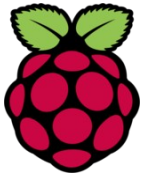
用 Raspberry Pi + LoRa



實做微型物聯網閘道器

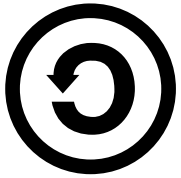
台灣樹莓派 <sosorry@piepie.com.tw>

CC (Creative Commons)

姓名標示 — 非商業性 — 相同方式分享

姓名標示 — 你必須給予 適當表彰、提供指向本授權條款的連結，以及 指出（本作品的原始版本）是否已被變更。你可以任何合理方式為前述表彰，但不得以任何方式暗示授權人為你或你的使用方式背書。

非商業性 — 你不得將本素材進行商業目的之使用。



相同方式分享 — 若你重混、轉換本素材，或依本素材建立新素材，你必須依本素材的授權條款來散布你的貢獻物。

關於我們

- Raspberry Pi 官方經銷商

Raspberry Pi
APPROVED RESELLER

- 專注 Raspberry Pi 應用與推廣，舉辦社群活動

分享 x 教學

- COSCUP,MakerConf,PyCon,HKOSCon 講者
- 投影片
  - https://speakerdeck.com/piepie_tw
- 程式碼
  - https://github.com/piepie-tw

▶ 學習路徑 ◀

Pi選購指南 → Pi設定安裝 → Linux系統管理 / Python程式設計

I/O硬體控制

- GPIO學習套件 (初)
- 感測器學習套件 (基礎/進階) (中)
- 空氣盒子套件 (PiM25) (初)
- Win10開發套件 (初)
- 智慧開關套件 (初)
- Linux Driver 學習套件 (進)

無線/IoT

- RFID/NFC 門禁系統 (初)
- LoRa IoT 閘道器套件 (初)
- 生理資訊 監控IoT(藍牙) (初)
- 毫米波人流/熱點監控 (mmWave) (初)

相機/影像處理

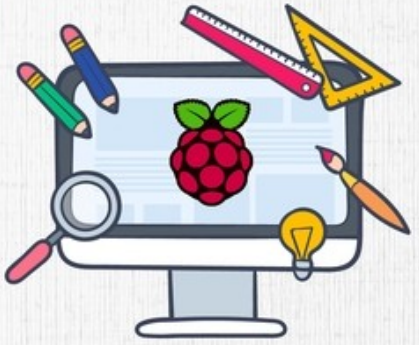
- 特色相機改裝套件 (初)
- 寵物小車套件 (初)
- 自控機器手臂套件 (中)
- 小鴨車套件 (Duckietown) (進)

人工智慧

- 驢車套件 (DonkeyCar) (初)
- AIY Vision Kit (中)
- Intel神經運算棒 (中)
- Google Coral USB加速器 (中)

語音/訊號處理

- 智慧音箱套件 (初)
- AIY Voice Kit (初)



(初) 初階課程　(中) 中階課程　(進) 進階課程

安裝今日所需軟體 (已安裝)

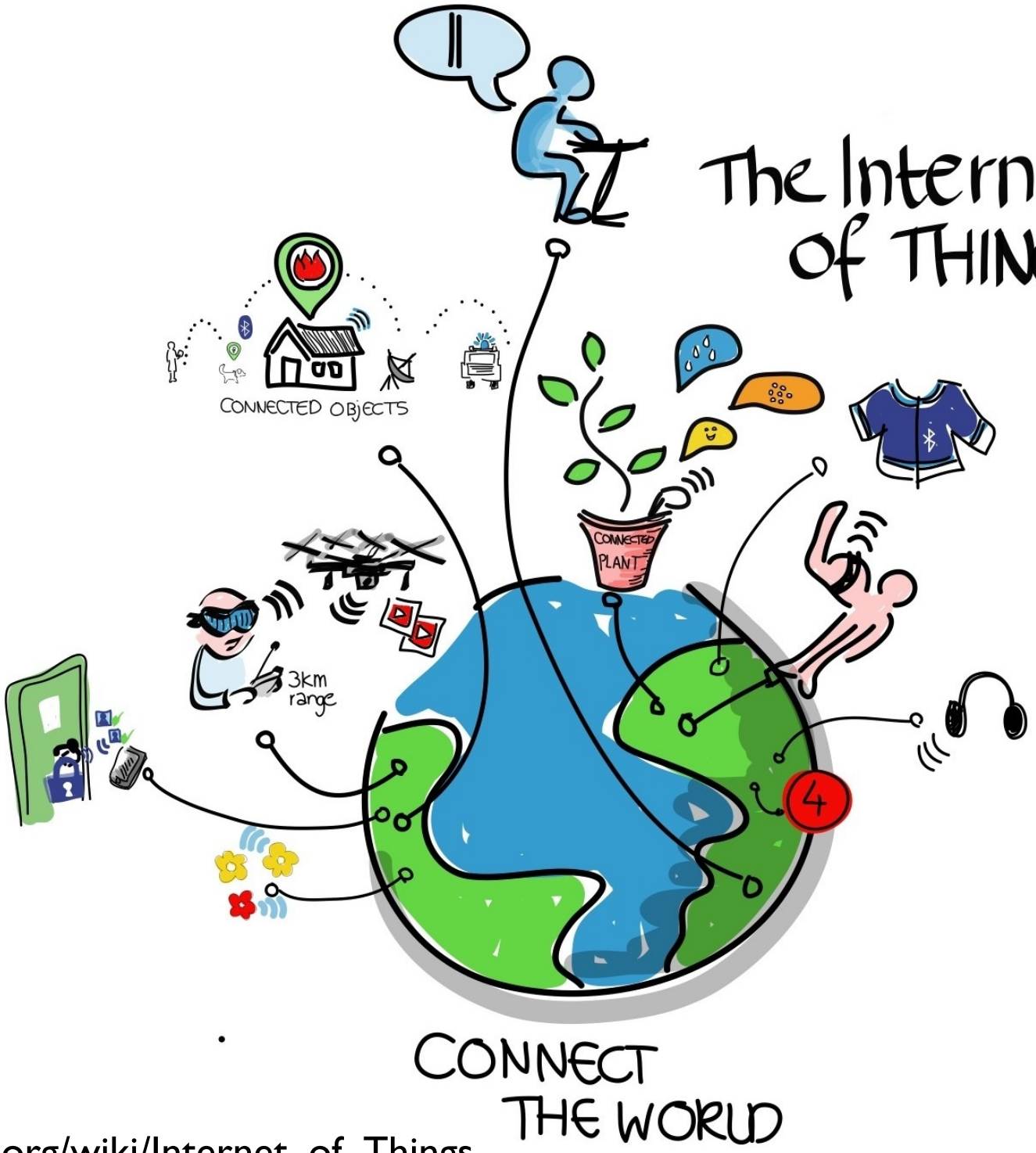
- $ sudo apt-get update
- $ sudo apt-get install -y python-dev python-pip x11vnc mosquitto mosquitto-clients
- $ sudo pip install spidev numpy pyserial paho-mqtt base58 pycrypto
- $ sudo pip3 install paho-mqtt

参考範例程式

- pySX127x
  - https://github.com/mayeranalytics/pySX127x
- LoRaWAN
  - https://github.com/jeroennijhof/LoRaWAN
- iot-lorawan-bme
  - https://github.com/emirez/iot-lorawan-bme
- single_chan_pkt_fwd
  - https://github.com/tftelkamp/single_chan_pkt_fwd
- packet_forwarder
  - https://github.com/Lora-net/packet_forwarder

大綱

- LoRa 簡介
- 控制 SX127x
- 連接 LoRaWAN
- 自製 LoRa 閘道器



http://en.wikipedia.org/wiki/Internet_of_Things

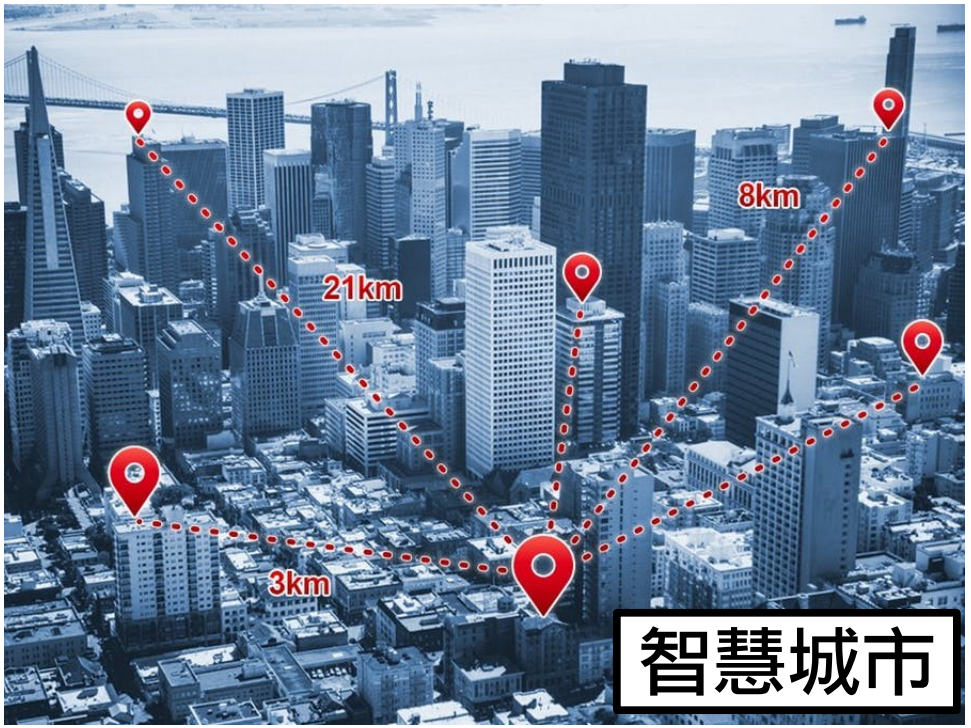
物聯網技術重點之一在”無線傳輸”

這些場景需要哪種無線技術？

農業監控

智慧電錶

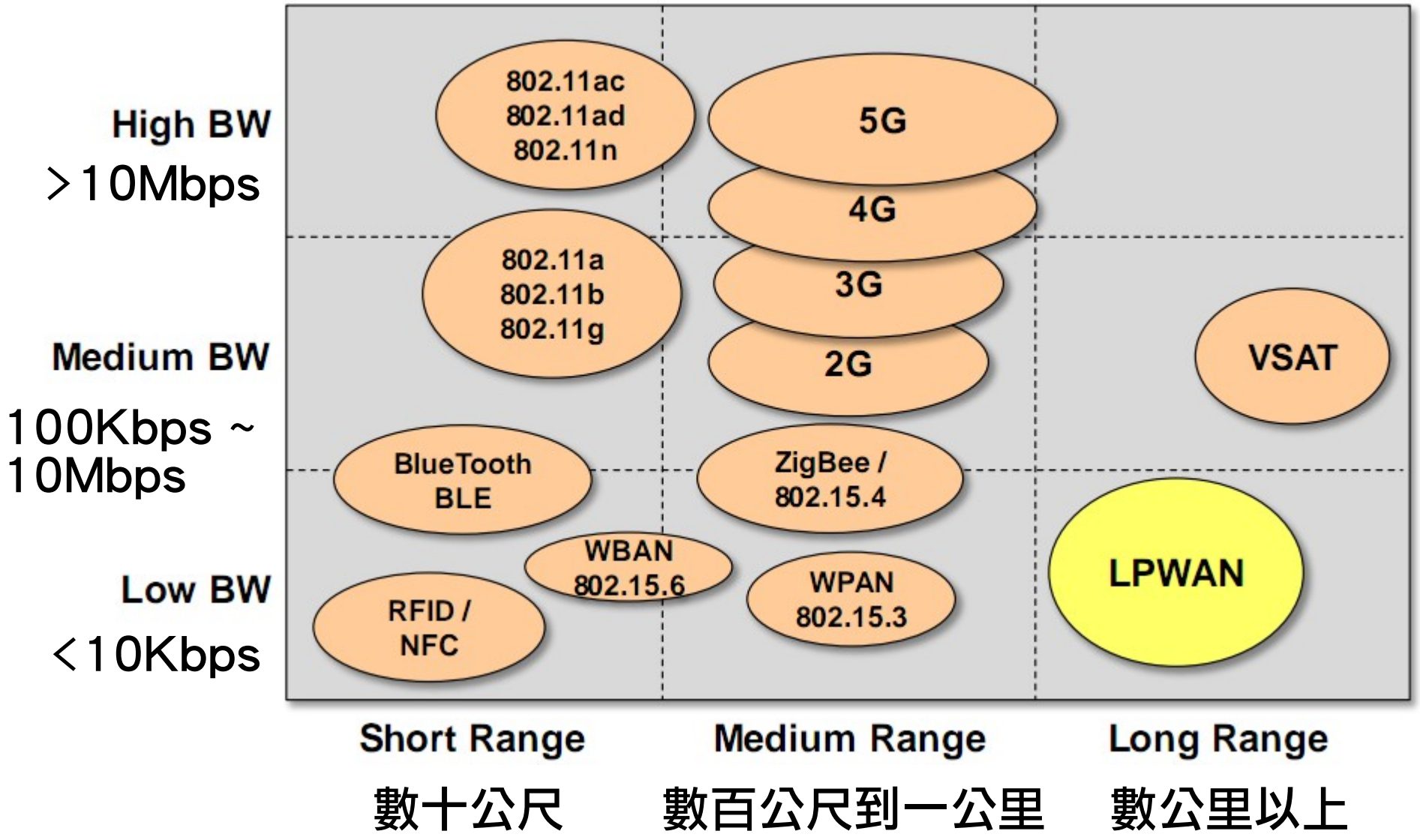
寵物追蹤



智慧城市

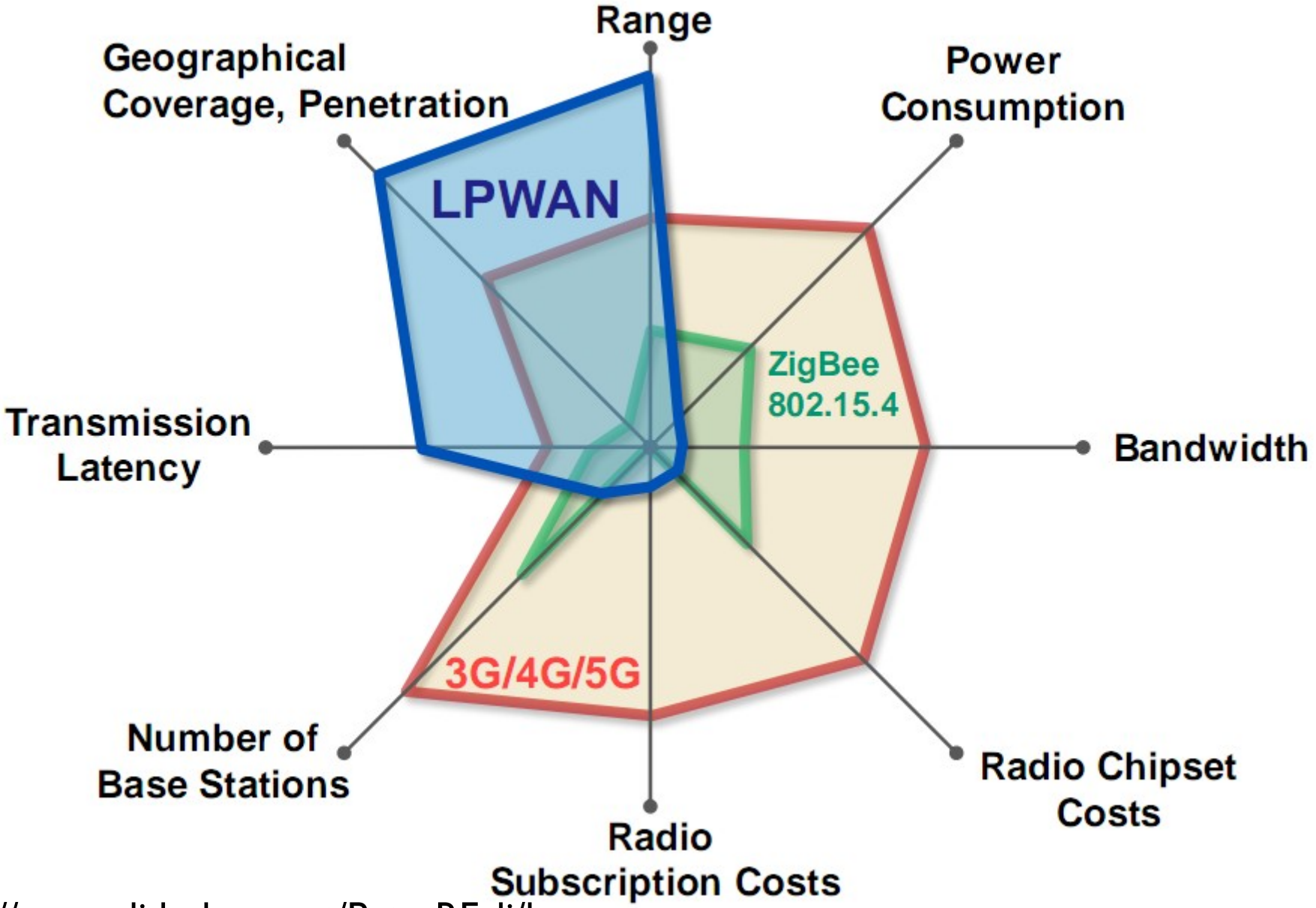
https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRa-Alliance-Whitepaper_NBIoT_vs_LoRa.pdf

速度與距離的考量



https://www.slideshare.net/PeterREgli/lpwan

不同無線傳輸技術的特點



https://www.slideshare.net/PeterREgli/lpwan

最近很夯的低功耗廣域網路

- LPWAN(Low Power Wide Area Network)

