LSN50 LoRa Sensor Node User Manual

Document Version: 1.5.2
Image Version: v1.5

Version	Description	Date
1.0	Release	2018-Dec-4
1.1	Add steps of install STM320x; Add ST-Link Upload firmware method	2018-Dec-27
1.2	Add trouble shooting for UART upload, Add change log for firmware v1.4	2019-Jan-23
1.2.1	More detail description for 8 channel mode and trouble shooting for using in US915/AU915	2019-Feb-21
1.2.2	Modify trouble shooting for upload via Flashloader	2019-Mar-13
1.2.3	Add ISP Mode / Flash mode different/ Add working flow diagram (Chapter 2.1 how it works) Add FAQ for how to configure the Keys	2019-Apr-1
1.5.0	Upgrade to v1.5 version firmware Add ultrasonic sensor support and description. Add downlink description Change decoder for v1.5 Add working flow chart Add Cayenne support	2019-Apr-19
1.5.1	Improve Interrupt feature, change interrupt example to use door sensor	
1.5.2	Various minor text and format edits.	2019-Jun-10

1 Introduction

1.1 What is LSN50 LoRa Sensor Node

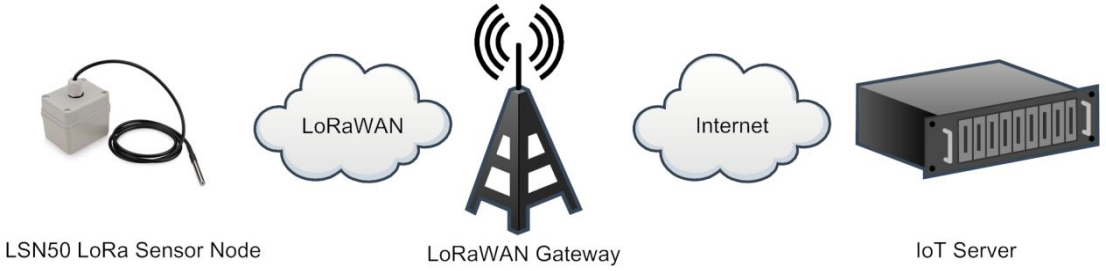
LSN50 is a Long Range LoRaWAN Sensor Node. It is designed for **outdoor data logging** and powered by **Li/SOCl2 battery** for long term use and secure data transmission. It is designed to facilitate developers to quickly deploy industrial level LoRa and IoT solutions. It helps users to turn the idea into a practical application and make the Internet of Things a reality. It is easy to program, create and connect your things everywhere.

It is based on SX1276/SX1278 allows the user to send data and reach extremely long ranges at low data-rates. 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.

LSN50 uses STM32l0x chip from ST, STML0x is the **ultra-low-power** STM32L072xx microcontrollers incorporate the connectivity power of the universal serial bus (USB 2.0 crystal-less) with the high-performance ARM® Cortex®-M0+ 32-bit RISC core operating at a 32 MHz frequency, a memory protection unit (MPU), high-speed embedded memories (192 Kbytes of Flash program memory, 6 Kbytes of data EEPROM and 20 Kbytes of RAM) plus an extensive range of enhanced I/Os and peripherals.

LSN50 is an **open source product**, it is based on the STM32Cube HAL drivers and lots of libraries can be found in ST site for rapid development.

LSN50 Network Structure



1.2 Specifications

Micro Controller:

- STM32L072CZT6 MCU
- MCU: STM32L072CZT6
- Flash: 192KB
- RAM: 20KB
- EEPROM: 6KB
- Clock Speed: 32Mhz

Common DC Characteristics:

- Supply Voltage: 2.1v ~ 3.6v
- Operating Temperature: -40 ~ 85°C
- I/O pins: Refer to STM32L072 datasheet

LoRa Spec:

- Frequency Range,
  - Band 1 (HF): 862 ~ 1020 Mhz

  or
  - 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.
- LoRaWAN 1.0.2 Specification

Battery:

- Li/SOCI2 un-chargeable battery
- Capacity: 4000mAh
- Self Discharge: <1% / Year @ 25°C
- Max continuously current: 130mA
- Max boost current: 2A, 1 second

Power Consumption

- STOP Mode: 2.7uA @ 3.3v
- LoRa Transmit Mode: 125mA @ 20dBm 44mA @ 14dBm

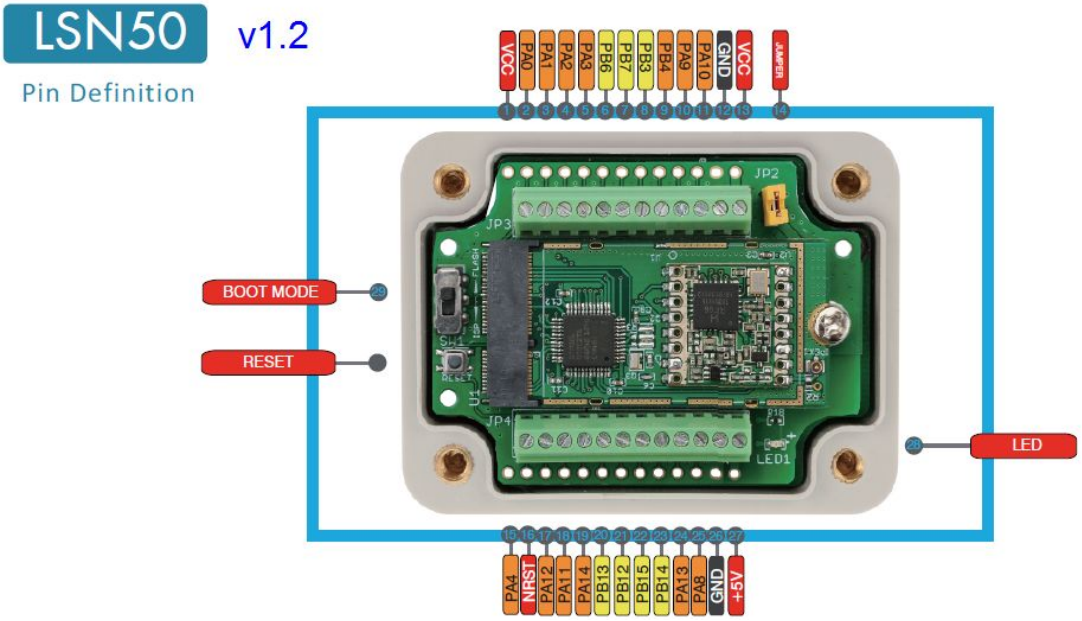
1.3 Features

- LoRaWAN 1.0.2 Class A,Class C
- STM32L072CZT6 MCU
- SX1276/78 Wireless Chip
- Pre-load bootloader on USART1/USART2
- MDK-ARM Version 5.24a IDE
- I2C, LPUSART1, USB, SPI2
- 3x12bit ADC, 1x12bit DAC
- 20xDigital I/Os
- LoRa™ Modem
- Preamble detection
- Baud rate configurable
- CN470/EU433/KR920/US915/IN865
- EU868/AS923/AU915
- Open source hardware / software
- Available Band:433/868/915/920 Mhz
- IP66 Waterproof Enclosure
- Ultra Low Power consumption
- AT Commands to change parameters
- 4000mAh Battery for long term use

1.4 Applications

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

1.5 Pin Definitions



Pin No.	Signal	Direction	Function	Remark
1	VCC(2.9V)	OUTPUT	VCC	Directly connect to main power for board
2	PA0	In/Out	Directly from STM32 chip	Used as ADC in LSN50 image
3	PA1	In/Out	Directly from STM32 chip	
4	PA2	In/Out	Directly from STM32 chip, 10k pull up to VCC	Used as UART_TXD in LSN50 image
5	PA3	In/Out	Directly from STM32 chip, 10k pull up to VCC	Used as UART_RXD in LSN50 image
6	PB6	In/Out	Directly from STM32 chip, 10k pull up to VCC	
7	PB7	In/Out	Directly from STM32 chip, 10k pull up to VCC	
8	PB3	In/Out	Directly from STM32 chip, 10k pull up to VCC	
9	PB4	In/Out	Directly from STM32 chip	
10	PA9	In/Out	Directly from STM32 chip, 10k pull up to VCC	
11	PA10	In/Out	Directly from STM32 chip, 10k pull up to VCC	
12	GND		Ground	
13	VCC(2.9V)	OUTPUT	VCC	Directly connect to main power for board
14	Jumper		Power on/off jumper	
15	PA4	In/Out	Directly from STM32 chip	
16	NRST	In	Reset MCU	

17	PA12	In/Out	Directly from STM32 chip	
18	PA11	In/Out	Directly from STM32 chip	
19	PA14	In/Out	Directly from STM32 chip	
20	PB13	In/Out	Directly from STM32 chip	
21	PB12	In/Out	Directly from STM32 chip	
22	PB15	In/Out	Directly from STM32 chip	
23	PB14	In/Out	Directly from STM32 chip	
24	PA13	In/Out	Directly from STM32 chip	
25	PA8	In/Out	Directly from STM32 chip	Default use to turn on/off LED1 in LSN50 image
26	GND		Ground	
27	+5V	Out	5v output power	Controlled by PB5(Low to Enable, High to Disable)
28	LED1		Controlled by PA8	Blink on transmit
29	BOOT MODE		Configure device in working mode or ISP program mode	Flash: Normal Working mode and send AT Commands ISP: UART Program Mode
30	NRST	In	Reset MCU	

1.6 Hardware Change log

LSN50 v1.2:

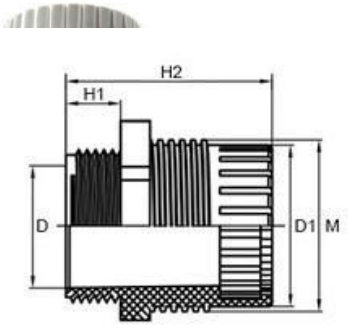
- Add LED. Turn on for every LoRa transmit
- Add pin PA4, PB13, NRST
- Add 5V Output, on/off control by PB5(Low to Enable, High to Disable)

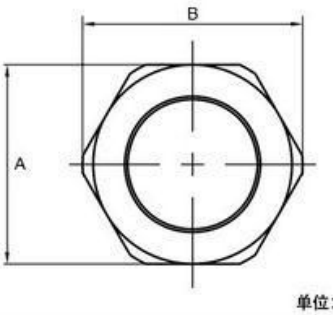
LSN50 v1.3:

- Add P-MOS to control 5V output

1.7 Hole Option

The LSN50 provides different hole size options for different size sensor cable. The options provided are M12, M16 and M20. The definition is as below:





单位：mm

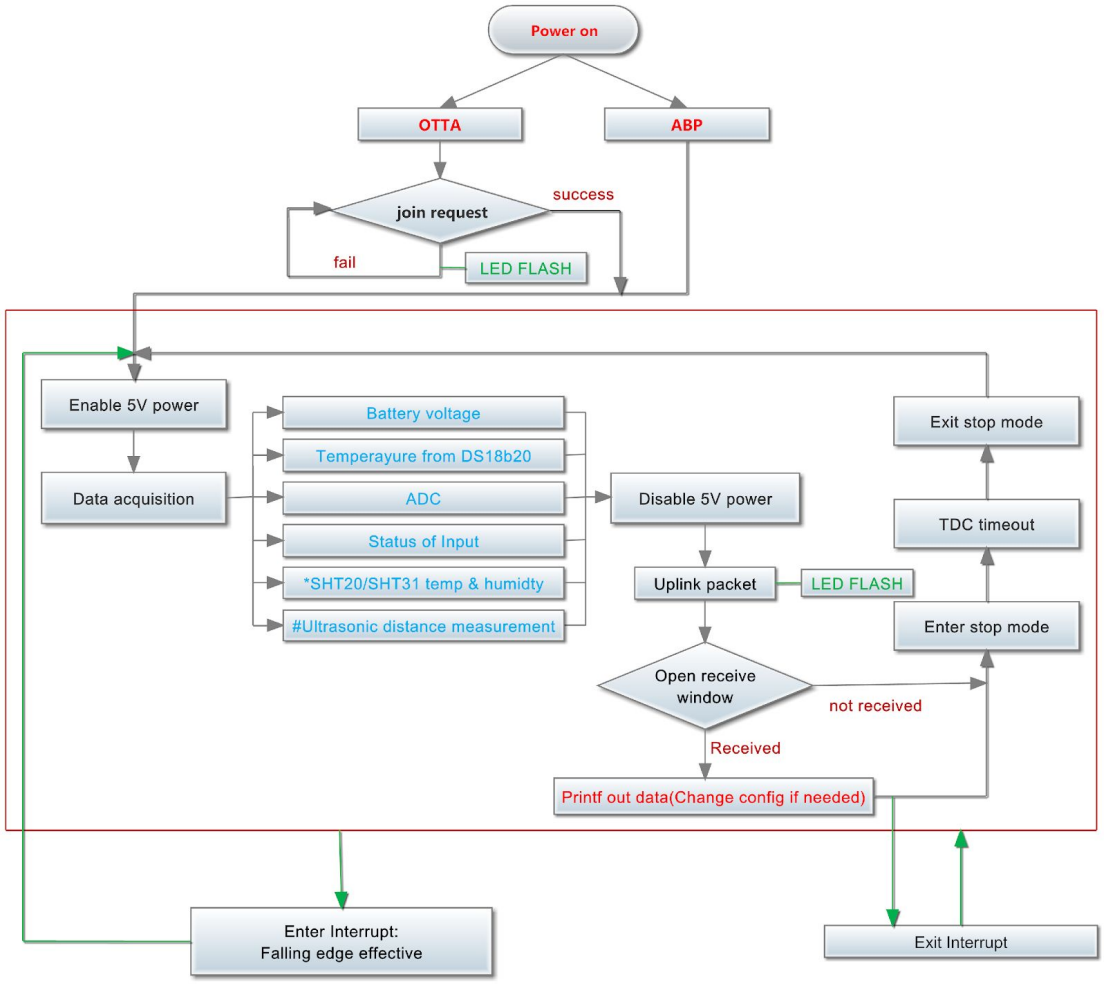
MODEL	H1	H2	M	M1	D	D1	A	B
M12*1.5	8	28.3	12.0	12.0	10.4	8.5±0.2	16±0.2	18±0.2
M16*1.5	8	30.7	15.1	16.0	13.5	10.9±0.2	18.8±0.2	20.6±0.2
M20*1.5	9	34.0	20.2	20.0	18.7	16.2±0.2	22.8±0.2	25.2±0.2

2 Using LSN50 with LoRaWAN firmware

2.1 How it works

The LSN50 is pre-loaded with a firmware and is configured as LoRaWAN OTAA Class A mode by default. It has OTAA keys to join LoRaWAN network. To connect a local LoRaWAN network, you just need to input the OTAA keys in the LoRaWAN IoT server and power on the LSN50. It will automatically join the network via OTAA.

The diagram below shows the working flow in default firmware (Ver 1.5):



* MOD=1 IIC mode (default)
\# MOD=2 ULT mode

In case you can not set the OTAA keys in the LoRaWAN OTAA server, and you have to use the keys from the server, you can use AT Commands to set the keys in the LSN50.

2.2 Quick guide to connect to LoRaWAN server (OTAA)

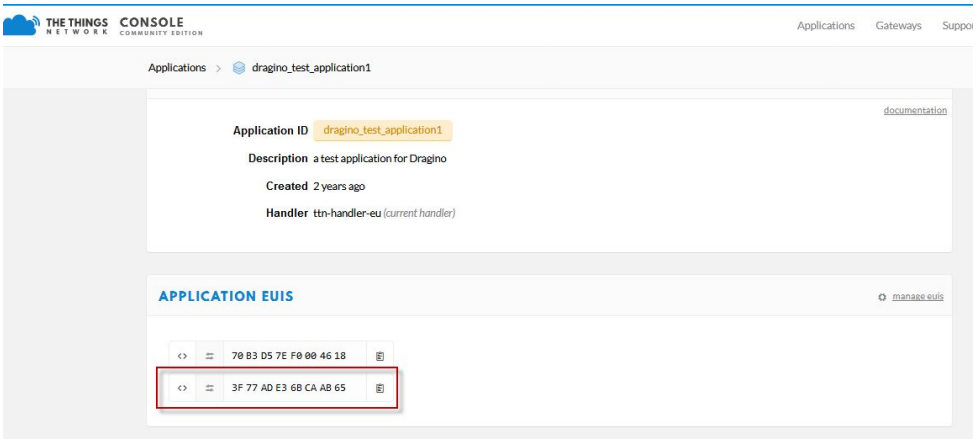
Following is an example for how to join the TTN LoRaWAN Network. Below is the network structure; we use the LG308 drvice as a LoRaWAN gateway in this example.

