

M-GW1302S 用户手册

LoRa 网关模组

V1.0

Released

南京芮捷电子科技有限公司

地址：南京市浦口高新区星火路 20 号

电话：156 5102 8736

邮箱：sales@rejee.com

网址：www.rejee.com

◆ 关于本手册

本文档主要介绍 M-GW1302S 的使用方式、操作说明和注意事项等。

◆ 文档变更通知

用户可以通过芮捷官网 www.rejee.com，淘宝商店或相关技术支持人员获取技术资料。

◆ 免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归 © 2020 芮捷所有。保留所有权利。

目 录

1. 产品介绍.....	4
1.1. 产品概述.....	4
1.2. 产品特性.....	4
2. 系统说明.....	5
2.1. 原理框图.....	5
2.2. 树莓派 4 特点.....	5
3. 使用说明.....	6
3.1. 登录系统.....	6
3.2. 源码介绍.....	6
3.3. 网关射频模组.....	8
3.3.1. 寄存器读写测试.....	8
3.3.2. 发送测试.....	9
3.3.3. 接收测试.....	9
3.4. 总结.....	10
4. 版本修订历史.....	11

1. 产品介绍

1.1. 产品概述

芮捷电子推出的 M-GW1302S 是采用 SX1302 芯片、工业级标准设计、基于 mini PCI-E 接口的 LoRa 网关射频模组。核心部件 SX1302 采用 SPI 方式与外部通信，可满足对接各种平台和系统，使用户能够灵活集成在自己的项目中。

SX1302 芯片相对于上一代 SX1301 不仅增加了 SF5 和 SF6 的支持，提升了传输速率，同时极大降低了射频的收发功耗。另外，支持 64 位的唯一标识号可用于标记网关编号或地址。

1.2. 产品特性

M-GW1302S 是针对 LPWAN 领域应用需求推出的新一代网关模组，兼容芮捷电子上一代 mini PCI-E 接口的射频网关评估板和底板，采用 SPI 方式通信。

主要无线特性如下：

8 路 125KHz 接收通道，可同时接收来自 SX127x 或 SX126x 系列芯片信号；

1 路 LoRa 通道（支持 125KHz、250KHz、500KHz），1 路(G)FSK；

每个接收通道接收灵敏度高达-142dBm；

每个接收通道可解 SF5~SF12 信号；

发送功率可调，最大 27dBm。

2. 系统说明

本文档为了便于说明，采用官方参考的 Linux 运行方式，以树莓派系统为平台，编译和执行 M-GW1302S 的驱动库和示例代码。

2.1. 原理框图

本文档介绍的 M-GW1302S 产品为不含系统的 SPI 通信接口的 LoRa 网关射频模组，主要提供给客户进行 LoRa 网关产品的开发，支持 LoRaWAN Class A/B/C 及所有地区标准。只需驱动 SPI 接口操作 SX1302 即可。如需透传版本的，可选择 M-GW1302。