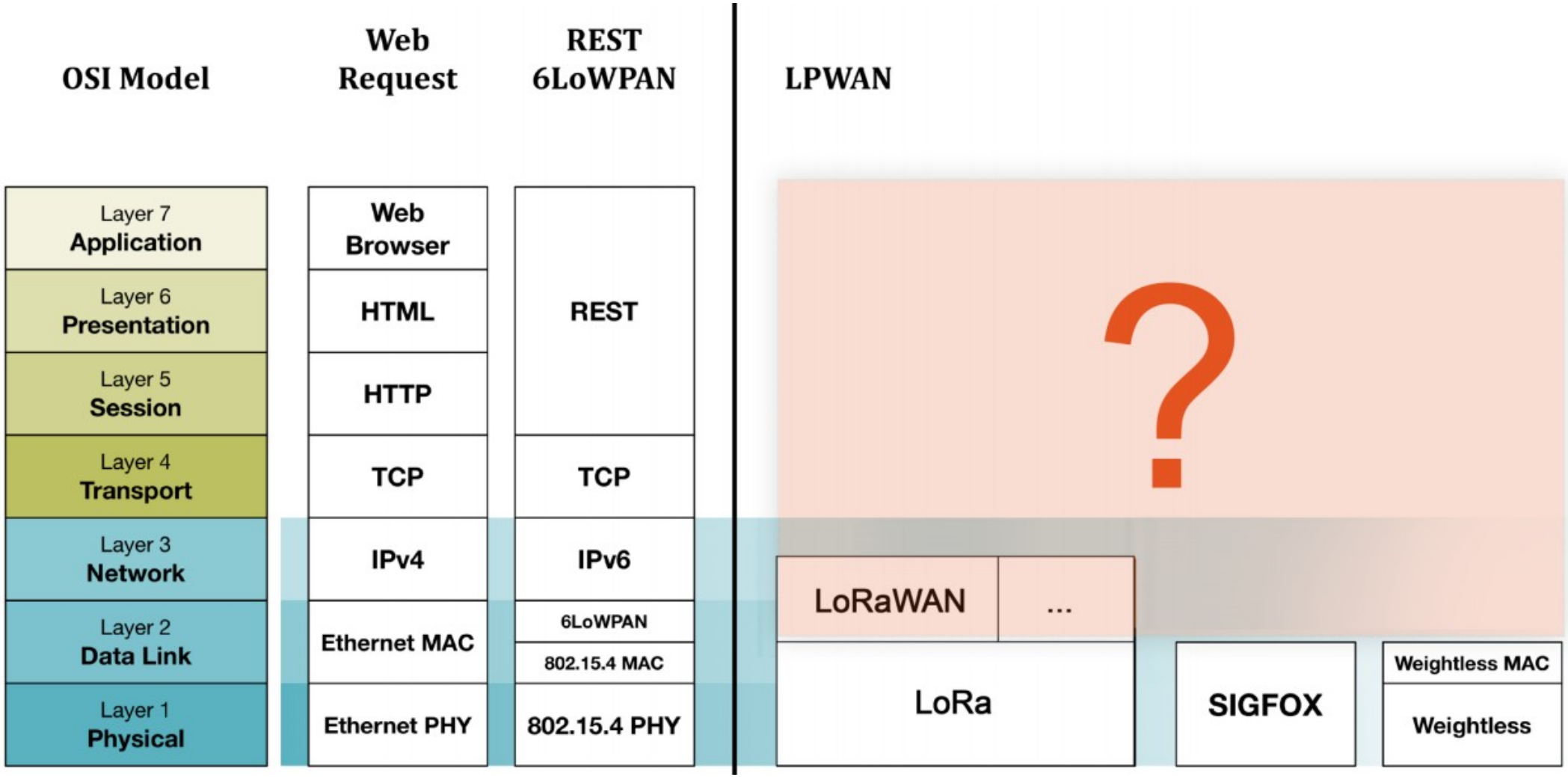
技術協定	主要推動者	成立	目前佈建國家	基站連接數目	使用頻段	傳輸距離	傳輸速度	技術範疇	產業生態系
SIGFOX Company	Sigfox	2009	17	100萬個	ISM Band Sub-1GHz	市區：10KM 郊區：50KM	100bps	終端設備至前端應用	已發展
LoRaWAN Alliance	IBM/Cisco	2015	12	25萬個	ISM Band Sub-1GHz	市區：3~5KM 郊區：15KM	300bps~50kbps	通訊協定	已發展
Weightless SIG	ARM/neul	2015	3	100萬個	ISM Band Sub-1GHz	5KM+(-N) 2KM+(-P)	300bps~100kbps(-N) 100kbps(-P)	通訊協定	尚未發展
RPMA Company	Ingenu	2008	25	20萬個	2.4GHz	4KM	8bps~8kbps	通訊協定及硬體規格	已發展
HaLow Alliance	IEEE	2013	NA	~1萬個	ISM Band Sub-1GHz	1KM	>100kbps	通訊協定	尚未發展
NB-IoT Alliance	3GPP	2016	NA	10萬個	GSM or LTE Band	20KM	~50kbps	通訊協定	尚未發展

http://www.techbang.com/posts/49656

資料傳輸限制

- Sigfox 限制每個裝置每天只能送 140 個 12-byte
- Weightless-N 只能上傳
- LoRa class A 只能在收到訊息後才能開始上傳

從 OSI 模型看 LPWAN



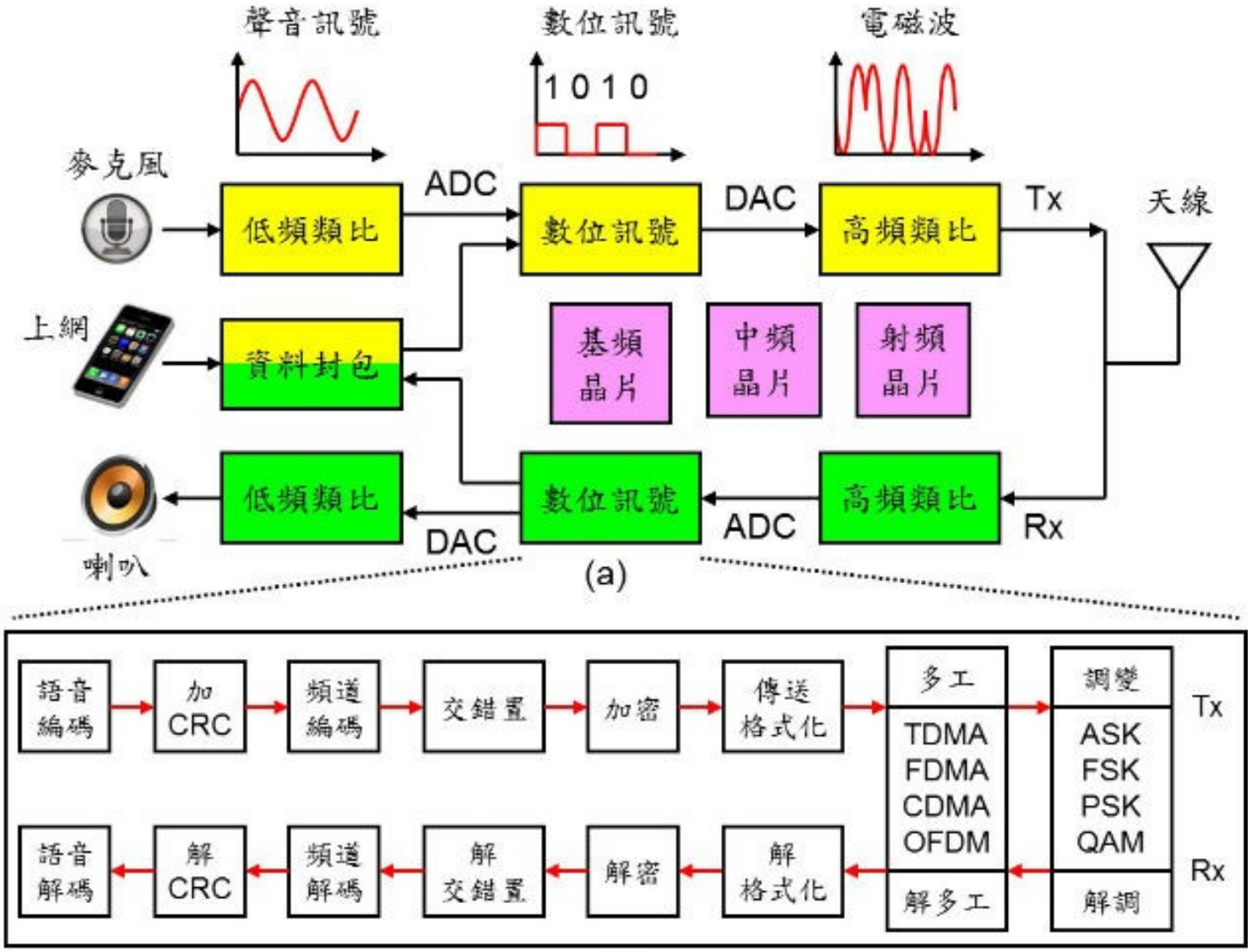
https://www.slideshare.net/infiswift/lpwan-for-iot-62591541

LoRa(Long Range)

- 法國公司 Cycleo 設計，Semtech 在 2012 收購
- LoRa 是實體層（PHY）的調變技術
- 採用 CSS(Chirp Spread Spectrum) 調變技術
- 常用頻段：433/470~510/868/900-925MHz
- 低功耗，低資料率，長距離，低價格
  - 低功耗：RX<10mA, Sleep<200nA
  - 長距離：500m 到 15Km
  - 靈敏度：低於 -137 dBm
  - 資料率：0.3kbps 到 50kbps

通訊系統 101

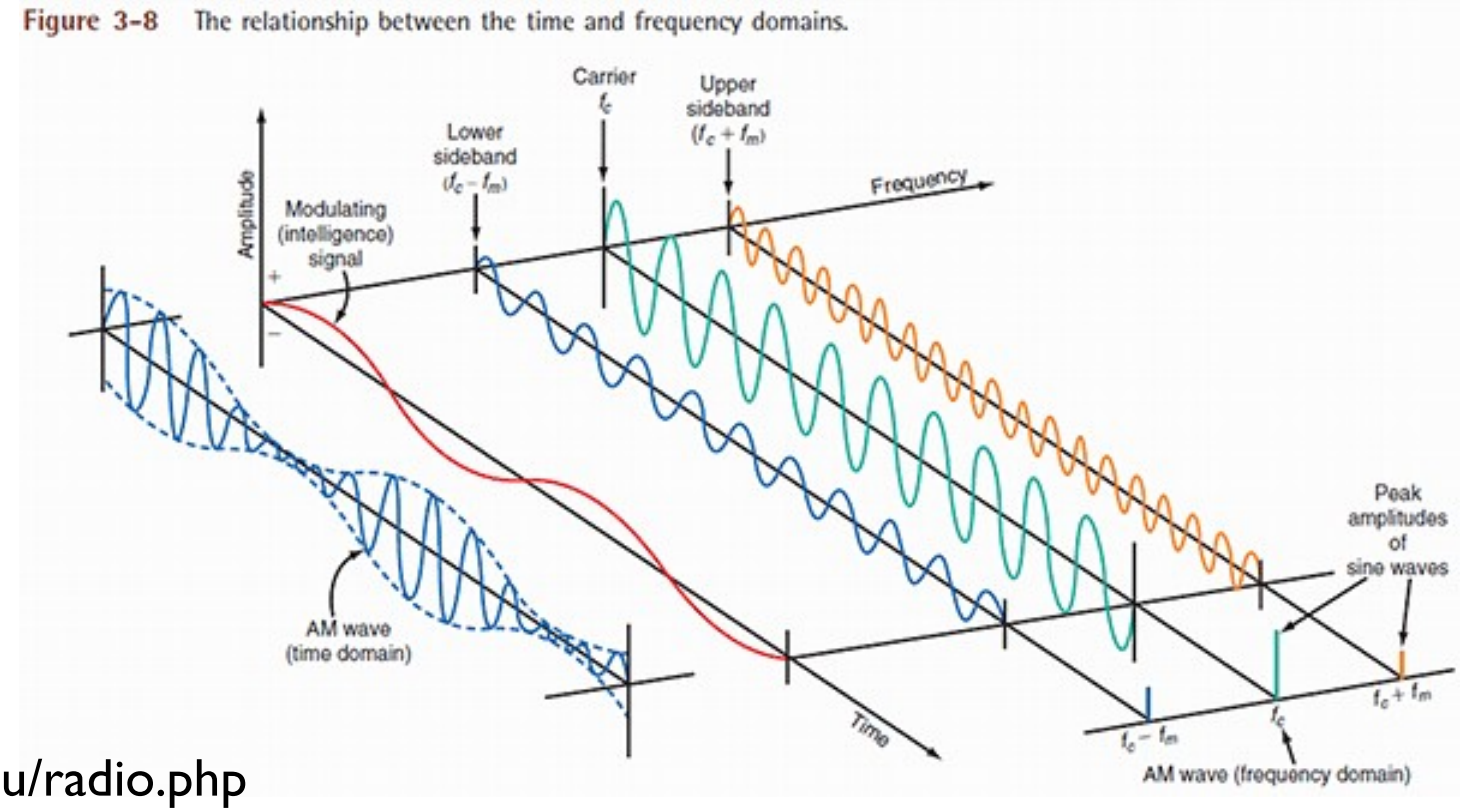
數位通訊系統架構



https://technews.tw/2015/10/22/communicating-element-smart-phone-chip/

射頻電波基礎要素

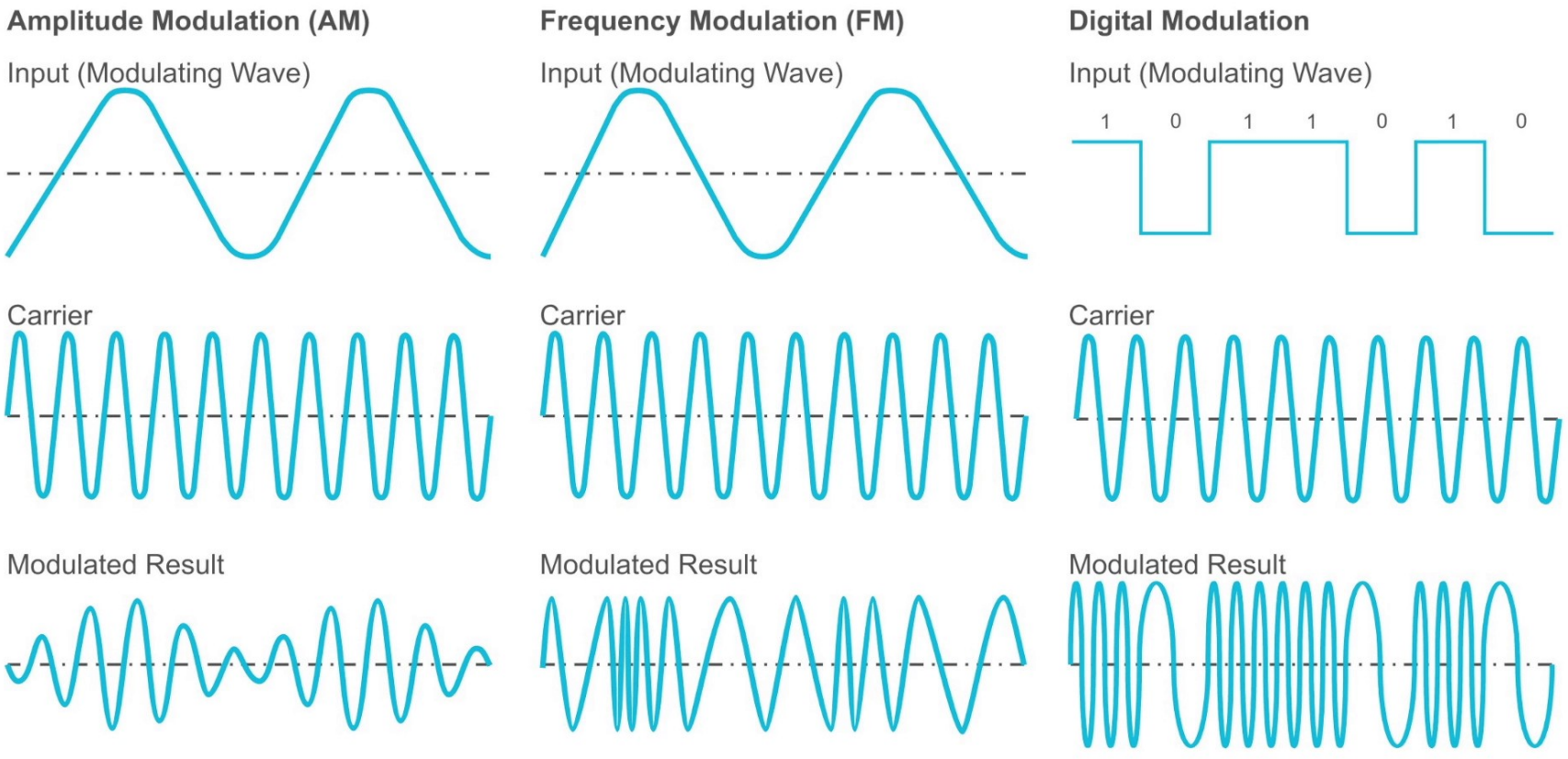
- Amplitude(振幅)
- Frequency(頻率)
- Phase(相位)



https://www.survival.org.au/radio.php

調變 / 解調變

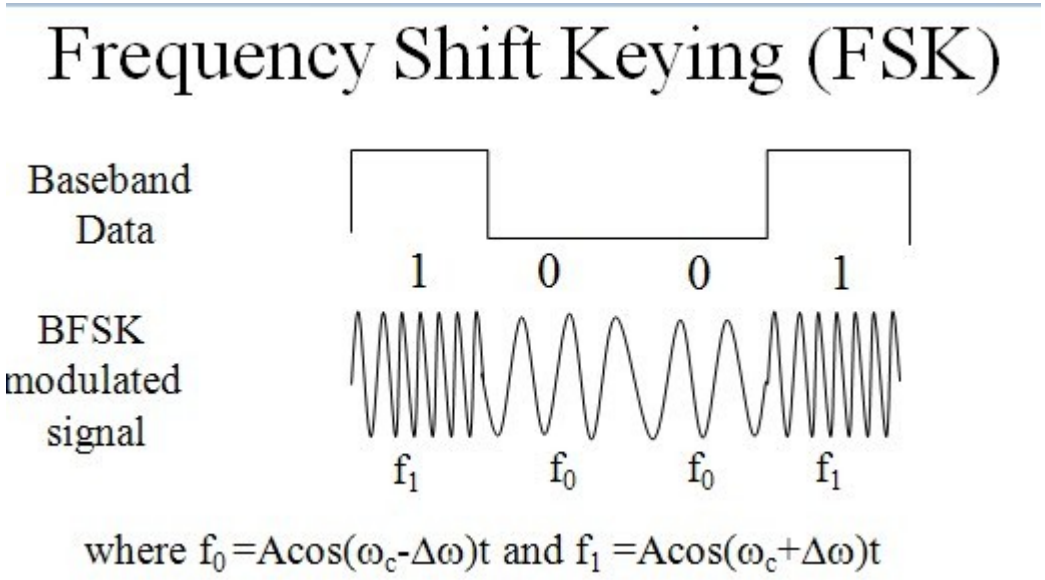
- 調變：將一個或多個週期性的載波混入訊號的技術
- 解調變：調變的逆過程稱為解調變

