RS485-LN -- RS485 to LoRaWAN Converter User Manual

Document Version: 1.3

Image Version: v1.3.0

Version	Description	Date
1.0	Release	2019-Dec-8
1.0.1	Improve product photos and network structure	2019-Dec-30
1.0.2	Add AT Command to set UART parity	
1.1	Upgrade manual for v1.1 firmware version	2020-Feb-19
1.2	Add downlink command 0x09, Add Command AT+CMDDL	2020-May-16
1.3	Remove FAQ for LG01, Add FAQ for slave numbers, Add Case Study Link. Update content for firmware v1.3	2020-Dec-5

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, smartphone detection, 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.
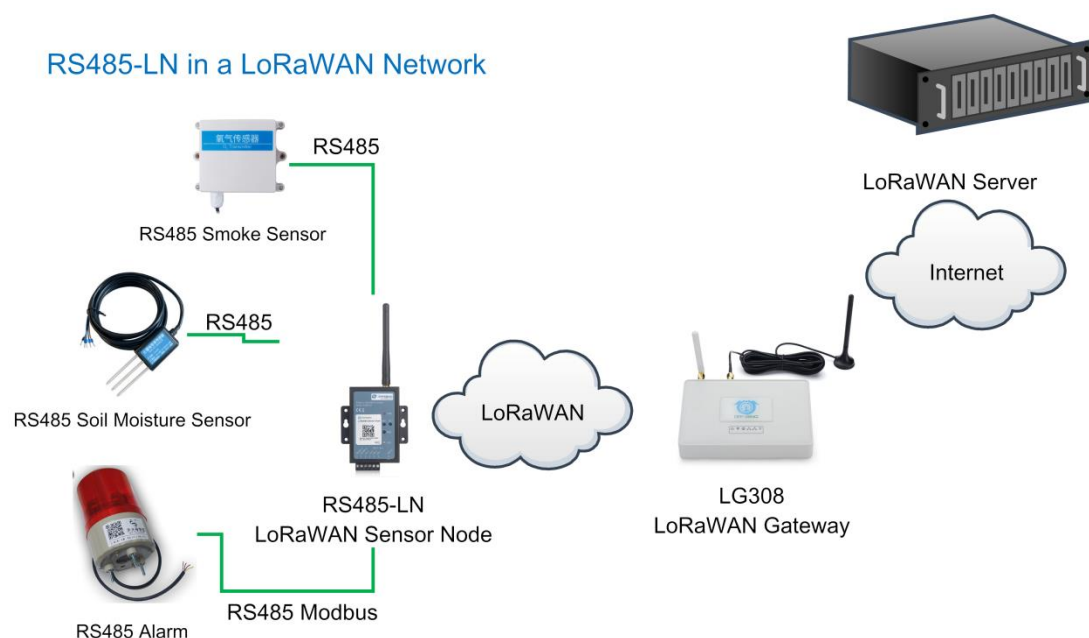
1.2 Specifications

Hardware System:

- STM32L072CZT6 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
- ✓ 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)

