HexaPad-10 (M7E) & HexaPad-10 BC (M7E)

User Guide

Public

Copyright

© 2025 Acura/HID Global Corporation/ASSA ABLOY AB. All rights reserved.

This document may not be reproduced, disseminated, or republished in any form without the prior written permission of HID Global Corporation.

Trademarks

Acura is a trademark or registered trademark of HID Global, ASSA ABLOY AB, or its affiliate(s) in the US and other countries and may not be used without permission. All other trademarks, service marks, and product or service names are trademarks or registered trademarks of their respective owners.

Contacts

For technical support, please contact: **acura.support@hidglobal.com**.

What's new

Date	Description	Revision
December 2025	Updates for latest firmware version.	A.2

A complete list of revisions is available in **Revision history**.

Products covered by this document

Model	Code
HexaPad-10 (M7E)	Please consult.
HexaPad-10 BC (M7E)	Please consult.

Introduction 6
1.1 Introduction 7
1.2 Main features 7
1.3 Interfaces 9
1.4 Approval notes 10
1.4.1 Anatel (Brazil) 10
1.4.2 FCC (USA) 10
1.4.3 Authorized antenna 10
1.4.4 CE/RED (Europe) 11
General information 12
2.1 HexaPad-10 USB 13
2.1.1 Dimensions 13
2.2 HexaPad-10 BC 14
2.2.1 Dimensions 14
2.3 Specification 15
2.4 Barcode reading zones 17
2.5 Barcode location to improve reader performance 18
2.6 Built-in Antenna 19
2.6.1 VSWR 19
2.6.2 Far field gain 19
Operation 20
3.1 Turning on the reader 21
3.2 Driver installation 22
3.2.1 Windows 10 and 11 - Identifying a serial port 23
3.2.2 Linux - Identifying a serial port 24
3.2.3 macOS - Identifying a serial port 24
3.3 Reader operating modes 25
3.3.1 Autonomous mode 25
3.3.2 Turning tag reading ON/OFF/TIME - Autonomous mode 26
3.3.3 General diagram 27
3.3.4 Reading tags engine 28
3.3.5 Reading result format 28
3.3.6 Configuring the reader - Autonomous mode 29
3.3.7 How to set up the reader 29
3.3.8 Operating logic 30
3.3.9 Syntax of messages and responses 30
3.3.10 Windows - how to send commands to the reader 31
3.3.11 Linux and macOS - how to send commands to the reader 32

3.4 Configuration commands 33
3.4.1 help command 33
3.4.2 summary command 34
3.4.3 dumpconfig command 35
3.4.4 Summary of configuration commands and default values 36
3.4.5 Response messages to commands 38
3.5 Configuration command descriptions 39
3.5.1 readmode 39
3.5.2 rfon 40
3.5.3 rfoff 41
3.5.4 linkread 42
3.5.5 gen2blf 43
3.5.6 gen2tari 44
3.5.7 gen2tagencoding 45
3.5.8 gen2session 46
3.5.9 gen2target 47
3.5.10 gen2q 48
3.5.11 gen2accesspassword 49
3.5.12 selectfilter 50
3.5.13 initreadtag 51
3.5.14 readtag 52
3.5.15 tagtimeout 53
3.5.16 rssifilter 54
3.5.17 tagop 55
3.5.18 reportonlytagopok 56
3.5.19 epcdecode 57
3.5.20 separator 58
3.5.21 endofline 59
3.5.22 keystrokespeed 60
3.5.23 reportrssi 61
3.5.24 reportantenna 62
3.5.25 reportreadcount 63
3.5.26 summary 63
3.5.27 temperature 63
3.5.28 version 63
3.5.29 rfidversion 63
3.5.30 rfidmodule 64
3.5.31 rfidregion 64
3.5.32 serialnumber 64

3.5.33 dumpconfig 64
3.5.34 barcodemode 65
3.5.35 initreadbarcode 66
3.5.36 readbarcode 67
3.5.37 barcodecommand 68
3.5.38 barcodedecode 70
3.5.39 configbybarcode 71
3.5.40 configbarcodeprefix 72
3.5.41 enablebeep 73
3.5.42 enablecapslock 74
3.5.43 echochar 75
3.5.44 facreset 76
3.5.45 upgradefirmware 77
3.5.46 help 78
3.6 Write tag commands 79
3.6.1 writetag 79
3.6.2 writedata 80
3.7 Switching between Autonomous and Transparent modes 82
3.7.1 Use the readmode command 82
3.7.2 Use the Caps Lock key (Windows and Linux only) 82
3.8 Read rate of the same tag 83
3.9 Hexpad reading and non-reading area 85
3.9.1 How to measure the RSSI cutoff value 85
3.10 Use cases - Autonomous mode 86
3.10.1 Read tags for 30 seconds and then remain idle 86
3.10.2 Checkout – GS1/EAN standard 86
3.10.3 Read tag encoded in SGTIN-96/198 and show result in GS1 element string format 87
3.10.4 Read tag encoded in SGTIN-96/198 and show result in EPC Tag URI format 88
3.10.5 Read TID memory bank and EPC id 89
3.11 Configure the reader using a QR code 90
3.12 Reader - Transparent mode 91
3.12.1 readmode 91
3.12.2 Software programming and development 91
3.12.3 Demo software – Universal Reader Assistant (URA) 92
3.12.4 Demo software instructions 92
3.13 Controlling the LED and Beep 95
3.14 Barcode scanning in Transparent mode 96

Section 01

Introduction

1.1 Introduction

The HexaPad-10 USB is a desktop reader designed with Ultra High Frequency (UHF) RFID technology. A built-in Near Field antenna is optimized for checkout operations. The HexaPad-10 BC (BarCode) is a versatile solution that seamlessly integrates UHF RFID technology with a built-in Barcode 1D/2D reader. This dual function allows both RFID and barcode capabilities, ensuring efficiency and adaptability in a single device.

1.2 Main features

The integrated antenna is of the Near Field type, restricting tag readings in a region above the reader, minimizing unwanted readings of nearby tags on the sides of the reader.

The reader has two modes, Autonomous, and Transparent.

Autonomous Mode

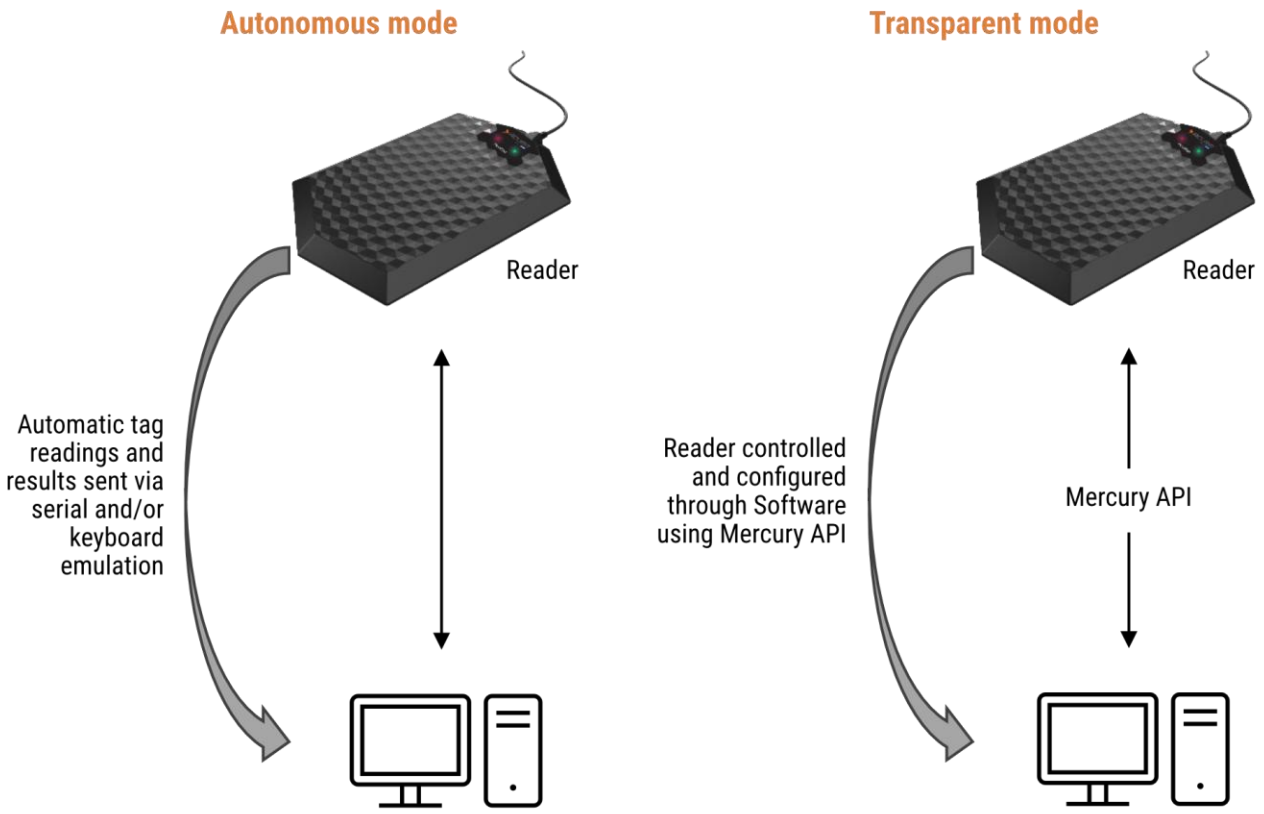
- Automatically reads tags in a continuous loop.
- Turn off/on the RF via command.
- Sends the results of the tags reads in the following interfaces:
  - Via keyboard emulation (HID class).
  - Via serial communication (CDC class).
  - Via keyboard emulation + serial communication.
- Various configuration parameters via serial interface.
  - Reading power.
  - Write power.
  - RF period times.
  - Gen2 parameters, (tag >> reader) and (reader << tag).
  - Reader information.
  - Read operations.
  - Select filter.
  - Tag read timeout configurable.
  - Read filter by RSSI value of the tag.
  - Barcode configuration.
- Can decode tags with GS1 SGTIN-96 and SGTIN-198 standards for checkout applications, sends in the following formats:
  - gs1string, for example: "(01)03608449920322(21)1540341".
  - gs1epcuri, for example: "urn:epc:tag:sgtin-96:1.360844.0992032.1540341".
  - gs1epcpureuri, for example: "urn:epc:id:sgtin:360844.0992032.1540341".
  - gs1gtin13, for example: "3608449920322".
  - gs1gtin13slz, the same as gs1gtin13, but suppressing leading zeros.
  - gs1sgtin13, for example: "3608449920322 1540341".
  - gs1sgtin13slz, same as gs1sgtin13 but suppressing leading zeros.
  - gs1aistrjson, for example: " {"01":"00004589876544","21":"3333444"}".
  - gs1bareid, for example: " gtin=00004589876544;serial=3333444".
- Can decode tags with ASCII standard.
- Can convert from hexadecimal to decimal with configurable range.
- Can split the tag EPC identifier with a configurable range.

Public

- Option to not decode and send the tag EPC hexadecimal raw value.
- Can read any memory bank of the tag as additional information to the tag EPC identifier.
- The built-in barcode reader has great performance to read most types of common 1d /2d bar codes, whatever read from paper or from phone screen.
- Can perform write operations with embedded select filter.
  - writetag, to write the EPC identifier.
  - writedata, to write the USER or RESERVED memory bank.

Transparent Mode

- The reader stays idle, does not perform autonomous tag readings.
- The reader must be controlled by a software that uses the mercuryapi, a complete API/SDK for reading and writing tags supporting the following languages:
  - C, .Net (C#) and Java.
- Compatible with Windows 10, macOS and Linux (no driver required). For Windows 7 and 8, the driver is provided for installation.
- Wide variation of reading power: 0 to 27dBm.



Public

1.3 Interfaces

HexaPad-10 readers use a USB interface. The reader does not require an external power supply, the power transmitted through the USB cable from a host Computer is sufficient for operation.

HID (keyboard), and Serial modes are enabled by default. When you connect the reader USB cable to a host PC, HID will be detected under the Human Interface Devices section and the CDC port will be detected under the Ports section of the Device Manager (Windows OS). The readers can be configured to HID mode, Serial mode, or both.

1.4 Approval notes

1.4.1 Anatel (Brazil)

HexaPad-10 USB and HexaPad-10 BC were tested and approved under the Regulation for Certification and Homologation of telecommunications Products, approved by Anatel Resolution No. 242 of November 30, 2000.

- **Types:** Radio Frequency Identification Systems - Category II.
- **Service/Application:** Restricted Radiation Radiocommunication.

1.4.2 FCC (USA)

Federal Communication Commission Interference Statement (FCC):

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operating in a commercial environment. This equipment generates, uses, and can radiate radiofrequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Note: The marking information is located at the bottom of the apparatus.

FCC Caution:	Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

MPE Warning:	Radiation Exposure Statement - This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 21cm between the radiator & your body. Contains FCCID: QV5MERCURY7EP

1.4.3 Authorized antenna

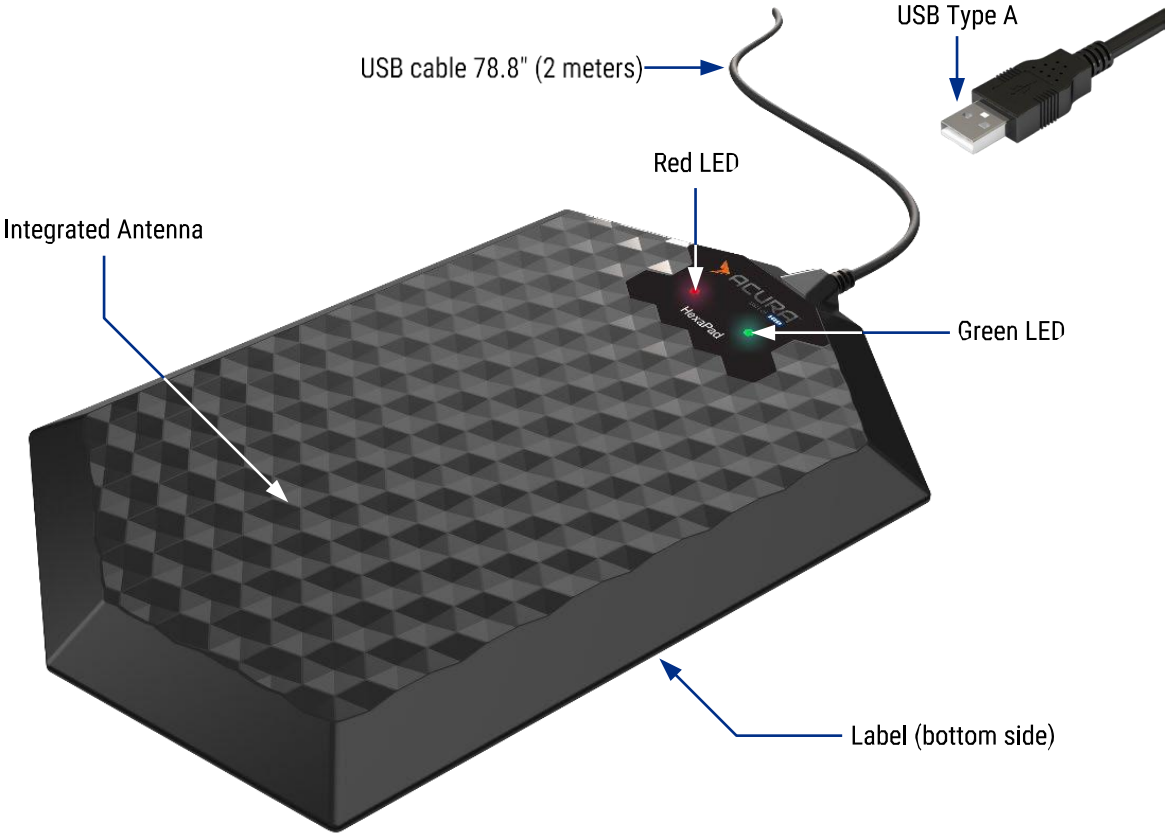
Vendor	Acura
Model	Integrated Near Field Antenna
Frequency Range (MHz)	915-928 MHz
Antenna Gain (dBi)	-4 dBi @920 MHz
Polarization	Circular (RHCP)
Type	Patch Near Field Antenna

