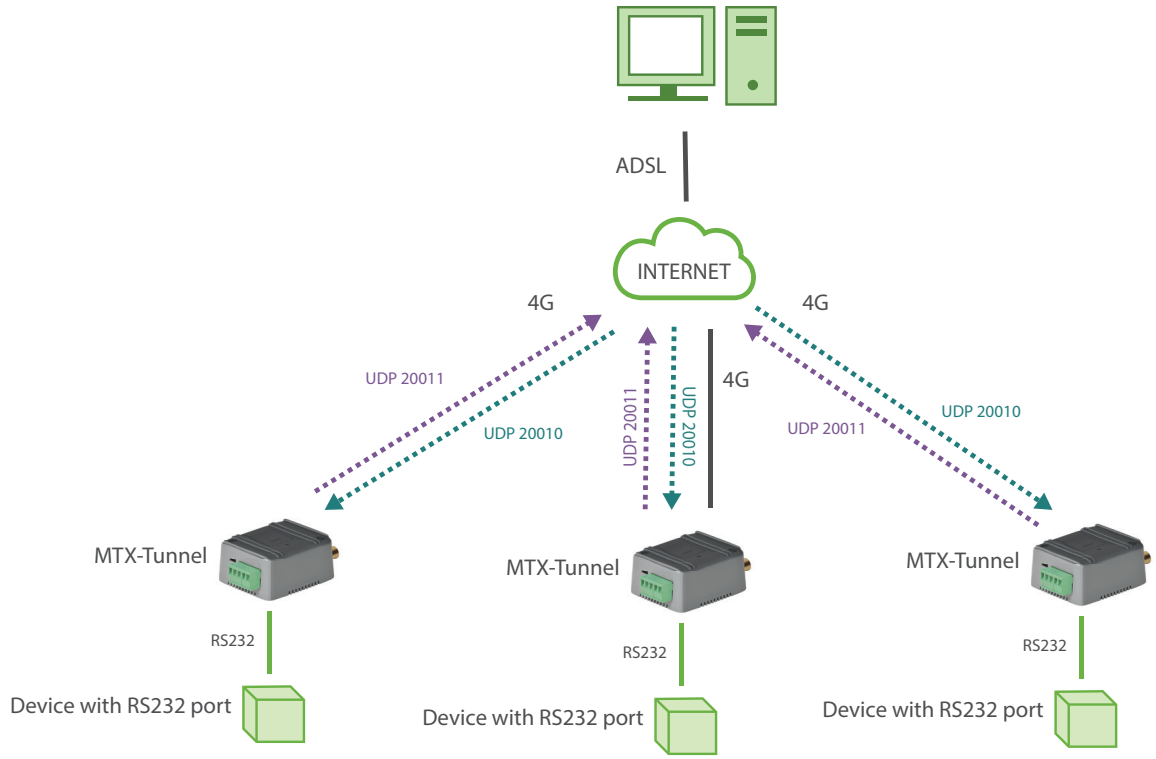
2.9 EXAMPLE: Serial 3G-UDP Tunnel. MTX configured as Client/Server UDP connected permanently to 3G and using a SIM card with a fixed IP address.

Scenario details:

- We have 3 devices with an RS232 port (configured as 115200, 8, N, 1 and HW flow control) and they need to be monitored from a PC which is located in a Central Control station with Internet access
- It is important to have access to the RS232 devices at all times; therefore the modem is connected to the serial port device and must remain connected to 4G at all times
- The used protocol is not going to be orientated to the (TCP) connection because we want to use UDP for communications. The MTX Tunnel will be waiting for data packets on the UDP 20010 port sent from the Central Control PC. Every time a packet is received, it will be retransmitted by the serial port. Similarly the data packets received by MTX through the RS232 serial port will be retransmitted to the Control PC via UDP using the UDP 20011 port
- SIM cards with fixed IP addressing will be used. The Control PC IP is fixed but in anticipation of possible future changes, the DNS “oficina.domino.com” will be used instead of their IP address. Then the MTX Tunnel must resolve the DNS

Solution:

MTX-T [4-N] modem+MTX-Tunnel



Config.txt configuration file:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: internetestatico.movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	DNS server IP address
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_model: 199801422	MTX-Terminal modem model used
MTX_mode: udp	UDP mode
MTX_urc: off	URC messages will not be sent
UDP_IP: oficina.dominio.com	Public IP address form central server
UDP_localPort: 20010	Local port waiting for incoming connections
UDP_remotePort: 20011	UDP server port waiting for incoming connections
FIREWALL_enabled: off	Any incoming connection form any IP is allowed

Details:

- Remember that it is necessary to configure the router in the PC server location to use NAT with the UDP port, NOT the TCP port

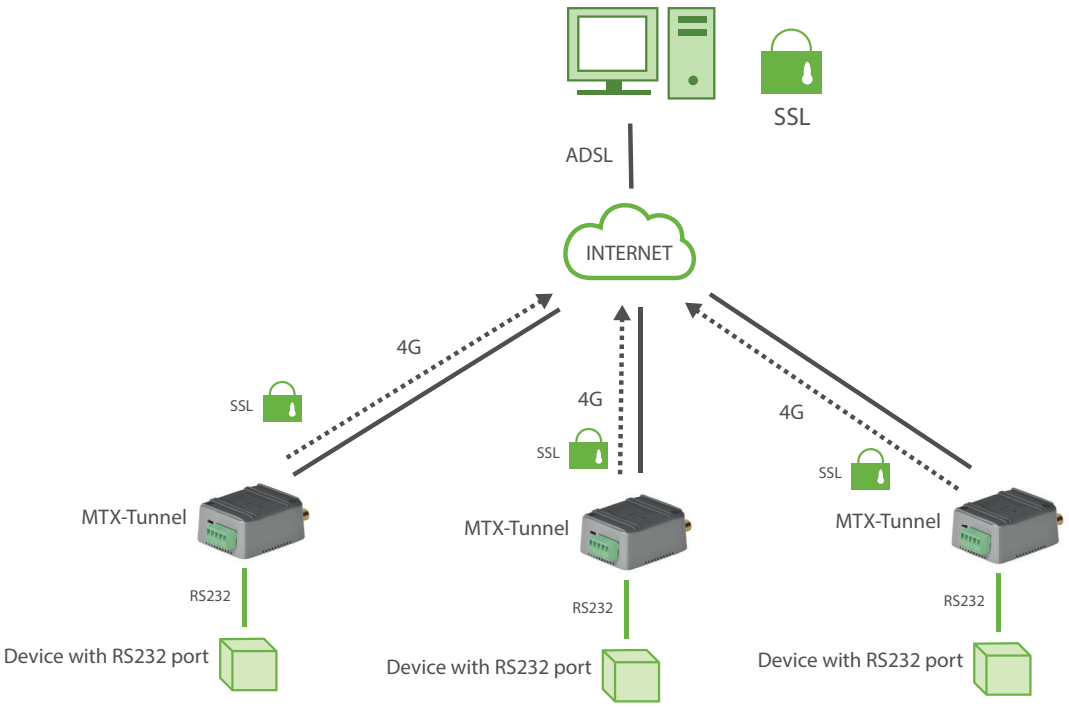
2.10 EXAMPLE: 3G-Serial Tunnel with SSL security. MTX-Tunnel configured as TCP/IP Client connected to GPRS permanently using a SIM card with a dynamic IP address.

Scenario details:

- There are three devices with RS232 (configured as 9600,8,N,1 and without flow control) which send information on a regular basis through the serial port. We want to collect and store all of that information from a PC server located at the central office
- The PC server waits for incoming connections from the different MTX-Tunnels. This means that the server PC does not connect with the MTX Tunnel; instead it is the MTX Tunnel which connects to the office IP address where the PC server is located (this is "oficina.midominio.com" and TCP 20010 port)
- The PC software control supports SSL socket communications and, as the information sent is very sensitive, it is important that we block access to this it, even if Ethernet sniffer is used (within the own LAN net where the PC server is located)
- MTXTunnel modems must remain permanently connected to the PC server, as information is frequently sent. MTX-Tunnel connects to the central office IP address so it does not make sense to use a SIM with fixed IP. Instead, SIMs with dynamic IP addresses will be used which are much cheaper

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	CTS hardware control disabled
COMM_autorts: off	RTS hardware control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	DNS server IP address
GPRS_timeout: 0	Modem is connected to GPRS permanently
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_model: 199801393	MTXTerminal modem model used
MTX_mode: client	TCP client mode
MTX_IDClient: equipoX	MTXTunnel identification string different for each modem
MTX_urc: off	URC messages will not be sent
MTX_clientSSL: on	SSL secure communication enabled
TCP_IP: oficina.midominio.com	Server IP address or DNS information
TCP_port: 20010	Server TCP port for connection

Details:

- Remember MTX-Tunnel can only make SSL connections if it is configured in “client” mode
- You can install SSL Root certificates according to your needs

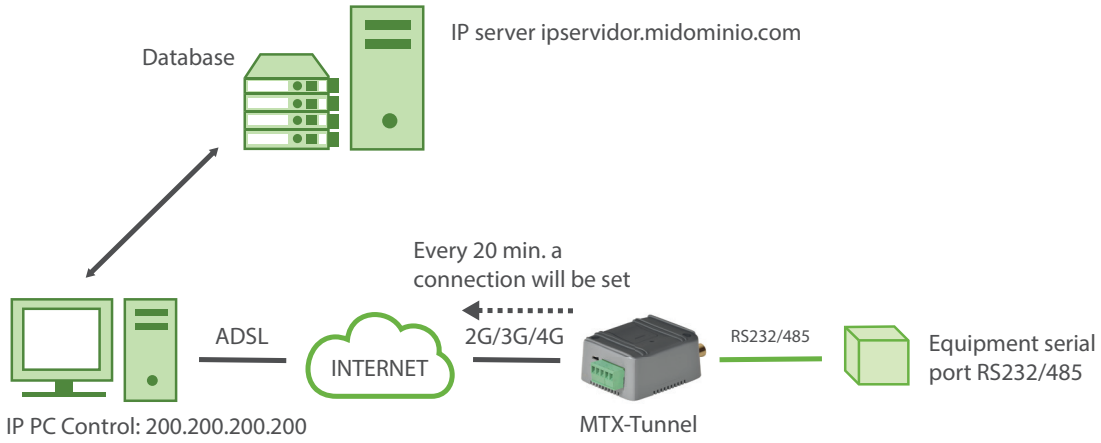
2.11 EXAMPLE: Advanced tunnel 4G/3G/2G-Serial - Settings for MTX-Tunnel as TCP/IP Client with a connection 3 times an hour.

Scenario details:

- We have 100 devices with an RS232 port (115200,8,N,1 and flow control HW), which need to be monitored from a Central Control Post via GPRS. For this MTX-Tunnel will be used, which will act as a 4G/3G/2G-Serial transparent tunnel
- The modems will be periodically connected, for 5 minutes every 20 minutes, to a central server via a TCP client socket
- Once connected to the central server, the modems must identify themselves by sending an identifier string+IMEI. After sending the identifier string, the modem must set a 4G/3G/2G-RS232 transparent gateway for those 5 minutes
- To be able to access the remote maintenance of the modem, the Telnet service and SMS settings must be enabled

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel software (para túnel 4G/3G/2G-RS232)



EXAMPLE of settings (file config.txt) for this scenario:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled

COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 5	The gateway will last 5 minutes
GPRS_autoTimeout: off	Duration doesn't restart when data is received
MTX_PIN: 0000	If the SIM card does not have a PIN, 0000
MTX_model: 199801436	MTX chosen
MTX_mode: client	The modem is set up as TCP server
MTX_urc: off	We do not need URC information messages
MTX_IDClient: ID0001	Modem identifier
MTX_IDClientExtended: imei	Besides the identifier, we want the IMEI to be sent
MTX_TPServer: null	We don't want a real time server
MTX_alwaysConnectedClient: off	We don't want the socket to reconnect
WAKEUP_timeEnabled: on	We want a time delay per hour
WAKEUP_time1: XXXX00	The socket will be set at 00 minutes of every hour
WAKEUP_time2: XXXX20	The socket will be set at 20 minutes of every hour
WAKEUP_time3: XXXX40	The socket will be set at 40 minutes of every hour
TCP_IP: ipServidor.midominio.com	DNS (o IP) where the modem will be connected
TCP_port: 20010	Where the modem will be connected
FIREWALL_enabled: off	Firewall is disabled

TELNET_enabled: on	We enable Telnet to send remote commands...
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_firewall: off	The MTX accepts telnet connections from any IP
SMS_allPhones: on	All phones are authorized
SMS_ATEnabled: on	We enable AT command via SMS
SMS_ATResponse: on	We enable replies via SMS to the AT commands

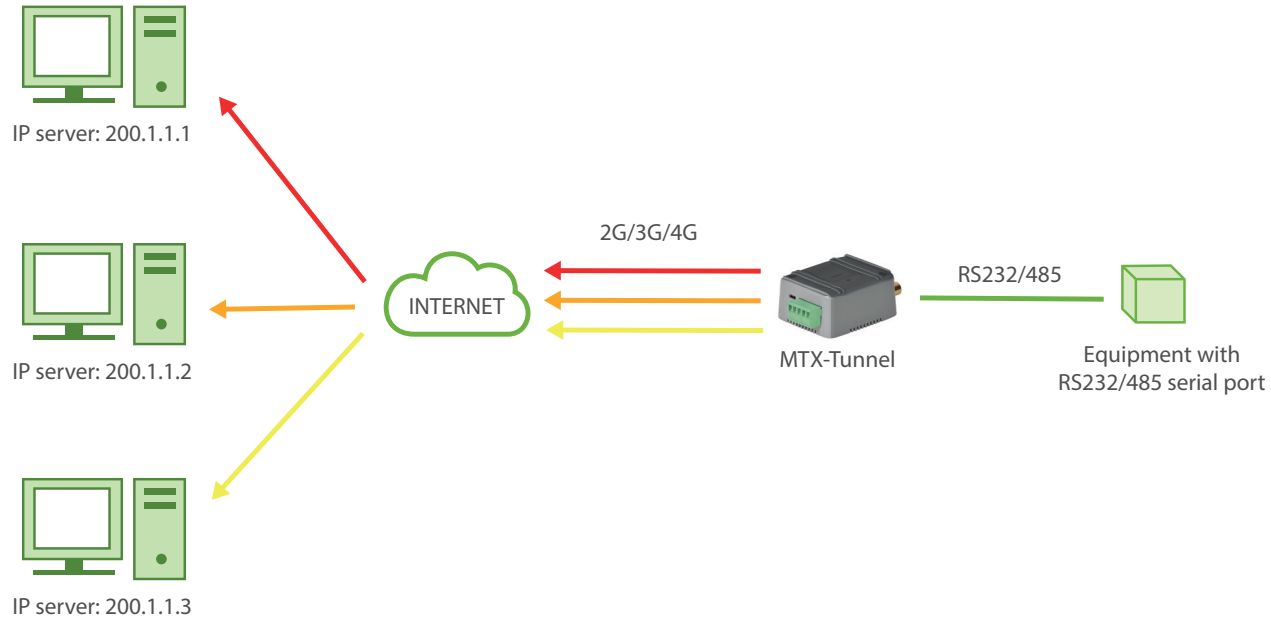
2.12 EXAMPLE: Advanced tunnel 4G-serial - Receiving and sending data to servers with UDP data packets.

Scenario details:

- We have a device with an RS232port (9600,8,N,1 and without flow control HW). This device, when an alarm is created, sends a data packet that, for security reasons, must be sent to 3 different servers.
- The communication with the servers must be made using UDP data packets. The different server addresses are: 200.1.1.1, 200.1.1.2 and 200.1.1.3. The 3 servers expect to receive the data in the UDP port 20010
- Servers also must be able to send data to the device for settings and supervision issues. They send the data packets to the port UDP 20000

Solution:

MTX-T [4-N] + MTX-Tunnel software (for GPRS-RS232 tunnel)



EXAMPLE of settings (file config.txt) for this scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control
COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	GPRS connection is active 100% of time
MTX_PIN: 0000	If SIM card does not have PIN, leave as 0000
MTX_model: 199801445	The MTX model chosen
MIX_mode: udp	The modem is set up as UDP
MTX_urc: off	We do not need URC information messages
MTX_ping: 35	Every 35 min. without communications, one ping
MTX_pingIP: 8.8.8.8	Address where the ping is made
UDP_IP: 200.1.1.1,200.1.1.2,200.1.1.3	The 3 servers addresses, separated by commas
UDP_remotePort: 20010	UDP remote port of servers where data will be sent
UDP_localPort: 20000	UDP local port where servers will send data
FIREWALL_enabled: off	Firewall disabled

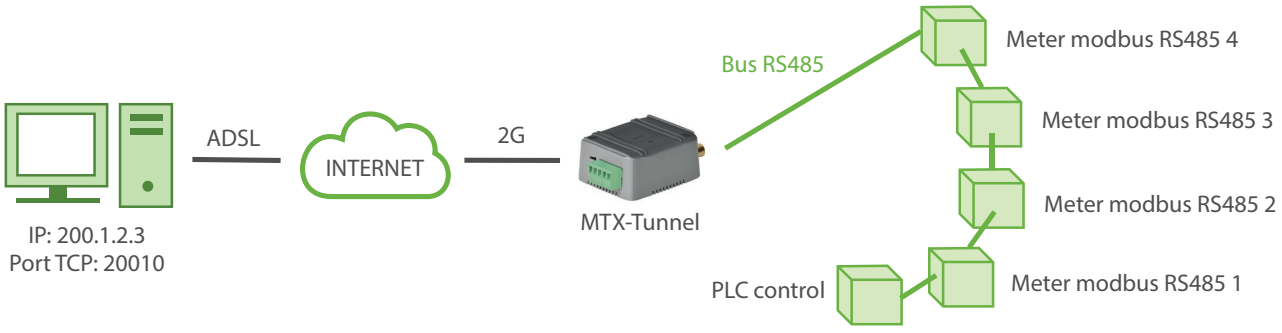
2.13 EXAMPLE: A 3G-serial gateway, broadcasting data that begins with a chosen header to a server.

Scenario details:

- Although MTX-Tunnel allows independent readings from ModBus devices to be made, we consider a PLC being the master in a MODBUS communication with a network of meters
- The PLC only allows the use of the MODBUS protocol and we want to be able to send some data from the meters to a central server
- To do so the PLC will consider the MTX-Tunnel as any other modbus (slave) device, with an address @100. That is, the MTX-Tunnel must not send, via 3G, all the traffic of bus RS485 (that is, not all the traffic between the control PLC and the meters), but only the frames going to the address @100, so the central server will receive only that data
- The communication must be made in TCP Client mode, i.e., the MTX-Tunnel must be connected in client mode against the IP server (200.1.2.3) and the TCP port 20010

Solution:

MTX-T [3-N] modem+MTX-Tunnel software



EXAMPLE of settings (file config.txt) for this scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control
COMM_autorts: off	No flow control

COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	3G APN by GSM operator
GPRS_login: MOVISTAR	3G Login
GPRS_password: MOVISTAR	3G Password
GPRS_timeout: 0	3G connection is active 100% of time
MTX_PIN: 0000	If SIM card does not have PIN, leave as 0000
MTX_mode: client	Working mode is TCP client
MTX_model: MTX-T [3-N]	MTX model
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_filter: 100	Only frames beginning by byte 100 will be sent
MTX_msToSend: 200	Minimum stop with no data to be a new frame
TCP_IP: 200.1.2.3	IP of the server the MTX-Tunnel will connect to
TCP_port: 20010	Minimum stop with no data to be a new frame

Details:

- In a MODBUS frame the first byte is the address. Therefore, when specifying the value 100in the MTX_filter parameter, only the frames beginning with 100 will be sent to the central server.
- If, for example, we want to resend to the server only the frames whose modbus command is the write command (0x10), since this command is the second byte in a modbus frame, the parameter MTX_filter will have to be:

  MTX_filter: 100,16

  (100 corresponding to the modbus address 100 and 16 to the decimal value of the 0x10 command)

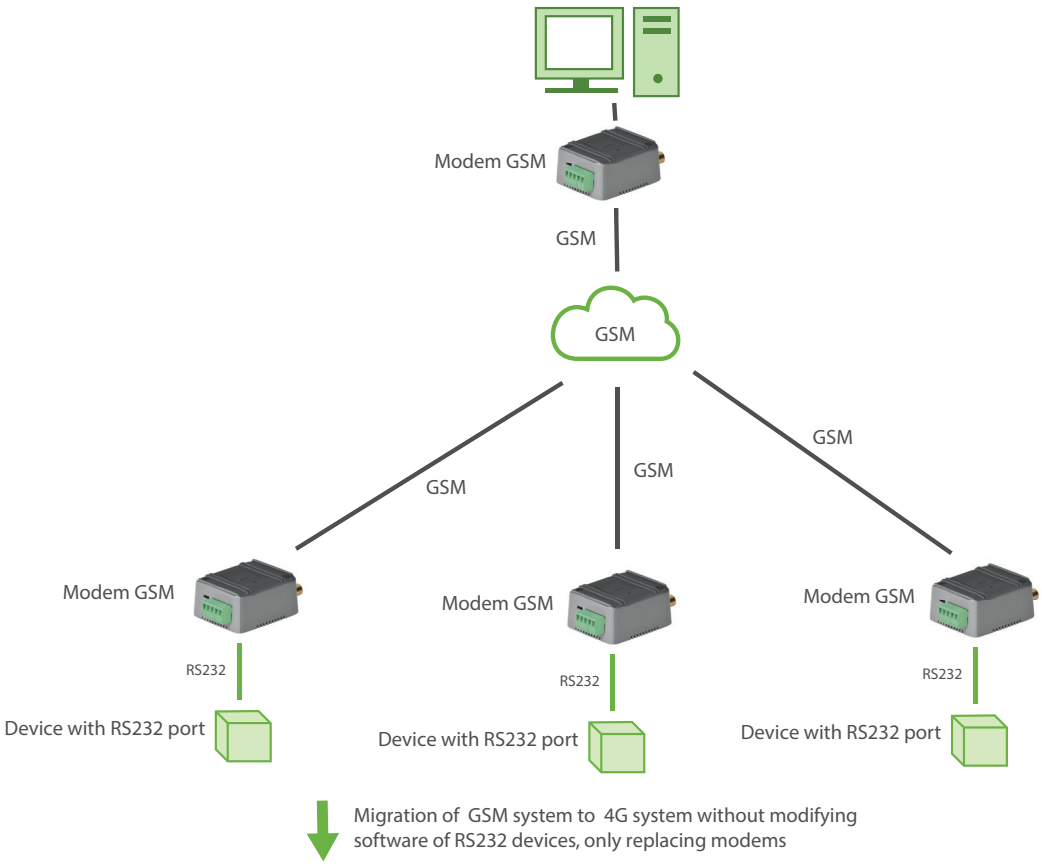
2.14 EXAMPLE: Conversion of a GSM communications system into an IP (3G/2G) communication system.

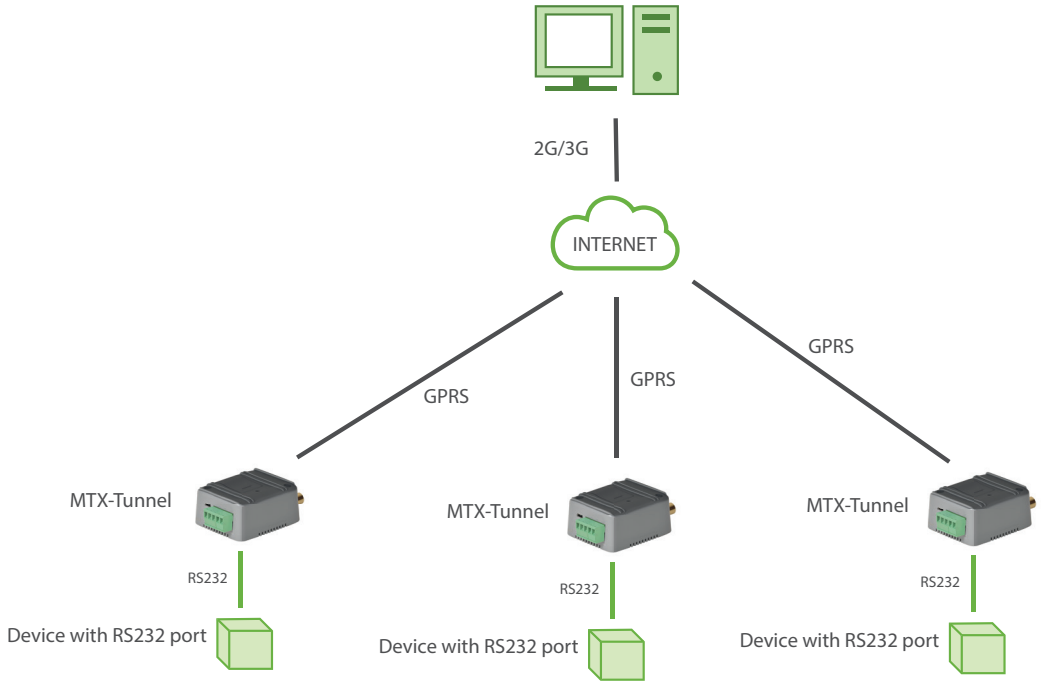
Scenario details:

- We have a series of old PLCs that use a GSM modem to make and receive conventional GSM calls
- We intend to replace the old GSM based communication system with one using 3G communications, modifying the previous system as little as possible. That is, without modifying the PLC software and modifying the server PC software as little as possible. Each PLC must be able to make calls, like until now, and must be able to receive them, but instead of making GSM calls, they will be IP (3G) connections. That is, it will only be necessary to replace the PLC current GSM modem with an MTX modem + MTX-Tunnel. It will also be necessary to replace the GSM modem connected to the PC server with a TCP/RS232 software converser
- For this, the MTX modem behaves as follows: To emulate the reception of GSM calls the modem remains in listen mode in a TCP port. For this application the chosen port is TCP 20010. When it receives a 3G/2G connection against that TCP port, the MTX modem must set a 3G/2G-Serial gateway the same way it would with a GSM conventional call. On the other hand, it must also be possible to establish a 3G/2G connection from the PLC emulating a GSM call. To do this, the PLC will send the typical connection command GSM ATD xxx.xxx.xxx.xxx (as if it was a GSM call but specifying an IP or DNS address instead of a phone number). The MTX-Tunnel must also be connected to this address via the 20010 port
- The PLC will control the DTR modem line, disconnecting 3G/2G communications with that line. The modem will send CONNECT when setting/receiving a connection and NO CARRIER when this ends

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel software





EXAMPLE of settings (file config.txt) for this scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control
COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 10	A value >5 is compulsory for this scenario
GPRS_autoTimeout: off	Mandatory off for this scenario
MTX_PIN: 0000	If SIM card does not have PIN, leave as 0000
MTX_mode: server	Working mode is TCP server
MTX_model: 199801393	MTX model
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_ATMux: modem	Compulsory mode for this scenario
MTX_urc: modem	Compulsory to receive CONNECT and NO CARRIER
MTX_DTR: modem	Compulsory for this scenario
TCP_port: 20010	Minimum stop with no data to be a new frame

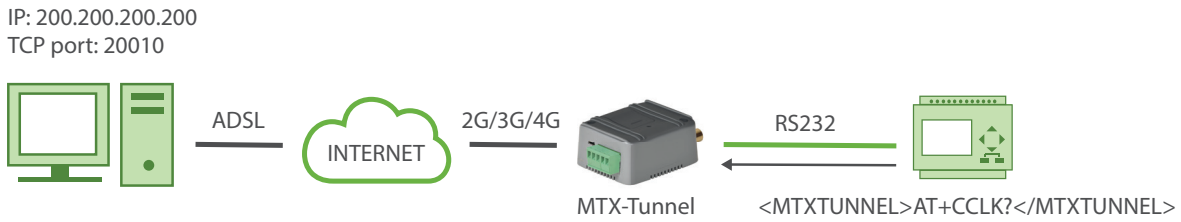
2.15 EXAMPLE: Use of MTX-Tunnel as 4G/3G/2G-RS232 gateway and for time synchronization of an external device connected to the modem.

Scenario details:

- A PLC system needs 4G/3G/2G-RS232 gateway to send data to a central server and to receive setting orders
- The modem must remain in listen mode, waiting for a server connection, in the TCP 20010 port. On the other hand, the PLC can send alarm messages to the server. In the event of the modem receiving data via the serial port, an alarm must open a socket client against the server to send the data. The destination IP of the alarms will be 200.200.200.200. The connection port will be to the TCP20010
- The PLC must also be able to use the modem to synchronize the time. The MTX-Tunnel must be set up so the time is automatically and periodically synchronized with an external server, with a command to read the modem time from the PLC
- The PLC serial port works at 9600bps, 8 data bits, 1 stop bit and no parity

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel software



EXAMPLE of settings (file config.txt) for this scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control
COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit

COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem will be connected permanently
MTX_PIN: 0000	If SIM card does not have PIN, leave as 0000
MTX_mode: server	Working mode is TCP server
MTX_model: 199801393	MTX model
MTX_TPServer: es.pool.ntp.org	Time server (the MTX must synchronize the time)
MTX_TPServer2: 2.europe.pool.ntp.org	Backup time server
MTX_TPProtocol: ntp	NTP protocol used
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_radioBand: europe	Not necessary if the MTX is installed in Europe
MTX_temporalClient: on	We enable the Temporary Client service
MTX_urc: off	We do not need the information messages URC
MTX_rssiLevel: 10	We enable the MTX-65i coverage LED
MTX_ATMux: on	Sends commands v. serial port (to check the time)
MTX_ATLimited: off	To execute any AT command
TCP_IP: 200.200.200.200	Server IP the Temporary Client will connect to
TCP_port: 20010	Minimum stop with no data to be a new frame
FIREWALL_enabled: off	Firewall disabled

Details:

- To read the time with an AT command, you should use the command AT+CCLK? In the following example PLC has a serial connection. To know the time, having configured the parameter “MTX_ATMux: on,” you will have to obtain information about the time by sending the following command:

```
<MTXTUNNEL>AT+CCLK?</MTXTUNNEL>
```

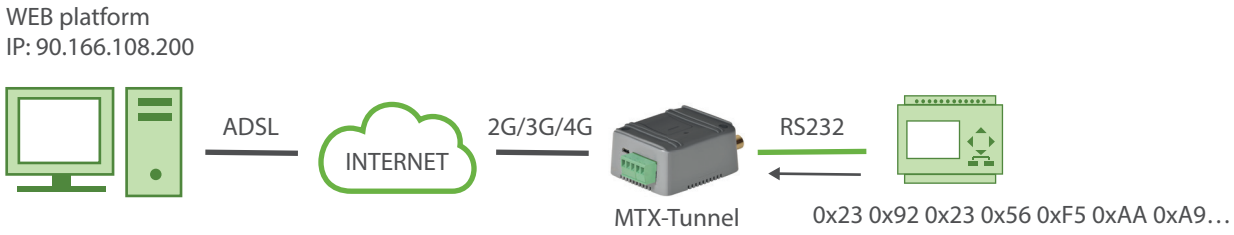
2.16 EXAMPLE: Use of MTX-Tunnel as a serial datalogger.

Scenario details:

- We have a device that sends bytes of data, which correspond to a sensor's readings, via its serial port every 10 seconds. The data will be a maximum of 256 bytes
- Taking readings every 10 seconds can be excessive for the application. MTX must only take one reading every minute (therefore eliminating 5 every minute), and send it to the web platform
- The modem must send the serial data that is collected to a Web platform via a JSON object. A timestamp should be attached
- The serial port configuration is 115200bps, 8 data bits, 1 stop bit and no parity

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel software



EXAMPLE of settings (file config.txt) for this scenario:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control
COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN by the GSM operator
GPRS_login: MOVISTAR	GPRS Login

GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_mode: server	GPRS/serial gateway in server mode
MTX_model: 199801393	MTX-Terminal modem model used
MTX_TPServer: es.pool.ntp.org	Time server (the MTX must synchronize the time)
MTX_TPServer2: 2.europe.pool.ntp.org	Backup time server
MTX_TPProtocol: ntp	NTP protocol used
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_ATEmbedded: on	To send AT commands from Web platform
MTX_urc: off	We don’t need URC info messages
MTX_ATLimited: off	To execute any AT command
TELNET_enabled: on	To send remote commands, etc.
TELNET_login: user	Telnet Login
TELNET_password: 1234	Telnet Password
TELNET_firewall: off	MTX accepts connections from all IPs
TELNET_port:20023	Telnet port
SMS_allPhones: on	All phone numbers are authorized
SMS_ATEnabled: on	AT commands can be sent via SMS
SMS_ATResponse: on	SMS replies to AT commands
LOGGER_enabled: on	Logger activated

LOGGER_password: ID-12345678	Password to be sent to the Web platform
LOGGER_server: 90.166.108.200/json/set.asp?data=	Web platform address
LOGGER_registerSize: 600	Size of the record
LOGGER_numRegistersFlash: 200	Number of records in flash storage
LOGGER_numRegistersRam: 3	Number of records in RAM memory
LOGGER_serialFrequency: 6	1 of 6 pieces of data received is logged
LOGGER_serverLogin: user	Username of the Webserver
LOGGER_serverPassword: 1234	Password of the Webserver
LOGGER_ httpMode: getjson	Mode HTTP GET (JSON)

Details:

- The JSON object received by the server will have the following format:

  {“IMEI”:353234028104337,”TS”:”08/06/14 13:39:33”,”P”:”ID-12345678”,”TYPE”:”SERIAL” ,”SER”:”3132333435363738393300d0a”}

  Where:

  IMEI: is the modem’s IMEI

  TS: is the Timestamp (the date/time of the modem)

  P: The password of the user

  TYPE: JSON type

  SER: The serial data in hexadecimal format (2 digits per byte)

  This means that for every string logged by MTX-Tunnel, a JSON object will be sent via GPRS or 3G (depending on the model of the modem) in its previous format (received by HTTP GET in the “data” variable, as can be seen in the LOGGER_server parameter).

2.17 EXAMPLE: Use of MTX-Tunnel as a datalogger. Sending customized and periodic serial data for proprietary protocols.

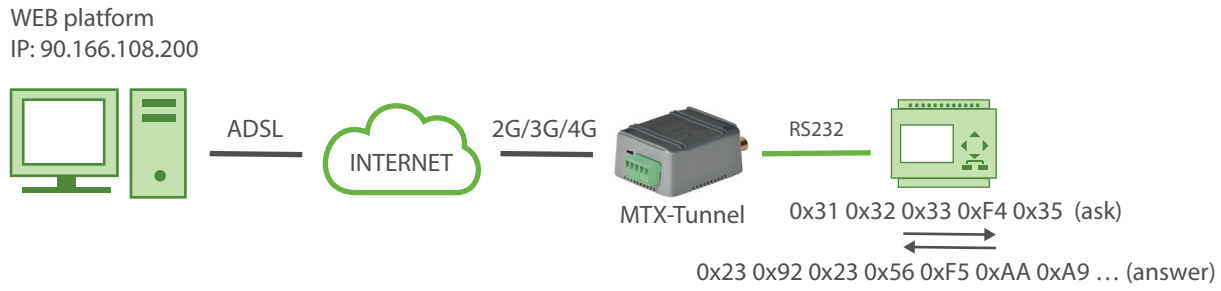
Scenario details:

- We have a device with an RS232 serial port. Internal registers need to be read and sent to a Web platform every 10 minutes
- In order to read the records, a proprietary protocol is used. This means that certain bytes of data (proprietary protocol) must be sent via the device's serial port for it to respond with the internal registry values
- Because of this, the modem must periodically send pre-programmed bytes of data via its RS232 serial port, collect the device's responses and, attaching a timestamp, send them to a Web server via a JSON object
- There are two pieces of data to be sent in order to be read every 10 minutes. These are:

  313233F435 and 42A12B42421F4343
- The serial port configuration is 115200bps, 8 data bits, 1 stop bit and no parity

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel software



EXAMPLE of settings (file config.txt) for this scenario:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control
COMM_autorts: off	No flow control

COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_mode: server	GPRS/serial gateway in server mode
MTX_model: 199801393	MTX-Terminal modem model used
MTX_TPServer: es.pool.ntp.org	Time server (the MTX must synchronize the time)
MTX_TPServer2: 2.europe.pool.ntp.org	Backup time server
MTX_TPProtocol: ntp	NTP protocol used
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_ATEmbedded: on	To send AT commands from Web platform
MTX_urc: off	We don’t need URC information messages
MTX_ATLimited: off	To execute any AT command
TELNET_enabled: on	To send remote commands, etc.
TELNET_login: user	Telnet Login
TELNET_password: 1234	Telnet Password
TELNET_firewall: off	MTX accepts connections from all IPs
TELNET_port: 20023	Telnet port

SMS_allPhones: on	All phone numbers are authorized
SMS_ATEnabled: on	AT commands can be sent via SMS
SMS_ATResponse: on	SMS replies to AT commands
LOGGER_enabled: on	Logger activated
LOGGER_password: ID-12345678	Password to be sent to the Web platform
LOGGER_server: 90.166.108.200/json/set.asp?data=	Web platform address
LOGGER_registerSize: 600	Size of the record
LOGGER_numRegistersFlash: 200	Number of records in flash storage
LOGGER_numRegistersRam: 3	Number of records in RAM memory
LOGGER_serverLogin: user	Username of the Webserver
LOGGER_serverPassword: 1234	Password of the Webserver
LOGGER_serialData1: 313233F435	First data to be sent
LOGGER_serialData2: 4141204242204343	Second data to be sent
LOGGER_ serialPeriod: 600	Data is sent every 600 secs (10 mins)
LOGGER_ httpMode: getjson	Mode HTTP GET (JSON)
LOGGER_serialFrequency: 1	We want to log all answers

Details:

- The JSON object received by the server will have the following format:

  {“IMEI”:353234028104337,”TS”:”08/06/14 13:39:33”,”P”:”ID-12345678”,”TYPE”: “SERIAL”,”SER”:”3132333435363738393300d0a”}

  Where:

  IMEI: is the modem’s IMEI

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

TS: is the Timestamp (the date/time of the modem)

P: The password of the user

TYPE: JSON type

SER: The serial data in hexadecimal format (2 digits per byte)

This means that for every string logged by MTX-Tunnel, a JSON object will be sent via GPRS or 3G (depending on the model of the modem) in its previous format (received by HTTP GET in the "data" variable, as can be seen in the LOGGER_server parameter).

The MTX-Tunnel will send a JSON object for every reading that is taken. IN the case of the example, 2 JSON objects will be received every 10 minutes.

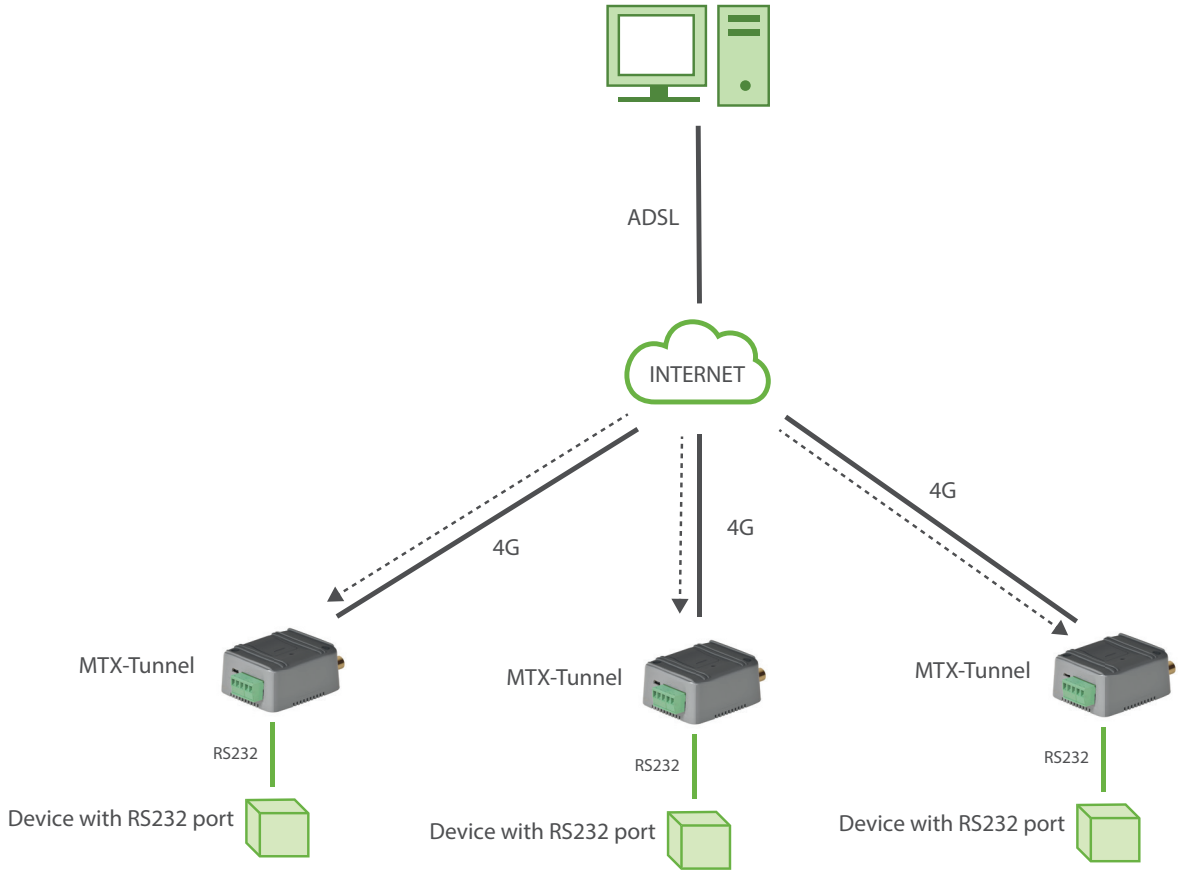
2.18 EXAMPLE: 3G-Serial transparent gateway with SNMP service activated.

Scenario details:

- There are a few hundreds of devices with RS232 port (configurated at a 115200,8,N,1 and flow control HW). We need to monitor them from a Computer with Internet connection
- It should be possible to access RS232 devices at any moment, therefore the modem connected to the serial port of the device should be connected to 3G 100 % of the time waiting for a connection (or to 2G if there is no 3G coverage). The modems will remain waiting for incoming connections through TCP 20010 port to create 3G-RS232 transparent gateway
- SIM cards with private APN providing fixed IP address will be used
- The modems should be supervised by SNMPv2c protocol. By means of the protocol it should be possible to read the state of the modems as well as their configuration. Via SNMP it should be also possible to change remote configuration, as well as carry out remote resets, if necessary, and fulfill any other task

Solution:

Modem MTX-IoT [4-S-N-N]-STD-N+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file) for the scenario that each MTX-Tunnel should have:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: mycompany.movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
GPRS_mode: auto	3G and 2G connection in case there is no 3G
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_model: 199801393	MTX-Terminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will be not output
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_urc: off	We don’t need URC information messages
MTX_ATLimited: off	To execute any AT command
TCP_port: 20010	TCP port used
FIREWALL_enabled: off	Any incoming connection form any IP is allowed

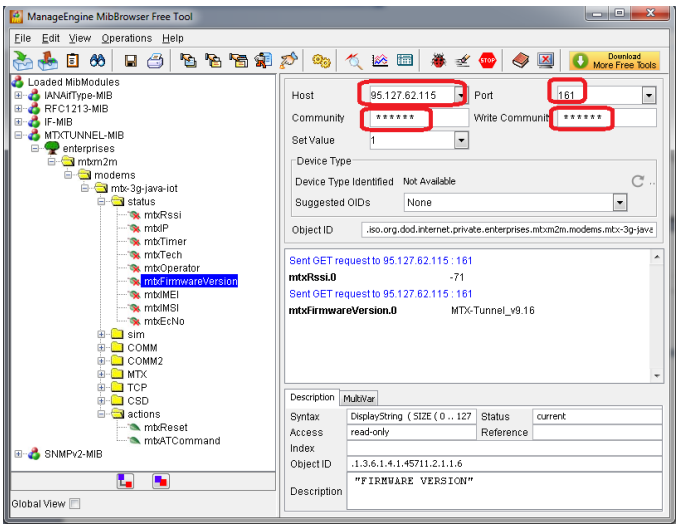
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Port chosen for Telnet
SMS_allPhones: on	All phone numbers are authorized
SMS_ATEnabled: on	AT commands via SMS enabled
SMS_ATResponse: on	SMS responses to sent AT commands enabled
SNMP_enabled: on	SNMP service enabled
SNMP_port: 161	UDP port forSNMP. 161 is standard port
SNMP_community: public	“Password” for GET operations (reading)
SNMP_communityW: public	“Password” for SEToperation (writing)

Details:

- To download MIBS database for SNMP use the following link:

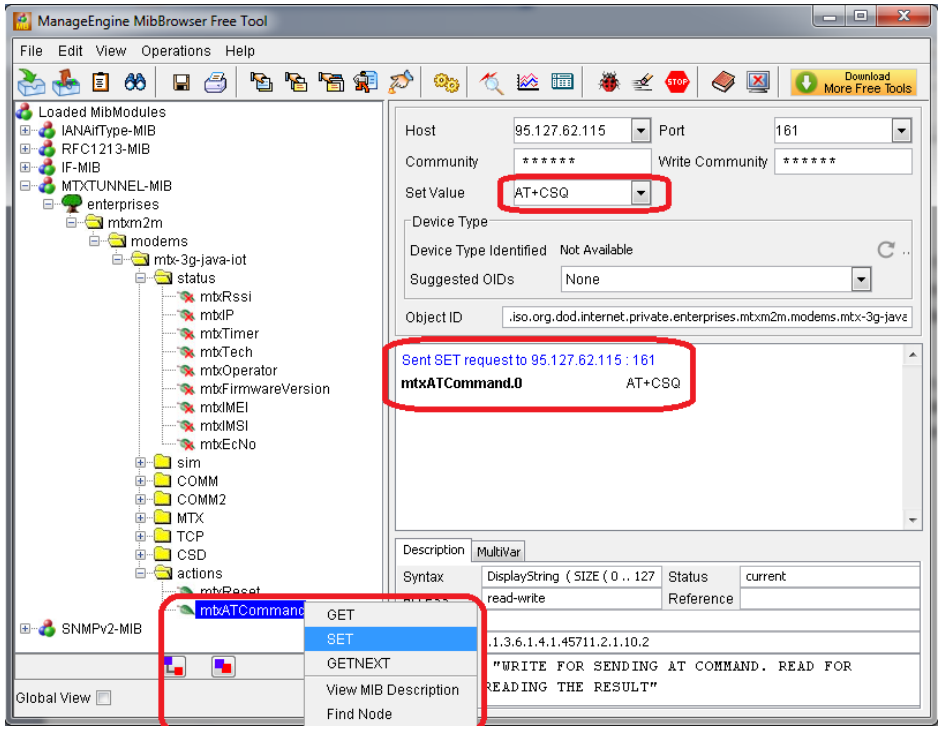
  https://www.dropbox.com/s/g6bfa0xxh4xcykp/MTXTUNNEL-MIB?dl=0

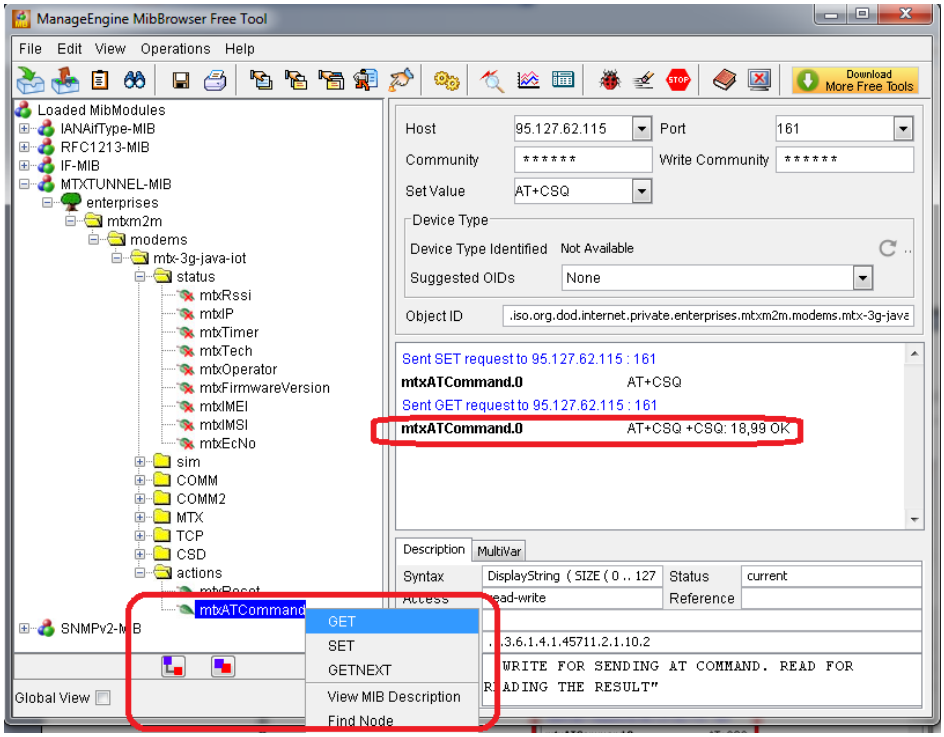
- The MIB provided in the previous link can import it in its SNMP management software. Here is an example of SNMP query where the coverage (in dBm) and firmware version were read.If you want to try out the same software, be sure to fill in the fields marked in red



- Use 2 OIDs to carry out actions: the one indicated as mtxReset and the one indicated as mtxATCommand. The first one will help you to execute a remote reset of the device. The second one will allow to execute an AT command in the modem the same way it would do it through Telnet,SMS...

  To execute an AT command via SNMP, simply send it to be executed together with a SET command of the OID mtxATCommand. Then, to read the result of the AT command, execute a GET command of the same OID. The execution of the command AT+CSQ for coverage reading is shown on the screen below





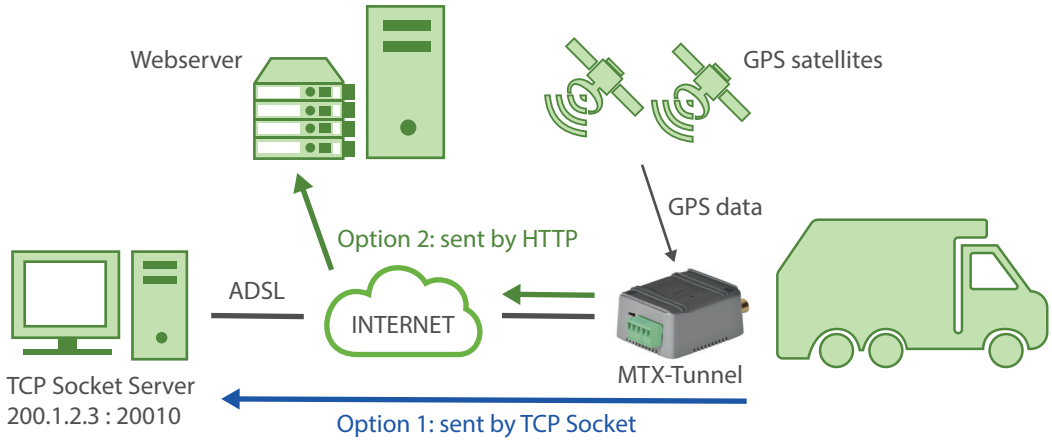
2.19 EXAMPLE: Sending GPS positions in real time to server using TCP socket or HTTP.

Scenario details:

- Some port facilities dispose of vehicles that lack being equipped with a real-ime GPS locating system
- Being a real-time system it does not need to save GPS locations history to the internal datalogger of the device in order to be sent when it later (as it is done in a conventional fleet control system),but these GPS locations should be sent as soon as possible to a sentral server, every second, if possible
- GPS locations should be sent in JSON format to the central server, and the transmission method should be chosen between TCP Socket (the quickest method with the minimal rate of 1-2 seconds) or through HTTP (it is a bit slower). The IP address for transmission via socket is 200.1.2.3 and the port is TCP 20010

Solution:

Modem MTX-IoT [4-S-N-N]-STD-N-GPS+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file) for the indicated scenario. Solution for quick communication via Socket TCP

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	8 bit data
COMM2_autocts: off	No flow control

COMM2_autorts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Permanent 3G session
GPS_mode: socket	Sending location vía SOCKET TCP
GPS_period: 2	Attempt to send GPS location every 2 sconds
GPS_ip: 200.1.2.3	IP address receiving GPS locations
GPS_port: 20010	TCP port where GPS location is sent
MTX_mode: none	We will not use TCP serial port gateways
MTX_pin: 0000	The SIM card PIN (if there is any)
MTX_model: 199801452	MTX modem model being used
MTX_atLimited: off	No limits for AT commands
MTX_numGSMErrors: 180	N. of GSM errors to reset
MTX_ping: 30	Ping every 30 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	Commands SMS sent from any mobile phone
SMS_sendIP: on	The modem will send IP to a missed call or SMS
SMS_ATEnabled: on	It is possible to send commands to MTX via SMS
SMS_ATResponse: on	SMS replies to sent AT commands

FIREWALL_enabled: off	Any incoming connection form any IP is allowed
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Port chosen for Telnet

Details:

- MTX modem, having connected to a 3G network, will connect to the IP address 200.1.2.3 and the port TCP 20010
- GPS location will be sent via sending a JSON object. Here is an example of JONSON transmission:

  {“IMEI”:”358884051192529”,”TYPE”:”GPS”,”DATE”:”2016/11/04”,”TIME”:”20:44:35”, “LAT”:”41.62963”,”NS”:”N”,”LON”:”2.3609116”,”EW”:”E”,”ALT”:”185.7”,”SPE”:”0.25”, “COU”:”0.00”,”TA”:”3”,”HPO”:”1.90”,”VDO”:”1.44”,”SAT”:”4”}

  Where:

  IMEI: the unique identification number of the modem

  TYPE: JSON type (GPS in this case)

  DATE: UTC date returned via GPS

  TIME: UTC time returned via GPS

  LAT: GPS Latitude

  NS: N=North, S=South

  LON: GPS Longitude

  EW E=East , W=West

  ALT: Altitude (meters)

  SPE: Speed (km/h)

  COU: Coure

  STA: Status. 0=no Fix, 2=2D, 3=3D

  HPO: Horizontal accuracy indication. The lower the better

  VDO: Vertical accuracy indication. The lower the better

  SAT: satellite Nº

EXAMPLE of configuration (config.txt file) for the scenario. Solution for quick communication via HTTP

COMM2_baudrate: 9600	Serial port (where internal GPS is connected) rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autorts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Permanent 3G session
GPS_mode: logger	Sending location via logger (HTTP)
GPS_period: 30	Attempt to send GPS location every 30 sconds
LOGGER_enabled: on	Logger activated to store GPS locations
LOGGER_httpMode: getjson	JSON sent via HTTP GET
LOGGER_password: 12345678	JSON user's field
LOGGER_server: www.myServer.com/page.asp?data=	Web platform address where Son is sent
LOGGER_registerSize: 500	Size of the record
LOGGER_numRegistersRam: 2	Only RAM memory (real time)
LOGGER_numRegistersFlash: 0	Only RAM memory real time
MTX_mode: none	We will not use TCP serial gateways
MTX_pin: 0000	The SIM card PIN (if there is any)

MTX_model: 199801452	MTX modem model being used
MTX_atLimited: off	No limits for AT commands
MTX_numGSMErrors: 180	N. GSM errors to reset
MTX_ping: 30	Ping every 30 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_TPServer: es.pool.ntp.org	Time server (MTX has to synchronized the time)
MTX_TPServer2: 2.europe.pool.ntp.org	Backup time server
MTX_TPProtocol: ntp	NTP protocol used
SMS_allPhones: on	SMS with commands can be sent from any cell
SMS_sendIP: on	Modem will reply with its IP to a missed call/SMS
SMS_ATEnabled: on	It is possible to send commands to MTX via SMS
SMS_ATResponse: on	SMS replies to sent AT commands
FIREWALL_enabled: off	Any incoming connection form any IP is allowed
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Port chosen for Telnet

Details:

- Stores locations collected every 30 secs in LOGGER_ and sends it to an URL in LOGGER_server
- Not necessary to store locations. Recommended to set LOGGER_numRegistersFlash to “0”
- JSON format is exactly the same as shown in the previous method
- If the method GPS_mode is specified, fasthttp can reduce the value of GPS_period to 10 secs. Keep in mind that when applying this method, LOGGER_ will be only used for sending GPS positions via HTTP in real time, not for additional tasks like MODBUS registers readings, etc.

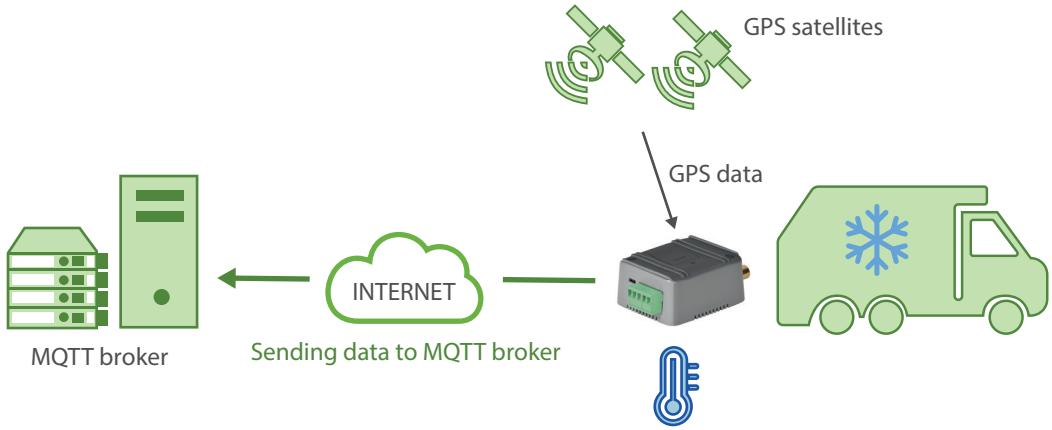
2.20 EXAMPLE: Sending GPS positions, temperature and trailer opening detection for cold chain monitoring.

Scenario details:

- We have a freezer truck transporting frozen goods. We need to install a GPS location device that also allows to monitor the temperature of the truck as well as to control its door
- For that, the GPS location device must check the GPS location each 60 seconds and send it to a central server via MQTT. In the sent data there must be a field with the temperature (the temperature sensor must have a temperature range from -50ºC to 80ºC). We also need to check the opening of the truck door. In case there is an opening, the system must gather the information about the opening time (date/time) and send the GPS position where the opening was made
- The truck will drive by places with no 4G/3G/2G coverage. So we don't lose any location, temperature or door opening data, the device must store all data on its flash memory in order to send them when there is 4G/3G/2G connectivity

Solution:

Modem MTX-IoT [4-S-N-N]-STD-N-GPS+ MTX-TEMP-RS232+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file) for the indicated scenario. Solution for quick communication via Socket TCP.

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control

COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM_power: on	Activation of the port power lines
COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	8 bit data
COMM2_autocts: off	No flow control
COMM2_autorts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Permanent 3G session
GPS_mode: logger	GPS working vía logger
GPS_period: 60	Attempt to send GPS location every 60 sconds
TEMPERATURE_enabled: on	Temperature sensor enabled
TEMPERATURE_period: 0	Temperature will be read along with GPS
MTX_mode: none	We will not use TCP serial port gateways
MTX_pin: 0000	The SIM card PIN (if there is any)
MTX_model: 199801452	MTX modem model being used
MTX_atLimited: off	No limits for AT commands

MTX_numGSMErrors: 180	N. of GSM errors to reset
MTX_ping: 30	Ping every 30 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_redLed: gps	Red LED lights when there's NO valid GPS location
MTX_TPServer: es.pool.ntp.org	Time server 1
MTX_TPServer2: 2.europe.pool.ntp.org	Time server 2
MTX_TPProtocol: ntp	Time synch protocol
SMS_allPhones: on	Commands SMS sent from any mobile phone
SMS_sendIP: on	The modem will send IP to a missed call or SMS
SMS_ATEnabled: on	It is possible to send commands to MTX via SMS
SMS_ATResponse: on	SMS replies to sent AT commands
FIREWALL_enabled: off	Any incoming connection form any IP is allowed
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Port chosen for Telnet

Details:

- Connections: the modem MTX-IoT [4-S-N-N]-STD-N-GPS has a serial port in the DB9 connector. That's where the temperature sensor MTX-TEMP-RS232 will be connected. Both devices are DCE (DB9 female) so you will need a converter Null-Modem DB9-DB9



To detect if the door is open/closed we will use digital input GPIO2. According to the table in AnnexA, it corresponds with PIN 11 of MTX's DB15. That is, if we connect the PIN 11 to PIN 14 of DB15 the modem will think that input has a "1" (for instance, open door) and if PIN 11 is not connected to anything the modem will think that input has a "0" (for instance, closed door)

- With the previous configuration, 3 data frames will be sent.

{"IMEI":354033090128458,"TYPE":"GPS","P":"","DATE":"2019/01/13","TIME":"11:35:30","LAT":"41.62964","NS":"N","LON":"2.361005","EW":"E","ALT":"187.9","SPE":"0","COU":"000.0","STA":"3","HPO":"1.2","VDO":"1.9","SAT":"08","TEMP":"16.9"}

Where:

IMEI: the unique identification number of the modem

TYPE: JSON type (GPS in this case)

DATE: UTC date returned via GPS

TIME: UTC time returned via GPS

LAT: GPS Latitude

NS: N=North, S=South

LON: GPS Longitude

EW E=East , W=West

ALT: Altitude (meters)

SPE: Speed (km/h)

COU: Coure

STA: Status. 0=no Fix, 2=2D, 3=3D

HPO: Horizontal accuracy indication. The lower the better

VDO: Vertical accuracy indication. The lower the better

SAT: satellite Nº

TEMP: temperature read from the temperature sensor

The second data frame kind happens when there's a change in the digital input 2 that controls if the door is open or closed. This data frame "TYPE":"IOS" will send E/S data (digital I/Os, analog inputs) and the GPS location at those moments as well as the temperature. For example:

{"IMEI":354033090128458,"TS":"13/01/19 10:45:40","TYPE":"IOS","P":"","IO1":0,"IO2":1,"IO3":0,"IO4":0,"IO5":0,"IO6":0,"IO7":0,"IO8":0,"IO9":0,"IO10":0,"AD1":1259,"AD2":1333,"CO1":"1","CO2":"3","CO3":"3","GPSDATA":{"DATE":"2019/01/13","TIME":"10:45:41","LAT":"41.62964","NS":"N","LON":"2.36099","EW":"E","ALT":"195.2","SPE":"0","COU":"000.0","STA":"3","HPO":"1.3","VDO":"2.3","SAT":"07","TEMP":"16.4"}}

IMEI: the unique identification number of the modem

TYPE: JSON type (IOS in this case)

TS: TimeStamp (modem time when the event happened)

IOx: digital input x value

ADx: analog input x value

COx: pulse counter x value

GPSDATA.DATE: date given by GPS module

GPSDATA.TIME: time given by GPS module

GPSDATA.LAT: GPS latitude

GPSDATA.NS: N=North, S=South

GPSDATA.LON: GPS longitude

GPSDATA.EW: E=East, W=West

GPSDATA.ALT: altitude

GPSDATA.SPE: speed (km/h)

GPSDATA.COU: course

GPSDATA.STA: status, 0=no Fix, 2=2D, 3=3D

GPSDATA.HPO: horizontal precision, the lower the better

GPSDATA.VDO: vertical precision, the lower the better

GPSDATA.SAT: how many satellites are being used

GPSDATA.TEMP: temperature read from the temperature sensor

Note there are 2 hours back in the JSON. One in the TS field (modem time) and another one in the GPS data (GPS module time). There can be a difference in seconds due to the moment when the data is read.

The third kind of data frame that is produced are the DNS data frames. These data frames are configured (DNS_period) to be sent every 120 seconds. They show important information about the device time, its IP address, GSM coverage, technology used, digital and analog E/S status, GPS location, temperature, meter boxes and information regarding the GSM used.

This data frame is very interesting for several reasons. On one hand, the data frame shows data in real time. Sometimes, when the vehicle drives through areas with low GPS coverage (due to weather, geography), this DNS data frame is useful because supports information from the CID field, and that's why it is possible to obtain the approximate location of the vehicle thanks to GSM location. It is also useful in areas of low 4G/3G/2G coverage. In case of long time periods, a lot of data is storaged until coverage is back. During those minutes we can still receive the GPS location information in real time thanks to the DNS data frame, without the need to wait to download the pending history.

Example of the DNS data frame:

{“IMEI”:354033090128458,”TYPE”:”DNS”,”TS”:”13/01/19 11:58:36”,”P”:””,”IP”:”95.124
.172.178”,”CSQ”:14,”TECH”:”4G”,”VER”:”10.04”,”AUX”:””,”MOD”:”MTX-IoT [4-S-N-N]-STD-N-
GPS”,”VCC”:12000,”IO1”:0,”IO2”:0,”IO3”:0,”IO4”:0,“IO5”:0,”IO6”:0,”IO7”:0,”IO8”:0,”IO9”:0,
”IO10”:0, “AD1”:1284,”AD2”:1333,”GPSDATA”:{“DATE”:”2019/01/13”,”TIME”:”11:58:37”,
“LAT”:”41.62964”,”NS”:”N”,”LON”:”2.361005”,“EW”:”E”,”ALT”:”187.9”,”SPE”:”0”,
”COU”:“000.0”,”STA”:”3”,”HPO”:”1.3”,”VDO”:”1.8”,“SAT”:”07”,”TEMP”:”16.9”},”CO1”:”1”,
”CO2”:“3”,”CO3”:”3”,”CID”:”07;21E0;13B6D0A;87;--”}

IMEI: the unique identification number of the modem

TYPE: JSON type (IOS in this case)

TS: TimeStamp (modem time when the event happened)

IP: modem IP address

CSQ: modem coverage (0 ... 31)

TECH: technology used in that moment (4G/3G/2G)

VER: MTX-Tunnel firmware version

MOD: MTX model (field MTX_model)

VCC: MTX supply voltage (in millivolts)

IOx: digital input x value

ADx: analog input x value

COx: pulse counter x value

CID: cell ID of the telephone station used

GPSDATA.DATE: date given by GPS module

GPSDATA.TIME: time given by GPS module

GPSDATA.LAT: GPS latitude

GPSDATA.NS: N=North, S=South

GPSDATA.LON: GPS longitude

GPSDATA.EW: E=East, W=West

GPSDATA.ALT: altitude

GPSDATA.SPE: speed (km/h)

GPSDATA.COU: course

GPSDATA.STA: status, 0=no Fix, 2=2D, 3=3D

GPSDATA.HPO: horizontal precision, the lower the better

GPSDATA.VDO: vertical precision, the lower the better

GPSDATA.SAT: how many satellites are being used

GPSDATA.TEMP: temperature read from the temperature sensor

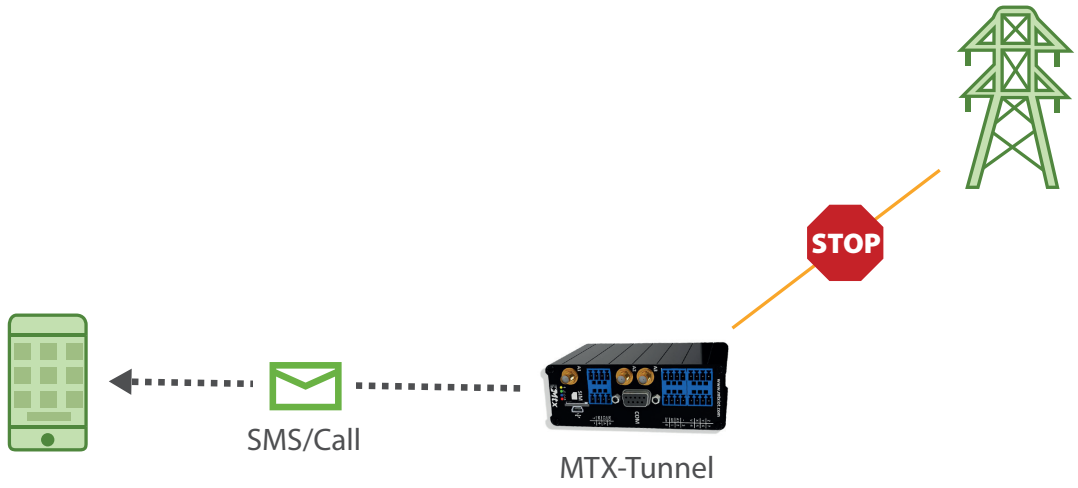
2.21 EXAMPLE: Alarms sent by SMS and VOICE call upon detection of 220V power loss and upon return of power.

Scenario details:

- It is necessary to monitor a 220V power system to prevent merchandise from industrial freezers
- For this, a battery modem will be used. This modem should send an SMS alert to 5 different phone numbers when a 220V power failure is detected. You should do the same when it detects that the power has recovered
- In addition, only in case of power loss, the modem will make a voice call to each of the 5 phones in order to make the call more noticeable

Solution:

Modem MTX-IOT-S [4-N]+firmware MTX-Tunnel



Example of configuration (config.txt file) for the indicated scenario:

MTX_pin: 0000	El PIN de la tarjeta SIM
MTX_mode: none	Modo de trabajo ninguno
MTX_model: 199802407	Modelo de MTX seleccionado para el escenario
ALARM_powerEnabled: on	Alarma SMS por fallo alimentación habilitada
ALARM_powerMessageOn: Alimentacion on	Texto alarma alimentación reestablecida
ALARM_powerMessageOff: FALLO ALIMENTACION 220V	Texto alarma SMS por fallo alimentación
ALARM_powerVoiceCall: on	Llamada de voz cuando se produzca el corte
ALARM_smsNumber1: +34666123455	Teléfonos de envío de alarma
ALARM_smsNumber2: +34666123456	
ALARM_smsNumber3: +34666123457	
ALARM_smsNumber4: +34666123458	
ALARM_smsNumber5: +34666123459	

Details:

- The voice call is only made when the power failure occurs. When power is restored, only SMS message is sent, no voice call
- Remember that voice calls are only indicative (to increase the awareness of the alarm), that is, the receiver will receive the call, but if you go off-hook you will not hear any type of audio

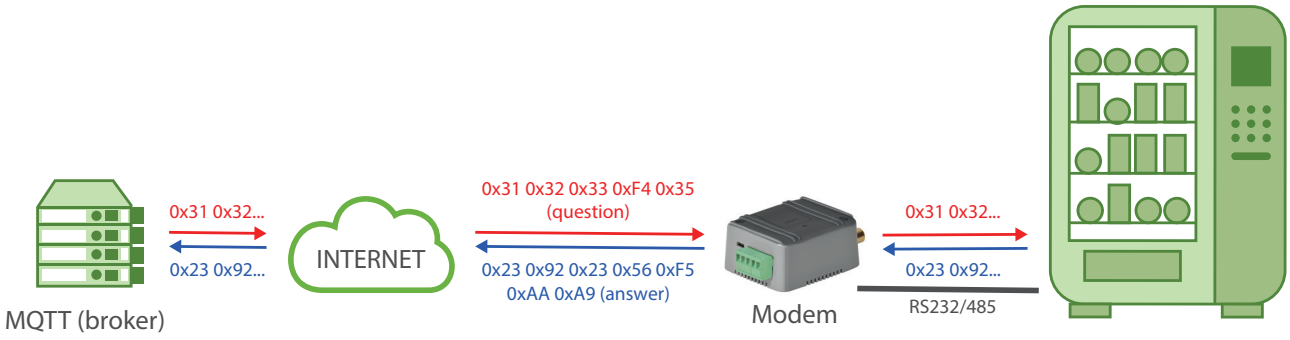
2.22 EXAMPLE: Access RS232/RS485 port of a device via MQTT (for serial RAW data management). Transparent gateway "RS232/485 - MQTT".

Scenario details:

- There is a vending machine with an RS232 serial port. We want to connect the RS232 serial port of the vending machine directly to an MQTT platform, where the communications protocol will be implemented. When the reading protocol of the vending machine is implemented in the MQTT platform, the modem must behave as a "transparent RS232-MQTT gateway," providing the MQTT platform access to the machine's data bus in RAW format
- The modem must be able to be managed from the platform, being able to change its configuration remotely at any time via MQTT. The modem must also inform periodically about its status (coverage, technology used, ...)
- The modem must immediately inform the MQTT platform when it detects a change in one of its digital inputs, which will be connected to the open door sensor of the machine

Solution:

Modem MTX-IoT [4-S-N-N]-STD-N+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file) for the indicated scenario. Solution for quick communication via Socket TCP.

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit

COMM_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Permanent 3G session
MTX_pin: 0000	The SIM card PIN (if there is any)
MTX_mode: mqtt	MQTT serial gateways will be used
MTX_model: 199801436	MTX modem model being used
MTX_ping: 35	One ping every 35 min. without communications
MTX_pingIP: 8.8.8.8	Ping address
MTX_invertedCom: off	Gateway port RS485
MTX_msToSend: 100	No fragmented networks
MTX_ATLimited: off	No limitations to AT commands
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
SMS_allPhones: off	Send SMS with commands from any phone
SMS_sendIP: off	Modem won't respond IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
MQTT_enabled: on	MQTT service enabled

MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
MQTT_commrxtopic: [IMEI]/rx	Data received will be retransmitted via serial
MQTT_commtxtopic: [IMEI]/tx	Data received v/serial, retransmitted to this topic
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	Sending mode MQTT
DNS_mqttTopic: [IMEI]/dns	Topic where status data will be sent
DNS_extended: off	Extended data not sent (E/S, ADCs...)
DNS_period: 300	Every 300 secs. (5 mins.) there's a sending
LOGGER_enabled: on	Logger activated
LOGGER_registerSize: 300	Size of the record
LOGGER_numRegistersFlash: 1500	Number of records in flash storage
LOGGER_mode: mqtt	Sending mode MQTT
LOGGER_mqttTopic: [IMEI]/logger	Sending topic to MQTT broker of the data
LOGGER_ioEvent: on	Changes in digital inputs automatically sent

Details:

- A transparent RS232-MQTT gateway allows to exchange raw data between the MQTT broker (or another application connected to the MQTT broker) and the machine connected to the RS232 port of the modem. All data the modem receives through its RS232 serial port will be forwarded via MQTT to the broker's topic specified in the "MQTT_commtxtopic" parameter, and viceversa

- If, instead of specifying an RS232-MQTT gateway, we need an RS485-MQTT gateway (and the MTX modem has an RS485 port), we only need to set the “MTX_invertedCom: on” parameter in the configuration file
- Keep in mind that communications latencies may be somewhat greater than direct communication latencies since there is an intermediary (the mqtt broker) and the speed of communications will depend on the power of the latter. Set the timeout if necessary
- In addition, the modem will send its status periodically (every 300 sec) to the MQTT broker (in a JSON object) to the topic configured in the “DNS_mqtttopic” parameter. Similarly, each time there is a change in one of the digital inputs, the modem will send a JSON with the changes to the topic configured in the “LOGGER_mqtttopic” parameter

2.23 EXAMPLE: Data transmission by RS232 via SMS for the control of electronic equipment with proprietary protocols. Use of ALIAS to send non-printable characters.

Scenario details:

- It is necessary to send a series of commands by SMS to an electronic team, so that it executes actions
- The communication with the electronic equipment is through a proprietary binary protocol and is done through an RS232 port at 9600.8, n, 1. Therefore, it is necessary to implement an SMS-RS232 gateway
- The user needs to be able to send two types of messages by SMS. The message “OPEN” sends via serial port of the modem to the electronic equipment the data frame {0x01} {0x02} OPE1 {0x03}. The message “CLOSE” causes the sending by the serial port of the modem to the electronic equipment the data frame {0x01} [0x02} OPE2 {0x03}. Where 0x01, 0x02 and 0x03 are binary characters (not printable), while OPE1 and OPE2 are alphanumeric characters

Solution:

Modem MTX-4G-IOT+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file) for the indicated scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit

COMM_parity: none	No parity
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
MTX_PIN: 0000	SIM card PIN
MTX_mode: none	Work mode
MTX_model: 199801145	MTX model
SMS_alias1: OPEN>AT^MTXTUNNEL=RS232,0,<HEX> 0102</HEX>OPE1<HEX>03</HEX>	
SMS_alias2: CLOSE>AT^MTXTUNNEL=RS232,0,<HEX> 0102</HEX>OPE2<HEX>03</HEX>	
SMS_aliasResponse: result	Response isn't executed AT command

Details:

- When the modem receives an SMS message with the text "OPEN" the modem will send by its serial port the bytes 0x01 0x02 followed by the alphanumeric text OPE1 and ending with the 0x03 byte
- It is possible to put multiple sections with the <HEX> </ HEX> tags. Among them, all hexadecimal values must have 2 digits. That is, the value "1" should be written as "01"

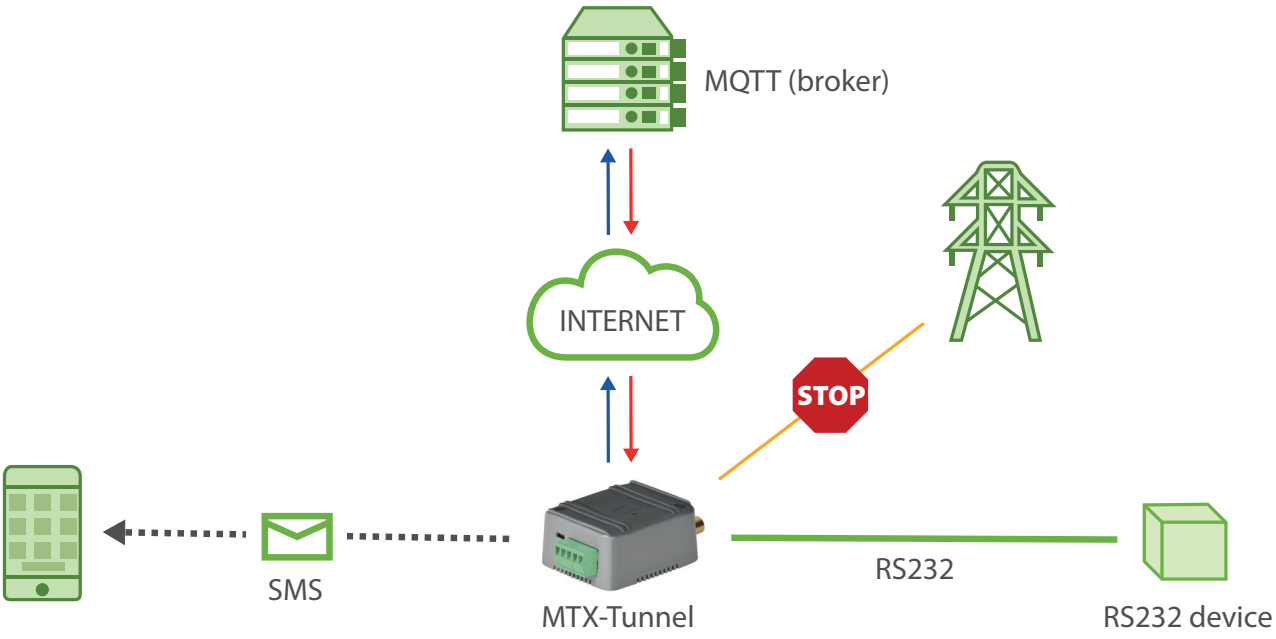
2.24 EXAMPLE: Alarms sent by SMS and MQTT message before detection of 220V power loss and before the return of power. Modem with supercap of 1 minute of autonomy.

Scenario details:

- A modem is required to make a transparent IP-RS232 gateway to read a counter. In addition, the modem must be able to report in the event of a power failure, so the modem must have sufficient autonomy to send an alarm when the event occurs.
- For this, a modem with supercap will be used, which will give it an approximate autonomy of 1 minute. This modem should send an SMS alert to a phone number when a 220V power failure is detected. You should do the same when it detects that the power has recovered
- In addition, the modem will send an alert message via MQTT to a control platform

Solution:

Modem MTX-Tunnel+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file) for the indicated scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data

COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Permanent 3G session
MTX_pin: 0000	The SIM card PIN (if there is any)
MTX_mode: server	RS232 IP gateway in server mode
MTX_model: 199801464	MTX modem model being used
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
LOGGER_enabled: on	Saves the alarm to be sent by MQTT
LOGGER_mode: mqtt	Logger will send alarm by MQTT
LOGGER_mqttTopic: [IMEI]/logger	MQTT topic where the alarm will be sent
MQTT_enabled: on	MQTT service enabled in the modem
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS, including port
MQTT_id: [IMEI]	Identifier

MQTT_attopic1: [IMEI]/AT	Topic to which subscribes to receive commands
MQTT_atrtopic: [IMEI]/ATR	Topic where it sends the responses
MQTT_qos: 1	Service quality
MQTT_keepalive: 300	300 seconds
MQTT_persistent: off	Not necessary
ALARM_powerEnabled: on	Power failure alarm enabled
ALARM_powerMessageOn: Alimentacion on	Refresh feed alarm text
ALARM_powerMessageOff: FALLO ALIMENTACION 220V	Text alarm SMS for power failure
ALARM_smsNumber1: +34666123456	Alarm sending telephone

Details:

- The modem model used has an internal supercap that allows it to have an approximate autonomy of 1 minute once it loses power
- When trying to reset the modem (turning off the power), remember to wait 1 minute for this reset to occur. Remember that you can also reset the modem by sending the command AT + CFUN = 1.1
- The MQTT message that the modem will send when an external power failure occurs will be formatted as the following example shows:

  {“IMEI”:357299070187619,”TS”:”30/03/04 12:55:50”,”TYPE”:”POWER”,“POW”:0,”VBAT”:4100}

- The MQTT message that the modem will send when power is restored will be formatted as shown in the following example:

  {“IMEI”:357299070187619,”TS”:”30/03/04 12:58:32”,”TYPE”:”POWER”,“POW”:1,”VBAT”:4102}

Where:

IMEI:	is the unique identifier of the modem
TS:	is the time stamp of when the event occurred
TYPE:	Indicates the type of data frame. POWER indicates power alarm
POW:	1 = external power supply active, 0 = external power supply inactive

VBAT: battery level in mV

- This example is valid for any MTX modem that has an internal battery

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

3. ANNEX: ULP Configuration and Example Scenarios

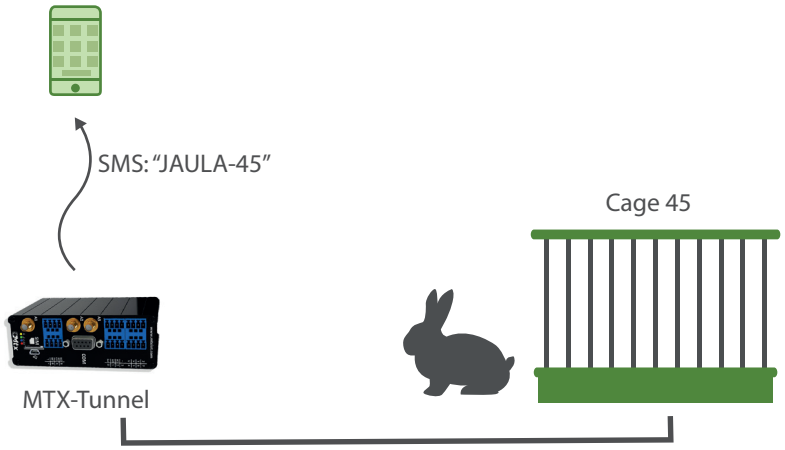
3.1 EXAMPLE: Sending an SMS alarm when the digital input is activated with ultra low power consumption.

