Pi Zero LoRa Configuration

Setup per:

<https://circuitdigest.com/microcontroller-projects/raspberry-pi-with-lora-peer-to-peer-communication-with-arduino>

See also:

<https://pypi.org/project/pyLoRa/#files>  
<https://store.uputronics.com/index.php?route=product/product&product_id=99>

By way of explanation for some of the comments below, I generally work on a Mac, running SSH to work on the Raspberry Pi. In this process, I generally do any file editing on the Mac (BBEdit) and transfer files between the Mac and Pi as required (via SFTP using Transmit).

Installation notes:

Enable SPI:

sudo raspi-config  
 [5 Interfacing Options -> P4 SPI (Enable SPI)]

Install pip:

sudo apt install python3-pip -y

There were a couple of problems reported during execution of this commend, with the recommendation (provided in the instructions) to execute the following:

sudo apt-get update

Still had trouble, execute following:

sudo apt install python-pip python3-pip

Still had trouble, execute following:

sudo apt install python-pip python3-pip —-fix-missing

Seemed to be OK at this point…

pip install RPi.GPIO

Returned:

Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/…  
Requirement already satisfied: RPi.GPIO in /usr/lib/python2.7/dist-pac…

Install spidev:

pip install spidev  
 [successfully installed spider-3.4]

Install the pyLoRa package:

pip install pyLoRa  
 [successfully installed pyLoRa-0.3.1]

Not sure what was going on there, because the next thing we’re instructed to do is effectively download all that stuff again…

sudo apt-get install python-rpi.gpio python3-rpi.gpio  
sudo apt-get install python-spidev python3-spidev

A couple more bits and pieces…

sudo apt-get install git  
sudo git clone https://github.com/rpsreal/pySX127x

Note that there is now an SX127x folder in the pi home folder.

Use SFTP (via Transmit) to transfer files (downloaded from the CircuitDigest website) from Mac to Pi—transfer the downloaded .zip file contents from the Mac to the pi user home directory.

Modify LoRa.py to reflect local configuration:

Freq: 917MHz

This didn’t seem to have the desired effect, so the actual Python script (LORA\_PI\_RX.py) was modified to explicitly set the frequency:

lora.set\_freq(917)

I did, however, subsequently go through the LoRa.py file and replace any occurrence of 434 with 917.

Everything else seems to be OK with the default settings, except for the sync word, which was explicitly set to 0x12 in LoRa.py:

@setter(REG.LORA.SYNC\_WORD)  
def set\_sync\_word(self, sync\_word=0x12):  
 return sync\_word

This setting was subsequently moved to the main Python script (LORA\_PI\_RX.py):

lora.set\_sync\_word(0x12)

I also had to fiddle around a bit (because I’d never previously worked with Python) to print the output as a hex string rather than whatever it was originally (LORA\_PI\_RX.py):

print(bytes(payload).hex())

This operation was subsequently enhanced considerably by adding a new method (print\_packet(self, payload)) to break out the packet content.

To match the current hardware configuration, which is a little different to the original, I had to modify the board\_config.py file to specify the new pin configuration[[1]](#footnote-1):

DIO0 = 25 # RaspPi GPIO 25  
DIO1 = 24 # RaspPi GPIO 24  
DIO2 = 23 # RaspPi GPIO 23  
DIO3 = 27 # RaspPi GPIO 27  
RST = 22 # RaspPi GPIO 22  
LED = 13 # RaspPi GPIO 13

set the low\_band parameter to reflect the fact that we’re operating at 917.0MHz:

low\_band = False

and remove the second board definition (just to be tidy, because we don’t have a second board).

The second board configuration details were also removed from the lora.py file.

1. This configuration was chosen for two reasons. First, it clustered the connections together on the Raspberry Pi, making it easier to build a ‘hat’ for the job, and second, to make the configuration compatible with that described by Charles-Henri Hallard (https://github.com/hallard/LoRasPI), which is reportedly supported by the default LMIC setup (which I might get around to testing one day, because, as I understand, the LMIC library is available for all the platforms I am using). [↑](#footnote-ref-1)