Public

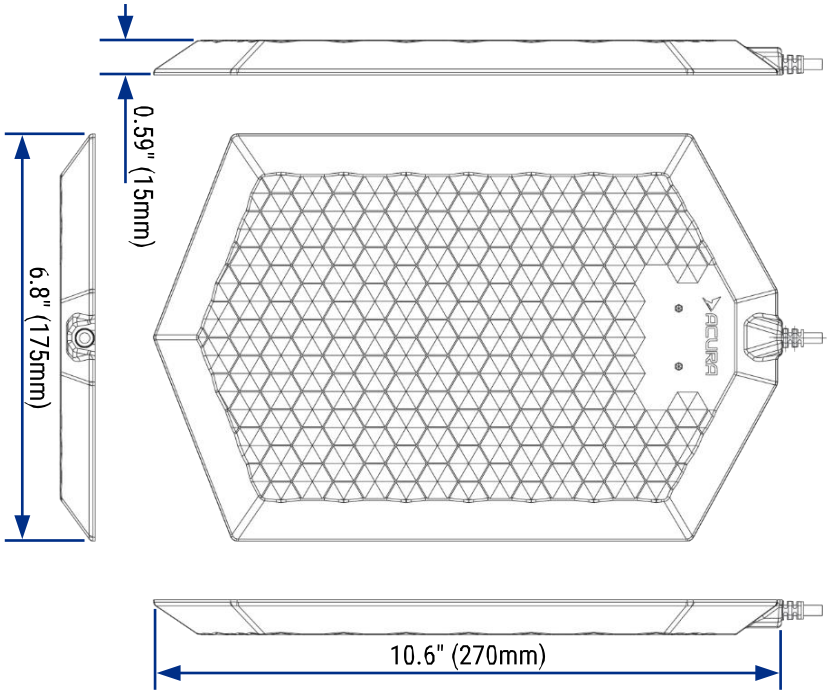
Section 02

General information

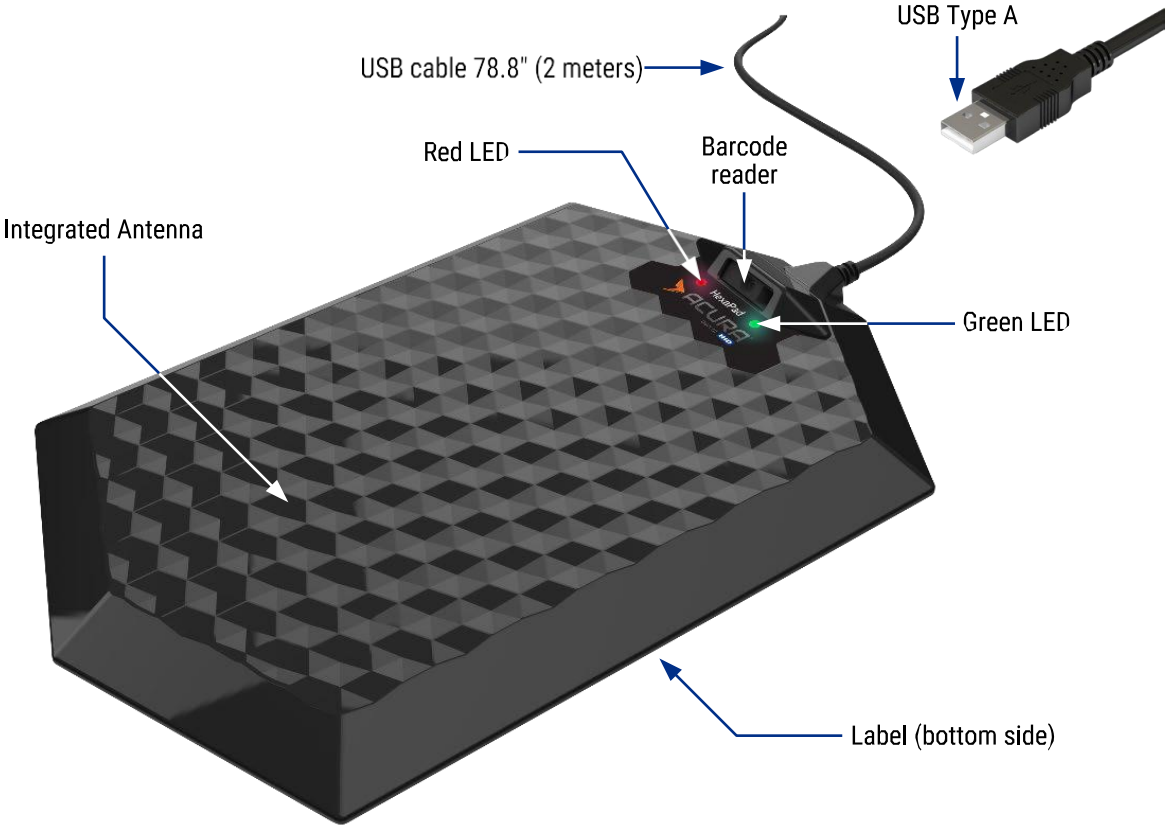
2.1 HexaPad-10 USB



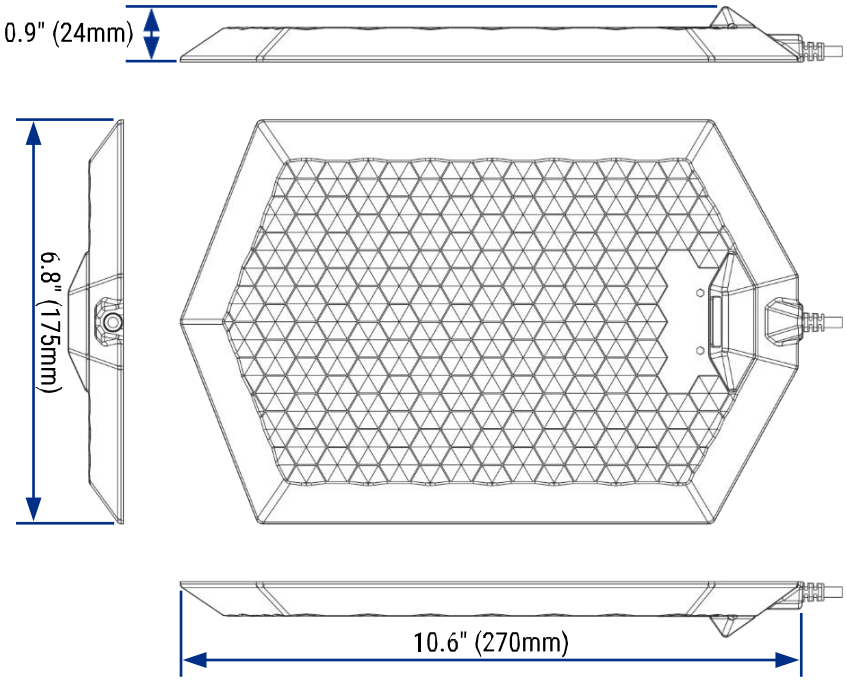
2.1.1 Dimensions



2.2 HexaPad-10 BC



2.2.1 Dimensions



2.3 Specification

Transponder Protocol	
Protocol	ISO 18000-6C (EPC Gen2) ISO 18000-63 (EPC Gen2 V2) (Transparent Mode)
RF interface	
RF Output power	From 0 to 27 dBm with 0.5 dBm increment
Regulation	ANATEL (BR) 902 – 907MHz and 915 – 928MHz FCC (USA) 902 – 928MHz CE/RED (EUROPE) upper band (916.3MHz, 917.5MHz, 918.7MHz)
Mode	Frequency Hopping
Modulation/RF Coding	PR-ASK/M4
Backscatter Link Frequency (BLF)	250KHz
Performance	
Reading distance (the reading distance can vary depending on the tag model, usage mode, and environment)	19.5" (500mm)
Data/Control Interface	
Connectors	Communication and Power: USB Type-A male connector
Communication interface	USB HID keyboard: The reader is recognized by the Operating System (Windows, macOS, and Linux) as a HID device. USB Serial CDC: When the reader is connected via USB, a serial port is automatically created on Windows 10, macOS, and Linux operating systems. (For Windows 7 and 8, driver installation is required).
Programming	Autonomous mode: The reader can be configured via serial communication using ASCII commands through a terminal application (such as Putty, Minicom, or Screen) without the need to use any API or SDK. It supports tag read-only functionality. Transparent mode: The reader responds to software using the Mercury API and has tag read and write functionality. (Note: barcode reading works in this mode only as keyboard emulation). Android not supported.
Energy	
Power	5 VDC +/- 1% (Host USB port). Maximum ripple from source: 200 m Vpp @ 20 MHz
Consumption	Max 3,8 W@ 5 VCC | Read power 27 dBm

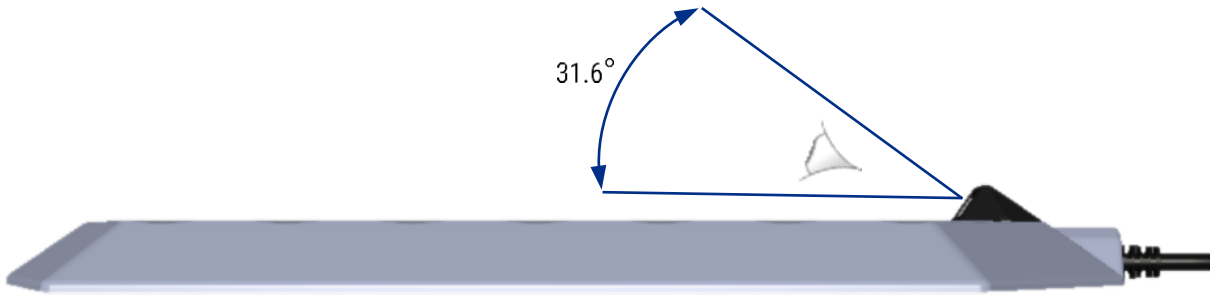
Public

Physical characteristics		
IP rating	Internal use	
Built-in antenna	Near field circular polarized antenna with -4dBi of gain (RHCP)	
Dimension (H x W x D)	HexaPad-10 USB: 6.7" x 6.8" x 0.6" (170 X 175 X 15 mm) HexaPad-10 BC: 6.7" x 6.8" x 0.94" (170 X 175 X 24 mm)	
Weight	17 oz 1 lb (484 g) (both)	
Cable length	78.8" (2 meters)	
Operation temperature	14°F to 131°F (-10°C to 55°C)	
Storage temperature	14°F to 140°F (-10°C to 60°C)	
Humidity	95%	
Mounting type	Desktop reader	
Built-in antenna		
Frequency range	902 - 928MHz	
Gain	-4 dBi @920 MHz	
Polarization	Circular (RHCP)	
Type	Patch Near Field Antenna	
Built-in barcode reader		
Image sensor	640 * 480 CMOS	
Symbologies	2D	PDF417 , QR Code, Data Matrix, Micro QR
	1D	EAN-13, EAN-8, UPC-A, UPC-E, Code128, Code 39, Codabar, UCC/EAN128, RSS, ITF- 14, ITF6, Standard 2 of 5, Matrix 2 of 5, Industrial 2 of 5, Plessey, MSI Plessey, Code 11, Code 93.
Resolution	≥3mil (differ from different type bar codes)	
Typical depth of field	EAN-13	2.3"-13.77" (60-350mm) (13mil)
	Code 39	1.57"-5.9" (40-150mm) (5mil)
	PDF417	1.96"-4.92" (50-125mm) (6.7mil)
	Data Matrix	1.77"-4.72" (45-120mm) (10mil)
	QR Code	1.18"-6.69" (30-170mm) (15mil)
Scan angle	Roll: 360°, Pitch: ±60°, Skew:±60°	
Field of view	Horizontal 42°, Vertical 31.6°	
Min. Symbol Contrast	25%	

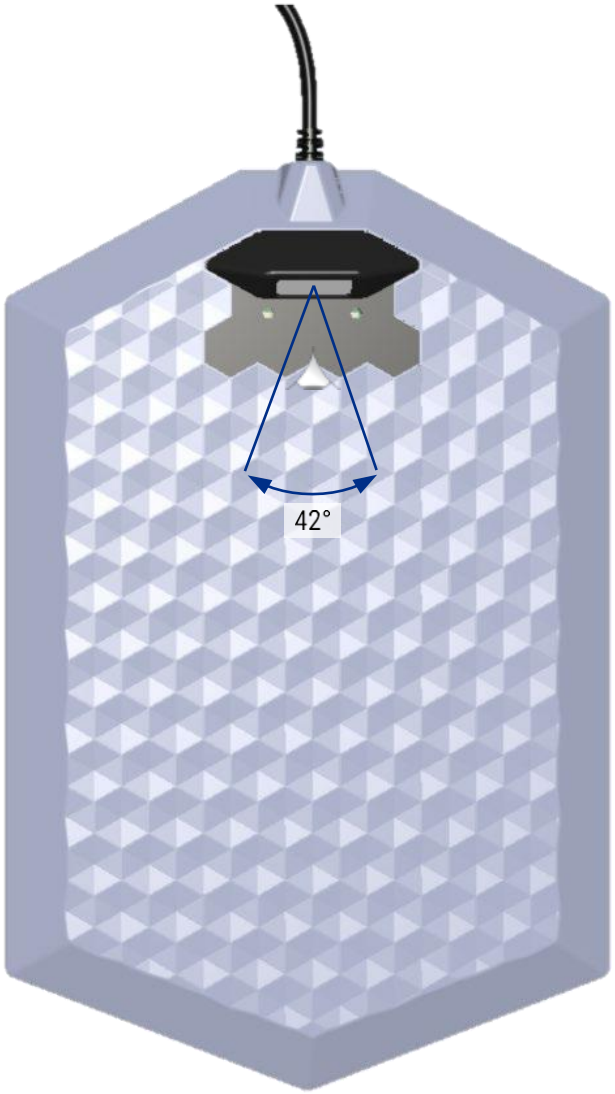
Public

2.4 Barcode reading zones

Vertical field of view



Horizontal field of view



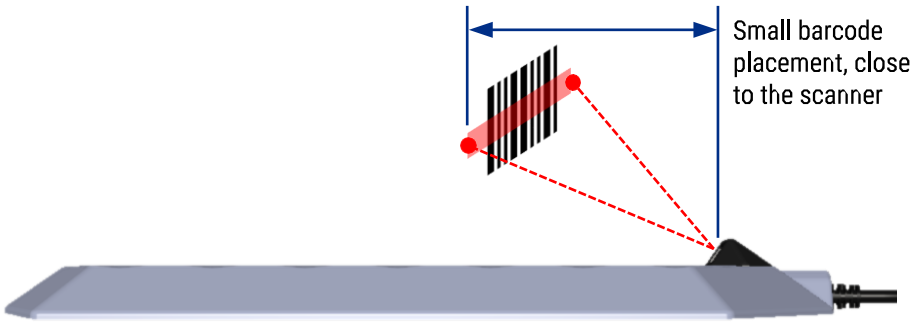
2.5 Barcode location to improve reader performance

When reading a barcode, it is important to use the appropriate distance and angle, in relation to the field of view between the scanner and the size of the barcode you are reading.

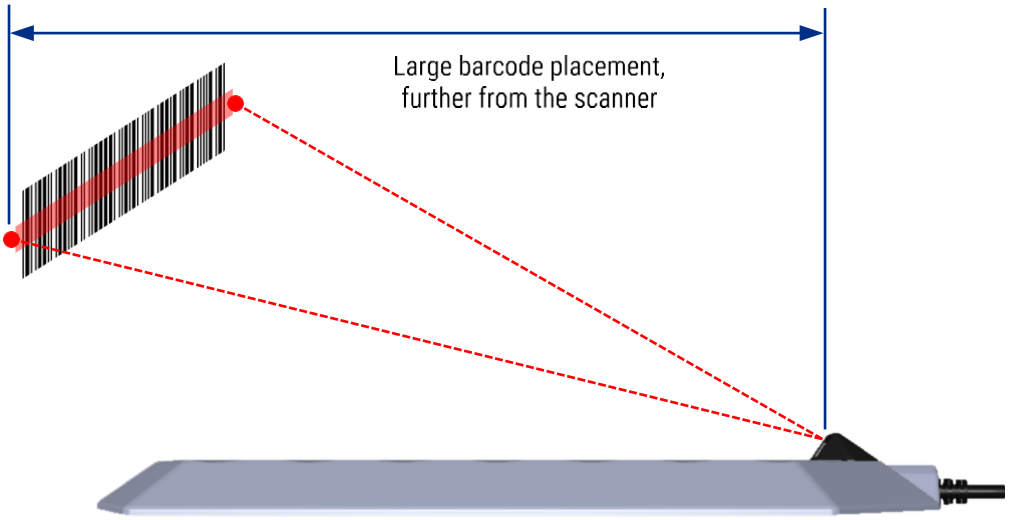
A small barcode will need to be close to the scanner, and a larger barcode will need to be further away from it.

For example:

Small barcode



Large barcode

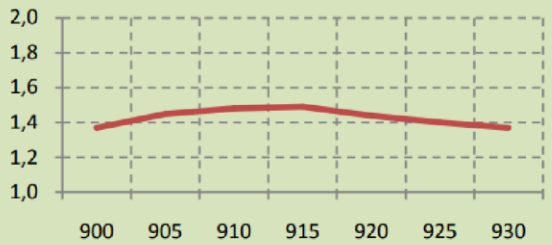


2.6 Built-in Antenna

The built-in antenna is a UHF Near Field Antenna, designed to contain the spread of UHF energy and limit most of the energy above the reader.

2.6.1 VSWR

MHz	VSWR
900	1,37
905	1,45
910	1,48
915	1,49
920	1,44
925	1,4
930	1,37



2.6.2 Far field gain

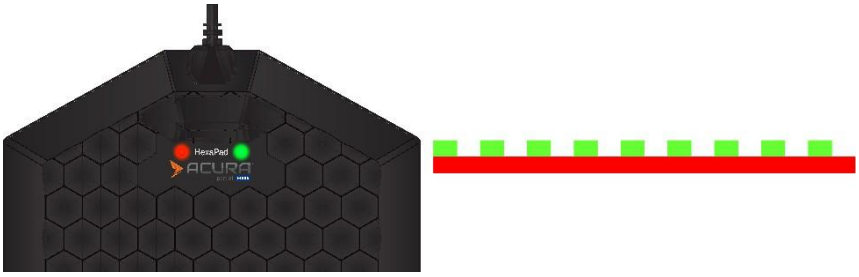
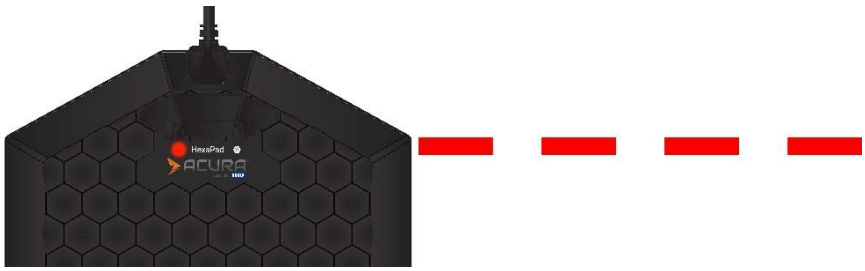
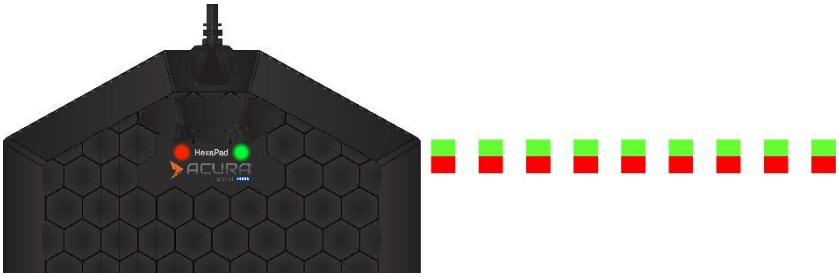
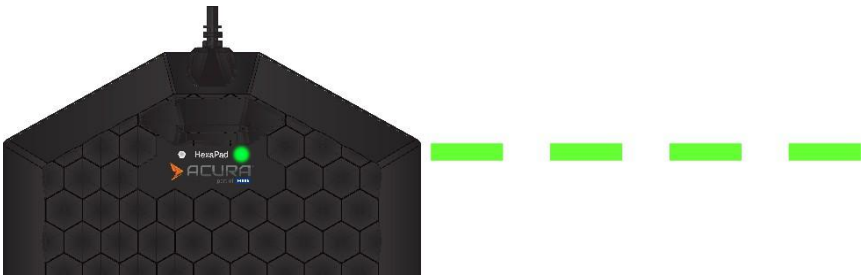
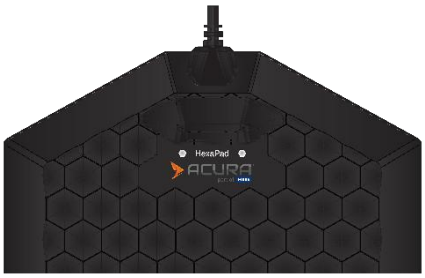
MHz	dBd	dBi
900	-11,9	-9,69
905	-9,7	-7,49
910	-7,7	-5,49
915	-6,5	-4,29
920	-6,3	-4,09
925	-6,4	-4,19
930	-6,5	-4,29

Section 03

Operation

3.1 Turning on the reader

After connecting the reader to a host Computer USB port, the LED status will exhibit the following combinations:

	Green LED: flashes fast. Red LED: solid on. Reader: in Autonomous Mode. Reading tags: activated (RF on).
	Green LED: turned off. Red LED: flashing at a consistent interval. Reader: in Autonomous Mode. Reading tags: deactivated (RF idle).
	Green and Red LED: flashing at same time in a consistent interval. Reader is powered on but with no USB communication with the Computer host.
	Green LED: flashing at a consistent interval. Red LED: turned off. Reader: in Transparent Mode. Note: In Transparent Mode, the red LED and beep are controlled by the user's software.
 LEDs turned off.	The reader is not powered on or is not functioning.

3.2 Driver installation

> **Note:** Any computer running an OS prior to Windows 10, will require a driver to be installed for the creation of the **COMxx** serial communication port. Please contact our support channel **acura.support@hidglobal.com** for more information.

For Linux, Android, and macOS, the driver is not required. When the reader is turned on, the following devices will be created in the Operating System:

• HID class type keyboard.