The LG308 is already set to connected to TTN network , so what we need to now is configure the TTN server.

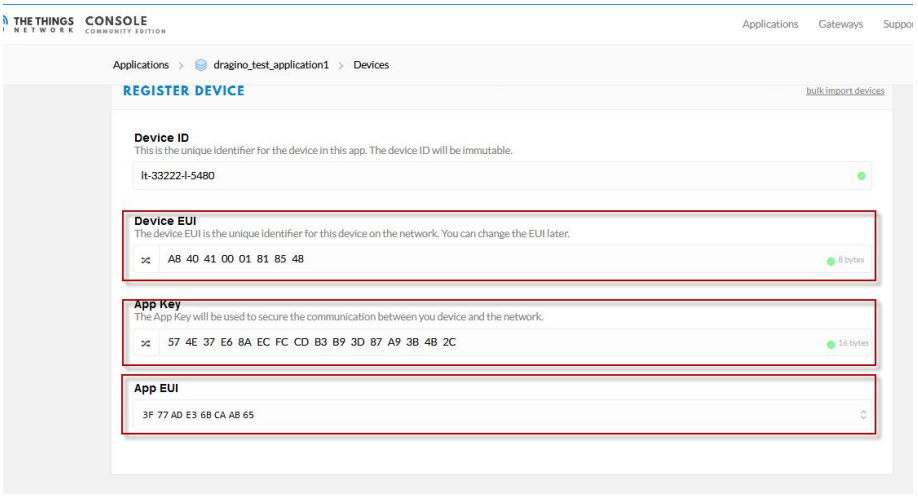
Step 1: Create a device in TTN with the OTAA keys from LSN50.
Each LSN50 is shipped with a sticker with the default device EUI as below:

You can enter this key in the LoRaWAN Server portal. Below is TTN screen shot:

Add APP EUI in the application

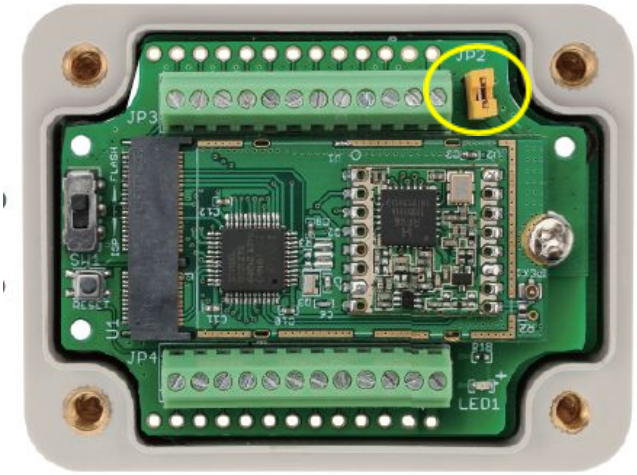


Add APP KEY and DEV EUI

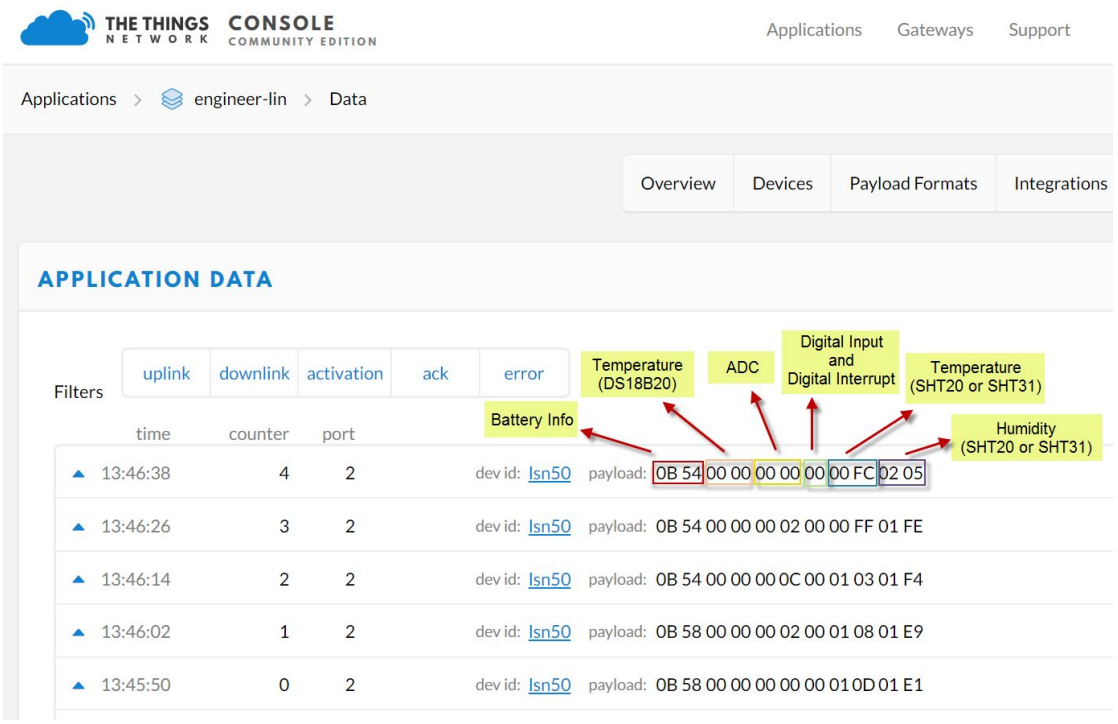


Step 2: Power on LSN50

Put a Jumper on JP2 to power on the device.

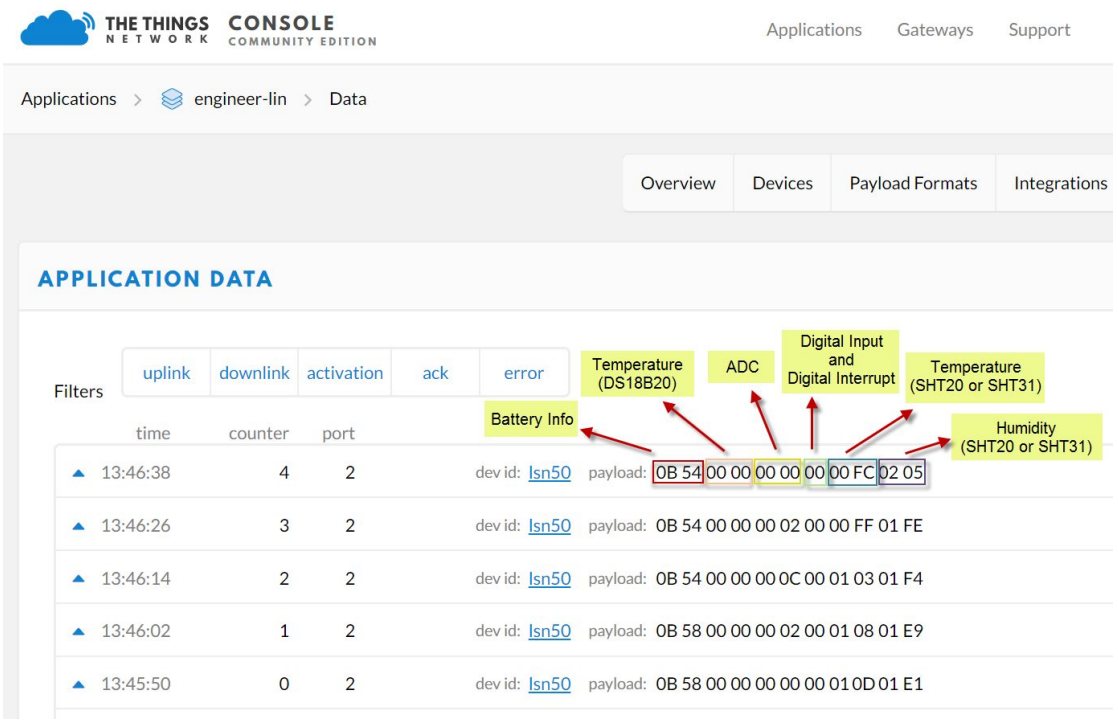


Step 3: The LSN50 will auto join to the TTN network. After join success, it will start to upload messages to TTN and you can see the messages in the panel.



2.3 Uplink Payload

The uplink payload is 11 bytes in total as shown below:



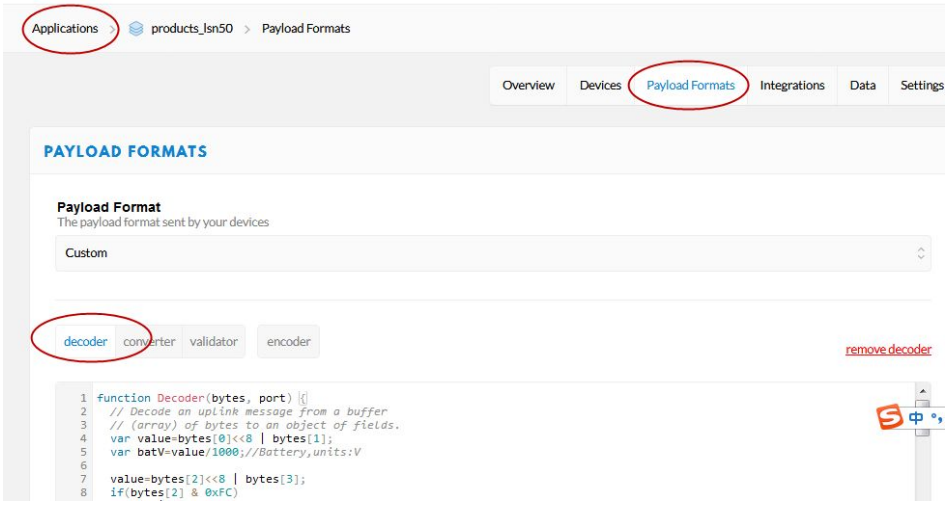
2.3.1 Payload Explanation

The uplink payload includes in total 11 bytes. Uplink packets use FPORT=2.
By default, the device will send an uplink message every 10 minutes.

Size(bytes)	2	2	1	2	2	2
Value	BAT	Temperature (DS18B20)	Digital in & Digital Interrupt	ADC	Temperature (SHT20 or SHT31 or Ultrasonic Sensor)	Humidity (SHT20)

2.3.2 Decode payload in The Things Network

While using TTN network, you can add the payload format to decode the payload.



The payload decoding function is as follows:

```
function Decoder(bytes, port) {
 // Decode an uplink message from a buffer
 // (array) of bytes to an object of fields.
 var value=bytes[0]<<8 | bytes[1];
 var batV=value/1000;//Battery,units:V
 value=bytes[2]<<8 | bytes[3];
 if(bytes[2] & 0x80)
 {value |= 0xFFFF0000;}

 var tempc=(value/10).toFixed(2);//DS18B20,PB3,units:℃
 var adc_ch0=(bytes[4]<<8 | bytes[5])/1000;//PA0,ADC Channel 0,units:V

 if(bytes[6] & 0x08)
 {
   var mod3="3ADC";  //work mode
 }

 else if(bytes[6] & 0x04)
 {
   var mod2="ULT";  //work mode
 }

 else if(!(bytes[6] & 0x04))
 {
   var mod1="IIC";  //work mode
 }

 var digital_IS=(bytes[6] & 0x02)? "H":"L";//PA12,   Digital Input Status

 var exti_trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";//PB14,GPIO_MODE_IT_FALLING

 var switch_=(bytes[6] & 0x80)? "CLOSE":"OPEN";//PB14,GPIO_MODE_IT_FALLING

if(mod3=="3ADC")
{
  var adc_ch1=(bytes[7]<<8 | bytes[8])/1000;//PA1,ADC Channel 1,units:V
  var adc_ch4=(bytes[9]<<8 | bytes[10])/1000;//PA4,ADC Channel 4,units:V
}
```

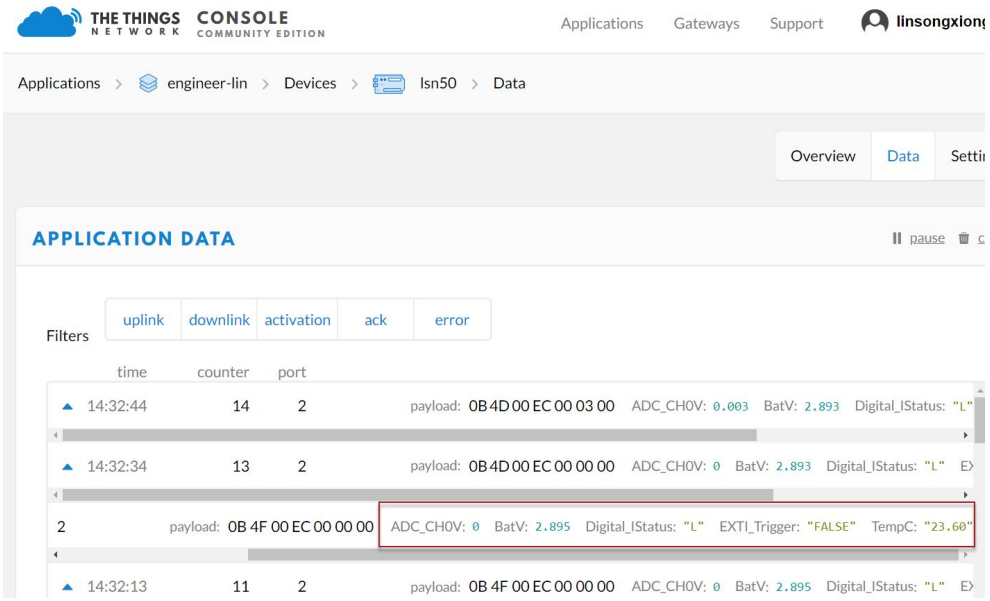
```
else if(mod2=="ULT")
{
 value=bytes[7]<<8 | bytes[8];
 var ult=(value/10); //ultrasonic,units:cm
}

else if(mod1=="IIC")
{
 value=bytes[7]<<8 | bytes[8];
 if(bytes[7] & 0x80)
 {value |= 0xFFFF0000;}
 var temp_SHT=(value/10).toFixed(2);//SHT20,temperature,units:℃

 value=bytes[9]<<8 | bytes[10];
 var hum_SHT=(value/10).toFixed(1);//SHT20,Humidity,units:%
}

 return {
 BatV:batV,
 TempC:tempc,
 ADC_CH0V:adc_ch0,
 ADC_CH1V:adc_ch1,
 ADC_CH4V:adc_ch4,
 Digital_IStatus:digital_IS,
 EXTI_Trigger:exti_trigger,
 Door_status:switch_,
 MOD1:mod1,
 MOD2:mod2,
 MOD3:mod3,
 Ultrasonic:ult,
 TempC_SHT:temp_SHT,
 Hum_SHT:hum_SHT
 };
}
```

The uplink payload will show as below:



2.4 Payload Explanation and Sensor Interface

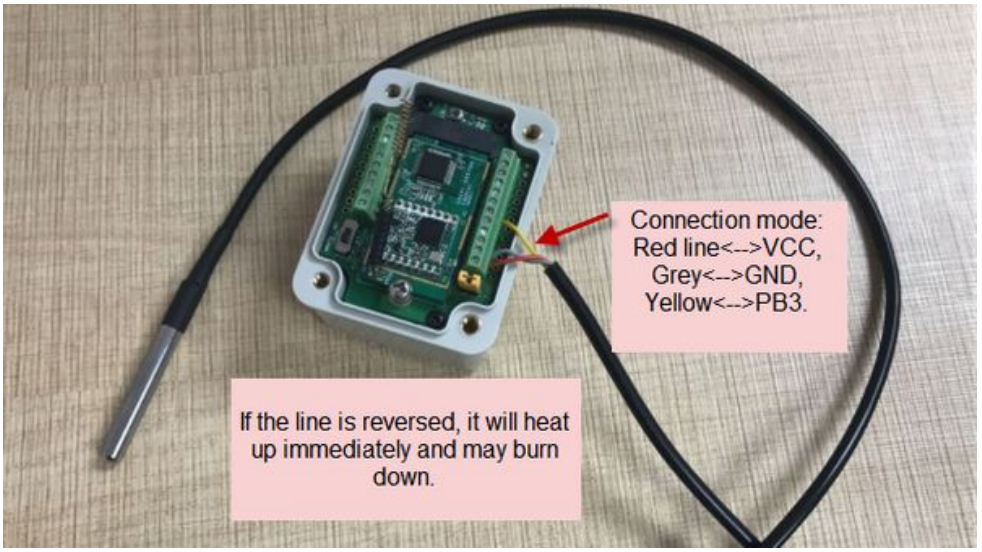
2.4.1 Battery Info

Check the battery voltage for LSN50.
Ex1: 0x0B45 = 2885mV
Ex2: 0x0B49 = 2889mV

2.4.2 Temperature (DS18B20)

If there is a DS18B20 connected to PB3 pin. The temperature will be uploaded in the payload.