Scenario details:

- Small cages have been placed in a forest to monitor veterinarily of a certain animal species. It is intended to send an alarm SMS when a digital input is activated that is connected to the door of some cages
- As the equipment is in the field, the MTX is powered by an internal battery (although an external battery can be used for more durability), so it must remain in ultra low consumption mode (~ 6 uAmps) until the closing of the door is detected. a cage (signal that an animal has entered), since the cage door lock detector will be connected to a digital input on the modem, specifically the digital input called “Tamper”
- Then, the moment the cage door is detected, the modem will wake up and send an SMS with the text “CAGE-45”
- After sending the SMS, the modem must return to ULP mode until the door is opened again

Solution:

MTX-IOT-S [4-N] modem+MTX-Tunnel



Config.txt configuration file:

MTX_model: 199801454	MTX-Terminal modem model: MTX-65-ULP
MTX_mode: none	GPRS-Serial tunnel not needed
ULP_enabled: on	ULP mode
ULP_sleepMode: minutes	Sleep mode
ULP_minutesoff: 0	Wake up by interrupt tamper input
ULP_secondsOn: 60	Modem sleeps after X seconds
ALARM_ulpEnabled: on	Alarm activated by ULP
ALARM_ulpMessage: JAULA-45	SMS text
ALARM_smsNumber1: +34666123456	Phone number to send SMS

Details:

- When the modem wakes up by activating the tamper input (digital dry contact input) it will try to send the alarm SMS to the specified numbers. In case of problems with SMS sending, it will retry every minute. If after 5 retries it is not possible to send the SMS, the modem will enter low power again until the tamper digital input is activated again

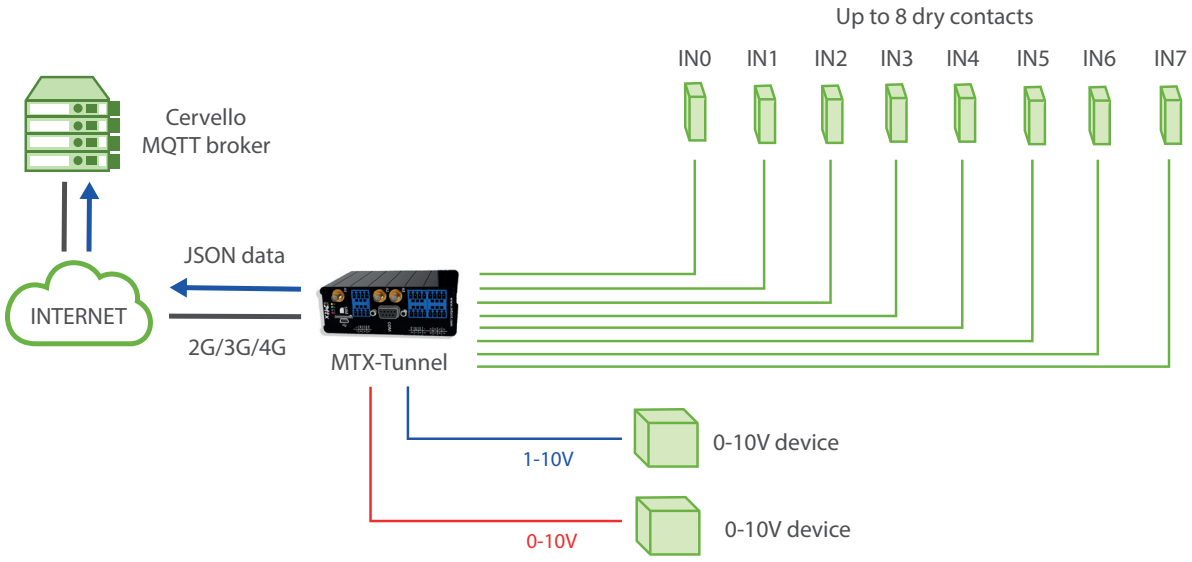
3.2 EXAMPLE: Periodic sending of digital and analog input value telemetry by 4G/3G/2G using ULP modem. Sending to MQTT broker.

Scenario details:

- It is intended to monitor various digital / analog inputs with a 4G/3G/2G modem
- The modem is going to be powered by batteries, so it must remain in ultra low consumption mode (~ 6 uAmps) waking up just to make the measurements and send them via 4G/3G/2G (depending on the modem model). The modem has an internal battery, but to increase durability, an external battery will be added
- Every 24 hours the MTX-Tunnel must wake up, read the status of the digital / analog inputs and send the readings to an MQTT broker
- After sending the information, the modem must return to ultra low consumption mode for another 24 hours

Solution:

MTX-IOT-S [4-N] modem+MTX-Tunnel



Config.txt configuration file:

GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password

GPRS_timeout: 0	Connected data while ULP is awake
MTX_model: 199801454	MTX terminal modem model used
MTX_mode: none	We don’t use gateways
ULP_enabled: on	Modem ULP service activated
ULP_sleepMode: minutes	Modem will be awake
ULP_minutesOff: 1438	Awake every 1438 minutes
ULP_secondsOn: 120	Awake during 2 minutes
MQTT_enabled: on	MQTT service in the modem activated
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS, including port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT1	MTX-Tunnel subscribes to this topic for commands
MQTT_qos: 1	Service quality
MQTT_keepalive: 300	Keep alive MQTT connection (300 seconds)
MQTT_persistent: off	We don’t need persistence
DNS_enabled: on	Sending status data activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: [IMEI]/dns	Topic where status data will be sent
DNS_extended: on	Sending extended data (E/S, ADCs, etc.)
DNS_period: 120	Every 120 seconds data is sent
GPIO_mode0: input	GPIO0 configured as an input
GPIO_config0: normal	No special configuration

GPIO_mode1: input	GPIO0 configured as an input
GPIO_config1: normal	No special configuration
GPIO_mode2: input	GPIO0 configured as an input
GPIO_config2: normal	No special configuration
GPIO_mode3: input	GPIO0 configured as an input
GPIO_config3: normal	No special configuration
GPIO_mode4: input	GPIO0 configured as an input
GPIO_config4: normal	No special configuration
GPIO_mode5: input	GPIO0 configured as an input
GPIO_config5: normal	No special configuration
GPIO_mode6: input	GPIO0 configured as an input
GPIO_config6: normal	No special configuration
GPIO_mode7: input	GPIO0 configured as an input
GPIO_config7: normal	No special configuration
ADC_mode0: voltage	ADC0 as voltage input
ADC_config0: normal	No special configuration
ADC_mode1: voltage	ADC1 as voltage input
ADC_config1: normal	No special configuration

Details:

- The frame sent to the server has the following JSON format:

  {“IMEI”:”354033091483894”,”TYPE”:”DNS”,”TS”:”2020-05-26T15:46:50Z”,”P”:””,”IP”:
  ”95.126.81.105”,”CSQ”:12,”TECH”:“4G”,”VER”:”11.07”,”AUX”:””,”MOD”:”199802407”,
  ”VCC”:12000,”CID”:”214;07;21E0;13B6D0A;405”,”IO0”:1,”IO1”:1,”IO2”:1,”IO3”:1,
  ”IO4”:1,”IO5”:0,“IO6”:0,”IO7”:0,”IO8”:0,”AD0”:0,”ADM0”:”voltage”,”AD1”:0,
  ”ADM1”:“voltage”,”POW”:1}

  Where:

IMEI:	is IMEI (modem identifier)
TYPE:	DNS frame type
TS:	Timestamp
IP:	current IP of the MTX
CSQ:	GSM coverage (0-31)
TECH:	Indicates if the modem is working in 2G, 3G or 4G
SEE:	MTX-Tunnel version
MOD:	Model. Indicates the value you have in the parameter MTX_model
VCC:	MTX supply voltage (in millivolts)
CID:	Identifier of the BTS station used (useful for location by cells)
IOx:	Digital input x
AD0:	Analog input 0
ADM0:	Analog input mode 0
AD1:	Analog input 1
ADM1:	Analog input mode 1
POW:	For equipment with internal battery. 1 = External power present

- Keep in mind that the ULP_minutesOff that the MTX-Tunnel will remain asleep start counting just as the modem enters ultra-low power mode
- From the MQTT platform you can end the ULP session at any time (without waiting for the end of the 2 minutes). To do this, you can send the command AT ^ MTXTUNNEL = SETULPSECONDS, 0 to the modem (to topic [IMEI] / AT1)
- 4Please note that the modem sends a JSON where the time is included. This example does not use time synchronization for consumption reasons, so the time received may not be correct. As the data is sent in real time, you can take the time from your own data reception server

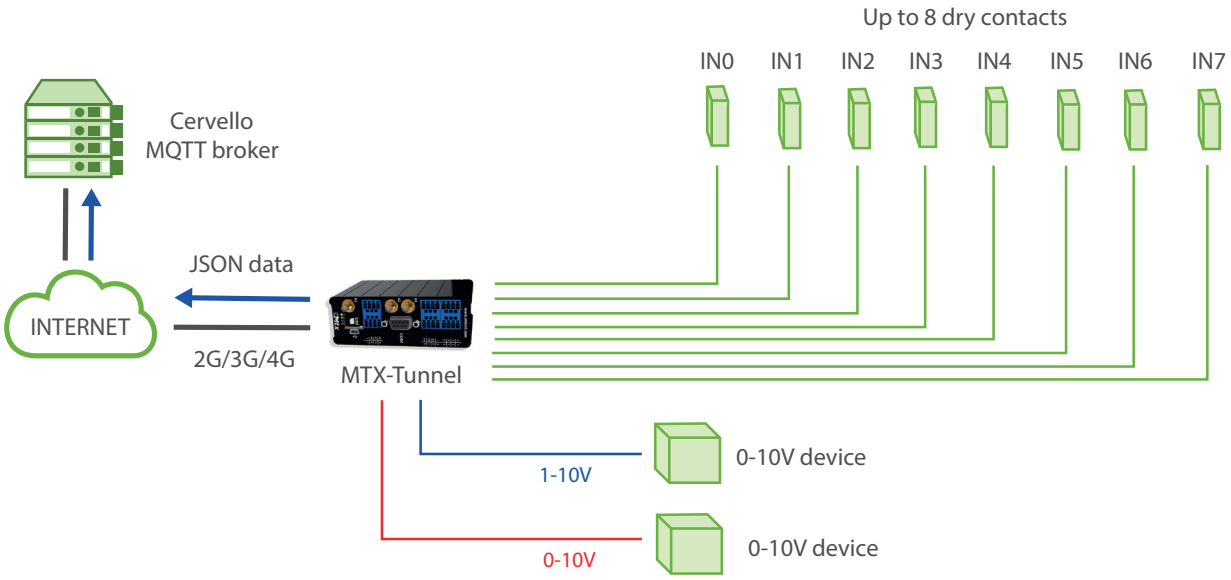
3.3 EXAMPLE: Sending digital and analog input value telemetry by 4G/3G/2G according to an event (not periodically) using ultra low power modem. Sending to MQTT broker.

Scenario details:

- It is intended to monitor various digital / analog inputs with a 4G/3G/2G modem
- The modem will be powered by batteries, so it must remain in ultra low power mode (~ 10 uAmps) until there is an activation in the digital input of the modem's tamper
- When such a change is detected on a tamper input, the modem should wake up, read the status of the digital / analog inputs, send them to an MQTT broker
- After sending the information, the modem must return to ultra low consumption mode until it detects another activation of a digital tamper input, at which time it will repeat the indicated process

Solution:

MTX-IOT-S [4-N] modem+MTX-Tunnel



Config.txt configuration file:

GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	Google DNS. Must be used if TCP_IP is set to DNS
GPRS_timeout: 0	Connected data while ULP is awake
MTX_model: 199801454	MTX terminal modem model used
MTX_mode: none	We don't use gateways
ULP_enabled: on	Modem ULP service activated
ULP_sleepMode: minutes	Modem will be awake
ULP_minutesOff: 0	Indefinitely asleep until tamper input changes
ULP_secondsOn: 60	Awake during 1 minutes
MQTT_enabled: on	MQTT service in the modem activated
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS, including port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT1	MTX-Tunnel subscribes to this topic for commands
MQTT_qos: 1	Service quality
MQTT_keepalive: 300	Keep alive MQTT connection (300 seconds)
MQTT_persistent: off	We don't need persistence
DNS_enabled: on	Sending status data activated
DNS_mode: mqtt	MQTT sending mode

DNS_mqttTopic: [IMEI]/dns	Topic where status data will be sent
DNS_extended: on	Sending extended data (E/S, ADCs, etc.)
DNS_period: 120	Every 120 seconds data is sent
GPIO_mode0: input	GPIO0 configured as an input
GPIO_config0: normal	No special configuration
GPIO_mode1: input	GPIO0 configured as an input
GPIO_config1: normal	No special configuration
GPIO_mode2: input	GPIO0 configured as an input
GPIO_config2: normal	No special configuration
GPIO_mode3: input	GPIO0 configured as an input
GPIO_config3: normal	No special configuration
GPIO_mode4: input	GPIO0 configured as an input
GPIO_config4: normal	No special configuration
GPIO_mode5: input	GPIO0 configured as an input
GPIO_config5: normal	No special configuration
GPIO_mode6: input	GPIO0 configured as an input
GPIO_config6: normal	No special configuration
GPIO_mode7: input	GPIO0 configured as an input
GPIO_config7: normal	No special configuration
ADC_mode0: voltage	ADC0 as voltage input
ADC_config0: normal	No special configuration

ADC_mode1: voltage	ADC1 as voltage input
ADC_config1: normal	No special configuration

Details:

The frame sent to the server has the following JSON format: {“IMEI”:”354033091483894”,”TYPE”:”DNS”,”TS”:”2020-05-26T15:46:50Z”,”P”:””,”IP”:”95.126.81.105”,”CSQ”:12,”TECH”:”4G”,”VER”:”11.07”,”AUX”:””,”MOD”:-“199802407”,VCC”:12000.”CID”:”214;07;21E0;13B6D0A;405”,”IO0”:1,”IO1”:1,”IO2”:1,”IO3”:1,”IO4”:1,”IO5”:0,”IO6”:0,”IO7”:0,”IO8”:0,”AD0”:0,”ADM0”:”voltage”,“AD1”:0,”ADM1”:”voltage”,”POW”:1}

Where

IMEI:	is IMEI (modem identifier)
TYPE:	DNS frame type
TS:	Timestamp
IP:	current IP of the MTX
CSQ:	GSM coverage (0-31)
TECH:	Indicates if the modem is working in 2G, 3G or 4G
SEE:	MTX-Tunnel version
MOD:	Model. Indicates the value you have in the parameter MTX_model
VCC:	MTX supply voltage (in millivolts)
CID:	Identifier of the BTS station used (useful for location by cells)
IOx:	Digital input x
AD0:	Analog input 0
ADM0:	Analog input mode 0
AD1:	Analog input 1
ADM1:	Analog input mode 1
POW:	For equipment with internal battery. 1 = External power present

- Keep in mind that the ULP_minutesOff that the MTX-Tunnel will remain asleep start counting just as the modem enters ultra-low power mode
- From the MQTT platform you can end the ULP session at any time. To do this, you can send the command AT ^ MTXTUNNEL = SETULPSECONDS, 0 to the modem (to topic [IMEI] / AT1)

- The modem sends a JSON with the time. As the data is sent in real time, you can take the time from your own data reception server

3.4 EXAMPLE: Timing alarm activation of IP-RS232 tunnel using MTX-ULP.

Scenario details:

- A series of meteorological stations with RS232 port are available. These run on batteries and continuously collect meteorological parameters that they store inside. It is necessary to collect these meteorological data through 4G/3G/2G once a day, but without compromising the consumption of the system, that is, a minimum consumption is required by the 4G/3G/2G modem connected to the meteorological station
- For this reason, the modems must remain in ultra low consumption mode (~ 10 uAmps) for 24 hours.
- Every 24 hours the modems must wake up, connect to the data network (4G/3G/2G) and open a client TCP / IP-RS232 gateway for 10 minutes, enough time for central offices to access the serial port of the weather station and extract the data. Every time the IP-RS232 gateway is opened, the modem must send an identifying ID (so that the server to which the TCP / IP-RS232 gateway connects can know which weather station it is)
- Once the information from the weather station has been read, the modem must sleep for another 24 hours, after which the process will be repeated. If the station read process ends before the configured 10 minutes, you must be able to send a command to bring the modem into ULP (Ultra Low Power) mode immediately, without waiting to complete the 10 minutes

Config.txt configuration file:

COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	Google DNS. Must be used if TCP_IP is set to DNS

GPRS_timeout: 0	Connected to data while active
MTX_model: 199802407	MTX terminal modem model used
MTX_mode: client	TCP client mode
MTX_IDClient: ID-0001	ID sent by modem when TCP client socket opens
MTX_ATEmbedded: on	Modem allows sending embedded AT from gtw
TCP_IP: 200.1.2.3	Central offices server IP address
TCP_port: 20010	TCP port used
FIREWALL_enabled: off	Any incoming connection form any IP is allowed
ULP_enabled: on	Modem ULP activated
ULP_sleepMode: minutes	Wake up every X minutes
ULP_minutesOff: 1440	Asleep during 24 hours
ULP_secondsOn: 600	Awake during 10 minutes

Details:

- Basically the operation is this:

  1. The modem wakes up

  2. The modem opens an IP-RS232 gateway, connecting to IP 200.1.2.3 and TCP port 20010

  3. Once the socket is opened, the modem sends ID-0001 so that the server knows which modem it is.

  4. The server uses the IP-RS232 gateway to read the weather station

  5. Once the reading is finished, you can choose to let the modem turn itself off, after turning ULP_secondsOn on, or, since embedded AT commands are enabled, send the following AT command by the IP-RS232 gateway itself to be executed. by the modem:

  <MTXTUNNELR> AT ^ MTXTUNNEL = SETULPSECONDS, 5 </MTXTUNNELR>

  This will adjust the remaining time of the modem to enter ULP mode (to turn off) to 5 seconds, so the shutdown will be immediate

3.5 EXAMPLE: IP-RS232/RS485 Tunnel using an ULP consumption modem. Programmed timing activation.

Scenario details:

- A series of energy meters with RS232 port are available. It is necessary to read these counters for 4G/3G/2G once a day at 22:00, but without compromising the consumption of the system, that is, a minimum consumption is required by the modem connected to the meter, since the modem is to be powered by batteries
- The modem must remain in ultra low power mode (~ 10 uAmps) until 22:00
- At 22:00 the modem must wake up, connect to the data network (4G/3G/2G) and open a client TCP / IP-RS232 gateway for 10 minutes, enough time for central offices to access the serial port from the energy meter and extract the data. Every time the IP-RS232 gateway is opened, the modem must send an identifying ID (so that the server to which the TCP / IP-RS232 gateway connects can find out which electric meter it is)
- Once the meter data has been read, the modem must sleep until 22:00 the next day, after which the process will be repeated

Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: on	CTS hardware control enabled
COMM_autorts: on	RTS hardware control enabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_DNS: 8.8.8.8	Google DNS. Must be used if TCP_IP is set to DNS

GPRS_timeout: 0	Connected while awake
MTX_model: 199802407	MTX terminal modem model used
MTX_mode: client	TCP server mode
MTX_TPProtocol: ntp	Time synchronization protocol
MTX_TPServer: ntp.roa.es	Time server (MTX must synch time)
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_ATEmbedded: on	Modem allows sending embedded AT from gtw
MTX_IDClient: ID-0001	ID sent by modem when TCP client socket opens
TCP_IP: 200.1.2.3	Central offices server IP address
TCP_port: 20010	TCP port used
FIREWALL_enabled: off	Any incoming connection form any IP is allowed
ULP_enabled: on	Modem ULP service activated
ULP_sleepMode: date	It will wake up by date/time
ULP_secondsOn: 600	It will wake up by date/time
ULP_time1: XX2200	MTX will wake up every day (XX) at 22:00

Details:

- Basically the operation is this:

  1. The modem wakes up at 22:00

  2. The modem opens an IP-RS232 gateway, connecting to IP 200.1.2.3 and TCP port 20010

  3. Once the socket is open, the modem sends ID-0001 so that the server knows which modem it is.

  4. The server uses the IP-RS232 gateway to read the energy meter

  5. Once the reading is finished, you can choose to let the modem turn itself off, after turning ULP_secondsOn on, or, since embedded AT commands are enabled, send the following AT command by the IP-RS232 gateway itself to be executed. by the modem:

<MTXTUNNELR> AT ^ MTXTUNNEL = SETULPSECONDS, 5 </MTXTUNNELR>

This will set the modem's remaining time to enter ULP mode (to shutdown) to 5 seconds, so the shutdown will be immediate in ULP mode (to shutdown) to 10 seconds, so the shutdown will be immediate.

The modem will wake up again at 10:00 PM the next day, repeating the cycle

- The modem needs to synchronize the time, so time servers have been included in the configuration

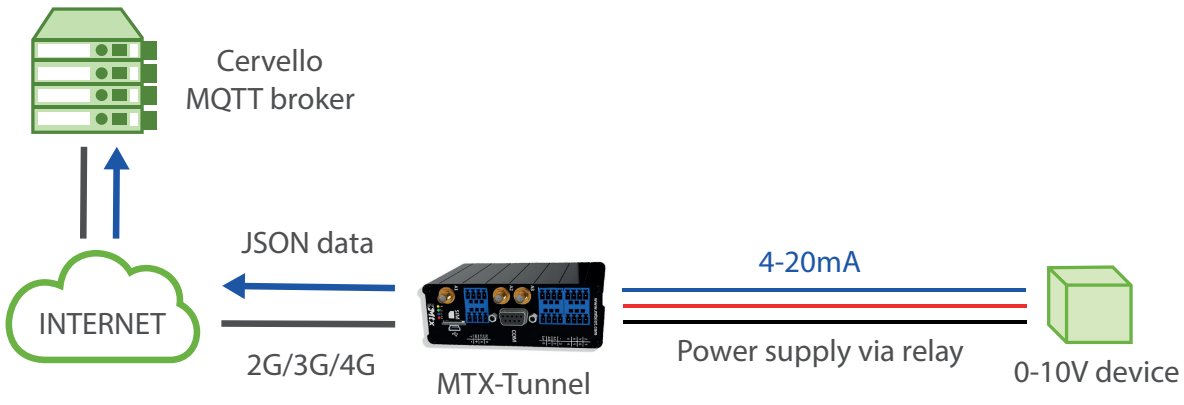
3.6 EXAMPLE: Periodic monitoring via 4G/3G/2G of a 4-20mA sensor with ultra-low consumption modem. Activating sensor supply via relay. Sending to broker MQTT.

Scenario details:

- It is intended to monitor a 4-20mA sensor with a 4G/3G/2G modem
- The modem is going to be powered by batteries, so it must remain in ultra low consumption mode (~ 10 uAmps) waking up just to make the measurement and send it via 4G/3G/2G (depending on the modem model) to an MQTT broker
- Every 24h the MTX modem must wake up, then it must activate its internal relay to power the 4-20mA sensor to be measured, wait a few seconds for the signal to stabilize and then proceed to read. After reading, an attempt will be made to send the data to an MQTT broker. If the data cannot be sent, they must be stored in flash memory (non-volatile) for sending the next time
- After the information sending process (satisfactory or not), the modem must disconnect the internal relay (to remove power to the 4-20mA sensor and save consumption) and re-enter ultra-low power mode for a period of another 24h

Solution:

MTX-IOT-S [4-N] modem + MTX-Tunnel



Config.txt configuration file:

GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Connected while awake

MTX_pin: 0000	SIM without a PIN
MTX_model: 199802407	MTX terminal modem model used
MTX_TPProtocol: ntp	Time synchronization protocol
MTX_TPServer: ntp.roa.es	Time server (MTX must synch time)
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_mode: none	No gateways used
ULP_enabled: on	Modem ULP service activated
ULP_sleepMode: minutes	It will wake up by date/time
ULP_minutesOff: 1440	It will wake up every 1440 minutes (24 hours)
ULP_secondsOn: 120	It will be awaken 2 minutes
ULP_relayMode: 1	Relay will activate when modem awake
MQTT_enabled: on	MQTT service in the modem activated
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS, including port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT1	MTX-Tunnel subscribes to this topic for commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send answers to AT commands
MQTT_qos: 1	Service quality
MQTT_keepalive: 300	Keep alive MQTT connection (300 seconds)
MQTT_persistent: off	We don’t need persistence
LOGGER_enabled: on	Logger activated
LOGGER_mode: mqtt	Sending mode MQTT

LOGGER_numRegistersFlash: 90	Upto 90 flash readings (3 months)
LOGGER_numRegistersRam: 0	No readings stored in RAM
LOGGER_mqttTopic: [IMEI]/logger	Sending topic to MQTT broker of the data
LOGGER_ioPeriod: 300	There's time for just 1 reading (the first one)
LOGGER_ioPeriodDelay: 10	10 seconds to stabilize reading 4-20mA
ADC_mode0: current	ADC0 as a current input
ADC_config0: normal	No special configuration

Details:

- The frame sent to the server has the following JSON format:

  {"IMEI":"354033091483894","TYPE":"IOS","TS":"27/05/2020 11:23:11","IO0":1,"IO1":1,"IO2":1,"IO3":1,"IO4":1,"IO5":0,"IO6":0,"IO7":0,"IO8":1,"AD0":10000,
  "ADM0":"current","AD1":0,"ADM1":"voltage"}

  Where:

  TYPE: IOS frame type

  IMEI: internal identification of the MTX

  Q: User field specified in the LOGGER_password parameter

  TS: TimeStamp of when data was collected

  IO0: It is the value of the digital input / output 0 of the modem (if available)

  IO1: Is the value of the digital input / output 1 of the modem (if available)

  IO2: It is the value of the digital input / output 2 of the modem (if available)

  IO3: Is the value of digital input / output 3 of the modem (if available)

  IO4: It is the value of the digital input / output 4 of the modem (if available)

  IO5: Is the value of the digital input / output 5 of the modem (if available)

  IO6: Is the value of the digital input / output 6 of the modem (if available)

  IO7: Is the value of the digital input / output 7 of the modem (if available)

  IO8: Is the value of the digital input / output 8 of the modem (if available)

  AD0: It is the value of the analog input 0 of the modem (sensor reading 4-20mA)

ADM0: AD0 operating mode

AD1: It is the value of the analog input 1 of the modem (sensor reading 4-20mA)

ADM1: AD1 operating mode

- Remember that to configure the analog inputs as 4-20mA (instead of 0-50V as the modem comes from the factory), you must configure microswitches 7 and 8 to ON, as indicated in the tables in “Annex A” of this manual
- Keep in mind that the ULP_minutesOff that the MTX-Tunnel will remain asleep start counting just as the modem enters ultra-low power mode
- From the MQTT platform you can end the ULP session at any time (without waiting for the end of the 2 minutes). To do this, you can send the command AT ^ MTXTUNNEL = SETULPSECONDS, 0 to the modem (to topic [IMEI] / AT1)
- Note that the modem sends a JSON where the time is included. For this reason, the configuration of 2 NTP time servers is included.
- Example operation summary is as follows:
    - The modem wakes up
    - The modem initiates internal internet connection processes and activates the relay to power the sensor
    - Once connected to the Internet, the time is synchronized
    - After the seconds configured in Logger_ioPeriodDelay, to stabilize the sensor after its activation with the relay, the modem takes a sample of its I / O, taking the sensor reading 4-20mA
    - The modem stores the reading in Flash memory and tries to send it to the MQTT server, along with other readings that it may have stored from previous days that due to some problem could not be sent
    - Each reading sent correctly is removed from the flash memory
    - After “ULP_secondsOn” seconds, or the end having been forced from the MQTT server, the modem enters low power mode 1440 minutes (1 day), but not before deactivating the relay and therefore removing the power supply to the 4-20mA sensor

4. ANNEX: API Configurations and Example Scenarios

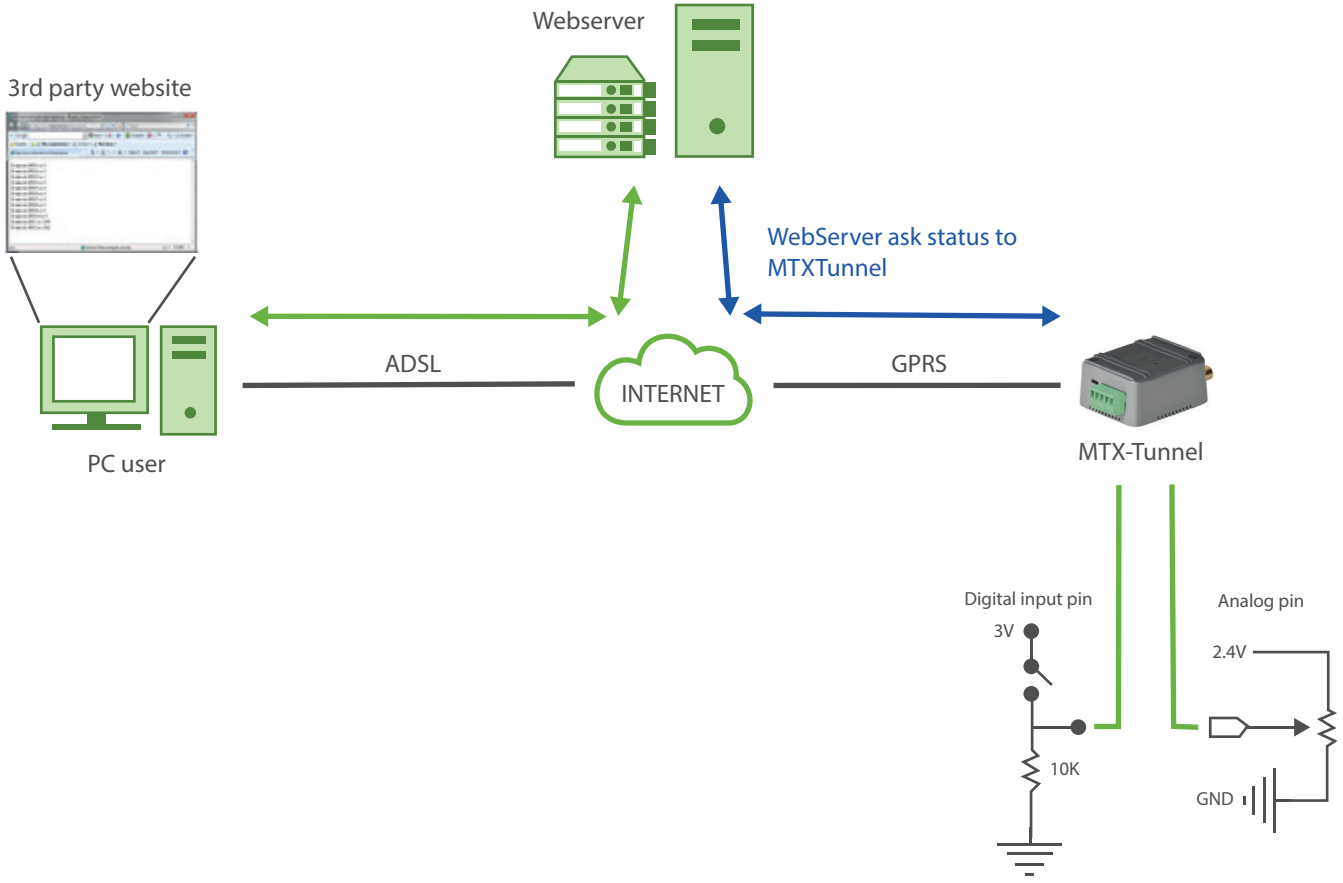
4.1 EXAMPLE: API used for reading MTX-Tunnel GPIOs and ADCs from 3rd party site.

Scenario details:

- Monitor the modem's inputs remotely and present values in a 3rd-party web page
- MTX-Tunnel has to be permanently connected to GPRS in server mode, waiting for HTTP GET connections coming from this web page
- The end customer web page will be ASP language programmed and will collect all of the values and display them

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel



Config.txt configuration file:

GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_model: 199801436	MTX-Terminal modem model used
MTX_mode: none	No gateways
MTX_ping: 30	Minutes for connectivity supervisor
MTX_pingIP: 8.8.8.8	Connectivity supervisor IP address
WEBSERVER_enabled: on	Webserver sevice enable
WEBSERVER_firewall: off	Firewall disabled
WEBSERVER_login: user	Webserver login
WEBSERVER_password: 1234	Webserver password
WEBSERVER_skin: http://www.mtxtunnel.com/webserverimg/	Webserver skin
WEBSERVER_gsmScript: http://www.blogelectronica.com/gps/gsm.php	Script GSM positioning

ASP page programming example:

The ASP programming example for this scenario is very simple.

Basically, in ASP code we use AT commands that are executed remotely and collect the responses. The command used is AT^MTXTUNNEL=GETIOS which obtain all of the digital and analog input values in one shot.

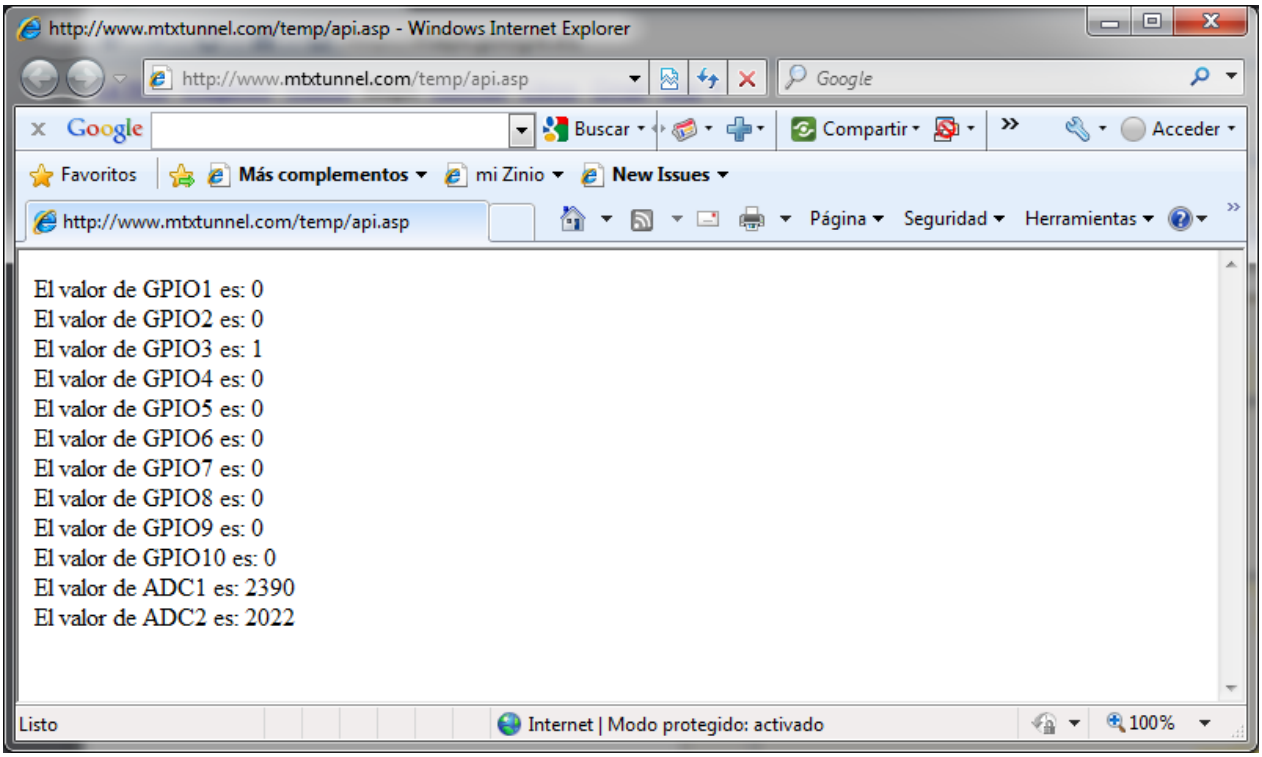
The results of these values are shown in the web explorer:

```
<%
      set XmlObj = Server.CreateObject("Microsoft.XMLHTTP")
      XmlObj.open       "GET",       "http://mtxtunnel.dyndns.org/api.
html?ATCOMMAND=AT%5EMTXTUNNEL=GETIOS&LOGIN=user&PASS=1234", false
      XmlObj.send
      datosDelMTXTunnel = XmlObj.responseText


      cadenaInicio="<MTXTUNNEL>AT^MTXTUNNEL=GETIOS "
      posiIni=instr(1,datosDelMTXTunnel,cadenaInicio)
      posiFin=instr(1,datosDelMTXTunnel,"</MTXTUNNEL>")
      datosIO=mid(datosDelMTXTunnel,len(cadenaInicio),posiFin-
len(cadenaInicio))


      vectorDatos=split(datosIO,",")
      for i=0 to 9
           Response.write("El  Value  de  GPIO"  &  i+1  &  "  es:  "  &
vectorDatos(i) & "<br>")
      next
      Response.write("El Value de ADC1 es: " & vectorDatos(10) & "<br>")
      Response.write("El Value de ADC2 es: " & vectorDatos(11) & "<br>")
%>
```

The web page will show:



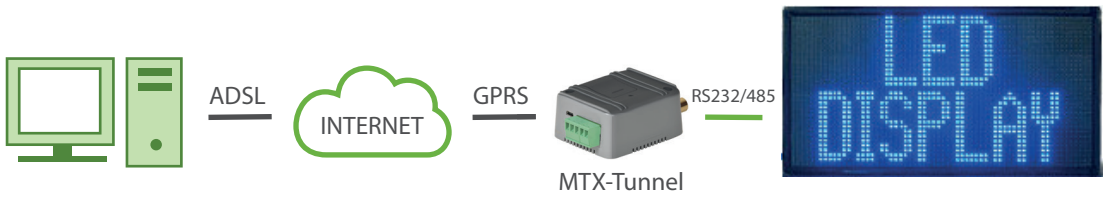
4.2 EXAMPLE: Using API to send data coming from a 3rd party webpage form and output serial port MTX-Tunnel.

Scenario details:

- We have a special display/LCD with a serial RS232 input. The data present in the RS232 port is shown in the display
- We need to remotely control the LCD using the MTX-Tunnel GPRS-Serial tunnel
- We can control the LCD in several ways:
- We could develop a code for the server PC and use the TCP/IP port socket to connect to MTX-Tunnel
- This example shows how to do this in any third-party web page using a form. The form will have a user defined entry box, so the text data captured will be resent to the LCD using the RS232 port. The LCD will answer with a code which will be displayed in the Web page form

Solution:

MTX-T [4-N]2 modem+MTX-Tunnel



Config.txt configuration file:

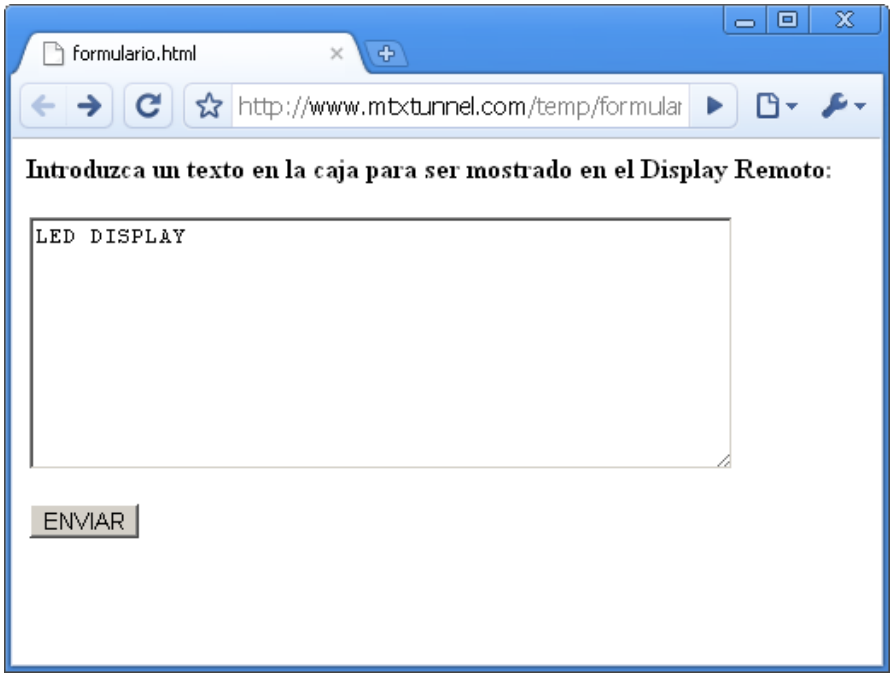
COMM_baudrate: 115200	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	CTS hardware control disabled
COMM_autorts: off	RTS hardware control disabled
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM2_baudrate: 115200	Serial port baud rate

COMM2_bitsperchar: 8	8 bit data
COMM2_autocts: off	CTS hardware control disabled
COMM2_autorts: off	RTS hardware control disabled
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_model: 199801438	MTX-Terminal modem model used
MTX_mode: none	No gateway is activated
MTX_ping: 30	Minutes for connectivity supervision ping
MTX_pingIP: 8.8.8.8	Connectivity supervision IP address
WEBSERVER_firewall: off	Firewall disabled
WEBSERVER_enabled: on	Internal webserver enabled
WEBSERVER_login:user	Webserver login
WEBSERVER_password: 1234	Webserver password
WEBSERVER_skin: http://www.mtxtunnel.com/webserverimg/	Webserver skin
WEBSERVER_gsmScript: http://www.blogelectronica.com/gps/gsm.php	Script GSM positioning

Third-party Web page form example:

This example has a very simple HTML code web page. It has a form with a text field and a button. HTML code also has a small script –JavaScript code- calling MTX-Tunnel.

This is the Web Page aspect:



HTML code for this webpage is as follows. Please ask iotsupport@mtxm2m.com for the full code.

```
<html>
<head></head>
<body>

<script type="text/javascript">
    function enviar()
    {
        var texto = document.FORMU.TEXTO.value;
        document.location.href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT%5EMTXTUNNEL=RS232,1," +
        texto + "&LOGIN=user&PASS=1234";
    }
</script>

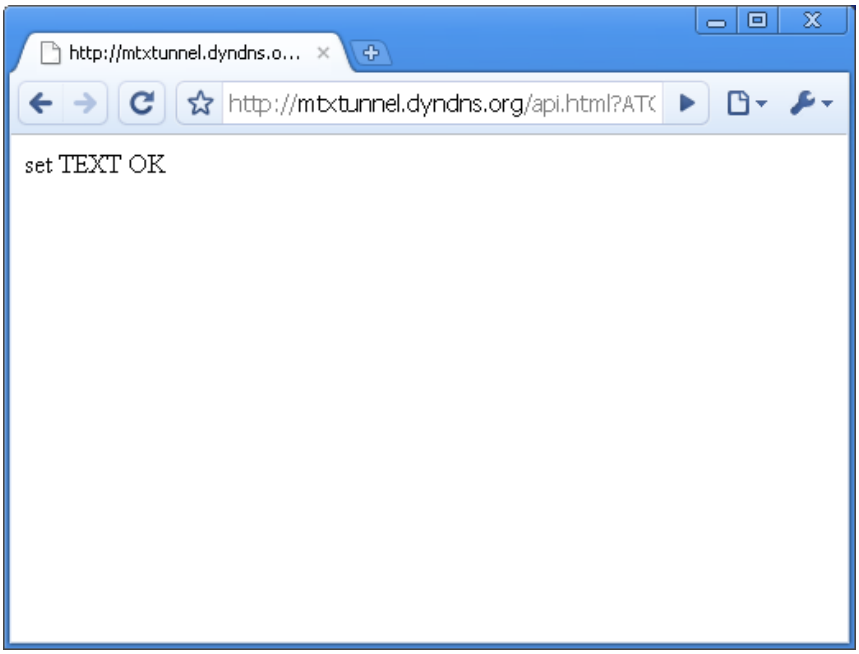
<p><b>Introduzca un texto en la caja para ser mostrado en el Display Remoto:</b></p>
<form method="GET" action="http://mtxtunnel.dyndns.org/api.html" name="FORMU">
    <p><textarea rows="9" name="TEXTO" cols="49"></textarea></p>
    <p><input type="button" value="ENVIAR" name="B1" onclick="enviar()"></p>
</form>

</body>
</html>
```

This example “LED DISPLAY” text has been written in Webpage form and it is sent to the remote MTX-Tunnel and GPRS-serial tunnel. After this the data is now in the R232 port which will be shown in the LCD display.

The LCD screen responds with “Set TEXT OK” code which is collected by MTX-Tunnel and resent as a response to the form. Obviously, the given response is very simple since we are looking at an example. In

real applications, the response would lead to a more complex page being shown to the end user.



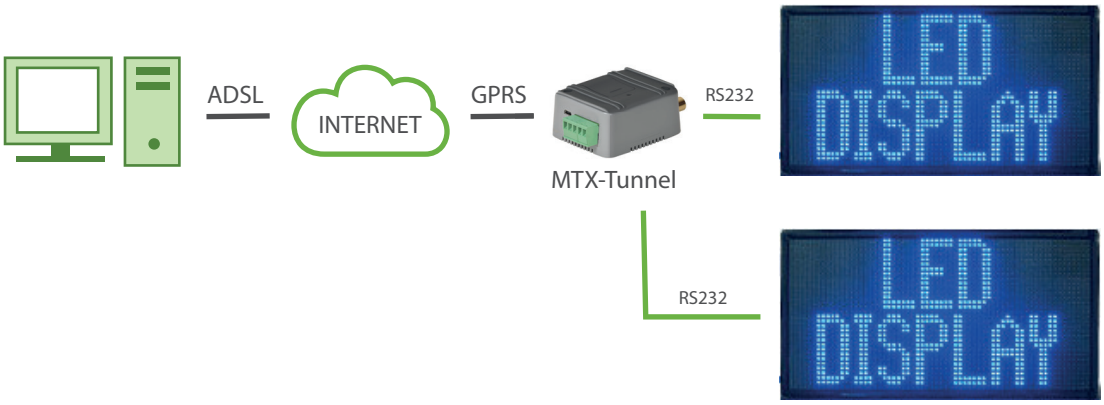
This is very simple example; in real application complex codes should be implemented.

We can control two LCD screens with just one MTXTerminal modem with MTX-Tunnel installed.

The key is the AT^MTXTUNNEL command.

In the example, we used AT^MTXTUNNEL=RS232,1But this command can send data to COM2 changing the “value” parameter (shown in red) to 2.

So we can control 2 LCD screens as follows:



Details:

- Remember if you want to configure COM2 port as RS232 you need to turn off the microswitch 2 next to the SIM card slot (if not it is configured as RS485)

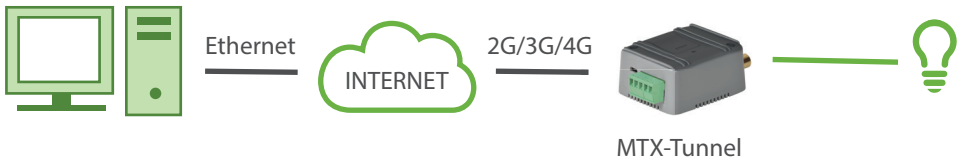
4.3 EXAMPLE: Using API to switch a relay from a third-party or end user webpage.

Scenario details:

- We want to program a web page and switch 1 relay, to switch a lamp on/off. For that we will use MTX-IoT [4-S-N-N]-STD-RL with internal relay
- An end user will open the web page and will switch the external lamps on/off with the click of a mouse. MTX-Tunnel has to communicate the webpage data with modem's GPIOs

Solution:

MTX-IoT [4-S-N-N]-STD-RL modem+MTX-Tunnel



Config.txt configuration file:

GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_model: 199801437	MTX-Terminal modem model used
MTX_ping: 30	Minutes for connectivity supervision ping
MTX_pingIP: 8.8.8.8	Connectivity supervision IP address
WEBSERVER_firewall: off	Firewall disabled
WEBSERVER_enabled: on	Internal webserver enabled

WEBSERVER_login:user	Webserver login
WEBSERVER_password: 1234	Webserver password
WEBSERVER_skin: http://www.mtxtunnel.com/webserverimg/	Webserver skin
WEBSERVER_gsmScript: http://www.blogelectronica.com/gps/gsm.php	Script GSM positioning

HTML web page example code:

We will switch the relay. We will not show the initial relay state to simplify the example.

The web page aspect is shown in the following picture. Only this simple HTML code translates the link in an AT command to be sent remotely. The AT command changes GPIO output level.



Source code

```
<html>
<head></head>
<body>

<p><b>Para conmutar un RELÉ pulse en On/Off:</b></p>
<table border="0" width="318" id="table1">
    <tr>
        <td width="110"><b>Relé1:</b></td>
        <td width="86"><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=0,0&LOGIN=user&PASS=1234">On</a></td>
        <td><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=0,1&LOGIN=user&PASS=1234">Off</a></td>
    </tr>
    <tr>
        <td width="110"><b>Relé2:</b></td>
        <td width="86"><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=1,0&LOGIN=user&PASS=1234">On</a></td>
        <td><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=1,1&LOGIN=user&PASS=1234">Off</a></td>
    </tr>
    <tr>
        <td width="110"><b>Relé3:</b></td>
        <td width="86"><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=2,0&LOGIN=user&PASS=1234">On</a></td>
        <td><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=2,1&LOGIN=user&PASS=1234">Off</a></td>
    </tr>
    <tr>
        <td width="110"><b>Relé7:</b></td>
        <td width="86"><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=6,0&LOGIN=user&PASS=1234">On</a></td>
        <td><a href="http://mtxtunnel.dyndns.org/api.html?ATCOMMAND=AT^SSIO=6,1&LOGIN=user&PASS=1234">Off</a></td>
    </tr>
</table>

</body>
</html>
```

Details:

- The address 176.80.120.20 shown in the example is the MTX SIM IP. If you don't have a SIM with a fixed IP you can check the parameters and examples DYNDNS_ on this guide
- To switch the MTX-IoT [4-S-N-N]-STD-RL relay is necessary to act on GPIO6, like in the E/S tables of the different models, that can be found on this guide

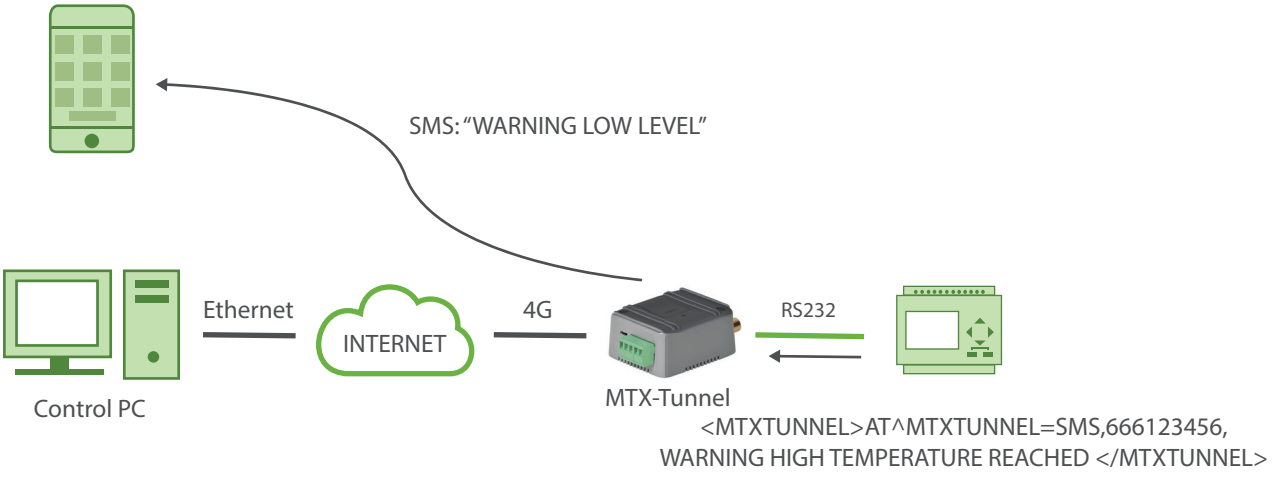
4.4 EXAMPLE: Using API to send SMS to external PLC devices connected to COM1 at the same time that the 4G-RS232 tunnel is active.

Scenario details:

- From the central offices we need to access periodically via 4G the PLC serial port to send instructions and collect results. To access it we'll use MTX-Tunnel as a 4G-RS232 gateway
- MTX-Tunnel must always have the IP session established to be able to access the PLC device at any moment. It will use the port TCP 20010
- The PLC device should be able to send an SMS alarm at any moment. PLC is connected to MTX-Tunnel via the only serial port, so the serial 4G tunnel can coexist with seding AT commands by the PLC using the same serial port (to send the SMS)

Solution:

MTX-T [4-N] modem+MTX-Tunnel



Config.txt configuration file:

COMM_baudrate: 115200	Serial port baudrate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	CTS hardware control disabled
COMM_autorts: off	RTS hardware control disabled
COMM_stopbits: 1	1 bit stop

COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_model: 199801445	MTX terminal modem model used
MTX_mode: server	TCP server mode
MTX_ATMux: on	AT multiplexing mode on serial port enabled
MTX_ping: 30	Minutes for connectivity supervision ping
MTX_pingIP: 8.8.8.8	Connectivity supervision IP address
TCP_port: 20010	TCP port for 4G-RS232
FIREWALL_enabled: off	Firewall disabled
TELNET_enabled: on	Modem's Telnet unabled
TELNET_login: user	Telnet username
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet port

Detailed explanation:

The simplest way to solve this scenario is to use an AT command multiplexer. This allows you to send AT commands at same time that the GPRS-serial tunnel is established. Enable the MTX_ATMux parameter to "on" to activate the multiplexer.

The AT commands must have a special format, between tags:

```
<MTXTUNNEL></MTXTUNNEL>
```

In the example, to send “WARNING HIGH TEMPERATURE REACHED” to the end phone number 666123456 the command will be:

```
<MTXTUNNEL>AT^MTXTUNNEL=SMS,666123456,WARNING HIGH TEMPERATURE REACHED
</MTXTUNNEL>
```

The command is interpreted and MTX TUNNEL executes it. The AT response will also be between the same tags.

5. ANNEX: Scenario and Configuration Examples for Wavenis

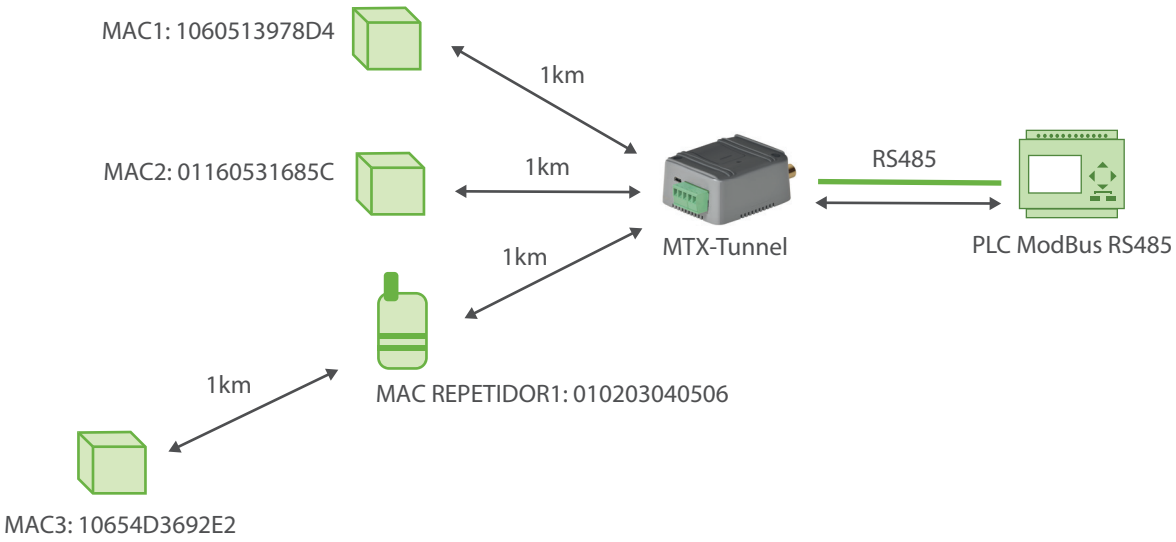
5.1 EXAMPLE: Remote reading of pulse counters with radio communications (868 MHz) from PLC through modbus RTU protocol.

Scenario details:

- We have a Modbus RTU PLC. It is intended to read 3 pulse counters with RF868MHz radio communication
- Therefore, an MTX-IoT [4-S-N-N]-STD-N-WC868 device will be used that will act as a Modbus RTU-Wavenis protocol gateway. The MTX will therefore act as a modbus slave and must have the modbus address 50. The communication between the PLC and the MTX will be at 9600.8, N, 1
- The MTX-IoT [4-S-N-N]-STD-N-WC868 must also be remotely accessible for maintenance or to read a radio counter at any time

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel



Configuration file config.txt:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: modbuswavenis	MTX-Tunnel mode
MTX_model: 199801404	MTX modem model
MTX_portAux: wavenis	Aux port communicates with RF card
MTX_ping: 35	Ping every 35 minutes without comms

MTX_pingIP: 8.8.8.8	Ping address
SMS_allPhones: on	SMS with commands to be sent from any phone
SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	TCP port for telnet
MODBUS_localAddress: 50	Local modbus address of the MTX
MODBUS_readCommand: 4	Modbus reading command
WAVENIS_mac1: 1060513978D4	MAC address of Waveflow1
WAVENIS_mac2: 01160531685C	MAC address of Waveflow2
WAVENIS_mac3: 10654D3692E2;1	MAC address of Waveflow3. Repeater 1 used
WAVENIS_rep1: 010203040506	MAC address of repeater 1

Details:

- The WAVENIS_macX parameters specify the MAC address of each waveflow device to be managed. The MTX-Tunnel allows managing up to 32 devices (from WAVENIS_mac1 to WAVENIS_mac32)
- Each WAVEFLOW device has 4 pulse counting cables. In other words, the MTX-Tunnel would be able to control up to 32 x 4 = 128 pulse counters
- The WAVENIS_repX parameters indicate the MAC addresses of the repeaters (also called WAVETALK). Up to 16 repeaters can be configured. Although WAVETALK devices are exclusively designed to work as repeaters, it is also possible to use WAVEFLOWs as repeaters (that is, in addition to acting as counters, it allows acting as a repeater simultaneously)

- If one or more repeaters must be used to read a WAVEFLOW device, it must be specified in its MAC in the following way: WAVENIS_maxX: AABBCCDDEE; 1; 2; 4 This would make 3 repeaters to be used to read the WAVEFLOW from mac AABBCDDEE, 1, 2 and 4 (to be specified in parameters WAVENIS_rep1, WAVENIS_rep2 and WAVENIS_rep4
- It is possible to specify up to a maximum of 3 repeaters between the MTX and a Waveflow. The maximum distance (direct vision) that 1 repeater covers is 1 KM in its 25mW version. A 500mW repeater covers up to 4Km. It is preferable to use 25mW repeaters, as they use FHSS (frequency hopping), which makes them more immune to radio interference, while the 500mW repeaters are single-channel
- When the MTX-Tunnel is used as a Wavenis modbus gateway (MTX_mode: modbuswavenis), the MTX becomes a SLAVE modbus device, whose address is the one specified in the MODBUS_localAddress parameter
- If you want to use the RS232 port (DB9 of the MTX) to communicate the PLC with the MTX, you must configure the microswitches that you will find next to the SIM port with the following values: SW1: ON, SW2: ON, SW5: OFF
- If, on the other hand, you want to use the RS485 port (green MTX terminal) to communicate the PLC with the MTX, you must configure the microswitches that you will find next to the SIM port with the following values: SW1: OFF, SW2, OFF, SW5: ON
- When the MTX-Tunnel is in modbuswavenis mode (MTX_mode: modbuswavenis) the MTX modbus registers memory table is as follows:

DIRECTION	R/W	DESCRIPTION
0	R/W	Action: 1>Read; 2>Reset counters ;3>Reset Alarms
1	R/W	ID Device 0=_MAC1, 1=_MAC2, 2=_MAC3, ...
2	R	Status: 0=OK ; 1=Executing command; 2= ERROR
3	R	COUNT 1 WORD_H
4	R	COUNT 1 WORD_L
5	R	COUNT 2 WORD_H
6	R	COUNT 2 WORD_L
7	R	COUNT 3 WORD_H
8	R	COUNT 3 WORD_L
9	R	COUNT 4 WORD_H
10	R	COUNT 4 WORD_L

11	R	BATTERY ALARM
12	R	ALARM CUT 1
13	R	ALARM CUT 2
14	R	ALARM CUT 3
15	R	ALARM CUT 4

- To write to a register, the use of the write modbus command 0x06 is mandatory. To read registers it is possible to use the MODBUS command 0x03 or 0x04 (whichever is specified in the MODBUS_readCommand parameter)
- How to read a Waveflow? The steps to carry out this process are detailed below:

  STEP 1.- We write in register @ 1 a value to specify which WAVEFLOW it is intended to read: if we want to read WAVEFLOW with the MAC address specified in WAVENIS_mac1 we will write the value “0” in register @ 1, if we want to read WAVEFLOW with the MAC address specified in WAVENIS_mac2 we will write a “1” in register @ 1, etc.

  STEP 2.- We write a value to register @ 0 to specify the action we intend to perform: to read a WAVEFLOW, write the value “1” in register @ 0

  STEP 3.- We read registers @ 2 to @ 15 until the value of register @ 2 is “0” or “2”: if the action went well, that is, if the reading could be carried out satisfactorily, in this Register @ 2 will return a value of “0”. If it returns a value of “2”, it indicates that the reading was wrong. In the case of being a correct reading, between the addresses @ 3 and @ 10, the values of the 4 counters of a WAVEFLOW will be available, each separated by the WORD with the highest weight (H) and the one with the lowest weight (L) . If desired, registers @ 11 to @ 15 can also be consulted, where battery alarms and cable cut-offs are returned
- If you only want to use the MTX as a Modbus-Wavenis gateway and do not need 4G/3G/2G communications (and therefore do not need a SIM), you can use the MTX without a SIM. To do this, specify the parameters: MTX_configMode: reverse and TELNET_enabled: off

6. ANNEX: Examples and Scenarios for Automatic Readings of Modbus Devices

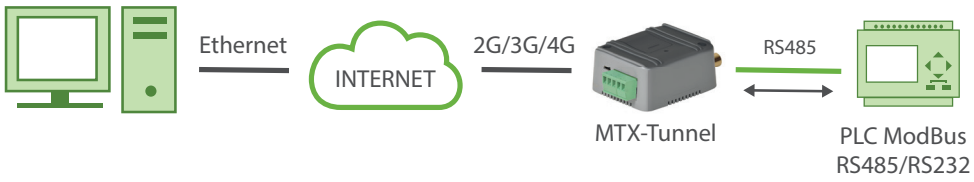
6.1 EXAMPLE: PLC Modbus RTU periodically registers readings, sends them as a JSON object to webserver.

Scenario details:

- There is a Modbus RTU PLC. This PLC records some variables such as temperature, pulses, etc. in their register memory table from external devices connected to it. We need to read them and send them to a WebServer
- MTXTunnel will connect every 15 minutes to the PLC serial port. The number for storing temperature is number 20. Pulse counters are stored in registers table number 21, 22 and 23 respectively
- MTXTunnel will get all data values and convert them to JSON objects in order to send them to a WebServer using HTTP GET procedure. In case the 4G/3G/2G network or server fails, it needs to store up to 1500 readings in MTX-Tunnel's non volatile internal memory and then it can retry communicating later on
- You also need to be able to access the PLC at any time in order to read the register values in real time or write/change any of PLC's configuration parameters

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel



Config.txt configuration file:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	MTX-Tunnel mode
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_rssiLevel: 10	Coverage LED activated
SMS_allPhones: on	All phone numbers are authorized

SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_firewall: off	Telnet port 20023
LOGGER_enabled: on	We enable the MTX Logger, to store the records
LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: www.miservidorWeb.com/ json.asp?data=	Server URL, will receive JSON data
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_httpMode: getjson	HTTP GET (JSON) mode
MODBUS_address: 1	ModBus equipment address
MODBUS_start: 20	ModBus register address to be read
MODBUS_numwords: 4	Number of registers read
MODBUS_readCommand: 3	Reading command
MODBUS_period: 900	Timing -seconds- read is repeated

Details:

- MTX-IoT [4-S-N-N]-STD-N modem is used if the PLC port is the RS485 type, but MTX-IoT [4-S-N-N]-STD-N-RS232 terminals are more suitable if PLC has the RS232 port

- Procedure:

  Every 15 minutes the modem reads a series of PLC ModBus records. They are translated to JSON objects and sent to a WebServer. WebServer URL is configured in LOGGER_server parameter.

  In the event of there being no coverage in GPRS link or the server is down, MTX-Tunnel stores data in the internal FLASH memory.

  Using TELNET, it is possible to remotely access the system and read or write any PLC records, readings or configuration parameters. Use the AT^MTXTunnel=getmodbus and AT^MTXTUNNEL=setmodbus commands.

  The JSON object data sent to the server has following syntax:

  {“IMEI”:353234028103206,”P”:”ID00001”,”TYPE”:”MODB”,”A”:1,”TS”:”20/08/12 08:31:44”,”V1”:23,”V2”:275,”V3”:274,”V4”:32765}

  WebServer receives a JSON object with the following information:

  IMEI: Modem’s IMEI identifier

  P: Password user definable field

  TYPE: data type

  A: ModBus address

  TS: Timestamp

  Vx are variables read

NOTE: You can also develop an easy modbusTCP/modbusRTU gateway as explained in example number 2.15

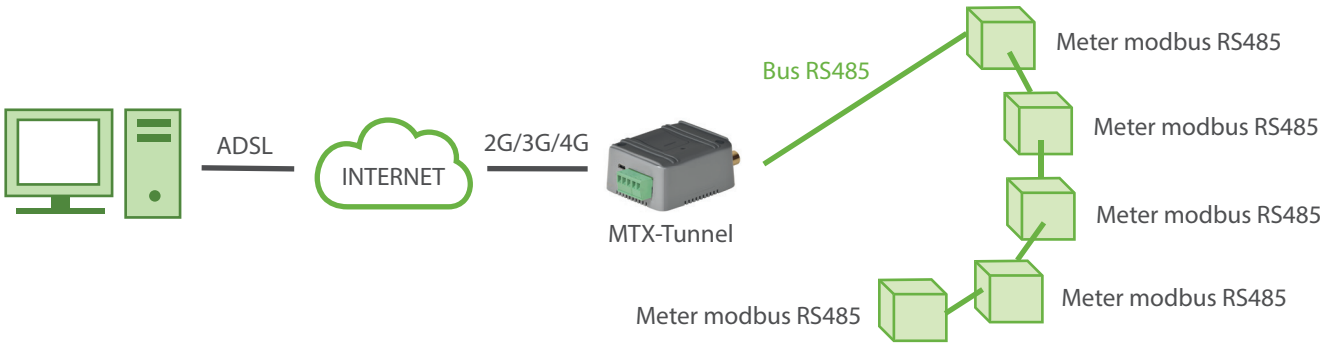
6.2 EXAMPLE: Periodically obtaining readings from 5 electronic meters and sending them to a webserver using the ModBus RTU protocol.

Scenario details:

- We have 5 electric meters with ModBus RTU protocol. These meters have a series of readings in their internal memory (e.g. cumulative consumption and average consumption) which should be read and sent to a Web server periodically
- Therefore every 15 minutes the MTX-Tunnel has to question the 5 meters to read the records via a serial port. The records to be read are for the cumulative consumption (record nº20) and for average consumption (record nº 21)
- After every reading MTX tunnel has to send the records to a web server via HTTP GET using a JSON object. However it must be able to store up to 1500 readings in flash memory in case there is a communication failure with the 4G/3G/2G so that it can send them when the connection is restored. For each meter, it will send a JSON frame where it indicates the Modbus address/ identifier of the meter that has been read
- You need to be able to access MTX-Tunnel at any moment to be able to read PLC records in real time as well as being able to write them and modify the PLC configuration records

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmare



Config.txt configuration file:

COMM2_baudrate: 9600	Serial port (where internal GPS is connected) rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autorts: off	No flow control

COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	MTX-Tunnel mode
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login

TELNET_password: 1234	Telnet password
TELNET_firewall: off	Telnet port 20023
LOGGER_enabled: on	We enable the MTX Logger, to store the records
LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: www.miservidorWeb.com/ json.asp?data=	Server URL, will receive JSON data
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_httpMode: getjson	HTTP GET (JSON) mode
MODBUS_address: 1;2;3;4;5	ModBus equipment address
MODBUS_start: 20	ModBus register address to be read
MODBUS_numwords: 4	Number of registers read
MODBUS_readCommand: 3	Reading command
MODBUS_period: 900	Timing -seconds- read is repeated

Details:

- The summary of this example is as follows: The modem will periodically read, every 15 minutes, two ModBus records from each counter and they will be sent to a web server via a JSON object (to the URL specified in the LOGGER_server parameter). In the event that the record cannot be sent (if there is no GPRS coverage or the server fails), it will store the data to memory in order to send it later on. With Telnet you can connect to the equipment directly in order to check/ change the PLC records in real time (for this, search for the AT^MTXTunnel=getmodbus and AT^MTXTUNNEL=setmodbus commands in this manual)
- A JSON object sent to a specified URL in LOGGER_server is encoded in the following manner for example:

  {“IMEI”:353234028103206,”P”:”ID00001”,”TYPE”:”MODB”,”A”:1,”TS”:”20/04/13 08:31:44”,”V1”:23,”V2”:275}

  This means that the web Server receives a JSON object with the modem’s IMEI (IMEI), the password field (P) which can be used to identify the computer if you do not want to use the IMEI, the counter’s modbus address (A) which distinguishes the counter that relates to the

readings, the time stamp (TS) which indicates when the modbus data has been read by each counter and “V1”/”V2” which indicates the readings that have been taken

- At the end of the manual you will find how to configure the internal microswitch to activate the RS485 bus of this modem model

6.3 EXAMPLE: Reading and periodic sending of records of 5 Electric meters with ModBus RTU protocol and memory map for different records to a webserver.

Scenario details:

- We have 5 electric meters with modbus RTU protocol. These meters have, in their internal memory, a series of variables/records (for example, accumulated and average consumption) that must be periodically read and sent to a web server
- To do so, the MTX-Tunnel must periodically question, every 15 minutes, via a serial port, the 5 meters in order to read the records. The records to read are different for each meter

  Meter 1: records 20 to 21 and records 100 to 105

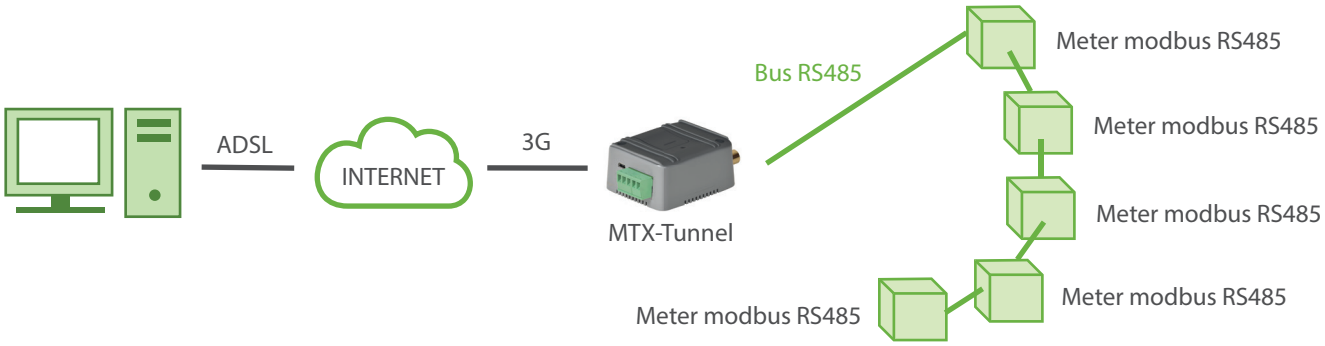
  Meter 2 and 3: records 30 to 35

  Meter 4 and 5: records 40 to 60

  Meters 1, 2 and 3 will use the modbus read command 0x03 (the most usual), meters 4 and 5 will use the modbus read command 0x04
- After each reading the MTX-Tunnel must send the value of the records to a web server via HTTP GET using a JSON object, but must be able, in case of 3G communication failure, to store up to 1500 records using flash memory that will be sent when communications are restored. For each meter it will send a JSON frame, where an identifier will specify (the modbus address) which meter has been read

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



EXAMPLE of settings (file config.txt) for such scenario:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	MTX-Tunnel mode
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM

SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet port 20023
LOGGER_enabled: on	We enable the MTX Logger, to store the records
LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: www.miservidorWeb.com/json.asp?data=	Server URL, will receive JSON data
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_httpMode: jsonget	HTTP GET (JSON) mode
MODBUS_address: 1;1;2;3;4;5	ModBus equipment address
MODBUS_start: 20;100;30;30;40;40	ModBus register address to be read
MODBUS_numwords: 2;6;6;6;21;21	Number of registers read
MODBUS_period: 900	Timing -seconds- read is repeated
MODBUS_readCommand: 3;3;3;3;4;4	Reading command

Details:

- The summary of this example is: the modem reads the Modbus records of every meter periodically, every 15 minutes, and sends them through a JSON object to a web server (to the URL specified

in LOGGER_server). In cases where it is not possible to send the records (because there is no GPRS coverage at that moment or because the server is down) it stores the data in the memory to send them later. It is possible to connect to the device directly via Telnet and check/ change the PLC records in real time (to do so please check the AT^MTXTunnel=getmodbus y AT^MTXTUNNEL=setmodbus commands set out in this manual)

- From version MTX-Tunnel v7.18 it is possible to read a different memory map for each device, as follows:

  MODBUS_address: 1;1;2;3;4;5

  Here the address of each meter is specified separately, using ';' (semicolon). Notice that meter 1 is duplicated. The reason is that it is treated as if it were 2 different devices (two readings must be made) because 2 ranges of different records must be read, from 20-21 and 100-105.

  MODBUS_start: 20;100;30;30;40;40

  The first record to be read is indicated for each meter. Notice that, as in the previous case, meter 1 is treated as two different meters, and the initial record of the initial registration of each range is as indicated (20 and 100)

  MODBUS_numwords: 2;6;6;6;21;21

  The number of records to be read for each meter is specified. It shows, for meter 1, the number of records to be read (2 records) for the first range, and for the second (6 records).

- The JSON object sent to the URL specified in LOGGER_server is codified as in this example:

  {"IMEI":353234028103206,"P":"ID00001","TYPE":"MODB","A":1,"TS":"20/04/13 08:31:44","ST":20, "V1":23,"V2":275}

  That is, the web server receives a JSON object with the modem's IMEI, a password field (P) that can also be used to identify the device (if you don't want to use the IMEI), the modbus address of the meter (A) which will be used to identify which meter the reading belongs to, the time stamp (TS) of when the modbus data of each meter has been read, and V1,V2 with the records read.

  Notice that from version MTX-Tunnel 7.18, the field ST (start), where the number of the first record to be read is specified, appears in the JSON object. Thanks to this, you will be able to tell which records they are when making 2 readings of the same device (for example if there are different ranges of records to read).

  You can quickly know the number of the record that has been read. If you want to know which record belongs to where, use Vx, NumReg= ST+x-1. For example V2, the case of the previous JSON: numReg=20+2-1 = 21

- At the end of the manual you will find how to configure the internal microswitch to activate the RS485 bus of this modem model

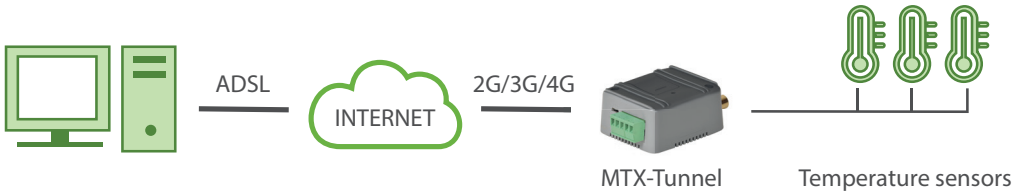
6.4 EXAMPLE: Regular reading of 3 Modbus temperature sensors and then sending to a webserver(Model MTX-Temp-RS485-IP65 of MTX accesories).

Scenario details:

- We have 3 temperature sensors MTX-Temp-RS485-IP65, which can be found in the MTX accessories catalogue of Matrix Electronica
- The MTX-Tunnel must periodically poll, every 10 minutes, the three temperature sensors, which use ModBus protocol. The modbus temperature record is 1. The ModBus addresses for the sensors have been previously set as 1, 2 and 3
- The MTX-Tunnel must send, after every reading, the value of the records to a web server via HTTP GET using a JSON object, but must be able, in case of 4G/3G/2G communication failure, to store up to 100 records in its flash memory, which will be sent when communications are restored. For each meter it will send a trauma JSON, where the meter that is read is specified by an identifier (the modbus address)

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



EXAMPLE of settings (file config.txt) for such scenario:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator

GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	MTX-Tunnel mode
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Port 20023 for Telnet
LOGGER_enabled: on	We enable the MTX Logger, to store the records

LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: www.miservidorWeb.com/json.asp?data=	Server URL, will receive JSON data
LOGGER_registerSize: 120	Register size
LOGGER_numRegistersFlash: 100	Maximum number of records in MTX
LOGGER_httpMode: getjson	HTTP GET (JSON) mode
MODBUS_address: 1;2;3	ModBus addresses of sensors to be read
MODBUS_start: 1;1;1	Initial addresses of each meter
MODBUS_numwords: 1;1;1	Number of records read from each meter
MODBUS_period: 600	Frequency of readings in seconds
MODBUS_readCommand: 4;4;4	Sensors read with modbus command0x04
MODBUS_regType: 2;2;2	2 byte registries

Details:

- The JSON object format that the MTX-Tunnel will send to the web server, will be similar to the following string, for sensor 1 (23.0º):

  {“IMEI”:353234028103206,”P”:”ID00001”,”A”:1,”TS”:”20/04/13 08:31:44”,”ST”:1, “V1”:230} ;

  for sensor 2 (24.5º):

  {“IMEI”:353234028103206,”P”:”ID00001”,”A”:2,”TS”:”20/04/13 08:31:44”,”ST”:1, “V1”:245};

  and for sensor 3 (22.1º):

  {“IMEI”:353234028103206,”P”:”ID00001”,”A”:2,”TS”:”20/04/13 08:31:44”,”ST”:1, “V1”:221}

- At the end of the manual you will find how to configure the internal microswitch to activate the RS485 bus of this modem model

6.5 EXAMPLE: Regular reading and delivery of Modbus records of a PLC + IP gateway for remote update of the PLC firmware to a webserver.

Scenario details:

- We have a PLC Modbus RTU. This PLC has a series of variables/records (for example, a temperature value and 3 meters, etc.) in its internal memory, which must be periodically read and sent to a Web server
- In order to read said records, the MTX-Tunnel must periodically poll the PLC, every 15 minutes, via an RS485serial port. The records to be read are: for temperature, record nº20, and for meters the records in 21, 22 y 23 respectively
- The MTX-Tunnel must send the records value to a web server via HTTP GET using a JSON object after every reading, but must be able, in case of a 4G/3G/2G communications failure, to store up to 1500 records in its flash memory that will be sent once communications are restored
- The PLC has a serial port RS232 that is used to update the firmware. Access to the PLC programming port must be possible at any given time. For this, the RS232 port of the MTX-IoT [4-S-N-N]-STD-N modem will be used

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



EXAMPLE of settings (file config.txt) for such scenario:

COMM_baudrate: 9600	Data rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: on	There is flow control
COMM_autocts: on	There is flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: server	Gateway activated for the PLC upgrade
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
TCP_port: 20010	TCP port for upgrade firmware gateway
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM

SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_firewall: off	Any IP will be able to connect to the MTX
LOGGER_enabled: on	We enable the MTX Logger, to store the records
LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: www.miservidorWeb.com/json.asp?data=	Server URL, will receive JSON data
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_httpMode: getjson	Sending data mode HTTP GET (JSON)
MODBUS_address: 1	Modbus address of device to be read
MODBUS_start: 20	Address of first modbus record to be read
MODBUS_numwords: 4	Number of records read from each meter
MODBUS_period: 900	Frequency of readings in seconds

Details:

- The summary of this example is as follows. The modem periodically reads, every 15 minutes, a series of PLC ModBus records and sends them to a web server (to the URL specified in the LOGGER_server parameter) via a JSON object. In cases where a record cannot be sent (because at that moment there is no GPRS coverage or because the server is down) it stores the records in the memory to send them later. Using Telnet it is possible to directly connect to the device and check/change the PLC records in real time (to do so please read the AT^MTXTunnel=getmodbus and AT^MTXTUNNEL=setmodbus commands laid out in this manual)

- The JSON object sent to the URL specified en LOGGER_server is codified as follows:

  {“IMEI”:353234028103206,”P”:”ID00001”,”A”:1,”TS”:”20/08/12 08:31:44”,”ST”:20,”V1”:23,”V2”:275,”V3”:274,”V4”:32765}

  That is, the web server receives a JSON object with the modem’s IMEI (IMEI), a password field (P) that can also be used to identify the device (if you do not want to use the IMEI), the modbus address (A), the time stamp (TS) of when the modbus data has been read, the initial address of the read variables (ST) and V1,V2, ... with each of the read variables.