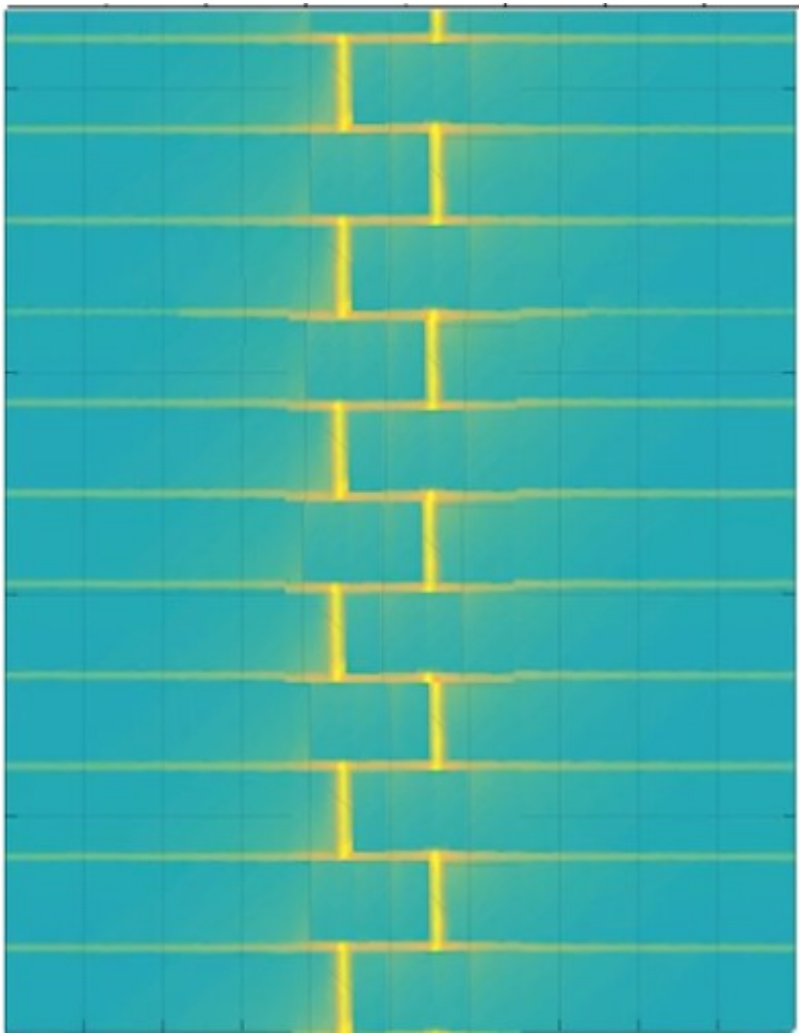


https://www.taitradioacademy.com/topic/how-does-modulation-work-1-1/

Frequency Shift Keying

- 用兩個不同頻率表示 0 和 1

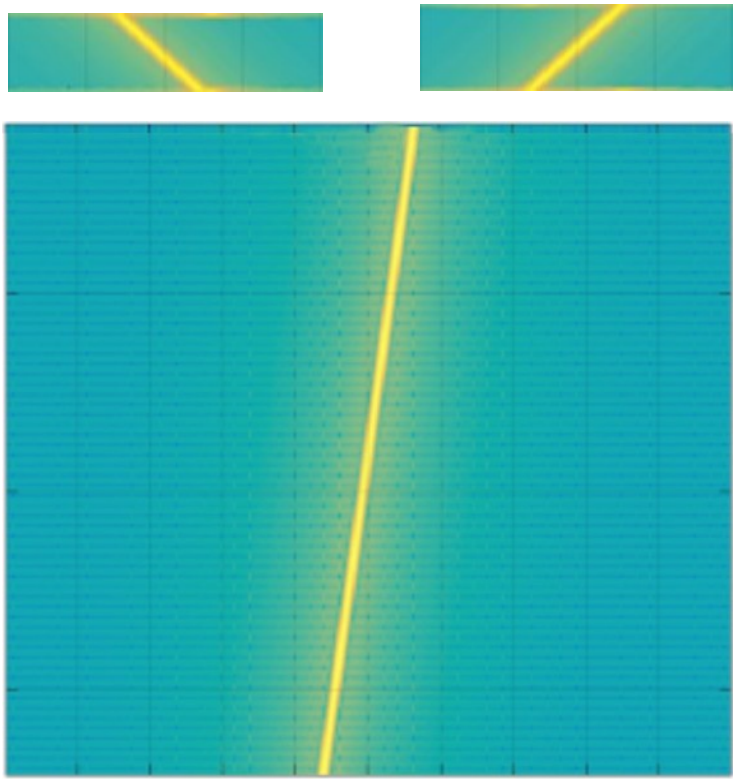




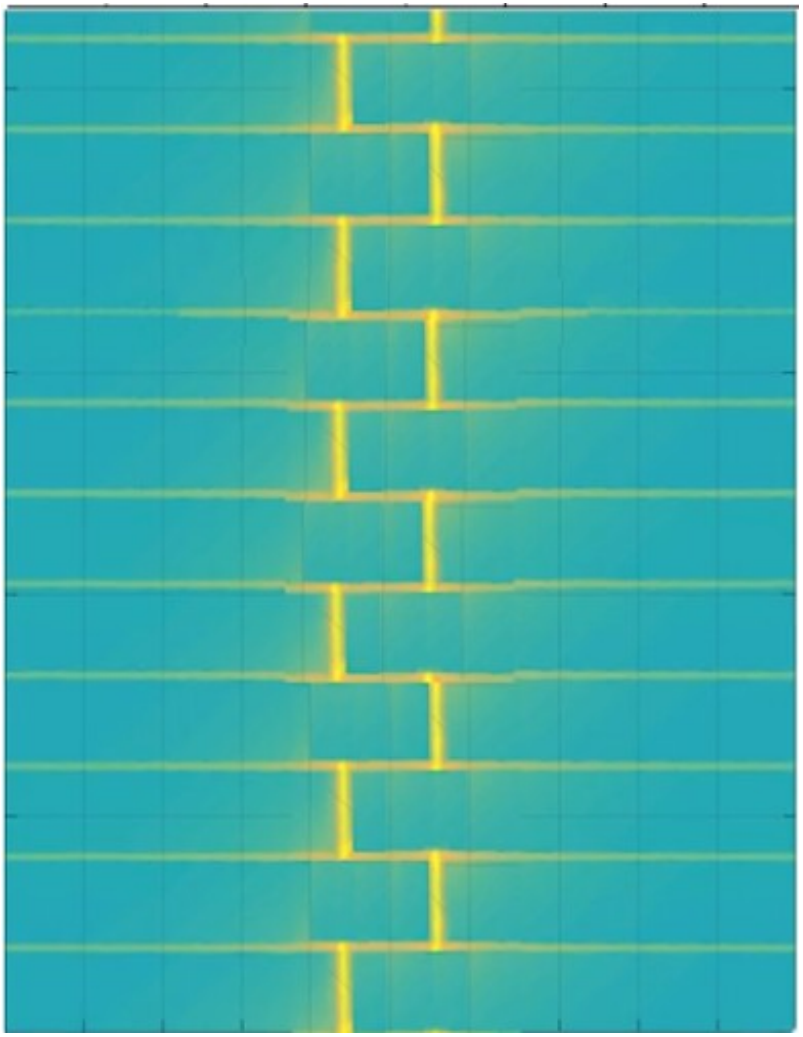
https://goo.gl/WxXhH5

Chirp Spread Spectrum

- 使用兩個載波做調頻
- Chirp 隨時間遞增 / 遞減



Chirp 載波

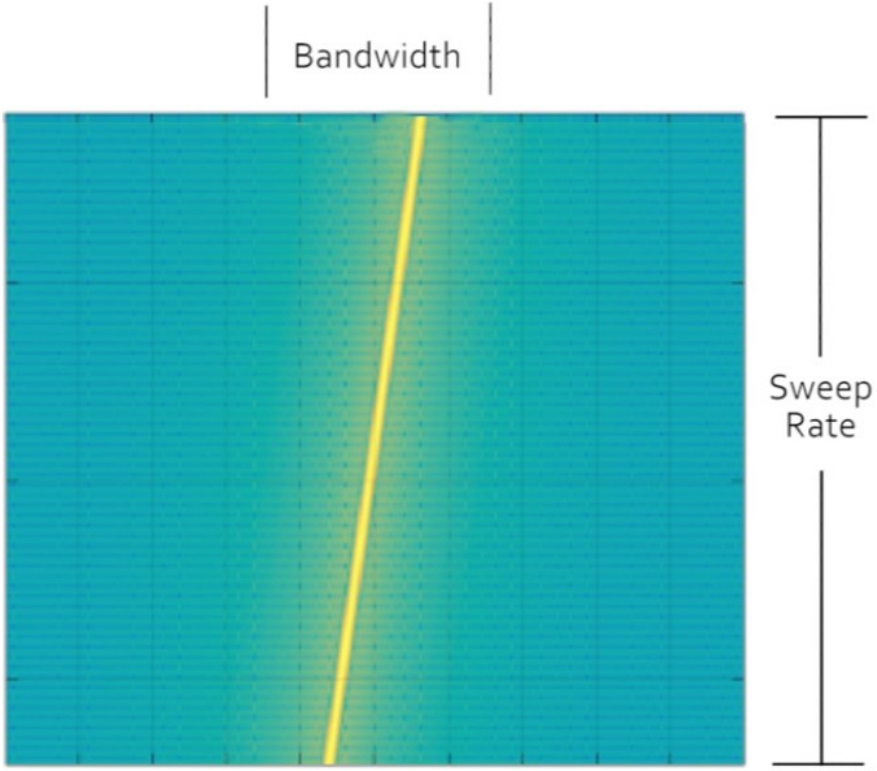


數位波形載波

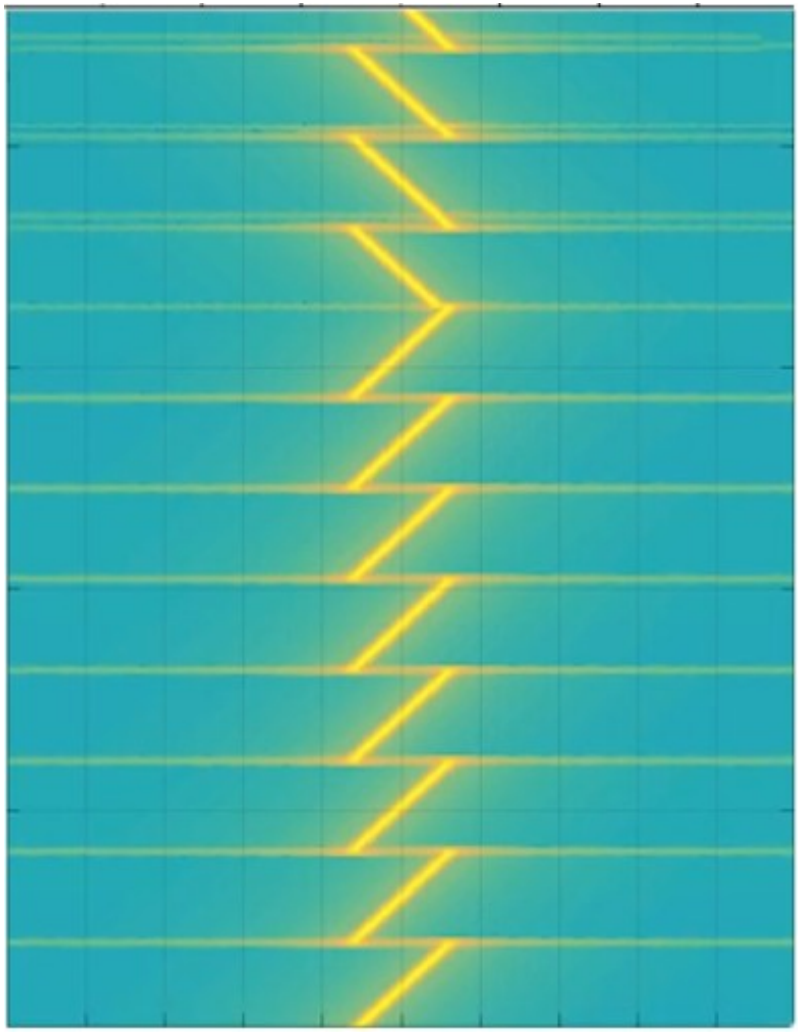
https://goo.gl/WxXhH5

Chirp Spread Spectrum

- Bandwidth 影響訊號品質
- Sweep Rate 影響處理增益



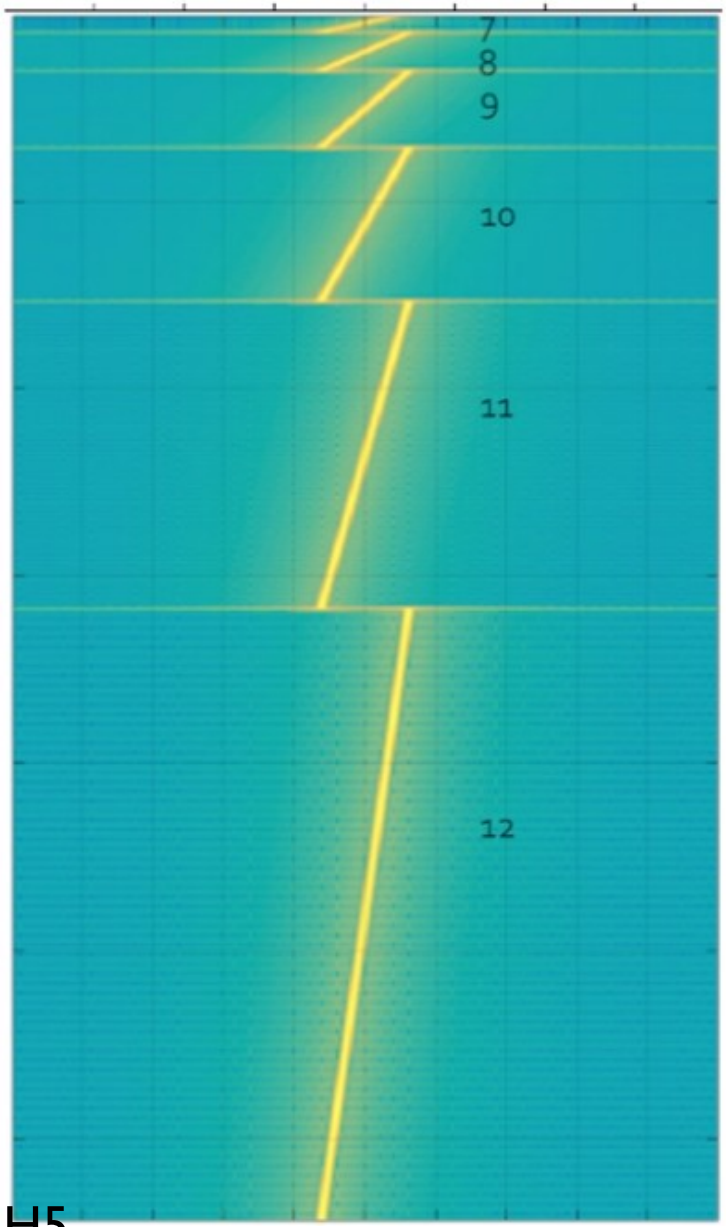
可調整 Bandwidth 和 Sweep Rate



Chirp Spread Spectrum

Spreading Factor

- Duration of the chirp
- 從SF7(**7)到SF12(**12)



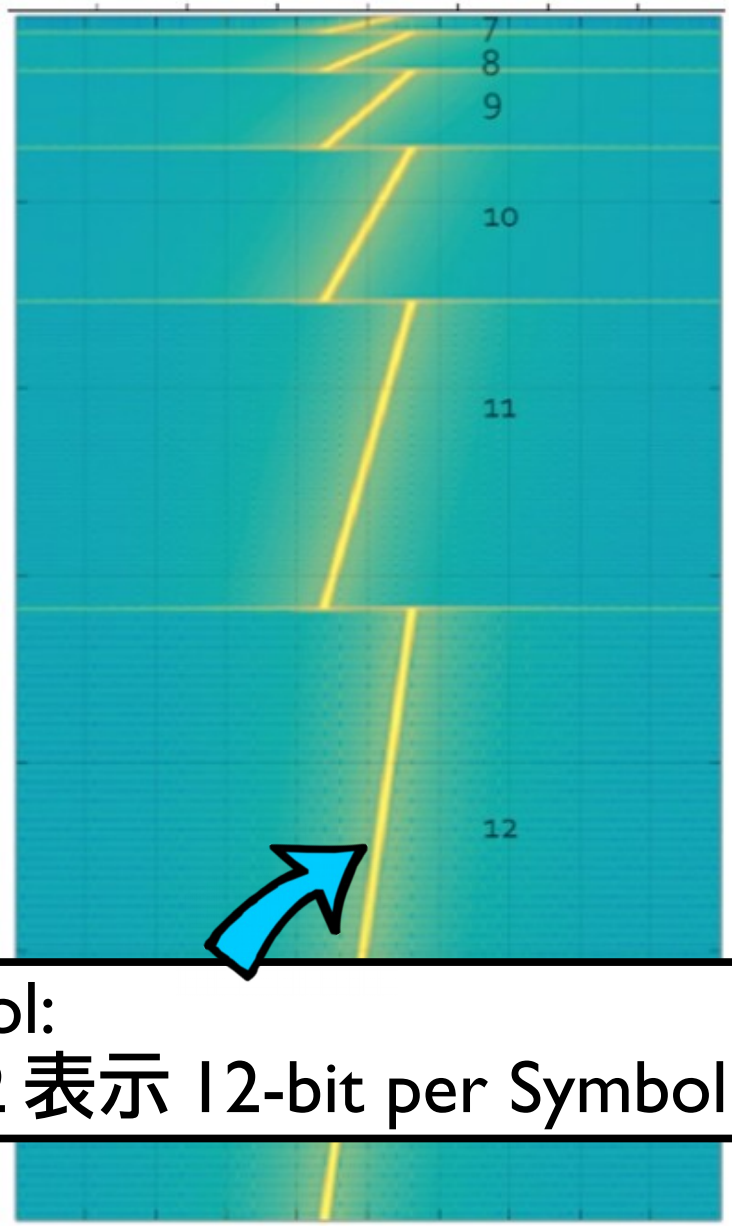
https://goo.gl/WxXhH5

Spreading Factor

- Symbole 中的 Chip 數量

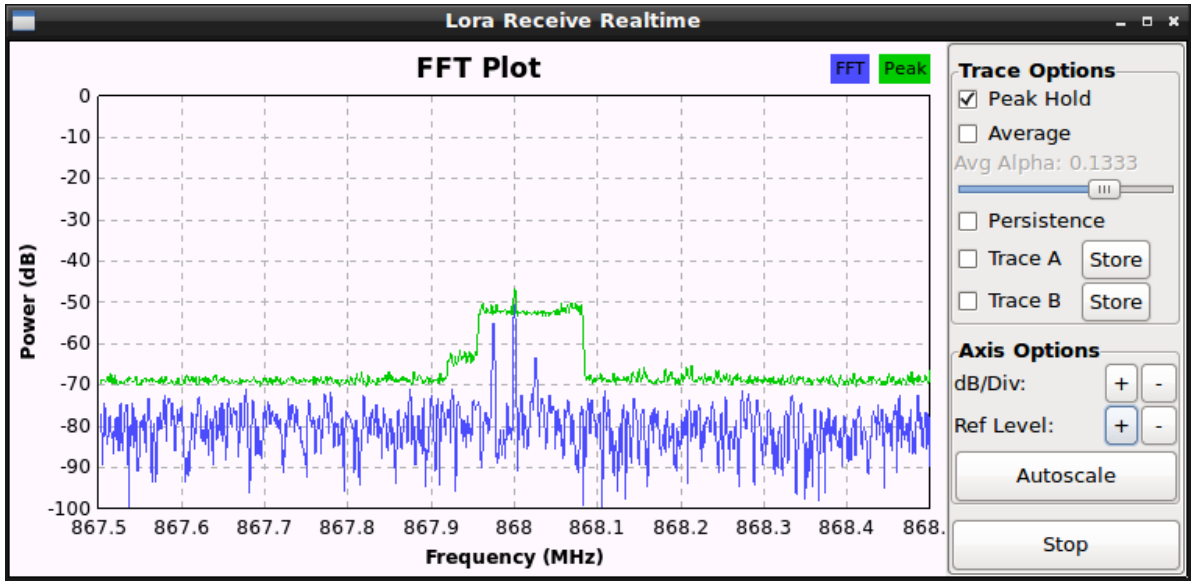
SpreadingFactor (RegModulationCfg)	Spreading Factor (Chips / symbol)	LoRa Demodulator SNR
6	64	-5 dB
7	128	-7.5 dB
8	256	-10 dB
9	512	-12.5 dB
10	1024	-15 dB
11	2048	-17.5 dB
12	4096	-20 dB

- $2^{SF}\, chips = 1\ symbol$

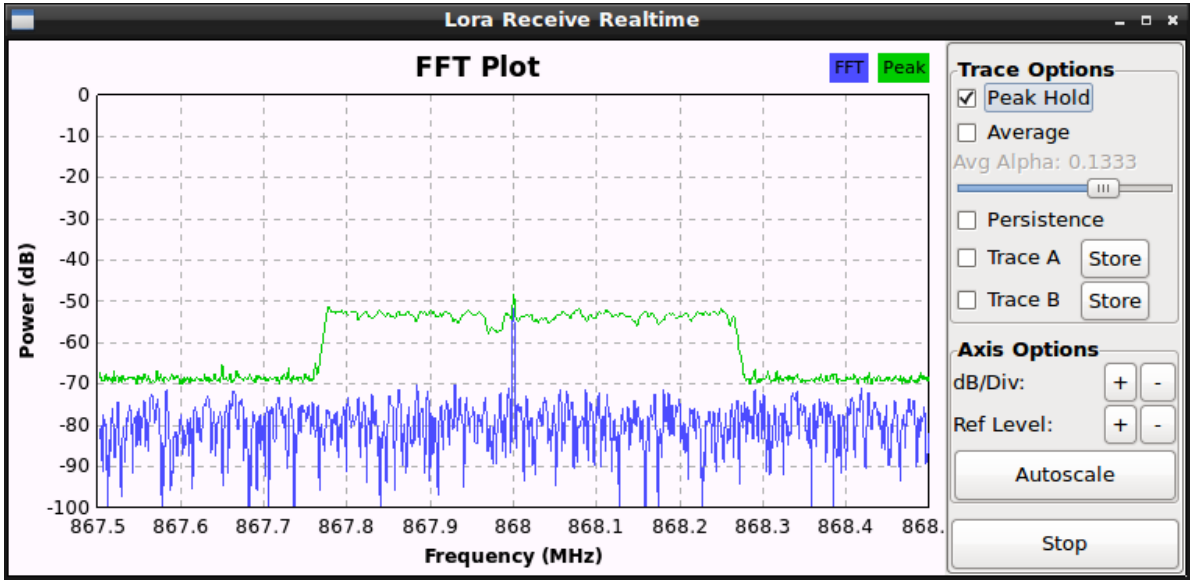


Symbol:
SF=12 表示 12-bit per Symbol

(Modulation) Bandwidth



BW=125kHz



BW=500kHz

計算公式

- Bit: 最小的資料單位

$$Bit\ rate = R_b \qquad R_b = SF \cdot \frac{BW}{2^{SF}}$$

- Symbol: 由 Chip 組成的有意義的資訊

$$Symbol\ rate = R_s \qquad R_s = \frac{BW}{2^{SF}}$$

- Chip: 最小的發送單位，為頻寬的倒數

$$Chip\ rate = R_c \qquad R_c = R_s \cdot 2^{SF}$$

https://electronics.stackexchange.com/questions/278192

計算題

- 如果資料率(Date Rate, DR)公式如下：

$$DR = SF \cdot \frac{BW}{2^{SF}} \cdot CR$$

- 給定以下參數，請問資料率多少？
- SF = 12(Spreading factor)
- BW = 125kHz(Bandwidth)
- CR = 4/8(Coding rate)

