RS485-NB -- RS485/UART to NB-IoT Converter User Manual

Last modified by Mengting Qiu (/xwiki/bin/view/XWiki/ting) on 2024/04/22 15:57

(http://wiki.dragino.com/xwiki/bi
NB_RS485UART_to_NB-IoT_Conv
width=521&height=414&rev=1.1)

Table of Contents:

1. Introduction

1.1 What is RS485 / UART to NB-IoT Converter

The Dragino RS485-NB is a **RS485 / UART to NB-IoT Converter** for Internet of Things solutions. User can connect RS485 or UART sensor to RS485-NB converter, and configure RS485-NB to periodically read sensor data and upload via NB-IoT network to IoT server.

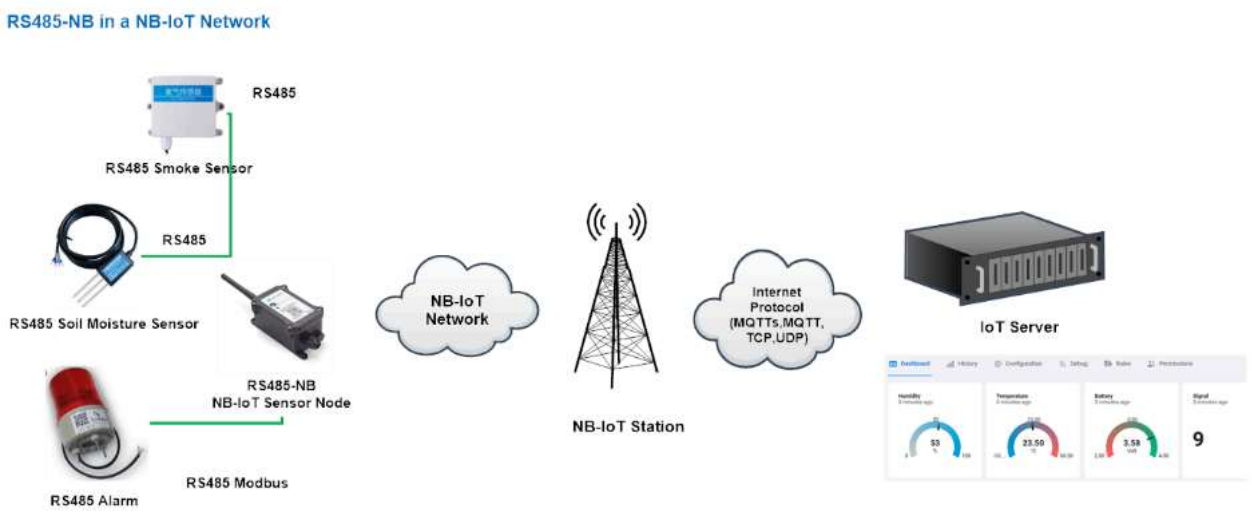
RS485-NB can interface to RS485 sensor, 3.3v/5v UART sensor or interrupt sensor. RS485-NB provides **a 3.3v output** and **a 5v output** to power external sensors. Both output voltages are controllable to minimize the total system power consumption.

RS485-NB supports different uplink methods including **MQTT, MQTTs, UDP & TCP** for different application requirement, and support uplinks to various IoT Servers.

RS485-NB supports **BLE configure and OTA update** which make user easy to use.

RS485-NB is IP67 **waterproof** and powered by **8500mAh Li-SOCI2 battery**, it is designed for long-term use up to several years.

RS485-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
- Support multiply RS485 devices by flexible rules
- Support Modbus protocol
- Support Interrupt uplink
- Supports connecting a UART sensors with 3.3V or 5V
- Multiply Sampling and one uplink
- Support Bluetooth v5.1 remote configure and update firmware
- Uplink via MQTT, MQTTs, TCP, or UDP

- Uplink on periodically
- Downlink to change configure
- 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

I/O Interface:

- Battery controllable output (2.6v ~ 3.6v depends on battery)
- +5v controllable output
- 1 x RS485 Interface
- 1 x UART Interface , 3.3v or 5v
- 1 x Interrupt or Digital IN/OUT pins
- 1 x I2C Interface
- 1 x one wire interface

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 Applications

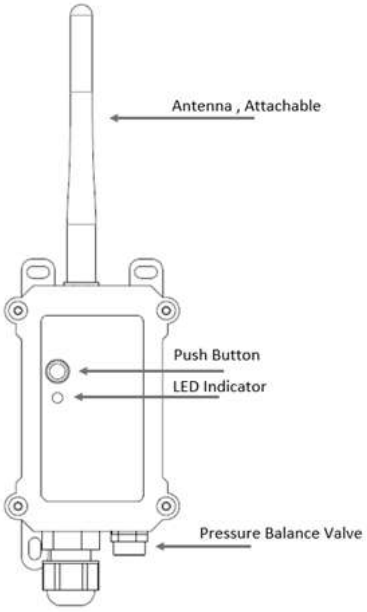
- Smart Buildings & Home Automation
- Logistics and Supply Chain Management
- Smart Metering
- Smart Agriculture
- Smart Cities
- Smart Factory

1.5 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.

1.6 Button & LEDs



Behavior on ACT	Function	Action
Pressing ACT between 1s < time < 3s	Send an uplink	If sensor has already attached to NB-IoT network, sensor will send an uplink packet, **blue led** will blink once. Meanwhile, BLE module will be active and user can connect via BLE to configure device.
Pressing ACT for more than 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. **Green led** will solidly turn on for 5 seconds after joined in network. 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.
Fast press ACT 5 times.	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.

1.7 BLE connection

RS485-NB support BLE remote configure and firmware update.

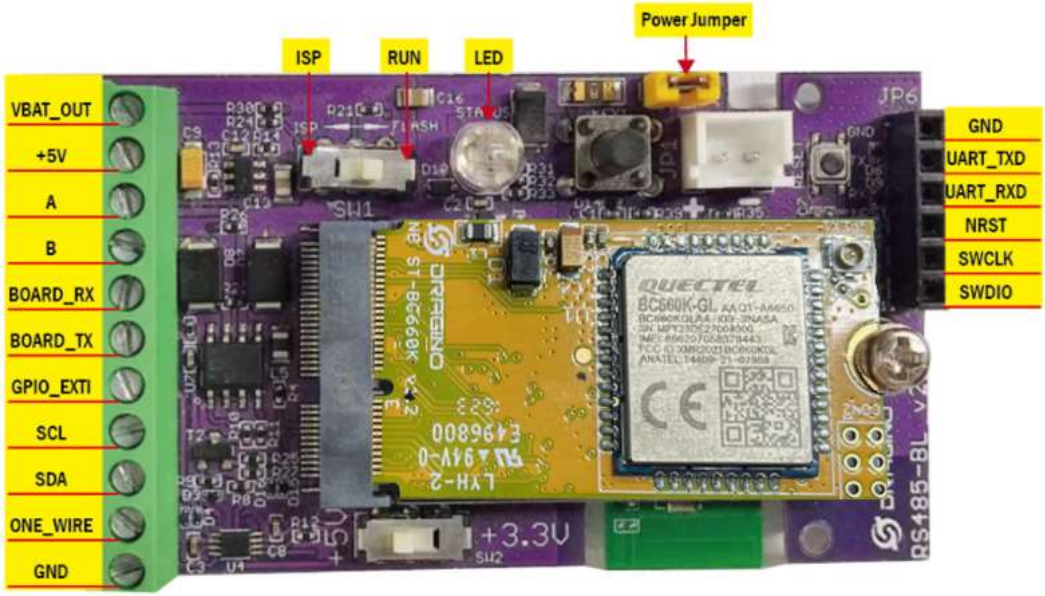
BLE can be used to configure the parameter of sensor or see the console output from sensor. BLE will be only activate on below case:

- 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.

1.8 Pin Definitions , Switch & SIM Direction

RS485-NB use the mother board which as below.



1.8.1 Jumper JP2

Power on Device when put this jumper.

1.8.2 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

1.8.3 Reset Button

Press to reboot the device.

1.8.4 SIM Card Direction

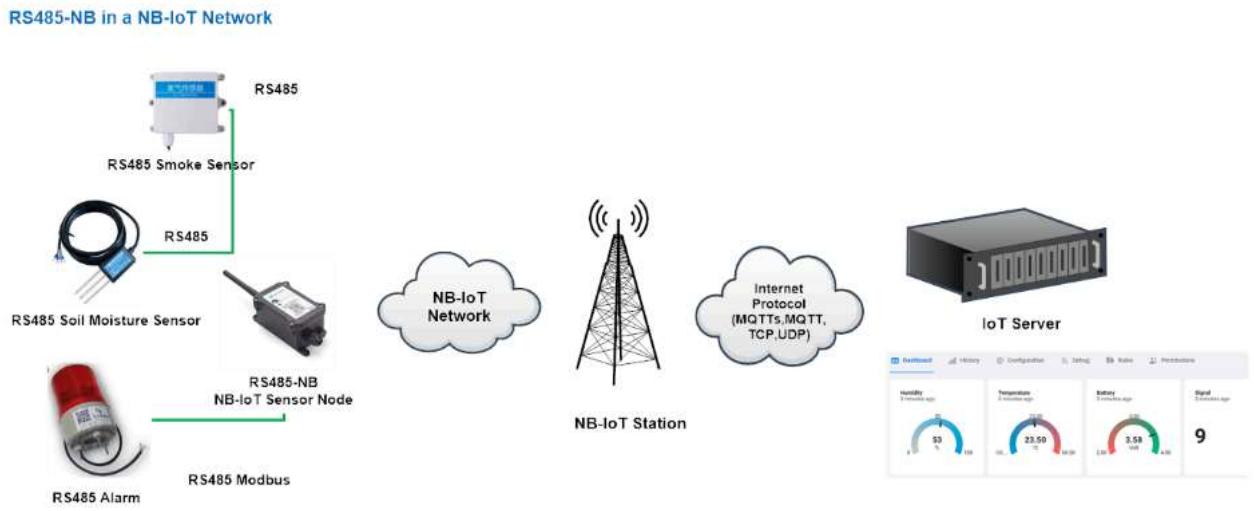
See this link. How to insert SIM Card (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H2.AttachNetwork) .

