LCM001
Utility User manual
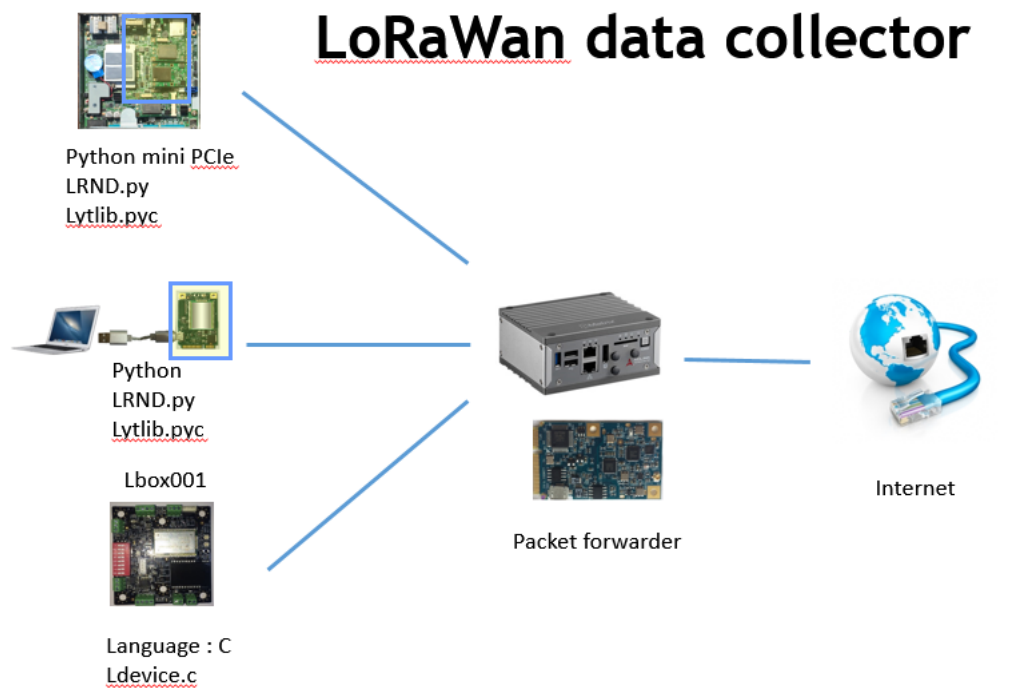
Version: 1.00

Date: 2017/07/13

CONTENT

LoRaWan data collector

Python mini PCIe
LRND.py
Lytlib.pyc

Python
LRND.py
Lytlib.pyc

Lbox001

Language : C
Ldevice.c

Packet forwarder

Internet

Chapter 1: Introduction

The purpose of this document is to set up an LCM001 in PC with Linux or Raspberry PI to forward the packets collected from the LoRaWan sensor nodes. The codes and program description are based on https://github.com/lora-net/ The LCM001 is based on Semtech SX1301. The SX1301 digital baseband chip is a massive digital signal processing engine specifically designed to offer breakthrough gateway capabilities in the ISM bands. The LoRa concentrator is a multi-channel high performance transmitter/receiver designed to simultaneously receive several LoRa packets using random spreading factors on random channels. Its goal is to enable robust connection between a central wireless data concentrator and a massive amount of wireless end-points spread over a very wide range of distances. The SX1301 is targeted at smart metering fixed networks and Internet of Things.
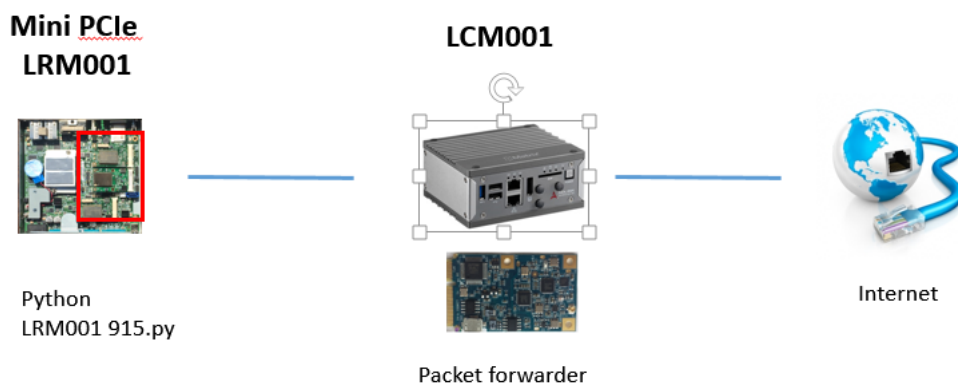
2. The following equipment are needed:

    i. One embedded system – PC or Raspberry Pi

    ii. One PC, 64 bits Windows 7 or Windows 10 with 32 bit Python 3.5.2 installed.

    iii. Internet access

    iv. One LCM001 (same frequency with LRM001 – 868 or 915MHz)

    v. One LRM001 (same frequency with LRM001 – 868 or 915MHz)

4. There following documents / files are for using this kit.

    i. LCM001 user manual.pdf

    ii. LRM001 user manual.pdf

    iii. LRM001 915.py (Python LoRaWan sample code for LRM001)

Chapter 2: Compile the libraries and utilities in lora_gateway

1. Semtech provide the source code of the library to build a gateway based on a Semtech LoRa multi-channel RF receiver (a.k.a. concentrator). Once compiled all the code is contained in the libloragw.a file that will be statically linked (ie. integrated in the final executable). The library also comes with a bunch of basic tests programs that are used to test the different sub-modules of the library.

2. Make sure your embedded system is with GCC and Make utility.

3. There are utilities like libloragw , util_pkt _logger and util_spi_stress

   Those programs are included in the project to provide examples on how to communicate with the packet forwarder, and to help the system builder use it without having to implement a full Lora network server.

   4.1 util_sink

   The packet sink is a simple helper program listening on a single port for UDP datagrams, and displaying a message each time one is received. The content of the datagram itself is ignored.

   4.2 util_ack

   The packet acknowledger is a simple helper progra listening on a single UDP port and responding to PUSH_DATA datagrams with PUSH_ACK, and to PULL_DATA datagrams with PULL_ACK.

   4.3 util_tx_test

   The network packet sender is a simple helper program used to send packets through the gateway-to-server downlink route.

Chapter 3: Setup driver

1. Please make sure that the embedded system is with both GCC and make utilities.

2. "Libmpsse" is the library of the Multi Protocol Synchronous Serial Engine (MPSSE) is generic hardware in FTDI chips that allows the chip to communicate with an SPI device. The LibMPSSE-SPI library has

been created by FTDI to aid the implementation of SPI designs using which incorporate FT232H by taking care of all the required MPSSE commands.

3.   Copy limpsse-1.3.tar.gz to ~ (Directory)

Chapter 4: Compile the utilities in packet_forwarder

4.   Semtech provide the source code of the library to build a gateway based on a Semtech LoRa multi-channel RF receiver (a.k.a. concentrator). The packet forwarder is a program running on the host of a Lora gateway that forwards RF packets receive by the concentrator to a server through a IP/UDP link.. The open source code is in https://github.com/Lora-net/packet_forwarder. The source code in Lora.gz of LCM001 is based on the specific version of the code.

