RS485-LN – RS485 to LoRaWAN Converter User Manual

last modified by Xiaoling

on 2023/04/26 17:49

Table of Contents

Table of Contents:

1. Introduction

1.1 What is RS485-LN RS485 to LoRaWAN Converter

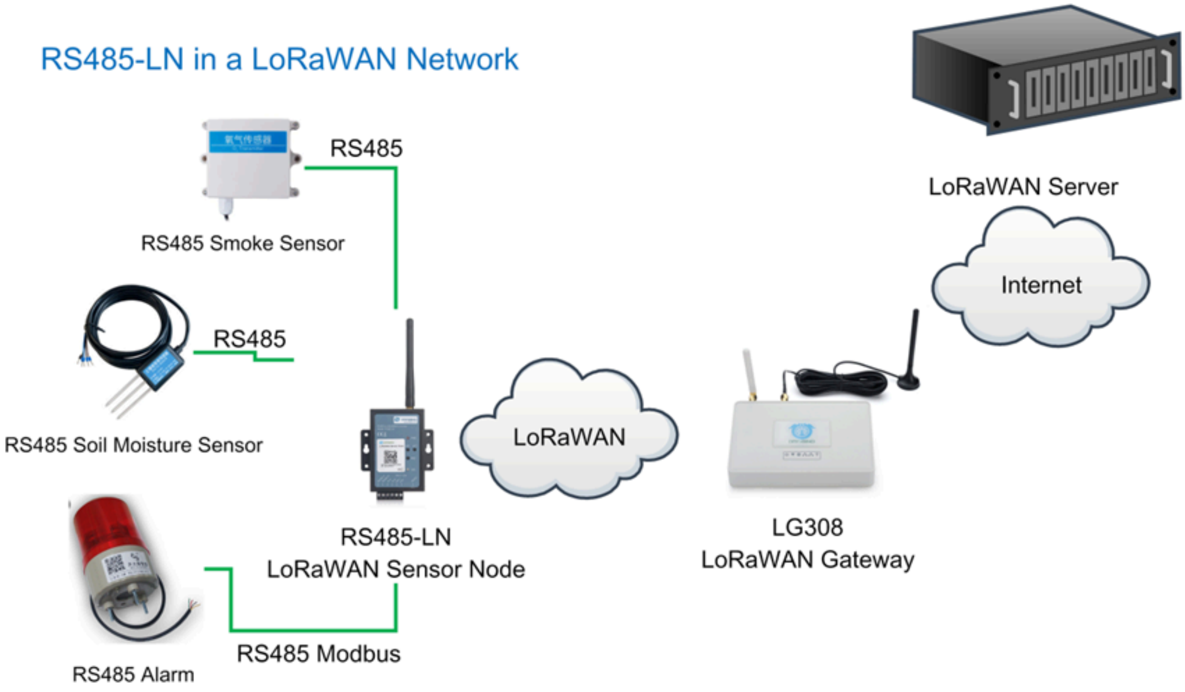
The Dragino RS485-LN is a **RS485 to LoRaWAN Converter**. It converts the RS485 signal into LoRaWAN wireless signal which simplify the IoT installation and reduce the installation/maintaining cost.

RS485-LN allows user to **monitor / control RS485 devices** and reach extremely long ranges. It provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption. It targets professional wireless sensor network applications such as irrigation systems, smart metering, smart cities, building automation, and so on.

For data uplink, RS485-LN sends user-defined commands to RS485 devices and gets the return from the RS485 devices. RS485-LN will process these returns according to user-define rules to get the final payload and upload to LoRaWAN server.

For data downlink, RS485-LN runs in LoRaWAN Class C. When there downlink commands from LoRaWAN server, RS485-LN will forward the commands from LoRaWAN server to RS485 devices.

Demo Dashboard for RS485-LN connect to two energy meters: https://app.datacake.de/dashboard/d/58844a26-378d-4c5a-aaf5-b5b5b153447a



1.2 Specifications

Hardware System:

- STM32L072xxxx MCU
- SX1276/78 Wireless Chip
- Power Consumption (exclude RS485 device):
  - Idle: 32mA@12v
  - 20dB Transmit: 65mA@12v

Interface for Model:

- RS485
- Power Input 7~ 24V DC.

LoRa Spec:

- Frequency Range:
  - Band 1 (HF): 862 ~ 1020 Mhz
  - Band 2 (LF): 410 ~ 528 Mhz
- 168 dB maximum link budget.
- +20 dBm - 100 mW constant RF output vs.
- +14 dBm high efficiency PA.
- Programmable bit rate up to 300 kbps.
- High sensitivity: down to -148 dBm.
- Bullet-proof front end: IIP3 = -12.5 dBm.
- Excellent blocking immunity.
- Low RX current of 10.3 mA, 200 nA register retention.
- Fully integrated synthesizer with a resolution of 61 Hz.
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
- Built-in bit synchronizer for clock recovery.
- Preamble detection.
- 127 dB Dynamic Range RSSI.
- Automatic RF Sense and CAD with ultra-fast AFC.
- Packet engine up to 256 bytes with CRC

1.3 Features

- LoRaWAN Class A & Class C protocol (default Class C)
- Frequency Bands: CN470/EU433/KR920/US915/EU868/AS923/AU915/IN865/RU864/MA869
- AT Commands to change parameters
- Remote configure parameters via LoRa Downlink
- Firmware upgradable via program port
- Support multiply RS485 devices by flexible rules
- Support Modbus protocol
- Support Interrupt uplink (Since hardware version v1.2)

1.4 Applications

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

1.5 Firmware Change log

RS485-LN Image files – Download link and Change log

1.6 Hardware Change log

v1.2: Add External Interrupt Pin.

v1.0: Release

2. Power ON Device

The RS485-LN can be powered by 7 ~ 24V DC power source. Connection as below

- Power Source VIN to RS485-LN VIN+
- Power Source GND to RS485-LN VIN-

Once there is power, the RS485-LN will be on.

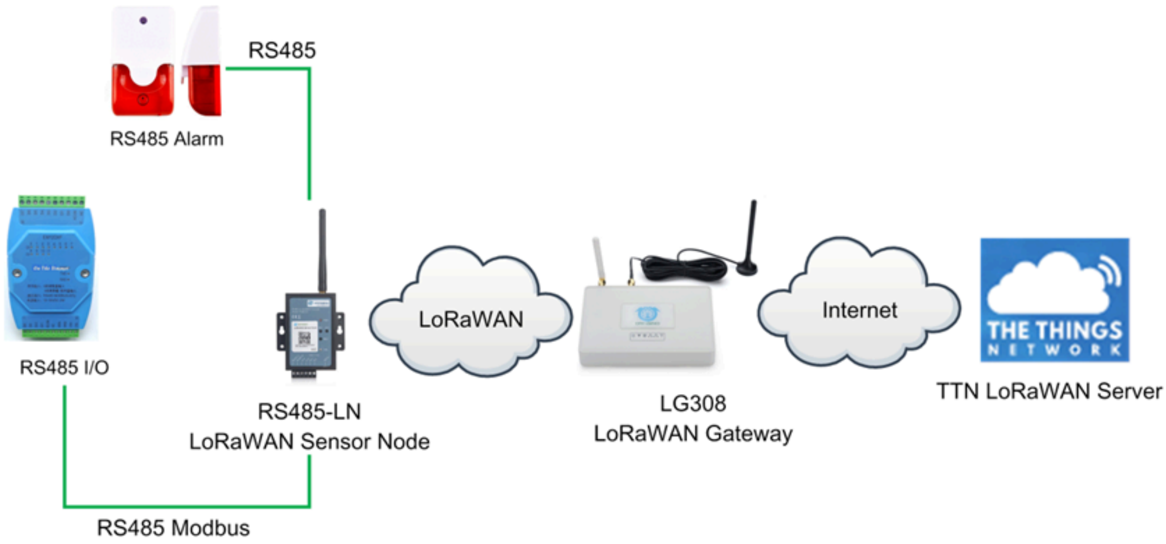
3. Operation Mode

3.1 How it works?

The RS485-LN is configured as LoRaWAN OTAA Class C mode by default. It has OTAA keys to join network. To connect a local LoRaWAN network, user just need to input the OTAA keys in the network server and power on the RS485-LN. It will auto join the network via OTAA.

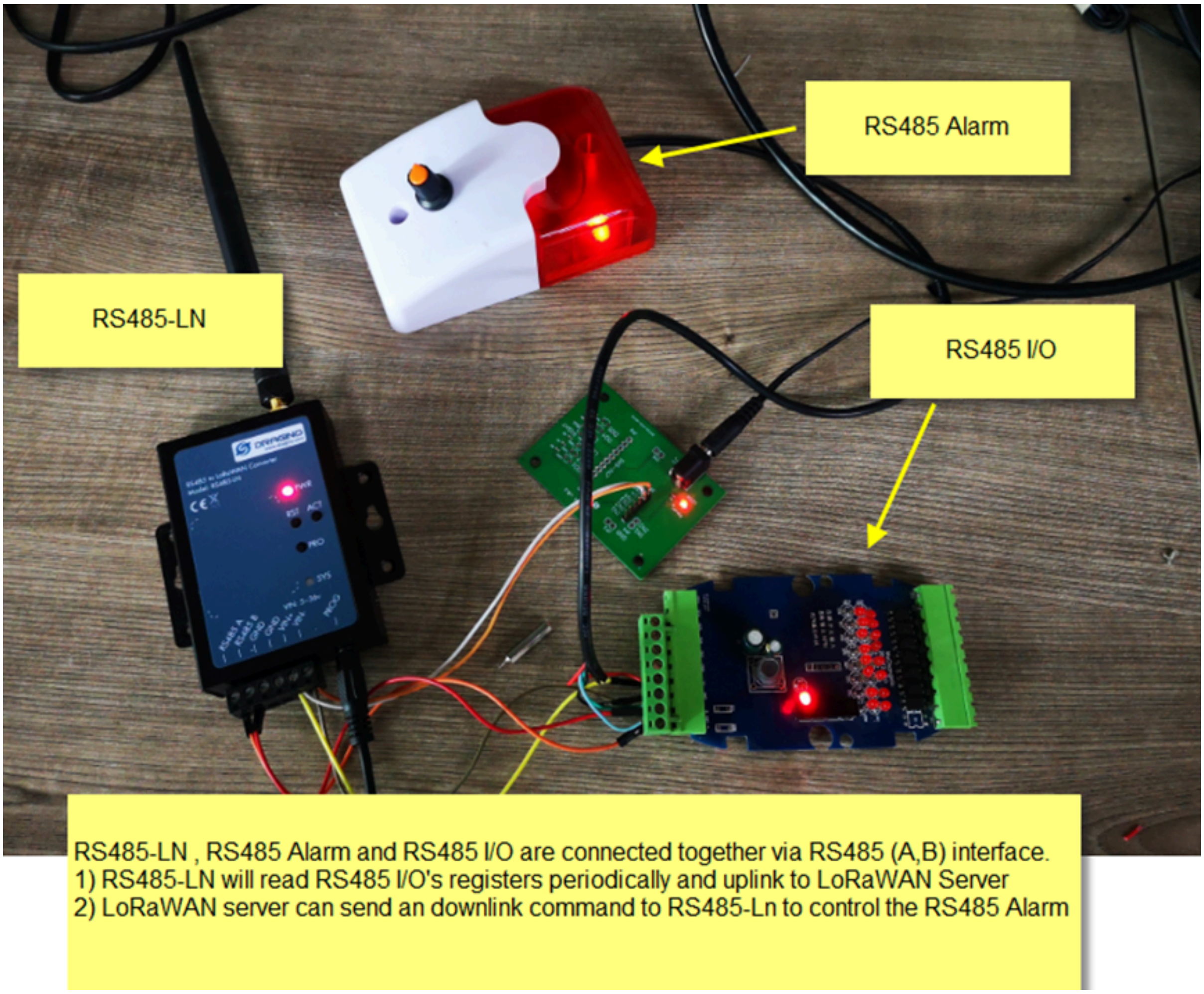
3.2 Example to join LoRaWAN network

Here shows an example for how to join the TTN V3 Network. Below is the network structure, we use LG308 as LoRaWAN gateway here.



The RS485-LN in this example connected to two RS485 devices for demonstration, user can connect to other RS485 devices via the same method. The connection is as below:

485A+ and 485B- of the sensor are connected to RS485A and RA485B of RS485-LN respectively.



RS485-LN , RS485 Alarm and RS485 I/O are connected together via RS485 (A,B) interface.
1) RS485-LN will read RS485 I/O's registers periodically and uplink to LoRaWAN Server
2) LoRaWAN server can send an downlink command to RS485-Ln to control the RS485 Alarm

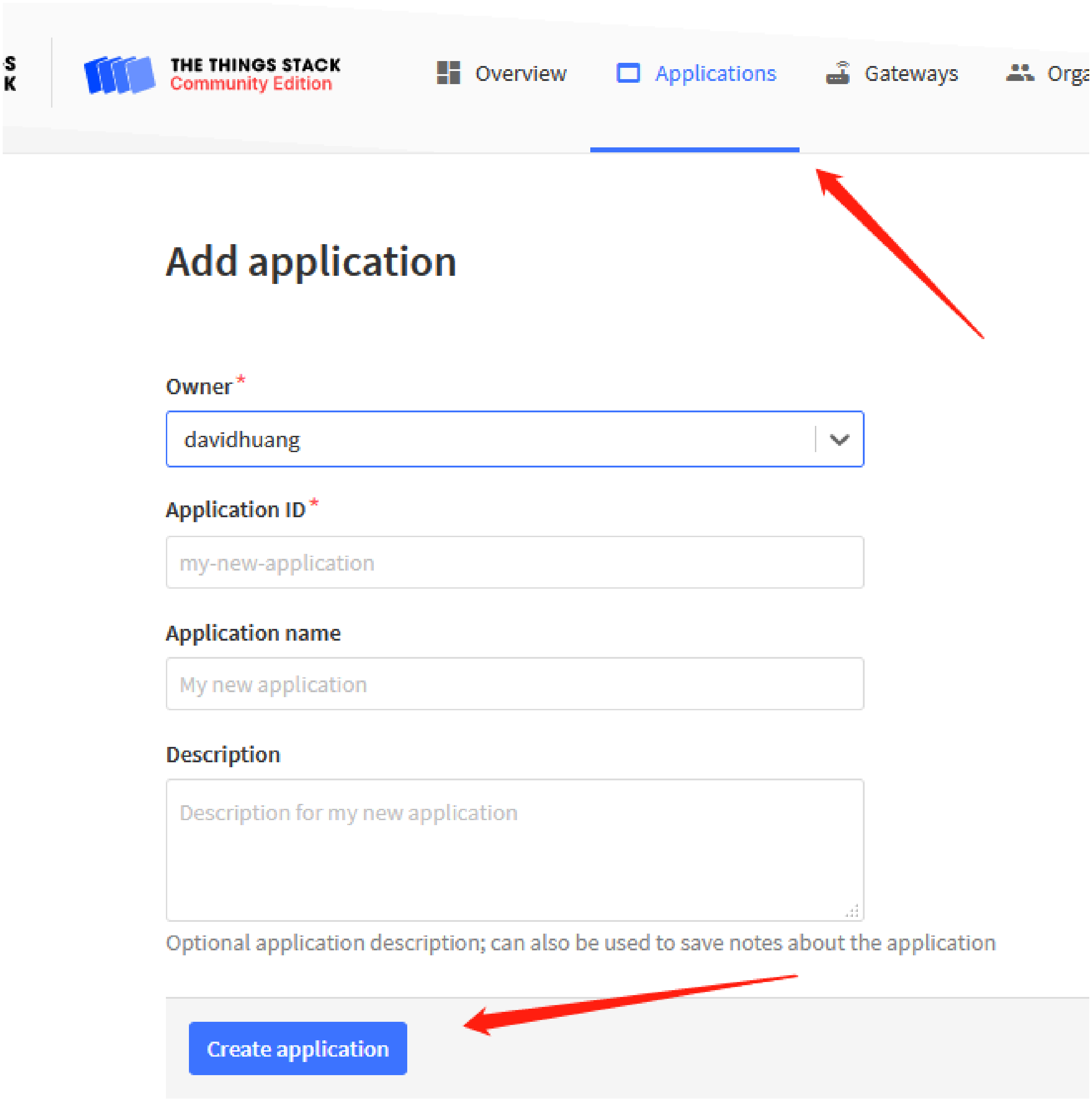
The LG308 is already set to connect to [TTN V3 network](). So what we need to now is only configure the TTN V3:

Step 1: Create a device in TTN V3 with the OTAA keys from RS485-LN.

Each RS485-LN is shipped with a sticker with unique device EUI:

User can enter this key in their LoRaWAN Server portal. Below is TTN V3 screen shot:

Add APP EUI in the application.



Add application

Owner *

davidhuang

Application ID *

my-new-application

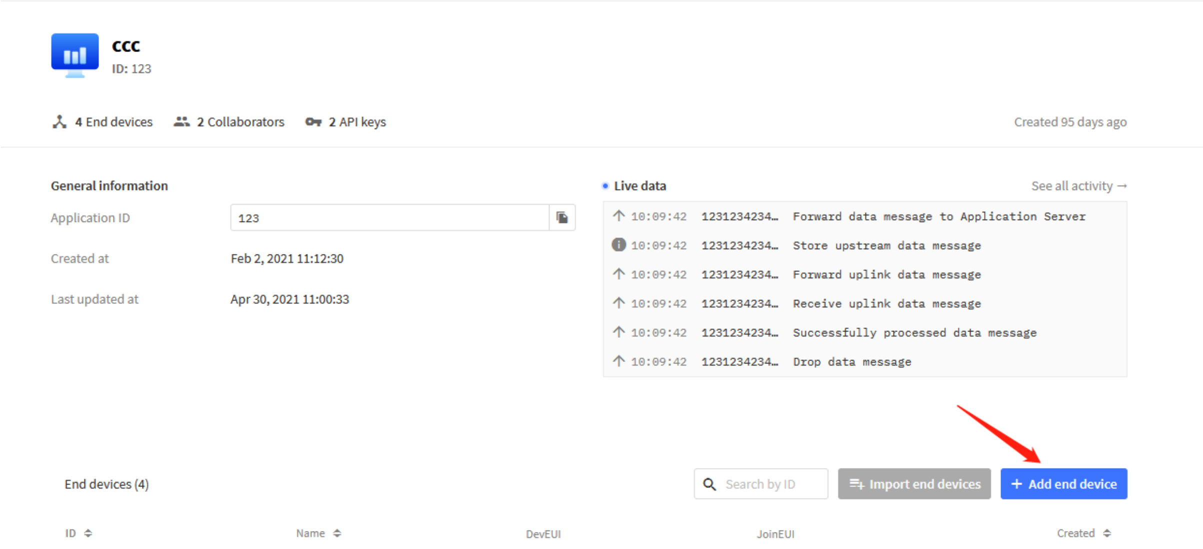
Application name

My new application

Description

Description for my new application

Optional application description; can also be used to save notes about the application



ccc

ID: 123

4 End devices   2 Collaborators   2 API keys   Created 95 days ago

General information

Application ID: 123

Created at: Feb 2, 2021 11:12:30

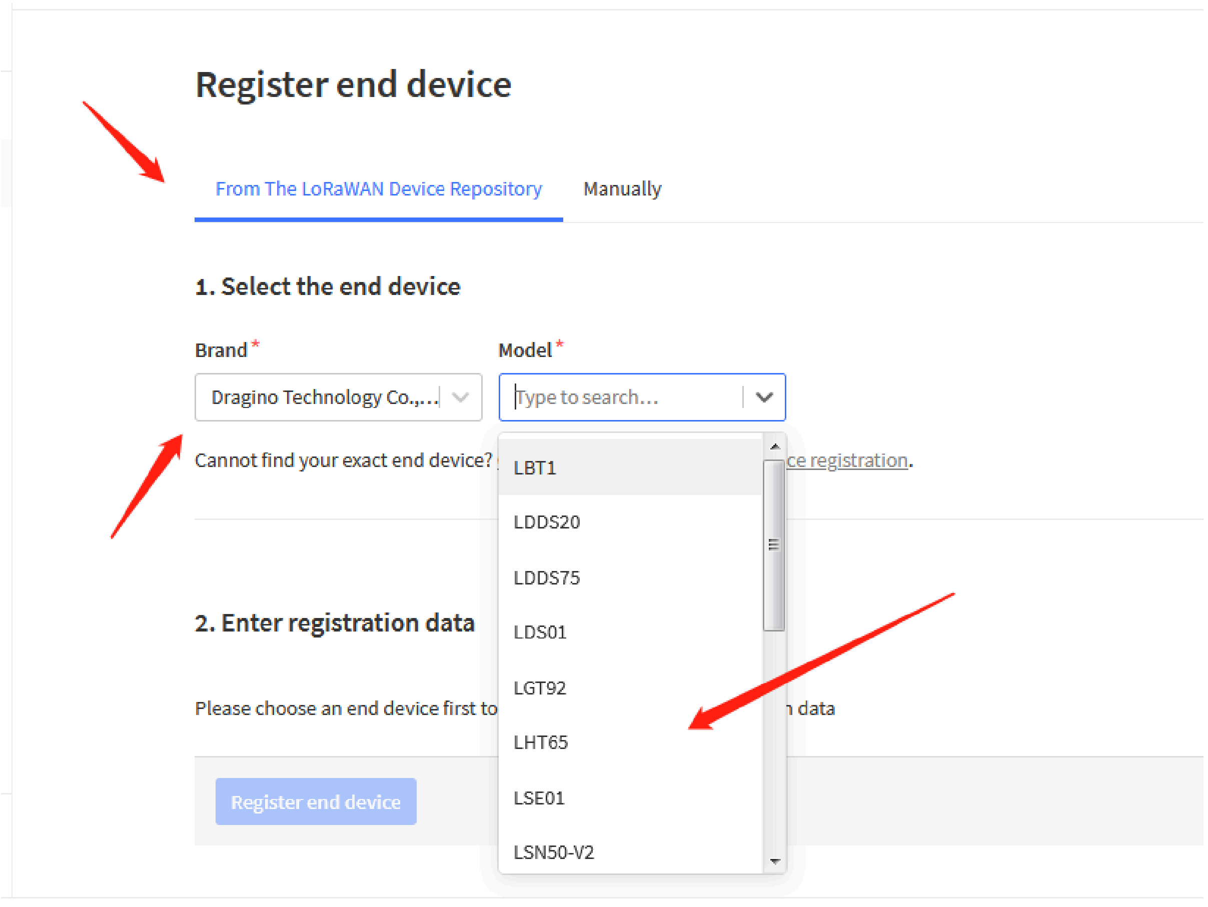
Last updated at: Apr 30, 2021 11:00:33

Live data

```
↑ 10:09:42  1231234234…  Forward data message to Application Server
ⓘ 10:09:42  1231234234…  Store upstream data message
↑ 10:09:42  1231234234…  Forward uplink data message
↑ 10:09:42  1231234234…  Receive uplink data message
↑ 10:09:42  1231234234…  Successfully processed data message
↑ 10:09:42  1231234234…  Drop data message
```

End devices (4)

ID	Name	DevEUI	JoinEUI	Created



Register end device

From The LoRaWAN Device Repository | Manually

1. Select the end device

Brand *: Dragino Technology Co.,...