2. Use RS485-NB to communicate with IoT Server

2.1 Send data to IoT server via NB-IoT network

The RS485-NB is equipped with a NB-IoT module, the pre-loaded firmware in RS485-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 RS485-NB.

Below shows the network structure:



There are two version: **-GE** and **-1D** version of RS485-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 RS485-NB send data to IoT server.

- Install NB-IoT SIM card and configure APN. See instruction of Attach Network (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H2.AttachNetwork) .
- Set up sensor to point to IoT Server. See instruction of Configure to Connect Different Servers (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H3.Configuretoconnecttodifferentservers) .

Below shows result of different server as a glance.

Servers	**Dash Board**	**Comments**
Node-Red (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H3.5A0Node-RedA028viaA0MQTT29)		
DataCake (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H3.4Datacake)		
General UDP (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H3.1GeneralA0UDPA0Connection)	Raw Payload. Need Developer to design Dash Board	
General MQTT (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H3.2GeneralA0MQTTA0Connection)	Raw Payload. Need Developer to design Dash Board	

1D Version: This version has 1NCE SIM card pre-installed and configure to send value to DataCake. User Just need to select the sensor type in DataCake and Activate RS485-NB and user will be able to see data in DataCake. See here for DataCake Config Instruction (http://wiki.dragino.com/xwiki/bin/view/Main/General%20Configure%20to%20Connect%20to%20IoT%20server%20for%20-NB%20%26%20-NS%20NB-IoT%20models/#H3.4Datacake) .

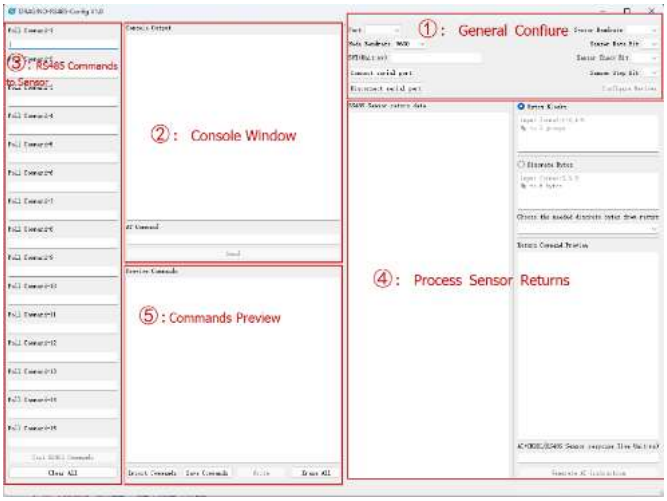
2.2 Configure Device to Read RS485 Sensors

There are plenty of RS485 and TTL level devices in the market and each device has different commands to read the valid data. To support these devices in most flexible, RS485-NB supports flexible command set. User can use Dragino RS485 Tool (https://www.dropbox.com/sh/us9qecn39fwt8n1/AABREdqUCzEmJMRrfuWuXasoa?dl=0) , AT Commands or LoRaWAN Downlink (http://wiki.dragino.com/xwiki/bin/view/Main/User%20Manual%20for%20LoRaWAN%20End%20Nodes/RS485-LB_Waterproof_RS485UART_to_LoRaWAN_Converter/#H3.5ConfigureRS485-LBviaATorDownlink) Command to configure how RS485-NB should read the sensor and how to handle the return from RS485 or TTL sensors.

2.2.1 Method 1 -- via RS485 Configure Tool

Use the RS485 Configure tool is the recommand method. Please see the instruction of how to use the tool:

- **RS485 Configure Tool Instruction (http://wiki.dragino.com/xwiki/bin/view/Main/RS485_Configure_Tool/)**



2.2.2 Method 2 -- via AT Commands

2.2.2.1 Configure UART settings for RS485 or TTL communication

RS485-NB can connect to either RS485 sensors or TTL sensor. User need to specify what type of sensor need to connect.

1. RS485-MODBUS mode:

AT+MOD=1 // Support RS485-MODBUS type sensors. User can connect multiply RS485 , Modbus sensors to the A / B pins.

2. TTL mode:

AT+MOD=2 // Support TTL Level sensors, User can connect one TTL Sensor to the TXD/RXD/GND pins.

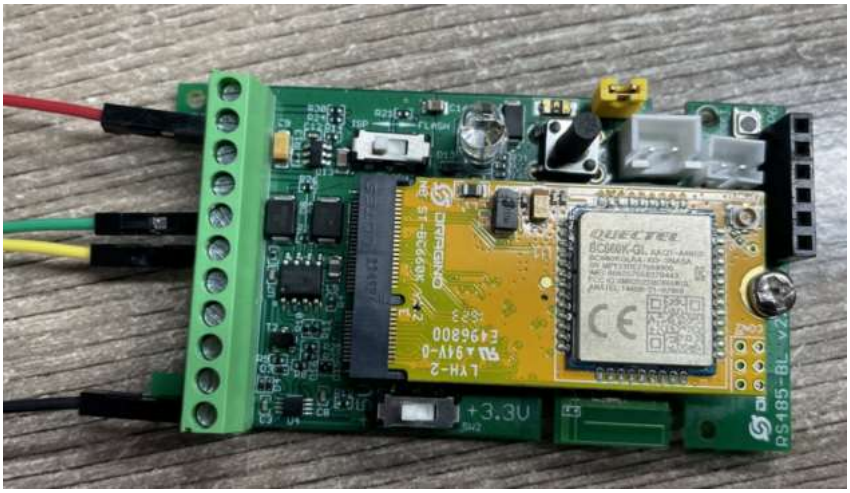
RS485-NB default UART settings is **9600, no parity, stop bit 1,data bit 8**. If the sensor has a different settings, user can change the RS485-NB setting to match.

AT Commands	Description	Example
AT+BAUDR	Set the baud rate. Default Value is: 9600.	AT+BAUDR=9600 Options: (1200,2400,4800,14400,19200,115200)
AT+PARITY	Set UART parity. Default Value is: no parity.	AT+PARITY=0 Option: 0: no parity, 1: odd parity, 2: even parity
AT+STOPBIT	Set serial stopbit Default Value is: 1bit.	AT+STOPBIT=0 for 1 bit AT+STOPBIT=1 for 1.5 bits AT+STOPBIT=2 for 2 bits
AT+DATABIT	Set serial databit. Default Value is: 8bits.	AT+DATABIT=7 for 7 bits AT+DATABIT=8 for 8 bits

Example(Soil three-parameter detector):

Wiring the UART sensor

GND <--------> GND
TX <--------> RX
RX <--------> TX
VCC <--------> 3.3/5V



Set the correct configuration:

AT+BAUDR=9600

AT+PARITY=0

AT+STOPBIT=1

AT+DATABIT=8

If the sensor needs 5v. Need to move the switch position to 5v and then use the command **AT+5VT=30000**

Configuration read command:

AT+CFGDEV=FE 03 00 00 00 03 11 C4,0

FE: Station address

03: Function code

00 00: Register start address

00 03: Number of registers

11 04: Check code

```
AT+CFGDEV=FE 03 00 00 00 03 11 C4,0

AT+CFGDEV=fe 03 00 00 00 03 11 c4 ,0
RETURN DATA:
fe 03 06 00 00 09 49 00 00 b6 cb

OK
```

Use AT+COMMAND1 to set it as a command, and use AT+DATACUT1 to intercept the bytes I need

```
AT+PAYVER=1
AT+MBFUN=0
AT+COMMAND1=fe 03 00 00 00 03 11 c4 ,0    AT+SEARCH1=0,0    AT+DATACUT1=11,2,4~9    AT+CMDDL1=0
AT+COMMAND2=0,0    AT+SEARCH2=0,0    AT+DATACUT2=0,0,0    AT+CMDDL2=0
AT+COMMAND3=0,0    AT+SEARCH3=0,0    AT+DATACUT3=0,0,0    AT+CMDDL3=0
AT+COMMAND4=0,0    AT+SEARCH4=0,0    AT+DATACUT4=0,0,0    AT+CMDDL4=0
AT+COMMAND5=0,0    AT+SEARCH5=0,0    AT+DATACUT5=0,0,0    AT+CMDDL5=0
AT+COMMAND6=0,0    AT+SEARCH6=0,0    AT+DATACUT6=0,0,0    AT+CMDDL6=0
AT+COMMAND7=0,0    AT+SEARCH7=0,0    AT+DATACUT7=0,0,0    AT+CMDDL7=0
AT+COMMAND8=0,0    AT+SEARCH8=0,0    AT+DATACUT8=0,0,0    AT+CMDDL8=0
AT+COMMAND9=0,0    AT+SEARCH9=0,0    AT+DATACUT9=0,0,0    AT+CMDDL9=0
AT+COMMANDA=0,0    AT+SEARCHA=0,0    AT+DATACUTA=0,0,0    AT+CMDDLA=0
AT+COMMANDB=0,0    AT+SEARCHB=0,0    AT+DATACUTB=0,0,0    AT+CMDDLB=0
AT+COMMANDC=0,0    AT+SEARCHC=0,0    AT+DATACUTC=0,0,0    AT+CMDDLC=0
AT+COMMANDD=0,0    AT+SEARCHD=0,0    AT+DATACUTD=0,0,0    AT+CMDDLD=0
AT+COMMANDE=0,0    AT+SEARCHE=0,0    AT+DATACUTE=0,0,0    AT+CMDDLE=0
AT+COMMANDF=0,0    AT+SEARCHF=0,0    AT+DATACUTF=0,0,0    AT+CMDDLF=0

OK
```

upload payload:

```
[41270]*****Upload start:0*****

CMD1    = fe 03 00 00 00 03 11 c4
RETURN1  = fe 03 06 00 00 09 49 00 00 b6 cb
Payload  = 01 00 00 09 49 00 00

[41928]BAT:3.584 V
[45458]Open a Socket Service successfully
[52516]Datagram is sent by RF
[53550]Send complete
```

2.2.3 Configure sensors

Some sensors might need to configure before normal operation. User can configure such sensor via PC or through RS485-NB AT Commands **AT+CFGDEV**.

When user issue an **AT+CFGDEV** command, Each **AT+CFGDEV** equals to send a command to the RS485 or TTL sensors. This command will only run when user input it and won't run during each sampling.