1.   Change directory to *~/Lora915/packet_forwarder/*

```
pi@raspberrypi:~/Lora $ dir
lora_gateway  packet_forward
pi@raspberrypi:~/Lora $
```

2.   Compile the source code, it will creates utilities like basic_pkt_fwd , util_sink and … (Shown in below)

```
pi@raspberrypi:~/Lora/packet_forward/packet_forwarder-2.1.0 $ dir
basic_pkt_fwd    gps_pkt_fwd  PROTOCOL.TXT     util_ack     VERSION
beacon_pkt_fwd   LICENSE      readme.md        util_sink
cfg              Makefile     reset_pkt_fwd.sh util_tx_test
```

Chapter 5: Execute the utility in packet_forwarder

1. Running Basic_pkt_fwd:

Change working directory to *~/Lora915/packet_forwarder/ packet_forwarder-2.1.0/basic_pkt_fwd*

Run the basic_pkt_fwd. The basic packet forwarder is a program running on the host of a Lora Gateway that forward RF packets receive by the concentrator to a server through an IP/UDP link. This program is statically linked with the libloragw Lora concentrator library.

The last line of running the "./basic_pkt_fwd" will show "INFO: [main] concentrator started, packet can now be received." If it failed, please make sure the LCM001 is attached to the host. Or you need to re-install the driver again.

It will start receiving the packet from the LoRaWan node. If there is no LoRaWan, you will see there is "RF packet received by concentrator: 0" (Shown below).

The way the program takes configuration files into account is the following:
 * if there is a debug_conf.json parse it, others are ignored
 * if there is a global_conf.json parse it, look for the next file
 * if there is a local_conf.json parse it
If some parameters are defined in both global and local configuration files, the local definition overwrites the global definition.

The global configuration file should be exactly the same throughout your network, contain all global parameters (parameters for "sensor" radio channels) and preferably default "safe" values for parameters that are specific for each gateway (e.g. specify a default MAC address).

The program display statistics on the RF packets received and sent, and the network datagrams received and sent. This basic variant of the packet forwarder does not send status report to the server.

```
##### 2017-07-06 09:56:48 GMT #####
### [UPSTREAM] ###
# RF packets received by concentrator: 0
# CRC_OK: 0.00%, CRC_FAIL: 0.00%, NO_CRC: 0.00%
# RF packets forwarded: 0 (0 bytes)
# PUSH_DATA datagrams sent: 0 (0 bytes)
# PUSH_DATA acknowledged: 0.00%
### [DOWNSTREAM] ###
# PULL_DATA sent: 3 (0.00% acknowledged)
# PULL_RESP(onse) datagrams received: 0 (0 bytes)
# RF packets sent to concentrator: 0 (0 bytes)
# TX errors: 0
##### END #####

##### 2017-07-06 09:57:18 GMT #####
### [UPSTREAM] ###
# RF packets received by concentrator: 0
# CRC_OK: 0.00%, CRC_FAIL: 0.00%, NO_CRC: 0.00%
# RF packets forwarded: 0 (0 bytes)
# PUSH_DATA datagrams sent: 0 (0 bytes)
# PUSH_DATA acknowledged: 0.00%
### [DOWNSTREAM] ###
# PULL_DATA sent: 3 (0.00% acknowledged)
# PULL_RESP(onse) datagrams received: 0 (0 bytes)
# RF packets sent to concentrator: 0 (0 bytes)
# TX errors: 0
##### END #####
```

Liyatech LCM001 user manual...doc

If there is LoRaWan node which send the packet you will see:

"RF packet received by concentrator: 3" (not necessary 3, maybe 2 or other integer)

```
##### 2017-07-06 10:03:48 GMT #####
### [UPSTREAM] ###
# RF packets received by concentrator: 3
# CRC_OK: 33.33%, CRC_FAIL: 66.67%, NO_CRC: 0.00%
# RF packets forwarded: 1 (16 bytes)
# PUSH_DATA datagrams sent: 1 (237 bytes)
# PUSH_DATA acknowledged: 0.00%
### [DOWNSTREAM] ###
# PULL_DATA sent: 3 (0.00% acknowledged)
# PULL_RESP(onse) datagrams received: 0 (0 bytes)
# RF packets sent to concentrator: 0 (0 bytes)
# TX errors: 0
##### END #####
```