Model *: Type to search...

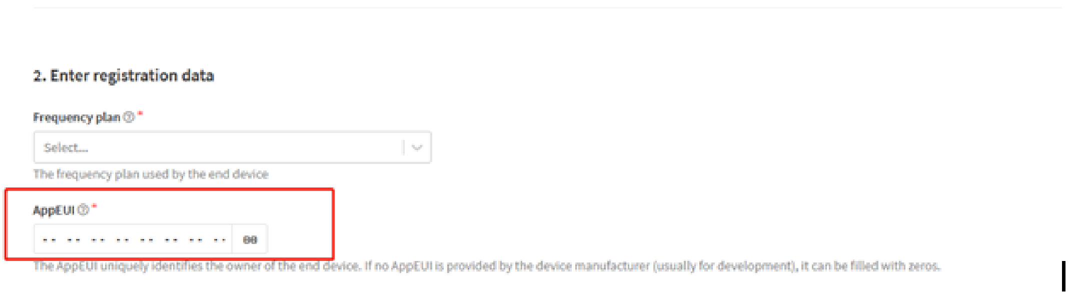
- LBT1
- LDDS20
- LDDS75
- LDS01
- LGT92
- LHT65
- LSE01
- LSN50-V2

Cannot find your exact end device? ... ce registration.

2. Enter registration data

Please choose an end device first to ... data

Register end device



2. Enter registration data

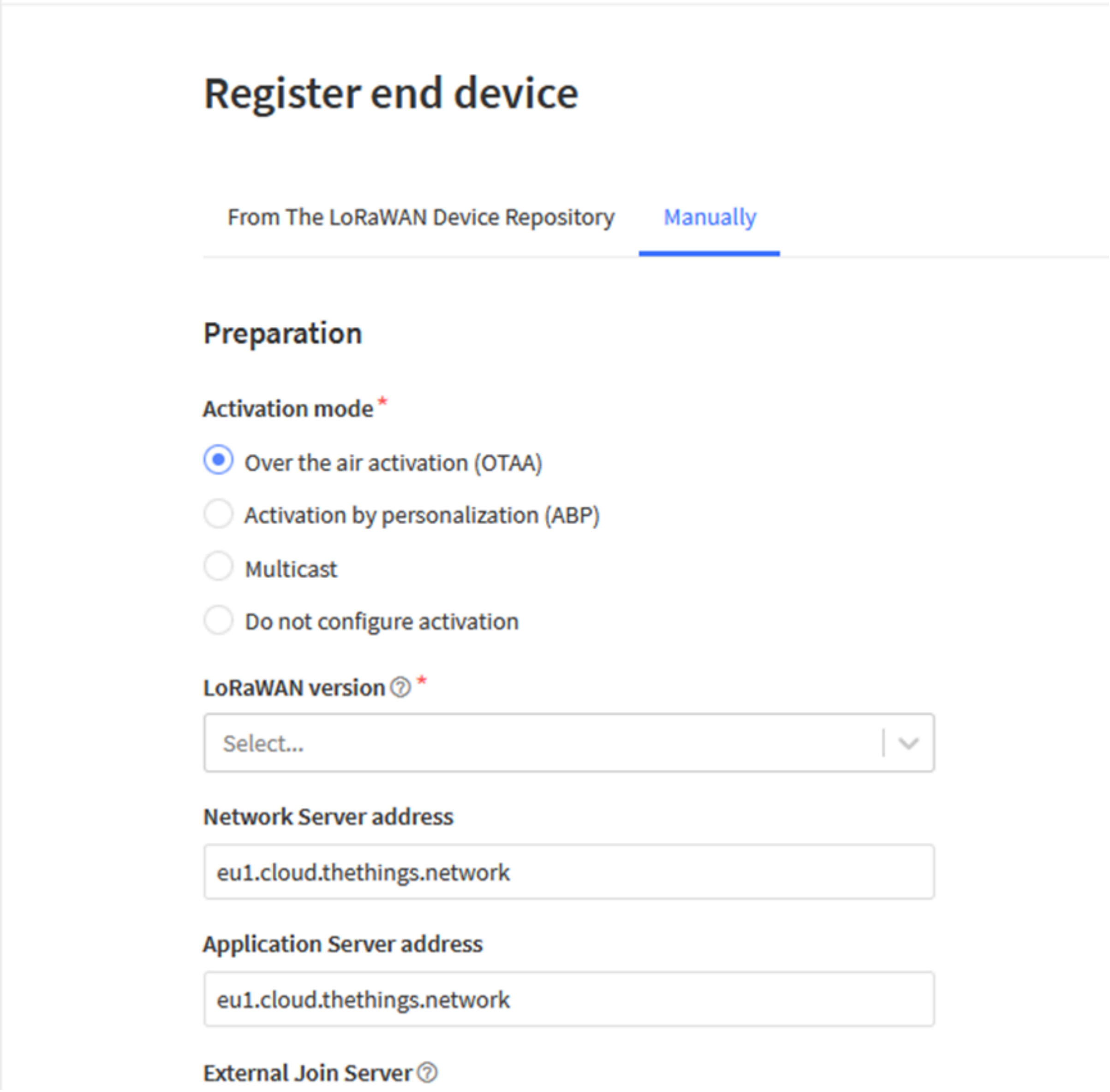
Frequency plan *: Select...

The frequency plan used by the end device

AppEUI *: -- -- -- -- -- -- -- --

The AppEUI uniquely identifies the owner of the end device. If no AppEUI is provided by the device manufacturer (usually for development), it can be filled with zeros.

You can also choose to create the device manually.



Register end device

From The LoRaWAN Device Repository | Manually

Preparation

Activation mode *

● Over the air activation (OTAA)

○ Activation by personalization (ABP)

○ Multicast

○ Do not configure activation

LoRaWAN version *

Select...

Network Server address

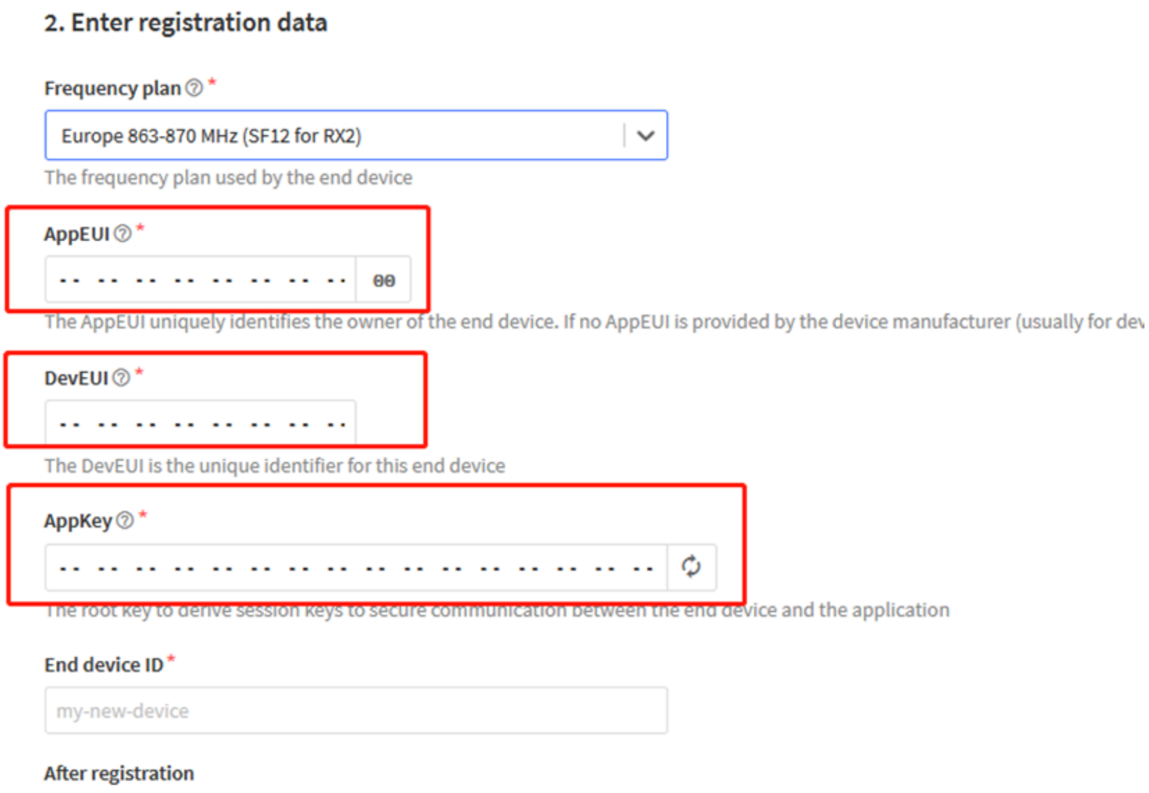
eu1.cloud.thethings.network

Application Server address

eu1.cloud.thethings.network

External Join Server

Add APP KEY and DEV EUI



2. Enter registration data

Frequency plan *

Europe 863-870 MHz (SF12 for RX2)

The frequency plan used by the end device

AppEUI *

-- -- -- -- -- -- -- -- 00

The AppEUI uniquely identifies the owner of the end device. If no AppEUI is provided by the device manufacturer (usually for dev

DevEUI *

-- -- -- -- -- -- -- --

The DevEUI is the unique identifier for this end device

AppKey *

-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

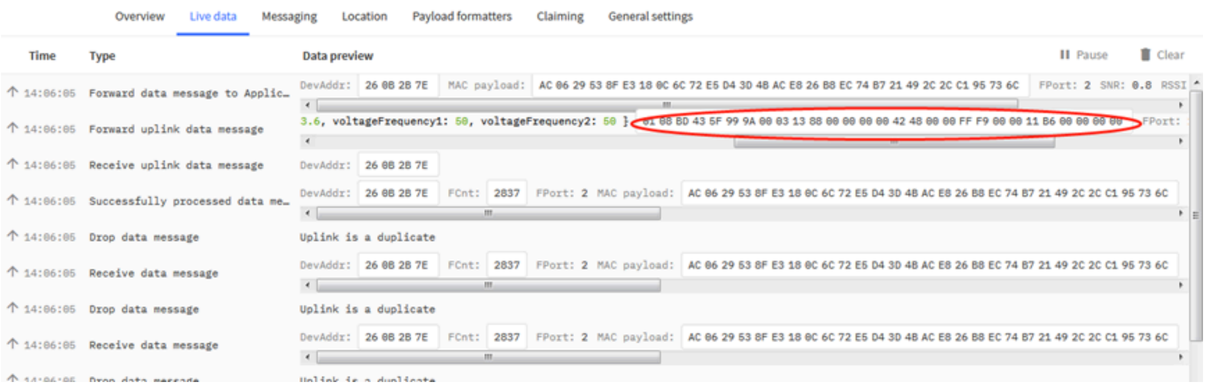
The root key to derive session keys to secure communication between the end device and the application

End device ID *

my-new-device

After registration

Step 2: Power on RS485-LN and it will auto join to the TTN V3 network. After join success, it will start to upload message to TTN V3 and user can see in the panel.



```
Time       Type                               Data preview
↑ 14:06:05 Forward data message to Applic…    DevAddr: 26 0B 2B 7E  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C  FPort: 2  SNR: 0.8  RSSI
↑ 14:06:05 Forward uplink data message        3.6, voltageFrequency1: 50, voltageFrequency2: 50 }  01 08 BD 43 5F 99 9A 00 03 13 88 00 00 00 00 42 48 00 00 FF F9 00 00 11 B6 00 00 00 00  FPort:
↑ 14:06:05 Receive uplink data message        DevAddr: 26 0B 2B 7E
↑ 14:06:05 Successfully processed data me…    DevAddr: 26 0B 2B 7E  FCnt: 2837  FPort: 2  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C
↑ 14:06:05 Drop data message                  Uplink is a duplicate
↑ 14:06:05 Receive data message               DevAddr: 26 0B 2B 7E  FCnt: 2837  FPort: 2  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C
↑ 14:06:05 Drop data message                  Uplink is a duplicate
↑ 14:06:05 Receive data message               DevAddr: 26 0B 2B 7E  FCnt: 2837  FPort: 2  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C
↑ 14:06:05 Drop data message                  Uplink is a duplicate
```

3.3 Configure Commands to read data

There are plenty of RS485 devices in the market and each device has different command to read the valid data. To support these devices in flexible, RS485-LN supports flexible command set. User can use AT Commands or LoRaWAN Downlink Command to configure what commands RS485-LN should send for each sampling and how to handle the return from RS485 devices.

Note: below description and commands are for firmware version >v1.1, if you have firmware version v1.0. Please check the user manual v1.0 or upgrade the firmware to v1.1

3.3.1 Configure UART settings for RS485 or TTL communication

To use RS485-LN to read data from RS485 sensors, connect the RS485-LN A/B traces to the sensors. And user need to make sure RS485-LN use the match UART setting to access the sensors. The related commands for UART settings are:

AT Commands	Description	Example
AT+BAUDR	Set the baud rate (for RS485 connection). Default Value is: 9600.	AT+BAUDR=9600 Options: (1200,2400,4800, 14400,19200,115200)
AT+PARITY	Set UART parity (for RS485 connection)	AT+PARITY=0 Option: 0: no parity, 1: odd parity, 2: even parity
AT+STOPBIT	Set serial stopbit (for RS485 connection)	AT+STOPBIT=0 for 1bit AT+STOPBIT=1 for 1.5 bit AT+STOPBIT=2 for 2 bits

3.3.2 Configure sensors

Some sensors might need to configure before normal operation. User can configure such sensor via PC and RS485 adapter or through RS485-LN AT Commands **AT+CFGDEV**. Each **AT+CFGDEV** equals to send a RS485 command to 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

3.3.3 Configure read commands for each sampling

During each sampling, we need confirm what commands we need to send to the RS485 sensors to read data. After the RS485 sensors send back the value, it normally include 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-LN can support 15 commands to read sensors. And combine the return to one or several uplink payloads.

Each RS485 commands include two parts:

1. What commands RS485-LN will send to the RS485 sensors. There are total 15 commands from **AT +COMMAD1**, **ATCOMMAND2**,..., to **AT+COMMANDF**. All commands are of same grammar.

2. How to get wanted value the from RS485 sensors returns from by 1). There are total 15 AT Commands to handle the return, commands are **AT+DATACUT1**,**AT+DATACUT2**,..., **AT+DATACUTF** corresponding to the commands from 1). All commands are of same grammar.

3. 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**.

Below are examples for the how above AT Commands works.

AT+COMMANDx : This command will be sent to RS485 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-LN, we should use this command AT+COMMAND1=01 03 0B B8 00 02,1 for the same.

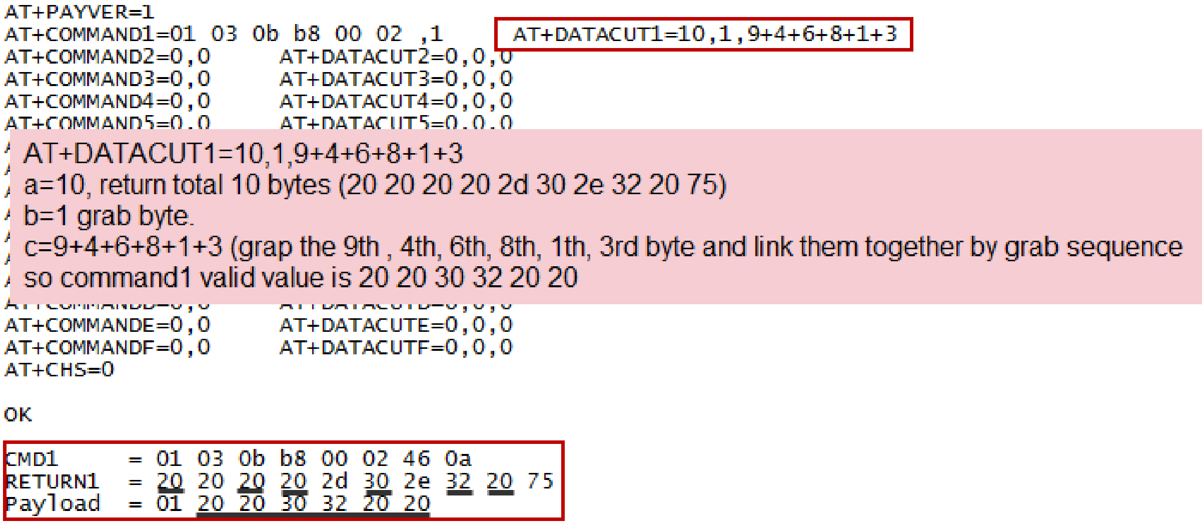
AT+DATACUTx : This command defines how to handle the return from AT+COMMANDx, max return length is 100 bytes.

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:

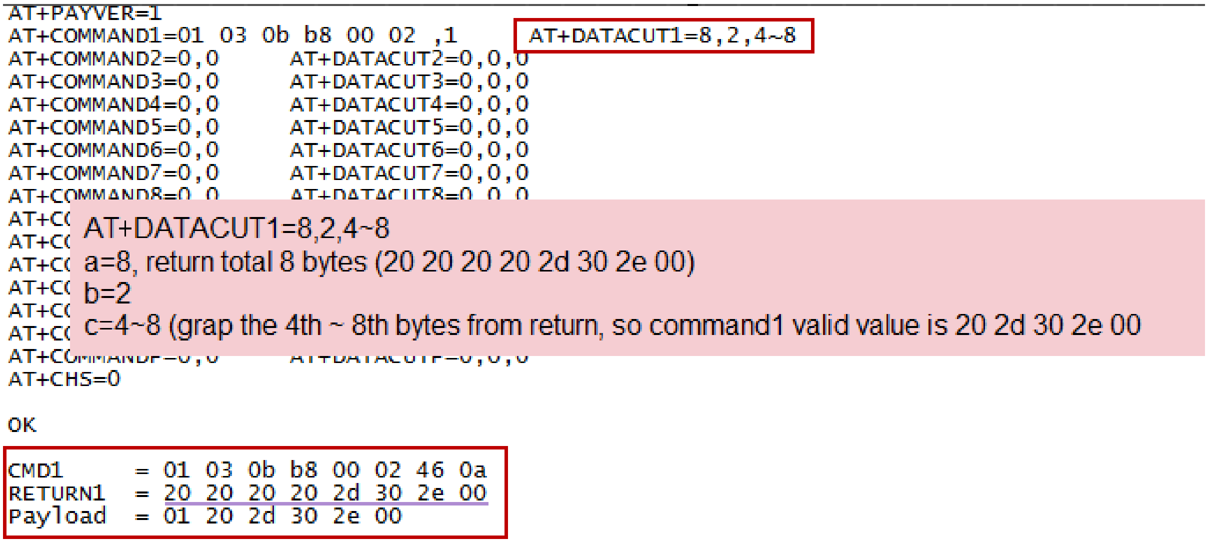
- **Grab bytes**



```
AT+PAYVER=1
AT+COMMAND1=01 03 0b b8 00 02 ,1    AT+DATACUT1=10,1,9+4+6+8+1+3
AT+COMMAND2=0,0       AT+DATACUT2=0,0,0
AT+COMMAND3=0,0       AT+DATACUT3=0,0,0
AT+COMMAND4=0,0       AT+DATACUT4=0,0,0
AT+COMMAND5=0,0       AT+DATACUT5=0,0,0
AT+COMMANDD=0,0       AT+DATACUTD=0,0,0
AT+COMMANDE=0,0       AT+DATACUTE=0,0,0
AT+COMMANDF=0,0       AT+DATACUTF=0,0,0
AT+CHS=0

OK

CMD1      = 01 03 0b b8 00 02 46 0a
RETURN1   = 20 20 20 20 2d 30 2e 32 20 75
Payload   = 01 20 20 30 32 20 20
```

AT+DATACUT1=10,1,9+4+6+8+1+3
a=10, return total 10 bytes (20 20 20 20 2d 30 2e 32 20 75)
b=1 grab byte.
c=9+4+6+8+1+3 (grap the 9th , 4th, 6th, 8th, 1th, 3rd byte and link them together by grab sequence
so command1 valid value is 20 20 30 32 20 20