- For the 4G/3G/2G-Series gateway of the PLC firmware updater we have chosen to work in Server mode (MTX_mode: server) so it is very simple to remotely connect to the MTX-Tunnel via the usual PLC firmware updater, simply stating the IP address and MTX-Tunnel port (if the IP is not fixed, it can be obtained through a missed call, an SMS or even using DynDNS among other options). If your PLC firmware updater does not allow you to enter an IP address, but only a COM port, you can use a virtual COM port such as the one shown in Annex 9

- At the end of the manual you will find how to configure the internal microswitch to activate the RS485 bus (by default) of this modem model

6.6 EXAMPLE: ADVANCED example for modbus devices reading and broadcasting to a web platform via a JSON object.

Scenario details:

- We have 3 modbus RTU devices. It is recommended that the MTX-Tunnel polls them periodically sending the records to a web platform via a JSON object. The web platform requires login and password authentication
- The MTX-Tunnel must periodically poll, every 2 minutes and via its serial port RS485, the 3 ModBus devices. The records to read are different for each device

  Device 1: records 20 to 21 and records 100 to 105

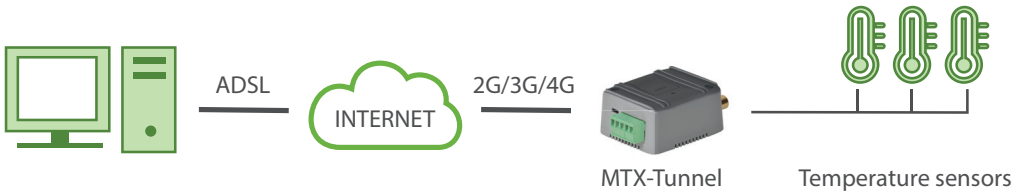
  Device 2: records 30 to 35

  Device 3: records 40 to 60

  Devices 1 and 2 will use the modbus read command 0x03 (the most common). Device 3 will use the modbus read command 0x04.
- Records 20 to 21 of Device 1 must be read and sent to the web platform every 2 minutes. Records 100 to 105 will also be read every 2 minutes but we only want to send them to the web platform every 10 minutes
- Records 30 to 35 of Device 2 will be read every 2 minutes, but they will only be sent to the web platform when the value of one of the records differs by “10” or more from the last record sent
- Records 40 to 60 of Device 3 will be read every 2 minutes. They will be sent to the web platform every 60 minutes but a reading will also be sent if any record differs by “25” or more from the last record sent

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



EXAMPLE of settings (file config.txt) for such scenario:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	MTX-Tunnel mode
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM

SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet port is 20023
LOGGER_enabled: on	We enable the MTX Logger, to store the records
LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: www.miservidorWeb.com/json.asp?data=	Server URL, will receive JSON data
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_serverLogin: miLogin	Web platform login
LOGGER_serverPassword: 12345678	The password of the web platform
LOGGER_httpMode: getjson	HTTP GET (JSON) mode
MODBUS_address: 1;1;2;3	ModBus addresses of devices to be read
MODBUS_start: 20;100;30;40	Initial addresses of each device
MODBUS_numwords: 2;6;6;21	Number of records read from each device
MODBUS_period: 120	Frequency of readings in seconds
MODBUS_readCommand: 3;3;3;4	First 2 devices use 0x03 and the other 2 0x04

MODBUS_logFrequency: 1;5;1;30	Device 1 sends variables every 1 and 5 records Device 3 every 30 records
MODBUS_changeDiff: 0;0;10;25	Device 2 controls if record variation is >=10 Device 2 controls if record variation is >=25
MODBUS_logType: 0;0;1;2	Device 1 sends with regards to time Device 2 sends change in value Device 3 sends time and/or change in value

Details:

- At the end of the manual you will find how to configure the internal microswitch to activate the RS485 bus (by default) of this modem model

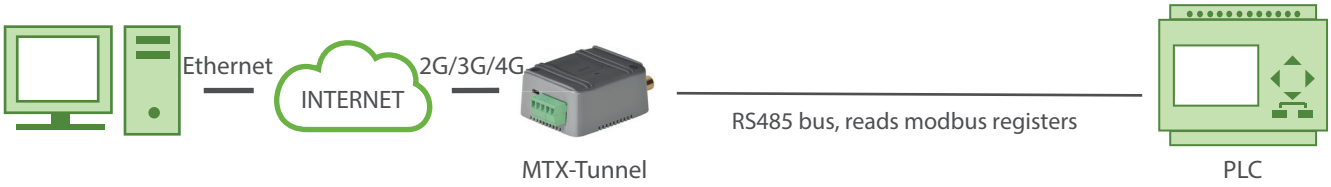
6.7 EXAMPLE: Reading modbus Word and Binary registries, sending data to webplaform using JSON object.

Scenario details:

- We want to read 12 modbus registers from a PLC
- 10 registers are binary and and the other 2 are word. It is necessary to use the modbus command 0x01 to read binary registers and the 0x03 modbus command to read word registers. Binary registers are in the addresses 10,11,12, ... 17 and Word registers are in the addresses 100,101
- We want to inform the web platorm about about the register type. For this reason we will use the parameter MODBUS_custom. Data will be read every 60 seconds
- The web platform is protected with a Login and Password, so the Logger needs to be configured in security mode (LOGGER_serverLogin, Logger_serverPassword)

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



EXAMPLE of settings (file config.txt) for such scenario:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator

GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	Not using gateway
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Send commands to MTX by SMS
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_firewall: off	To connect to MTX from any IP
LOGGER_enabled: on	We enable the MTX Logger, to store the records

LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: www.miservidorWeb.com/json.asp?data=	Server URL, will receive JSON data
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_serverLogin: miUsuario	Web platform login
LOGGER_serverPassword: miPassword	The password of the web platform
LOGGER_httpMode: getjson	HTTP GET (JSON) mode
MODBUS_address: 1;1	ModBus addresses of sensors to be read
MODBUS_start: 10;100	Initial addresses of each meter
MODBUS_numwords: 10;2	Number of records read from each meter
MODBUS_period: 60	Frequency of readings in seconds
MODBUS_readCommand: 1;3	Read commands 0x01 and 0x03
MODBUS_custom: BIN;WORD	User custom strings

Details:

- Examples of JSON received by the web platform:

  To read Words:

  {“IMEI”:353234028104337,”TS”:”02/03/14 10:15:51”,”TYPE”:”MODB”,”C”:”WORD”,”A”:2,”ST”:100,”V1”:77,”V2”:88 }

  To read Binary:

  {“IMEI”:353234028104337,”TS”:”02/03/14 16:32:00”,”TYPE”:”MODB”,”C”:”BIN”,”A”:1,”ST”:10,”NB”:10,”V1”:85,”V2”:3 }

- Note that in the case of reading binary registries these are returned grouped into byte-type registries (2 bytes, 8 bits, 2 bits). That is, the value of the binary register of address 10 would be in bit0 of V1, the value of the address registry 17 in bit 7 of V1, the address register 18 would be in bit 0 of V2 and the address register value 19 would be in bit1 of V2

In the previous case, the Custom variable is used to indicate that the records are of type Binary. Note that in the case of reading binaries also appears the variable NB, which indicates the number of bits read

- Remember that the modem model used in this example has 2 serial ports. The main port (RS232) and the secondary port (RS232 or RS485) used in this example. To configure the secondary port as RS485 you must set the microswitch 2 “ON” as shown in Annex A of this manual

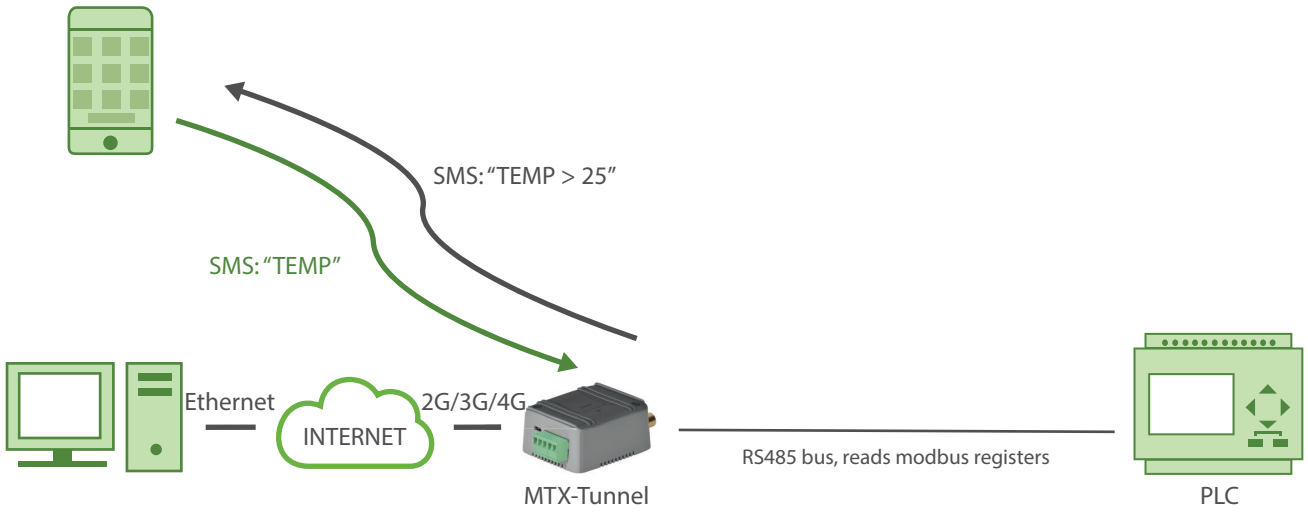
6.8 EXAMPLE: Reading and writing Modbus registers via SMS.

Scenario details:

- We have a PLC with Modbus protocol and address 1
- It is necessary to read different Modbus registers via SMS. Register 10 (current temperature), register 12 (pressure) and register 14 (current humidity)
- It is also necessary to be able to write “1” via SMS in Modus16 register. This way the PLC will activate a relay during a pre-configured time period
- SMS should be send only from the authorized mobile phones: 666123456 and 666123457
- SMS commands sent should be simple with the following texts: TEMP, PRES, HUM (to read registers 10,12 and 14 respectively). An SMS with the text RELE will write “1” in register 16 to activate the relay
- It also should be possible to carry out configuration changes remotely via TELNET

Solution:

Modem MTX-IoT [4-S-N-N]-STD-N+firmware MTX-Tunnel



EXAMPLE of settings (file config.txt) for such scenario:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN of SIM card
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	Working mode set as none as we do not use them
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_radioBand: europe	If modem is installed in Europe
SMS_allPhones: off	SMS with commands sent from authorized phone
SMS_validPhone1: +34666123456	Authorized mobile phone 1
SMS_validPhone2: +34666123457	Authorized mobile phone 2
SMS_sendIP: on	Modem replies with its IP to a missed call or SMS

SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
SMS_alias1: TEMP>AT^MTXTUNNEL=GETMODBUS,1;10;1;3	Alias for temperature reading
SMS_alias2: PRES>AT^MTXTUNNEL=GETMODBUS,1;12;1;3	Alias for pressure reading
SMS_alias3: HUM>AT^MTXTUNNEL=GETMODBUS,1;14;1;3	Alias for humidity reading
SMS_alias4: RELE>AT^MTXTUNNEL=SETMODBUS,1;16;1	Alias for relay activation
SMS_aliasResponse: result	Modem returns the result only, no AT command
SMS_aliasError: Error	Modem returns text “Error” when executing alias
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet port
TELNET_firewall: off	Any IP will be able to connect to the MTX

Details:

- When the value “result” is set in the parameter SMS_aliasResponse, the response SMS to ALIAS will not contain executed AT command

  For example, if SMS_aliasResponse parameter has the value “full”, the following will occur:

  SMS sent: TEMP

  SMS received: AT^MTXTUNNEL=GETMODBUS,1;10;1;3

  25

  OK

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

If SMS_aliasResponse parameter has the value “result”, the following will occur:

SMS sent: TEMP

SMS received: TEMP > 25

- If SMS_aliasError parameter configuration contains the text “Error”, the response SMS to ALIAS, in case of error, will have the following response:

  SMS sent: TEMP

  SMS received: TEMP > Error

- Remember that the modem model used in this example disposes of 2 serial ports. In the given example the main port (RS232) and the secondary port (RS232 or RS485) are used. To configure the secondary port as RS485 you should set “ON” in microswitch 2, as it is explained in Annex A of the this manual

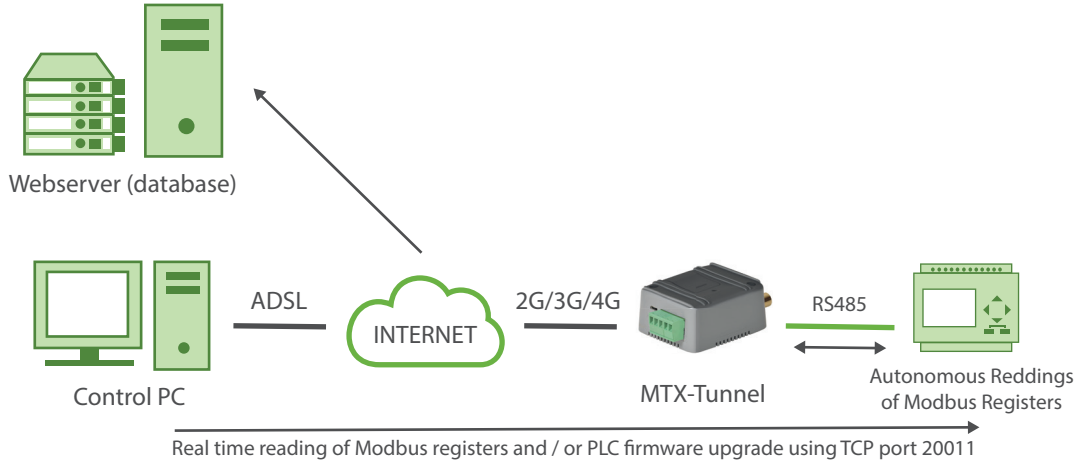
6.9 EXAMPLE: Reading and sending Modbus registers from a PLC to a webserver. Remote and occasional access to PLC registers in real time and for remote update of PLC's firmware.

Scenario details:

- We have a PLC Modbus RTU. In its internal memory this PLC contains a number of variables/ registers (for example, one temperature and three counters,...) which should be read and sent periodically to Web server
- Therefore, MTX-Tunnel should request periodically (every 15 minutes) the PLC through a serial port in order to read the above mentioned registers. The registers to be read are temperature with register number 20, and the counters' registers 21, 22 and 23 respectively
- After each reading MTX-Tunnel will send the register's value to a Web server via HTTP GET using a JSON object, but it should be able, in case of 4G/3G/2G communications failure, to store in flash memory up to 1500 readings that will be sent off when the communications are restored
- At any moment it should be possible to establish a IP-RS485 gateway to directly access Modbus registers of PLC. Additionally this gateway will be used for updating the firmware of PLC when needed

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+firmware MTX-Tunnel



EXAMPLE of settings (file config.txt) for such scenario:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits

COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: server	Gateway created
MTX_model: 199801436	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_rssiLevel: 10	Coverage LED activated
MTX_serverTimeout: 300	Socket closes if there’s no traffic in 300 secs.
TCP_port: 0	Disabled gateway
TCP_port2: 20011	Secondary serial port gateway active in 20011

SMS_allPhones: on	SMS with commands sent from any phone
SMS_sendIP: on	Modem replies with IP to a missed call or SMS
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet port
LOGGER_enabled: on	We enable MTX logger to store the records
LOGGER_password: ID00001	Field to identify the origin of the frames
LOGGER_server: www.miservidorWeb.com/json.asp?data=	URL to send the JSON with data
LOGGER_registerSize: 300	The size of internal registry of MTX
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_httpMode: getjson	Sending mode HTTP GET (JSON)
MODBUS_address: 1	Modbus address to be read
MODBUS_start: 20	Initial modbus register address to be read
MODBUS_numwords: 4	N. of records read from each device
MODBUS_readCommand: 3	Reading command
MODBUS_period: 900	Period in seconds within which a reading is done

Details:

- This scenario is only recommended from firmware MTX-Tunnelv 9.18
- The process is as follows: MTX independently reads Modbus registers of the PLC every 900 seconds via RS485 port. After being read, the registers are stored in MTX modem's internal memory.If there is 4G/3G/2G coverage, the modem will send the register to the configured Web server.If not, they will be sent later when there is 4G/3G/2G again
- Any moment it is possible to carry out a real-time reading of Modbus registers of the PLC from a Control PC, or update the PLC firmware. To do this task, a IP-RS485 transparent gateway is configured through TCP20011 port. When a connection to the port is established (and due to that the IP-RS485 is established), the internal process of MTX modem responsible for interval reading (every 900 seconds) of PLC Modbu registers will be temporarily suspended (to avoid collisions in RS485 bus). This way it is possible to access RS485 port of the PLC directly to carry out a real-time reading of the PLC registers or update its software. When the IP-RS485 gateway established in TCP 20011 is closed, the internal process of MTX modem is resumed, which allows again reading PLC registers every 900 seconds
- With the parameter TCP_port2: 20011 modem opens a socket server in the TCP port 20011 to be able to establish a IP-serial gateway of the secondary serial port of MTX modem; in this case it is RS485 port. TCP_port: 0 parameter does not configure any server socket associated with the main serial port, because we don't need it
- It is strongry recommended to set a value in the configuration parameter MTX_serverTimeout for this example. Otherwise, if a gateway is established in the port TCP 20011 and when finished working it does not close the connection correctly (for example, unexpected connection failure because of the lack of power supply of the Control PC, Ethernet cable extraccion,..), the modem will not resume the reading process every 900 seconds. The timeout corresponding to 300 set in the parameter MTX_serverTimeout makes it possible that if gateway disconnection in the port TCP20011 goes wrong, the modem will close the connection in the port automatically after 300 seconds (5 minutes) without traffic, and the reading process will be resumed
- Si desde el PC de control quiere realizar una pasarela Modbus TCP a Modbus RTU (en lugar de la pasarela Modbus RTU sobre TCP configurada en este ejemplo), puede añadir el parámetro MTX_gatewayModbus: on
- If you want to establish, from the Control PC, a Modbus TCP to Modbus RTU gateway (instead of Modbus RTU over TCP as show this example), you can add the parameter MTX_gatewayModbus: on
- JSON object sent to the URL specified in LOGGER_server is encoded the following way:

  {"IMEI":353234028103206,"P":"ID00001","TYPE":"MODB","A":1,"TS":"20/08/12 08:31:44","ST":20,"V1":23,"V2":275,"V3":274,"V4":32765}

  That means Web server receives a JSON object with modem's IMEI, a password field (P) that can be also used to identify the device (if you do not want to use IMEI), device modbus address (A), time stamp (TS) of when modbus data was read, the initial address read (ST) and V1,V2,... with each variable read.
- Remember that the modem used in the given example has 2 serial ports: the main port(RS232) and the secondary port (RS232 or RS485). To configure the secondary port RS485 you should set miscroswitch 2 in "ON", as it is shown in the Annex A of this manual

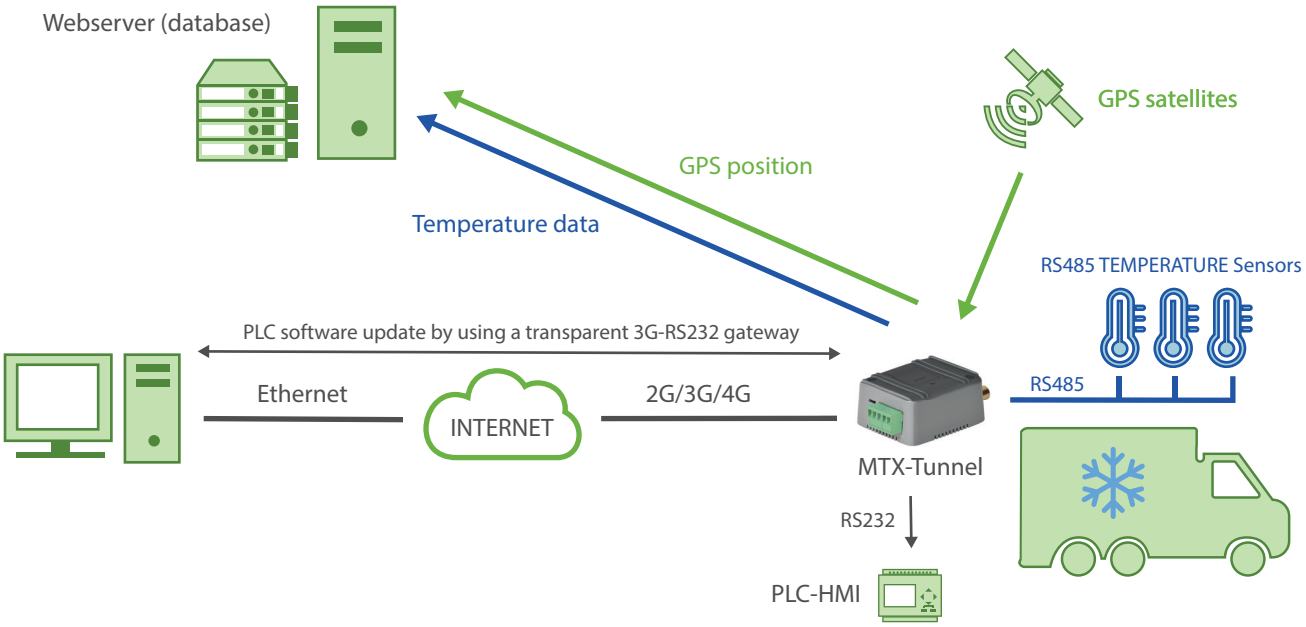
6.10 EXAMPLE: Periodical sending of vehicle's GPS locations and the registers of Modbus RTU temperature sensor RS485 to a Web server. Occasional 3G-RS232 gateway for PLC's firmware actualization.

Scenario details:

- We have a truck whose GPS location and temperature of the 3 temperature sensors should be monitored. The truck is also equipped with a control PLC with RS232 serial port.There should be the possibility to update the PLC's firmware remotely by the above mentioned serial port at any moment
- The modem must collect GPS location every 1 minute and send it in JSON format to a Web server throught HTTP POST
- The modem must read the temperature from the RS485sensors every 10 minutes and send it to a Web server throught HTTP POST
- The modem should be ready for updating the control PLC at any moment, therefore a 3G-RS232 gateway must be activated in TCP 20010 port
- The modem should be able to be configured remotely via SMS and Telnet

Solution:

MTX-IoT [4-S-N-N]-N-GPS modem+firmware MTX-Tunnel



EXAMPLE of settings (file config.txt) for such scenario:

COMM_baudrate: 19200	Serial port baud rate (PLC HMI communication)
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN of SIM card
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: server	3G-RS232 gateway configured for the current PLC
MTX_model: 199801452	MTX modem model
MTX_portAux: modbusmaster	The aux port used as master modbus MTX
MTX_TPProtocol: ntp	Time synch. protocol

MTX_TPServer: ntp.roa.es	Time server (the MTX must sync the time)
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_rssiLevel: 10	Coverage led of MTX-65i activated
MTX_serverTimeout: 300	If no gateway traffic in 300s socket server closed
TCP_port: 20010	Principal serial port gateway RS232
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet port 20023
LOGGER_enabled: on	We enable the MTX Logger, to store the records
LOGGER_password: ID001	Field to identify the origin of the frames
LOGGER_server: dominio.com/datos.asp?datos=	Server URL, will receive JSON data
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX

LOGGER_httpMode: postjson	HTTP POST mode (JSON) selected
MODBUS_address: 1;2;3	Modbus addresses to be read
MODBUS_start: 1;1;1	Initial address of each sensor to be read
MODBUS_numwords: 1;1;1	Registers number to be read
MODBUS_period: 300	Period in seconds within which a reading is done
MODBUS_readCommand: 4;4;4	Sensors read with modbus 0x04 command
MODBUS_regType: 2;2;2	The registers have 2 bytes (unsigned word)
GPS_period: 60	Every minute one GPS position is read
DNS_enabled: on	DNS service activated to inform periodically
DNS_password: ID00001	Field says where frames come from
DNS_server: www.miservidorWeb.com/json.asp	URL to send the JSON with data
DNS_mode: http	Configuration mode http
DNS_httpMode: postjson	HTTP POST is used to send data
DNS_period: 3600	Status sent when IP changes or every 3600s

Details:

- The format of JSON object that MTX-Tunnel will send to a Web server, for sensor 1 (23.0º) is similar to the following structure:

  {“IMEI”:353234028103206,”P”:”ID00001”,”TYPE”:”MODB”,”A”:1,”TS”:”20/04/13 08:31:44”,”ST”:1, “V1”:230}

  for sensor 2 (24.5º)

  {“IMEI”:353234028103206,”P”:”ID00001”,”TYPE”:”MODB”,”A”:2,”TS”:”20/04/13 08:31:44”,”ST”:1, “V1”:245}

for sensor 3 (22.1º)

{“IMEI”:353234028103206,”P”:”ID00001”,”TYPE”:”MODB”,”A”:2,”TS”:”20/04/13 08:31:44”,”ST”:1, “V1”:221}

And for GPS frames:

{“IMEI”:357044060013890,”TYPE”:”GPS”,”P”:”ID00001”,”DATE”:”2016/10/30”, “TIME”:”21:35:51”,”LAT”:”41.629803”,”NS”:”N”,”LON”:”2.3609767”,”EW”:”E”, ”ALT”:”197.61”,”SPE”:”2.98”,”COU”:”9.85”,”STA”:”3”,”HPO”:”1.25”,”VDO”:”0.75”, “SAT”:”05”}

Where:

IMEI: IMEI of the modem

TYPE: Data frame type

P: Text specified in the parameter LOGGER_password

DATE: UTC date collected directly from GPS

TIME: UTC time collected directly from GPS

LAT: Latitude

NS: N=North, S=South

LON: Longitude

EW: E=East, W=West

ATL: Altitude

SPE: Speed

COU: Direction (0 – 359º)

STA: gps status (0=No fix, 2=2D, 3=3D)

HPO: horizontal accuracy (the closer to 1 the better)

VDO: vertical accuracy (the closer to 1 the better)

SAT: Satellite nº

- Remeber that the modem model used in this example has 2 serial ports: the main port (RS232) and the secondary port (RS232 and RS485). To configure the secondary port as RS485 you should set microswitch 2 in “ON”, as it is shown in Annex A of this manual

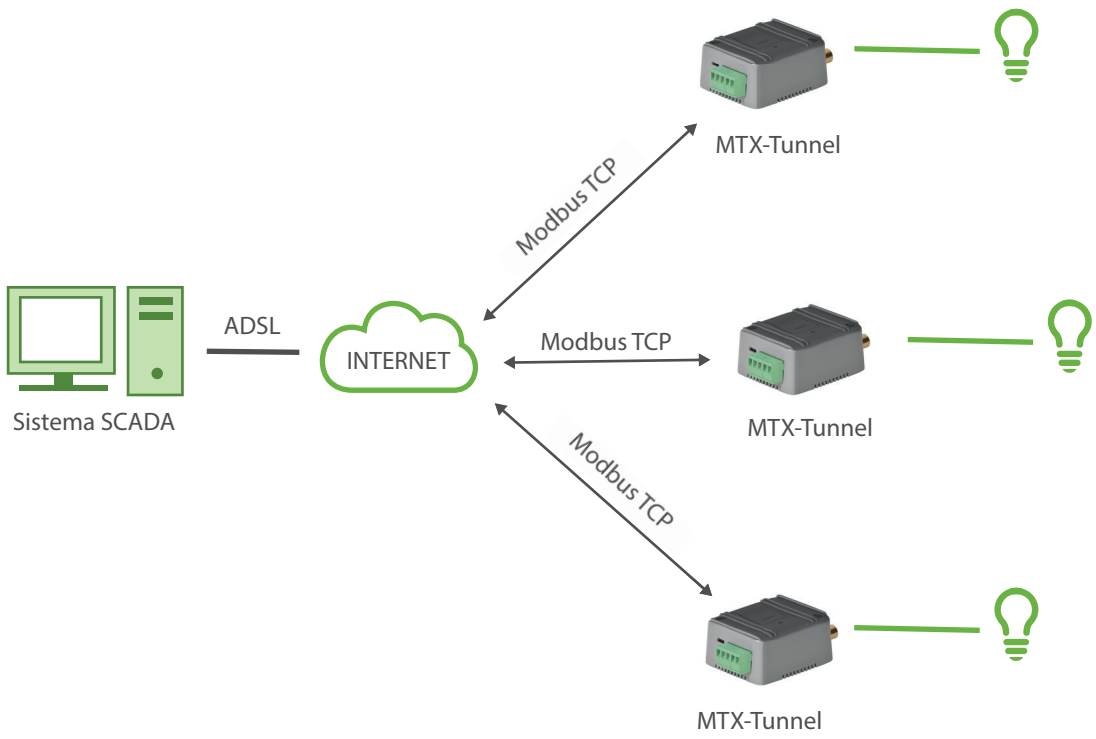
6.11 EXAMPLE: Relays control, MTX modem digital and analog inputs from a SCADA system using Modbus TCP protocol.

Scenario details:

- It is necessary to control 4 relays located in 3 locations, as well as 4 digital inputs and 2 analog inputs. This will use 3 4G/3G/2G modems, each in a different location
- The communication will be made from a SCADA system using Modbus TCP protocol
- For simplicity it is intended to use SIM cards with public IP address, so it is necessary for Modbus slave devices to have some security mechanism (via Password) for Modbus communications and to prevent any intruders from acting on the relays
- The modem must be able to be configured remotely by SMS, Telnet and Modbus

Solution:

MTX-IoT [4-S-N-N]-STD-N-RL modem+firmware MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: internetestatico.movistar.es	GPRS APN from your GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_PIN: 0000	The SIM card PIN
MTX_mode: none	No 3G-RS232 gateway is required
MTX_model: 199801451	Modem model where the MTX-Tunnel is installed
MTX_ping: 35	Every 35 mins without communication, we will ping
MTX_pingIP: 8.8.8.8	Address where the ping is performed
MTX_rssiLevel: 10	Activate the MTX-65i coverage LED
MODBUSTCP_enabled: on	Modbus TCP service is enabled for the MTX-Tunnel
MODBUSTCP_port: 502	Specifes the TCP port to use for modbus
MODBUSTCP_password: ABCD	Specifies a Password for modbus communications
SMS_allPhones: on	Possible to send SMS with commands from phone
SMS_sendIP: on	Modem responds with IP to a missed call or SMS
SMS_ATEnabled: on	It is possible to send commands to MTX via SMS
SMS_ATResponse: on	MTX responds with SMS to a sent SMS command
FIREWALL_enabled: off	It is possible to connect to the modem from any IP
TELNET_enabled: on	We enable the Telnet of the modem
TELNET_login: user	Login for Telnet
TELNET_password: 1234	Password for Telnet

Details:

- Remember that when specifying a Password in the MODBUSTCP_password parameter, each time your scada system establishes a communication socket with the MTX modem, you must enter the specified Password (in the case of this ABCD example) in the modbus addresses 50, 51, 52 and 53 as shown in this manual, which details the parameters MODBUSTCP_

  Specifically, you need to write 65 (ASCII corresponding to A) in the register @50, 66 (ASCII corresponding to B) in the register @51, 67 (ASCII corresponding to C) in the register @52 and 68 (ASCII Corresponding to D) in register @53.

- For simplicity's sake we have not used it, but remember there are FIREWALL_ parameters you can use to increase the security in all communications
- Remember in Annex A you'll find a table with each modem's I/Os, modbus addresses, etc. For example, the example's model:

EHS6	CONNECTOR	PIN N.	I/O	FUNCTION	MODBUS
GPIO1	DB15	4	Digital input	Wake up/pulse counter	1 / 13-14
GPIO2	DB15	11	Digital input	SMS alarm/pulse counter	2/ 15-16
GPIO3	DB15	9	Digital input	User	3
GPIO4	DB15	5	Digital output	User	4
GPIO5	DB15	12	Digital output	User	5
GPIO6	DB15		Relay	User	6
ADC1	DB15	15	Analog input	User	11
ADC2	DB15	13	Analog input	User	12
VExt	DB15	10	Output voltage	4V	--
GND	DB15	14	Ground	Ground	--

- For instance, if you want to activate the relay via modbus, you'll have to write "1" in the modbus registry 6. If you want to deactivate the relay, write "0"

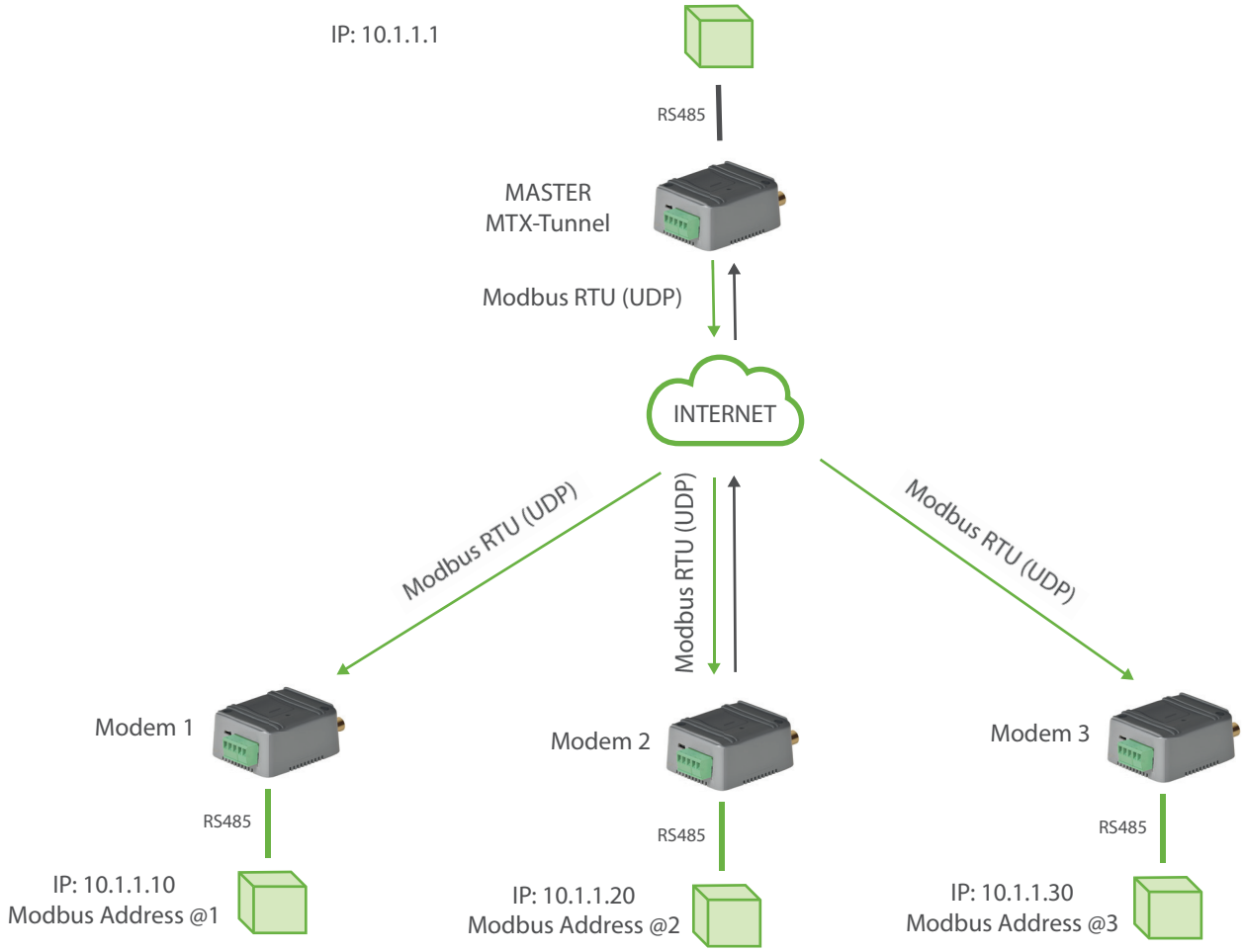
6.12 EXAMPLE: Providing IP to Master Modbus RTU and Modbus RTU slaves.

Scenario details:

- We have a PLC with Modbus RTU RS485 communications which, acting as master, performs periodic readings of 3 slave devices also with Modbus RTU RS485 communications
- It is necessary to replicate that same scenario, with the same RS485 devices, but with IP communications, since it is not possible to carry out a wiring when the different elements are located at kilometers distances between them
- SIM cards will be used with private APN, so each device will have a MTX-Tunnel modem with fixed IP address to facilitate communications and increase security

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+firmware MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario. MTX-Tunnel MASTER:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: udp	UDP communications will be used
MTX_model: 199801393	MTX modem model
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds with IP to missed call or SMS
SMS_ATEnabled: on	It is possible to send commands to MTX via SMS
SMS_ATResponse: on	MTX responds with SMS to a sent SMS command
FIREWALL_enabled: off	It's possible to connect to the modem from any IP
TELNET_enabled: on	We enable the Telnet of the modem

TELNET_login: user	Login for Telnet
TELNET_password: 1234	Password for Telnet
TELNET_port: 20023	Telnet port for remote configuration changes
UDP_IP: 10.1.1.10@1,10.1.1.20@2,10.1.1.30@3	IPs and slave Modbus RTU address are indicated
UDP_ localPort: 20010	Local communications UDP port
UDP_ remotePort: 20010	Remote communications UDP port

Configuration example (config.txt file) for the indicated scenario. MTX-Tunnel SLAVES:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: miapn.movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: udp	UDP communications will be used
MTX_model: 199801393	MTX modem model
MTX_ping: 35	Ping every 35 minutes without comms

MTX_pingIP: 8.8.8.8	IP address to ping
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds with IP to missed call or SMS
SMS_ATEnabled: on	It is possible to send commands to MTX via SMS
SMS_ATResponse: on	MTX responds with SMS to a sent SMS command
FIREWALL_enabled: off	It's possible to connect to the modem from any IP
TELNET_enabled: on	We enable the Telnet of the modem
TELNET_login: user	Login for Telnet
TELNET_password: 1234	Password for Telnet
TELNET_port: 20023	Telnet Port for Remote Configuration Changes
UDP_IP: 10.1.1.1	Modbus master IPs are indicated
UDP_ localPort: 20010	Local communications UDP port
UDP_ remotePort: 20010	Remote communications UDP port

Details:

- To solve this scenario UDP communications are used. Each time the modem is connected to the master PLC it receives a communication frame. Modbus analyzes the received frame by looking for the Modbus RTU address of the slave to which it is addressed. Once the address is found, the frame is sent only to the IP associated with the corresponding slave modbus RTU device

  Example for the following configuration:

  UDP_IP: 10.1.1.10@1.10.1.1.20@2.10.1.1.30@3

  If the master modem receives a modbus RTU frame and verifies that the destination address is for the slave device with address modbus @2, it forwards the received frame to the associated address, in this case 10.1.1.20.
- Other devices on the market forward the frame to all associated devices, with the consequent problem of increasing data traffic and, therefore, higher economic cost and lower communication speed
- Do not forget to establish a communication timeout in the master PLC of at least 2 seconds

6.13 EXAMPLE: Reading and sending Modbus registries periodically from a PLC to an FTP server.

Scenario details:

We have a PLC Modbus RTU. This PLC has a series of variables/registries (for instance, temperature and 3 meter boxes) inside its internal memory, which need to be read and sent periodically to an FTP server

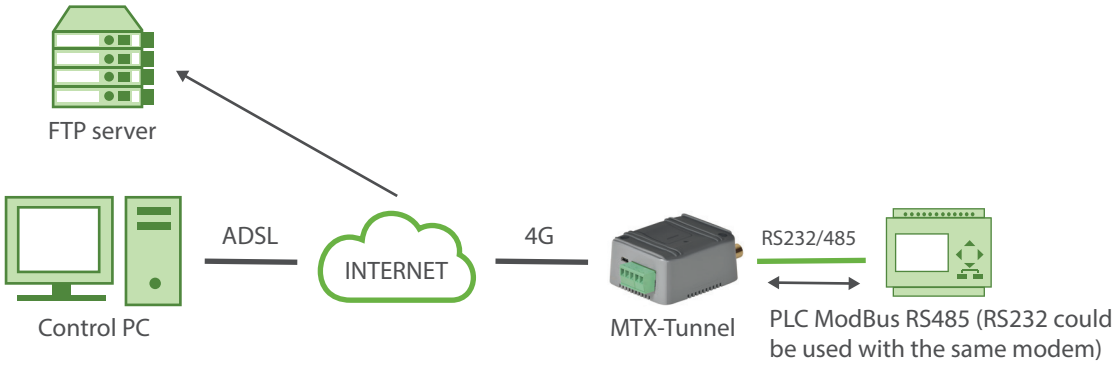
That is why the MTX-Tunnel needs to ask the PLC to read said registries every 15 minutes via a serial port. The registries to be read are, for the temperature the registry number 20, and for the meter boxes the registries number 21, 22 and 23 respectively

After every Reading, the MTX-Tunner should send the registries value to a web server via FTP using a JSON object, and in case of GPRS communication failure it should be able to store up to 1500 readings in its falsh memory that will be sent when communication is restored

You should be able to access MTX-Tunnel anytime so you can read the PLC registries in real time, as well as writing on them and modifying PLC configuration registries

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+firmware MTX-Tunnel



EJEMPLO de configuración (fichero config.txt) para el escenario indicado:

COMM2_baudrate: 9600	Serial port (where internal GPS is connected) rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autorts: off	No flow control

COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	MTX-Tunnel mode
MTX_model: 199801393	MTX modem model
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_rssiLevel: 10	AWe activate the MTX-65i coverage led
SMS_allPhones: on	All phone numbers are authorized
SMS_sendIP: on	IP sent to phone which called or “on” SM
SMS_ATEnabled: on	Remote AT commands by SMS enabled
SMS_ATResponse: on	Modem response to AT command with SMS
FIREWALL_enabled: off	Any IP will be able to connect to the modem
TELNET_enabled: on	Telnet is activated

TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Use TCP port 20023
LOGGER_enabled: on	We enable the MTX Logger, to store the records
LOGGER_password: ID00001	Password field can be used as ID device
LOGGER_server: ftp.myFTPServer.com	FTP server address
LOGGER_serverLogin: myUsername	FTP server username
LOGGER_serverPassword: myPassword	FTP server password
LOGGER_registerSize: 300	Register size
LOGGER_numRegistersFlash: 1500	Maximum number of records in MTX
LOGGER_mode: ftp	Sending mode: ftp
MODBUS_address: 1	Modbus address of the computer to read
MODBUS_start: 20	Address of the initial register modbus to read
MODBUS_numwords: 4	Number of records to be read from the beginning
MODBUS_readCommand: 3	Comando de lectura
MODBUS_period: 900	Every few seconds a reading is made

Details:

- This example is about an MTX-IoT [4-S-N-N] using an RS485 port, but we could do the same using the RS232 introducing the MTX_invertedCom parameter: on. At the end of the guide it is explained how to configure the internal microswitch to activate the RS485 bus
- Summarizing this exaple, the modem reads every 15 minutes a series of Modbus registries from the PLC and sends them through a JSON object to an FTP server (to the address specified in the LOGGER_server parameter). If the registry could not be sent because the server or the 4G/3G/2G connection is down, it stores the data in its memory and sends them afterwards. You can connect directly to the equipment through Telnet, and check/change the PLC registries in real time. To do that, look for the AT^MTXTunnel=getmodbus and AT^MTXTUNNEL=setmodbus

commands in this guide

- The JSON object sent to the FTP server is coded the following way, as an example:

  {“IMEI”:353234028103206,”P”:”ID00001”,”TYPE”:”MODB”,”A”:1,”TS”:”20/08/12 08:31:44”,”ST”:20,”V1”:23,”V2”:275,”V3”:274,”V4”:32765}

  That is, the web server receives a JSON object with the modem IMEI (IMEI), a password field (P) that can be also used to identify the equipment in case you do not want to use the IMEI, the modbus address of the equipment (A), the time stamp (TS) when the modbus data has been read, the initial address read (ST) and V1, V2, ... with each of the read variables.

Important note: for compatibility’s sake, a different file is written for each registry sent to the FTP server. The name of the file is IMEI-fechaUTCdeEnvio. In case you have several módems, you can easily manage the files in your FTP server since they have an IMEI (identifying number for each modem) in the name of each file. In “fechaHoraUTCdeEnvio” you can find the UTC time the modem had (may the time be synchronized or not) in the moment of being sent to the FTP server.

Example of name of file: 357042064802888-17-09-17-19-42-29.

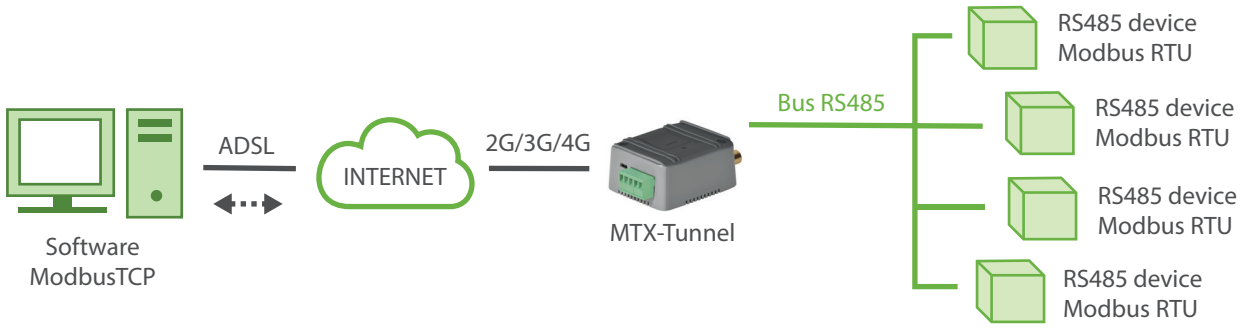
6.14 EXAMPLE: Modbus TCP/Modbus RTU 4G/3G/2G gateway.

Scenario details:

- We need to monitor remote ModBus RTU devices with the RS485 port from a central Server which has a software application to control ModBus TCP. MTX-Tunnel will create a ModBus TCP/ModBus RTU GPRS gateway
- ModBus remote devices need to be available at all times, so the modem connected to those devices is permanently connected to the 4G/3G/2G waiting for incoming requests in the standard ModBus TCP port number 502
- SIM cards provisioned with fixed IP addressing will be used, but we could also use dynamic IP addresses and DynDNS or our own DNS server in the same way

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autocts: off	No flow control
COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity

GPRS_apn: internetestatico.movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_model: 199801436	MTX-Terminal modem model used
MTX_mode: server	TCP server mode
MTX_urc: off	URC messages will not be sent
MTX_gatewayModBus: on	ModBus TCP – ModBus RTU gateway enabled
MTX_invertedCom: on	Main port (for GPRS-Serial gateway)
TCP_port: 502	TCP port used. Standard ModBus
FIREWALL_enabled: off	Any incoming connection form any IP is allowed

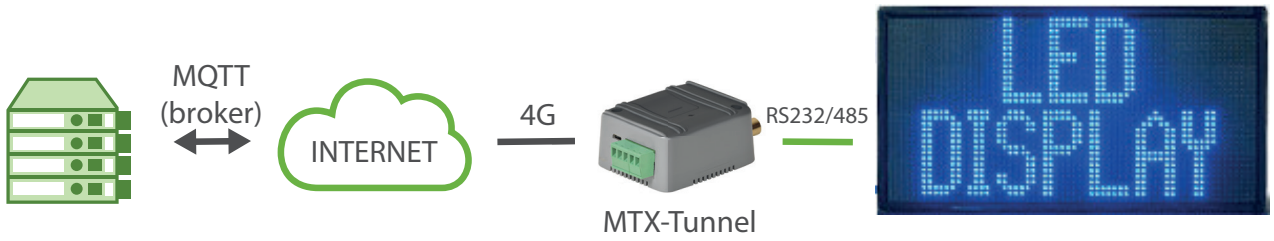
6.15 EXAMPLE: Reading/writing MODBUS records on RS232 device using MQTT communication.

Scenario details:

- A special screen with RS232 serial input is available to communicate with using modbus protocol. It is possible to display characters on the screen by writing in its modbus registries through the RS232 serial port
- We can control the screen remotely via 4G with an MTX modem with MTX-Tunnel firmware
- To avoid connectivity problems (SIMs with private IPs) MQTT protocol will be used, so the modem will automatically connect to the configured MQTT broker
- Once the modem is connected to the broker, the modem will periodically send its status (IP, coverage, etc) to a MQTT topic
- In order to write and read the modbus registries on the screen, the commands AT ^ MTXTUNNEL = GETMODBUS, ... and AT ^ MTXTUNNEL = SETMODBUS will be sent to the modem via MQTT

Solution:

MTX-4G-T modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	8 bit data
COMM2_autocts: off	No flow control
COMM2_autorts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity

GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_model: 199801421	MTX-Terminal modem model used
MTX_mode: none	Gategay disabled
MTX_urc: off	URC messages will not be sent
MTX_ping: 30	Minutes for connectivity supervision ping
MTX_pingIP: 8.8.8.8	Connectivity supervision IP address
MTX_invertedCom: on	RS232 port used as secondary
MTX_portAux: modbusmaster	Secondary port as modbus master
SMS_allPhones: on	All phones are authorized
SMS_ATEnabled: on	AT commands enabled by SMS
SMS_ATResponse: on	Replies to AT commands enabled by SMS
MQTT_enabled: on	MQTT service enabled
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT1	Subscribed to this topic to receive commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_qos: 1	Service quality

MQTT_keepalive: 300	Connection keep alive (300 seconds)
MQTT_persistent: off	Not relevant
DNS_enabled: on	Sending status periodically
DNS_mode: mqtt	Sending mode
DNS_mqttTopic: [IMEI]/mqtt	Topic to send status data to
DNS_extended: off	Won't send extended information (GPIOs)
DNS_period: 120	Sending status every 120 seconds

Details:

- Remember what modem will replace the [IMEI] tags with their IMEI (unique identifier)
- The modem will subscribe to the MQTT topic "[IMEI]/AT1", so all the AT commands sent to the MQTT topic will be received and executed by the modem
- The responses to the executed AT commands will be sent to the MQTT topic: "[IMEI]/ATR"
- Let's say the screen allows to visualize 10 characters, and the modbus registries on the screen are from @10 to @19. Let's suppose it is enough to write the ASCII code in those registries so they are visualized. The screen will have the modbus address @7

  If we want to write the word "HELLO" on the screen, which corresponds to the ASCII: 72, 79, 76 and 65, we will need to write those values in the modbus position 10, 11, 12 and 13, so we will send the following command via MQTT to the topic [IMEI]/AT1:

  AT^MTXTUNNEL=SETMODBUS,7;10;72;79;76;65

  This command writes in the modbus device with address @7, from register number @10, the values 72, 79, 76 and 65
- If we want to read the modbus registries on the screen, we use the AT ^ MTXTUNNEL = GETMODBUS command. If we want to read the same modbus registries from the previous section, we send the following AT command via MQTT:

  AT^MTXTUNNEL=GETMODBUS,7;10;3;4

  This command reads the modbus device with address @7, from register number @10, using the modbus 3 command and will read 4 registers. The result of the AT command the modem will send to the "[IMEI]/ATR" topic will be:

  AT^MTXTUNNEL=GETMODBUS,7;10;3;4

  72,79,76,65

  OK

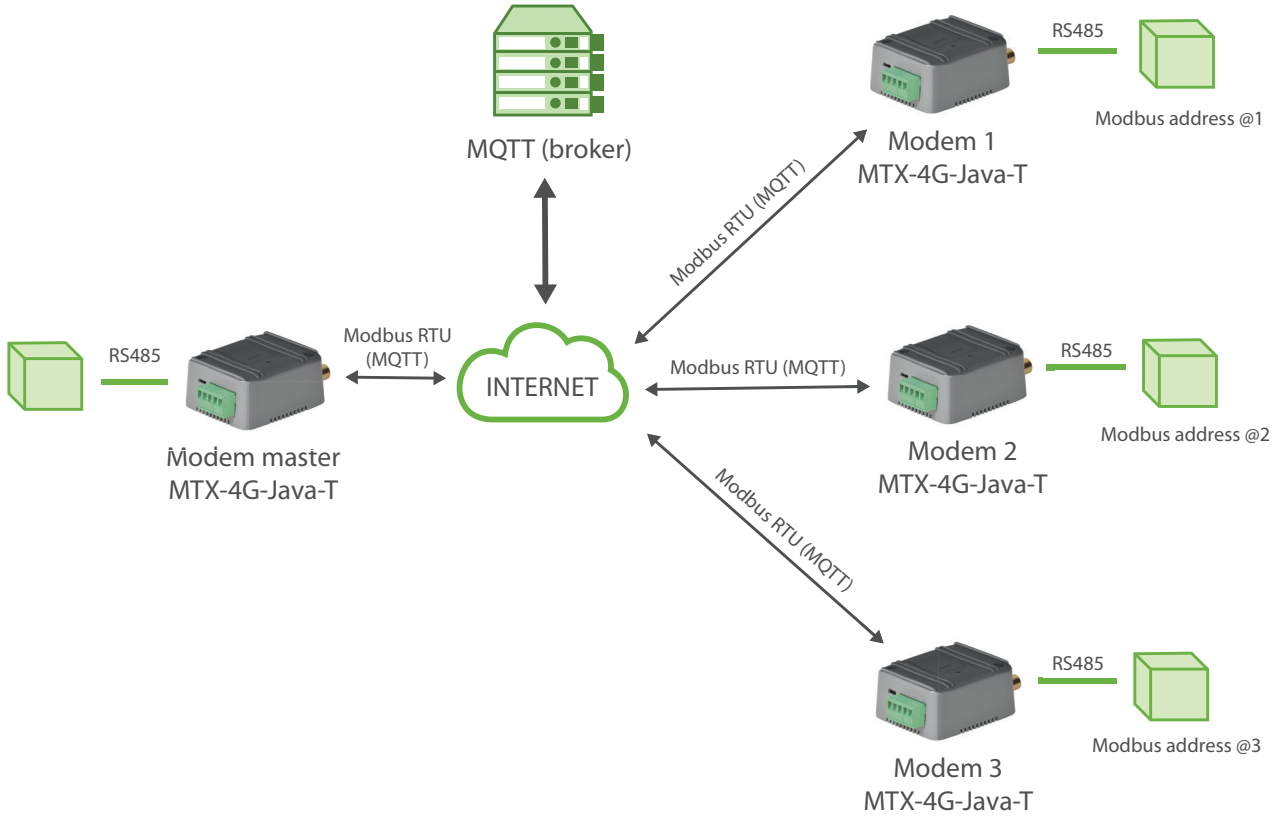
6.16 EXAMPLE: About how to provide MQTT communications to master Modbus RTU and Modbus RTU slaves.

Scenario details:

- We have a PLC with RS485 Modbus RTU communications that, acting as master, performs periodic readings of 3 slave devices also with Modbus RTU RS485 communications
- We want to replicate the same scenario with the same devices, but with IP communications, since it is not possible to make a wiring to find the different elements located far away
- We will use affordable SIM cards with dynamic and private IP, so the devices can't communicate with each other. An MQTT broker will be used as an intermediary. The master will send the modbus requests to the MQTT broker, which will forward the slaves via MQTT. The responses of the slave modems are forwarded to the MQTT broker, which in turn are forwarded to the master modem

Solution:

MTX-T [4-N] modem+MTX-Tunnel firmware



Config.txt configuration file (master):

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_mode: mqtt	MQTT serial gateways will be used
MTX_model: 199801445	MTX-Terminal modem model used
MTX_ping: 35	One ping every 35 min. without communications
MTX_pingIP: 8.8.8.8	Ping address
MTX_invertedCom: on	Gateway port RS485
MTX_msToSend: 250	No fragmented networks
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds with its IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS

FIREWALL_enabled: off	Any incoming connection form any IP is allowed
MQTT_enabled: on	MQTT service enabled
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
MQTT_commrxtopic: rxmaster	Data received will be retransmitted via serial
MQTT_commrxtopic: txmaster	Data received v/serial, retransmitted to this topic

Config.txt configuration file (slave):

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value

MTX_mode: mqtt	MQTT serial gateways will be used
MTX_model: 199801445	MTX-Terminal modem model used
MTX_ping: 35	One ping every 35 min. without communications
MTX_pingIP: 8.8.8.8	Ping address
MTX_invertedCom: on	Gateway port RS485
MTX_msToSend: 250	No fragmented networks
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds with its IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
FIREWALL_enabled: off	Any incoming connection form any IP is allowed
MQTT_enabled: on	MQTT service enabled
MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
MQTT_commrxtopic: txmaster	Data received will be retransmitted via serial
MQTT_commrxtopic: rxmaster	Data received v/serial, retransmitted to this topic

Details:

- The master modem forwards all the modbus requests it receives in its RS485 serial port to the MQTT broker, specifically to the “txmaster” topic. The slave modems, as they are subscribed to the topic “txmaster” of the MQTT broker, automatically receive said requests from the master, which in turn are forwarded by their RS485 serial port

  And the other way around, the slave modems forward all the data they receive through their RS485 serial port (the modbus responses) to the mqtt broker, to the “rxmaster” topic. As the master modem is subscribed to the topic “rxmaster”, the MQTT broker immediately sends us those responses, which are forwarded through its RS485 serial port

- Keep in mind that communications latencies may be somewhat greater than direct communication latencies since there is an intermediary (the mqtt broker) and the speed of communications will depend on the power of the latter. Set the timeout if necessary

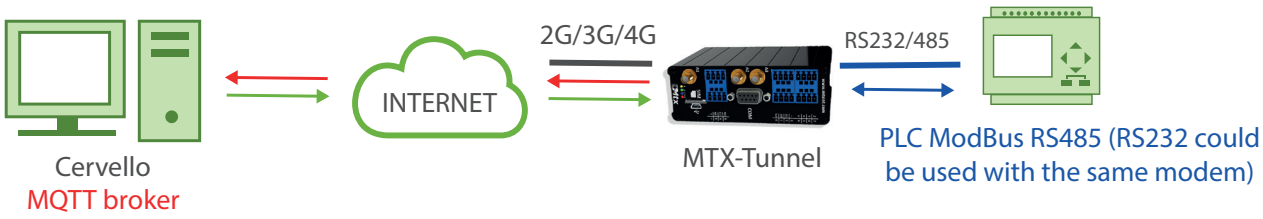
6.17 EXAMPLE: Reading and periodic sending to an MQTT broker of the MODBUS registers of a PLC. DUAL SIM configuration due to loss of connectivity of an operator.

Scenario details:

- We have a Modbus RTU PLC. This PLC has in its internal memory a series of variables / registers (for example, a temperature and 3 counters, ...) which must be read and periodically sent to an MQTT broker
- Therefore, the MTX-Tunnel must periodically interrogate the PLC through a serial port every 15 minutes to read these registers. The registers to be read are: for the temperature register nº20, and the counters are in registers 21, 22 and 23 respectively
- The MTX-Tunnel must send the value of the registers after each reading to an MQTT broker using a JSON object, but it must be able, in case of 2g / 3g / 4g communications failure, to store in flash memory up to 1500 readings that it will send when communications are restored
- For more security in communications, the modem must have DUAL SIM. In other words, the modem must have 2 SIM cards from 2 different telephone operators. The modem will need to change SIMs as long as it cannot get an IP address for more than 120 seconds

Solution:

MTX-IOT-S [4-N] modem+MTX-Tunnel firmware



Config.txt configuration file:

COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	8 bit data
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_apn2: airtelnet.es	GSM operator GPRS APN. Secondary SIM
GPRS_login2: vodafone	GSM operator GPRS login. Secondary SIM
GPRS_password2: vodafone	GSM operator GPRS password. Secondary SIM
GPRS_timeout: 0	The modem will be permanently connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_mode: none	Working mode. None because we do not use them
MTX_model: 199802407	MTX-Terminal modem model used
MTX_portAux: modbusmaster	The aux port is to be used as modbus master
MTX_TPProtocol: ntp	Time synchronization protocol
MTX_TPServer: ntp.roa.es	Time server (MTX must synchronize time)
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	One ping every 35 min. without communications

MTX_pingIP: 8.8.8.8	Ping address
MTX_rssiLevel: 10	We activate the coverage LED
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: on	Modem responds with its IP to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
FIREWALL_enabled: off	Connect to the modem (for Telnet) from any IP
LOGGER_enabled: on	We activate the MTX Logger, to store the readings
LOGGER_mode: mqtt	Shipping mode by MQTT
LOGGER_mqttTopic: /LOGGER	Sending topic to MQTT broker to send counter data
LOGGER_registerSize: 300	The size of the internal MTX register
LOGGER_numRegistersFlash: 1500	The maximum number of records within the MTX
MODBUS_address: 1	Modbus address of the device to read
MODBUS_start: 20	Address of the initial modbus register to read
MODBUS_numwords: 4	Number of registers to read from the initial
MODBUS_readCommand: 3	Read command
MODBUS_period: 900	Every few seconds a reading is taken
DUALSIM_select: dual	DUAL SIM mode activated with external initial SIM
DUALSIM_mode: ip	SIM change mode
DUALSIM_timeout: 120	Timeout for SIM change in case of problems
MQTT_enabled: on	MQTT service enabled

MQTT_server: tcp://test.mosquitto.org:1883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_login:	Username
MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Connection keep alive (60 seconds)

Details:

- In this example a modem is used using the RS485 port, but the RS232 port could be used without problems (to use the RS232 port the parameter MTX_invertedCom: on should be entered)
- The summary of this example is as follows: the modem periodically reads, every 15 minutes, a series of ModBus registers from the PLC and sends them via a JSON object to an MQTT broker (to the topic specified in the LOGGER_mqttTopic parameter). In case of not being able to send the record (because there is no coverage at that time or the server is down) it stores the data in memory to send them later. Using MQTT commands it is also possible to connect to the equipment directly and consult / change the PLC registers in real time (to do this, search in this manual for the commands AT ^ MTXTunnel = getmodbus and AT ^ MTXTUNNEL = setmodbus)
- The JSON object sent to the MQTT broker is encoded as follows, as an example:

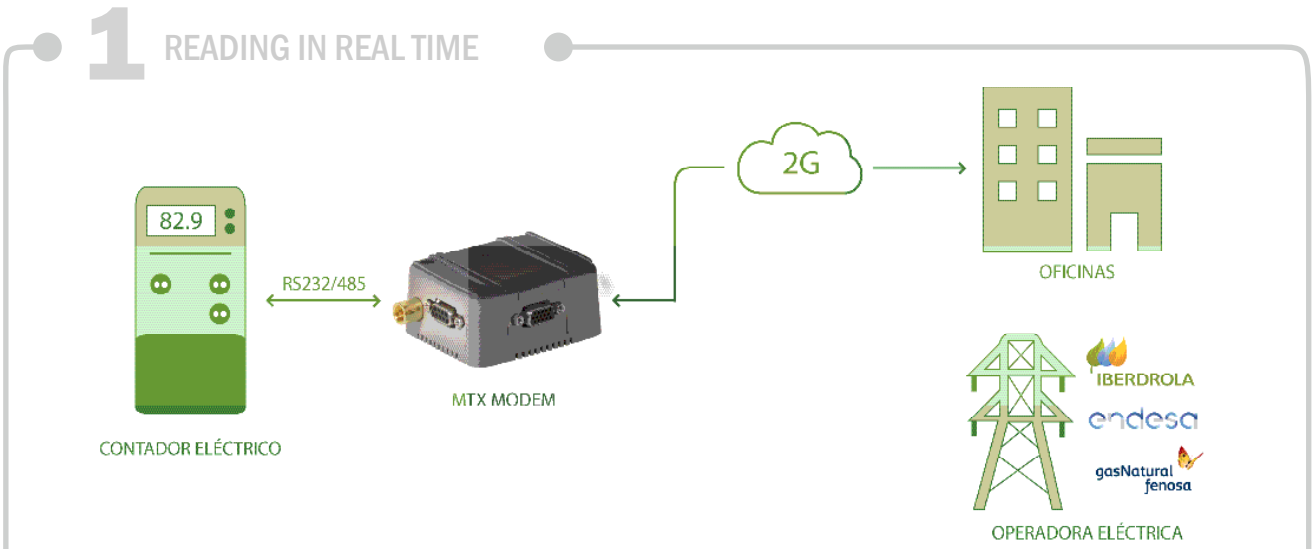
  {“IMEI”:353234028103206,”P”:”ID00001”,”TYPE”:”MODB”,”A”:1,”TS”:”20/08/12 08:31:44”,”ST”:20,”V1”:23,”V2”:275,”V3”:274,”V4”:32765}
- The modem has DUAL SIM (DUALSIM_mode: dual) configured. This will cause the modem to boot using the primary SIM card, which in the case of the MTX-IOT-S [4-N] modem is the external SIM card. In case coverage is lost and it is not possible to register on the operator network for 60 seconds (DUALSIM_timeout), the modem will switch to the secondary SIM card

7. ANNEX: Metering Scenario Examples. Meter Reading Scenarios via GSM, GPRS, 3G & 4G

7.1 EXAMPLE: Basic example for meter reading with simultaneous GPRS connections and GSM calls.

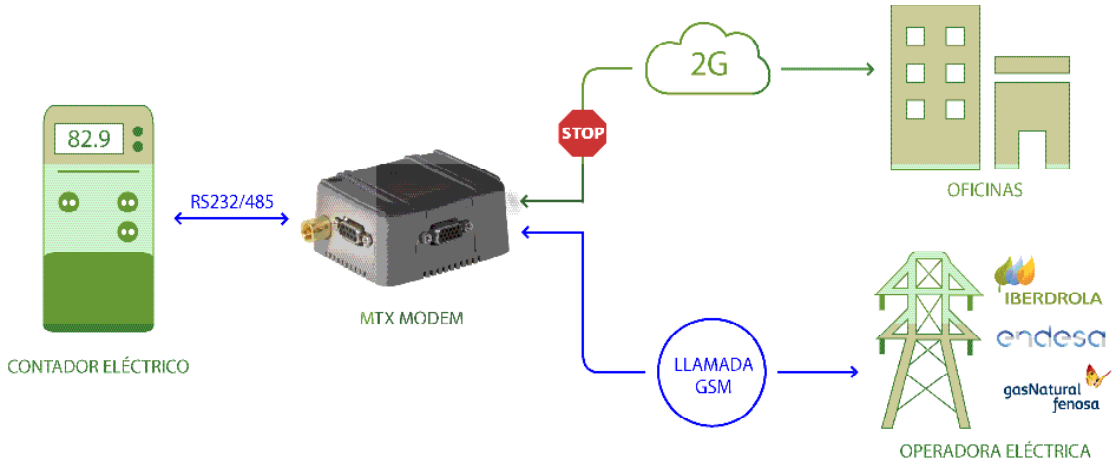
Scenario details:

- The intention is to take electrical meter readings periodically, every 10 minutes. By obtaining readings so frequently you will access the meter via GPRS instead of with GSM calls in order to save costs
- At the same time, the energy operator (Endesa, Iberdrola, etc.) will obtain a reading once a day with a conventional GSM call
- The GSM call will have priority. When the modem receives a call from the operator it will “freeze” GPRS connections to allow the operator to obtain readings. Once the GSM call has ended, GPRS connections are re-established



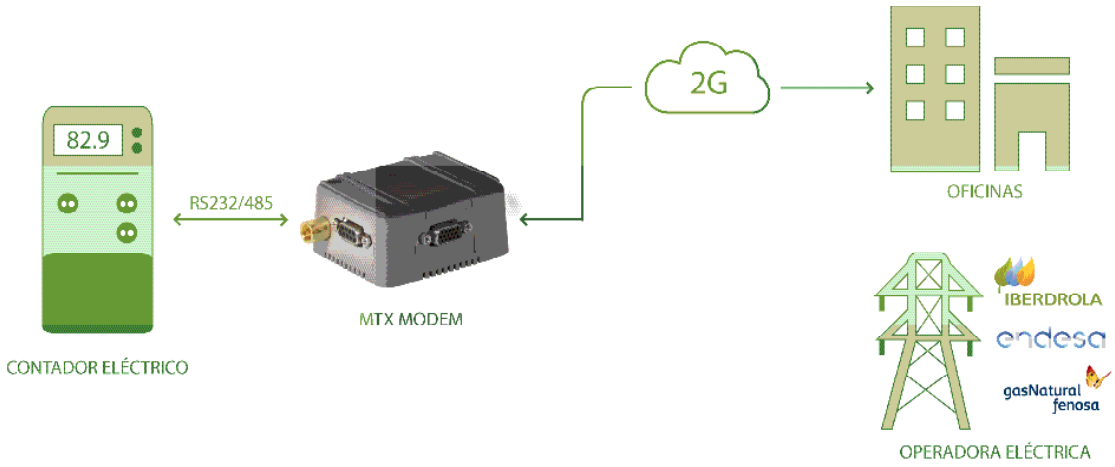
From its own offices, it is possible to carry out, in a continuous and real time, readings of the electric metering box via GPRS.

2 CALL FROM THE ELECTRIC COMPANY



When daily the electrical operator (Endesa, Iberdrola...) makes a GSM call to the modem to make a metering box reading, GPRS communications established with the counter “freeze” to give way to the operator’s call, which has priority.

3 BACK TO REAL TIME READING



Once the operator’s GSM call is finished, real-time GPRS communications are reset.

Config.txt configuration file:

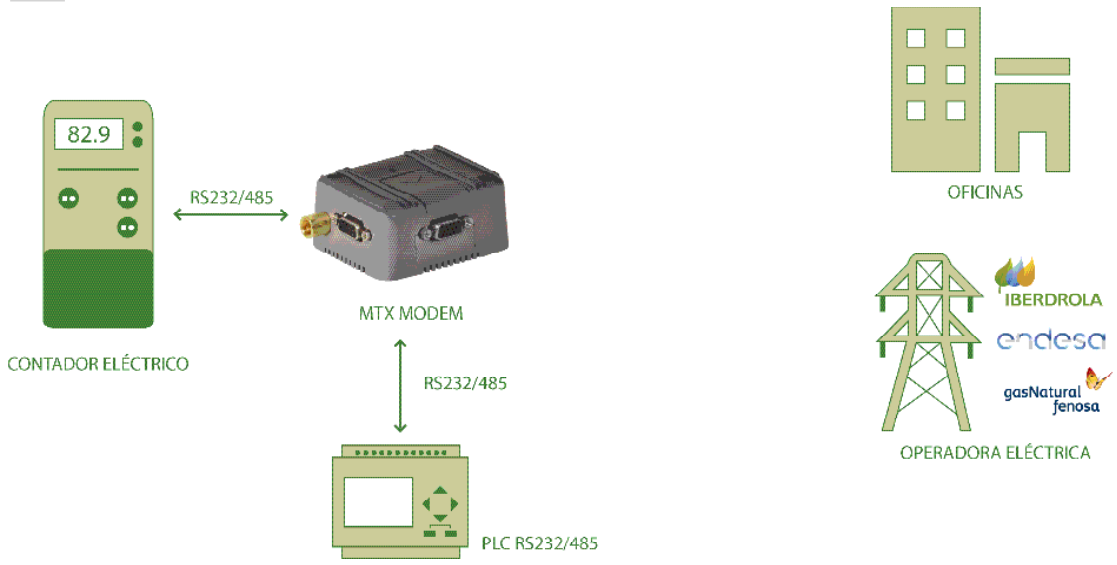
COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from the network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPRS_mode: 2g	GPRS mode
MTX_mode: server	MTX server mode
MTX_model: 199801421	MTX terminal model
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_radioBand: europe	If modem is installed in Europe
MTX_port: 20010	Establish GPRS Gateway, read the energy meter
Firewall_enabled: off	Accept incoming connections from any IP
CSD_enabled: on	Accepts GSM calls from energy operator

7.2 EXAMPLE: An advanced example for reading 2 serial devices using a single modem/SIM and simultaneously using GPRS connection and GSM calls. There is additional serial tunnelling when there are no GSM/ GPRS connections.

Scenario details:

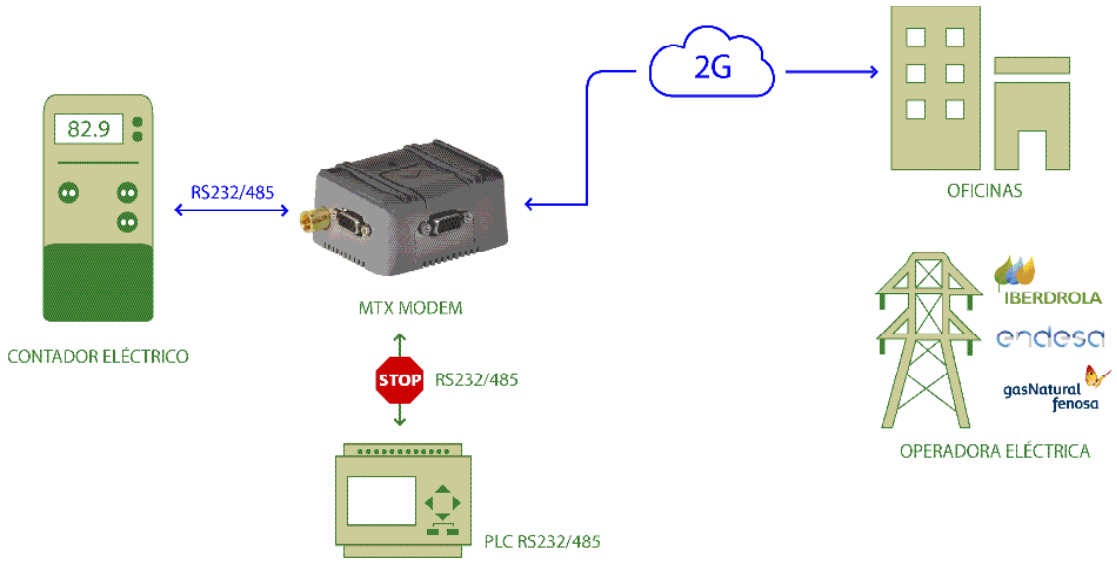
- In this scenario you have a modem with 2 serial ports. One serial port is connected to an energy meter; the other is connected to a PLC
- When there is no GPRS connection with the modem and a GSM call has not been established by the operator, the modem acts as a serial tunnel. This means that all data that arrives at the modem from the PLC must be redirected to the energy meter and vice versa – all serial data that the energy meter sends should be redirected to the PLC
- You should be able to establish 2 GPRS-serial gateways that function at the same time. Through one you will have access to the electric meter to take readings periodically; through the other you should be able to have access to modbus records from the PLC
- Additionally, the energy operator (Iberdrola, Endesa, etc.) will make a daily GSM call in order to read the meter's data. This GSM call must be a priority, freezing GPRS connections until the call is finished

1 LOCAL READING IN REAL TIME



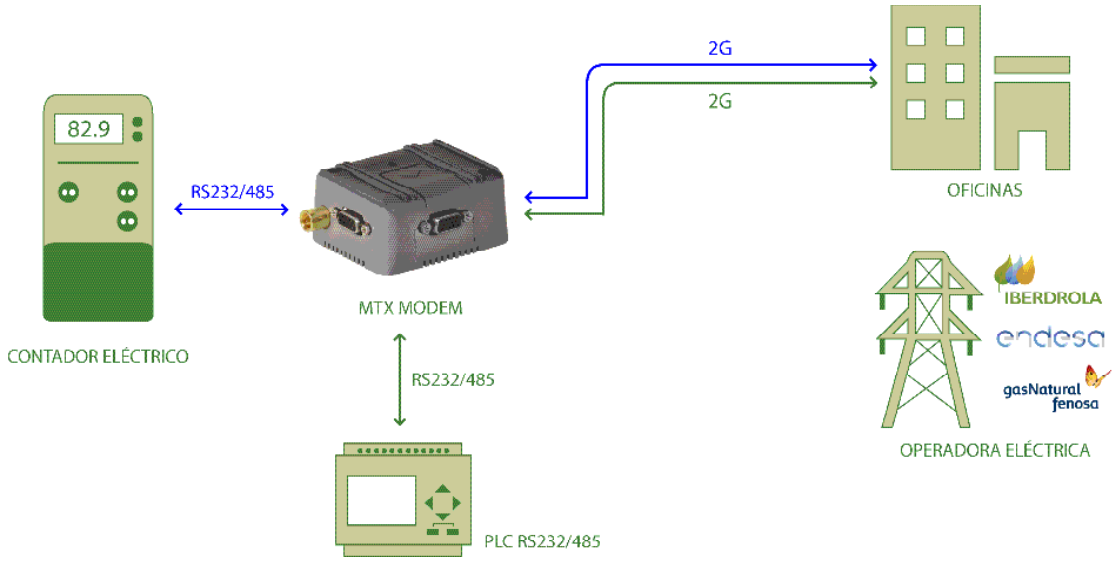
When there is no GPRS connection from your offices, neither to the modem, nor to the PLC, nor there is a GSM call from the electrical operator (Endesa, Iberdrola...), the modem acts as a transparent Serial-Serial gateway, ie, forwarding the data of the PLC directly to the metering box and vice versa, forwarding the metering box data to the PLC.

2 REMOTE READING IN REAL TIME



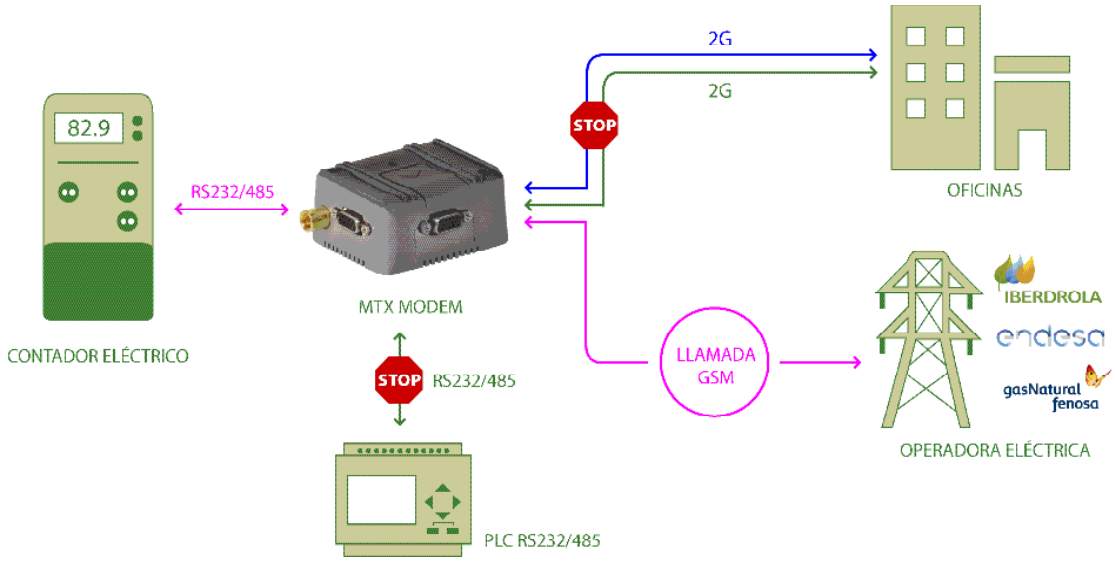
When a GPRS connection is made with the modem, the Serial-Serial gateway between the PLC and the meterig box is interrupted and the GPRS-RS232 transparent gateway between its offices and the metering box is passed, in order to carry out the readings of the metering box.

3 DUAL READING IN REAL TIME



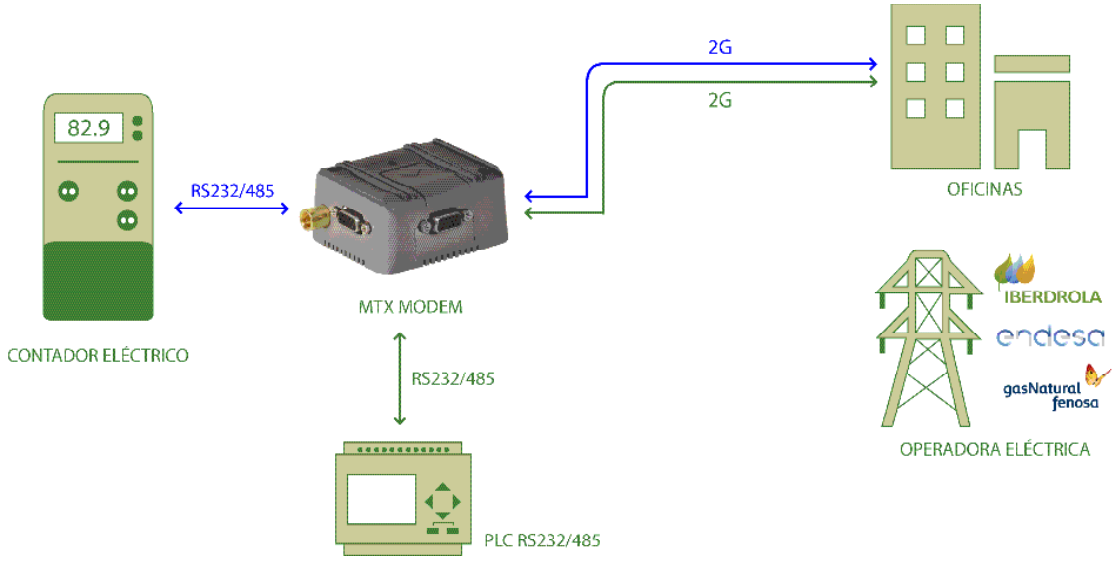
In addition, two simultaneous GPRS-RS232/485 gateways can be realized from their offices to the devices, one to access the metering box and the other to access the PLC, taking advantage of the fact that the MTX65 modem has two serial ports. Obviously the serial-serial gateway between the PLC and the metering box is still interrupted.

4 CALL FROM THE ELECTRIC COMPANY



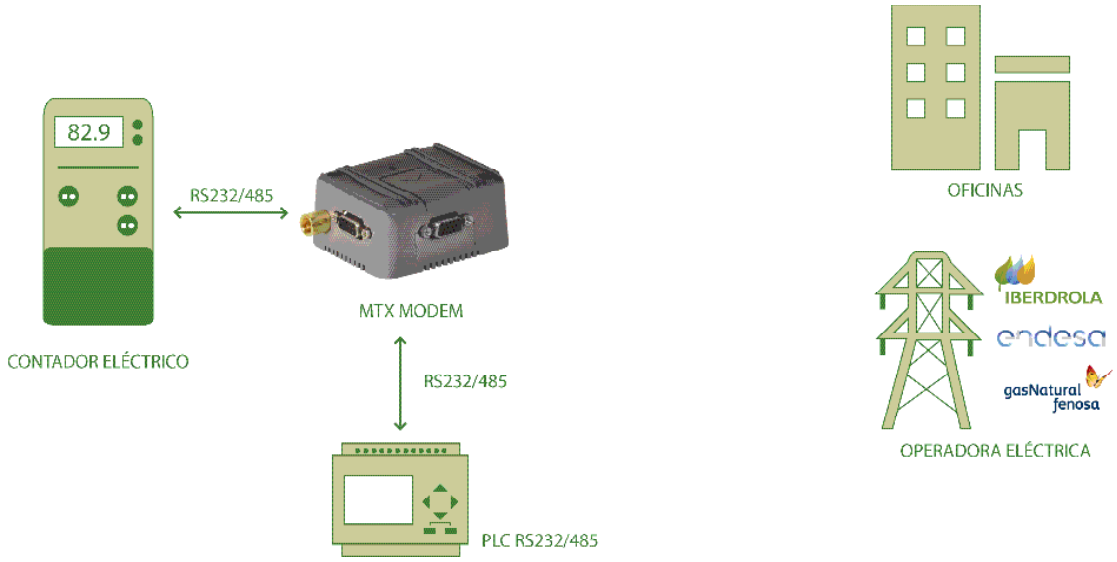
When you have two simultaneous GPRS-RS232/485 gateways, you can still access a GSM call from the metering box (Iberdrola, Endesa...) and the MTXTunnel will be able to manage it. At that moment the GPRS-RS232 / 485 gateways are frozen, giving priority to the metering box's GSM call.

5 BACK TO DUAL READING



Once the call of the energy operator (Endesa, Iberdrola...) has been completed, the GPRS-RS232/485 gateways that are established from their offices are reestablished either to read the metering box in real time or to access to the PLC, or both.