Connection



Example:
If payload is: 0105H: (0105 & FC00 == 0), temp = 0105H /10 = 26.1 degree
If payload is: FF3FH : (FF3F & FC00 == 1) , temp = (FF3FH - 65536)/10 = -19.3 degrees.

2.4.3 Digital Input

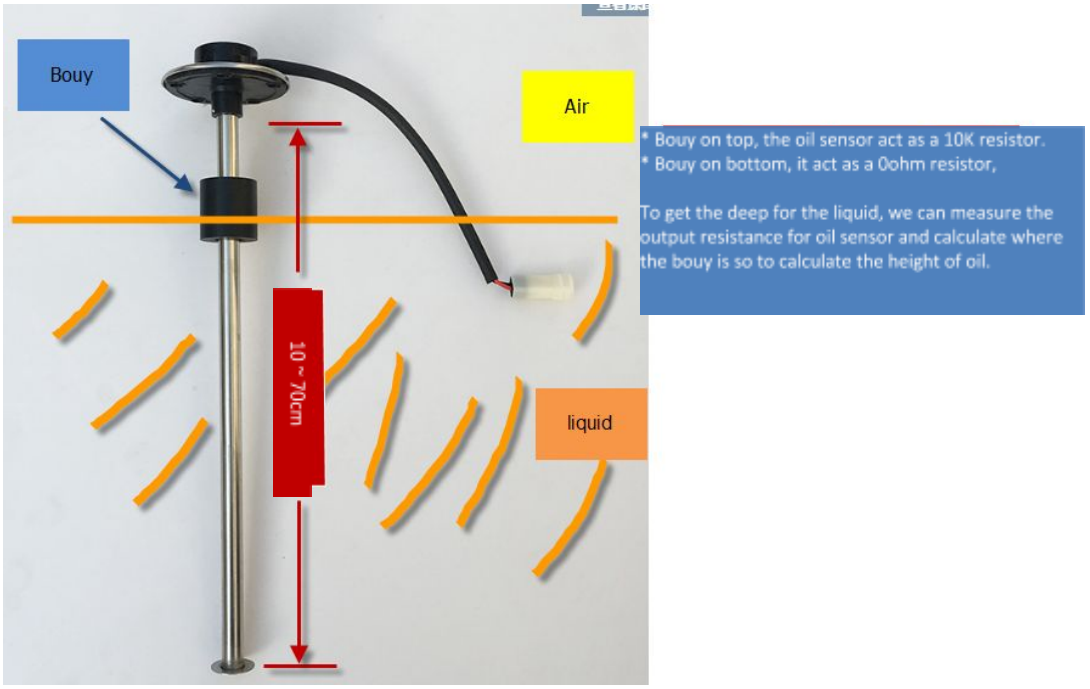
The digital input for pin PA12,

- When PA12 is high, the bit2 of payload byte 6 is 1.
- When PA12 is low, the bit2 of payload byte 6 is 0.

2.4.4 Analogue Digital Converter (ADC)

The ADC monitors the voltage on the PA0 line, in mV.
Ex: 0x021F = 543mv,

Example1: Reading an Oil Sensor (Read a resistance value):

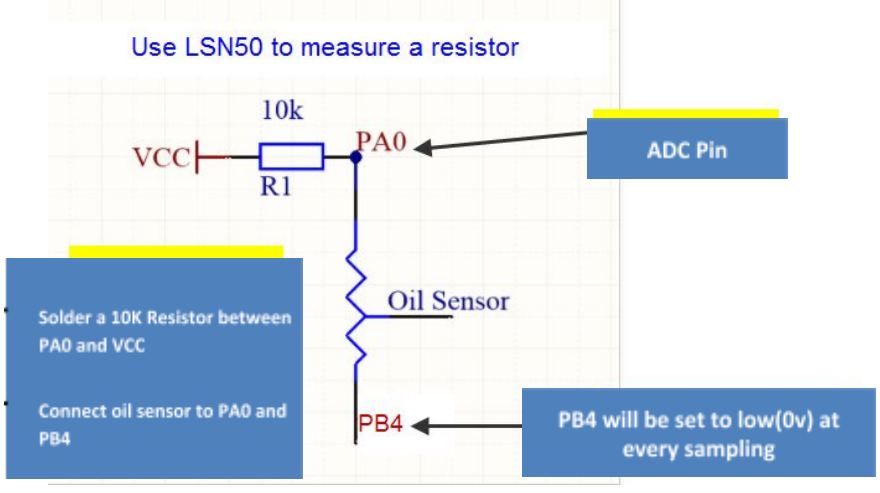


In the LSN50, we can use PB4 and PA0 pin to calculate the resistance for the oil sensor.

Steps:

1. Solder a 10K resistor between PA0 and VCC.
2. Screw oil sensor's two pins to PA0 and PB4.

The equipment circuit is as below:



According to above diagram:

$$(VCC - V_{PA0}) \frac{}{10} k = \frac{V_{PA0}}{R_{oil_{sensor}}}$$

So

$R_{oil_{sensor}} = V_{PA0} \times 10 \frac{K}{(VCC - V_{PA0})}$

V_{PA0} is the reading of ADC. So if ADC=0x05DC=0.9 v and VCC (BAT) is 2.9v

The $R_{oil_{sensor}} = 0.9 \times \frac{10K}{2.9 - 0.9} = 4.5K$ ohm

Since the Bouy is linear resistance from 10 ~ 70cm.

The position of Bouy is $\frac{4.5K}{10K} \times (70cm - 10cm) + 10cm = 37cm$, from the bottom of Bouy

2.4.5 Digital Interrupt

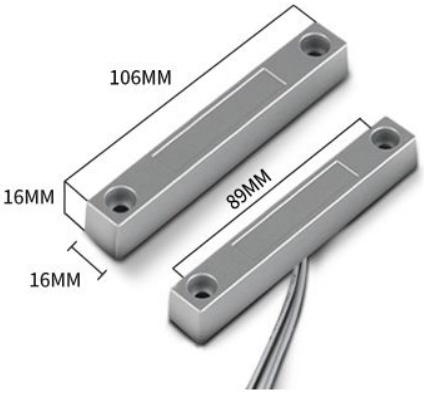
Digital Interrupt refers to pin PB14, and there are different trigger methods. When there is a trigger, the LSN50 will send a packet to the server.

Example to use with door sensor

(Requires firmware > 1.5.1)

The door sensor as shown at right. It is a two wire magnetic contact switch used for detecting the open/close status of doors or windows.

When the two pieces are close to each other, the 2 wire output will be short or open (depending on the type), while if the two pieces are away from each other, the 2 wire output will be the opposite status. So we can use LSN50 interrupt interface to detect the status for the door or window.

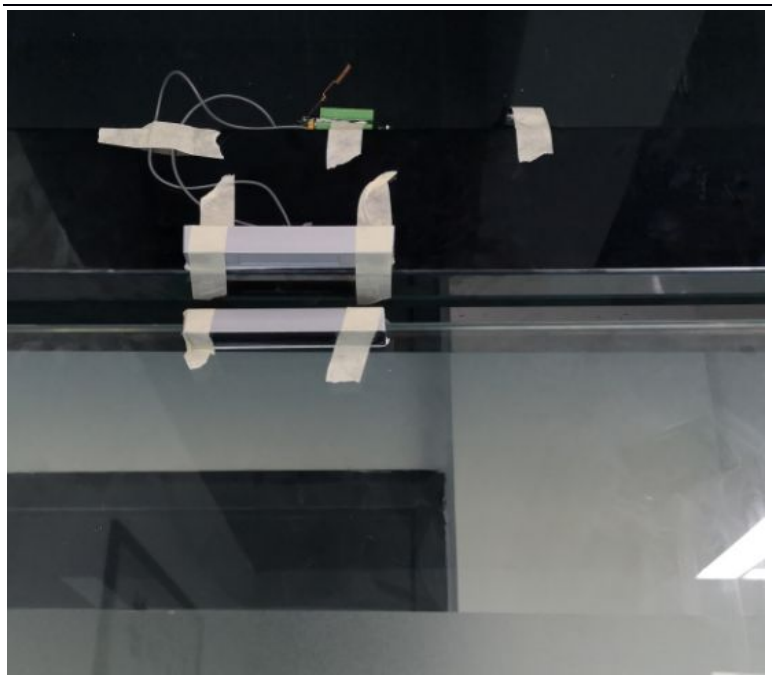


Below is the installation example:

Fix one piece of the magnetic sensor to the door and connect the two pins to LSN50 as follows:

- One pin to LSN50's PB14 pin
- The other pin to LSN50's VCC pin

Install the other piece to the door. Find a place where the two pieces will be close to each other when the door is closed. For this particular magnetic sensor, when the door is closed, the output will be short, and PB14 will be at the VCC voltage.



The above photos shows the two parts of the magnetic switch fitted to a door.

The software by default uses the falling edge on the signal line as an interrupt. We need to modify it to accept both the rising edge (0v --> VCC , door close) and the falling edge (VCC --> 0v , door open) as the interrupt.

The command is:

AT+INMOD=1 //(more info about INMOD please refer [AT Command Manual]().)

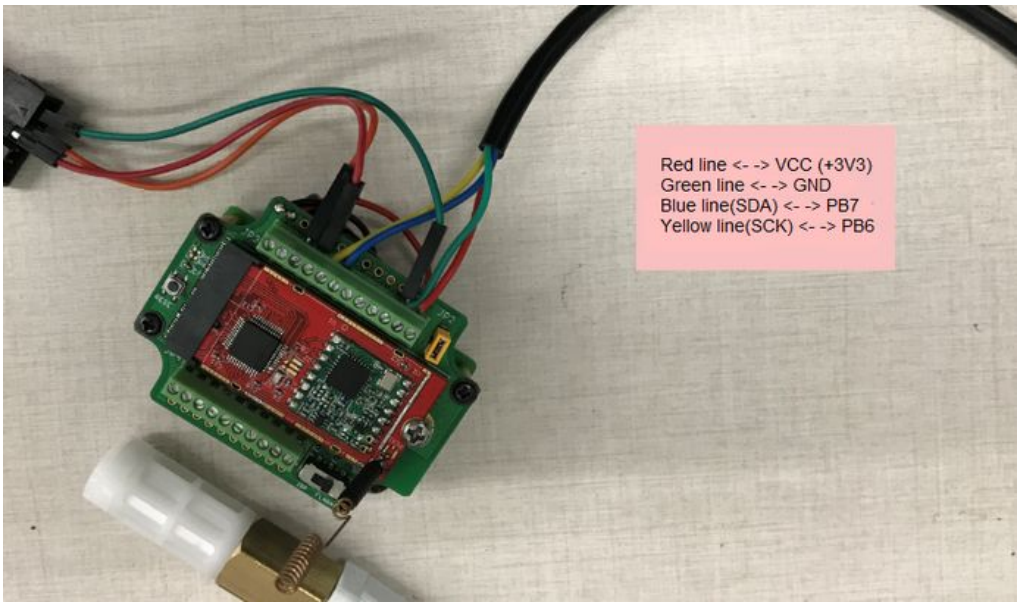
Below shows some screen captures in TTN:

2.4.6 I2C Interface (SHT20)

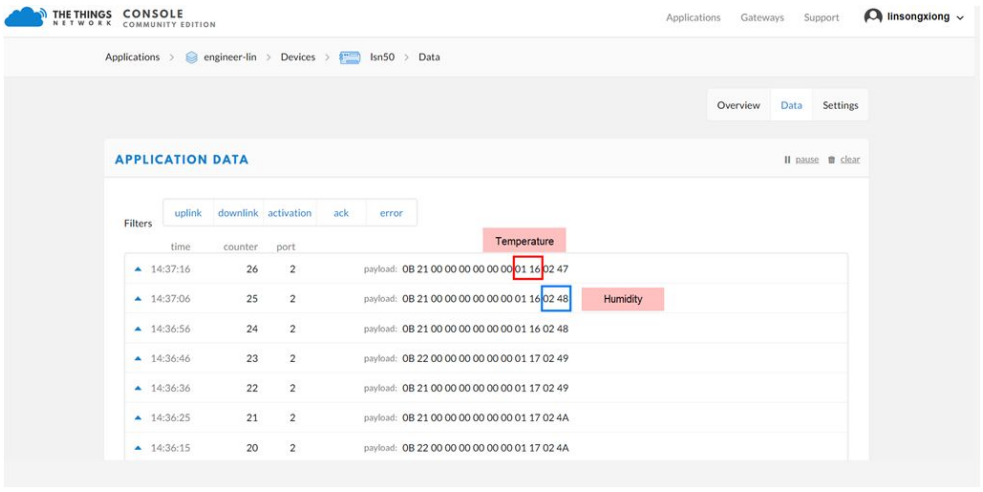
The PB6(SDA) and PB7(SCK) are I2C interface lines. You can use these to connect to an I2C device and get the sensor data.

We have made an example to show how to use the I2C interface to connect to the SHT20 Temperature and Humidity Sensor. This is supported in the stock firmware since v1.5 with **AT+MOD=1 (default value).**

Below is the connection to SHT20.



The device will be able to get the I2C sensor data now and upload to IoT Server.



Convert the read byte to decimal and divide it by ten.

Example:

Temperature: Read:0116(H) = 278(D) Value: 278 /10=27.8℃;

Humidity: Read:0248(H)=584(D) Value: 584 / 10=58.4, So 58.4%

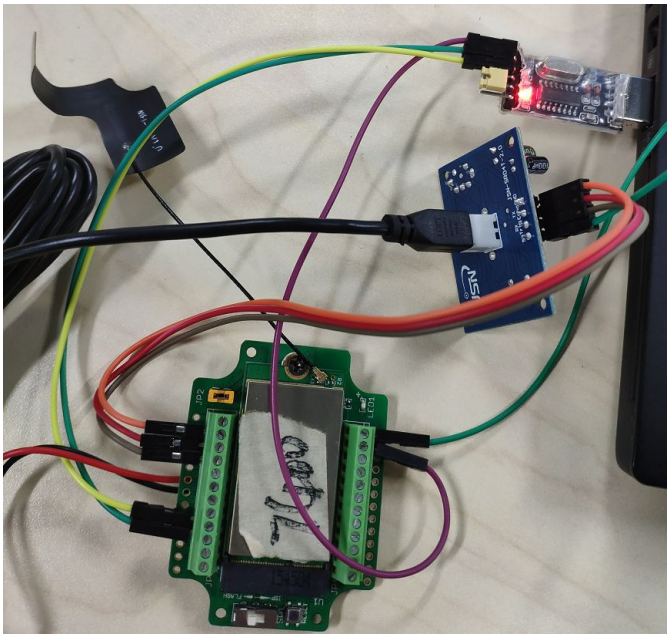
If you want to use other I2C device, please refer the SHT20 part source code as reference.

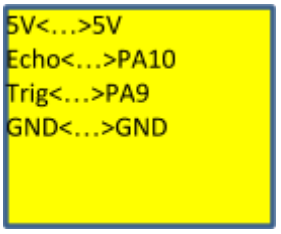
2.4.7 Ultrasonic Sensor

The LSN50 v1.5 firmware supports ultrasonic sensor (with AT+MOD=2) such as SEN0208 from DF-Robot. This Fundamental Principles of this sensor can be found at this link: https://wiki.dfrobot.com/Weather_-_proof_Ultrasonic_Sensor_with_Separate_Probe_SKU__SEN0208

The LSN50 detects the pulse width of the sensor and converts it to mm output. The accuracy will be within 1 centimeter. The usable range (the distance between the ultrasonic probe and the measured object) is between 24cm and 600cm.