- 答：183.1055bps https://electronics.stackexchange.com/questions/277722

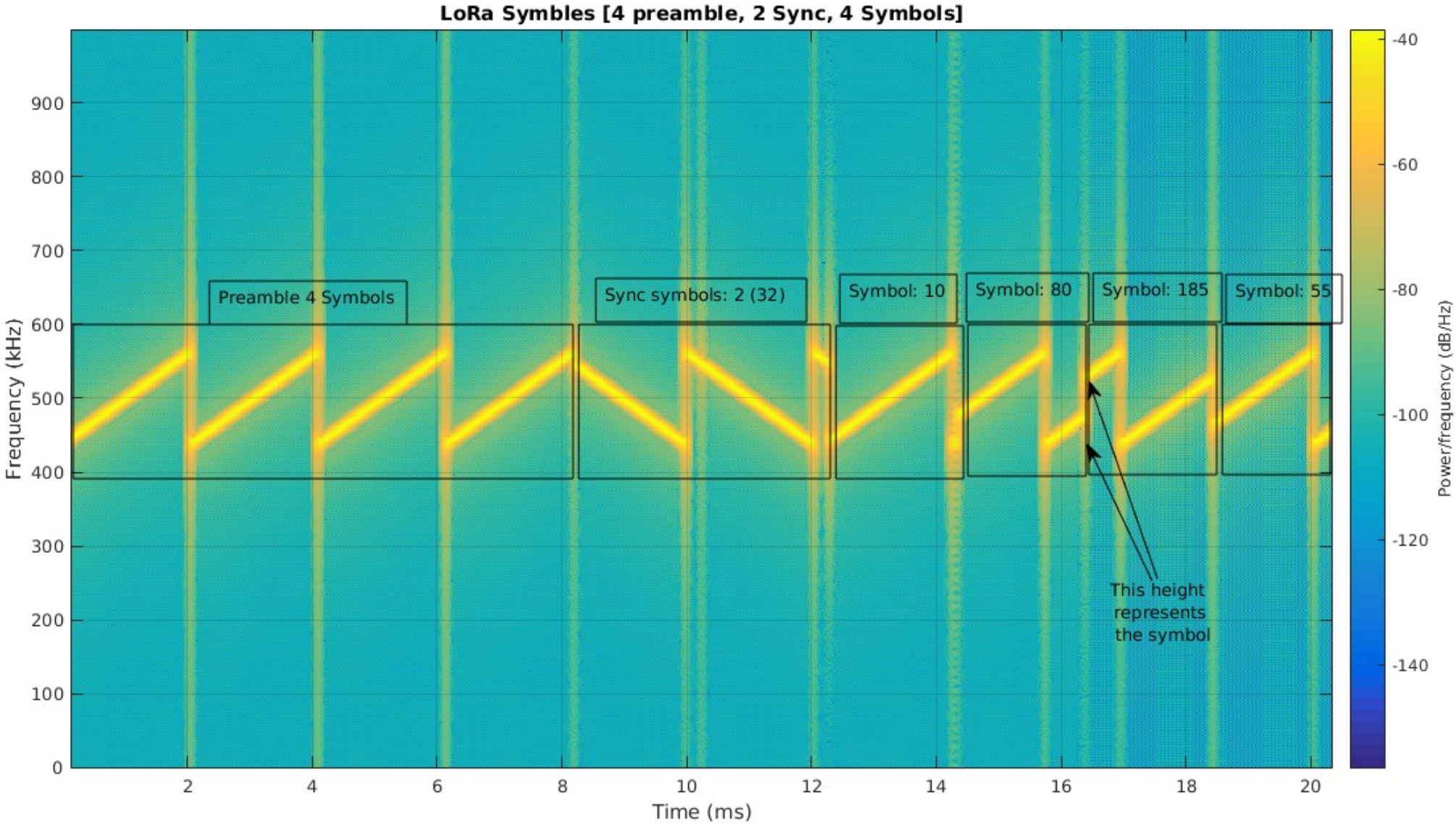
Chirp Spread Spectrum 名詞解釋

- Bit: 最小的資料單位
- Chip: 最小的發送單位，為頻寬的倒數(p.28)
- Symbol: 由 Chip 組成的有意義的資訊(p.28)
- Bandwidth: 調變頻寬
- Spreading Factor: 一個 symbol 中被編碼的 bit 數
- Chirp: 連續遞增或遞減的頻率
- Chirp Rate: 為 Chirp Frequency 一階導數
- Preamble: 前導訊號，接續在後為資料訊號
- Code Rate:k/r 表示每 k 位有用資訊，但產生 n 位數據