AT Commands	Description	Example
AT+CFGDEV	This command is used to configure the RS485/TTL devices; they won't be used during sampling. AT+CFGDEV=xx xx xx xx xx xx xx xx xx xx xx xx, mm: 0: no CRC, 1: add CRC-16/MODBUS in the end of this command	AT+CFGDEV=xx xx xx xx xx xx xx xx xx xx xx xx,m

2.2.4 Configure read commands for each sampling

Note: Please upgrade firmware to v1.0.3 or above. Otherwise, if a data error occurs in AT+COMMANDx check crc mode, the device will enter an inoperable state.

RS485-NB is a battery powered device; it will sleep most of time. And wake up on each period and read RS485 / TTL sensor data and uplink.

During each sampling, we need to confirm what commands we need to send to the sensors to read data. After the RS485/TTL sensors send back the value, it normally includes some bytes and we only need a few from them for a shorten payload.

To save the LoRaWAN network bandwidth, we might need to read data from different sensors and combine their valid value into a short payload.

This section describes how to achieve above goals.

During each sampling, the RS485-NB can support 15 commands to read sensors. And combine the return to one or several uplink payloads.

Command from RS485-NB to Sensor:

RS485-NB can send out pre-set max 15 strings via **AT+COMMAD1**, **ATCOMMAND2**,..., to **AT+COMMANDF** . All commands are of same grammar.

Handle return from sensors to RS485-NB:

After RS485-NB send out a string to sensor, RS485-NB will wait for the return from RS485 or TTL sensor. And user can specify how to handle the return, by **AT+DATACUT or AT+SEARCH commands**

- **AT+DATACUT**

When the return value from sensor have fix length and we know which position the valid value we should get, we can use AT+DATACUT command.

- **AT+SEARCH**

When the return value from sensor is dynamic length and we are not sure which bytes the valid data is, instead, we know what value the valid value following. We can use AT+SEARCH to search the valid value in the return string.

Define wait timeout:

Some RS485 device might has longer delay on reply, so user can use AT+CMDDL to set the timeout for getting reply after the RS485 command is sent. For example, AT+CMDDL1=1000 to send the open time to 1000ms

After we got the valid value from each RS485 commands, we need to combine them together with the command **AT+DATAUP**.

Examples:

Below are examples for the how above AT Commands works.

AT+COMMANDx : This command will be sent to RS485/TTL devices during each sampling, Max command length is 14 bytes. The grammar is:

AT+COMMANDx=xx xx xx xx xx xx xx xx xx xx xx xx,m

xx xx xx xx xx xx xx xx xx xx xx xx: The RS485 command to be sent

m: 0: no CRC, 1: add CRC-16/MODBUS in the end of this command

For example, if we have a RS485 sensor. The command to get sensor value is: 01 03 0B B8 00 02 46 0A. Where 01 03 0B B8 00 02 is the Modbus command to read the register 0B B8 where stored the sensor value. The 46 0A is the CRC-16/MODBUS which calculate manually.

In the RS485-NB, we should use this command AT+COMMAND1=01 03 0B B8 00 02,1 for the same.

If a single command exceeds 14 bytes, you can use the command splicing function.

When AT+CMDDLx=1, the commands of AT+COMMANDx and AT+COMMAND(x+1) will be merged.

Examples: To send 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F data it should be configured:

AT+COMMAND1=00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D,0

AT+COMMAND1=1

AT+COMMAND2=0E 0F,0

AT+SEARCHx: This command defines how to handle the return from AT+COMMANDx.

AT+SEARCHx=aa,xx xx xx xx xx

- **aa: 1: prefix match mode; 2: prefix and suffix match mode**
- **xx xx xx xx xx: match string. Max 5 bytes for prefix and 5 bytes for suffix**

Examples:

1) For a return string from AT+COMMAND1: 16 0c 1e 56 34 2e 30 58 5f 36 41 30 31 00 49

If we set AT+SEARCH1=1,1E 56 34. (max 5 bytes for prefix)

The valid data will be all bytes after 1E 56 34 , so it is **2e 30 58 5f 36 41 30 31 00 49**

```
CMD1     = 11 01 1e d0
SEARCH1  = 1e 56 34
RETURN1  = 2e 30 58 5f 36 41 30 31 00 49
Payload  = 8d 2d 01 2e 30 58 5f 36 41 30 31 00 49
```

2) For a return string from AT+COMMAND1: 16 0c 1e 56 34 2e 30 58 5f 36 41 30 31 00 49

If we set AT+SEARCH1=2, 1E 56 34+31 00 49

Device will search the bytes between 1E 56 34 and 31 00 49. So it is **2e 30 58 5f 36 41 30**

```
CMD1     = 11 01 1e d0
SEARCH1  = 1e 56 34
RETURN1  = 2e 30 58 5f 36 41 30 31 00 49
Payload  = 8d 2d 01 2e 30 58 5f 36 41 30 31 00 49
```

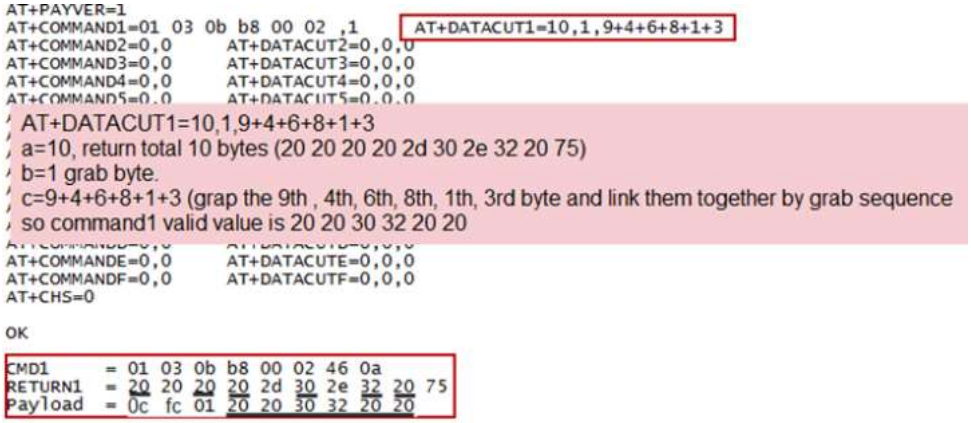
AT+DATACUTx : This command defines how to handle the return from AT+COMMANDx, max return length is 100 bytes.

AT+DATACUTx=a,b,c

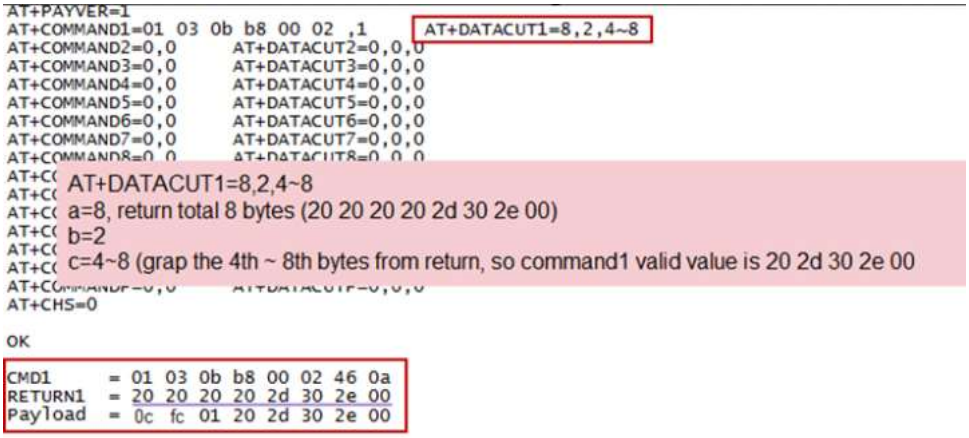
- **a: length for the return of AT+COMMAND**
- **b:1: grab valid value by byte, max 6 bytes. 2: grab valid value by bytes section, max 3 sections.**
- **c: define the position for valid value.**

Examples:

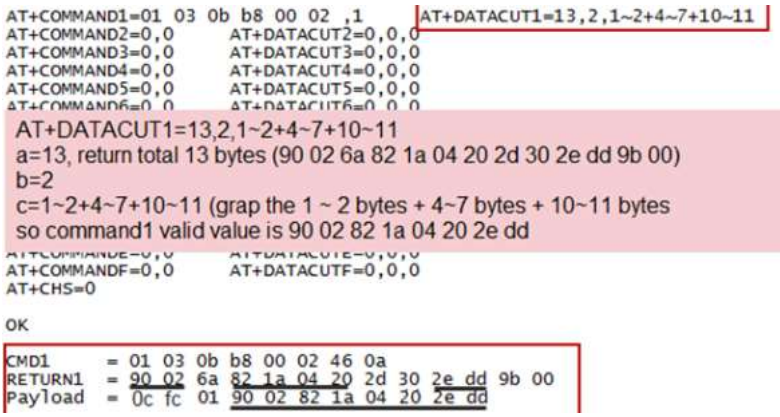
- **Grab bytes:**



- **Grab a section.**



- **Grab different sections.**



Note:

AT+SEARCHx and **AT+DATACUTx** can be used together, if both commands are set, RS485-NB will first process AT+SEARCHx on the return string and get a temporary string, and then process AT+DATACUTx on this temporary string to get the final payload. In this case, AT+DATACUTx need to set to format **AT+DATACUTx=0,xx,xx** where the return bytes set to **0**.

Example:

AT+COMMAND1=11 01 1E D0,0

AT+SEARCH1=1,1E 56 34

AT+DATACUT1=0,2,1~5

Return string from AT+COMMAND1: 16 0c 1e 56 34 2e 30 58 5f 36 41 30 31 00 49

String after SEARCH command: 2e 30 58 5f 36 41 30 31 00 49

Valid payload after DataCUT command: 2e 30 58 5f 36

CMD1 = 11 01 1e d0
SEARCH1 = 1e 56 34
RETURN1 = 2e 30 58 5f 36 41 30 31 00 49
Payload = 8d 2d 01 2e 30 58 5f 36

2.2.5 Uplink when the sensor is not responding

1) When **AT+MOD=1**, if the data intercepted by **AT+DATACUT** or **AT+MBFUN** is empty, it will display **NULL**, and the payload will be filled with **n FFs**.