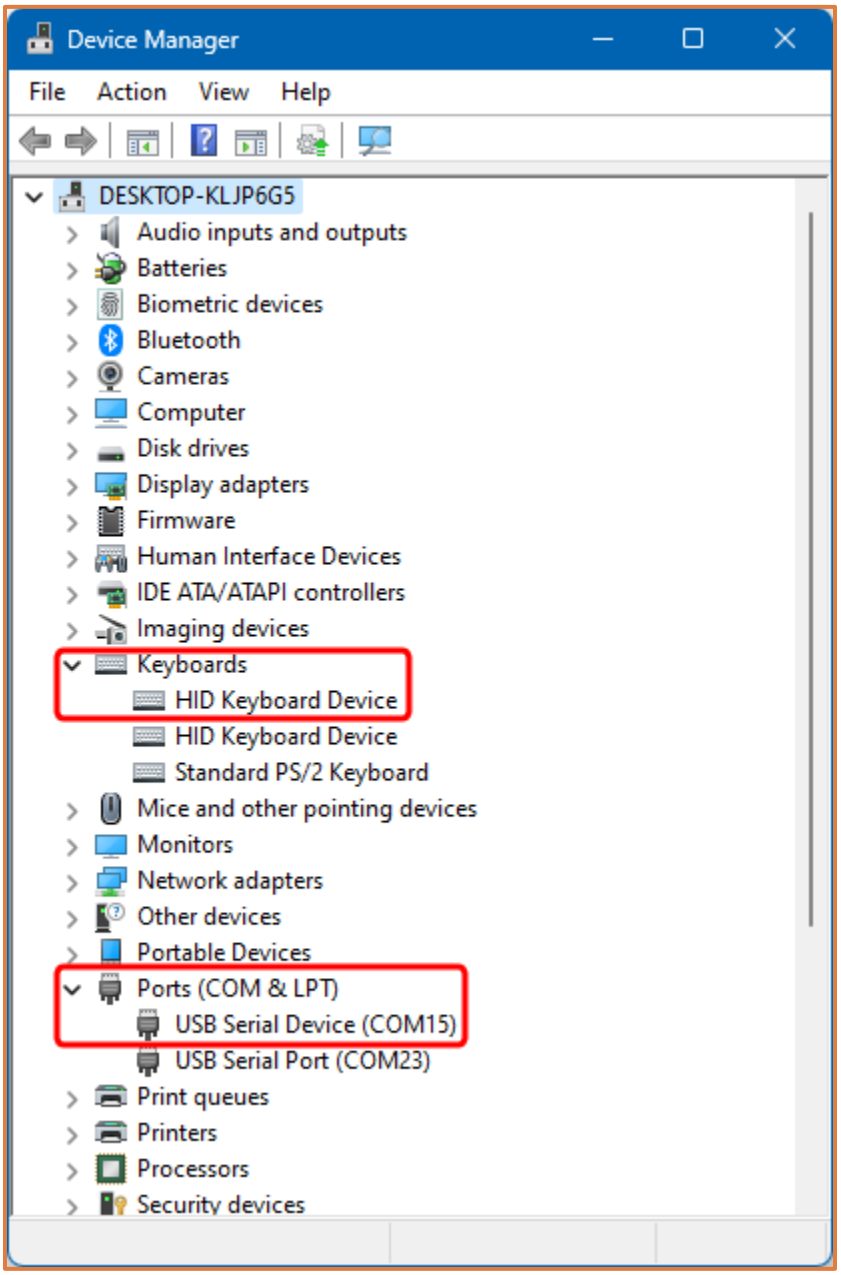
• Serial port, (COM&LPT on Windows), (CDC on Linux, Android, and macOS).

Since the reader uses the HID class keyboard type for keyboard emulation when sending reading results in Autonomous Mode, the Serial Port will be used for communication and configuration in both Autonomous and Transparent modes.

3.2.1 Windows 10 and 11 - Identifying a serial port

After turning on the reader, open the **Device Manager**. In **Ports (COM and LPT)**, a virtual serial communication port should appear a few seconds after the reader is connected. A new **HID Keyboard Device** will also appear in **Keyboards**.

Once the reader has been recognized by the Operating System, it is ready to start work.

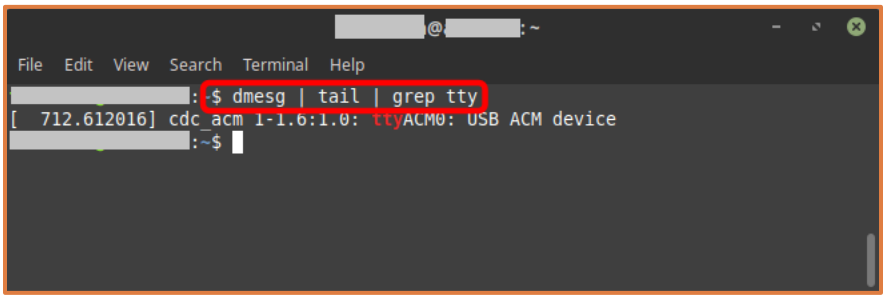


-

3.2.2 Linux - Identifying a serial port

To identify the name of the serial port created after connecting the reader to a USB port, type the following in the Terminal:

```
$dmesg | tail | grep tty
```



In the example, serial communication with the reader will be through **/dev/ttyACM0**.

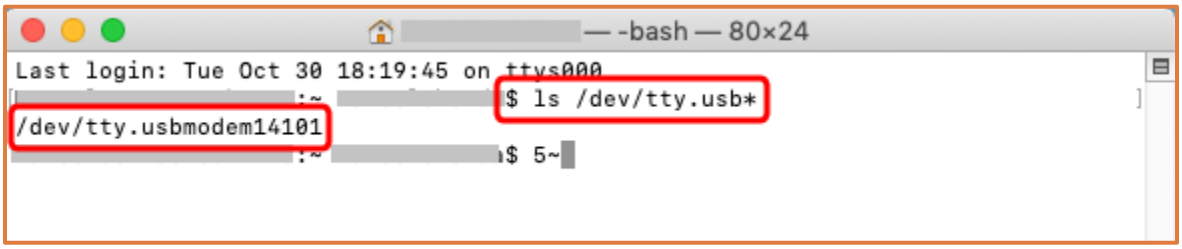
For some Linux distribution like Ubuntu, Mint, Fedora, OpenSuse. it is recommended to disable the ModemManager service to not try to connect to the serial port every time the reader is connect to the USB port:

```
$sudo systemctl stop ModemManager.service
$sudo systemctl disable ModemManager.service
```

3.2.3 macOS - Identifying a serial port

To identify the serial port name created after connecting the reader to USB port, type the following in the Terminal:

```
$ls /dev/tty.usb*
```



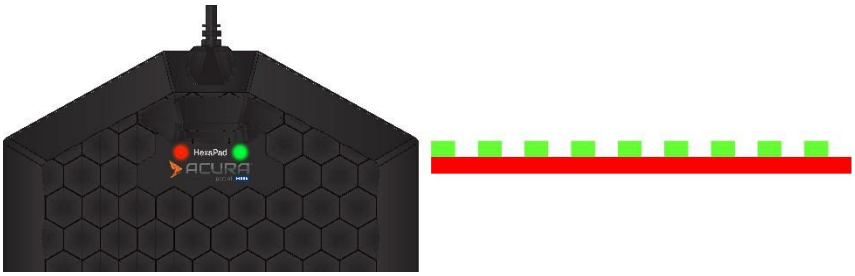
In the example, serial communication with the reader will be via **/dev/tty.usbmodem14101**.

3.3 Reader operating modes

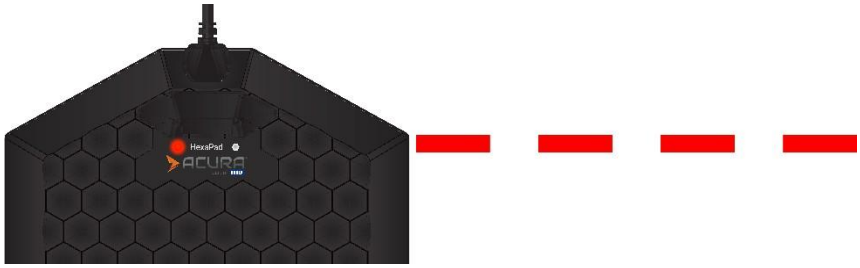
3.3.1 Autonomous mode

In Autonomous mode, the reader continuously attempts to read a tag. When a tag is successfully read, the information is transmitted through the configured communication interface. Starting from firmware version 1.1.3, the option to disable tag reading has been introduced. This is achieved by deactivating the RF signal using the **readtag** command or the **CapsLock** key, effectively putting the reader into an idle state until tag reading is reactivated.

Visual feedback from the reader when it is in Autonomous mode, and performing tag reading:

	Green LED: flashes fast. Red LED: solid on. Reader: in Autonomous Mode. Reading tags: activated (RF on).

Visual feedback from the reader when it is in Autonomous mode, and not performing tag reading:

	Green LED: turned off. Red LED: flashing at a consistent interval. Reader: in Autonomous Mode. Reading tags: deactivated (RF idle).

Public

3.3.2 Turning tag reading ON/OFF/TIME - Autonomous mode

To turn tag reading **ON** and **OFF**, the following options can be used:

1. Using the **readtag command**:
   a. The command **readtag off<CR>** disables the RF (Radio Frequency) and stops reading tags and results report.
   b. The command **readtag on<CR>** activates the RF (Radio Frequency) and resumes tag reading and results report.
   c. The command **readtag 3000<CR>** activates the RF (Radio Frequency) for 3 seconds, then stops reading tags. The reading timeout must be set in milliseconds, from 1000 to 600000 (1second to 600 seconds).

   > **Note: <CR>** indicates a carriage return.

2. Using the **Caps Lock** key from the keyboard (Windows and Linux only):

   To switch between **ON** and **OFF** for reading tags, quickly **press the Caps Lock key two times**, then check the readers LEDs, and beep, to see the change.

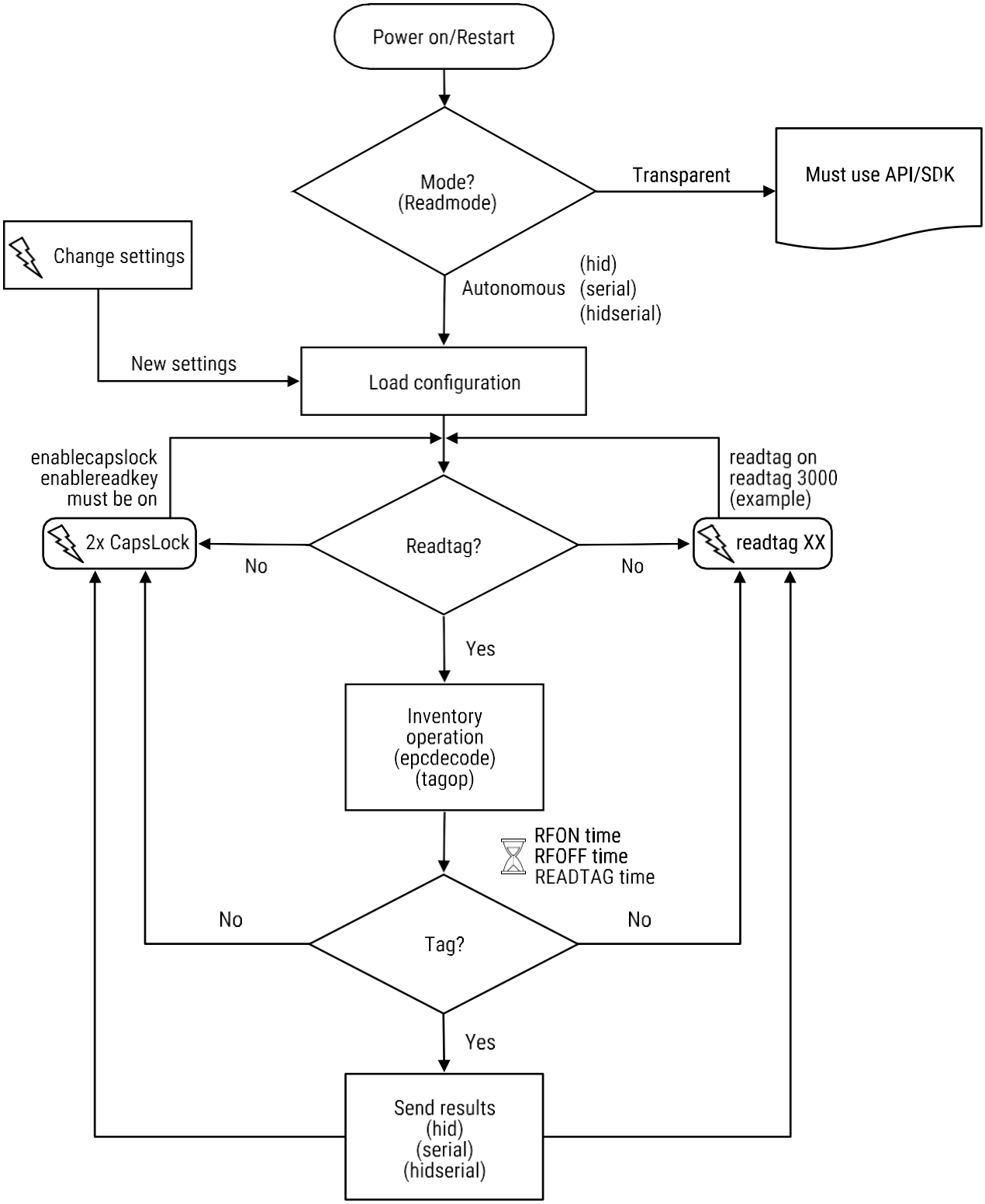
   The command **readtagtimeout** value will determine the tag reading time out initiated via Caps Lock key, if it is set to zero, there will be no time out.

   The command **enablecapslock** and **enablereadtagkey** enables or disables this function.

   > **Note:** The commands **enablecapslock** and **enablereadtagkey** enables or disables this function. To use it, **enablecapslock** and **enablereadtagkey** must be on.

Public

3.3.3 General diagram



Public

3.3.4 Reading tags engine

The algorithm for reading tags in Autonomous mode is described as follows:

- The tag reading can be on,off, or on during a configurable timeout. This can be controlled by the commands **readtag**, **readtagtimeout**, **initreadtag** and **Capslock** keyboard key.
- The tag reading result string format is fixed, not configurable, this is explained in **Reading result format** section.
- The EPC id of the tag will always be reported by the reader in a format determined by the **epcdecode** command.
- Additionally, to the EPC id, a tag memory bank read operation, configured by the **tagop** command, will be reported by the reader concatenated with a configurable **separator** character.
- More tag reading data can be included in the tag reading result report, such as **readcount** and signal level **rssi**, separated by the **separator** character.

If the interface chosen by the **readmode** command is serial, and a tag is read while there is no connection in the serial port generated, the reader will generate audible feedback with fast beeps every time the tag is read.

3.3.5 Reading result format

In Autonomous mode, the reader transmits a fixed formatted string containing the results of tag reads via the selected communication interface.

> **Note:** The size of the result string may vary depending on the tags read. The software that will receive this string, must always handle it using the **<separator>** and **<endofline>** fields to split the desired fields.

The format of the string is as follows:

<epcdecode><separator><tagop><separator><readcount><separator><rssi><endofline>
<epcdecode> EPC id read result, decoded as specified by the epcdecode command. Note: Always reported by the reader in Autonomous mode.
<separator> Separator character is specified by the separator command. Note: It is only reported by the reader if there are more fields in addition to <epcdecode> to report.
<tagop> The result of the tagreading is operation specified by the tagop command. Note: Optional.
<readcount> The quantity of readings for the same tag during an inventory period. The choice to display this value or not is determined by the reportreadcount command. Note: Optional.
<rssi> The RF signal level value in dBm and negative of tag response. The decision to display or hide this value is determined by the reportrssi command. Note: Optional.
<endofline> The end-of-line type is configured using the endofline command. Note: The reader reports this in Autonomous mode if it is configured as anything other than none.

Public

3.3.6 Configuring the reader - Autonomous mode

When the reader is in Autonomous mode, the following configurations can be made:

- Type of communication
- Perform tag reading with or without time out or stay idle
- Reading power
- Reading time
- Write power
- Idle time
- EPC id tag decoding type
- Reading operation in the tag's memory bank
- Select filter to read only specific tag
- EPC Gen2 protocol parameters:
  - BLF (Backscatter Link Frequency)
  - Tari
  - Tag Encoding
  - Session
  - Target
  - Q (Anti-collision)
  - Access password
- Character that separates the fields in the result string
- Type of end-of-line in the result string
- Typing speed for keyboard emulation
- Filter by RSSI (Received Signal Strength Indicator)
- Show or do not show the RSSI value of the read tag
- Show or do not show the quantity of tag reads
- Audible tag reading alert (beep)
- Check internal temperature, status, and firmware versions

3.3.7 How to set up the reader

When the reader is powered **on** in Autonomous mode, it must be configured using ASCII commands sent through the serial port created by the operating system.

> **Note:** To find out which serial port the reader will connect to, please consult **Turning on the reader**.

3.3.8 Operating logic

The configuration of the reader via ASCII commands works with the Message/Response structure:

1. An external host sends a command to the reader.
2. The reader interprets and executes the command included in the received message.
3. The reader responds to the external host.

The messages and responses use ASCII characters with a carriage return terminator **<CR>** (13d) (0x0D), which is the **[ENTER]** in terminal. The messages may be in uppercase, lowercase, or a mixture of both.

> **Note:** The reader echoes each character typed in the serial terminal, and with each new line the reader sends the character **>** to indicate the cursor.

3.3.9 Syntax of messages and responses

> **Note:** Only one value must be passed for each command, the reader will not accept more than one value per command.

The syntax of messages from the host to the reader:

COMMAND VALUE<CR>	Changes the value of the command/parameter.
COMMAND<CR>	Reads the value of the command/parameter.

The syntax of reader response messages to host:

RESULT<CR>	Returns the result of the execution of the last command sent to the reader.