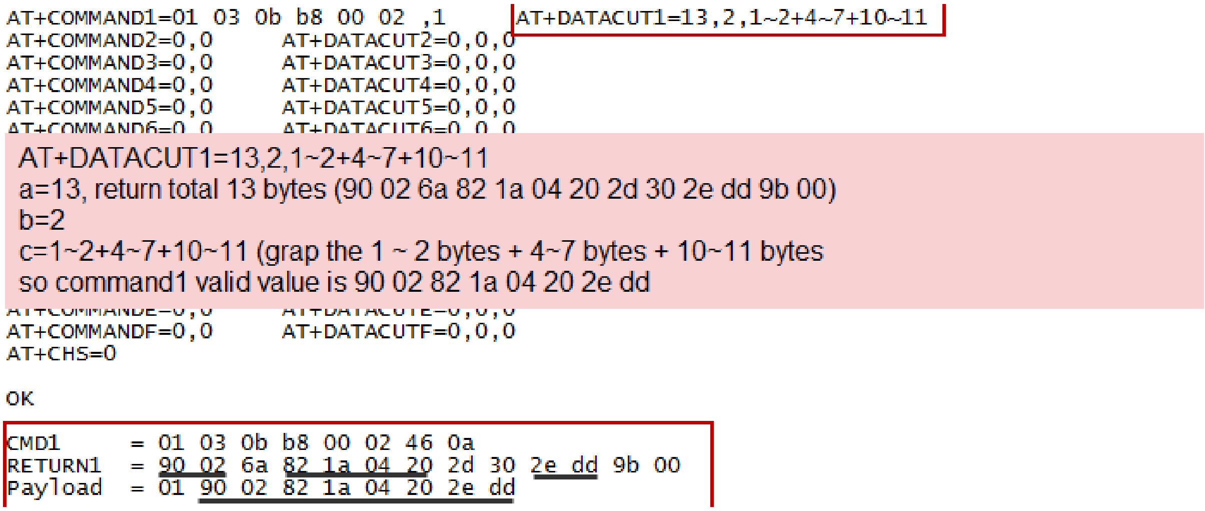
- **Grab a section**



```
AT+PAYVER=1
AT+COMMAND1=01 03 0b b8 00 02 ,1     AT+DATACUT1=8,2,4~8
AT+COMMAND2=0,0       AT+DATACUT2=0,0,0
AT+COMMAND3=0,0       AT+DATACUT3=0,0,0
AT+COMMAND4=0,0       AT+DATACUT4=0,0,0
AT+COMMAND5=0,0       AT+DATACUT5=0,0,0
AT+COMMAND6=0,0       AT+DATACUT6=0,0,0
AT+COMMAND7=0,0       AT+DATACUT7=0,0,0
AT+COMMAND8=0,0       AT+DATACUT8=0,0,0
AT+COMMANDF=0,0       AT+DATACUTF=0,0,0
AT+CHS=0

OK

CMD1      = 01 03 0b b8 00 02 46 0a
RETURN1   = 20 20 20 20 2d 30 2e 00
Payload   = 01 20 2d 30 2e 00
```

AT+DATACUT1=8,2,4~8
a=8, return total 8 bytes (20 20 20 20 2d 30 2e 00)
b=2
c=4~8 (grap the 4th ~ 8th bytes from return, so command1 valid value is 20 2d 30 2e 00

- **Grab different sections**



```
AT+COMMAND1=01 03 0b b8 00 02 ,1      AT+DATACUT1=13,2,1~2+4~7+10~11
AT+COMMAND2=0,0      AT+DATACUT2=0,0,0
AT+COMMAND3=0,0      AT+DATACUT3=0,0,0
AT+COMMAND4=0,0      AT+DATACUT4=0,0,0
AT+COMMAND5=0,0      AT+DATACUT5=0,0,0
AT+COMMAND6=0,0      AT+DATACUT6=0,0,0
```

AT+DATACUT1=13,2,1~2+4~7+10~11
a=13, return total 13 bytes (90 02 6a 82 1a 04 20 2d 30 2e dd 9b 00)
b=2
c=1~2+4~7+10~11 (grap the 1 ~ 2 bytes + 4~7 bytes + 10~11 bytes
so command1 valid value is 90 02 82 1a 04 20 2e dd

```
AT+COMMANDE=0,0      AT+DATACUTE=0,0,0
AT+COMMANDF=0,0      AT+DATACUTF=0,0,0
AT+CHS=0

OK

CMD1      = 01 03 0b b8 00 02 46 0a
RETURN1   = 90 02 6a 82 1a 04 20 2d 30 2e dd 9b 00
Payload   = 01 90 02 82 1a 04 20 2e dd
```

3.3.4 Compose the uplink payload

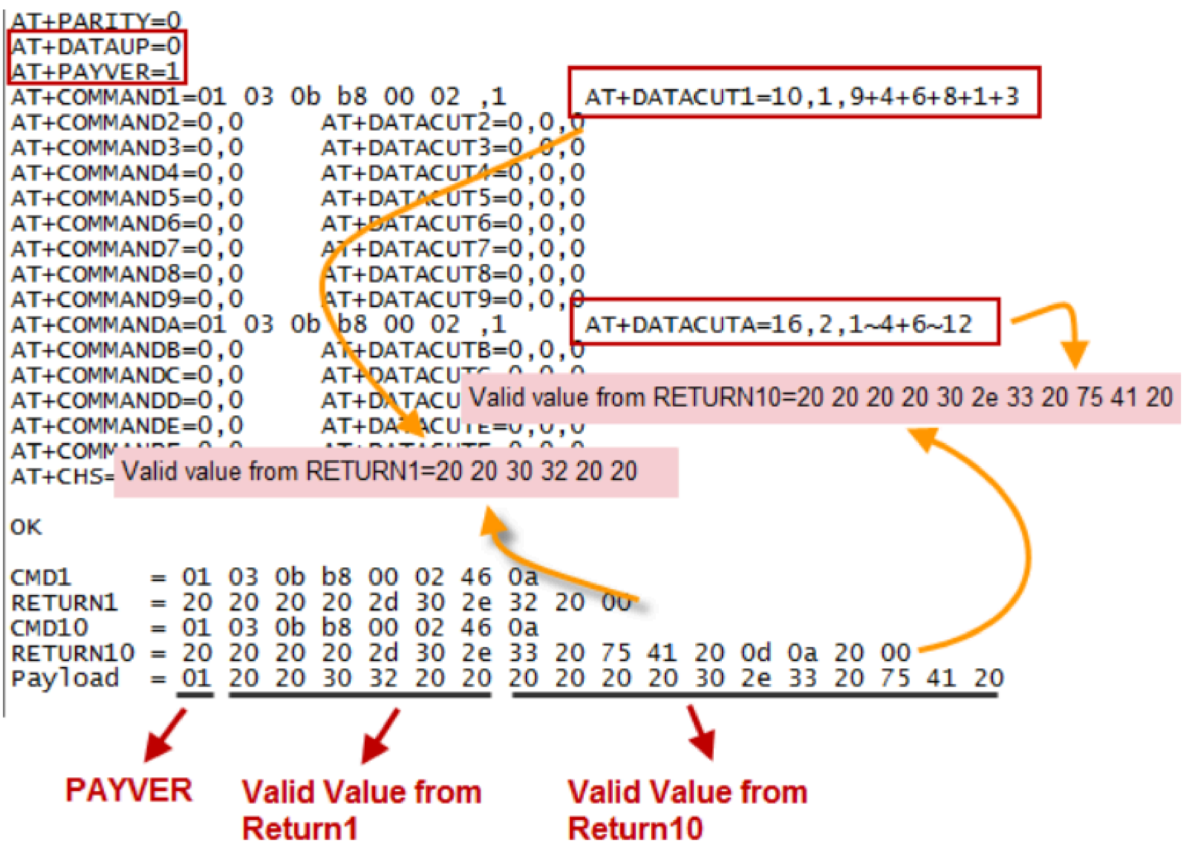
Through AT+COMMANDx and AT+DATACUTx we got valid value from each RS485 commands, Assume these valid value are RETURN1, RETURN2, .., to RETURNx. The next step is how to compose the LoRa Uplink Payload by these RETURNs. The command is **AT+DATAUP.**

Examples: AT+DATAUP=0

Compose the uplink payload with value returns in sequence and send with **A SIGNLE UPLINK**.

Final Payload is **Battery Info+PAYVER + VALID Value from RETURN1 + Valid Value from RETURN2 + ... + RETURNx**

Where PAYVER is defined by AT+PAYVER, below is an example screen shot.



```
AT+PARITY=0
AT+DATAUP=0
AT+PAYVER=1
AT+COMMAND1=01 03 0b b8 00 02 ,1       AT+DATACUT1=10,1,9+4+6+8+1+3
AT+COMMAND2=0,0      AT+DATACUT2=0,0,0
AT+COMMAND3=0,0      AT+DATACUT3=0,0,0
AT+COMMAND4=0,0      AT+DATACUT4=0,0,0
AT+COMMAND5=0,0      AT+DATACUT5=0,0,0
AT+COMMAND6=0,0      AT+DATACUT6=0,0,0
AT+COMMAND7=0,0      AT+DATACUT7=0,0,0
AT+COMMAND8=0,0      AT+DATACUT8=0,0,0
AT+COMMAND9=0,0      AT+DATACUT9=0,0,0
AT+COMMANDA=01 03 0b b8 00 02 ,1       AT+DATACUTA=16,2,1~4+6~12
AT+COMMANDB=0,0      AT+DATACUTB=0,0,0
AT+COMMANDC=0,0      AT+DATACUTC=0,0,0
AT+COMMANDD=0,0      AT+DATACU
AT+COMMANDE=0,0      AT+DATACUTE=0,0,0
AT+COMM
AT+CHS=

OK

CMD1     = 01 03 0b b8 00 02 46 0a
RETURN1  = 20 20 20 20 2d 30 2e 32 20 00
CMD10    = 01 03 0b b8 00 02 46 0a
RETURN10 = 20 20 20 20 2d 30 2e 33 20 75 41 20 0d 0a 20 00
Payload  = 01 20 20 30 32 20 20 20 20 20 20 30 2e 33 20 75 41 20
```

Valid value from RETURN10=20 20 20 20 30 2e 33 20 75 41 20

Valid value from RETURN1=20 20 30 32 20 20

Examples: AT+DATAUP=1

Compose the uplink payload with value returns in sequence and send with **Multiply UPLINKs**.

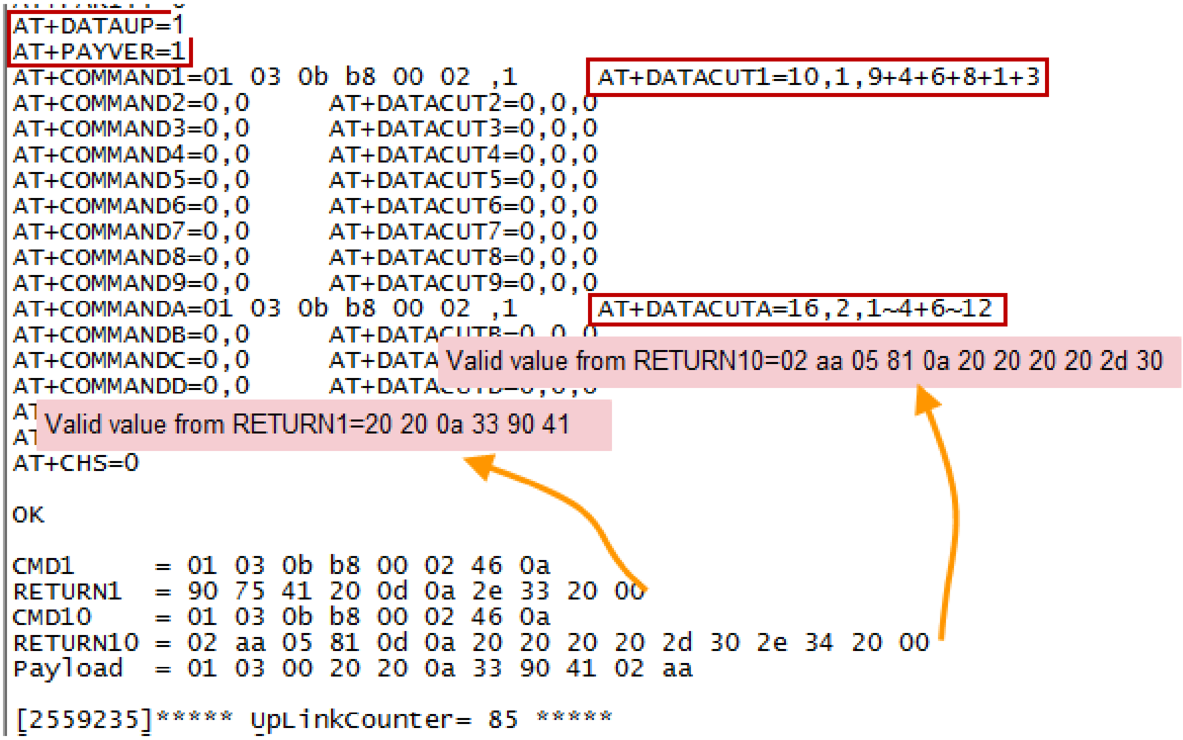
Final Payload is **PAYVER + PAYLOAD COUNT + PAYLOAD# + DATA**

1. PAYVER: Defined by AT+PAYVER

2. PAYLOAD COUNT: Total how many uplinks of this sampling.

3. PAYLOAD#: Number of this uplink. (from 0,1,2,3...,to PAYLOAD COUNT)

4. DATA: Valid value: max 8 bytes for each uplink so each uplink <= 11 bytes. For the last uplink, DATA will might less than 8 bytes



```
AT+DATAUP=1
AT+PAYVER=1
AT+COMMAND1=01 03 0b b8 00 02 ,1       AT+DATACUT1=10,1,9+4+6+8+1+3
AT+COMMAND2=0,0      AT+DATACUT2=0,0,0
AT+COMMAND3=0,0      AT+DATACUT3=0,0,0
AT+COMMAND4=0,0      AT+DATACUT4=0,0,0
AT+COMMAND5=0,0      AT+DATACUT5=0,0,0
AT+COMMAND6=0,0      AT+DATACUT6=0,0,0
AT+COMMAND7=0,0      AT+DATACUT7=0,0,0
AT+COMMAND8=0,0      AT+DATACUT8=0,0,0
AT+COMMAND9=0,0      AT+DATACUT9=0,0,0
AT+COMMANDA=01 03 0b b8 00 02 ,1       AT+DATACUTA=16,2,1~4+6~12
AT+COMMANDB=0,0      AT+DATACUTB=0,0,0
AT+COMMANDC=0,0      AT+DATA
AT+COMMANDD=0,0      AT+DATA
AT
AT
AT+CHS=0

OK

CMD1      = 01 03 0b b8 00 02 46 0a
RETURN1   = 90 75 41 20 0d 0a 2e 33 20 00
CMD10     = 01 03 0b b8 00 02 46 0a
RETURN10  = 02 aa 05 81 0d 0a 20 20 20 20 2d 30 2e 34 20 00
Payload   = 01 03 00 20 20 0a 33 90 41 02 aa

[2559235]***** UpLinkCounter= 85 *****
```

Valid value from RETURN10=02 aa 05 81 0a 20 20 20 20 2d 30

Valid value from RETURN1=20 20 0a 33 90 41

So totally there will be 3 uplinks for this sampling, each uplink include 8 bytes DATA

DATA1=RETURN1 Valid Value + the first two of Valid value of RETURN10= **20 20 0a 33 90 41 02 aa**

DATA2=3rd ~ 10th byte of Valid value of RETURN10= **05 81 0a 20 20 20 20 2d**

DATA3=the rest of Valid value of RETURN10= **30**

Notice: In firmware v1.3, the Max bytes has been changed according to the max bytes in different Frequency Bands for lowest SF. As below:

* For AU915/AS923 bands, if UplinkDwell time=0, max 51 bytes for each uplink.

* For AU915/AS923 bands, if UplinkDwell time=0, max 11 bytes for each uplink.

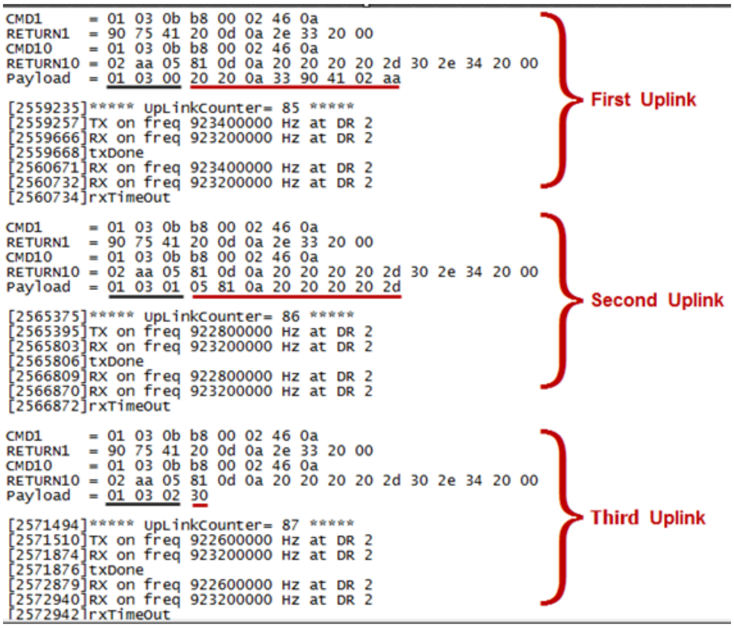
* For US915 band, max 11 bytes for each uplink.

* For all other bands: max 51 bytes for each uplink.

When AT+DATAUP=1, the maximum number of segments is 15, and the maximum total number of bytes is 1500;

When AT+DATAUP=1 and AT+ADR=0, the maximum number of bytes of each payload is determined by the DR value. (Since v1.4.0)

Below are the uplink payloads:



```
CMD1     = 01 03 0b b8 00 02 46 0a
RETURN1  = 90 75 41 20 0d 0a 2e 33 20 00
CMD10    = 01 03 0b b8 00 02 46 0a
RETURN10 = 02 aa 05 81 0d 0a 20 20 20 20 2d 30 2e 34 20 00
Payload  = 01 03 00 20 20 0a 33 90 41 02 aa

[2559235]***** UpLinkCounter= 85 *****
[2559257]TX on freq 923400000 Hz at DR 2
[2559666]RX on freq 923200000 Hz at DR 2
[2559668]txDone
[2560671]RX on freq 923400000 Hz at DR 2
[2560732]RX on freq 923200000 Hz at DR 2
[2560734]rxTimeOut

CMD1     = 01 03 0b b8 00 02 46 0a
RETURN1  = 90 75 41 20 0d 0a 2e 33 20 00
CMD10    = 01 03 0b b8 00 02 46 0a
RETURN10 = 02 aa 05 81 0d 0a 20 20 20 20 2d 30 2e 34 20 00
Payload  = 01 03 01 05 81 0a 20 20 20 20 2d

[2565375]***** UpLinkCounter= 86 *****
[2565395]TX on freq 922800000 Hz at DR 2
[2565803]RX on freq 923200000 Hz at DR 2
[2565806]txDone
[2566809]RX on freq 922800000 Hz at DR 2
[2566870]RX on freq 923200000 Hz at DR 2
[2566872]rxTimeOut

CMD1     = 01 03 0b b8 00 02 46 0a
RETURN1  = 90 75 41 20 0d 0a 2e 33 20 00
CMD10    = 01 03 0b b8 00 02 46 0a
RETURN10 = 02 aa 05 81 0d 0a 20 20 20 20 2d 30 2e 34 20 00
Payload  = 01 03 02 30

[2571494]***** UpLinkCounter= 87 *****
[2571510]TX on freq 922600000 Hz at DR 2
[2571874]RX on freq 923200000 Hz at DR 2
[2571876]txDone
[2572879]RX on freq 922600000 Hz at DR 2
[2572940]RX on freq 923200000 Hz at DR 2
[2572942]rxTimeOut
```

3.3.5 Uplink on demand

Except uplink periodically, RS485-LN is able to uplink on demand. The server send downlink command to RS485-LN and RS485 will uplink data base on the command.

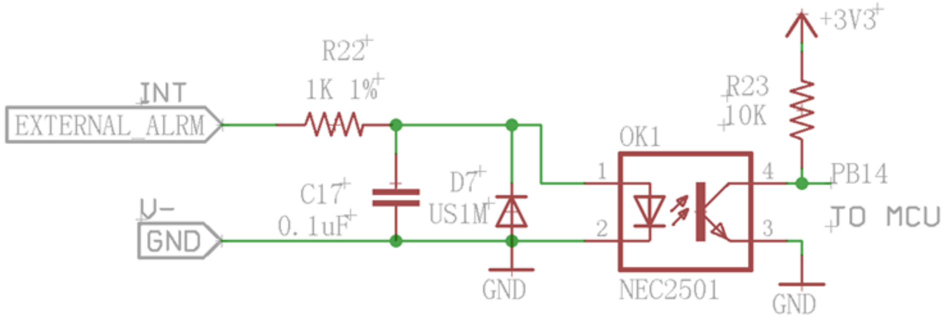
Downlink control command:

0x08 command: Poll an uplink with current command set in RS485-LN.

0xA8 command: Send a command to RS485-LN and uplink the output from sensors.

3.3.6 Uplink on Interrupt

RS485-LN support external Interrupt uplink since hardware v1.2 release.



Connect the Interrupt pin to RS485-LN INT port and connect the GND pin to V- port. When there is a high voltage (Max 24v) on INT pin. Device will send an uplink packet.

3.4 Uplink Payload

Size(bytes)	1	Length depends on the return from the commands

Value	PAYLOAD_VER	If the valid payload is too long and exceed the maximum support payload length in server, server will show payload not provided in the LoRaWAN server.

Below is the decoder for the first 3 bytes. The rest bytes are dynamic depends on different RS485 sensors.