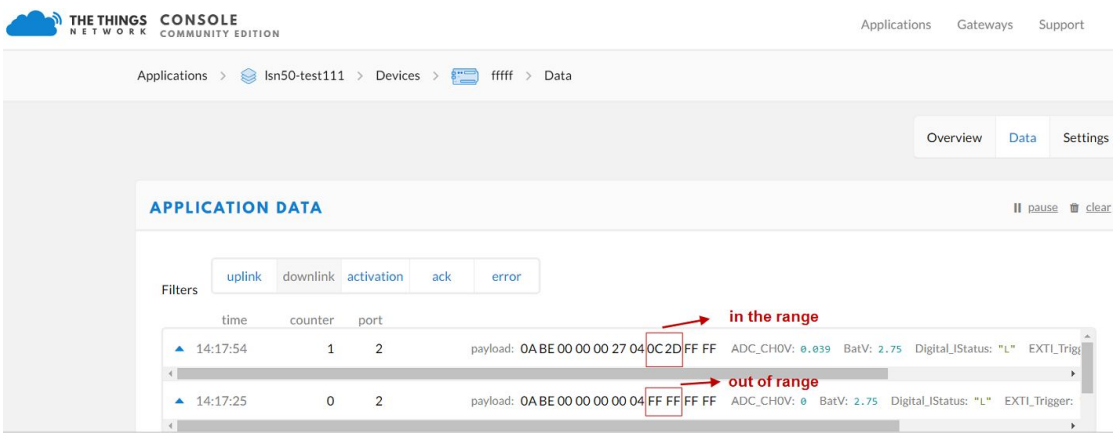
The picture below shows the connection:





Connect to the LSN50 and run **AT+MOD=2** to switch to ultrasonic mode (ULT).
The ultrasonic sensor uses the 8th and 9th byte for the measurement value.

Example:
Distance: Read:0C2D(Hex) = 3117(D) Value: 3117 mm=311.7 cm



You can see the serial output in ULT mode as below:

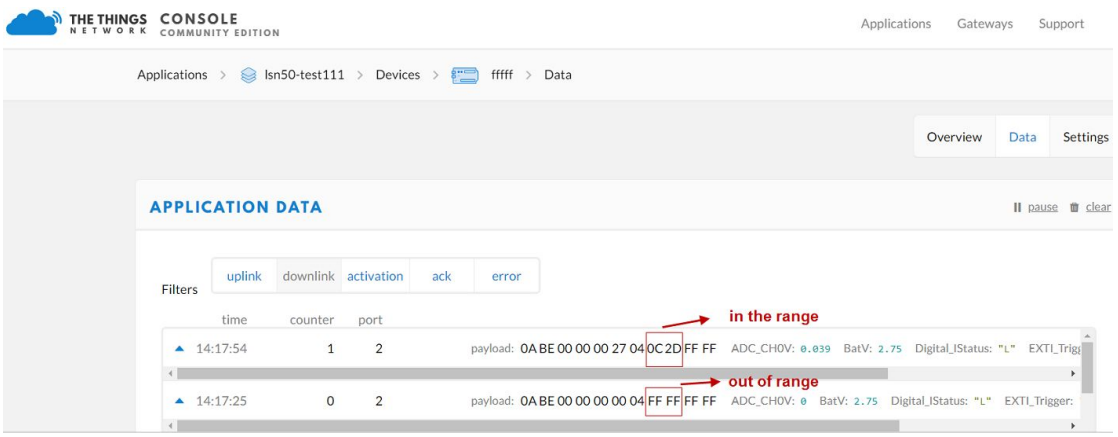
```
LSN50 Device
Image Version: v1.4
Frequency Band: EU868
DevEui= 00 97 16 CF B8 37 CF 4E

***** UpLinkCounter= 0 *****
TX on freq 868300000 Hz at DR 5
txDone
rxDone
JOINED
Distance is out of range

***** UpLinkCounter= 0 *****
TX on freq 867700000 Hz at DR 0
txDone
rxTimeOut
rxTimeOut
Distance=3117 mm

***** UpLinkCounter= 1 *****
TX on freq 868100000 Hz at DR 0
txDone
rxTimeOut
rxDone
```

In TTN server:



2.4.8 +5V Output

Since v1.2 hardware version, a +5v output is added in the hardware. The +5V output will be valid for every sampling.

2.5 Downlink Payload

By default, LSN50 prints the downlink payload to console port.

Downlink Control Type	FPort	Type Code	Downlink payload size(bytes)
TDC (Transmit Time Interval)	Any	01	4
RESET	Any	04	2
AT+CFM	Any	05	4
INITMOD	Any	06	4

Examples

Set TDC
If the payload=0100003C, it means set the END Node's TDC to 0x00003C=60(S), while type code is 01.
Payload: 01 00 00 1E TDC=30S
Payload: 01 00 00 3C TDC=60S

Reset
If payload = 0x04FF, it will reset the LSN50

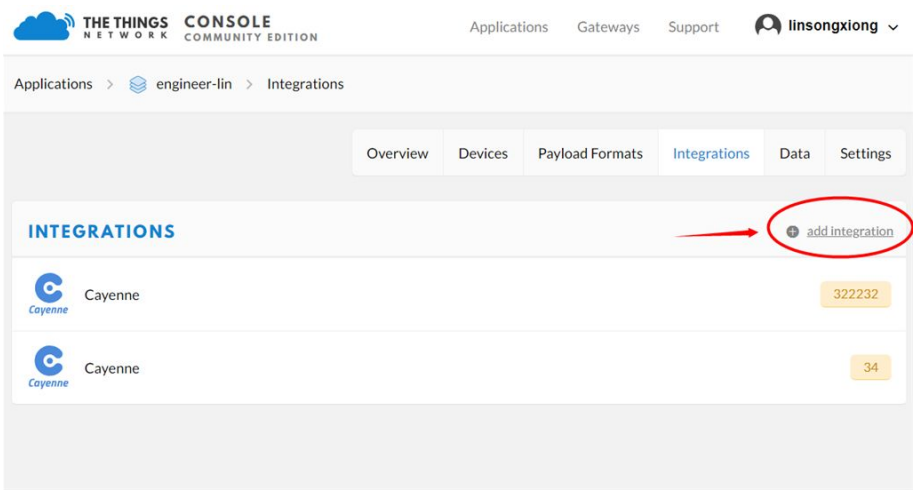
CFM
Downlink Payload: 05000001, Set AT+CFM=1

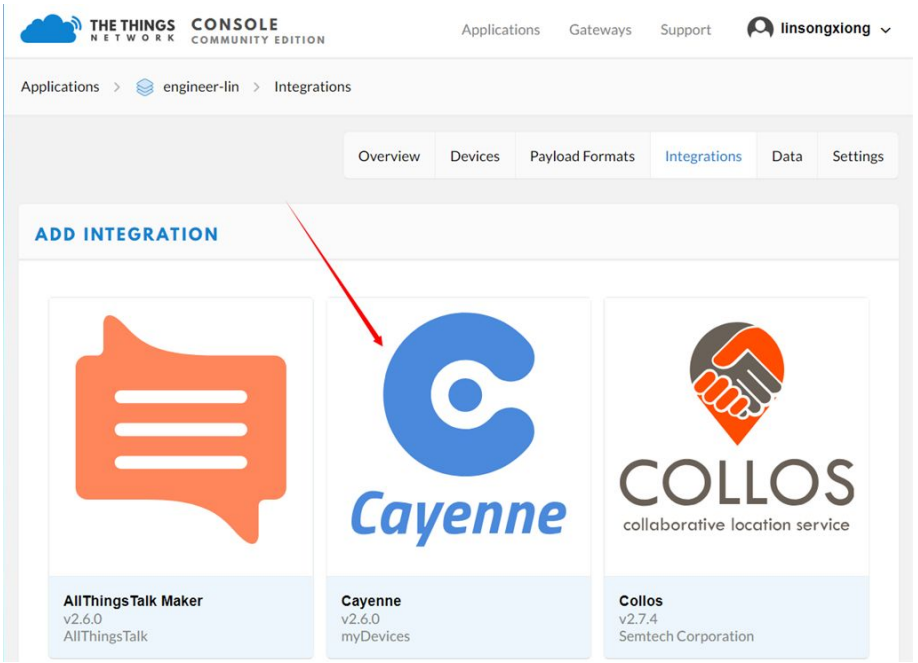
INITMOD
Downlink Payload: 06000003, Set AT+INITMOD=3

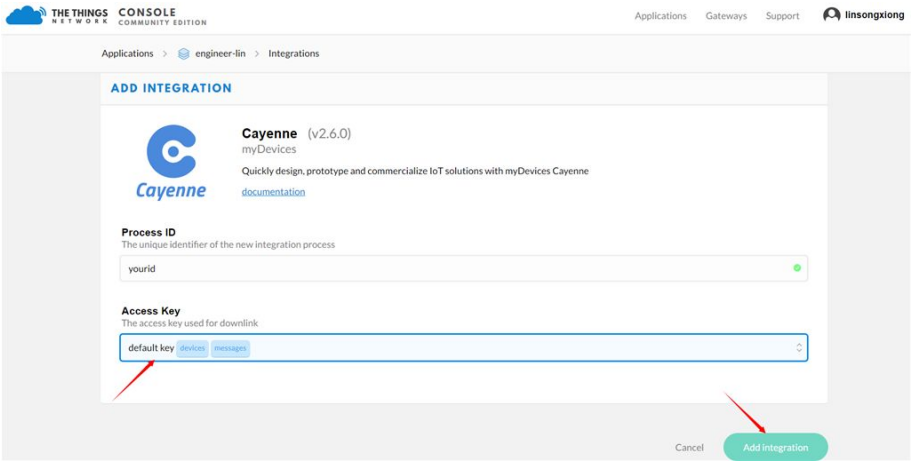
2.6 Show Data in Cayenne IoT Server

Cayenne provides a human friendly interface to show the sensor data, once we have data in TTN, we can use Cayenne to connect to TTN and see the data in Cayenne. Below are the steps:

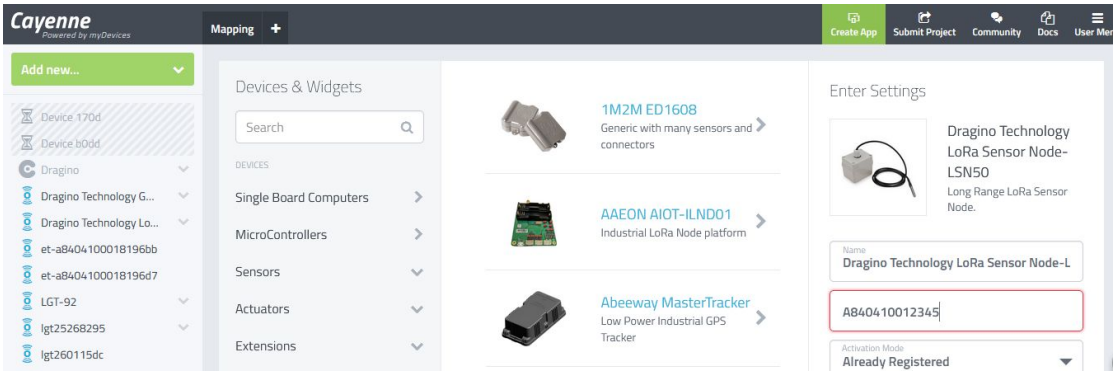
Step 1: Be sure that your device is programmed and properly connected to the network at this time.
Step 2: To configure your Application to forward data to Cayenne you will need to add an Integration. To add the Cayenne integration, perform the following steps:





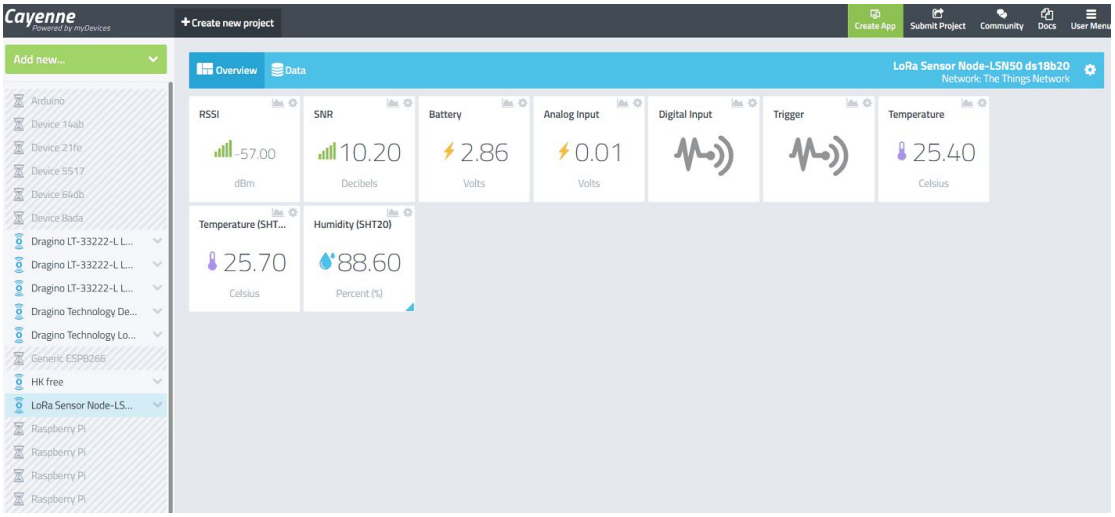


Step 3: Create an account or log in in the mydevices cayenne.
Step 4:Search the LSN50 and add DevEUI

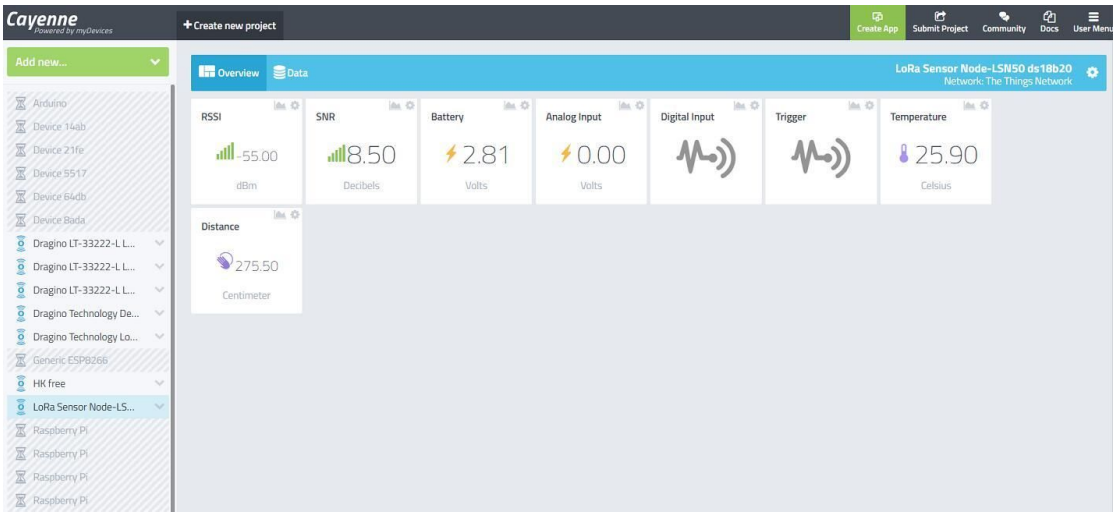


After added, when you have data arrive TTN, it will also arrive and show in Cayenne.

