**Table of Contents:**


- 1. Introduction
  - 1.1 What is DDS75-NB NB-IoT Distance Detection Sensor
  - 1.2 Features
  - 1.3 Specification
  - 1.4 Rated environmental conditions
  - 1.5 Effective measurement range Reference beam pattern
  - 1.6 Applications
  - 1.7 Sleep mode and working mode
  - 1.8 Button & LEDs
  - 1.9 BLE connection
  - 1.10 Pin Definitions , Switch & SIM Direction
    - 1.10.1 Jumper JP2
    - 1.10.2 BOOT MODE / SW1
    - 1.10.3 Reset Button
    - 1.10.4 SIM Card Direction
  - 1.11 Mechanical
- 2. Use DDS75-NB to communicate with IoT Server
  - 2.1 Send data to IoT server via NB-IoT network
  - 2.2 Payload Types
    - 2.2.1 General Json Format(Type=5)
    - 2.2.2 HEX format Payload(Type=0)
    - 2.2.3 ThingsBoard Payload(Type=3)


SE NO
Direktronik AB | Box 234 , 149 23 Nynäshamn | Besöksadress Konsul Johnsons väg 15 149 45 Nynäshamn
Telefon 08 52 400 700 | Fax | Epost info@direktronik.se | Org.nr 556281-9663 | Bankgiro 922-0179

- 2.2.4 ThingSpeak Payload(Type=1)
- 2.3 Test Uplink and Change Update Interval
- 2.4 Multi-Samplings and One uplink
- 2.5 Trggier an uplink by external interrupt
- 2.6 Distance Alarm
- 2.7 Clock logging (Since firmware version v1.2.1)
- 2.8 Example Query saved historical records
- 2.9 Uplink log query
- 2.10 Scheduled domain name resolution
- 2.11 Set the QoS level
- 2.12  Set CoAP option
- 2.13 Set the downlink debugging mode(Since firmware v1.3.0)
- 2.14 Domain name resolution settings(Since firmware v1.3.0)

- 3. Configure DDS75-NB
  - 3.1 Configure Methods
  - 3.2  Serial Access Password
  - 3.3 AT Commands Set
- 4. Battery & Power Consumption
- 5. Firmware update
- 6. FAQ
  - 6.1 How can I access t BC660K-GL AT Commands?
  - 6.2 Can I use DDS75-NB in condensation environment?
  - 6.3 How to configure the certificate?
- 7. Trouble Shooting
  - 7.1 Why does the sensor reading show 0 or "No sensor"
  - 7.2 Abnormal readings The gap between multiple readings is too large or the gap between the readings and the actual value is too large
- 8. Order Info
- 9. Packing Info
- 10. Support


# 1. Introduction

## 1.1 What is DDS75-NB NB-IoT Distance Detection Sensor

The Dragino DDS75-NB is a **NB-IoT Distance Detection Sensor** for Internet of Things solution. It is used to measure the distance between the sensor and a flat object. The distance detection sensor is a module that uses **ultrasonic sensing technology** for distance measurement, and **temperature compensation** is performed internally to improve the reliability of data. The DDS75-NB can be applied to scenarios such as horizontal distance measurement, liquid level measurement, parking management system, object proximity and presence detection, intelligent trash can management system, robot obstacle avoidance, automatic control, sewer, bottom water level monitoring, etc.

It detects the distance **between the measured object and the sensor**, and and send IoT platform via NB-IoT network.

DDS75-NB supports different uplink methods including **MQTT, MQTTs, UDP & TCP** for different application requirement, and support uplinks to various IoT Servers.

DDS75-NB **supports BLE configure** and **OTA update** which make user easy to use.

DDS75-NB is powered by **8500mAh Li-SOCI2 battery**, it is designed for long-term use up to several years.

DDS75-NB has optional built-in SIM card and default IoT server connection version. Which makes it works with simple configuration.

## 1.2 Features

- NB-IoT Bands: B1/B2/B3/B4/B5/B8/B12/B13/B17/B18/B19/B20/B25/B28/B66/B70/B85 @H-FDD
- Ultra-low power consumption
- Distance Detection by Ultrasonic technology
- Flat object range 280mm - 7500mm
- Accuracy: ±(1cm+S*0.3%) (S: Distance)
- Measure Angle: 40°
- Multiply Sampling and one uplink
- Support Bluetooth v5.1 remote configure and update firmware
- Uplink on periodically
- Downlink to change configure
- IP66 Waterproof Enclosure
- 8500mAh Battery for long term use
- Nano SIM card slot for NB-IoT SIM

## 1.3 Specification

**Common DC Characteristics:**

- Supply Voltage: 2.5v ~ 3.6v
- Operating Temperature: -40 ~ 85°C

**NB-IoT Spec:**

**NB-IoT Module: BC660K-GL**

**Support Bands:**

- B1 @H-FDD: 2100MHz
- B2 @H-FDD: 1900MHz
- B3 @H-FDD: 1800MHz
- B4 @H-FDD: 2100MHz
- B5 @H-FDD: 860MHz
- B8 @H-FDD: 900MHz
- B12 @H-FDD: 720MHz
- B13 @H-FDD: 740MHz
- B17 @H-FDD: 730MHz

- B18 @H-FDD: 870MHz
- B19 @H-FDD: 870MHz
- B20 @H-FDD: 790MHz
- B25 @H-FDD: 1900MHz
- B28 @H-FDD: 750MHz
- B66 @H-FDD: 2000MHz
- B70 @H-FDD: 2000MHz
- B85 @H-FDD: 700MHz

**Battery:**

- Li/SOCI2 un-chargeable battery
- Capacity: 8500mAh
- Self Discharge: <1% / Year @ 25°C
- Max continuously current: 130mA
- Max boost current: 2A, 1 second

**Power Consumption**

- STOP Mode: 10uA @ 3.3v
- Max transmit power: 350mA@3.3v

## 1.4 Rated environmental conditions

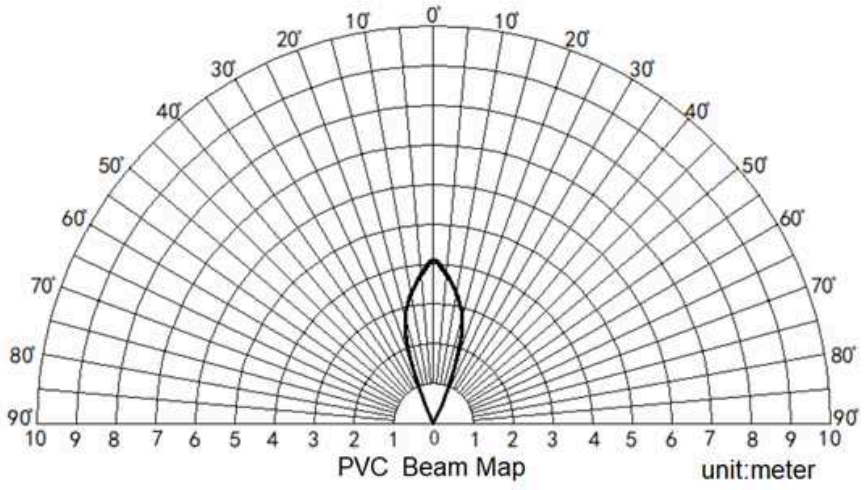
| Item | Minimum value | Typical value | Maximum value | Unit | Remarks |
|---|---|---|---|---|---|
| Storage temperature | -25 | 25 | 80 | ℃ | |
| Storage humidity | | 65% | 90% | RH | (1) |
| Operating temperature | -15 | 25 | 60 | ℃ | |
| Working humidity | | 65% | 80% | RH | (1) |

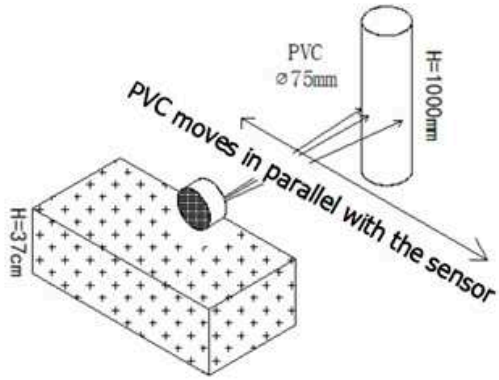
**Remarks: (1) a. When the ambient temperature is 0-39 ℃, the maximum humidity is 90% (non-condensing);**

**b. When the ambient temperature is 40-50 ℃, the highest humidity is the highest humidity in the natural world at the current temperature (no condensation)**

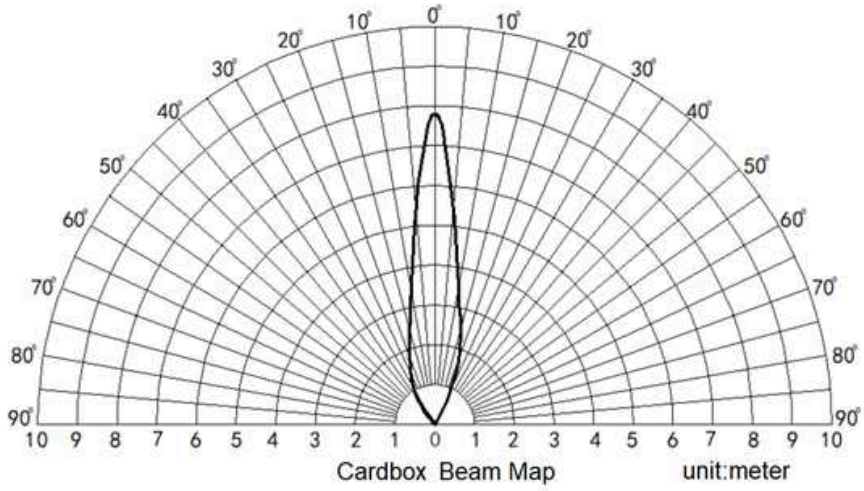
## 1.5 Effective measurement range Reference beam pattern

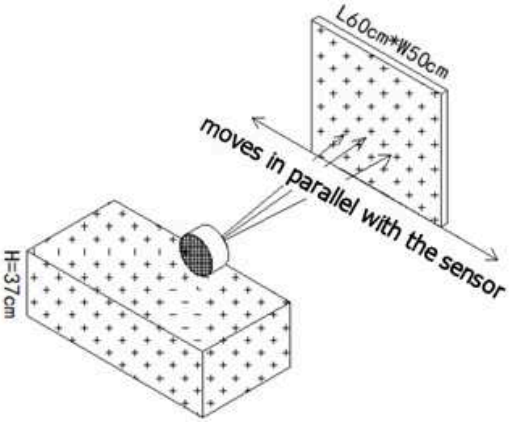
**1. The tested object is a white cylindrical tube made of PVC, with a height of 100cm and a diameter of 7.5cm.**




**2. The object to be tested is a "corrugated cardboard box" perpendicular to the central axis of 0 °, and the length * width is 60cm * 50cm.**




