



E106系列LORA无线模块

E106系列模块在树莓派上使用官方SDK说明

目录

简介:	3
第一步 配置树莓派环境	3
第二步 下载编译源码:	3
第三步 连接树莓派和E106	5
第四步 测试E106模块	6
修订历史	9
关于我们	9

简介:

该模块是硬件模块需要外部的设备来驱动它工作，本文档使用semtech官方例程所使用的树莓派来驱动该模块。此网关模块是使用semtech的sx1302作为主芯片，该芯片官方未提供寄存器手册，故驱动此模块和二次开发都需要参考官方的例程。

第一步 配置树莓派环境

首先树莓派中安装好了raspi系统并能通过SSH远程登陆。如没有系统可从以下链接下载镜像

- [raspbmin lite 精简版](#)
- [raspbmin full 完整版](#)

首先通过SSH登录到树莓派系统

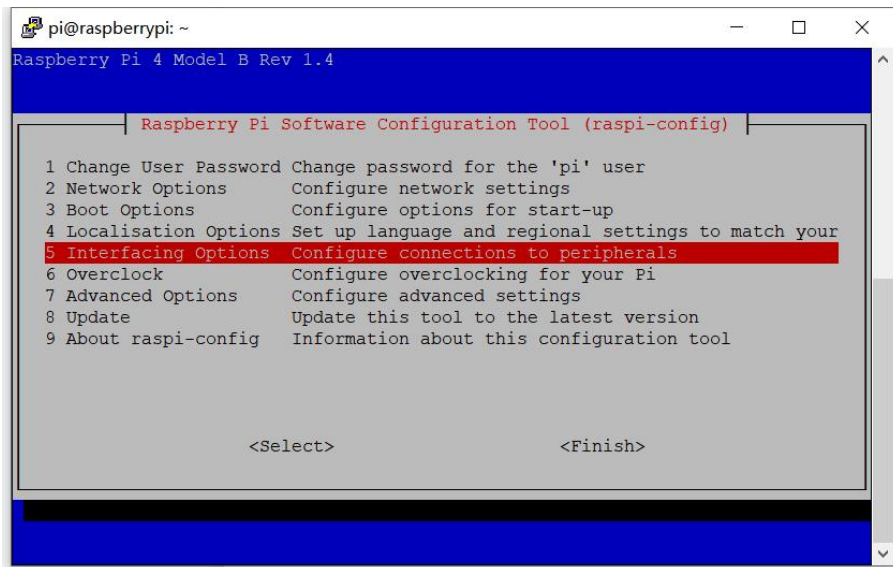
- 更新Linux ~ 按下列指令输入更
- \$ sudo apt-get update
- \$ sudo apt-get upgrade
- \$ sudo apt-get dist-upgrade
- \$ sudo rpi-update

- 安装Git 服务

- \$ sudo apt install git

- 开启Linux SPI/I2C/UART (输入指令后进入选项选择启动SPI/I2C/UART)

- \$ sudo raspi-config:



- 输入reboot指令重新启动

- \$ sudo reboot

第二步 下载编译源码:

- 从LoRa®Github获取最新的Semtech软件包 (需要连接到互联网)

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

- 由于我们没有使用i2c温度传感器和ad，所以需要屏蔽掉这部分，不然使用时会报错。(也可以直接使用我们官网修改好了的sdk，无需修改)

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

```

205行:
#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

1095行:
#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

1225行:
#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) {
printf("ERROR: failed to close I2C AD5338R device
(err=%i)\n", x);
err = LGW_HAL_ERROR;
}
}
}
#endif

```

```
1289行:
#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

1592行:
int err = LGW_HAL_ERROR;改为int err = LGW_HAL_SUCCESS;
1600行:
//err = stts751_get_temperature(ts_fd, ts_addr, temperature);
```