Example for AT+MOD=1 plus SHT20 + DS18B20 sensor:



Example for AT+MOD=2 plus Ultrasonic + DS18B20 sensor:



2.7 Firmware Change Log

V1.6 Firmware (Not released):

- Improve Interrupt feature.
- Downlink to change AT+CFM. Downlink to change AT+INITMOD

V1.5 Firmware:

- Add ultrasonic sensor support.
- Add AT+MOD command to select difference sensors: (Ultrasonic, I2C) (See update AT Command manual)
- Add Downlink command to change TDC and reset the device .
- Add AT+TXP command to be able manually set the exact TX Gain (See update AT Command manual)

V1.4 Firmware:

- Adjust payload, the default firmware include SHT20 and SHT31, If there is no SHT20, SHT31, the related filed will show FF FF FF FF
- Adjust 868 & 915 payload into 11 bytes, now 868 & 915 has same payload
- Fix the 85 degree bug for DS18B20
- Add new AT command which can adjust RX window time for LG01/LG02
- Add AT command to print all parameters.
- Any FPORT can accept downlink message and print.

v1.3 Firmware:

- Add new AT Commands: AT+CHS & AT+CHE
- Change AT+FDR command. This command will reset to factory except the keys
- +5v power will only enable when read sensor data
- Optimize OTAA join procedure. The first 50 joins will act as per LoRaWAN request(request join every few seconds), if devices have not joined in network, the Join Interval will extend to 30 minutes. If devices still not join at 200 tries, it will restart and start to Join again.
- Now print Device Model/Frequency bands/ Image Version/Dev EUI at start.

V1.2 Firmware:

- Support Class C
- After the configuration key can be stored in. No need to configure again even after power off.
- Add auto send feature after power on
- Solve negative temperature issue.
- Support Cayenne_LPP payload, user need to recompile firmware again.

V1.1 Firmware:

- Support Battery Voltage(mV) ,the data of Oil Sensor ,the data of DS18B20, Digital I/0, ADC_IN1(PA1),
- Proximity switch, I2C Device Example

V1.0 Firmware:

- Support ADC monitoring (See how to in the case study of Oil Sensor) and DS18B20 (See how to in the case study of DS18B20)

2.8 Battery Analysis

2.8.1 Battery Type

The LSN50 battery is a combination of a 4000mAh Li/SOCI2 Battery and a Super Capacitor. The battery is non-rechargeable battery type with a low discharge rate (<2% per year). This type of battery is commonly used in IoT devices such as water meter.

The battery is designed to last for more than 5 years for the LSN50.

The battery related documents as below:

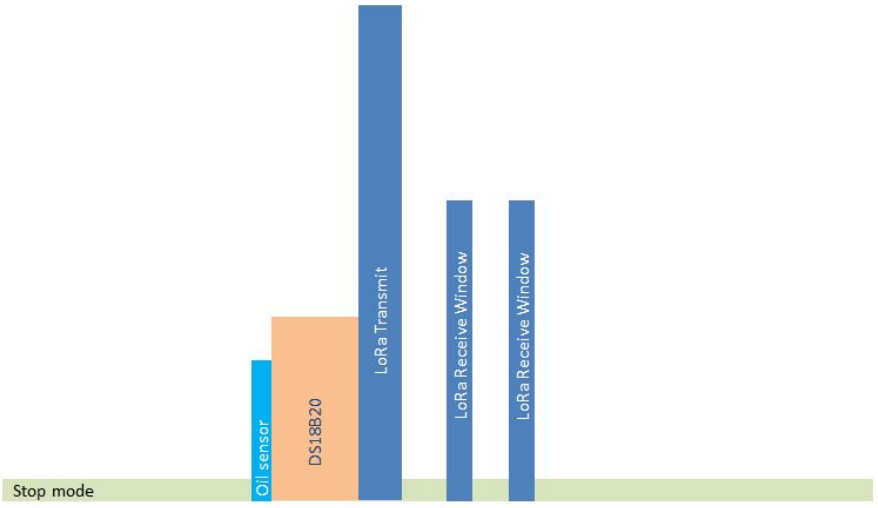
- Battery Dimension,
- Lithium-Thionyl Chloride Battery datasheet, Tech Spec
- Lithium-ion Battery-Capacitor datasheet, Tech Spec

2.8.2 Power consumption Analyze

In a minimum system with DS18B20 and Oil Sensor and default firmware, the power consumption includes:

1. Deep Sleep (Stop mode) for STM32. ~ 5uA
2. Sampling current while reading DS18B20 and Oil Sensor
   - Oil Sensor sampling time: 200us, current: 0.3mA
   - DS18B20 sampling time: 750ms, current: 0.64mA
   - Above power should add 8mA CPU power in working mode.
3. LoRaWAN transmit and receive time consumption. The LoRa TX / RX time and power can be found in the LoRa calculator tool.

In a typical LoRaWAN data transmit. The energy profile is as below:



LoRaWAN Energy Profile in one period

In the LoRaWAN protocol, the device will transfer in different LoRa Radio, and have different energy profile in LoRa part. We can calculate the battery life in two cases:

1) Lower power LoRa radio. Device has a good signal to gateway
2) Higher power LoRa radio. Device has a poor signal to gateway

Lower Power Case:

- Radio Parameter: SF7, 125kHz, 20dbm
- Transmit interval: 15 minutes.
- Payload: 8 Bytes.

High Power Case:

- Radio Parameter: SF10, 125kHz, 20dbm
- Transmit interval: 15 minutes.
- Payload: 8 Bytes.

To simplify the calculation, we can:

- Combine oil sensor and DS18B20 sampling energy together to **751ms@8.64ma**
- Combine the two RX windows together.

There is a power consumption tool for easy analysis. Below is the analysis result.

Scenarios		A	B	C	D	E	F
Time	Units	Scenario_A	Scenario_B	Scenario_C	Scenario_D	Scenario_E	Scenario_F
Sleep	min	15	15	15			
Sampling	ms	751	751	5000			
Transmit	ms	100	274.4	34.3			
Receive	ms	72	491.4	82			
Radio type		SF7_125K_20dB	SF10_125K_20dB	SF7_125K_14dB			
# of bytes transmitted		8	8	8			
Total System Current							
Sleep	mA	0.005	0.005	0.005			
Sampling	mA	0.64	0.64	0.64			
Transmit	mA	133	133	52			
Receive	mA	18.8	18.8	18.8			
					Micro-Controller Active power (mA):		8
Power usage comparison							
Sleep	%	22.92%	8.87%	40.82%	0.00%	0.00%	0.00%
Sampling	%	2.45%	0.95%	29.02%	0.00%	0.00%	0.00%
Transmit	%	67.74%	71.96%	16.18%	0.00%	0.00%	0.00%
Receive	%	6.89%	18.22%	13.98%	0.00%	0.00%	0.00%
						Legend: Red > 100%, Green <= 100%	
Average current	mA	**0.021793472**	**0.056254259**	**0.012180976**	**0**	**0**	**0**

Design Goals							
System efficiency		90%	90%	90%	90%	90%	90%
Target battery life	yr	2	2	2	2	2	2
Required battery capac	mAh	**424.54**	**1095.83**	**237.29**	**0.00**	**0.00**	**0.00**
or							
Given battery capacity	mAh	4000	4000	4000	4000	4000	4000
Estimated battery life	yr	**18.84**	**7.30**	**33.71**	**0.00**	**0.00**	**0.00**

Note: Ignore the 18 year result, because the battery has a max 2% discharge per year.

2.8.3 Battery Note

The Li-SICO battery is designed for small current / long period application. It is not good to use a high current, short period transmit method. The recommended minimum period for use of this battery is 5 minutes. If you uses a shorter period time to transmit LoRa, then the battery life may be decreased.

2.8.4 Replace the battery

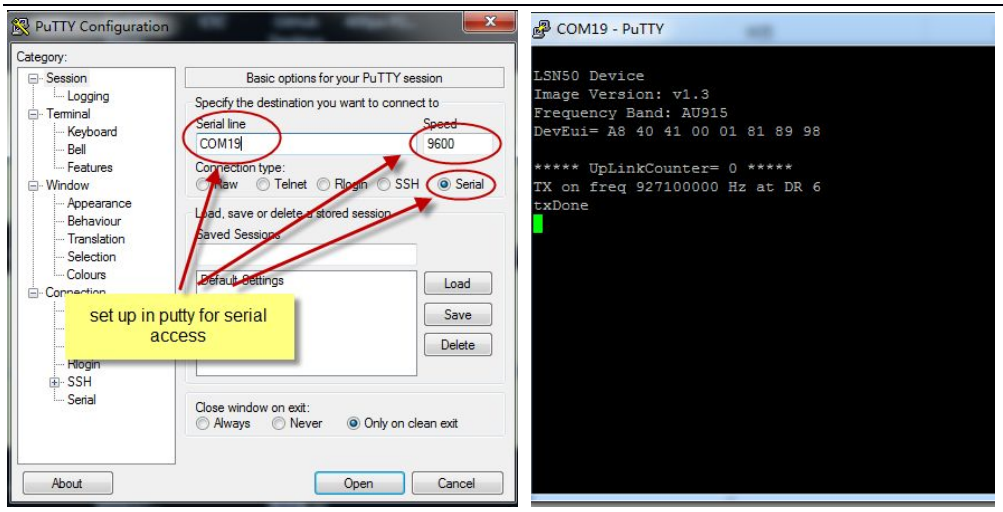
You can change the battery in the LSN50. On the main board, there is a diode (D1) between the battery and the main circuit. If you need to use a battery with less than 3.3v, please remove the D1 and shortcut the two pads of it so there won't be voltage drop between battery and main board.

3 Using the AT Commands

3.1 Access AT Commands

LSN50 supports AT Command set in the stock firmware. You can use a USB to TTL adapter to connect to LSN50 for using AT command, as below.

In the PC, you need to set the serial baud rate to **9600** to access the serial console for LSN50. LSN50 will output system info once power on as below:



Below are the available commands, a more detailed AT Command manual can be found at [AT Command Manual](http://www.dragino.com/downloads/index.php?dir=LSN50-LoRaST/&file=DRAGINO_STM_AT_Commands_v1.3.pdf)

AT+<CMD>? : Help on <CMD>
AT+<CMD> : Run <CMD>
AT+<CMD>=<value> : Set the value
AT+<CMD>=? : Get the value

General Commands

AT : Attention
AT? : Short Help
ATZ : MCU Reset
AT+TDC : Application Data Transmission Interval

Keys, IDs and EUIs management

AT+APPEUI : Application EUI
AT+APPKEY : Application Key
AT+APPSKEY : Application Session Key
AT+DADDR : Device Address
AT+DEUI : Device EUI
AT+NWKID : Network ID (You can enter this command change only after successful network connection)
AT+NWKSKEY : Network Session Key Joining and sending date on LoRa network
AT+CFM : Confirm Mode
AT+CFS : Confirm Status
AT+JOIN : Join LoRa? Network
AT+NJM : LoRa? Network Join Mode
AT+NJS : LoRa? Network Join Status
AT+RECV : Print Last Received Data in Raw Format
AT+RECVB : Print Last Received Data in Binary Format
AT+SEND : Send Text Data
AT+SENB : Send Hexadecimal Data

LoRa Network Management

AT+ADR : Adaptive Rate
AT+CLASS : LoRa Class(Currently only support class A
AT+DCS : Duty Cycle Setting

AT+DR	: Data Rate (Can Only be Modified after ADR=0)
AT+FCD	: Frame Counter Downlink
AT+FCU	: Frame Counter Uplink
AT+JN1DL	: Join Accept Delay1
AT+JN2DL	: Join Accept Delay2
AT+PNM	: Public Network Mode
AT+RX1DL	: Receive Delay1
AT+RX2DL	: Receive Delay2
AT+RX2DR	: Rx2 Window Data Rate
AT+RX2FQ	: Rx2 Window Frequency
AT+TXP	: Transmit Power

Information

AT+RSSI	: RSSI of the Last Received Packet
AT+SNR	: SNR of the Last Received Packet
AT+VER	: Image Version and Frequency Band
AT+FDR	: Factory Data Reset
AT+PORT	: Application Port
AT+CHS	: Get or Set Frequency (Unit: Hz) for Single Channel Mode
AT+CHE	: Get or Set eight channels mode, Only for US915, AU915, CN470

3.2 Common AT Command Sequence

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

If device has not joined network via OTAA:

AT+FDR
AT+NJM=0
ATZ

If device already joined network:

AT+NJM=0
ATZ

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

See Sect 6.7

4 Upload Firmware

Notes:

- Since image v1.3, the firmware will show version info during boot. If your device doesn't show version info, you may have a very old image version.
- Always run AT+FDR to reset parameters to factory default after an update image.
  If the update is from image >= v1.3 to another image version >=v1.3, then the keys will be kept after running AT+FDR.
  Otherwise (e.g. from v1.2 to v1.3), AT+FDR may erase the keys.

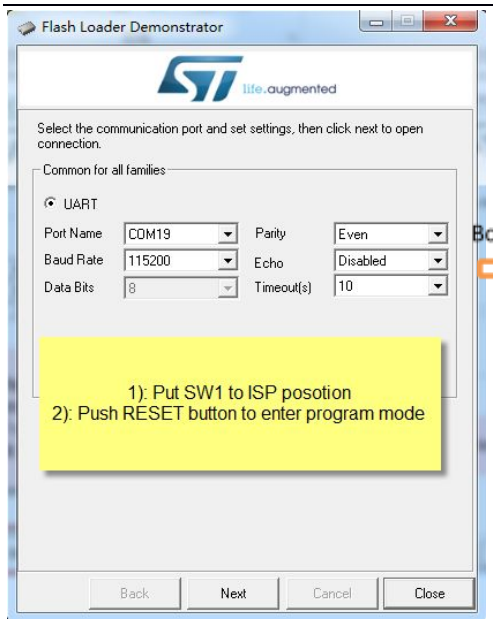
4.1 Upload Firmware via Serial Port

The LSN50's AT Command port can be used for firmware upgrade. The hardware connection for upgrade firmware is as below:

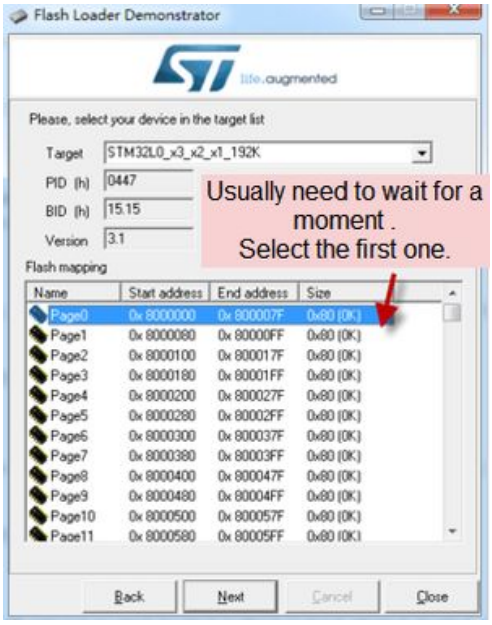
Step1: Download flash loader.

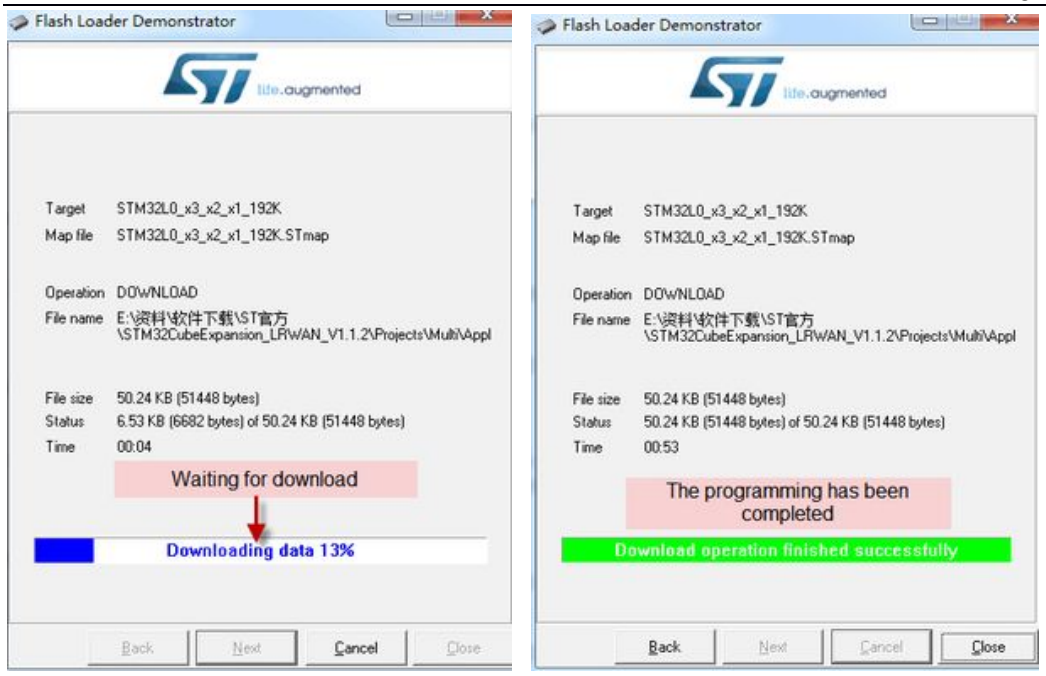
Step2: Download the LSN50 Image files.