RS485 to LoRaWAN Converter User Manual

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 Network. Below is the network structure, we use our LG308 as LoRaWAN gateway here.
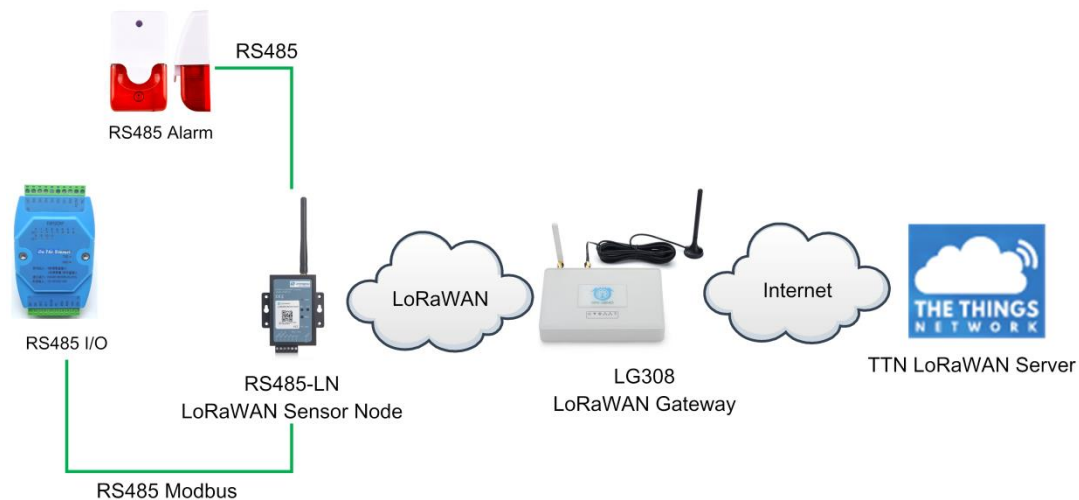
The RS485-LN in this example connected to two RS485 devices for demonstration, user can connect to other RS485 devices via the same method. The connection is as below:
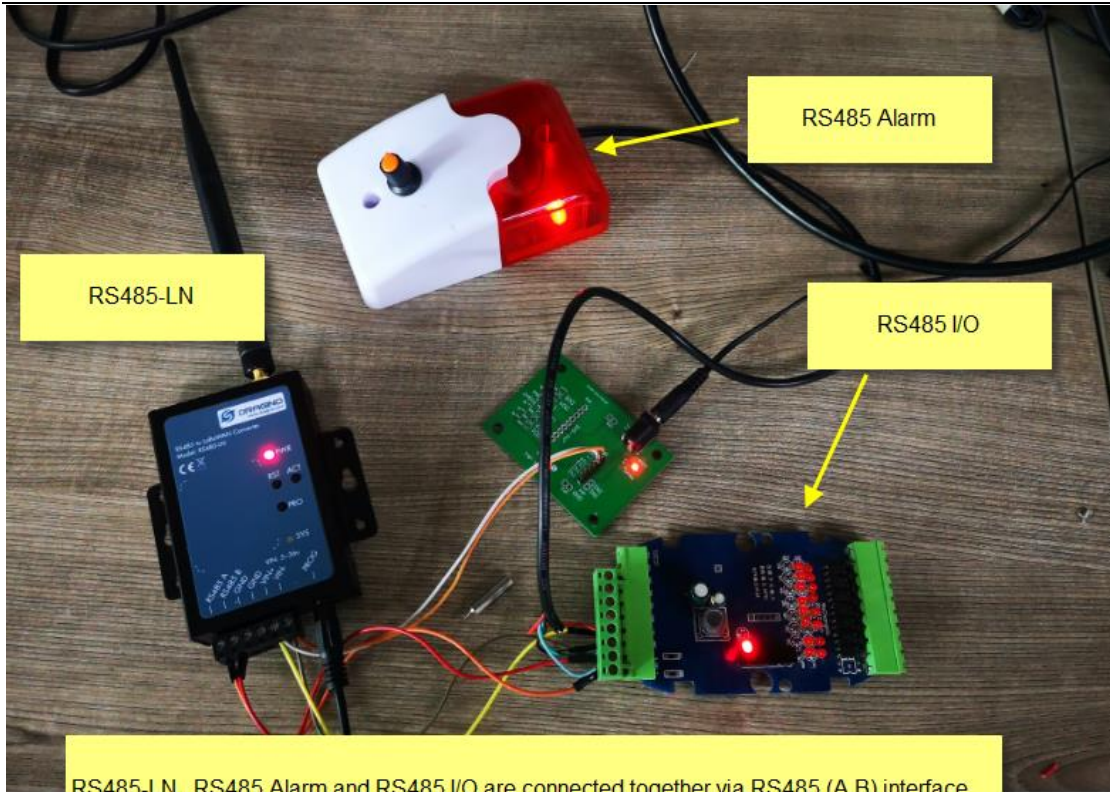
RS485-LN , RS485 Alarm and RS485 I/O are connected together via RS485 (A,B) interface.
1) RS485-LN will read RS485 I/O's registers periodically and uplink to LoRaWAN Server
2) LoRaWAN server can send an downlink command to RS485-Ln to control the RS485 Alarm