## 1.6 Applications

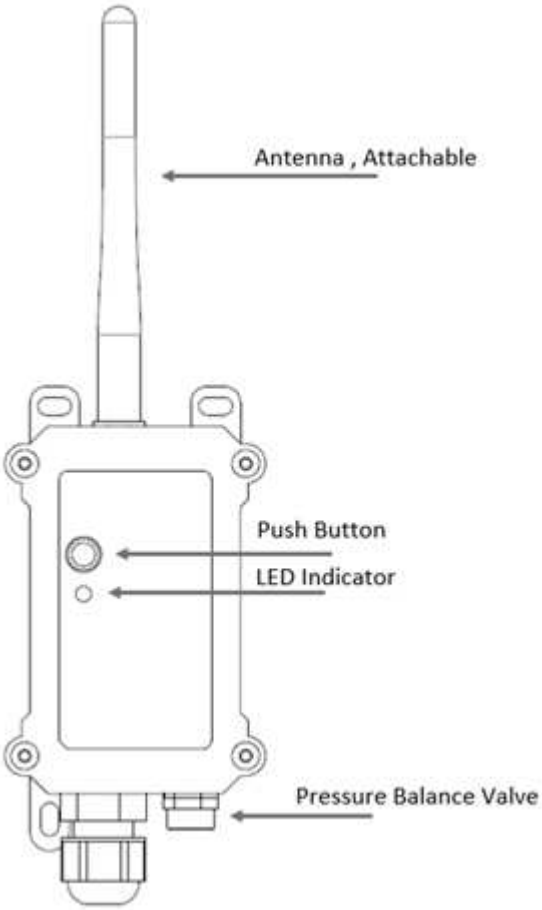
- Horizontal distance measurement
- Liquid level measurement
- Parking management system
- Object proximity and presence detection
- Intelligent trash can management system
- Robot obstacle avoidance
- Automatic control
- Sewer
- Bottom water level monitoring

## 1.7 Sleep mode and working mode

**Deep Sleep Mode:** Sensor doesn't have any NB-IoT activate. This mode is used for storage and shipping to save battery life.

**Working Mode:** In this mode, Sensor will work as NB-IoT Sensor to Join NB-IoT network and send out sensor data to server. Between each sampling/tx/rx periodically, sensor will be in IDLE mode), in IDLE mode, sensor has the same power consumption as Deep Sleep mode.

## 1.8 Button & LEDs




| Behavior on ACT | Function | Action |
|---|---|---|
| 1~3s | Send an uplink | If sensor has already attached to NB-IoT network, sensor will send an uplink packet, **blue led** will blink once.<br>Meanwhile, BLE module will be active and user can connect via BLE to configure device. |
| >3s | Active Device | **Green led** will fast blink 5 times, device will enter **OTA mode** for 3 seconds. And then start to attach NB-IoT network.<br>Once sensor is active, BLE module will be active and user can connect via BLE to configure device, no matter if device attach NB-IoT network or not. |

| | | |
|---|---|---|
| x5 | Deactivate Device | **Red led** will solid on for 5 seconds. Means device is in Deep Sleep Mode. |

**Note: When the device is executing a program, the buttons may become invalid. It is best to press the buttons after the device has completed the program execution.**

## 1.9 BLE connection

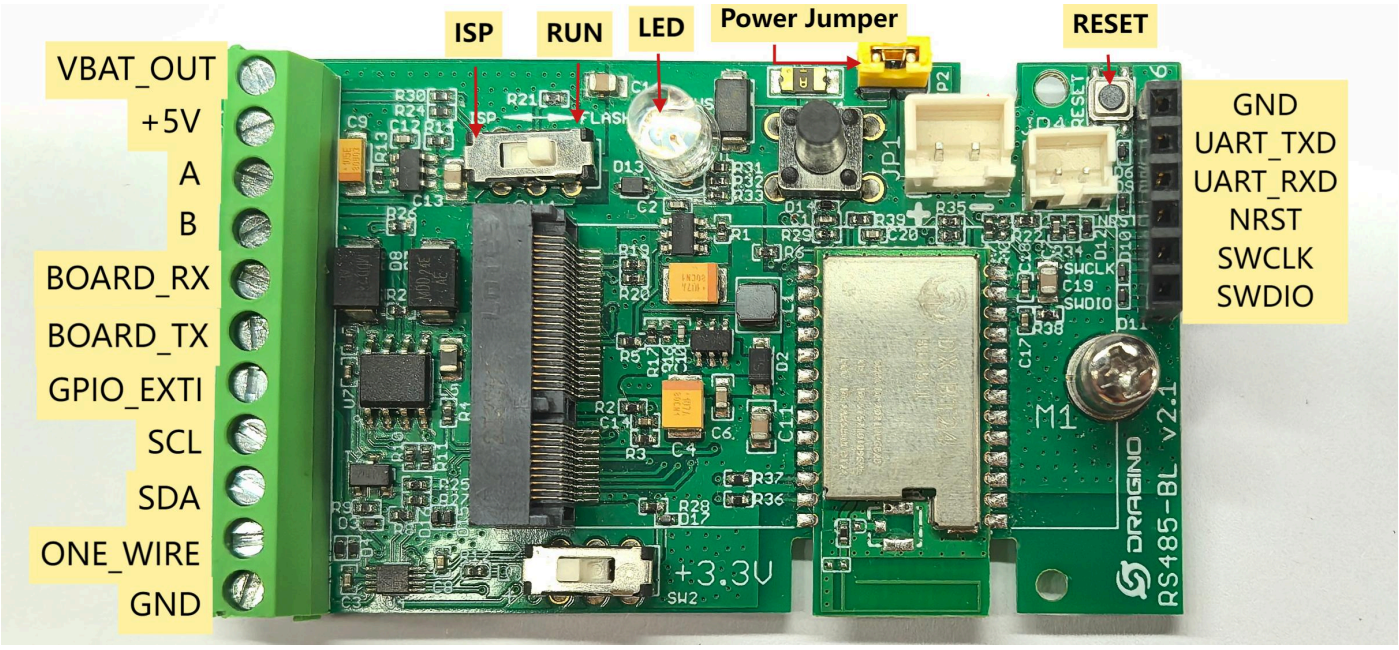
DDS75-NB support BLE remote configure and firmware update.

BLE can be used to configure the parameter of sensor or see the console output from sensor. BLE will be only activate on below case:

- Press button to send an uplink
- Press button to active device.
- Device Power on or reset.

If there is no activity connection on BLE in 60 seconds, sensor will shut down BLE module to enter low power mode.

## 1.10 Pin Definitions , Switch & SIM Direction




### 1.10.1 Jumper JP2

Power on Device when put this jumper.

### 1.10.2 BOOT MODE / SW1

**1) ISP**: upgrade mode, device won't have any signal in this mode. but ready for upgrade firmware. LED won't work. Firmware won't run.

**2) Flash**: work mode, device starts to work and send out console output for further debug

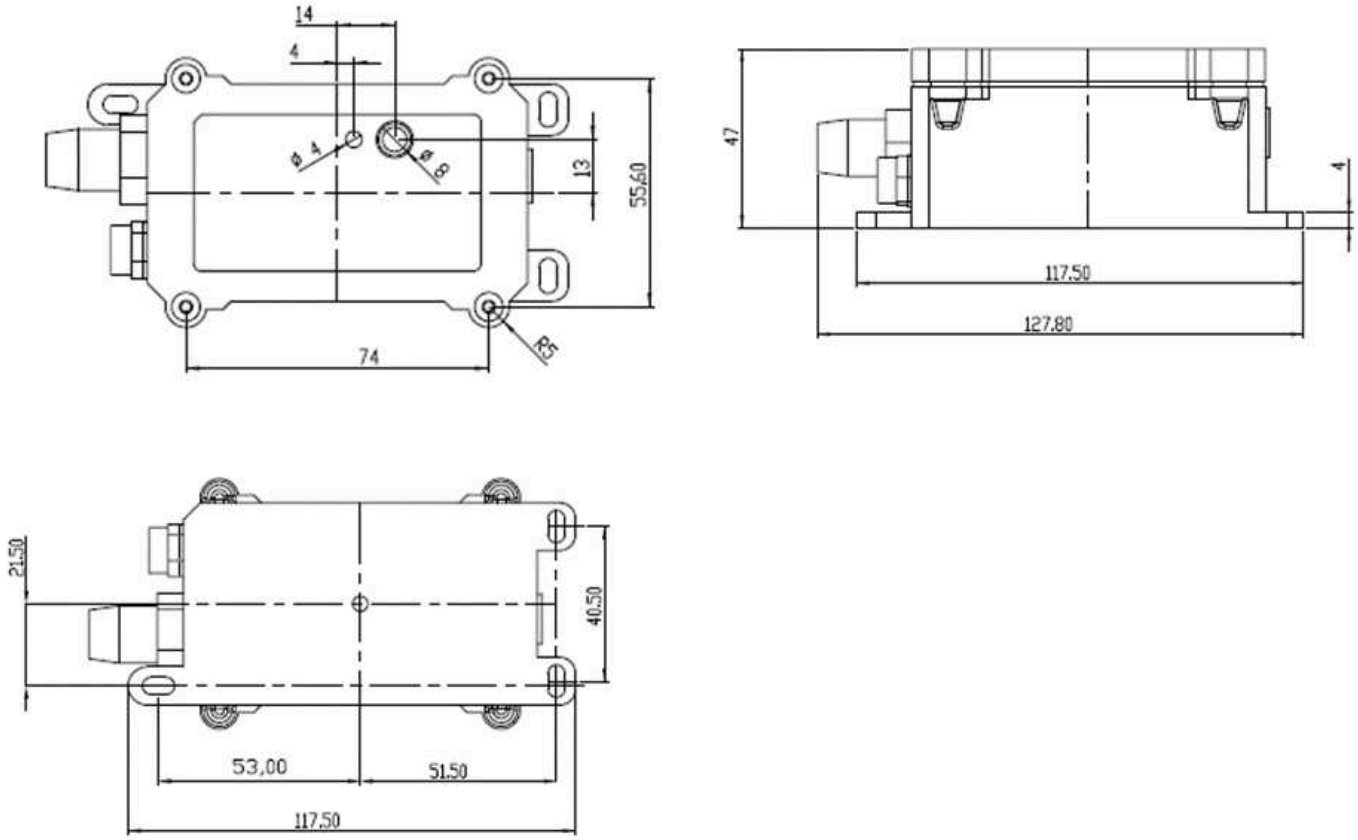
### 1.10.3 Reset Button

Press to reboot the device.