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



Board detected

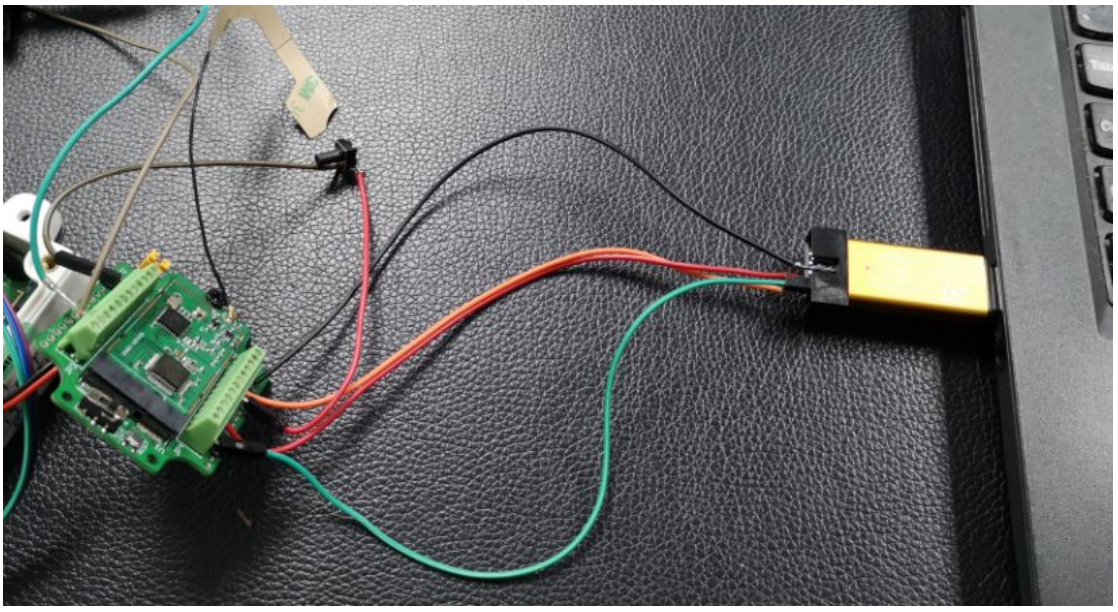




Step4: Switch SW1 back to flash state and push the RESET button.
The LSN50 will then run the new firmware.

4.2 Upload Firmware via ST-Link V2

You can use ST-LINK to upgrade firmware into LSN50. The hardware connection for upgrade firmware is as below:



Connection:

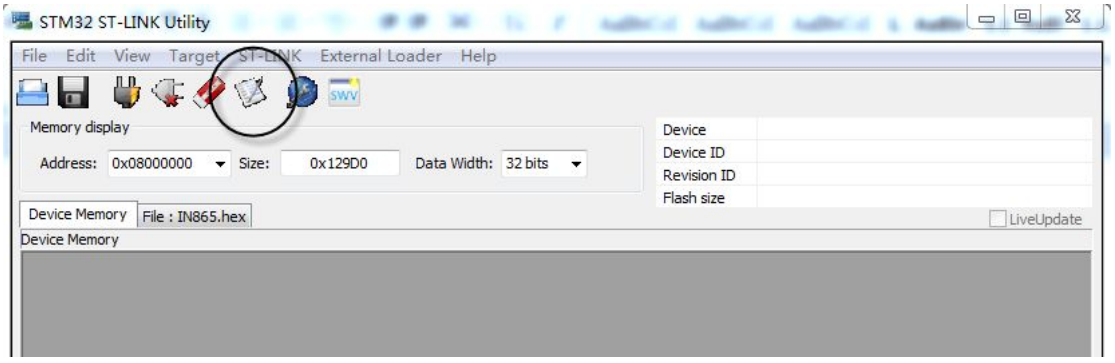
- ST-LINK v2 GND <--> LSN50 GND
- ST-LINK v2 SWCLK <--> LSN50 PA14
- ST-LINK v2 SWDIO <--> LSN50 PA13
- ST-LINK v2 RST <-->LSN50 NRST.

Step1: Install ST-LINK driver first and then install ST-LINK Utility

Step2: Download the LSN50 Image files.

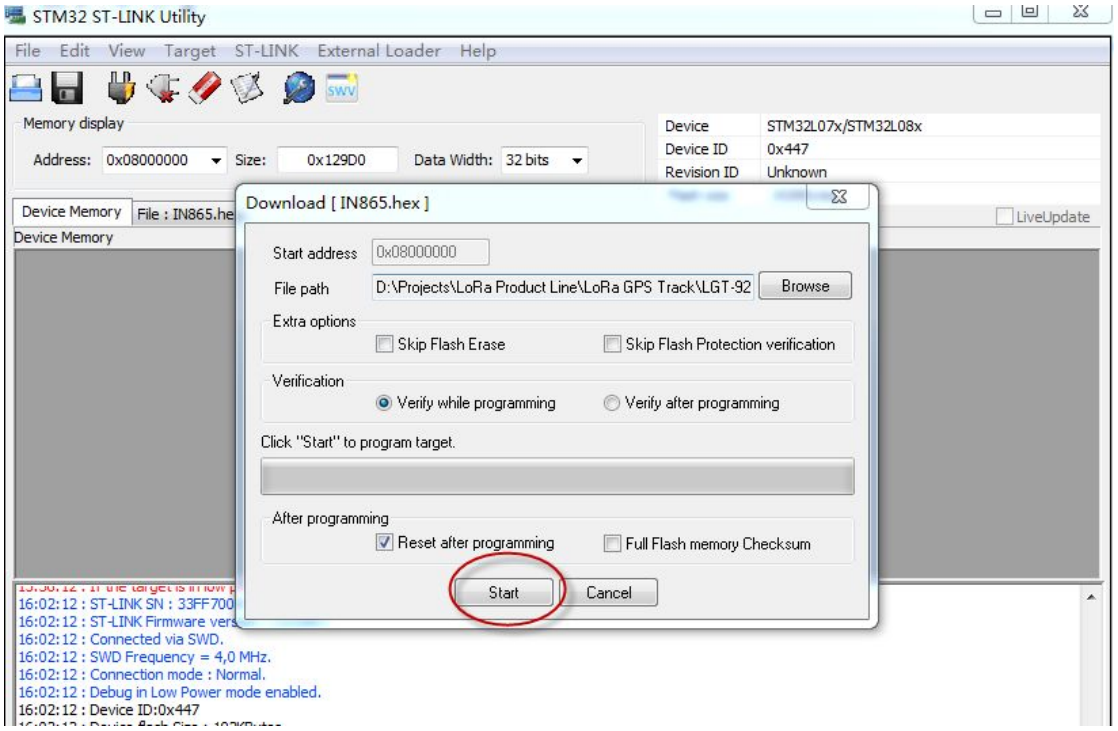
Step3: Open ST-LINK utility, **file --> open file** to select the image to be upgraded.

Step4: Click the "**Program Verify**" button on ST-LINK.



Step5: The led on the ST-LINK adapter will now blinking, and the ST-Link utility will pop up a download window. Click the start button to download the image to LSN50.

NOTE: If this step fails, ST-LINK can't establish connection to LSN50, please try to swap SWDIO & SWCLK pin. Some ST-LINK v2 devices are incorrectly marked.



5 Developer Guide

5.1 Source Code

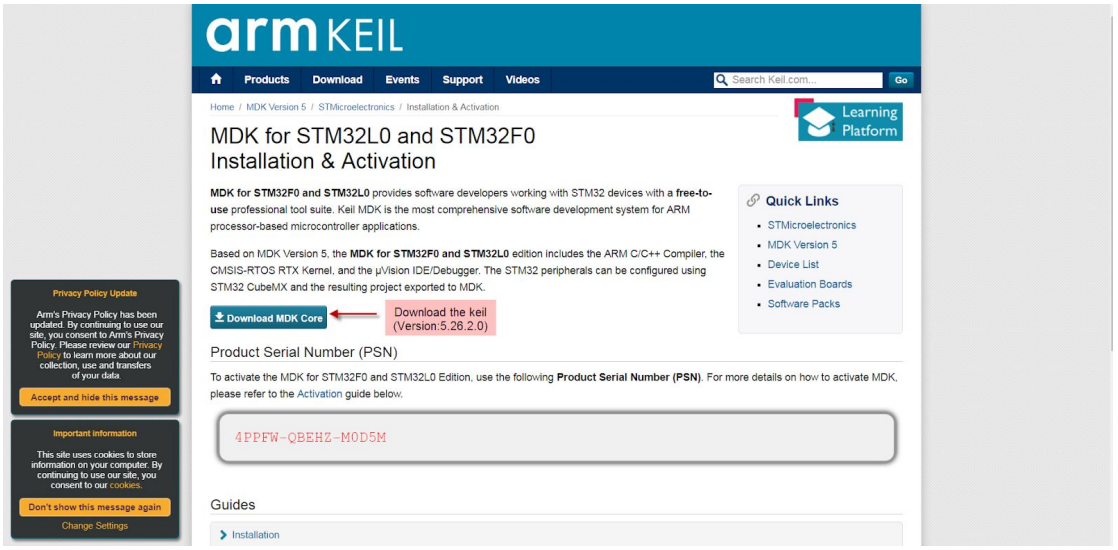
Software Source Code Download Link.
(https://github.com/dragino/LoRa_STM32/tree/master/STM32CubeExpansion_LRWAN)

Hardware Source Code Download Link
(https://github.com/dragino/Lora/tree/master/LSN50)

5.2 Compile Source Code

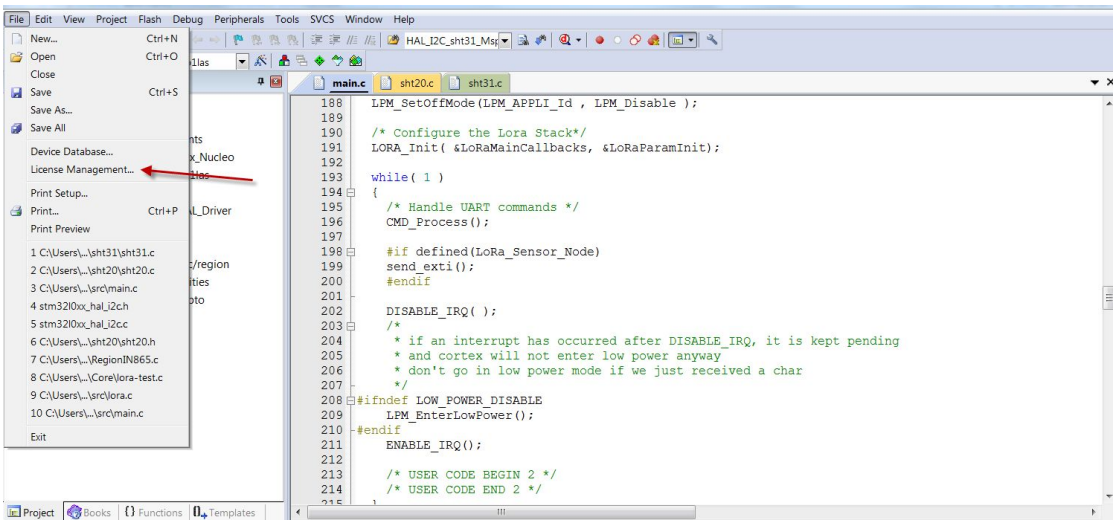
5.2.1 Set up Keil Compile Environment

Assuming you already have Keil uVision5 installed, the steps below show how to install the MDK support and get a license.
1: Open the **Webpage**: http://www2.keil.com/stmicroelectronics-stm32/mdk
2: Download the **Keil MDK**:

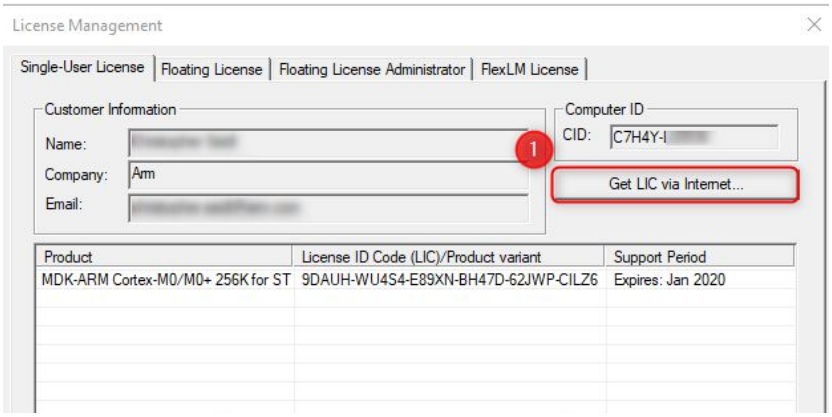


3: Login with an account that has administration rights.
4: Right-click the µVision icon and select **Run as Administrator...** from the context menu.

5: Open the dialog **File — License Management...** and select the **Single-User License tab**.

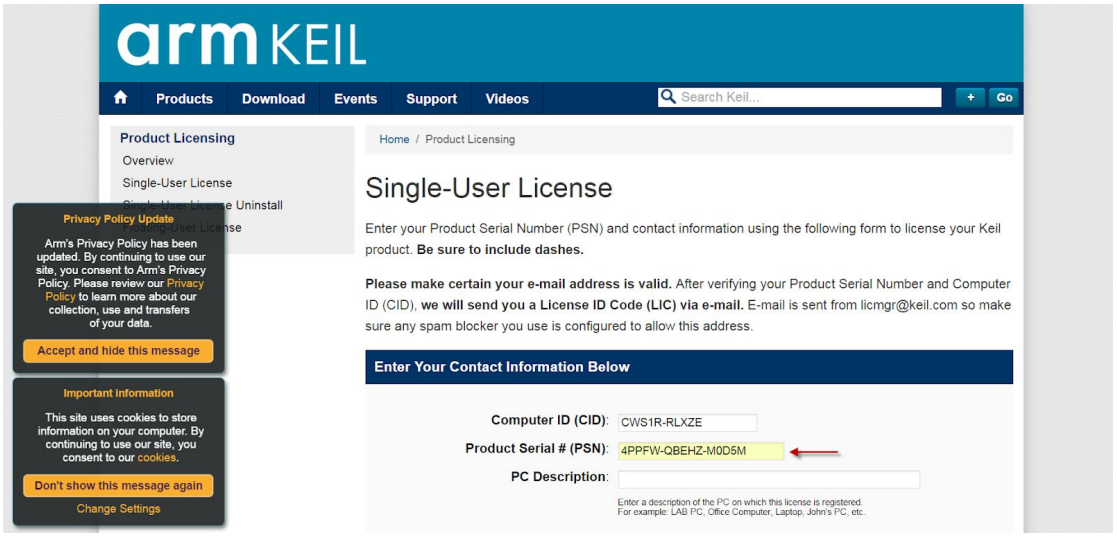


6: Click the button **Get LIC via Internet...**, then click the button **OK** to register the product. This action opens the License Management page on the Keil web site.