3.5 Configure RS485-LN via AT or Downlink

User can configure RS485-LN via AT Commands or LoRaWAN Downlink Commands

There are two kinds of Commands:

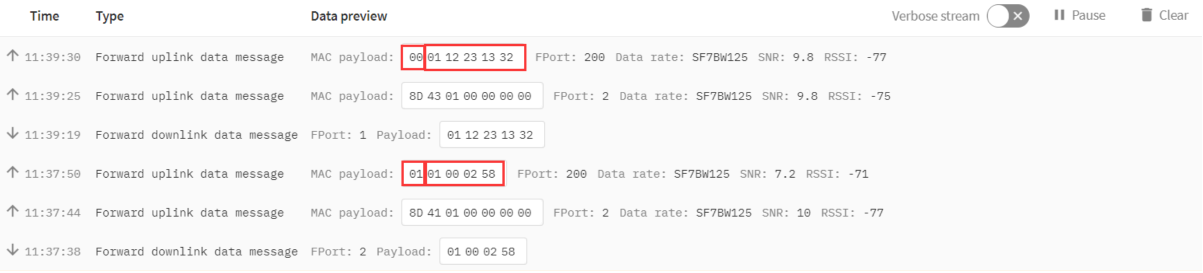
- **Common Commands**: They should be available for each sensor, such as: change uplink interval, reset device. For firmware v1.3, user can find what common commands it supports: AT Commands and Downlink Command
- **Sensor Related Commands**: These commands are special designed for RS485-LN. User can see these commands below:

3.5.1 Common Commands

They should be available for each of Dragino Sensors, such as: change uplink interval, reset device. For firmware v1.3, user can find what common commands it supports: End Device AT Commands and Downlink Command

3.5.2 Downlink Response(Since firmware v1.4)

Response feature is added to the server's downlink, a special package with a FPort of 200 will be uploaded immediately after receiving the data sent by the server.



```
Time      Type                            Data preview
11:39:30  Forward uplink data message     MAC payload: 00 01 12 23 13 32   FPort: 200  Data rate: SF7BW125  SNR: 9.8  RSSI: -77
11:39:25  Forward uplink data message     MAC payload: 8D 43 01 00 00 00 00  FPort: 2  Data rate: SF7BW125  SNR: 9.8  RSSI: -75
11:39:19  Forward downlink data message   FPort: 1  Payload: 01 12 23 13 32
11:37:50  Forward uplink data message     MAC payload: 01 01 00 02 58   FPort: 200  Data rate: SF7BW125  SNR: 7.2  RSSI: -71
11:37:44  Forward uplink data message     MAC payload: 8D 41 01 00 00 00 00  FPort: 2  Data rate: SF7BW125  SNR: 10  RSSI: -77
11:37:38  Forward downlink data message   FPort: 2  Payload: 01 00 02 58
```

The first byte of this package represents whether the configuration is successful, 00 represents failure, 01 represents success. Except for the first byte, the other is the previous downlink. (All commands except A8 type commands are applicable)

3.5.3 Sensor related commands

RS485 Debug Command

This command is used to configure the RS485 devices; they won't be used during sampling. Max Length of AT +CFGDEV is **40 bytes**.

- **AT Command**

AT+CFGDEV=xx xx xx xx xx xx xx xx xx xx xx xx,m m: 0: no CRC, 1: add CRC-16/MODBUS in the end of this command

- **Downlink Payload**

Format: **A8 MM NN XX XX XX XX YY**

Where:

- MM: 1: add CRC-16/MODBUS ; 0: no CRC
- NN: The length of RS485 command
- XX XX XX XX: RS485 command total NN bytes
- YY: How many bytes will be uplink from the return of this RS485 command,
  - if YY=0, RS485-LN will execute the downlink command without uplink;
  - if YY>0, RS485-LN will uplink total YY bytes from the output of this RS485 command; Fport=200
  - if YY=FF, RS485-LN will uplink RS485 output with the downlink command content; Fport=200.

Example 1: --> Configure without ask for uplink (YY=0)

To connect a Modbus Alarm with below commands.

- The command to active alarm is: 0A 05 00 04 00 01 4C B0. Where 0A 05 00 04 00 01 is the Modbus command to read the register 00 40 where stored the DI status. The 4C B0 is the CRC-16/MODBUS which calculate manually.
- The command to deactivate alarm is: 0A 05 00 04 00 00 8D 70. Where 0A 05 00 04 00 00 is the Modbus command to read the register 00 40 where stored the DI status. The 8D 70 is the CRC-16/MODBUS which calculate manually.

So if user want to use downlink command to control to RS485 Alarm, he can use:

A8 01 06 0A 05 00 04 00 01 00: to activate the RS485 Alarm

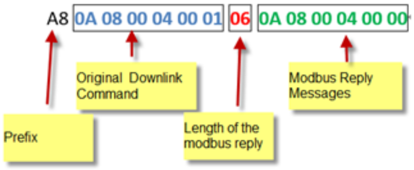
A8 01 06 0A 05 00 04 00 00 00: to deactivate the RS485 Alarm

A8 is type code and 01 means add CRC-16/MODBUS at the end, the 3rd byte is 06, means the next 6 bytes are the command to be sent to the RS485 network, the final byte 00 means this command don’t need to acquire output.

Example 2: --> Configure with requesting uplink and original downlink command (**YY=FF**)

User in IoT server send a downlink command: **A8 01 06 0A 08 00 04 00 01 YY**

RS485-LN got this downlink command and send **0A 08 00 04 00 01** to Modbus network. One of the RS485 sensor in the network send back Modbus reply **0A 08 00 04 00 00**. RS485-LN get this reply and combine with the original downlink command and uplink. The uplink message is: **A8 0A 08 00 04 00 01 06 0A 08 00 04 00 00**



Set Payload version

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

- **AT Command:**

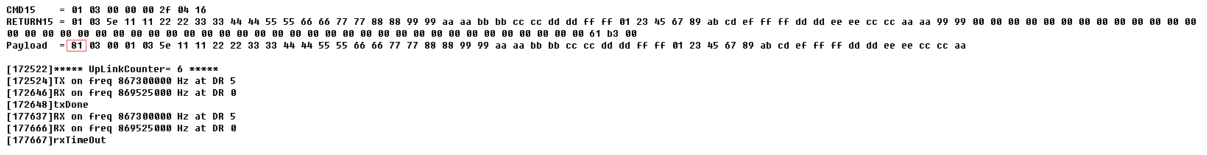
AT+PAYVER: Set PAYVER field = 1

- **Downlink Payload:**

0xAE 01   -->   Set PAYVER field = 0x01

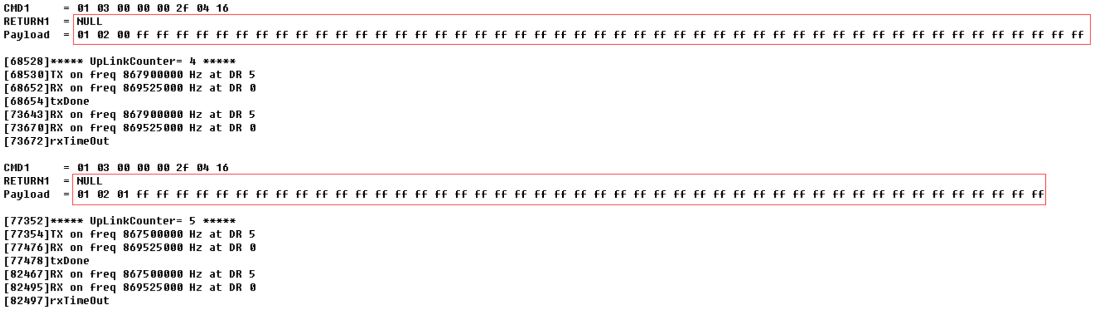
0xAE 0F   -->   Set PAYVER field = 0x0F

1) Add the interrupt flag at the highest bit of the Payver byte, that is, Byte7 of the first byte. (Since v1.4.0)



```
CMD15    = 01 03 00 00 00 2f 04 16
RETURN15 = 01 03 5e 11 11 22 22 33 33 44 44 55 55 66 66 77 77 88 88 99 99 aa aa bb bb cc cc dd dd ff ff 01 23 45 67 89 ab cd ef ff ff dd dd ee ee cc cc aa aa 99 99 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 b3 00
Payload  = 81 03 00 01 03 5e 11 11 22 22 33 33 44 44 55 55 66 66 77 77 88 88 99 99 aa aa bb bb cc cc dd dd ff ff 01 23 45 67 89 ab cd ef ff ff dd dd ee ee cc cc aa

[172522]***** UpLinkCounter= 6 *****
[172524]TX on freq 867300000 Hz at DR 5
[172646]RX on freq 869525000 Hz at DR 0
[172648]txDone
[177637]RX on freq 867300000 Hz at DR 5
[177666]RX on freq 869525000 Hz at DR 0
[177667]rxTimeOut
```

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 FFs.



```
CMD1     = 01 03 00 00 00 2f 04 16
RETURN1  = NULL
Payload  = 01 02 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

[68528]***** UpLinkCounter= 4 *****
[68530]TX on freq 867900000 Hz at DR 5
[68652]RX on freq 869525000 Hz at DR 0
[68654]txDone
[73643]RX on freq 867900000 Hz at DR 5
[73670]RX on freq 869525000 Hz at DR 0
[73672]rxTimeOut

CMD1     = 01 03 00 00 00 2f 04 16
RETURN1  = NULL
Payload  = 01 02 01 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

[77352]***** UpLinkCounter= 5 *****
[77354]TX on freq 867500000 Hz at DR 5
[77476]RX on freq 869525000 Hz at DR 0
[77478]txDone
[82467]RX on freq 867500000 Hz at DR 5
[82495]RX on freq 869525000 Hz at DR 0
[82497]rxTimeOut
```

Set RS485 Sampling Commands

AT+COMMANDx or AT+DATACUTx

These three commands are used to configure how the RS485-LN polling data from Modbus device. Detail of usage please see : polling RS485 device.

- **AT Command:**

AT+COMMANDx: Configure RS485 read command to sensor.

AT+DATACUTx: Configure how to handle return from RS485 devices.

- **Downlink Payload:**

0xAF downlink command can be used to set AT+COMMANDx or AT+DATACUTx.

Note : if user use AT+COMMANDx to add a new command, he also need to send AT+DATACUTx downlink.

Format: **AF MM NN LL XX XX XX XX YY**

Where:

- MM: the ATCOMMAND or AT+DATACUT to be set. Value from 01 ~ 0F,
- NN: 0: no CRC; 1: add CRC-16/MODBUS ; 2: set the AT+DATACUT value.
- LL: The length of AT+COMMAND or AT+DATACUT command
- XX XX XX XX: AT+COMMAND or AT+DATACUT command
- YY: If YY=0, RS485-LN will execute the downlink command without uplink; if YY=1, RS485-LN will execute an uplink after got this command.

Example:

AF 03 01 06 0A 05 00 04 00 01 00: Same as AT+COMMAND3=0A 05 00 04 00 01,1

AF 03 02 06 10 01 05 06 09 0A 00: Same as AT+DATACUT3=**16**,**1**,**5+6+9+10**

AF 03 02 06 0B 02 05 07 08 0A 00: Same as AT+DATACUT3=**11**,**2**,**5~7+8~10**

Fast command to handle MODBUS device

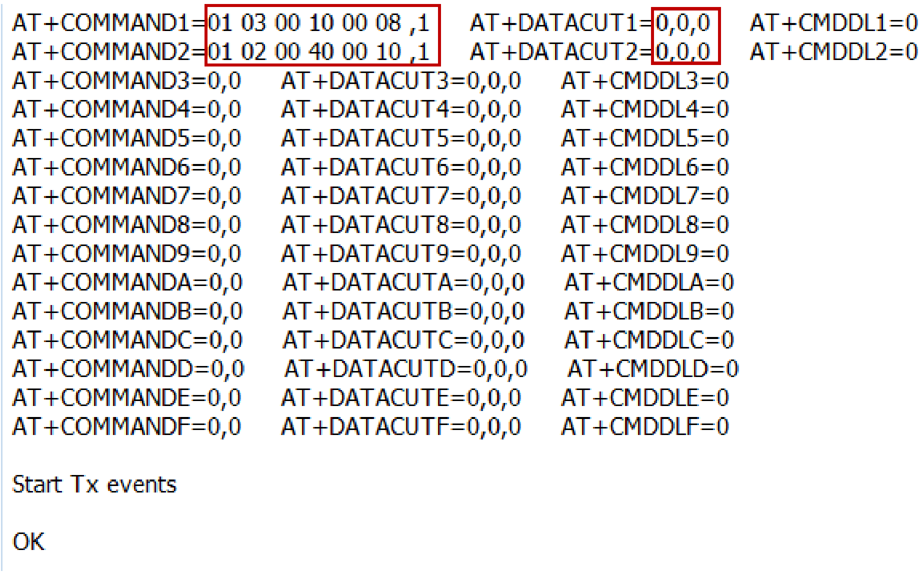
AT+MBFUN is valid since v1.3 firmware version. The command is for fast configure to read Modbus devices. It is only valid for the devices which follow the MODBUS-RTU protocol.

This command is valid since v1.3 firmware version

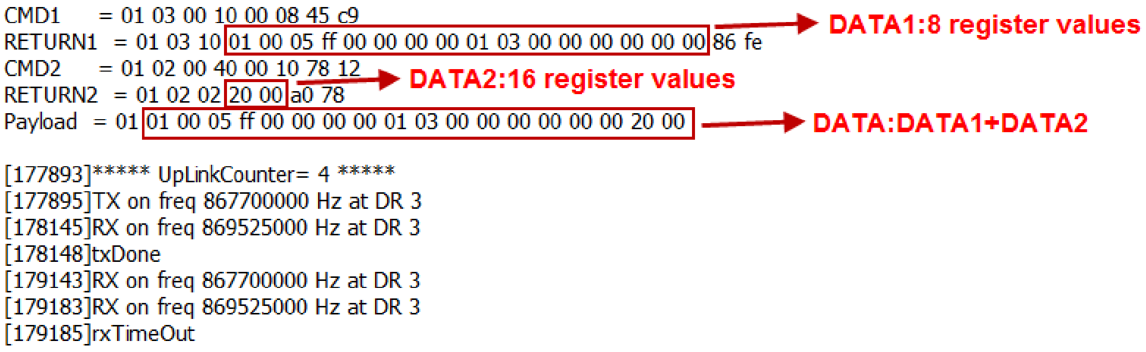
AT+MBFUN can auto read the Modbus function code: 01, 02, 03 or 04. AT+MBFUN has lower priority vs AT +DATACUT command. If AT+DATACUT command is configured, AT+MBFUN will be ignore.

Example:

- AT+MBFUN=1 and AT+DATACUT1/AT+DATACUT2 are not configure (0,0,0). So RS485-LN.
- 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.



```
AT+COMMAND1=01 03 00 10 00 08 ,1    AT+DATACUT1=0,0,0    AT+CMDDL1=0
AT+COMMAND2=01 02 00 40 00 10 ,1    AT+DATACUT2=0,0,0    AT+CMDDL2=0
AT+COMMAND3=0,0    AT+DATACUT3=0,0,0    AT+CMDDL3=0
AT+COMMAND4=0,0    AT+DATACUT4=0,0,0    AT+CMDDL4=0
AT+COMMAND5=0,0    AT+DATACUT5=0,0,0    AT+CMDDL5=0
AT+COMMAND6=0,0    AT+DATACUT6=0,0,0    AT+CMDDL6=0
AT+COMMAND7=0,0    AT+DATACUT7=0,0,0    AT+CMDDL7=0
AT+COMMAND8=0,0    AT+DATACUT8=0,0,0    AT+CMDDL8=0
AT+COMMAND9=0,0    AT+DATACUT9=0,0,0    AT+CMDDL9=0
AT+COMMANDA=0,0    AT+DATACUTA=0,0,0    AT+CMDDLA=0
AT+COMMANDB=0,0    AT+DATACUTB=0,0,0    AT+CMDDLB=0
AT+COMMANDC=0,0    AT+DATACUTC=0,0,0    AT+CMDDLC=0
AT+COMMANDD=0,0    AT+DATACUTD=0,0,0    AT+CMDDLD=0
AT+COMMANDE=0,0    AT+DATACUTE=0,0,0    AT+CMDDLE=0
AT+COMMANDF=0,0    AT+DATACUTF=0,0,0    AT+CMDDLF=0

Start Tx events

OK
```



```
CMD1      = 01 03 00 10 00 08 45 c9
RETURN1   = 01 03 10 01 00 05 ff 00 00 00 00 01 03 00 00 00 00 00 00 86 fe
CMD2      = 01 02 00 40 00 10 78 12
RETURN2   = 01 02 02 20 00 a0 78
Payload   = 01 01 00 05 ff 00 00 00 00 01 03 00 00 00 00 00 00 20 00

[177893]***** UpLinkCounter= 4 *****
[177895]TX on freq 867700000 Hz at DR 3
[178145]RX on freq 869525000 Hz at DR 3
[178148]txDone
[179143]RX on freq 867700000 Hz at DR 3
[179183]RX on freq 869525000 Hz at DR 3
[179185]rxTimeOut
```

DATA1:8 register values

DATA2:16 register values

DATA:DATA1+DATA2

RS485 command timeout

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

Default value: 0, range: 0 ~ 65 seconds

- **AT Command:**

AT+CMDDLaa=hex(bb cc)*1000

Example:

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

- **Downlink Payload:**

0x AA aa bb cc Same as: AT+CMDDLaa=hex (bb cc)

Example:

0xAA 01 03 E8   --> Same as **AT+CMDDL1=1000 ms**

Uplink payload mode

Define to use one uplink or multiple uplinks for the sampling.

The use of this command please see: Compose Uplink payload

- **AT Command:**

AT+DATAUP=0

AT+DATAUP=1

0xAD 01 00 00 14 **-->** Same as AT+DATAUP=1, 20000 // (00 00 14 is 20 seconds)

Each uplink is sent to the server at 20-second intervals when segmented.

- **Downlink Payload:**

0xAD 00 **-->** Same as AT+DATAUP=0

0xAD 01 **-->** Same as AT+DATAUP=1 //Each uplink is sent to the server one after the other as it is segmented.

- **AT Command:**

AT+DATAUP=1,Timeout

- **Downlink Payload:**

0xAD 01 00 00 14 **-->** Same as AT+DATAUP=1, 20000 // (00 00 14 is 20 seconds)

Each uplink is sent to the server at 20-second intervals when segmented.

Manually trigger an Uplink

Ask device to send an uplink immediately.

- **AT Command:**

No AT Command for this, user can press the ACT button for 1 second for the same.

- **Downlink Payload:**

0x08 FF, RS485-LN will immediately send an uplink.

Clear RS485 Command

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

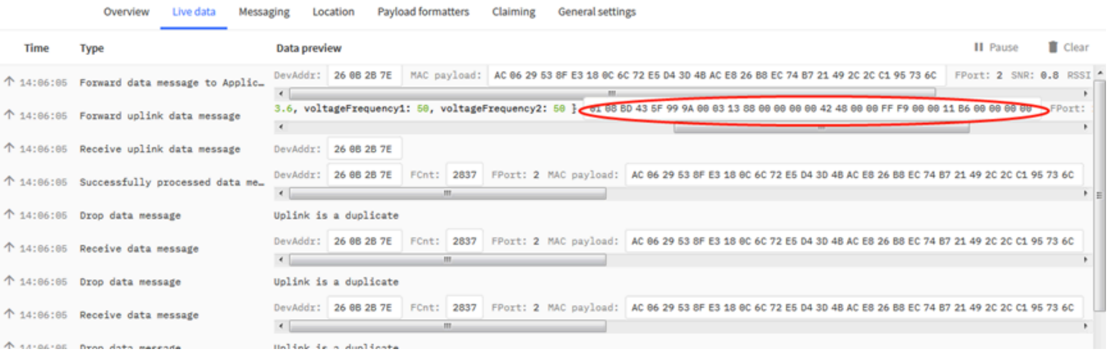
- **AT Command:**

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 to AT+COMMAND10/AT+DATACUT10

Example screen shot after clear all RS485 commands.

The uplink screen shot is:



```
Time      Type                              Data preview
14:06:05  Forward data message to Applic…   DevAddr: 26 0B 2B 7E  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C  FPort: 2  SNR: 0.8  RSSI
14:06:05  Forward uplink data message       3.6, voltageFrequency1: 50, voltageFrequency2: 50 } 01 08 BD 43 5F 99 9A 00 03 13 88 00 00 00 00 42 48 00 00 FF F9 00 00 11 B6 00 00 00 00  FPort:
14:06:05  Receive uplink data message       DevAddr: 26 0B 2B 7E
14:06:05  Successfully processed data me…   DevAddr: 26 0B 2B 7E  FCnt: 2837  FPort: 2  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C
14:06:05  Drop data message                 Uplink is a duplicate
14:06:05  Receive data message              DevAddr: 26 0B 2B 7E  FCnt: 2837  FPort: 2  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C
14:06:05  Drop data message                 Uplink is a duplicate
14:06:05  Receive data message              DevAddr: 26 0B 2B 7E  FCnt: 2837  FPort: 2  MAC payload: AC 06 29 53 8F E3 18 0C 6C 72 E5 D4 3D 4B AC E8 26 B8 EC 74 B7 21 49 2C 2C C1 95 73 6C
14:06:05  Drop data message                 Uplink is a duplicate
```

- **Downlink Payload:**

0x09 aa bb same as AT+CMDEAR=aa,bb

Set Serial Communication Parameters

Set the Rs485 serial communication parameters:

AT Command:

- Set Baud Rate

AT+BAUDR=9600   // Options: (200~115200) When using low baud rate or receiving multiple bytes, you need to use AT+CMDDL to increase the receive timeout (the default receive timeout is 400ms), otherwise data will be lost

- Set UART Parity

AT+PARITY=0 // Option: 0: no parity, 1: odd parity, 2: even parity

- Set STOPBIT

AT+STOPBIT=0 // Option: 0 for 1bit; 1 for 1.5 bit ; 2 for 2 bits

Downlink Payload:

A7 01 aa bb: Same AT+BAUDR=hex(aa bb)*100

Example:

- A7 01 00 60 same as AT+BAUDR=9600
- A7 01 04 80 same as AT+BAUDR=115200
- A7 02 aa: Same as AT+PARITY=aa (aa value: 00 , 01 or 02)
- A7 03 aa: Same as AT+STOPBIT=aa (aa value: 00 , 01 or 02)

Configure Databit (Since Version 1.4.0)

- **AT Command:**

AT+DATABIT=7 // Set the data bits to 7

AT+DATABIT=8 // Set the data bits to 8

- **Downlink Payload:**

A7 04 07: Same as AT+DATABIT=7

A7 04 08: Same as AT+DATABIT=8

Encrypted payload(Since Version 1.4.0)

- **AT Command:**

AT+DECRYPT=1 // The payload is uploaded without encryption

AT+DECRYPT=0 // Encrypt when uploading payload (default)

Get sensor value(Since Version 1.4.0)

- **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.

