# E106 Series LORA Wireless Modules

Official SDK instructions for using E106 series modules on Raspberry Pi



成都亿佰特电子科技有限公司
Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Catalogue

Introduction: ..... 3
Step 1 Configure the Raspberry Pi environment: ..... 错误！未定义书签。
Step 2 Download compiled source code: ..... 3
Step 3 Connecting the Raspberry Pi to the E106 ..... 错误！未定义书签。
Step 4 Testing the E106 module ..... 错误！未定义书签。
Revision History ..... 错误！未定义书签。
About Us ..... 错误！未定义书签。

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# Introduction:

This module is a hardware module that requires an external device to drive it. This document uses the Raspberry Pi used in the official semtech routines to drive the module. This gateway module uses semtech's sx1302 as the main chip, which does not have an official register manual, so you will need to refer to the official routines to drive this module and for secondary development.

## Step 1 Configure the Raspberry Pi environment

First of all the raspi system is installed on the Raspberry Pi and can be logged in remotely via SSH. If you do not have the system, you can download the image from the following link.

- raspbin lite Lite version
- raspbin full Full Version

First login to the Raspberry Pi system via SSH

- Update Linux ~ Enter the following command

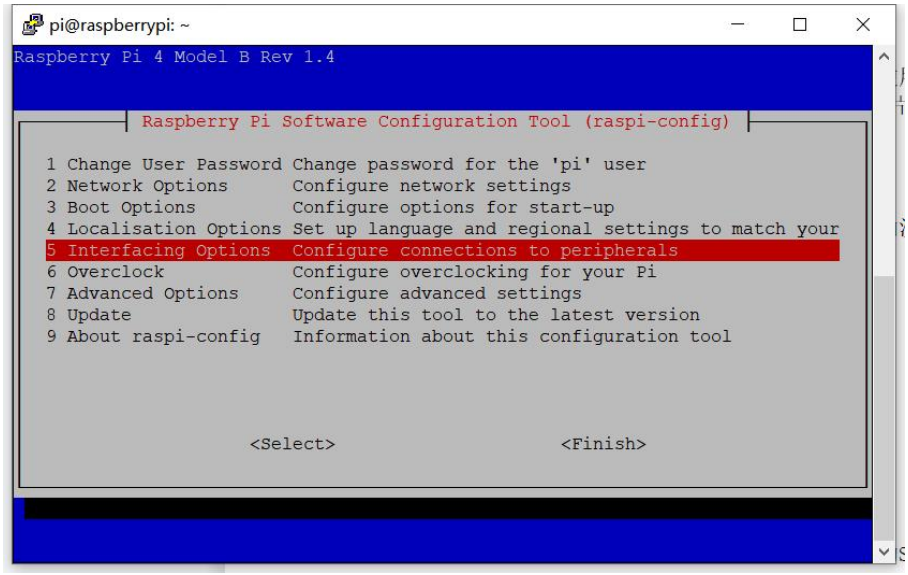
```
- $ sudo apt-get update
- $ sudo apt-get upgrade
- $ sudo apt-get dist-upgrade
- $ sudo rpi-update
```

- Installing the Git service

```
- $ sudo apt install git
```

- Turn on Linux SPI/I2C/UART (enter the command and go to Options to select Start SPI/I2C/UART)

```
- $ sudo raspi-config:
```




- Reboot by entering the reboot command

```
- $ sudo reboot
```

## Step2 Download compiled source code:

- Get the latest Semtech package from LoRa® Github (internet connection required)

```
- $ git clone https://github.com/Lora-net/sx1302_hal.git
- $ cd ~/sx1302_hal/
```

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

• As we do not use the i2c temperature sensor and ad, we need to block this part, otherwise we will get an error when using it. (You can also use the modified sdk directly from our official website, no need to modify it)

```
sudo vim libloragw/src/loragw_hac.c
```

Line 205:

```
#if 0
/* I2C temperature sensor handles */
static int ts_fd = -1;
static uint8_t ts_addr = 0xFF;
/* I2C AD5338 handles */
static int ad_fd = -1;
#endif
```