6 BACK TO LOCAL READING



When there are no GSM calls or GPRS connections set to access the metering box or the PLC, MTX-Tunnel restores the transparent serial-to-serial gateway between PLC and the electric metering box.

Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autorts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM2_baudrate: 9600	Serial port baud rate
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit

COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS from your network operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPRS_mode: 2g	IMPORTANT. For GSM call, 2G is needed
MTX_mode: server	MTX-Tunnel mode
MTX_model: 199801406	MTX terminal modem
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_radioBand: Europe	If modem is installed in Europe
MTX_portAux: gateway	Tunnelling between ports
MTX_port: 20010	Establish GPRS Gateway, read the energy meter
Firewall_enabled: off	Accepts incoming connections from any IP
CSD_enabled: on	Accepts GSM calls from energy operator
Telnet_enabled: on	Enabled as a secondary Gateway for PLC access
Telnet_port: 20011	TCP port receiving connections to access PLC
Telnet_bypass: on	Set to “on” to use Telnet as a second Gateway

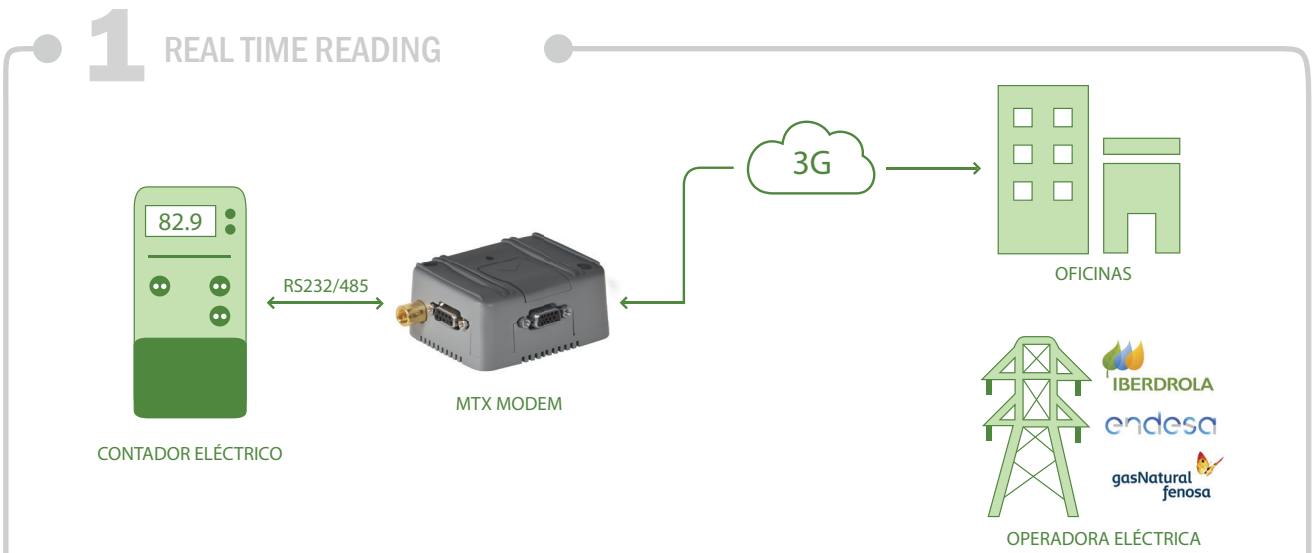
Details:

- If the secondary gateway doesn’t need to be established, you can use Telnet
- You’ll still have remote access to the modem with embedded commands (<MTXTUNNELR></MTXTUNNELR>) or SMS. Via SMS you can check the coverage and change the configuration

7.3 EXAMPLE: Example of meter reading via IP connection for real-time readings and for Energy operator. Preference for Energy operator's IP connection.

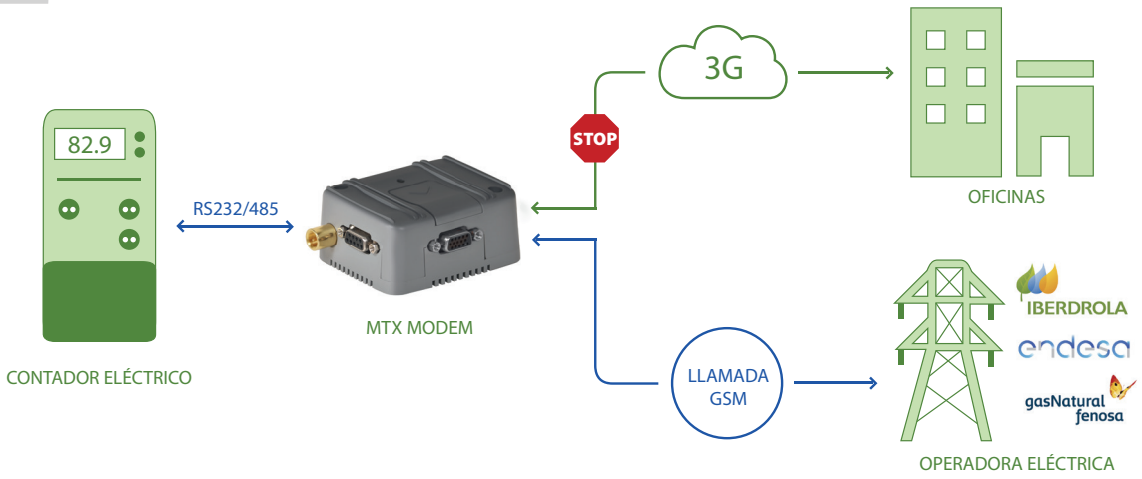
Scenario details:

- We need to read an energy meter, periodically, every 10 minutes through a 3G-RS232 gateway. We will call it "Real-Time" connection
- In parallel, an energy operator (like Endesa, Iberdrola, ...) once a day will establish a IP connection (i.e., a 3G-RS232 gateway AS WELL, NOT A GSM CALL) to carry out the daily meter reading
- Operator's IP connection should be of TOP PRIORITY. That is to say, when receiving a IP connection from the energy operator, the modem should "freeze" the 'Real Time" connection to give way to operator's readings. "Real Time" communication should be re-established after the operator's IP connection is ended



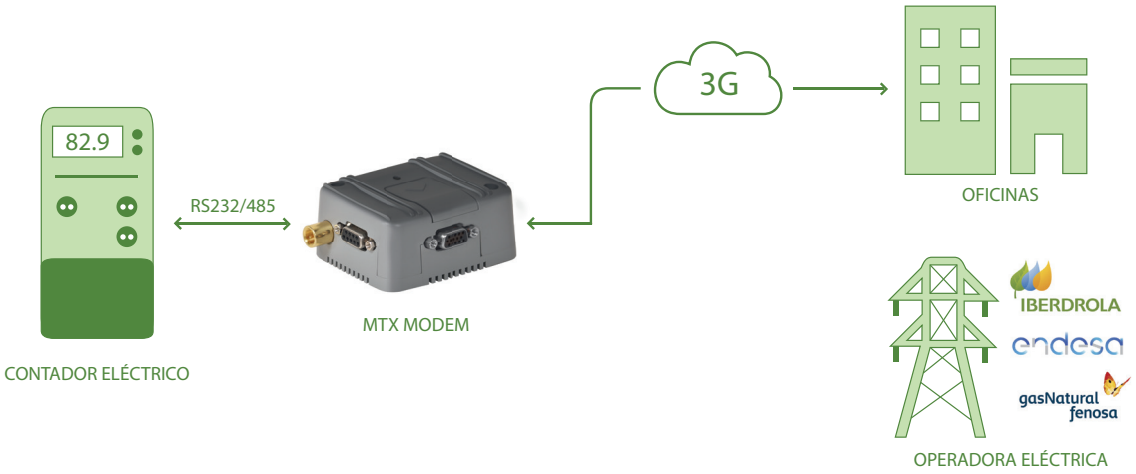
From its own offices, it is possible to carry out, in a continuous and real time, readings of the electric metering box via 3G.

2 CALL FROM THE ELECTRIC COMPANY



When daily the electrical operator (Endesa, Iberdrola...) makes a GSM call to the modem to make a metering box reading, 3G communications established with the counter “freeze” to give way to the operator’s call, which has priority.

3 BACK TO REAL TIME READING



Once the operator’s GSM call is finished, real-time 3G communications are reset.

Config.txt configuration file:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: internetestatico.movistar.es	GPRS APN
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_mode: server	MTX server mode
MTX_model: 199801422	MTX terminal model
MTX_ping: 35	Ping every 35 minutes without comms
MTX_pingIP: 8.8.8.8	IP address to ping
MTX_port: 20010	TCP port for real time reading
MTX_portb: 20011	ETCP port for Energy Operator
Firewall_enabled: off	Accept incoming connections from any IP

Details:

- RS232 port of MTX-T [3-N] modem is used to establish a 3G-RS232 transparent gateway
- 2 simultaneous 3G-RS232 gateways are created. 1 in TCP20010 port, other in TCP 20011 port
- Both gateways can't run simultaneously. When the operator establishes connects with 20011, communications with 20010 are suspended. Then, only the operator has access to the meter
- When the operator connection ends in 20011, communications with 20010 are resumed

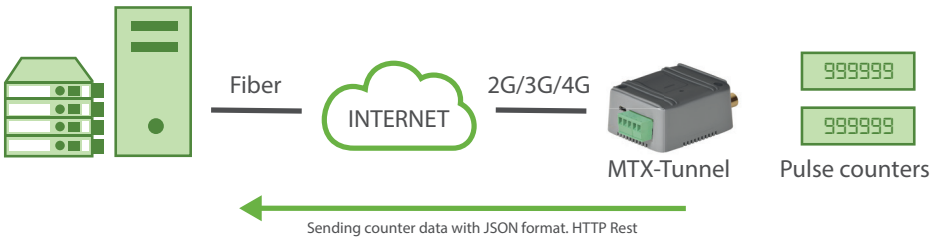
7.4 EXAMPLE: Example of reading of several pulse counters (dry contact or reed). Sending data to web platform.

Scenario details:

- Every water meters in a multiple-site installation has a pulse output. At every site there are 2 counters in need of a modem capable of carrying out pulse counting of each counter
- Every hour the modem must send counter data to a WEB platform to be dealt with

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



EXAMPLE of configuration (config.txt file) for the indicated scenario:

MTX_pin: 0000	SIM Card PIN
MTX_mode: none	We don't need 3G-RS232 gateways
MTX_model: 199801436	Modem model
MTX_ping: 30	Keep alive every 30 minutes
MTX_pingIP: 8.8.8.8	IP address for ping
MTX_ATLimited: off	No limits for user AT Commands
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	TimeServer 1 (time synch)
MTX_TPServer2: es.pool.ntp.org	TimeServer 2 (backup)
GPRS_apn: movistar.es	SIM card APN

GPRS_login: MOVISTAR	SIM card Username
GPRS_password: MOVISTAR	SIM card Password
GPRS_timeout: 0	2G/3G permanent connection
SMS_allPhones: on	All phones are allowed
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_ATResponse: on	SMS AT responses activated
FIREWALL_enabled: off	No firewall
TELNET_enabled: on	Telnet enabled
TELNET_login: user	Telnet username
TELNET_password: 1234	Telnet password
LOGGER_enabled: on	Datalogger enabled
LOGGER_password: ID12345678	User field
LOGGER_server: www.metering.es/json/set.asp?data=	URL for sending data
LOGGER_registerSize: 600	Register size
LOGGER_numRegistersFlash: 500	Number of registers in internal datalogger
LOGGER_ioPeriod: 3600	We want to read counters every 3600s (1h)
LOGGER_httpmode: getjson	HTTP GET JSON sensing mode

Details:

- How does MTX-Tunnel send counter data to the Web Server?

  Tunnel sends JSON data via HTTP(S) GET. In the previous example data will be sent to the URL:

  http://www.metering.es/json/dataset.php?data=

- What is JSON data format like?

  Tunnel will send data from all inputs and counters due to LOGGER_ioPeriod being >0

  {“IMEI”:357042060414951,”TYPE”:”IOS”,”TS”:”30/07/2016 13:14:36”,”P”:”ID12345678”,”IP”:”80.23.1.3”,”CSQ”:10,”VER”:”9.12”, “AUX”:””,”MOD”:”101”,”IO1”:0,”IO2”:0,”IO3”:0,”IO4”:0,”IO5”:0,”IO6”:0, “IO7”:0,”IO8”:0,”IO9”:0,”IO10”:0,”AD1”:0,”AD2”:0,”CO1”:”1023”, “CO2”:”18425”,”CO3”:”0”}

  Where:

  “IMEI”:357042060414951: IMEI of the modem. Unique for each modem.

  “TYPE”:”IOS”: JSON type. In this case: IOS

  “TS”:30/07/2016 13:14:36”: Time stamp of when data was collected from the modem

  “P”:”ID12345678”: User field specified in LOGGER_password

  “IP”:”80.23.1.3”: Current IP of the modem

  “CSQ”:10: Signal level. Between 0 ... 31

  “VER”:”9.12”: MTX-Tunnel version

  “IO1” ... “IO10”: I/O value

  “AD1”:0: Analog input 1 value

  “AD2”:0: Analog input 2 value

  “CO1”: “1023”: Counter 1 value

  “CO2”: “18425”: Counter 2 value

  “CO3”: “0”: Counter 3 value

- If the modem is periodically reset, will the counters reset continuously?

  No. Counters reset when lacking power supply. They aren’t stored in flash memory because it can shorten its life. When receiving data on your server always keep in mind that if you receive a value inferior to the previos reading, that means that there was a power supply failure. Therefore you should add it to the previous reading information stored.

- What is the max. value the counters can count, and what happens when they reach the max.?

  4 bytes. When surpassing they’re set to 0. If 1 pulse/sec is generated continuously.

- Is it possible to read counter data at a certain time?

  Yes, via Telnet or SMS by means of AT^MTXTUNNEL=GETCOUNTER,x command (x=counter).

- Is it possible to initialize the counters?

  Yes, via AT^MTXTUNNEL=SETCOUNTER,x,value command (x=counter, value=value to write in).

- In which pins of MTX-IoT [4-S-N-N]-STD-N modem pulse generator cable is connected?

  PIN4 or PIN11 of DB15 (GND PIN 14 of DB15). Use PIN4/14 for Counter 1, PIN11/14 for 2.

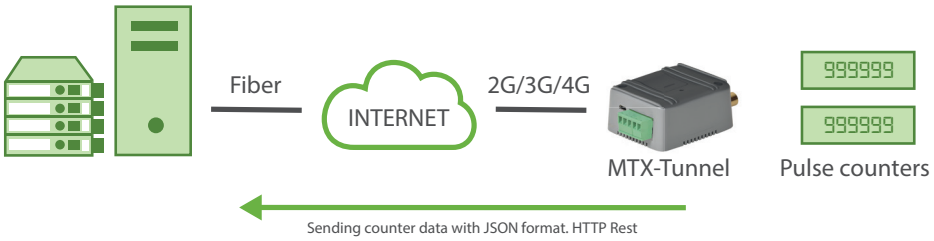
7.5 EXAMPLE: Example of pulse meter reading (dry contact or reed) with sending data to the Web Platform.

Scenario details:

- A park of water meters, in multiple sites, that have 1 pulse output (dry contact or reed) with which the water flow can be read. There are 2 meters at each site and a modem is required at each location capable of counting the pulses of each counter.
- Every hour the modem must send the meter data to a platform for processing

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+MTX-Tunnel firmware



EXAMPLE of configuration (config.txt file) for the indicated scenario:

MTX_pin: 0000	SIM Card PIN
MTX_mode: none	We don't need 3G-RS232 gateways
MTX_model: 199801436	Modem model
MTX_ping: 30	Keep alive every 30 minutes
MTX_pingIP: 8.8.8.8	IP address for ping
MTX_ATLimited: off	No limits for user AT Commands
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	TimeServer 1 (time synch)
MTX_TPServer2: es.pool.ntp.org	TimeServer 2 (backup)
GPRS_apn: movistar.es	SIM card APN

GPRS_login: MOVISTAR	SIM card Username
GPRS_password: MOVISTAR	SIM card Password
GPRS_timeout: 0	2G/3G permanent connection
SMS_allPhones: on	All phones are allowed
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_ATResponse: on	SMS AT responses activated
FIREWALL_enabled: off	No firewall
TELNET_enabled: on	Telnet enabled
TELNET_login: user	Telnet username
TELNET_password: 1234	Telnet password
LOGGER_enabled: on	Datalogger enabled
LOGGER_password: ID12345678	User field
LOGGER_server: www.metering.es/json/set.asp?data=	URL for sending data
LOGGER_registerSize: 600	Register size
LOGGER_numRegistersFlash: 500	Number of registers in internal datalogger
LOGGER_ioPeriod: 3600	We want to read counters every 3600s (1h)
LOGGER_httpmode: getjson	HTTP GET JSON sensing mode

Details:

- How does the MTX-Tunnel send the count data to the Web Server?

  The MTX-Tunnel always sends the data in JSON format, via HTTP GET or HTTPS GET. In the case

of the previous example, it would send the data to the URL:

http://www.metering.es/json/dataset.php?data=

- How is the JSON format of the data?

  For the case of the example, the MTX-Tunnel will send the data of all the possible digital and analog inputs and counters because the LOGGER_ioPeriod option is configured with a value> 0. An example of data sent is:

  {“IMEI”:357042060414951,”TYPE”:”IOS”,”TS”:”30/07/2016 13:14:36”,”P”:”ID12345678”,”IP”:”80.23.1.3”,”CSQ”:10,”VER”:”9.12”, “AUX”:””,”MOD”:”101”,”IO1”:0,”IO2”:0,”IO3”:0,”IO4”:0,”IO5”:0,”IO6”:0, “IO7”:0,”IO8”:0,”IO9”:0,”IO10”:0,”AD1”:0,”AD2”:0,”CO1”:”1023”, “CO2”:”18425”,”CO3”:”0”}

  Where:

  “IMEI”: 357042060414951 IMEI of the modem. Unique for each one.

  “TYPE”: “IOS” Type of JSON sent. In this example IOS type

  “TS”: 07/30/2016 13:14:36 “Time stamp of when the data was collected in the modem

  “P”: “ID12345678” User field defined in LOGGER_password

  “IP”: “80.23.1.3” Current IP of the modem

  “CSQ”: 10 Signal level. Between 0... 31

  “VER”: “9.12” MTX-Tunnel version

  “IO1”... “IO10” Value of the I / O

  “AD1”: 0 Value of analog input 1

  “AD2”: 0 Value of analog input 2

  “CO1”: “1023” Counter value 1

  “CO2”: “18425” Counter value 2

  “CO3”: “0” Counter value 3

- If a periodic reset is performed on the modem, do the counters continually initialize?

  No. The value of the counters will not be lost when the equipment is reset or if it is reset. Although the counters will be reset in case of power failure in the modem. The counters are NOT stored in flash memory, since doing so continuously would shorten its life very quickly. On your server, when you receive the data, you always have to bear in mind that if you receive a value lower than the last reading, there has been a supply failure, so you must perform a summation with the previously saved value.

- What is the maximum value that the counters can count and what happens when it reaches the maximum?

-

- The maximum value of the counters is 4294967295 (4 bytes). After that limit, they are initialized to 0. In any case (as an example) if 1 pulse per second is generated continuously

over time, this limit would be reached after 136 years. Enough time for neither you nor me to affect us.

- Is it possible to read the counter values at a specific moment?

  Yes, via TELNET or via SMS using the command AT ^ MTXTUNNEL = GETCOUNTER, x (where X = 0 or 1 depending on the counter to be read)

- Is it possible to initialize the counters?

  Yes, through the command AT ^ MTXTUNNEL = SETCOUNTER, x, value (where “x” = 0 or 1 depending on the counter you want to write and “value” the value to write)

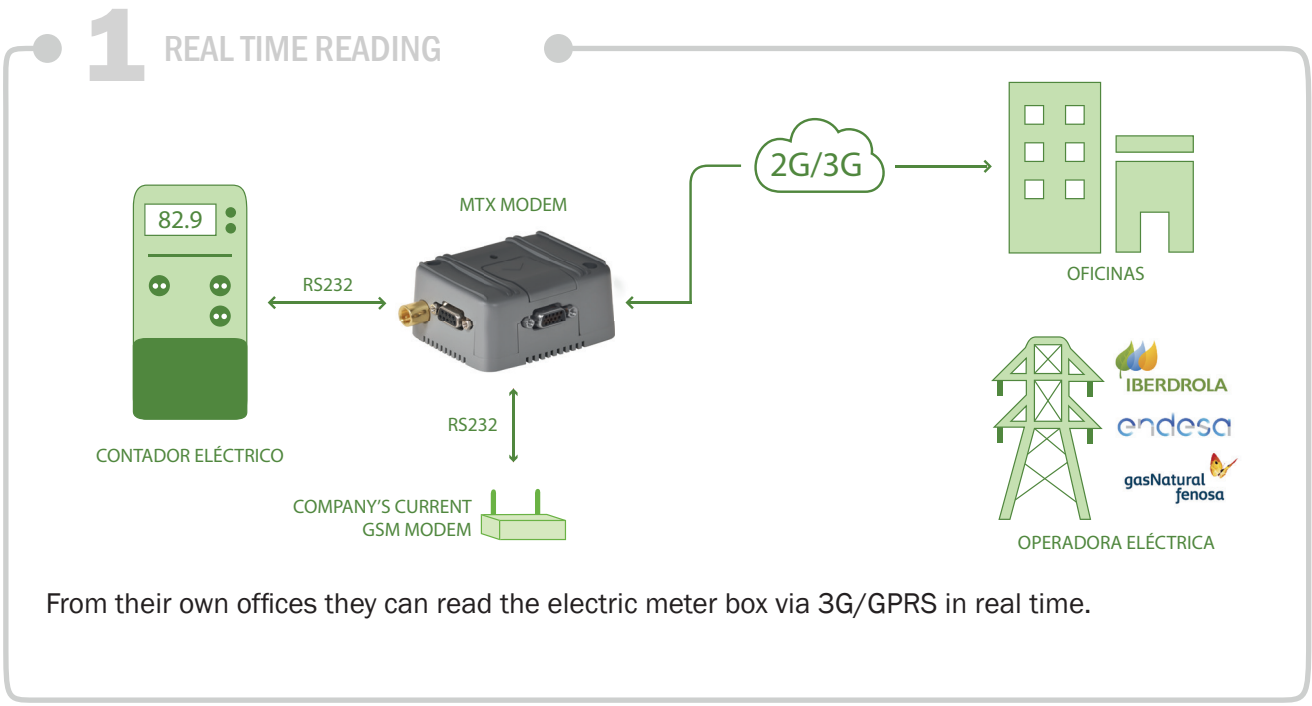
- To which pins of the MTX-IoT [4-S-N-N] -STD-N modem is the meter’s pulse generator cable connected?

  The modem will count pulses each time the modem’s DB15 connector PIN4 or PIN11 is brought to GND (DB15 pin 14). Therefore you must use PIN4 and PIN14 for Counter 1 and PIN11 and PIN14 for Counter 2. You will find the complete table of DB15 connector connections in the Annexes at the end of the manual

7.6 EXAMPLE: Reading meter boxes via 3G in scenarios with a modem that can't be disconnected (the current modem must be kept).

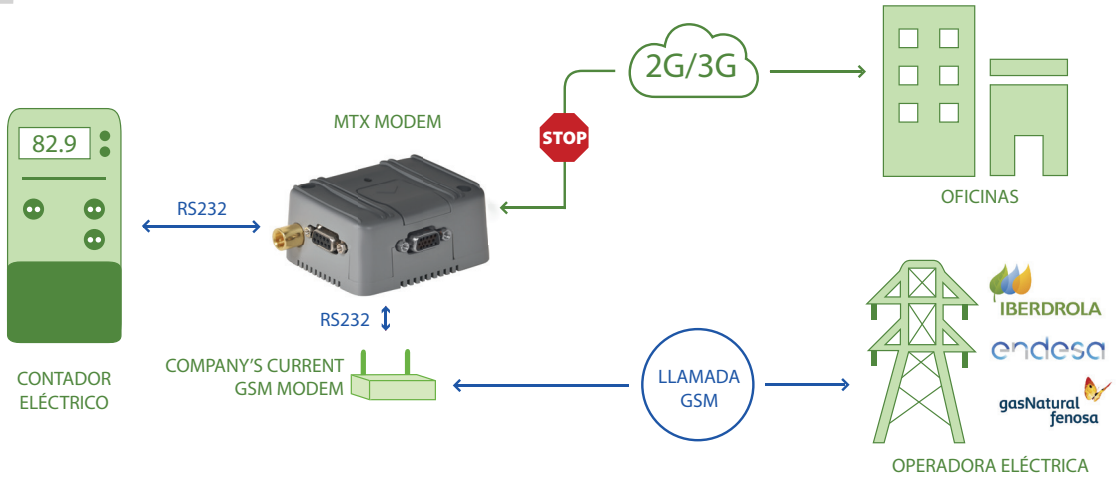
Scenario details:

- We want to read an electric meter box every 10 minutes. Since the frequency is high, we'll access the meter box via GPRS/3G instead of with GSM calls to reduce costs
- Simultaneously, the electric company will make a daily conventional GSM data pone call to read the meter box
- In this scenario there's a company modem already installed, which can't be disconnected, since the company makes the daily call to their modem SIM. The solution is to install an MTX-T2 [3-N] modem between the meter box which needs to be read and the company modem



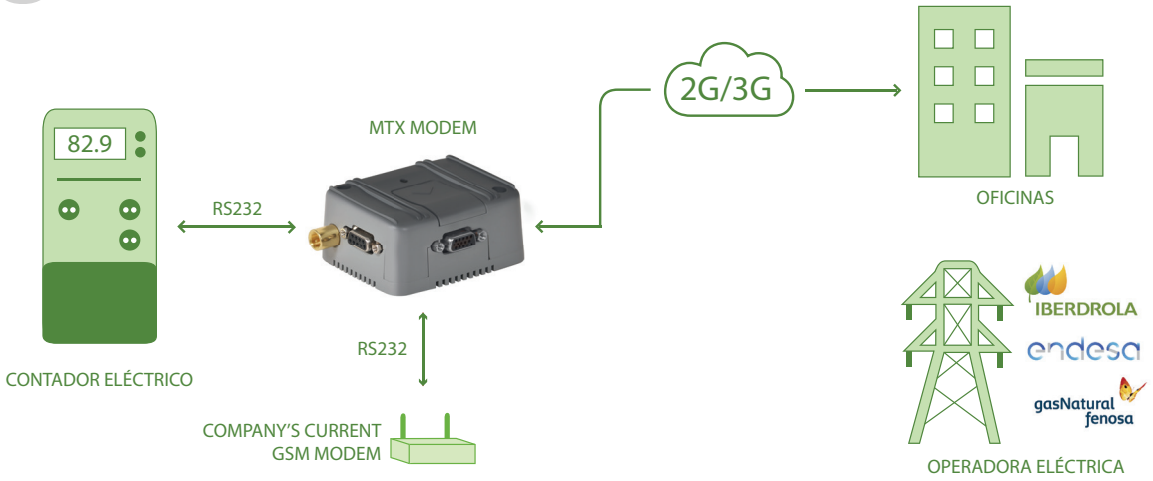
From their own offices they can read the electric meter box via 3G/GPRS in real time.

2 CALL FROM THE ELECTRIC COMPANY



The electric company calls daily to their own modem. The call is redirected to the MTX modem which stops the 3G/GPRS communication to make a bypass between serial ports (making itself invisible).

3 BACK TO REAL TIME READING



When the GSM call ends (when the MTX doesn't detect data coming through the serial port for 30 seconds) IP communications in real time are reactivated.

Configuration example (config.txt file) for said scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM2_baudrate: 9600	Company's modem serial port speed
COMM2_bitsperchar: 8	Number of data bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit configured
COMM2_parity: none	No parity bit
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_mode: server	GPRS connection server type
MTX_model: 199801406	Modem model
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_portAux: bypass	Gateway between serials with preference

TCP_port: 20010	Establish the gateway between ports
SMS_allPhones: on	All phones are allowed
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_ATResponse: on	SMS AT responses activated
FIREWALL_enabled: off	No firewall
TELNET_enabled: on	Telnet enabled
TELNET_login: user	Telnet username
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet TCP port

Details:

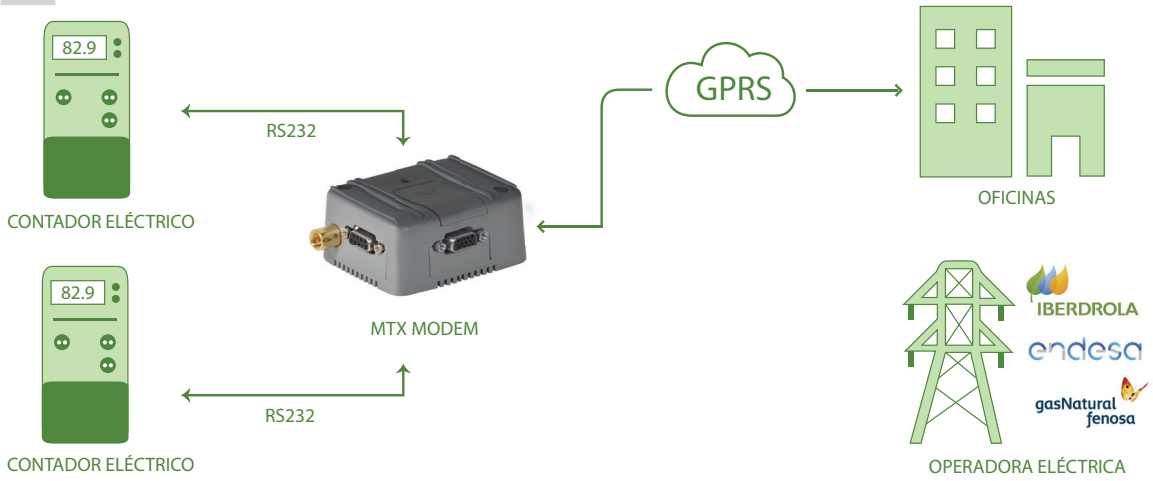
- It's necessary to connect the meter box to the MTX COM1 connector (which is the DB9 connector that's closer to the MTX modem GSM antenna) and the GSM modem to the MTX COM2 connector
- Remember you may need a crossover cable between the MTX modem and the company's GSM modem

7.7 EXAMPLE: Basic example for reading 2 meters with 1 modem (and a single SIM), combining GPRS connections and GSM calls.

Scenario details:

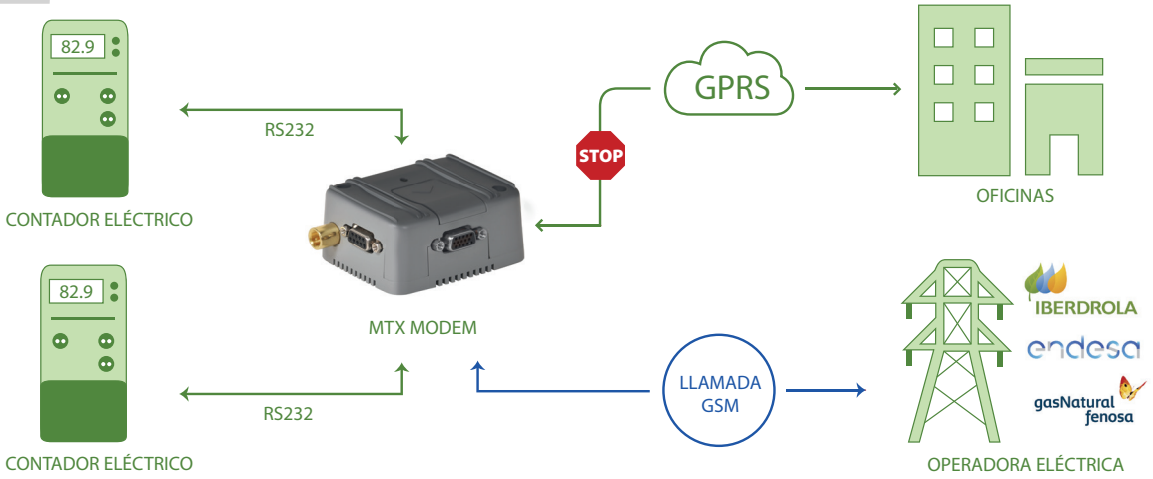
- It is intended to read 2 electricity meters periodically every 15 minutes. The two meters are located in the same place, so it is desired to use a single modem, with 2 serial ports, in order to lower costs (saving a modem and an additional SIM).
- Being such a frequent reading period (15 minutes), the meter will be accessed via GPRS instead of GSM calls, also with the aim of reducing costs.
- At the same time, the energy operator (Endesa, Iberdrola,...) will make a conventional GSM data call once a day to read the meters.
- The GSM call must be a priority. When the modem receives a call from the power operator, it must “freeze” the GPRS connections to make way for the operator’s readings. Once the GSM call is finished, the GPRS connections must be reestablished

1



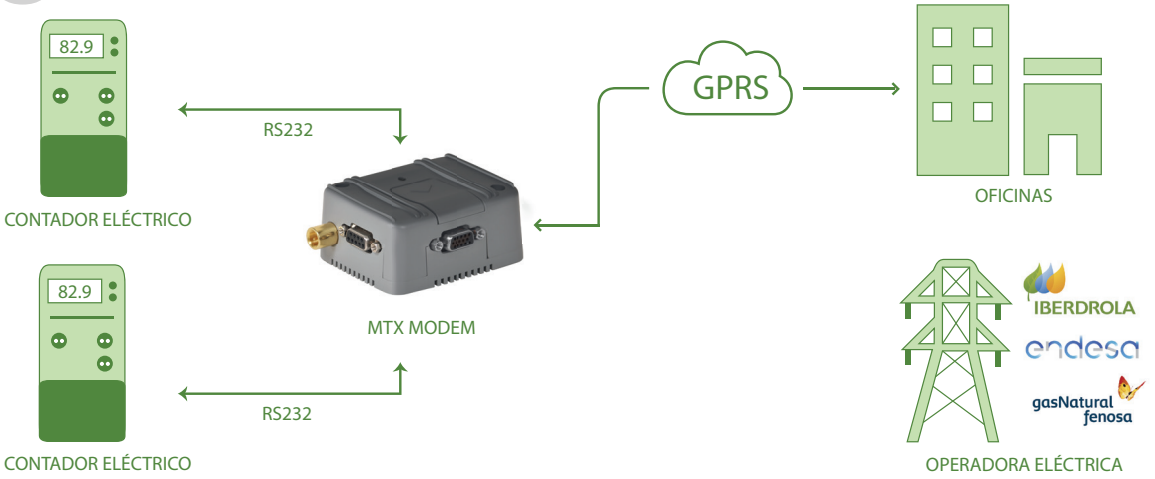
Electric meter readings can be carried out continuously and in real time from their own offices via GPRS (each one through a TCP port).

2



When the electricity operator (Endesa, Iberdrola ...) makes a GSM call to the modem to read the meters on a daily basis, the meter's GPRS communications "freeze" to make way for the operator's call, which has priority.

3



Once the operator's GSM call is finished, real-time GPRS communications are restored.

Configuration example (config.txt file) for said scenario:

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
COMM2_baudrate: 9600	Company's modem serial port speed
COMM2_bitsperchar: 8	Number of data bits
COMM2_autorts: off	No flow control
COMM2_autocts: off	No flow control
COMM2_stopbits: 1	1 stop bit configured
COMM2_parity: none	No parity bit
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPRS_mode: 2g	Modem is always GPRS connected
MTX_mode: server	GPRS connection server type
MTX_model: 199801406	Modem model
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping

TCP_port: 20010	Establish the gateway between ports
TCP_port2: 20011	Establish the gateway between ports
Firewall_enabled: off	Firewall status
CSD_enabled: on	Enables reception of GSM data calls
CSD_commPort: 3	GSM call must be used via both serial ports

Details:

- Remember that if you are using a modem with 3G technology to receive GSM calls, it is essential to configure it to work with 2G technology. Otherwise the GSM calls will not enter the modem. This is done with the parameter: GPRS_mode: 2g
- For the GSM call to be received by the 2 serial ports simultaneously, it is necessary to configure the modem with the CSD_commPort parameter: 3
- The example shows the possibility of managing 2 meters with RS232 port. The most suitable modem model for this is the MTX-T2 [3-N], as it is the most economical. If you want to control 2 meters, but one RS232 and the other RS485 it is also possible to do so, but you must opt for the MTX-IoT [3-S-N-N] modem model

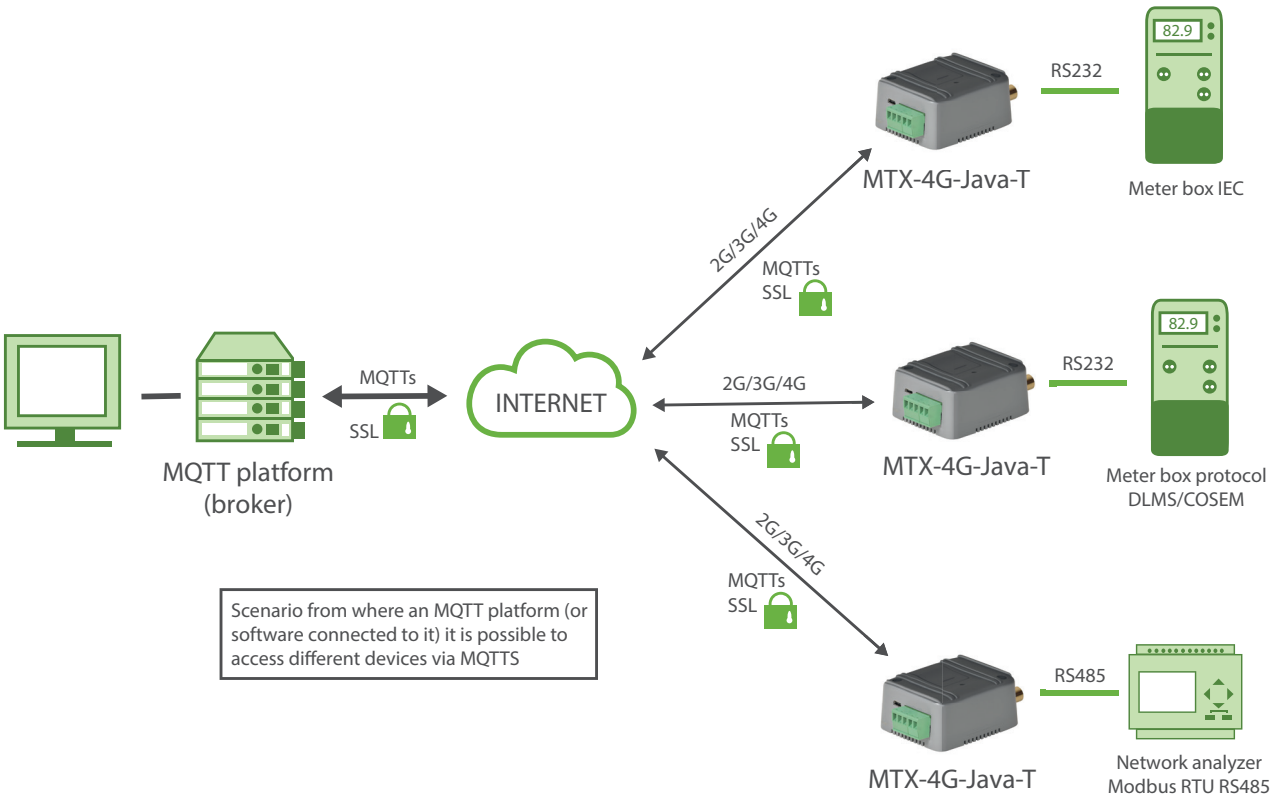
7.8 EXAMPLE:. Reading IEC counters, DLMS/Cosem, Modbus with protocols implemented in an MQTT platform.

Scenario details:

- We want to read a large number of different meter boxes (IEC counters, counters with DLMS/ Cosem protocol, modbus protocol network analyzers, etc.) All devices have an RS232 or RS485 serial port
- In order to make the system cheaper, economic devices with 4G/3G/2G connectivity will be used, such as the MTX-T [4-N] modem, whose model has an RS232 and RS485 bus, making it suitable for all cases. The SIM cards used may be from any operator, so it is not guaranteed that the IPs are public or fixed. Therefore, connectivity must always be established from the counter to the platform
- The communication between platform and devices will be via MQTTS, which guarantees the connectivity required in the previous section and adds a layer of SSL security in the communications between platform and counter
- The modems will act as “Transparent gateways Series - MQTT,” therefore the interpretation of the protocols (ie. the IEC protocol intelligence, DLMS/Cosem, modbus...) will be on the platform. This allows that in case of adding new features it is not necessary to remotely update the firmware of the entire modem pool, with the risk it involves, but only to update the software in one place, that of the MQTT platform
- Likewise, in addition to being able to read from the platform the different meters, a “Device Manager” must be implemented, in order to be able to constantly visualize the status of the modems (status, coverage...) as well as to be able to change their configuration anytime
- The modems must also have communication enabled by SMS, allowing that, from several authorized telephone numbers, the status can be checked, changing configurations (for example, an incorrect APN that does not allow communication with the MQTT platform), executing remote reset or any other operation
- The modem must have the firmware (MTX-Tunnel) signed and locked. That is, it can’t be manipulated or changed physically. In addition, the configuration of the modem will be encrypted and for greater security, the memory blocked. It can only be unlocked, if necessary, from the MQTT platform or from an authorized telephone number
- The modem must have watchdogs that allow it to recover from problematic connectivity situations. Although it will not be used because it is not necessary, the modem must be ready to activate, from the MQTT platform or via SMS, a daily authorset every 24h

Solution:

MTX-T [4-N] modem+MTX-Tunnel firmware



Config.txt configuration file (master):

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	8 bit data
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN from your network operator
GPRS_login: MOVISTAR	GPRS Login

GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	MTX-Tunnel is always GPRS connected
MTX_PIN: 0000	If SIM card has no PIN security, use 0000 value
MTX_mode: mqtt	MQTT serial gateways will be used
MTX_model: 199801445	MTX-Terminal modem model used
MTX_ping: 35	One ping every 35 min. without communications
MTX_pingIP: 8.8.8.8	Ping address
MTX_invertedCom: off	Gateway port RS485
MTX_msToSend: 100	No fragmented networks
MTX_ATLimited: off	No limitations
SMS_allPhones: off	Send SMS with commands from any phone
SMS_sendIP: off	Modem won't respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
SMS_validPhone1: +34666123456	Authorized phone number 1
SMS_validPhone2: +34666123457	Authorized phone number 2
FIREWALL_enabled: off	No firewall
MQTT_enabled: on	MQTT service enabled
MQTT_server: ssl://test.mosquitto.org:8883	Broker IP/DNS specified, including identifying port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands

MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
MQTT_commrxtopic: rxmaster	Data received will be retransmitted via serial
MQTT_commrxtopic: txmaster	Data received v/serial, retransmitted to this topic
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: [IMEI]/dns	Topic where status data are sent
DNS_extended: off	Extended data (E/S, ADCs...) are not sent
DNS_period: 300	One sending every 300 secs (5 mins.)

Details:

- The config.txt file shown corresponds to the modems connected to RS232 devices. If you need to use the RS485 port, you only need to change the MTX_invertedCom parameter to “on”
- Likewise, you must adjust the values of the COMM_ parameters, adjusting them to the configuration of the serial port of the connected device
- If you use SSL communications and need to incorporate the certificates of your broker, at the end of this document you will find an Annex with the procedure
- Remember that with this example the MQTT platform directly access the serial port of the device and it must be this one the one to interpret the protocol, because the modem treats the information transparently. In addition, the platform will monitor the status of modems (modems send their status every X minutes) and can also use the MQTT_ATTopic1 topics to send AT commands to the modem from the platform at any time (to make configuration changes, resets, etc.). Remember you can set up to 3 topics of AT commands for each device. If you put the same topic in all modems you can, executing a single command on your MQTT platform, execute an action (AT command) on all modems
- With this configuration, all the modems subscribe to the topic [IMEI]/rx. This causes all the data frames sent by the platform to this topic to be received by the corresponding modem and retransmitted, as is, by its RS232 or RS485 serial port. In the same way, all the data frames received by the RS232 or RS485 serial port of the modems are automatically retransmitted via MQTT to the topic [IMEI]/tx. Remember that it is not necessary to replace [IMEI] with the corresponding IMEI, the modem does it automatically
- Keep in mind that communication latencies can be somewhat greater than with direct communication (since there is an intermediary, the mqtt broker, and the speed of communications will depend on the power of the latter). Set the timeout if necessary

7.9 EXAMPLE: Meter reading via GSM call and/or IP communications (without SSL/TLS security). Incorporation of Device Manager (with SSL/TLS security) for modem management.

Scenario details:

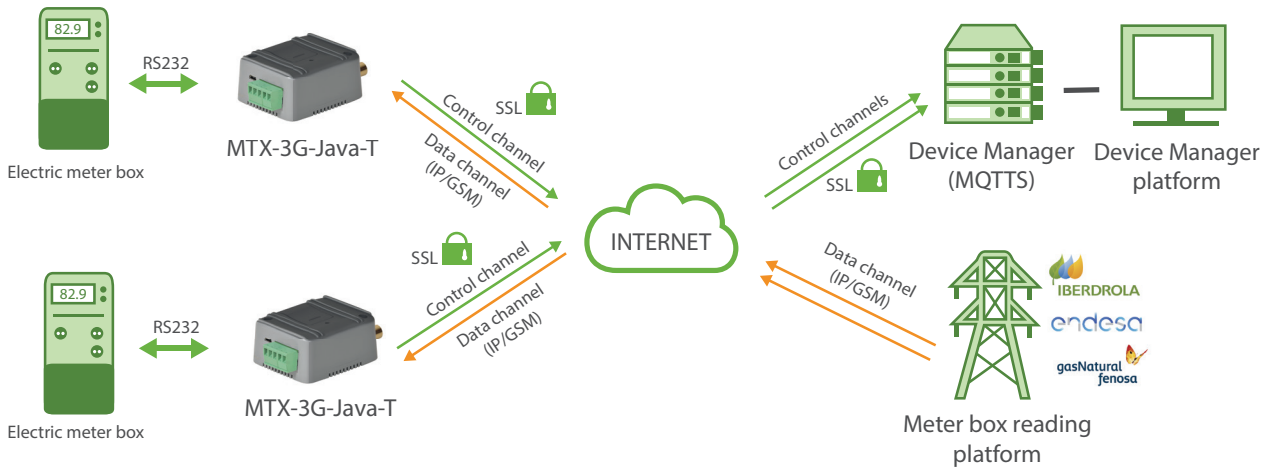
- A large pack of electric meter boxes is available. For 15 years until today the readings of the contractors are carried out through GSM data calls (CSD) made to the modems connected to the electric meter boxes through the RS232 port. This project aims to make an evolution of this system, going from a GSM communications system (CSD) to one of IP communications (3G/2G). In addition, it is intended to add a remote modem management system (Device Manager) that allows remote monitoring of them, as well as being able to make changes to remote configurations, firmware update, certificate management, etc.
- It is necessary to add high security to the system. Thus:

  1. The modems will be listening on TCP port 20010 (TCP Server mode) to make a transparent IP-RS232 gateway, but only accept connections from authorized IP addresses

  2. Communications between modems and Device Manager must be secure. Connections will be established from the modem (TCP Client) and must be SSL/TLS. MQTTS protocol will be used

  3. Modems should only accept CSD calls that are made from authorized telephone numbers, that is, from the telephone numbers of the Accountant Reading Center

Solution:



Config.txt configuration file (master):

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPRS_mode: 2g	GPRS connection server type
MTX_PIN: 0000	
MTX_mode: server	GPRS connection server type
MTX_model: 199801422	Modem model
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_ATLimited: off	No AT commands limitations
SMS_allPhones: off	Send SMS with commands from any phone
SMS_sendIP: off	Modem won't respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS

SMS_validPhone1: +34666123456	Authorized phone number 1
SMS_validPhone2: +34666123457	Authorized phone number 2
FIREWALL_enabled: on	Authorized IP will be able to connect to modem
FIREWALL_IP1: 80.1.2.3	IP address authorized 1
FIREWALL_IP2: 80.4.5.6	IP address authorized 2
TCP_port: 20010	Establish the gateway between ports
MQTT_enabled: on	MQTT service enabled
MQTT_server: ssl://broker.cervello.io:8883	Broker IP/DNS specified, including identifying port
MQTT_id: yku41420t957oh8t	Identifier
MQTT_login: jfj1usly8ijhh9hizfr453	Username
MQTT_password: gthhdte67y3ttes33fgg	Password
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: [IMEI]/dns	Topic where status data are sent
DNS_extended: off	Extended data (E/S, ADCs...) are not sent
DNS_period: 600	One sending every 600 secs (5 mins.)
CSD_enabled: on	CSD calls are enabled
CSD_allPhones: off	Only telephone numbers admitted for CSD

CSD_validPhone1: 666333444	Authorized phone number 1
CSD_validPhone2: 666444555	Authorized phone number 2

Details:

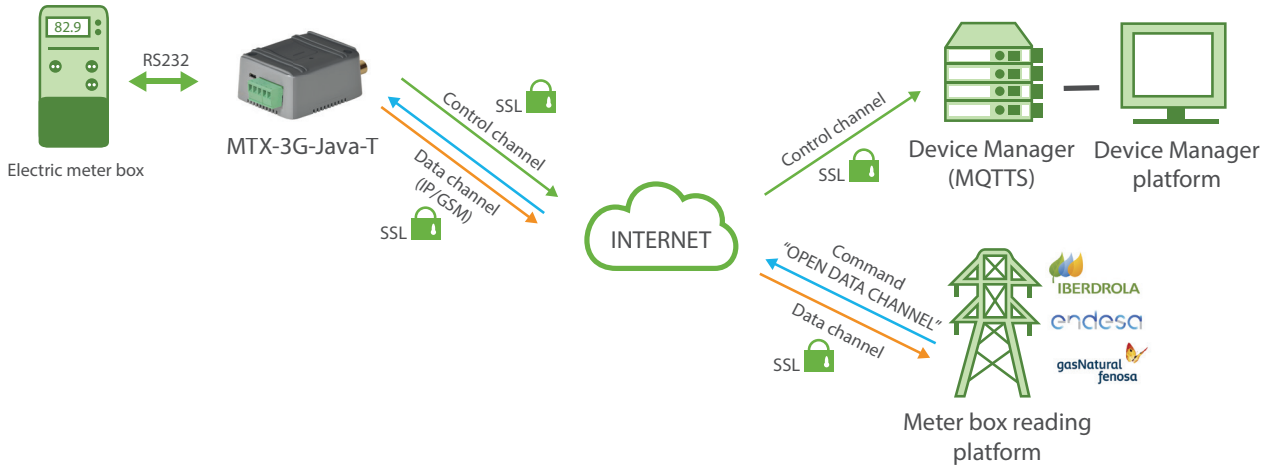
- In this scenario, it is possible to connect to the meter box by making a TCP/IP connection (on TCP port 20010) from the reading platform to the modem, but the connection via GSM data call (CSD) is also allowed. That is why in this scenario the modem must be forced to work using the 2G network (parameter GPRS_mode: 2g). In case of not needing GSM data calls, the GPRS_mode parameter can be changed to an “auto” value, that way the modem would use the 3G/2G network according to availability
- We must adjust the values of the COMM_ parameters to match the configuration of the serial port of the meter box to which the modem is connected
- In this example, the Cervello broker has been used as Device Manager, which is a platform fully compatible with MTX modems. Consult the Cervello documentation for operational monitoring of modem status, remote configuration changes, etc.
- If we use SSL communications against the Device Manager and need to incorporate the root certificates of your broker, at the end of this document there’s an Annex with the procedure

7.10 EXAMPLE: Meter reading via GSM data call (CSD) and IP communications (with SSL/TLS security). Incorporation of Device Manager (with SSL/TLS security) for modem management.

Scenario details:

- A large pack of electric meter boxes is available. For 15 years the readings of the contractors are carried out through GSM Data Calls (CSD) made to the modems which are connected to the electric meter boxes through the RS232 port. This project aims to make an evolution of this system, going from a GSM communications system (CSD) to one of IP communications. In addition, it is intended to add a remote modem management system (Device Manager) that allows remote monitoring of modems, as well as being able to make changes to remote configurations, firmware update, certificate management, etc.
- It is necessary that the new system has high security. Therefore, both the IP data channel of the meter reading (communication between the modem and the Reading Center) and the modem monitoring control channel (that is, the communication between the modem and the Device Manager) must be performed through an IP communication with SSL/TLS security and mutual authentication
- Currently, the reading infrastructure of the electric meter boxes is carried out by means of a GSM data call (CSD). Therefore, new modems must also be compatible with this type of CSD calls until the infrastructure migrates to IP communications. Taking advantage of the modem replacement, it is intended to add an additional security layer to the current GSM data calls (CSD), and that modems should only accept CSD calls that are made from authorized telephone numbers, that is, from the numbers Telephone Counter Reading Center

Solution:



Config.txt configuration file (master):

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPRS_mode: 2g	GPRS connection server type
MTX_PIN: 0000	Pin of the SIM
MTX_mode: server	GPRS connection server type
MTX_model: 199801422	Modem model
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_ATLimited: off	No AT commands limitations
MTX_ATEmbedded: temporalclient	Embedded command only for temporal TCP socket
MTX_temporalClientTimeout: 120	Socket closes if no traffic during x seconds
MTX_IDClient: [IMEI]	Identifying chain
MTX_clientSSL: on	TCP client socket

SMS_allPhones: off	Send SMS with commands from any phone
SMS_sendIP: off	Modem won't respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
SMS_validPhone1: +34666123456	Authorized phone number 1
SMS_validPhone2: +34666123457	Authorized phone number 2
FIREWALL_enabled: on	Authorized IP will be able to connect to modem
FIREWALL_IP1: 80.1.2.3	IP address authorized 1
FIREWALL_IP2: 80.4.5.6	IP address authorized 2
TCP_port: 20010	Establish the gateway between ports
TCP_IP: 80.1.2.3	Reading platform IP address
MQTT_enabled: on	MQTT service enabled
MQTT_server: ssl://broker.cervello.io:8883	Broker IP/DNS specified, including identifying port
MQTT_id: yku41420t957oh8t	Identifier
MQTT_login: jfj1usly8ijhh9hizfr453	Username
MQTT_password: gthhdte67y3ttes33fgg	Password
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: [IMEI]/dns	Topic where status data are sent

DNS_extended: off	Extended data (E/S, ADCs...) are not sent
DNS_period: 600	One sending every 600 secs (5 mins.)
CSD_enabled: on	CSD calls are enabled
CSD_validPhone1: 666333444	Authorized phone number 1
CSD_validPhone2: 666444555	Authorized phone number 2

Details:

- The IP connection between the modem and the Device Manager is permanent and secure (SSL/TLS). It is always set and at any time it is possible to send an action command (configuration change, reset, etc.) from the Device Manager to the modem
- The connection between the modem and the reading platform is not permanent. The procedure by the reading platform to start reading an accountant safely is as follows:

  1. The reading platform connects to TCP port 20010 of the modem

  2. The reading platform sends the command:

  <MTXTUNNELR> AT ^ MTXTUNNEL = DEFAULTTEMPORALCLIENT </MTXTUNNELR>

  to the modem through said socket so that it opens a secure channel (SSL/TLS) of communications in TCP mode Client against the platform.

  3. The modem opens an SSL/TLS socket against the “TCP_port” port of the platform IP. Mutual authentication is performed between the reading platform and modem through digital certificates

  4. Once the secure socket (SSL/TLS) is established, the modem sends the value of its MTX_IDClient configuration parameter through this socket, an alphanumeric string that allows the reading platform to identify the modem that has been connected (in case you want to read several counters simultaneously)

  This socket will remain established until it is closed on the side of the reading platform or “MTX_temporalClientTimeout” seconds pass without traffic on it.

  5. The platform can start the meter reading through said socket as a transparent IP-RS232 gateway with SSL/TLS security

  6. Once the meter is read, the reading platform will close the reading socket and the modem will be ready to accept the command through the TCP_port port. To start a new meter reading, the process indicated in 1 is restarted
- You must adjust the values of the COMM_ parameters by adjusting them to the serial port configuration of the connected device
- For SSL communications, if you need to incorporate the root certificates of your servers, at the end of this document you will find an annex with the procedure

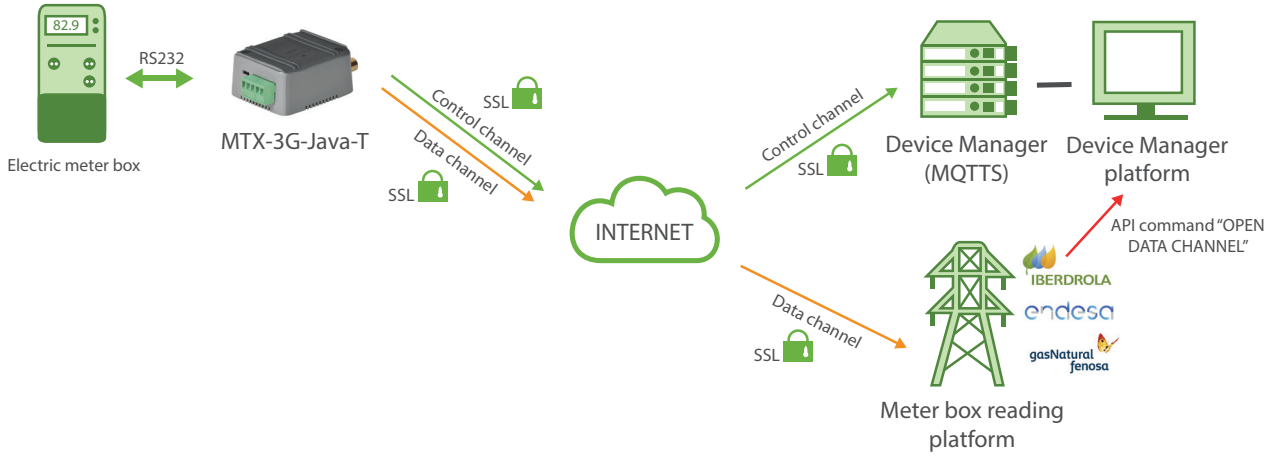
- The modem will only accept connections to the TCP_port port from the IP addresses indicated in the FIREWALL_IP1 and FIREWALL_IP2 parameters
- In this scenario, in addition to IP communications, it is allowed to connect to the meter box via GSM data call (CSD). That is why in this scenario the modem must be forced to use the 2G network (parameter GPRS_mode: 2g). If you do not need a GSM data call, you can change the GPRS_mode configuration parameter to an “auto” value, that way the modem will use the 3G/2G network according to availability

7.11 EXAMPLE: Reading of counters per CSD call and IP communications with SSL/TLS. Incorporation of Device Manager for modem management and connection between modems and the meter box reading platform.

Scenario details:

- A large pack of electric meter boxes is available. Until today the readings of the contractors are made through GSM Data Calls (CSD) made to the modems which are connected to the electric meter boxes through the RS232 port. It is intended to move from GSM (CSD) to IP, and add a remote modem management system (Device Manager) that allows remote monitoring of modems, as well as being able to make changes to remote configurations, firmware update, certificate management, etc.
- High security is required. Both the IP data channel of the meter reading (communication between the modem and the reading center) and the modem monitoring control channel (that is, the communication between the modem and the Device Manager) must be carried out through a communication IP with SSL/TLS security and mutual authentication. To further increase security, the modem must NOT have any TCP port on listen, that is, all TCP/IP connections must be made from the modem. Therefore, every time the reading platform needs to read a counter, it must execute an API command against the Plataform Device Manager, indicating that it wants to read a certain counter. The Device Manager will send a command to the modem (through the command channel already established with SSL/TLS security) to open another secure data channel against the reading platform
- Currently, the reading infrastructure of the electric meter boxes is carried out by means of a GSM data call (CSD). Therefore, new modems must also be compatible with this type of CSD calls until the infrastructure migrates to IP communications. Taking advantage of the modem replacement, it is intended to add an additional security layer to the current GSM data calls (CSD), and that modems should only accept CSD calls that are made from authorized telephone numbers, that is, from the numbers Telephone Counter Reading Center

Solution:



Config.txt configuration file (master):

COMM_baudrate: 9600	Serial port baud rate
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPRS_mode: 2g	GPRS connection server type
MTX_PIN: 0000	Pin of the SIM
MTX_mode: none	GPRS connection server type
MTX_model: 199801422	Modem model
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_ATLimited: off	No AT commands limitations
MTX_IDClient: [IMEI]	Identifying chain
MTX_temporalClientTimeout: 120	Socket closes if no traffic during x seconds
MTX_clientSSL: on	TCP client socket
SMS_allPhones: off	Send SMS with commands from any phone

SMS_sendIP: off	Modem won't respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
SMS_validPhone1: +34666123456	Authorized phone number 1
SMS_validPhone2: +34666123457	Authorized phone number 2
MQTT_enabled: on	MQTT service enabled
MQTT_server: ssl://broker.cervello.io:8883	Broker IP/DNS specified, including identifying port
MQTT_id: yku41420t957oh8t	Identifier
MQTT_login: jfj1usly8ijhh9hizfr453	Username
MQTT_password: gthhdte67y3ttes33fgg	Password
MQTT_attopic1: [IMEI]/AT	MQTT topic to send AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic to send replies to commands to
MQTT_keepalive: 300	Connection keep alive (300 seconds)
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: [IMEI]/dns	Topic where status data are sent
DNS_extended: off	Extended data (E/S, ADCs...) are not sent
DNS_period: 600	One sending every 600 secs (5 mins.)
CSD_enabled: on	CSD calls are enabled
CSD_validPhone1: 666333444	Authorized phone number 1
CSD_validPhone2: 666444555	Authorized phone number 2

Details:

- The IP connection between the modem and the Device Manager is permanent and secure (SSL/TLS). It is always set and at any time it is possible to send an action command (configuration change, reset, etc.) from the Device Manager to the modem
- The connection between the modem and the reading platform is not permanent. The procedure by the reading platform to start reading an accountant safely is as follows:

  1. The reading platform executes an API (webservice) command against the Device Manager Platform, indicating the modem to be read. (See documentation of the API of the Cervello platform)

  2. The Device Manager Platform sends a command to the modem (through its always established data channel, with SSL/TLS security) so that it opens another secure channel (SSL/TLS) of communications in TCP mode client against the platform of meter box reading. In this case, the IP command and connection TCP port against the platform will be indicated in the API command. The command that the Device Manager platform sends to the modem is:

  AT ^ MTXTUNNEL = TEMPORALCLIENT, IP, TCPPORT, TIMEOUT

  Where the IP address TCP port of the meter box reading platform, respectively, is specified in the IP and TCPPORT parameter. In TIMEOUT you specify the time (seconds) after which the socket must be closed in case there is no traffic through it.

  3. The modem opens an SSL/TLS socket against the TCP and IP port that indicates the command received from the Device Manager Platform. Mutual authentication is performed between the reading platform and modem through digital certificates

  4. Once the secure socket (SSL/TLS) is established, the modem sends the value of its MTX_IDClient configuration parameter through this socket, an alphanumeric string that allows the reading platform to identify the modem that has been connected (in case you want to read several counters simultaneously)

  This socket will remain established until it is closed on the side of the reading platform or "MTX_temporalClientTimeout" seconds pass without traffic on it.

  5. The platform can start the meter reading through said socket as a transparent IP-RS232 gateway with SSL/TLS security

  6. Once the meter is read, the reading platform will close the socket. To start a new meter reading, the process indicated in 1 is restarted

- You must adjust the values of the COMM_ parameters by adjusting them to the serial port configuration of the connected device
- For SSL communications, if you need to incorporate the root certificates of your servers, at the end of this document you will find an Annex with the procedure
- The security in this communication model is very high, since there's no TCP port open (on listening) in the modem, all communications are of the client TCP type, secured by SSL/TLS
- In this scenario, in addition to IP communications, it is allowed to connect to the Counter via GSM data call (CSD). That is why in this scenario the modem must be forced to use the 2G network (parameter GPRS_mode: 2g). If you do not need a GSM data call, you can change the GPRS_mode configuration parameter to an "auto" value, that way the modem will use the 3G/2G network according to availability

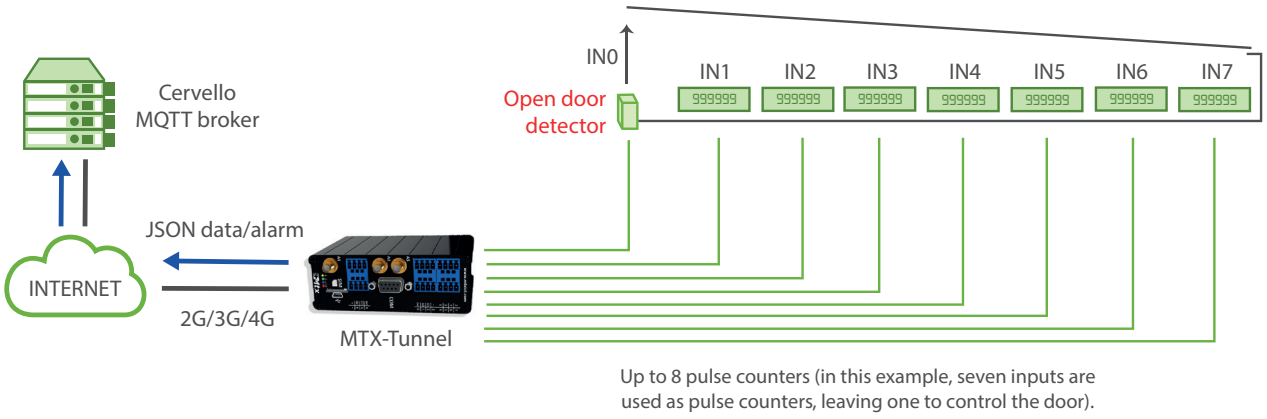
7.12 EXAMPLE: Monitoring of 7 meters with pulse outputs via MQTT with alarm to detect the opening of the meter access door.

Scenario details:

- 7 counters with pulse output need to be monitored. Therefore, a modem with 7 pulse counter inputs is required. The accumulated pulses must be sent to a web platform via MQTT every 60 minutes
- The meters will be installed in manholes and register boxes which are locked. It is necessary to monitor the status of the door and, if it is opened, a message with the status of the door must be sent to the Web platform via MQTT. In this way, from the control post they will have proof that said location should be reviewed

Solution:

Modem MTX-IOT-S [4-N] + firmware MTX-Tunnel



Up to 8 pulse counters (in this example, seven inputs are used as pulse counters, leaving one to control the door).

Configuration file config.txt (master):

GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	Pin of the SIM
MTX_model: MTX-IOT-4G-S	MTX model
MTX_mode: none	GPRS connection server type
MTX_ping: 30	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_numGSMErrors: 180	Reset if it doesn't register in 1800 secs. in GSM
MTX_TPServer: es.pool.ntp.org	Time server NTP1
MTX_TPServer2: 2.europe.pool.ntp.org	Time server NTP2
MTX_TPProtocol: ntp	NTP protocol
SMS_allPhones: on	Send SMS with commands from any phone
SMS_allPhones: off	Send SMS with commands from any phone
SMS_sendIP: on	Modem will respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
MQTT_enabled: on	MQTT service enabled
MQTT_server: ssl://broker.cervello.io:8883	Broker IP/DNS specified, including identifying port
MQTT_id: psdjs334jjsd8345	Identifier

MQTT_login: 3ddg435g67899	Username
MQTT_password: 2345433456567	Password
MQTT_attopic1: /cervello/devices/[MQTT_ID]/rpc	MQTT topic to send AT commands
MQTT_atrtopic: /cervello/devices/[MQTT_ID]/rpc/response	Topic to send replies to commands to
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Connection keep alive (60 seconds)
MQTT_defaultIOQos: 0	Qos for the topic defaultIOTopic
MQTT_defaultIOTopic: IOCHANGE	MQTT topic to send quick telemetries
GPIO_mode0: input	GPIO0 configured as input
GPIO_config0: mqtt;2;0	GPIO0 MQTT configuration. Gate status
GPIO_mode1: input	GPIO1 configured as input
GPIO_config1: counter	GPIO1 MQTT configuration as pulse counter
GPIO_mode2: input	GPIO2 configured as input
GPIO_config2: counter	GPIO2 MQTT configuration as pulse counter
GPIO_mode3: input	GPIO3 configured as input
GPIO_config3: counter	GPIO3 MQTT configuration as pulse counter
GPIO_mode4: input	GPIO4 configured as input
GPIO_config4: counter	GPIO4 MQTT configuration as pulse counter
GPIO_mode5: input	GPIO5 configured as input
GPIO_config5: counter	GPIO5 MQTT configuration as pulse counter
GPIO_mode6: input	GPIO6 configured as input

GPIO_config6: counter	GPIO6 MQTT configuration as pulse counter
GPIO_mode7: input	GPIO7 configured as input
GPIO_config7: counter	GPIO7 MQTT configuration as pulse counter
LOGGER_enabled: on	Logger on to store readings
LOGGER_registerSize: 600	Internal registry size
LOGGER_numRegistersFlash: 1500	Max. number of registries
LOGGER_mode: mqtt	Sending mode via MQTT
LOGGER_mqttTopic: /LOGGER	Sending to MQTT broker topic
LOGGER_ioPeriod: 3600	Period of data sendig
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: /DNS	Topic where modem status data will be sent
DNS_mqttTopic: /DNS	Period of data sendig

Detalles:

- The configuration of the inputs as “mqtt; 2; 0; 1; / DINPUT” indicates the following. Remember that all parameters are separated by semicolons ;

  “mqtt” > The input is configured to send the states of the digital inputs by MQTT

  “2” > The 2 indicates that the digital input is configured to send an MQTT message both for activation of the input (when it closes taking it to ground) and for deactivation of the input (when it opens). In case you want to send an MQTT message only when closing the entry (bringing it to ground), you should indicate a value of “1”

  “0” > Indicates the timeout of the digital input. This means that the change of the digital input will be sent whenever it occurs. If, for example, a value of “10” was configured, as occurs with GPIO4 and GPIO5, even if multiple activations occur in the digital input, more than 1 MQTT message will never be sent in those 10 seconds

  “1” > The QoS of the MQTT message indicates the message, which can have the value 0.1.2

  “/DINPUT” > Indicates the topic to send the MQTT message to

- The configuration of the inputs as “counter” indicates the following:

  “counter” > The input is configured as a pulse counter

- The sending format of the GPIO0 input messages (the input that controls the opening of the door) follows the JSON structure, shown in the following example:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”DINPUT”,

  “DATA”:

  {

  “GPIO”:0,

  “VALUE”:1

  }

  }

  Where:

  - IMEI: indicates the IMEI of the MTX modem

  - TYPE: indicates the type of frame. DINPUT = Digital Input

  - DATA: contains a JSON with the event data

  - GPIO: indicates the GPIO index (0... 7)

  - VALUE: indicates the value of the input (0,1)

- The sending format of the IOS messages, where the data of the counters are included, is as follows:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”IOS”,

  “TS”:”19/01/20 07:16:08”,

  “IO0”:0,

  “IO1”:0,

  “IO2”:0,

  “IO3”:0,

  “IO4”:0,

  “IO5”:0,

“IO6”:0,
“IO7”:0,
“AD0”:0,
“AD1”:0,
“CO0”:”10005”,
“CO1”:”11005”,
“CO2”:”14303”,
“CO3”:”16001”,
“CO4”:”14425”,
“CO5”:”11901”,
“CO6”:”11124”,
“CO7”:”15373”
}

Where:

- IMEI: indicates the IMEI of the MTX modem
- TYPE: indicates the type of frame. IOS
- TS: timeStamp
- IOx: 0.1 (state of the digital input, where x = 0, … 7)
- COx: counters accounts, where x = 0 … 7)

7.13 EXAMPLE: Monitoring of 1 IEC870-5-102 meter via RS232 in real time (every 5 minutes) with data sent to the HTTP platform.

Scenario details:

- An energy efficiency company needs to monitor in real time (every 15 minutes) the data collected in the following table:

Active absolute energy (VabA)	Power Factor Phase 2 (thousandths)
Inductive Reactive Absolute Energy (VabRi)	Active Power Phase 3 (KW)
Capacitive Reactive Absolute Energy (VabRc)	Reactive Power Phase 3 (KVA)
Total Active Power (KW)	Power Factor Phase 3 (thousandths)
Total Reactive Power (KVA)	Current Phase 1 (tenths of an amp)
Total Power Factor (thousandths)	Voltage Phase 1 (tenths of a volt)
Active Power Phase 1 (KW)	Phase 2 current (tenths of an amp)
Reactive Power Phase 1 (KVA)	Phase 2 Voltage (tenths of a volt)
Power Factor Phase 1 (thousandths)	Current Phase 3 (tenths of an amp)
Active Power Phase 2 (KW)	Voltage Phase 3 (tenths of a volt)
Reactive Power Phase 2 (KVA)	

  Said data will be collected from a meter with IEC 870-5-102 protocol with RS232 port and the data will be sent to an HTTP platform.

- The meter is in a location where coverage is not very good and connectivity is sometimes lost, so the modem must store the readings in its internal memory to send them when coverage returns. The modem must support 4G/3G/2G technologies

Solution:

Modem MTX-T + firmware MTX-Tunnel



Configuration file config.txt (master):

COMM2_baudrate: 9600	Speed of serial port
COMM2_bitsperchar: 8	Data bit
COMM2_autorts: off	No flux control
COMM2_autocts: off	No flux control
COMM2_stopbits: 1	Stop bits
COMM2_parity: none	Bit parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	Pin of the SIM
MTX_model: 199801445	MTX model
MTX_mode: none	GPRS connection server type
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping

MTX_TPProtocol: ntp	NTP protocol
MTX_TPServer: 1.europe.pool.ntp.org	Time server
MTX_TPServer: 2.europe.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Time format
MTX_porAux: iec102	Auxiliar port has function IEC102
MTX_invertedCom: on	Inverted port on
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: off	Modem won't respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
IEC102_meter1: id001;1715;1;7	Identifier id001, address 1715, pt. 1, password 7
IEC102_period: 15	Period to read counter
LOGGER_enabled: on	Logger enabled or disabled
LOGGER_password: ID-12345678	User field
LOGGER_server: www.miweb.es/set.asp?data=	Web platform url
LOGGER_registerSize: 600	Register size
LOGGER_numRegistersFlash: 200	Number of registers in flash
LOGGER_mode: http	Sending mode
LOGGER_httpMode: getjson	Mode used

Details:

- The data is sent to the HTTP server using a JSON object, which has the following example format:

{“IMEI”:”353085090011697”,”TYPE”:”IEC102”,”TS”:”2020-05-15T18:00:05Z”,”P”:”ID-12345678”,”ID”:”ID001”,”VABA”:4000,”VABRI”:41,”VABRC”:1328,”PAT”:5,”PRT”:0,”FPT”:990,”PAF1”:2,”PRF1”:0,”FPF1”:970,”PAF2”:2,”PRF2”:0,”FPF2”:970,”PAF3”:0,”PRF3”:0,”FPF3”:760,”IF1”:94,”TF1”:2392,”IF2”:107,”TF2”:2367,”IF3”:37,”TF3”:2427}

Where:

- IMEI: IMEI of the modem (unique identifier of the MTX modem)
- TYPE: Indicates the type of frame. In this case IEC102
- TS: Timestamp of when the data was read
- P: User field
- ID: User identifier indicated in IEC102_meter1
- VBAB: Active absolute energy (VabA)
- VINA: Absolute Reactive Inductive Energy (VabRi)
- VABRI: Capacitive Reactive Absolute Energy (VabRc)
- PAT: Total Active Power (KW)
- PRT: Total Reactive Power (KVA)
- FPT: Total Power Factor (thousandths)
- PAF1: Active Power Phase 1 (KW)
- PRF1: Phase 1 Reactive Power (KVA)
- FPF1: Phase 1 Power Factor (thousandths)
- PAF2: Phase 2 Active Power (KW)
- PRF2: Reactive Power Phase 2 (KVA)
- FPF2: Phase 2 Power Factor (thousandths)
- PAF3: Phase 3 Active Power (KW)
- PRF3: Reactive Power Phase 3 (KVA)
- FPF3: Phase 3 Power Factor (thousandths)
- IF1: Phase 1 current (tenths of an ampere)
- TF1: Phase 1 Voltage (tenths of a volt)
- IF2: Phase 2 current (tenths of an ampere)
- TF2: Phase 2 Voltage (tenths of a volt)
- IF3: Phase 3 current (tenths of an ampere)
- TF3: Phase 3 Voltage (tenths of a volt)

- Remember that the configuration data of the RS232 serial port of the modem must match the configuration of the RS232 port of the meter
- If you need to monitor other meter parameters in addition to VabA, VabRi, VabRc, check the possibility at iotsupport@mtxm2m.com
- The IEC102_ configuration parameters are only available in the “MTX-Tunnel IEC870”, it is a product that is supplied separately

7.14 EXAMPLE: Monitoring of 16 IEC870-5-102 counters via RS485 in real time (every 5 minutes) with data sent to the MQTT platform.

Scenario details:

- An energy efficiency company needs to monitor in real time (every 15 minutes) the data collected in the following table:

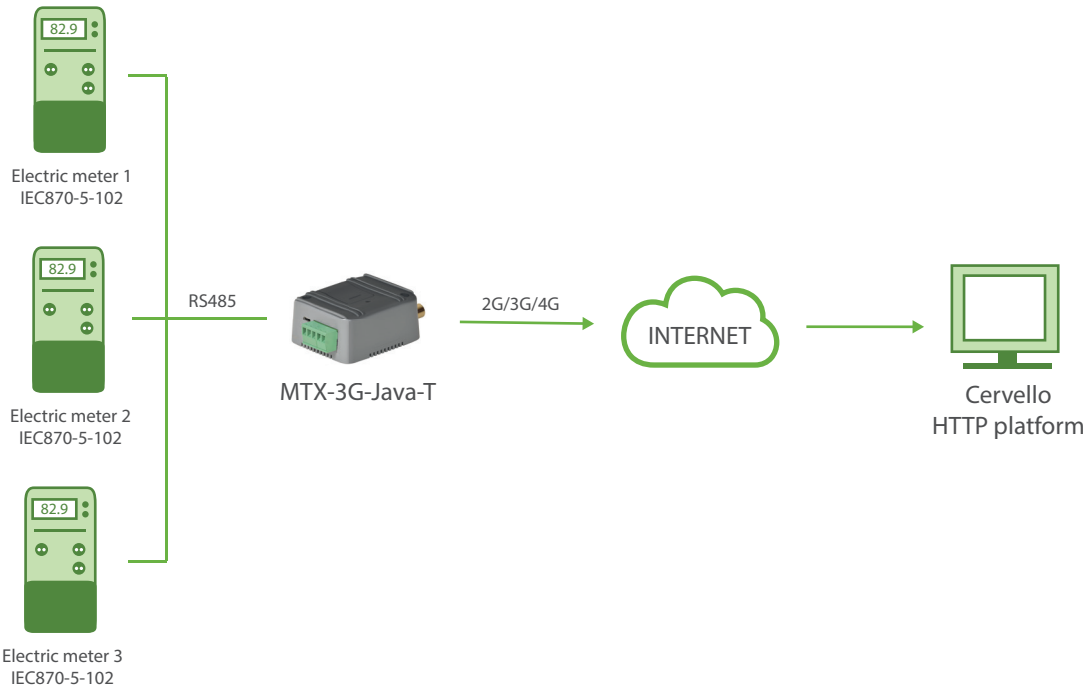
Active absolute energy (VabA)	Power Factor Phase 2 (thousandths)
Inductive Reactive Absolute Energy (VabRi)	Active Power Phase 3 (KW)
Capacitive Reactive Absolute Energy (VabRc)	Reactive Power Phase 3 (KVA)
Total Active Power (KW)	Power Factor Phase 3 (thousandths)
Total Reactive Power (KVA)	Current Phase 1 (tenths of an amp)
Total Power Factor (thousandths)	Voltage Phase 1 (tenths of a volt)
Active Power Phase 1 (KW)	Phase 2 current (tenths of an amp)
Reactive Power Phase 1 (KVA)	Phase 2 Voltage (tenths of a volt)
Power Factor Phase 1 (thousandths)	Current Phase 3 (tenths of an amp)
Active Power Phase 2 (KW)	Voltage Phase 3 (tenths of a volt)
Reactive Power Phase 2 (KVA)	

  Said data will be collected from 16 counters with IEC 870-5-102 protocol with RS485 port and send the data to an MQTT platform

- The meters are in a location where coverage is not very good and connectivity is sometimes lost, so the modem must store the readings in its internal memory to send them when coverage returns. The modem must support 4G/3G/2G technologies

Solution:

Modem MTX-T + firmware MTX-Tunnel



Configuration file config.txt (master):

COMM2_baudrate: 9600	Speed of serial port
COMM2_bitsperchar: 8	Data bit
COMM2_autorts: off	No flux control
COMM2_autocts: off	No flux control
COMM2_stopbits: 1	Stop bits
COMM2_parity: none	Bit parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password

GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	Pin of the SIM
MTX_model: 199801445	MTX model
MTX_mode: none	GPRS connection server type
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_TPProtocol: ntp	NTP protocol
MTX_TPServer: 1.europe.pool.ntp.org	Time server
MTX_TPServer: 2.europe.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Time format
MTX_porAux: iec102	Auxiliar port has function IEC102
MTX_invertedCom: off	Inverted port off
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: off	Modem won't respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
IEC102_meter1: id001;1;2;3	Identifier id001, address 1, pt. , password 3
IEC102_meter2: id002;2;2;3	Identifier id002, address 2, pt. , password 3
IEC102_meter3: id003;3;2;3	Identifier id003, address 3, pt. , password 3
IEC102_meter4: id004;4;2;3	Identifier id004, address 4, pt. , password 3
IEC102_meter5: id005;5;2;3	Identifier id005, address 5, pt. , password 3
IEC102_meter6: id006;6;2;3	Identifier id006, address 6, pt. , password 3

IEC102_meter7: id007;7;2;3	Identifier id007, address 7, pt. , password 3
IEC102_meter8: id008;8;2;3	Identifier id008, address 8, pt. , password 3
IEC102_meter9: id009;9;2;3	Identifier id009, address 9, pt. , password 3
IEC102_meter10: id010;10;2;3	Identifier id0010, address 10, pt. , password 3
IEC102_meter11: id011;11;2;3	Identifier id0011, address 11, pt. , password 3
IEC102_meter12: id012;12;2;3	Identifier id0012, address 12, pt. , password 3
IEC102_meter13: id013;13;2;3	Identifier id0013, address 13, pt. , password 3
IEC102_meter14: id014;14;2;3	Identifier id0014, address 14, pt. , password 3
IEC102_meter15: id015;15;2;3	Identifier id0015, address 15, pt. , password 3
IEC102_meter16: id016;16;2;3	MTX responds with an SMS to a command SMS
IEC102_period: 15	Period to read counter
LOGGER_enabled: on	Logger enabled or disabled
LOGGER_registerSize: 600	Register size
LOGGER_numRegistersFlash: 200	Number of registers in flash
LOGGER_mode: mqtt	Sending mode
LOGGER_mqttTopic: /LOGGER	Data sending MQTT topic

Details:

- The data is sent to the HTTP server using a JSON object, which has the following example format:
{“IMEI”:”353085090011697”,”TYPE”:”IEC102”,”TS”:”2020-05-15T18:00:05Z”,”P”:”ID-12345678”,”ID”:”ID001”,”VABA”:4000,”VABRI”:41,”VABRC”:1328,”PAT”:5,-”PRT”:0,”FPT”:990,”PAF1”:2,”PRF1”:0,”FPF1”:970,”PAF2”:2,”PRF2”:0,”FPF2”:970,”PAF3”:0,”PRF3”:0,”FPF3”:760,”IF1”:94,”TF1”:2392,”IF2”:107,”TF2”:2367,”IF3”:37,”TF3”:2427}

  Where:

  - IMEI: IMEI of the modem (unique identifier of the MTX modem)

- TYPE: Indicates the type of frame. In this case IEC102
- TS: Timestamp of when the data was read
- P: User field
- ID: User identifier indicated in IEC102_meter1
- VBAB: Active absolute energy (VabA)
- VINA: Absolute Reactive Inductive Energy (VabRi)
- VABRI: Capacitive Reactive Absolute Energy (VabRc)
- PAT: Total Active Power (KW)
- PRT: Total Reactive Power (KVA)
- FPT: Total Power Factor (thousandths)
- PAF1: Active Power Phase 1 (KW)
- PRF1: Phase 1 Reactive Power (KVA)
- FPF1: Phase 1 Power Factor (thousandths)
- PAF2: Phase 2 Active Power (KW)
- PRF2: Reactive Power Phase 2 (KVA)
- FPF2: Phase 2 Power Factor (thousandths)
- PAF3: Phase 3 Active Power (KW)
- PRF3: Reactive Power Phase 3 (KVA)
- FPF3: Phase 3 Power Factor (thousandths)
- IF1: Phase 1 current (tenths of an ampere)
- TF1: Phase 1 Voltage (tenths of a volt)
- IF2: Phase 2 current (tenths of an ampere)
- TF2: Phase 2 Voltage (tenths of a volt)
- IF3: Phase 3 current (tenths of an ampere)
- TF3: Phase 3 Voltage (tenths of a volt)

- Remember that the configuration data of the RS485 serial port of the modem must match the configuration of the RS485 port of the meter
- If you need to monitor other meter parameters in addition to VabA, VabRi, VabRc, check the possibility at iotsupport@mtxm2m.com
- The IEC102_ configuration parameters are only available in the “MTX-Tunnel IEC870”, it is a product that is supplied separately
- Not all types of meters allow simultaneous reading connected to the same RS485 bus. If in doubt, consult the meter manufacturer

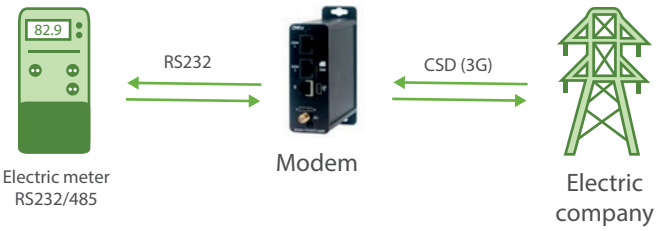
7.15 EXAMPLE: Basic example for meter reading via CSD call in environments with only 3G coverage (that is, where there is no 2G coverage).