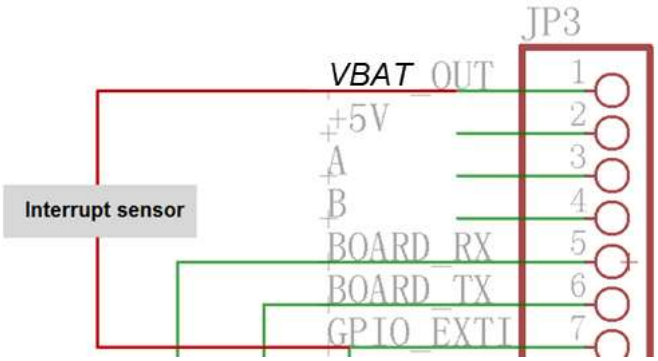
```
[146157]Signal Strength:23
[149091]*****Upload start:4*****

CMD1    = fe 03 00 00 00 03 11 c4
RETURN1 = NULL

Payload = 01 ff ff ff ff ff ff

[149645]BAT:3.594 V
[156683]Datagram is sent by RF
[157718]Send complete
[158743]*****End of upload*****
```

2) When **AT+MOD=2**, if the data intercepted by **AT+DATACUT** or **AT+MBFUN** is empty, it will display **NULL**, and the payload will be filled with **n 00s**.

```
[172119]Signal Strength:23
[175053]*****Upload start:5*****

CMD1    = fe 03 00 00 00 03 11 c4
RETURN1 = NULL

Payload = 01 00 00 00 00 00 00

[176587]BAT:3.591 V
[183623]Datagram is sent by RF
[184658]Send complete
[185683]*****End of upload*****
```

2.2.6 Uplink on Interrupt

Put the interrupt sensor between 3.3v_out and GPIO ext.



AT+INTMOD=0 Disable Interrupt.(Default Value)

AT+INTMOD=1 Interrupt trigger by rising or falling edge.

AT+INTMOD=2 Interrupt trigger by falling edge.

AT+INTMOD=3 Interrupt trigger by rising edge.

2.3 Payload Types

To meet different server requirement, RS485-NB supports different payload type.

Includes:

- General JSON format payload. (Type=5)
- HEX format Payload. (Type=0)

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

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

2.3.1 General Json Format(Type=5)

This is the General Json Format. As below:

{"IMEI":"866207053462705","Model":"RS485-NB","Payload":01111122223333,"battery":3.581,"signal":23}

Notice:The maximum number of bytes in the Payload is 1 payload version number plus 687 RS485 data.

2.3.2 HEX format Payload(Type=0)

This is the HEX Format. As below:

f866207053462705 1764 0df4 11 00 00 64f02eb0 01 111122223333

HEX Format for RS485-NB								
f866207058386669	1764	0df4	11	00	00	64f02eb0	01	Length depends on the return from the commands
f+IMEI	Version	BAT	Singal	GPIO_EXIT Level	GPIO_EXIT Flag	Timestamp	payload_version	RS485_sensor_data
8 Bytes	12+n Bytes(The maximum n value is 710)							

Version:

These bytes include the hardware and software version.

Higher byte: Specify Sensor Model: 0x17 for RS485-NB

Lower byte: Specify the software version: 0x64=100, means firmware version 1.0.0

BAT (Battery Info):

Ex1: 0x0DF4 = 3572mV

Signal Strength:

NB-IoT Network signal Strength.

Ex1: 0x11 = 17

0 -113dBm or less

1 -111dBm

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

31 -51dBm or greater

99 Not known or not detectable

GPIO_EXIT Level :

GPIO_EXTI is used as Interrupt Pin.

Example:

01 (H): GPIO_EXTI pin is high level.

00 (L): GPIO_EXTI pin is low level.

GPIO_EXIT Flag:

This data field shows if this packet is generated by **Interrupt Pin** or not.

Note: The Interrupt Pin is a separate pin in the screw terminal.

Example:

0x00: Normal uplink packet.

0x01: Interrupt Uplink Packet.

TimeStamp:

Unit TimeStamp Example: 64F02EB0(H) = 1693462192(D)

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

Payload_version:

The payload version number is used to parse different decodes.

RS485_sensor_data:

The data returned by the RS485 sensor or UART sensor needs to be cut out the AT+DATACUTx and AT+MBFUN or AT+SEARCHx commands.

2.4 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+TDC=600 // Set Update Interval to 600s

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

2.5 Fast command to handle MODBUS device

The command is for fast configure to read Modbus devices. It is only valid for the devices which follow the MODBUS-RTU protocol (https://www.modbustools.com/modbus.html) .

AT+MBFUN has only two value:

- **AT+MBFUN=1**: Enable Modbus reading. And get response base on the MODBUS return

AT+MBFUN=1, device can auto read the Modbus function code: 01, 02, 03 or 04. AT+MBFUN has lower priority vs AT+DATACUT command. If AT+DATACUT command is configured, AT+MBFUN will be ignore.

- **AT+MBFUN=0**: Disable Modbus fast reading.

Example:

- AT+MBFUN=1 and AT+DATACUT1/AT+DATACUT2 are not configure (0,0,0).
- AT+COMMAND1= 01 03 00 10 00 08,1 --> read slave address 01 , function code 03, start address 00 01, quantity of registers 00 08.
- AT+COMMAND2= 01 02 00 40 00 10,1 --> read slave address 01 , function code 02, start address 00 40, quantity of inputs 00 10.

2.6 RS485 command timeout

Some Modbus device has slow action to send replies. This command is used to configure the RS485-NB to use longer time to wait for their action.

Default value: 0, range: 0 ~ 10 seconds

- **AT Command:**

AT+CMDDLaa=hex(bb cc)

Example:

AT+CMDDL1=1000 to send the open time to 1000ms

2.7 Clear RS485 Command

The AT+COMMANDx and AT+DATACUTx AT+SEARCHx and AT+CMDDLx settings are stored in special location, user can use below command to clear them.

- **AT Command:**

AT+CMDEAR=mm,nn mm: start position of erase ,nn: stop position of erase Etc. AT+CMDEAR=1,10 means erase AT+COMMAND1/AT+DATACUT1/AT+SEARCH1/AT+CMDDL1 to AT+COMMANDA/AT+DATACUTA/AT+SEARCHA/AT+CMDDLA.

2.8 Set Payload version

This is the first byte of the uplink payload. RS485-NB can connect to different sensors. User can set the PAYVER field to tell server how to decode the current payload.

- **AT Command:**

AT+PAYVER: Set PAYVER field = 1

2.9 Control output power duration

User can set the output power duration before each sampling.

- **AT Command:**

Example:

AT+3V3T=1000 // 3V3 output power will open 1s before each sampling.

AT+3V3T=0 // Normally open 3V3 power supply.

AT+3V3T=65535 // Normally closed 3V3 power supply.

AT+5VT=1000 // +5V output power will open 1s before each sampling.

AT+5VT=0 // Normally closed +5V power supply.

AT+5VT=65535 // Normally open +5V power supply.

2.10 Get sensor value

- **AT Command:**

AT+GETSENSORVALUE=0 // The serial port gets the reading of the current sensor

AT+GETSENSORVALUE=1 // The serial port gets the current sensor reading and uploads it.

2.11 +3V3 Output

RS485-NB has a Controllable +3V3 output, user can use this output to power external sensor.

The +3V3 output will be valid for every sampling. RS485-NB will enable +3V3 output before all sampling and disable the +3V3 after all sampling.

The +3V3 output time can be controlled by AT Command.

AT+3V3T=1000

Means set +3v3 valid time to have 1000ms. So, the real +3v3 output will actually have 1000ms + sampling time for other sensors.

By default, the AT+3V3T=0. This is a special case, means the +3V3 output is always on at any time

2.12 +5V Output

RS485-NB has a Controllable +5V output, user can use this output to power external sensor.

The +5V output will be valid for every sampling. RS485-NB will enable +5V output before all sampling and disable the +5v after all sampling.

The 5V output time can be controlled by AT Command.

AT+5VT=1000

Means set 5V valid time to have 1000ms. So, the real 5V output will actually have 1000ms + sampling time for other sensors.

By default, the AT+5VT=0. If the external sensor which require 5v and require more time to get stable state, user can use this command to increase the power ON duration for this sensor.

2.13 Switch Jumper

Switch Jumper	Feature
SW1	ISP position: Upgrade firmware via UART Flash position: Configure device, check running status.
SW2	5V position: set to compatible with 5v I/O. 3.3v position: set to compatible with 3.3v I/O.

+3.3V: is always ON

+5V: Only open before every sampling. The time is by default, it is **AT+5VT=0**.

Note: If SW2 is at +5V and AT+5VT=0, work mode 2 will not be able to send data.

2.14 Clock logging (Since firmware version v1.0.7)

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.