Resets the downlink packet count(Since Version 1.4.0)

- **AT Command:**

AT+DISFCNTCHECK=0 // When the downlink packet count sent by the server is less than the node downlink packet count or exceeds 16384, the node will no longer receive downlink packets (default)

AT+DISFCNTCHECK=1 // When the downlink packet count sent by the server is less than the node downlink packet count or exceeds 16384, the node resets the downlink packet count and keeps it consistent with the server downlink packet count.

When the limit bytes are exceeded, upload in batches(Since Version 1.4.0)

- **AT Command:**

AT+DISMACANS=0 // When the MACANS of the reply server plus the payload exceeds the maximum number of bytes of 11 bytes (DR0 of US915, DR2 of AS923, DR2 of AU195), the node will send a packet with a payload of 00 and a port of 4. (default)

AT+DISMACANS=1 // When the MACANS of the reply server plus the payload exceeds the maximum number of bytes of the DR, the node will ignore the MACANS and not reply, and only upload the payload part.

- **Downlink Payload:**

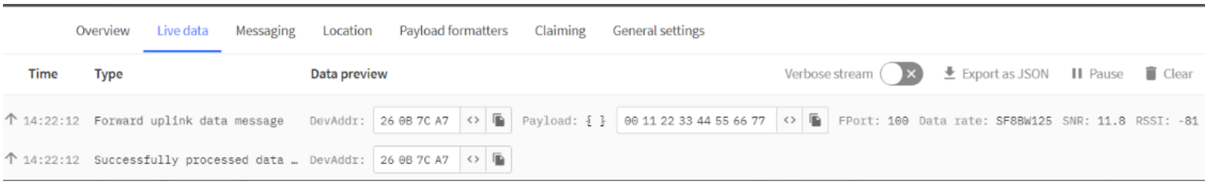
0x21 00 01 // Set the DISMACANS=1

Copy downlink to uplink (Since Version 1.4.0)

- **AT Command:**

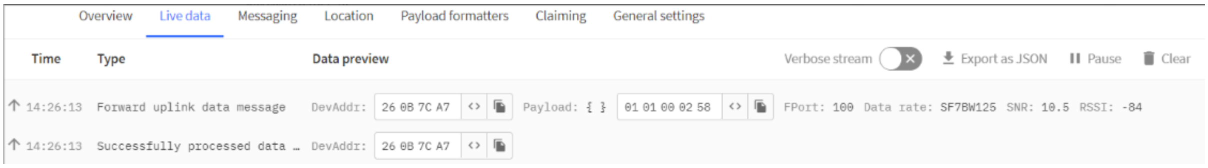
AT+RPL=5 // After receiving the package from the server, it will immediately upload the content of the package to the server, the port number is 100.

Example: **aa xx xx xx xx** // aa indicates whether the configuration has changed, 00 is yes, 01 is no; xx xx xx xx are the bytes sent.



```
↑ 14:22:12  Forward uplink data message    DevAddr: 26 0B 7C A7   Payload: { }  00 11 22 33 44 55 66 77   FPort: 100  Data rate: SF8BW125  SNR: 11.8  RSSI: -81
↑ 14:22:12  Successfully processed data …  DevAddr: 26 0B 7C A7
```

For example, sending 11 22 33 44 55 66 77 will return invalid configuration 00 11 22 33 44 55 66 77.



```
↑ 14:26:13  Forward uplink data message    DevAddr: 26 0B 7C A7   Payload: { }  01 01 00 02 58   FPort: 100  Data rate: SF7BW125  SNR: 10.5  RSSI: -84
↑ 14:26:13  Successfully processed data …  DevAddr: 26 0B 7C A7
```

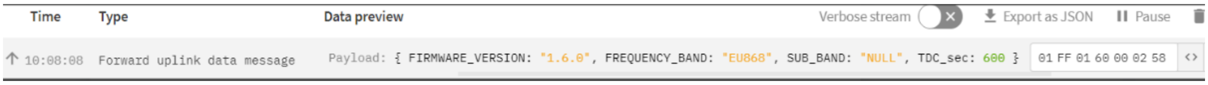
For example, if 01 00 02 58 is issued, a valid configuration of 01 01 00 02 58 will be returned.

Query version number and frequency band 、TDC(Since Version 1.4.0)

- **Downlink Payload:**

26 01 // Downlink 26 01 can query device upload frequency, frequency band, software version number, TDC time.

Example:



```
↑ 10:08:08  Forward uplink data message   Payload: { FIRMWARE_VERSION: "1.6.0", FREQUENCY_BAND: "EU868", SUB_BAND: "NULL", TDC_sec: 600 }  01 FF 01 60 00 02 58
```

Monitor RS485 communication of other devices(Since Version 1.4.0)

- **AT Command:**

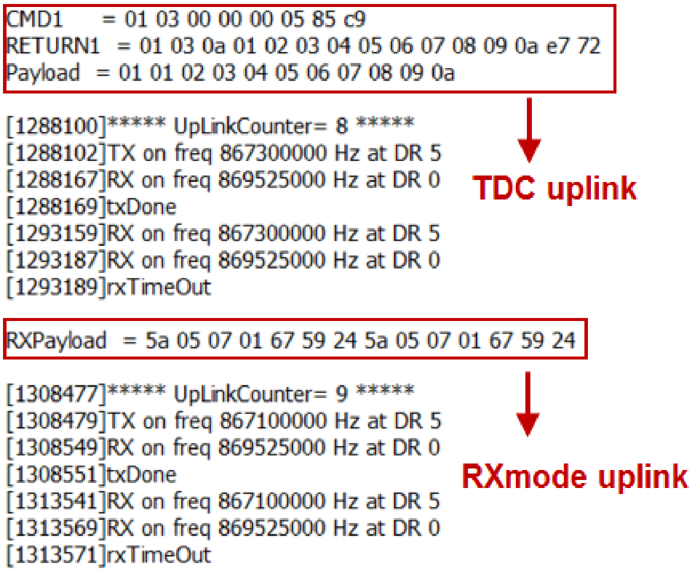
AT+RXMODE=1,10 // When the RS485-LN receives more than 10 bytes from the RS485, it immediately sends the uplink of the received data.

AT+RXMODE=2,500 // RS485-LN uploads data as uplink from the first byte received by RS485 to the data received within 500ms after that.

AT+RXMODE=0,0 // Disable this mode (default)

- **Downlink Payload:**

A6 aa bb bb // same as AT+RXMODE=aa,bb

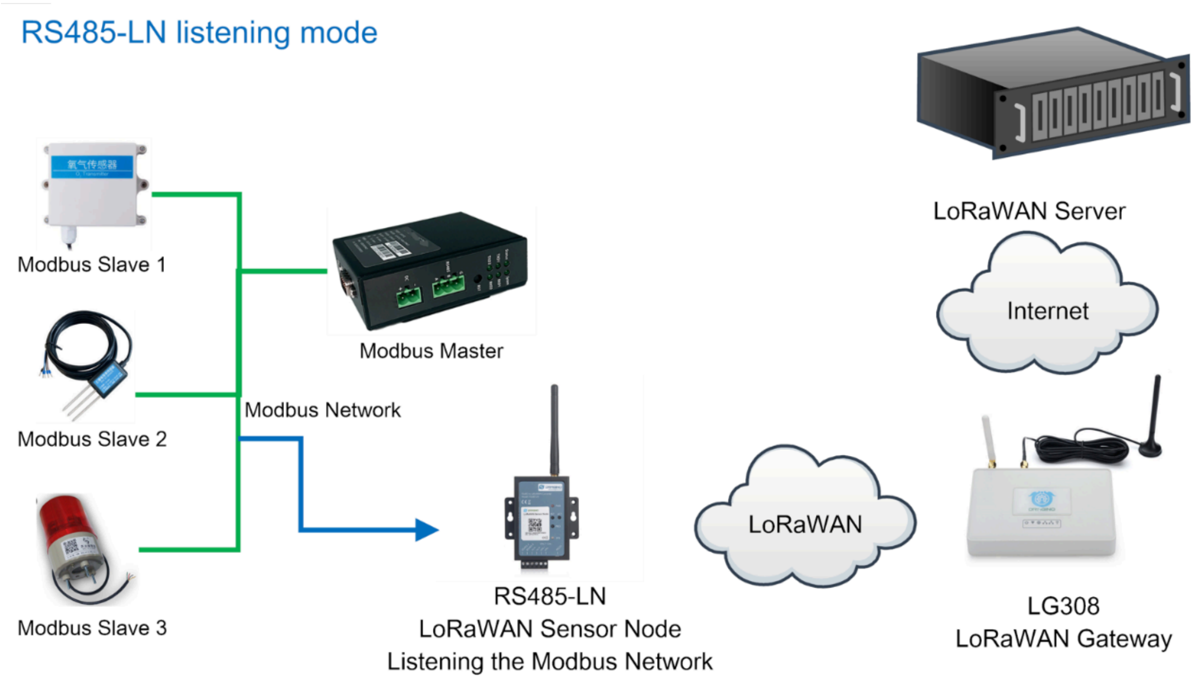


```
CMD1      = 01 03 00 00 00 05 85 c9
RETURN1  = 01 03 0a 01 02 03 04 05 06 07 08 09 0a e7 72
Payload  = 01 01 02 03 04 05 06 07 08 09 0a

[1288100]***** UpLinkCounter= 8 *****
[1288102]TX on freq 867300000 Hz at DR 5
[1288167]RX on freq 869525000 Hz at DR 0
[1288169]txDone
[1293159]RX on freq 867300000 Hz at DR 5
[1293187]RX on freq 869525000 Hz at DR 0
[1293189]rxTimeOut

RXPayload  = 5a 05 07 01 67 59 24 5a 05 07 01 67 59 24

[1308477]***** UpLinkCounter= 9 *****
[1308479]TX on freq 867100000 Hz at DR 5
[1308549]RX on freq 869525000 Hz at DR 0
[1308551]txDone
[1313541]RX on freq 867100000 Hz at DR 5
[1313569]RX on freq 869525000 Hz at DR 0
[1313571]rxTimeOut
```

3.6 Listening mode for RS485 network

This feature support since firmware v1.4

RS485-LN supports listening mode, it can listen the RS485 network packets and send them via LoRaWAN uplink. Below is the structure. The blue arrow shows the RS485 network packets to RS485-LN.



To enable the listening mode, use can run the command **AT+RXMODE**.

Command example	Function
AT+RXMODE=1,10	Enable listening mode 1, if RS485-LN has received more than 10 RS485 commands from the network. RS485-LN will send these commands via LoRaWAN uplinks.
AT+RXMODE=2,500	Enable listening mode 2, RS485-LN will capture and send a 500ms content once from the first detect of character. Max value is 65535 ms
AT+RXMODE=0,0	Disable listening mode. This is the default settings.
	A6 aa bb cc same as AT+RXMODE=aa,(bb<<8 | cc)

Downlink Command:

0xA6 aa bb cc same as AT+RXMODE=aa,(bb<<8 | cc)

Example:

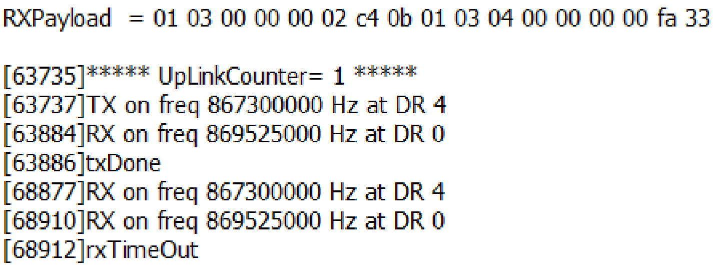
The RS485-LN is set to AT+RXMODE=2,1000

There is a two Modbus commands in the RS485 network as below:

The Modbus master send a command: **01 03 00 00 00 02 c4 0b**

And Modbus slave reply with: **01 03 04 00 00 00 00 fa 33**

RS485-LN will capture both and send the uplink: **01 03 00 00 00 02 c4 0b 01 03 04 00 00 00 00 fa 33**



```
RXPayload  = 01 03 00 00 00 02 c4 0b 01 03 04 00 00 00 00 fa 33

[63735]***** UpLinkCounter= 1 *****
[63737]TX on freq 867300000 Hz at DR 4
[63884]RX on freq 869525000 Hz at DR 0
[63886]txDone
[68877]RX on freq 867300000 Hz at DR 4
[68910]RX on freq 869525000 Hz at DR 0
[68912]rxTimeOut
```

Notice: Listening mode can work with the default polling mode of RS485-LN. When RS485-LN is in to send the RS485 commands (from AT+COMMANDx), the listening mode will be interrupt for a while.

3.7 Buttons

Button	Feature
ACT	If RS485 joined in network, press this button for more than 1 second, RS485 will upload a packet, and the SYS LED will give a **Blue blink**
RST	Reboot RS485
PRO	Use for upload image, see How to Update Image

3.8 LEDs

LEDs	Feature
PWR	Always on if there is power
SYS	After device is powered on, the SYS will **fast blink in GREEN** for 5 times, means RS485-LN start to join LoRaWAN network. If join success, SYS will be **on GREEN for 5 seconds** **.** SYS will **blink Blue** on every upload and **blink Green** once receive a downlink message.

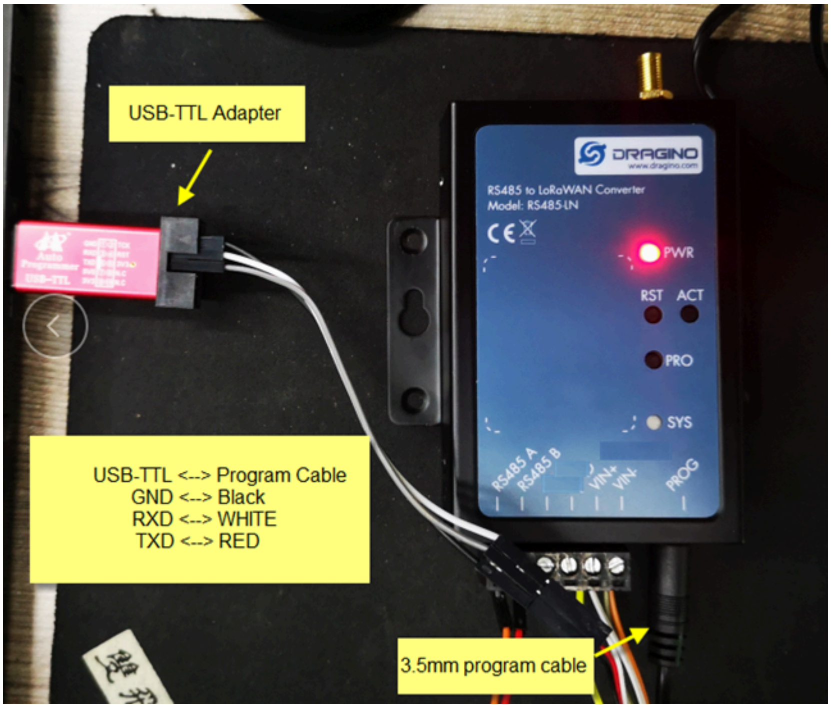
4. Case Study

User can check this URL for some case studies: APP RS485 COMMUNICATE WITH SENSORS

5. Use AT Command

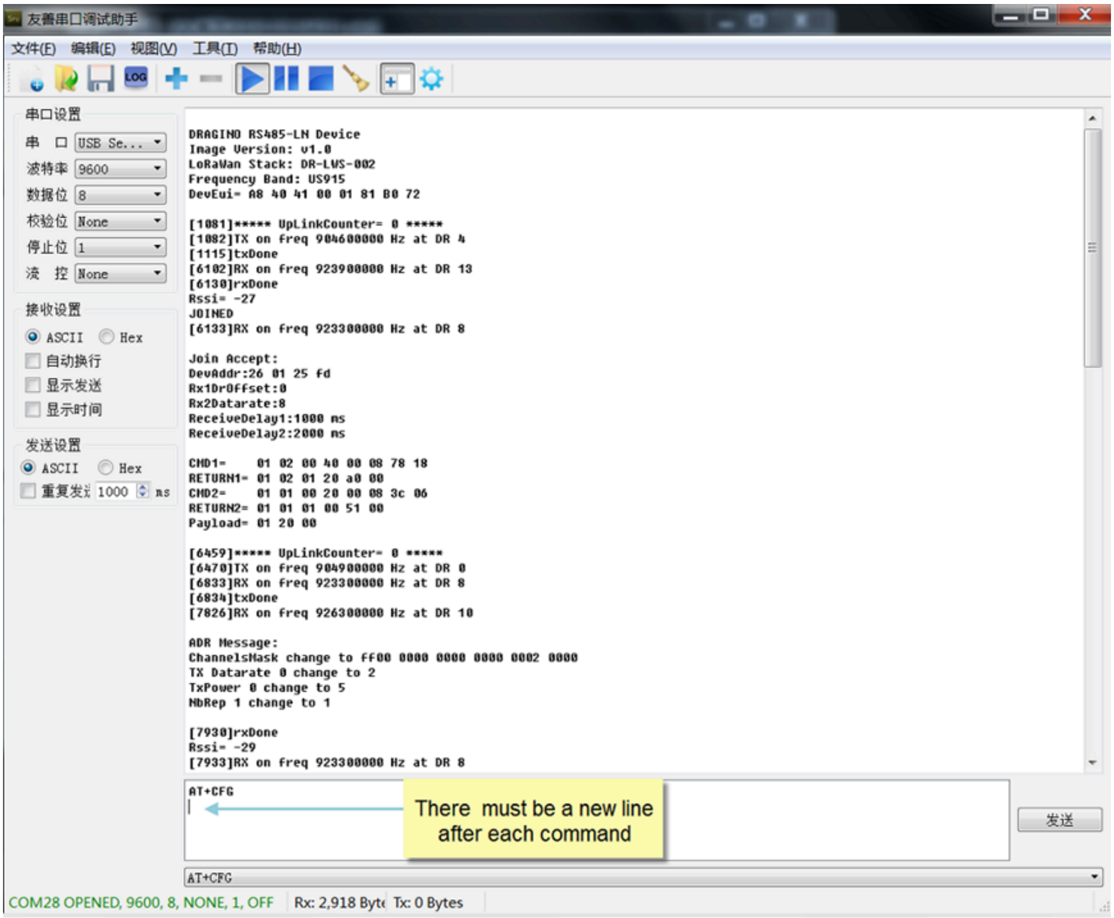
5.1 Access AT Command

RS485-LN supports AT Command set. User can use a USB to TTL adapter plus the 3.5mm Program Cable to connect to RS485-LN to use AT command, as below.



USB-TTL <--> Program Cable
GND <--> Black
RXD <--> WHITE
TXD <--> RED

In PC, User needs to set **serial tool**(such as putty, SecureCRT) baud rate to **9600** to access to access serial console of RS485-LN. The default password is 123456. Below is the output for reference:



串口设置
串　口 USB Se...
波特率 9600
数据位 8
校验位 None
停止位 1
流　控 None

接收设置
● ASCII ○ Hex
□ 自动换行
□ 显示发送
□ 显示时间

发送设置
● ASCII ○ Hex
□ 重复发送 1000 ms

```
DRAGINO RS485-LN Device
Image Version: v1.0
LoRaWan Stack: DR-LWS-002
Frequency Band: US915
DevEui= A8 40 41 00 01 81 B0 72

[1081]***** UpLinkCounter= 0 *****
[1082]TX on freq 904600000 Hz at DR 4
[1115]txDone
[6102]RX on freq 923900000 Hz at DR 13
[6130]rxDone
Rssi= -27
JOINED
[6133]RX on freq 923300000 Hz at DR 8

Join Accept:
DevAddr:26 01 25 fd
Rx1DrOffset:0
Rx2Datarate:8
ReceiveDelay1:1000 ms
ReceiveDelay2:2000 ms

CMD1=    01 02 00 40 00 08 78 18
RETURN1= 01 02 01 20 a0 00
CMD2=    01 01 00 20 00 08 3c 06
RETURN2= 01 01 01 00 51 00
Payload= 01 20 00

[6459]***** UpLinkCounter= 0 *****
[6470]TX on freq 904900000 Hz at DR 0
[6833]RX on freq 923300000 Hz at DR 8
[6834]txDone
[7826]RX on freq 926300000 Hz at DR 10

ADR Message:
ChannelsMask change to ff00 0000 0000 0000 0002 0000
TX Datarate 0 change to 2
TxPower 0 change to 5
NbRep 1 change to 1

[7930]rxDone
Rssi= -29
[7933]RX on freq 923300000 Hz at DR 8
```

```
AT+CFG
|
```

There must be a new line after each command

AT+CFG

COM28 OPENED, 9600, 8, NONE, 1, OFF　Rx: 2,918 Byte　Tx: 0 Bytes

More detail AT Command manual can be found at AT Command Manual

5.2 Common AT Command Sequence

5.2.1 Multi-channel ABP mode (Use with SX1301/LG308)

If device has not joined network yet:

- **AT+FDR**
- **AT+NJM=0**
- **ATZ**

If device already joined network:

- **AT+NJM=0**
- **ATZ**

5.5.2 Single-channel ABP mode (Use with LG01/LG02)

AT+FDR Reset Parameters to Factory Default, Keys Reserve

AT+NJM=0 Set to ABP mode

AT+ADR=0 Set the Adaptive Data Rate Off

AT+DR=5 Set Data Rate

AT+TDC=60000 Set transmit interval to 60 seconds

AT+CHS=868400000 Set transmit frequency to 868.4Mhz

AT+RX2FQ=868400000 Set RX2Frequency to 868.4Mhz (according to the result from server)

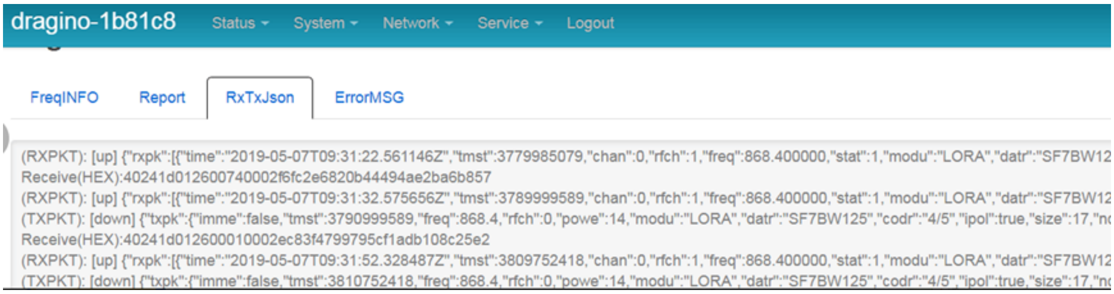
AT+RX2DR=5 Set RX2DR to match the downlink DR from server. see below

AT+DADDR=26 01 1A F1 Set Device Address to 26 01 1A F1, this ID can be found in the LoRa Server portal.