https://media.ccc.de/v/33c3-7945-decoding_the_lora_phy

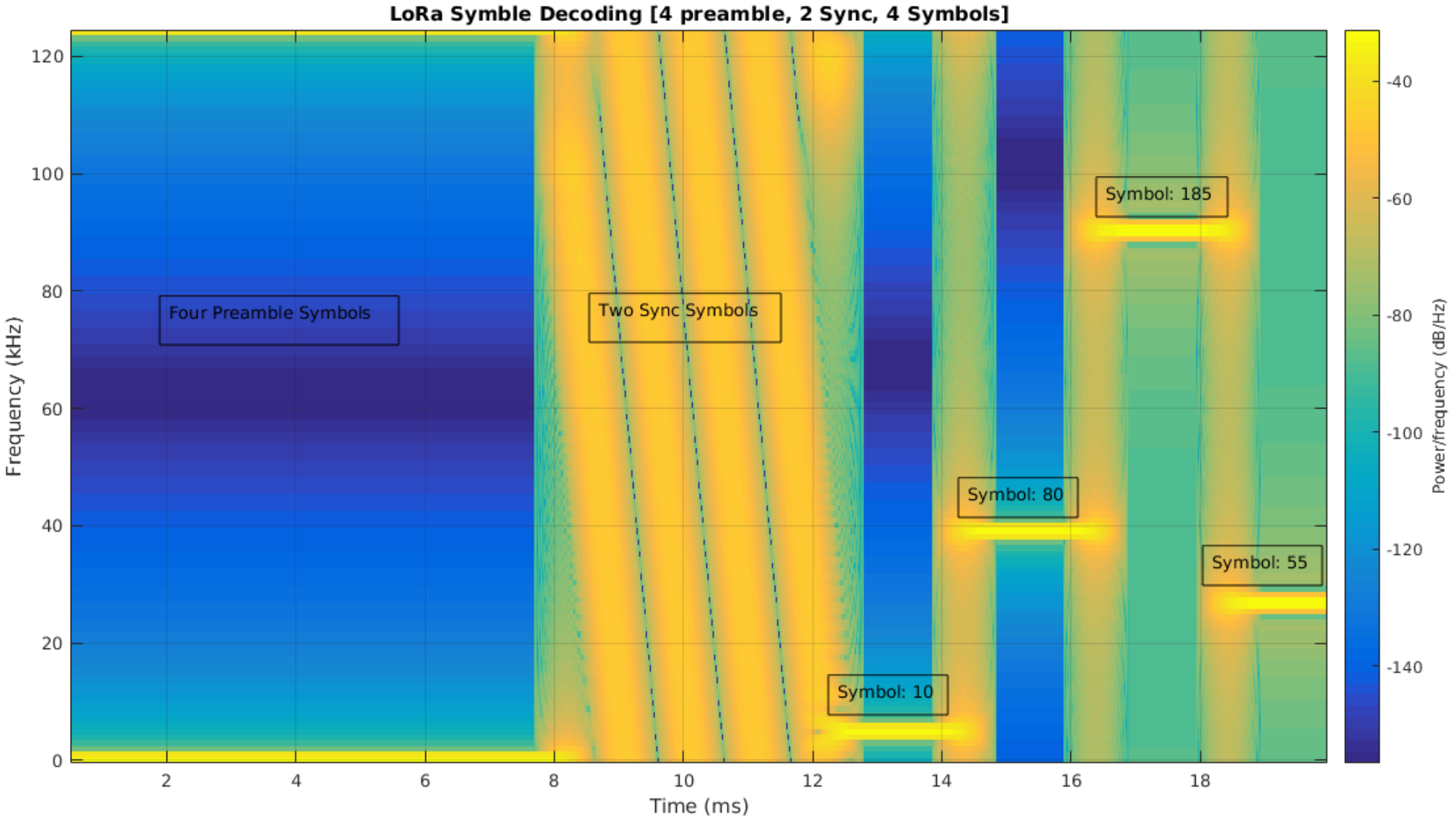
接收端與解調變

接收端收到的訊號



http://www.sghoslya.com/p/lora_9.html

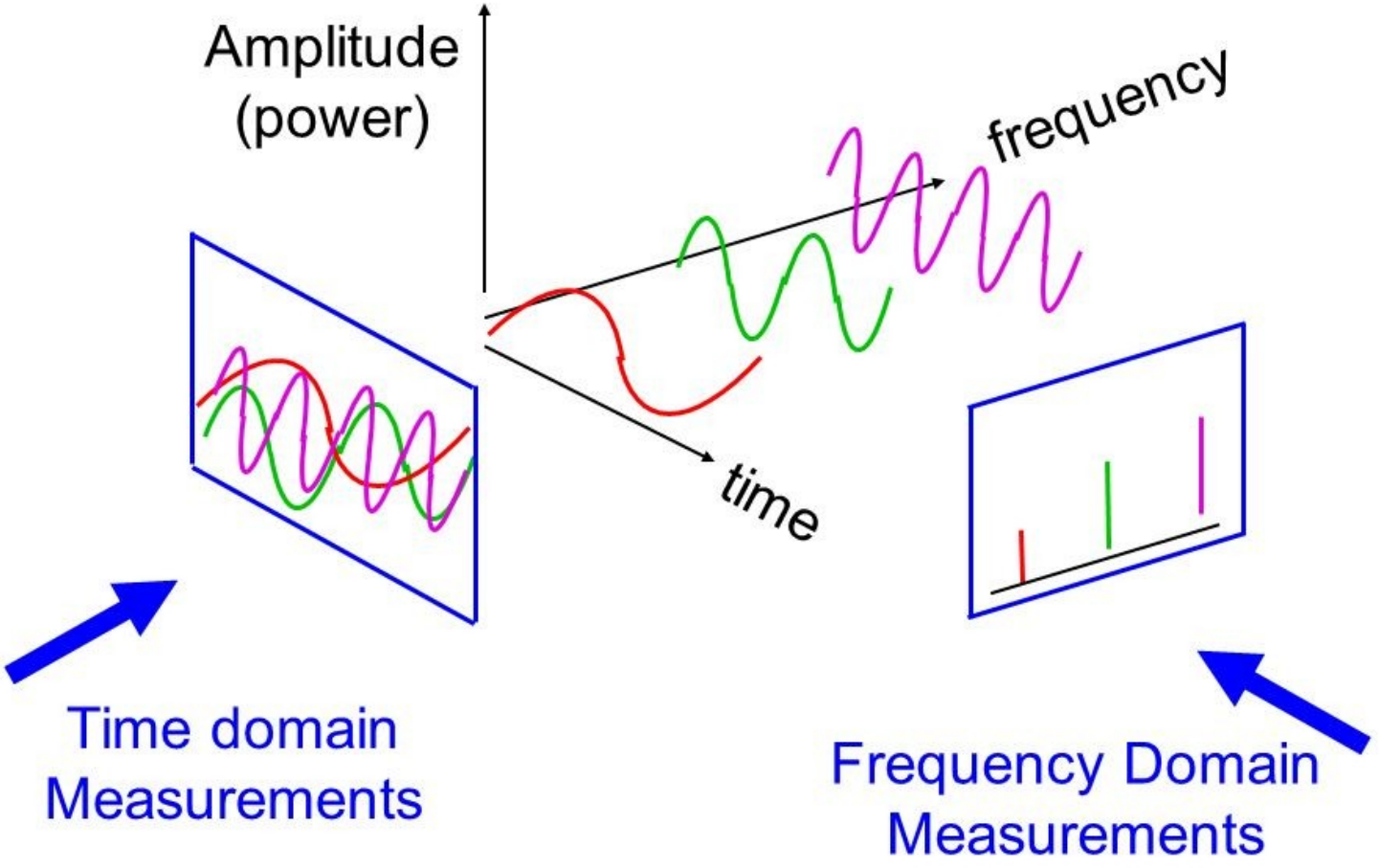
和 up-chirp 與 down-chirp 相乘



http://www.sghoslya.com/p/lora_9.html

Frequency Domain vs. Time Domain

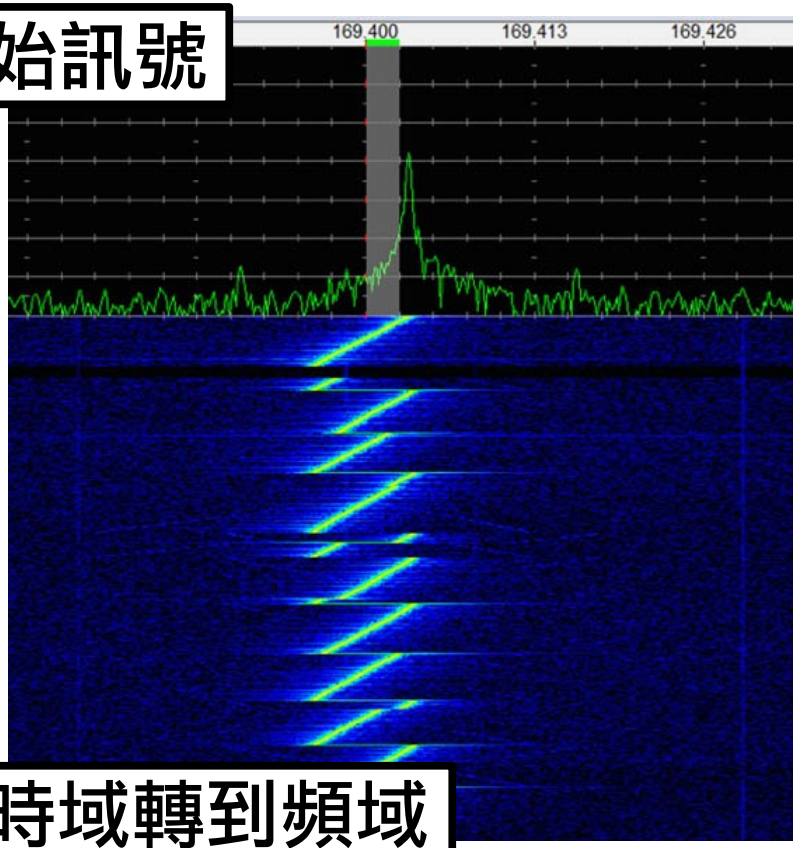
- 使用 FFT 將時域轉到頻域



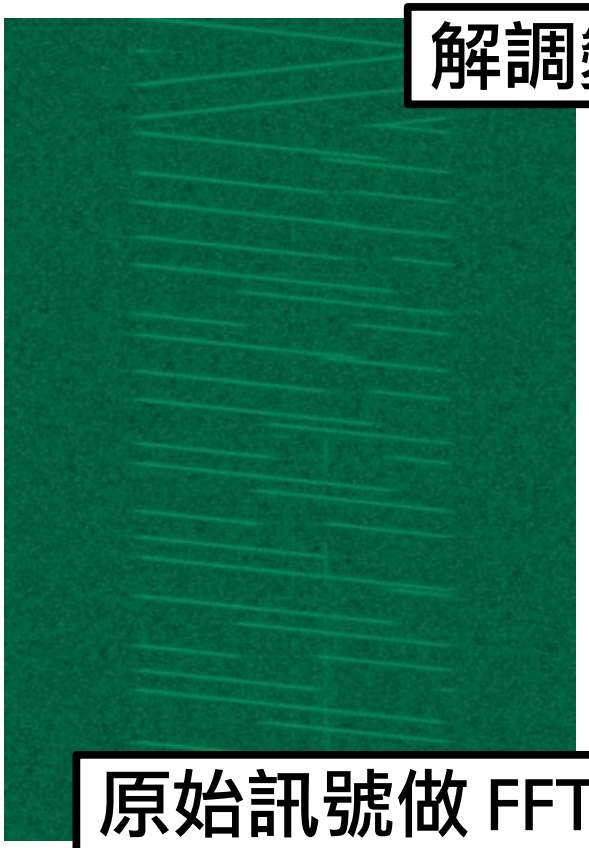
http://slideplayer.com/slide/8521080/

可讀取到 Symbol 數值

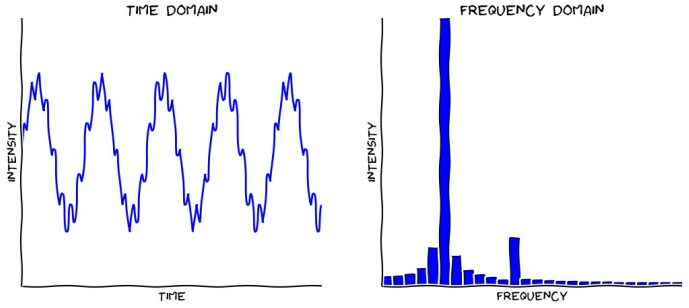
原始訊號



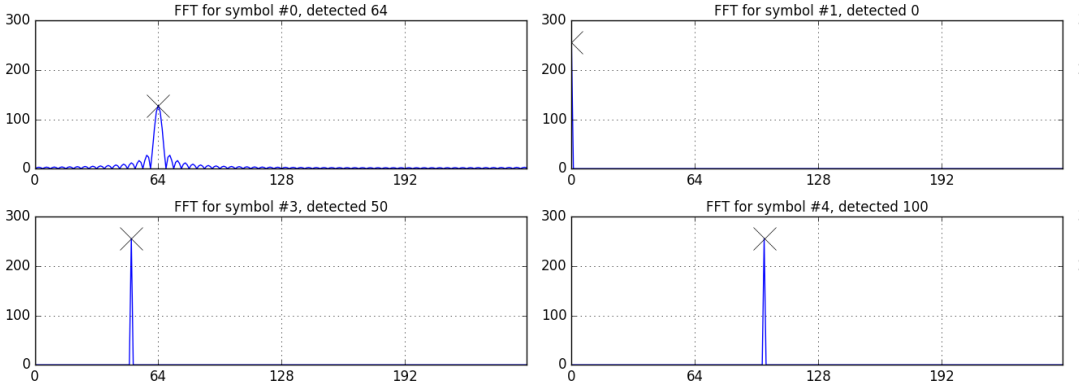
解調變後



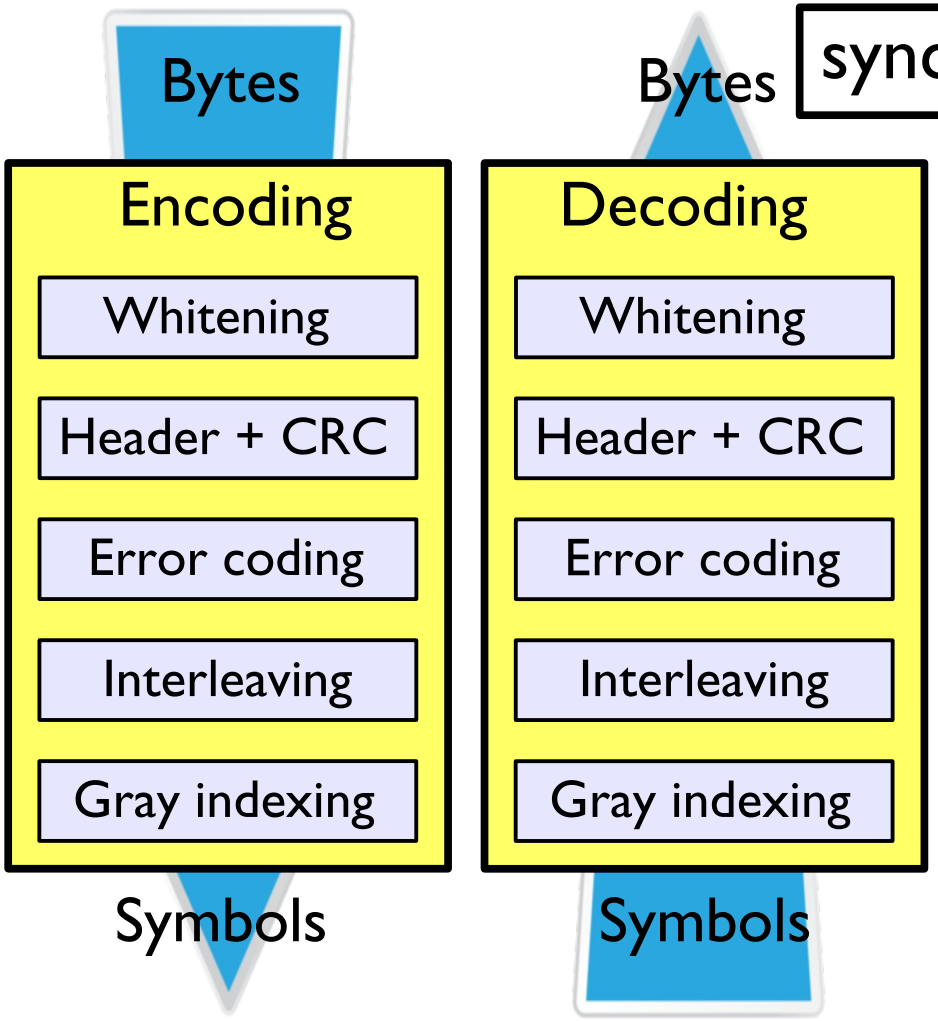
FFT 將時域轉到頻域

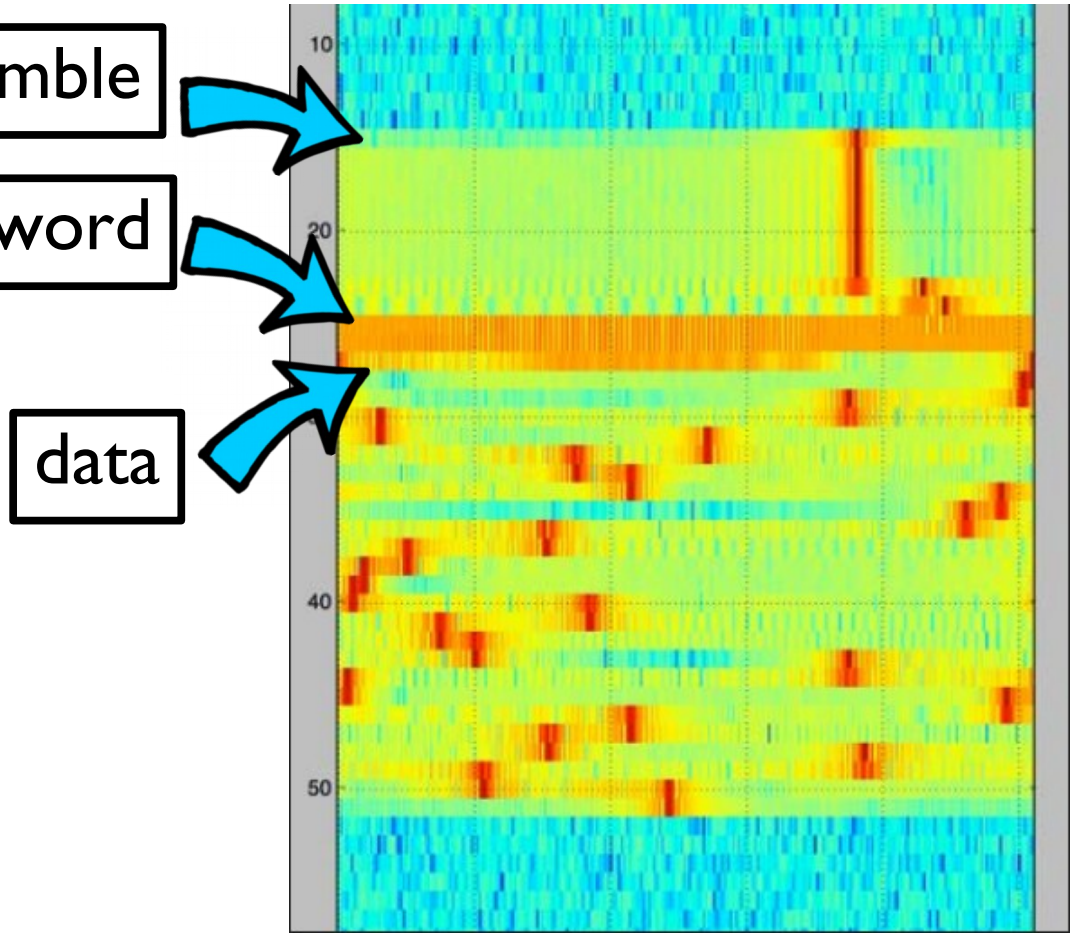


原始訊號做 FFT 的結果



從 Symbols 解碼回原始的 Bytes





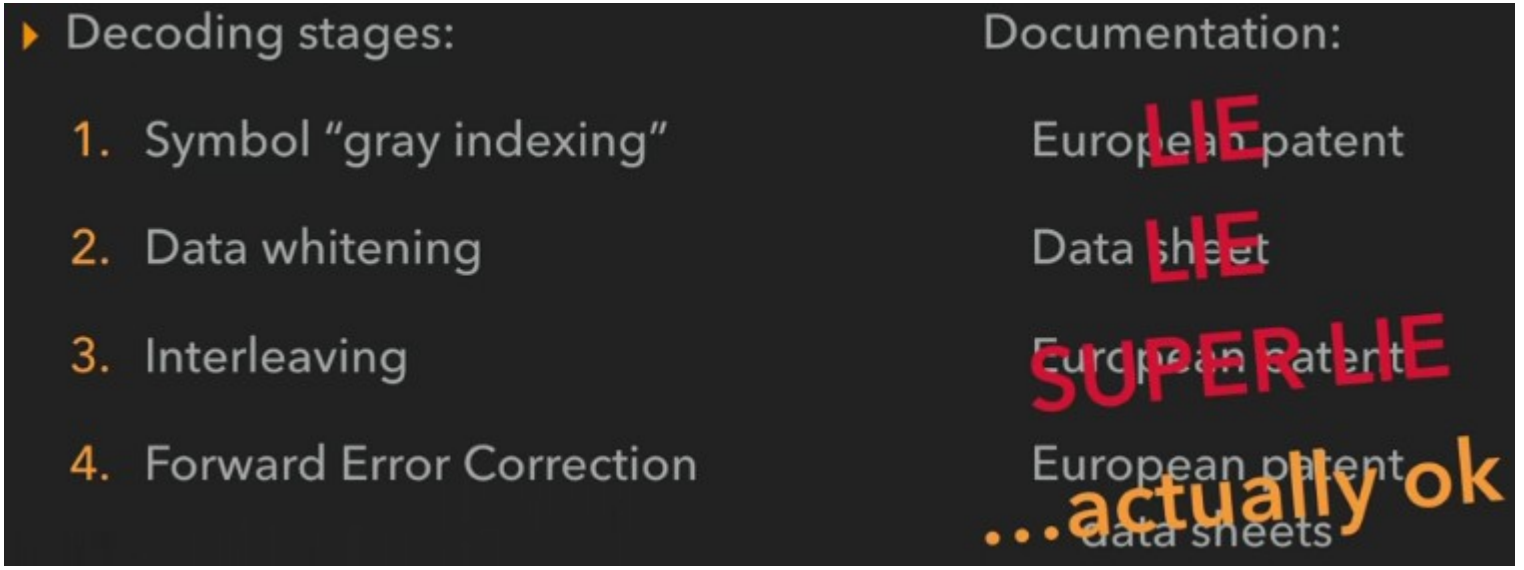
https://myriadrf.org/blog/lora-modem-limesdr/

更多參考資料

- 技術文件
  - Semtech European patent application 13154071.8
  - LoRa Alliance LoRaWAN spec
  - Semtech app notes AN1200.18 and AN1200.22

- 參考實做
  - Partial implementation in open source rtl-sdrangelove
  - Observations at https://revspace.nl/DecodingLora

特別是 Matt Knight 的影片



https://www.youtube.com/watch?v=NoquBA7IMNc

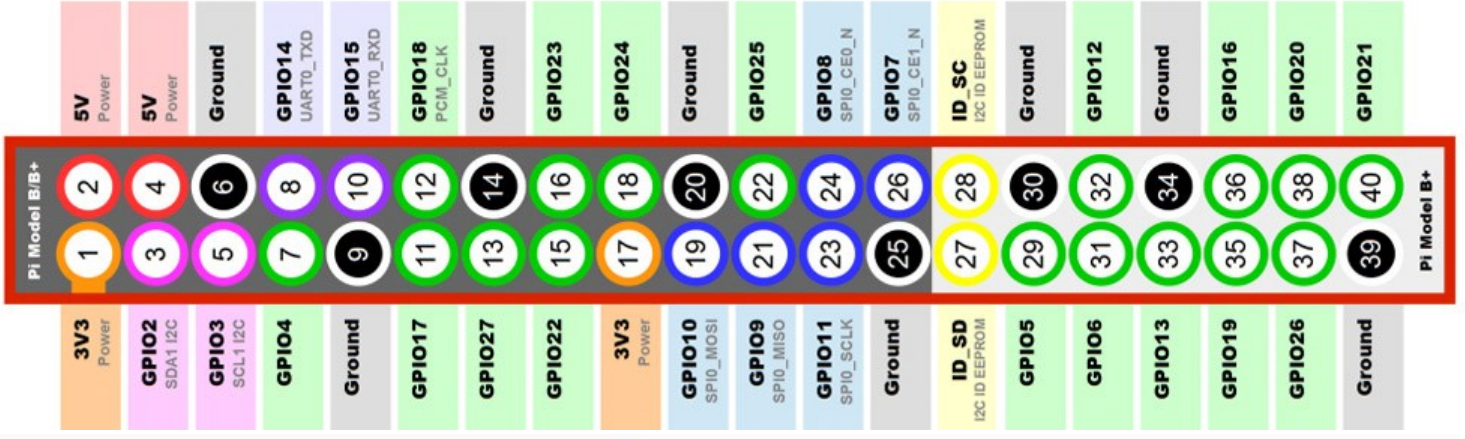
有了基本概念就好動手了

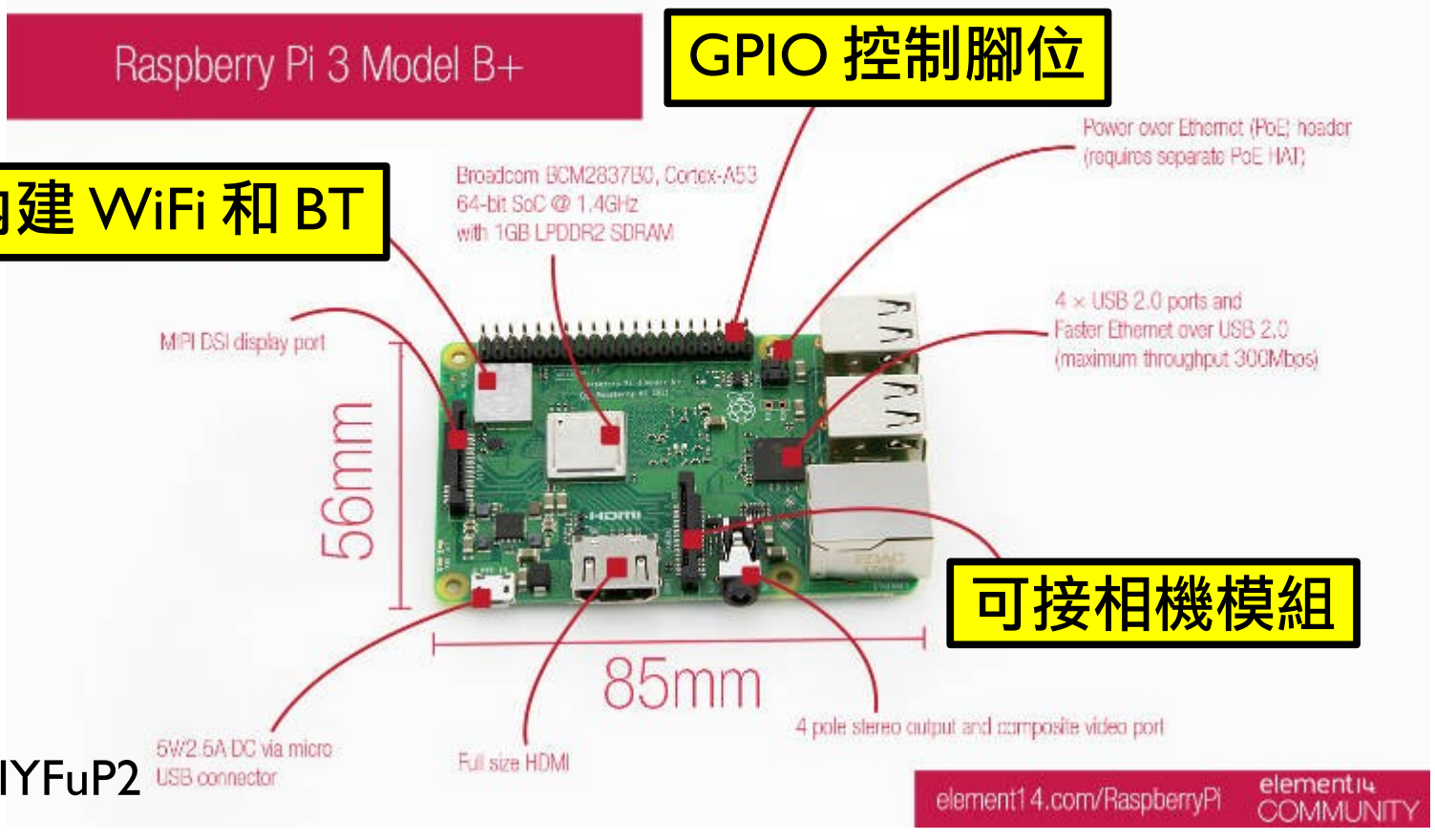
Raspberry Pi

- 信用卡大小般的電腦

http://www.flickr.com/photos/fotero/7697063016/

Pi 3B+ 硬體規格





http://bit.ly/2IYFuP2

今日環境

- 硬體：Raspberry Pi 3B/3B+
- 作業系統：2018-06-27-raspbian-stretch.img

- 為了可以使用USB轉TTL傳輸線
  - 修改/boot/config.txt，新增三行
    - dtoverlay=pi3-miniuart-bt
    - core_freq=250
    - enable_uart=1

```
55 # Enable audio (loads snd_bcm
56 dtparam=audio=on
57 dtoverlay=pi3-miniuart-bt
58 core_freq=250
59 enable_uart=1
```

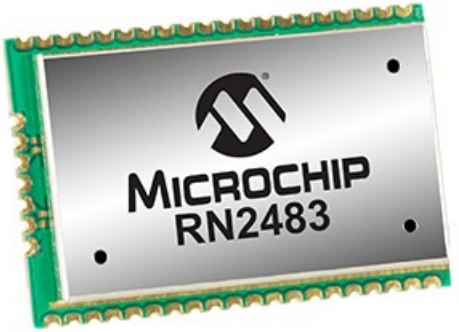
新增三行

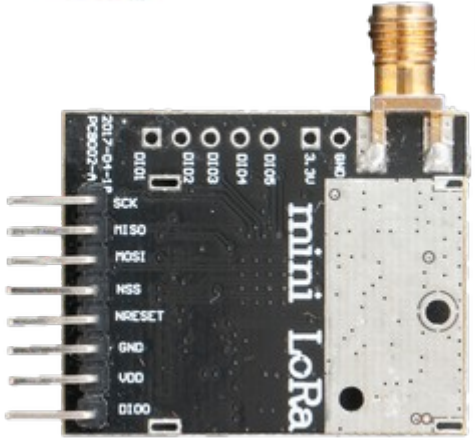
  - 修改/boot/cmdline.txt，將quiet splash的quiet移除

```
1 dwc_otg.lpm_enable=0 console=serial0,115200
  console=tty1 root=/dev/mmcblk0p2 rootfstype=
  ext4 elevator=deadline fsck.repair=yes rootw
  ait quiet splash plymouth.ignore-serial-cons
  oles quiet init=/usr/lib/raspi-config/init_r
  esize.sh
```