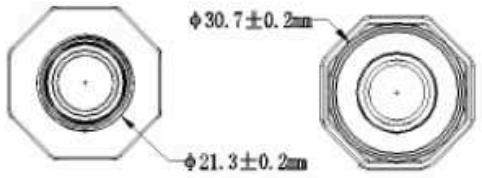
### 1.10.4 SIM Card Direction

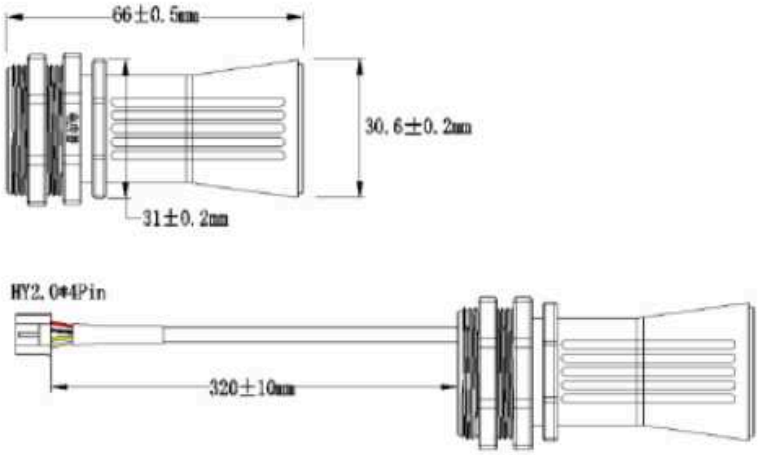
See this link. [How to insert SIM Card](#).

## 1.11 Mechanical




**Probe Mechanical:**




# 2. Use DDS75-NB to communicate with IoT Server

## 2.1 Send data to IoT server via NB-IoT network

The DDS75-NB is equipped with a NB-IoT module, the pre-loaded firmware in DDS75-NB will get environment data from sensors and send the value to local NB-IoT network via the NB-IoT module.  The NB-IoT network will forward this value to IoT server via the protocol defined by DDS75-NB.

Below shows the network structure:

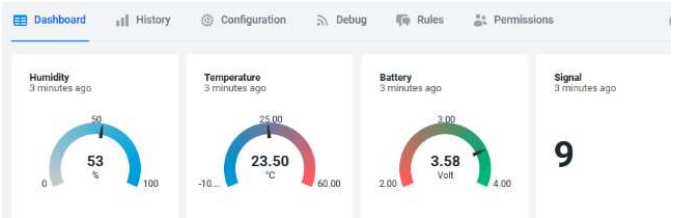
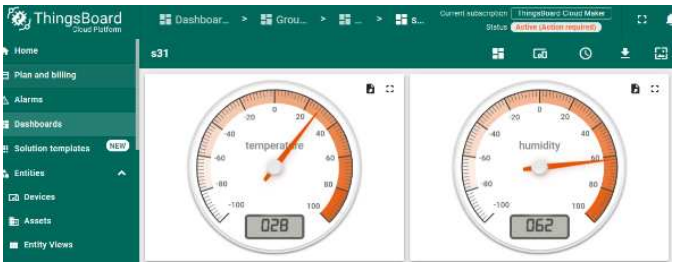
There are two version: **-GE** and **-1T** version of DDS75-NB.

**GE Version**: This version doesn't include SIM card or point to any IoT server. User needs to use AT Commands to configure below two steps to set DDS75-NB send data to IoT server.

- Install NB-IoT SIM card and configure APN. See instruction of Attach Network.
- Set up sensor to point to IoT Server. See instruction of Configure to Connect Different Servers.

Below shows result of different server as a glance.

| Servers | Dash Board | Comments |
|---|---|---|
| Node-Red | | |

| | | |
|---|---|---|
| DataCake |  | |
| Tago.IO | | |
| General UDP | Raw Payload. Need Developer to design Dash Board | |
| General MQTT | Raw Payload. Need Developer to design Dash Board | |
| ThingSpeak | | |
| ThingsBoard |  | |

**1T Version**: This version has 1NCE SIM card pre-installed and configure to send value to ThingsEye. User Just need to select the sensor type in ThingsEyeand Activate DDS75-NB and user will be able to see data in ThingsEye. See here for ThingsEye Config Instruction.

## 2.2 Payload Types

To meet different server requirement, DDS75-NB supports different payload type.

**Includes:**

- General JSON format payload. (Type=5)
- HEX format Payload. (Type=0)
- ThingSpeak Format. (Type=1)
- ThingsBoard Format. (Type=3)

User can specify the payload type when choose the connection protocol. Example:

**AT+PRO=1,0** // Use COAP Connection & hex Payload

**AT+PRO=1,5** // Use COAP Connection & Json Payload

**AT+PRO=2,0** // Use UDP Connection & hex Payload

**AT+PRO=2,5** // Use UDP Connection & Json Payload

**AT+PRO=3,0** // Use MQTT Connection & hex Payload

**AT+PRO=3,5** // Use MQTT Connection & Json Payload

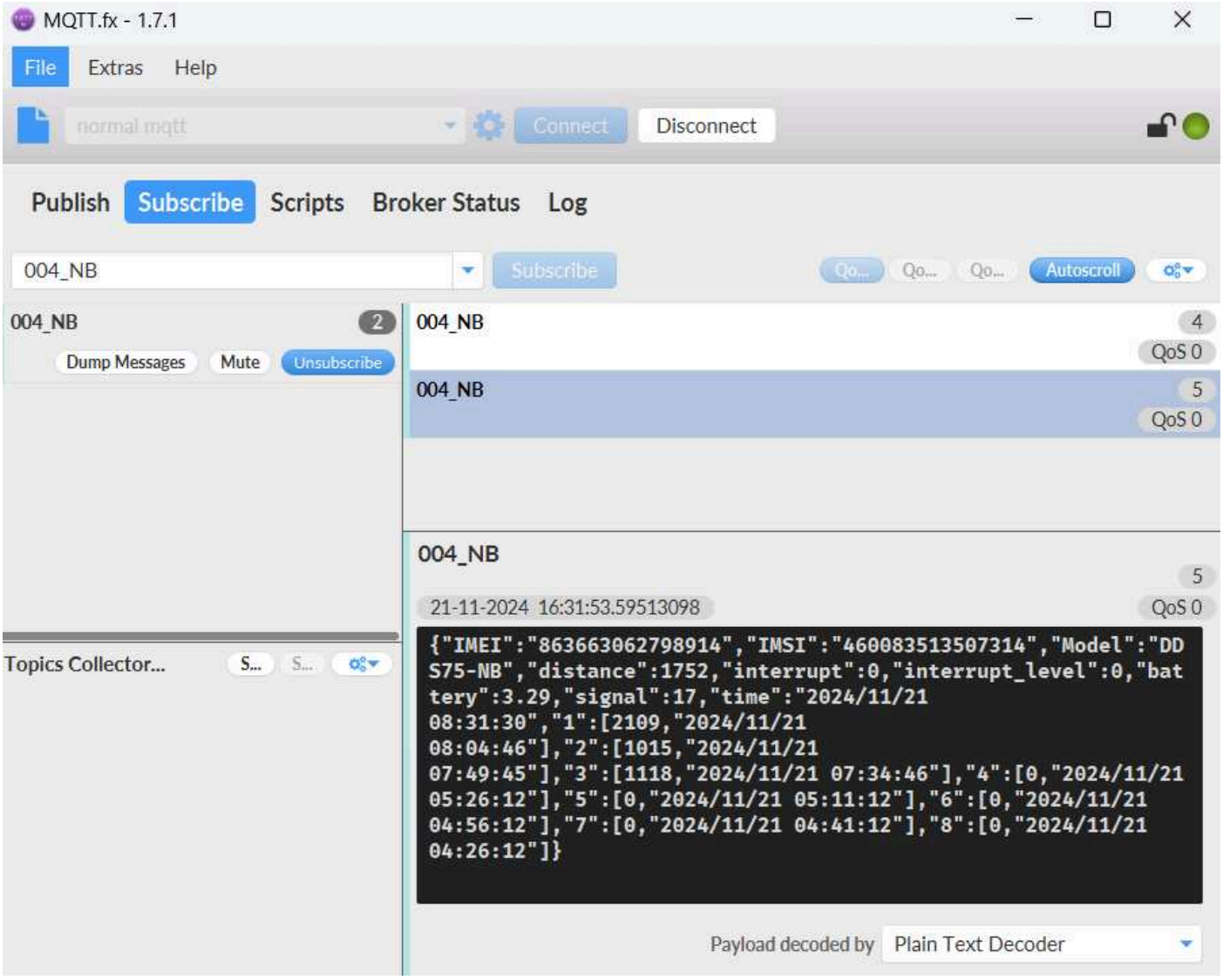
**AT+PRO=4,0** // Use TCP Connection & hex Payload

**AT+PRO=4,5** // Use TCP Connection & Json Payload

### 2.2.1 General Json Format(Type=5)

This is the General Json Format. As below:

**{"IMEI":"863663062798914","IMSI":"460083513507314","Model":"DDS75-NB","distance":1752,"interrupt":0,"interrupt_level":0,"battery":3.29,"signal":17,"time":"2024/11/21 08:31:30","1":[2109,"2024/11/21 08:04:46"],"2":[1015,"2024/11/21 07:49:45"],"3":[1118,"2024/11/21 07:34:46"],"4":[0,"2024/11/21 05:26:12"],"5":[0,"2024/11/21 05:11:12"],"6":[0,"2024/11/21 04:56:12"],"7":[0,"2024/11/21 04:41:12"],"8":[0,"2024/11/21 04:26:12"]}**




**Notice, from above payload:**

- Distance , Battery, Signal & time are the value at uplink time.
- Json entry 1 ~ 8 are the last 1 ~ 8 sampling data as specify by **AT+CLOCKLOG=1,65535,15,8** Command. Each entry includes (from left to right): Temperature, Humidity, Sampling time.

