Internet of Things and Smart Systems

LoRa

Asst. Prof. Dr.Panarat Cherntanomwong
Asst. Prof. Dr.Sumek Wisayataksin

Faculty of Engineering
King's Mongkut Institute of Technology Ladkrabang

Long Range (LoRa)

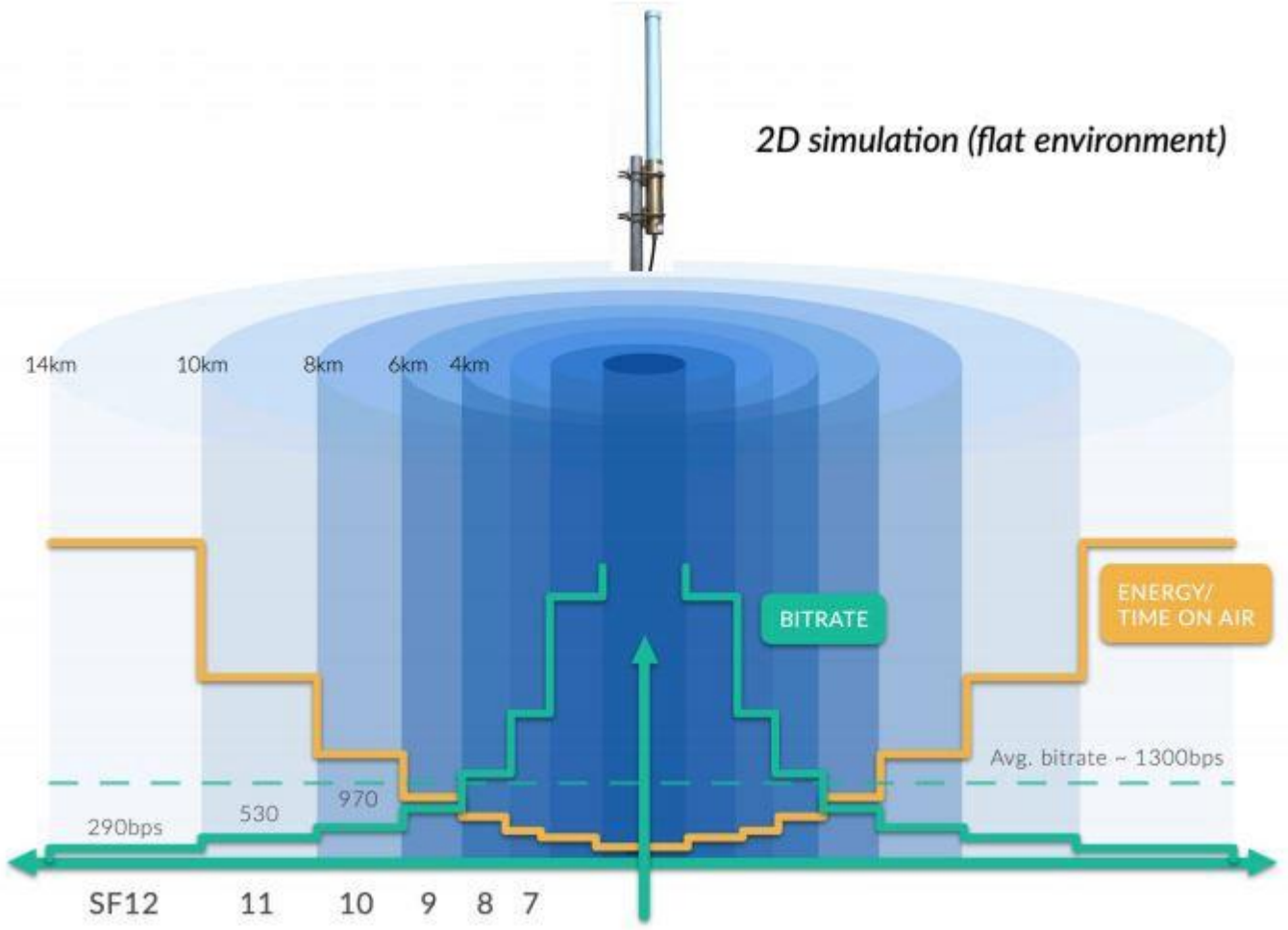
- Wireless modulation technology.
- Physical layer for long range communications.
- Low bandwidth.
- Low battery usage.
- Operates in the license-free ISM bands all around the world
  - 443, 868, 915 MHz