刪除 quiet

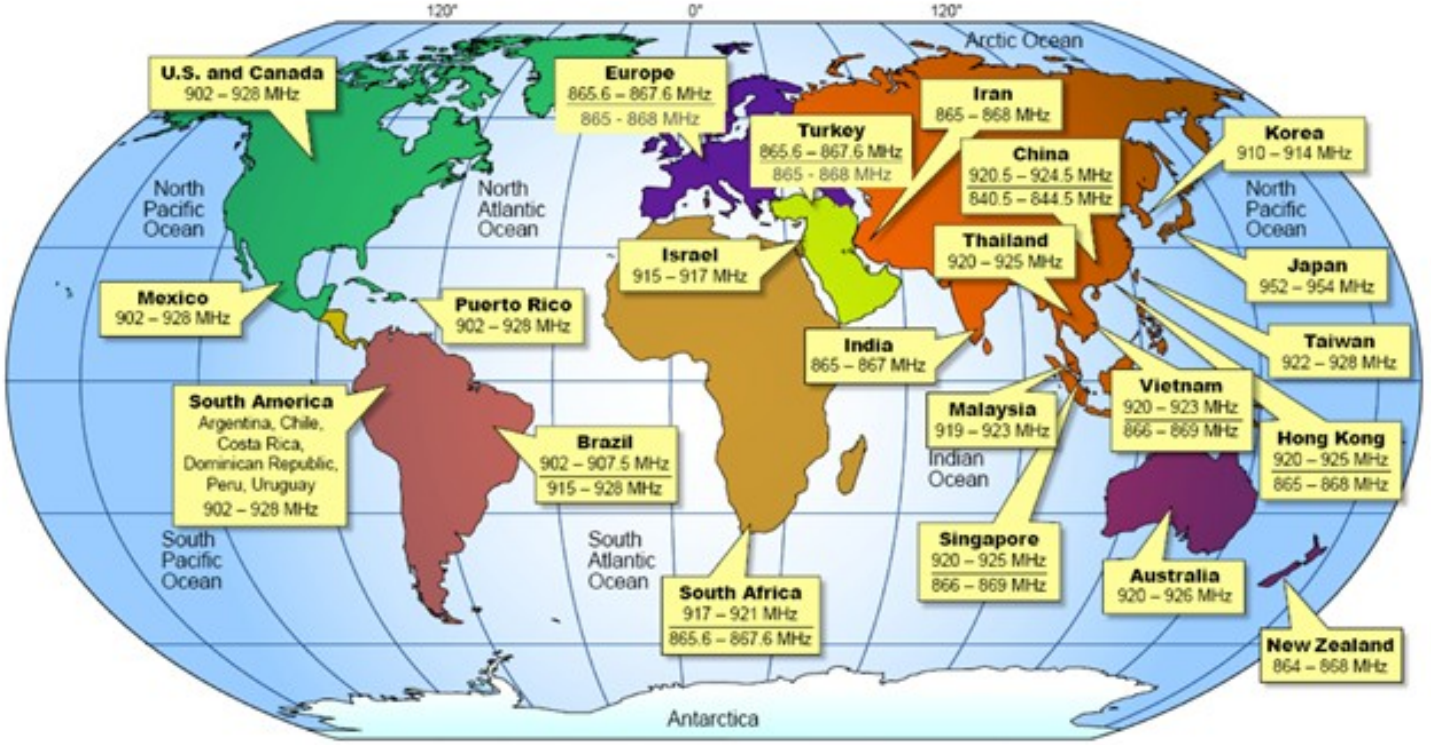
LoRa Module 百百種





Semtech SX127x 系列和 ISM 頻段

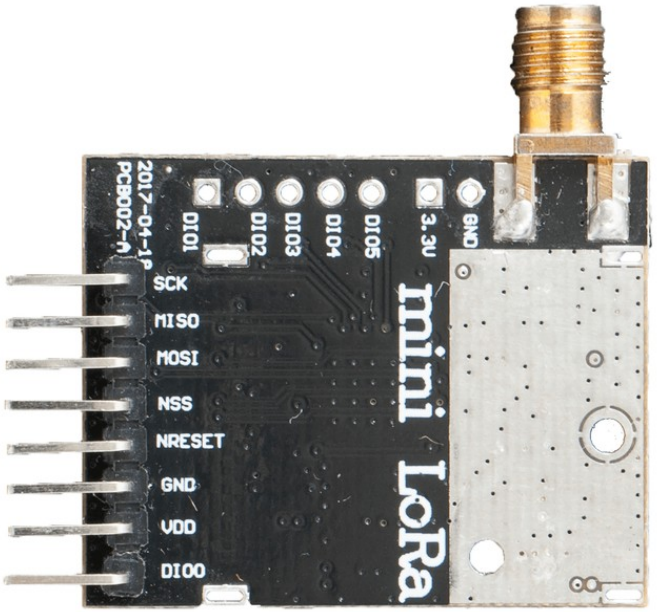
Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
SX1276	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1277	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
SX1278	137 - 525 MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1279	137 - 960MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm



http://rfid4u.com/rfid-basics-resources/basics-rfid-regulations/

SX1276 LoRa Module

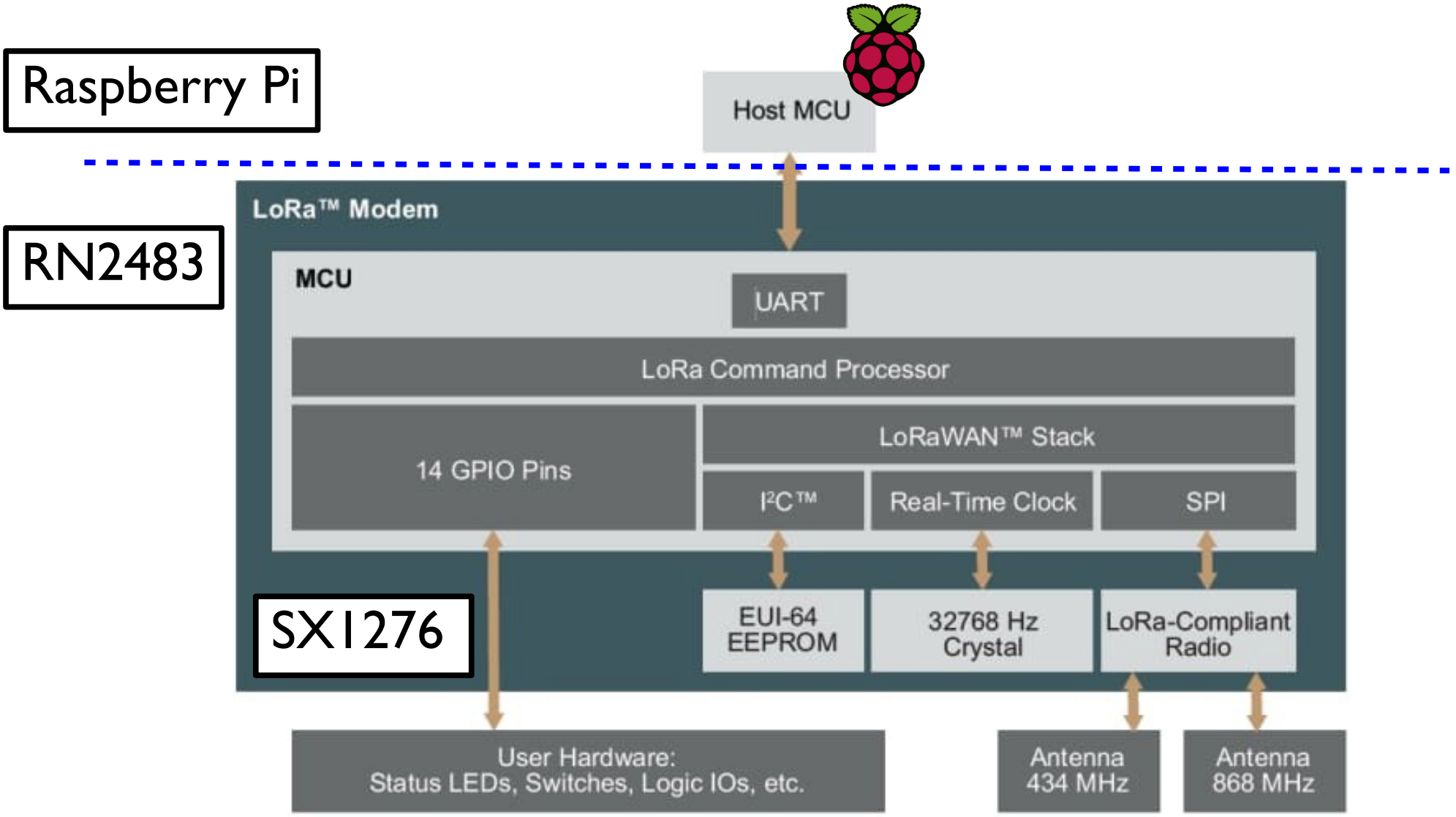
- 特色
  - 傳輸距離可達 15km
  - 長達 10 年的低功耗 (使用鈕扣電池)
  - 支援 868-928MHz 頻率範圍
  - 支援 FSK,OOK,GFSK,LoRa 等調變技術
  - 最快傳輸速度可達 300kbps
  - 最高鏈路預算可達 168dB
  - 最大輸出功率可達 +20dBm
  - 靈敏度可達 -148dBm
  - 每次最大傳輸 256bytes(含 CRC)
  - 低功耗如 RX<10mA 和 Sleep<200nA
  - 工作電壓 :1.8-3.7V(預設 3.3V)



應用

- 自動化抄表
- 智慧家居和安全系統
- 工業監控和控制
- 遠端植栽灌溉系統
- 無線感測器收集系統

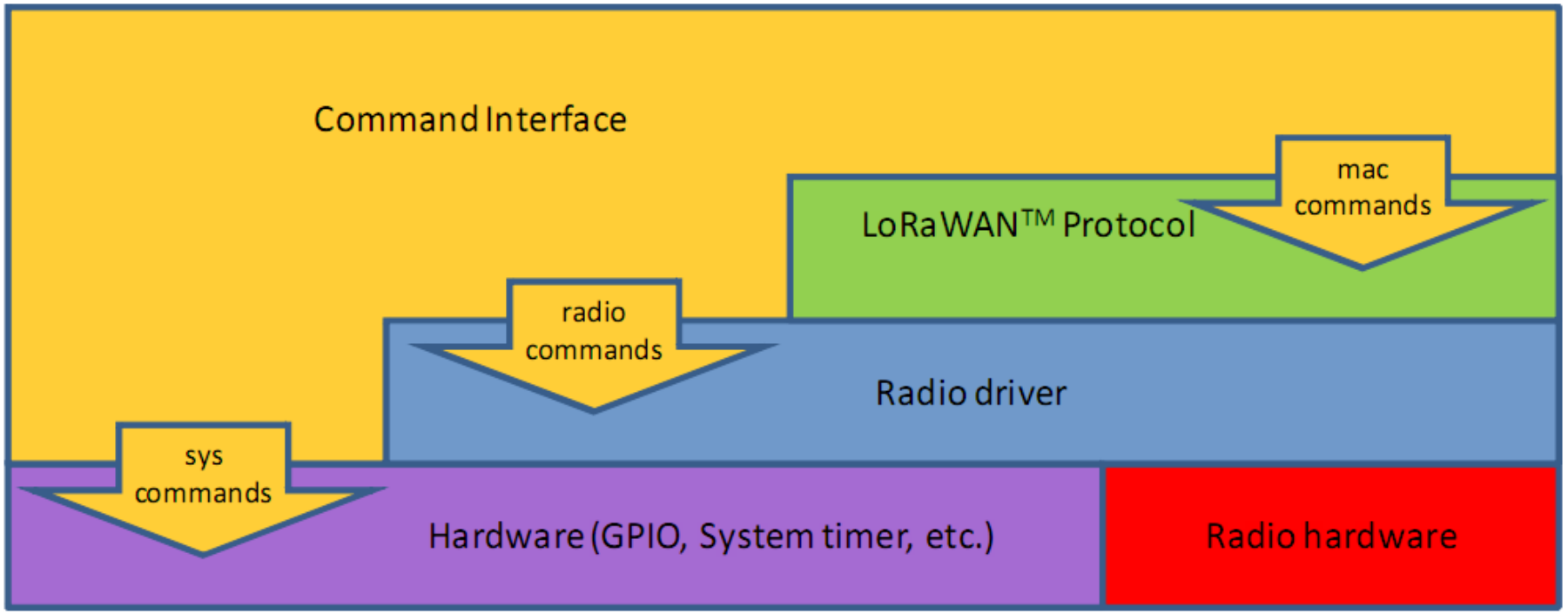
Microchip RN2483 系統架構



http://www.mouser.tw/new/microchip/microchip-rn2483-module/

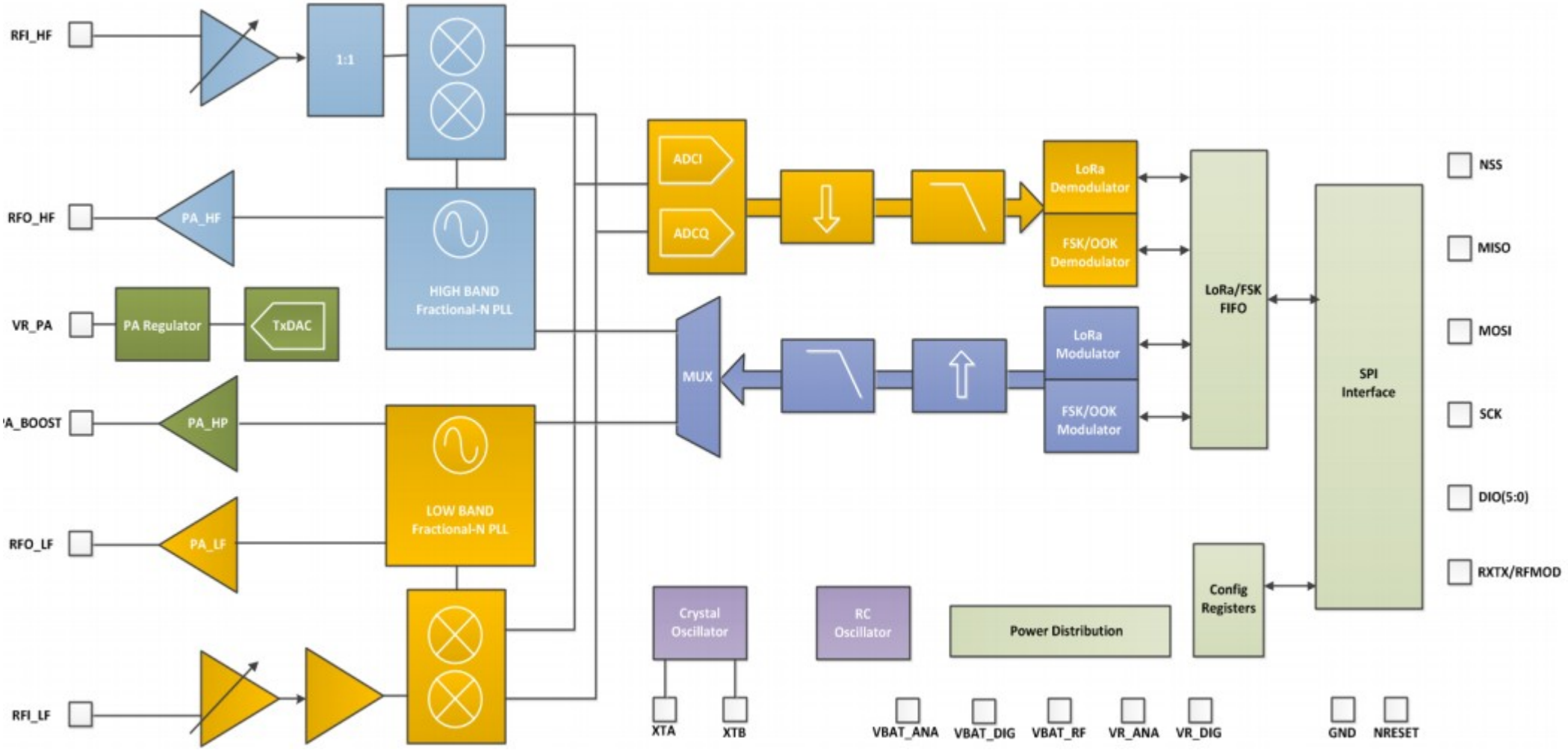
RN2483 三種指令

- sys 設定系統與 GPIO
- radio 設定射頻 ← 主要設定指令
- mac 設定 LoRaWAN



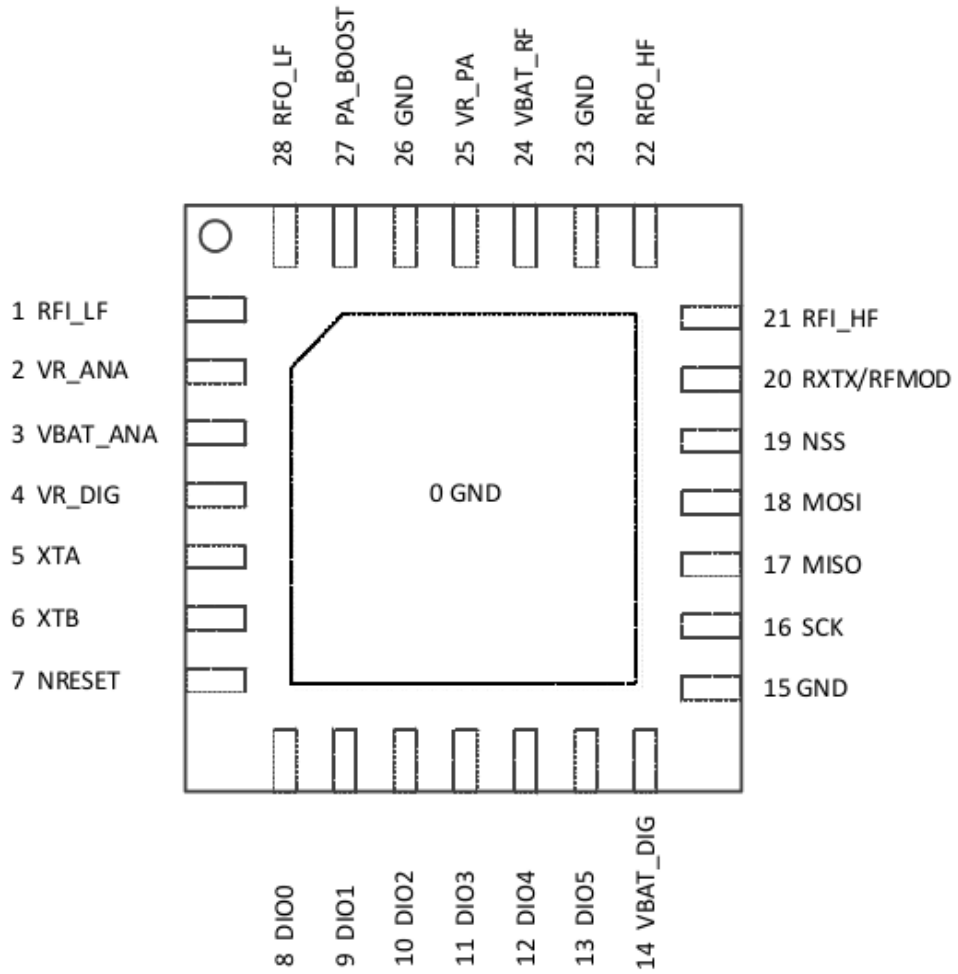
http://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf

SX127x 功能方塊圖



http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

SX1276 腳位定義

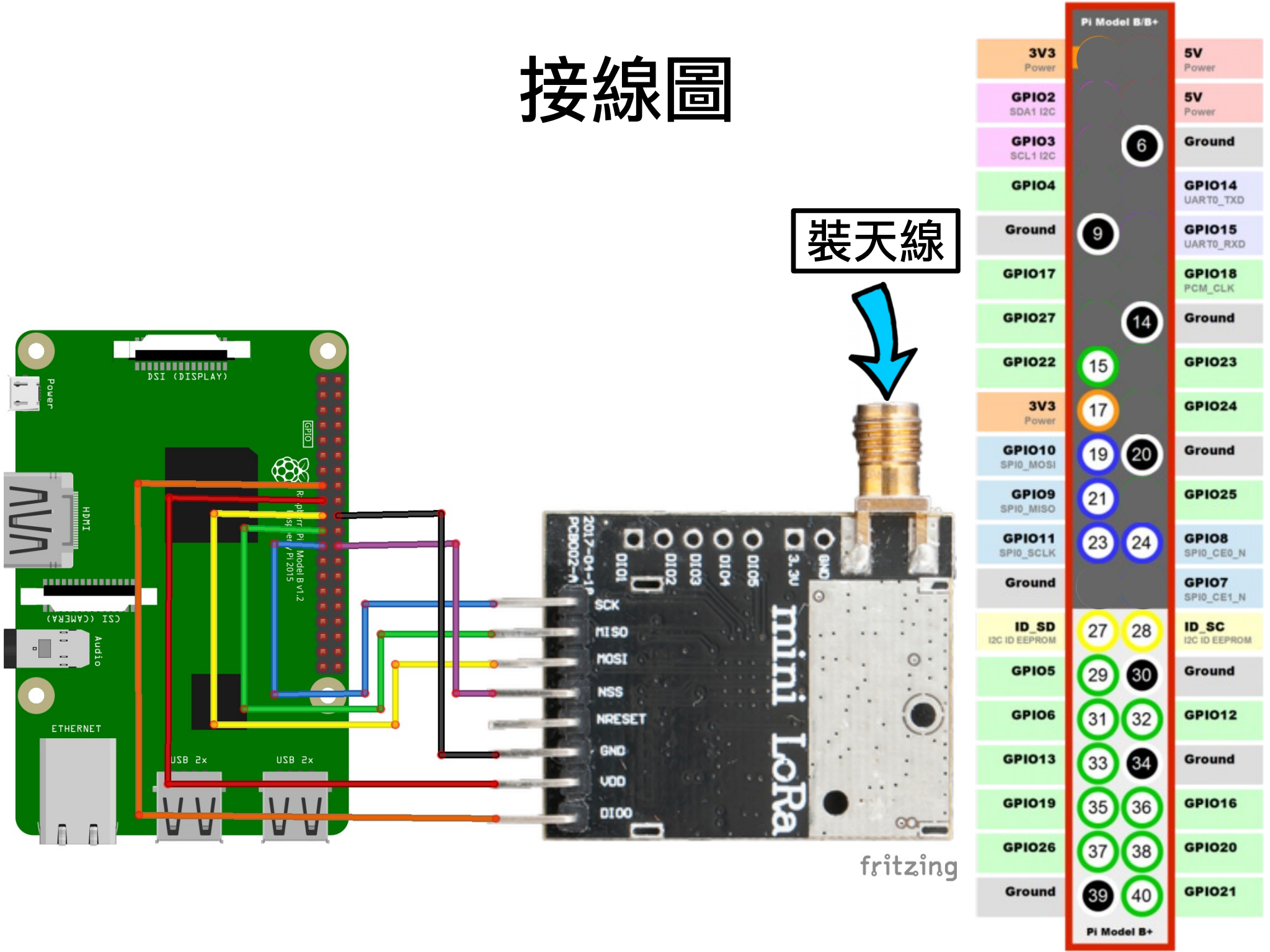


SX1276/77/79

Number	Name	Type	Description
	SX1276/77/79/(78)	SX1276/77/79/(78)	SX1276/77/79/(78)
0	GROUND	-	Exposed ground pad
1	RFI_LF	I	RF input for bands 2&3
2	VR_ANA	-	Regulated supply voltage for analogue circuitry
3	VBAT_ANA	-	Supply voltage for analogue circuitry
4	VR_DIG	-	Regulated supply voltage for digital blocks
5	XTA	I/O	XTAL connection or TCXO input
6	XTB	I/O	XTAL connection
7	NRESET	I/O	Reset trigger input
8	DIO0	I/O	Digital I/O, software configured
9	DIO1/DCLK	I/O	Digital I/O, software configured
10	DIO2/DATA	I/O	Digital I/O, software configured
11	DIO3	I/O	Digital I/O, software configured
12	DIO4	I/O	Digital I/O, software configured
13	DIO5	I/O	Digital I/O, software configured
14	VBAT_DIG	-	Supply voltage for digital blocks
15	GND	-	Ground
16	SCK	I	SPI Clock input
17	MISO	O	SPI Data output
18	MOSI	I	SPI Data input
19	NSS	I	SPI Chip select input
20	RXTX/RF_MOD	O	Rx/Tx switch control: high in Tx
21	RFI_HF (GND)	I (-)	RF input for band 1 (Ground)
22	RFO_HF (GND)	O (-)	RF output for band 1 (Ground)
23	GND	-	Ground
24	VBAT_RF	-	Supply voltage for RF blocks
25	VR_PA	-	Regulated supply for the PA
26	GND	-	Ground
27	PA_BOOST	O	Optional high power PA output, all frequency bands