ATZ Reset MCU

Note:

1. Make sure the device is set to ABP mode in the IoT Server.
2. Make sure the LG01/02 gateway RX frequency is exactly the same as AT+CHS setting.
3. Make sure SF / bandwidth setting in LG01/LG02 match the settings of AT+DR. refer this link to see what DR means.
4. The command AT+RX2FQ and AT+RX2DR is to let downlink work. to set the correct parameters, user can check the actually downlink parameters to be used. As below. Which shows the RX2FQ should use 868400000 and RX2DR should be 5



```
(RXPKT): [up] {"rxpk":[{"time":"2019-05-07T09:31:22.561146Z","tmst":3779985079,"chan":0,"rfch":1,"freq":868.400000,"stat":1,"modu":"LORA","datr":"SF7BW12
Receive(HEX):40241d012600740002f6fc2e6820b44494ae2ba6b857
(RXPKT): [up] {"rxpk":[{"time":"2019-05-07T09:31:32.575656Z","tmst":3789999589,"chan":0,"rfch":1,"freq":868.400000,"stat":1,"modu":"LORA","datr":"SF7BW12
(TXPKT): [down] {"txpk":{"imme":false,"tmst":3790999589,"freq":868.4,"rfch":0,"powe":14,"modu":"LORA","datr":"SF7BW125","codr":"4/5","ipol":true,"size":17,"n
Receive(HEX):40241d012600010002ec83f4799795cf1adb108c25e2
(RXPKT): [up] {"rxpk":[{"time":"2019-05-07T09:31:52.328487Z","tmst":3809752418,"chan":0,"rfch":1,"freq":868.400000,"stat":1,"modu":"LORA","datr":"SF7BW12
(TXPKT): [down] {"txpk":{"imme":false,"tmst":3810752418,"freq":868.4,"rfch":0,"powe":14,"modu":"LORA","datr":"SF7BW125","codr":"4/5","ipol":true,"size":17,"n
```

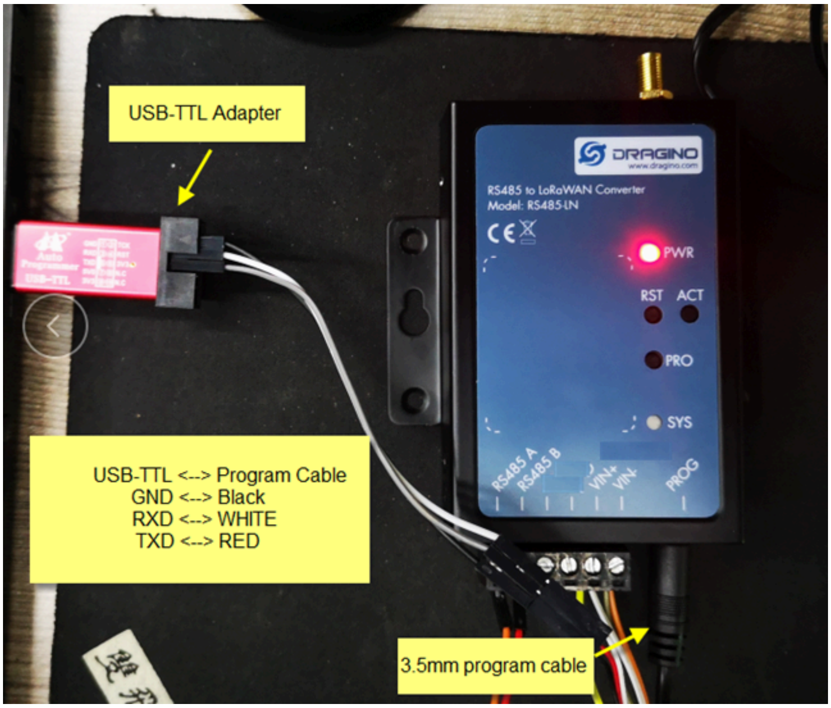
6. FAQ

6.1 How to upgrade the image?

The RS485-LN LoRaWAN Controller is shipped with a 3.5mm cable, the cable is used to upload image to RS485-LN to:

- Support new features
- For bug fix
- Change LoRaWAN bands.

Below shows the hardware connection for how to upload an image to RS485-LN:

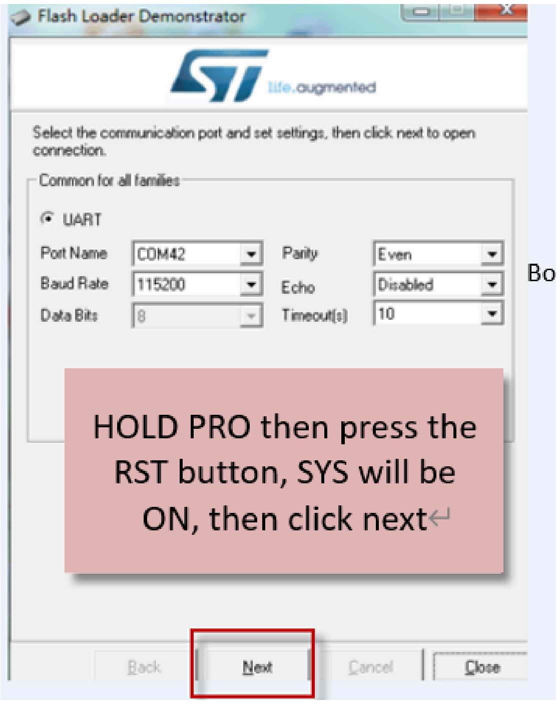


Step1: Download flash loader.

Step2: Download the LT Image files.

Step3: Open flashloader; choose the correct COM port to update.

Hold down the PRO button and then momentarily press the RST reset button and the SYS led will change from OFF to ON, While SYS LED is RED ON, it means the RS485-LN is ready to be program.



Flash Loader Demonstrator

Select the communication port and set settings, then click next to open connection.

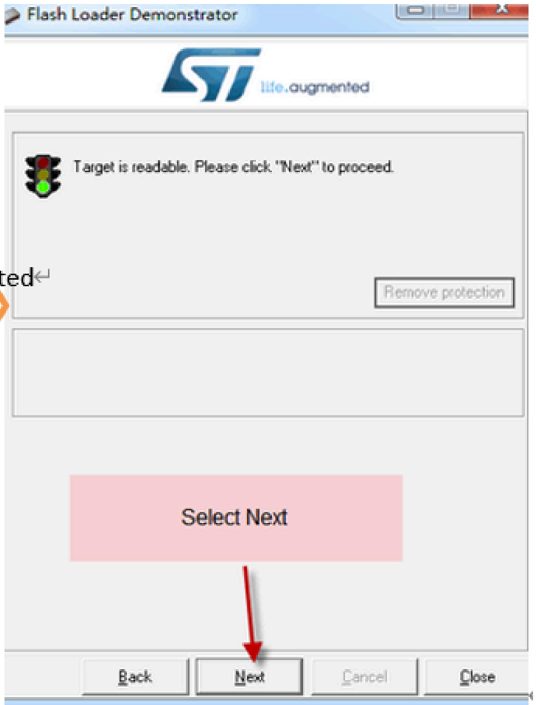
Common for all families

● UART

Port Name	COM42	Parity	Even
Baud Rate	115200	Echo	Disabled
Data Bits	8	Timeout(s)	10

HOLD PRO then press the RST button, SYS will be ON, then click next

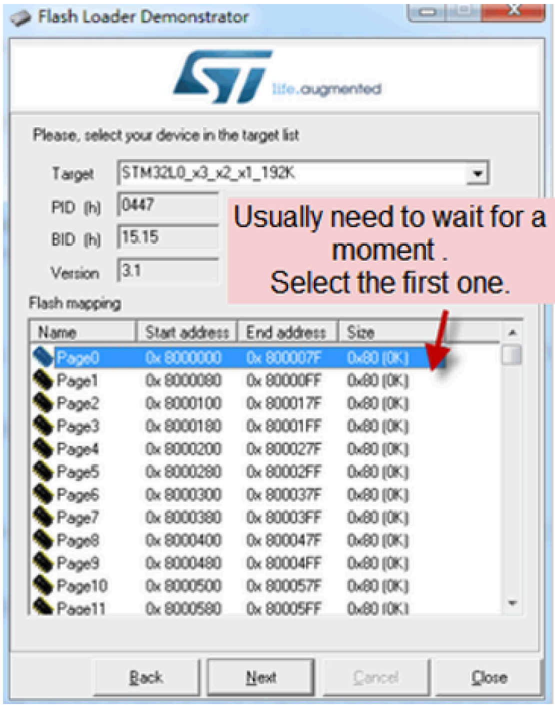
Board detected



Flash Loader Demonstrator

Target is readable. Please click "Next" to proceed.

Select Next



Flash Loader Demonstrator

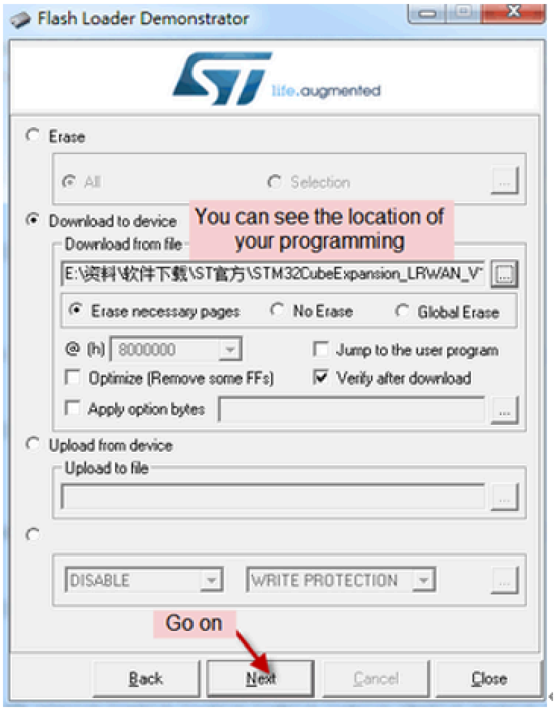
Please, select your device in the target list

Target	STM32L0_x3_x2_x1_192K
PID (h)	0447
BID (h)	15.15
Version	3.1

Usually need to wait for a moment . Select the first one.

Flash mapping

Name	Start address	End address	Size
Page0	0x 8000000	0x 800007F	0x80 (0K)
Page1	0x 8000080	0x 80000FF	0x80 (0K)
Page2	0x 8000100	0x 800017F	0x80 (0K)
Page3	0x 8000180	0x 80001FF	0x80 (0K)
Page4	0x 8000200	0x 800027F	0x80 (0K)
Page5	0x 8000280	0x 80002FF	0x80 (0K)
Page6	0x 8000300	0x 800037F	0x80 (0K)
Page7	0x 8000380	0x 80003FF	0x80 (0K)
Page8	0x 8000400	0x 800047F	0x80 (0K)
Page9	0x 8000480	0x 80004FF	0x80 (0K)
Page10	0x 8000500	0x 800057F	0x80 (0K)
Page11	0x 8000580	0x 80005FF	0x80 (0K)



Flash Loader Demonstrator

○ Erase

○ All ○ Selection

● Download to device

You can see the location of your programming

Download from file

E:\资料\软件下载\ST官方\STM32CubeExpansion_LRWAN_V

● Erase necessary pages ○ No Erase ○ Global Erase

@ (h) 8000000 □ Jump to the user program

□ Optimize (Remove some FFs) ☑ Verify after download

□ Apply option bytes

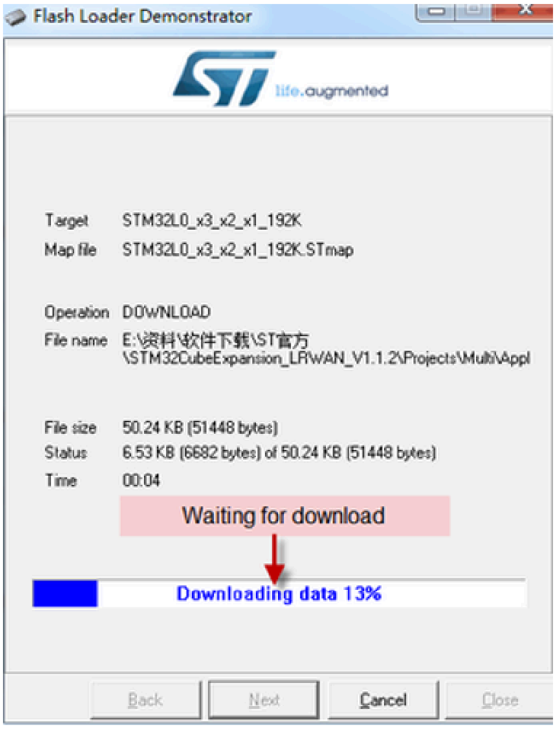
○ Upload from device

Upload to file

○

DISABLE WRITE PROTECTION

Go on



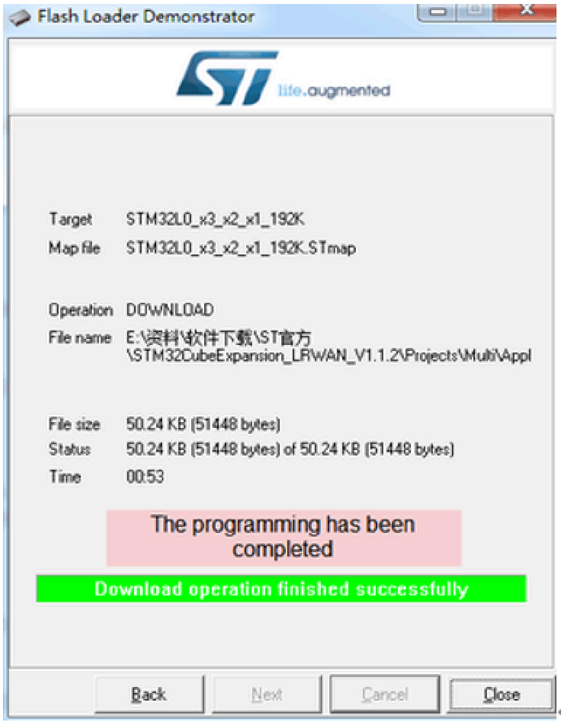
Target STM32L0_x3_x2_x1_192K
Map file STM32L0_x3_x2_x1_192K.STmap

Operation DOWNLOAD
File name E:\资料\软件下载\ST官方\STM32CubeExpansion_LRWAN_V1.1.2\Projects\Multi\Appl

File size 50.24 KB (51448 bytes)
Status 6.53 KB (6682 bytes) of 50.24 KB (51448 bytes)
Time 00:04

Waiting for download

Downloading data 13%



Target STM32L0_x3_x2_x1_192K
Map file STM32L0_x3_x2_x1_192K.STmap

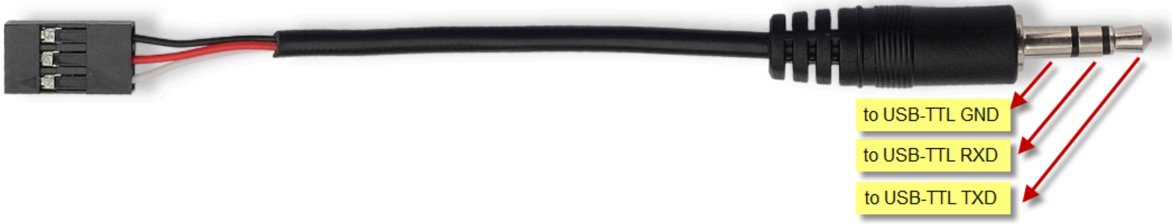
Operation DOWNLOAD
File name E:\资料\软件下载\ST官方\STM32CubeExpansion_LRWAN_V1.1.2\Projects\Multi\Appl

File size 50.24 KB (51448 bytes)
Status 50.24 KB (51448 bytes) of 50.24 KB (51448 bytes)
Time 00:53

The programming has been completed

Download operation finished successfully

Notice: In case user has lost the program cable. User can hand made one from a 3.5mm cable. The pin mapping is:



6.2 How to change the LoRa Frequency Bands/Region?

User can follow the introduction for how to upgrade image. When download the images, choose the required image file for download.

6.3 How many RS485-Slave can RS485-LN connects?

The RS485-LN can support max 32 RS485 devices. Each uplink command of RS485-LN can support max 16 different RS485 command. So RS485-LN 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.

6.4 Compatible question to ChirpStack and TTI LoRaWAN server ?

When user need to use with ChirpStack or TTI. Please set AT+RPL=4.

Detail info check this link: Set Packet Receiving Response Level

6.5 Can i use point to point communication for RS485-LN?

Yes, please updating point-to-point firmware， then refer Point to Point Communication for RS485-LN.

6.6 How to Use RS485-LN to connect to RS232 devices?

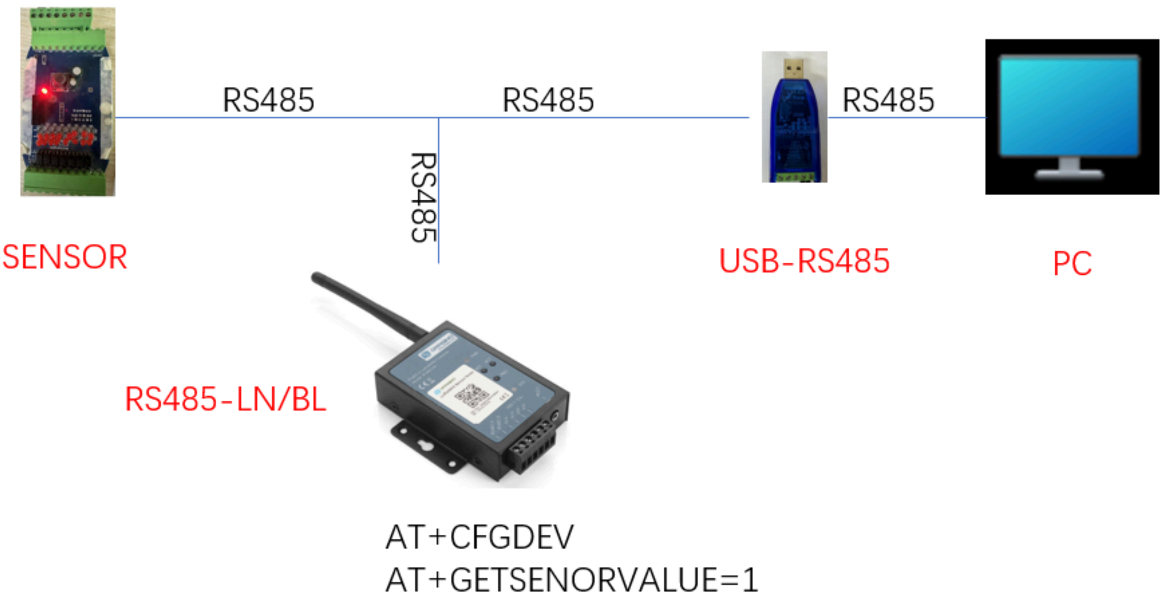
Use RS485-BL or RS485-LN to connect to RS232 devices. - DRAGINO

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

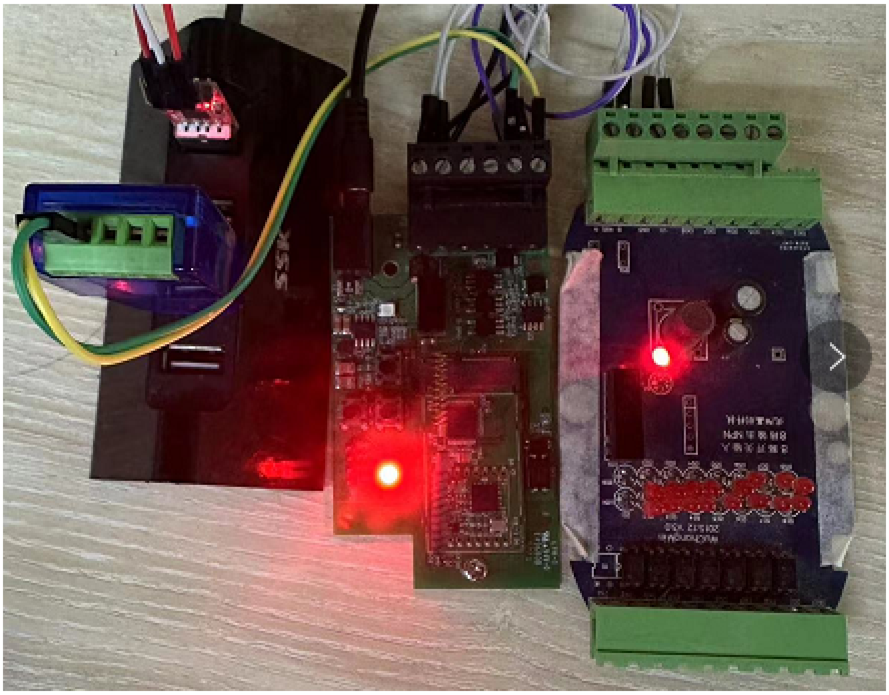
6.7.1 Introduce:

Users can use below the structure to fast debug the communication between RS485BL and RS485-LN. The principle is to put the PC in the RS485 network and sniff the packet between Modbus MTU and RS485-BL/LN. We can use this way to:

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

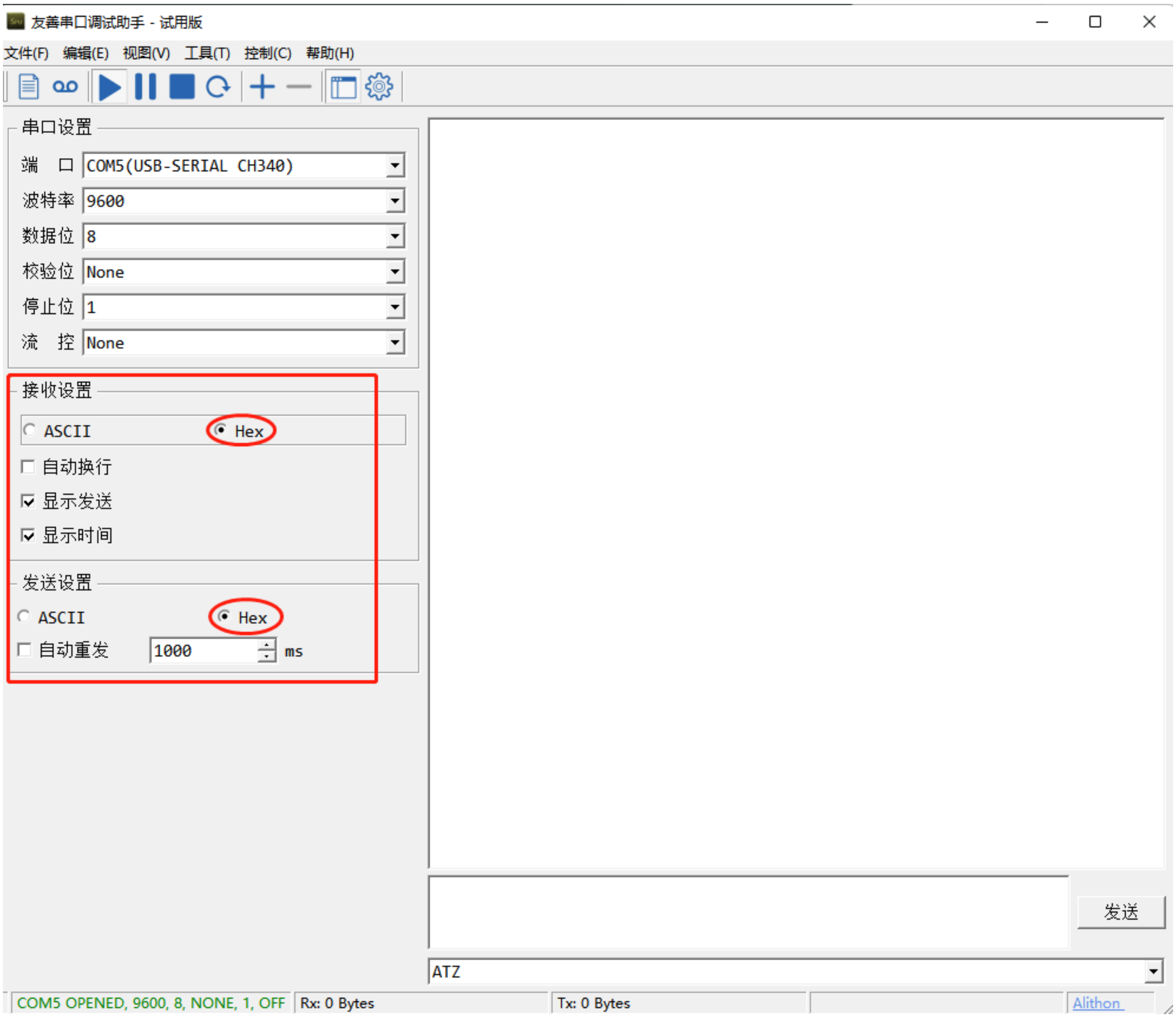


Example Connection:



6.7.2 Set up PC to monitor RS485 network With Serial tool

Note: Receive and send set to hex mode


友善串口调试助手 - 试用版
文件(F) 编辑(E) 视图(V) 工具(T) 控制(C) 帮助(H)
发送
Alithon

串口设置

端 口 COM5(USB-SERIAL CH340)

波特率 9600

数据位 8

校验位 None

停止位 1

流 控 None

接收设置

○ ASCII ● Hex

□ 自动换行

☑ 显示发送

☑ 显示时间

发送设置

○ ASCII ● Hex

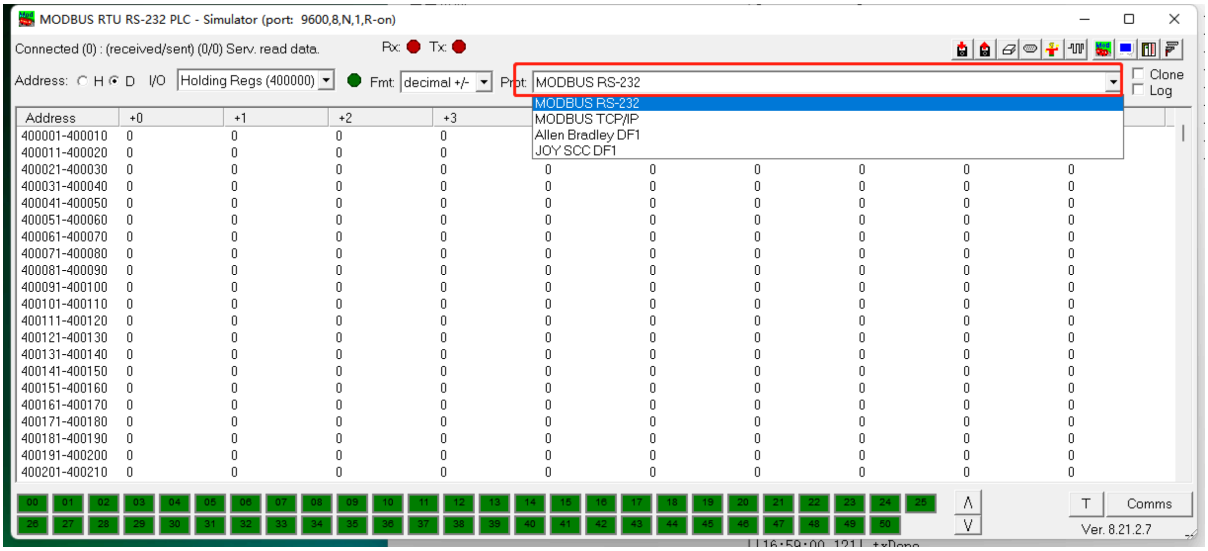
□ 自动重发 1000 ms

ATZ

COM5 OPENED, 9600, 8, NONE, 1, OFF | Rx: 0 Bytes | Tx: 0 Bytes

6.7.3 With ModRSsim2:

(1) Select serial port MODBUS RS-232



MODBUS RTU RS-232 PLC - Simulator (port: 9600,8,N,1,R-on)

Connected (0) : (received/sent) (0/0) Serv. read data.

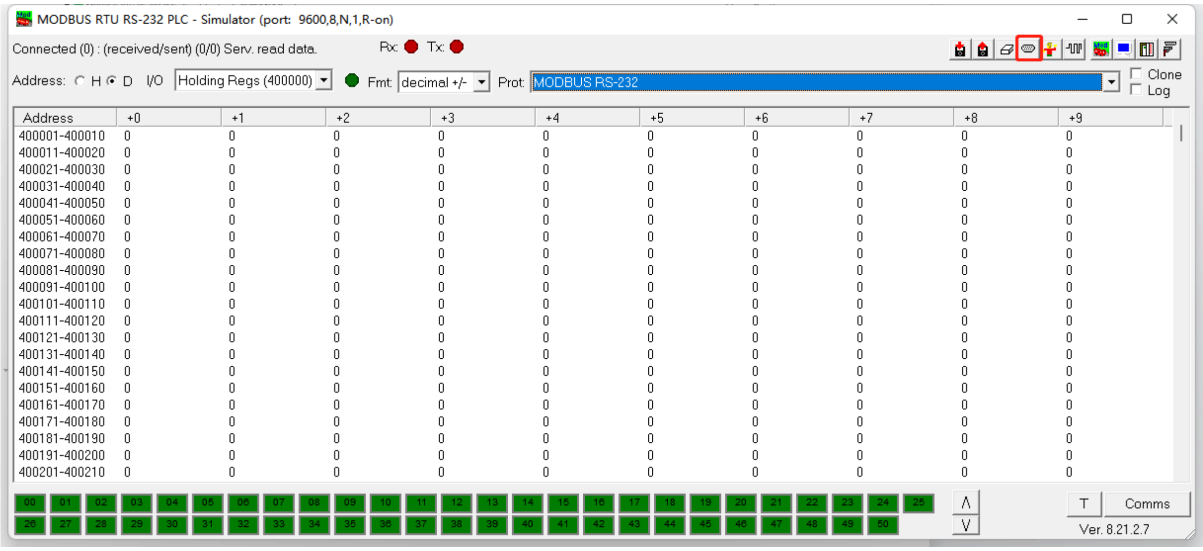
Address: ○ H ● D I/O Holding Regs (400000) Fmt: decimal +/- Prot: MODBUS RS-232

□ Clone
□ Log

MODBUS RS-232
MODBUS TCP/IP
Allen Bradley DF1
JOY SCC DF1

Address	+0	+1	+2	+3						
400001-400010	0	0	0	0						
400011-400020	0	0	0	0						
400021-400030	0	0	0	0	0	0	0	0	0	0
400031-400040	0	0	0	0	0	0	0	0	0	0
400041-400050	0	0	0	0	0	0	0	0	0	0
400051-400060	0	0	0	0	0	0	0	0	0	0
400061-400070	0	0	0	0	0	0	0	0	0	0
400071-400080	0	0	0	0	0	0	0	0	0	0
400081-400090	0	0	0	0	0	0	0	0	0	0
400091-400100	0	0	0	0	0	0	0	0	0	0
400101-400110	0	0	0	0	0	0	0	0	0	0
400111-400120	0	0	0	0	0	0	0	0	0	0
400121-400130	0	0	0	0	0	0	0	0	0	0
400131-400140	0	0	0	0	0	0	0	0	0	0
400141-400150	0	0	0	0	0	0	0	0	0	0
400151-400160	0	0	0	0	0	0	0	0	0	0
400161-400170	0	0	0	0	0	0	0	0	0	0
400171-400180	0	0	0	0	0	0	0	0	0	0
400181-400190	0	0	0	0	0	0	0	0	0	0
400191-400200	0	0	0	0	0	0	0	0	0	0
400201-400210	0	0	0	0	0	0	0	0	0	0

(2) Click the serial port icon



MODBUS RTU RS-232 PLC - Simulator (port: 9600,8,N,1,R-on)

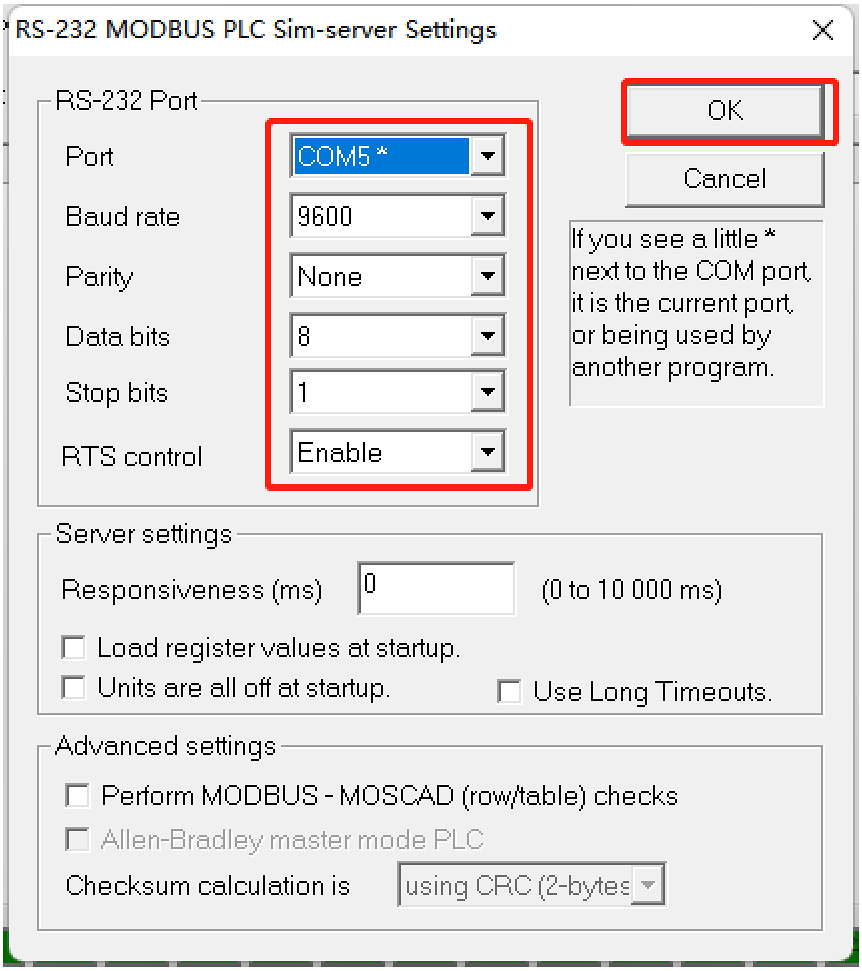
Connected (0) : (received/sent) (0/0) Serv. read data.

Address: ○ H ● D I/O Holding Regs (400000) Fmt: decimal +/- Prot: MODBUS RS-232

□ Clone
□ Log

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
400001-400010	0	0	0	0	0	0	0	0	0	0
400011-400020	0	0	0	0	0	0	0	0	0	0
400021-400030	0	0	0	0	0	0	0	0	0	0
400031-400040	0	0	0	0	0	0	0	0	0	0
400041-400050	0	0	0	0	0	0	0	0	0	0
400051-400060	0	0	0	0	0	0	0	0	0	0
400061-400070	0	0	0	0	0	0	0	0	0	0
400071-400080	0	0	0	0	0	0	0	0	0	0
400081-400090	0	0	0	0	0	0	0	0	0	0
400091-400100	0	0	0	0	0	0	0	0	0	0
400101-400110	0	0	0	0	0	0	0	0	0	0
400111-400120	0	0	0	0	0	0	0	0	0	0
400121-400130	0	0	0	0	0	0	0	0	0	0
400131-400140	0	0	0	0	0	0	0	0	0	0
400141-400150	0	0	0	0	0	0	0	0	0	0
400151-400160	0	0	0	0	0	0	0	0	0	0
400161-400170	0	0	0	0	0	0	0	0	0	0
400171-400180	0	0	0	0	0	0	0	0	0	0
400181-400190	0	0	0	0	0	0	0	0	0	0
400191-400200	0	0	0	0	0	0	0	0	0	0
400201-400210	0	0	0	0	0	0	0	0	0	0

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



RS-232 MODBUS PLC Sim-server Settings

RS-232 Port

Port: COM5 *

Baud rate: 9600

Parity: None

Data bits: 8

Stop bits: 1

RTS control: Enable

If you see a little * next to the COM port, it is the current port, or being used by another program.

Server settings

Responsiveness (ms): 0 (0 to 10 000 ms)

□ Load register values at startup.

□ Units are all off at startup.

□ Use Long Timeouts.

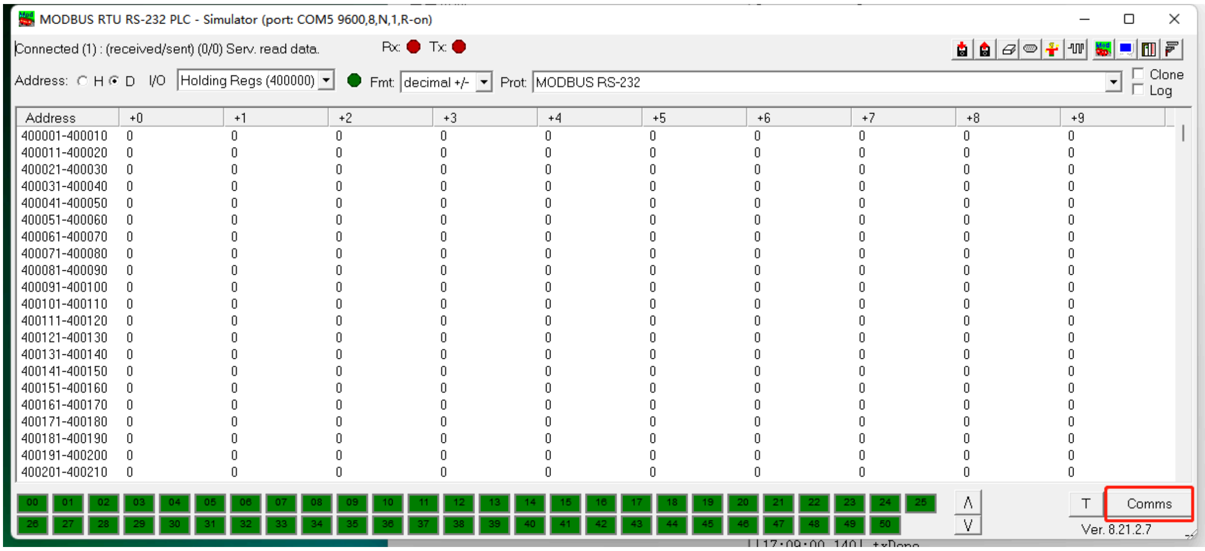
Advanced settings

□ Perform MODBUS - MOSCAD (row/table) checks

□ Allen-Bradley master mode PLC

Checksum calculation is: using CRC (2-bytes

(4) Click the comms.



MODBUS RTU RS-232 PLC - Simulator (port: COM5 9600,8,N,1,R-on)

Connected (1) : (received/sent) (0/0) Serv. read data.

Address: ○ H ● D I/O Holding Regs (400000) Fmt: decimal +/- Prot: MODBUS RS-232

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
400001-400010	0	0	0	0	0	0	0	0	0	0
400011-400020	0	0	0	0	0	0	0	0	0	0
400021-400030	0	0	0	0	0	0	0	0	0	0
400031-400040	0	0	0	0	0	0	0	0	0	0
400041-400050	0	0	0	0	0	0	0	0	0	0
400051-400060	0	0	0	0	0	0	0	0	0	0
400061-400070	0	0	0	0	0	0	0	0	0	0
400071-400080	0	0	0	0	0	0	0	0	0	0
400081-400090	0	0	0	0	0	0	0	0	0	0
400091-400100	0	0	0	0	0	0	0	0	0	0
400101-400110	0	0	0	0	0	0	0	0	0	0
400111-400120	0	0	0	0	0	0	0	0	0	0
400121-400130	0	0	0	0	0	0	0	0	0	0
400131-400140	0	0	0	0	0	0	0	0	0	0
400141-400150	0	0	0	0	0	0	0	0	0	0
400151-400160	0	0	0	0	0	0	0	0	0	0
400161-400170	0	0	0	0	0	0	0	0	0	0
400171-400180	0	0	0	0	0	0	0	0	0	0
400181-400190	0	0	0	0	0	0	0	0	0	0
400191-400200	0	0	0	0	0	0	0	0	0	0
400201-400210	0	0	0	0	0	0	0	0	0	0

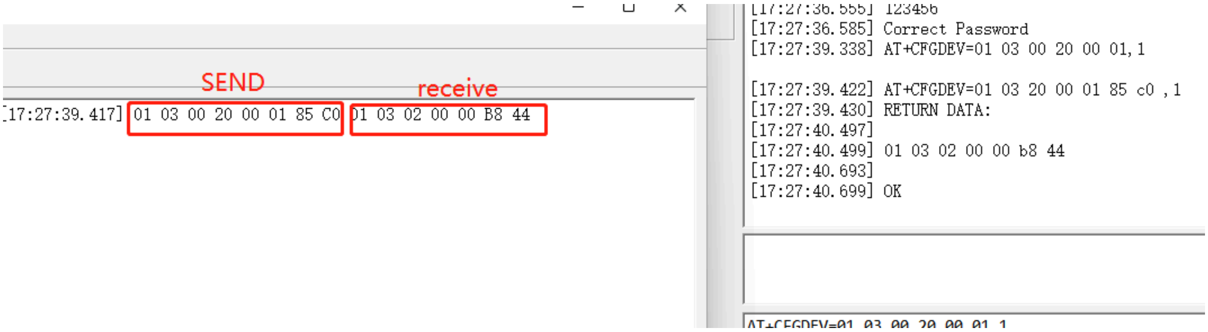
Run RS485-LN/BL command and monitor if it is correct.

6.7.4 Example – Test the CFGDEV command

RS485-LN sent below command:

AT+CFGDEV=01 03 00 20 00 01,1 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.