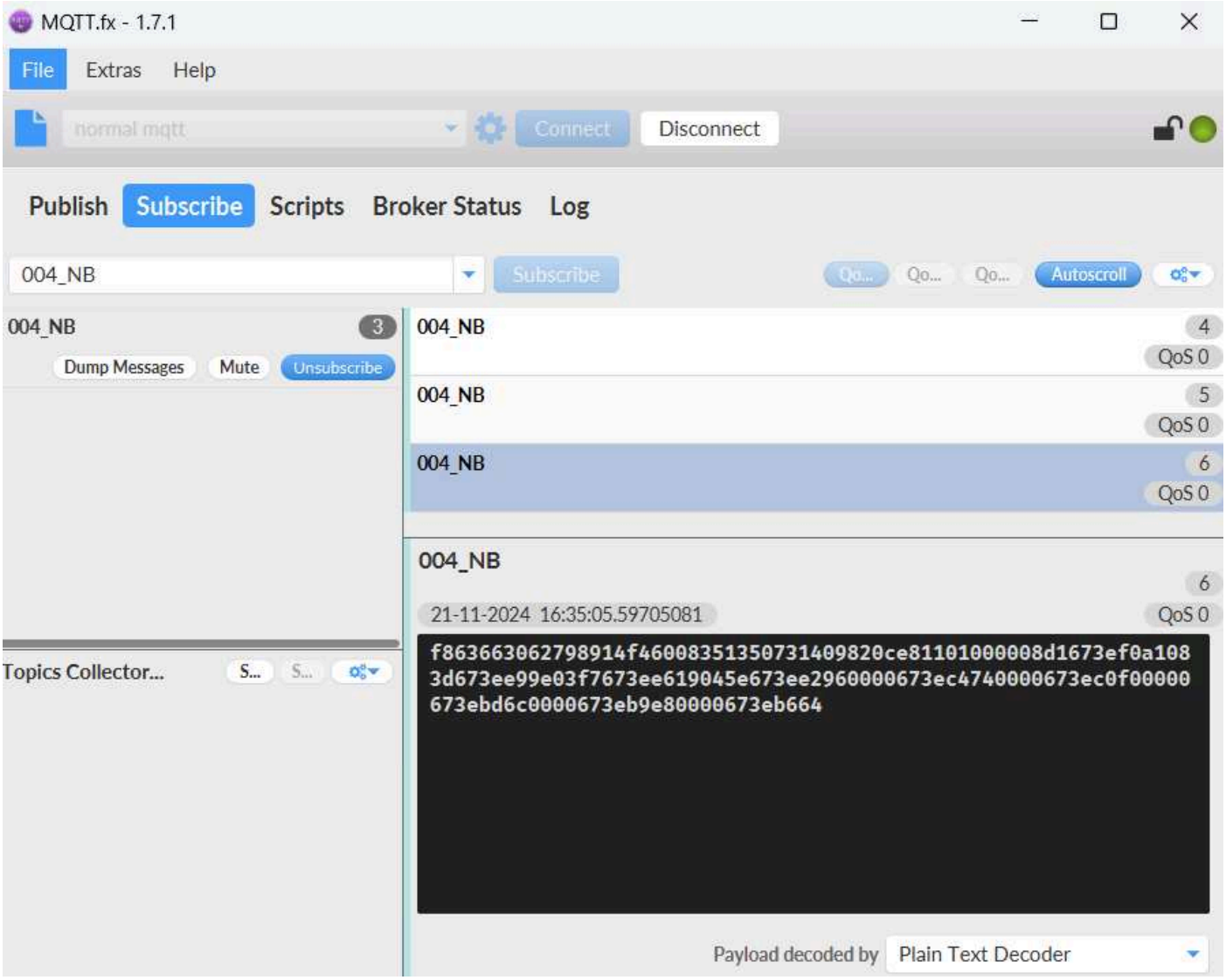
## 2.2.2 HEX format Payload(Type=0)

This is the HEX Format. As below:

**f863663062798914f460083513507314098 20ce81101000008d1673ef0a1083d673ee99e03f7673ee619045e673**

| HEX Format for DDS75-NB(AT+CLOCKLOG=1,65535,15,8) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| f863663062798914 | f460083513507314 | 0982 | 0ce8 | 11 | 01 | 00 | 00 | 08d1 | 673ef0a1 |
| f+IMEI | f+IMSI | Version | BAT | signal | Mod | Interrupt | Interrupt_level | Distance | Timestamp |
| 8 Bytes | 8 Bytes | | | | | 14 Bytes | | | |
| | | | | | | | | | |
| 083d 673ee99e | 03f7673ee619 | 045e673ee296 | | | | 0000673ec474 | | | |
| Distance Timestamp | last 2nd data | last 3rd data | | | | last 4th data | | | |
| 6 Bytes | 6 Bytes | 6 Bytes | | | | 6 Bytes | | | |
| | | | | | | | | | |
| 0000673ec0f0 | 0000673ebd6c | 0000673eb9e8 | | | | 0000673eb664 | | | |
| last 5th data | last 6th data | last 7th data | | | | last 8th data | | | |
| 6 Bytes | 6 Bytes | 6 Bytes | | | | 6 Bytes | | | |

If we use the MQTT client to subscribe to this MQTT topic, we can see the following information when the NB sensor uplink data.




**Device ID(f+IMEI):** f863663062798914 = 863663062798914

**SIM Card ID(f+IMSI):** f460083513507314 = 460083513507314

**Version:**

These bytes include the hardware and software version.

**Higher byte:** Specify Sensor Model: 0x09 for DDS75-NB

**Lower byte:** Specify the software version: 0x82=130, means firmware version 1.3.0

**BAT (Battery Info):**

Ex1: 0x0D38 = 3384mV

**Signal Strength:**

NB-IoT Network signal Strength.

**Ex1: 0x13 = 19**

**0** -113dBm or less

**1** -111dBm

**2...30** -109dBm... -53dBm

**31** -51dBm or greater

**99** Not known or not detectable

**Interrupt:**

If this packet is generated by interrupt or not.

**Example:**

0x00: Normal uplink packet.

0x01: Interrupt Uplink Packet.

**Interrupt_level:**

This byte shows whether the interrupt is triggered by a high or low level.

**Ex1:** 0x00 Interrupt triggered by falling edge (low level)

**Ex2:** 0x01 Interrupt triggered by rising edge (high level)

**Distance:**

Get the distance. Flat object range 280mm - 7500mm.

For example, if the data you get from the register is **0x0B 0x05**, the distance between the sensor and the measured object is

**0B05(H) = 2821 (D) = 2821 mm.**

- If the sensor value is 0x0000, it means system doesn't detect ultrasonic sensor.

**Timestamp:**

Unit Timestamp Example: 6653ddb4(H) = 1716772276(D)

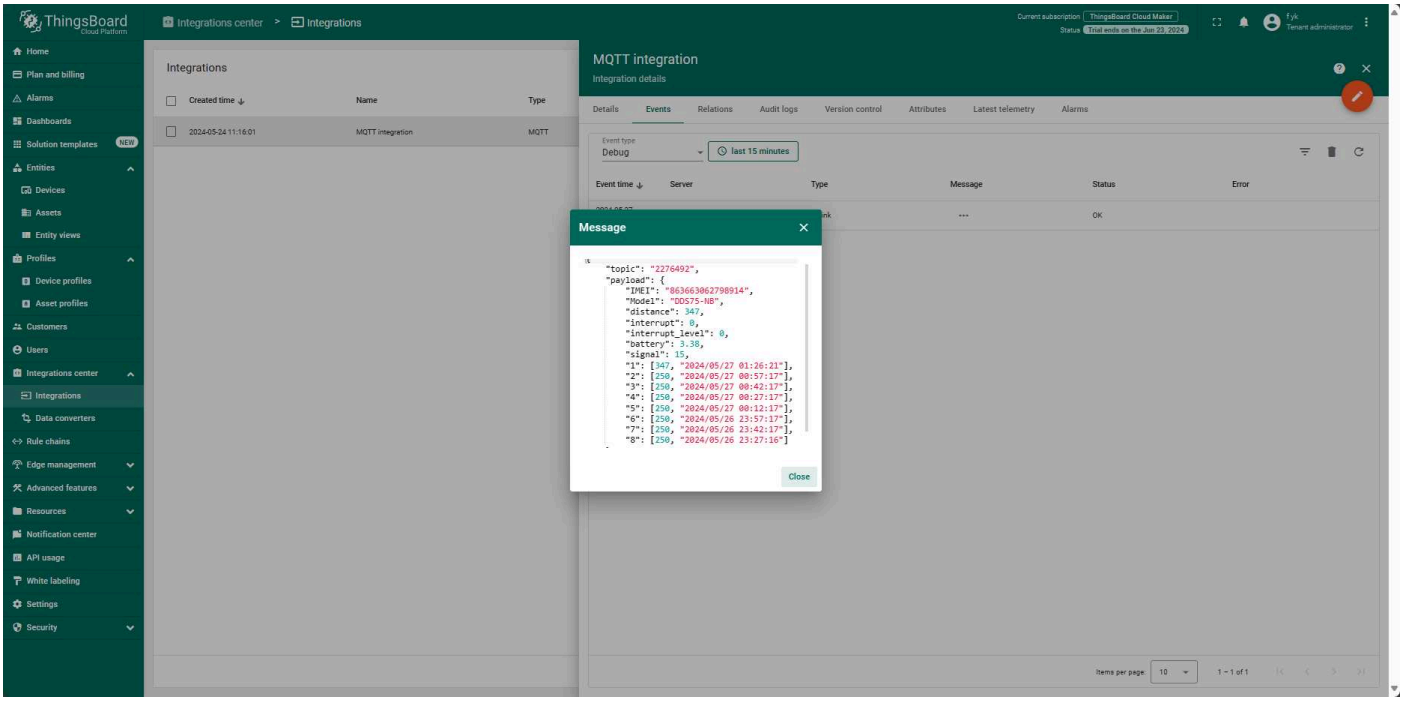
Put the decimal value into this link(https://www.epochconverter.com)) to get the time.

## 2.2.3 ThingsBoard Payload(Type=3)

Type3 payload special design for ThingsBoard, it will also configure other default server to ThingsBoard.


```
{
  "topic": "2276492",
  "payload": {
    "IMEI": "863663062798914",
    "Model": "DDS75-NB",
```

```
    "distance": 347,
    "interrupt": 0,
    "interrupt_level": 0,
    "battery": 3.38,
    "signal": 15,
    "1": [347, "2024/05/27 01:26:21"],
    "2": [250, "2024/05/27 00:57:17"],
    "3": [250, "2024/05/27 00:42:17"],
    "4": [250, "2024/05/27 00:27:17"],
    "5": [250, "2024/05/27 00:12:17"],
    "6": [250, "2024/05/26 23:57:17"],
    "7": [250, "2024/05/26 23:42:17"],
    "8": [250, "2024/05/26 23:27:16"]
  }
}
```