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

接線圖

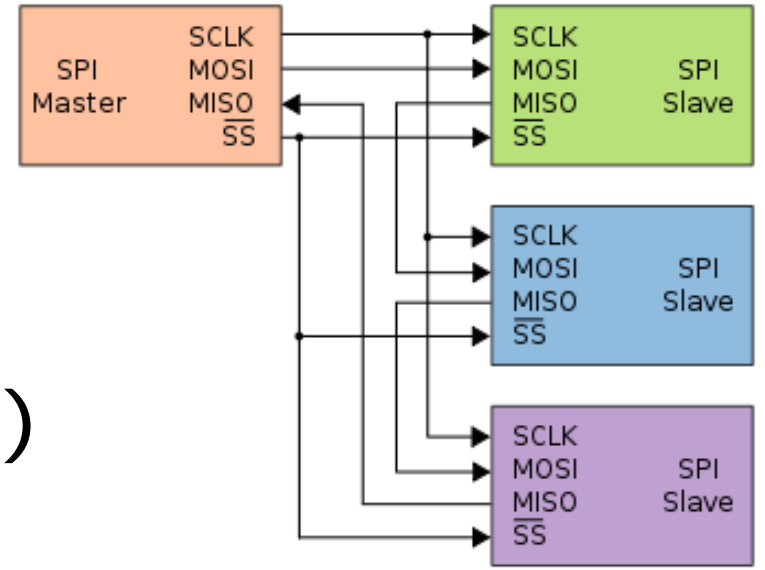

裝天線
DSI (DISPLAY)
Power
GPIO
HDMI
CSI (CAMERA)
Audio
ETHERNET
USB 2x
USB 2x
SCK
MISO
MOSI
NSS
NRESET
GND
VDD
DIO0
mini LoRa
fritzing
Pi Model B/B+
3V3 Power
5V Power
GPIO2 SDA1 I2C
5V Power
GPIO3 SCL1 I2C
Ground
GPIO4
GPIO14 UART0_TXD
Ground
GPIO15 UART0_RXD
GPIO17
GPIO18 PCM_CLK
GPIO27
Ground
GPIO22
GPIO23
3V3 Power
GPIO24
GPIO10 SPI0_MOSI
Ground
GPIO9 SPI0_MISO
GPIO25
GPIO11 SPI0_SCLK
GPIO8 SPI0_CE0_N
Ground
GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM
ID_SC I2C ID EEPROM
GPIO5
Ground
GPIO6
GPIO12
GPIO13
Ground
GPIO19
GPIO16
GPIO26
GPIO20
Ground
GPIO21
Pi Model B+

根據規格書寫程式

- 步驟：
  1. 遵循硬體所使用的通訊協定
  2. 從規格書找出暫存器位址
  3. 讀取 / 寫入暫存器數值

Serial Peripheral Interface(SPI)

- 主從式架構，可一對多
- 四線同步序列資料協定
  - SS: 週邊選擇線 (CE)
  - SCK: 序列時脈線 (SCLK)
  - MOSI: 主往從送
  - MISO: 從往主送



https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

遵循硬體所使用的通訊協定 (SPI)

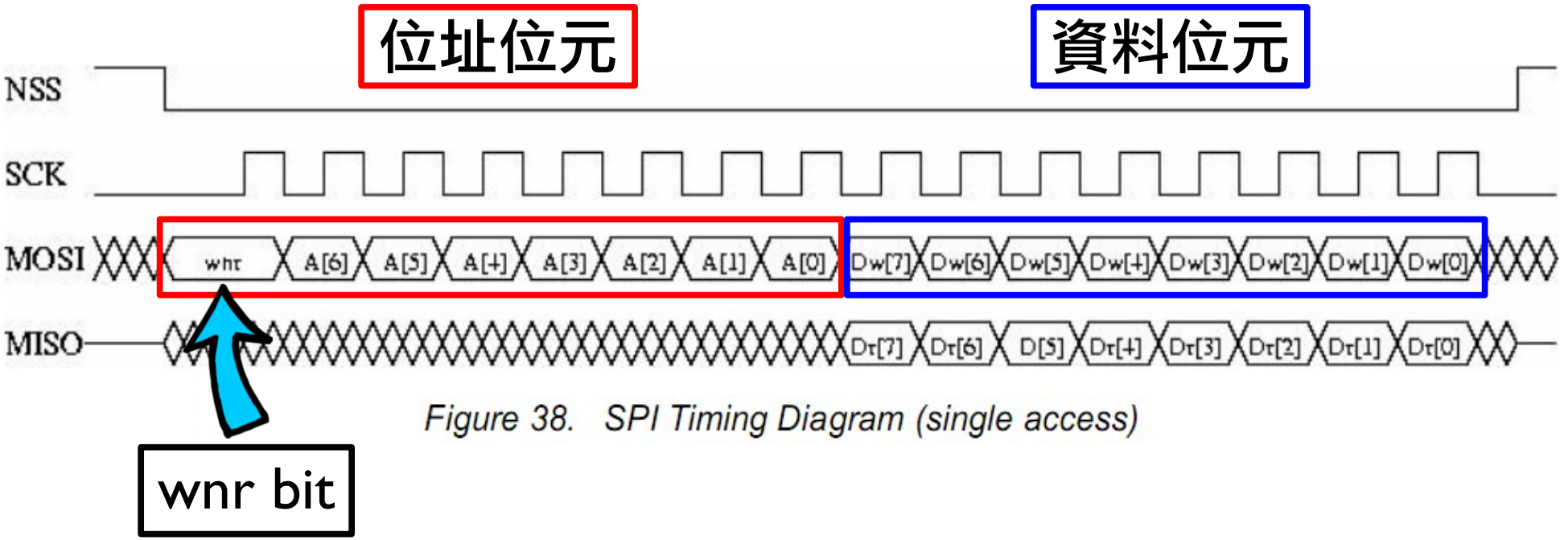


Figure 38. SPI Timing Diagram (single access)

- 傳送的第一個位元為位址位元
- 位址位元的第一個 bit(wnr) 決定讀 (0) 或寫 (1)
- 位址位元的後七個位元為為暫存器位址 (MSB)
- 兩邊的速率 (clock rate) 要一致

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

從暫存器位址對應表開始

6.1. Register Table Summary

Table 41 Registers Summary

Address	Register Name		Reset (POR)	Default (FSK)	Description	
	FSK/OOK Mode	LoRa™ Mode			FSK Mode	LoRa™ Mode
0x00	RegFifo		0x00		FIFO read/write access	
0x01	RegOpMode		0x01		Operating mode & LoRa™ / FSK selection	
0x02	RegBitrateMsb	Unused	0x1A		Bit Rate setting, Most Significant Bits	
0x03	RegBitrateLsb		0x0B		Bit Rate setting, Least Significant Bits	
0x04	RegFdevMsb		0x00		Frequency Deviation setting, Most Significant Bits	
0x05	RegFdevLsb		0x52		Frequency Deviation setting, Least Significant Bits	
0x06	RegFrfMsb		0x6C		RF Carrier Frequency, Most Significant Bits	
0x07	RegFrfMid		0x80		RF Carrier Frequency, Intermediate Bits	
0x08	RegFrfLsb		0x00		RF Carrier Frequency, Least Significant Bits	
0x09	RegPaConfig		0x4F		PA selection and Output Power control	
0x0A	RegPaRamp		0x09		Control of PA ramp time, low phase noise PLL	
0x0B	RegOcp		0x2B		Over Current Protection control	
0x0C	RegLna		0x20		LNA settings	
0x0D	RegRxConfig	RegFifoAddrPtr	0x08	0x0E	AFC, AGC, ctrl	FIFO SPI pointer
0x0E	RegRssiConfig	RegFifoTxBa-seAddr	0x02		RSSI	Start Tx data

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

現在是 FSK 還是 LoRa 模式？

FSK/OOK Mode Register Map

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
Registers for Common settings					
RegOpMode (0x01)	7	LongRangeMode	r	0x00	0 → FSK/OOK Mode 1→ LoRa™ Mode This bit can be modified only in Sleep mode. A write operation on other device modes is ignored.
	6-5	ModulationType	rw	0x00	Modulation scheme: 00 → FSK 01 → OOK 10 → 11 → reserved
	4	reserved	r	0x0	reserved
	3	LowFrequencyModeOn	rw	0x01	Access Low Frequency Mode registers (from address 0x61 on) 0 → High Frequency Mode (access to HF test registers) 1 → Low Frequency Mode (access to LF test registers)
	2-0	Mode	rw	0x01	Transceiver modes 000 → Sleep mode 001 → Stdby mode 010 → FS mode TX (FSTx) 011 → Transmitter mode (Tx) 100 → FS mode RX (FSRx) 101 → Receiver mode (Rx) 110 → reserved 111 → reserved

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

LoRa Mode Register Map

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
Common Register Settings					
RegOpMode (0x01)	7	LongRangeMode	rw	0x0	0 → FSK/OOK Mode 1 → LoRa™ Mode This bit can be modified only in Sleep mode. A write operation on other device modes is ignored.
	6	AccessSharedReg	rw	0x0	This bit operates when device is in Lora mode; if set it allows access to FSK registers page located in address space (0x0D:0x3F) while in LoRa mode 0 → Access LoRa registers page 0x0D: 0x3F 1 → Access FSK registers page (in mode LoRa) 0x0D: 0x3F
	5-4	reserved	r	0x00	reserved
	3	LowFrequencyModeOn	rw	0x01	Access Low Frequency Mode registers 0 → High Frequency Mode (access to HF test registers) 1 → Low Frequency Mode (access to LF test registers)
	2-0	Mode	rwt	0x01	Device modes 000 → SLEEP 001 → STDBY 010 → Frequency synthesis TX (FSTX) 011 → Transmit (TX) 100 → Frequency synthesis RX (FSRX) 101 → Receive continuous (RXCONTINUOUS) 110 → receive single (RXSINGLE) 111 → channel activity detection (CAD)

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

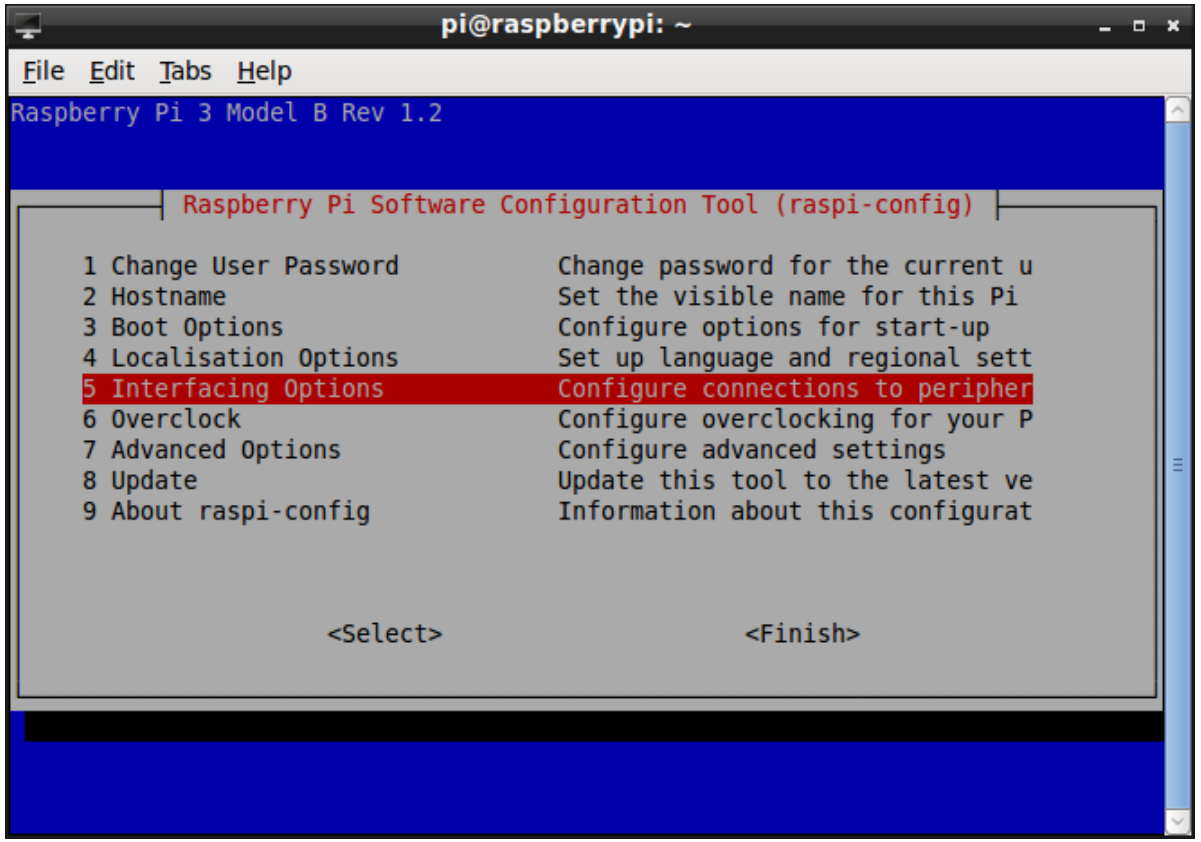
透過 SPI 讀取暫存器數值

```
import spidev

spi = spidev.SpiDev()
spi.open(0, 0)
spi.max_speed_hz = 5000000

RegOpMode = 0x01
Value     = 0

# Read
ret = spi.xfer([RegOpMode & 0x7F, Value])[1]
print(ret)          # 128
print(ret >> 7)     # 1

spi.close()
```