本文档以最新树莓派系统，搭载树莓派 4 硬件进行操作说明。评估套件原理框图如下图所示，与公司上一代评估套件兼容。

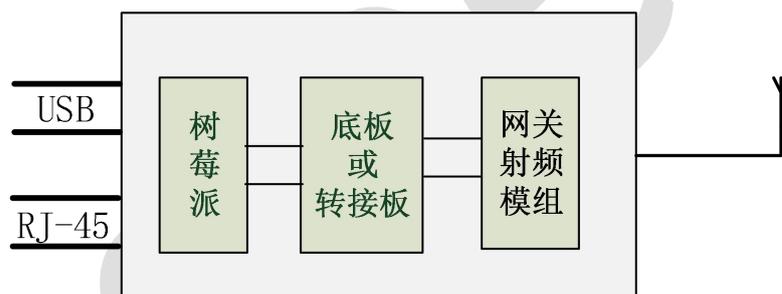


图 1 MPCI-GW2-EV 原理框图

注意：树莓派新的操作系统向下兼容，可在树莓派 3B+和树莓派 4 上运行。但是旧的树莓派操作系统或镜像（之前 3B+的旧镜像），不能运行在树莓派 4 上。

2.2. 树莓派 4 特点

CPU 频率提升到 1.5GHZ，内存升级为 LPDDR4；

全尺寸 HDMI 端口升级为 micro-HDMI，并支持 4K 视频输出；

支持千兆网卡；供电口升级为 USB-C 接口。

3. 使用说明

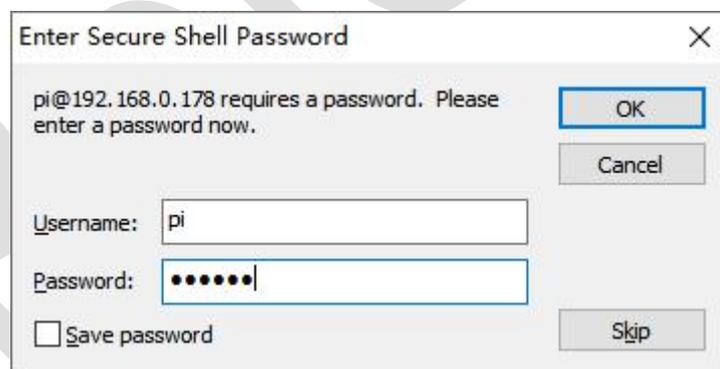
树莓派系统采用最新版本 4.19。有线网络默认使用 DHCP (IP 可以使用 IP 扫描工具扫, 如 advanced-ip-scanner 之类工具), 在无 DHCP 的情况下 **默认 IP 为: 192.168.0.178**。登录用户 **pi**, 密码: **rejeee**。

由于新树莓派 4 硬件采用是 micro-HDMI 接口, 需要专门的转接线, 可能不太方便接显示器。因此内置远程桌面服务, 如果需要使用桌面, 也可通过 Windows 远程桌面登录系统。

3.1. 登录系统

树莓派评估板配套的树莓派系统为桌面系统, 可外接显示器和键鼠操作, 也可通过 SSH 登录, 如果在 Windows 下操作, 相关 SSH 工具如 Putty 或 SecureCRT, 下载安装请参考对应工具网站, 在此就不赘述。

使用 SSH 登录的方式, 如 SecureCRT, 通过快速连接, 新建一个 session, 登录系统, 如下所示。用户 **pi**, 密码: **rejeee**



注意: 如果像上述类似操作, 通过电脑网线直连树莓派, 需要将电脑的有线网卡配置为 192.168.0.x 网段, 如可设置电脑 IPv4 地址为 192.168.0.1。

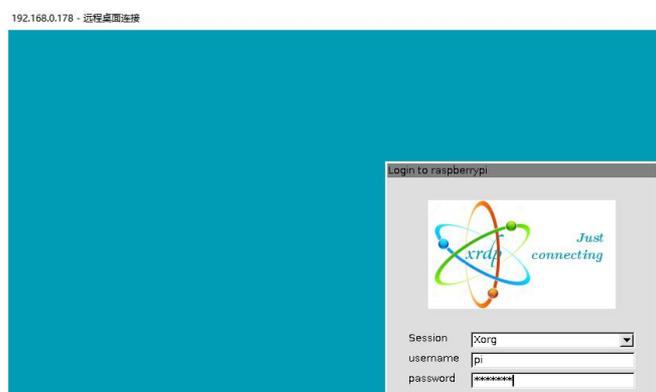
3.2. 源码介绍

源码程序在默认用户 **pi** 下的桌面上 **gw1302s**, 也可自行下载: <https://gitee.com/rejeee/gw1302s.git>, 参考官方代码, 为了确保编译通过,