2. Running util_sink:

Change directory to *~/Lora915/packet_forwarder/ packet_forwarder-2.1.0/util_sink*

Run the $./util_sink 1680. You will see "Got packet from host 127.0.0.1 port 34924, 12 bytes long" or "237 bytes long". 12 bytes long means the pull data packet. The 237 bytes long is the real packet from LoRaWan node – like LRM001-915.

```
pi@raspberrypi:~/Lora/packet_forward/packet_forwarder
-2.1.0/util_sink $ ./util_sink 1680
INFO: util_sink listening on port 1680

Got packet from host 127.0.0.1 port 34924, 12 bytes l
ong
Got packet from host 127.0.0.1 port 34924, 12 bytes long
Got packet from host 127.0.0.1 port 34924, 12 bytes long
Got packet from host 127.0.0.1 port 34924, 12 bytes long
Got packet from host 127.0.0.1 port 41784, 237 bytes long
Got packet from host 127.0.0.1 port 34924, 12 bytes long
```

Chapter 6: Send packet from LoRaWan node (LRM001)

A. Attach LRM001 to a PC and running "LRM001 915.py" with LRM001, you will see the result below. (If you are not familiar with how to use LRM001, please read LRM001 user manual.)

$python "LRM001 915.py"

```
RESTART: C:\Users\Robert Apple\Dropbox\Product\sucessful code\LRM001 915 LoRaWa
n success.py
Device Name : COM100
Device Opening : True
b'RN2903 0.9.5 Sep 02 2015 17:19:55\r\n'
sys get hweui
 radio set pwr  b'0004A30B001A5B98\r\n'
radio set pwr  b'ok\r\n'
excute mac resume
 b'ok\r\n'
b'' b''
---- init ok ----
 b'ok\r\n' b''
b'ok\r\n' b''
b'ok\r\n' b''
b'mac save' b'ok\r\n'
28
mac set devaddr = 0x177B3534 b'ok\r\n' b''
mac set nwkskey 28AED22B7E1516A609CFABF715884F3C b'ok\r\n' b''
mac set appskey 1628AE2B7E15D2A6ABF7CF4F3C158809 b'ok\r\n' b''
mac set deveui 177B3534171A803B b'ok\r\n' b''
b'mac save' b'ok\r\n' b''
mac set dr = 5, 125KHz b'invalid_param\r\n' b''
mac join abp b'ok\r\n' b'accepted\r\n'
```

Chapter 7: Summary

Thanks for your patience in using our products. If you had any question, please contact the distributor or info@liyatech.com .

Appendix : FCC Statement

Federal Communication Commission Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.

- Increase the separation between the equipment and receiver.

- Connect the equipment into an outlet on a circuit different from that

to which the receiver is connected.

- Consult the dealer or an experienced radio/TV technician for help.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

IMPORTANT NOTE:

Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Country Code selection feature to be disabled for products marketed to the US/CANADA

This device is intended only for OEM integrators under the following conditions:

1) The antenna must be installed such that 20 cm is maintained between the antenna and users, and
2) The transmitter module may not be co-located with any other transmitter or antenna,

As long as 2 conditions above are met, further <u>transmitter</u> test will not be required. However, the OEM integrator is still responsible for testing their end-product for any additional compliance requirements required with this module installed

IMPORTANT NOTE

Liyatech LCM001 user manual...doc

In the event that these conditions can not be met (for example certain laptop configurations or co-location with another transmitter), then the FCC authorization is no longer considered valid and the FCC ID can not be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

End Product Labeling

This transmitter module is authorized only for use in device where the antenna may be installed such that 20 cm may be maintained between the antenna and users. The final end product must be labeled in a visible area with the following: "Contains FCC ID: 2ALXW-LCM001915010".

Manual Information to the End User

The OEM integrator has to be aware not to provide information to the end user regarding how to install or remove this RF module in the user's manual of the end product which integrates this module.

The end user manual shall include all required regulatory information/warning as show in this manual.