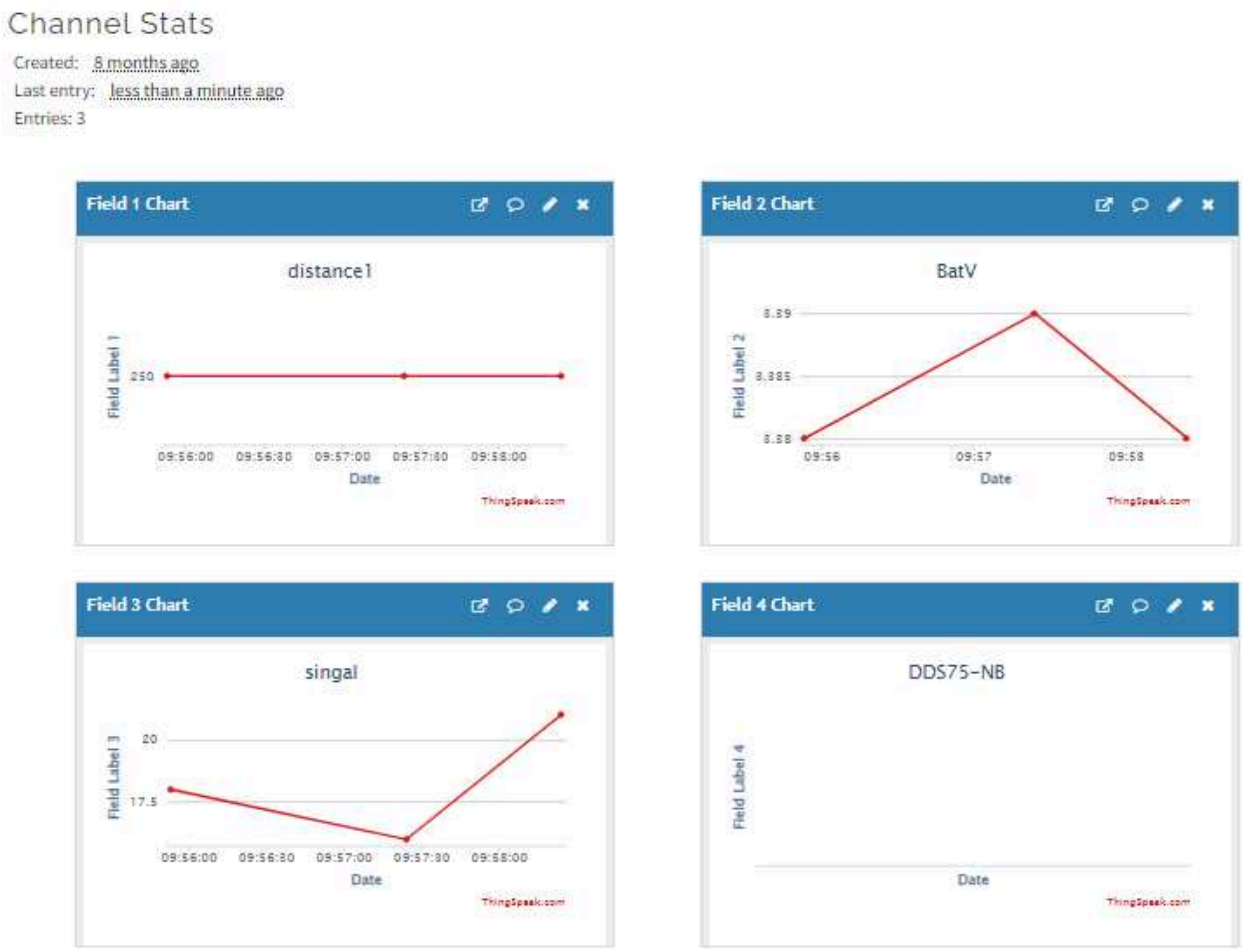

## 2.2.4 ThingSpeak Payload(Type=1)

This payload meets ThingSpeak platform requirement. It includes only four fields. Form 1~3 are:

Distance, Battery & Signal. This payload type only valid for ThingsSpeak Platform.

As below:

**field1=Distance value&field2=Battery value&field3=Signal value**




## 2.3 Test Uplink and Change Update Interval

By default, Sensor will send uplinks **every 2 hours**

User can use below commands to change the **uplink interval**.

**AT command: AT+TDC**

Example: AT+TDC=7200  // Set Update Interval to 7200 seconds

**Downlink command: 0x01**

Format: Command Code (0x01) followed by 3 bytes.

Example:  12 hours= 43200 seconds  43200(D)=0xA8C0(H)

Downlink Payload: **01 00 A8 C0**  // AT+TDC=43200, Set Update Interval to 12 hours.

**Note: User can also push the button for more than 1 seconds to activate an uplink.**

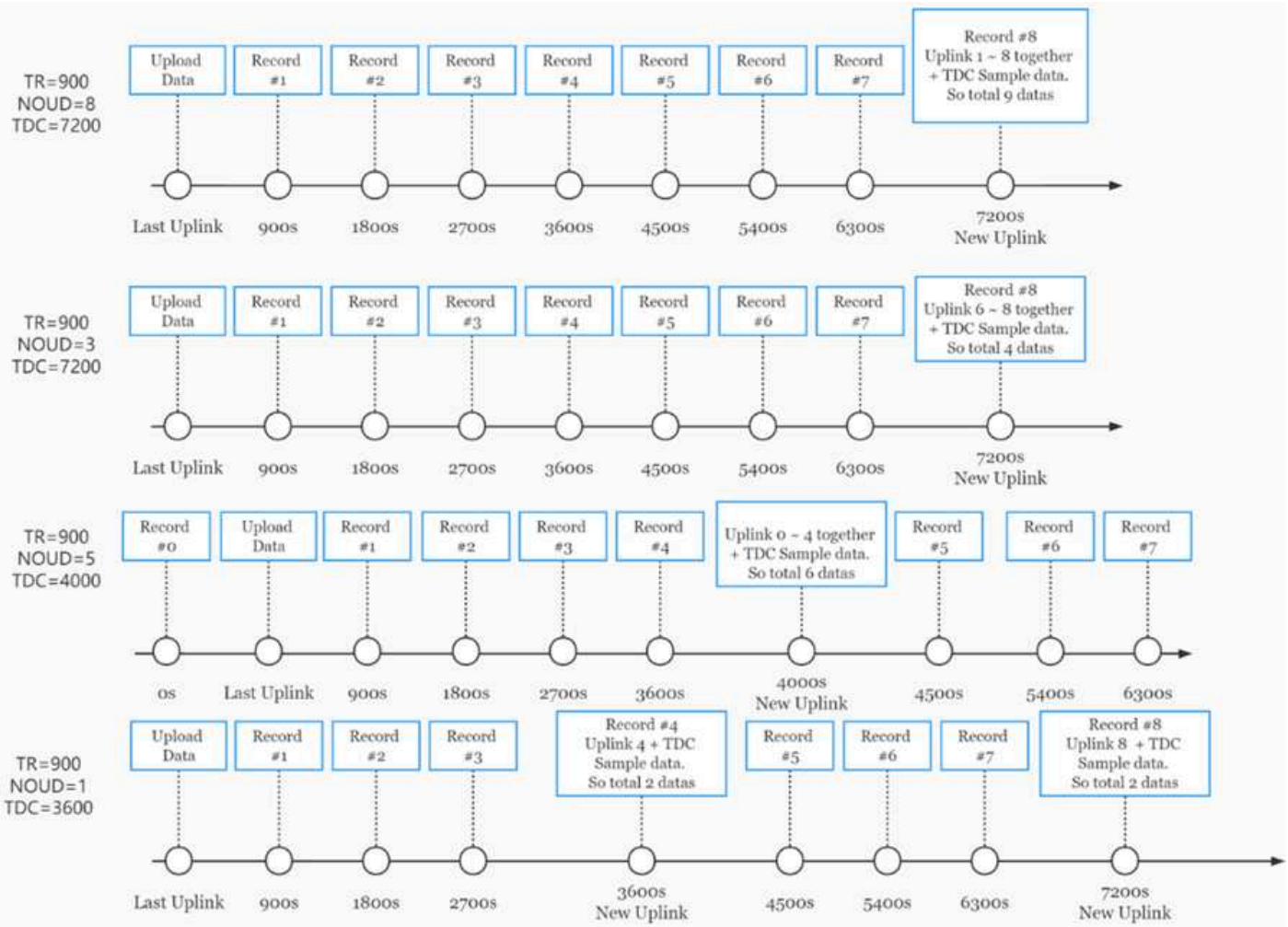
## 2.4 Multi-Samplings and One uplink

**Notice: The AT+NOUD feature is upgraded to Clock Logging, please refer [Clock Logging Feature]**

To save battery life, DDS75-NB will sample distance data every 15 minutes and send one uplink every 2 hours. So each uplink it will include 8 stored data + 1 real-time data. They are defined by:

- **AT+TR=900** // The unit is seconds, and the default is to record data once every 900 seconds (15 minutes, the minimum can be set to 180 seconds)
- **AT+NOUD=8** // The device uploads 8 sets of recorded data by default. Up to 32 sets of record data can be uploaded.

The diagram below explains the relationship between TR, NOUD, and TDC more clearly:




## 2.5 Trggier an uplink by external interrupt

DDS75-NB has an external trigger interrupt function. Users can use the GPIO_EXTI pin to trigger the upload of data packets.

**AT command:**

- **AT+INTMOD** // Set the trigger interrupt mode
- **AT+INTMOD=0** // Disable Interrupt
- **AT+INTMOD=1** // Trigger by rising and falling edge

- **AT+INTMOD=2** // Trigger by falling edge
- **AT+INTMOD=3** // Trigger by rising edge

## 2.6 Distance Alarm

Feature: Set alarm of LDDS and NMDS.

**AT command: AT+LDDSALARM** (Range:280mm - 7500mm)

Example: AT+LDDSALARM=500,2000 // Set the alarm threshold

**Downlink command: 0X08**

Format: Command Code (0x08) followed by 4 bytes.

Example: Downlink Payload: **08 01 F4 07 D0** //AT+LDDSALARM=500,2000

## 2.7 Clock logging (Since firmware version v1.2.1)

Sometimes when we deploy lots of end nodes in field. We want all sensors sample data at the same time, and upload these data together for analyze. In such case, we can use clock loging feature.

We can use this command to set the start time of data recording and the time interval to meet the requirements of the specific collection time of data.

- **AT command: AT+CLOCKLOG=a,b,c,d**

**a: 0:** Disable Clock logging. **1:** Enable Clock Logging

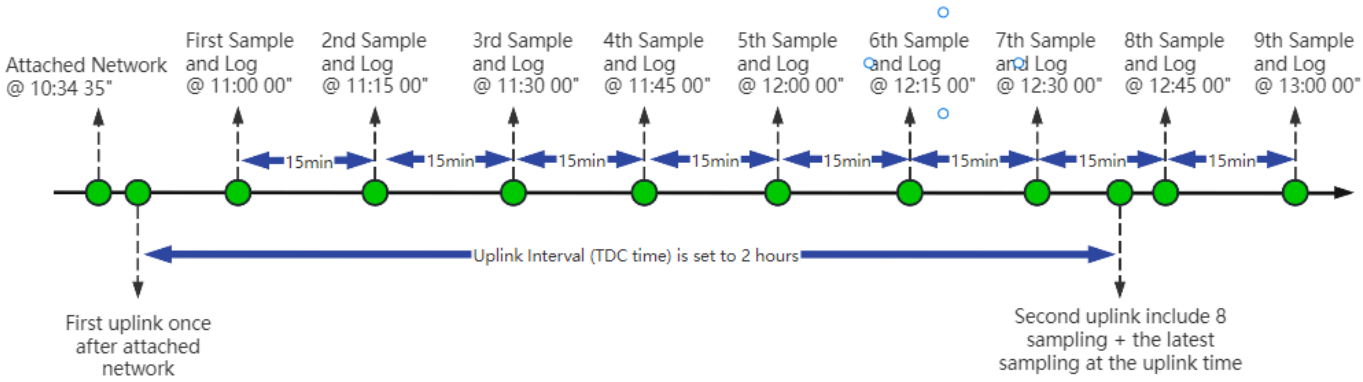
**b:** Specify First sampling start second: range **(0 ~ 3599, 65535)** // **Note:** If parameter b is set to 65535, the log period starts after the node accesses the network and sends packets.