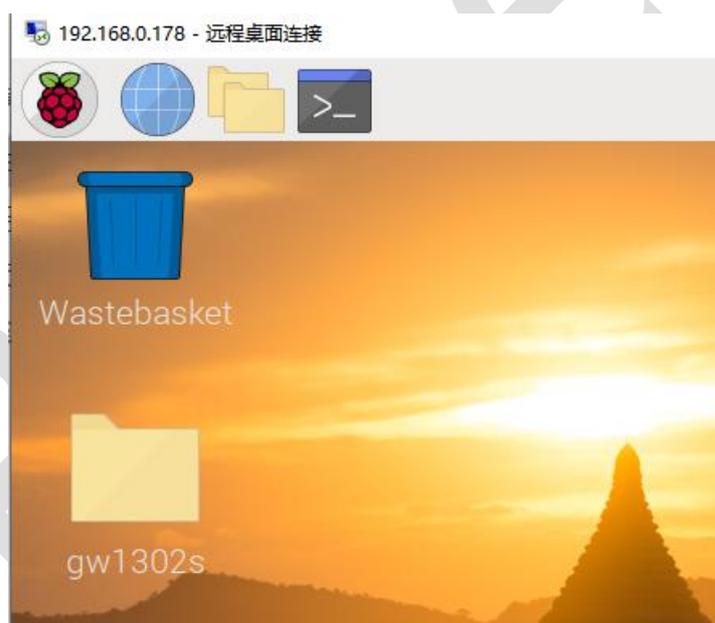
LoRa 网关模组手册

注释了 I2C 接口的温度传感器代码，并将收发测试程序的同步字修改为 private 即 0x12。如需官方源码也可自行到 semtech 网站或 github 进行下载。

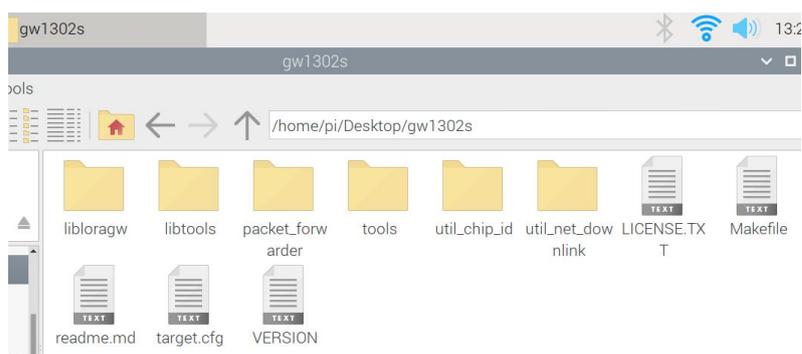
为了图形化显示源码文件目录，远程桌面到树莓派，如下所示。



远程桌面，输入用户名和密码，然后进入系统。显示如下。



打开文件夹，目录如下所示，主要库源码目录为 libloragw。



同样跟 SX1301 一样，在树莓派上启动 SX1302 需要先执行 `reset_lgw.sh` 脚本（在 `tools` 目录下）启动芯片。这一次 SX1302 的官方代码改进了执行动作，将脚本执行动作写入到测试程序中了。因此无需像 SX1301 一样每次手动执行一遍 `reset_lgw.sh` 脚本。只需保证 `reset_lgw.sh` 脚本与可执行测试程序在同一目录即可。

`reset_lgw.sh` 脚本将 SX1302 的复位改为 GPIO7，为了保持与 SX1301 的评估套件相兼容，采用同样的树莓派硬件管脚复位 SX1302。

在 `gw1302s` 目录下，如果修改代码，执行 `make clean`，然后再 `make` 一下即完成所有项编译。也可单独编译各自目录模块。

3.3. 网关射频模组

3.3.1. 寄存器读写测试

如果使用评估套件，需要将 mini PCI-E 的 M-GW1302S 模组扣在评估套件的底板（转接板）上，然后 USB 供电启动，登录系统，进入 `libloragw` 目录，然后执行 `test_loragw_reg`，如下所示。执行提示找不到 `reset_lgw.sh` 脚本，所以首先我们需要将脚本 `copy` 到当前执行程序同目录下再执行 `test_loragw_reg`，过程如下所示，测试完成，显示测试结果，通过则说明硬件连接正常。

```
192.168.0.178 x
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ ls
inc          Makefile    src          test_loragw_counter  test_loragw_hal_tx  test_loragw_spi
libloragw.a  obj         test_loragw_cal    test_loragw_gps      test_loragw_i2c     test_loragw_spi1250
library.cfg  readme.md  test_loragw_capture_ram  test_loragw_hal_rx  test_loragw_reg     tst
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ ./test_loragw_reg
sh: 1: ./reset_lgw.sh: not found
ERROR: failed to reset SX1302, check your reset_lgw.sh script
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ sudo cp ../tools/reset_lgw.sh .
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ ./test_loragw_reg
CoreCell reset through GPIO7...
## TEST#1: read all registers and check default value for non-read-only registers
-----
TEST#1 PASSED
-----

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

CoreCell reset through GPIO7...
pi@raspberrypi:~/Desktop/gw1302s/libloragw$
```