The LG308 is already set to connect to TTN network . So what we need to now is only configure the TTN:

Step 1: Create a device in TTN 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 screen shot:

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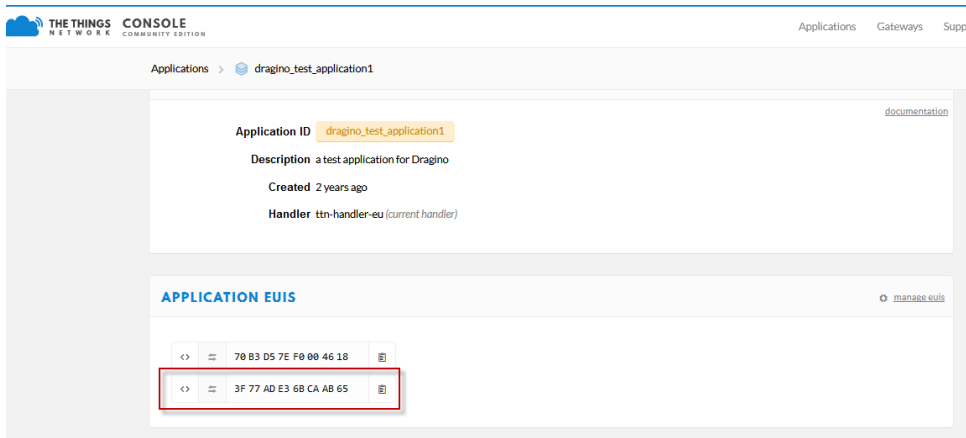
Add APP KEY and DEV EUI
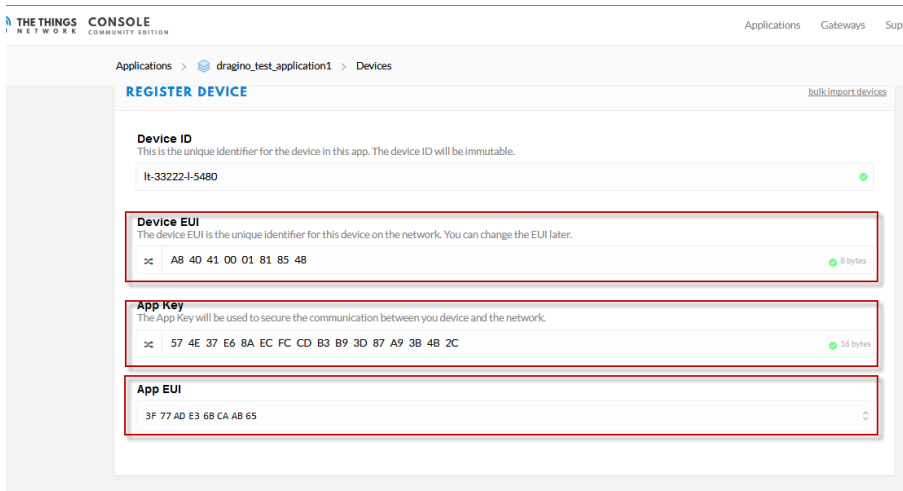
Step 2: Power on RS485-LN and it will auto join to the TTN network. After join success, it will start to upload message to TTN and user can see in the panel.