Line 1095:

```
#if 0
/* Find the temperature sensor on the known supported ports */
for (i = 0; i < (int)(sizeof I2C_PORT_TEMP_SENSOR); i++) {
ts_addr = I2C_PORT_TEMP_SENSOR[i];
err = i2c_linuxdev_open(I2C_DEVICE, ts_addr, &ts_fd);
if (err != LGW_I2C_SUCCESS) {
printf("ERROR: failed to open I2C for temperature sensor on
port 0x%02x\n", ts_addr);
return LGW_HAL_ERROR;
}
err = stts751_configure(ts_fd, ts_addr);
if (err != LGW_I2C_SUCCESS) {
printf("INFO: no temperature sensor found on port 0x%02x\n",
ts_addr);
i2c_linuxdev_close(ts_fd);
ts_fd = -1;
} else {
printf("INFO: found temperature sensor on port 0x%02x\n",
ts_addr);
break;
}
}
if (i == sizeof I2C_PORT_TEMP_SENSOR) {
printf("ERROR: no temperature sensor found.\n");
return LGW_HAL_ERROR;
}
#endif
/* Configure ADC AD338R for full duplex (CN490 reference design) */
if (CONTEXT_BOARD.full_duplex == true) {
err = i2c_linuxdev_open(I2C_DEVICE, I2C_PORT_DAC_AD5338R,
&ad_fd);
if (err != LGW_I2C_SUCCESS) {
printf("ERROR: failed to open I2C for ad5338r\n");
return LGW_HAL_ERROR;
}
err = ad5338r_configure(ad_fd, I2C_PORT_DAC_AD5338R);
if (err != LGW_I2C_SUCCESS) {
printf("ERROR: failed to configure ad5338r\n");
i2c_linuxdev_close(ad_fd);
ad_fd = -1;
return LGW_HAL_ERROR;
}/* Turn off the PA: set DAC output to 0V */
uint8_t volt_val[AD5338R_CMD_SIZE] = { 0x39,
(uint8_t)VOLTAGE2HEX_H(0), (uint8_t)VOLTAGE2HEX_L(0) };
err = ad5338r_write(ad_fd, I2C_PORT_DAC_AD5338R, volt_val);
if (err != LGW_I2C_SUCCESS) {
printf("ERROR: AD5338R: failed to set DAC output to 0V\n");
return LGW_HAL_ERROR;
}
printf("INFO: AD5338R: Set DAC output to 0x%02X 0x%02X\n",
(uint8_t)VOLTAGE2HEX_H(0), (uint8_t)VOLTAGE2HEX_L(0));
}
#endif
```

Line 1225:

```
#if 0
DEBUG_MSG("INFO: Closing I2C for temperature sensor\n");
x = i2c_linuxdev_close(ts_fd);
if (x != 0) {
printf("ERROR: failed to close I2C temperature sensor device
(err=%i)\n", x);
err = LGW_HAL_ERROR;
}
if (CONTEXT_BOARD.full_duplex == true) {
DEBUG_MSG("INFO: Closing I2C for AD5338R\n");
x = i2c_linuxdev_close(ad_fd);
if (x != 0) {
```

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
printf("ERROR: failed to close I2C AD5338R device
(err=%i)\n", x);
err = LGW_HAL_ERROR;
}
}
#endif
```

```
Line 1289:
#if 0
res = lgw_get_temperature(&current_temperature);
if (res != LGW_I2C_SUCCESS) {
printf("ERROR: failed to get current temperature\n");
return LGW_HAL_ERROR;
}
#endif
```

```
Line 1592:
int err = LGW_HAL_ERROR;Change to int err = LGW_HAL_SUCCESS;
Line 1600:
//err = stts751_get_temperature(ts_fd, ts_addr, temperature);
```

• Compiling the source code

In the sx1302_hal directory

- $ make clean all
- $ ssh-keygen -t rsa The following two commands are executed to avoid entering the user password when installing the file
- $ ssh-copy-id -i ~/.ssh/id_rsa.pub pi@localhost You should install all programs after entering the raspberry pi password.
- $ make install
- $ make install_conf将global_conf.json Copy the executable file to the bin folder

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## Step3 Connecting the Raspberry Pi to the E106

This document is about connecting the E106 module to the Raspberry Pi using a mini-pcie to spi adapter board. The main connections are the four SPI pins, the sx1302_reset pin, vcc, and gnd.

Use different adapter boards with different sx1302_reset pins connected to the Raspberry Pi, modify the reset_lgw.sh file under the sx1302_hal/bin folder as appropriate