**c:** Specify the sampling interval: range **(0 ~ 255 minutes)**

**d:** How many entries should be uplink on every TDC **(max 32)**

**Note: To disable clock recording, set the following parameters: AT+CLOCKLOG=1,65535,0,0**

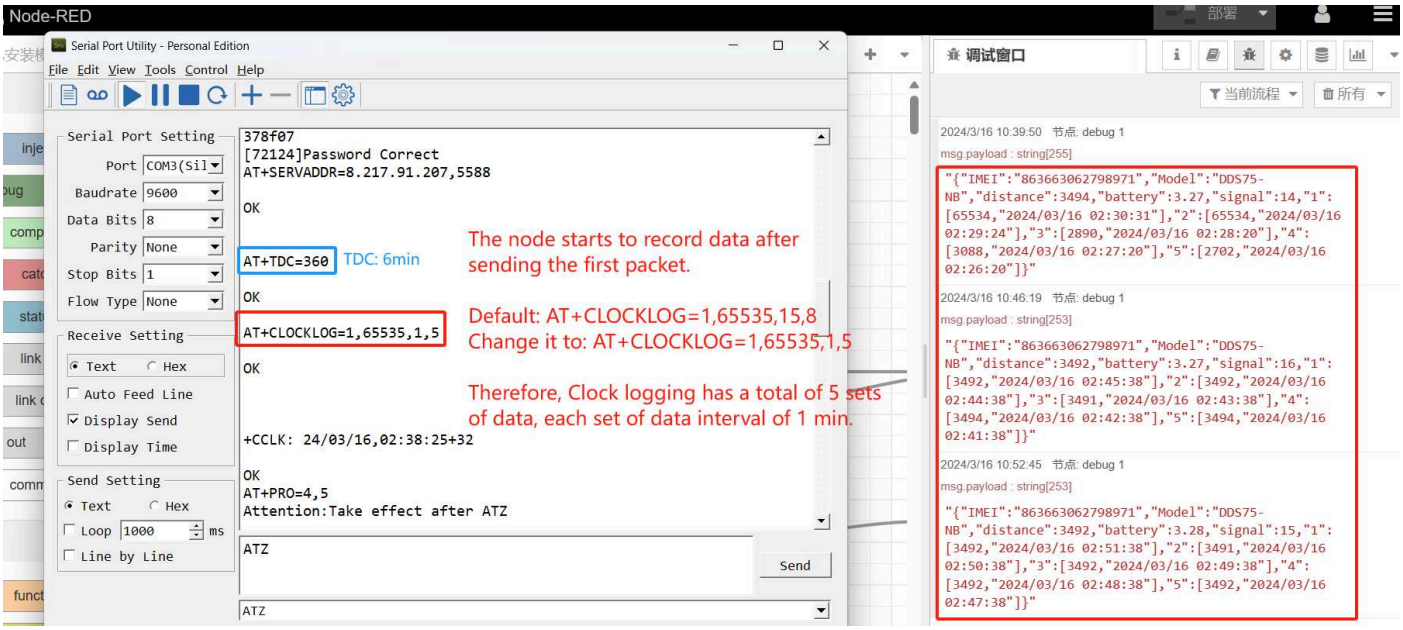
Example: AT+CLOCKLOG=1,0,15,8
Device will log data to memory start from the 0" second (11:00 00" of first hour and then sampling and log every 15 minutes.Every TDC uplink, the uplink payload will consist: Battery info + last 8 memory record with timestamp + the latest sample at uplink time) . See below for the example.




**Example:**

**AT+CLOCKLOG=1,65535,1,5**

After the node sends the first packet, data is recorded to the memory at intervals of 1 minute. For each TDC uplink, the uplink load will include: battery information + the last 5 memory records (payload + timestamp).




**Note: Users need to synchronize the server time before configuring this command. If the server time is not synchronized before this command is configured, the command takes effect only after the node is reset.**

- **Downlink command: 0x0A**

Format: Command Code (0x0A) followed by 5 bytes.

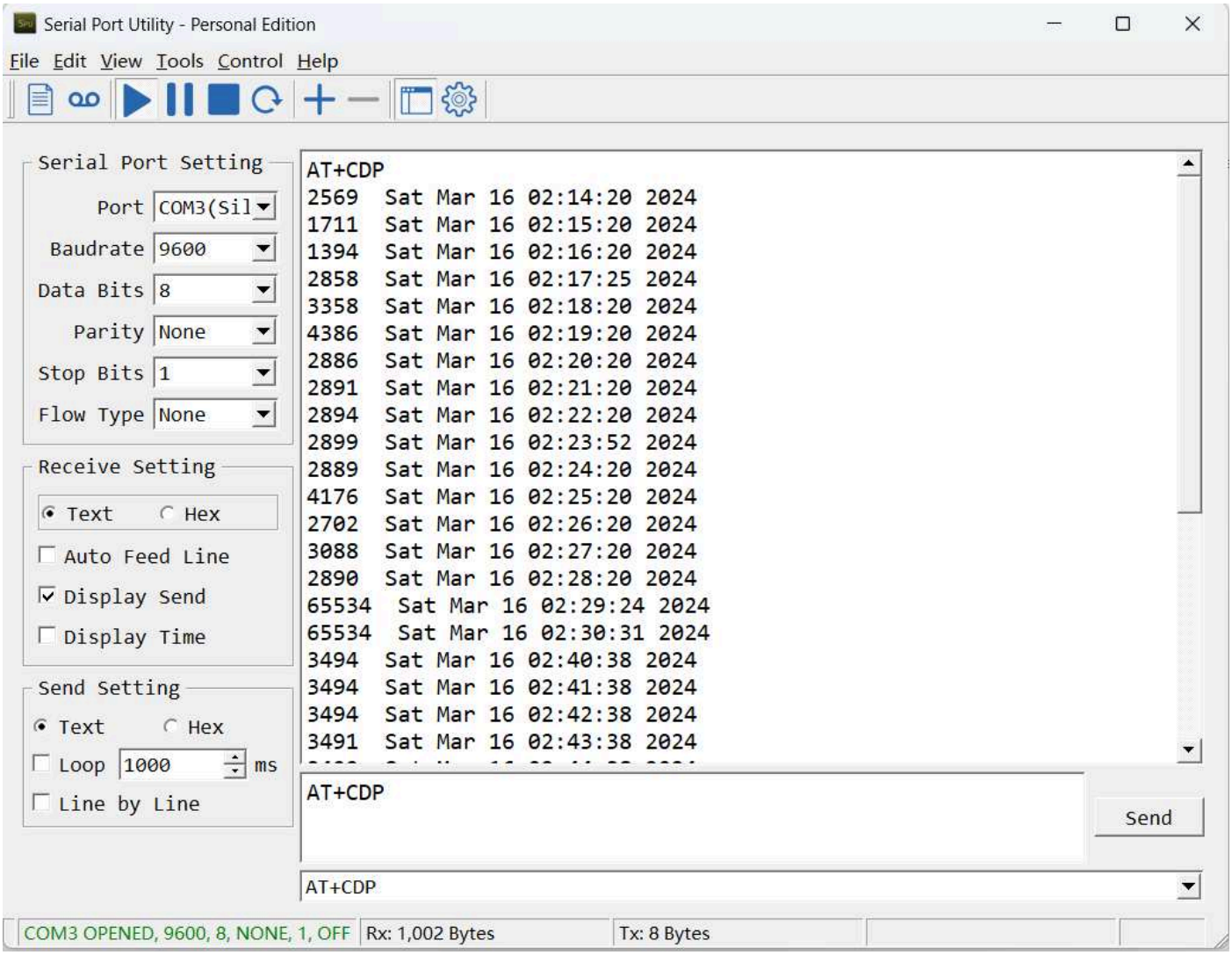
- **Example 1**: Downlink Payload: **0A01FFFF0F08** // Set SHT record time: AT+CLOCKLOG=1,65535,15,8
- **Example 1**: Downlink Payload: **0A0104B00F08** // Set SHT record time: AT+CLOCKLOG=1,1200,15,8

**Note: When entering the downlink payload, there must be no Spaces between bytes.**

## 2.8 Example Query saved historical records

- **AT command: AT+CDP**

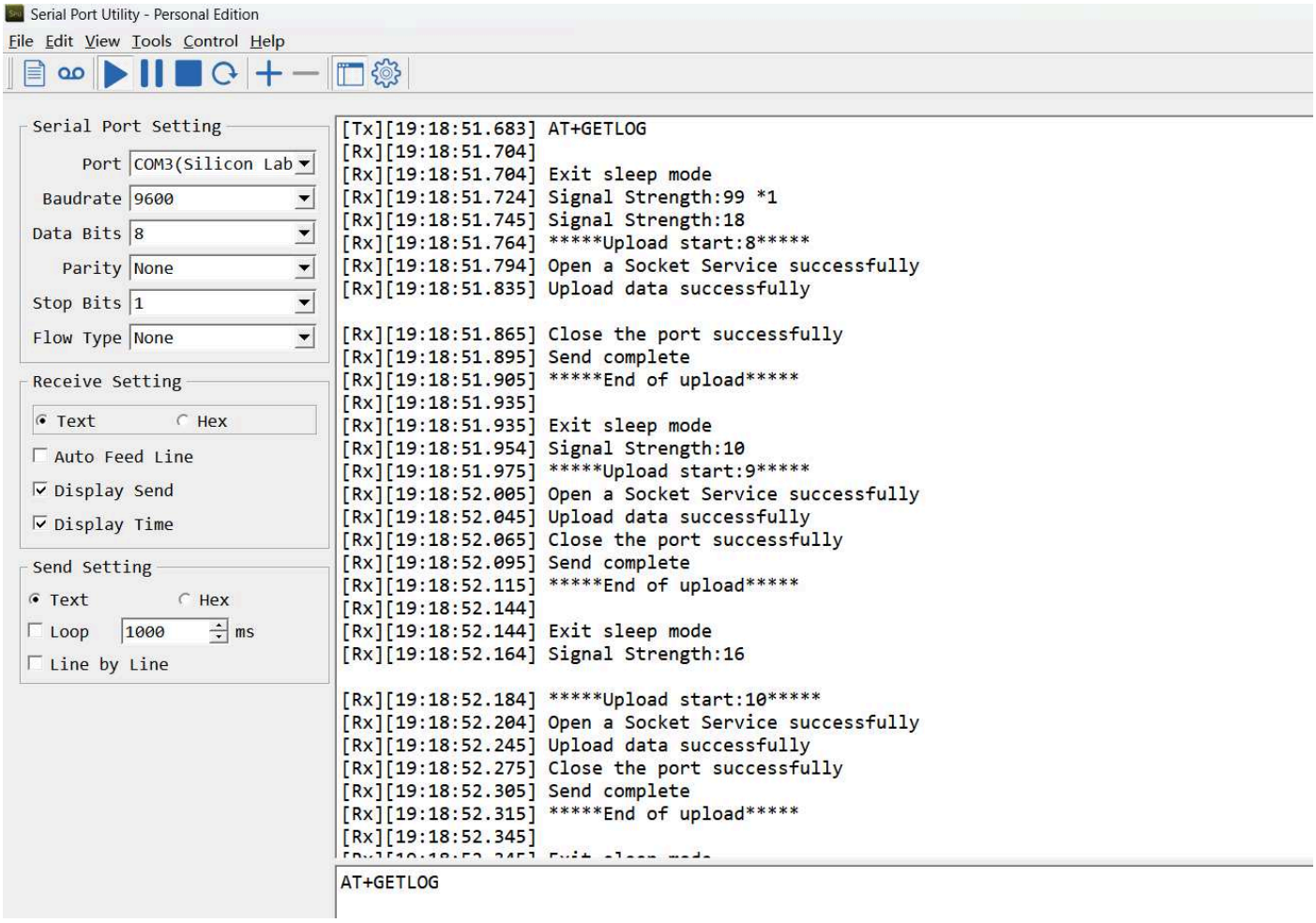
This command can be used to search the saved history, recording up to 32 groups of data, each group of historical data contains a maximum of 100 bytes.