Scenario details:

- A modem is required to read an energy meter by CSD call (GSM data call)
- The modem must be configured to communicate with the meter via RS232 serial communication, with a serial port setting at 9600.8, N, 1
- The modem must be accessible via SMS in order to read status, make configuration changes, etc. It should only be possible to send SMS commands to the modem from authorized phone numbers
- The modem is going to be located in a place with no 2G coverage. Only the area has 3G coverage, so the modem must be able to accept a data call through this technology

Solution:

MTX-T modem + MTX-Tunnel firmware



Configuration file config.txt (master):

COMM_baudrate: 9600	Energy meter serial port speed
COMM_bitsperchar: 8	Number of data bits
COMM_autorts: off	No flow control
COMM_autocts: off	No flow control
COMM_stopbits: 1	We set 1 stop bit
COMM_parity: none	No parity bit
GPRS_timeout: 1	The modem will not connect to the Internet
GPRS_mode: auto	The modem will connect to 2g or 3g
MTX_PIN: 0000	SIM card has no PIN
MTX_mode: none	There are no TCP gateways
MTX_model: 199801492	Modem Model
MTX_numGSMErrors: 180	Reset in 30 minutes if there is no network registration
SMS_allPhones: off	All numbers are NOT authorized
SMS_ATEnabled: on	SMS commands enabled
SMS_ATResponse: on	SMS command responses enabled
SMS_validPhone1: +34666123456	Authorized phone number
CSD_enabled: on	Enables the reception of GSM data calls

Details:

- The GPRS_mode parameter is configured in “auto” mode. In this way the modem will connect to the 3g network if it is available and otherwise to the 2g network. It is possible to configure the modem to register only in “2g” or “3g”, however the “auto” mode is generally more recommended.
- Remember that if you configure the modem in “auto” or “3g” mode, it is not possible to modify

this example to combine CSD calls with IP communications, since in many locations it is not supported by the telephone operator. If you need to simultaneously CSD communications with IP communications, the modem must be configured in “2g” mode as indicated in the previous examples in this chapter.

7.16 EXAMPLE: Monitoring (instantaneous values and fiscal closure) of 1 IEC870-5-102 counter via RS232 in real time (every 5 minutes) with data sent to the MQTT platform. Support for CSD calls in environments with 2G coverage.

Scenario details:

- An energy efficiency company needs to monitor in real time (every 5 minutes) the instantaneous energy and power values collected in the following table:

Active absolute energy (VabA)	Power Factor Phase 2 (thousandths)
Inductive Reactive Absolute Energy (VabRi)	Active Power Phase 3 (KW)
Capacitive Reactive Absolute Energy (VabRc)	Reactive Power Phase 3 (KVA)
Total Active Power (KW)	Power Factor Phase 3 (thousandths)
Total Reactive Power (KVA)	Current Phase 1 (tenths of an amp)
Total Power Factor (thousandths)	Voltage Phase 1 (tenths of a volt)
Active Power Phase 1 (KW)	Phase 2 current (tenths of an amp)
Reactive Power Phase 1 (KVA)	Phase 2 Voltage (tenths of a volt)
Power Factor Phase 1 (thousandths)	Current Phase 3 (tenths of an amp)
Active Power Phase 2 (KW)	Voltage Phase 3 (tenths of a volt)
Reactive Power Phase 2 (KVA)	

  Said data will be collected from 1 meter with IEC 870-5-102 protocol with RS232 port and the data will be sent to an MQTT platform

- Punctual request for Instantaneous values. It will be possible to query the instantaneous energy and power values in real time, by means of an AT command sent via MQTT. The data read will have the same format as that of the periodic submission, but will be sent to a different MQTT topic

- Reading and periodic sending of the fiscal closing. The reading and sending period will be daily, taking the current day as the final day, and the corresponding day one month ago as the initial day. Example:

CURRENT DAY	INITIAL TIMESTAMP	FINAL TIMESTAMP
04/22/2021	03/21/2021-00:00	04/22/2021-00:00

NOTE: The sending of the parameters referring to the memorized Charging information (Contract I) of the closing readings will be sent via MQTT encapsulated in a JSON object (in an array), with all the data read.

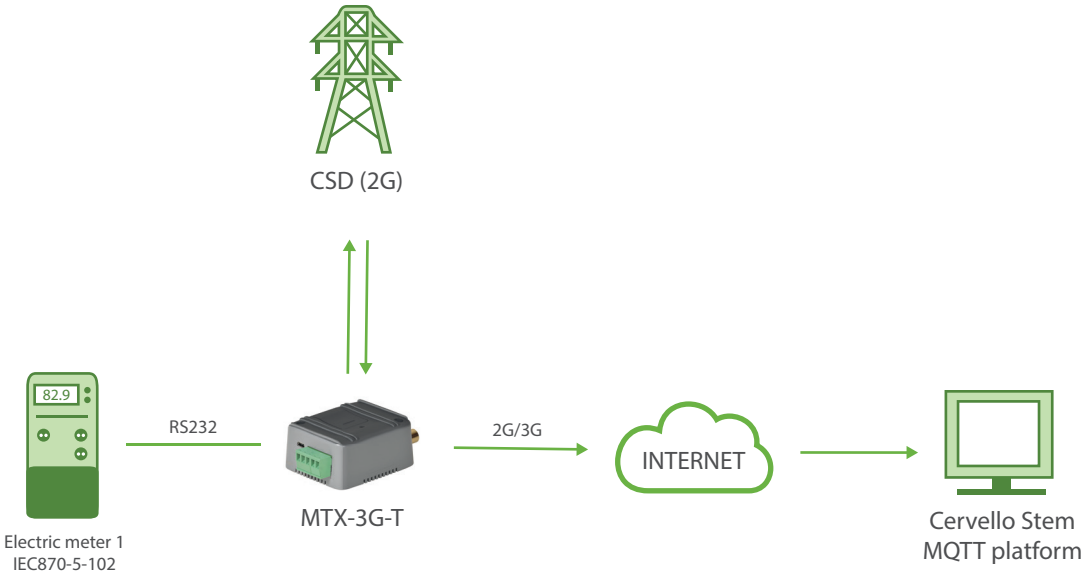
ELEMENT	TYPE	BYTES
Absolute energy A [A + / A-]	Whole without sign	4
Incremental energy A [A + / A-]	Whole without sign	4
Qualifier A	8 bit array	1
Absolute energy Ri [Ri + / Ri-]	Whole without sign	4
Incremental energy Ri [Ri + / Ri-]	Whole without sign	4
Qualifier Ri	8 bit array	1
Absolute energy Rc [Rc + / Rc-]	Whole without sign	4
Incremental energy Rc [Rc + / Rc-]	Whole without sign	4
Rc qualifier	8 bit array	1
Reserve 7	To define	4
Qualifier 7	8 bit array	1
Reserve 8	To define	4
Qualifier 8	8 bit array	1
Maximum power A [A + / A-]	Whole without sign	4
Maximum power A date	Label time a	5

Maximum qualifier A	8 bit array	1
Power excesses A [A + / A-]	Whole without sign	4
Excess qualifier	8 bit array	1
Start of period	Label time a	5
End of period	Label time a	5

- Reading and punctual sending of the fiscal closing. By means of AT command it will be possible to carry out the punctual reading and sending of the closures in real time. This command will allow you to specify the start date and the end date
- CSD call support. In case of receiving a CSD call for real-time value queries, the autonomous parameter reading processes must be suspended until the CSD call ends, at which point they will be resumed.
- The meters can be in a place where the coverage is not very good and connectivity is sometimes lost, so the modem must store the readings in its internal memory to send them when the coverage returns. The modem must support 2g / 3g technologies
- The meter has network address 2816, measurement point 1 and pass 7

Solution:

MTX-T modem + MTX-Tunnel firmware



Configuration file config.txt (master):

COMM2_baudrate: 9600	Speed of serial port
COMM2_bitsperchar: 8	Data bit
COMM2_autorts: off	No flux control
COMM2_autocts: off	No flux control
COMM2_stopbits: 1	Stop bits
COMM2_parity: none	Bit parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPRS_mode: 2g	Modem model where the MTX-Tunnel is installed
MTX_PIN: 0000	Pin of the SIM
MTX_model: 199801422	MTX model
MTX_mode: none	GPRS connection server type
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_TPProtocol: ntp	NTP protocol
MTX_TPServer: time1.google.com	Time server
MTX_TPServer: time2.google.com	Time server backup
MTX_TPFormat: unix	Time format
MTX_porAux: iec102	Auxiliar port has function IEC102

MTX_invertedCom: on	Inverted port on
SMS_allPhones: on	Send SMS with commands from any phone
SMS_sendIP: off	Modem won't respond to a missed call/SMS
SMS_ATEnabled: on	Commands can be sent to the MTX by SMS
SMS_ATResponse: on	MTX responds with an SMS to a command SMS
IEC102_meter1: id001;2816;1;7	Counter, address, measured point, pass
IEC102_period: 5	Counters are intended to be read every 5 minutes
IEC102_attempts: 3	Maximum number of meter reading retries
LOGGER_enabled: on	Logger enabled or disabled
LOGGER_registerSize: 5000	Register size
LOGGER_numRegistersFlash: 250	Number of registers in flash
LOGGER_mode: mqtt	Sending mode
LOGGER_mqttTopic: /LOGGER	Data sending MQTT topic
MQTT_enabled: on	We enable the MQTT service on the modem
MQTT_server: tcp:// broker.mqttdashboard.com:1883	We specify the IP / DNS of the broker, port incl
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: AT	Topic to subscribe to receive commands
MQTT_attopic: ATR	Topic MQTT for sending responses to AT commands
MQTT_qos: 1	Quality of service
MQTT_keepalive: 300	MQTT connection keep alive (300 seconds)
CSD_enabled: on	Enables reception of GSM data calls from operator

CSD_commPort: 2	Important: ndicates call should be through COM2
TELNET_enabled: on	We enable the modem Telnet
TELNET_login: user	Login for Telnet
TELNET_password: 1234	Password for Telnet
TELNET_password: 20023	Telnet port
FIREWALL_enabled: off	Firewall disabled, accepts connections from any IP

Details:

The data is sent to the MQTT server using a JSON object, which has the following example format:

For the case of the readings of the instantaneous values of energy and power:

{“IMEI”:”353085090011697”,”TYPE”:”IEC102”,”TS”:”2020-05-15T18:00:05Z”,”P”:”ID-12345678”,”ID”:”ID001”,”VABA”:4000,”VABRI”:41,”VABRC”:1328,”PAT”:5,”PRT”-:0,”FPT”:990,”PAF1”:2,”PRF1”:0,”FPF1”:970,”PAF2”:2,”PRF2”:0,”FPF2”:970,”PAF3”:0,”PRF3”:0,”FPF3”:760,”IF1”:94,”TF1”:2392,”IF2”:107,”TF2”:2367,”IF3”:37,”TF3”:2427}

Where:

- IMEI: IMEI of the modem (unique identifier of the MTX modem)
- TYPE: Indicates the type of frame. In this case IEC102
- TS: Timestamp of when the data was read
- P: User field
- ID: User identifier indicated in IEC102_meter1
- VBAB: Active absolute energy (VabA)
- VINA: Absolute Reactive Inductive Energy (VabRi)
- VABRI: Capacitive Reactive Absolute Energy (VabRc)
- PAT: Total Active Power (KW)
- PRT: Total Reactive Power (KVA)
- FPT: Total Power Factor (thousandths)
- PAF1: Active Power Phase 1 (KW)
- PRF1: Phase 1 Reactive Power (KVA)
- FPF1: Phase 1 Power Factor (thousandths)

- PAF2: Phase 2 Active Power (KW)
- PRF2: Reactive Power Phase 2 (KVA)
- FPF2: Phase 2 Power Factor (thousandths)
- PAF3: Phase 3 Active Power (KW)
- PRF3: Reactive Power Phase 3 (KVA)
- FPF3: Phase 3 Power Factor (thousandths)
- IF1: Phase 1 current (tenths of an ampere)
- TF1: Phase 1 Voltage (tenths of a volt)
- IF2: Phase 2 current (tenths of an ampere)
- TF2: Phase 2 Voltage (tenths of a volt)
- IF3: Phase 3 current (tenths of an ampere)
- TF3: Phase 3 Voltage (tenths of a volt)

In the case of tax closings:

{“IMEI”:”353085090011697”,”TYPE”:”IEC102”,”TS”:”2020-05-15T18:00:05Z”,”P”:”ID-12345678”,”ID”:”ID001”,”CTAVM2”:[{“DO”:20,”EaA”:2813,”EiA”:1403,”CA”:0,”EaRi”:29,”EiRi”:18,”CRi”:0,”EaRc”:937,”EiRc”:450,”CRc”:0,”R7”:0,”C7”:0,”R8”:0,”C8”:0,”MPA”:6,”FMPA”:”2021-06-14T12:45-1”,”CMA”:0,”EPA”:0,”CE”:0,”DINI”:”2021-06-01T00:00-1”,”DEND”:”2021-07-01T00:00-1”},{“DO”:21,”EaA”:853,”EiA”:431,”CA”:0,”EaRi”:7,”EiRi”:7,”CRi”:0,”EaRc”:54,”EiRc”:23,”CRc”:0,”R7”:0,”C7”:0,”R8”:0,”C8”:0,”MPA”:6,”FMPA”:”2021-06-14T12:45-1”,”CMA”:0,”EPA”:0,”CE”:0,”DINI”:”2021-06-01T00:00-1”,”DEND”:”2021-07-01T00:00-1”},{“DO”:22,”EaA”:1379,”EiA”:708,”CA”:0,”EaRi”:14,”EiRi”:9,”CRi”:0,”EaRc”:298,”EiRc”:141,”CRc”:0,”R7”:0,”C7”:0,”R8”:0,”C8”:0,”MPA”:6,”FMPA”:”2021-06-07T11:00-1”,”CMA”:0,”EPA”:0,”CE”:0,”DINI”:”2021-06-01T00:00-1”,”DEND”:”2021-07-01T00:00-1”},{“DO”:23,”EaA”:302,”EiA”:151,”CA”:0,”EaRi”:1,”EiRi”:1,”CRi”:0,”EaRc”:266,”EiRc”:136,”CRc”:0,”R7”:0,”C7”:0,”R8”:0,”C8”:0,”MPA”:4,”FMPA”:”2021-06-11T08:00-1”,”CMA”:0,”EPA”:0,”CE”:0,”DINI”:”2021-06-01T00:00-1”,”DEND”:”2021-07-01T00:00-1”},{“DO”:24,”EaA”:40,”EiA”:18,”CA”:0,”EaRi”:0,”EiRi”:0,”CRi”:0,”EaRc”:55,”EiRc”:25,”CRc”:0,”R7”:0,”C7”:0,”R8”:0,”C8”:0,”MPA”:1,”FMPA”:”2021-06-06T14:30-1”,”CMA”:0,”EPA”:0,”CE”:0,”DINI”:”2021-06-01T00:00-1”,”DEND”:”2021-07-01T00:00-1”},{“DO”:25,”EaA”:161,”EiA”:60,”CA”:0,”EaRi”:7,”EiRi”:1,”CRi”:0,”EaRc”:152,”EiRc”:73,”CRc”:0,”R7”:0,”C7”:0,”R8”:0,”C8”:0,”MPA”:3,”FMPA”:”2021-06-06T18:15-1”,”CMA”:0,”EPA”:0,”CE”:0,”DINI”:”2021-06-01T00:00-1”,”DEND”:”2021-07-01T00:00-1”},{“DO”:26,”EaA”:78,”EiA”:35,”CA”:0,”EaRi”:0,”EiRi”:0,”CRi”:0,”EaRc”:112,”EiRc”:52,”CRc”:0,”R7”:0,”C7”:0,”R8”:0,”C8”:0,”MPA”:0,”FMPA”:”2021-06-27T01:30-1”,”CMA”:0,”EPA”:0,”CE”:0,”DINI”:”2021-06-01T00:00-1”,”DEND”:”2021-07-01T00:00-1”}]}

Where:

- IMEI: IMEI of the modem (unique identifier of the MTX modem)
- TYPE: Indicates the type of frame. In this case IEC102

- TS: Timestamp of when the data was read
- P: User field
- ID: User identifier indicated in IEC102_meter1
- DO: Object Address (in this case we have 7, from 20 to 26)
- EaA: Active absolute energy
- EiA: Active incremental energy
- AC: Active energy qualifier
- EaRi: Inductive reactive absolute energy
- EiRi: Incremental reactive inductive energy
- Cri: Inductive reactive energy qualifier
- EaRc: Capacitive reactive absolute energy
- EiRc: Capacitive reactive incremental energy
- CRc: Capacitive Reactive Energy Qualifier
- R7: Register 7 reserve
- C7: Qualifier of Reserve Record 7
- R8: Register 8 reserve
- C8: Qualifier of Reserve Record 8
- MPA: Maximum of Powers
- FMPA: Maximum Power Date
- CMA: Maximum Qualifier
- EPA: Excess of Powers
- CE: Excess Qualifier
- DINI: Beginning of the period
- DEND: End of period

- Remember that the configuration data of the RS232 serial port of the modem must match the configuration of the RS232 port of the meter.
- If you need to monitor other parameters of the meter in addition to the above, check the possibility at iotsupport@mtxm2m.com
- The IEC102_ configuration parameters are only available in the “MTX-Tunnel IEC870”, it is a product that is supplied separately
- Not all types of meters allow simultaneous reading connected to the same RS485 bus. If in doubt, consult the meter manufacturer

8. ANNEX: EXAMPLE SCENARIOS AND CONFIGURATIONS FOR THE USE OF DIGITAL OUTPUTS AND RELAYS

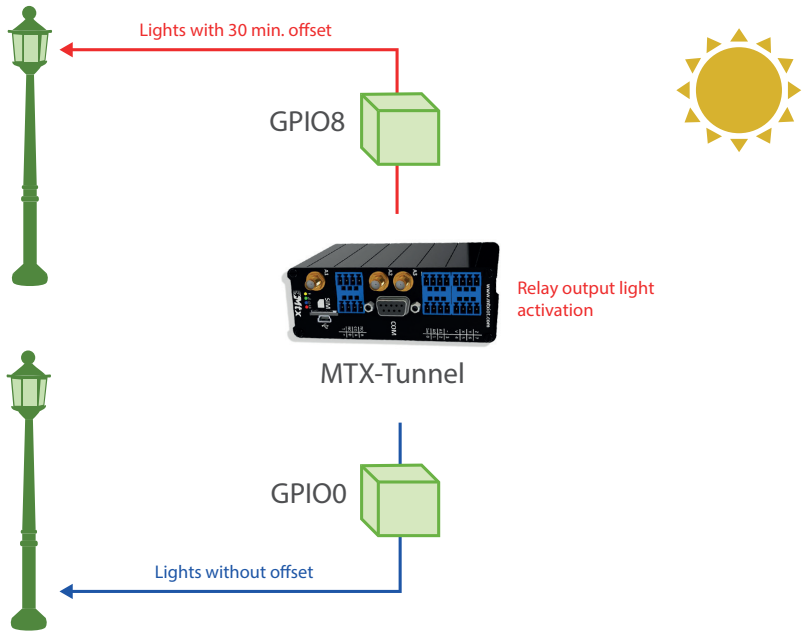
8.1 EXAMPLE: Switching of relays based on the MTX astronomical clock.

Scenario details:

- We want to switch a relay based on the sunrise and sunset to activate a lighting system
- It is not possible to set a fixed time, as the time for sunrise/sunset depends on the location (sunrise/ sunset does not happen at the same time in Madrid than it does in Barcelona or Mexico DF) and on the time of year (sunrise/ sunset does not happen at the same time in Summer as it does in Winter). To do so, the modem MTX must be set up as an astronomical clock
- We also want the lights to connect 30 minutes prior to sunset and disconnect 30 minutes after sunrise
- The device location is the town of Cardedeu, with the GPS latitude: 41.6333 and longitude: 2.36667

Solution:

MTX-IoT [4-S-N-N]-STD-N-RL modem+MTX-Tunnel software



EXAMPLE of settings (file config.txt) for such scenario:

MTX_pin: 0000	SIM Card PIN
MTX_model: 199802407	MTX model
MTX_mode: none	We do not want GPRS
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server (the MTX must sync the time)
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_latitude: 41.6333	GPS latitude
MTX_longtude: 2.36667	GPS longitude
SMS_allPhones: on	All phones are allowed
SMS_ATEnabled: on	IP by SMS authorized
SMS_ATResponse: on	AT by SMS allowed
SMS_sendIP: on	SMS AT responses activated
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS login provided by the GSM operator
GPRS_password: MOVISTAR	GPRS password provided by the GSM operator
GPRS_timeout: 1	The modem will not be permanently connected
GPIO_mode0: output	GPIO0 as output
GPIO_config0: astronomical;0;0	Astronomical mode without offset
GPIO_mode8: output	GPIO8 as output
GPIO_config8: astronomical;-30;30	Astronomical mode with 30 min. offset

Details:

- Of course, this scenario can be combined with others that simultaneously perform Serial / GPRS gateways to access serial devices (modbus or others) as well as others in which the MTX-Tunnel autonomously collects modbus data and sends it via JSON to Web platforms
- As of the MTX-Tunnel v11.12 version, it is possible to use the command AT ^ MTXTUNNEL = SETIOMAINTENANCE, idGPIO, mode to establish a maintenance mode (mode = 1) in a specific output GPIO. When an output is in maintenance mode, the output does not obey the astronomical clock, but is managed by the command AT ^ MTXTUNNEL = SETIO, idGPIO, value

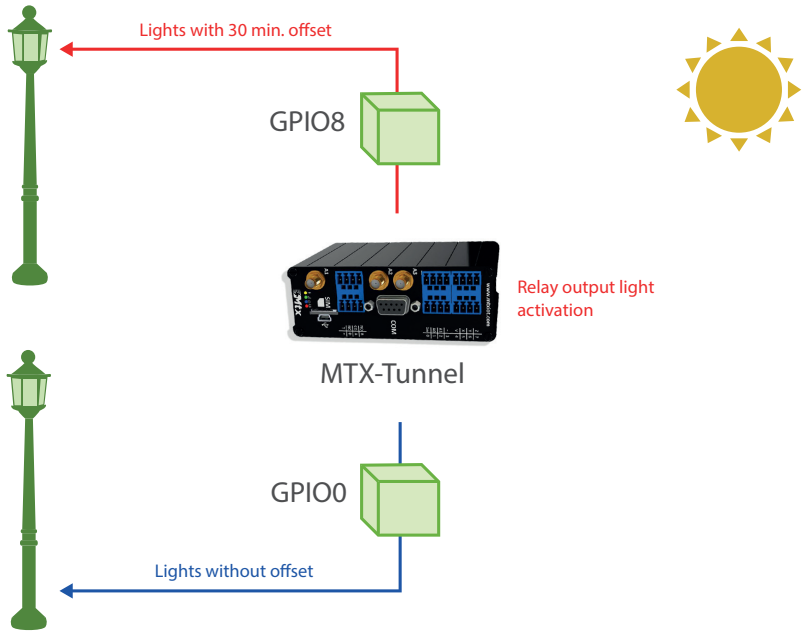
8.2 EXAMPLE: Switching of a relay based on the MTX astronomical clock and based on scheduled times for special days.

Scenario details:

- It is necessary to switch a relay depending on the sunrise and sunset to activate a lighting system
- It is not possible to establish a fixed time, since the Sunrise and Sunset time depends on the place (it does not dawn / dusk at the same time in Madrid as Barcelona or Mexico City) and the day of the year (it does not dawn / dusk at the same time in Summer as in Winter). For this, the MTX modem must be configured as an astronomical clock
- For special days, such as Christmas or local holidays, it must be possible to configure schedules to activate / deactivate the relay in a preprogrammed way at a certain time
- The location of the device is the town of Cardedeu, with GPS position latitude: 41.6333 and longitude: 2.36667

Solution:

MTX-IoT [4-S-N-N]-STD-P modem+software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

MTX_PIN: 0000	If SIM card doesn't have PIN, leave as 0000
MTX_model: 199802407	The MTX model chosen is MTX-DIN
MTX_mode: none	We do not want GPRS gateways
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server (the MTX must sync the time)
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_latitude: 41.6333	GPS latitude position
MTX_longitude: 2.36667	GPS longitude position
SMS_allPhones: on	IP by SMS authorized
SMS_ATEnabled: on	IP by SMS authorized
SMS_ATResponse: on	AT by SMS allowed
SMS_sendIP: on	SMS AT responses activated
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
GPIO_mode0: output	GPIO0 as output
GPIO_config0: astronomical;0;0	Astronomical mode without offset
GPIO_mode8: output	GPIO8 as output
GPIO_config8: astronomical;0;0	Astronomical mode without offset

Details:

- Of course this scenario can be combined with others that simultaneously perform Serial / GPRS gateways to access serial devices (modbus or others) as well as others in which the MTX-Tunnel independently collects modbus data and sends them via JSON to Web platforms
- The MTX-Tunnel works in the following way with respect to its astronomical clock: every time a change of day occurs, it calculates the hour / minute of the moments “orto” (sunrise) and “ocaso” (sunset) . But before calculating these moments, consult the file “astroschedule.txt”, where exceptions can be defined for certain days. For example, if you define a special sunrise / sunset time for January 5 and it is included in the “astroschedule.txt” file, the MTX-Tunnel will use that time instead of the calculated one
- The file “astroschedule.txt” has the following format:

  ID:day;month;hourDawn;minutesDawn;hourSunset;minutesSunset\r\n

  Where ID is a value that indicates the hour identifier (can range from 1 to 200). Note that after the ID it must be written “:” (colon), unlike the rest of the parameters that are separated by “;” (semicolon)

  day: indicates the day of the month (1... 31)

  month: indicates the month (1... 12)

  hourDawn: indicates the “forced hour” of sunrise (0... 23)

  minutesDawn: indicates the “forced minutes” of sunrise (0... 59)

  hourSunset: indicates the “forced hour” of sunset (0... 23)

  minutesSunset: indicates the “forced minutes” of sunset (0... 59)

  For example, if we want to specify 2 times, one for January 5 and the other for October 12, the file “astroschedule.txt” could be written like this:

  1:5;1;8;0;17;30

  2:12;10;7;45;18;30

  Where on January 5 the time of forced ortho would be 08:00 and sunset 17:30 and on October 12 they would be 07:45 and 18:30 respectively
- The file “astroschedule.txt” can be written manually in the root directory of the modem’s memory (next to the file config.txt “), it can also be downloaded completely into the modem’s memory with the command AT ^ MTXTUNNEL = DOWNLOAD, ... (It would be downloaded via http from a URL), but it is also possible to use a series of AT commands, such as: AT^MTXTUNNEL=SETASTROSCHEDULE, AT^MTXTUNNEL=GETASTROSCHEDULE, AT^MTXTUNNEL=GETASTROSCHEDULES, AT^MTXTUNNEL=DELASTROSCHEDULES, AT^MTXTUNNEL=DELASTROSCHEDULE. See the information for these commands in this manual for more information.
- Keep in mind that if you configure the parameter OUTPUT_config3 (to indicate an offset of sunrise and sunset) the times specified in the file “astroschedule.txt” will not be affected
- Keep in mind that, as always, the time specified in the file “astroschedule.txt” must be UTC time

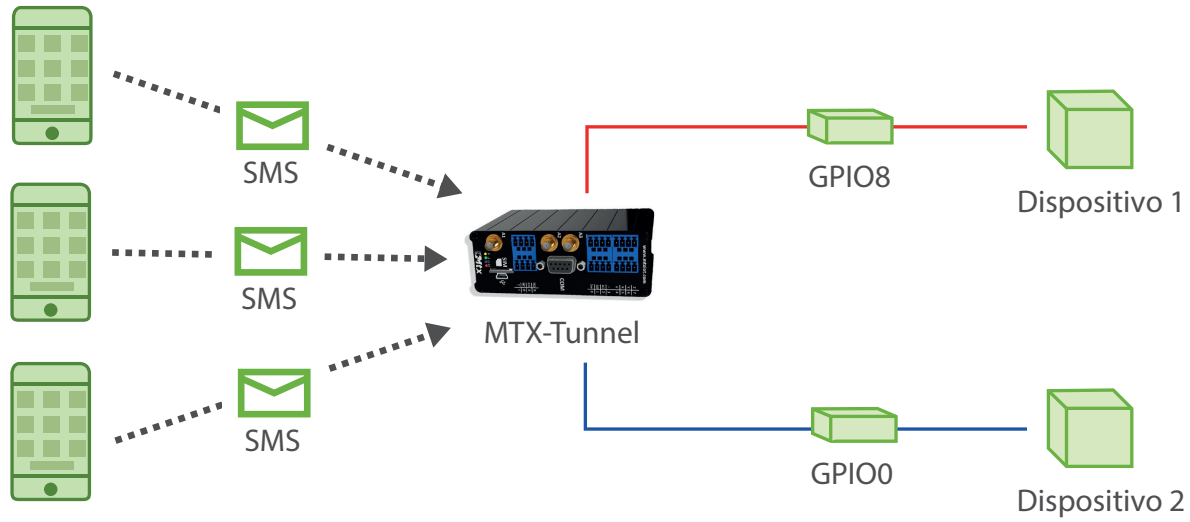
8.3 EXAMPLE: Switching of a relay for 5 seconds with an SMS message.

Scenario details:

- We need to be able to activate a relays for some seconds by sending an SMS message
- By sending an SMS with the text “ON”, the relay will be activated for X seconds. Then, the relay (when an SMS with “ON” is received) will be activated for 10 seconds
- Only the authorized telephone numbers (666123456 and 666123457) will be capable of activating the relay
- The relay needs to be activated remotely, via Telnet, as well as being able to change the MTX configuration remotely

Solution:

MTX-IoT [4-S-N-N]-STD-N-RL modem+MTX-Tunnel software



EXAMPLE of settings (file config.txt) for such scenario:

GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected

MTX_PIN: 0000	If SIM card doesn't have PIN, leave as 0000
MTX_model: 199802407	The MTX model chosen is MTX-DIN
MTX_mode: none	We do not want GPRS gateways
MTX_urc: off	We do not need URC information messages
FIREWALL_enabled: off	Connections from any IP can be made
GPIO_mode0: output	GPIO0 as output
GPIO_config0: timer	GPIO0 as output TIMER (temporized)
GPIO_mode8: output	GPIO8 as output
GPIO_config8: timer	GPIO8 as output TIMER (temporized)
SMS_allPhones: off	Not all phones will be used
SMS_validPhone1: +34666123456	Authorized number 1
SMS_validPhone2: +34666123457	Authorized number 2
SMS_ATEnabled: on	AT commands can be received
SMS_ATResponse: on	SMS responses sent to AT commands
SMS_sendIP: on	The modem returns IP after activating IP session
SMS_alias1: RELE1>AT^MTXTUNNEL= SETOUTPUTTIMER,0,5	GPIO0 activated during 5 secs.
SMS_alias2: RELE2>AT^MTXTUNNEL= SETOUTPUTTIMER,8,	GPIO8 activated during 5 secs.
SMS_aliasOk: CommandOk	If command correctly executed, sent by SMS
SMS_aliasError: CommandError	If command incorrectly executed, sent by SMS
TELNET_enabled: on	Telnet is activated
TELNET_login: user	Login for Telnet

TELNET_password: 1234	Password for Telnet
TELNET_port: 20023	Telnet port

Details:

- Remember that in order to activate the 4G/3G/2G session it's enough to send an SMS with the text "mtxtunnel on." From then on you can connect to the equipment via Telnet
- At the end of this guide you'll find a table with the description of the modem I/Os
- To change the relay via Telnet, you just need to send the command AT^MTXTUNNEL=SETOUTPUTTIMER,3,10

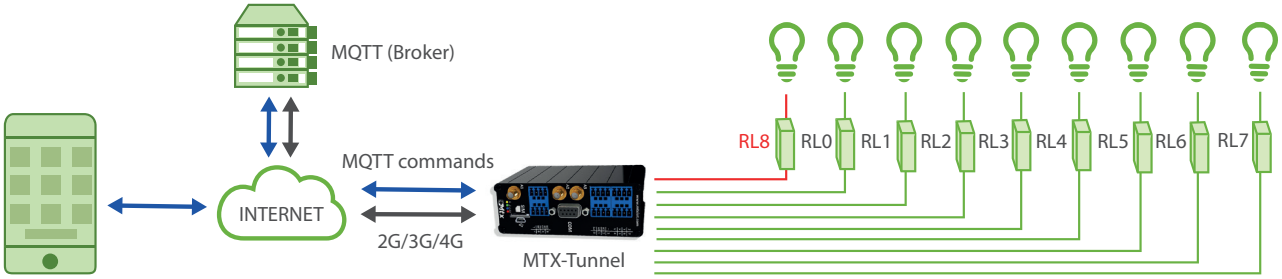
8.4 EXAMPLE: Example of using MQTT to activate/deactivate relays of a 3G modem from an Android/IO mobile phone.

Scenario details:

- Intended to be able to activate/deactivate a relay output from an Android or Iphone mobile phone
- The SIM card to be used is very affordable and does not have a public IP address, but a private one of the kind 10.x.x.x

Solution:

MTX-IoT [4-S-N-N]-STD-N-RL modem+software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_model: 199802407	MTX-Terminal modem model used
MTX_mode: none	We do not configure gateways
MTX_PIN: 0000	PIN if it has one
MTX_ping: 30	Every 30 minutes PING check

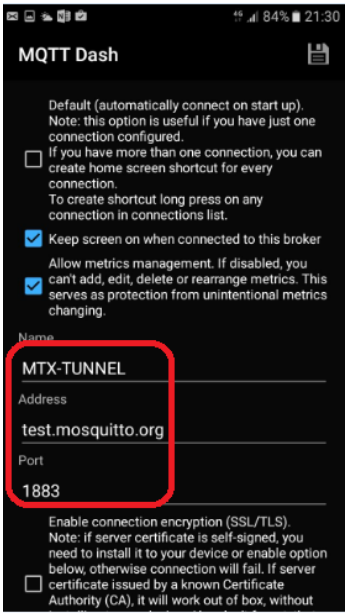
MTX_pingIP: 8.8.8.8	IP address for connection checking
MTX_ATLimited: off	We do not want limited AT commands
FIREWALL_enabled: off	All authorized Ips
TELNET_enabled: on	We enable the Telnet of the modem
TELNET_login: user	Login for Telnet
TELNET_password: 1234	Password for Telnet
TELNET_firewall: off	They can be connected by telnet from any IP
TELNET_port: 20023	Port for Telnet
GPIO_mode0: output	GPIO0 as output
GPIO_mode1: output	GPIO1 as output
GPIO_mode2: output	GPIO2 as output
GPIO_mode3: output	GPIO3 as output
GPIO_mode4: output	GPIO4 as output
GPIO_mode5: output	GPIO5 as output
GPIO_mode6: output	GPIO6 as output
GPIO_mode7: output	GPIO7 as output
GPIO_mode8: output	GPIO8 as output mandatory (it's a relay)
MQTT_enabled: on	We enable the MQTT service on the modem
MQTT_server: tcp://test.mosquitto.org:1883	We specify the broker IP/DNS, including port
MQTT_id: [IMEI]	Identifier
MQTT_attopic1: [IMEI]/AT1	To receive commands

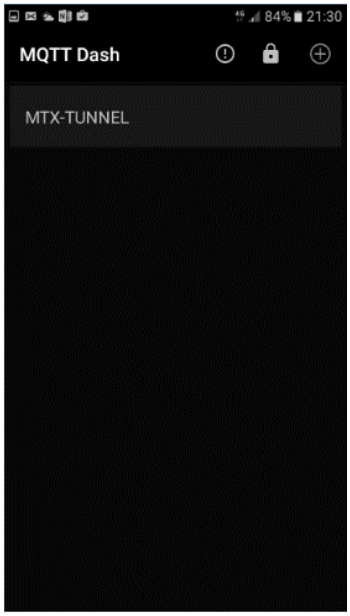
MQTT_qos: 2	Quality of service
MQTT_keepalive: 300	Keep alive MQTT connection (300 seconds)
MQTT_persistent: off	We do not need persistence

Testing the example:

To test this example, let's assume that the modem has already loaded the config.txt file and with the inserted SIM. What we are going to configure here is an example for the phone. Suppose it is an Android phone.

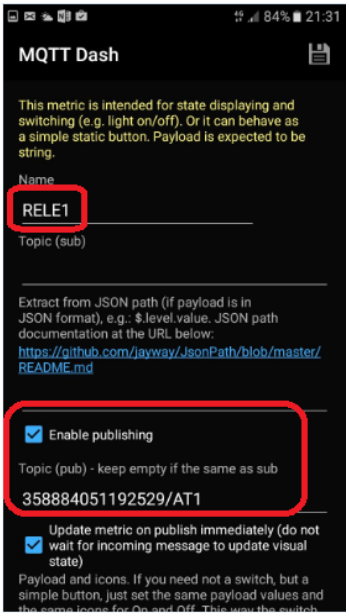
- We download an MQTT application of the PlayStore. For example MQTT Dash
- We configure certain details of the application. For example, we created a MTX-TUNNEL device





- Once the MTX-TUNNEL device has been created, we create and configure a button, for example one with the name RELÉ1

  We configure the name of the relay and the topic that we configure in MQTT_attopic1 (obviously replacing [IMEI] tag with the corresponding imei. We configure the command AT ^ MTXTUNNEL = SETIO,5,1 in the active button, and in the disabled one AT ^ MTXTUNNEL = SETIO,5,0. We can specify a QoS 2





- We can generate an application like the following one:



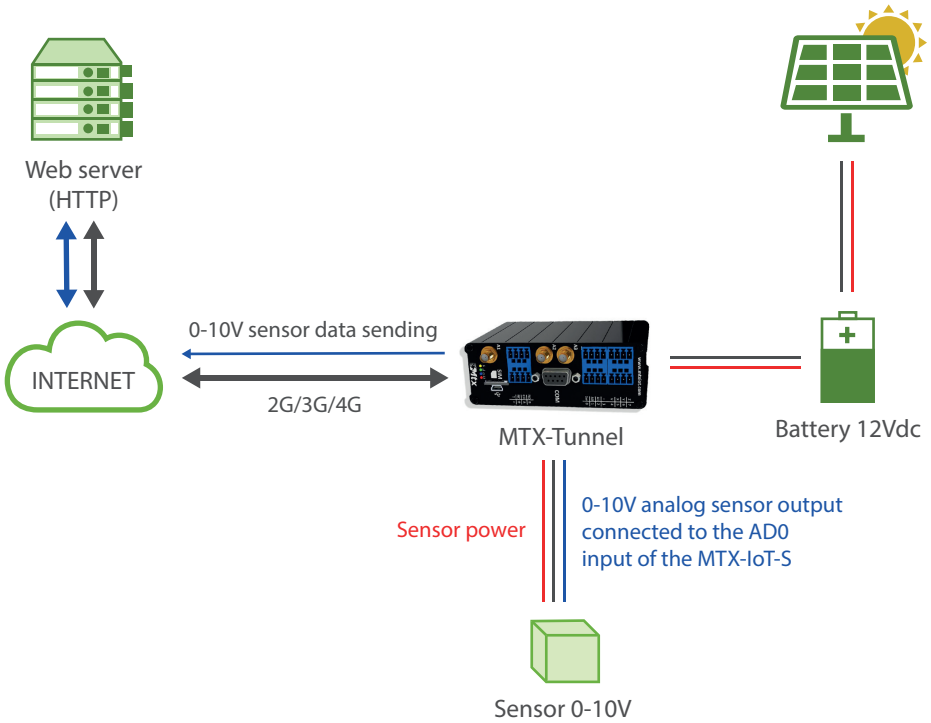
8.5 EXAMPLE: Periodic monitoring by 4G/3G/2G of 0-10V sensor activating the power supply of the sensor through a relay output of the modem. Sending to web server using HTTP GET.

Scenario details:

- It is intended to monitor a 0-10V sensor with a 4G/3G/2G modem
- The modem and the sensor will be powered by solar panels, so energy consumption is important
- For this reason, the sensor to be read will be powered through the internal relay with the MTX-IOT-S modem associated with its GPIO8 output. That is, the MTX-IOT-S modem, before reading the sensor, will activate the GPIO8 (internal relay) to supply said sensor 0-10V. After a 5 second wait for sensor stabilization, the MTX-IOT-S modem will read the sensor that will be connected to its AD0 input (which allows an input between 0 and 50V) and then deactivate the relay again to save consumption
- The monitoring must be at certain times (at 00:00 UTC, 06:00 UTC, 14:30 UTC and 18:30 UTC) the data will be stored in the flash memory of the modem (datalogger) and will be sent to a server web via HTTP GET as soon as there is 4G/3G/2G coverage

Solution:

Modem MTX-IoT [4-S-N-N]-STD-N + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server (the MTX must sync the time)
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_TPFormat: unix	Sent JSON format
MTX_mode: none	Gateways used
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
FIREWALL_enabled: off	Authorized IPs
SMS_allPhones: on	IP by SMS authorized
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_ATResponse: on	SMS AT responses activated
GPIO_mode8: output	GPIO configured as output
GPIO_config8: normal	Configuration of GPIO
ADC_mode0: voltage	ADC configured as voltage

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
LOGGER_enabled: on	Logger on to store readings
LOGGER_server: www.metering.es/json/set.asp?data=	Data sending URL
LOGGER_password: 12345678	“P” field of JSON
LOGGER_numRegistersFlash: 1500	Registry numbers in flash
LOGGER_registerSize: 300	Registry size in flash
LOGGER_numRegistersRam: 0	Number of readings stored in RAM
LOGGER_httpMode: getjson	Data sending mode to server
TELNET_enabled: on	Telnet status
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	TCP port for telnet

Details:

- The frame sent to the server has the following JSON format

  31/01/2020 12:20:33 --- {“IMEI”:354033091487838,”TYPE”:”IOS”,”TS”:”2020-01-05T06:00:33Z”,”P”:”12345678”,”IO0”:1,”IO1”:1,”IO2”:1,”IO3”:1,”IO4”:1,”IO5”:0,“IO6”:0,”IO7”:0,”-IO8”:1,”AD0”:5935,”AD1”:0}

  Where:

  TYPE: IOS frame type

  IMEI: internal identification of the MTX

  Q: User field specified in the LOGGER_password parameter

  TS: TimeStamp of when data was collected

  IO0: It is the value of the GPIO0 of the modem (not used in this ex.)

  IO1: It is the value of the GPIO1 of the modem (not used in this ex.)

  IO2: It is the value of the GPIO2 of the modem (not used in this ex.)

IO3: It is the value of the GPIO3 of the modem (not used in this ex.)

IO4: It is the value of the GPIO4 of the modem (not used in this ex.)

IO6: It is the value of the GPIO5 of the modem (not used in this ex.)

IO7: It is the value of the GPIO6 of the modem (not used in this ex.)

IO8: It is the value of the GPIO7 of the modem (not used in this ex.)

IO9: It is the value of the GPIO8 of the modem (Internal relay of the modem)

AD0: It is the value of the AD0 input of the modem (sensor reading 0-10V in millivolts, from 0 to 50000)

AD1: Is the value of the AD1 input of the modem. Not used in this example

- This example makes use of the files “iologger_start.txt”, “iologger_end.txt”. These text files allow you to enter commands that will be executed automatically BEFORE reading the modem I/O data and AFTER reading. Both files must be placed in the /atscripts folder

  The content of “iologger_start.txt” for this example is as follows:

  EXECUTE AT^MTXTUNNEL=SETIO,8,1

  PAUSE 5

  Basically it executes an AT command that activates the internal relay of the MTX-IOT-S modem (which is associated with the GPIO8) and waits 5 seconds.

  The content of “iologger_end.txt” for this examplei is as follows:

  EXECUTE AT^MTXTUNNEL=SETIO,8,0

  Where the relay (GPIO8) is simply deactivated after reading the 0-10V sensor

- This example makes use of the file “schedule.txt”, whose documentation you will find in this manual, which allows you to enter the execution times of certain AT commands. The content of the file “schedule.txt” (which must be located in the root directory of the modem) for this application is as follows:

  1:-1;0;0;AT^MTXTUNNEL=IOEVENT

  2:-1;6;0;AT^MTXTUNNEL=IOEVENT

  3:-1;14;30;AT^MTXTUNNEL=IOEVENT

  4:-1;18;30;AT^MTXTUNNEL=IOEVENT

  4 times are entered in this file. Every day of the week, at 00:00, 06:00, 14:30 and 18:30 the command AT ^ MTXTUNNEL = IOEVENT will be executed. This AT command will launch the process of reading the modem’s I/O, storing it in the modem’s datalogger and sending it to the web server

- Remember that the modem always works with UTC time
- Properly configure the microswitches of the MTX-IOT-S modem to be able to read voltage in ADC0 (description of the microswitches in the Annexes of the manual)

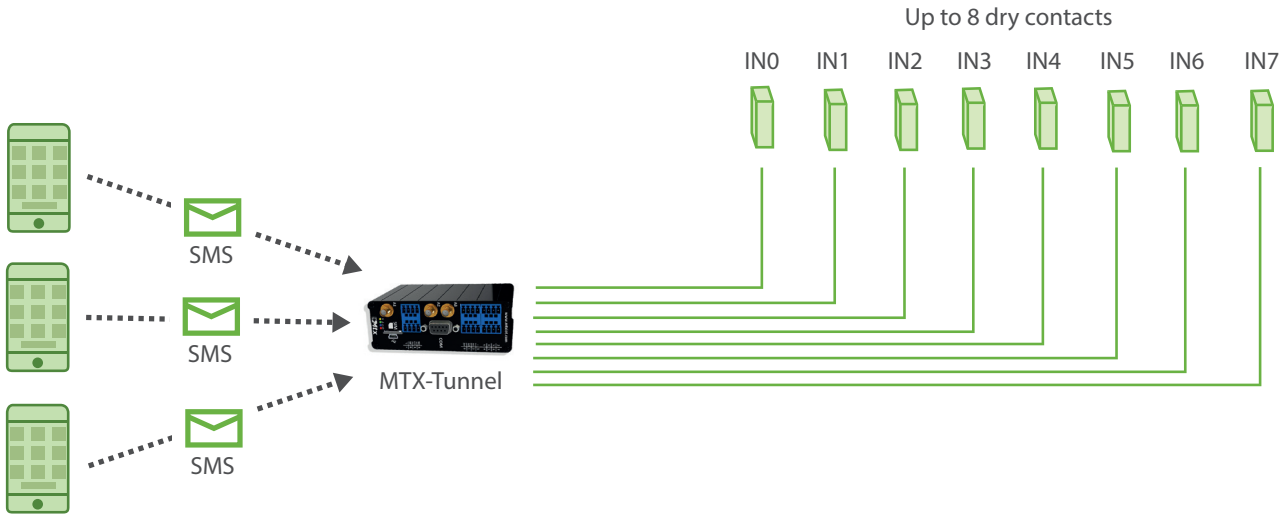
8.6 EXAMPLE: SMS alarms to change up to 8 digital inputs.

Scenario details:

- 8 dry contact sensors need to be monitored
- In the event of a state change in any of them, an SMS message must be sent to several phone numbers. Each dry contact must have a personalized message that identifies the sensor that caused the alarms, as well as whether the alarm was activated (dry contact closed) or opened (dry contact open).
- The phone numbers that will receive the SMS alerts are +34666123456, + 34666123457 and +346661234568
- To avoid sending a high number of SMS (in case of rapid changes in a certain entry), a timeout of 60 seconds must be established per entry, that is, no more than 1 SMS per minute will be sent for changes in a certain entry

Solution:

Modem MTX-IOT-S [4-N] + firmware MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
SMS_allPhones: off	Phone numbers
SMS_validPhone1: +34666123456	Phone number 1
SMS_validPhone2: +34666123457	Phone number 2
SMS_validPhone3: +34666123458	Phone number 3
ALARM_smsNumber1: +34666123456	Phone number 1
ALARM_smsNumber2: +34666123457	Phone number 2
ALARM_smsNumber3: +34666123458	Phone number 3
GPIO_mode0: input	GPIO0 configured as input
GPIO_config0: sms;2;60;Alarma GPIO0 ON;Alarma GPIO0 OFF	GPIO0 SMS configuration
GPIO_mode1: input	GPIO1 configured as input
GPIO_config1: sms;2;60;Alarma GPIO1 ON;Alarma GPIO1 OFF	GPIO1 SMS configuration
GPIO_mode2: input	GPIO2 configured as input
GPIO_config2: sms;2;60;Alarma GPIO2 ON;Alarma GPIO2 OFF	GPIO2 SMS configuration
GPIO_mode3: input	GPIO3 configured as input
GPIO_config3: sms;2;60;Alarma GPIO3 ON;Alarma GPIO3 OFF	GPIO3 SMS configuration
GPIO_mode4: input	GPIO4 configured as input

GPIO_config4: sms;2;60;Alarma GPIO4 ON;Alarma GPIO4 OFF	GPIO4 SMS configuration
GPIO_mode5: input	GPIO5 configured as input
GPIO_config5: sms;2;60;Alarma GPIO5 ON;Alarma GPIO5 OFF	GPIO5 SMS configuration
GPIO_mode6: input	GPIO6 configured as input
GPIO_config6: sms;2;60;Alarma GPIO6 ON;Alarma GPIO6 OFF	GPIO6 SMS configuration
GPIO_mode7: input	GPIO07 configured as input
GPIO_config7: sms;2;60;Alarma GPIO7 ON;Alarma GPIO7 OFF	GPIO7 SMS configuration

Details:

- The configuration of the inputs as “sms; 2; 30; GPIO0 ON alarm; GPIO0 OFF alarm” indicates the following:

  “sms” > The input is configured to send alarm by SMS

  “2” > The 2 indicates that the digital input is configured to send an SMS message both by activating the input (when it closes, bringing it to ground) and by deactivating the input (when it opens). If you want to send an SMS message only when closing the entry (taking it to ground), you should indicate a value of “1”

  “60” > Indicates the timeout of the digital input. This means that, even if there are multiple changes in the digital input, more than 1 SMS will never be sent in those 60 seconds. Note: in the event that the entrance closes and it opens again after 5 seconds, an SMS will be received indicating the closure and, after 60 seconds, the opening

  “Alarm GPIO0 ON” > Indicates a text message to be sent when the input is activated (when the input is ground)

  “Alarm GPIO0 OFF” > Indicates a text message to be sent when the input is deactivated

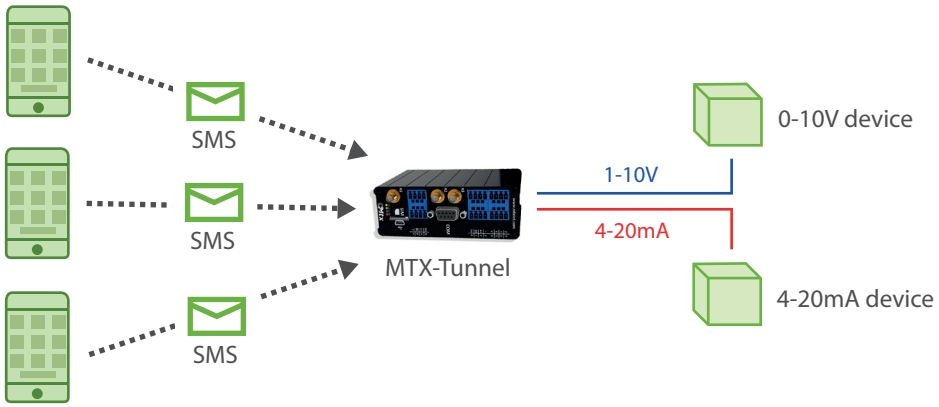
8.7 EXAMPLE: SMS alarms to change up to 2 analog inputs.

Scenario details:

- 2 analog sensors need to be monitored, one 0-10V and the other 4-20mA
- If the 0-10V sensor registers a voltage higher than 8V or lower than 2V, an alarm SMS message must be sent, indicating the type of alarm. You must also inform with another SMS in case the alarm is deactivated
- If the 4-20mA sensor registers a current higher than 16mA or less than 8mA, an alarm SMS message must be sent, indicating the type of alarm. You must also inform with another SMS in case the alarm is deactivated
- The phone numbers that will receive the SMS alerts are +34666123456, +34666123457 and +346661234568
- To avoid sending a high number of SMS (when the measured value is just within the alarm limits), a hysteresis of 100mV must be configured for the 0-10V sensor and 0.5mA for the 4-20mA sensor

Solution:

Modem MTX-IOT-S [4-N] + firmware MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
SMS_allPhones: off	Phone numbers
SMS_validPhone1: +34666123456	Phone number 1
SMS_validPhone2: +34666123457	Phone number 2
SMS_validPhone3: +34666123458	Phone number 3
ALARM_smsNumber1: +34666123456	Phone number 1
ALARM_smsNumber2: +34666123457	Phone number 2
ALARM_smsNumber3: +34666123458	Phone number 3
ADC_mode0: voltage	ADC0 configured as voltage
ADC_config0: sms;2000;8000;100;60;Alarm ADC0 Low;Alarm ADC0 High;Alarm ADC0 normal	
ADC_mode1: current	ADC1 configured as current
ADC_config1: sms;8000;16000;500;60;Alarm ADC1 Low;Alarm ADC1 High;Alarm ADC1 normal	

Details:

- The configuration of ADC0 input as “sms;2000;8000;100;60;Alarm ADC0 Low;Alarm ADC0 High;Alarm ADC0 normal” indicates:

  “sms” > The analog input is configured to send an alarm by SMS depending on its value

  “2000” > Millivolts below which the modem will send a low voltage alarm SMS message

  “8000” > Millivolts above which the modem will send a high voltage alarm SMS message

  “100” > 100mV hysteresis. This means that when an alarm is generated for exceeding, in the

case of this example, 8000mV, it is not considered that it has returned to a normal state until it has a reading of 8000-100 = 7900mV. In other words, an SMS message with the “Alarm ADC0 normal” message will not be sent until then. This prevents 8000mV, 7999mV, 8001mV ... from periods of time when the modem is reading continuously high-voltage SMS messages

“60” > 60 second timeout. Sending more than 1 SMS every 60 seconds will not be allowed

“Alarm ADC0 Low” > Indicates a text message that will be sent to when the read voltage is below the setpoint, in the case of the example, 2000mV)

“Alarm ADC0 High” > Indicates a text message to be sent to when the read voltage is above the setpoint, in the case of the example, 8000mV)

“Alarm ADC0 Normal” > Indicates a text message to be sent to when, after an alarm, the read voltage returns within limits (note hysteresis)

- The configuration of ADC1 input as “sms;8000;16000;500;60;Alarm ADC1 Low;Alarm ADC1 High;Alarm ADC1 normal” indicates:

  “sms” > The analog input is configured to send an alarm by SMS depending on its value

  “8000” > Microamps below which the modem will send an alarm message for low current

  “16000” > Microamps above which the modem will send a high current alarm SMS message

  “500” > Hysteresis of 500 Microamps (0.5mA). This means that when an alarm is generated to exceed, in the case of this example, the 16000uA, it is not considered that it has returned to a normal state until it has a reading of 16000-500 = 15500uA. In other words, an SMS message with the “Alarm ADC1 normal” message will not be sent until then. This prevents 16000uA, 15998uA, 16002uA from being read in periods of time ... the modem is continuously sending SMS messages

  “60” > 60 second timeout. Sending more than 1 SMS every 60 seconds will not be allowed

  “Alarm ADC1 Low” > Indicates a text message to be sent to when the current read is below the setpoint, in the case of the example, 8000uA (8mA)

  “Alarm ADC1 High” > Indicates a text message to be sent to when the read current is above the setpoint, in the case of the example, 16000uA (16mA)

  “Alarm ADC1 Normal” > Indicates a text message to be sent to when, after an alarm, the read current returns within limits (note hysteresis)

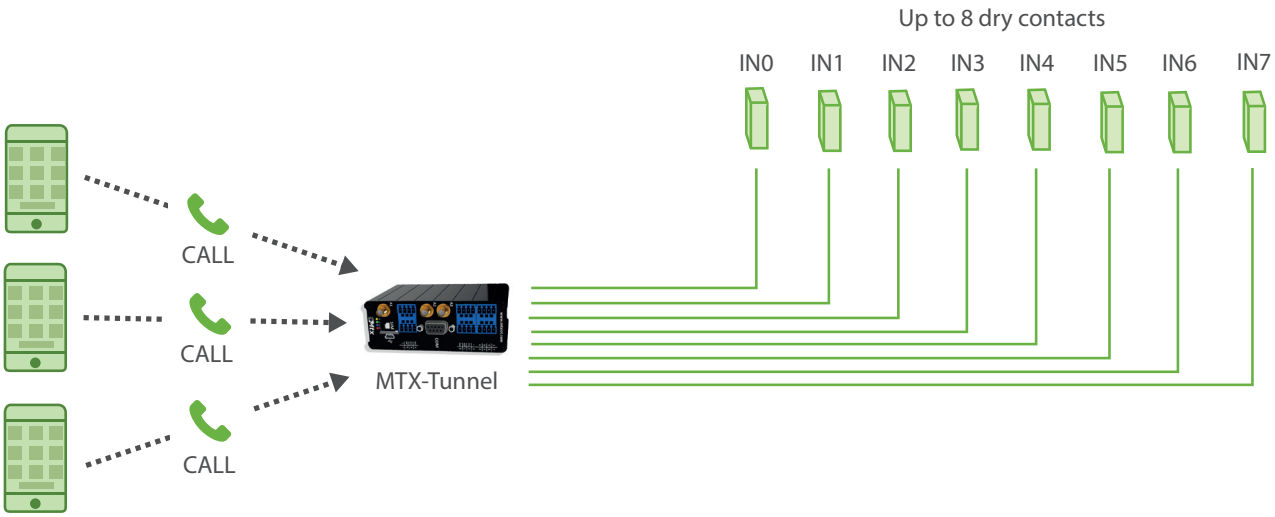
8.8 EXAMPLE: Voice call alarm to change up to 8 digital inputs.

Scenario details:

- 8 dry contact sensors need to be monitored
- In the event of an activation in any of them, a voice call must be made to several phone numbers
- The phone numbers that will receive the SMS alerts are +34666123456, + 34666123457 and +346661234568
- To avoid making a large number of voice calls (in case of rapid changes in a certain entry), a timeout of 5 minutes should be established per entry, that is, no more than 1 call will be made in a 5-minute window by activating a certain input

Solution:

Modem MTX-IOT-S [4-N] + firmware MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
SMS_allPhones: off	Phone numbers
SMS_validPhone1: +34666123456	Phone number 1
SMS_validPhone2: +34666123457	Phone number 2
SMS_validPhone3: +34666123458	Phone number 3
ALARM_smsNumber1: +34666123456	Phone number 1
ALARM_smsNumber2: +34666123457	Phone number 2
ALARM_smsNumber3: +34666123458	Phone number 3
GPIO_mode0: input	GPIO0 configured as input
GPIO_config0: call;1;300	GPIO0 call configuration
GPIO_mode1: input	GPIO1 configured as input
GPIO_config1: call;1;300	GPIO1 call configuration
GPIO_mode2: input	GPIO2 configured as input
GPIO_config2: call;1;300	GPIO2 call configuration
GPIO_mode3: input	GPIO3 configured as input
GPIO_config3: call;1;300	GPIO3 call configuration
GPIO_mode4: input	GPIO4 configured as input
GPIO_config4: call;1;300	GPIO4 call configuration
GPIO_mode5: input	GPIO5 configured as input

GPIO_config5: call;1;300	GPIO5 call configuration
GPIO_mode6: input	GPIO6 configured as input
GPIO_config6: call;1;300	GPIO6 call configuration
GPIO_mode7: input	GPIO07 configured as input
GPIO_config7: call;1;300	GPIO7 call configuration

Details:

- The configuration of the inputs as “call; 1; 300” indicates the following:

  “call” > The input is configured to send alarm by voice call (the voice call will be made, but will not play any audio on the other end of the phone

  “1” > The 2 indicates that the digital input is configured to make the voice call by activation of the input (when it closes taking it to ground)

  “300” > Indicates the timeout of the digital input, in seconds. This means that, even if there are multiple changes in the digital input, there will never be more than 1 call in a 5-minute window (5 minutes = 300 seconds)

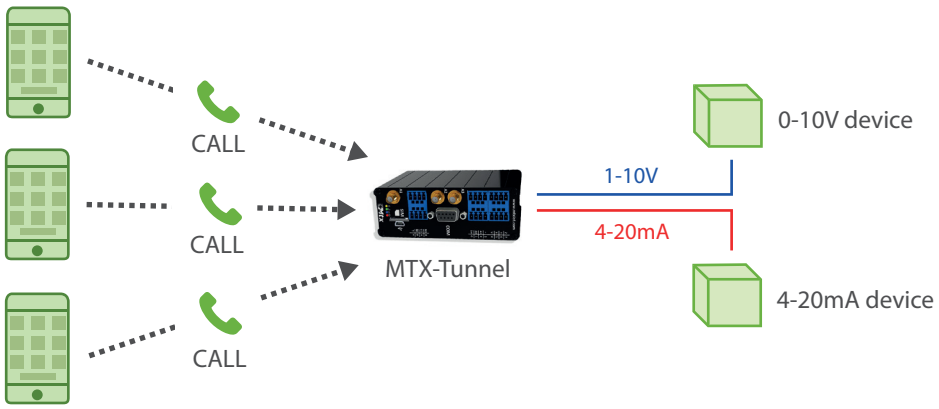
8.9 EXAMPLE: Voice call alarm to change up to 2 analog inputs of 0-10V and 4-20mA.

Scenario details:

- 2 analog sensors need to be monitored, one 0-10V and the other 4-20mA
- If the 0-10V sensor registers a voltage greater than 8V or less than 2V, an alert voice call must be made to a series of telephone numbers
- If the 4-20mA sensor registers a current greater than 16mA or less than 8mA, an alert voice call must be made to a series of telephone numbers
- The phone numbers that will receive the alert calls are +34666123456, +34666123457 and +346661234568
- To avoid sending a high number of calls (when the measured value is just within the alarm limits), a hysteresis of 100mV must be configured for the 0-10V sensor and 0.5mA for the 4-20mA sensor.
- A timeout of 5 minutes per input must also be established, that is, no more than 1 call will be made in a 5-minute window due to the activation of a certain analog input

Solution:

Modem MTX-IOT-S [4-N] + firmware MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
SMS_allPhones: off	Phone numbers
SMS_validPhone1: +34666123456	Phone number 1
SMS_validPhone2: +34666123457	Phone number 2
SMS_validPhone3: +34666123458	Phone number 3
ALARM_smsNumber1: +34666123456	Phone number 1
ALARM_smsNumber2: +34666123457	Phone number 2
ALARM_smsNumber3: +34666123458	Phone number 3
ADC_mode0: voltage	ADC0 configured as voltage
ADC_config0: ADC_config0: call;2000;8000;100;300	
ADC_mode1: current	ADC1 configured as current
ADC_config1: ADC_config1: call;8000;16000;500;300	

Details:

- The configuration of the inputs as “call; 1; 300” indicates the following:

  “call” > The input is configured to send alarm by voice call (the voice call will be made, but will not play any audio on the other end of the phone)

  “1” > The 2 indicates that the digital input is configured to make the voice call by activation of the input (when it closes taking it to ground)

  “300” > Indicates the timeout of the digital input, in seconds. This means that, even if there are multiple changes in the digital input, there will never be more than 1 call in a 5-minute window (5 minutes = 300 seconds)

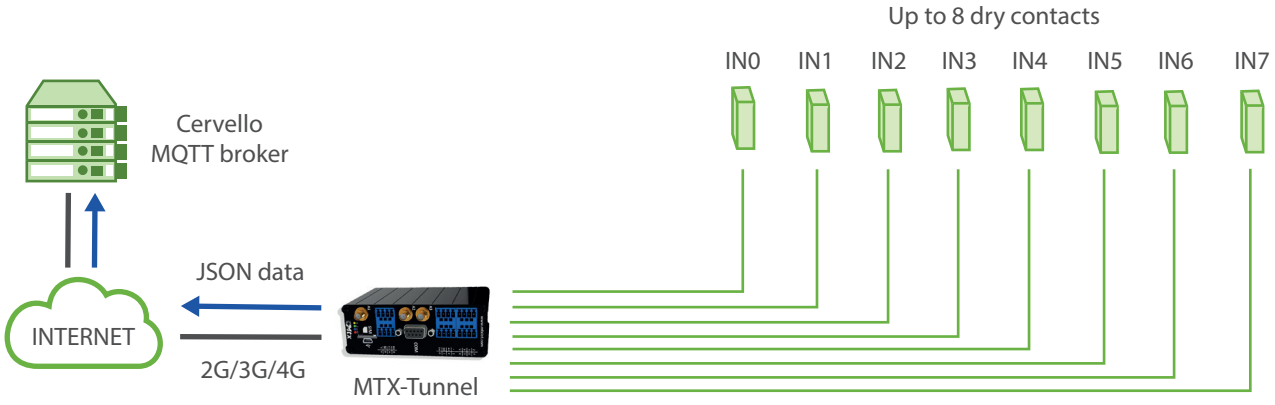
8.10 EXAMPLE: Instant sending of telemetry via MQTT of up to 8 digital inputs.

Scenario details:

- 8 dry contact sensors need to be monitored. In the state of the entries, it must be sent quickly to an MQTT platform (example Cervello) when a change occurs in one of them
- For the first four inputs, the modem must be configured to send via MQTT any state change that occurs in them (input enabled (to ground) / input disabled (open)). In the last four digital inputs, it should only be sent via MQTT when there is an activation on one input (ground input)
- To avoid sending a large number of messages (in case of rapid changes in certain inputs), a timeout of 10 seconds should be established for inputs GPIO4 and GPIO5. This implies that no matter how many changes occur in GPIO4 and GPIO5, no more than 1 MQTT message can be sent in a 10 second window

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_TPProtocol: ntp	Time synchronization protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
SMS_allPhones: on	IP by SMS authorized
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_ATResponse: on	SMS AT responses activated
MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker. Protocol format://url:port
MQTT_id: [IMEI]	Device ID in broker
MQTT_login:	Username

MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	Topic of the MTX to receive AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends its answers to AT commands
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
MQTT_defaultIOQos: 0	Qos for defaultIOTopic
MQTT_defaultIOTopic: IOCHANGE	MQTT topic to send quick telemetries
GPIO_mode0: input	GPIO0 configured as input
GPIO_config0: mqtt;2;0	GPIO0 MQTT configuration
GPIO_mode1: input	GPIO1 configured as input
GPIO_config1: mqtt;2;0	GPIO1 MQTT configuration
GPIO_mode2: input	GPIO2 configured as input
GPIO_config2: mqtt;2;0	GPIO2 MQTT configuration
GPIO_mode3: input	GPIO3 configured as input
GPIO_config3: mqtt;2;0	GPIO3 MQTT configuration
GPIO_mode4: input	GPIO4 configured as input
GPIO_config4: mqtt;2;0	GPIO4 MQTT configuration
GPIO_mode5: input	GPIO5 configured as input
GPIO_config5: mqtt;2;0	GPIO5 MQTT configuration

Details:

- The configuration of the inputs as “mqtt; 2; 0” indicates the following. Remember that all parameters are separated by semicolons ;

  “mqtt” > The input is configured to send the states of the digital inputs by MQTT

  “2” > The 2 indicates that the digital input is configured to send an MQTT message both for activation of the input (when it closes taking it to ground) and for deactivation of the input (when it opens). If you want to send an MQTT message only when closing the entry (bringing it to ground), you should indicate a value of “1”

  “0” > Indicates the timeout of the digital input. This means that the change of the digital input will be sent whenever it occurs. If, for example, a value of “10” were configured, as is the case with GPIO4 and GPIO5, even if there are multiple activations in the digital input, more than 1 MQTT message will never be sent in those 10 seconds

- The sending format of these messages follows the JSON structure, shown in the following example:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”GPIO”,

  “TS”:” 2020-02-08T18:35:15Z”

  “ID”:”0”,

  “VALUE”:1,

  “DIR”:”INPUT”

  }

  Where:

  - IMEI: indicates the IMEI of the MTX modem

  - TYPE: indicates the type of frame. “GPIO” is for digital I / O frames

  - TS: Timestamp (unix format specified in MTX_TPFormat)

  - ID: indicates the index of the GPIO (0 = GPIO0, 1 = GPIO1 ..., 7 = GPIO7)

  - VALUE: indicates the value of the input (0,1)

  - DIR: indicates the type of pin (INPUT / OUTPUT)

- The data of the digital inputs / outputs configured as “mqtt” are sent to the topic configured in the parameter “MQTT_defaultIOTopic”

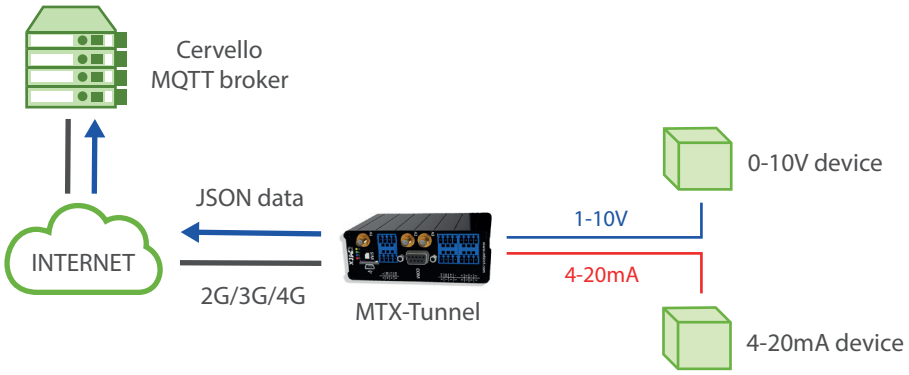
8.11 EXAMPLE: Instant sending of telemetries via MQTT of up to 2 analog inputs 0-10V and 4-20mA.

Scenario details:

- 2 analog sensors need to be monitored, one of the 0-10V type and the other of the 4-20mA type
- The monitoring of the analog sensors will be carried out using MQTT, that is, the modem must send the data from both sensors to an MQTT broker
- In order to save bandwidth, there is no need to continuously send data from the sensors to the MQTT broker. Only the changes will be sent, that is, when the 0-10V input varies by 0.1V or the 4-20mA input varies 0.15mA, these values must be sent quickly to an MQTT platform (such as Cervello)

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_TPProtocol: ntp	Time synchronization protocol
MTX_numGSMErrors: 180	Reset if no registry in GSM network in 1800 secs.
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
SMS_allPhones: on	IP by SMS authorized
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_ATResponse: on	SMS AT responses activated
MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker. Protocol format://url:port
MQTT_id: [IMEI]	Device ID in broker

MQTT_login:	Username
MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	Topic of the MTX to receive AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends its answers to AT commands
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
MQTT_defaultIOQos: 0	Qos for defaultIOTopic
MQTT_defaultIOTopic: IOCHANGE	MQTT topic to send quick telemetries
ADC_mode0: voltage	ADC0 as votage input
ADC_config0: mqtt;100;0	ADC0 MQTT configuration
ADC_mode1: current	ADC1 as current input
ADC_config1: mqtt;150;0	ADC1 MQTT configuration

Details:

- Setting the ADC0 input as “mqtt; 100; 0” indicates the following. Remember that all parameters are separated by semicolons ;

  “mqtt” > The input is configured to send the analog input quickly by MQTT

  “100” > Hysteresis in mV. Indicates how many mV the input must change to send the data to the MQTT broker. This avoids continuously sending data to the broker and excessive and unnecessary consumption of traffic

  “0” > Indicates the timeout of the analog input. This means that the change of the analog input will be sent to the MQTT broker quickly. If, for example, a value of “60” were configured, even if there are multiple changes in the analog input, more than 1 MQTT message will never be sent in those 60 seconds

- The sending format of these messages follows the JSON structure, shown in the following example:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”ADC”,

“TS”:” 2020-02-08T19:15:12Z”

“ID”:0,

“VALUE”:7750

“MODE”:”voltage”

}

Where:

- IMEI: indicates the IMEI of the MTX modem

- TYPE: indicates the type of frame. ADC = analog input

- TS: Timestamp (unix format specified in MTX_TPFormat)

- ID: indicates the ADC index (0 = ADC0.1 = ADC1)

- VALUE: indicates the value of the input (in mV or mA)

- MODE: indicates the working mode of the input (“voltage” / “current”)

- The data of the digital inputs configured as “mqtt” are sent to the topic configured in the parameter “MQTT_defaultIOTopic”

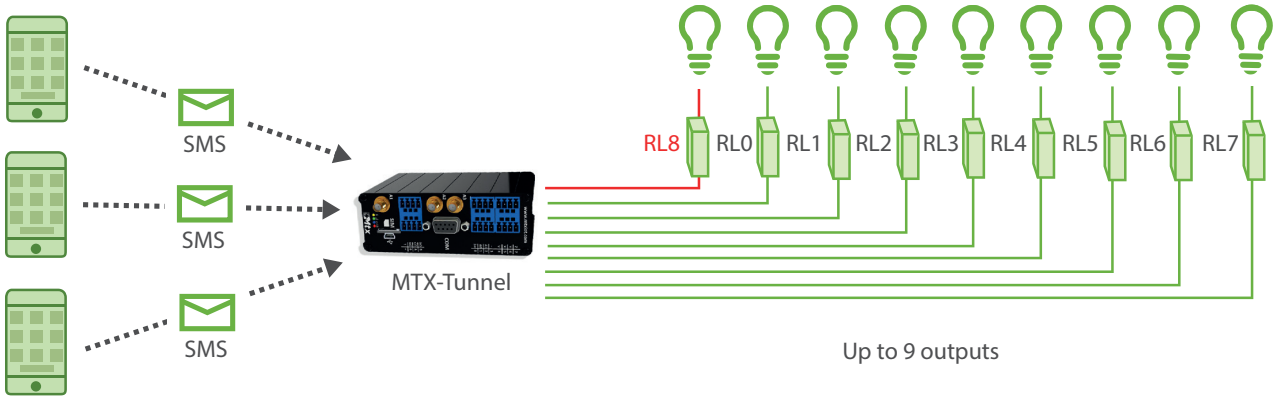
8.12 EXAMPLE: Change by SMS of up to 9 digital outputs or 9 relays.

Scenario details:

- You need to be able to change the status of 9 remote relays to enable / disable a number of devices connected to them. Such remote activation needs to be done by SMS messaging
- It should only be possible to change the status of the relays from the authorized telephone numbers: +34666123456, + 34666123457 and +346661234568
- The SMS messages to make the change of state must be simple. Therefore, to activate Relay 0 the SMS message will be “RL00N” and to deactivate it “RL00FF”. Exactly the same for the rest of the relays: “RL10N” / “RL10FF”, “RL20N” / “RL20FF”, ... “RL80N” / “RL80FF”

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
SMS_allPhones: off	Phone numbers
SMS_validPhone1: +34666123456	Phone number 1
SMS_validPhone2: +34666123457	Phone number 2
SMS_validPhone3: +34666123458	Phone number 3
SMS_atEnabled: on	Remote AT commands
SMS_atResponse: on	Responses to AT commands
SMS_alias1: RL0ON>AT^MTXTUNNEL=SETIO,0,1	Alias to activate the relay RL0
SMS_alias2: RL0OFF>AT^MTXTUNNEL=SETIO,0,0	Alias to deactivate the relay RL0
SMS_alias3: RL1ON>AT^MTXTUNNEL=SETIO,1,1	Alias to activate the relay RL1
SMS_alias4: RL1OFF>AT^MTXTUNNEL=SETIO,1,0	Alias to deactivate the relay RL1
SMS_alias5: RL2ON>AT^MTXTUNNEL=SETIO,2,1	Alias to activate the relay RL2
SMS_alias6: RL2OFF>AT^MTXTUNNEL=SETIO,2,0	Alias to deactivate the relay RL2
SMS_alias7: RL3ON>AT^MTXTUNNEL=SETIO,3,1	Alias to activate the relay RL3
SMS_alias8: RL3OFF>AT^MTXTUNNEL=SETIO,3,0	Alias to deactivate the relay RL3

SMS_alias9: RL4ON>AT^MTXTUNNEL=SETIO,4,1	Alias to activate the relay RL4
SMS_alias10: RL4OFF>AT^MTXTUNNEL=SETIO,4,0	Alias to deactivate the relay RL4
SMS_alias11: RL5ON>AT^MTXTUNNEL=SETIO,5,1	Alias to activate the relay RL5
SMS_alias12: RL5OFF>AT^MTXTUNNEL=SETIO,5,0	Alias to deactivate the relay RL5
SMS_alias13: RL6ON>AT^MTXTUNNEL=SETIO,6,1	Alias to activate the relay RL6
SMS_alias14: RL6OFF>AT^MTXTUNNEL=SETIO,6,0	Alias to deactivate the relay RL6
SMS_alias15: RL7ON>AT^MTXTUNNEL=SETIO,7,1	Alias to activate the relay RL7
SMS_alias16: RL7OFF>AT^MTXTUNNEL=SETIO,7,0	Alias to deactivate the relay RL7
SMS_alias17: RL8ON>AT^MTXTUNNEL=SETIO,8,1	Alias to activate the relay RL8
SMS_alias18: RL8OFF>AT^MTXTUNNEL=SETIO,8,0	Alias to deactivate the relay RL8
SMS_aliasResponse: result	Not obtain as a response AT command executed
GPIO_mode0: output	GPIO0 configured as an output
GPIO_config0: normal	Configuration as a normal output
GPIO_mode1: output	GPIO1 configured as an output
GPIO_config1: normal	Configuration as a normal output
GPIO_mode2: output	GPIO2 configured as an output
GPIO_config2: normal	Configuration as a normal output

GPIO_mode3: output	GPIO3 configured as an output
GPIO_config3: normal	Configuration as a normal output
GPIO_mode4: output	GPIO4 configured as an output
GPIO_config4: normal	Configuration as a normal output
GPIO_mode5: output	GPIO5 configured as an output
GPIO_config5: normal	Configuration as a normal output
GPIO_mode6: output	GPIO6 configured as an output
GPIO_config6: normal	Configuration as a normal output
GPIO_mode7: output	GPIO7 configured as an output
GPIO_config7: normal	Configuration as a normal output
GPIO_mode8: output	GPIO8 configured as an output
GPIO_config8: normal	Configuration as a normal output

Details:

- The MTX-IOT-S family modems have up to 8 digital outputs (from GPIO0 to GPIO7) and 1 integrated relay (GPIO8). The 8 digital outputs are open collector type with enough current to switch an external relay
- Aliases, for convenience, are not case sensitive. That is, it is the same to send “RL1ON” as “RI1ON” as “RL1on” or “rl1on”, among other combinations
- If for your application you only need to manage a low power relay (up to 1Amp), the use of GPIO8 is recommended, since it is connected to an internal relay of the MTX-IoT-S

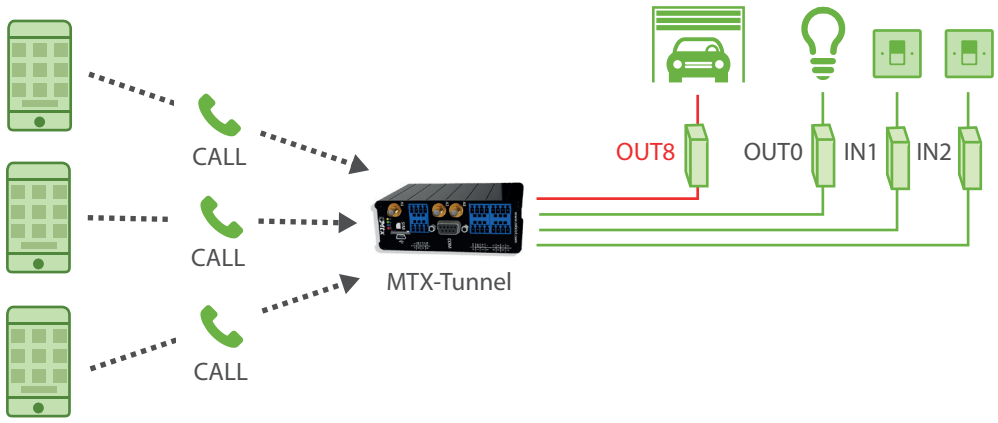
8.13 EXAMPLE: Activate a relay by voice call and by dry contact inputs to open the parking gate.