```
- $ sudo vim reset_lgw.sh
Due to the adapter board we changed the SX1302_RESET_PIN from 23 to 7
SX1302_RESET_PIN=7 #SX1302 RESET
```



## Step 4 Testing the E106 module

1、Register test

After logging in, go to the directory where the test instructions are located: cd lorawan/sx1302_hal/bin

Then look at the files under the current folder: ls

```
pi@raspberrypi:~ $ cd lorawan/sx1302_hal/bin
pi@raspberrypi:~/lorawan/sx1302_hal/bin $ ls
boot                                  spectral_scan
chip_id                               test_loragw_cal_sx125x
global_conf.json                      test_loragw_capture_ram
global_conf.json.sx1250.AS923.USB     test_loragw_com
global_conf.json.sx1250.CN490         test_loragw_com_sx1250
global_conf.json.sx1250.CN490.USB     test_loragw_com_sx1261
global_conf.json.sx1250.EU868         test_loragw_counter
global_conf.json.sx1250.EU868.USB     test_loragw_gps
global_conf.json.sx1250.US915         test_loragw_hal_rx
global_conf.json.sx1250.US915.USB     test_loragw_hal_tx
global_conf.json.sx1255.CN490.full-duplex  test_loragw_i2c
global_conf.json.sx1257.EU868         test_loragw_reg
lora_pkt_fwd                          test_loragw_sx1261_rssi
net_downlink                          test_loragw_toa
reset_lgw.sh
pi@raspberrypi:~/lorawan/sx1302_hal/bin $
```

Enter the test command

```
- $ ./test_loragw_reg
```

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
pi@raspberrypi:~/lorawan/sx1302_hal/bin $ ./test_loragw_reg
CoreCell reset through GPIO7...
SX1261 reset through GPIO7...
CoreCell power enable through GPIO18...
CoreCell ADC reset through GPIO13...
Opening SPI communication interface
Note: chip version is 0x10 (v1.0)
## TEST#1: read all registers and check default value for non-read-only register
s
-----------------
 TEST#1 PASSED
-----------------

## TEST#2: read/write test on all non-read-only, non-pulse, non-w0clr, non-w1clr
 registers
-----------------
 TEST#2 PASSED


Closing SPI communication interface
CoreCell reset through GPIO7...
SX1261 reset through GPIO7...
CoreCell power enable through GPIO18...
CoreCell ADC reset through GPIO13...
pi@raspberrypi:~/lorawan/sx1302_hal/bin $
```

TEST1 ,TEST2 are PASSED means the register test is passed, if it fails, please check if the reset pin is correct, if the spi communication is normal, if the register test is passed, it means the module is basically OK.

2、Sending tests

First use the command: . /test_loragw_hal_tx -h to see the help information and set the different parameters as required.

```
pi@raspberrypi:~/lorawan/sx1302_hal/bin $ ./test_loragw_hal_tx -h
Available options:
 -h print this help
 -u         Set COM type as USB (default is SPI)
 -d <path>  COM path to be used to connect the concentrator
            => default path: /dev/spidev0.0
 -k <uint>  Concentrator clock source (Radio A or Radio B) [0..1]
 -c <uint>  RF chain to be used for TX (Radio A or Radio B) [0..1]
 -r <uint>  Radio type (1255, 1257, 1250)
 -f <float> Radio TX frequency in MHz
 -m <str>   modulation type ['CW', 'LORA', 'FSK']
 -o <int>   CW frequency offset from Radio TX frequency in kHz [-65..65]
 -s <uint>  LoRa datarate 0:random, [5..12]
 -b <uint>  LoRa bandwidth in khz 0:random, [125, 250, 500]
 -l <uint>  FSK/LoRa preamble length, [6..65535]
 -n <uint>  Number of packets to be sent
 -z <uint>  size of packets to be sent 0:random, [9..255]
 -t <uint>  TX mode timestamped with delay in ms. If delay is 0, TX mode GPS trigger
 -p <int>   RF power in dBm
 -i         Send LoRa packet using inverted modulation polarity
 -j         Set radio in single input mode (SX1250 only)
 --fdev <uint>  FSK frequency deviation in kHz [1:250]
 --br   <float> FSK bitrate in kbps [0.5:250]
 --pa   <uint> PA gain SX125x:[0..3], SX1250:[0,1]
 --dig  <uint> sx1302 digital gain for sx125x [0..3]
 --dac  <uint> sx125x DAC gain [0..3]
 --mix  <uint> sx125x MIX gain [5..15]
 --pwid <uint> sx1250 power index [0..22]
 --nhdr        Send LoRa packet with implicit header
 --loop        Number of loops for HAL start/stop (HAL unitary test)
 --fdd         Enable Full-Duplex mode (CN490 reference design)
pi@raspberrypi:~/lorawan/sx1302_hal/bin $
```