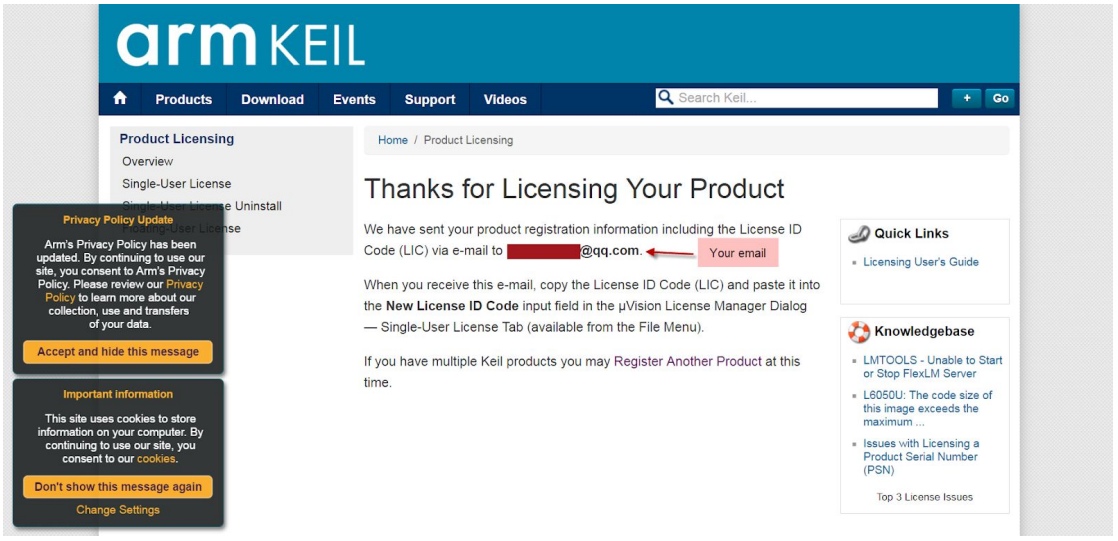


7: Enter the **Product Serial Number 4PPFW-QBEHZ-M0D5M** along with your contact information and click the button **Submit**. An e-mail is sent back with the **License ID Code (LIC)** within a few minutes.

(1)



(2)



(3)

Thank you for licensing your Keil product. Your License ID Code (LIC) is printed below. Print a copy of this e-mail to keep for your records.

MDK-ARM Cortex-M0/M0+ 256K
For ST Only
Support Ends 31 Jan 2020

PC Description : 111
Computer ID (CID): CWS1R-RLXZE

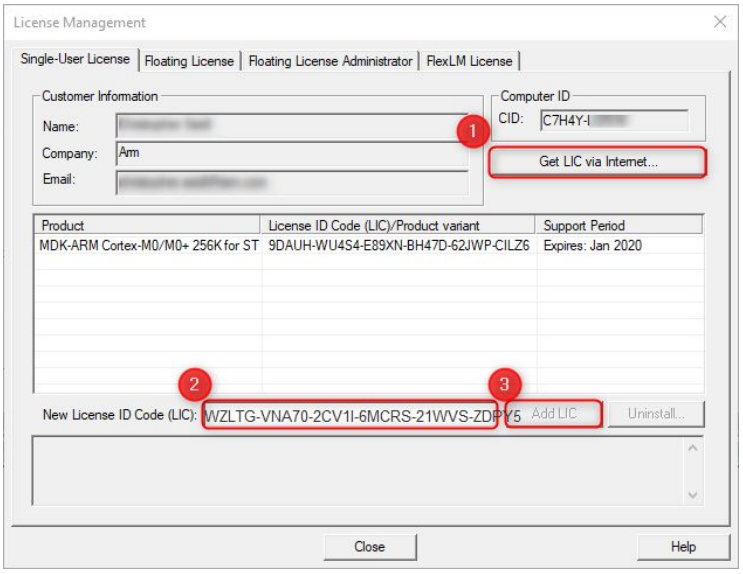
License ID Code (LIC): WZLTG-VNA70-2CV1I-6MCRS-21WVS-ZDPY5

To activate your Keil product, copy the License ID Code (LIC) and paste it into the New License ID Code input field on the Single-User License Tab in the uVision4 License Manager Dialog (available from the File menu).

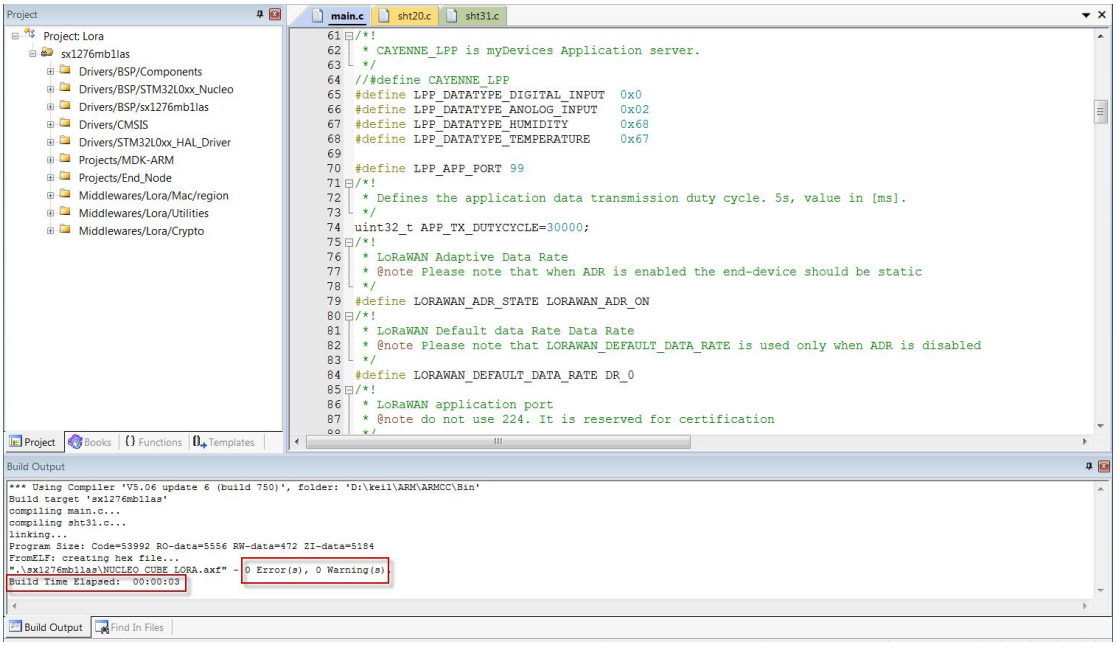
*** DO NOT REPLY TO THIS EMAIL: For licensing problems or questions, please contact Keil Technical Support.

Thank You,
Technical Support

8: To activate the Software Product, enter the **LIC** in the field **New License ID Code (LIC)** of the dialog **License Management...** and click **Add LIC**.



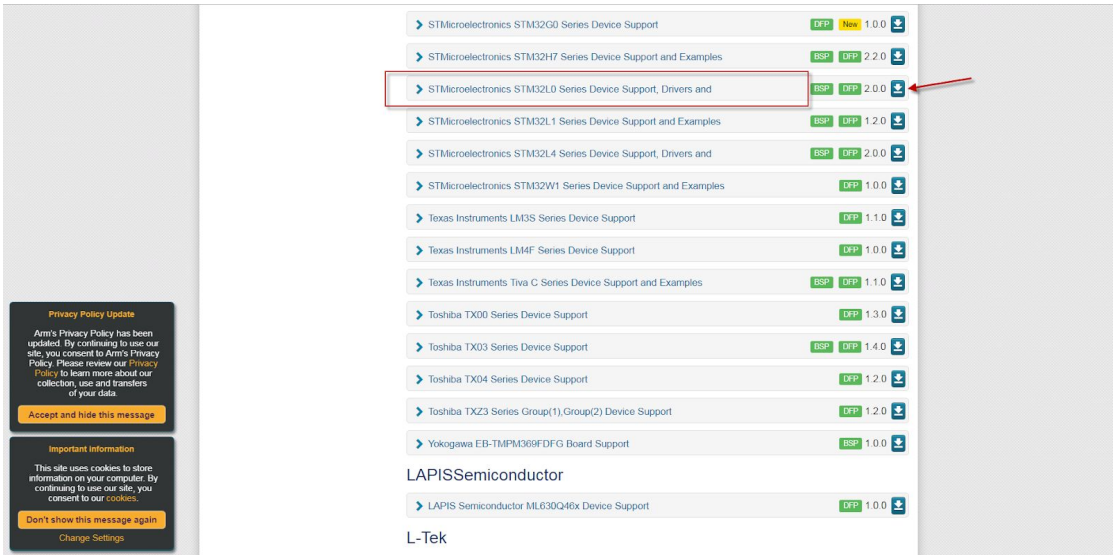
9: Finish



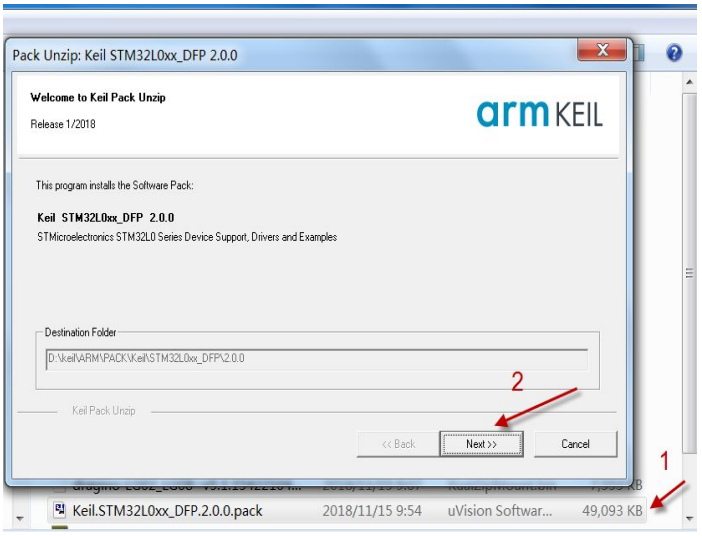
5.2.3 Install STM32L0 Series Device

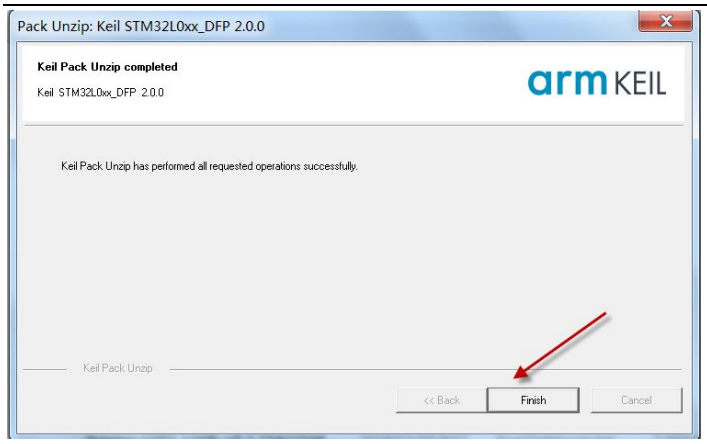
1: **Open** the webpage: http://www.keil.com/dd2/pack/eula-container;

2: **Find** the **STMicroelectronics STM32L0 Series Device** and **download** it.

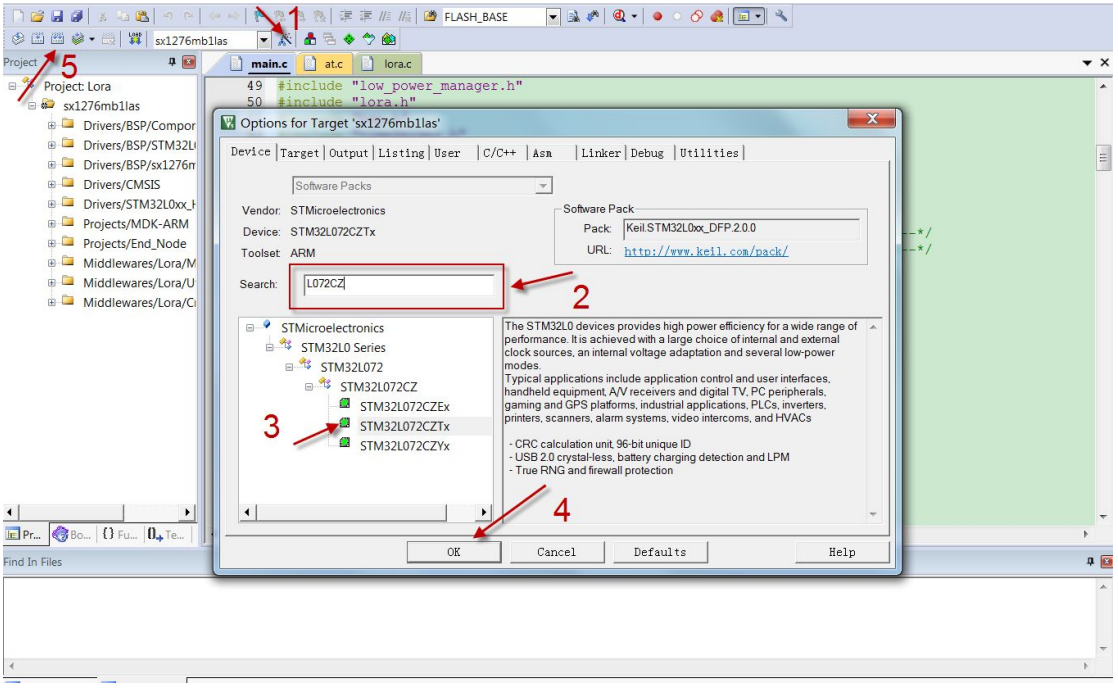


3: Find the Software Pack and install it.

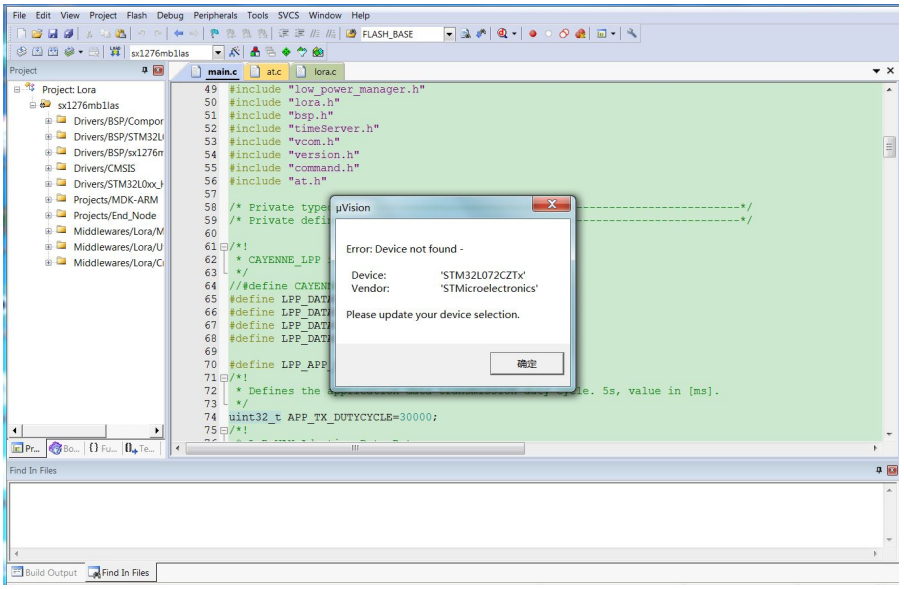




4: **Add** the Device, then you can **rebuild** the project.

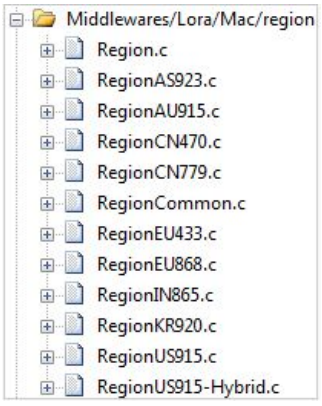


Note: If you don't add the Device, then Keil will report this error:

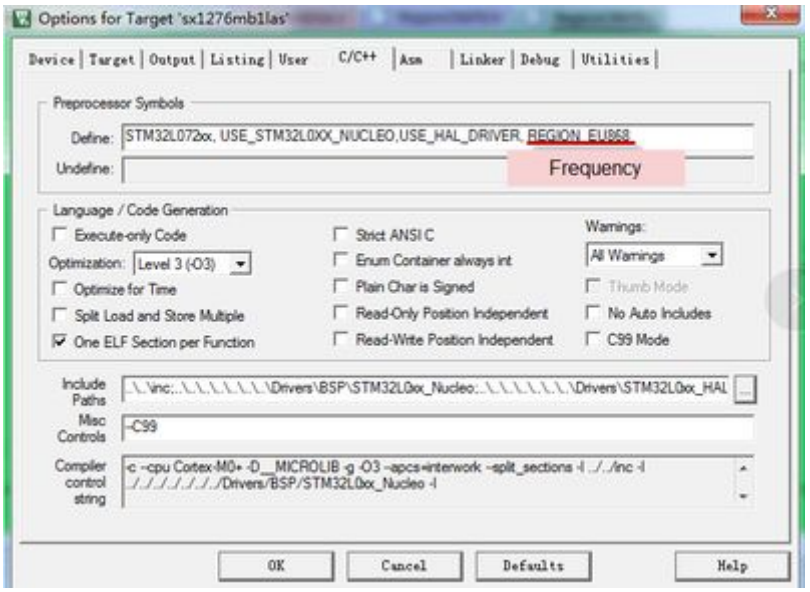


5.2.4 Compile Source Code

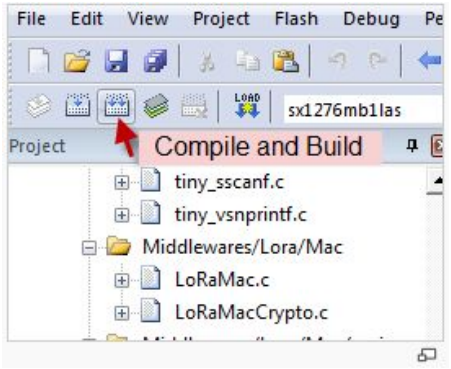
1. Download the source code from Software Source Code Download Link.
2. Use Keil to open the project file: STM32CubeExpansion_LRWAN/Projects/Multi/Applications/LoRa/DRAGINO-LRWAN(AT) /MDK-ARM/STM32L072CZ-Nucleo/Lora.uvprojx
3. In Keil, you can see what frequency band the code support.