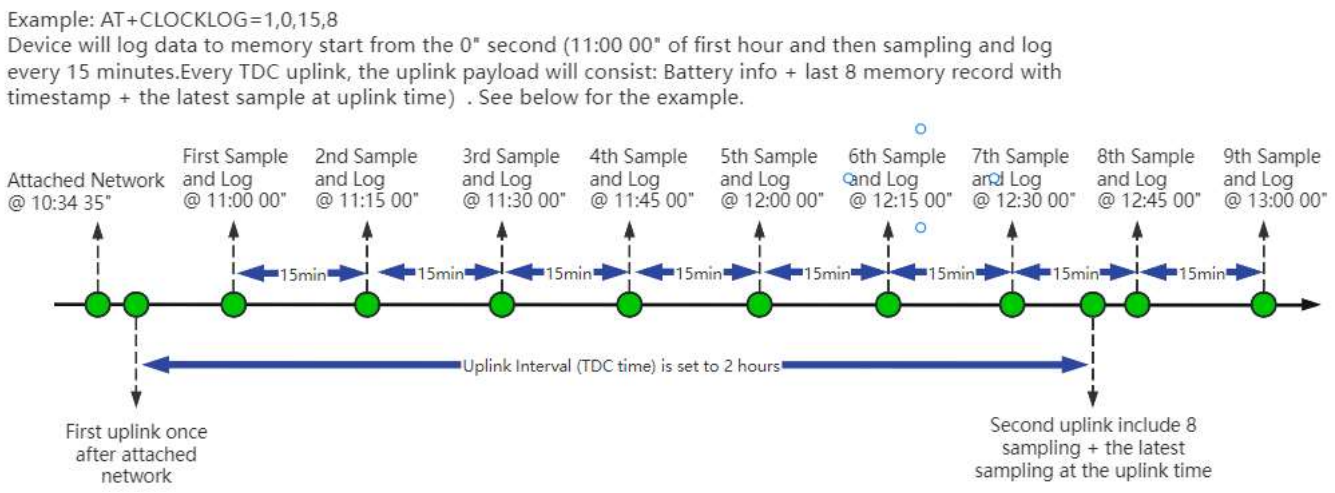
- **AT Command: AT+CLOCKLOG=a,b,c,d**

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)**

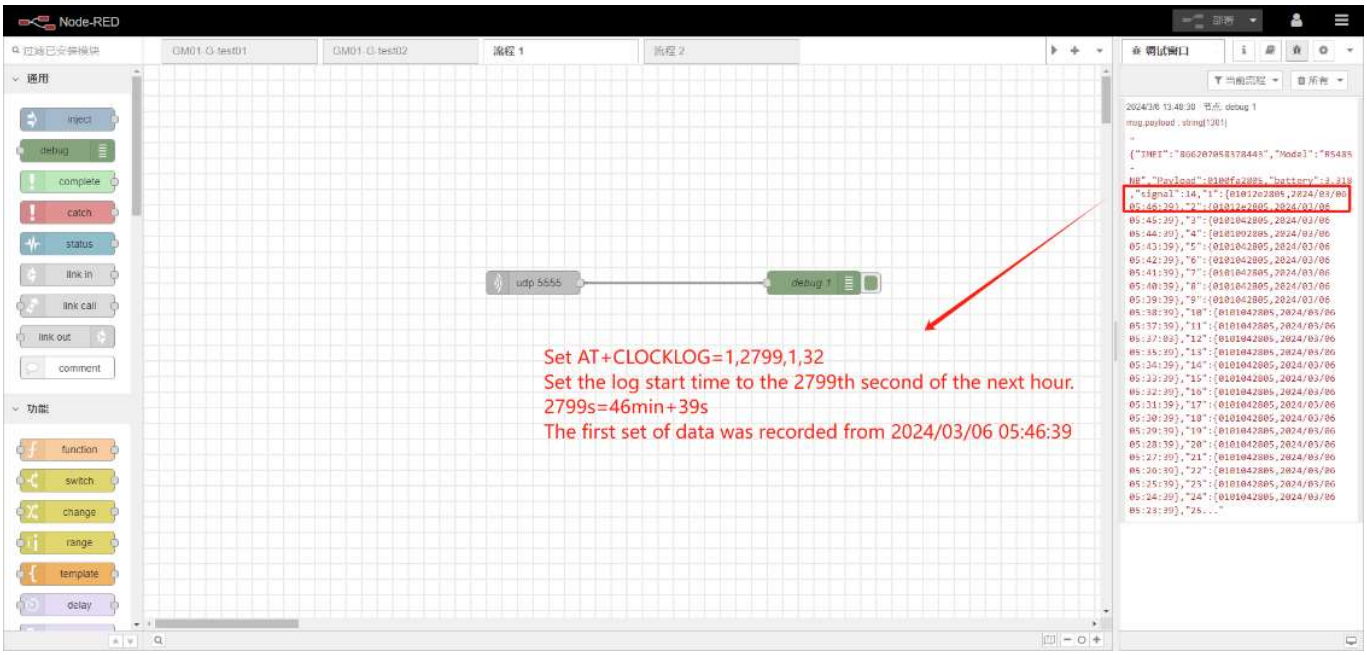
d: How many entries should be uplink on every TDC **(max 32)**



Example:

AT+CLOCKLOG=1,2799,1,32

Device will log data to memory start from 2799' and then log every 1 minutes. And every TDC uplink, the uplink payload will consist: Battery info + last 32 memory record (payload+Timestamp).

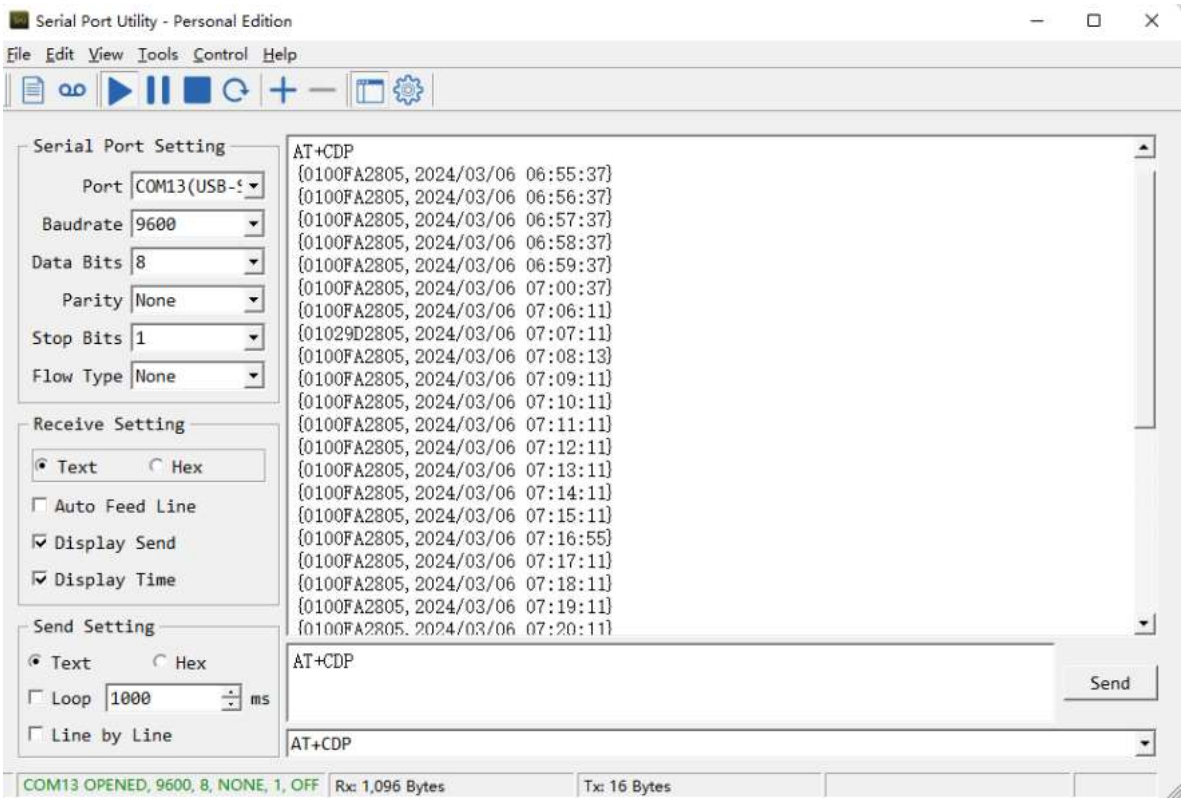


Note: Users need to synchronize the server time before configuring this command. If the server time is not synchronized before this command is configured, the command takes effect only after the node is reset.

2.15 Example Query saved historical records

- **AT Command: AT+CDP**

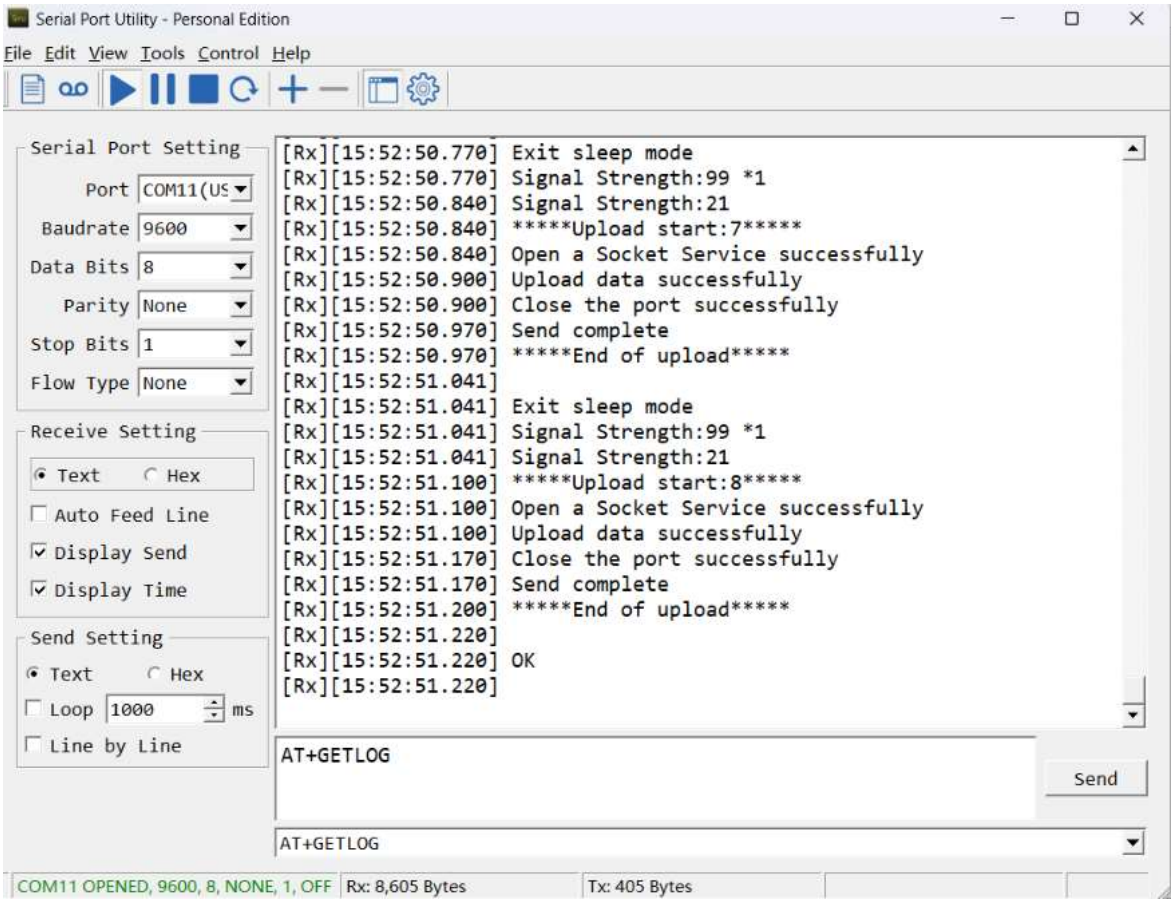
This command can be used to search the saved history, recording up to 32 groups of data, each group of historical data contains a maximum of 100 bytes.