LoRa Characteristic

KMITL
พระจอมเกล้าลาดกระบัง

Characteristic	Target Value for LPWAN Technologies
Long range	5 – 40km in the open field
Ultra low power	Battery lifetime of 10 years
Throughput	Depends on the application, but typically a few hundred bit / s or less (15 seconds per 12-20 bytes package)
Radio chipset costs	$2 or less
Radio subscription costs	$1 per device and year
Transmission latency	Not a primary requirement for LPWAN. IoT applications are typically insensitive to latency.
Required number of base stations for coverage	Very low. LPWAN base stations are able to serve thousands of devices.
Geographic coverage, penetration	Excellent coverage also in remote and rural areas. Good in-building and in-ground penetration (e.g. for reading power meters).

Spreading Factor (SF)



Spreading Factor (SF)

$$\text{SNR} = 10log\left(\frac{S}{N}\right)$$

Spreading Factor	SNR limit	Time-on-air (10 byte packet)	Bitrate
7	-7.5	56 ms	5469 bps
8	-10	103 ms	3125 bps
9	-12.5	205 ms	1758 bps
10	-15	371 ms	977 bps
11	-17.5	741 ms	537 bps
12	-20	1483 ms	293 bps

LoRa Module (SX1278)

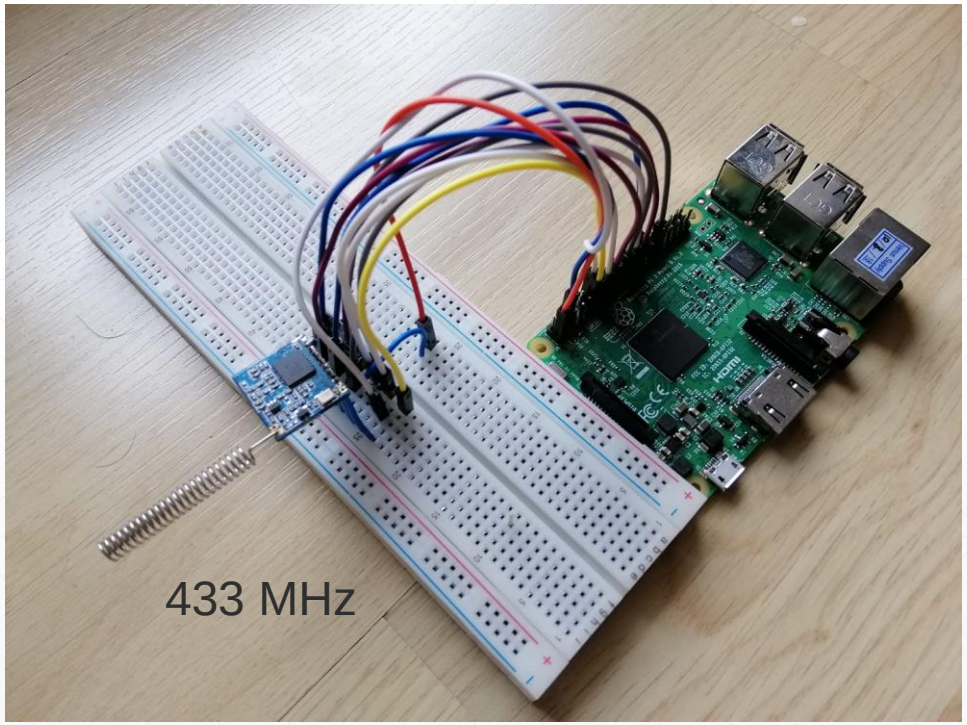
Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate
SX1276	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps
SX1277	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps
SX1278	137 - 525 MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps
SX1279	137 - 960MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps

Semtech SX1278

137MHz to 525MHz Long Range Low Power Transceiver

The SX1276/77/78/79 transceivers feature the LoRa® long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption.