- 编译源码

在sx1302_hal目录下

```
- $ make clean all
- $ ssh-keygen -t rsa执行下述两个命令是为了避免在安装文件时输入用户密码
- $ ssh-copy-id -i ~/.ssh/id_rsa.pub pi@localhost您应该在输入raspberrypi密码后安装所有程式
- $ make install
- $ make install_conf将global_conf.json等可执行文件复制到bin文件夹中
```

第三步 连接树莓派和E106

本文档是使用mini-pcie转spi转接板把E106模块与数莓派相连接，主要连接是SPI的四根引脚、sx1302_reset脚、vcc、gnd就可以了。

使用不同的转接板的sx1302_reset脚连接到树莓派上的位置不同，根据实际情况修改sx1302_hal/bin文件夹下面的reset_lgw.sh文件

```
- $ sudo vim reset_lgw.sh
```

由于转接板的原因我们修改SX1302_RESET_PIN由23改为7
SX1302_RESET_PIN=7 #SX1302 RESET



第四步 测试E106模块

1、寄存器测试

登陆后进入测试指令所在目录：`cd lorawan/sx1302_hal/bin`
 然后查看当前文件夹下面的文件：`ls`

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

输入测试命令
`- $./test_loragw_reg`

```
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 registers
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都PASSED表示寄存器测试通过, 如果失败请检查reset脚是否正确、spi通信是否正常, 如果寄存器测试通过表示模块基本无问题。

2、发送测试

首先使用指令：`./test_loragw_hal_tx -h` 查看帮助信息, 根据需求设置不同的参数。

```

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 $

```

几个重要参数配置

命令	参数	作用
-c	0或1	配置射频芯片A(0)或射频或者射频芯片B(1)为发送
-r	1255, 1257, 1250	射频芯片型号选择，我们模块使用的是1250
-f	频率 (MHZ)	设置发送的频率
-m	'CW' , 'LORA' , 'FSK'	工作模式
-o	[-65..65]khz	发送频偏
-s	[5...12]	扩频因子
-b	[125, 250, 500]	频宽
-l	[6...65535]	Fsk/lora的前导码长度
-n	任意正数	发送的包数
-z	[9...255]	每包发送多少字节
--pa	[0, 1]	射频芯片pa
--pwid	[0...22]	发射功率

示例如：
./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、配置为网关使用

不同的地区标准需要使用不同的json文件，列如：此时我们使用EU868频段，网关模块使用spi连接，所以此时使用global_conf.json.sx1250.EU868 文件。

编辑此文件设置网关id和指定服务器地址

- \$ sudo vim global_conf.json.sx1250.EU868

```

"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": "FFFE45F011396FE",
  /* change with default server address/ports */
  "server address": "eul.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,

```

设置好之后，在sx1302_hal/bin目录下输入

- \$./lora_pkt_fwd -c global_conf.json.sx1250.EU868

此处我们是使用的TTN服务器，并提前在上面注册了该网关，此时查看网关状态发现其已经上线。

gw1302
ID: sx1302-gw1

↑ 0 ↓ 0 • Last activity 7 seconds ago

1 Collaborator 0 API keys

General information

Gateway ID: sx1302-gw1

Gateway EUI: FF FE E4 5F 01 13 96 FE

Gateway description: 树莓派网关

Created at: Mar 4, 2022 14:19:26

Last updated at: Mar 4, 2022 14:19:26

Gateway Server address: eu1.cloud.thethings.network

LoRaWAN information

Frequency plan: EU_863_870

Global configuration: Download global_conf.json

Live data

14:26:28 Receive gateway status Metrics: { ackr: 0, rxfw: 0, rxin: 0, ... }

14:25:58 Receive gateway status Metrics: { ackr: 0, rxfw: 0, rxin: 0, ... }

14:25:28 Connect gateway

14:19:26 Create gateway

Location

Change location settings

修订历史

版本	修订日期	修订说明	维护人
1.0	2023-1-4	初始版本	Luo

关于我们



销售热线：4000-330-990

技术支持：support@cdebyte.com

公司地址：四川省成都市高新西区西区大道 199 号 B5 栋

公司电话：028-61399028

官方网站：www.ebyte.com

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