2.16 Uplink log query

- **AT Command: AT+GETLOG**

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



3. Configure RS485-NB

3.1 Configure Methods

RS485-NB supports below configure method:

- AT Command via Bluetooth Connection (**Recommended**): BLE Configure Instruction (http://wiki.dragino.com/xwiki/bin/view/Main/BLE%20Bluetooth%20Remote%20Configure/) .
- AT Command via UART Connection : See UART Connection (http://wiki.dragino.com/xwiki/bin/view/Main/UART%20Access%20for%20LoRa%20ST%20v4%20base%20model/#H2.4UARTConnectionforRS485-BLbasemotherboard) .

3.2 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+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+SERVADDR : Server Address

AT+MOD: Get or Set work mode

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

AT+3V3T : Get or Set extend the time of 3V3 power

AT+INTMOD : Get or Set the trigger interrupt mode (0:input,1:falling or rising,2:falling,3:rising)

AT+BAUDR: Get or Set baudr of uart

AT+DATABIT: Get or Set databit(7:7 bit,8:8 bit) of uart

AT+PARITY: Get or Set parity(0:none,1:odd,2:even) of uart

AT+STOPBIT: Get or Set stopbit(0:1 bit,1:1.5 bit,2:2 bit) of uart

AT+CMDEAR: Erase command(number of begin to number of ending)

AT+PAYVER: Get or Set payload version

AT+MBFUN: Get or Set Modbus Funtion code(0:none,1:01 or 02,2:03 or 04)for automatic cut

AT+COMMAND: Send data of command1-15 to RS485 deceive for payload

AT+SEARCH: Retrieve the hexadecimal character that appears in command 1-15

AT+DATACUT: Cut receive data after use command1-15

AT+CMDDL: Set delay timeout after send rs485 command1-15

AT+CFGDEV: Send data to RS485 deceive

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

4. Battery & Power Consumption

RS485-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
(http://wiki.dragino.com/xwiki/bin/view/Main/How%20to%20calculate%20the%20battery%20life%20of%20Dragino%20sensors%3F/) .

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 (https://www.dropbox.com/sh/eywl6agsdcwrn4q/AAD3eMki9zLj2GZtndfHkO_Ra?dl=0)**

Methods to Update Firmware:

- (Recommended way) OTA firmware update via BLE: **Instruction** (http://wiki.dragino.com/xwiki/bin/view/Main/BLE_Firmware_Update_NB_Sensors_BC660K-GL/) .
- Update through UART TTL interface : **Instruction (http://wiki.dragino.com/xwiki/bin/view/Main/UART_Access_for_NB_ST_BC660K-GL/#H4.2UpdateFirmware28Assumethedevicealreadyhaveabootloader29)** .

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 (https://www.dropbox.com/sh/5f6ssda5fum8rvs/AABT68l8ZzWOvZ5eg2qwOoFda?dl=0)

6.2 How many RS485-Slave can RS485-NB connects?

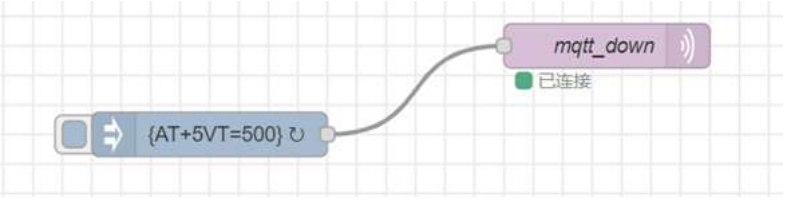
The RS485-NB can support max 32 RS485 devices. Each uplink command of RS485-NB can support max 16 different RS485 command. So RS485-NB can support max 16 RS485 devices pre-program in the device for uplink. For other devices no pre-program, user can use the downlink message (type code 0xA8) to poll their info.

6.3 How to configure the device through the MQTT subscription function?

Subscription content: {AT COMMAND}

Example:

Setting AT+5VT=500 through Node-RED requires MQTT to send the content {AT+5VT=500}.



The serial port displays:

```
CMD1    = fe 03 00 00 00 03 11 c4
RETURN1  = fe 03 06 11 11 22 22 33 33 65 54
Payload  = 01 11 11 22 22 33 33

[42203]BAT:3.584 V
[47629]Opened the MQTT client network successfully
[51187]Successfully connected to the server
AT+PWRM2
[54905]Upload data successfully
[58443]Received downlink data:AT+5VT=500
[59330]Subscribe to topic successfully
[62877]Close the port successfully
[63916]Send complete
[64940]*****End of upload*****
```

6.4 How to Use RS485-NB to connect to RS232 devices?

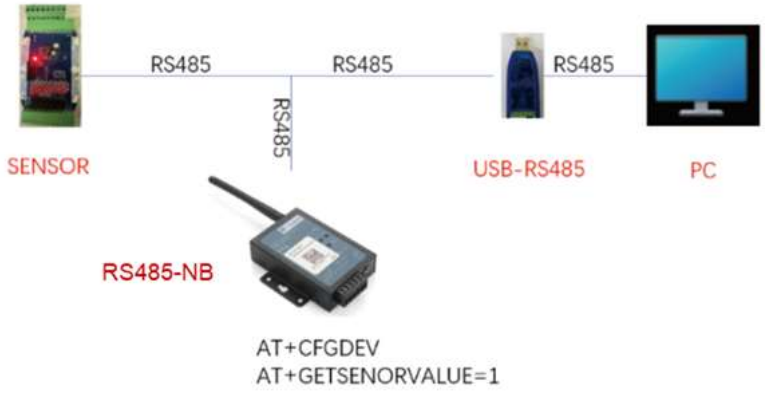
Use RS485-NB to connect to RS232 devices. - DRAGINO (http://wiki.dragino.com/xwiki/bin/view/Main/RS485%20to%20RS232/)

6.5 How to judge whether there is a problem with the set COMMAND

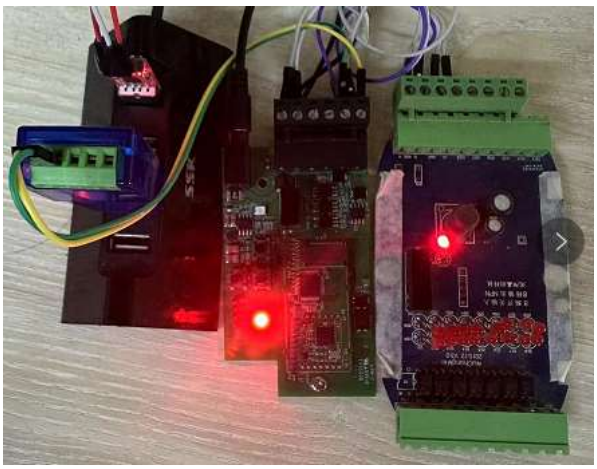
6.5.1 Introduce:

Users can use below the structure to fast debug the communication between RS485-NB. The principle is to put the PC in the RS485 network and sniff the packet between Modbus MTU and RS485-NB. We can **use this way to:**

1. Test if Modbus-MTU works with PC commands.
2. Check if RS485-NB sent the expected command to Mobus-MTU
3. Check if Modbus-MTU return back the expected result to RS485-NB.
4. If both b) and c) has issue, we can compare PC's output and RS485-NB output.

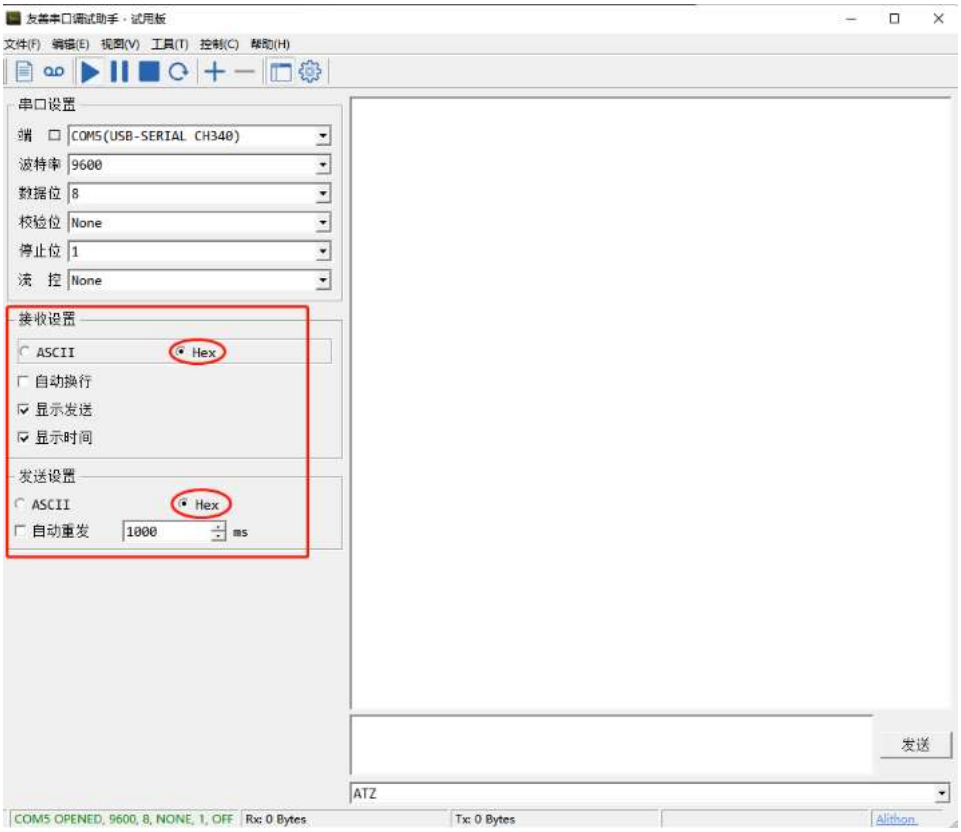


Example Connection:



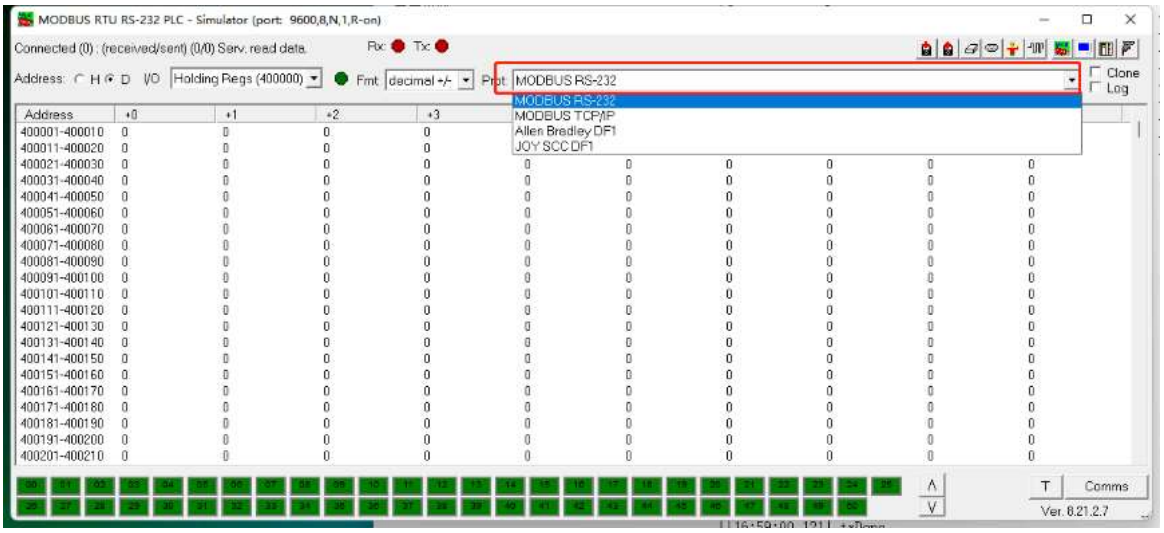
6.5.2 Set up PC to monitor RS485 network With Serial tool

Note: Receive and send set to hex mode

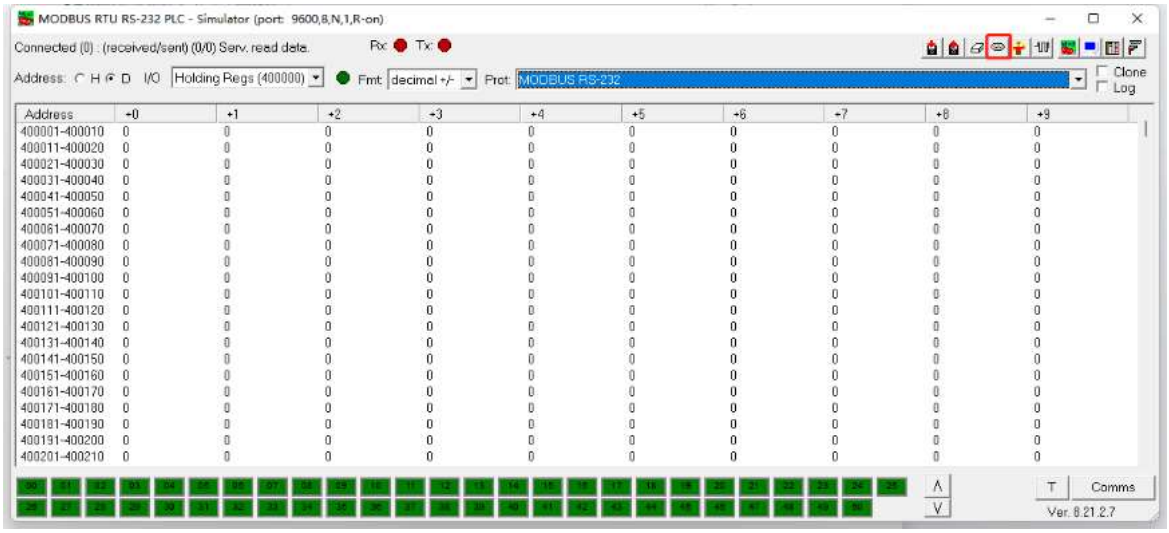


6.5.3 With ModRSsim2:

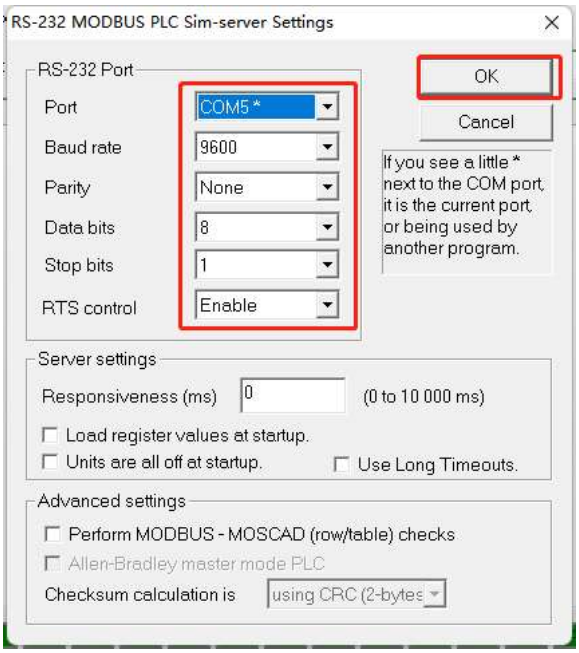
(1) Select serial port MODBUS RS-232



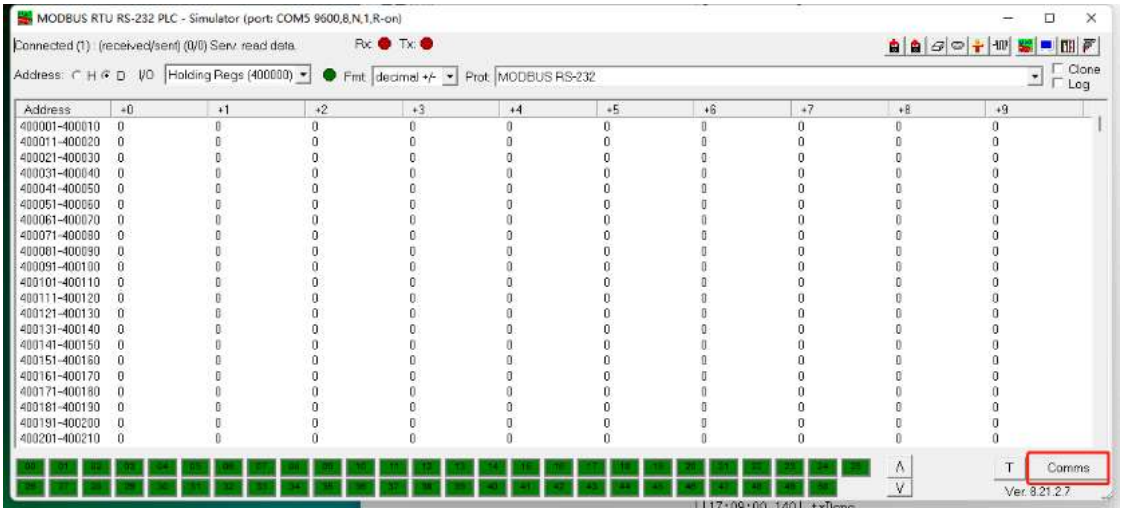
(2) Click the serial port icon



(3) After selecting the correct serial port and baud rate, click ok



(4) Click the comms.



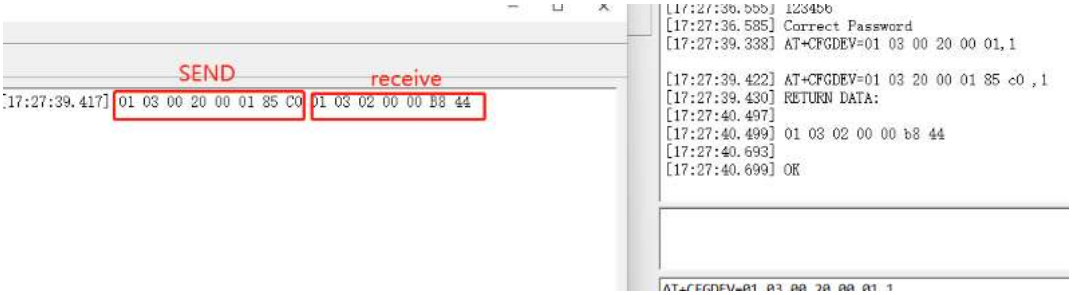
Run RS485-NB command and monitor if it is correct.

6.5.4 Example – Test the CFGDEV command

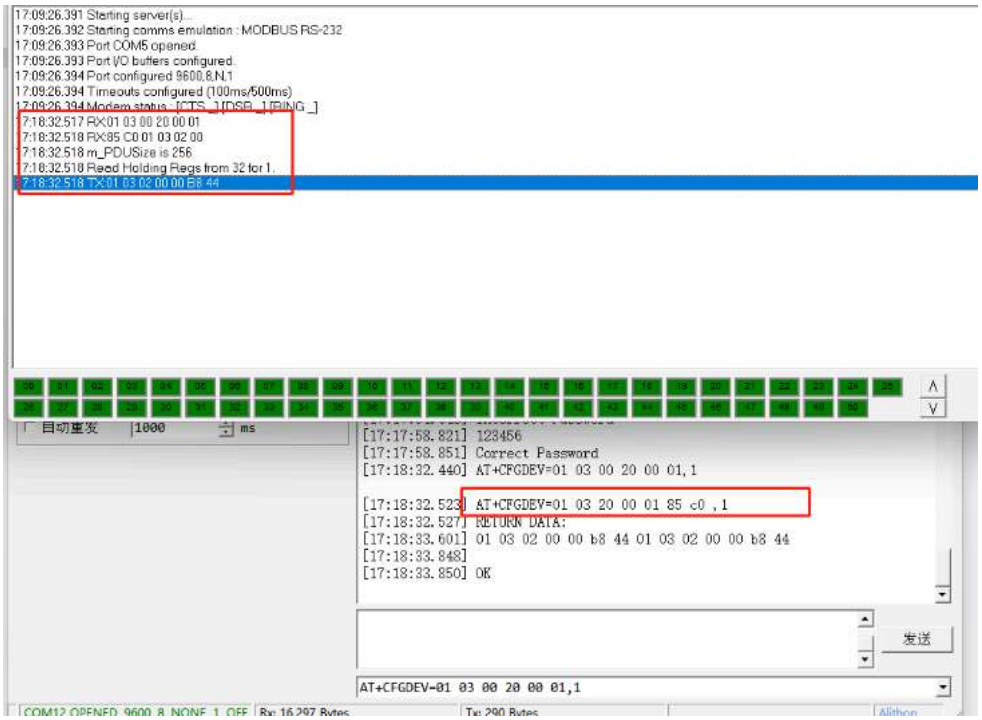
RS485-LN sent below command:

AT+CFGDEV=01 03 20 00 01 85 c0,1 to RS485 network, and PC is able to get this command and return commands from MTU to show in the serial tool.