Scenario details:

- You need to be able to operate a relay with a 5 second pulse to open a motorized garage door. Such activation needs to be done by voice call through a mobile phone
- It should only be possible to operate the relay from the authorized telephone numbers: +34666123456, +34666123457 and +346661234568
- It must also be possible to operate the relay to open the garage door through 2 buttons located inside the garage. That is, when one of these buttons is pressed, the relay that activates the garage door must pulse for 5 seconds, in the same way as when receiving a voice call.
- Finally, both when receiving the call and when pressing the buttons, a luminaire must be activated for 60 seconds to allow good maneuverability for the vehicle driver

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
MTX_mode: 2g	Mode to receive audio calls
MTX_defaultPrefix: +34	Default prefix for national calls
SMS_allPhones: off	Phone numbers
SMS_validPhone1: +34666123456	Phone number 1
SMS_validPhone2: +34666123457	Phone number 2
SMS_validPhone3: +34666123458	Phone number 3
SMS_atEnabled: on	Remote AT commands
SMS_atResponse: on	Responses to AT commands
GPIO_mode0: output	GPIO0 configured as an output
GPIO_config0: timer	Configuration as temporized output
GPIO_mode1: input	GPIO1 configured as an input
GPIO_config1: at;AT^MTXTUNNEL=EXECUTE,doorlight.txt;	Configuration as AT input
GPIO_mode2: input	GPIO2 configured as an input
GPIO_config2: at;AT^MTXTUNNEL=EXECUTE,doorlight.txt;	Configuration as AT input
GPIO_mode8: input	GPIO8 mandatory output (relay)
GPIO_config8: call;AT^MTXTUNNEL=EXECUTE,light.txt	Configuration as call output

Details:

- The MTX-IOT-S family modems have up to 8 digital I / Os (from GPIO0 to GPIO7) and 1 integrated relay (GPIO8). The digital outputs are open collector type with sufficient current to switch an external relay. In this example 1 gpio has been configured as output (GPIO0) and gpios GPIO1 and GPIO2 as inputs. Obviously the GPIO8 that corresponds to an internal relay of the MTX, is compulsorily configured as an output

- In this example GPIO8 (the internal relay of the MTX) is used for door control. The GPIO0, configured as an output, is the one used to activate an external control relay for the luminaire. GPIO0 and GPIO1 are inputs, used to control button 1 and button 2 respectively

- GPIO_config0: timer. This configuration configures the GPIO0 as TIMER OUTPUT, that is, as a timed output

- GPIO_config8: call; AT ^ MTXTUNNEL = EXECUTE, light.txt. This configuration configures the GPIO8 as OUTPUT CALL. This implies that when the MTX-IOT-S modem receives a voice call from an authorized telephone number, it will activate said GPIO8 output for 5 seconds. Additionally it will execute the command AT ^ MTXTUNNEL = EXECUTE, light.txt

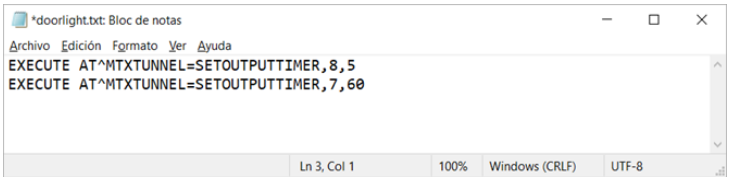
  What this command does is execute the file found in the atscripts / light.txt directory, which is made up of the AT commands to execute. Said atscripts / light.txt file must have the following content



  That is, the modem will execute these AT commands (there is actually only one). The command AT ^ MTXTUNNEL = SETOUTPUTTIMER, 7.60 will activate GPIO7 (the output connected to the luminaire) for 60 seconds

- GPIO_config1: at; AT ^ MTXTUNNEL = EXECUTE, doorlight.txt ;. This configuration configures GPIO1 as INPUT AT. This implies that when the MTX-IOT-S modem activates the GPIO1 input (which has button 1 connected, bringing that input to GND when pressed) it will execute the command AT ^ MTXTUNNEL = EXECUTE, doorlight.txt

  What this command does is execute the file found in the atscripts / doorlight.txt directory, which is made up of the AT commands to execute. Said atscripts / doorlight.txt file must have the following content:



  In other words, the modem will execute these two AT commands. The command AT ^ MTXTUNNEL = SETOUTPUTTIMER, 8.5 will activate GPIO8 (the output connected to the motorized door) for 5 seconds AT ^ MTXTUNNEL = SETOUTPUTTIMER, 7.60 will activate GPIO7 (the output connected to the luminaire) for 60 seconds

- It is possible to modify the authorized phone numbers by sending an SMS message to the MTX modem. For this it is necessary to send an SMS message from an authorized phone number with the text, for example:

mtxtunnel at^mtxtunnel=setparam,SMS_validPhone1,+34666333444

- The previous header marked in red (mtxtunnel) can be customized with the text you want, for this see the configuration parameter SMS_header

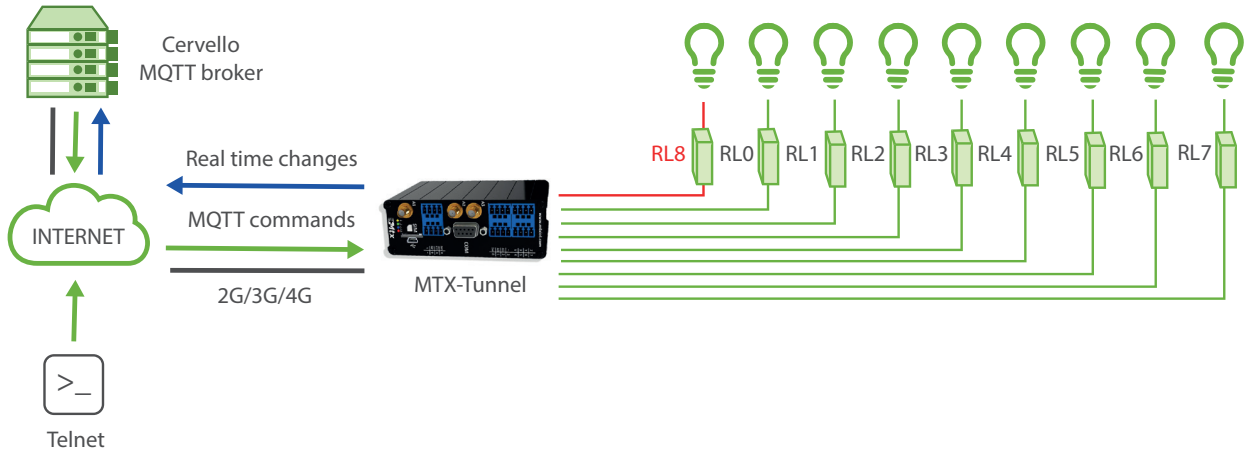
8.14 EXAMPLE: Remote management of up to 9 digital outputs and relays through Telnet and/or MQTT/S.

Scenario details:

- You need to be able to change the status of 9 remote relays to enable / disable a number of devices connected to them. Such remote activation needs to be done through telnet and / or an MQTT / S platform.
- For telnet access, only TCP connections must be allowed from authorized IPs 1.2.3.4 and 1.2.3.5
- In the event of a Telnet change in the status of an exit, said change must be transmitted immediately to the MQTT platform so that the new status is reflected in the MQTT platform dashboard so that an operator can consult the current status

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_numGSMErrors: 180	Reset if no registry on GSM network in 1800 secs.
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
FIREWALL_enabled: off	Authorized IPs
TELNET_enabled: on	Telnet service
TELNET_login: user	Telnet username
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet TCP port
MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker, format protocol://url:port

MQTT_id: [IMEI]	Device ID in broker
MQTT_login:	Username
MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MTX topic to recieve AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends responses to AT commands
MQTT_persistent: off	Persistence
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
MQTT_defaultOTopic: /IOCHANGE	MQTT topic to inform of output changes in real time
MQTT_defaultIOQos: 1	MQoS to inform of output changes in real time
GPIO_mode0: output	GPIO0 configured as an output
GPIO_config0: normal	GPIO0 configured as a normal output
GPIO_mode1: output	GPIO1 configured as an output
GPIO_config1: normal	GPIO1 MQTT configuration
GPIO_mode2: output	GPIO2 configured as an output
GPIO_config2: normal	GPIO2 MQTT configuration
GPIO_mode3: output	GPIO3 configured as an output
GPIO_config3: normal	GPIO3 MQTT configuration
GPIO_mode4: output	GPIO4 configured as an output
GPIO_config4: normal	GPIO4 MQTT configuration
GPIO_mode5: output	GPIO5 configured as an output
GPIO_config5: normal	GPIO5 MQTT configuration

GPIO_mode6: output	GPIO6 configured as an output
GPIO_config6: normal	GPIO6 MQTT configuration
GPIO_mode7: output	GPIO7 configured as an output
GPIO_config7: normal	GPIO7 MQTT configuration
GPIO_mode8: output	GPIO8 configured as an output
GPIO_config8: normal	GPIO8 MQTT configuration

Details:

- The MTX-IOT-S family modems have up to 8 digital outputs (from GPIO0 to GPIO7) and 1 integrated relay (GPIO8). The 8 digital outputs are open collector type with enough current to switch an external relay. In this example all GPIOs have been configured as OUTPUT
- If for your application you only need to manage a low power relay (up to 1Amp), the use of GPIO8 is recommended, since it is connected to an internal relay of the MTX-IOT-S
- To remotely change the status of an output it must be done using an AT command sent remotely via Telnet and/or MQTT. The command to send is AT ^ MTXTUNNEL = SETIO, X, Y where X indicates the GPIO to act on (0... 8) and Y indicates the value of the output (0 = not activated/1 = activated)
- To send the AT command to the modem via MQTT you must do it on the topic configured in MQTT_attopic1. Remember that if you configure something like [IMEI] / AT the modem will replace that text [IMEI] with its real IMEI, that is, for example for something like 354033091487838/AT. The modem will send the response to the command to the TOPIC specified in the MQTT_atropic parameter, which in the case of this example is [IMEI] / ATR
- 5 When configuring the MQTT_defaultIOTopic parameter, the modem will report in that topic MQTT in real time of any change in the output GPIOs. For this reason, every time an output is changed from Telnet, the modem will send a JSON to said MQTT topic informing of the new status

  The sending format of these messages follows the JSON structure, shown in the following example:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”GPIO”,

  “TS”:” 2020-02-08T19:55:11Z”

  “ID”:0,

  “VALUE”:1

contact@webdyn.com | webdyn.com
V11.19 subject to changes | Webdyn © by Flexitron Group

“DIR”:”OUTPUT”

}

Where:

- IMEI: indicates the IMEI of the MTX modem

- TYPE: indicates the type of frame. GPIO = Digital input / output

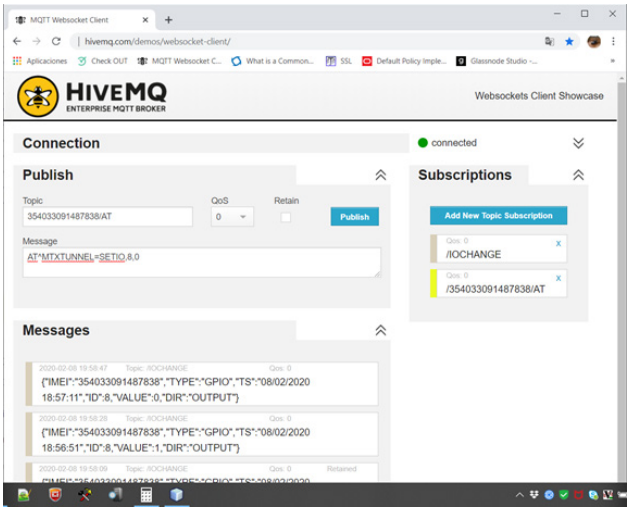
- TS: Timestamp (unix format specified in MTX_TPFormat)

- ID: indicates the index of the GPIO (0 = GPIO0, 1 = GPIO1 ..., 7 = GPIO7)

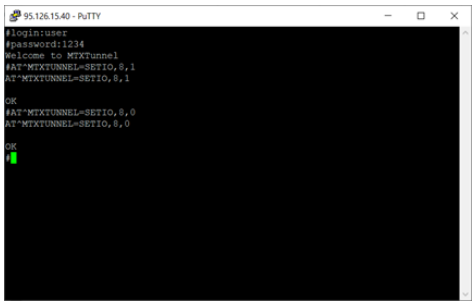
- VALUE: indicates the value of the input (0,1)

- DIR: indicates the type of pin (INPUT / OUTPUT)

- Example of sending AT commands to remotely switch GPIO8 output from an MQTT broker and receiving the AT command response



- Example of sending AT commands to switch GPIO8 output remotely from Telnet



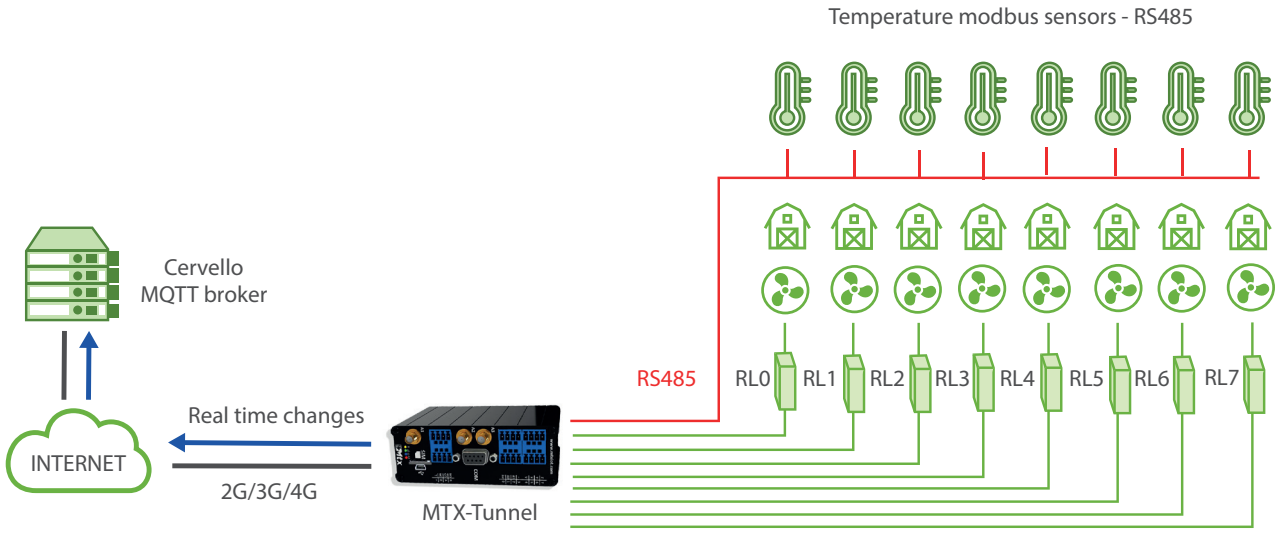
8.15 EXAMPLE: Automatic activation of relays according to the value of Modbus registers. Automatic sending of state changes to the MQTT/S platform.

Scenario details:

- It is necessary to control 8 fans of a farm autonomously. Each fan has an associated Modbus RTU protocol temperature probe, with RS485 connection
- When the Temperature Sensor X is above 30 degrees, the associated Fan X should be activated to facilitate air circulation and lower the temperature of the room. When Temperature Sensor X detects a temperature of 25 degrees, Fan X must be deactivated again. In this way, each farm enclosure will always be between 25 and 30 degrees
- In addition to activating the fans, a MQTT platform must be informed in real time each time a fan is activated / deactivated, in order to have constantly updated information on the application dashboard. Loss of communications with the MQTT control platform must not interfere with the control of the fans

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

COMM2_baudrate: 9600	Data rate of communication of serial port
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	CTS Hardware flow control deactivated
COMM2_autocts: off	RTS Hardware flow control deactivated
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_numGSMErrors: 180	Reset if no registry on GSM network in 1800 secs.
MTX_portAux: modbusmaster	Read modbus
MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker, format protocol://url:port
MQTT_id: [IMEI]	Device ID in broker
MQTT_login:	Username

MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MTX topic to recieve AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends responses to commands
MQTT_persistent: off	Persistence
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
MQTT_defaultOTopic: /IOCHANGE	MQTT topic about output changes in real time
MQTT_defaultIOQos: 1	MQoS to inform of output changes in real time
GPIO_mode0: output	GPIO0 configured as an output
GPIO_config0: modbus;10;3;1;250;300;;	GPIO0 configured as a modbus output
GPIO_mode1: output	GPIO1 configured as an output
GPIO_config1: modbus;11;3;1;250;300;;	GPIO1 configured as a modbus output
GPIO_mode2: output	GPIO2 configured as an output
GPIO_config2: modbus;12;3;1;250;300;;	GPIO2 configured as a modbus output
GPIO_mode3: output	GPIO3 configured as an output
GPIO_config3: modbus;13;3;1;250;300;;	GPIO3 configured as a modbus output
GPIO_mode4: output	GPIO4 configured as an output
GPIO_config4: modbus;14;3;1;250;300;;	GPIO4 configured as a modbus output
GPIO_mode5: output	GPIO5 configured as an output

GPIO_config5: modbus;15;3;1;250;300;;	GPIO5 configured as a modbus output
GPIO_mode6: output	GPIO6 configured as an output
GPIO_config6: modbus;16;3;1;250;300;;	GPIO6 configured as a modbus output
GPIO_mode7: output	GPIO7 configured as an output
GPIO_config7: modbus;17;3;1;250;300;;	GPIO7 configured as a modbus output

Details:

- The MTX-IOT-S family modems have up to 8 digital outputs (from GPIO0 to GPIO7) and 1 integrated relay (GPIO8). The 8 digital outputs are open collector type with enough current to switch an external relay. In this example, all GPIOs from 0 to 7 have been configured as OUTPUT, to control the fans
- If for your application you only need to manage a low power relay (up to 1Amp), the use of GPIO8 is recommended, since it is connected to an internal relay of the MTX-IOT-S
- By configuring the MQTT_defaultOTopic parameter, the modem will report in that MQTT topic in real time of any change in the output GPIOs. For this reason, every time an output is changed from Telnet, the modem will send a JSON to said MQTT topic informing of the new status

  The sending format of these messages follows the JSON structure, shown in the following example:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”GPIO”,

  “TS”:” 2020-02-08T08:12:45Z”

  “ID”:0,

  “VALUE”:1

  “DIR”:”OUTPUT”

  }

  Where:

  - IMEI: indicates the IMEI of the MTX modem

  - TYPE: indicates the type of frame. GPIO = Digital input / output

- TS: Timestamp (unix format specified in MTX_TPFormat)
- ID: indicates the index of the GPIO (0 = GPIO0, 1 = GPIO1 ..., 7 = GPIO7)
- VALUE: indicates the value of the input (0,1)
- DIR: indicates the type of pin (INPUT / OUTPUT)

- The configuration of an output as modbus;10;3;1;250;300;; does:

  modbus: configure the output in modbus mode

  10: GPIO2 output is associated with the modbus temperature probe with address 10

  3: the modbus command to read the temperature register is 3

  1: the register to read from the probe, where it has the temperature is register number 1

  250: if the fan falls below 250 (25 degrees) the GPIO0 will deactivate, turning off the fan

  300: if more than 300 (30 degrees) is read the GPIO0 output will activate, turning on the fan

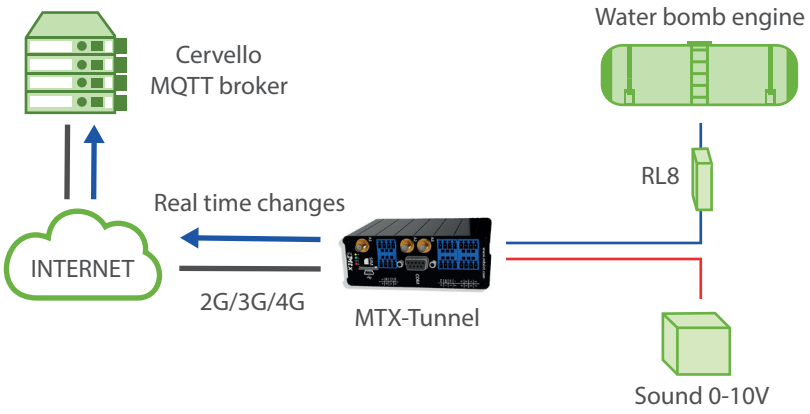
8.16 EXAMPLE: Automatic activation of relay output depending on the value of analog 0-10V probe.

Scenario details:

- It is necessary to control 1 relay to activate the pumping motor of a well
- An analog 0-10V level sensor is available. When the level is low, that is, when the sensor returns a value below 2V, the relay acting on the motor must be activated to pump water. When the sensor returns a value above 8V, the relay must be deactivated to stop pumping water.
- The relay status (on/off) must be sent via 4G/3G/2G to an MQTT broker every time a change occurs. It must also be possible to activate/deactivate the modem relay that controls the well motor at any time from the MQTT broker, that is, remotely. A drop in connectivity with the MQTT broker should not interfere with the behavior of the analog sensor input to the relay

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_numGSMErrors: 180	Reset if no registry on GSM network in 1800 secs.
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker, format protocol://url:port
MQTT_id: [IMEI]	Device ID in broker
MQTT_login:	Username
MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MTX topic to recieve AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends responses to commands

MQTT_persistent: off	Persistence
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
MQTT_defaultOTopic: /IOCHANGE	MQTT topic to inform of output changes in real time
MQTT_defaultIOQos: 1	MQoS to inform of output changes in real time
ADC_mode0: voltage	ADC0 configured as voltage analog input
ADC_config0: at;2000;8000;0;AT ^MTXTUNNEL=SETIO,8,1;AT ^MTXTUNNEL=SETIO,8,0;AT	
GPIO_mode8: output	GPIO8 configured as a mandatory output (relay)
GPIO_config8: normal	GPIO8 configured as a normal output

Details:

- The MTX-IOT-S family modems have up to 8 digital outputs (from GPIO0 to GPIO7) and 1 integrated relay (GPIO8). In this example GPIO8, which is associated with the MTX-IOT-S modem relay, has been configured to enable / disable the motor relay
- The analog input ADC0 of the MTX-IOT-S modem must be configured to read voltage (do not forget to properly configure the microswitches to configure the ADC0 in voltage mode, you will find more information in the Annexes of this manual)
- By configuring the MQTT_defaultIOTopic parameter, the modem will report in that MQTT topic in real time any change in the input / output GPIOs. For this reason, each time the relay associated with GPI8 is activated / deactivated, the modem will send a JSON to said topic MQTT informing of the new status

  The sending format of these messages follows the JSON structure, shown in the following example:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”GPIO”,

  “TS”:” 2020-02-09T12:33:18Z”

  “ID”:”8”,

  “VALUE”:1,

“DIR”:”INPUT”

}

Where:

- IMEI: indicates the IMEI of the MTX modem

- TYPE: indicates the type of frame. “GPIO” is for digital I / O frames

- TS: Timestamp (unix format specified in MTX_TPFormat)

- ID: indicates the index of the GPIO (0 = GPIO0, 1 = GPIO1 ..., 7 = GPIO7)

- VALUE: indicates the value of the input (0,1)

- DIR: indicates the type of pin (INPUT / OUTPUT)

- The configuration of ADC0 input as at;2000;8000;0;AT^MTXTUNNEL=SETIO,8,1; AT^MTXTUNNEL=SETIO,8,0;AT does:

  at: sets the analog input to at, which will cause specific AT commands to be executed when the analog value is above or below a certain threshold

  2000: minimum value (in mV) to execute low voltage AT commands

  8000: maximum value (in mV) to execute the high voltage AT commands

  0: hysteresis (in mV). In this example you don’t need

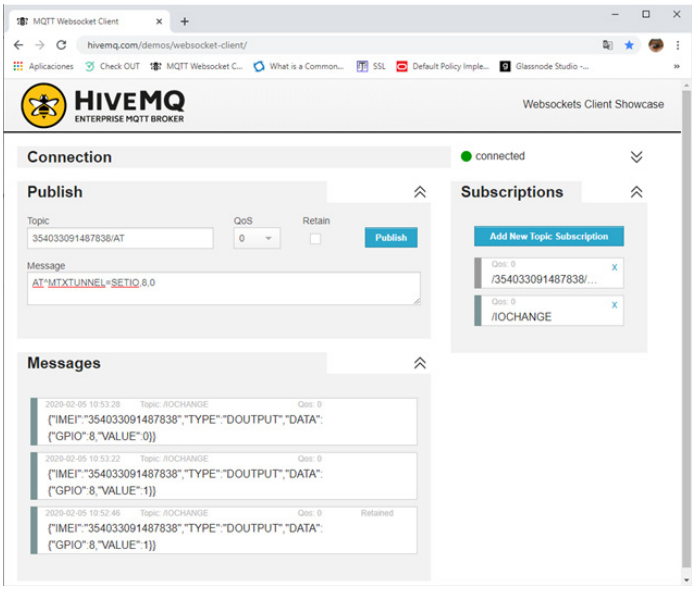
  AT ^ MTXTUNNEL = SETIO, 8.1: the AT command that is executed when the analog input value is below 2000 mV (i.e. the AT command that activates the GPIO8 output (the relay) to activate the motor)

  AT ^ MTXTUNNEL = SETIO, 8.0: the AT command that is executed when the analog input value is above 8000 mV (that is, the AT command that disables the GPIO8 output (the relay) to stop the motor)

  AT: the AT command that is executed when the sensor value is in the normal zone (that is, between 2000mV and 8000mV). In this example it is not used

- To remotely change the state of the GPIO8 (the relay), simply send the command AT ^ MTXTUNNEL = SETIO, 8.1 (to activate the relay) and AT ^ MTXTUNNEL = SETIO, 8.0 (to deactivate the relay) from the MQTT platform.). AT commands must be sent to the TOPIC MQTT configured in the MQTT_attopic1 parameter and the modem will send the responses to the AT commands to the TOPIC MQTT MQTT_atrtopic

Below is an example of sending the command:



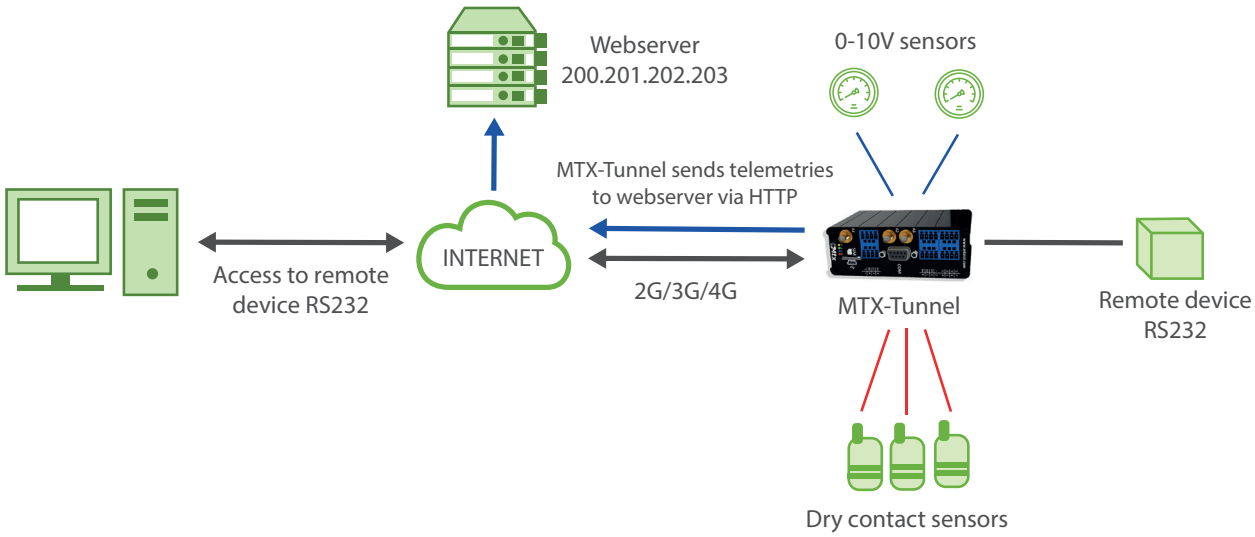
8.17 EXAMPLE: DATALOGGER. Telemetry (digital and analog inputs) sent periodically to the web server via HTTP (via JSON object) + transparent IP-RS232 gateway for remote access to the RS232 device.

Scenario details:

- 3 digital inputs and 2 analog inputs must be sent to a web server via HTTP request using a JSON object
- The MTX will collect digital and analog data every 10 minutes, sending it to the web server via HTTP request using a JSON object. If there is no 4G/3G/2G coverage or the web server is not available, no readings should be lost and the MTX modem must store the read records (a maximum of 1000 records) in its flash memory to try to send it when there is coverage
- Likewise, it is necessary to be able to activate a transparent IP-RS232 gateway in parallel to the telemetries for the remote reading of a device with RS232 serial port at 9600.8, N, 1. For convenience there is a SIM with a fixed IP address in the MTX modem
- The modem must also send its status (coverage, IP, etc.) periodically (every 10 minutes) to the web server via HTTP request

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

COMM_baudrate: 9600	Data rate of communication of serial port
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	CTS Hardware flow control deactivated
COMM_autocts: off	RTS Hardware flow control deactivated
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: server	Gateways used
MTX_urc: off	Every 30 minutes PING check
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
MTX_ping: 35	Time server backup
MTX_pingIP: 8.8.8.8	Unix time format
GPIO_mode0: input	GPIO0 configured as an input for sensor, dry

GPIO_config0: normal	GPIO0 configured as a normal output
GPIO_mode1: input	GPIO1 configured as an output
GPIO_config1: normal	GPIO1 MQTT configuration
GPIO_mode2: input	GPIO2 configured as an output
GPIO_config2: normal	GPIO2 MQTT configuration
ADC_mode0: voltage	ADC0 configured to read voltage
ADC_config0: normal	ADC0 configured as normal
ADC_mode1: voltage	ADC1 configured to read voltage
ADC_config1: normal	ADC1 configured as normal
TCP_port: 20010	MTX TCP port for incoming connections, any IP
FIREWALL_enabled: off	Firewall
LOGGER_enabled: on	Logger status
LOGGER_password: ID00001	Logger password
LOGGER_server: www.miservidorWeb.com/json.asp?data=	URL where JSON data will be sent
LOGGER_registerSize: 300	Size of MTX internal registry
LOGGER_numRegistersFlash: 1000	Max. number of registries in MTX
LOGGER_ioPeriod: 600	Digital/analog inputs of the modem read, sent
LOGGER_httpMode: getjson	Logger mode
DNS_enabled: on	To send status
DNS_server: www.miservidorWeb.com/json.asp?data=	URL to send status data
DNS_password: ID00001	DNS password

DNS_mode: http	HTTP data sending mode
DNS_httpMode: getjson	HTTP GET (JSON) data sending mode
DNS_exdended: on	Also sending GPIOs and ADCs
DNS_period: 600	Period of sending, also when data changes

Details:

- Remember that you have the I / O descriptions of the MTX model in ANNEX 10 of this manual. Be sure to configure the microswitches appropriately to read voltage or current on the AD0 and AD1 inputs of the MTX modem
- The MTX modem will send the value of its digital and analog inputs, the Logger data, every 600 seconds (10 minutes). In the HTTP request that the MTX modem will make to the Web server every 10 minutes, a JSON is included with the following format shown through an example

  {“IMEI”:354033091487838,”TYPE”:”IOS”,”TS”:”2020-02-08T15:35:07Z”,”P”:”ID 00001”,”IO0”:1,”IO1”:0,”IO2”:0,”IO3”:0,”IO4”:0,”IO5”:0,”IO6”:0,”IO7”:0,”IO8”:1-,”AD0”:1200,”AD1”:4850}

  Where:

  IMEI: indicates the IMEI of the modem

  TS: timestamp of when the data was read in the modem

  TYPE: frame type. In this case it is “IOS”

  Q: the field indicated in LOGGER_password

  IOx: digital input/output x. see Annex 10 for more information. x = 0... 8

  ADx: analog input 1 and 2 (values from 0 to 50,000 mV)

  In the same way, the modem status data, DNS frames, sent every DNS_period seconds to the web server via HTTP, will have the following format:

  {“IMEI”:”354033091487838”,”TYPE”:”DNS”,”TS”:”2020-02-08T10:23:13Z”,”P”:”ID00001”,”IP”:”95.124.213.236”,”CSQ”:18,”TECH”:”4G”,”VER”:”11.00”,”AUX”:””,”MOD”:”MTX-IOT-4G-S”,”VCC”:12000}

  Where:

  IMEI: indicates the IMEI of the modem

  TS: timestamp with modem time

  TYPE: frame type. In this case it is “DNS”

  Q: the field indicated in DNS_password

IP: current IP of the modem

CSQ: RSSI (0... 31)

TECH: technology (2G, 3G, 4G)

SEE: FW MTX-Tunnel version

AUX: auxiliary field defined in DNS_aux

MOD: model of the modem (MTX_model field)

VCC: MTX supply voltage (in millivolts)

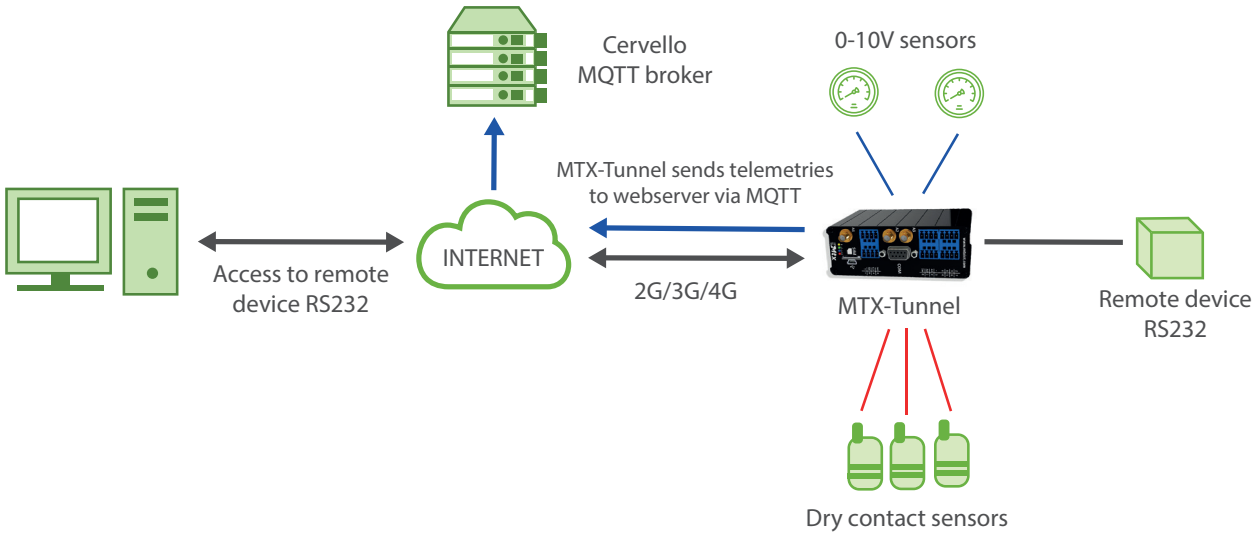
8.18 EXAMPLE: DATALOGGER. Telemetry (digital and analog inputs) sent periodically to broker MQTT (via JSON object) + transparent IP-RS232 gateway for remote access to RS232 device.

Scenario details:

- 3 digital inputs and 2 analog inputs must be sent to a web server via HTTP request using a JSON object
- The MTX will collect the digital and analog data every 10 minutes, sending it to an MQTT broker using a JSON object. If there is no 4G/3G/2G coverage or the web server is not available, no readings should be lost and the MTX modem must store the read records (a maximum of 1000 records) in its flash memory to try to send it when there is coverage
- Likewise, it is necessary to be able to activate a transparent IP-RS232 gateway in parallel to the telemetries for the remote reading of a device with RS232 serial port at 9600.8, N, 1. For convenience there is a SIM with a fixed IP address in the MTX modem
- The modem must also send its status (coverage, IP, etc.) periodically (every 10 minutes) to the MQTT broker

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

COMM_baudrate: 9600	Data rate of communication of serial port
COMM_bitsperchar: 8	Number of bits
COMM_autorts: off	CTS Hardware flow control deactivated
COMM_autocts: off	RTS Hardware flow control deactivated
COMM_stopbits: 1	1 stop bit
COMM_parity: none	No parity
GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: server	Gateways used
MTX_urc: off	Every 30 minutes PING check
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
MTX_ping: 35	Time server backup
MTX_pingIP: 8.8.8.8	Unix time format
MQTT_enabled: on	MQTT service

MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker, format protocol://url:port
MQTT_id: [IMEI]	Device ID in broker
MQTT_login:	Username
MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MTX topic to recieve AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends responses to commands
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
GPIO_mode0: input	GPIO0 configured as input for dry contact sensor
GPIO_config0: normal	GPIO0 normal
GPIO_mode1: input	GPIO1 configured as input for dry contact sensor
GPIO_config1: normal	GPIO1 normal
GPIO_mode2: input	GPIO2 configured as input for dry contact sensor
GPIO_config2: normal	GPIO2 normal
ADC_mode0: voltage	ADC0 configured to read voltage
ADC_config0: normal	ADC0 normal
ADC_mode1: voltage	ADC1 configured to read voltage
ADC_config1: normal	ADC1 normal
TCP_port: 20010	TCP port to receive IP connections
FIREWALL_enabled: off	Firewall status
LOGGER_enabled: on	Logger status to store readings

LOGGER_registerSize: 300	Size of internal MTX registry
LOGGER_numRegistersFlash: 1000	Max. number of registries
LOGGER_mode: mqtt	Logger sending mode
LOGGER_mqttTopic: /LOGGER	Topic to send modem data
LOGGER_ioPeriod: 600	Period to read send digital/analog inputs
DNS_enabled: on	DNS to send status
DNS_mode: mqtt	Sending mode
DNS_exdended: on	Also sending GPIOs and ADCs
DNS_period: 600	Period of sending, also when data changes
DNS_mqttTopic: /DNS	Topic to send status data

Details:

- Remember that you have the I / O descriptions of the MTX model in ANNEX 10 of this manual. Be sure to configure the microswitches appropriately to read voltage or current on the AD0 and AD1 inputs of the MTX modem
- The MTX modem will send the value of its digital and analog inputs, the Logger data, every 600 seconds (10 minutes). In the MQTT sending that the MTX modem will carry out the topic configu

  {“IMEI”:354033091487838,”TYPE”:”IOS”,”TS”:”2020-02-08T15:35:07Z”,”P”:”ID 00001”,”IO0”:1,”IO1”:0,”IO2”:0,”IO3”:0,”IO4”:0,”IO5”:0,”IO6”:0,”IO7”:0,”IO8”:1-,”AD0”:1200,”AD1”:4850}

  Where:

  IMEI: indicates the IMEI of the modem

  TS: timestamp of when the data was read in the modem

  TYPE: frame type. In this case it is “IOS”

  Q: the field indicated in LOGGER_password

  IOx: digital input/output x. see Annex 10 for more information. x = 0... 8

  ADx: analog input 1 and 2 (values from 0 to 50,000 mV)

In the same way, the modem status data, DNS frames, sent every DNS_period seconds to the web server via HTTP, will have the following format:

{“IMEI”:”354033091487838”,”TYPE”:”DNS”,”TS”:”2020-02-08T10:23:13Z”,”P”:”ID00001”,”IP”:”95.124.213.236”,”CSQ”:18,”TECH”:”4G”,”VER”:”11.00”,”AUX”:””,”MOD”:”MTX-IOT-4G-S”,”VCC”:12000}

Where:

IMEI: indicates the IMEI of the modem

TS: timestamp with modem time

TYPE: frame type. In this case it is “DNS”

Q: the field indicated in DNS_password

IP: current IP of the modem

CSQ: RSSI (0... 31)

TECH: technology (2G, 3G, 4G)

SEE: FW MTX-Tunnel version

AUX: auxiliary field defined in DNS_aux

MOD: model of the modem (MTX_model field)

VCC: MTX supply voltage (in millivolts)

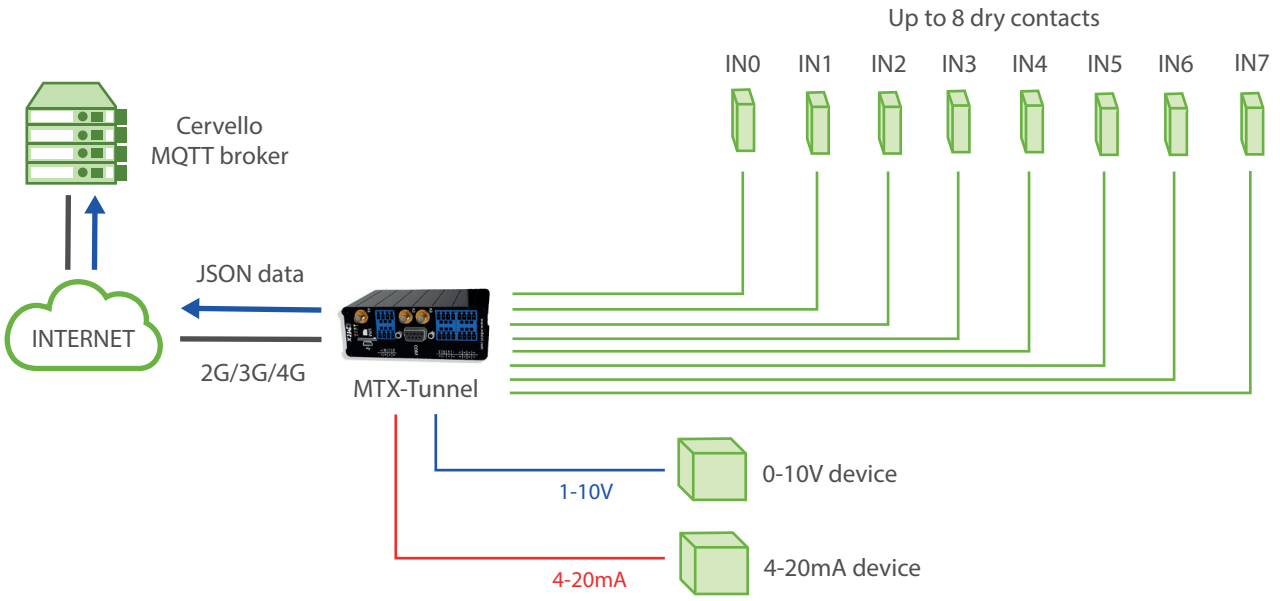
8.19 EXAMPLE: JSON customization and sending topics in the data frames sent by MTX-Tunnel.

Scenario details:

- 8 dry contact sensors need to be monitored. In the status of the entries, they must be sent quickly to an MQTT platform (example Cervello) when a change occurs in one of them. It is also necessary to monitor 2 analog sensors, one of the 0-10V type and the other of the 4-20mA type. Every time the 0-10V sensor varies 100mV or the 4-20mA sensor varies 0.15mA, the value of the sensors should be sent to the MQTT broker
- The status of the modem (coverage, used technology, current IP, etc.) should also be sent periodically, every 60 seconds
- The MQTT broker needs to receive the telemetries in certain topics, as well as the JSON format must have a specific format, so the MTX-Tunnel must be configured appropriately to allow adjusting these requirements

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	Gateways used
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_numGSMErrors: 180	Reset if no registry on GSM network in 1800 secs.
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
SMS_allPhones: on	IP by SMS authorized
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_ATResponse: on	SMS AT responses activated
MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker, format protocol://url:port
MQTT_id: [IMEI]	Device ID in broker

MQTT_login:	Username
MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MTX topic to recieve AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends responses to AT commands
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
MQTT_defaultIOQos: 1	MQoS to inform of output changes in real time
MQTT_defaultOTopic: /IOCHANGE	MQTT topic to inform of output changes in real time
DNS_enabled: on	DNS to send status
DNS_mode: mqtt	Sending mode
DNS_exdended: on	Also sending GPIOs and ADCs
DNS_period: 60	Period of sending, also when data changes
DNS_mqttTopic: DNS	Topic to send status data
GPIO_mode0: input	GPIO0 configured as an input
GPIO_config0: mqtt;2;0	GPIO0 MQTT configuration
GPIO_mode1: input	GPIO1 configured as an input
GPIO_config1: mqtt;2;0	GPIO1 MQTT configuration
GPIO_mode2: input	GPIO2 configured as an input
GPIO_config2: mqtt;2;0	GPIO2 MQTT configuration
GPIO_mode3: input	GPIO3 configured as an input
GPIO_config3: mqtt;2;0	GPIO3 MQTT configuration
GPIO_mode4: input	GPIO4 configured as an input

GPIO_config4: mqtt;2;0	GPIO4 MQTT configuration
GPIO_mode5: input	GPIO5 configured as an input
GPIO_config5: mqtt;2;0	GPIO5 MQTT configuration
GPIO_mode6: input	GPIO6 configured as an input
GPIO_config6: mqtt;2;0	GPIO6 MQTT configuration
GPIO_mode7: input	GPIO7 configured as an input
GPIO_config7: mqtt;2;0	GPIO7 MQTT configuration
ADC_mode0: voltage	ADC0 as voltage input
ADC_config0: mqtt;100;0	ADC0 MQTT configuration
ADC_mode1: current	ADC0 as current input
ADC_config1: mqtt;150;0	ADC1 MQTT configuration

Details:

- The configuration of the inputs as “mqtt; 2; 0” indicates the following. Remember that all parameters are separated by semicolons ;

  “mqtt” > The input is configured to send the states of the digital inputs by MQTT

  “2” > The 2 indicates that the digital input is configured to send an MQTT message both by activating the input (when it closes, bringing it to ground) and by deactivating the input (when it opens). If you want to send an MQTT message only when closing the entry (bringing it to ground), you should indicate a value of “1”

  “0” > Indicates the timeout of the digital input. This means that the change of the digital input will be sent whenever it occurs. If, for example, a value of “10” were configured, as is the case with GPIO4 and GPIO5, even if there are multiple activations in the digital input, more than 1 MQTT message will never be sent in those 10 seconds

- The standard format for sending Digital input messages follows the JSON structure shown in the following example:

  {

  “IMEI”:”354033091487838”,

  “TYPE”:”GPIO”,

  “TS”:” 2020-02-08T18:35:15Z”

  “ID”:”0”,

“VALUE”:1,

“DIR”:”INPUT”

}

Where:

- IMEI: indicates the IMEI of the MTX modem

- TYPE: indicates the type of frame. “GPIO” is for digital I / O frames

- TS: Timestamp (unix format specified in MTX_TPFormat)

- ID: indicates the index of the GPIO (0 = GPIO0, 1 = GPIO1 ..., 7 = GPIO7)

- VALUE: indicates the value of the input (0,1)

- DIR: indicates the type of pin (INPUT / OUTPUT)

- The standard sending format of the messages related to analog inputs follow the JSON structure shown in the following example:

{

“IMEI”:”354033091487838”,

“TYPE”:”ADC”,

“TS”:” 2020-02-08T19:15:12Z”

“ID”:0,

“VALUE”:7750

“MODE”:”voltage”

}

Where:

- IMEI: indicates the IMEI of the MTX modem

- TYPE: indicates the type of frame. ADC = analog input

- TS: Timestamp (unix format specified in MTX_TPFormat)

- ID: indicates the ADC index (0 = ADC0.1 = ADC1)

- VALUE: indicates the value of the input (in mV or mA)

- MODE: indicates the working mode of the input (“voltage” / “current”)

- The data of the digital inputs / outputs configured as “mqtt” are sent to the topic configured in the parameter “MQTT_defaultIOTopic” and QoS specified in the parameter “MQTT_

defaultIOQos”

- DNS frames, state frames, would have a format like the one shown in the following example:

  {“IMEI”:”354033091487838”,”TYPE”:”DNS”,”TS”:”2020-02-09T13:02:24Z”,”P”:””,”IP”:”95.126.2.167”,”CSQ”:9,”TECH”:”4G”,”VER”:”11.00”,”AUX”:””,”MOD”:”MTX-IOT-4G-S”}

- Customizing the JSON and the Topics. Now imagine that you want to send the DNS frames in the json format:

  {“data”:

  {“IMEI”:”354033091487838”,”TYPE”:”DNS”,”TS”:”2020-02-09T13:04:25Z”,”P”:””,”IP”:”95.126.2.167”,”CSQ”:9,”TECH”:”4G”,”VER”:”11.00”,”AUX”:””,”MOD”:”MTX-IOT-4G-S”,”VCC”:12000}

  }

  We also want to customize the frames of the digital inputs and the analog inputs in the same way.

  {“data”:

  {“IMEI”:”354033091487838”,”TYPE”:”ADC”,”TS”:”2020-02-09T13:01:21Z”,”ID”:1,”VALUE”:22774,”MODE”:”current”}

  }

  And we also want to send DNS frames to the topic “topicDNS”, and each GPIO and ADC to a certain topic, for example “topicGPIO0”, “topicGPIO1”, “topicGPIO2”, …. , “TopicGPIO8”, “topicADC0” and “topic ADC1”

  Well, for this, the following configuration structure must be added to the file config.txt:

  JSON_config1: {“TYPE”:”DNS”,”MQTT”:{“TOPIC”:”topicDNS”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config2:{“TYPE”:”GPIO0”,”MQTT”:{“TOPIC”:”topicGPIO0”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config3: {“TYPE”:”GPIO1”,”MQTT”:{“TOPIC”:”topicGPIO1”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config4: {“TYPE”:”GPIO2”,”MQTT”:{“TOPIC”:”topicGPIO2”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config5: {“TYPE”:”GPIO3”,”MQTT”:{“TOPIC”:”topicGPIO3”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config6: {“TYPE”:”GPIO4”,”MQTT”:{“TOPIC”:”topicGPIO4”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config7: {“TYPE”:”GPIO5”,”MQTT”:{“TOPIC”:”topicGPIO5”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config8: {“TYPE”:”GPIO6”,”MQTT”:{“TOPIC”:”topicGPIO6”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

  JSON_config9: {“TYPE”:”GPIO7”,”MQTT”:{“TOPIC”:”topicGPIO7”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config10:{“TYPE”:”GPIO8”,”MQTT”:{“TOPIC”:”topicGPIO8”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config11: {“TYPE”:”ADC0”,”MQTT”:{“TOPIC”:”topicADC0”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config12: {“TYPE”:”ADC1”,”MQTT”:{“TOPIC”:”topicADC1”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

With the previous configuration we managed to encapsulate the standard JSON sent by the MTX-Tunnel, in another custom JSON. Let’s break down an example:

JSON_config1: {“TYPE”:”DNS”,”MQTT”:{“TOPIC”:”topicDNS”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

This setting indicates:

TYPE: Tipo de trama que se pretende formatear. En este caso la trama “DNS”.

MQTT: Parámetros de MQTT que se utilizarán para dicha trama (en caso de no especificar el JSON

MQTT, se toma como topic el especificado en DNS_mqttTopic y el QoS será “0” para esta trama

FORMAT: Indica el formato que se pretende enviar. MTX-Tunnel utilizará el formato indica, SUBSTITUYENDO el texto indicado en “JSON-MTXTUNNEL” (comillas incluidas) por la trama original DNS que utiliza el MTX-Tunnel.

This setting indicates:

{“data”:”JSON-MTXTUNNEL”}}

The red highlighted text will be replaced by the standard JSON for DNS frames.

{“data”:

{“IMEI”:”354033091487838”,”TYPE”:”DNS”,”TS”:”2020-02-09T13:04:25Z”,”P”:””,”IP”:”95.126.2.167”,”CSQ”:9,”TECH”:”4G”,”VER”:”11.00”,”AUX”:””,”MOD”:”MTX-IOT-4G-S”,”VCC”:12000}

}

Another example:

JSON_config2:{“TYPE”:”GPIO0”,”MQTT”:{“TOPIC”:”topicGPIO0”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

This configuration indicates:

TYPE: Type of frame to be formatted. In this case the “GPIO0” frame. (Note that for GPIO and ADC frames, the index of the associated entry must be specified in TYPE, i.e. GPIOx and ADCx)

MQTT: MQTT parameters to be used for this frame (in case you don’t specify the JSON

MQTT, for I / O frames, the one specified in MQTT_defaultIOTopic is taken as topic and the QoS will be the one specified in MQTT_defaultIOQos for this frame.

FORMAT: Indicates the format to be sent. MTX-Tunnel will use the format indicated,

SUBSTITUTING the text indicated in “JSON-MTXTUNNEL” (quotes included) by the original DNS frame used by the MTX-Tunnel

That is, for this new plot format:

{“data”:”JSON-MTXTUNNEL”}}

The red highlighted text will be replaced by the standard JSON for DNS frames.

{“data”:

{

“IMEI”:”354033091487838”,”TYPE”:”GPIO”,”TS”:” 2020-02-08T18:35:15Z”,”ID”:”0”,

“VALUE”:1,”DIR”:”INPUT”

}

}

MTX-Tunnel uses various frame types: DNS, GPIOx, ADCx as seen in the previous paragraphs, but also sends other types of data such as IOS (I / O datalogger), GPS (GPS positioning), MODB (modbus data), TEMP (temperature probe readings), POWER (external power status) and SERIAL (serial datalogger)

All JSON types can be customized, including TOPIC and QoS (in case of using MQTT), adding in the file config.txt, the following configurations in the JSON_config parameters

JSON_config13: {“TYPE”:”IOS”,”MQTT”:{“TOPIC”:”topicIOS”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config14: {“TYPE”:”GPS”,”MQTT”:{“TOPIC”:”topicGPS”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config15:{“TYPE”:”MODB”,”MQTT”:{“TOPIC”:”topicMODB”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config16: {“TYPE”:”TEMP”,”MQTT”:{“TOPIC”:”topicTEMP”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config17: {“TYPE”:”POWER”,”MQTT”:{“TOPIC”:”topicPOWER”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

JSON_config18:{“TYPE”:”SERIAL”,”MQTT”:{“TOPIC”:”topicSERIAL”,”QOS”:1},”FORMAT”:{“data”:”JSON-MTXTUNNEL”}}

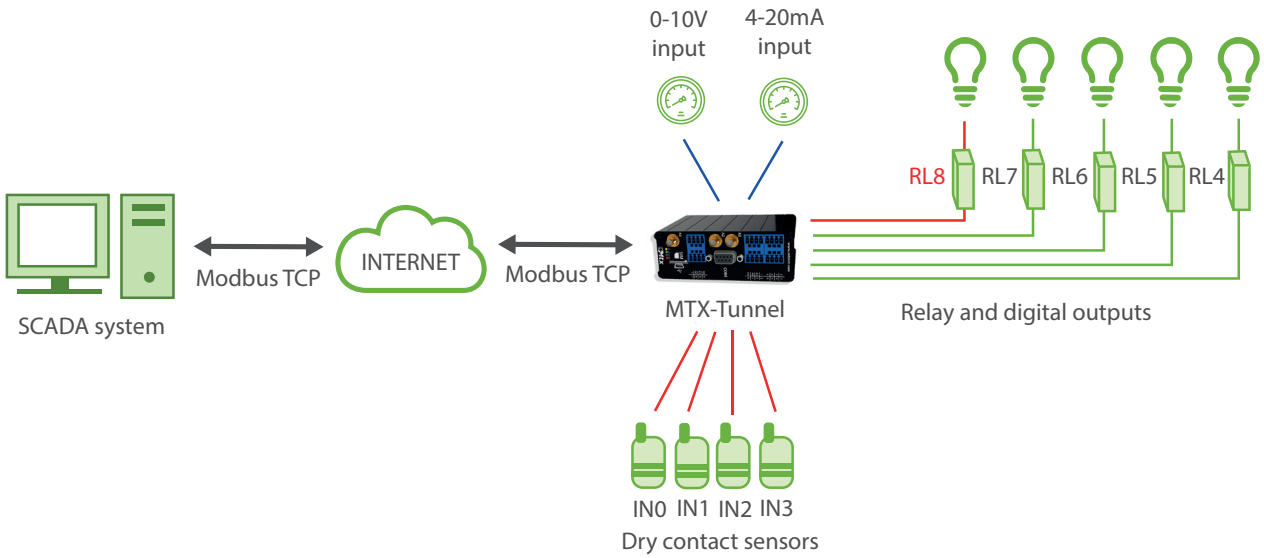
8.20 EXAMPLE: Control of the relays, digital outputs, digital inputs and 0-10V/4-20mA analog inputs of an MTX modem from a SCADA system using the MODBUS TCP protocol.

Scenario details:

- It is necessary to control 1 relay, 4 digital outputs (also connected to relays), 4 dry contact inputs and 2 analog inputs (one 0-10V type and another 4-20mA) located in a remote location
- Communication must be done from a SCADA system using Modbus TCP protocol
- For simplicity it is intended to use SIM cards with a public IP address
- The modem must be remotely configurable via SMS, Telnet and even Modbus

Solution:

Modem MTX-IoT [4-S-N-N]-STD-P + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_ping: 35	Every 35 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MODBUSTCP_enabled: on	Google IP (f.e.) to ping
MODBUSTCP_port: 502	Reset if no registry on GSM network in 1800 secs.
MODBUSTCP_password: ABCD	Time synch protocol
SMS_allPhones: on	IP by SMS authorized
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_allPhones: on	SMS AT responses activated
FIREWALL_enabled: off	Firewall status
TELNET_enabled: on	Telnet status
TELNET_login: user	Telnet login
TELNET_password: 1234	Telnet password
TELNET_port: 20023	Telnet port
GPIO_mode0: input	GPIO0 configured as an input

GPIO_config0: normal	GPIO0 normal configuration
GPIO_mode1: input	GPIO1 configured as an input
GPIO_config1: normal	GPIO1 normal configuration
GPIO_mode2: input	GPIO2 configured as an input
GPIO_config2: normal	GPIO2 normal configuration
GPIO_mode3: input	GPIO3 configured as an input
GPIO_config3: normal	GPIO3 normal configuration
GPIO_mode4: input	GPIO4 configured as an input
GPIO_config4: normal	GPIO4 normal configuration
GPIO_mode5: input	GPIO5 configured as an input
GPIO_config5: normal	GPIO5 normal configuration
GPIO_mode6: input	GPIO6 configured as an input
GPIO_config6: normal	GPIO6 normal configuration
GPIO_mode7: input	GPIO7 configured as an input
GPIO_config7: normal	GPIO7 normal configuration
GPIO_mode8: input	GPIO8 configured as an input
GPIO_config8: normal	GPIO8 normal configuration
ADC_mode0: voltage	ADC0 as voltage input
ADC_config0: normal	ADC0 normal configuration
ADC_mode1: voltage	ADC0 as voltage input
ADC_config1: normal	ADC1 normal configuration

Details:

- For simplicity, in this example the firewall has not been used, but remember that it has the FIREWALL_ parameters that could be used for greater communications security, allowing only communication with the Titan router from the public IP address of Scada.
- Remember that in the tables in Annex A of this manual you will find a table with the I / O of each modem model. Next to them you will find the modbus addresses of each one. For example, from the example model:

ID	@MODBUS	READING COMMAND	WRITING COMMAND
GPIO0	1	0x03	0x10
GPIO1	2	0x03	0x10
GPIO2	3	0x03	0x10
GPIO3	4	0x03	0x10
GPIO4	5	0x03	0x10
GPIO5	6	0x03	0x10
GPIO6	7	0x03	0x10
GPIO7	8	0x03	0x10
GPIO8 (Relé)	9	0x03	0x10
ADC1	11	0x03	---
ADC2	12	0x03	---

- If you want to activate / deactivate via GPbus a GPIO configured in the config.txt as output, you must write a “0” or a “1” in the associated modbus register (@Modbus)

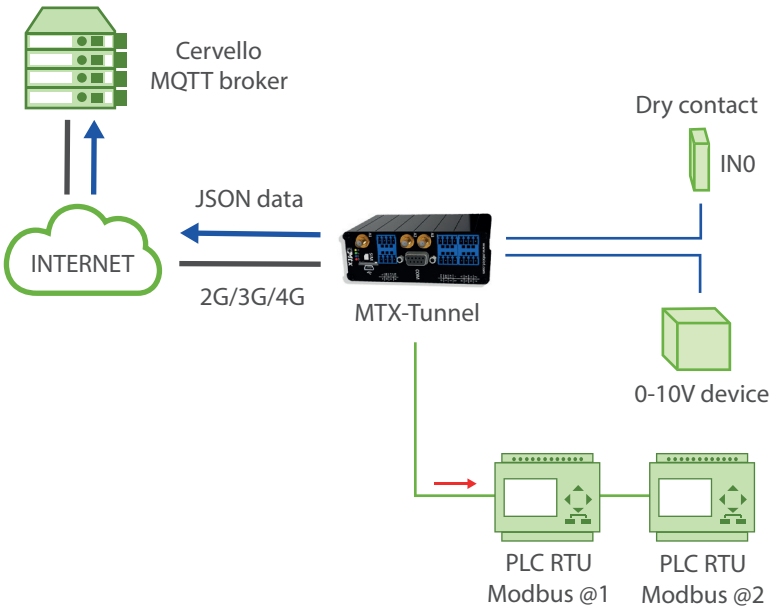
8.21 EXAMPLE: About writing to Modbus RTU devices based on the digital and analog inputs of the modem.

Scenario details:

- 2 sensors need to be monitored. An analog 0-10V sensor and a digital dry contact sensor
- The modem has 2 PLCs with Modbus RTU support connected to its RS485 port. One has the Modbus RTU @ 1 address and the other has the Modbus RTU @ 2 address.
- When changing the state of the dry contact sensor, which is connected to a digital input of the modem, the modem must write a value “1” in register 10 of the PLC @ 1 and PLC @ 2 in case the input and a “0” in case input is disabled
- When the analog sensor has a value> = 5000mV, the modem will write to register 20 of PLC @ 1 and PLC @ 2 a value of “2”. In case the analog input is <= 1000mV it will write a “0” in both registers and otherwise (between 1000mV and 5000mV) it will write a “1” in the Modbus registers of both PLCs
- In each digital event, the value of the digital and analog input must be sent to the MQTT broker

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_numGSMErrors: 180	Reset if no registry on GSM network in 1800 secs.
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
SMS_allPhones: on	IP by SMS authorized
SMS_sendIP: on	IP by SMS authorized
SMS_ATEnabled: on	AT by SMS allowed
SMS_allPhones: on	SMS AT responses activated
MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker, format protocol://url:port
MQTT_id: [IMEI]	Device ID in broker
MQTT_login:	Username

MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MTX topic to recieve AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends responses to commands
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive
GPIO_mode0: input	GPIO0 configured as an input
GPIO_config0: at;AT^MTXTUNNEL=EXECUTE,digitalon.txt;AT^MTXTUNNEL=EXECUTE, digitaloff.txt	GPIO0 AT configuration
ADC_mode0: voltage	ADC0 as voltage input
ADC_config0: at;1000;5000;100;AT^MTXTUNNEL=EXECUTE,analoglow.txt; AT^MTXTUNNEL=EXECUTE,analoghigh.txt;AT^MTXTUNNEL=EXECUTE, analognormal.txt	

Details:

- The configuration of the digital GPIO0 (where it is connected to the dry meter sensor) as “at; AT ^ MTXTUNNEL = EXECUTE, digitalon.txt; AT ^ MTXTUNNEL = EXECUTE, digitaloff.txt” indicates the following. Remember that all parameters are separated by semicolons ;

  “At”> The input is configured to execute an AT command every time it changes state

  “AT ^ MTXTUNNEL = EXECUTE, digitalon.txt”> The second parameter indicates the AT command that will be executed when the digital input is activated. In this case, the AT command script file is located in /atscripts/digitalon.txt

  “AT ^ MTXTUNNEL = EXECUTE, digitaloff.txt”> The third parameter indicates the AT command to be executed when the digital input is deactivated. In this case, the AT command script file is located in /atscripts/digitaloff.txt

  The file “digitalon.txt” will have the following content



The file “digitaloff.txt” will have the following content



The first two lines of this file write in register 10 of the PLCs with address @ 1 and address @ 2 a value of “1” in the first case (when the input is activated) and a “0” in the second case (when input is disabled).

The third line in both files executes an AT command that allows an MQTT message to be sent to the broker. NOTE that MTX-Tunnel substitutes the tags [GPIOx], [ADCx] and [COUNTERx] for ANY COMMAND AT with their corresponding values. In this case a JSON with the value of GPIO0 and current ADC0 is sent to the MQTT broker.

- The configuration of the analog ADC0 (where it is connected to the 0-10V analog sensor) as “at; 1000; 5000; 100; AT ^ MTXTUNNEL = EXECUTE, analoglow.txt; AT ^ MTXTUNNEL = EXECUTE, analoghigh.txt; AT ^ MTXTUNNEL = EXECUTE, analognormal.txt ”indicates the following. Remember that all parameters are separated by semicolons;

  “At”> The input is configured to execute an AT command when a certain condition occurs

  “1000”> Minimum value from which the AT command of minimum value reached will be executed. In this case 1000mV

  “5000”> Maximum value from which the AT command of maximum value reached will be executed. In this case 5000mV

  “100”> Hysteresis. 100mV

  “AT ^ MTXTUNNEL = EXECUTE, analoglow.txt”> AT command to be executed when the analog input is less than or equal to 1000mV. In this case, the AT command script file is run in / atscripts/analoglow.txt

  “AT ^ MTXTUNNEL = EXECUTE, analoghigh.txt”> AT command to be executed when the analog input is greater than or equal to 5000mV. In this case, the AT command script file is located in /atscripts/analoghigh.txt

  “AT ^ MTXTUNNEL = EXECUTE, analognormal.txt”> AT command to be executed when the analog input returns to a normal state between 1000V and 5000mV. In this case, the AT command script file is located in /atscripts/analognormal.txt

  The file “analoglow.txt” will have the following content



  The file “analoghigh.txt” will have the following content

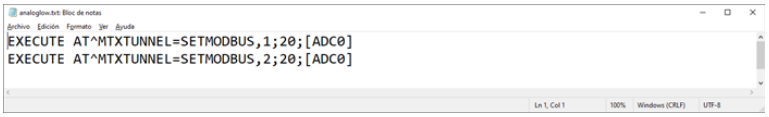


  El fichero “analognormal.txt” tendrá el siguiente contenido



These files write in register 20 of the PLCs with address @ 1 and address @ 2 a value of “2”, “0” or “1” according to the statement in this example

- Note that MTX-Tunnel substitutes the tags [GPIOx], [ADCx] and [COUNTERx] for ANY COMMAND AT with their corresponding values. For example, the current value of the sensor connected to the ADC0 could be written to the PLCs @ 1 and @ 2. To do this, simply replace the value to write with the tag [ADC0], as shown in the following file:



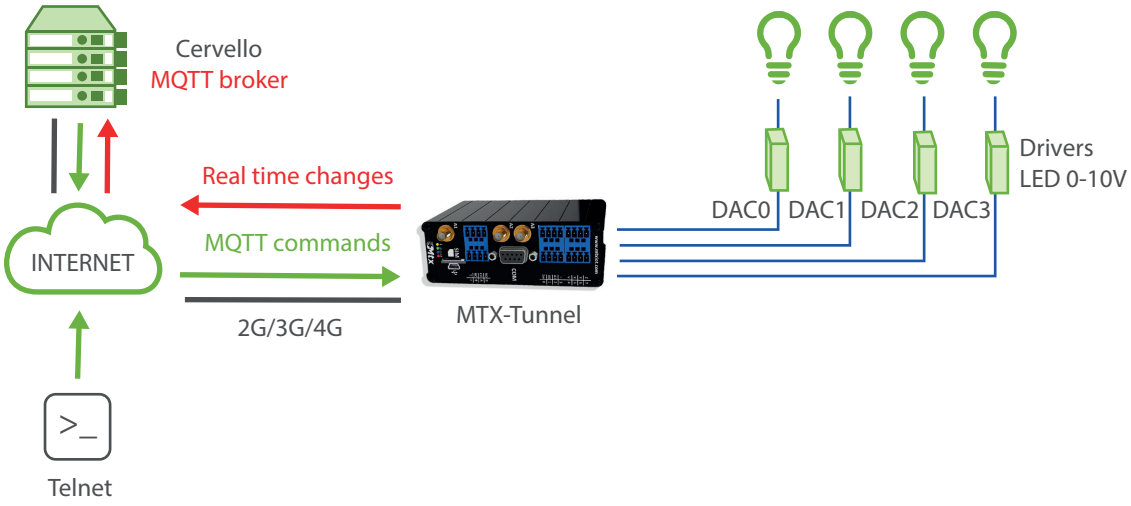
8.22 EXAMPLE: Remote management of up to 4 0-10V analog outputs via Telnet and/or MQTT/S.

Scenario details:

- It is necessary to be able to change the state of 4 remote 0-10V analog outputs to dim some of the luminaires connected to them. Such remote activation must be done through telnet and/or an MQTT/S platform.
- For telnet access, only TCP connections from authorized IPs 1.2.3.4 and 1.2.3.5 must be allowed

Solution:

Modem MTX-IOT-S [4-N] + software MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario:

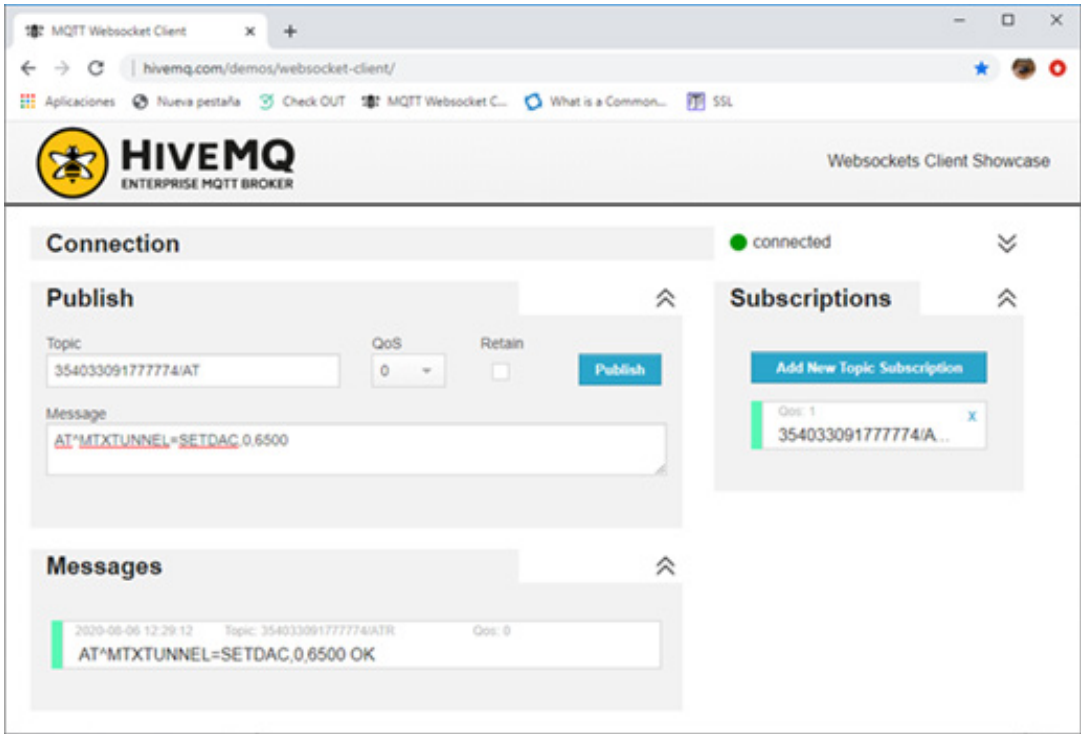
GPRS_apn: movistar.es	GPRS APN provided by GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is permanently connected to GPRS
MTX_pin: 0000	PIN if it has one
MTX_model: 199802407	Device model
MTX_mode: none	No need for serial-IP tunnels
MTX_ping: 30	Every 30 minutes PING check
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_numGSMErrors: 180	Reset if no registry on GSM network in 1800 secs.
MTX_TPProtocol: ntp	Time synch protocol
MTX_TPServer: ntp.roa.es	Time server
MTX_TPServer2: es.pool.ntp.org	Time server backup
MTX_TPFormat: unix	Unix time format
FIREWALL_enabled: on	All IPs are not authorized
FIREWALL_IP1: 1.2.3.4	Authorized IP 1
FIREWALL_IP2: 1.2.3.5	Authorized IP 2
TELNET_enabled: on	We activate the Telnet service
TELNET_login: user	Username
TELNET_password: 1234	Password
TELNET_port: 20023	TCP port

MQTT_enabled: on	MQTT service
MQTT_server: tcp://broker.mqttdashboard.com:1883	MQTT broker, format protocol://url:port
MQTT_id: [IMEI]	Device ID in broker
MQTT_login:	Username
MQTT_password:	Password
MQTT_attopic1: [IMEI]/AT	MTX topic to recieve AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends responses to commands
MQTT_persistent: off	No persistence
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	Keepalive

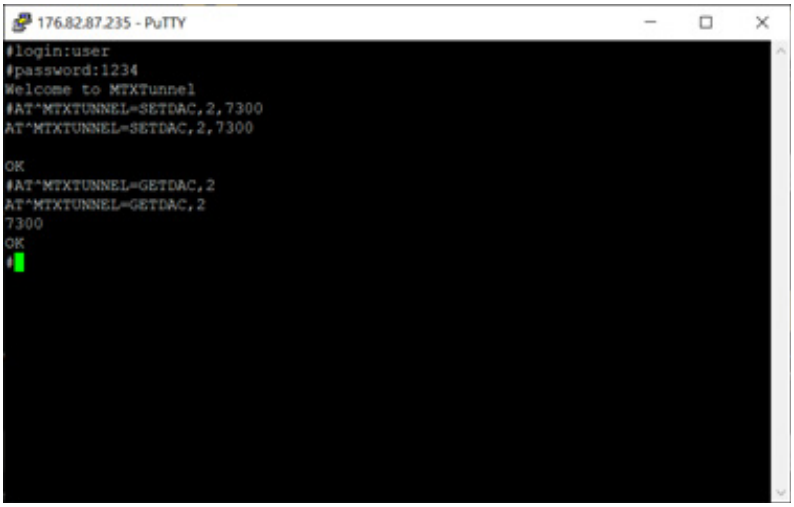
Details:

- MTX-IOT-S family modems have up to 4 0-10V analog outputs (DAC) (from DAC0 to DAC3)
- To remotely change the status of an analog output, it must be done by using an AT command sent remotely via Telnet and / or MQTT (it is also possible via SMS, modbus tcp, etc). The command to send is AT ^ MTXTUNNEL = SETDAC, X, Y where X indicates the DAC on which to act (0... 3) and Y indicates the value of the analog output (0... 10000) expressed in millivolts.
- To send the AT command to the modem via MQTT you must do it on the topic configured in MQTT_attopic1 Remember that if you configure something like [IMEI] / AT, the modem will replace that text [IMEI] with its real IMEI, that is, for example by something like 354033091777774 / AT. The modem will send the response to the command to the TOPIC specified in the MQTT_atropic parameter, which in the case of this example is [IMEI] / ATR

- Example of sending AT commands to remotely change the value of DAC0 to 6.5V from an MQTT broker



- Example of sending AT commands via telnet to change the value of the DAC2 analog output to 7.3V. Also example of querying the set value.



9. ANNEX: MTX-TUNNEL CONNECTION WITH WEB PLATFORMS

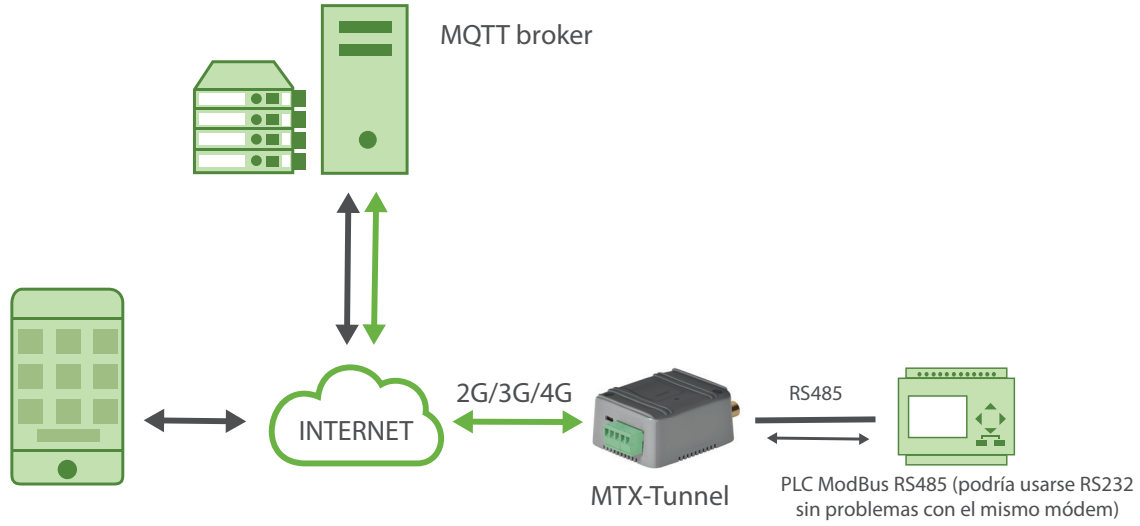
9.1 EXAMPLE: MTX-Tunnel connection with Platform/MQTT broker.

Scenario details:

- We have a PLC Modbus RTU, in its internal memory a series of variables/registries (for instance una temperature, 3 meter boxes, etc.) which must be read and sent periodically to a webserver
- To do that the MTX modem must ask periodically, every minute, for a serial port to the PLC to read those registries. Registries to be read are, for temperature the registry number 20, and the meter boxes are in the registries 21, 22 and 23 respectively
- Also, the MTX modem must connect to an MQTT platform and send the modbus registries read
- The MTX modem can also send information about its status every 15 minutes (information about its IP, coverage, technology used (4G/3G/2G), etc.) to the MQTT platform
- We need to be able to send AT commands to the modem via MQTT in order to change the configuration, remote reset, etc. We also want to be able to send AT commands from a cell phone via MQTT

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file):

COMM2_baudrate: 9600	Data rate of communication of serial port
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	CTS Hardware flow control deactivated
COMM2_autocts: off	RTS Hardware flow control deactivated
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	No gateways
MTX_model: 199801436	Modem is configured as TCP server
MTX_portAux: modbusmaster	AUXILIAR COM port used as master modbus
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: ntp.roa.es	Time server (the MTX must sync the time)
MTX_TPServer2: es.pool.ntp.org	Backup time server
MTX_ping: 35	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_rssiLevel: 10	AWe activate the MTX-65i coverage led
SMS_allPhones: on	IP by SMS authorized

SMS_sendIP: on	SMS AT responses activated
SMS_ATEnabled: on	IP by SMS authorized
SMS_ATResponse: on	AT by SMS allowed
FIREWALL_enabled: off	Firewall disabled
LOGGER_enabled: on	Logger enabled
LOGGER_registerSize: 300	Max. size of MTX internal registries
LOGGER_numRegistersFlash: 1500	Max. number of MTX internal registries
LOGGER_mode: mqtt	MQTT sending mode
LOGGER_mqttTopic: [IMEI]/logger	Sending topic of internal datalogger data
MQTT_enabled: on	MQTT enabled
MQTT_server: tcp://test.mosquitto.org:1883	Broker MQTT to be used
MQTT_id: [IMEI]	Identification of the equipment in the broker
MQTT_login:	No username
MQTT_password:	No password
MQTT_attopic1: [IMEI]/AT	MTX topic to receive AT commands
MQTT_atrtopic: [IMEI]/ATR	Topic where MTX sends answers to commands
MQTT_qos: 1	QoS established
MQTT_keepalive: 60	60 seconds keepalive
MQTT_persistent: off	No persistence
MODBUS_address: 1	Modbus address of the equipment to be read
MODBUS_start: 20	Address of the initial modbus registry to be read
MODBUS_numwords: 4	Number of registries to be read from the initial

MODBUS_readCommand: 3	Reading command
MODBUS_period: 60	A reading is made every 60 seconds
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: [IMEI]/dns	Topic where status data will be sent to
DNS_extended: on	Sending extended data (I/O, ADCs, etc.)
DNS_period: 30	Every 30 seconds a sending will be made

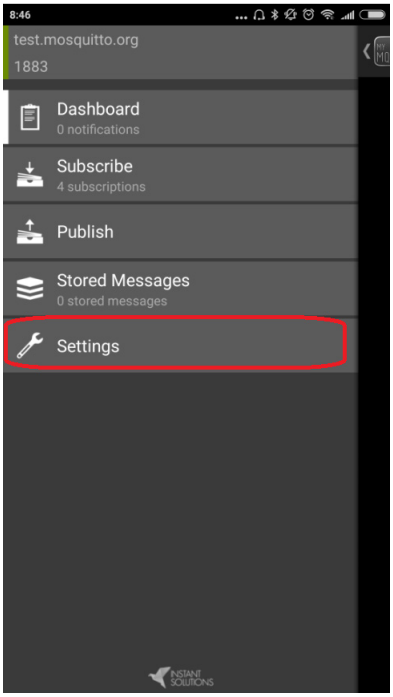
Details:

- The modem reads every 15 minutes the modbus registries of the PLC and sends them via JSON to an MQTT broker (for this test we use a free one: test.mosquitto.org). MTX sends datalogger (where modbus registries read are internally stored) data to MQTT broker, especifically to the topic [IMEI]/logger (MTX substitues the tag [IMEI] for the real IMEI). In case of being unable to send the registrie (no coverage or broker down) it stores the data in its memory to send them later
- It is also possible to send AT commands to the modem via MQTT from a cell phone. To do that, the MTX modem subscribes to the topic indicated in the parameter MQTT_attopic1. All the AT commands sent from a cell phone or any other device to said topic will be received and executed by the modem
- Remember you can specify 3 topics to receive AT commands: MQTT_attopic1, MQTT_attopic2, MQTT_attopic3. For example, we can configure MQTT_attopic1 to receive AT commands exclusively in that modem, MQTT_attopic2 for a several modems, and MQTT_attopic3 for all the modems
- The JSON object sent to the topic LOGGER_mqttTopic is coded like in the following example:

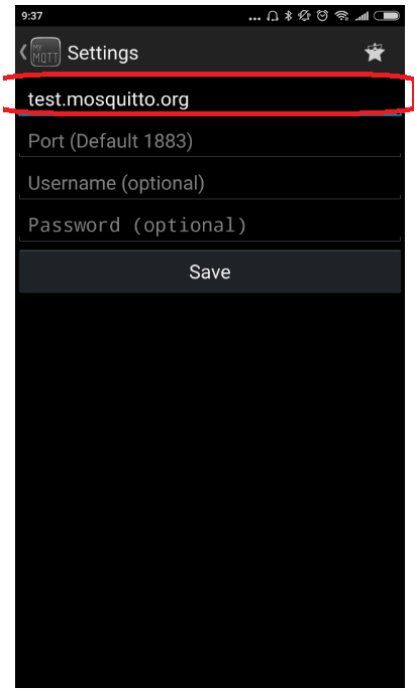
  {“IMEI”:353234028103206,”P”:””,”TYPE”:”MODB”,”A”:1,”TS”:”20/08/12 08:31:44”,”ST”:20,”V1”:23,”V2”:275,”V3”:274,”V4”:32765}

  That is, the web server receives a JSON object with the modem IMEI, the modbus address of the equipment (A), the time stamp (TS) when modbus data has been read, the initial address read (ST) and V1, V2... with each of the variables read.
- To test this example we use the application MyMQTT you can find in Google Play for Android: https://play.google.com/store/apps/details?id=at.tripwire.mqtt.client

Once installed on our phone, we click on “Settings.”