## 2.9 Uplink log query

- **AT command: AT+GETLOG**

This command can be used to query upstream logs of data packets.




## 2.10 Scheduled domain name resolution

This command is used to set up scheduled domain name resolution.

**AT command:**

- **AT+DNSTIMER=XX** // Unit: hour

After setting this command, domain name resolution will be performed regularly.

## 2.11 Set the QoS level

This command is used to set the QoS level of **MQTT**.

**AT command:**

- **AT+MQOS=xx** // 0~2

**Downlink command: 0x07**

Format: Command Code (0x07) followed by 1 byte.

**Ex1:** Downlink payload: **0x0700** //AT+MQOS=0

**Ex2:** Downlink payload: **0x0701** //AT+MQOS=1

## 2.12 Set CoAP option

This command sets the connection parameters of the COAP.

**AT command:**

- AT+URI1 // CoAP option name, CoAP option length, "CoAP option value"
- AT+URI2 // CoAP option name, CoAP option length, "CoAP option value"
- AT+URI3 // CoAP option name, CoAP option length, "CoAP option value"
- AT+URI4 // CoAP option name, CoAP option length, "CoAP option value"

**Example:**

- AT+URI1=11,38,"i/faaa241f-af4a-b780-4468-c671bb574858"

## 2.13 Set the downlink debugging mode(Since firmware v1.3.0)

Feature: Set the conversion between the standard version and 1T version downlinks.

**AT command: AT+DOWNTE**

| Command Example | Function/Parameters | Response/Explanation |
|---|---|---|
| AT+DOWNTE=? | Get current Settings | 0,0 (default)<br>OK |
| AT+DOWNTE=a,b | **a**: Set the conversion between the downlink of the standard version and 1T version | **0**: Set the downlink of the standard version.<br>**1**: Set the downlink of the 1T version(ThingsEye platform) |
| | **b**: Enable/Disable downlink debugging | **0**: Disable downlink debugging mode.<br>**1**: Enable downlink debugging mode, users can see the original downlink reception. |

**Example:**

- AT+DOWNTE=0,1 // Set to standard version downlink, and enable downlink debugging.
- AT+DOWNTE=1,1 // Set to 1T version downlink, and enable downlink debugging.

**Downlink Command:**

No downlink commands for feature

## 2.14 Domain name resolution settings(Since firmware v1.3.0)

Feature: Set dynamic domain name resolution IP.

**AT command: AT+BKDNS**

| Command Example | Function/Parameters | Response/Explanation |
|---|---|---|
| AT+BKDNS=? | Get current Settings | 0,0,NULL (default)<br>OK |
| AT+BKDNS=a,b,c | **a**: Enable/Disable dynamic domain name resolution. | **1**: Disable dynamic domain name update. The ip address will be saved after the domain name is resolved, if the next domain name resolution fails, the last saved ip address will be used.<br>**2**: Enable dynamic domain name update. The ip address will be saved after domain name resolution, if the next domain name resolution fails, the last saved ip address will be used, and the domain name resolution will be updated regularly according to the time set by the customer. |
| | **b**: Set the time to update the domain name resolution at regular intervals. | Unit: hour |
| | **c**: Set the IP address manually. | The format is the same as AT+SERVADDR.<br>If domain name resolution fails, this ip address will be used directly, if domain name resolution succeeds, parameter c will be updated to the successfully resolved IP address. |

**Example:**

- AT+BKDNS=1,0 // Dynamic domain name resolution is disabled.
- AT+BKDNS=2,1 // The dynamic domain name resolution function is enabled and the automatic update time is set to 1 hour.
- AT+BKDNS=2,4,3.69.98.183,1883 // The dynamic domain name resolution function is enabled and the automatic update time is set to 4 hour, and manually set the ip address, if the domain name failed to resolve, it will directly use this ip to communicate. When the next domain name resolution is successful, it will be updated to the ip address of the successful resolution.

**Downlink Command:**

No downlink commands for feature

# 3. Configure DDS75-NB

## 3.1 Configure Methods

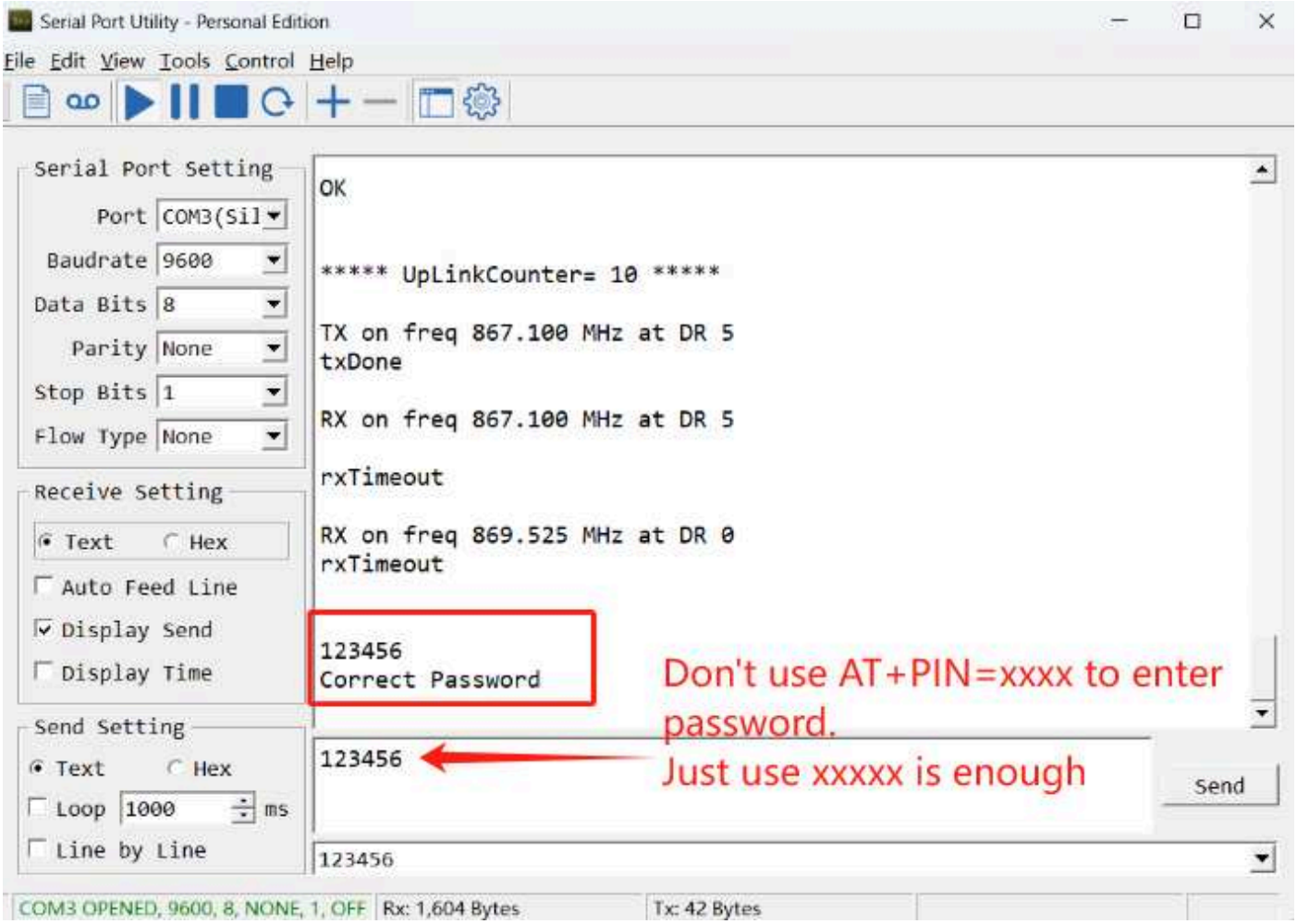
DDS75-NB supports below configure method:

- AT Command via Bluetooth Connection (**Recommended**): BLE Configure Instruction.
- AT Command via UART Connection : See UART Connection.