Using Semtech's patented LoRa modulation technique SX1276/77/78/79 can achieve a sensitivity of over -148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20dBm power amplifier yields industry leading link budget making it optimal for any application requiring range or robustness. LoRa also provides significant advantages in both blocking and selectivity over conventional modulation techniques, solving the traditional design compromise between range, interference immunity and energy consumption.



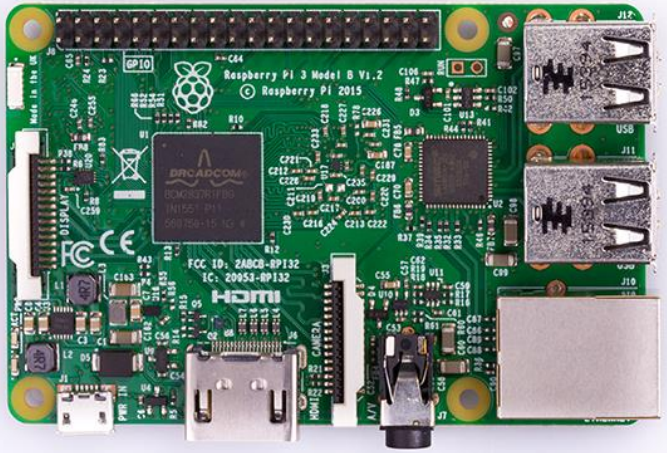
Lab 5.1 : LoRa Transmit and Receive

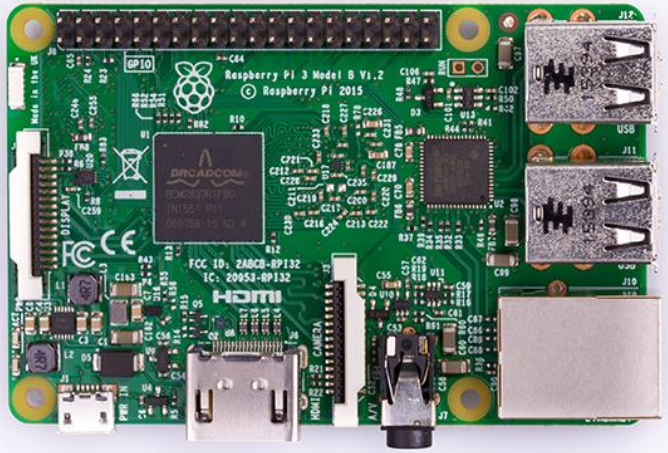
“Hello from LoRa1”

“Acknowledge from LoRa2”