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Several important parameters are configured.

| Command | Parameters | Functions |
|---|---|---|
| -c | 0 or 1 | Configure RF chip A(0) or RF or RF chip B(1) for transmitting |
| -r | 1255,1257,1250 | RF chip model selection, our module uses the 1250 |
| -f | Frequency（MHZ） | Set the frequency of sending |
| -m | ‘CW’ , ’ LORA’ , ‘FSK’ | Working mode |
| -o | [-65..65]khz | Transmit frequency bias |
| -s | [5…12] | Spread Spectrum Factor |
| -b | [125,250,500] | Bandwidth |
| -l | [6…65535] | Leading code length of Fsk/lora |
| -n | Arbitrary positive numbers | Number of packets sent |
| -z | [9…255] | How many bytes to send per packet |
| --pa | [0,1] | RF chip pa |
| --pwid | [0…22] | Transmitting power |

Example:

./test_loragw_hal_tx -r 1250 -f 915.2 -m LORA -b 125 -s 12 -z 20 --pa 1 --pwid 20

```
pi@raspberrypi:~/lorawan/sx1302_hal/bin $ ./test_loragw_hal_tx -r 1250 -f 915.2 -m LORA -b 125 -s 12 -z 20 --p
a 1 --pwid 2
Sending 1 LoRa packets on 915200000 Hz (BW 125 kHz, SF 12, CR 1, 20 bytes payload, 8 symbols preamble, explici
t header, non-inverted polarity) at 0 dBm
CoreCell reset through GPIO7...
SX1261 reset through GPIO7...
CoreCell power enable through GPIO18...
CoreCell ADC reset through GPIO13...
Opening SPI communication interface
Note: chip version is 0x10 (v1.0)
INFO: using legacy timestamp
ARB: dual demodulation disabled for all SF
TX done

Nb packets sent: 1 (1)
Closing SPI communication interface
CoreCell reset through GPIO7...
SX1261 reset through GPIO7...
CoreCell power enable through GPIO18...
CoreCell ADC reset through GPIO13...
========== Test End ==========
pi@raspberrypi:~/lorawan/sx1302_hal/bin $
```

2、Configured for use as a gateway

Different regional standards require different json files, e.g., we are using EU868 band and the gateway module uses spi connection, so we use global_conf.json.sx1250.EU868 file.

Edit this file to set the gateway id and specify the server address

- $ sudo vim global_conf.json.sx1250.EU868

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        "chan_multiSF_3": {"enable": true, "radio": 0, "if": -400000},
        "chan_multiSF_4": {"enable": true, "radio": 0, "if": -200000},
        "chan_multiSF_5": {"enable": true, "radio": 0, "if":  0},
        "chan_multiSF_6": {"enable": true, "radio": 0, "if":  200000},
        "chan_multiSF_7": {"enable": true, "radio": 0, "if":  400000},
        "chan_Lora_std":  {"enable": true, "radio": 1, "if": -200000, "bandwidth
": 250000, "spread_factor": 7,
                           "implicit_hdr": false, "implicit_payload_length": 17,
 "implicit_crc_en": false, "implicit_coderate": 1},
        "chan_FSK":       {"enable": true, "radio": 1, "if":  300000, "bandwidth
": 125000, "datarate": 50000}
    },