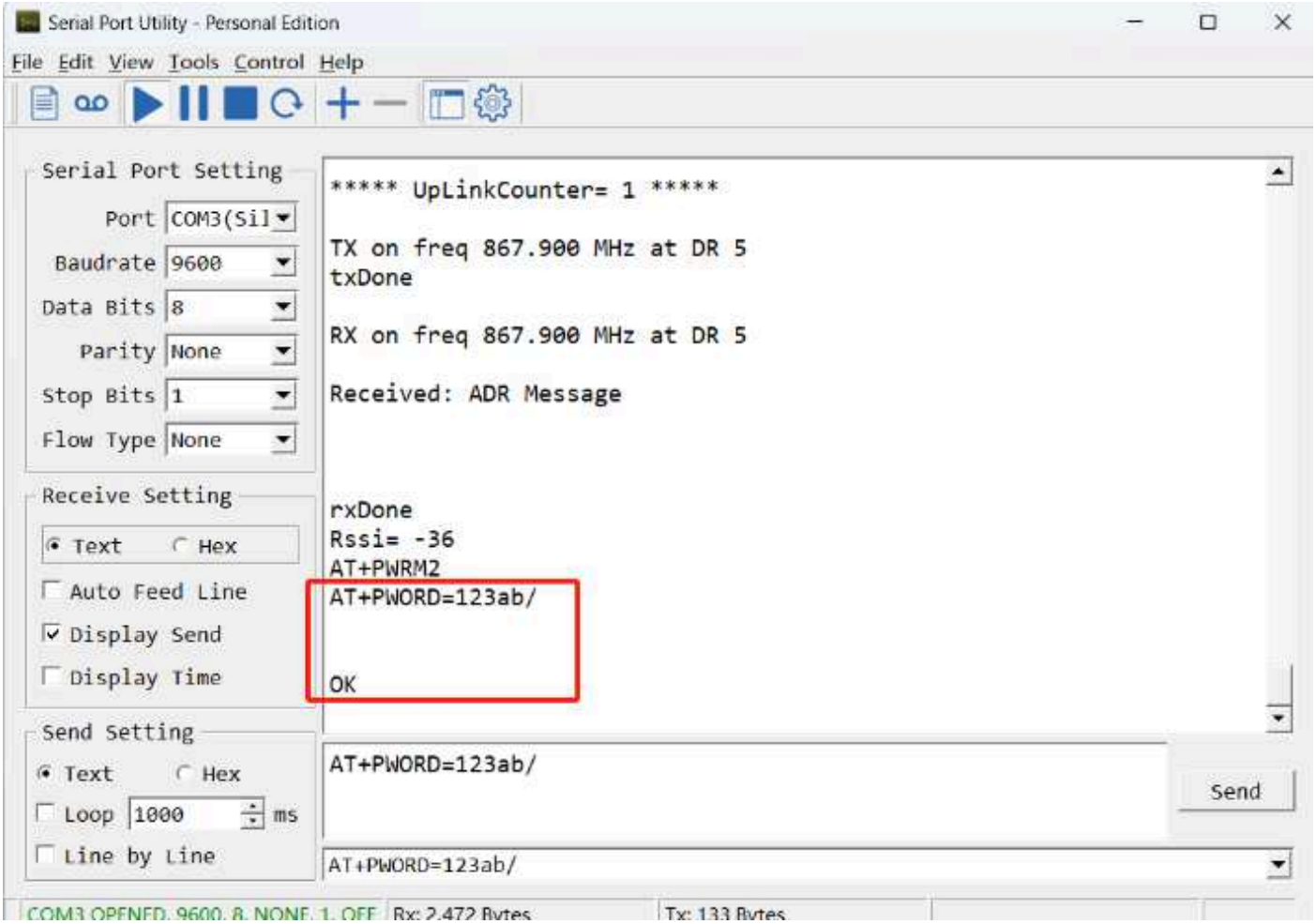
## 3.2 Serial Access Password

After the Bluetooth or UART connection is successful, use the Serial Access Password to enter the AT command window.

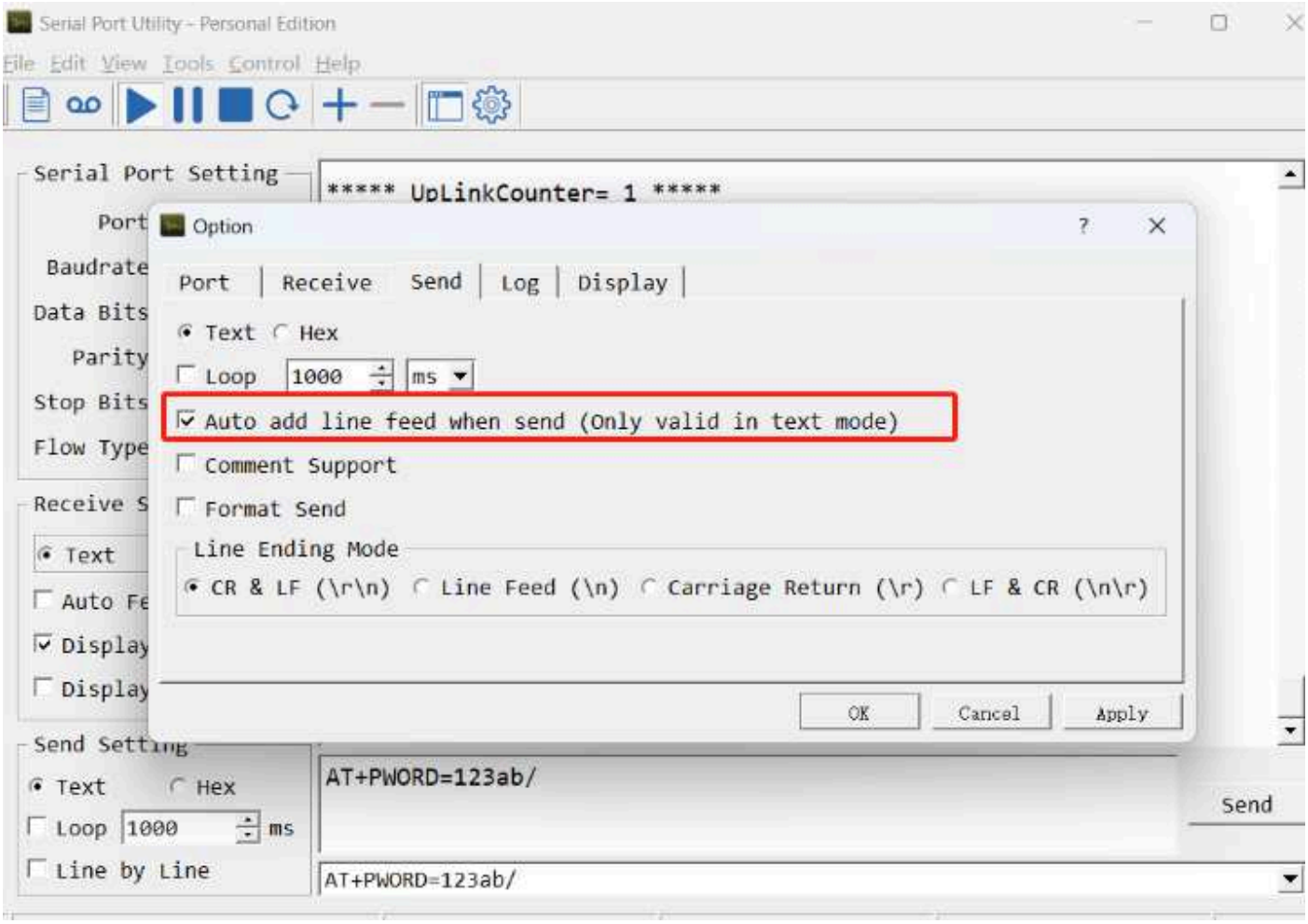
The label on the box of the node will print the initial password: AT+PIN=**xxxxxx**, and directly use the six-digit password to access the AT instruction window.




If you need to change the password, use **AT+PWORD=**xxxxxx (6 characters), NB nodes only support lowercase letters.




**Note: After entering the command, you need to add a line break, and you can also set automatic line breaks in the Bluetooth tool or UART connection tool.**




## 3.3 AT Commands Set

AT+<CMD>? : Help on <CMD>

AT+<CMD> : Run <CMD>

AT+<CMD>=<value> : Set the value

AT+<CMD>=? : Get the value

### General Commands

AT : Attention

AT? : Short Help

ATZ : MCU Reset

AT+TDC : Application Data Transmission Interval

AT+CFG : Print all configurations

AT+CFGMOD : Working mode selection

AT+DEUI : Get or set the Device ID

AT+INTMOD : Set the trigger interrupt mode

AT+5VT : Set extend the time of 5V power

AT+PRO : Choose agreement

AT+RXDL : Extend the sending and receiving time

AT+DNSCFG : Get or Set DNS Server

AT+GETSENSORVALUE : Returns the current sensor measurement

AT+NOUD : Get or Set the number of data to be uploaded

AT+CDP : Read or Clear cached data

AT+SERVADDR : Server Address

**MQTT Management**

AT+CLIENT : Get or Set MQTT client

AT+UNAME : Get or Set MQTT Username

AT+PWD : Get or Set MQTT password

AT+PUBTOPIC : Get or Set MQTT publish topic

AT+SUBTOPIC : Get or Set MQTT subscription topic

**Information**

AT+FDR : Factory Data Reset

AT+PWORD : Serial Access Password

AT+LDATA : Get the last upload data

AT+CDP : Read or Clear cached data

# 4. Battery & Power Consumption

DDS75-NB use ER26500 + SPC1520 battery pack. See below link for detail information about the battery info and how to replace.

**Battery Info & Power Consumption Analyze** .

# 5. Firmware update

User can change device firmware to:

- Update with new features.
- Fix bugs.

Firmware and changelog can be downloaded from : **Firmware download link**

Methods to Update Firmware:

- (Recommended way) OTA firmware update via BLE: **Instruction**.
- Update through UART TTL interface : **Instruction**.

# 6. FAQ

## 6.1 How can I access t BC660K-GL AT Commands?

User can access to BC660K-GL directly and send AT Commands.

See BC660K-GL AT Command set

## 6.2 Can I use DDS75-NB in condensation environment?

DDS75-NB is not suitable to be used in condensation environment. Condensation on the DDS75-NB probe will affect the reading and always got 0.

## 6.3 How to configure the certificate?

User can can refer to this description to configure the certificate.

# 7. Trouble Shooting

## 7.1 Why does the sensor reading show 0 or "No sensor"

1. The measurement object is very close to the sensor, but in the blind spot of the sensor.

2. Sensor wiring is disconnected

3. Not using the correct decoder

## 7.2 Abnormal readings The gap between multiple readings is too large or the gap between the readings and the actual value is too large

1) Please check if there is something on the probe affecting its measurement (condensed water, volatile oil, etc.)

2) Does it change with temperature, temperature will affect its measurement

3) If abnormal data occurs, you can turn on DEBUG mode, Please use downlink or AT COMMAN to enter DEBUG mode.

downlink command: **F1 01**, AT command: **AT+DDEBUG=1**

4) After entering the debug mode, it will send 20 pieces of data at a time, and you can send its uplink to us for analysis

Its original payload will be longer than other data. Even though it is being parsed, it can be seen that it is abnormal data.

Please send the data to us for check.

# 8. Order Info

Part Number: **DDS75-NB-XX**

**XX**:

- **GE**: General version ( Exclude SIM card)
- **1T**: with 1NCE * 10 years 500MB SIM card and Pre-configure to ThingsEye server

# 9. Packing Info

**Package Includes**:

- DDS75-NB NB-IoT Distance Detection sensor x 1
- External antenna x 1

**Dimension and weight**:

- Device Size: 13.0 x 5 x 4.5 cm
- Device Weight: 150g
- Package Size / pcs : 14.0 x 8x 5 cm
- Weight / pcs : 180g

# 10. Support

- Support is provided Monday to Friday, from 09:00 to 18:00 GMT+8. Due to different timezones we cannot offer live support. However, your questions will be answered as soon as possible in the before-mentioned schedule.
- Provide as much information as possible regarding your enquiry (product models, accurately describe your problem and steps to replicate it etc) and send a mail to Support@dragino.cc.