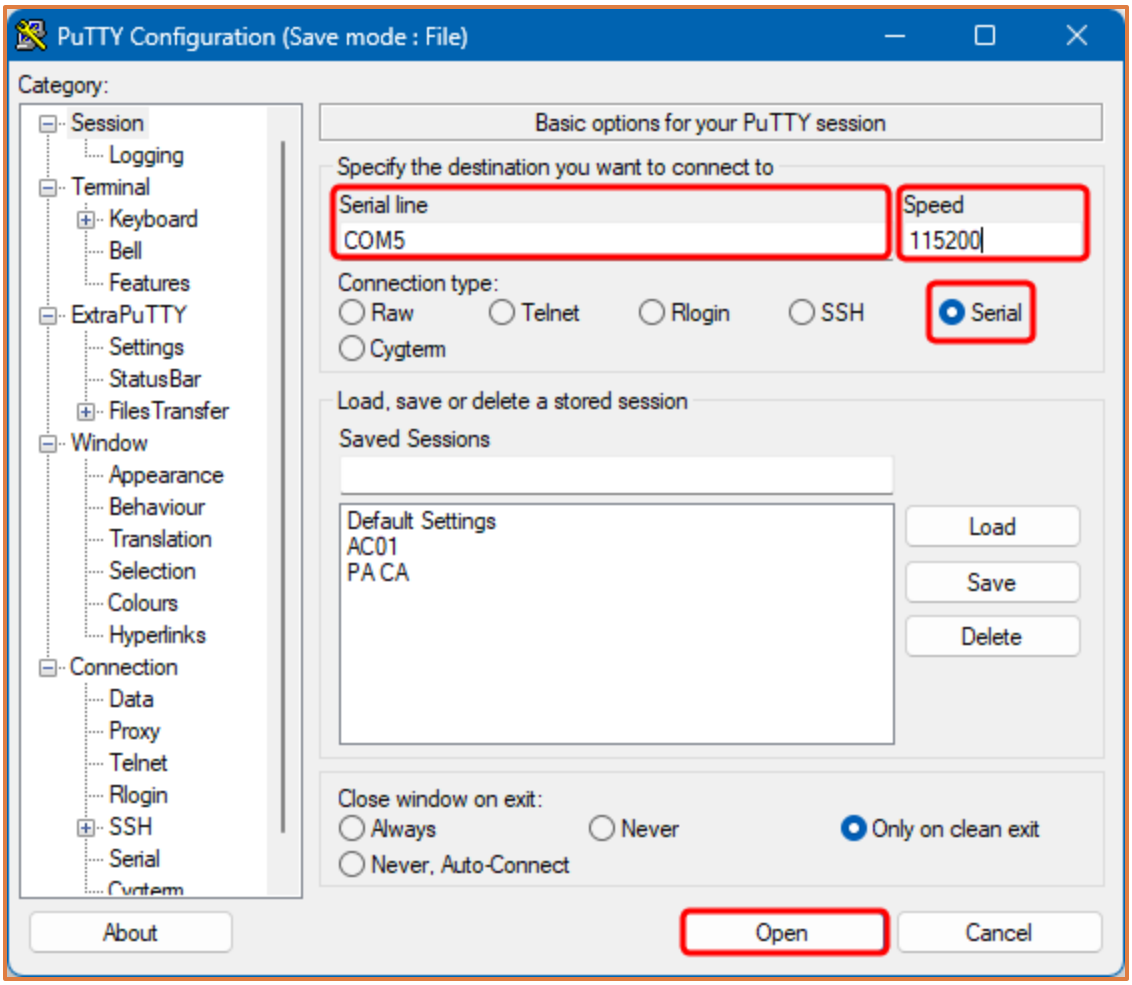
Public

3.3.10 Windows - how to send commands to the reader

To change the readers settings in Autonomous mode, a terminal emulation software can be used. Putty is a free open-source terminal emulation application: **https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html**

To configure Putty:

1. Open Putty and select **Connection type:** > **Serial**, enter the **Serial line** as the port generated by the reader **(COMxx)**, and set the **Speed** to **115200**.



2. Click **Open** to open the terminal screen for entering commands.

3.3.11 Linux and macOS - how to send commands to the reader

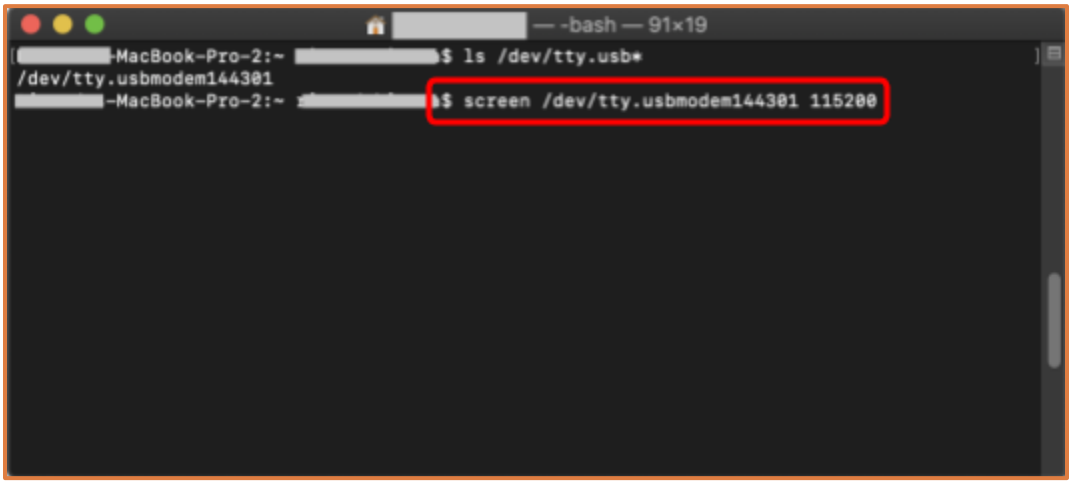
> **Note:** On the Linux operating system, you can also use Putty, see: **Windows - how to send commands to the reader**.

For macOS and Linux, you can use the **screen** program.

> **Note:** To find out on which serial port the reader will respond, please consult **Turning on the reader**.

macOS

```
$ screen /dev/tty.usbmodemXXXXXX 115200
```

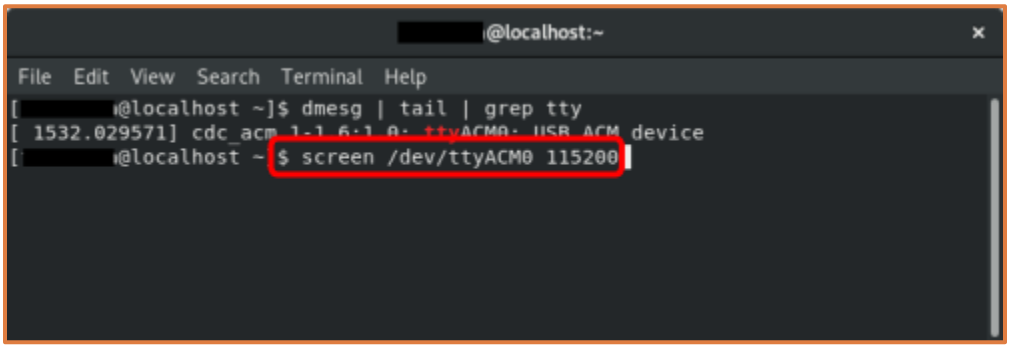


Linux

> **Note:** In Linux, to have permission to use the serial port, run the command with superuser privileges (sudo) or add your user to the dialout group using the following command: **$ sudo usermod -a -G dialout $USER**
> For this to take effect, you need to restart the system.

To connect using the screen:

```
$ screen /dev/ttyACMx 115200
```



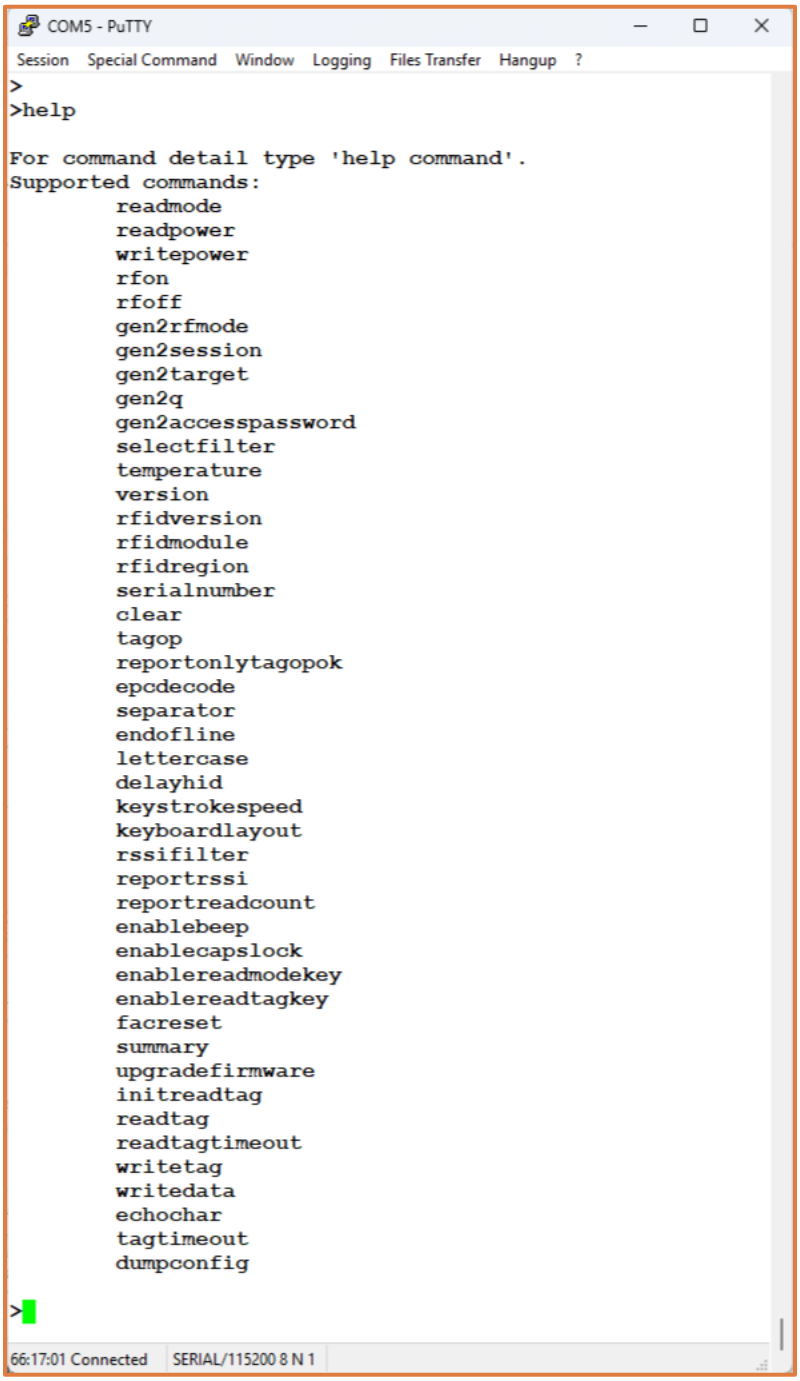
To exit and terminate the screen command, use **Ctrl+A** then **d.**

Public

3.4 Configuration commands

Note: It is not always easy to remember the command name or accepted values, for that, use the **help** command. When executed without any values, it lists all the commands supported by the reader. To get details on a specific command, simply type **help (command)<CR>** – this is a quick way to access the readers configuration information.

3.4.1 help command



Public

3.4.2 summary command

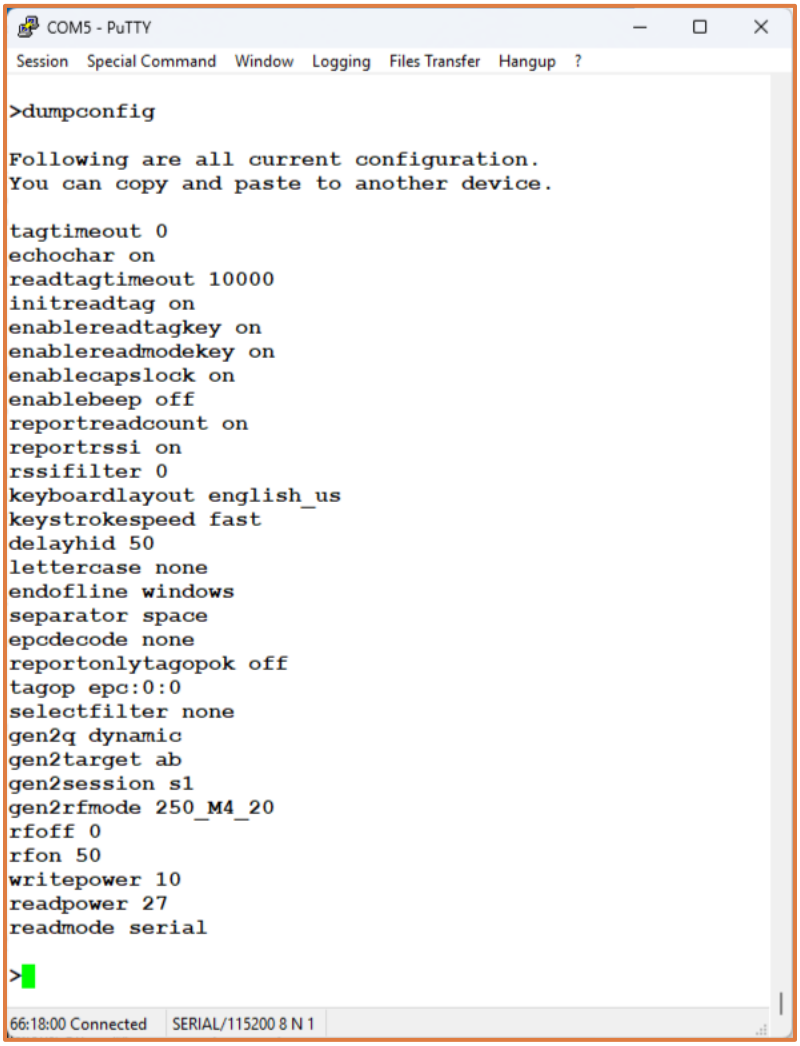
The **summary** command returns all the current values and status of the reader parameters in a formatted list as follows:

```
COM5 - PuTTY
Session  Special Command  Window  Logging  Files Transfer  Hangup  ?

>summary

Summary of all commands with their respective values.
        Command                 Value:

        readmode                serial
        readpower               27 dbm
        writepower              10 dbm
        rfon                    50 ms
        rfoff                   0 ms

        gen2rfmode              250_M4_20
        gen2session             s1
        gen2target              ab
        gen2q                   dynamic
        gen2accesspassword      00000000
        selectfilter            none

        temperature             33 C
        version                 2.1.6
        rfidversion             02.01.07.1C-20
        rfidmodule              deka-module
        rfidregion              FCC Reduced
        serialnumber            1901408

        tagop                   epc:0:0
        reportonlytagopok       off
        epcdecode               none
        separator               ' '
        endofline               windows
        lettercase              none
        delayhid                50 ms
        keystrokespeed          fast
        keyboardlayout          english_us

        rssifilter              0 (no filter)
        reportrssi              on
        reportreadcount         on
        enablebeep              off
        enablecapslock          on
        enablereadmodekey       on
        enablereadtagkey        on
        initreadtag             on
        readtag                 off
        readtagtimeout          10000 ms

        echochar                on
        tagtimeout              0 ms (no filter)

>D
66:15:40 Connected   SERIAL/115200 8 N 1
```

Public

3.4.3 dumpconfig command

The **dumpconfig** command returns all the current values of the reader parameters in a simple formatted list. This entire list can be copied from the terminal and pasted to another terminal connected to another reader.



3.4.4 Summary of configuration commands and default values

	Command	Short description	Default value
Operation	readmode	Reader operating mode.	hid
	readpower	Read power in dBm.	10
	writepower	Write power in dBm.	10
	rfon	Time in milliseconds of RF ON.	250
	rfoff	Time in milliseconds of RF OFF.	250
	linkread	Link both tag and barcode read operations.	none
Tag Gen2 protocol	gen2blf	Air Backscatter Link Frequency in KHz.	250
	gen2tari	Tari (bit pulse length) in us.	25
	gen2rfmode	RF mode (BLF_TagEncoding_Tari)	250 m4 20
	gen2tagencoding	Air tag-reader communication encoding.	m4
	gen2session	Tag Session value.	s1
	gen2target	Tag Target value.	a
	gen2q	q value, which determines the tag population.	dynamic
	gen2accesspassword	Tag gen2 access password.	00000000
	selectfilter	Tag select filter.	none
Tag reading	initreadtag	Initial state of reading tag.	on
	readtag	Current reading tag condition.	on
	readtagtimeout	Read timeout in ms trigger by Capslock key	0
	tagtimeout	Same tag id filter, in milliseconds.	0
	rssifilter	Filter by the value of tag RSSI in dbm.	0
	tagop	Read tag operation.	none
	reportonlytagopok	Report only if the tagop was successfully executed.	off
	epcdecode	How to decode the tag EPC id.	none
Tag writing	writetag	Write tag EPC id with or without select filter.	
	writedata	Write tag memory bank with or without select filter.	

Public

	Command	Short description	Default value
Messages reply	separator	Character between the results fields.	"space"
	endofline	Endofline for tag reads report string.	windows
	keystrokespeed	Keystroke speed for Wedge emulation.	fast
	keyboardlayout	Layout for Wedge emulation.	english_us
	lettercase	Force letters to be in uppercase or lowercase via keyboard emulation.	
	delayhid	Delay in ms between multiple transmission via keyboard emulation.	
	reportrssi	Report the tag rssi value.	off
	reportantenna	Report the antenna the tag was read.	off
	reportreadcount	Report the tag readcount value of each rfon.	off
information	summary	List all commands with their respective values.	
	temperature	Temperature of the reader in Celsius degree.	
	version	Processor firmware version.	
	rfidversion	RFID firmware version.	
	rfidmodule	RFID module type.	
	rfidregion	Current RFID region.	
	serialnumber	Reader serial number.	
	dumpconfig	Lists all the current configuration of the device in a format that can be copied and pasted via terminal in another device.	
Barcode reading	barcodemode	Barcode scan mode operation.	hid
	initreadbarcode	Initial state of barcode reading.	on
	readbarcode	Current barcode reading condition.	on
	barcodecommand	Send command to barcode reader.	
	barcodedecode	How to decode the barcode readings.	none
	configbybarcode	Permission to configure the reader using barcodes.	off
	configbarcodeprefix	Prefix character of the barcode to configure the reader.	*
Miscellaneous	enablebeep	Enable beep sound.	on
	enablecapslock	CapsLock to switch readmode Transparent/HidMaster switch for Capslock key.	on
	enablereadmodekey	Capslock key to switch readmode Transparent/Autonomous.	
	enablereadtagkey	Capslock key to swtch readtag.	
	echochar	Echo character through serial comm.	on
	facreset	Returns all configuration to factory default.	
	clear	To clear the terminal screen.	
	status	Status of the reader.	
	help	List all commands or display details of specific one.	

Public

3.4.5 Response messages to commands

When receiving a command, the reader executes and responds with one of the following messages:

Type	Message
Success	ok - command executed. ok - parameter has been changed. ok - parameter already has this value. ok - Transparent mode off, resetting reader. !!! Factory Reset in action !!! The reader will be restarted. !!! Bootloader will be executed !!! The reader will be restarted.
Errors	error - command not found. error - too many args. error - value not supported for this RF Module. error - command not supported for this hardware. error - command not supported for this comm interface. error - value out of range. error - rs232baudrate must be 115200. error - gen2 wrong parameters combination. error - internal buffer full. error - need a command to send. error - barcode is not enabled.

Public

3.5 Configuration command descriptions

3.5.1 readmode

Syntax	readmode<CR> readmode value<CR>
Function	Check (no arguments) or change the mode of operation of the reader.
Description	The reader can work in Autonomous mode, with automatic tag readings, or in Transparent mode, where a software can have full control of the reader for tag reading and writing.
Values	hid - Keyboard emulation mode. serial - Serial communication mode, the read result is sent via serial communication on the USB virtual port. hidserial - Keyboard emulation mode and USB Serial, the reading result is sent via keyboard emulation and via serial communication on the USB virtual port. transparent - The reader must be controlled by a software that has full control to read and write tags.
Default	hid
Example	To get the current operation mode: readmode<CR> hid<CR> To change the mode of operation (Autonomous reporting via serial only): readmode serial<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.2 rfon

Note:

- The reading results are reported by the reader after the sum of the **RF On** and **RF Off** time periods.
- If the application requires reading multiple tags simultaneously, set **RF On** time to more than **250ms** to allow sufficient time to power up all tags within the readers field.