RS485 to LoRaWAN Converter User Manual
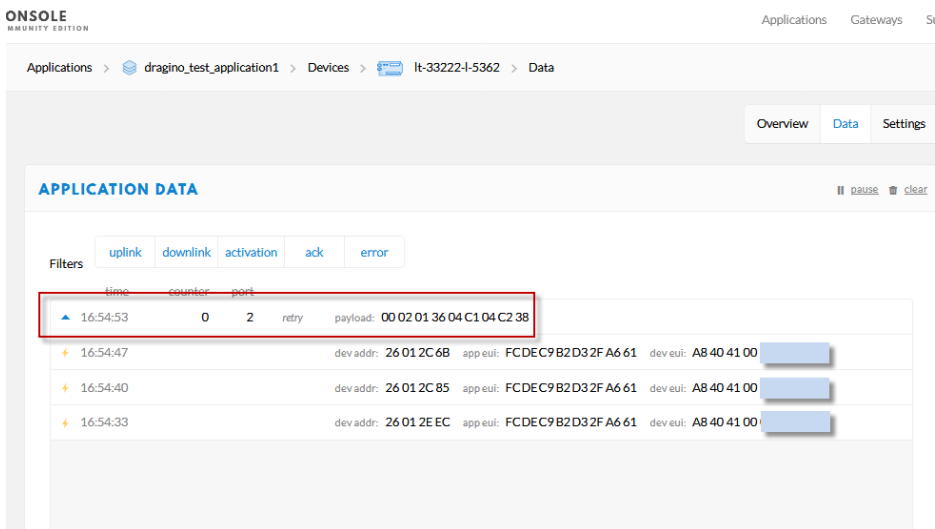
3.3 Configure Commands to read data

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 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 RS485 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 devices; they won't be used during sampling. AT+CFGDEV=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	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=<u>01 03 0B B8 00 02,1</u> for the same.

AT+DATACUTx : This command defines how to handle the return from AT+COMMANDx, max return length is 45 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+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

```
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
```

➢ 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+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

```
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
```

➢ 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

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+DATACUT C=0,0,0
AT+COMMANDD=0,0      AT+DATACU Valid value from RETURN10=20 20 20 20 30 2e 33 20 75 41 20
AT+COMMANDE=0,0      AT+DATACUTE=0,0,0
AT+COMMANDF=0,0      AT+DATACUTF=0,0,0
AT+CHS= Valid value from RETURN1=20 20 30 32 20 20

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
```

PAYVER   Valid Value from   Valid Value from
**   Return1   Return10**

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 Valid value from RETURN10=02 aa 05 81 0a 20 20 20 20 2d 30
AT+COMMANDD=0,0        AT+DATACUTD=0,0,0
AT  Valid value from RETURN1=20 20 0a 33 90 41
A
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 20 2d 30 2e 34 20 00
Payload  = 01 03 00 20 20 0a 33 90 41 02 aa

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

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.

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
```

First Uplink

Second Uplink

Third Uplink

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.
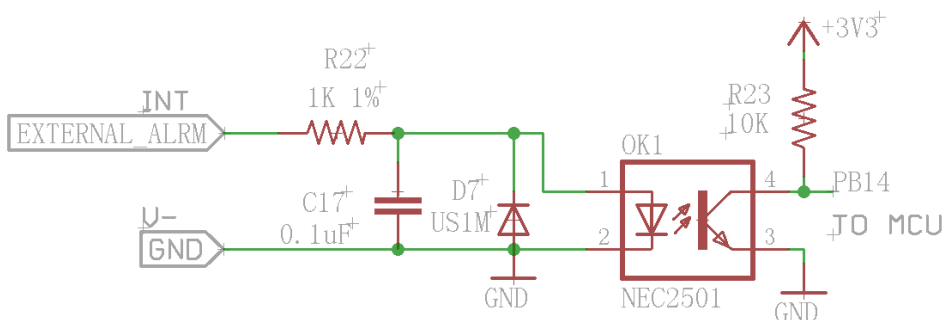
Downlink control command:

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.

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: http://wiki.dragino.com/index.php?title=End_Device_AT_Commands_and_Downlink_Commands

✓ **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:
http://wiki.dragino.com/index.php?title=End_Device_AT_Commands_and_Downlink_Commands

3.5.2 Sensor related commands:

RS485 Debug Command

This command is used to configure the RS485 devices; they won't be used during sampling.

➢ AT Command

AT+CFGDEV=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

For example,

To connect a Modbus Alarm with below commands.

