<https://github.com/raspberrypi-tw/lora-sx1278/commit/4d3b539e39a775c4e0d28201f6fa7cefb9626e8d>

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

sudo apt-get install python-spidev python3-spidev

sudo apt-get install git

sudo git clone https://github.com/rpsreal/pySX127x

If it is necessary to run the library from anywhere:

nano ~/.bashrc

Put this at the end of the file:

*export PYTHONPATH=/home/pi/pySX127x/*

source ~/.bashrc

from SX127x.LoRa import \*

from SX127x.board\_config import BOARD

BOARD.setup()

lora = LoRa()

lora.set\_mode(MODE.STDBY)

print lora.get\_version() # this prints the sx127x chip version

print lora.get\_freq() # this prints the frequency setting

lora.set\_freq(433.0) # Set the frequency to 433 MHz

class LoRaRcvCont(LoRa):

def \_\_init\_\_(self, verbose=False):

super(LoRaRcvCont, self).\_\_init\_\_(verbose)

self.set\_mode(MODE.SLEEP)

self.\_id = "GW\_01"

def on\_rx\_done(self):

print '----------------------------------'

self.clear\_irq\_flags(RxDone=1)

payload = self.read\_payload(nocheck=True)

data = ''.join([chr(c) for c in payload])

print "Time:", str(time.ctime())

print "Rawinput:", payload

try:

\_length, \_data = packer.Unpack\_Str(data)

print "Time:", str(time.ctime())

print "Length:", \_length

print "Receive:", \_data

except:

print "Non-hexadecimal digit found..."

print "Receive:", data

#sleep(0.5)

#self.set\_mode(MODE.SLEEP)

#self.reset\_ptr\_rx()

#sleep(1)

for i in range(3):

self.set\_mode(MODE.STDBY)

self.clear\_irq\_flags(TxDone=1)

data = {"id":self.\_id, "data":packer.ACK}

\_length, \_ack = packer.Pack\_Str( json.dumps(data) )

ack = [int(hex(ord(c)), 0) for c in \_ack]

print "ACK:", self.\_id

self.write\_payload(ack)

self.set\_mode(MODE.TX)

# ALOHA(1~3)

t = i\*2 + np.random.random() \* 3

sleep(t)

self.set\_mode(MODE.RXCONT)

def start(self):

print 'start to receive...'

self.reset\_ptr\_rx()

self.set\_mode(MODE.RXCONT)

while True:

sleep(.5)

#

# initialize sx1278

#

BOARD.setup()

lora = LoRaRcvCont()

lora.set\_mode(MODE.STDBY)

lora.set\_pa\_config(pa\_select=1)

lora.set\_freq(433)

lora.set\_spreading\_factor(7) # 7-12

lora.set\_bw(7) # 0-9

lora.set\_coding\_rate(1) # 1-4

lora.clear\_irq\_flags(RxDone=1)

print(lora)

try:

lora.start()

finally:

lora.set\_mode(MODE.SLEEP)

BOARD.teardown()

print "exit()"