Syntax	rfon<CR> rfon value<CR>
Function	Check (no arguments) or change the read time in ms (milliseconds).
Description	The reading time is equivalent to the time when the reader transmits RF power to its antenna. When you have large number of tags to read at the same time, the reading time should be increased.
Values	50..5000
Default	250
Note	Tag reading results are reported by the reader after the sum rfon + rfoff time periods.
Example	To get the current reading period: rfon<CR> 250<CR> To change the reading time to 320ms: rfon 320<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.3 rfoff

Note: The reading results are reported by the reader after the sum of the **RF On** and **RF Off** time periods.

Syntax	rfoff<CR> rfoff value<CR>
Function	Check (no arguments) or change the period of RF idle in ms (milliseconds).
Description	This is the period of RF idle with no tag reading.
Values	0..1000
Default	250
Note	Tag reading results are reported by the reader after the sum rfon + rfoff periods.
Example	To get the current reading period: rfoff<CR> 250<CR> To change the RF idle to 0ms: rfoff 0<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

3.5.4 linkread

Syntax	linkread<CR> linkread value<CR>
Function	Check (no arguments) or change the type of link between the events of the read RFID tags and barcodes.
Description	The reader can work with independent readings between RFID tags and barcodes or establish reading links between them.
Values	none No reading link is established; the read RFID tags and barcodes have independent reading control. equal It establishes a link between RFID tags and barcode reading controls, forcing both to be in the same state (on/off). inverse It reverses the link between RFID tag reading and barcode reading controls (related to readtag and readbarcode). Only one remains on and operational at a time.
Default	none
Note	For inverse, you may change the initreadbarcode or initreadtag. Supported only by HexaPad-10 BC model.
Example	To check the value: linkread<CR> none<CR> To establish a link between RFID tags and the barcode: linkread equal<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.5 gen2blf

Syntax	gen2blf<CR> gen2blf value<CR>
Function	Check (no arguments) or change the "Backscatter Link Frequency" parameter of the Gen2 air protocol.
Description	BLF is the data rate in KHz of transmission by air between a tag and the reader.
Values	250 320 640
Default	250
Note	By changing the BLF to 640 or 320 KHz, the tari value and tag encoding will automatically change to 6.25 us and fm0 respectively.
Example	To get the current BLF: gen2blf<CR> 250<CR> To change the BLF to 640Khz gen2blf 640<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR> error - value not supported for this RF Module.<CR>

Public

3.5.6 gen2tari

Note: The value of Tari should be combined with the value of BLF.

Syntax	gen2tari<CR> gen2tari value<CR>
Function	Check (no arguments) or change the "Tari" parameter of the Gen2 air protocol.
Description	Tari is the time in us (microseconds) of the pulse of each bit transmitted by the tag.
Values	6.25 12.5 25
Default	25
Note	Values 12.5 us and 25 us can only be set if the gen2blf is set to 250 KHz. The values 6.25 us and 12.5 us are supported only by the internal RF module micro-module type, which can be checked by rfidmodule command.
Example	To get the current Tari: gen2tari<CR> 25<CR> To change the Tari to 6.25Khz gen2tari 6.25<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR> error - value not supported for this RF Module.<CR> error - gen2 wrong parameters combination.<CR>

3.5.7 gen2tagencoding

Syntax	gen2tagencoding<CR> gen2tagencoding value<CR>
Function	Check (no arguments) or change the "Tag data encoding" parameter of the Gen2 air protocol.
Description	It is the encoding format of the tag response.
Values	fm0 m2 m4 m8
Default	m4
Note	m2, m4 and m8 can only be configured if the blf is 250 KHz. fm0 has a low read sensitivity rate but a higher speed in decoding each bit. m8 has a high read sensitivity rate but a lower speed in decoding each bit.
Example	To get the current Tari: gen2tagencoding<CR> m4<CR> To change the Encoding format to FM0Khz gen2tagencoding FM0<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR> error - value not supported for this RF Module.<CR> error - gen2 wrong parameters combination.<CR>

3.5.8 gen2session

Syntax	gen2session<CR> gen2session value<CR>
Function	Check (no arguments) or change the "Session" parameter the Gen2 air protocol inventory operation.
Description	In which session the reader will read the tag.
Values	s0 s1 s2 s3
Default	s1
Note	The combination of gen2session and gen2target values reflects the read rate of the same tag.
Example	To get the current Session: gen2session<CR> s1<CR> To change the Session to S0 gen2session s0<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.9 gen2target

Syntax	gen2target<CR> gen2target value<CR>
Function	Check (no arguments) or change the "Target", or Inventory Flag of the Gen2 air protocol.
Description	Which inventory flag (target), the reader should search for a tag.
Values	a b ab ba
Default	a
Note	The combination of gen2session and gen2target values reflects the read rate of the same tag.
Example	To get the current Target: gen2target<CR> a<CR> To change the Target to ab gen2target ab<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.10 gen2q

Note: Use **gen2q** as **dynamic** if the quantity of tags in the reading field varies or is unknown.

Syntax	gen2q<CR> gen2q value<CR>
Function	Check (no arguments) or change the "Q" parameter of the Gen2 air protocol.
Description	It is a numeric value that the reader uses to regulate the probability of a tags response, used for anti-collision when multiple tags are in the readers reading field. 2^q equals the number of tags to be read at the same time.
Values	dynamic - The reader automatically adjusts the Q value during readings. 0..15 - Fixed value for 2^q number of tags in the reading field of the reader.
Default	dynamic
Note	If the number of tags to be read at the same time varies, leave the q value in dynamic, so the reader automatically adjusts the best value for each reading.
Example	To get the current Q value: gen2q<CR> dynamic<CR> To change the Q value to an estimated tag population of 8 tags. gen2q 3<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.11 gen2accesspassword

Note: To read the protected memory region of the tag (Reserved), the tags access password must be provided to the reader through this command.

Syntax	gen2accesspassword<CR> gen2accesspassword value<CR>
Function	Check (no arguments) or change the "Access password" of the tag.
Description	It is 32 bits length hexa number.
Values	00000000..ffffffff
Default	0
Note	To read the reserved memory bank of the tag with the tagop command, the password must be the same as the tag.
Example	To get the current access password: gen2accesspassword<CR> 00000000<CR> To change the access password to 12345678. gen2accesspassword 12345678<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.12 selectfilter

Syntax	selectfilter<CR> selectfilter value<CR>
Function	Check (no arguments) or set the Select filter tag using a mask value in a specific bank.
Description	This filter allows you to specify which tags should respond in an inventory. This feature allows you to isolate specific tags in large population. Tags that do not match the specified filter will not transmit a response, which greatly reduces the amount of RF traffic.
Values	res:mask:bp:bl - Reserved bank.* epc:mask:bp:bl - EPC bank.* tid:mask:bp:bl - TID bank.* user:mask:bp:bl - User bank.* none - No Select filter. * mask - mask value [hexa], bp - bit pointer, bl - bit length, max 96.
Default	none
Example	To get the current Select filter: selectfilter<CR> none<CR> To read only tags with EPC id starting with 0xAC01: selectfilter epc:AC01:32:16 ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.13 initreadtag

Syntax	initreadtag<CR> initreadtag value<CR>
Function	Check (no arguments) or set the read tag state (on/off) of the reader when it is initialized.
Description	The reader can always be preset to have the RF signal either turned off or turned on when initialized.
Values	on off
Default	on
Example	To check the initreadtag value: initreadtag<CR> on<CR> To disable the reader (RF signal) when it is initialized: initreadtag off<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.14 readtag

Syntax	readtag<CR> readtag value<CR>
Function	Check (no arguments) or turn RF on/off.
Description	Enables or disables tag reading by turning off the readers RF signal.
Values	on - Enables tag reading, RF on. off - Disables tag reading, RF idle.
Default	Depends on how initreadtag is setup. Volatile data.
Note	RF on/off can be toggled by pressing 2x fast the CapsLock key, the enablecapslock must be set to on.
Example	To check the current value: readtag<CR> on<CR> To change the RF to idle: readtag off<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.15 tagtimeout

Syntax	tagtimeout<CR> tagtimeout value<CR>
Function	Check (no arguments) or change the reading interval time in milliseconds (ms) for the same reported tag (same EPC filter).
Description	With this time filter it is possible to manipulate the same tag read rate.
Values	0..300000 (time in milliseconds)
Default	0 ms
Example	To check the reading interval time of the same tag: tagtimeout<CR> 1000<CR> To change the reading interval time of the same tag to 500 ms (0.5 s): tagtimeout 500<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.16 rssifilter

Note: The filter based on the tags RSSI value is used to restrict undesired reading areas around the reader. It should always be used in conjunction with the reading power.

Syntax	rssifilter<CR> rssifilter value<CR>
Function	Check (no arguments) or change the cutoff value of the RSSI filter for reading.
Description	The signal level of the tag indicates whether the tag is near or away from the antenna, when closer to the antenna, the stronger the RSSI (closer to zero) and vice versa. If a value is specified for the filter, only tags with RSSI equal to or greater than that specified will be reported by the reader. The signal level of the tag is measured in dBm and is a negative value, for example, the power in Watts is always less than 1mW.
Values	0..99
Default	0
Notes	To disable the filter, write the value 0. The specified value should not contain the minus sign.
Example	To check the RSSI filter value: rssifilter<CR> 0<CR> To change the RSSI filter value to 45: rssifilter 45<CR> ok - parameter has been changed.<CR>
Returns	"value"<CR> ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.17 tagop

Syntax	tagop<CR> tagop value<CR>
Function	Check (no arguments) or change the tag read operation.
Description	With the tag read operation, it is possible to read any memory bank of the tag.
Values	none - No memory bank read operation. autoid - Reads Acura Autoid System tags. reserved:wp:wl - Reads the Reserved memory bank of the tag. epc:wp:wl - Reads the EPC memory bank of the tag. tid:wp:wl - Reads the TID memory bank of the tag. user:wp:wl - Reads the USER memory bank of the tag.
Default	none
Note	wp - word pointer, indicates the initial word (16 bits) read position in the specified memory bank. wl - word length, indicates the number of words to read in the specified memory bank. Maximum value for wp and wl is 32. To read the entire memory bank, simply specify zero for wp and wl.
Example	To read all the contents of the User memory bank of the tag: tagop user:0:0<CR> ok - parameter has been changed.<CR> To read four words of the Tid memory bank from the second word: tagop tid:2:4<CR> ok - parameter has been changed.<CR> To not read any specific tag memory bank, just the EPC number: tagop none<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.18 reportonlytagopok

Syntax	reportonlytagopok <CR> reportonlytagopok value<CR>
Function	Check (no arguments) or set the choice of report only if the tagop was successfully executed.
Description	This command is used to force the reader to send only the results when the tagop command is executed successfully.
Values	on off
Default	off
Note	This has effect only if the command tagop is other than none.
Example	To check the reportonlytagopok value: reportonlytagopok <CR> off<CR> To change the value to set the reportonlytagopok on: reportonlytagopok on <CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.19 epcdecode

Syntax	epcdecode<CR> epcdecode value<CR>
Function	Check (no arguments) or change the type of decodification will be used to process the EPC id of the tag.
Description	The tag to be read can be encoded for a specific application; for example, in checkout applications, the tag will be encoded using the GS1 standard, while in maintenance applications, the tag may be encoded using the ASCII table. The EPCID decoding provides a readable result for each respective application.
Values	none - No decoding is performed and the epcid result of the tag is reported in hexadecimal. gs1string - GS1 standard, that returns the "GS1 element string". gs1epcuri - GS1 standard, that returns the "EPC Tag URI". gs1epcpureuri - GS1 standard, that returns the "EPC Pure Identity URI". gs1gtin13 - GS1 standard, that returns code GTIN13 without the serial number. gs1gtin13slz - Suppress leading zeros. gs1sgtin13 - GS1 standard, that returns code GTIN13 + serial number. gs1sgtin13slz - Suppress leading zeros. gs1crystal - Crystal custom standard, that returns GTIN13 + RFID + serial 12. gs1aistrjson – GS1 AI String in JSON. gs1bareid - GS1 Bare Identifier. ascii - Decodes the EPCID of the tag according to the Ascii table. decimal:dp:dl - Decimal value of EPC split range of dp:dl. mid:dp:dl - Substring of EPC in hexadecimal. Max of 32 digits.
Default	None
Note	For the GS1 standard, the sgtin-96 and sgtin-198 schemes are supported. The wiegand26 and magstripe values can be used in conjunction with the autoid type tagop. If the reader is set to a specific epcdecode type and the tag being reading has a wrong format, a beep sound pattern different from the read sound pattern is executed by the reader. dp - digit pointer, indicates the initial digit to read in the EPC. dl - digit length, indicates the number of digits to read in the EPC. Maximum value for dl is 32.
Example	To read the GTIN13 + serial number of a tag encoded using the GS1 standard: epcdecode gs1sgtin13<CR> ok - parameter has been changed.<CR> To read the GS1 element string of a tag encoded using the GS1 standard: epcdecode gs1string<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.20 separator

Note: The separator character, along with the end-of-line, should be used to split the result string by the software that will receive this string.

Syntax	separator<CR> separator value<CR>
Function	Check (no arguments) or change the separation character of the fields of the reading result of a tag.
Description	It is a character that separates the fields in the tag reading result string.
Values	space To have a space as a separator. ‘char’ Any readable character in the Ascii table.
Default	' ' (space)
Notes	For space, you must write space as a value for the command.
Example	To check the separator: separator<CR> space<CR> To change the tab to '|': separator |<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.21 endofline

Note: The separator character, along with the end-of-line, should be used to split the result string by the software that will receive this string.

Syntax	endofline<CR> endofline value<CR>
Function	Check (no arguments) or change the end-of-line character.
Description	Get (no arguments) or set the endofline for tag reads report string.
Values	windows:qt Windows standard end-of-line <CR><LF>. unix:qt Unix standard end-of-line <LF>. macintosh:qt macOS standard end-of-line <CR>. tab:qt The end-of-line will be the tabulation character Tab. none No end-of-line in the result string.
Default	windows
Notes	qt - Number of end-of-line repetitions, values 1 to 10. If :qt is omitted, it is considered only one end of line.
Example	To check which end-of-line is configured: endofline<CR> windows<CR> To change the end of line to Tab and repeat 3 times: endofine tab:3<CR> ok - parameter has been changed.<CR> To change the end-of-line to Windows and repeat it 5 times: endofline windows:5<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.22 keystrokespeed

Note: This value should only be changed if the Host that will receive the result string via keyboard emulation experiences issues due to the typing speed exerted by the reader.

Syntax	keystrokespeed<CR> keystrokespeed value<CR>
Function	Check (no arguments) or change the typing speed of the keyboard emulation.
Description	The typing speed in keyboard emulation, when readmode is as hid or hidserial, can be changed to three different values (fast, medium, and slow).
Values	fast Keystroke at fast speed, T=4ms. medium Keystroke at medium speed, T=32ms. slow Keystroke at slow speed, T=64ms.
Default	fast
Example	To check the typing speed: keystrokespeed<CR> fast<CR> To change to slow speed: keystrokespeed slow<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.23 reportrssi

Command	reportrssi<CR> reportrssi value<CR>
Function	Check (no arguments) or change the transmission of the RSSI value from the read tag.
Description	Enables or disables sending the RSSI value of the tag read in the result string.
Values	on off
Default	off
Notes	The RSSI value is reported with the negative sign '-'.
Example	To check the value: reportrssi<CR> off<CR> To change the value to include the RSSI value in the result string: reportrssi on<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.24 reportantenna

Command	reportantenna<CR> reportantenna value<CR>
Function	Check (no arguments) or change the transmission of the antenna from the read tag.
Description	Enables or disables sending the antenna the tag was read in the result string.
Values	on off
Default	off
Example	To check the value: reportrssi<CR> off<CR> To change the value to include the RSSI value in the result string: reportrssi on<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

3.5.25 reportreadcount

Syntax	reportreadcount<CR> reportreadcount value<CR>
Function	Check (no arguments) or change the transmission of the tags read quantity.
Description	Enables or disables the transmission of the tags read quantity during the "rfon" period in the result string.
Values	on off
Default	off
Notes	The maximum value for the read quantity is 255.
Example	To check the value: reportreadcount<CR> off<CR> To change the value to include the read quantity in the result string: reportreadcount on<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

3.5.26 summary

Syntax	summary<CR>
Function	Lists all the values configured in the reader.
Errors	error - too many args.<CR>

3.5.27 temperature

Syntax	temperature<CR>
Function	Check (no arguments) the internal temperature of the reader in Celsius degree.
Errors	error - too many args.<CR>

3.5.28 version

Syntax	version<CR>
Function	Get (no arguments) the processor firmware version.
Errors	error - too many args.<CR>

3.5.29 rfidversion

Syntax	rfidversion<CR>
Function	Get (no arguments) the RFID firmware version.
Errors	error - too many args.<CR>

Public

3.5.30 rfidmodule

Syntax	rfidmodule<CR>
Function	Get (no arguments) the RFID module type.
Errors	error - too many args.<CR>

3.5.31 rfidregion

Note: The readers regional configuration is established within the firmware.