We can see the output from the Serial port tool to analyze. And check if they are expected result.



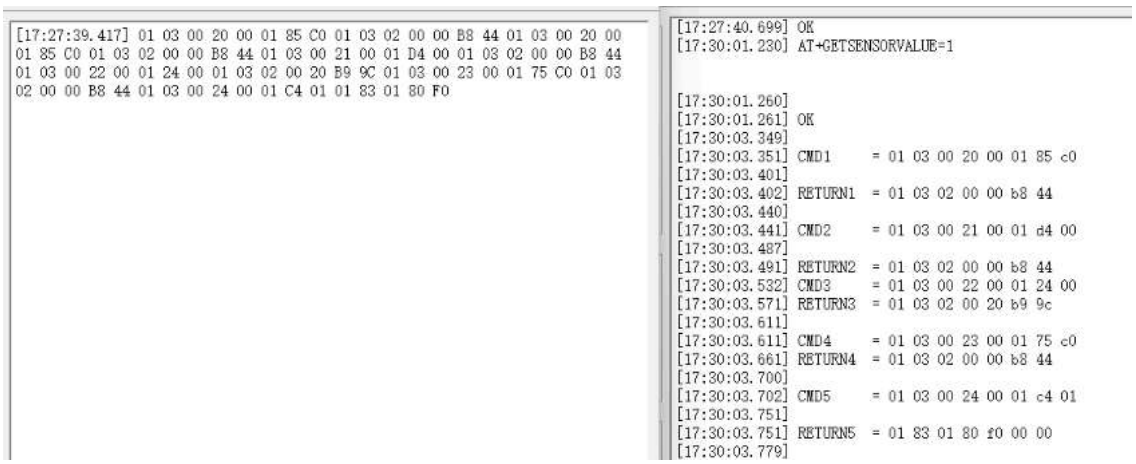
We can also use ModRSsim2 to see the output.



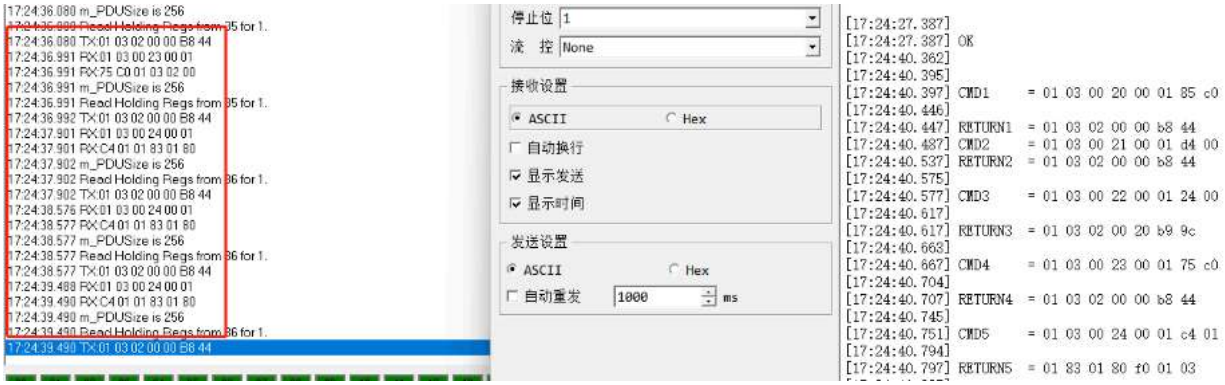
6.5.5 Example – Test CMD command sets.

Run **AT+SENSORVALUE=1** to test the CMD commands set in RS485-LN.

Serial port tool:



ModRSsim2:

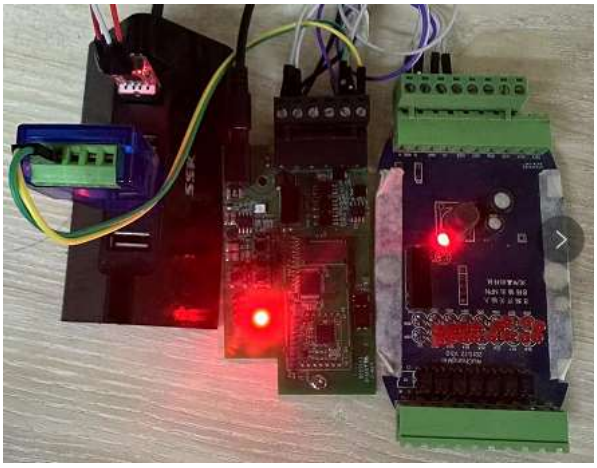


6.5.6 Test with PC

If there is still have problem to set up correctly the commands between RS485-LN and MTU. User can test the correct RS485 command set in PC and compare with the RS485 command sent out via RS485-LN. as long as both commands are the same, the MTU should return correct result.

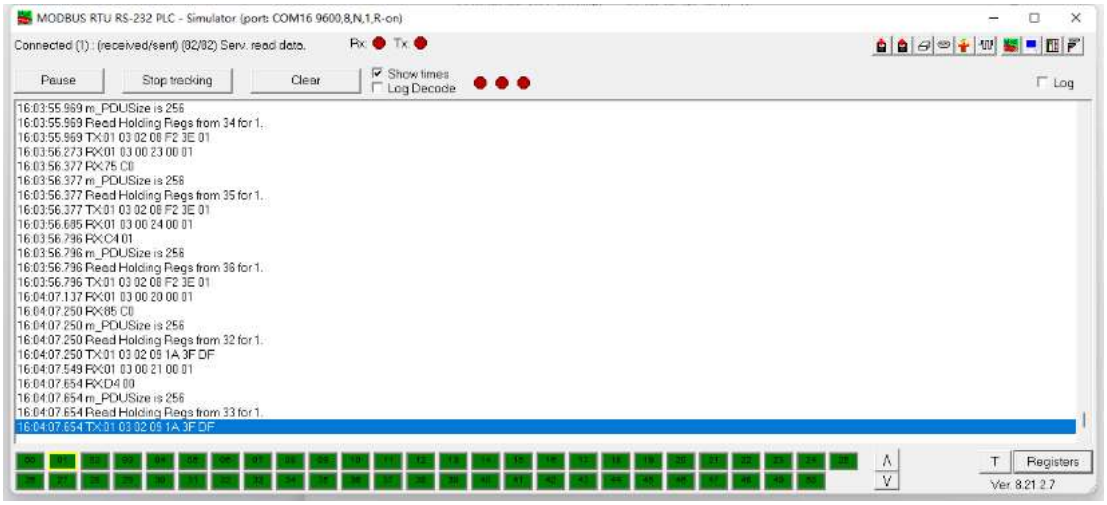
Or User can send the working commands set in PC serial tool to Dragino Support to check what should be configured in RS485-LN.

Connection method:



Link situation:

```
[16:03:35.274] CMD4     = 01 03 00 23 00 01 75 c0
[16:03:35.320] RETURN4  = 01 03 02 08 98 be 2e
[16:03:35.367] CMD5     = 01 03 00 24 00 01 c4 01
[16:03:35.412] RETURN5  = 01 03 02 08 98 be 2e
[16:03:35.444] Payload  = 01
[16:03:35.951]
[16:03:35.961] [134251]***** UpLinkCounter= 11 *****
[16:03:35.992] [134252]TX on freq 910700000 Hz at DR 0
[16:03:36.245] [134545]RX on freq 923300000 Hz at DR 8
[16:03:36.287] [134547]txDone
[16:03:41.239] [139539]RX on freq 924500000 Hz at DR 10
[16:03:41.281] [139571]RX on freq 923300000 Hz at DR 8
[16:03:41.311] [139573]rxTimeOut
[16:03:46.016]
[16:03:46.019] CMD1     = 01 03 00 20 00 01 85 c0
[16:03:46.065] RETURN1  = 01 03 02 08 c0 bf d4 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
[16:03:46.204] CMD2     = 01 03 00 21 00 01 d4 00
[16:03:46.252] RETURN2  = 01 03 02 08 c0 bf d4
[16:03:46.297] CMD3     = 01 03 00 22 00 01 24 00
[16:03:46.344] RETURN3  = 01 03 02 08 c0 bf d4
[16:03:46.374] CMD4     = 01 03 00 23 00 01 75 c0
[16:03:46.421] RETURN4  = 01 03 02 08 c0 bf d4
[16:03:46.467] CMD5     = 01 03 00 24 00 01 c4 01
[16:03:46.515] RETURN5  = 01 03 02 08 c0 bf d4
[16:03:46.545] Payload  = 01
[16:03:47.052]
[16:03:47.057] [145352]***** UpLinkCounter= 12 *****
[16:03:47.088] [145354]TX on freq 910500000 Hz at DR 0
[16:03:47.347] [145646]RX on freq 923300000 Hz at DR 8
[16:03:47.383] [145648]txDone
[16:03:52.340] [150639]RX on freq 923900000 Hz at DR 10
[16:03:52.376] [150671]RX on freq 923300000 Hz at DR 8
[16:03:52.407] [150673]rxTimeOut
```



7. Order Info

Part Number: **RS485-NB-XX**

XX:

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

YY: The grand connector hole size

- **M12**: M12 hole
- **M16**: M16 hole

8. Packing Info

Package Includes:

- RS485-NB NB-IoT Sensor Node 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

9. 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 (mailto:Support@dragino.cc) .