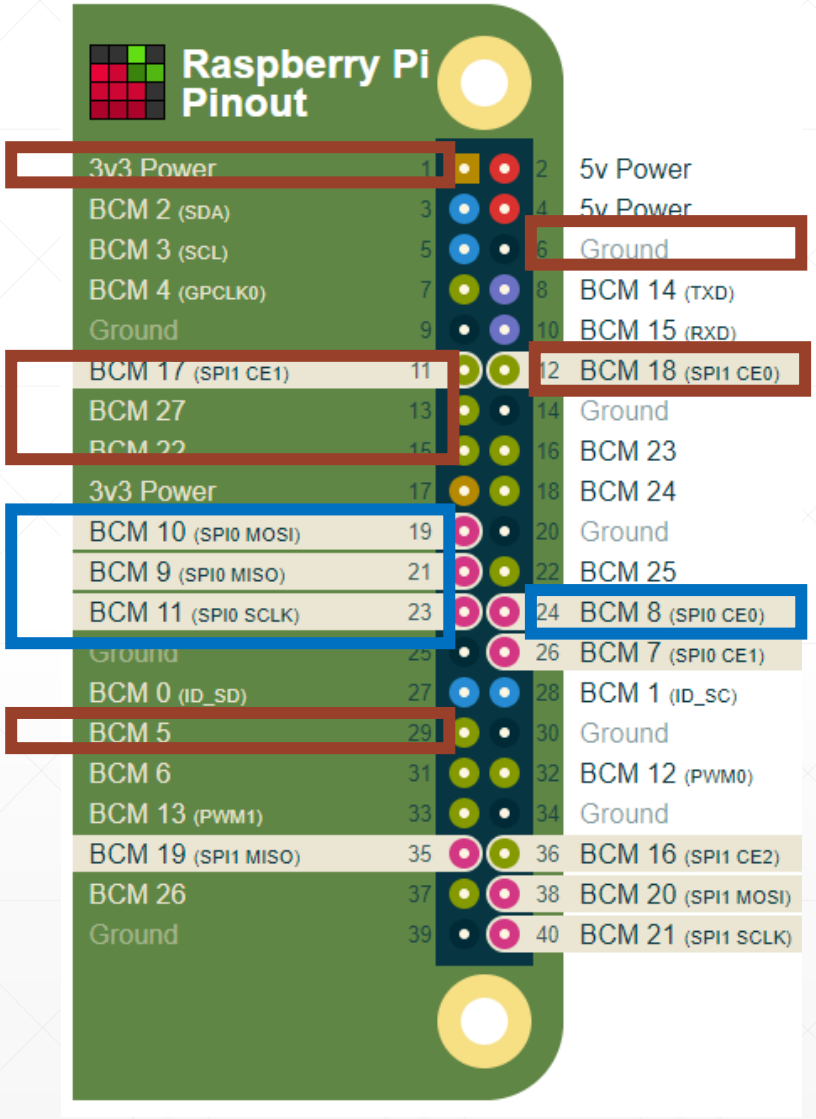
LoRa1

LoRa2

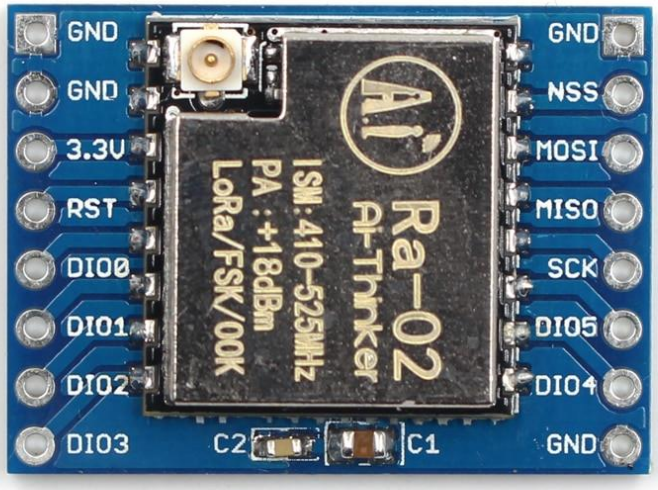




Lab 5.1 : LoRa Transmit and Receive



PIN 6 -> GND
PIN 1 -> 3.3V
PIN 15 -> RST
PIN 29 -> DIO0
PIN 11 -> DIO1
PIN 12 -> DIO2
PIN 13 -> DIO3

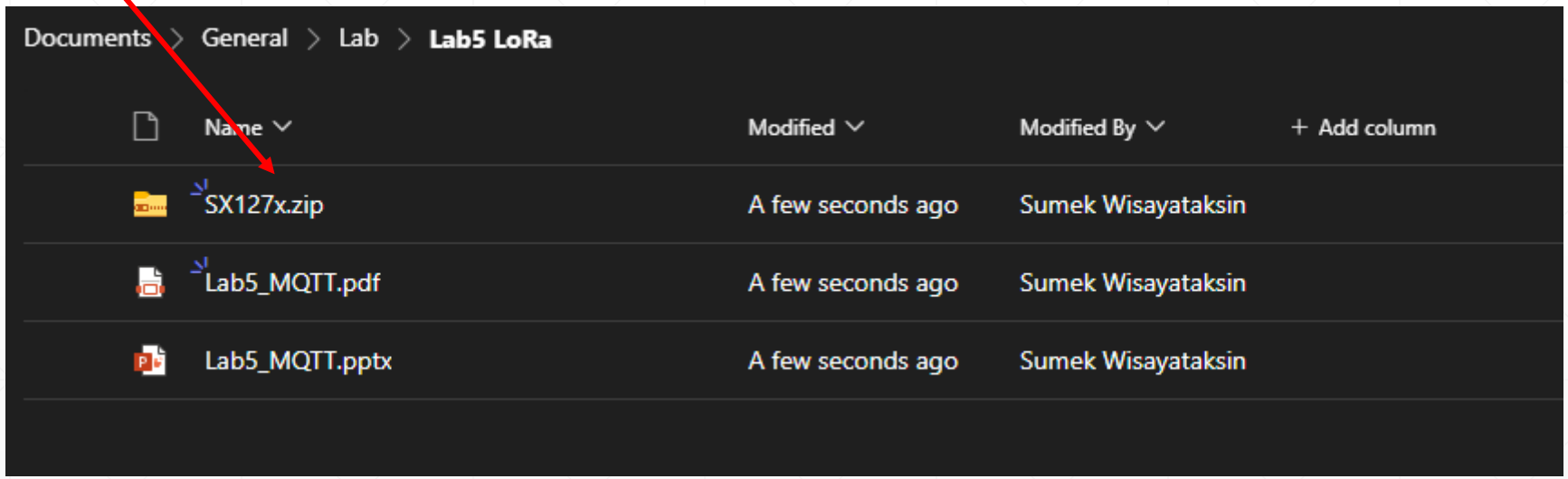


PIN 24 -> NSS
PIN 19 -> MOSI
PIN 21 -> MISO
PIN 23 -> SCK

Totally 11 Pins

Lab 5.1 : LoRa Transmit and Receive

- Download SX127x Library and put in Raspberry Pi
- Unzip next to your python code.



Lab 5.1 : LoRa1 Transmit

KMITL
พระจอมเกล้าลาดกระบัง

```
import time
from SX127x.LoRa import *
from SX127x.board_config import BOARD

BOARD.setup()
BOARD.reset()

#-----------------------------------------------------------------

class mylora(LoRa):

    def __init__(self, verbose=False):
        super(mylora, self).__init__(verbose)
        self.set_mode(MODE.SLEEP)
        self.set_dio_mapping([0] * 6)
        self.val=0

    def on_rx_done(self):

        self.clear_irq_flags(RxDone=1)
        payload = self.read_payload(nocheck=True)
        mens=bytes(payload).decode("utf-8",'ignore')
        mens=mens[2:-1] #to discard \x00\x00 and \x00 at the end
        print("Receive from Lora2 : "+mens+"\n")

        time.sleep(2)

    def on_tx_done(self):

        print("\nTxDone")