Syntax	rfidregion<CR>
Function	Get (no arguments) the current RFID region.
Errors	error - too many args.<CR>

3.5.32 serialnumber

Syntax	serialnumber<CR>
Function	Get (no arguments) the reader serial number.
Errors	error - too many args.<CR>

3.5.33 dumpconfig

Syntax	dumpconfig<CR>
Function	Lists all the current configuration of the device in a format that can be copied and pasted via terminal in another device.
Description	This command can compile a list of the present reader configurations, streamlining the process of archiving or duplicating them across other readers with greater efficiency.
Errors	error - too many args.<CR>

Public

3.5.34 barcodemode

Syntax	barcodemode<CR> barcodemode value<CR>
Function	Check (no arguments) or change the barcode reader operating mode.
Description	The barcode reader will always operate in Autonomous mode with automatic readings.
Values	hid Keyboard emulation mode, where when a barcode is read, the result is sent to the computer as if it were typed quickly. hidserial Keyboard emulation mode and USB Serial mode, where the reading result is sent via keyboard emulation and also through serial communication on the virtual USB port. serial Serial communication mode, where the reading result is sent through serial communication on the virtual USB port.
Default	hid
Note	When readmode is set to transparent, the barcodemode will change to hid. Supported only by HexaPad-10 BC model.
Example	To check the operating mode: barcodemode<CR> hid<CR> To change the operating mode (Autonomous reporting via serial only): barcodemode serial<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.35 initreadbarcode

Syntax	initreadbarcode<CR> initreadbarcode value<CR>
Function	Check (no arguments) or set the state (on/off) of the barcode scanner when the reader is initialized.
Description	The reader can always be preset to have the barcode scanner either turned off or turned on when initialized.
Values	on off
Default	on
Note	Supported only by HexaPad-10 BC model.
Exemplo	To check the value: initreadbarcode<CR> on<CR> To disable the scanner when the reader is initialized: initreadbarcode off<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.36 readbarcode

Syntax	readbarcode<CR> readbarcode value<CR>
Function	Check (no arguments) or enable/disable the barcode scanner.
Description	Enables or disables barcode reading by turning off the readers scanner.
Values	on off
Default	Depends on how initreadbarcode is setup. Volatile data.
Note	Supported only by HexaPad-10 BC model.
Example	To check the value: readbarcode<CR> off<CR> To enable the barcode scanner and barcode reading: readbarcode on<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

3.5.37 barcodecommand

Syntax	barcodecommand value<CR>
Function	• Set the state of the barcode scanner LEDs. • Enable re-read timeout. • Set the re-read time in milliseconds for the same barcode.
Description	According to the barcode commands, it is possible to turn on/off the white and red LEDs of the barcode scanner, enable re-read timeout, set the time in milliseconds (1s, 2s, and 3s) for the same barcode.
Values	LED STATUS: Turn on the white LED: barcodecommand @ILLSCN1 Turn off the white LED: barcodecommand @ILLSCN0 Turn on the red LED: barcodecommand @AMLENA1 Turn off the red LED: barcodecommand @AMLENA0 RE-READ TIMEOUT: Enable re-read timeout: barcodecommand @RRDENA1 RE-READ TIME IN MILLISECONDS FOR THE SAME BARCODE: 1s barcodecommand @RRDDUR1000 2s barcodecommand @RRDDUR2000 3s barcodecommand @RRDDUR3000
Examples	To turn off the red scanner LED: barcodecommand @AMLENA0<CR> 0000@AMLENA0.<CR> To enable the re-read timeout. barcodecommand @RRDENA1<CR> 0000@RRDNA1<CR> To set the barcode reread time to 1 second (1000 ms): barcodecommand @RRDDUR1000<CR> 0000@RRDDUR1000<CR>

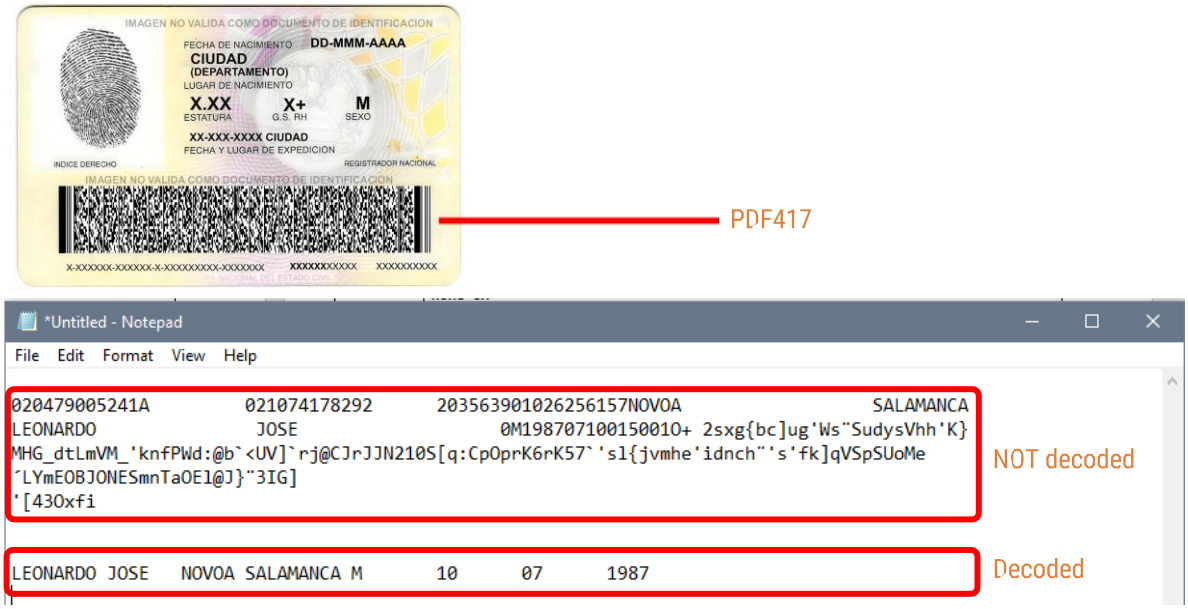
Public

Default	white LED on red LED on re-read time enabled re-read time value 2s
Note	Supported only by HexaPad-10 BC model.
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

3.5.38 barcodedecode

Syntax	barcodedecode<CR> barcodedecode value<CR>
Function	Check (no arguments) or set how to decode the barcode readings.
Description	The barcode scanner can be configured to either read pure barcode information or, for specialized applications, it can conduct pre-decoded scanning of the Colombian ID (PDF417).
Values	none - No decode. colombianidcard - PDF417 decode for Colombian ID card.
Default	none
Note	Supported only by HexaPad-10 BC model.
Example	To check the barcodedecode value: barcodedecode <CR> none<CR> To enable decoding for Colombian ID card. barcodedecode colombianidcard<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

The **PDF417** format for Colombian identification is shown, demonstrating how it is read when not decoded, and then how it is read once it has been decoded:



3.5.39 configbybarcode

Note:

- After activating the barcode configuration, it is also important to check or set the prefix for the barcodes that will be considered configuration barcodes.
- The configuration barcode must be generated in **text** format.

Syntax	configbybarcode<CR> configbybarcode value<CR>
Function	Check (no arguments) or set the permission to configure the reader using QR code.
Description	Enables or disables the ability to use a QR code for reader configuration. When enabled, it is possible to generate a QR Code to configure the reader, following the required standard.
Values	on off
Default	off
Note	Supported only by HexaPad-10 BC model.
Example	To check the value: configbybarcode <CR> off<CR> To enable the readers configuration using barcode: configbybarcode on<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

3.5.40 configbarcodeprefix

Note: Only a single character can be used as the prefix for the configuration barcode.

Syntax	configbarcodeprefix<CR> configbarcodeprefix value<CR>
Function	Check (no arguments) or set the prefix character of the QR code to configure the reader.
Description	The configuration of the prefix character to be used. All configuration QR codes should begin with this designated prefix followed with no spaces with the desired ASCII commands.
Values	space –For the space character. 'char' –For any character, only one character is accepted.
Default	*
Example	To check the character configured as prefix to the config barcode: configbarcodeprefix <CR> *<CR> To set the barcode prefix with the character 'A': configbarcodeprefix A<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.5.41 enablebeep

Syntax	enablebeep<CR> enablebeep value<CR>
Function	Check (no arguments) or change the beep with each reading of the tag.
Description	Enables or disables beeping with each tag reading.
Values	on off
Default	on
Example	To check the value: enablebeep<CR > off<CR> To turn off the tag read alert sound: enablebeep off<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

3.5.42 enablecapslock

Note: For more details on changing the readers operating mode, please refer to **Switching between Autonomous and Transparent modes**.

Syntax	enablecapslock<CR> enablecapslock value<CR>
Function	Check (no arguments) or change the use of the shortcut to change read mode.
Description	Enables or disables the use of the CapsLock 5x quick-type key, to switch between Autonomous and Transparent modes.
Values	on off
Default	on
Notes	This feature only works on Windows and Linux operating systems. An alternative to exit Transparent mode is to send the command $ off<CR> to the reader.
Example	To check the value: enablecapslock<CR> off<CR> To change the value to disable shortcut use: enablecapslock on<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

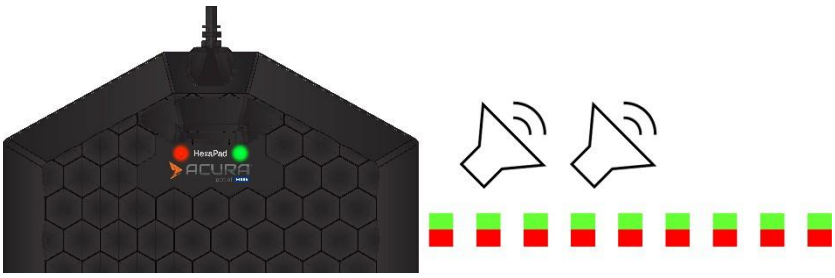
3.5.43 echochar

Syntax	echochar<CR> echochar value<CR>
Function	Check (no arguments) or turn echo on/off.
Description	Enables or disables the echo of each byte sent by serial communication in Autonomous mode.
Values	on off
Default	on
Notes	When configured to off" typing in terminal software will not appear unless character echoing is set up in the terminal software. It is recommended to turn off the echochar when custom developed software will send the ASCII commands to the reader.
Example	To check the value: echochar<CR> on<CR> To disable the echo of sent bytes: echochar off<CR> ok - parameter has been changed.<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

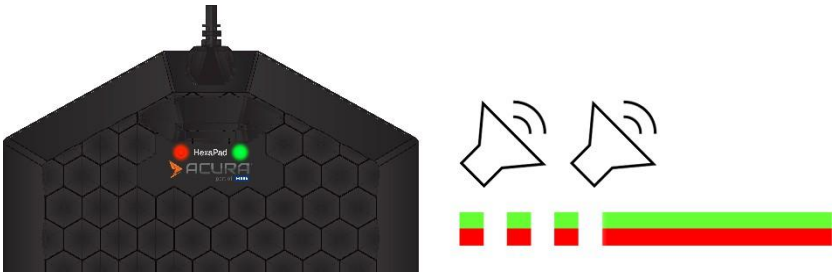
3.5.44 facreset

Syntax	facreset<CR>
Function	Returns all values to the factory default.
Errors	error - too many args.<CR>

	Beep, Green and Red LEDs flash rapidly simultaneously for five seconds to indicate that it is in the factory reset process.

3.5.45 upgradefirmware

Syntax	upgradefirmware<CR>
Function	Starts the reader firmware update process.
Example	To check the commands accepted by the reader: help<CR> command list... <CR> upgradefirmware !!! Bootloader will be executed!!! The reader will be restarted.
Errors	error - too many args.<CR>

	Beep, Green and Red LEDs flash rapidly three times, and then both LEDs are solid, to indicate the bootloader execution.

3.5.46 help

Syntax	help<CR> help value<CR>
Function	Lists all commands (no arguments) or provides detailed help for the given command.
Description	For detailed help on a particular command just type: "help command<CR>".
Example	To check the commands accepted by the reader: help<CR> command list... <CR> For detailed help on the tagop command: help tagop<CR> Command: tagop arg<CR> Description: Get (no arguments) or set the read tag operation.<CR> Arguments: reserved:wp:wl Reserved bank.*<CR> epc:wp:wl EPC bank.*<CR> tid:wp:wl TID bank.*<CR> user:wp:wl User bank.*<CR> autoid Acura Autoid tags.<CR> pa Artefato SJ5511 tags.<CR> none No tagop.<CR> * wp - word pointer, wl - word length, max 32, membank:0:0 entire bank Default: none<CR>
Returns	ok - parameter already has this value.<CR> ok - parameter has been changed.<CR> error - value out of range<CR> error - too many args.<CR>

Public

3.6 Write tag commands

Starting with firmware version 2.1.3 it is possible to write tags using the reader in Autonomous mode with the following commands.

- **writetag** to write the tag EPC id with or without a select filter.
- **writedata** to write data to the tag User and Reserved memory banks.

> **Note:** Multiple tag write commands can be sent at once to the reader, each command will be executed in sequence, to be used when bulk tag writing is necessary.

3.6.1 writetag

Command sintax:

- `writetag newEPC:targetEPC:accessPassword`
- `writetag newEPC:none:accessPassword`

newEpc is the new EPC id to be written, must be a string in hexadecimal format. Size must be word multiple, min. 1 word (2 bytes) (4 digits) max. 16 words (32 bytes) (64 digits).

targetEPC is the current EPC id of the tag you want to change to run a Select filter or 'none' to not perform a filter and write the first tag that reply. Size must be word multiple, min. 1 word (2 bytes) (4 digits) max. 16 words (32 bytes) (64 digits).

accessPassword is the 32b password in hexadecimal format, in case of no password use '00000000'. Fixed size 4 bytes (8 digits).

Reply messages from reader:

```
“ok - wrote <newEpc>”
“error - lock: <msg error>”
“error - unlock: <msg error>”
“error - write: <msg error>”
```

Examples for the writetag command:

a. Write new EPC '123456789012345678901234' of 96 bits/12 bytes/6 words in a tag in the reading field with EPC '3632A80E4EDBBBE244A93E6C', this tag is not password protected (locked) (accesspassword).

   Command to be executed:

   - `writetag 123456789012345678901234:3632A80E4EDBBBE244A93E6C:00000000`

   When the 'epcTarget' field is different from 'none' the reader will execute the write command together with a Select filter so that only the tag with EPC '3632A80E4EDBBBE244A93E6C' responds to the write command.

b. Write new EPC '123456789012345678901234' of 96 bits/12 bytes/6 words in the first tag that reply (recommended to have only 1 tag in the reading field), this tag is not password protected (locked) (accesspassword).

   Command to be executed:

   - `writetag 123456789012345678901234:none:00000000`

c. Write new EPC '123456789012345678901234' of 96 bits/12 bytes/6 words in a tag in the reading field with EPC '3632A80E4EDBBBE244A93E6C', this tag is protected (locked) with password '0E0ABBCC'.

   Command to be executed:

   - `writetag 123456789012345678901234:3632A80E4EDBBBE244A93E6C:0E0ABBCC`

Public

3.6.2 writedata

Command sintax:

- `writedata bank:wordAddr:data:epcTarget:accessPassword`
- `writedata bank:wordAddr:data:none:accessPassword`

bank is the memory bank to written, '**user**' for User and '**res**' for Reserved membank

wordAddr is the start word address, **0..999**

data is a string in hexadecimal format containing the data to be recorded. Size must be word multiple, min. 1 word (2 bytes) (4 digits) max. 16 words (32 bytes) (64 digits).

epcTarget is the current EPC id of the tag you want to change to run a Select filter or 'none' to not perform a filter and write the first tag that reply. Size must be word multiple, min. 1 word (2 bytes) (4 digits) max. 16 words (32 bytes) (64 digits).

accessPassword is the 32b password in hexadecimal format, in case of no password use '00000000'. Fixed size 4 bytes (8 digits).

Examples for the writedata command:

a. Write 43 words/86 bytes to the user membank in a tag within the reading field with EPC '3074257BAF369A40000011D4', this tag is not password protected (locked) (accesspassword).
   Since the maximum size of the 'data' field is 16 words, 3 writedata commands must be executed as below.
   Command to be executed:
   - `writedata user:0:AAAA0000000000000000000000000000000000000000000000000000000000AAAA:3074257BAF369A40000011D4:00000000`
   - `writedata user:16:BBBB0000000000000000000000000000000000000000000000000000000000BBBB:3074257BAF369A40000011D4:00000000`
   - `writedata user:32:CCCC00000000000000000000000000000000CCCC:3074257BAF369A40000011D4:00000000`

   The first run will write AAAA000...000AAAA in the first 16 words in the user membank of the tag with epc id '3074257BAF369A40000011D4'
   The second execution will write BBBB000...000BBBB to the second block of 16 words in the user membank of the tag with epc id '3074257BAF369A40000011D4'
   The third execution will write CCCC000...000CCCC to the remaining block of 11 words in the user bank of the tag with epc id '3074257BAF369A40000011D4'

   > **Note:** The reader has 1024 bytes of buffer for command input, so the 3 lines can be sent at once, the reader will execute them in sequence.

b. Write a new (accesspassword) in a tag with EPC '3074257BAF369A40000011D4' this tag is not password protected (locked) (accesspassword).

Command to be executed:

- `writedata res:2:0E0E0F0F: 3074257BAF369A40000011D4:00000000`

> **Note:** When the reader receives this combination (**bank** 'res', **wordaddr** '2' and **data** with size 2 words) after recording the access password, a Read/Write Lock command is executed in the reserved bank with this new password.

3.7 Switching between Autonomous and Transparent modes

The following options can be used for changing the operation mode of the reader:

3.7.1 Use the readmode command

Example:

> **Note:** When in Transparent mode, the reader does not echo the characters typed in the serial terminal.

1. The reader is in Autonomous mode, and the command **readmode transparent<CR>** is sent. The reader enters Transparent mode of operation.
2. The reader is in Transparent mode and the command **$ off<CR>** is sent. The reader enters Autonomous mode with hid interface.

3.7.2 Use the Caps Lock key (Windows and Linux only)

Example:

> **Important: Using the Caps Lock shortcut for changing the operating mode will only works if the readers serial port is not connected (not being used by any software).**

1. Close any serial communication with the reader.
2. To switch between one operating mode and another, **quickly click the Caps Lock key five times**, and check the readers LEDs to see the change.

   > **Note:** The commands **enablecapslock** and **enablereadmodekey** enables or disables this function. To use it, **enablecapslock** and **enablereadmodekey** must be on.

Public

3.8 Read rate of the same tag

To control the read rate, it is recommended to use the **tagtimeout** command.

The read rate of the same tag can be controlled through the following commands:

- **gen2session**
- **gen2target**
- **rfon**
- **rfoff**
- **tagtimeout**

The listed commands, except for **tagtimeout**, change the parameters of the UHF EPC Gen2 air protocol, and influence the communication between the tag and reader.

Combinations of values for the Session and Target of the tag reflect how often a tag that has already responded to an inventory will respond again.

Examples:

a. I would like to read the EPC id of a tag only once, and I do not want the same EPC id to be repeated while the tag is in the reading field.
   Configure it as follows:
   gen2session s2<CR>
   gen2target a<CR>
b. I would like to read the same tag repeatedly while it is in the reading field.
   Configure it as follows:
   gen2session s0<CR>
   gen2target ab<CR>
   In this case, what will determine the read repetition rate are the **rfon** and **rfoff** commands, meaning the total time will dictate the repetition rate.
   The highest possible read repetition rate is every 50ms:
   rfon 50<CR>
   rfoff 0<CR>

The read rate of the same tag can also be tweaked through the **tagtimeout** command. This command acts as a filter for tag reading, unlike the EPC Gen2 commands (Session and Target) mentioned above, which are part of the air protocol definition.

Public

The **tagtimeout** command ranges from 0 to 300000 ms, up to 5 minutes. During this time, the same tag EPC id will not be reported by the reader again, even if it remains in the reading field.

Example:

- I would like to read the tag and only be able to read it again after 20 seconds. Configure the reader as follows:
  tagtimeout 20000<CR>
  gen2session s0<CR>
  gen2target ab<CR>

> **Note:** An important point is that depending on the Session and Target configurations in the reader, the tag response time can override the time defined in **tagtimeout**.