测试命令文本: `./test_loragw_reg`

3.3.2. 发送测试

执行对应的发送测试程序 `test_loragw_hal_tx`，可以通过命令查询参数帮助，根据需要设置不同测试参数。过程如下所示，测试完成，显示测试结果，发送成功。如下调整发射功率，可增加输入参数。示例如：`--pa 1 --pwid 22`。具体参考-h 参数说。测试命令文本如下，所需参数根据帮助说明自行删减。

```
./test_loragw_hal_tx -h
```

```
./test_loragw_hal_tx -r 1250 -f 480.1 -m LORA -b 125 -s 12 -z 20
```

```
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ ./test_loragw_hal_tx -h
Available options:
-h print this help
-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]
-d <int> FSK frequency deviation in khz [1:250]
-q <float> FSK bitrate in kbps [0.5:250]
-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)
-----
--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 MX 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)
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ ./test_loragw_hal_tx -r 1250 -f 480.1 -m LORA -b 125 -s 12 -z 20
Sending 1 LoRa packets on 480100000 Hz (BW 125 kHz, SF 12, CR 1, 20 bytes payload, 8 symbols preamble, explicit header, non-inverted polarity) at 0 dbm
CoreCell reset through GPIO7...
TX done

Nb packets sent: 1 (1)
CoreCell reset through GPIO7...
===== Test End =====
pi@raspberrypi:~/Desktop/gw1302s/libloragw$
```

3.3.3. 接收测试

执行对应的发送测试程序 `test_loragw_hal_rx`，可以通过命令查询参数帮助，根据需要设置不同测试参数。

测试命令文本如下，所需参数根据帮助说明自行删减。

```
./test_loragw_hal_rx -h
```

```
./test_loragw_hal_rx -r 1250 -a 475.5 -b 476.5
```

过程如下所示，当芯片启动成功后，显示：

```
Waiting for packets...
```

如果有数据接收到，则打印显示相关信息，如下所示。

```
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ ./test_loragw_hal_rx -h
Available options:
-h print this help
-k <uint>   Concentrator clock source (Radio A or Radio B) [0..1]
-r <uint>   Radio type (1255, 1257, 1250)
-a <float>  Radio A RX frequency in MHz
-b <float>  Radio B RX frequency in MHz
-o <float>  RSSI offset to be applied in dB
-n <uint>   Number of packet received with CRC OK for each HAL start/stop loop
-z <uint>   Size of the RX packet array to be passed to lgw_receive()
-m <uint>   Channel frequency plan mode [0:LoRAWAN-like, 1:Same frequency for all channels (-400000Hz on RF0)]
-j         set radio in single input mode (SX1250 only)
pi@raspberrypi:~/Desktop/gw1302s/libloragw$ ./test_loragw_hal_rx -r 1250 -a 475.5 -b 476.5
===== sx1302 HAL RX test =====
INFO: rxpkt buffer size is set to 16
INFO: select channel mode 0
CoreCell reset through GPIO7...
waiting for packets...

----- LoRa packet -----
count_us: 3941730
size: 18
chan: 5
status: 0x10
dctr: 12
codr: 1
rf_chain 0
freq_hz 475500000
snr_avg: 6.8
rssi_chan:154.0
rssi_sig :154.0
crc: 0x6A0A
03 02 0B 41 0C 00 2A C6 00 20 1D 04 00 E7 05 21 EA 96
Received 1 packets (total:1)
```

3.4. 总结

总体来说，从运行的情况看，SX1302 的启动比 SX1301 会更加稳定，因为原来 1301 的代码校验相对较多，在某些情况下会出现启动失败的情况，只有调教软件适配自己 CPU 才能运行稳定。另外，可能第 2 代芯片吸取了第 1 代芯片的经验和教训，使内部的 SPI 操作更加稳定。

从源码上看，官方测试代码相对来说操作更加方便。收发测试独立，方便用户进行测试和编程参考。

由于 M-GW1302S 是完全基于 CoreCell 的 SX1302 网关参考而设计的，因此用户可以很方便的采用官方源码进行 LoRa 网关通信设计和开发。同时硬件接口与公司上一代 mini-PCI-E 的 SX1301 网关模组兼容，也方便此类客户进行升级迭代，以便使用更具优势的 LoRa 设计方案。

4. 版本修订历史

日期	版本	发布说明
2020.04	V1.0	初稿发布

Rejeeee