        print(self.get_irq_flags())
```

Receive callback

Lab 5.1 : LoRa1 Transmit

Select 420-480

Change to your name

char to bin

```
def start(self):

    while True:

        info = "Hello From LoRa1 No."+str(self.val);
        self.val = self.val+1
        print ("Send to Lora2 : "+info)

        word1 = list(info)
        word2 = []
        for f in word1: word2.append(ord(f))
        data = [0,0]+word2+[0]
        self.write_payload(data) # Send INF
        self.set_mode(MODE.TX)
        time.sleep(3)
        self.reset_ptr_rx()
        self.set_mode(MODE.RXCONT) # Receiver mode
        time.sleep(10)
```

```
#-----------------------------------------------------------------

lora = mylora(verbose=False)

lora.set_pa_config(pa_select=1, max_power=21, output_power=15)
lora.set_freq(433.2)
lora.set_bw(BW.BW250)
lora.set_coding_rate(CODING_RATE.CR4_8)
lora.set_spreading_factor(7)
lora.set_rx_crc(True)
lora.set_low_data_rate_optim(True)

assert(lora.get_agc_auto_on() == 1)

try:
    print("START Lora1 \n")
    lora.start()

except KeyboardInterrupt:
    sys.stdout.flush()
    print("Exit")
    sys.stderr.write("KeyboardInterrupt\n")

finally:
    sys.stdout.flush()
    print("Exit")
    lora.set_mode(MODE.SLEEP)
    BOARD.teardown()
```