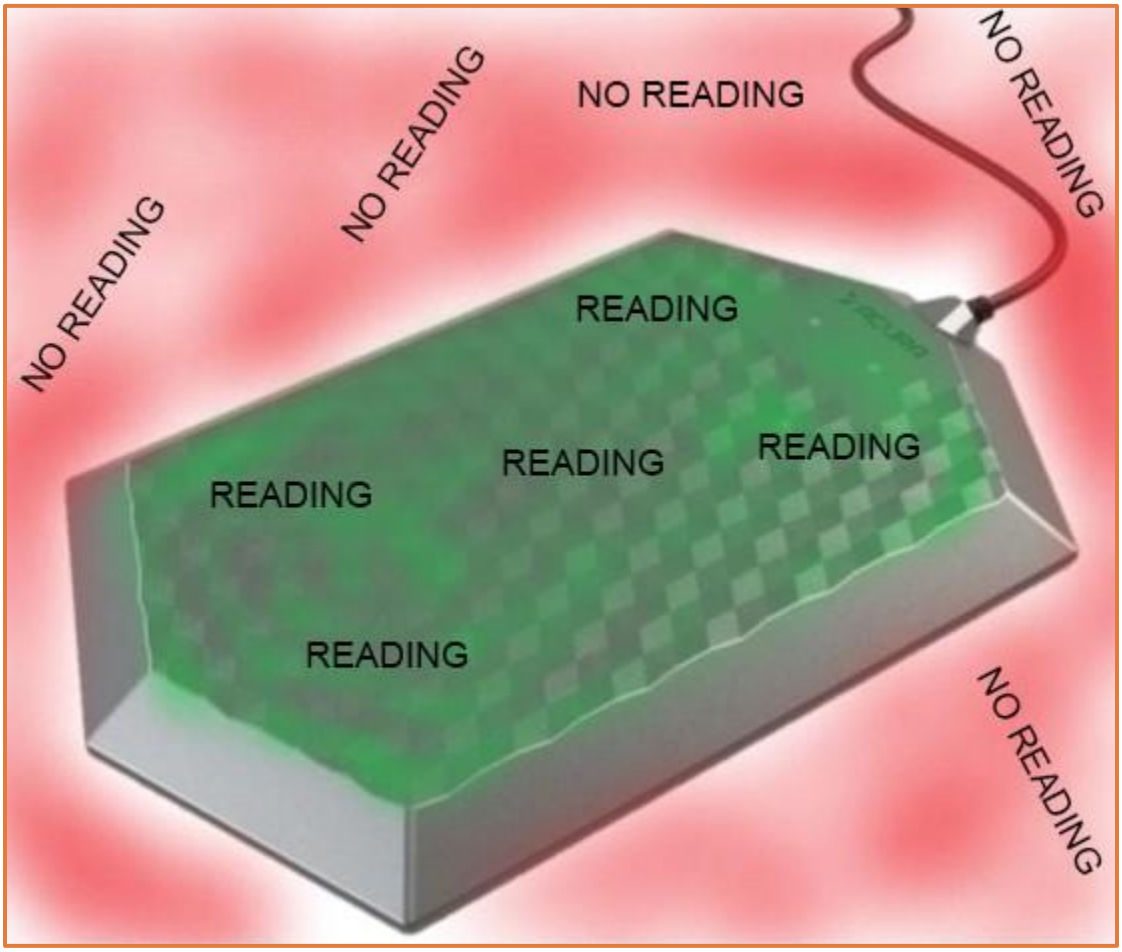
Example:

- I would like to configure the tagtimeout to 1 second, but I will keep the UHF Gen2 configuration as Session 2.
  tagtimeout 1000<CR>
  gen2session s2<CR>
  gen2target a<CR>

> **Important: The tag read rate can be much faster than the rate at which results are sent through keyboard emulation (HID). If a high reading rate is chosen, and the chosen interface is keyboard emulation, the reader will store the reading result and continue sending it even after the tag is removed from the reading field.**

3.9 Hexpad reading and non-reading area

The reading and non-reading areas for both the HexaPad-10 USB and HexaPad-10 BC models:



To obtain a reading field as close as possible to the image above, the RSSI reading filter should be used in conjunction with the reading power setting. With these adjustments, along with the integrated near-field antenna, the rejection of side readings greatly increases.

3.9.1 How to measure the RSSI cutoff value

The cutoff value for the tag signal reading level should be obtained in a practical manner, with the tags that will be used in the application and conducting reading tests to verify the RSSI value reported on the sides and above the reader.

> **Note:** To obtain the RSSI value for each reading, use the **reportrssi on<CR>** command. To configure the reader to report readings only above a specific RSSI value, use the **rssifilter xx<CR>** command, where **xx** is the desired value.

> **Note:** To facilitate the execution of RSSI level readings, the read repetition rate for the tags can be increased as described in **Reading rate of the same tag**.

Public

3.10 Use cases - Autonomous mode

The following are examples of using and applying the HexaPad-10 family readers, with details of the configuration commands used.

3.10.1 Read tags for 30 seconds and then remain idle

To read tags during a specific time duration, the readtag command can be used with time duration as value or the Capslock (double hit) can be used and the value of readtagtimeout command will be used:

- **readtag 30000**
- **readtagtimeout 30000**

The value of **readtagtimeout** will be used only when the Capslock key is used to toggle read tag state, the **enablecapslock** and **enablereadtagkey** commands must be on.

3.10.2 Checkout – GS1/EAN standard

For applications that use UHF EPC Gen2 tags encoded according to the GS1 EPC Tag Data Standards (TDS) 1.11 standard, the HexaPad-10 family readers can be used and supports the following tag encodings:

- **sgtin-96**
- **sgtin-198**

Is an encoding scheme that stands for Serialized Global Trade Identification Number and is sometimes called a serialized GTIN. The 96 in the name refers to the number of bits in the final EPC binary form.

If you need alphanumeric characters in the serial numbers, SGTIN-198 must be used. This means that you must use RFID tags with higher capacity in the EPC memory, than if you use the SGTIN-96.

The results of reading the encoded tag can be sent in the following formats:

- **GS1 element string**, example "(01)03608449920322(21)1540341";
- **EPC Tag URI**, example "urn:epc:tag:sgtin-96:1.360844.0992032.1540341";
- **EPC Pure Identity URI**, example "urn:epc:id:sgtin:360844.0992032.1540341";
- **GTIN13 without the serial number**, example "3608449920322";
- **GTIN13 + serial number**, example "3608449920322 1540341";

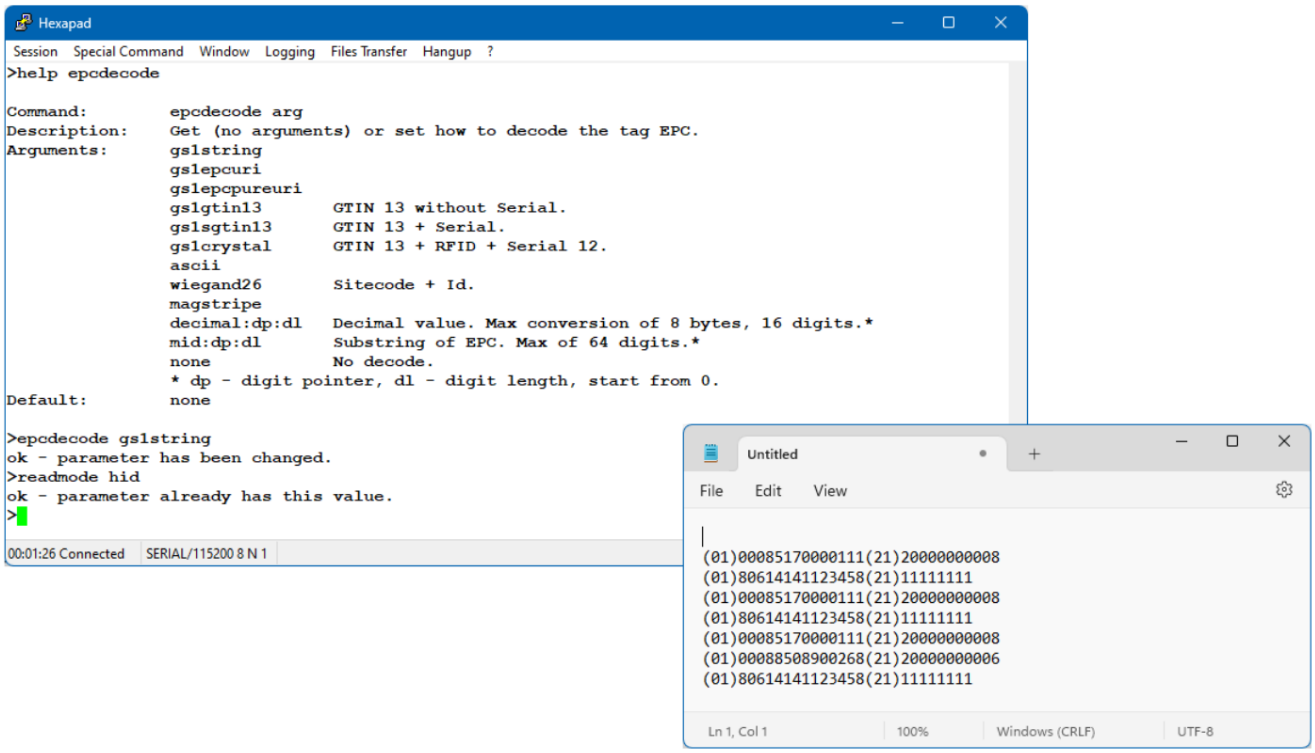
Public

3.10.3 Read tag encoded in SGTIN-96/198 and show result in GS1 element string format

Configure the reader as follows:

epcdecode gs1string<CR>

(to decode sgtin-96 or sgtin-198 and send in GS1 element string format)

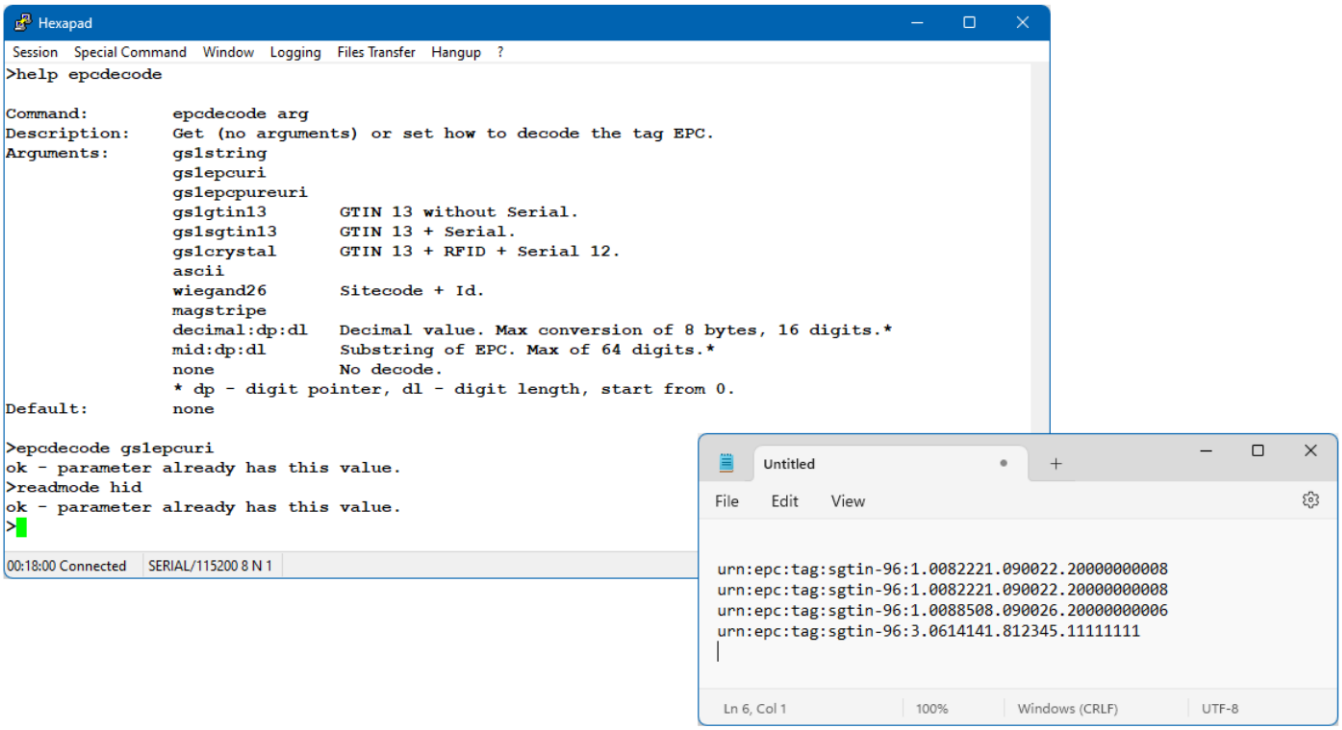


3.10.4 Read tag encoded in SGTIN-96/198 and show result in EPC Tag URI format

Configure the reader as follows:

epcdecode gs1epcuri<CR>

(to decode sgtin-96 or sgtin-198 and send in EPC Tag URI format)



For the other formats use the **epcdecode** command with the equivalent format that can be checked in **Configuration commands > epcdecode**.

> **Note:** The keyboard/language layout configured in the Operating System can change the "**:**" character of the result string of the epcdecode type **gs1epcuri** and **gs1pureepcuri**.

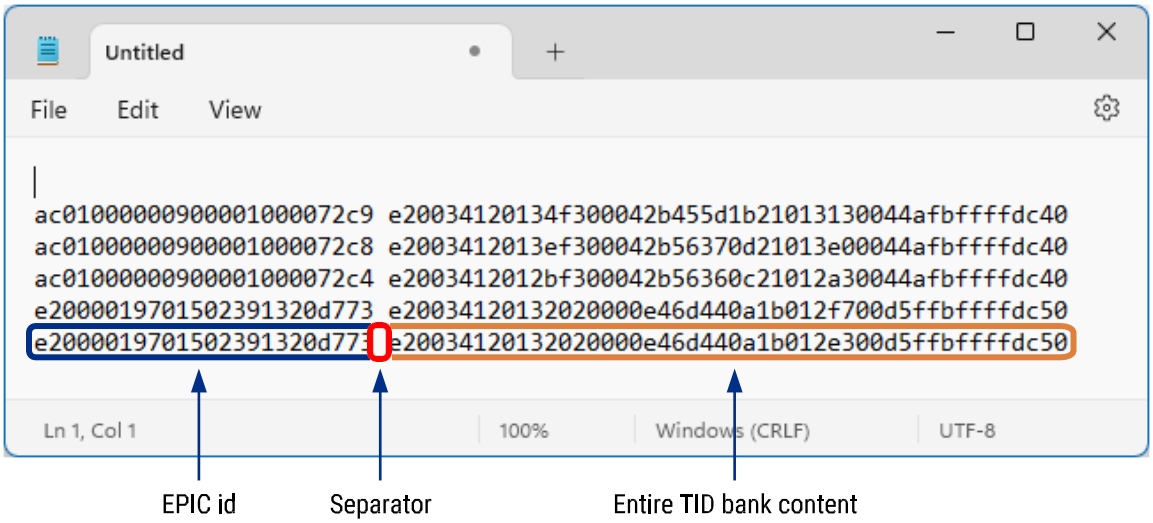
3.10.5 Read TID memory bank and EPC id

The HexaPad-10 readers are able to read any memory bank of the EPC Gen2 tag, (reserved, EPC, TID and user memory). Configure using the **tagop** command in whichever database you desire to read, along with the initial position and the size of the data (in word) that you want to read.

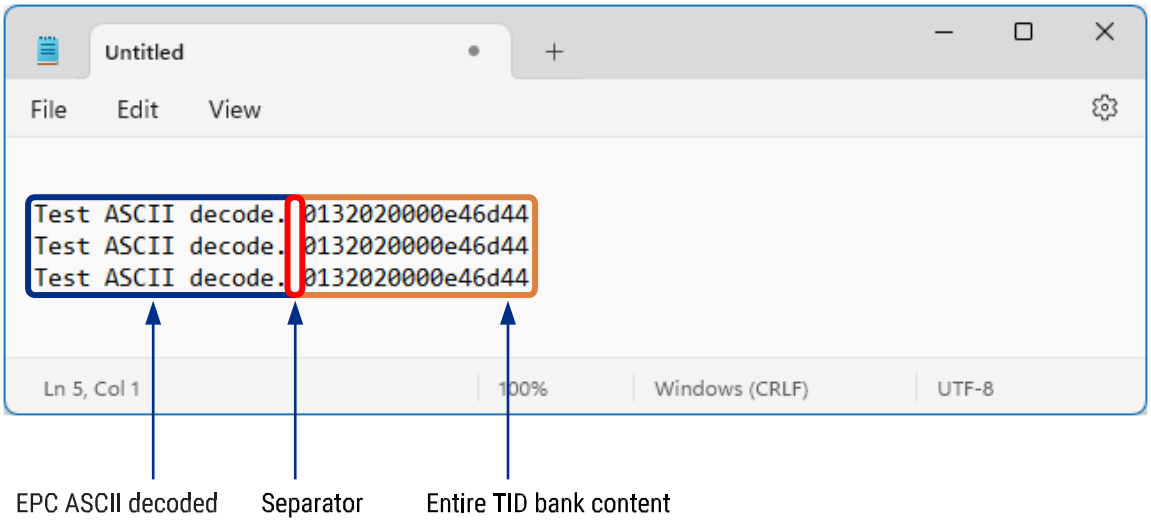
The syntax of the **tagop** command is explained in detail in the **tagop** section.

For example:

1. Read the entire TID bank of the tag plus the EPC id in hexa:
   tagop tid:0:0<CR>
   epcdecode none<CR>
   Example of reading two tags with different sizes from the TID bank:



2. Read from the TID bank 4 words skipping the first 2 words and decoding the EPC id by the ASCII table:
   tagop tid:2:4<CR>
   epcdecode ascii<CR>
   Example of reading two different tags:



Public

In addition to TID bank, RESERVED, USER, and EPC banks can also be read in conjunction with the EPC tag id.

> **Note:**
> - The count unit in the EPC Gen2 standard is **word**, which is equivalent to 2 bytes or 16 bits.
> - For the RESERVED bank, if the tag has an access password other than 0x00000000, the tags password must be configured in the reader using the **gen2accesspassword** command.
> - More details of commands in **Configuration commands**.

3.11 Configure the reader using a QR code

The HexaPad-10 BC (Barcode) can be configured using a QR Code containing a list of the ASCII commands. This feature is convenient, as it allows the reader to be configured in the field, using, for example, a mobile phone.

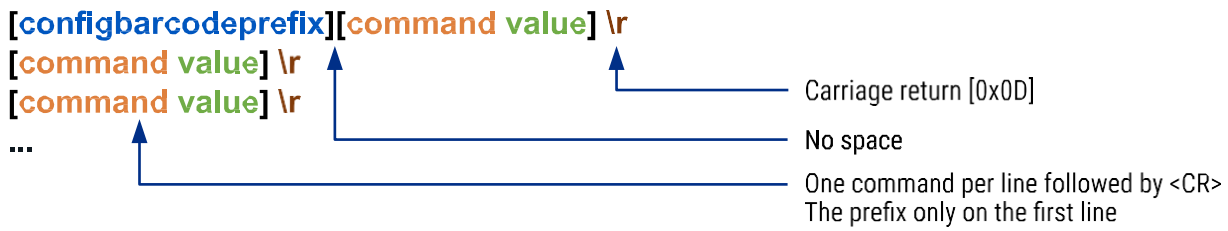
To be able to use this feature, the following set up must be made:

configbybarcode on<CR>

configbarcodeprefix [char value]<CR>

readbarcode on<CR>

The QR code must have a specific syntax, as outlined below:



Example with prefix default value:

*readmode serial\r
readpower 21\r

Generated QR code:

> **Note:** The QR code generator engine must be able to evaluate/generate the special character '**\r**' carriage return.