✓ The command to active alarm is: <u>0A 05 00 04 00 01</u> 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: <u>0A 05 00 04 00 00</u> 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.

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

Set RS485 Sampling Commands

AT+COMMANDx or AT+DATACUTx

These two commands are use 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 00 86 fe    ---> DATA1:8 register values
CMD2    = 01 02 00 40 00 10 78 12    ---> DATA2:16 register values
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 00 20 00    ---> DATA:DATA1+DATA2

[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
```

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)*1000

Example:

0xAA 01 00 01   ➔ 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

➢ Downlink Payload:

0xAD 00   ➔ Same as AT+DATAUP=0

0xAD 01   ➔ Same as AT+DATAUP=1

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:

RS485 to LoRaWAN Converter User Manual

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:

		link	activation	ack	error
Filters					
	time	counter	port		
▲ 16:36:00	252	2		payload: 02 20 00	
▲ 16:35:40	251	2		payload: 02 20 00	
▲ 16:35:20	250	2		payload: 02 20 00	
▲ 16:35:00	249	2		payload: 02 20 00	

➢ 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: (1200,2400,4800,14400,19200,115200)

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)

3.6 Buttons

Button	Feature

RS485 to LoRaWAN Converter User Manual

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

3.7 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.

http://wiki.dragino.com/index.php?title=APP_RS485_COMMUNICATE_WITH_SENSORS
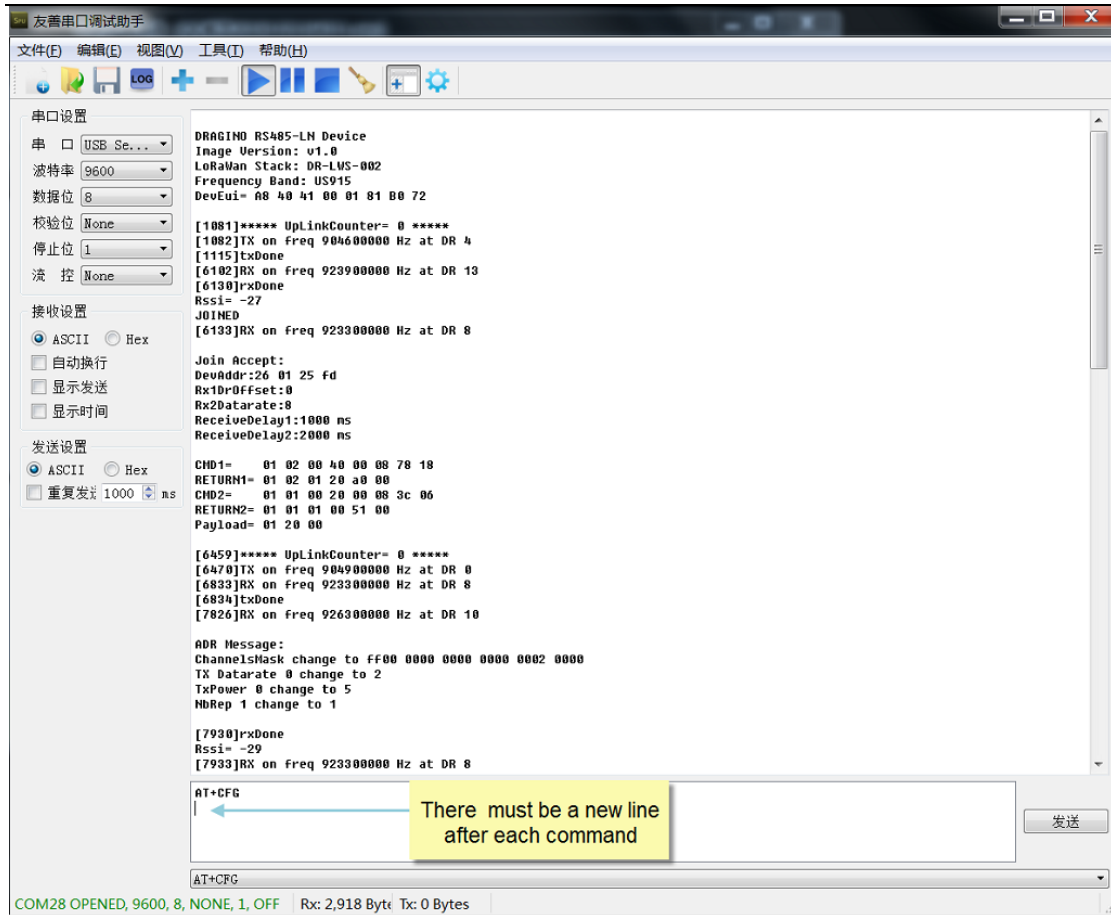
5. Use AT Command

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.
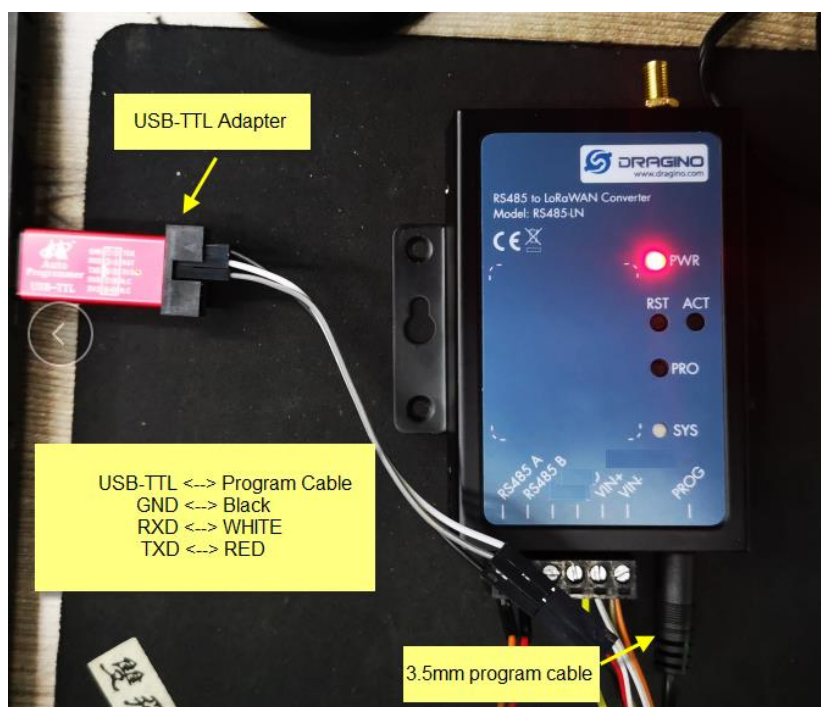