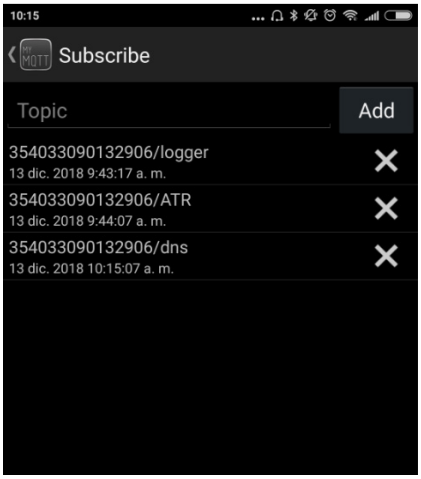
In the section “Settings” we enter the MQTT broker address, in this case test.mosquitto.org in the port 1883. There is no need to specify user or password, since test.mosquitto.org doesn’t allow it. We do it as shown below and press “Save.”



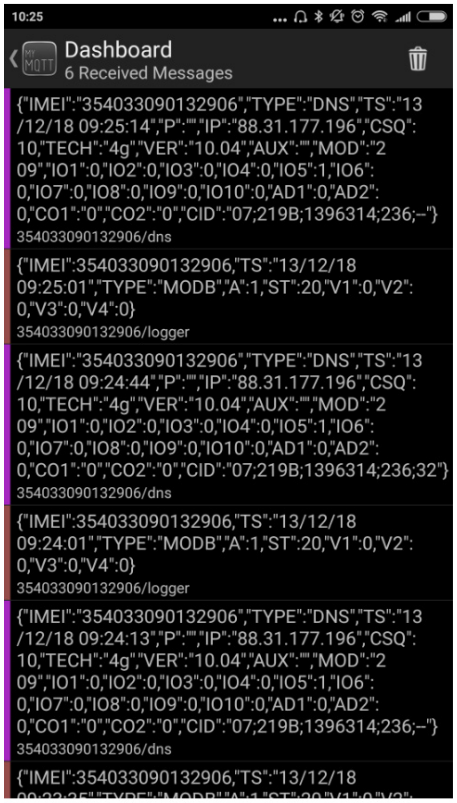
Next we will subscribe to 3 topcs. We want to see the modbus data send by the MTX modem and it is configured to send it to the topic [IMEI]/logger. In the cell phone app we will subscribe to that topic, specifying the IMEI numerically. You can find the MTX modem IMEI in the sticker on the bottom of the box.

Same for status data, that the modem sends to the topic [IMEI]/dns.

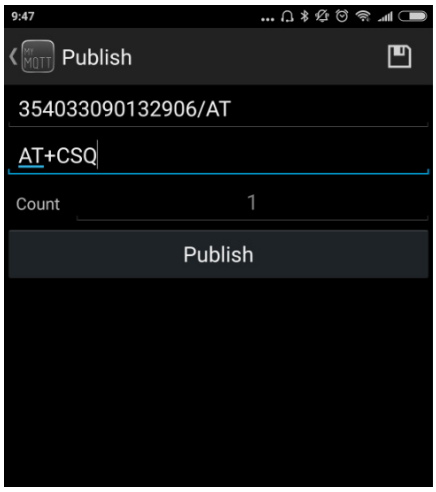
The modem also sends answers to AT commands to the topic [IMEI]/ATR, and since we want to see them, we will enter that topic. We have 3 topics, as seen below:



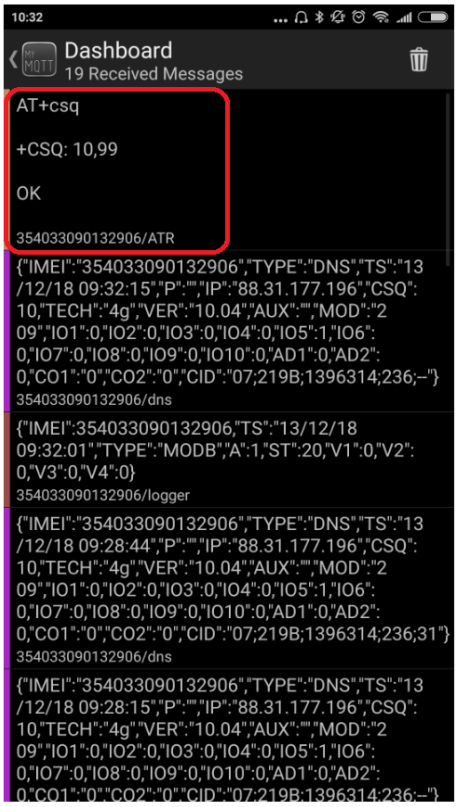
Once these steps are done, if we turn on the MTX modem we will start receiving modbus and DNS data sent by the modem. DNS every 30 seconds, modbus every 2 minutes, as it is configured:



Finally, we will send an AT command to the modem via MQTT from the cell phone to know the GSM coverage (we could send any AT command, for instance to reset, to read the configuration, to change it, to activate a relay, etc.). To do that we go to the menu “Publish” and enter the command as shown below:



Once we press “Publish” the command will be executed on the modem. Since we are subscribed to the answers too, we can see those as well:



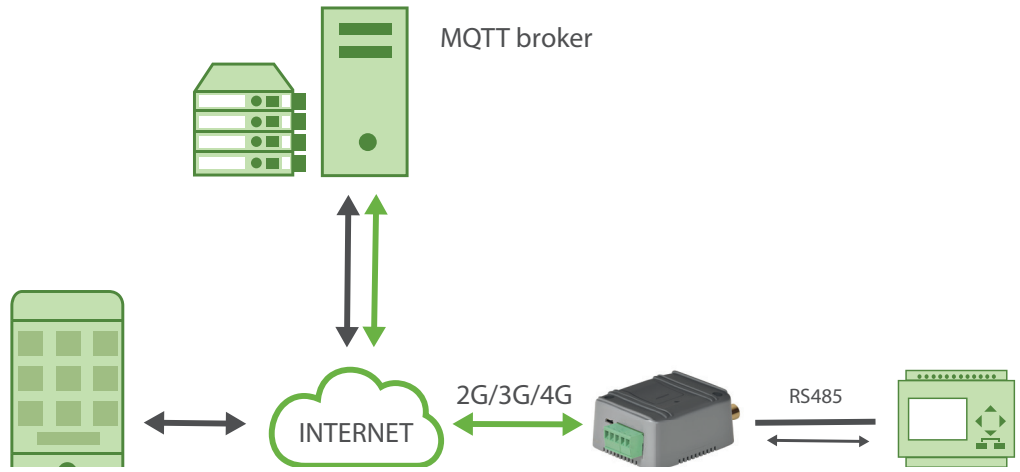
9.2 EXAMPLE: MTX-Tunnel connection with Control Platform via dedicated TCP Socket.

Scenario details:

- There are 100 devices with RS232 port (115200,8,N,1 and HW flow control) which need to be monitored from a Central Control site via 3G.For this purpose MTX-Tunnel will be used, acting as a 4G serial transparent tunnel
- It is necessary to monitor the modem from a Control Platform at any moment. That means, it should be possible to read the status of each modem (for example, coverage reading) and read/change the configuration of all the sets of modems at any time. For this purpose a TCP socket should be installed permanently FROM the modem TO the Control Server (in this sense you will be able to avoid problems with private IPs, etc). The Control Server will be able to send AT commands to the modem through this socket at any moment in order to perform tasks

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file):

COMM2_baudrate: 115200	Data rate of communication of serial port
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: on	CTS Hardware flow control activated
COMM2_autocts: on	RTS Hardware flow control activated
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_model: 199801393	MTX modem model
MTX_mode: server	Modem is configured as TCP server
MTX_urc: off	We do not need URC notification messages
TCP_port: 20010	TCP port of MTX for 3G-RS232 gateway
FIREWALL_enabled: off	Firewall desactivated
LINK_enabled: on	LINK Service activated
LINK_IP: myserver.mydomain.com	Control IP or DNS where modem is connected to
LINK:port: 20020	TCP port of the server specified in LINK_IP
LINK_retryPeriod: 60	Secs before attempt to establish connection

LINK_timeout: 900	Timeout in case no commands are received
LINK_keyId: ID12345678	The first text modem will send via LINK socket

Details:

- Via Link socket (with LINK_parameters) you can send AT commands to modem at any moment
- Specify LINK_retryPeriod increased to avoid high data consumption if connectivity problems
- Use the parameter LINK_keyid to identify the modem that connects to your server
- Sending ATcommand to modem from server: embedded ATcommand (MTX_ATEmbedded: on):
  - Modem coverage reading request from the Server:

    Command sent from the server:

    <MTXTUNNELR>AT+CSQ</MTXTUNNELR>

    Response from the modem:

    <MTXTUNNELR>AT+CSQ +CSQ: 22,99

    OK</MTXTUNNELR>

  - Configuration parameter COMM_baudrate reading request:

    Command sent from the server:

    <MTXTUNNELR>AT^MTXTUNNEL=GETPARAM,COMM_baudrate</MTXTUNNELR>

    Response from the modem:

    <MTXTUNNELR>AT^MTXTUNNEL=GETPARAM,COMM_baudrate

    115200

    OK</MTXTUNNELR>

  - Configuration parameter COMM_baudrate change request. Command sent from server:

    <MTXTUNNELR>AT^MTXTUNNEL=SETPARAM,COMM_baudrate,9600</MTXTUNNELR>

    Response from the modem:

    <MTXTUNNELR>AT^MTXTUNNEL=SETPARAM,COMM_baudrate,9600

    OK</MTXTUNNELR>

9.3 EXAMPLE: MTX-Tunnel configuration modification from a PHP, ASP... webserver.

Scenario details:

There are 1000 devices with RS232 port (115200,8,N,1 and HW flow control) which need to be monitored from a Central Control site via GPRS.For this purpose MTX-Tunnel will be used, acting as a 4G serial transparent tunnel

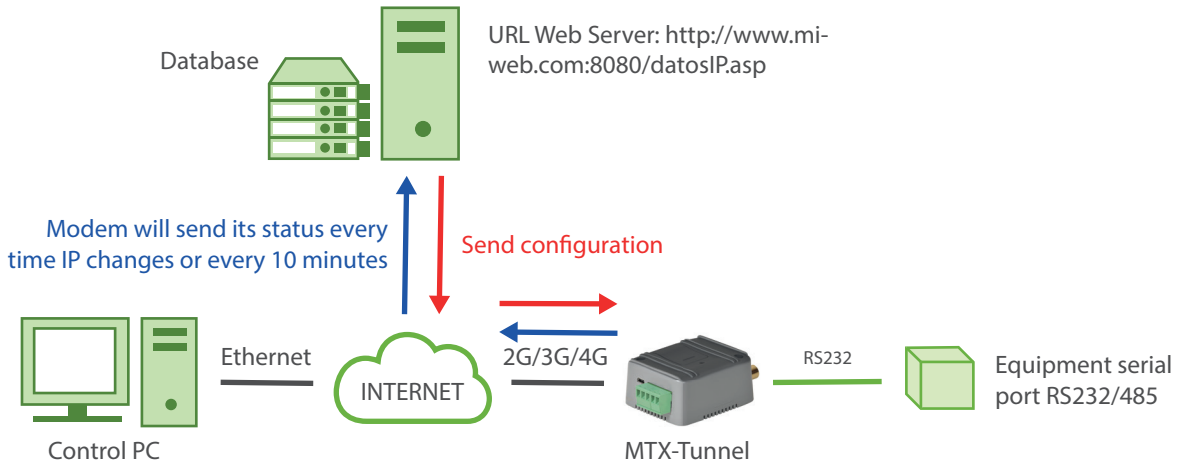
It is important to be able to access RS232 devices at any moment, therefore the modem connected to the serial port of the device should remain connected to 4G permanently waiting for a connection

For reasons of economy, SIM cards with dynamic IP address will be used.Taking into consideration the number of modems DynDNS use is discarded, and it is MTX-Tunnel that should send to a WEB server via HTTP GET (JSON) the IP (assigned by GSM operator) it has at any moment along with its status(coverage,etc.). Every time the modem changes IP address, it should sent to the control center (with URL http://www.miweb.com/datosIP.asp) its new IP to communicate the change. Independently on IP change, the modem will send its status data every 600 seconds (10 minutes)

It is important to be able to change the configuration of any MTX-Tunnel from a WEB Server. Since MTX-Tunnel will be sending its IP every 10 minutes minimum, HTTP request of the modem will be used to send configuration as a response (if there is any new configuration). This way it is very easy to make a massive configuration change of the 1000 modems

Solution:

MTX-IoT [4-S-N-N]-STD-N modem+firmware MTX-Tunnel



EXAMPLE of configuration file (config.txt file):

COMM2_baudrate: 115200	Data rate of communication of serial port
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: on	CTS hardware flow control activated
COMM2_autocts: on	RTS hardware flow control activated
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_model: 199801393	MTX modem model
MTX_mode: server	Modem is configured as TCP server
MTX_urc: off	We do not need URC notification messages
MTX_rssiLevel: 10	Modem coverage LED enabled
MTX_ATEmbedded: on	Embedded AT commands sending allowed
TCP_port: 20010	TCP port of MTX for 3G-RS232 gateway
FIREWALL_enabled: off	Firewall desactivated
DNS_enabled: on	IP to DNS sending service activated
DNS_mode: http	Sending information via HTTP
DNS_httpMode: getjson	HTTP GET JSON mode

DNS_password: 12345678	Password string for higher security
DNS_server: www.miweb.com/datosIP.asp?data=	URL of DNS sever where modem status is sent
DNS_period: 600	Every 10 secs status is sent to the webserver
DNS_aux: 0	Auxiliary field to control configuraion version

Details:

- With this configuration a JSON will be sent via HTTP GET to a specified Web server. It will be sent every time the device is reset, or its IP is changed every 10 minutes (as it happened before). Therefore, at least once every 10 minutes MTX-Tunnel will communicate with the WEB Server sending a JSON similar to the one shown in the following example:

  {“IMEI”:357042060366409,”TYPE”:”DNS”,”P”:”12345678”,”IP”:”88.28.253.206”, ”CSQ”:26, “VER”:”9.12”,”AUX”:”0”,”MOD”:”201”,”VCC”:12000}

  Where:

  IMEI: IMEI of the modem (Unique for each modem)

  TYPE: JSON type sent (DNS in this case)

  P: User field specified in DNS_password

  IP: Current modem IP

  CSQ: rssi of the modem (between 0 and 31)

  VER: MTX-Tunnel version

  AUX: Auxiliary field to control configuraion version

  MOD: MTX modem model

  VCC: MTX supply voltage (in millivolts)

- When our WEB Server receives a JSON, the only thing for us to do is to compare the configuration version sent by MTX-Tunnel (field AUX of JSON) to the version we have on our Web Server. Let’s imagine, for example, that we have changed the configuration of a MTX-Tunnel on the server and indicated that the current configuration version is DNS_AUX: 1. When the MTX-Tunnel sends its JSON, we will receive a value AUX “0”. Since it is diferent from “1” (this is what we have on our server), we will send a new configuration to the moden
- To send the new configuration to the modem we can send an AT command from the WEB server containing something like:

  <MTXTUNNELR>AT^MTXTUNNEL=SETCONFIG,

  COMM_baudrate: 115200

COMM_bitsperchar: 8

COMM_autocts: on

COMM_autorts: on

COMM_stopbits: 1

.....

DNS_password: 12345678

DNS_server: www.miweb.com/datosIP.asp?data=

DNS_period: 600

DNS_aux: 1

</MTXTUNNELR>

That means that, as a response to periodic HTTP transmission of the modem, we send form the WEB server ALL the configuration file containing changes made and specify the value of the new configuration in DNS_aux.Really you are substituting modem's "config.txt" file with the one you are sending.

- Once the new configuratioof is received, the modem will reboot automatically. In few seconds it will communicate with the WEB Platform again to indicate its new IP and will check again the AUX parameter received inside of JSON. If everything has done correctly, the corresponding value of the new configuration will be "1"
- Data transmission from your WEB Platform to the modem should be carried out according to the Server type you work with. For example, if the variable "configuration" contains all the configuration to be sent, from an ASP Server you could use something like:

```
<%
Response.Write(configuration)
%>
```

In a PHP server it could be something like:

```
<?php
echo $configuration;
?>
```

- If you need to test this method, we recommend you to ask us for an account on our test server www.metering.es You will find it useful for your first tests and better understanging of how it works. For that and in case of any doubt send us an email to iotsupport@mtxm2m.com

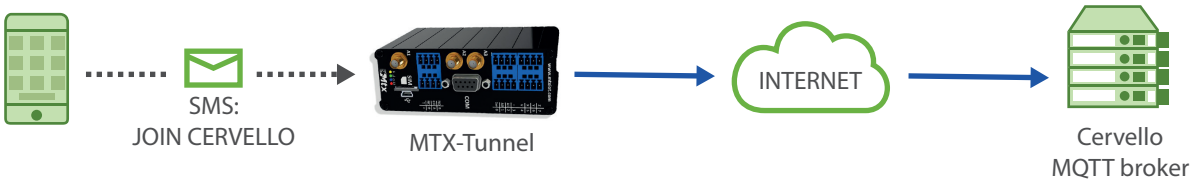
9.4 EXAMPLE: Automatic connection to the Cervello platform via SMS

Scenario details:

- It is necessary to connect the MTX modem to the Cervello Web Platform. The Web Platform will be used as “Device Manager” and through which the status of the MTX modem can be known in real time (coverage, IP, etc.) as well as being able to perform operations on the MTX modem (readings and configuration changes) in a way comfortable and simple
- The operation of connection with the Cervello platform must be a very simple process in order not to require any technical knowledge for it. For this reason, the operator must only supply an MTX modem and send an SMS message to the MTX modem with the text “JOIN CERVELLO”

Solution:

Modem MTX-IOT-S [4-N] + firmware MTX-Tunnel



Configuration example (config.txt file) for the indicated scenario that each MTX-Tunnel must contain (factory configuration):

No special configuration is necessary, just have the factory configuration on the MTX modem.

Details:

- In order to avoid typographical errors, it is possible to send the SMS message in the following ways: “JOIN CERVELLO”, “JOINCERVELLO”, “join cervello” “joincervello”, “Joincervello”, “Join cervello”, etc. (any combination of uppercase and lowercase letters with or without a space between the two words)
- The procedure is as explained in the example statement

  a) The user must connect the antennas, the SIM (without PIN) and power the MTX modem

  b) The user must send the modem an SMS message with the text “JOIN CERVELLO”

c) If everything is correct, the user will receive an SMS message as indicated in the following figure. After a few seconds the modem will auto-reset and connect to Cervello



d) The user can connect to the Cervello Platform, where he will see the MTX modem linked and can make the necessary configuration changes, as well as know the status of the modem

- When sending the sms message “JOIN CERVELLO” the MTX modem is configured to use the generic APN “internet”. If it is necessary to configure a specific APN for the SIM used, it is possible to do it by sending the apn to be used, after a “,” (comma). Example, if we need to use the apn “movistar.es”, the SMS message to send would be:

  JOIN CERVELLO,movistar.es

- If, in addition to the APN, it is necessary to specify a username and password, it is also possible to specify it in the SMS. For example:

  JOIN CERVELLO,movistar.es,miUsername,miPassword

- Another simple option to automatically link an MTX modem with Cervello is simply to configure, within the config.txt configuration file, the parameter:

  CERVELLO_modeAuto: on

10. ANNEX: EXAMPLE SCENARIOS OF READING AND SENDING DATA FROM W-MBUS DEVICES WITH MTX-TUNNEL

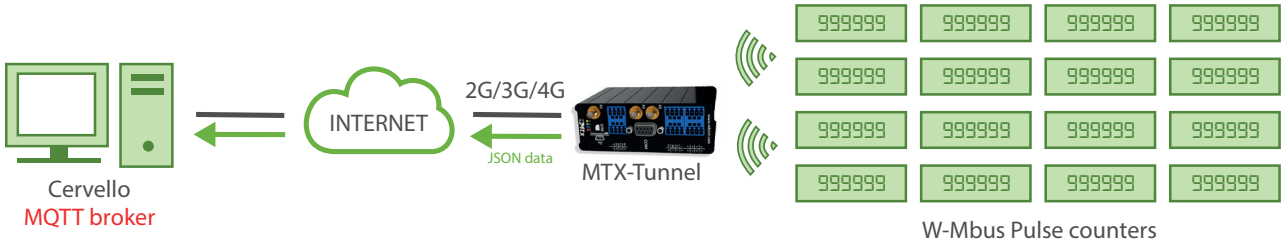
10.1 EXAMPLE: Reading data from W-Mbus water meters and sending it transparently to the MQTT platform. Configuration of time windows and filter by manufacturer.

Scenario details:

- We need to monitor 300 W-Mbus pulse counters from the manufacturer Adeunis
- Each counter emits 1 RF frame with the counting data every 1 minute, but only the W-Mbus concentrator must store and send to the data platform 1 frame every hour, in other words, the concentrator must configure time windows of 1 hour
- The meters will be installed in an urban area with many other W-Mbus devices nearby, so a filter must be configured, in addition to the aforementioned 1 hour window, so that only the meters from a certain manufacturer are read: Adeunis
- The data must be read and stored within the hub's memory and automatically sent to a platform via MQTT using a JSON object whenever there is data coverage. To save data, the W-Mbus frame must be base-64 encoded

Solution:

MTX-IoT-S [4-N] modem+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file):

COMM2_baudrate: 9600	Data rate of communication of serial port
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	CTS Hardware flow control deactivated
COMM2_autocts: off	RTS Hardware flow control deactivated
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	No gateways
MTX_model: 199802407	Modem is configured as TCP server
MTX_portAux: wmbus	Port COMM2 to communicate with WMbus card
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: time1.google.com	Time server (the MTX must sync the time)
MTX_TPServer2: time2.google.com	Backup time server
MTX_ping: 30	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_TPFormat: unit	Unix time format
MTX_numGSMErrors: 180	Reset if no registration in network in 1800 secs

SMS_allPhones: on	Send SMS with commands from any mobile
SMS_sendIP: on	SMS AT responses activated
SMS_ATEnabled: on	IP by SMS authorized
SMS_ATResponse: on	AT by SMS allowed
LOGGER_enabled: on	Logger enabled
LOGGER_registerSize: 1000	Max. size of MTX internal registries
LOGGER_numRegistersRam: 1000	Max. number of MTX internal registries
LOGGER_numRegistersFlash: 0	We don't need to use FLASH memory for logger
LOGGER_mode: mqtt	MQTT sending mode
LOGGER_mqttTopic: /LOGGER	Sending topic of internal datalogger data
MQTT_enabled: on	MQTT enabled
MQTT_server: tcp://broker.release.cervello.io.com:1883	Broker MQTT to be used
MQTT_id: psdjs334jjsd8345	Identification of the equipment in the broker
MQTT_login: 3ddg435g67899	No username
MQTT_password: 2345433456567	No password
MQTT_attopic1: /cervello/devices/[MQTT_ID]/rpc	MTX topic to receive AT commands
MQTT_atrtopic: /cervello/devices/[MQTT_ID]/rpc/response	Topic where MTX sends answers to commands
MQTT_qos: 1	QoS established
MQTT_keepalive: 360	360 seconds keepalive
DNS_enabled: on	Status data sending activated

DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: /DNS	Topic where status data will be sent to
DNS_period: 3600	Every 3600 seconds a sending will be made
WMBUS_mode:9	We configure work mode 9 for Wireless MBus
WMBUS_interval: 60	We set a window of 1 hour (60 minutes)
WMBUS_filter: ARF	We configure a filter for Adeunix devices
WMBUS_data: jsonrawbase64	We configure the data in base 64

Details:

- After the configuration is carried out, the MTX-Tunnel will store in memory a single frame of each W-Mbus device from the manufacturer Adeunis. Said data will be logged in RAM (it could be in flash if necessary modifying the configuration), encapsulated in JSON format and sent to an MQTT broker at topic / LOGGER. The data corresponding to the counter within the JSON, will be encoded in BASE64.

  Example:

  {“IMEI”:”354033091777774”,”TYPE”:”WMBUS”,”TS”:”2020-12-11T08:58:07Z”,”WDATA”:”HURGBioQACABG3IqEAAgRgYBGz8AAAAEZrcAAAASLw==”}

  Where:

  IMEI: the IMEI of the modem sending the data

  TYPE: type of frame sent

  TS: Timestamp of when the W-Mbus frame was collected in the MTX-Tunnel

  WDATA: W-Mbus data in BASE64 format

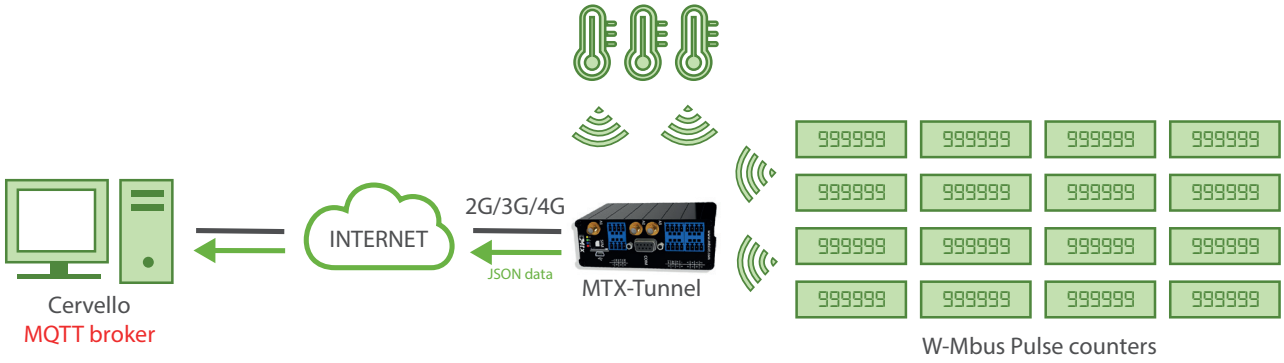
10.2 EXAMPLE: Reading data from W-Mbus water meters and W-MBus temperature sensors. Sending data transparently to the MQTT platform. Configuration of time windows, period and filters by manufacturer and serial number of each device.

Scenario details:

- It is necessary to monitor 300 W-Mbus pulse counters from the manufacturer Adeunis and 30 temperature sensors also from the manufacturer Adeunis
- Each meter emits 1 RF frame with the count data every 1 minute, but only the W-Mbus concentrator must store and send 1 frame every hour to the data platform. The temperature sensors send data every 30 seconds, but we only need to send the temperature every 15 minutes. In other words, the hub must configure time windows of 15 minutes. The temperature will be sent every 1 window and the water meter, every 4 time windows
- The meters will be installed in an urban area with many other W-Mbus devices nearby. Filter by manufacturer and serial number must be configured
- The data must be read and stored within the hub's memory and automatically sent to a platform via MQTT using a JSON object whenever there is data coverage. To save data, the W-Mbus frame must be base-64 encoded

Solution:

MTX-IoT-S [4-N] modem+firmware MTX-Tunnel



EXAMPLE of configuration (config.txt file):

COMM2_baudrate: 9600	Data rate of communication of serial port
COMM2_bitsperchar: 8	Number of bits
COMM2_autorts: off	CTS Hardware flow control deactivated
COMM2_autocts: off	RTS Hardware flow control deactivated
COMM2_stopbits: 1	1 stop bit
COMM2_parity: none	No parity
GPRS_apn: movistar.es	APN GPRS provided by the GSM operator
GPRS_login: MOVISTAR	GPRS Login
GPRS_password: MOVISTAR	GPRS Password
GPRS_timeout: 0	Modem is always GPRS connected
MTX_PIN: 0000	SIM Card PIN
MTX_mode: none	No gateways
MTX_model: 199802407	Modem is configured as TCP server
MTX_portAux: wmbus	Port COMM2 to communicate with WMbus card
MTX_TPProtocol: ntp	Time synch. protocol
MTX_TPServer: time1.google.com	Time server (the MTX must sync the time)
MTX_TPServer2: time2.google.com	Backup time server
MTX_ping: 30	Ping time to oversee connection
MTX_pingIP: 8.8.8.8	Google IP (f.e.) to ping
MTX_TPFormat: unit	Unix time format
MTX_numGSMErrors: 180	Reset if no registration in network in 1800 secs

SMS_allPhones: on	Send SMS with commands from any mobile
SMS_sendIP: on	SMS AT responses activated
SMS_ATEnabled: on	IP by SMS authorized
SMS_ATResponse: on	AT by SMS allowed
LOGGER_enabled: on	Logger enabled
LOGGER_registerSize: 1000	Max. size of MTX internal registries
LOGGER_numRegistersRam: 1000	Max. number of MTX internal registries
LOGGER_numRegistersFlash: 0	We don't need to use FLASH memory for logger
LOGGER_mode: mqtt	MQTT sending mode
LOGGER_mqttTopic: /LOGGER	Sending topic of internal datalogger data
DNS_enabled: on	Status data sending activated
DNS_mode: mqtt	MQTT sending mode
DNS_mqttTopic: /DNS	Topic where status data will be sent to
DNS_period: 3600	Every 3600 seconds a sending will be made
WMBUS_mode:9	We configure work mode 9 for Wireless MBus
WMBUS_interval: 15	We set a window of 15 min.
WMBUS_data: jsonrawhex	We configure the data in base 64 (no compression)

Details:

- The data corresponding to the counter within the JSON, will be encoded in BASE64.

  Example:

  {"IMEI":"354033091777774","TYPE":"WMBUS","TS":"2020-12-11T08:58:07Z","WDATA":"17444606180200100307 7aef8a00002f2f0412c40900001237"}

Where:

IMEI: the IMEI of the modem sending the data

TYPE: type of frame sent

TS: Timestamp of when the WMBus frame was collected in the MTX-Tunnel

WDATA: W-Mbus data in HEX format

- For the correct operation of the example, a file named “wmbus.txt” must be introduced, where it is allowed to introduce different filters, in addition to the sampling period. The file “wmbus.txt” is in CSV format and must be specified like this, indicating a line for each device

  <FAB>,<NUM_SERIE>,<VERSION>,<TIPO>,<PERIODO>

  Where:

  <FAB> (optional): ASCII. It is the name of the manufacturer, see annex D of this manual for the list of names. If not specified, it will not filter by manufacturer (unless WMBUS_filter is used)

  <NUM_SERIE> (required): HEX. It is the serial number of the W-Mbus device

  <VER> (optional): HEX. It is the firmware version of the device. If it is not specified, it is not filtered by this field

  <TYPE> (optional): HEX. It is the type of device. If it is not specified, it is not filtered by this field

  <PERIOD> (optional): DECIMAL. Period for reading frames. For example, if a 15 minute time window (WMBUS_interval: 15) is configured and <PERIOD> has a value of 4, 1 W-MBUS frame from the sensor will be captured every 15x4 = 60 minutes (1 hour). If not specified, consider period 1

  Wmbus.txt file example:

  ARF,10000218,,,4

  ARF,2000102a,,,

  ....

  Line 1 description:

  ARF manufacturer filter (Adeunis), for a device with serial number 10000218, without specifying VERSION filter, without specifying TYPE filter and specifying a period 4 (that is, as WMBus_interval: 15, implies a reading every 15x4 = 60 minutes)

  Line 2 description:

  ARF manufacturer filter (Adeunis), for a device with serial number 2000102a, without specifying VERSION filter, without specifying TYPE filter and without specifying period, so period is 1 (that is, as WMBus_interval: 15, implies a reading every 15x1 = 15 minutes)

- The wmbus.txt file must be entered in the modem in the same way and level as the configuration file “config.txt”, that is, in the root directory of the modem

1. ANNEX A: MTX-TUNNEL I/O DESCRIPTION

1.1 MTX-IoT-S

Description of I/O interfaces:

- Terminal connection: 8 GPIOs configurable by software as Digital Input or as Digital Output, and 2 A/D converters (0-10V in voltage mode. 0-20mA in current mode, configurable by microswitches)
- Serial interfaces: 1 RS232 (DB9) port. and 1 RS485 port
- 1x latch relay 30V 1A
- Internal GPS connected to the secondary serial port (supported on models: MTX-IoT-S [4-N-GPS])
- The MTX-Tunnel software uses the inputs and outputs as follows:

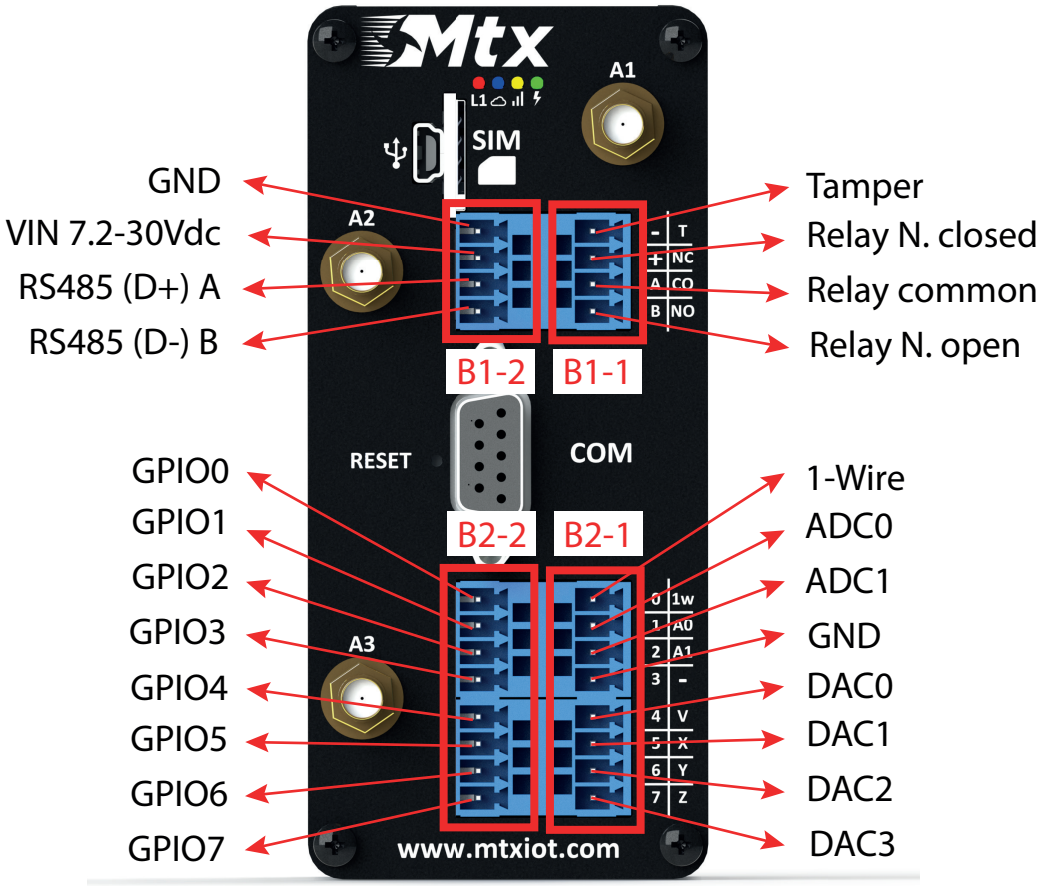
GPIO	SITUATION	PIN	I/O	FUNCTION
GPIO0	B2-2	1	Digital I/O	I: active via dry contact to GND O: open collector
GPIO1	B2-2	2	Digital I/O	I: active via dry contact to GND O: open collector
GPIO2	B2-2	3	Digital I/O	I: active via dry contact to GND O: open collector
GPIO3	B2-2	4	Digital I/O	I: active via dry contact to GND O: open collector
GPIO4	B2-2	5	Digital I/O	I: active via dry contact to GND O: open collector
GPIO5	B2-2	6	Digital I/O	I: active via dry contact to GND O: open collector
GPIO6	B2-2	7	Digital I/O	I: active via dry contact to GND O: open collector
GPIO7	B2-2	8	Digital I/O	I: active via dry contact to GND O: open collector
Tamper	B1-1	1	Tamper	Tamper input to ground

GPIO8	B1-1	2 (NC) 3 (COM) 4 (NO)	Latch relay output	30V / 1A
1-Wire	B2-1	1	1-Wire	Future use of 1Wire devices
ADC0	B2-1	2	Analog input	0-10V / 0-20mA
ADC1	B2-1	3	Analog input	0-10V / 0-20mA
GND	B1-2	1	GND (power)	
VIN	B1-2	2	VIN (power)	Voltage 7.2Vdc - 30Vdc
RS485 A	B1-2	3	RS485 A (D+)	RS485 COM2_ serial port
RS485 B	B1-2	4	RS485 A (D-)	RS485 COM2_ serial port
DAC0	B2-1	1	Analog output	0-10V 20mA max.
DAC1	B2-1	2	Analog output	0-10V 20mA max.
DAC2	B2-1	3	Analog output	0-10V 20mA max.
DAC3	B2-1	4	Analog output	0-10V 20mA max.

PIN	DB9	RS232
1	Carrier detector (DCD)	CD
2	Receive data (Rx)	RD
3	Transmit data (Tx)	TD
4	Data terminal ready	DTR
5	Signal ground/common (SG)	GND
6	Data set ready	DSR

7	Request to send	RTS
8	Clear to send	CTS
9	Voltage output: 4.05V with external power supply, 4.05-3.45V form internal battery	VOUT

- The “GPIO” column refers to the name of the MTX modem I / O signal
- The “Situation” column indicates in which MTX modem connector this I / O is located.
- The “PIN No.” column indicates which pin of the connector the I / O is on.
- The “I / O” column indicates the type of I / O
- Each digital input can be configured for: normal input without special functionality, sms alarm for state change, wakeup, pulse counter, execution of at commands for state change, sending of MQTT message for state change, alarm for voice call by status change. See examples in Annex 8 for more information.
- Each analog input can be configured for: normal input without special functionality, sms alarm for exceeding configurable thresholds, AT command execution for exceeding configurable thresholds, sending of MQTT message for variation of the configurable input value, alarm for voice call for exceeding configurable thresholds wakeup. See examples in Annex 8 for more information.
- Each digital output or latch relay can be configured for: normal output without special functionality (change of state using AT commands), on / off depending on the time, timed output, output on / off based on a digital input, output on / off depending on the value of one of the analogue inputs of the modem, output on / off based on the value of a modbus register of a modbus RTU device connected to the serial port (232/485) of the MTX modem, output activated X seconds per missed call from a phone number, output on / off based on astronomical clock, etc.



Antenna 3 is intended for GPS on models that support it.

1.2 MTX-IoT

Description of I/O interfaces:

- DB15 expansion connection: 3 digital inputs (3 of them pulse counters), 2 digital outputs and 2 A / D converters (0-50V in voltage mode. 0-20mA in current mode)
- Serial interfaces: 1 RS232 (DB9) port. 1 RS232 (DB15) port or 1 RS485 port (configurable by swtiches)
- Internal GPS connected to the secondary serial port (supported by MTX-IoT [3-S-N-GPS] and MTX-IoT [4-S-N-GPS] models)
- Wavecard 868MHz: internal radio card, connected to the secondary serial port (supported by MTX-IoT model [4-S-N-WCA868])
- The MTX-Tunnel software uses the inputs and outputs as follows:

GPIO	SITUATION	PIN	I/O	FUNCTION
GPIO0	DB15	4	Digital input	Activates GND via dry contact
GPIO1	DB15	11	Digital input	Activates GND via dry contact
GPIO2	DB15	9	Digital input	Activates by positive (>3V)
GPIO3	DB15	5	Digital output	Open collector
GPIO4	DB15	12	Digital output	Open collector
ADC0	DB15	15	Analog input	0-50V / 0-20mA
ADC1	DB15	13	Analog input	0-50V / 0-20mA
VExt	DB15	10	Output voltage	
GND	DB15	14	Ground	

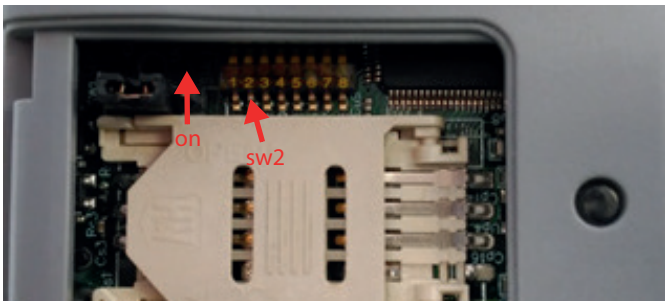
- The “GPIO” column refers to the name of the MTX modem I / O signal
- The “Situation” column indicates in which MTX modem connector this I / O is located.
- The “PIN No.” column indicates which pin of the connector the I / O is on.
- The “I / O” column indicates the type of I / O
- Each digital input can be configured for: normal input without special functionality, sms alarm for state change, wakeup, pulse counter, execution of at commands for state change, sending of MQTT message for state change, alarm for voice call by status change. See examples in Annex 8 for more information.

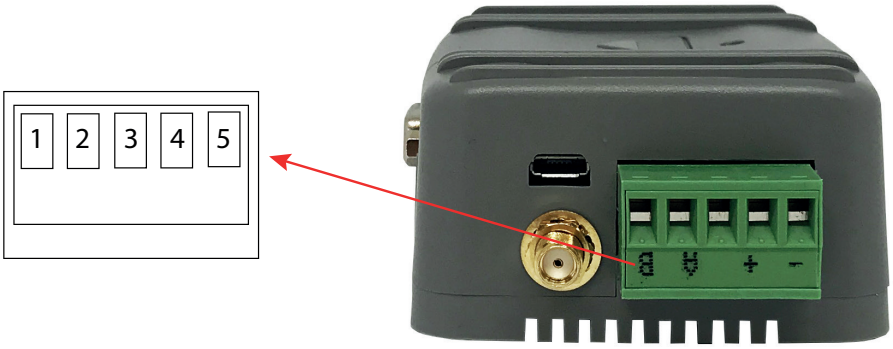
- Each analog input can be configured for: normal input without special functionality, sms alarm for exceeding configurable thresholds, AT command execution for exceeding configurable thresholds, sending of MQTT message for variation of the configurable input value, alarm for voice call for exceeding configurable thresholds wakeup. See examples in Annex 8 for more information.
- Each digital output can be configured for: normal output without special functionality (change of state using AT commands), on / off depending on the time, timed output, output on / off based on a digital input, output on / off based on the value of one of the modem's analog inputs, output on / off based on the value of a modbus register of a modbus RTU device connected to the serial port (232/485) of the MTX modem, output activated X seconds per missed call from a phone number, output on / off based on astronomical clock, etc.

Jumper configuration:

The MTX modem has a series of switches, which allow you to configure the modem in different ways:

- [SW1 off] + [SW2 on] > RS232 DB9 on, RS232 DB15 off, RS485 on
- [SW1 off] + [SW2 off] > RS232 DB9 on. RS232 DB15 on, o RS485 off
- [SW7 off] > Tension analog input 1 (0-50V)
- [SW7 on] > Current analog input 1 (0-20mA)
- [SW8 off] > Tension analog input 2 (0-50V)
- [SW8 on] > Current analog input 2 (0-20mA)





PIN	SIGNAL	DIRECTION	LIMITS	DESCRIPTION
1	-RxB	I/O		RS485 B signal (see section Interface Descripción 5 (RS485 bus) for details)
2	+RxA	I/O		RS485 A signal (see section Interface Descripción 5 (RS485 bus) for details)
3	NC			Not connected
4	VIN	Input	7-50VDC	Positive power input
5	GND	Input		Negative power (ground)

1.3 MTX-T

Description of I/O interfaces:

- Serial interfaces: 1x RS232 (DB9), 1x RS485
- GPS: internal, only in MTX-T [3-N]-G model
- MTX-Tunnel software uses these inputs/outputs as follows:

GPIO	SITUATION	PIN	I/O	FUNCTION
GPIO2	Terminal block	3	Digital input	Activates by positive (>3V)

- The “GPIO” column refers to the name of the MTX modem I / O signal
- The “Situation” column indicates in which MTX modem connector this I / O is located.
- The “PIN No.” column indicates which pin of the connector the I / O is on.
- The “I / O” column indicates the type of I / O
- Each digital input can be configured for: normal input without special functionality, sms alarm for state change, wakeup, pulse counter, execution of at commands for state change, sending of MQTT message for state change, alarm for voice call by status change. See examples in Annex 8 for more information

1.4 MTX-T2

Description of I/O interfaces:

- Serial interfaces: 1x RS232 (DB9), 1x RS232 (secondary DB9)
- MTX-Tunnel software uses these inputs/outputs as follows:

GPIO	SITUATION	PIN	I/O	FUNCTION
GPIO2	Terminal block	3	Digital input	Activates by positive (>3V)

- The “GPIO” column refers to the name of the MTX modem I / O signal
- The “Situation” column indicates in which MTX modem connector this I / O is located.
- The “PIN No.” column indicates which pin of the connector the I / O is on.
- The “I / O” column indicates the type of I / O
- Each digital input can be configured for: normal input without special functionality, sms alarm for state change, wakeup, pulse counter, execution of at commands for state change, sending of MQTT message for state change, alarm for voice call by status change. See examples in Annex 8 for more information

2. ANNEX B: HOW TO USE VIRTUAL SERIAL PORTS (VSPE)

2.1 EMULATION SOFTWARE FOR SERIAL PORTS: VSPE

Quick guide to the emulation software for serial ports

Introduction:

One of the most used pieces of software to emúlate serial ports nowadays is VSPE. This is a free piece of software for 32 bit Windows operating systems. There is also a paid version for 64 bit platforms.

This software should be used if you are using older software in order to connect your serial devices which only allow connections via a COM serial port. This software allows you to create virtual COMs on your PC (COM1, COM2, COM3, etc.) which, in reality, refer to a specific IP/TCP port.

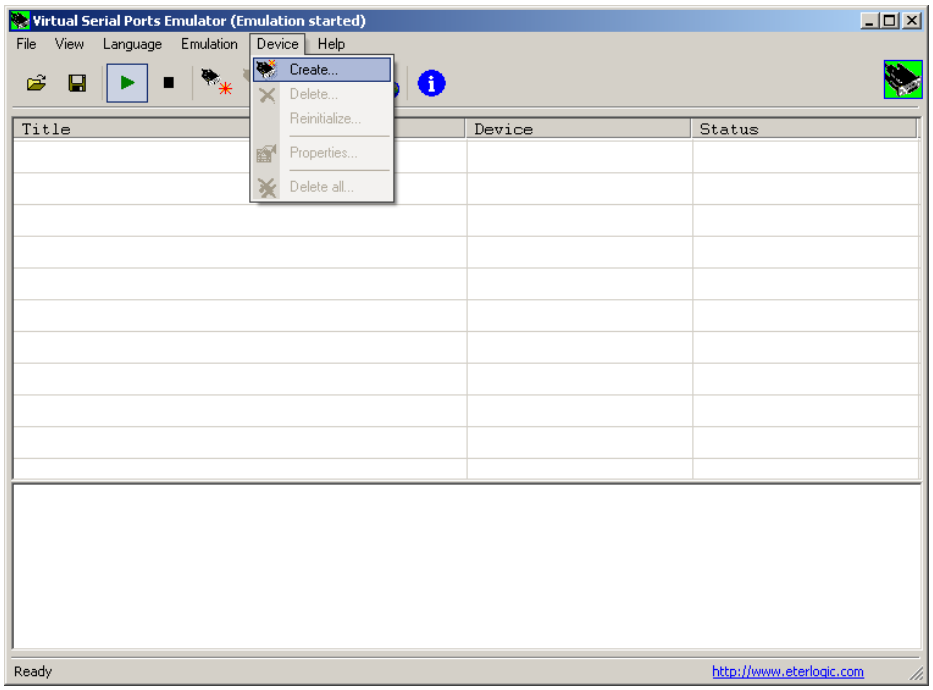
The software can be downloaded from the following link:

http://www.eterlogic.com/downloads/SetupVSPE.zip

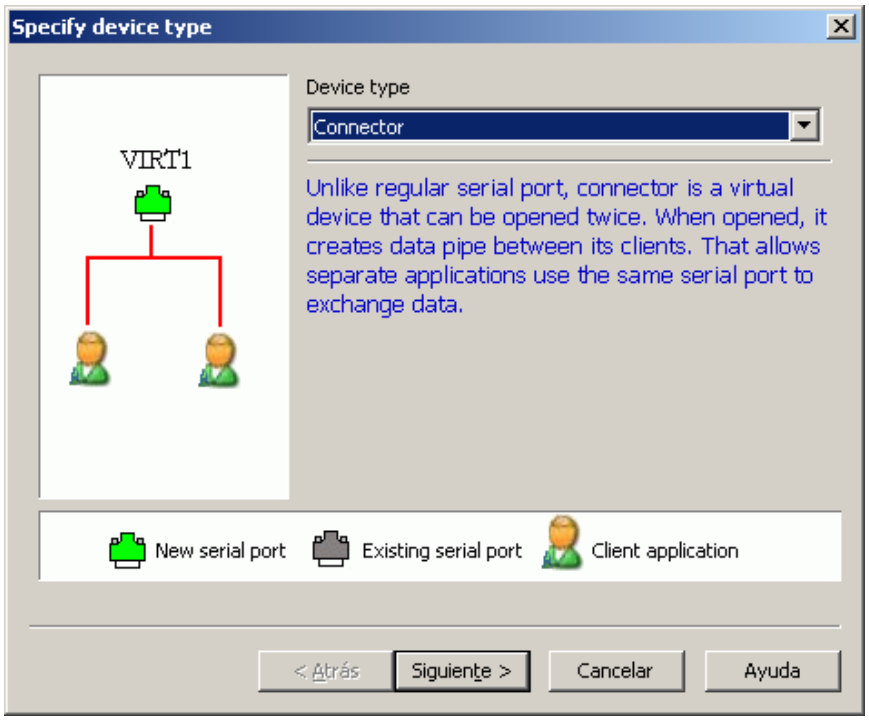
Usage Example:

The following steps show how to configure a virtual COM port that refers to a specific IP address or TCP port.

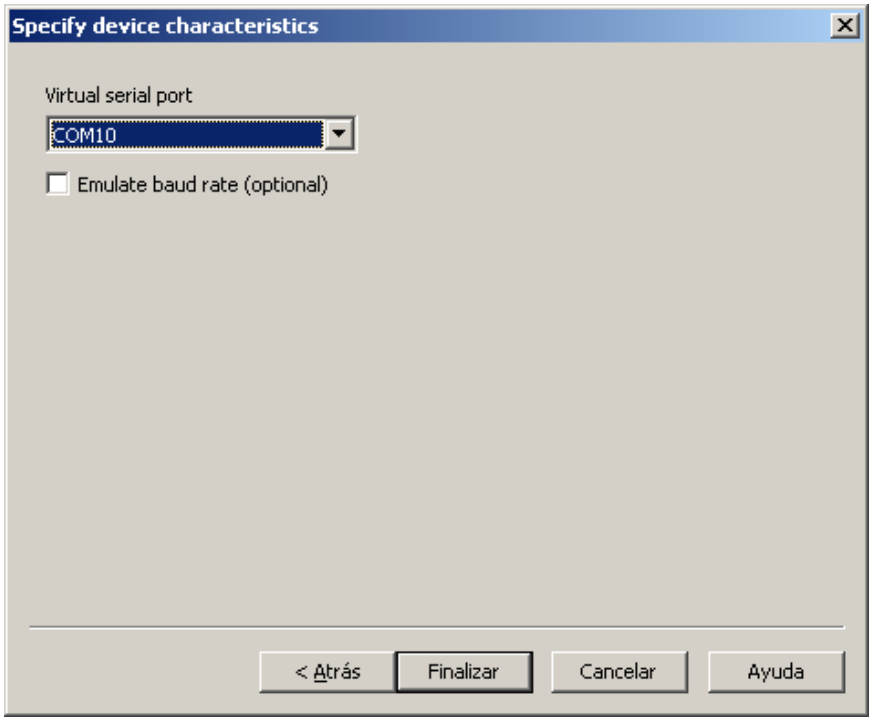
- Menu: Device > Create



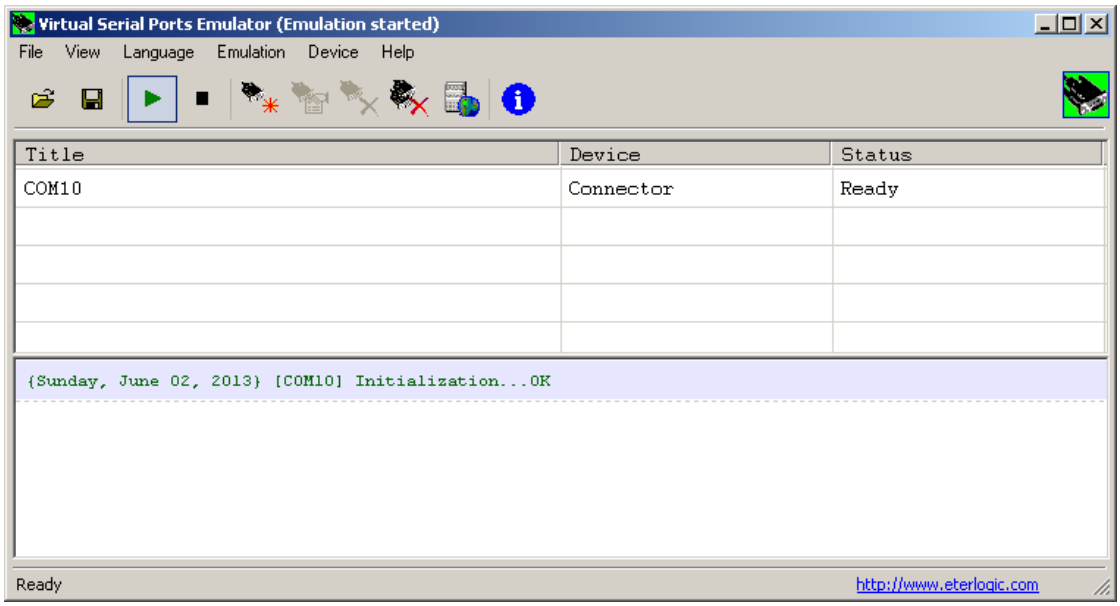
- We select “Connector”



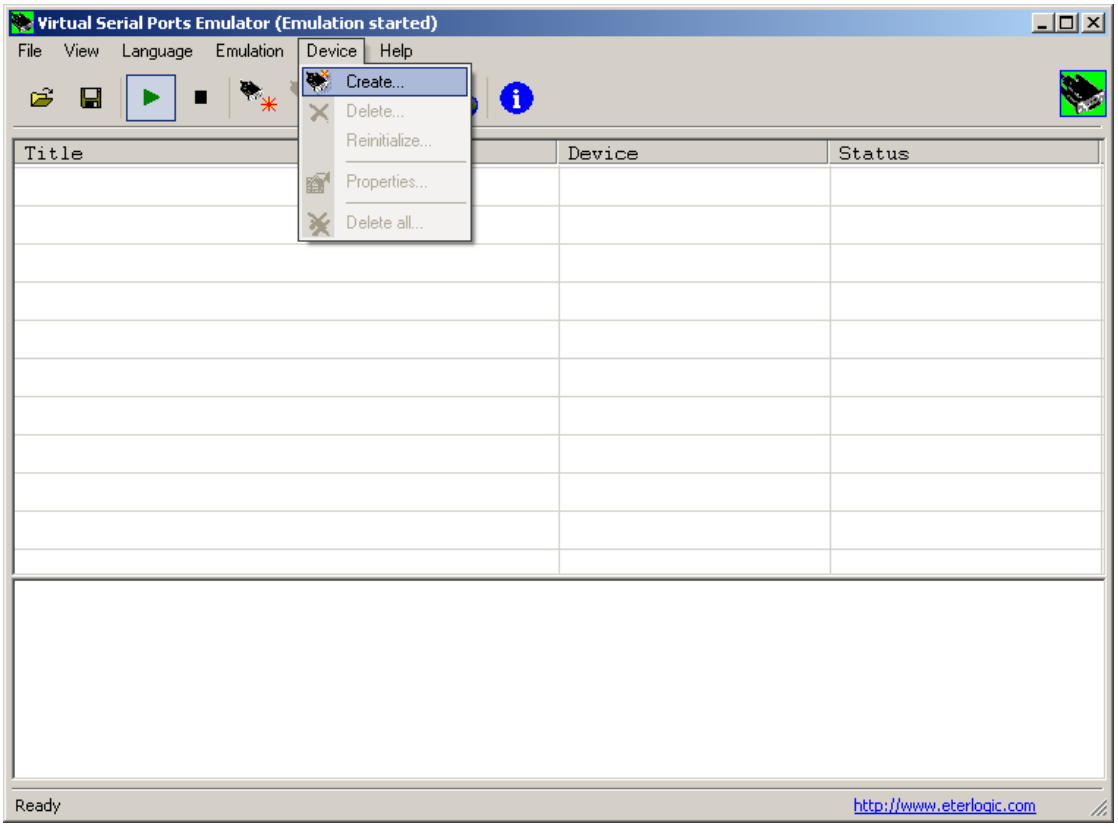
- Select the number of the virtual COM that we want to use. For example, COM10



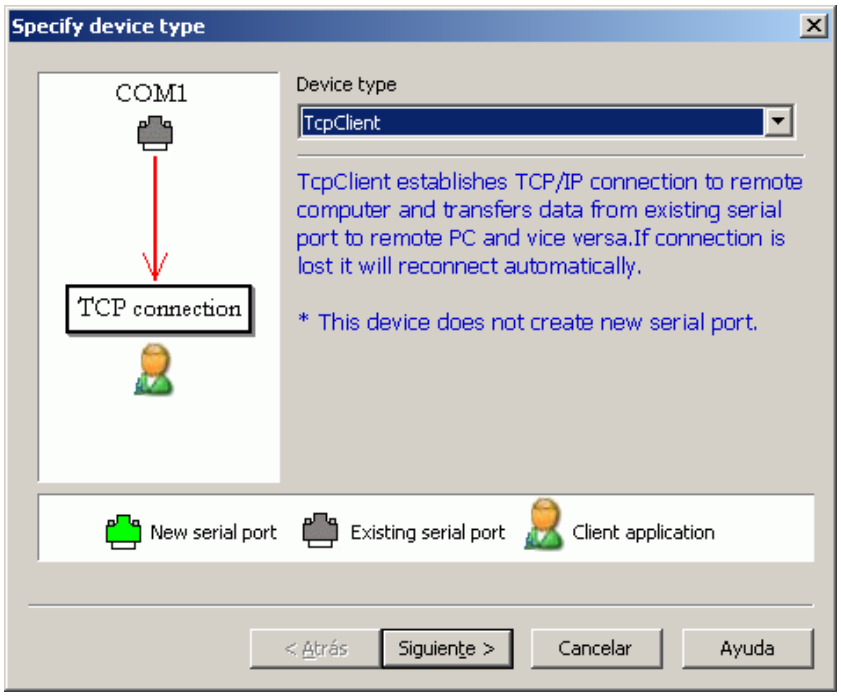
- Click Finish. The virtual COM created will appear in the list



- Return to the Menu: Device > Create



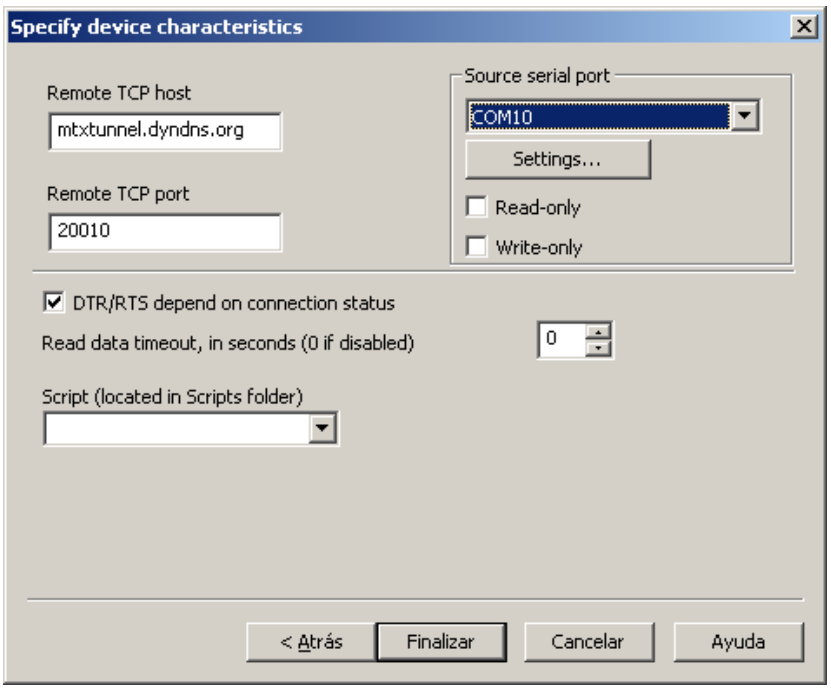
- Select TCPClient in the drop down menu



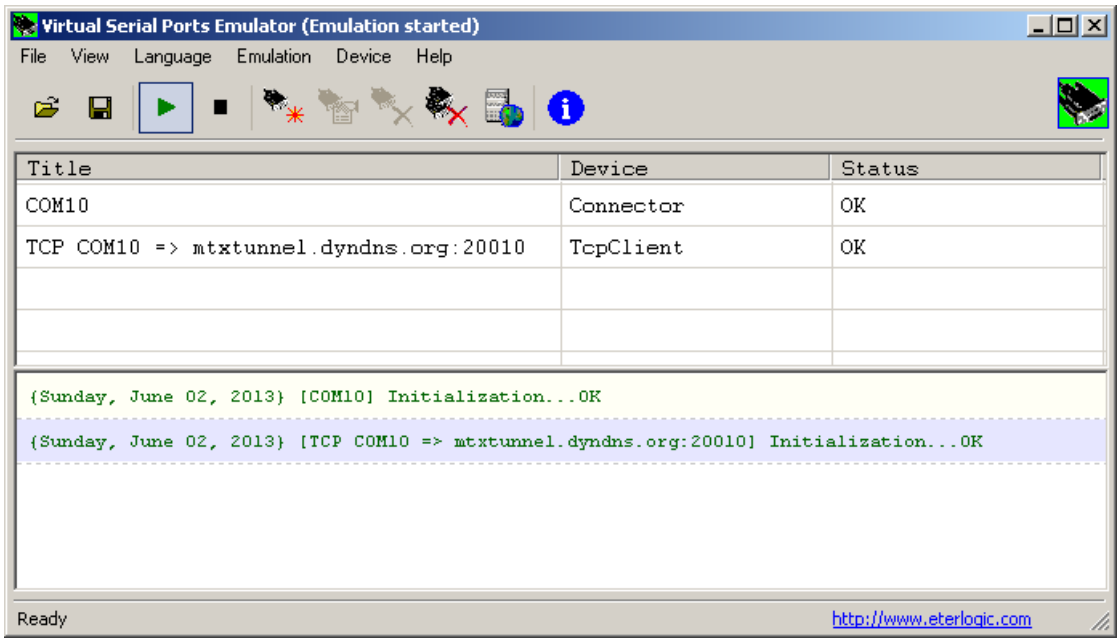
- On the following screen we indicate the remote IP address of MTX-Tunnel. We can use a numerical IP address (if we use a SIM with a fixed IP address, or because we obtain the actual IP address by SMS or a missed call), or we can use a DNS (if we use the DynDNS service for example)

  We also select the TCP port in which MTX-Tunnel is connected to (the default is 20010). Finally, we select the virtual COM port that we previously created (in our case COM10).

  Click Finish.



- Now COM10 is connected to our MTX-Tunnel



- The last step is to Access our control application and select COM10 as the communications port. Now, we can access our serial devices remotely. Remember that the configuration of the modem's serial port (bauds, number of data bits, etc.) must be established in the modem's configuration file "config.txt", as indicated in this manual

3. ANNEX C: GENERATION AND USE OF SSL CERTIFICATES

3.1 Certificate usage documentation

Introduction:

To use your server certificates and/or client certificates in SSL communications you need to follow a series of steps, adapting the certificate files to be compatible with MTX-Tunnel. You can use up to 10 root certificates and 1 client certificate.

Preparing a server root certificate:

To use a server certificate, it must first be converted into a file accepted by the MTX-Tunnel. To carry out the conversion you will need: 1) have JAVA installed on your computer, 2) have the "jseccmd.jar" file corresponding to your MTX model (if you don't have it, request it at iotsupport@mtxm2m.com, indicating the exact model of your modem), 3) the file with the root certificate to use in .der format (for example, servercertificate.der). With this, you can execute the following command (assuming you have the files hosted in a folder named "C: \ certs \"

"C: \ Program Files (x86) \ Java \ jre7 \ bin \ java" -jar "C: \ certs \ jseccmd.jar" -cmd AddHttpsCertificateUntrusted -filename C: \ certs \ servercertificate.der> C: \ certs \ servercertificate1 .jar

The resulting file, servercertificateX.jar (mandatory name), is the transformed file that can be used in the MTX-Tunnel. You can use up to 10 certificates (servercertificate1.jar... servercertificate10.jar).

Preparation of a client certificate:

To use a client certificate, it must first be converted into a file accepted by the MTX-Tunnel. To carry out the conversion you will need: 1) have JAVA installed on your computer, 2) have the "jseccmd.jar" file corresponding to your MTX model (if you don't have it, request it at iotsupport@mtxm2m.com, indicating the exact model of your modem), 3) the .key and .crt files of the certificate (for client.crt and key.crt). With this, you can execute the following command (assuming that you have the files hosted in a folder named "C: \ certs \"

"C: \ Program Files (x86) \ Java \ jre7 \ bin \ java" -jar "C: \ certs \ jseccmd.jar" -cmd AddHttpsClientCertificateUntrusted -filename c: \ certs \ client.crt -keyfilename c: \ certs \ client.key> c: \ certs \ clientcertificate1.jar

The resulting file, clientcertificate1.jar (mandatory name), is the transformed file that can be used in the MTX-Tunnel.

Local installation of server root certificates:

If you wish, you can install your root SSL certificates locally (via the USB port). To do this, follow the steps below:

1. Copy the files "servercertificate1.jar", "servercertificate2.jar", ... "servercertificate10.jar" that you have previously generated in the path "security / certs / servers /" that you will find inside the modem (procedure similar to the one performed in copy the file "config.txt" into the modem)

2. Copy a file (without content) with the name “installservercertificates” in the path “security / certs /”
3. Restart the MTX. After restarting, the MTX-Tunnel, upon detecting the file “installservercertificates”, will install the root certificates copied in step 1

Local installation of client certificate:

If you wish, you can install your root SSL certificates locally (via the USB port). To do this, follow the steps below:

1. Copy the file “clientcertificate1.jar” that you have previously generated in the path “security / certs / client /” that you will find inside the modem (procedure similar to the one you do when copying the file “config.txt” inside the modem)
2. Copy a file (without content) with the name “installclientcertificate” in the path “security / certs /”
3. Restart the MTX. After restarting, the MTX-Tunnel, upon detecting the file “installclientcertificate”, will install the root certificates copied in step 1

Remote update of server root certificates:

You can remotely update the server root certificates. To do this, follow the steps below:

1. Use the command AT ^ MTXTUNNEL = DOWNLOAD,.... to download from your web server the certificates you want to update / upload to the MTX modem. For example, suppose you want to update / load the certificates “servercertificate1.jar” and “servercertificate2.jar”. To do this, you must execute the following commands remotely (via telnet, mqtt, ...) in the MTX-Tunnel. This will make the certificates download to the modem

   AT ^ MTXTUNNEL = DOWNLOAD, https: //www.mydomain.com/ myPath,myUsername,myPass,servercertificate1.jar,security/certs/servers/

   AT ^ MTXTUNNEL = DOWNLOAD, https: //www.mydomain.com/ myPath,myUsername,myPass,servercertificate2.jar,security/certs/servers/
2. Run the command AT ^ MTXTUNNEL = CERTIFICATE, INSTALLSERVERS

Remember that you can use the following commands to perform more operations with the certificates, such as listing the certificates or deleting one of them. See the use of the following commands in the AT command section of this manual.

AT^MTXTUNNEL=CERTIFICATE,LISTSERVERS

AT^MTXTUNNEL=CERTIFICATE,DELETESERVER,servercertificateX.jar

3. ANNEX D: MANUFACTURER FILTERS W-MBUS

List of W-Mbus device manufacturer filters for the WMBus_filter parameter:

FILTER	MANUFACTURER
ABB	ABB AB, P.O. Box 1005, SE-61129 Nyköping, Nyköping,Sweden
ABN	ABN Braun AG, Platenstraße 59, 90441 Nürnberg, Germany
ACA	Acean, Zi de la Liane, BP 439, 62206 Boulogne Sur Mer Cedex, FRANCE
ACB	AcBel Polytech Inc., No. 159, Sec. 3, Danjin Rd., Tamsui Dist.,New Taipei, Taiwan (R.O.C.)
ACE	Actaris, France. (Electricity)
ACG	Actaris, France. (Gas)
ACW	Actaris, France. (Water and Heat)
ADD	"ADD-Production" S.R.L., 36, Dragomirna str., MD-2008, Chisinau, Republic of Moldova
ADX	ADD-Production S.R.L., 36, Dragomirna str., MD-2008, Chisinau, Republic of Moldova
ADN	Aidon Oy, 40101 Jyvaskyla, Finland
ADU	Adunos GmbH, Am Schlangengraben 16, D-13597 Berlin, Germany
AEC	Advance Electronics Company, Riyadh, Saudi Arabia
AEE	Atlas Electronics, 17530 Surdulica, Serbia and Montenegro
AEG	AEG
AEL	Kohler, Turkey
AEM	S.C. AEM S.A. Romania
AER	Aerzener Maschinenfabrik GmbH, Reherweg 28, 31855 Aerzen, Germany
AFX	Alflex Products, Zoetermeer, Holland
AGE	AccessGate AB, Rissneleden 144, 174 57 Sundbyberg, Sweden

ALF	Alfatech - Elektromed Elektronik, Ankara, Turkey
ALG	Algodue Elettronica srl, Via Passerina, 3/A, Fontaneto D\'Agogna, Italy
ALT	Amplitec GmbH, Gootkoppel 28, Reinfeld, Germany
AMB	Amber wireless GmbH, Hawstra e 2a, 54290 Trier, Germany
AME	AVON METERS PRIVATE LIMITED, D-15/16/17, INDUSTRIAL FOCAL POINT, DERABASSI,PUNJAB-140507INDIA
AMH	AMiHo Ltd, 1010 Cambourne Business Park, Cambourne, Cambridge, CB1 9AY, UK
AMI	AMI Tech(I) Pvt. Ltd, #205&206,NSIC-EMDBP, Kamalanagar, ECIL PO, Hyderabad-500062, India
AML	Eon Electric Ltd., C-124, Hosierry Complex, Noida Phase II, NOIDA, INDIA
AMP	Ampy Automation Digilog Ltd
AMS	Zhejiang Chaoyi Electronic Technology Co. Ltd., Zhejiang, China.
AMT	Aquametro
AMX	APATOR METRIX SA, Piaskowa 3, 83-110 Tczew, Poland
ANA	Anacle Systems PTE LTD, 1A International Business Park #05-02, Singapore
AND	ANDIS sro, Bratislava, Slovakia
AON	ASTRON d.o.o., Cesta XIV. divizije 51, Maribor, Slovenia
APA	APATOR SA (Electricity), Zolkiewskiego 21/29, 87-100, Torun,Poland
APR	Apronecs Ltd, Gabrovo, Bulgaria
APS	Apsis Kontrol Sistemleri, Turkey
APT	Apator SA (Gas, water and heat), ó kiewskiego 21/29, Toru , Poland
APX	Amplex A/S, Aarhus C, Denmark
AQM	Aquametro AG, Ringstrasse 75, Therwil, Switzerland

AQT	AQUATHERM P.P.H, Kujawinski, Lomianki, Poland
ARC	Arcelik AS., Istanbul, Turkey
ARF	ADEUNIS_RF, 283 rue Louis NEEL, CROLLES 38920, France
ARM	arivus metering GmbH, Mielestr. 2, 14542 Werder (Havel), Germany
ARS	ADD - Bulgaria Ltd, Bul. \"6 septemvri\" 252 Et.7, Plovdiv, Bulgaria
ART	Electrotécnica Arteche Smart Grid, Derio Bidea 28, Zabalondo Industrialdea, 48100 Mungia, Bizkaia. Spain
ASR	Erelsan Elektrik ve Elektronik, Malzeme, Istanbul, Turkey
AST	ASTINCO Inc., 114 Anderson Ave. Suite 7A, ON, L6E1A5, Markham, Canada
ATF	AKTIF Otomasyon ve GS ve Tic, Turkey
ATI	ANALOGICS TECH INDIA LIMITED, Plot No.9/10, Road No.6, Nacharam Industrial Estate, HYDERABAD, INDIA
ATL	Atlas Elektronik, ANKARA, Turkey
ATM	Atmel, Torre C2, Polígono Puerta norte, A-23, 50820, (Zaragoza) Spain
ATS	Atlas Sayaç Sanayi A. ., Erciyes Teknopark 4.Bina Talas, Kayseri, Turkey
AUX	Ningbo Sanxing Electric Co., Ltd., No.1166, Mingguang North Rd. Jiangshan Town, Ningbo, China
AXI	UAB "Axis Industries", LT-47190, Lithuania
AZE	AZEL Electronics, B. Ankara, Turkey
BAM	Bachmann GmbH & Co KG, Ernsthaldenstr, 33 70565, Stuttgart, Germany
BAR	Baer Industrie-Elektronik GmbH, Fuerth, Germany
BAS	BASIC INTELLIGENCE TECHNOLOGY CO.,LTD., 1st Floor, No.1 NanLi Rd. PanYu District, GuangZhou, GuangDong, China.
BBS	BBS Electronics, Singapore

BCE	ShenZhen B.C Electronic CO.Ltd, 4F, Strength Building,GaoXin Ave.1.s, South, Hi-technology industry Zone, ShenZhen, China
BEE	Bentec Electricals & Electronics Pvt Ltd, 150,Upen Banerjee Road,Kolkata, Kolkata, India
BEF	BEFEGA GmbH, Reichenbacher Str. 22, Schwabach, Germany
BEG	begcomm Communication AB, Brunnehagen 109, GöteborgSweden
BER	Bernina Electronic AG
BHG	Brunata A/S, DK-2730 Herlev, Denmark
BKB	Boendekomfort AB, Box 37, 260 40 Viken, SWEDEN
BKT	Bekto Precisa, Ibrahima Popovi a bb., Gora de, Bosnia and Herzegovina
BKO	Beko Elektronik AS., Istanbul, Turkey
BLU	Bluering, Brescia, Italy
BME	Beifeng GmbH, 60599 Frankfurt am Main, Germany
BMI	Badger Meter Inc., 6116 E 15th St., Tulsa, USA
BMP	BMETERS Polska Sp.z.o.o., Glowna 60, Psary, Poland
BMT	BMETERS srl, Via Friuli, 3, 33050, Gonars (UD), ITALY
BRA	Brandes GmbH, D-23701 Eutin, Germany
BSC	Sanayeh Sanjesh Energy Behineh Sazan Toos, Toos Industrial Estate,Mashhad,Iran
BSE	Basari Elektronik A.S. Turkey
BSM	Bluestar Electrical Meter Research Institute, Nanjing, China
BSP	Byucksan Power Co. Ltd., 6th Fl. New Hosung Bldg. Yoido-Dong, Youngdeungpo-Gu, Seoul, Korea.
BST	BESTAS Elektronik Optik, Turkey
BSX	BS-Messtechnik UG, Kassel, Germany

BTL	BIT-LAB
BTR	RIA-BTR Produktions-GmbH, D-78176 Blumberg
BTS	Basari Teknolojik Sistemler AS, Ankara, Turkey
BUR	Bopp und Reuther Messtechnik GmbH, Speyer, Germany
BYD	BYD Company Limited, BYD Road NO.3009 PingShan, ShenZhen, China
BYL	BAYLAN, Turkey
BXC	Beijing Fuxing Xiao-Cheng Electronic Technology Stock Co., Ltd, Room 503, Block D, IFEC Blog, No.87 Xisanhuan Beilu, Haidian District, Beijing, China
BZR	Gebr. Bauer GbR, 87719 Mindelheim, Germany
CAH	MAEC GROUPE CAHORS, ZI DE REGOURD BP 149, 46003 CAHORS CEDEX 9, FRANCE
CAT	Cubes And Tubes OY, Olli Kytölän tie 1, MUURAME, FINLAND
CBI	Circuit Breaker Industries, South Africa
CDL	Customised Data Ltd., 44 Allerburn Lea, Alnwick, UK
CEB	Cebyc AS, Vestre Rosten 81 / 13th Floor, Tiller, Norway
CET	Cetinkaya Aydinlatma, Istanbul, Turkey
CGC	Contor Group S.A., Romania - 310059 Arad, Calea Bodrogului nr. 2-4
CIR	Circutor, Viladecavalls/Barcelona, Spain
CLE	Shen Zhen Clou Electronics Co Ltd, Guangdong, China
CLO	Clorius Raab Karcher Energi Service A/S
CLY	Clayster, FO Petersons gata 6, 421 31 Västra Frölunda, Sweden
CMC	CMC EKOCON d.o.o., IOC ZAPOLJE I/10, LOGATEC, SLOVENIA
CMP	CM Partner Inc, Yongin, South Korea

CMV	Comverge, Inc., 5390 Triangle Parkway, Suite 300, Norcross, GA 30041, USA
CNM	COSTEL, #462-870 COSTEL Bldg., 223-39, Sangdaewon-Dong, Jungwon-Gu, Sungnam-Si, Kyunggi-Do, Korea
COM	COMMON S.A., Aleksandrowska 67/93, LODZ, POLAND
CON	Conlog
CPL	CPL CONCORDIA Soc.Coop., Via A. Grandi, 39 - 41033 Concordia s/S (MO), Italy
CPO	C3PO S.A., Alejandro Goicoechea 6, Sant Just Desvern, Spain
CRW	Xuzhou Runwu Science and Technology Development Co. Ltd., NO.5,Huijin Road, Damiao Industry Park,Economic Development Zone, Xuzhou,Jiangsu, P.R.China
CTR	Contar Electronica Industrial, Lisboa, Portugal
CTX	Contronix GmbH, Nizzastr 6, Radebeul, Germany
CUC	cuculus GmbH, Ehrenbergstrasse 11, D-98693 Ilmenau,Germany
CUR	CURRENT Group, LLC, 20420 Century Boulevard, Germantown, MD, USA
CWI	Cewe Instrument AB, Nykoping, Sweden
CWV	CMEC Electric Import & Export Co. Ltd, Beijing 100055, China
CYE	Quanzhou Chiyoung Electronics Technology Co. Ltd., #20 Hongshan Rd,Shudou community, Changtai St, Quanzhou City, Fujian 362000, China
CYI	QUANZHOU CHIYOUNG INSTRUMENT CO., LTD, #20 Hongshan Rd,Shudou community, Changtai St, Quanzhou City, Fujian, China 362000
CYN	Cynox, Weinart Engineering, Bad Zwischenahn, Germany
CZA	Contazara, Zaragoza, Spain
CZM	Cazzaniga S.p.A.
DAE	DAE Srl, Via Trieste, 4/E, Santa Lucia di Piave, Italy
DAF	Daf Enerji Sanayi ve Ticaret A.S, Atasehir Bulvari Ata Carsi Kat:4 No:52 34758 Atasehir, Istanbul, Turkey

DAN	Danubia
DBE	Decibels Electronics P Ltd, Decibels Electronics Pvt Ltd., 6-1-85/4, Saifabad, Hyderabad, AP, India
DDE	D&D Elettronica srl, Via XXV Aprile, 37, Bresso (MI), ITALY
DEC	DECODE d.o.o. Data Communications, Belgrade
DEL	DELTAMESS DWWF GmbH, Sebenter Weg 42, 23758 Oldenburg in Holstein, Germany
DES	Desi (Alarms) Ltd, Turkey
DEV	Develco Products, Olof Palmes Allé 40, 8200 Aarhus N, Denmark
DFE	Dongfang Electronics Co., Ltd., JiChang road 2#, Yantai City, Shandong Province, China
DFS	Danfoss A/S
DGC	Digicom S.p.A., Via A.Volta 39, 21010 Cardano al Campo (VA), Italy
DGM	Diehl Gas Metering GmbH, Industriestrasse 13, Ansbach, Germany
DIE	Dielen GmbH, Zeppelinstrasse 9, 47638 Straelen, Deutschland
DJA	De Jaeger Automation bvba, Molenstraat 200, B-9900 EEKLO, Belgium
DKY	Electric Power Research Institute of Guangdong Power Grid Corporation, No. 8, Shui jungang Dongfengdong Road, Guangzhou, China
DIL	DECCAN INFRATECH LIMETED, A3-4/A,Electronic Complex, Kushaiguda, HYDERABAD, INDIA
DMC	DMC International, Al Gharhoud, Dubai, UAE
DME	DIEHL Metering, Industriestrasse 13, 91522 Ansbach, Germany
DMP	DM Power Co., Ltd, #SB118 Megavalley, Gwanyang,-Dong, Anyang City, South Korea
DNO	DENO d.o.o, Zagreb, Croatia
DNT	Dr Neuhaus Telekommunikation GmbH, Hamburg, Germany
DNV	DNV KEMA, Utrechtseweg 310, Arnhem, Netherlands