Tested QR code generator tools:

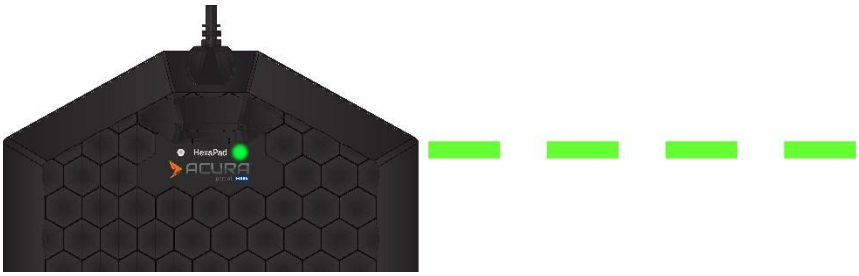
- Mobile phone Android:
  https://play.google.com/store/apps/details?id=com.teacapps.barcodescanner
  (No need to type '\r' at end of each line)
- Online generator with:
  https://barcode.tec-it.com/en/QRCode?data=*readmode%20serial%5Cr%0Areadpower%2021%5Cr
  (Need to type '\r' at end of each line)

Public

3.12 Reader - Transparent mode

In Transparent mode, the reader starts to respond through the serial to a software developed using the Mercury API, which is a library package for C#/.Net, Java and C languages. With this development package, it is possible to develop software that can have full control of the reader, for reading and writing UHF EPC Gen2 tags (ISO 18000-6C) and UHF EPC Gen2 V2 (ISO 18000-63).

Visual feedback from the reader when in Transparent mode:

	Green LED flashes slowly, and red LED is turned off.

The default operation mode of the reader is the Autonomous mode, to change to Transparent mode please refer to **Switching between Autonomous and Transparent modes**.

3.12.1 readmode

The readers red LED and buzzer are controlled by the users software through the manipulation of two available GPIOs in MercuryAPI. For more details, refer to **Controlling the LED and Beep**.

Communication between the user software and the reader is conducted through a serial port created by the operating system when the reader is connected via USB. To identify this port, please refer to section **Turning on the reader**.

> **Note:** In Transparent mode, the barcode reader interface is HID keyboard.

3.12.2 Software programming and development

For software development, it is necessary to use the MercuryAPI SDK.

The latest recommended version for development can be found at:

https://drive.google.com/drive/folders/1AOXUCQPtyAfcvyUeZf2gsHyLh3vfAEsf?usp=drive_link

In the API file package, there are numerous examples and source code in all supported languages. This development package has no additional cost and is open-source.

The complete API documentation is also available at the same link.

Public

3.12.3 Demo software – Universal Reader Assistant (URA)

The reader in Transparent mode can be tested using the Jadak Universal Reader Assistant software. The software download and its manual are available at:

https://drive.google.com/drive/folders/1SsEdqQSkHQI_n-WqaQb7Kx8oi2INgzj2?usp=drive_link

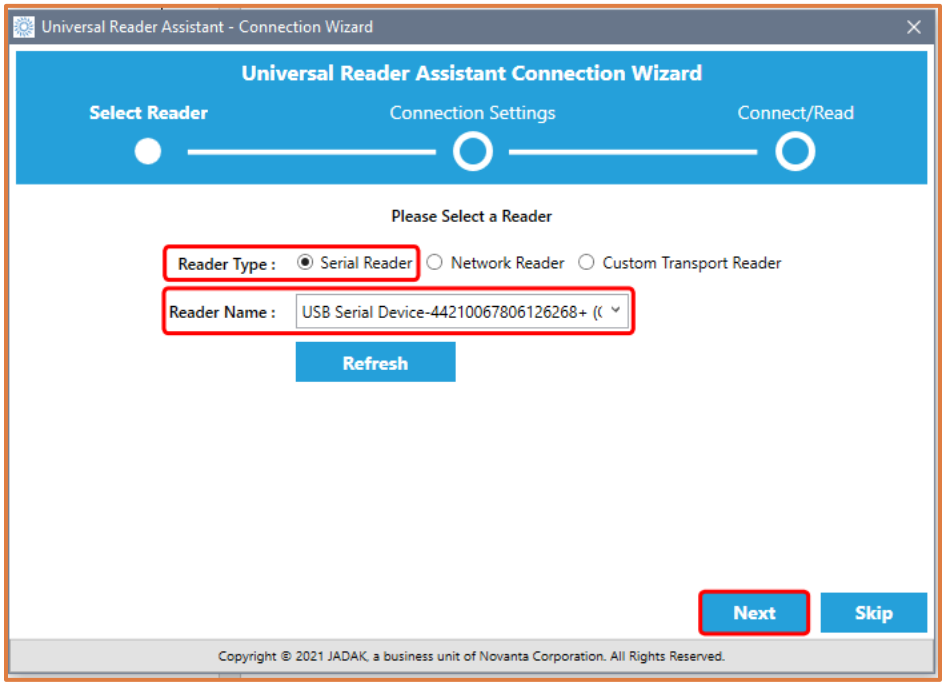
3.12.4 Demo software instructions

> **Note:** Ensure that the reader is set to Transparent mode. Please refer to **Reader - Transparent mode**.

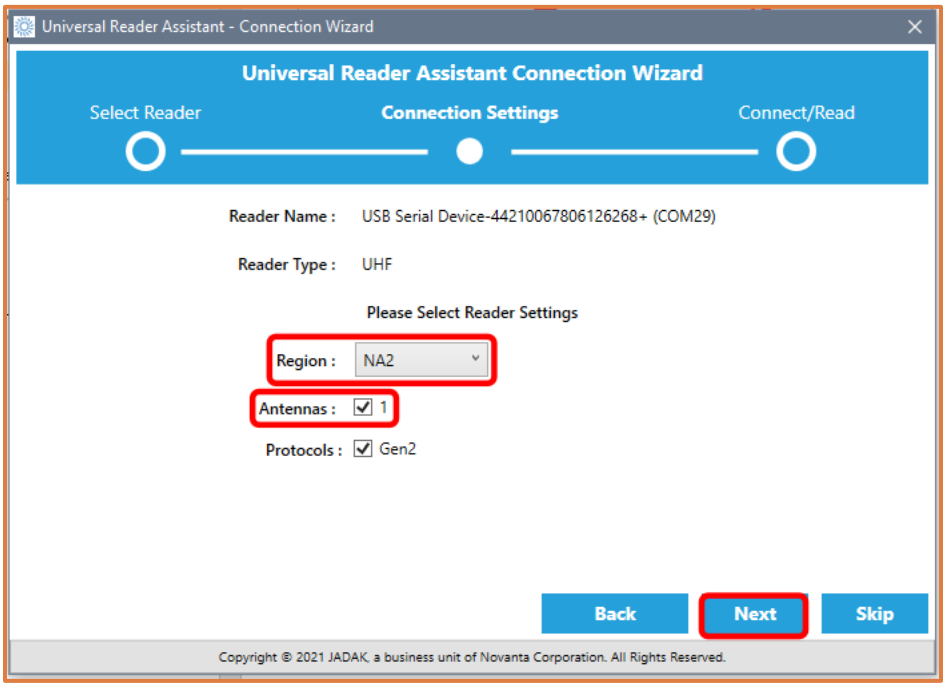
1. Launch the Universal Reader Assistant.

   > **Note:** The software will automatically identify the device connected to your computer.

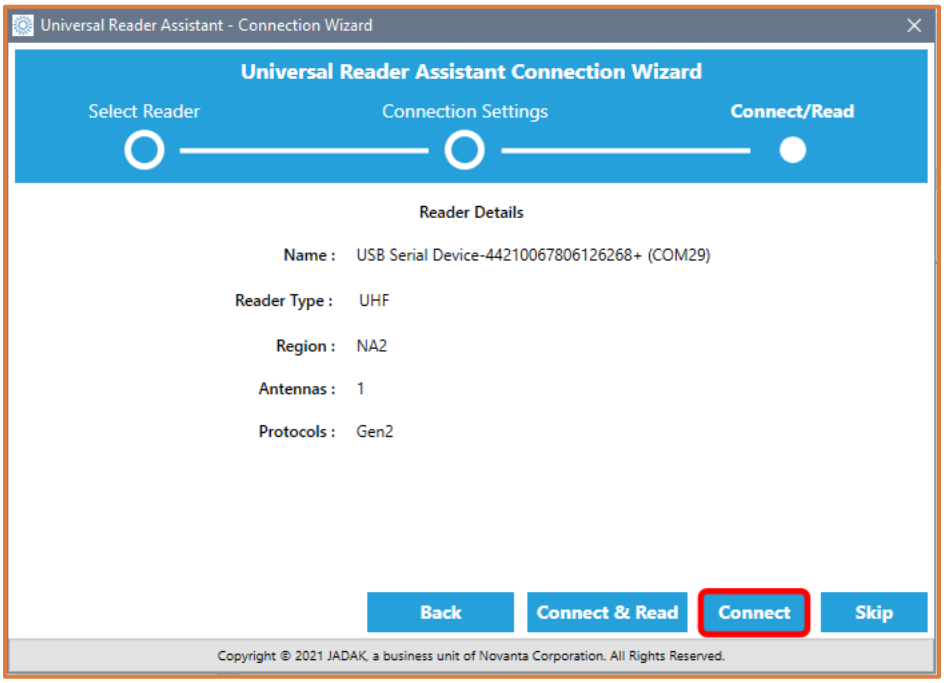
2. Click **Next**.



3. Select **Region** and **Antennas**.
4. Click **Next**.

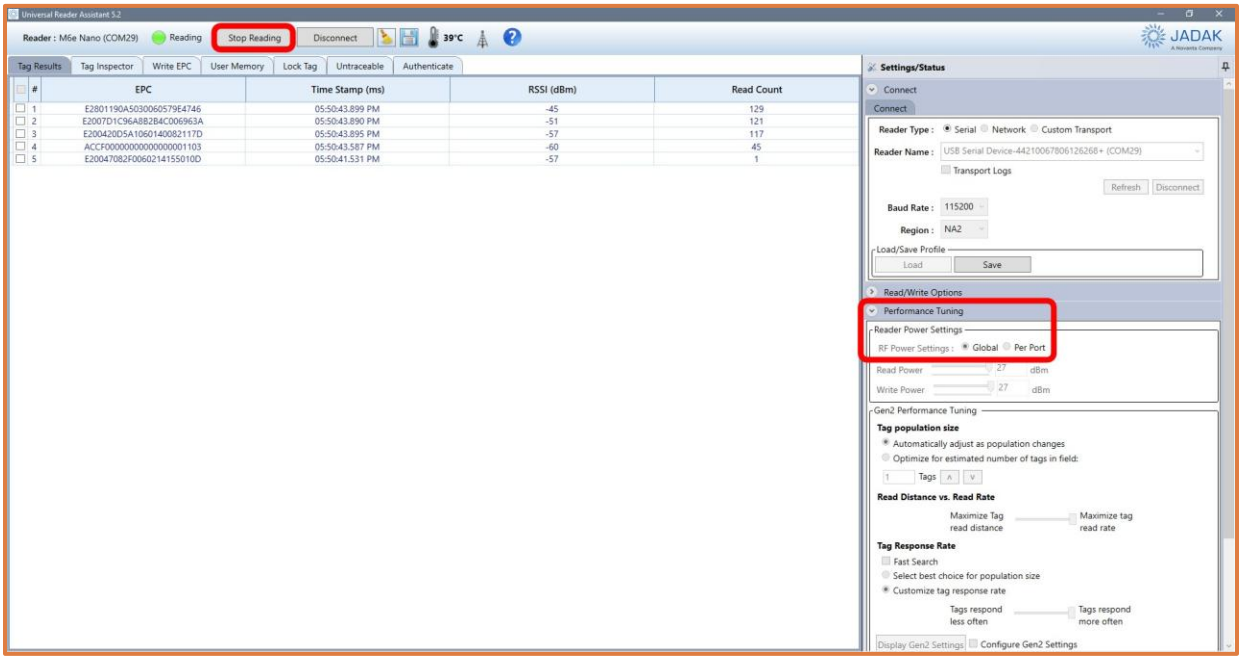


5. Click **Connect**.



Public

6. In **Performance Tuning** select the **RF Power Setting**.
7. Click **Read** to start the data readings.



3.13 Controlling the LED and Beep

In Transparent mode, control of the red LED and the beep should be managed by the users software using the MercuryAPI.

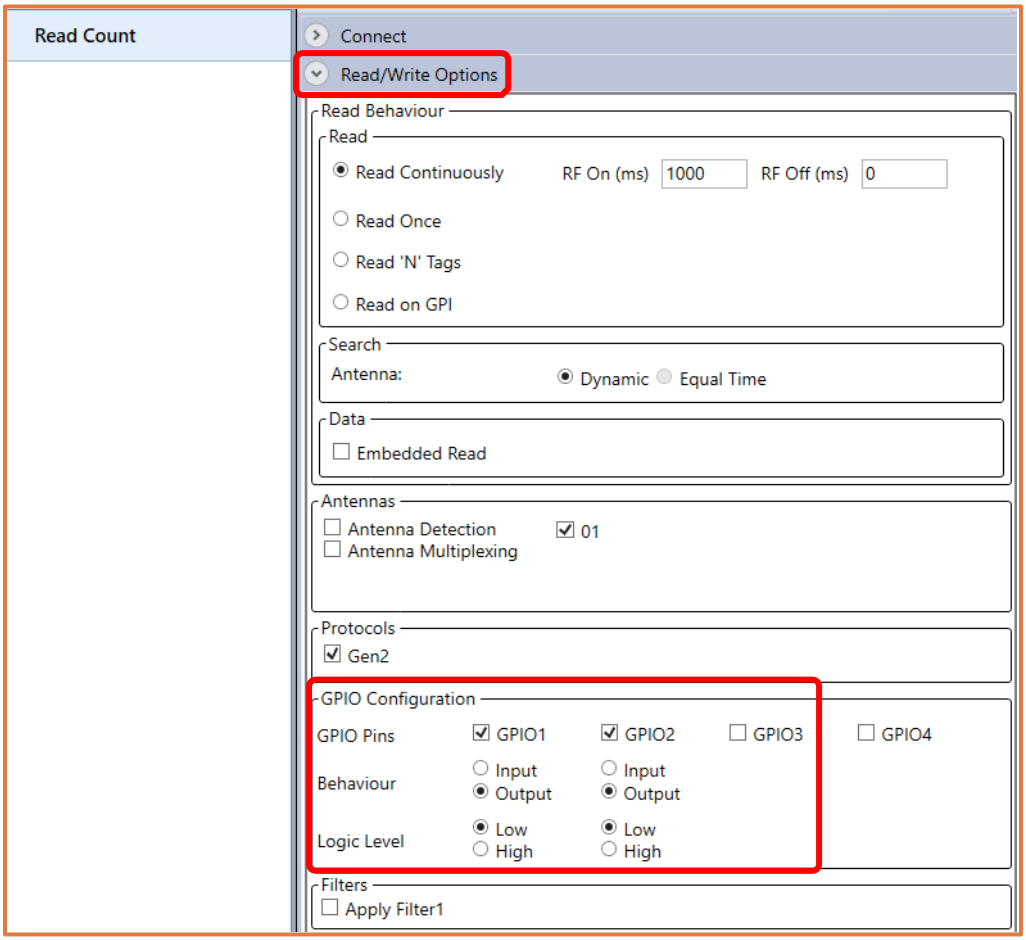
> **Note:** In the MercuryAPI file package for all languages, there is an example for manipulating the readers GPIO called "**gpiocommands.**"

The usage map for the respective GPIOs:

Output	Type feedback	Drive (on)
GPIO1	Red LED	High
GPIO2	Beep	High

Using the URA for controlling the red LED and beep of the reader in Transparent mode:

1. Expand the **Read/Write Options** to access the configuration menu.
2. In **GPIO Configuration** check **GPIO Pins GPIO1** (for the LED) and **GPIO2** (for the buzzer).
3. In **Behavior** set to **Output**.
4. The **Logic Level** when set to **Low**, causes the LED or buzzer to **turn off**. Setting the **Logic Level** to **High**, causes the LED or buzzer to **turn on**.

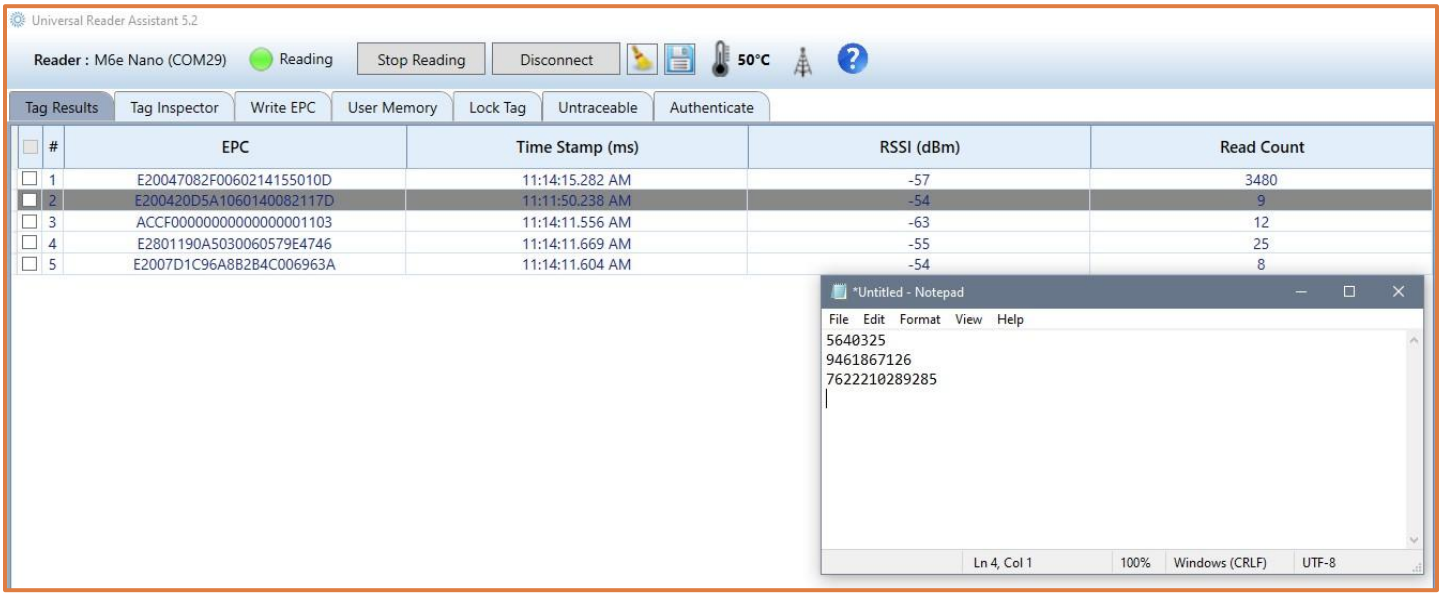


Public

3.14 Barcode scanning in Transparent mode

Even though the RFID tag reader is configured to operate in Transparent mode, the barcode reader remains in Autonomous mode, where it scans barcodes and sends the results through keyboard emulation (HID).

The reader performs RFID tag scans in the URA software (Transparent mode). When opening a notepad, it is also possible to scan barcode data, for example:



Revision history

Date	Description	Revision
December 2025	Updates for latest firmware version.	A.2
February 2025	Contact email address changed.	A.1
November 2023	Initial release.	A.0

Tel.: 11 3028-4600 | E-mail: acura.support@hidglobal.com | Site: www.ACURA.com.br

© 2025 HID Global Corporation/ASSA
ABLOY AB. All rights reserved.
PLT-07484, Rev. A.2

Part of ASSA ABLOY