    "gateway_conf": {
        "gateway_ID": "FFFEE45F011396FE",
        /* change with default server address/ports */
        "server_address": "eu1.cloud.thethings.network",
        "serv_port_up": 1700,
        "serv_port_down": 1700,
        /* adjust the following parameters for your network */
        "keepalive_interval": 10,
        "stat_interval": 30,
        "push_timeout_ms": 100,
```

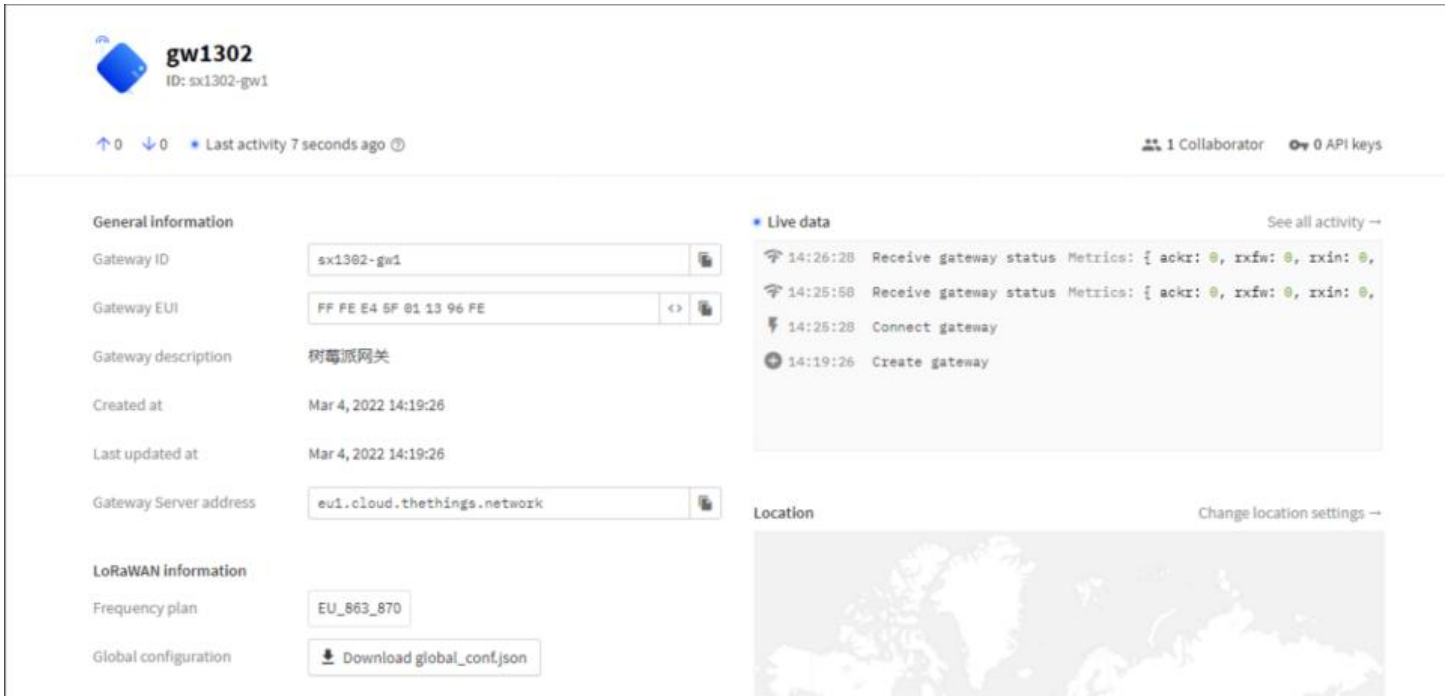
网关id

服务器地址

端口号

Once set up, in the sx1302_hal/bin directory, type
- $ ./lora_pkt_fwd -c global_conf.json.sx1250.EU868

Here we are using the TTN server, and registered the gateway on it in advance, at this time to check the status of the gateway found that it has been online.




Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# Revision History

| Version | Revision Date | Revision Notes | Maintainer |
|---|---|---|---|
| 1.0 | 2023-1-4 | Initial Version | Luo |

# About us

Sales Hotline: 4000-330-990 Company Phone: 028-61399028
Technical Support: support@cdebyte.com Address: Building B5, No. 199, West Avenue, High-tech West District, Chengdu, Sichuan Province

成都亿佰特电子科技有限公司
Chengdu Ebyte Electronic Technology Co.,Ltd.

Copyright ©2012–2020， Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000