DPP	DECCAN POWER PRODUCTS PVT. LTD., A3-4/A, Electronic Complex, Kushaiguda, Hyderabad-500062, INDIA
DRT	DRESSER Italia S.r.l., Via Roma, 772, Talamona (SO), Italy
DSE	Digitech Systems and Engineering Private Limited, 18 Ramamurthy Street, Nehru Nagar, Chromepet, Chennai-600044, Tamil Nadu, India
DSE	DSE energy Co., Ltd, 8F, No.531, HsinTien, Taipei, Taiwan
DWZ	Lorenz GmbH & Co. KG, Burgweg 3, 89601 Schelklingen, Germany
DZG	Deutsche Zahlergesellschaft
EAA	Electronic Afzar Azma, Iran
EAH	Endress+Hauser, 87484 Nesselwang, Germany
EAS	EAS Elektronik San. Tic. A.S., Ankara, Turkey
ECH	Echelon Corporation, 550 Meridian Avenue, San Jose, California, USA.
ECL	Electronics Corporation of India Ltd, Hyderabad, India
ECS	Herholdt Controls srl, Milan, Italy
EDI	Enel Distribuzione S.p.A, Via Ombrone, 2, Rome, Italy
EDM	EDMI Pty. Ltd.
EEE	3E s.r.l., Via Biandrate, 24, Novara, Italy
EEO	Eppeltone Engineers, A 293/1 Okhla Industrial Area Phase 1, New Delhi, India
EFA	EFACEC Engenharia e Sistemas SA, Apartado 3078, MAIA, PORTUGAL
EFE	Engelmann Sensor GmbH, Rudolf-Diesel-Straße 24-28, 69168 Wiesloch, Germany
EFR	Europäische Funk-Rundsteuerung, Nymphenburger Strasse 20b, Munich, Germany
EGA	eGain International AB, Faktorvägen 9, Kungsbacka, Sweden
EGM	Elgama-Elektronika Ltd, Lithuania

EHL	Secure Meters Limited
EIT	EnergyICT NV, 8500 Kortrijk, Belgium
EKA	Eka Systems, Germantown, MD 20874, USA
EKT	PA KVANT J.S., Russian Federation
ELD	Elektromed Elektronik Ltd, Turkey, O.S.B. Uygurlar Cad. No:4 Sincan, Ankara, Turkey
ELE	Elster Electricity LLC, 208 Rogers Lane, Raleigh, USA
ELG	Elgas s.r.o., Pardubice, Czech Republic
ELM	Elektromed Elektronik Ltd, Turkey
ELO	ELO Sistemas Eletronicos S.A., Brazil
ELQ	ELEQ b.v., Karl-Ferdinand-Braun-Straße 1, Kerpen, Germay
ELR	Elster Metering Limited, 130 Camford Way, Luton, UK
ELS	Elster GmbH, 55252 Mainz-Kastell, Germany
ELT	ELTAKO GmbH, Hofener Straße 54, 70736 Fellbach, Germany
ELV	Elvaco AB, Kungsbacka, Sweden
EMC	Embedded Communication Systems GmbH, vom Staal-Weg 10, 4500 Solothurn, Switzerland
EME	SC. Electromagnetica SA, Bucharest, Romania
EMH	EMH metering GmbH & Co. KG (formerly EMH Elektrizitatszahler GmbH & CO KG)
EML	Emlite ltd, 10 Reynolds Business Park, Stevern Way, PE1 5EL Peterborough, UK
EMM	Email Metering, Australia
EMO	Enermet
EMS	EMS-PATVAG AG, CH-7013 Domat/Ems, Switzerland

EMT	Elster Messtechnik GmbH, Lampertheim, Germany
EMU	EMU Elektronik AG, 6432 Rickenbach SZ, Switzerland
END	ENDYS GmbH
ENE	ENERDIS, 16 rue Georges Besse SILIC44, 92182 ANTONYFRANCE
ENG	ENER-G Switch2 Ltd, The Waterfront, Salts Mill Rd, Bradford, BD17 7EZ, UK
ENI	entec innovations GmbH, Hebelstr. 1, 79379 Müllheim, Germany
ENL	ENEL d.o.o. Beograd, Belgrade, Serbia and Montenegro
ENM	ENMAS GmbH, Holzkoppelweg 23, Kiel, Germany
ENO	ennovatis GmbH, Stammheimer 10Kornwestheim, Germany
ENP	"Kiev Polytechnical Scientific Research"
ENR	Energisme
ENS	ENSO NETZ GmbH, Postfach 12 01 23, 01002 Dresden, Dresden, Deutschland
ENT	ENTES Elektronik, Istambul
ERE	Enermatics Energy (PTY) LTD, Mertech Building, Glenfield Office Park, Oberon str., Faerleglen, Pretoria, South Africa
ERI	Easun Reyrolle Limited, 389, Rasukumaki, Hulimavu, Bannerghatta Road, Bangalore-560076, India
ERL	Erelsan Elektrik ve Elektronik, Turkey
ESE	ESE Nordic AB, Slottagårdsgatan 9, Vellinge, Sweden
ESI	Monosan Monofaze Elektrik Motorlari, Turkey
ESM	Monosan Monofase Elekrik Motorlari, Turkey
ESO	Monosan Monofaze Elektrik Motorlari, Turkey
ESS	Energy Saving Systems LTD., Zroshyvalna, 15b, Kiev, Ukraine

ESY	EasyMeter GmbH
EUE	E+E Electronic, Langwiesen 7, 4209 Engerwitzdorf, Austria
EUR	Eurometers Ltd
EVK	EV KUR ELEKTRIK, Istanbul, Turkey
EWG	EWG DOO, Bulevar Svetog Cara Konstantina 80-82, Ni , 18106, Serbia
EWT	Elin Wasserwerkstechnik
EYT	Enerlyt Potsdam GmbH
FAN	Fantini Cosmi S.p.A., Via dell"Osio 6, 20090 Caleppio di Settala, Miano, Italy
FAR	FARAB, No. 18, Mirhadi St., Jooybar St., Fatemi Sq., Tehran, IRAN
FED	Federal Elektrik, Turkey
FFD	Fast Forward AG, Ruedesheimer Strasse 11, Munich, Germany
FIM	Frodexim Ltd, Sofia, Bulgaria
FIN	Finder GmbH, Hans-Böckler-Starße 44, 65468 Trebur-Astheim, Deutschland
FLE	XI'AN FLAG ELECTRONIC CO.,LTD, Flag Electronic Industry Park,No.11,Zhangba 6 Rd.(New Zone), Hi-Tech Development Zone, Xi'an, ShaanXi, PRC., China
FIO	Pietro Fiorentini, Via Rosellini,1, Milano, Italy
FLG	FLOMAG s.r.o, Brno, Czech Republic
FLO	Flonidan A/S, 8700 Horsens, Denmark
FLS	FLASH o.s, Istanbul, Turkey
FMG	Flow Meter Group, Menisstraat 5c, 7091 ZZ Dinxperlo, The Netherlands
FML	Siemens Measurements Ltd. (Formerly FML Ltd.)
FNX	Flownetix Ltd, Marlow Bottom, Bucks, UK

FRE	Frer Srl, Viale Europa, 12, Cologno Monzese (MI), Italy
FSP	Finmek Space S.p.A., I-34012 Trieste
FTL	Tritschler GmbH, Schönaustr. 10+12, Laufenburg, Deutschland
FUS	Fuccesso, 98 Yingchundong, Taizhou, China
FUT	first:utility, Tachbrook Park, Warwick, UK
FWS	FW Systeme GmBH, Ehnkenweg 11, 26125 Oldenburg, Germany
GAV	Carlo Gavazzi Controls S.p.A, Via Safforze 8 C.A.P. 32100, Belluno, Italy
GBJ	Grundfoss A/S
GCE	Genergica, Caracas, Venezuela
GEC	GEC Meters Ltd.
GEE	GE Energy, Lauder House, Almondvale Business Park, Livingston, UK
GEL	Industrial Technology Research Institute, Rm. 809, Blg.51, No. 195, Sec. 4, Chung Hsing Rd., Chutung, Hsinchu, Taiwan
GEN	Goerlitz AG, Germany
GEO	Green Energy Options Limited, 3 St. Mary"s Court, Main Street Hardwick, Camridge, England, CB23 7QS
GET	Genus Electrotech Ltd., Survey No-43,Galpadar Road, Taluka anjar, District-kutch, gandhidham-370110 Gujrat, Taluka anjar,India
GFM	GE Fuji Meter Co.,Ltd., Horigane Karasugawa 2191, Azumino-City Nagano, Japan
GIL	Genus Innovations Limited, SPL-2B, RIICO Industrial Area, Sitapura, Jaipur, India
GIN	Gineers Ltd, 1756 Sofia, Bulgaria
GMC	GMC-I Messtechnik GmbH, Südwestpark 15, D-90449 Germany
GME	Global Metering Electronics, Amsterdam, Netherlands
GMM	Gamma International Egypt, Abour, St 130, industrial area, Cairo, Egypt

GMT	GMT GmbH, Odenwaldstraße 19, 64521 Groß-Gerau, Germany
GOE	Genus Overseas Electronics Ltd, Jaipur, India
GRE	GE2, Green Energy Electronics, R. Fonte Caspolina, N.6,2.C, 2774-521, PACO DE ARCOS, Portugal.
GRI	Grinpal Energy Management, 50 Oak Avenue, Pretoria, South Africa
GSP	Ingenieurbuero Gasperowicz
GST	Shenzhen Golden Square Technology Co.,Ltd, Zone C&D,5/F,Block A3,Shenzhen Digital Technology Park,Hi-Tech South 7 Rd.,Nanshan ,Shenzhen,Guangdong, China
GTE	GREATech GmbH, Lindenstrasse 66a, 45478 Muelheim an der Ruhr, Germany
GTS	GIGATRONIK Stuttgart GmbH, Hortensienweg 21, 70374, Stuttgart, Germany
GUH	ShenZhen GuangNing Industrial CO.,Ltd, Room 802, 8th Floor, ShenZhen Software Building ,NanShan, District,ShenZhen ,China
GWF	Gas- u. Wassermesserfabrik Luzern
HAG	Hager Electro GmbH, 66131 Saarbruecken, Germany
HCE	Hsiang Cheng Electric Corp, Hsin-Tien City, Taipei, R.o.China
HEG	Hamburger Elektronik Gesellschaft
HEI	Hydro-Eco-Invest SP. Z O.O., Gliwice, Poland
HEL	Heliowatt
HER	Hermes Systems, Australia
HFR	SAERI HEAT METERING TECHNOLOGY CO.,LTD, WANLIAN ROAD 1,SHENHE DISTRICT SHENYANG CHINA
HIE	Shenzhen Holley South Electronics Technology Co., Ltd., 7/F, No.2 Jianxing Building, Chaguang Industrial Zone, Nanshan District, Shenzhen, China
HLY	Holley Metering Ltd

HMI	HMI Energy Co., Ltd, No.38, Alley 175, Lane 75, Sec3, Kongning Rd., Neihu, Taipe, Taiwan
HMS	Hermes Systems, Australia
HMU	Hugo Müller GmbH & Co KG, Sturmbühlstraße 145-149, 78054 VS-Schwenningen, Germany
HND	Shenzhen Haoningda Meters Co., Ltd., 6/F, Huake Mansion, East Science Park, Qiaoxiang Rd, Nanshan District, Shenzhen, China
HOE	HOENTZSCH GMBH, Gottlieb-Daimler-Str.37, 71334 Waiblingen, Germany
HOL	Holosys d.o.o, Matije Gupca 7, 49243, Oroslavje, Croatia
HON	Honeywell Technologies Sarl, Ecublens, Switzerland
HPL	HPL-Socomec Pvt. Ltd., 133 Pace City 1, Sector 37, Gurgaon, India
HRM	Hefei Runa Metering Co., Ltd, 1102# jinchi Rd. Luyang industrial park, Hefei, Anhui Province, Hefei, CHINA
HRS	HomeRider SA, France
HSD	Ningbo Histar Meter Technology Co.,Ltd., No.181 Haichuan Road Jiangbei District, Ningbo City, Zhejiang Province, CHINA
HST	HST Equipamentos Electronicos Ltda
HTC	Horstmann Timers and Controls Ltd.
HTL	Ernst Heitland GmbH & Co. KG, Erlenstr. 8-10, 42697 Solingen, Deutschland
HTS	HTS-Elektronik GmbH
HWT	Huawei Technologies Co. Ltd., Department of Industry Standards, Huawei Industrial Base, Shenzhen, China
HXD	Beijing HongHaoXingDa Meters CO.,LTD, HouXing,the third street,18,HuoXian, TongZhou., Beijing, China(P.R.C)
HXE	Hexing Electrical Co., Ltd, Hangzhou, China

HXW	Hangzhou Xili Watthour Meter Manufacture Co. Ltd., No. 14, JiaQi Road, XianLin Industrial Park, Yuhang District, Hangzhou, China
HYD	Hydrometer GmbH
HYE	Zhejiang Hyayi Electronic Industry Co Ltd, Zhejiang, China
HYG	Hydrometer Group, 91522 Ansbach, Germany
HZC	TANGSHAN HUIZHONG INSTRUMENTATION CO., LTD., Qinghua Road,New and Hi-Tech Development, Zone,Tangshan,Hebei Province,China, Tangshan, China
HZI	TANGSHAN HUIZHONG INSTRUMENTATION CO., LTD., Qinghua Road,New and Hi-Tech Development, Zone,Tangshan,Hebei Province,China, Tangshan, China
ICM	Intracom, Greece
ICP	PT Indonesia Comnets Plus, PLN building 9th Floor Jl.Jendral Gatot Subroto kav.18, Jakarta Selatan, Indonesia
ICS	ICSA (India) Limited, Plot No. 12, 1st Floor, Software units Layout, Cyberabad, Hyderabad, India
ICT	International Control Metering-Technologies GmbHWillhoop 7, D-22453 Hamburg, Germany
IDE	IMIT S.p.A
IEC	leonnardo Corporation, Peremogy, 31, Sutysky, Ukraine
IEE	I.E. Electromatic, S.L, Quart de Poblet (Valencia), Spain
IFX	Infineon Technologies, AM Campeon 1-12, Nuebiberg, Germany
IHM	Shenzhen Inhemeter Co., Ltd., 7/F, Science & Industry Park Building, Science & Industry Park, Nanshan District, Shenzhen, China
IJE	ILJIN Electric, Kyunggi-Do, Korea
IKS	IKASIAN, Av. Josep Tarradellas, 38. SBC Office 55, Barcelona, Spain
IMS	IMST GmbH, Carl-Friedrich-Gauss-Straße 2-4, 47475 Kamp-Lintfort, Germany
INC	Incotex, 16th Parkovaya st, 26, Moscow, Russia

IND	INDRA SISTEMAS, Avda. Bruselas, 35, Alcobendas (Madrid)
INE	INNOTAS Elektronik GmbH, Rathenaustr. 18a, 02763 Zittau, Germany
INF	Infometric, Sollentunavägen 50, 19140 Sollentuna, Sweden
INI	Altero AB, 21211 Malmoe, Sweden
INP	INNOTAS Produktions GmbH, Rathenaustr. 18a, 02763 Zittau, Germany
INS	INSYS MICROELECTRONICS GmbH, Hermann-Köhl-Str. 22, 93049, Regensburg, GERMANY
INT	Infranet Technologies GmbH, 21079 Hamburg, Germany
INV	Sensus Metering Systems, Ludwigshafen/Rh, Germany
INX	Innolex Engineering BV, Molenlei 2A, Akersloot, The Netherlands
IPD	IPD Industrial Products Australia, Sydney, Australia
ISI	Akcionarsko Drustvo "Insa Industrija Satova, Trscanska 21, Belgrade-Zemun, Serbia
ISK	Iskraemeco, Slovenia
ISO	Isoil Industria spa, via F.lli Gracchi n.27, Cinisello Balsamo (Milan), Italy
IST	Ista
ITA	iTrona GmbH, CH-6432 Rickenbach SZ, Switzerland
ITB	ITRON Brazil, rua Fioravante Mancino, 1560, CEP: 13175-575 Sumaré, Brazil
ITC	INTECH TUNISIE, Rue de Tozeur ZI Hammam Zriba, Zaghouan, Tunisia
ITE	ITRON (Electricity), 52, Avenu Camille Desmoulin, 92130 Issy les Moulineaux, FRANCE
ITG	ITRON (Gas), 52, Avenue Camille Desmoulin, 92130 Issy les Moulineaux, FRANCE
ITH	INTELTEH d.o.o., Bozidara Magovca 87, 10000, Zagreb, Croatia
ITF	ITF Fröschl GmbH, Hauserbachstrasse 9, 93194 Walderbach, Germany

ITI	ITRON Asia, EJIP Plot 6B-2, Lemah Abang, Bekasi 17550, Jawa Barat, Indonesia
ITR	Itron
ITS	ITRON Australia, Rosberg Road, Wingfield, SA, 5013, Adelaide, Australia
ITU	ITRON United States, 2111 N Molter Road, Liberty Lake, WA 99019, United States
ITW	ITRON (Water), 52, Avenue Camille Desmoulin, 92130 Issy les Moulineaux, FRANCE
ITZ	ITRON South Africa, Tygerberg Office Park, Hendrik, Verwoerd Drive, 7500 Plattekloof, Cape Town, South Africa
IUS	IUSA SA DE CV, Km 109 Carr Panamericana, Pasteje Jocotitlan Edo. de Mex., Mexico
IWK	IWK Regler und Kompensatoren GmbH
IZE	iZenze AB, Slottagårdsgatan 9, 235 35 Vellinge, Sweden
JAN	Janitza electronics GmbH, Lahnau-Waldgirmes, Germany
JCE	Janz Contadores de Energia SA, Lisbon, Portugal
JED	JED Co Ltd, Dongan-Gu, Anyang, Kyunggi-Do, South Korea
JGF	Janz Contagem e Gestao de Fluidos SA, Lisbon, Portugal
JHM	Changzhou Jianhu Intelligentize Meter Co.,Ltd., No.11 Lijia Industrial District,Wujin, Changzhou, China
JMT	JM-TRONIC Sp. z o.o., ul. Wybrze e Ko ciuszkowskie 31/33, 00-379 Warszawa, Poland
JOY	Zhejiang Joy Electronic Technology Co., Ltd., No. 333 North Chayuan Road, Youchegang Town, Xiuzhou District, Jiaxing, China
JUM	JUMO GmbH & Co. KG, Herrmann - Muth - Strasse 1, 36039 Fulda, Germany
KAM	Kamstrup Energi A/S
KAS	Kamstrup A/S, Industrivej 28, 8660 Skanderborg, Denmark
KAT	KATHREIN-Werke KG, Anton-Kathrein-Straße 1-3, D-83022 Rosenheim, Germany
KBN	Alpamis IT Ltd, ANKARA, Turkey

KEL	KELEMINIC d.o.o., Zagreb, Croatia
KER	KERMS UG (haftungsbeschränkt), Fontanestraße 39, 15569 Woltersdorf, Germany
KFM	Shenzhen KAIFA Technology Co, Ltd, Shenzhen, China
KGE	Guangzhou Keli General Electric Co., Ltd, No1.Sui Hua Nan street, Jiang Nan Da Dao Zhong Road, Guangzhou P.R.China
KHL	Kohler, Turkey
KKE	KK-Electronic A/S
KMB	Kamstrup A/S, Industrivej 28, Stilling, DK 8660 Skanderborg, Denmark
KMT	Krohne Messtechnik GmbH, Ludwig-Krohne-Straße, Duisburg,Germany
KNX	KONNEX-based users (Siemens Regensburg)
KRO	Kromschroder
KST	Kundo SystemTechnik GmbH
KSY	KUNDO SystemTechnik GmbH, St Georgen, Germany
KTC	Kerman Tablo Co, Tehran, Iran
LAC	Heinz Lackmann GmbH & Co KG, Harkortstrasse 15, 48163 Münster, Germany
LAN	Langmatz GmbH, Am Gschwend 10, Garmisch-Partenkirchen, Germany
LCG	Landis+Gyr Meter & System (Zhuhai) Co., Ltd, No.12 Pingdong 3RD,Nanping Industry Community,Zhuhai 519060, P.R.China
LCR	ShanDong LiChuang Science and Technology Co.,Ltd, No. 9 Fenghuang Road High-tech District, Laiwu ShandongChina
LDE	Shenzhen Londian Electrics Co., Ltd, 3/F, Build 107#, 1st Nanyou Industrial Zone, Nanshan District, Shenzhen, China
LEC	Lectrotek Systems Pvt Ltd, 33 Parvati Industrial Estate, 411009, Pune, India
LEM	LEM HEME Ltd., UK

LFS	Payolcer Metering Devices, Erciyes Un versity Cybertech Center-4 Num:25, KAYSERI, TURKEY
LGB	Landis+Gyr Ltd., UK
LGD	Landis+Gyr GmbH, Germany
LGS	Landis+Gyr (Pty) Ltd. South Africa
LGU	LG Uplus Corp, Namdaemunno 5-ga, Jung-gu, Seoul, Korea
LGZ	Landis+Gyr AG Zug
LHA	Atlantic Meters, South Africa
LML	LUMEL, Poland
LNT	Larsen & Toubro Ltd, MPS, KHebbal-Hootagalli, Mysore, India
LSE	Landis & Staefa electronic
LSK	LS Industrial Systems Co Ltd, Cheongju, South Korea
LSP	Landis+Gyr GmbH, Germany
LSZ	Siemens Building Technologies
LUG	Landis+Gyr GmbH, Germany
LUN	Protokol Sanayi ve Ticaret, Karacaoglan Mah., 167 Sok., No 42 Isikkent, Izmir, Turkey
LYE	Jiangsu Linyang Electronics Co.,Ltd., No.666,Linyang Road, Qidong, China
MAC	RUDNAP Group Meter & Control, Omladinskih brigada 182, Belgrade, Serbia
MAD	Maddalena S.p.A.. Italy
MAE	Mates Elektronik Metin Ates, Ankara, Turkey
MAN	Manthey GmbH, Walter-Freitag-Str. 30, 42897 Remscheid, Deutschland
MAT	Mitsubishi Electric Automation, Bangkok, Thailand

MAX	MAXMET Inc, Seogu, Daejeon 302-834, Korea
MBS	MBS AG, Eisnachstraße 51, 74429 Sulzbach-Laufen, Germany
MCR	MICRORISC, D Inická 222, Ji ín, Czech Republic
MDE	Diehl Metering Deutschland, Industriestraße 13, Ansbach, Germany
MEC	Mitsubishi Electric Corporation, 1-8, Midorimachi Fukuyama-city Hiroshima, 720-8647, Japan
MED	MAHARASHTRA STATE ELECTRICITY DISTRIBUTION COMPANY LIMITED, PLOT NO. G-9, PRAKASHGAD, 5TH FLOOR, PROF. ANANT KANEKAR MARG, BANDRA (EAST)MUMBAI 400051, INDIA
MEH	Mueller-electronic GmbH, Fritz-Garbe-Str. 2, 30974, Wennigsen, Germany
MEI	Sensus Metering Systems, Ludwigshafen/Rh, Germany
MEL	Mikroelektronika a.d, Banja Luka, Bosnia and Herzegovina
MEM	MEMS AG, Segelhofstrasse, CH-5405 Baden-Dättwil, Switzerland
MET	METRA Energie-Messtechnik GmbH, Speyer, Germany
MIC	Microtronics Engineering GmbH, Hauptstrasse 7, A-3244 Ruprechtshofen, Austria
MII	Miitors ApS, Miitors ApS, VBI Park, Chr. M Østergaardsvej 4aDK-8700 Horsens, Denmark
MIM	Malaysian Intelligence Meters Sdn. Bhd., No. 3, Jalan Pemberita U1/49, Temasya Industrial Park, Seksyen U1, Glenmarie Shah Alam, Selangor Darul Ehsan, Malaysia
MIR	MIR Research and Production Association, 51 Uspeshnaya644105 Omsk, Russia
MIS	Iskra MIS d.d, 4000 Kranj, Slovenia
MKE	MKEK Genel Mudurlugu Gazi Fisek (ELSA), Ankara, Turkey
MKL	MAKEL Elektrik Malzemeleri, Turkey
MKS	MAK-SAY Elektrik Elektronik, Turkey
MMC	Modern Meters Co., Damascus Sahnaya, Syria,

MMI	MyMeterInfo, 95 rue du Morellon, 38070 Saint Quentin, Fallavier, France
MMS	Brunswick Bowling and Billiards UK Ltd, Unit L1, Temple Court, Knights Park, Knight Road, Strood, Kent, UK
MNS	MANAS Elektronik, Turkey
MOS	MOMAS SYSTEMS NIGERIA LIMITED, #4, Bode Thomas Street, Surulere, Lagos, NIGERIA
MOT	The Motwane Manufacturing Company Private Limited, Gyan Baug, Motwane Road, Nasik Road, Nasik, India
MPA	Mega Power Automation International Limited, 16/F., Block A-1, Fortune Factory Building, 40 Lee Chung Street, Chai Wan, Hong Kong.
MPR	Michael Rac GmbH, Sonnenfeld 29, 91522 Ansbach, Germany
MPS	Multiprocessor Systems Ltd, Bulgaria
MRT	MIRTEK LTD., Gagarin street, Building 4, Stavropol, Russia
MSB	MISA SDN BHD, LOT 30, JALAN MODAL 23/2, 40300, SHAH ALAM, SELANGOR, MALAYSIA
MSE	Mahashakti Energy Limited, A-8, New Focal Point, Dabwali Road, Bathinda (Punjab), India
MSM	MS-M Co., Ltd., 237 Bukjung-Dong Yangsan-City, Korea
MSO	Metiso, Tr anska 21, Zemun, Belgrade, Serbia
MSY	MAK-SAY Elektrik Elektronik Malzemeleri, Turkey
MTD	Removed - November 2006
MTC	Metering Technology Corporation, USA
MTH	njmeter, Binjiang Development Zone of Jiangning Road No. 6Nanjing, China
MTI	Micrtotech Industries Pakistan, Plot#2,Street#2,Attari industrial estate 18-Km. Ferozepure Raod, Lahore, Pakistan
MTM	Metrum Sweden AB, Vestagatan 2A, Gothenburg, Sweden
MTS	MeteRSit S.r.L., Viale dell"Industria, 31, Padova, Italy

MTX	Matrix Energy Pvt. Ltd., Soni Arcade, No. 242, 2nd Floor, 7th Cross, 6th Block, Banashankari 3rd Stage, Bangalore, INDIA
MUK	Meters UK Ltd, Whitegate, White Lund Trading Estate, Lancaster, UK
MWU	METRONA Wärmemesser Union GmbH, Aidenbachstraße 40, 81379 München, Germany
MXM	Maxim India Integrated Circuit Design Pvt Ltd., 132/133, Divyasree Technolopolis, Off Airport Road, Bangalore, India
NAR	NARI Group Corporation-NARI Technology Development Co., Ltd, No.8 NARI Rd. Gulou District, Nanjing, China
NDF	NÚCLEO DURO FELGUERA, Avda. de la Industria, 24, 28760, Tres Cantos, Madrid
NDM	Northern Design, 228 Bolton Road, Bradford, United Kingdom
NES	NORA ELK. MALZ. SAN. ve T C. A. ., nönü Cad. Sümer Sok. Zita Merkezi C1 Blok No:9 Kozyata -Kad köy- STANBUL, TURKEY
NIS	Nisko Industries Israel
NJC	NAMJUN Co Ltd, Gimhae Gyoungnam, South Korea
NMS	Nisko Advanced Metering Solutions Israel
NNT	2N Telekomunikace a.s., Modranska 621, 14301 Praha 4, Czech Republic
NRM	Norm Elektronik, Turkey
NTC	Nuri Telecom Co Ltd, Geumcheon-gu, Seoul, Korea
NTM	Netinium, Postbus 86, Wormerveer, The Netherlands
NVN	NOVEN ENERGY AND ICT LTD., Hacettepe University KOSGEB Technology Center T1-Blok B:14 Beytepe, Ankara,Turkiye
NWM	Ningbo Water Meter Co.Ltd., No.99 Lane 268 Beihai Road Jiangbei District, Ningbo City Zhejiang ProvinceCHINA
NXP	NXP Semiconductors, High Tech Campus 32, 5656AE Eindhoven, The Netherlands
NYG	Ningbo Yonggang Instrument Co.,Ltd, Weisan Road, West Industrial Zone, Xinpu Town, Cixi City, China

NYN	Nanjing Yuneng Instrument Co Ltd, Nanjing, China
NZR	Nordwestdeutsche Zählerrevision Ing. Aug. Knemeyer GmbH & Co. KG, Heideweg 33, 49196, Bad Laer, Germany
OAS	Omni Agate Systems, Chennai, India
ODI	OAS Digital Infrastructures Pvt. Ltd., No:4/3, Stringer Road, Periamet, Vepery, Chennai, INDIA
OEE	ONUR Elektrik ve Elektronik, Turkey
OMS	OMNISYSTEM Co., Ltd., Goyang-shi, Gyeonggi-do, Korea
ONR	ONUR Elektroteknik, Turkey
ONS	ONUR Elektroteknik, Turkey
OPT	Optec GmbH, Grundstrasse 22, 8344 Bäretswil, Switzerland
ORB	ORBIS Tecnologia Electrica, SA, Madrid, Spain
ORM	Ormazabal, B Basauntz, 2, Igorre, Spain
OSA	Osaki Electric Co., Ltd.(Europe), Gotanda-Square, Tokyo, Japan
OSK	Osaki Electric Co Ltd. (Japan), Shinagawa-ku, Tokyo, Japan
OZK	Oz-kar Enerji, Kayseri, Turkey
PAD	PadMess GmbH, Germany
PAF	FAP PAFAL S.A., 26 Lukasinskiego street, widnica, Poland
PAK	Paktim Energy Sp.zo.o., ul. Swiety Marcin 29/8, 61-806 Poznan, Poland
PAN	Panasonic Corporation, 800 Tsutsui-cho, Yamatokoriyama-shiNara Pref., Japan
PDX	Paradox Engineering SA, Via Ronchetto, 9, Cadempino, Switzerland
PEL	Pak Elektron Ltd. (PEL), 14-km Ferozpur Road, Lahore, Pakistan
PGP	P.G.P. - Smart Sensing s.a., Rue Fond Cattelain 2 / 1.15, Mont-St-Guibert, Belgium

PHL	HangZhou PAX Electronic Technology Co., Ltd., China
PII	PiiGAB Processinformation i Goteborg AB, Sweden
PIK	pikkerton GmbH, Kienhorststrasse 70, 13403 Berlin, Germany
PIL	Pilot Systems (London) Ltd, Chiswick, London
PIM	Power Innovation GmbH, Rehland 2, Achim,Germany
PIP	Hermann Pipersberg jr., Felder Hof 2, D-42899 Remscheid, Deutschland
PLO	Weihai Ploumeter Co. Ltd., : No. 28 Hengrui Street, Torch Hi-Tech Industries Development Zone, Weihai, Shandong, China
PMG	Sensus Metering Systems, Ludwigshafen/Rh, Germany
PMS	PMS-Elektronik GmbH, Humboldtstraße 14, D-74915, Waibstadt, Germany
POD	PowerOneData, Bangalore 560082, India
POW	PowerApp, Esromgade 15 opg. 2 - 2 sal., Copenhagen, Denmark
POZ	ZEUP Pozyton sp. z o.o, ul. Czestochowa, Poland
PPC	Power Plus Communications AG, Am Exerzierplatz 2, 68167 Mannheim, Germany
PRE	Predicate Software, 7 Protea Ave, Dooringkloof, 0140, Centurion, Gauteng, South Africa
PPS	Palace Power Systems, 50 Oak Avenue, Pretoria, South Africa
PRI	Polymeters Response International Ltd.
PRG	Paud Raad Industrial Group, No. 18, 2nd St., Shah Nazari Ave.,Madar Sq., Mirdamad Blvd., Tehran, Iran
PRO	Proton - Elektromed Ltd, Ankara, Turkey
PST	PSTec Co.,Ltd, Seoul, Korea
PUK	Paktim Consulting UK Ltd, 2 West Regent Street, Glasgow, G2 1RW, United Kingdom
PWB	Paul Wegener GmbH, Ballenstedt, Germany

PWR	Powrtec, Scotts Valley, CA 95066, USA
PYU	PYUNGIL Co. Ltd, Anyang-si, Gyeonggi-do, Korea
QDS	Qundis GmbH, Sondershaeuser Landstrasse 27, Mühlhausen, Germany
QTS	QT systems ab, Alfavägen 3, 92133 Lycksele, Sweden
RAC	Michael Rac GmbH, Sonnenfeld 29, Ansbach, Germany
RAD	Radiocrafts AS, Sandakerveien 64, 0484 OSLO, NORWAY
RAM	Rossweiner Armaturen und Messgeräte GmbH & Co. OHG, Wehrstraße 8, Roßwein, Germany
RAS	Rubikon Apskaitos Sistemos, Vilnius, Lithuania
RCE	RC ENERGY METERING PVT,LTD., B-65 GATE NO.1,N.I.A.PHASE-II NEW DELHI -110028, DELHI, INDIA
REC	Zhejiang Reallin Electron Co.,Ltd, 2F,Building3,No.202 Zhenzhong Rd,Sandun Technology Park,Xihu District, Hangzhou, China
REL	Relay GmbH, Germany
REM	Remote Energy Monitoring, Tring, UK
RIC	Richa Equipments Pvt. Ltd., Z B 5-6/487, Zulfe Bengal, Dilshad Garden Shahdara, Delhi, India
RIL	Rikken Instrumentation Limited, Plot No. 369, Phase 2, Industrial Area, Panchkula, India
RIM	CJSC \"Radio and Microelectronics\", 630082, Novosibirsk, the Red Prospectus, 220, the case 17, Novosibirsk, Russia
RIT	Ritz Instrument Transformers GmbH, Wandsbeker Zollstr. 92 98, 22041 Hamburg, Germany
RIZ	RIZ Transmitters, Bozidareviceva 13, Zagreb, Croatia
RKE	Viterra Energy Services (formerly Raab Karcher ES)
RML	ROLEX METERS RPIVATE LIMITED, Plot No 20&21, Prashanthi Nagar, Kukatpally Industrial Estate, Hyderabad, India

RMR	Advanced Technology RAMAR, Christchurch, UK
RSA	Rahrovan Sepehr Andisheh Pte. Co.
RSW	RSW Technik GmbH, Giessen, Germany
SAA	Sanjesh Afzar Asia Ltd. Co., 3 rd Flr/No. 8/16 St./Gandi Ave./Tehran, Iran
SAC	Sacofgas 1927 SpA, Via Ascanio Sforza 85, Milano, Italy
SAG	SAGEM, Cergy Saint-Christophe, France
SAM	Siemens AG Österreich, AMIS (Automated Metering and Information System), Ruthnergasse 3, Vienna, Austria
SAP	Sappel
SAT	SATEC Ltd, 7 Hamarpe Street, Jerusalem, Israel
SBC	Saia-Burgess Controls, Bahnhofstrasse 18, 3280 Murten, Switzerland
SCE	Seo Chang Electric Communication Co Ltd, Daegu, Korea
SCH	Schinzel GmbH
SCW	ScatterWeb GmbH, Charlottenstr. 16, Berlin, Germany
SDC	SdC Sistemas de Contagem, Vila Nova de Famalicao, PT
SDM	Shandong Delu Measurement Co., Ltd., Tower C, Qiln Software Park, High-Tech Industrial Development, JiNan, China
SEC	Schneider Electric Canada, Saanichton, Canada
SEE	El Sewedy Electrometer Egypt, 6th of October, Egypt
SEN	Sensus Metering Systems, Ludwigshafen/Rh, Germany
SGA	smartGAS Mikrosensorik GmbH, Kreuzenstraße 98, 74076 Heilbronn, Germany
SGM	Swiss Gas Metering AG, Reichenauerstrasse, Domat/Ems, Switzerland
SHD	Beijing SanHuaDeBao Energy Technoligy Co.,Ltd., Floor4 Jinyanlong R&D Building Jiancaicheng West Road Changping District Beijing City China

SHE	Shenzhen SingHang Elec-tech Co., Ltd., Rm203-206, Terra Science & Technology Park, Futian District, Shenzhen, China
SHM	Shanghai Metering, China, No.2065 Kongjiang Road, Shanghai, China
SIE	Siemens AG
SIT	SITEL doo, Belgrade, Serbia and Montenegro
SLB	Schlumberger Industries Ltd.
SLP	Sylop, ul. Jagiellonska 4, PL-32830 Wojnicz, Poland
SLX	Ymatron AG, Bruelstrasse 7, Dielsdorf, Switzerland
SMC	Pending
SME	Siame, Tunisia
SMG	Samgas s.r.l., SP 33 km 0,600 20080, Vernate (MI), Italy
SML	Siemens Measurements Ltd.
SMN	Saiman Corporation LLC, 162d Shevchenko Street, Almaty, Kazakhstan
SMT	Smarteh d.o.o., Trg tigrovcev 1, Tolmin, Slovenia
SNM	ShenZhen Northmeter Co.Ltd , floor 5, Dongshan Building, Huafeng first science park,Baoan, Shenzhen, China
SNR	NTN-SNR, 1 Rue des Usines, 74010 Annecy, France
SNS	Signals and Systems India Private Limited, MF-7, Cipet Hostel Road, Thiru-Vi-Ka Industrial Estate, Chennai, India
SOC	SOCOMEC, 1, rue de Westhouse, 67230 Benfeld, France
SOF	Softflow.de GmbH, Dorfstasse, 15834 Gross Machnow, Germany
SOG	Sogecam Industrial, S.A., C/ Rosalind Franklin, 22-24, Campanillas (Málaga), Spain
SOL	Soledia Srl, Via di Selva Candida 85, Rome, Italy
SOM	Somesca, 80 rue Jean Jaures, 92270 Boois colombes, France

SON	Sontex SA
SPL	Sappel
SPX	Sensus Metering Systems, Ludwigshafen/Rh, Germany
SRE	Guangzhou Sunrise Electronics Development Co., Ltd., Guangzhou Avenue South, Guangzhou, Guangdong, China
SRF	Saraf Industries, Saraf Industries, Bathinda Road, Rampura Phul - 151103, Punjab, India
SRN	Shandong SARON Intelligent Technology Co., Ltd., 3F,South E,International Business Center, Environmental Technology Area, Middle of Zhengfeng Road, Jinan City Shandong Province,China
SRV	Servic LLC, Kirova 16-9, Dniprodzerzhinsk, Ukraine
SSN	Silver Spring Networks, 555 Broadway Street, Redwood City, United States
SST	Qingdao Haina Electric Automation Systems Co., Ltd., No.151,Zhuzhou Road, Laoshan, Qingdao, China
STA	Shenzhen Star Instrument Co Ltd, Shenzhen, China
STC	Sunrise Technology Co., Ltd, Building C, Xiyuan 8th Road 2#, West-Lake Technological & Economic Zone, Hangzhou, China
STD	Stedin, Essebaan 71, Capelle a/d Ijssel, Netherlands
STR	Strike Technologies, South Africa
STV	STV Automation, Branch of STV Electronic, Detmold, Germ.
STZ	Steinbeis Innovation Center Embedded Design and Networking c/oUniversity of Cooperative Education Loerrach Hangstrasse 46-50, D79539 Loerrach
SVM	AB Svensk Värmemätning SVM
SWI	Swistec GmbH, Graue-Burg-Strasse 24-26, Bornheim, 53332 Germany
SWT	Beijing Swirling Technology Co. Ltd, Beijing, China
SYN	SMSISTEM Ltd., ANKARA, Turkey

TAG	Telma AG, Gewerbeweg 10, 3662 Seftigen, Switzerland
TAT	Tatung Co., 22, Chungshan N. Rd., 3rd Sec., Taipei, Taiwan
TAY	Taytech Otomasyon ve Bilisim Teknolojileri LTD. Sti, Tasdelen Gungoren Mah. Izan Sok. No:15 Cekmekoy., Istanbul, Turkey
TCE	Qindao Techen Electronic Technology Co.,LTD, No. 169 Songling Road. Laoshan District, Qingdao, China
TCH	Techem Service AG & Co. KG
TDC	Telecom Design, Rue Romaine Voie de Remora, GradignanFRANCE
TEC	TECSON Digital, Felde, Germany
TEP	TEPEECAL, 69730 Genay, France
TFC	Toos Fuse Co. 375 Sanat Blvd. Toos Industrial Estate, Mashad, Iran
THE	Theben AG, Hohenbergstrasse 32, 72401 Haigerloch, Germany
TIP	TIP GmbH, Bahnhofstr. 26, 99842 Ruhla, Germany
TIX	Tixi.Com GmbH, D-13465 Berlin
TLM	Theodor Lange Messgeräte GmbH, Rodeberg 7, 31226 Peine,Germany
TMK	Timi Kosova Sh.p.k.
TMS	TEMASS IMALAT A.S, Macunkoy, Ankara, Turkey
TPL	Teplocom Holding, 45, Vyborgskaya Naberezhnaya, ST Petersburg, Russian Federation
TRI	Tritech Technology AB, Sturegatan, 10-12 PO Box 1094, SE-172 22 Sundbyberg, Stockholm, Sweden
TRJ	SHENZHEN TECHRISE ELECTRONICS CO.,LTD, Building 112,1st Industrial park, Liantang, Luohu District, Shenzhen City, China.
TRL	Trilliant Inc., 610 du Luxembourg, Granby, (Quebec), Canada, J2J 2V2
TRV	Transvaro Elektron Aletleri A.S., Turkey

TSD	Theobroma Systems Design und Consulting GmbH, Gutheil-Schoder Gasse 17, Wien, Austria
TTM	Toshiba Toko Meter Systems Co., Ltd., 12-7, Shiba 1-chome, Minato-ku, TOKYO, JAPAN
TTR	Tetraedre Sarl, Epancheurs 34b, 2012 Auvernier, Switzerland
TTT	Telephone and Telegraph Technique Plc, Sofia, Bulgaria
TUR	TURKSAY ELEKTRONIK ELEKTRIK ENDUSTRISI
TXL	CETC46 TianJin New Top Electronics Technology Co.,Ltd., KEYAN East Road 15,Nankai District, Tianjin, China
UAG	Uher
UBI	Ubitronix system solutions gmbh, 4232 Hagenberg, Austria
UEI	United Electrical Industries Limited, Pallimukku, Kollam, India
UGI	United Gas Industries
UTI	Utilia Spa, Via Chiabrera, 34/D, Rimini (RN 47924), Italia
UTL	United Telecoms Limited, A-1/A, 2nd Floor, Revati Building, ECIL XRoads. Hyderabad, INDIA
VER	VERAUT GmbH, Siemensstr.52, Linz, Austria
VES	Viterra Energy Services
VIK	VI-KO ELEKTRIK, Istanbul, Turkey
VMP	VAMP Oy., Yrittäjänkatu 15 P.O. Box 810 FI-65101, Vaasa, Finland
VPI	Van Putten Instruments B.V.
VSE	Valenciana Smart Energy of Mediterranean Sea S.A, Sir Alexander Fleming, 12 . Warehouse 11, Parque tecnologico de Valencia, 46980, Valencia, Spain
VTC	Vitelec, Kapittelweg 18, NL 4827 HG Breda, Postbus 6543, NL 4802 HM Breda, Netherlands

VTK	Linkwell Telesystems Pvt. Ltd., Gowra Kalssic, 1-11-252/1/A, Begumpet, Hyderabad 500016, India
YTL	ZheJiang yongtailong electronic co.,ltd, No.8 KangDing Road, Tongxiang, China
WAH	WAHESOFT UG, Moeoerte 16, 26316 Varel, Germany
WAI	Chongqing WECAN Precision Instruments Co.,Ltd, #66 HuangShan Rd,HI-Tech Park , New North Zone, ChongqingP.R.China
WAL	Wallaby Metering Systems Pvt. Ltd., M-3, 9th Street, Dr.VSI Estate, Thiruvanmiyur, Chennai - 600 041, INDIA
WDN	Webdyn SA, 26 rue des Gaudines, 78100 Saint Germain en Laye, France
WEB	Webolution GmbH & Co. KG, Sendenhorsterstrasse 32, 48317 Drensteinfurt, Germany
WEG	WEG Equipamentos Elétricos S.A. Automação, Av. Pref. Waldemar Grubba, 3000, Jaraguá do Sul, Brazil
WEH	E. Wehrle GmbH, Obertalstraße 8, 78120 Furtwangen, Germany
WEL	WELLTECH automation, 263# HongZhong Road, Shanghai,P.R.China
WFT	Waft Embedded Circuit Solutions, A-109, Sahni Tower, Sector-5, Rajendra Nagar, Sahibabad, Ghaziabad (U.P.), India
WIN	Windmill Innovations BV, Paasbosweg 14-16, 3862 ZS, Nijkerk (GLD), The Netherlands
WMO	Westermo Teleindustri AB, Sweden
WSD	Yantai Wisdom Electric Co., Ltd., JiChang road 2#, Yantai, Shangdong Province, China
WSE	Changsha Weisheng Electronics Ltd, Changsha, P.R. China
WTI	Weihai Sunts Electric Meter Co., Ltd, 39-7#, Shenyang Middle Rd. Gaoji Weihai CHINA
WTL	Wipro Technologies, Doddakannelli, Sarjapur Road. Bangalore, India
WTM	Watertech S.r.l., Strada dell"Antica Fornace, 2/4, 14053 Canelli (At) Italy
WZG	Neumann & Co. Wasserzähler Glaubitz GmbH, Industriestraße A7, 01612 Glaubitz, Germany / Deutschland

WZT	Wizit Co Ltd, Ansin-City Gyeonggi-Do, S Korea
XAO	Info Solution SpA, Via della Burrona, 51, Vimodrone (MI), Italy
XEM	XEMTEC AG, Sarnen, Switzerland
XJM	XJ Metering Co., Ltd, No 416, Ruixiang Road, Xuchang, Henan, China
XMA	XMASTER s.c. ul. Gersona 41, Wroclaw, Poland
XMX	Xemex NV, B-2900 Schoten, Belgium
XTR	HENAN SUNTRONT TECH CO., LTD, No.19 Guohuai Street, High and New Tech Industrial Development Zone, Zhengzhou City, Henan Province, China
XTY	LianYuanGang Tengyue Electronics & Technology Co.LianYunGang.Jiangsu.china Haizhou, LianYunGang, China
YDD	Jilin Yongda Group Co., Ltd
YHE	Youho Electric Co Ltd, Yangjoo, South Korea
YSS	Yellowstone Soft, Brunnenstr. 32, 89584 Ehingen, Germany
YTE	Yuksek Teknoloji, Turkey
ZAG	Zellweger Uster AG
ZAP	Zaptronix
ZEL	Dr. techn. Josef Zelisko GmbH, Beethovengasse 43 45A-2340 Mödling, Austria
ZIV	ZIV Aplicaciones y Tecnologia, S.A.
ZJY	Zhejiang Jiayou Thermal Technology Equipment Co., LTD, Houwan Industrial Point,Yucheng Street,Yuhuan County, Taizhou, China
ZPA	ZPA Smart Energy a.s., Komenského 821, CZ-541 01 TrutnovCzech Republic
ZRI	ZENNER International GmbH & Co. KG, Postfach 10 33 39D-66033 Saarbrücken, Germany
ZRM	ZENNER International GmbH & Co. KG, Postfach 10 33 39D-66033 Saarbrücken, Germany

ZTY	Hangzhou Chint Meter Technology Co., Ltd, 7th Floor, New Building, No 313, Tianmushan Road, Hangzhou 310013, China

MTX-Tunnel Release Notes

V5.0

- First release of MTX-Tunnel v5

V5.1

- Private VPNs are supported using private APN (SIM cards must be provisioned)
- MTX_msToSend parameter description missed out in the MTX-Tunnel v5.0 manual

V5.2

- Text string modification to activate MTX-Tunnel.

  Previous version: Just SMS with text string “on” was enough.

  Current and later versions: Text string must be “mtxtunnel on” and has to be at the beginning of the SMS text string message.
- SMS remote AT commands

  Previous versions: Any AT command was considered if the SMS string text began with “AT”.

  Current and later versions: Text SMS string must be: “mtxtunnel [ATcommand]”.
- Default MTX parameter values are modified as follows:

  SMS_allPhones: off

  SMS_sendIP: off

  SMS_ATEnabled: off

  SMS_ATResponse: off

  FIREWALL_enabled: on
- Fixed bug: After power up, if the GPIO is configured as a high input level, it cannot be detected without being changed to a lower input level

V5.3

- DNS_gpios, DNS_adc1 and DNS_adc2 parameters added

  These new parameters allow us send a data string to a server using GPRS when a digital input changes level or when a programmed trigger level on an analog input is reached.
- In the MTX-65i (obsolete), GPIO3 and GPIO4 are digital inputs if the DNS_gpios parameter value is “on”
- See the example in Annex 2.13

V5.4

- MTX_gatewayModBus parameter added

  This will allow you to configure modbus TCP / modbus RTU tunnel Gateway.

- See the example in Annex 2.14

V5.5

- ALARM_gpioEnabled parameter corrected so you can use it with the MTX-IND terminal modem

V5.6

- RF Wavenis de Coronis protocol supported

  MTXTunnel can be used as a communications hub in GPRS-RF (868MHz) scenarios.

  Read Annex 5 for more information. New commands: AT^MTXTUNNEL=SETWAVENIS and AT^MTXTUNNEL=GETWAVENIS.

V5.7

- GPRS_auto parameter added

  The GPRS_auto parameter at “on” value plus “operator.txt” file allows you to use any network operator without changing the “config.txt” configuration file. This is intended for use when the SIM card may be changed or if MTX-Tunnel is sent without knowing which end operator will be used.

V5.8

- DYNDNS_period parameter added

  MTXTunnel now upgrades the DynDNS servers periodically, so it is more robust in the event of the DynDNS servers being down. Please remember MTX-Tunnel also accesses DynDNS servers when the IP address changes.

V7.6

- Version 7 and more recent versions are compatible with previous ones. Now the configuration file is “config.txt”. The “MTXTunnel.jad” file is not used
- The AT^MTXTUNNEL=GETIOS command is standardized and answers with OK as a normal AT command
- MTX_IDClientExtended has a new extra parameter value
- In the socket client scenario, the “imei” parameter option only sends the MTX-Terminal modem’s IMEI identification. GPIO and ADC values will NOT be sent. It only allows the socket client to establish a connection once every X minutes/hours. Example: the modem will connect to the

server every hour. After data exchange, the socket can close on the server's side. MTX-Tunnel will not reconnect until after one hour

- Added commands related to Wavenis 868MHz devices are included in the MTXIND-V2 terminal modem, which acts as a communications hub. Now you can read and control a mesh network of sensors such as temperature, 0-10V or 4-20mA sensors or you can read pulses from a counter. Sensor readings can be taken in real time or periodically. A new feature is that this data can be sent to a WEB server using JSON objects
- As well as the ModbusTCP / ModBusRTU Gateways, MTX-Tunnel V7 can now also autonomously send the memory tables of a modbus device to a Web server using JSON objects
- In version 7 the new LOGGER_ioPeriod parameter can use JSON objects to send the readings of analog and digital inputs to a WebServer. Please remember you could also use DNS_ parameters
- The parameters MTX_init1, MTX_init2, MTX_init3 have been added to configure the MTX-Terminal modem in a way that some AT commands will run after the power is turned on
- MTX_radioBand is a new useful parameter that will allow you to select GSM European bands or American ones in order to make it faster to register to the network
- AT^MTXTUNNEL=TEMPORALCLIENT,<IP>,<Puerto> is a new AT proprietary command which allows you to temporarily run a TCP client and connect to a server. This is useful in the event of the network operator blocking incoming requests and when Telnet cannot be used. Now using SMS, MTX-Tunnel can create a socket connection to our server and remotely send AT commands in order to change any configuration parameter or even some network or modem statuses
- New AT^MTXTUNNEL=ATEmbedded command. AT commands can be sent remotely using a client socket connection, which can be temporary. AT commands must be between the special strings <MTXTUNNELR> and </MTXTUNNELR
- MTX_TPServer2 is a new parameter used to backup time synchronization in case the server is down. It needs to be synchronized periodicall
- MTX_pingIP is a new parameter that can define the PING IP in order to check communications. However you need to activate the MTX_PING parameter to "on"

V7.7

- The possibility to read 868MHz Wavelog devices is added in order to be able to remotely read digital input

V7.8

- You can now send encapsulated AT commands in GPRS-serial gateways, both in server and client mode
- The MTX_ATEmbeddedPass parameter has been added in order to set a password for encapsulated AT commands in GPRS-serial gateways
- It incorporates DNS_mode: remoteat in order to replicate a modem's digital inputs in the relays of another modem via GPRS

- It includes the MTX_flushSerialBuffers parameter in order to clean serial buffers before a TCP/IP connection
- TCP_ICP2 and TCP_port2 parameters are available to simultaneously establish 2 GPRS-serial gateways in client mode
- The MTX_clientReconnection configuration parameter is included in order to specify the client socket's reconnection time when having problems with the server

V7.9

- The ALARM_gpioMessage1 was added so that it is possible to send a different alarm SMS when the digital input value is "1" or "0"

V7.11

- The CSD_enabled parameter allows the modem to pick up GSM calls at the same time as being connected to the GPRS. This is ideal for metering applications
- The "Gateway" possibility has been added to the MTX_portAux parameter. This allows you to establish a serial gateway between the modem's two serial ports when there are no GPRS connections/GSM calls
- MODBUS_onlyChanges has been added. When the value is "on", it allows you to send data read from MODBUS equipment to a server only when one of the read recordshas changed

V7.12

- It allows you to read multiple Modbus devices, specifying a list of Modbus addresses in the parameter MODBUS_address

V7.15

- The parameter GPRS_autoTimeout is added
- The parameter SMS_urc is added, to be able to automatically send an SMS via the serial port
- The AT^MTXTUNNEL=TEMPORALCLIENT,xxx.xxx.xxx.xxx,port,time command is modified
- The parameter MTX_urcPort is added, to be able to specify the URCs' port
- The parameter MTX_clientTimeout is added (maximum time for socket client without data before the socket is restarted)
- The setting AT+CSNS=4 is added by default, so calls without data bearer are treated as such and not as voice calls (before it was necessary to write MTX_init1: AT+CSNS=4)
- MTX_TPServer allows the establishment of the value "null". Very useful in order to not depend on external time servers in time delayed socket applications
- New command AT^MTXTUNNEL=GETCONFIG to read all modem settings from telnet

V7.16

- The parameter MTX_rssiLevel is added, useful if using MTX-65i (obsolete). Specifying a value >0, in case of low coverage, MTX-65i (obsolete) LED will light up warning of the lack of proper coverage

V7.17

- Up to 5 IP addresses can now be entered in the parameter UDP_IP, which allows the serial data received to be sent to 5 remote servers
- The parameter SMS_header is added, which allows the text to be written in the SMS commands to be customized

V7.18

- It is possible to read modbus records of more than one device but also with memory maps of different records. It is also possible to read several memory maps of a single device

V7.19

- The parameter MTX_fullDuplex is added, which allows for a better multiplexing of input and output serial data in the GPRS-Serial gateways
- The parameter MODBUS_readCommand is added, in order to be able to choose between 0x03 and 0x04 read commands

V7.20

- The parameter MTX_filter is added, which allows the introduction of a filter for GPRS-Serial gateways. It enables the sending of only those frames with a certain header
- The supervision of communications with the Wavecard in the MTX-IND-V2 terminal is improved

V7.21

- The parameters OUTPUT_mode and OUTPUT_config are added. These configuration parameters allow an increase of control in digital outputs and relays of MTX modems. It is possible to decide when to switch these on, for example one output per hour, per GSM call, per SMS, in function of a modbus record of an external device, of the value of an analog input, etc.
- The parameter LOGGER_ioEvent is added. It allows the scoring of the records in the logger when there is a change in a digital input

V7.22

- The Logger is improved by synchronizing the time in getting modbus parameters. That is, if modbus data from a device is read every 5 minutes, then it is done at exact times (00:00,

00:05, 00:10, ...)

V7.23

- Possibilities are added to the parameters MTX_ATMux, MTX_urc, and MTX_DTR to allow for the example scenario in ANNEX 2.14

V7.24

- GPRS-Serial gateways are improved when the UDP packets are used. The gateways are stronger and it allows the socket reconnection time to be set

V7.25

- The set up parameters ALARM_powerEnabled, ALARM_powerMessageOn and ALARM_powerMessageOff are added. These parameters are accepted by the model MTX-65+G in its battery version (obsolete). It allows SMS alarm messages to be sent in the event of battery failures

V7.27

- The set up parameters LOGGER_serverLogin and LOGGER_serverPassword are added, to work with M2M data collecting platforms that have an authentication system
- The parameters MODBUS_logFrequency, MODBUS_logType and MODBUS_changeDiff are added, which allow the detailed configuration of x autonomous readings of modbus records of connected devices
- The parameters MTX_latitude and MTX_longitude are added by the MTX for the new astronomical clock, which allows the switching of relays at the exact time of sunrise and sunset
- In the OUTPUT_mode parameter, the “astronomical” mode can be included so that said output (digital or relay) is controlled by the MTX-Tunnel astronomical clock. It is possible to add an offset for both sunrise and sunset
- It is possible to use radio communications against wavenis protocol devices and communications to read modbus devices at the same time
- It is possible to make GPRS-Serial gateways using 2 bits of stop in the configuration of the modem serial port
- The parameter SMS_replaceText is added, which allows the replacement of a text by another in an SMS message. Useful for SMS – RS232 gateways
- The parameter MTX_configMode is added, which allows the configuration mode of MTX to be reversed (so the configuration mode is with a SIM inserted and the working mode without a SIM card)

V7.29

- The parameter CSD_commPort is added, which allows you to choose whether the serial port used in a GSM (CSD) data call reception is COM1 or COM2 (until now it was set as COM1)

V8.02

- Added support for the new 3G modem MTX-3G-Java. That is, MTX-Tunnel is now compatible with a 3G device

V8.02

- Added support for the terminals MTX-65+Gv6 (obsolete) and MTX-65+Gv7 (obsolete)

V8.03

- Internal improvements for remote firmware updates via OTAP

V8.04

- New option “socket” in parameter OUTPUT_mode. It allows a relay to be opened/closed if a socket is established
- New parameter MTX_interface for the new MTX terminal MTX-3G-Java. It allows a USB-GPRS gateway to be established

V8.06

- Added support for the new 3G modem MTX-3G-Java-B (obsolete)
- Solved the problem with the parameter MTX_urc. In some special cases it could have affected access to the modem’s configuration file when in config mode

V8.07

- Now is not necessary to use “” with the MTX_PIN parameter with the MTX-3G-Java device in the config.txt file
- New options (“1” and “2”) with the MODBUS_readCommand parameter. It allows binary registers beside words to be read
- New parameter MODBUS_custom. It allows a custom value in JSON data to be added

V8.08

- The parameter MODBUS_reqType is added, which allows you to choose whether the read Modbus records are “word” or “doubleword”

V8.09

- The parameters BLUETOOTH_enabled, BLUETOOTH_pin, BLUETOOTH_mode, BLUETOOTH_name, BLUETOOTH_initCommands are added for scenarios that require 3G+Bluetooth
- The command AT^MTXTUNNEL=SETIO,X,Y is added, which allows the status of a digital output to be changed with MTX 3G devices
- Support for the MTX-3G-Java-GPS device is added

V8.10

- FTP support is added to download files inside an MTX terminal. For this, a new command has been created: AT^MTXTUNNEL=FTP (see manual). For example, the entire configuration file config.txt, the file operators.txt in full, or any other file can be downloaded
- It is possible to create a configuration file "config.txt" for security reasons. It is sufficient to download a "config.txt" file that is renamed "numTelef.bkp" (for example "+34666123456. bkp). If an SMS containing the text "MTXTUNNEL RESTORE" is sent to the modem from the number that corresponds with that in the name of the .bkp file, MTX will retrieve the backup configuration
- The parameters DNS_httpMode, DNS_serverLogin, DNS_serverPassword are added which allow communications with secured platforms via JSON objects
- The DNS JSON chain incorporates new parameters such as the GSM coverage, the MTX-Tunnel firmware versión and the MTX model
- When a DNS JSON string is sent to a Web platform, AT commands can now be sent in response whenever the MTX_ATEmbedded parameter is set to "on" and the command is sent from a Web platform between the tags <MTXTUNNELR></MTXTUNNELR>
- New command AT^MTXTUNNEL=SETCONFIG which is created to be sent only as a HTTP response to the sending of a JSON object to a server. This command allows the entire configuration file config.txt to be changed with just one command
- Internal MTX webserver changes made to correct the problem caused when SIMs issued by Vodafone were used and login to the server was not available
- The option "logger serie" is added. This means the MTX terminal is capable of sending user-configurable RS232 strings and storing the replies in an external device. It is also able to log serial received in its serial port without previously sending data. Please consult the examples for further details. For this, the following parameters have been created: LOGGER_serialFrequency, LOGGER_serialPeriod, LOGGER_serialData1,... LOGGER_serialData10
- Support is added for MTX-3G-Java-ULP (obsolete) and MTX-3G-Java-ULP-GPS (obsolete)
- New command AT^MTXTUNNEL=SETOUTPUTTIMER is added which allows an output (digital or relay) to be changed for X seconds with just one AT command
- The BITCOIN_ parameters are added to process online payments

V8.11

- AT^MTXTUNNEL=RS232 command is modified. Now in a SMS-RS232 tunnel it is possible to send the characters CR (0x13) and LF(0x10) via SMS. For example, if you want to send an SMS to MTX-Tunnel for it to send the text “COMANDO1[CR][LF]” via serial port, now it is possible to send an SMS with the text “0x130x10” , and MTX-Tunnel will replace the text with \r\n (i.e. carriage return)

V8.12

- New configuration parameters related to movement detection alarms (for MTX containing accelerometer) are added. In particular, the following prameters were added: MOVEMENT_enabled, MOVEMENT_messageOn, MOVEMENT_messageOff ,MOVEMENT_threshold, MOVEMENT_windowTime
- New configuration parameters related to temperature alarms (for MTX that can be connected to MTX-TEMP-RS232 temperature sensor) are added. In particular, the following prameters were added: TEMPERATURE_enabled, TEMPERATURE_period, TEMPERATURE_max, TEMPERATURE_min,TEMPERATURE_messageOn, TEMPERATURE_messageOff, TEMPERATURE_threshold

V8.15

- GPIO control is added to the following models: MTX-65i-RS485 (obsolete) and MTX-3G-Java-RS485 (obsolete)

V9.00

- The configuration parameter TELNET_instances is added. It allows having up to 2 simultaneous Telnet sessions for MTX 3G modems
- MTX modem models with 3G connectivity can now have 2 simultaneous 3G-serial gateways in conjunction with TELNET

V9.02

- The configuration parameter MTX_encryptedConfig is added. It allows to encrypt “config.txt”file. Only available for 3G modems
- Low-coverage led activation for MTX-3G-Java modems (related with the configuration parameter MTX_rssiLevel)
- When the modem is in “config” mode(i.e. when MTX-Tunnel is used without a SIM card) five-minute watchdog timer becomes active. That means that after 5 minutes of working in “config” mode the modem will restart. This solves certain problems occurring when some users carelessly insert a SIM card into the modem which is on. This way the modem remains in “config” mode instead of “running” mode

V9.04

- The new command AT^MTXTUNNEL=GETPOWERSTATUS. This command allows to know whether a MTX modem (modems with internal battery)is using an external power supply or the battery is being used after a power cut
- MTX-Tunnel support for the model MTX-3G-Java-GPS-BAT (obsolete)

V9.05

- MTX-Tunnel suppport for the new modem models: MTX-IoT 3G and MTX-IoT 2G
- New configuration parameters: LOGGER_https and DNS_https for SSL communications
- New configuration parameter: GPRS_mode. Useful for 3G models, because it lets you choose between “auto”(default), “2g” and “3g”

V9.06

- New configuration parameters related to real-time modem control from WEB platforms: LINK_enabled, LINK_ip, LINK_port, LINK_retryPeriod, LINK_timeout, LINK_keyId, LINK_ssl
- PULSE COUNTER service(300HZ maximum) is added to MTX-IoT 3G and MTX-IoT 2G modems
- New configuration parameters: TCP_IPb, TCP_portb, TCP_IP2b, TCP_port2b designed for Metering applications. Service similar to GSM call preference but with 3G connections. That means that it is possible to enable a 3G-RS232 gateway for a real-time counter reading, but when an operator IP connection is established(ENDESA, IBERDROLA, ...) the real-time IP connection is paused to give way to it

V9.11

- The parameters hdop, vdop and satllite number in the data frames that include GPS location are added. Useful for stating the horizontal (hdop) and vertical (vdop) accuracy of a sent location

V9.12

- The parameter“TYPE” is included in all JSON with data transmitted to a WEB server. This simplifies data recognition in aplications transmitting different data types. In particular, the sent “TYPE” can be: “DNS”, “IOS”, “MODB”, “MOV”, “BLUE”, “SERIAL”, “TEMP”, “WMBUS”, “WAVT”, “WAVSC”, “WAVSV”, “WAVF”, “WAVL”, “POWER”.
- “CELLID” data is included in JSON of all DNS frames transmission. This allows to obtain an approximate GPS location of MTX-Tunnel via GSM localisation (independently of whether the MTX disposes of internal GPS or not)
- New command AT^MTXTUNNEL=GETCELLID added. It returns the ID of the telephone cell of the modem.Useful for cell localization systems
- New command AT^MTXTUNNEL=RESET, seconds . It allows to reset the moden on a deferred basis (i.e. X seconds after the command is executed)

- New command AT^MTXTUNNEL=SETPARAMS,... It allows to change different configuration parameters simultaneously, which is how the command AT^MTXTUNNEL=SETPARAM (allowing only 1 configuration parameter change) is improved. Useful for configuration change from WEB platforms
- New parameter DNS_httpMode is added. It permits DNS frame transmission in “get” format (HTTP GET, as it was until now), “getjson”(i.e. JSON transmission through HTTP GET) or “postjson” (JSON transmission through HTTP POST)
- In MTX modems with internal battery, the JSON of DNS frames includes the parameter POW. This parameter indicates if a MTX modem disposes of external power supply (1) or is using its internal battery (0)
- New configuration parameters DNS_header1, DNS_header2, DNS_header3, LOGGER_header1, LOGGER_header2, LOGGER_header3 are added. These parameters are very useful for MTX-Tunnel integration with third-party WEB platforms, such as THINGWORX

V9.16

- The new modem model MTX-3G-Java-2DB9 (obsolete) is included
- FTP and Telnet services improved
- SNMPv2c protocol included
- New commands AT^MTXTUNNEL=SETSCHEDULE and AT^MTXTUNNEL=GETSCHEDULE are added

V9.17

- New configuration parameter SMS_aliasResponse

V9.18

- Support for the new model MTX-IoT [3-S-N-GPS] (3G modem+GPS)
- Support for the new model MTX-IOT [3-S-N-WC25] (3G modem+communication at 868MHz)
- New configuration parameter GPS_period to store GPS locations in Logger
- New configuration parameter MTX_serverTimeout for server gateways

V9.20

- Support for the new model MTX-3G-Java-ACCEL (obsolete)
- New parameters; ALARM_movementEnabled, ALARM_movementMessage, ALARM_movementPause that allow sending movement alarm SMS (for MTX modems with accelerometer) via HTTP
- New parameters: GPS_mode, GPS_ip, GPS_port. When using GPS_mode you can periodically log GPS position and send it to a WEB server via HTTP or send the position in real time via TCP

socket to a specific IP and port

- Security improvement in Telnet service. New authentication methods provided by the new parameter TELNET_auth. Allows the use of OTP keys (One Time Password) encrypted via SHA-256. It allows to use OTP keys returned in SMS
- Security improvement of SNMP service due to SNMP_auth and SNMP_password parameters. New OID ACTION_OTP that permits to enable/disable reading/writing service of the rest of OIDs with the help of an OTP Password (with SHA-256)
- Support for the new module GPS Skytrack (together with GPS + GLONASS support)
- NewtimingcommandsAT^MTXTUNNEL=GETSCHEDULESandAT^MTXTUNNEL=DELSCHEDULES

V9.21

- New parameter SMS_defaultPrefix
- New configuration parameters CSD_allPhones, CSD_validPhone1 ... CSD_validPhone16. With these parameters it is possible to configure authorized phone numbers for GSM calls (CSD calls)

V9.22

- New MODBUSTCP_enabled, MODBUSTCP_port and MODBUSTCP_password parameters that allow the MTX-Tunnel to behave like a Modbus Slave device

V9.23

- New commands AT * MTXTUNNEL = GETASTRONOMIC that returns the hour of orto and sundown for a certain GPS position and time
- New commands AT ^ MTXTUNNEL = GETIO and AT ^ MTXTUNNEL = GETADC to be able to individualize the values of the digital and analog inputs

V9.24

- Improvements in the encryption of the configuration file with the parameter MTX_ encryptedConfig, increasing it to 128 bits

V9.25

- New MQTT shipping service. New parameters DNS_mqttTopic, LOGGER_mode, LOGGER_ mqttTopic, MQTT_enabled, MQTT_server, MQTT_id, MQTT_login, MQTT_password, MQTT_ attopic1, MQTT_attopic2, MQTT_attopic3, MQTT_atrtopic, MQTT_qos, MQTT_keepalive, MQTT_ persistent

V9.26

- Possibility of partial customization of the sending JSON of Logger and DNS frames for compatibility with JSON Carriots and other platforms. See sample shipping to Carriots platform for more information
- New Tacacs+authentication service for Telnet and SNMP. New TACACS_server, TACACS_port, and TACACS_key parameters

V9.27

- New functionality for the LOGGER_mode parameter. We can specify the “ftp” method. This way, the MTX-Tunnel is able to send read data (modbus, IOs, ...) through HTTP/HTTPS/MQTT/MQTTS and now also FTP. A file will be created for each read registry

V9.28

- New MTX modem models supported: MTX-IoT [3-S-N-N]-G (obsolete), MTX-IoT [3-S-N-GPS] and MTX-IoT [3-S-N-BLE] (obsolete)
- Possibility of bypass between serial ports
- 7.5 and 9.4 examples

V9.29

- New MTX modem models supported: MTX-T [4-N]

V9.30

- New parameters TELNET_loginGuest and TELNET_passwordGuest which allow a Telnet user without permission for reading or writing AT^MTXTUNNEL= commands, that is, without the ability to read or change the configuration
- New MTX_TPProtocol parameter that allows to use time servers with TP and NTP protocols
- MODBUS_logType parameter with type “bit” registry readings

V9.39

- New MQTT parameters MQTT_filetopic1 and MQTT_filertopic
- New model MTX-T [3-G] (obsolete)
- New command AT^MTXTUNNEL=SETMODBUS2
- [IMEI] tag can be used in the MQTT_ID field
- MTX-IOT-3G-JAVA-IOT internal relay managing (optional relay depending on model)
- New parameter MTX_resetCond, that avoids the time reset of a modem when there are connected sockets

- New option “modem” for the parameter MTX_ATMux, that allows to send AT commands to the main serial port white there’s no TCP connection or CSD call going on
- New parameter MTX_configMode
- Modification of the command AT^MTXTUNNEL=SETPARAMS

V10.01

- SMS_defaultPrefix, MTX_rssiLevel eliminated
- DNS_mode remoteat eliminated
- New parameter ULP_minutesOff, ULP_secondsOn, ALARM_ulpEnabled,ALARM_ulpMessage
- Parameter AT^MTXTUNNEL=GETIO,X changes. X allows values from 0 to 9
- Parameter AT^MTXTUNNEL=GETADC,X changes. X alows value from 0 to 1
- WAKEUP_ADCVALUEMIN and WAKEUP_ADCVALUEMIN adjusted between 0 and 50000
- It allows CSD calls via 2 serial ports if “CSD_commPort:3”
- MTX_blueLed can have value “mqtt” to remain fixed when connected to MQTT broker
- New support for model MTX-T2 [3-N]
- Support for user SSL certificates

V10.04

- Internal improvements of the registry in 4G network and MQTT connectivity
- The technology used (4G/3G/2G) TS (TimeStamp) is added to DNS framework
- New parameter MTX_status
- New model MTX-IOT [4-S-N-GPS]
- CSD call ability in 4G models (Rel2)
- Field MTX_model can use PN (part number) too
- New models MTX-T [2-N] and MTX-T2 [2-N]
- New model MTX-IOT [4-S-U-N] (obsolete)
- New option “nogps” for the configuration parameter MTX_redLed (LED iluminates when there’s no GPS coverage)
- New parameter LOGGER_addGPS. “On” allows to include GPS position, if it exists, when the stored registry was taken (for instance from a modbus reading of an external equipment)
- The default value LOGGER_registerSize increases to 500
- If the temperature sensor is enabled (TEMPERATURE_enabled: on) temperature is added to the GPS reading JSON (for instance, to track the cold chain in a fleet control)
- New parameter COMM_power. “On” allows to activate CDC signs and CSR of the main serial

port to power some devices via the serial port

V10.05

- New parameter TEMPERATURE_voiceCall to make a voice call in case of temperature alarm
- Support for new model MTX-IOT [4-S-B-N] with internal battery (obsolete)
- New parameter ALARM_powerVoiceCall to make a voice call in case of alarm due to lack of power supply

V10.06

- New value “modbuswavenis” for the parameter MTX_mode, which allows the MTX to be converted into a modbus-WAVENIS slave (that is, it allows a protocol conversion to be done, through modbus commands sent to the MTX, to read radio devices (pulse counters, temperature, analog sensors...)
- New parameter MODBUS_localAddress, necessary when MTX_mode is configured in “modbuswavenis” mode

V10.07

- Renaming of MTX models MTX-IoT [4-S-N-N]-STD-N-RL -> MTX-IOT [4-S-N-P] (obsolete) and MTX-IoT [4-S-N-N]-STD-N-ULP -> MTX-IOT [4-S-N-U] (obsolete). Old names are still supported
- New features for the astronomical relay with time exceptions (see example 8.7)
- New commands: AT^MTXTUNNEL=SETASTROSCHEDULE, AT^MTXTUNNEL=GETASTROSCHEDULE, AT^MTXTUNNEL=GETASTROSCHEDULES, AT^MTXTUNNEL=DELASTROSCHEDULES, AT^MTXTUNNEL=DELASTROSCHEDULE AT^MTXTUNNEL=DOWNLOAD

V10.09

- New parameters MQTT_commrxtopic and MQTT_commtxtopic, that together with MTX_mode: mqtt allow to make a transparent gateway Series - MQTT

V10.10

- Support for new model MTX-IOT [4-S-UR-N] model ULP (Ultra Low Power) + latch relay (obsolete)
- New parameter ULP_relayMode, which allows the model MTX-IOT [4-S-UR-N] (obsolete) to activate its internal latch relay when it wakes up (when leaving power consumption mode) and deactivate its relay when sleeping (when entering low consumption mode). Useful to feed an external 4-20mA sensor
- New parameter LOGGER_ioPeriodDelay. Allows you to enter a few seconds of delay before reading your I/O. Useful for when the ULP_relayMode parameter is used to feed a sensor and this requires a few seconds to stabilize its reading

V10.11

- Support for new model MTX-IOT [4-S-UR-GPS] model ULP (Ultra Low Power) + relay latch + GPS (obsolete)
- New parameter GPS_agpsUrl, which allows to assist the internal GPS of the MTX (AGPS) decreasing the time necessary to obtain the first GPS position

V10.12

- Improvements in the internal supervision of execution of AT commands

V10.14

- New files “mtxtunnel_start.txt”, “iologger_start.txt” and “iologger_end.txt” that allow to execute small scripts of AT commands. All in the/atscripts folder. Allowed commands for the scripts: EXECUTE (which executes an AT command) and PAUSE (which generates a pause in seconds)
- “mtxtunnel_start.txt” is executed, if present, when starting the MTX-Tunnel
- “iologger_start.txt” is executed, if present, before proceeding to an I/O reading
- “iologger_end.txt” is executed, if present, after proceeding to an I/O reading
- New command AT ^ MTXTUNNEL = IOEVENT
- New feature in the timed AT command files “schedule.txt” where “day” is allowed to take the value “-1” to represent any day and “hour” is “-1” to also represent any time

V10.15

- Support for the new model with supercap MTX-T [4-S] PN: 199801464 (obsolete)
- New feature for the command AT ^ MTXTUNNEL = RS232... It is now possible to include commands with hexadecimal values, interesting to send commands by SMS and the use of ALIAS. The hexadecimal values must be sent between the tags <hex> </ hex>

V10.16

- Support is added to be able to use the MTX_portAux “on” parameter (to be able to send AT commands through the secondary serial port) and also be able to use the SMS-RS232 gateway through said secondary serial port

V10.18

- New MTX_temporalClientTimeout parameter, to specify the duration in seconds, of a TCP client temporary client
- New possibilities for the MTX_ATEmbedded parameter
- New examples for metering applications (meter reading) introducing SSL/TLS communication

with Device Manager and Counter Reading Platform. Examples 6, 7 and 8 of the annex 6

V10.20

- Support for new model 4G Cat4

V11.00

- Complete change of I / O management with respect to the MTX-Tunnel v10 version
- Elimination of the configuration parameters OUTPUT_mode and OUTPUT_config, replacing them with the new parameters GPIO_mode and GPIO_config, which refer to both the inputs and the outputs
- I / O numbering starts at "0" to make it compatible with the command AT ^ MTXTUNNEL = SETIO, X, Y
- New parameters MQTT_defaultIOTopic and MQTT_defaultIOQos, where the topic and qos are defined for sending events by MQTT due to GPIOs state changes
- New parameter MTX_saveOutputState, where the state of the outputs is allowed to be stored each time they change, to recover the state after a reset
- New possibilities for the MTX_TPFormat parameter, being able to specify the time in "mtxtunnel" format (the one available in MTX-Tunnel v10 and earlier) or "unix" or "epoch"
- Elimination of ALARM_gpioEnabled, ALARM_gpioValue, ALARM_gpioMessage, ALARM_gpioMessage1, ALARM_gpioPause, ALARM_gpioAT, DNS_adc1, DNS_adc2, WAKEUP_gpioEnabled, WAKEUP_gpioValue, WAKEUP_adcEnabled, WAKEUP_adcValueMin, WAKEUP_adcValueMax parameters
- Changes in Modbus TCP addresses (when MTX-Tunnel is configured as modbus slave)
- AT ^ MTXTUNNEL = GETIOS command response now returns a JSON
- New parameter "MODBUS_format"
- New JSON_ configuration parameters, which allow to wrapper the JSON sent by the datalogger
- New COMM_enabled and COMM2_enabled configuration parameters. This allows disabling the use of a certain serial port by the MTX-Tunnel, so it could be used by third-party applications to use the modem as a conventional modem (for the disabled port)
- New parameter SMS_enabled to allow disabling the use of SMS by the MTX-Tunnel, so a third-party application could manage the SMS externally from a serial or USB port of the modem
- New value "abort" for the MTX_mode parameter, which allows aborting the MTX-Tunnel firmware after the modem starts
- New parameter GPS_absolute to allow sending signed GPS positions (latitude and longitude)
- New parameter MTX_snifferMode
- Performance improvements on client TCP gateways

V11.07

- Support for ULP scenarios (Annex III examples) for MTX-IOT-S models
- New configuration parameter MTX_yellowLed for MTX-IOT-S models
- DNS frames of MTX-IOT-S models (which have internal battery) send external power status and battery level

V11.08

- New options for the MTX_gatewayModbus parameter. When establishing a Modbus TCP to Modbus RTU gateway it is now possible to use both serial ports (as before) or to choose which serial port to use

V11.09

- New configuration parameters GPRS_apn2, GPRS_login2, GPRS_password2, GPRS_DNS2, DUALSIM_select, DUALSIM_mode, DUALSIM_timeout, MTX_PIN2 for MTX modems with dual SIM capability
- New Example 6.19 Dual SIM demo
- New commands AT ^ MTXTUNNEL = SETDAC and AT ^ MTXTUNNEL = GETDAC to use the DAC port (analog / digical converter) of MTX models with support for this interface.
- Example 8.22 for DAC support

V11.10

-
- Default value of the DUALSIM_mode parameter changes to “ip”
- Default value of the DUALSIM_timeout parameter changes to “120”
- The DNS chain for sending periodic information includes the FW version of the I / O management micro
- Improvements to the AT ^ MTXTUNNEL = DOWNLOAD command, which allows you to enter a download timeout
- New AT ^ MTXTUNNEL = ADOWNLOAD command that allows asynchronous file downloads on the modem
- New command AT ^ MTXTUNNEL = ISFILE that allows to find out if a file is downloaded to the modem
- New command AT ^ MTXTUNNEL = SETIOMAINTENANCE that allows putting a digital output / relay in maintenance mode, temporarily releasing it from its configuration and going to manual mode
- New parameter MTX_api232Resp that allows specifying the format of the command AT ^ MTXTUNNEL = RS232. For example, it is very useful to be able to send a binary data frame

from an SMS to the MTX modem so that it redirects it through its serial port

V1.11

- The DNS chain for sending periodic information includes the input supply voltage of the MTX modem

V11.12

- ULP mode enhancements (for MTX modems with Ultra Low Power support, such as MTX-IOT-S family modems)

V11.15

- New WMBus hub functionality
- New configuration parameters WMBUS_interval, WMBus_mode, WMBUS_filter, WMBUS_data
- New command AT ^ MTXTUNNEL = SETWMBUSFILTERS, value

V11.18

- Support is included for new models 199801492 and 199801476
- DNS fix for older models of Gemalto EHS6 and EHS8 (Rel3) modules. For those models it is no longer necessary to include the GPRS_DNS parameter in MTX-Tunnel v11.xx

V11.19

- CSD call compatibility with standalone IEC102
- Inclusion of more instantaneous parameters (energy and power)
- Inclusion of integrated totals
- New commands to read (in JSON) the instantaneous values in real time and obtain the integrated ones between 2 specific dates
- New command “AT ^ MTXTUNNEL = SETIEC102, ...”
- New command “AT ^ MTXTUNNEL = GETIEC102, ...”
- New command “AT ^ MTXTUNNEL = SETIEC102_CTAVM2, ...”
- New command “AT ^ MTXTUNNEL = GETIEC102_CTAVM2, ...”
- Schedule to read each day of the integrated totals from 23:59 (UTC) of the current day to one month ago

Sales Contact

SPAIN

C/ Alejandro Sánchez 109
28019 Madrid

Phone 1: 902.19.81.46
Phone 2: +34-91.560.27.37
Email: contact@webdyn.com

FRANCE

26 Rue des Gaudines
78100 Saint-Germain-en-Laye

Phone: +33.139042940
Email: contact@webdyn.com

INDIA

803-804 8th floor, Vishwadeep Building
District Centre, Janakpurt, 110058 New Delhi

Phone: +91.1141519011
Email: purchase-india@webdyn.com

PORTUGAL

LusoMatrix Lda.
Av. Coronel Eduardo Galhardo 7-1ºC
1170-105 Lisbon, Portugal

Phone: +351.218162625
Email: comercial@lusomatrix.pt

APAC

9F, No. 156, Sec. 3, Minsheng E. Rd.
Songshan Dist., Taipei City 10596, Taiwan

Phone: +886.965333367
Email: ahsu@matrix.es

AUE

Dubai

Phone: +34.915602737
Email: hperchin@matrix.es

USA

Chicago

Phone: +34.915602737
Email: jcabezas@matrix.es

V11.19 subject to changes | Webdyn © by Flexitron Group