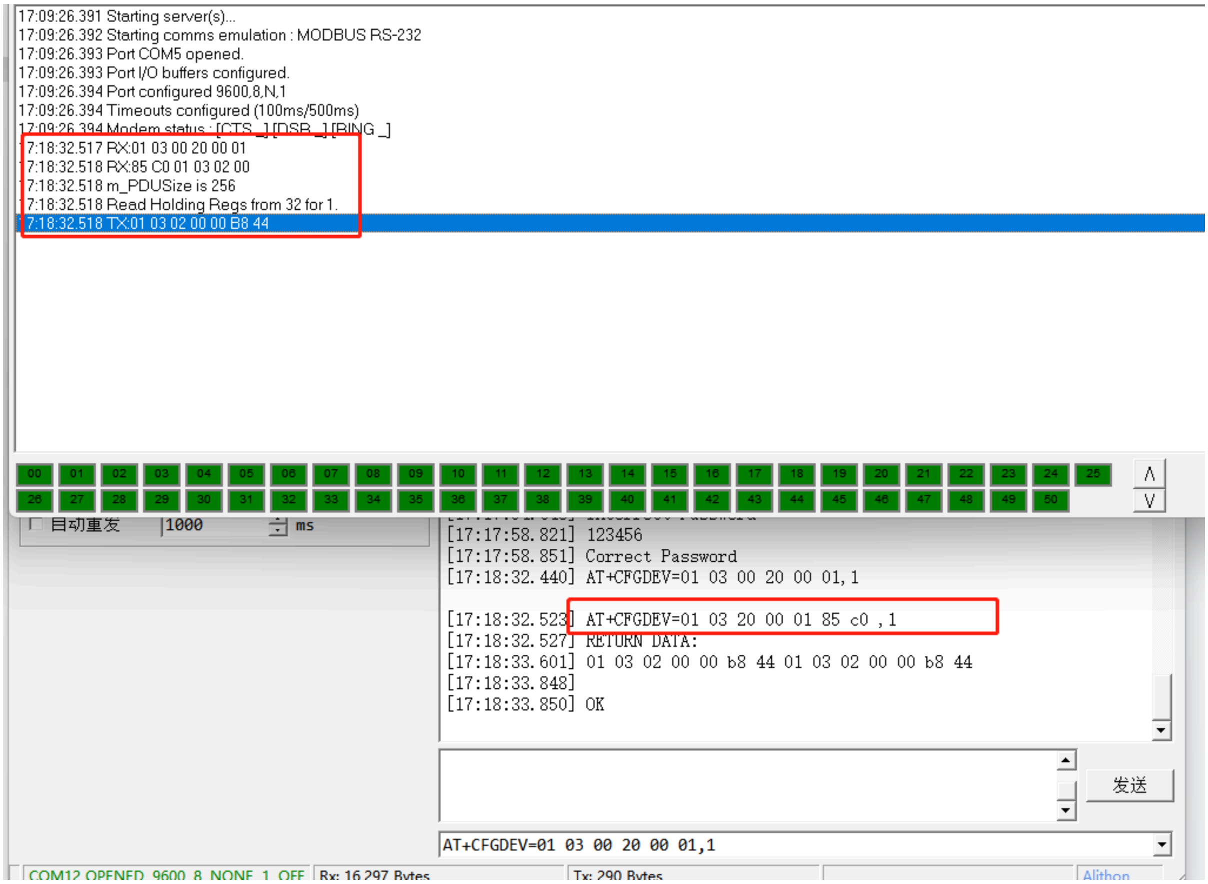


```
[17:27:39.417] 01 03 00 20 00 01 85 C0 01 03 02 00 00 B8 44
```

```
[17:27:36.555] 123456
[17:27:36.585] Correct Password
[17:27:39.338] AT+CFGDEV=01 03 00 20 00 01,1

[17:27:39.422] AT+CFGDEV=01 03 20 00 01 85 c0 ,1
[17:27:39.430] RETURN DATA:
[17:27:40.497]
[17:27:40.499] 01 03 02 00 00 b8 44
[17:27:40.693]
[17:27:40.699] OK
```

```
AT+CFGDEV=01 03 00 20 00 01,1
```

We can also use **ModRSsim2** to see the output.



```
17:09:26.391 Starting server(s)...
17:09:26.392 Starting comms emulation : MODBUS RS-232
17:09:26.393 Port COM5 opened.
17:09:26.393 Port I/O buffers configured.
17:09:26.394 Port configured 9600,8,N,1
17:09:26.394 Timeouts configured (100ms/500ms)
17:09:26.394 Modem status : [CTS _] [DSR _] [RING _]
7:18:32.517 RX:01 03 00 20 00 01
7:18:32.518 RX:85 C0 01 03 02 00
7:18:32.518 m_PDUSize is 256
7:18:32.518 Read Holding Regs from 32 for 1.
7:18:32.518 TX:01 03 02 00 00 B8 44
```

```
[17:17:58.821] 123456
[17:17:58.851] Correct Password
[17:18:32.440] AT+CFGDEV=01 03 00 20 00 01,1

[17:18:32.523] AT+CFGDEV=01 03 20 00 01 85 c0 ,1
[17:18:32.527] RETURN DATA:
[17:18:33.601] 01 03 02 00 00 b8 44 01 03 02 00 00 b8 44
[17:18:33.848]
[17:18:33.850] OK
```

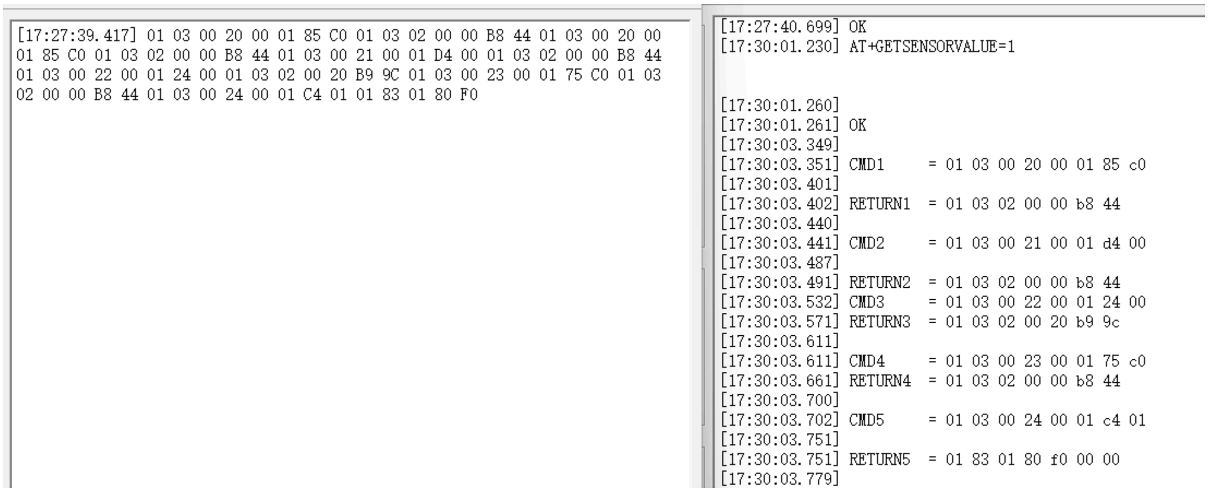
```
AT+CFGDEV=01 03 00 20 00 01,1
```

COM12 OPENED, 9600, 8, NONE, 1, OFF | Rx: 16,297 Bytes | Tx: 290 Bytes | Alithon

6.7.5 Example – Test CMD command sets.

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

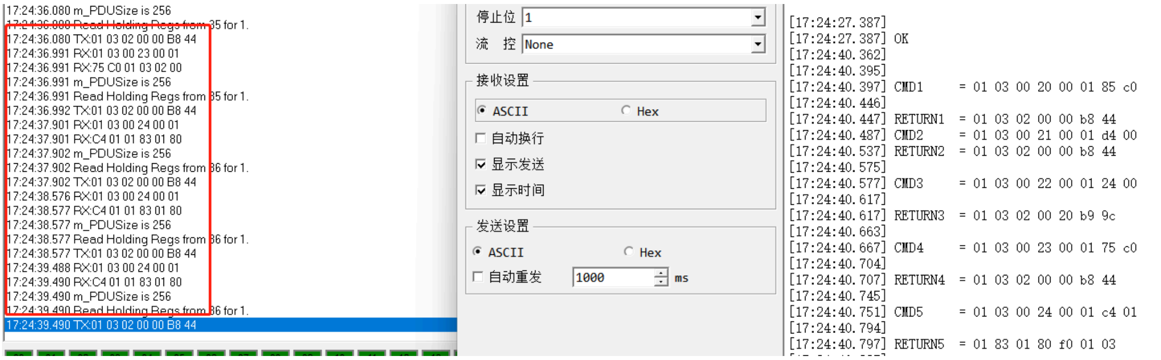
Serial port tool:



```
[17:27:39.417] 01 03 00 20 00 01 85 C0 01 03 02 00 00 B8 44 01 03 00 20 00
01 85 C0 01 03 02 00 00 B8 44 01 03 00 21 00 01 D4 00 01 03 02 00 00 B8 44
01 03 00 22 00 01 24 00 01 03 02 00 20 B9 9C 01 03 00 23 00 01 75 C0 01 03
02 00 00 B8 44 01 03 00 24 00 01 C4 01 01 83 01 80 F0
```

```
[17:27:40.699] OK
[17:30:01.230] AT+GETSENSORVALUE=1


[17:30:01.260]
[17:30:01.261] OK
[17:30:03.349]
[17:30:03.351] CMD1     = 01 03 00 20 00 01 85 c0
[17:30:03.401]
[17:30:03.402] RETURN1  = 01 03 02 00 00 b8 44
[17:30:03.440]
[17:30:03.441] CMD2     = 01 03 00 21 00 01 d4 00
[17:30:03.487]
[17:30:03.491] RETURN2  = 01 03 02 00 00 b8 44
[17:30:03.532] CMD3     = 01 03 00 22 00 01 24 00
[17:30:03.571] RETURN3  = 01 03 02 00 20 b9 9c
[17:30:03.611]
[17:30:03.611] CMD4     = 01 03 00 23 00 01 75 c0
[17:30:03.661] RETURN4  = 01 03 02 00 00 b8 44
[17:30:03.700]
[17:30:03.702] CMD5     = 01 03 00 24 00 01 c4 01
[17:30:03.751]
[17:30:03.751] RETURN5  = 01 83 01 80 f0 00 00
[17:30:03.779]
```

ModRSsim2:



```
17:24:36.080 m_PDUSize is 256
17:24:36.080 Read Holding Regs from 35 for 1.
17:24:36.080 TX:01 03 02 00 00 B8 44
17:24:36.991 RX:01 03 00 23 00 01
17:24:36.991 RX:75 C0 01 03 02 00
17:24:36.991 m_PDUSize is 256
17:24:36.991 Read Holding Regs from 35 for 1.
17:24:36.992 TX:01 03 02 00 00 B8 44
17:24:37.901 RX:01 03 00 24 00 01
17:24:37.901 RX:C4 01 01 83 01 80
17:24:37.902 m_PDUSize is 256
17:24:37.902 Read Holding Regs from 36 for 1.
17:24:37.902 TX:01 03 02 00 00 B8 44
17:24:38.576 RX:01 03 00 24 00 01
17:24:38.577 RX:C4 01 01 83 01 80
17:24:38.577 m_PDUSize is 256
17:24:38.577 Read Holding Regs from 36 for 1.
17:24:38.577 TX:01 03 02 00 00 B8 44
17:24:39.488 RX:01 03 00 24 00 01
17:24:39.490 RX:C4 01 01 83 01 80
17:24:39.490 m_PDUSize is 256
17:24:39.490 Read Holding Regs from 36 for 1.
17:24:39.490 TX:01 03 02 00 00 B8 44
```

停止位 1
流　控 None

接收设置
● ASCII ○ Hex
□ 自动换行
☑ 显示发送
☑ 显示时间

发送设置
● ASCII ○ Hex
□ 自动重发 1000 ms

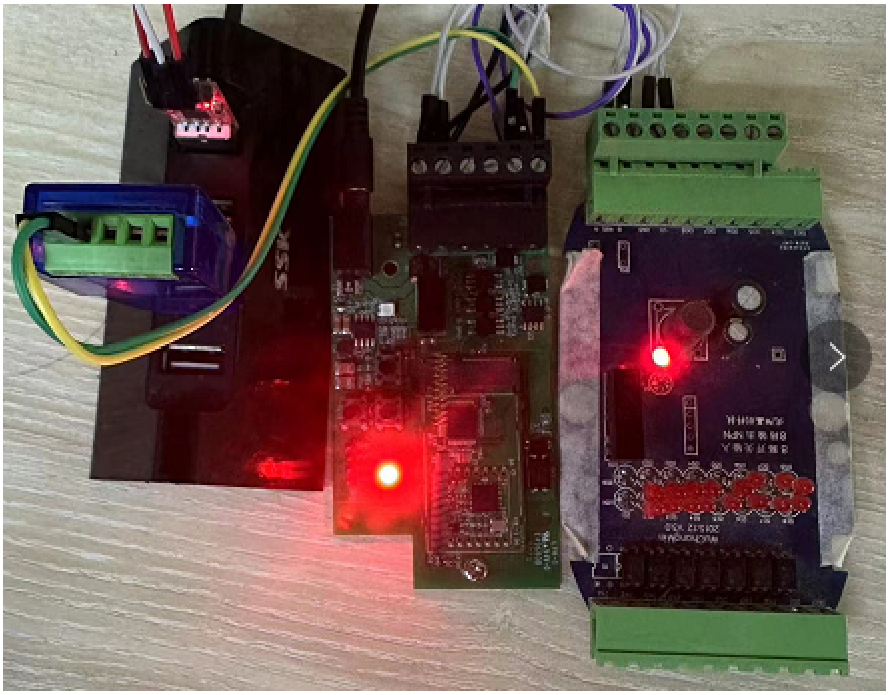
```
[17:24:27.387]
[17:24:27.387] OK
[17:24:40.362]
[17:24:40.395]
[17:24:40.397] CMD1     = 01 03 00 20 00 01 85 c0
[17:24:40.446]
[17:24:40.447] RETURN1  = 01 03 02 00 00 b8 44
[17:24:40.487] CMD2     = 01 03 00 21 00 01 d4 00
[17:24:40.537] RETURN2  = 01 03 02 00 00 b8 44
[17:24:40.575]
[17:24:40.577] CMD3     = 01 03 00 22 00 01 24 00
[17:24:40.617]
[17:24:40.617] RETURN3  = 01 03 02 00 20 b9 9c
[17:24:40.663]
[17:24:40.667] CMD4     = 01 03 00 23 00 01 75 c0
[17:24:40.704]
[17:24:40.707] RETURN4  = 01 03 02 00 00 b8 44
[17:24:40.745]
[17:24:40.751] CMD5     = 01 03 00 24 00 01 c4 01
[17:24:40.794]
[17:24:40.797] RETURN5  = 01 83 01 80 f0 01 03
```

6.7.6 Test with PC

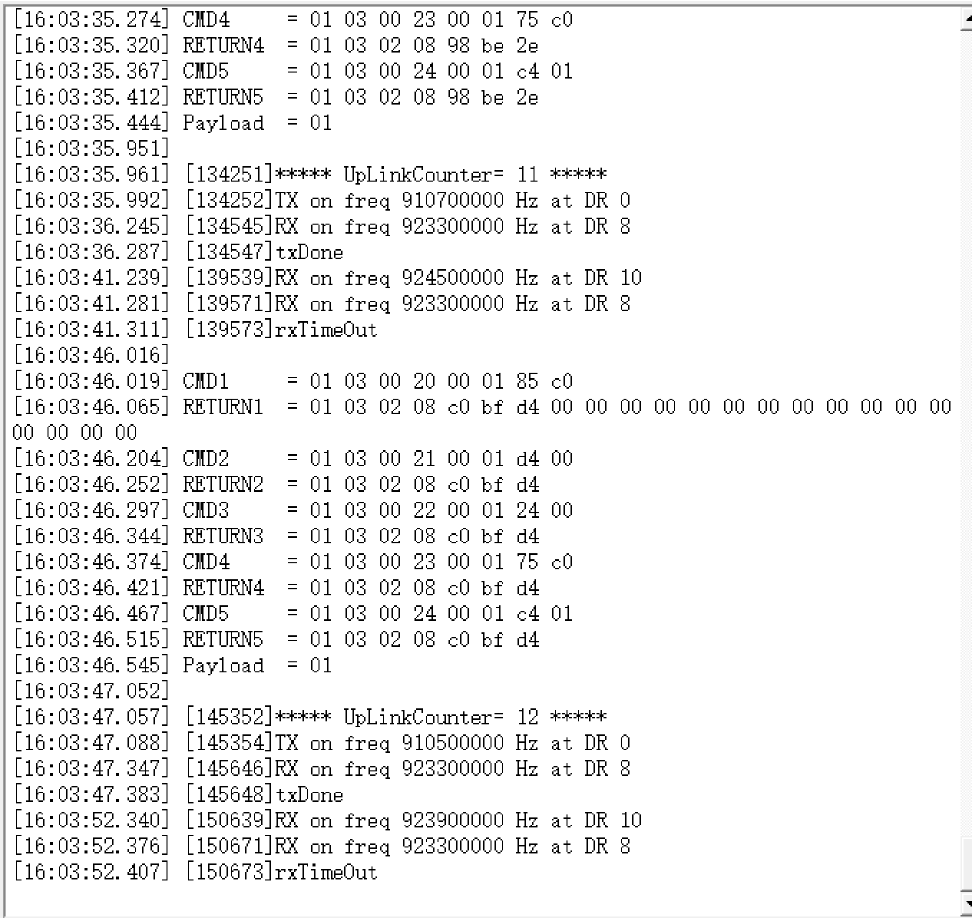
If there is still have problem to set up correctly the commands between RS485-LN and MTU. User can test the correct RS485 command set in PC and compare with the RS485 command sent out via RS485-LN. as long as both commands are the same, the MTU should return correct result.

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

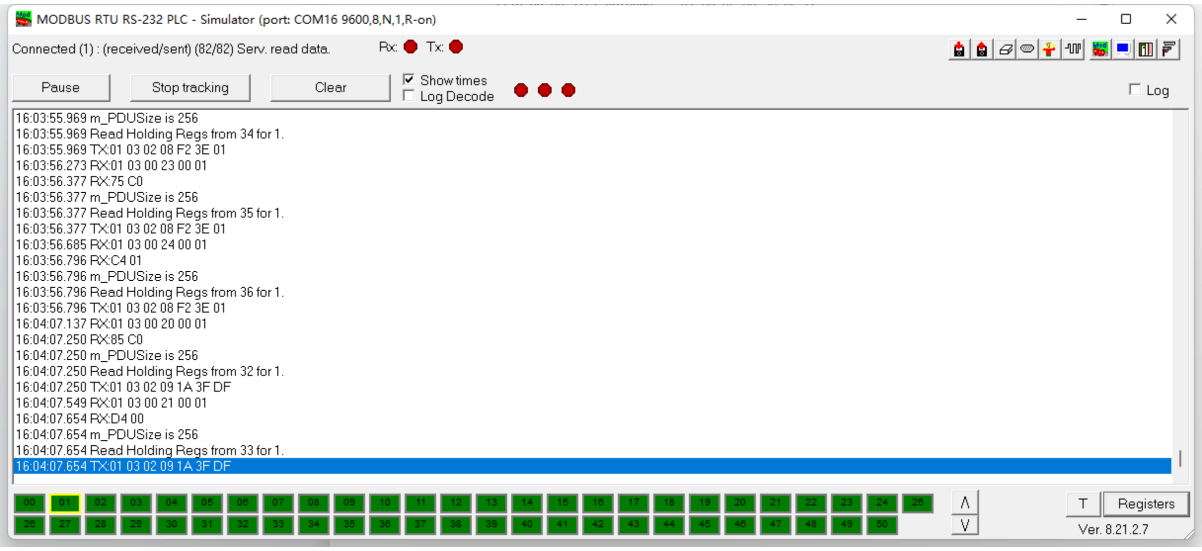
Connection method:



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
```



MODBUS RTU RS-232 PLC - Simulator (port: COM16 9600,8,N,1,R-on)

Connected (1) : (received/sent) (82/82) Serv. read data.

☑ Show times
□ Log Decode
□ Log

```
16:03:55.969 m_PDUSize is 256
16:03:55.969 Read Holding Regs from 34 for 1.
16:03:55.969 TX:01 03 02 08 F2 3E 01
16:03:56.273 RX:01 03 00 23 00 01
16:03:56.377 RX:75 C0
16:03:56.377 m_PDUSize is 256
16:03:56.377 Read Holding Regs from 35 for 1.
16:03:56.377 TX:01 03 02 08 F2 3E 01
16:03:56.685 RX:01 03 00 24 00 01
16:03:56.796 RX:C4 01
16:03:56.796 m_PDUSize is 256
16:03:56.796 Read Holding Regs from 36 for 1.
16:03:56.796 TX:01 03 02 08 F2 3E 01
16:04:07.137 RX:01 03 00 20 00 01
16:04:07.250 RX:85 C0
16:04:07.250 m_PDUSize is 256
16:04:07.250 Read Holding Regs from 32 for 1.
16:04:07.250 TX:01 03 02 09 1A 3F DF
16:04:07.549 RX:01 03 00 21 00 01
16:04:07.654 RX:D4 00
16:04:07.654 m_PDUSize is 256
16:04:07.654 Read Holding Regs from 33 for 1.
16:04:07.654 TX:01 03 02 09 1A 3F DF
```

Ver. 8.21.2.7

6.8 Where to get the decoder for RS485-LN?

The decoder for RS485-LN needs to be written by yourself. Because the sensor to which the user is connected is custom, the read device data bytes also need custom parsing, so there is no universal decoder. We can only provide templates for decoders (no intermediate data parsing part involved)

7. Trouble Shooting

7.1 Downlink doesn't work, how to solve it?

Please see this link for debug: LoRaWAN Communication Debug

7.2 Why I can't join TTN V3 in US915 /AU915 bands?

It might about the channels mapping. Please see for detail: Notice of Frequency band

8. Order Info

Part Number: RS485-LN-XXX

XXX:

- **EU433**: frequency bands EU433
- **EU868**: frequency bands EU868
- **KR920**: frequency bands KR920
- **CN470**: frequency bands CN470

- **AS923**: frequency bands AS923
- **AU915**: frequency bands AU915
- **US915**: frequency bands US915
- **IN865**: frequency bands IN865
- **RU864**: frequency bands RU864
- **KZ865**: frequency bands KZ865

9. Packing Info

Package Includes:

- RS485-LN x 1
- Stick Antenna for LoRa RF part x 1
- Program cable x 1

Dimension and weight:

- Device Size: 13.5 x 7 x 3 cm
- Device Weight: 105g
- Package Size / pcs : 14.5 x 8 x 5 cm
- Weight / pcs : 170g

10. FCC Caution for RS485LN-US915

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

IMPORTANT NOTE:

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

—Reorient or relocate the receiving antenna.

—Increase the separation between the equipment and receiver.

—Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

—Consult the dealer or an experienced radio/TV technician for help.

FCC 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 20cm between the radiator& your body.

11. 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.com.