In PC, User needs to set **serial tool**(such as putty, SecureCRT) baud rate to **9600** to access to access serial console of RS485-LN. The default password is 123456, or 0 if upgrade from old version. Below is the output for reference:

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

5.2.2 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
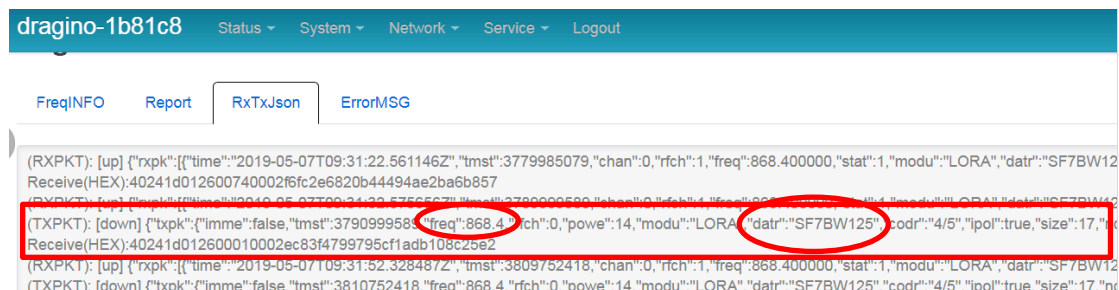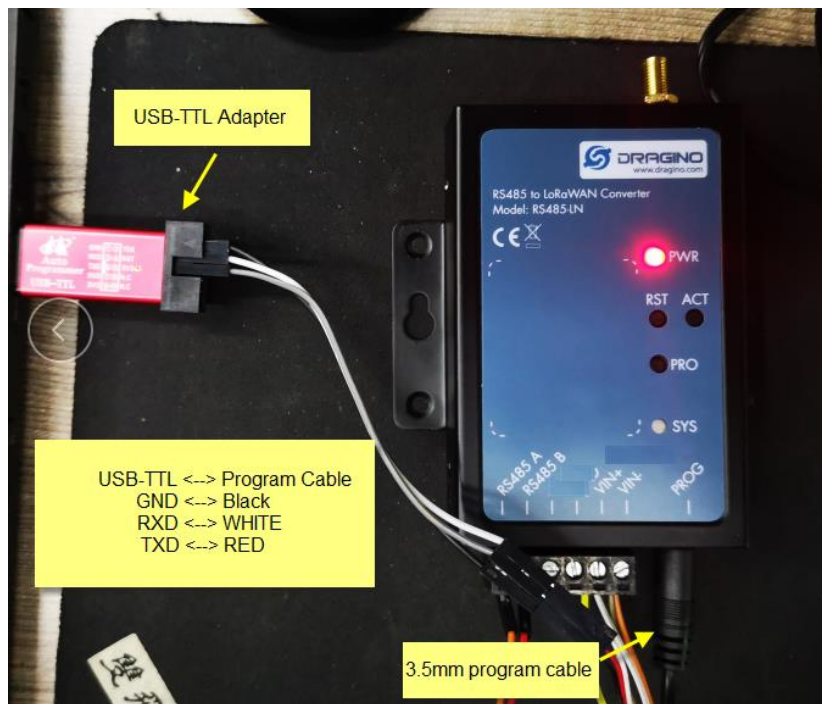

6. FAQ

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:

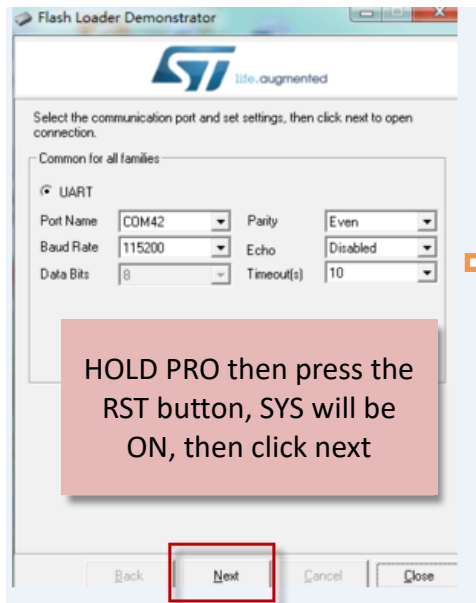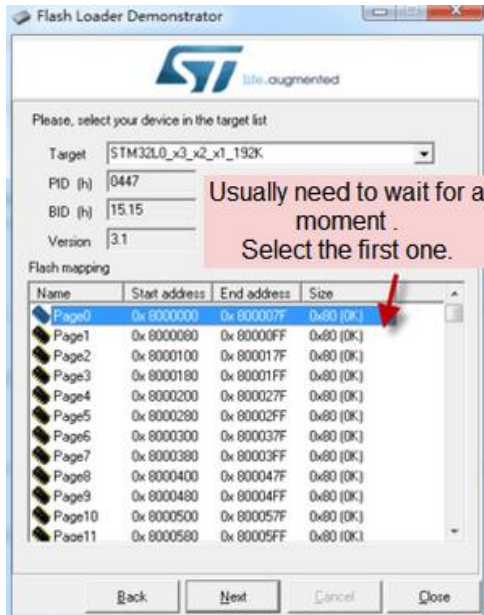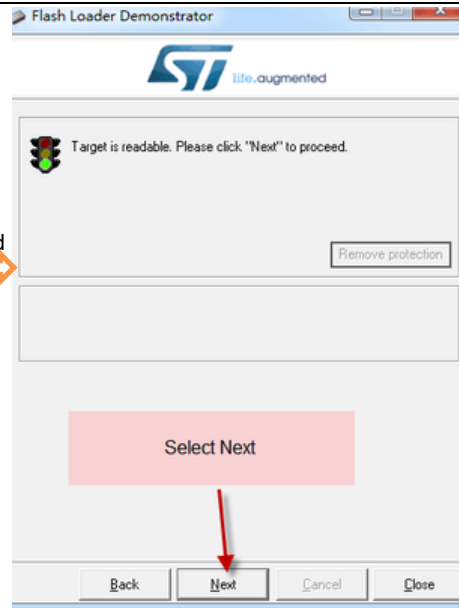**Step1:** Download flash loader.

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.

Board detected

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

Select Next

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

You can see the location of your programming

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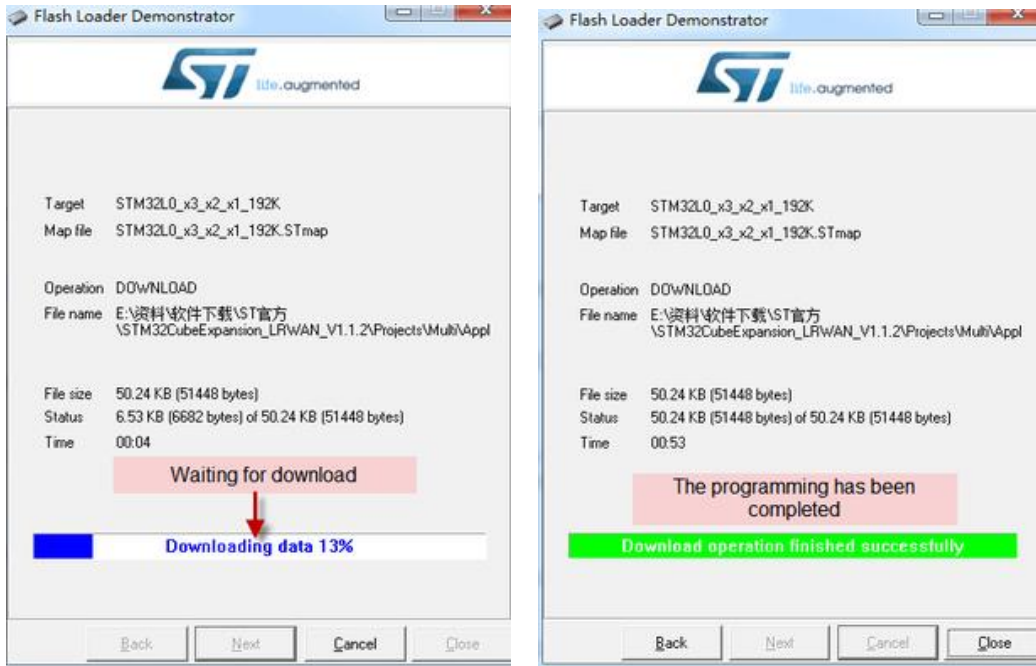
Notice: In case user has lost the program cable. User can hand made one from a 3.5mm cable. The pin mapping is:

to USB-TTL GND
to USB-TTL RXD
to USB-TTL TXD

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:

http://wiki.dragino.com/index.php?title=End_Device_AT_Commands_and_Downlink_Commands#Set_Packet_Receiving_Response_Level

7. Trouble Shooting

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

Please see this link for debug:

http://wiki.dragino.com/index.php?title=Main_Page#LoRaWAN_Communication_Debug

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

It might about the channels mapping. Please see for detail.

http://wiki.dragino.com/index.php?title=LoRaWAN_Communication_Debug#Notice_of_US915.2
FCN470.2FAU915_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
- ➢ **CN779**: frequency bands CN779
- ➢ **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. 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

<p align="center">support@dragino.com</p>