Lab 5.1 : LoRa2 Receive

KMITL
พระจอมเกล้าลาดกระบัง

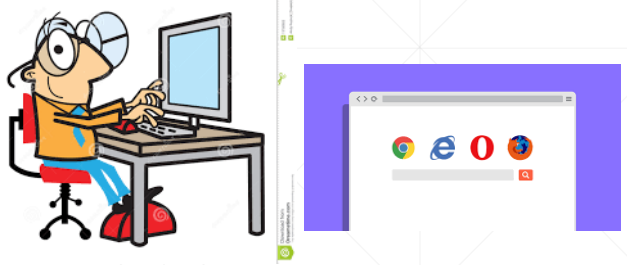
```python
import time
from SX127x.LoRa import *
from SX127x.board_config import BOARD

BOARD.setup()
BOARD.reset()

class mylora(LoRa):
    def __init__(self, verbose=False):
        super(mylora, self).__init__(verbose)
        self.set_mode(MODE.SLEEP)
        self.set_dio_mapping([0] * 6)
        self.val = 0

    def on_rx_done(self):

        self.clear_irq_flags(RxDone=1)
        payload = self.read_payload(nocheck=True )
        mens=bytes(payload).decode("utf-8",'ignore')
        mens=mens[2:-1] #to discard \x00\x00 and \x00 at the end
        print("Receive from Lora1 : "+mens)

        time.sleep(3)

        info  = "Acknowledge From Lora2 No."+str(self.val);
        self.val = self.val+1
        print ("Reply to Lora1 : "+info+"\n")
        word1 = list(info)
        word2 = []
        for f in word1: word2.append(ord(f))
        data = [0,0]+word2+[0]
        self.write_payload(data)
        self.set_mode(MODE.TX)
        time.sleep(3)
        self.reset_ptr_rx()
        self.set_mode(MODE.RXCONT)
```

Change to your name

```python
    def on_tx_done(self):
        print("\nTxDone")
        print(self.get_irq_flags())

    def start(self):
        while True:
            self.reset_ptr_rx()
            self.set_mode(MODE.RXCONT) # Receiver mode
            while True:
                pass;

lora = mylora(verbose=False)
lora.set_pa_config(pa_select=1, max_power=21, output_power=15)
lora.set_freq(433.2)
lora.set_bw(BW.BW250)
lora.set_coding_rate(CODING_RATE.CR4_8)
lora.set_spreading_factor(7)
lora.set_rx_crc(True)
lora.set_low_data_rate_optim(True)

assert(lora.get_agc_auto_on() == 1)

try:
    print("START Lora2 \n")
    lora.start()

except KeyboardInterrupt:
    sys.stdout.flush()
    print("Exit")
    sys.stderr.write("KeyboardInterrupt\n")

finally:
    sys.stdout.flush()
    print("Exit")
    lora.set_mode(MODE.SLEEP)
    BOARD.teardown()
```

Select 420-480 (Same as LoRa1

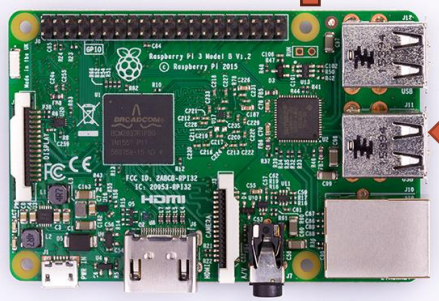
Lab 5.2 : Combine MQTT & LoRa



Web Browser

MQTT server

http://kmitl.ddns.net
Port : 9001
Username : kmitliot
Password : KMITL@iot1234



LoRa2

LoRa1