4. If you want to change frequency, modify the Preprocessor Symbols.
   For example, change EU868 to US915



5. Compile and build



6 FAQ

6.1 Why there is 433/868/915 version?

Different countries have different rules for the ISM band for LoRa. Although the LoRa chip can support a wide range of Frequencies, we provide different versions of the hardware for best tune of the LoRa hardware part.

6.2 What is the frequency range of LT LoRa part?

Different LT version supports different frequency range, below is the table for the working frequency and recommend bands for each model.

Version	LoRa IC	Working Frequency	Best Tune Frequency	Recommend Bands
433	SX1278	Band2(LF): 410 ~525 Mhz	433Mhz	CN470/EU433
868	SX1276	Band1(HF):862~1020 Mhz	868Mhz	EU868
915	SX1276	Band1(HF):862 ~1020 Mhz	915Mhz	AS923/AU915/ KR920/US915

6.3 How to change the LoRa Frequency Bands/Region?

You can follow the instructions for how to upgrade image.
When downloading the images, choose the required image file for download.

6.4 Can I use Private LoRa protocol?

The stock firmware is based on LoRaWAN protocol. You can use a private LoRa protocol in LSN50. This section describes an example for base LoRa transfer. It is a reference/demo and we do not provide further software development support on this topic.

In this demo, we will show the communication between LoRa Shield and LSN50, both of them using the basic LoRa library. LSN50 will send a message to a LoRa Shield and the LoRa Shield will print it to the console.

LoRa Shield + UNO:
Use the LoRa Library and upload the LoRa_Receive Sketch to Arduino.

Refs:
http://www.dragino.com/downloads/index.php?dir=LSN50-LoRaST/LoRa_Raw_Example/Arduino/&file=LoRa.zip
http://www.dragino.com/downloads/downloads/LSN50-LoRaST/LoRa_Raw_Example/Arduino/LoRaReceiver.ino

Open the serial monitor to Arduino. The device acts as a LoRa Receiver and listen on the frequency 868.3Mhz by default.

LSN50:
Use the <LoRa RAW code> . The project file is in: MDK-ARM\STM32L072CZ-Nucleo\Lora.uvprojx
Compile it and Upload it to LSN50, the LSN50 will transfer on the frequency 868.3Mhz.
In the Arduino Console, it will see the received packets as below.

6.5 How to set up LSN50 to work in 8 channel mode

By default, the frequency bands US915, AU915, CN470 work in 72 frequencies. Many gateways are 8 channel gateways, and in this case, the OTAA join time and uplink schedule is long and unpredictable while the end node is hopping in 72 frequencies.

You can configure the end node to work in 8 channel mode by using the AT+CHE command. The 500kHz channels are always included for OTAA.

For example, in **US915** band, the frequency table is as below. By default, the end node will use all channels (0~71) for OTAA Join process. After the OTAA Join, the end node will use these all channels (0~71) to send uplink packets.

CHE	US915 Uplink Channels(125KHz,4/5,Unit:MHz,CHS=0)								
0	ENABLE Channel 0-63								
1	902.3	902.5	902.7	902.9	903.1	903.3	903.5	903.7	Channel 0-7
2	903.9	904.1	904.3	904.5	904.7	904.9	905.1	905.3	Channel 8-15
3	905.5	905.7	905.9	906.1	906.3	906.5	906.7	906.9	Channel 16-23
4	907.1	907.3	907.5	907.7	907.9	908.1	908.3	908.5	Channel 24-31
5	908.7	908.9	909.1	909.3	909.5	909.7	909.9	910.1	Channel 32-39
6	910.3	910.5	910.7	910.9	911.1	911.3	911.5	911.7	Channel 40-47
7	911.9	912.1	912.3	912.5	912.7	912.9	913.1	913.3	Channel 48-55
8	913.5	913.7	913.9	914.1	914.3	914.5	914.7	914.9	Channel 56-63
Channels(500KHz,4/5,Unit:MHz,CHS=0)									
	903	904.6	906.2	907.8	909.4	911	912.6	914.2	Channel 64-71

When you use the TTN network, the US915 frequency bands use are:

- 903.9 - SF7BW125 to SF10BW125
- 904.1 - SF7BW125 to SF10BW125
- 904.3 - SF7BW125 to SF10BW125
- 904.5 - SF7BW125 to SF10BW125
- 904.7 - SF7BW125 to SF10BW125
- 904.9 - SF7BW125 to SF10BW125
- 905.1 - SF7BW125 to SF10BW125
- 905.3 - SF7BW125 to SF10BW125
- 904.6 - SF8BW500

Because the end node is now hopping in 72 frequency, it makes it difficult for the devices to Join the TTN network and uplink data. To solve this issue, you can access the device via the AT commands and run:

AT+CHE=2

ATZ

to set the end node to work in 8 channel mode. The device will work in Channel 8-15 & 64-71 for OTAA, and channel 8-15 for Uplink.

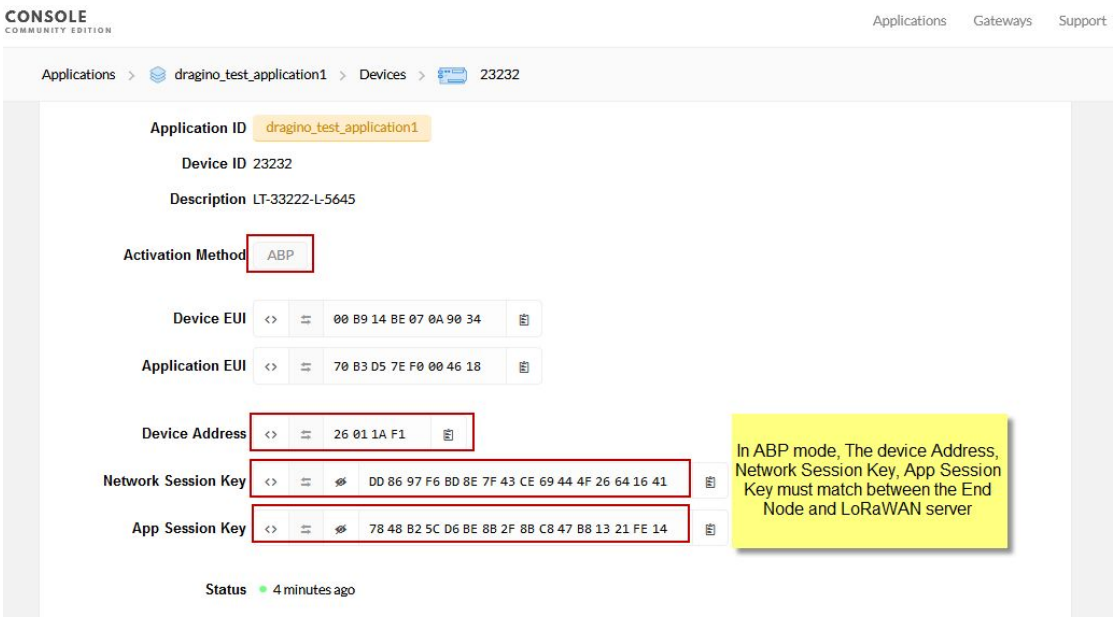
The **AU915** band is similar. Below are the AU915 Uplink Channels.

CHE	AU915 Uplink Channels(125KHz,4/5,Unit:MHz,CHS=0)								
0	ENABLE Channel 0-63								
1	915.2	915.4	915.6	915.8	916	916.2	916.4	916.6	Channel 0-7
2	916.8	917	917.2	917.4	917.6	917.8	918	918.2	Channel 8-15
3	918.4	918.6	918.8	919	919.2	919.4	919.6	919.8	Channel 16-23
4	920	920.2	920.4	920.6	920.8	921	921.2	921.4	Channel 24-31
5	921.6	921.8	922	922.2	922.4	922.6	922.8	923	Channel 32-39
6	923.2	923.4	923.6	923.8	924	924.2	924.4	924.6	Channel 40-47
7	924.8	925	925.2	925.4	925.6	925.8	926	926.2	Channel 48-55
8	926.4	926.6	926.8	927	927.2	927.4	927.6	927.8	Channel 56-63
Channels(500KHz,4/5,Unit:MHz,CHS=0)									
	915.9	917.5	919.1	920.7	922.3	923.9	925.5	927.1	Channel 64-71

6.6 How to set up LSN50 to work with Single Channel Gateway such as LG01/LG02?

In this case, users need to set LSN50 to work in ABP mode and transmit in only one frequency.
Assume we have a LG02 working in the frequency 868400000 now, below is the steps.

Step1: Log in TTN, Create an ABP device in the application and input the network session key (NETSKEY), app session key (APPSKEY) from the device.

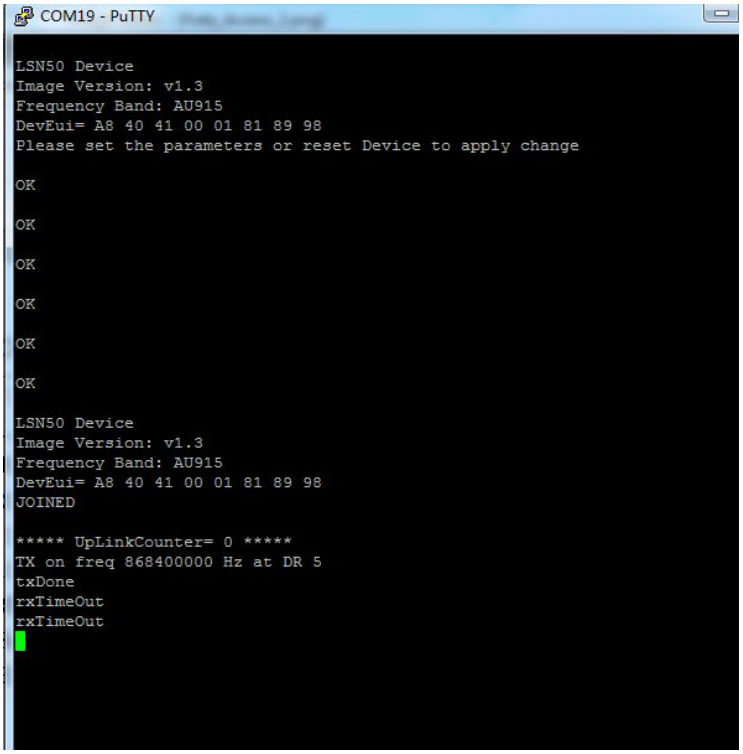


Note: You need to make sure the above three keys match in the device and in TTN. You can change them either in TTN or in the Device to make them match. In TTN, NETSKEY and APPSKEY can be configured in the setting page, but the Device Addr is generated by TTN. You can also change the Device ADDR in TTN by using the The Things Network CLI.

Step2: Run AT commands to make the LSN50 work in Single frequency and ABP mode.
Below are the AT commands:

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 (Set AT+DR=3 for 915 band)
AT+TDC=300000 Set transmit interval to 5 minutes
AT+CHS=868400000 Set transmit frequency to 868.4Mhz
AT+DADDR=26 01 1A F1 Set Device Address to 26 01 1A F1
ATZ Reset MCU

As shown below:



6.7 How to configure the EUI keys in LSN50?

The early version of LSN50 firmware doesn't have pre-configured keys.
It is recommended that you update the image to the latest version before configure the keys. Refer upgrade_image to update the firmware to the latest version.
Run AT commands to set the keys to desired keys; refer AT Command manual.

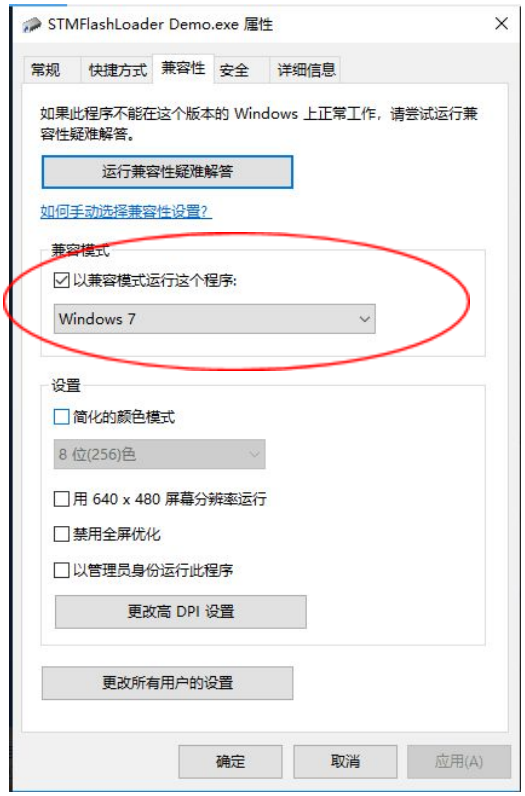
7 Trouble Shooting

7.1 Connection problem when uploading firmware.

Issue: While using USB to TTL to upload firmware via UART interface. It works for several times but most of times it fails.

Checklist:

1. Double check if follow up exactly the steps as manual.
2. Check if hardware works fine: a) check if AT command works, b) check if ISP / flash switch works: PA12 will have different output level while set the ISP/Flash Switch in different position. c) check if reset button works.
3. If you use Windows10 system. Please change the flash loader to run in Windows7 compatibility mode.



4. We have seen cases where the FT232 USB TTL adapter has a reliability issue with the PC USB chipset (Intel). In this case, even though points 1 and 2 above work, it still has a reliability issue for uploading. If this happens, change to a different PC or change the USB to TTL adapter to solve the issue.

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

It is due to channel mapping. Please see the Eight Channel Mode section above for details.

9 Order Info

Part Number: **LSN50-XX-YY**

XX: The default frequency band

- **AS923**: LoRaWAN AS923 band
- **AU915**: LoRaWAN AU915 band
- **EU433**: LoRaWAN EU433 band
- **EU868**: LoRaWAN EU868 band
- **KR920**: LoRaWAN KR920 band
- **US915**: LoRaWAN US915 band
- **IN865**: LoRaWAN IN865 band
- **CN470**: LoRaWAN CN470 band

YY:

- **12**: With M12 waterproof cable hole
- **16**: With M16 waterproof cable hole
- **20**: With M20 waterproof cable hole
- **NH**: No Hole

10 Packing Info

Package Includes:

- LSN50 LoRa Sensor Node x 1

Dimension and weight:

- Device Size: 8 x 6.5 x 5 cm
- Device Weight: 137g
- Package Size / pcs : 9 x 7 x 6cm
- Weight / pcs : 160g

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

13 References

- Product Page
  (http://www.dragino.com/products/lora/item/128-lsn50.html)

- Data Sheet
  (http://www.dragino.com/downloads/index.php?dir=datasheet/EN/&file=Datasheet_LoRaSensorNode.pdf)

- Image Download
  (https://github.com/dragino/LoRa_STM32/tree/master/LSN50.hex)

- AT Command Manual
  (http://www.dragino.com/downloads/index.php?dir=LSN50-LoRaST/&file=DRAGINO_STM_AT_Commands_v1.3.pdf)