0|||||||

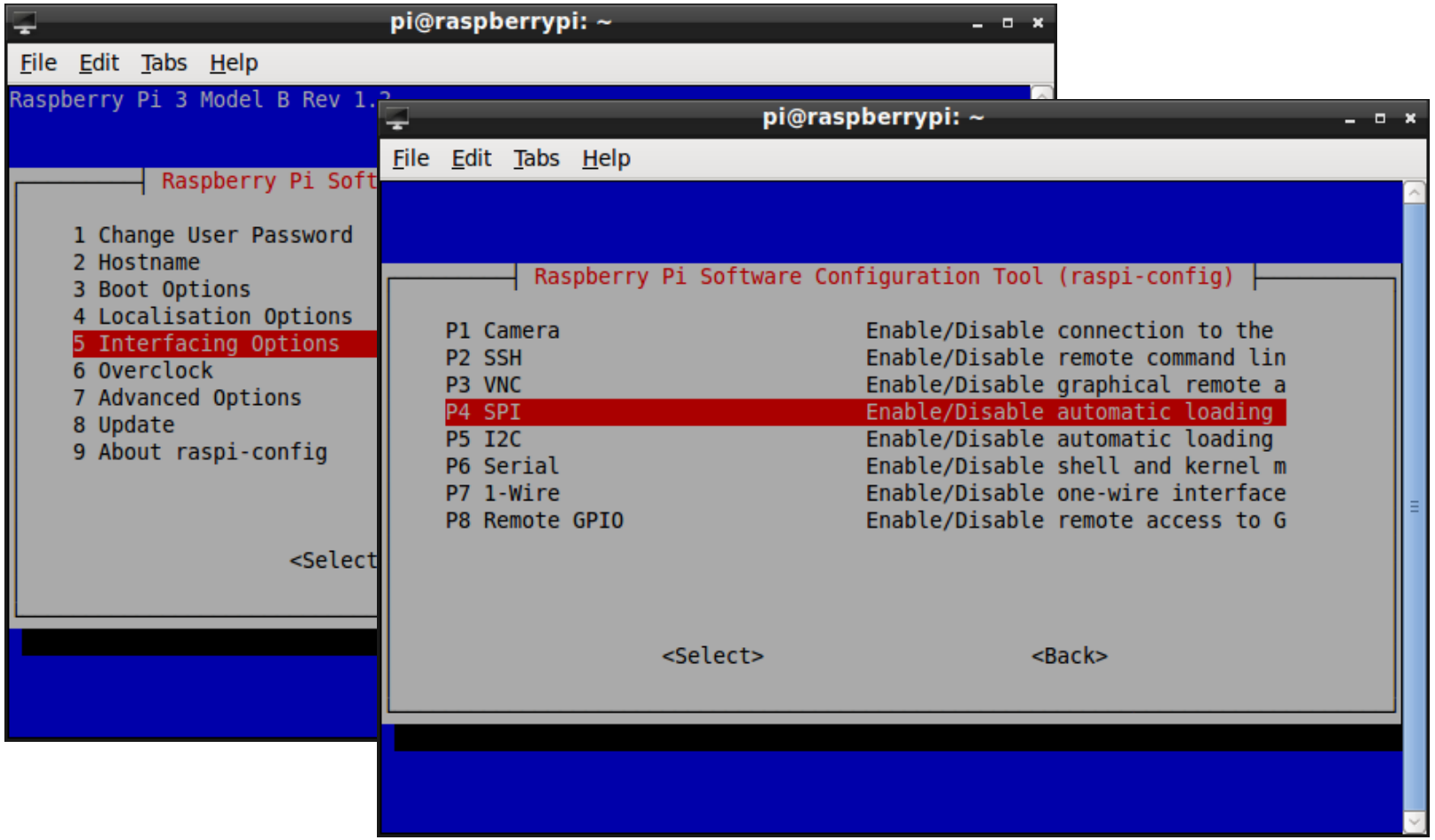
啟用 Raspberry Pi 的 SPI

- `$ sudo raspi-config`



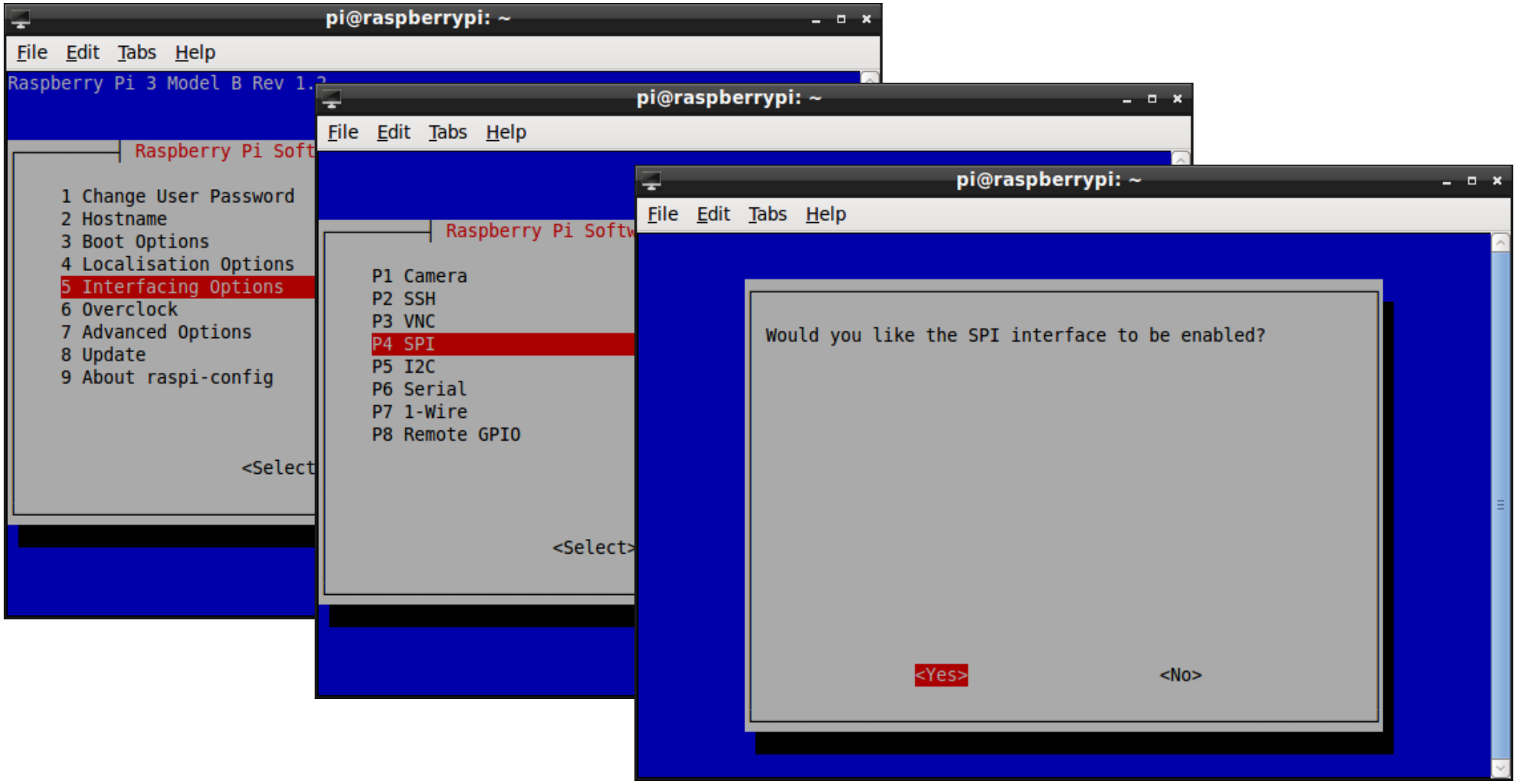
啟用 Raspberry Pi 的 SPI

- `$ sudo raspi-config`



啟用 Raspberry Pi 的 SPI

- `$ sudo raspi-config`



DEMO
get_regopmode.py

```
$ cd ~/lora-sx1276/01-register
$ python3 get_regopmode.py
```

那目前 LoRa 使用的 Frequency ？

從規格書找出暫存器位址

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegFrMsb (0x06)	7-0	Frf(23:16)	rw	0x6c	MSB of RF carrier frequency
RegFrMid (0x07)	7-0	Frf(15:8)	rw	0x80	MSB of RF carrier frequency
RegFrLsb (0x08)	7-0	Frf(7:0)	rwt	0x00	LSB of RF carrier frequency $f_{RF} = \frac{F(XOSC) \cdot Frf}{2^{19}}$ Resolution is 61.035 Hz if F(XOSC) = 32 MHz. Default value is 0x6c8000 = 434 MHz. Register values must be modified only when device is in SLEEP or STAND-BY mode.

- RF 頻率共分為三個暫存器儲存資料
  - MSB:0x06
  - MID:0x07
  - LSB:0x08

0x6c8000 = 434MHz
7110656/434=16384
轉換數值為 16384

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

Frequency 是三個暫存器數值的加總

```
import spidev

spi = spidev.SpiDev()
spi.open(0, 0)
spi.max_speed_hz = 5000000

RegFrMsb = 0x06
RegFrMid = 0x07
RegFrLsb = 0x08
Value    = 0

msb = spi.xfer([RegFrMsb & 0x7F, Value])[1]
mid = spi.xfer([RegFrMid & 0x7F, Value])[1]
lsb = spi.xfer([RegFrLsb & 0x7F, Value])[1]
f = lsb + 256*(mid + 256*msb)
print(f / 16384.0)

spi.close()
```

DEMO
get_regfr.py

```
$ cd ~/lora-sx1276/01-register
$ python3 get_regfr.py
```

怎麼修改 Frequency 的數值？

Frequency 是三個暫存器數值的加總

```
RegFrMsb = 0x06

def set_freq(f):
    i = int(f * 16384.)     # choose floor
    msb = i // 65536
    i -= msb * 65536
    mid = i // 256
    i -= mid * 256
    lsb = i
    return spi.xfer([RegFrMsb | 0x80, msb, mid, lsb])

set_freq(868)
```

10000000

位址的第一個 bit=1 為寫入

DEMO
set_regfr.py

```
$ cd ~/lora-sx1276/01-register
$ python3 set_regfr.py
```

練習

- 如何取得目前使用的 Spreading Factor?

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegModemConfig 2 (0x1E)	7-4	SpreadingFactor	rw	0x07	SF rate (expressed as a base-2 logarithm) 6 → 64 chips / symbol 7 → 128 chips / symbol 8 → 256 chips / symbol 9 → 512 chips / symbol 10 → 1024 chips / symbol 11 → 2048 chips / symbol 12 → 4096 chips / symbol other values reserved.
	3	TxContinuousMode	rw	0	0 → normal mode, a single packet is sent 1 → continuous mode, send multiple packets across the FIFO (used for spectral analysis)
	2	RxPayloadCrcOn	rw	0x00	Enable CRC generation and check on payload: 0 → CRC disable 1 → CRC enable If CRC is needed, RxPayloadCrcOn should be set: - in Implicit header mode: on Tx and Rx side - in Explicit header mode: on the Tx side alone (recovered from the header in Rx side)
	1-0	SymbTimeout(9:8)	rw	0x00	RX Time-Out MSB

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

如何實現最簡單的 P2P ？

先備知識

- 硬體初始化（模式切換）
- 觸發(DIO mapping)
- 資料搬移（傳送與接收）

http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

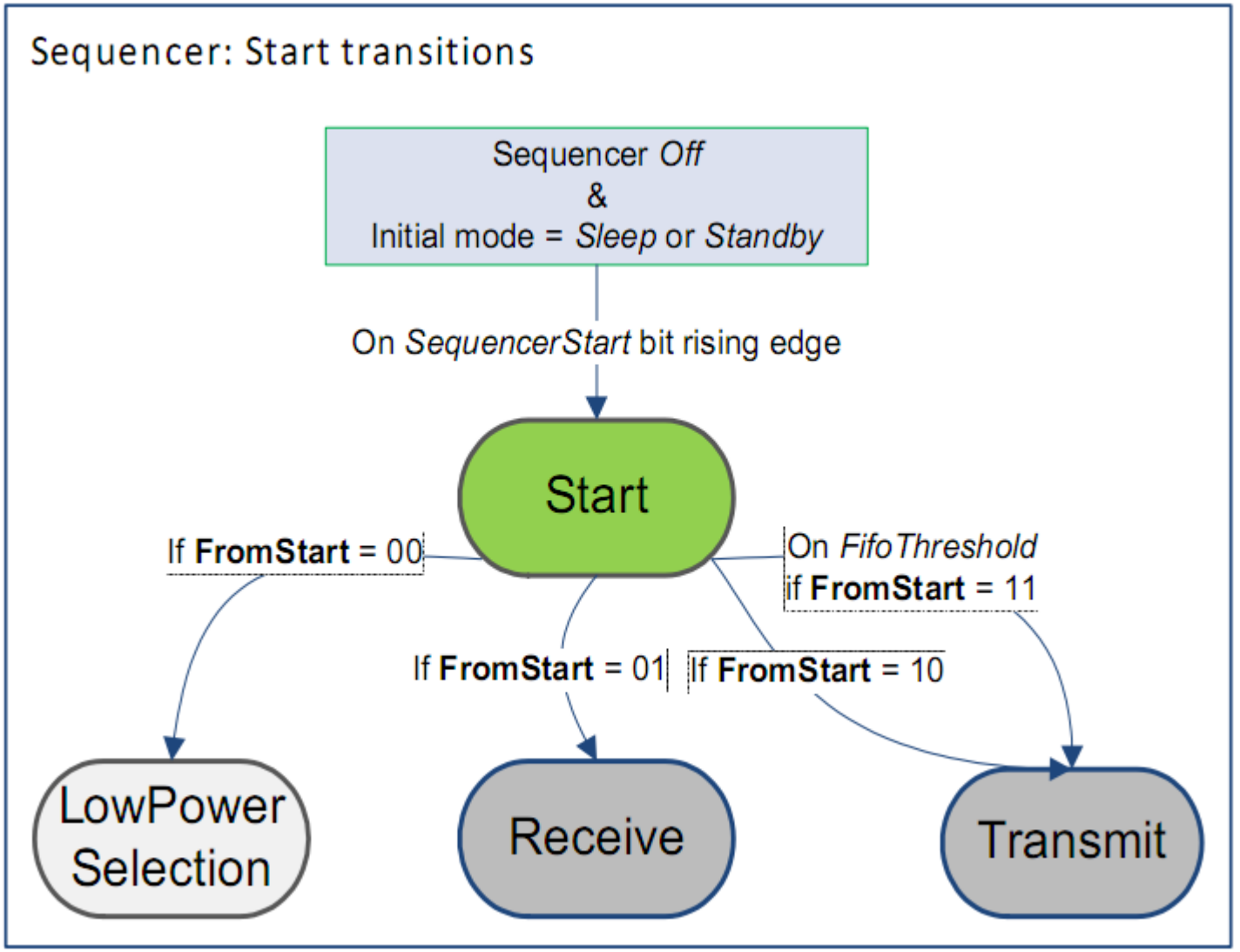
硬體初始化 (模式切換)

- 需要從 SLEEP Mode 轉成 RXCONTINUOUS Mode

Operating Mode	Description
SLEEP	Low-power mode. In this mode only SPI and configuration registers are accessible. Lora FIFO is not accessible. Note that this is the only mode permissible to switch between FSK/OOK mode and LoRa mode.
STANDBY	both Crystal oscillator and Lora baseband blocks are turned on.RF part and PLLs are disabled
FSTX	This is a frequency synthesis mode for transmission. The PLL selected for transmission is locked and active at the transmit frequency. The RF part is off.
FSRX	This is a frequency synthesis mode for reception. The PLL selected for reception is locked and active at the receive frequency. The RF part is off.
TX	When activated the SX1276/77/78/79 powers all remaining blocks required for transmit, ramps the PA, transmits the packet and returns to Standby mode.
RXCONTINUOUS	When activated the SX1276/77/78/79 powers all remaining blocks required for reception, processing all received data until a new user request is made to change operating mode.
RXSINGLE	When activated the SX1276/77/78/79 powers all remaining blocks required for reception, remains in this state until a valid packet has been received and then returns to Standby mode.
CAD	When in CAD mode, the device will check a given channel to detect LoRa preamble signal

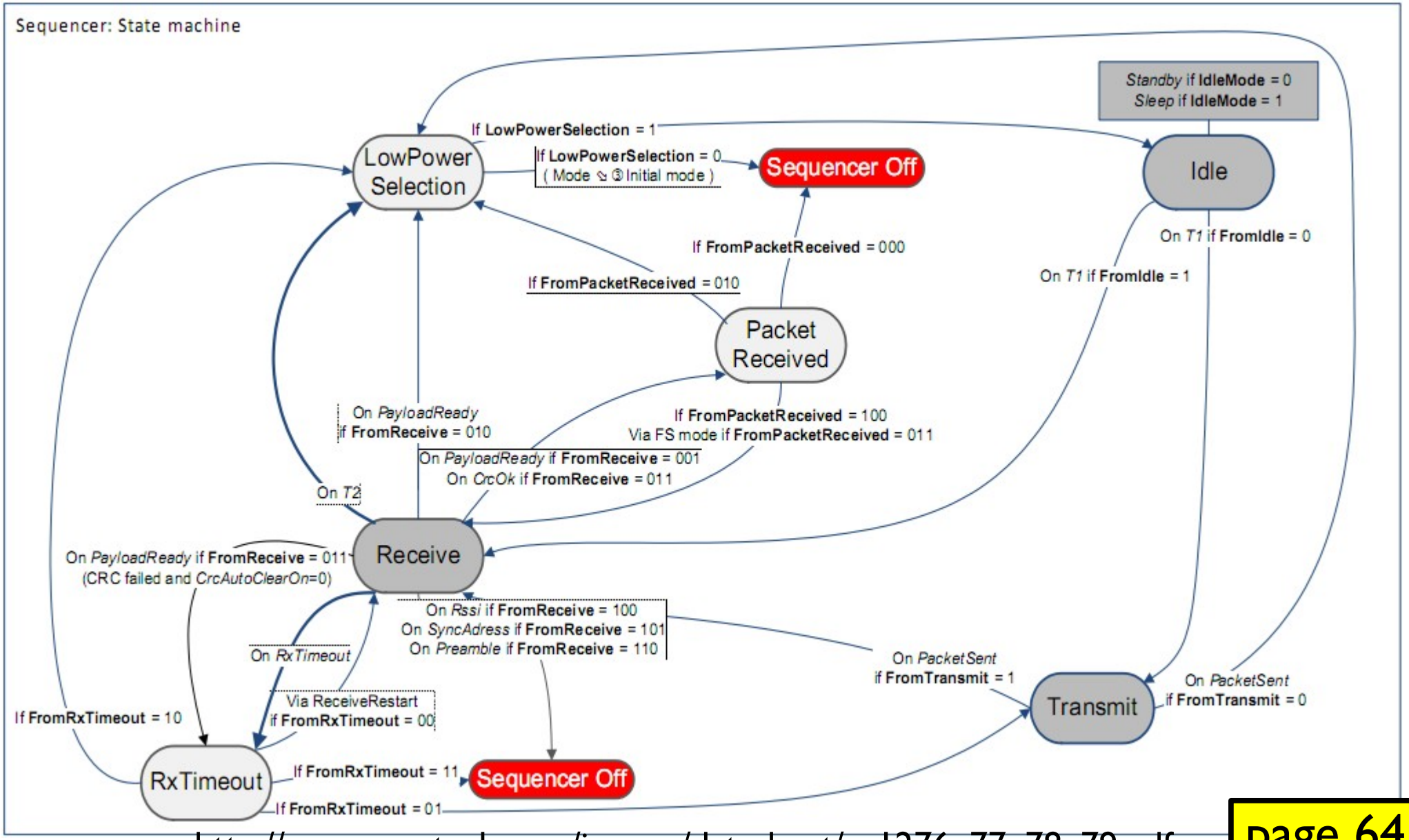
http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

狀態機 (State Machine)



http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

狀態轉移



http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

收到 Preamble 後的狀態

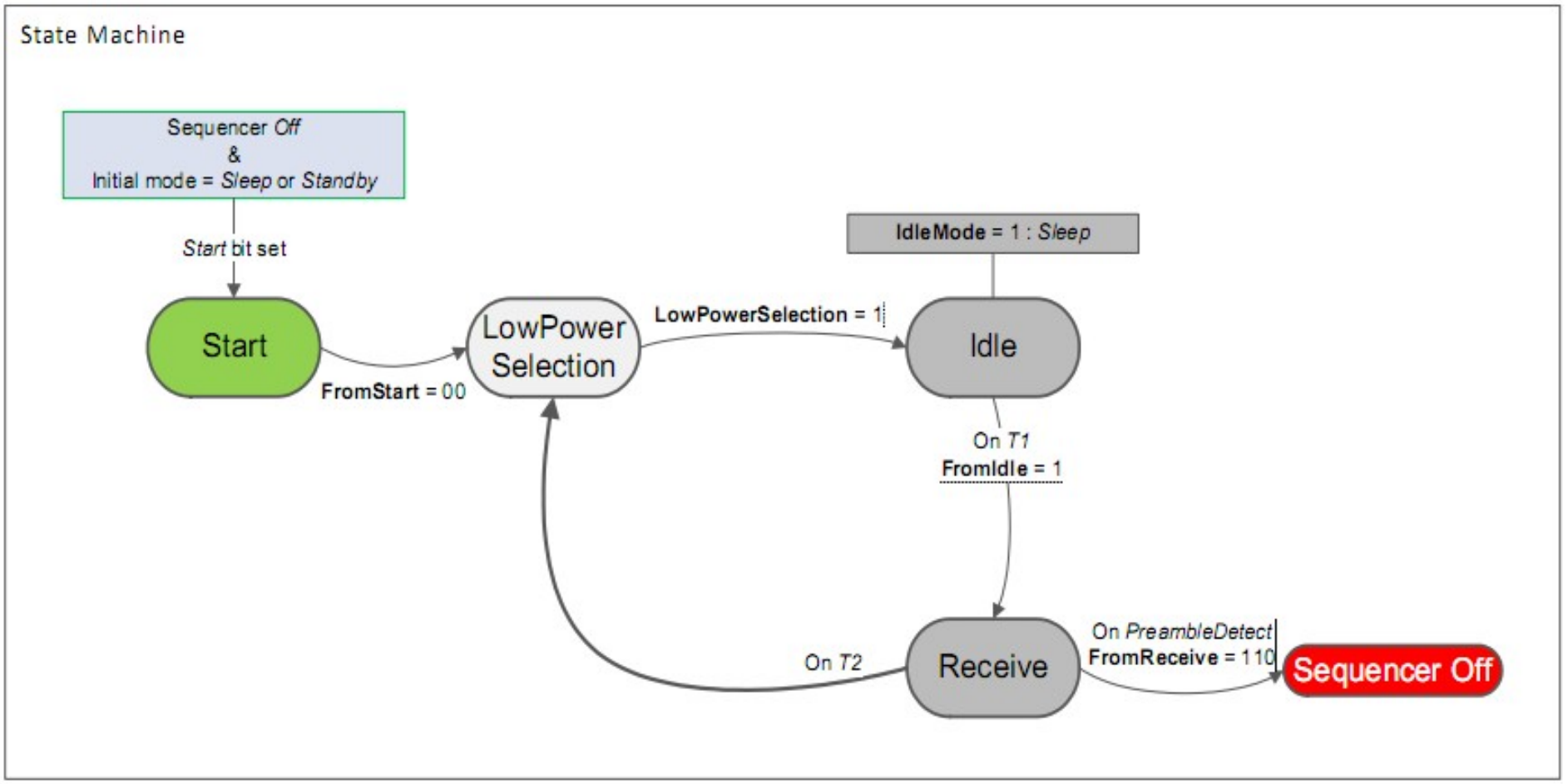
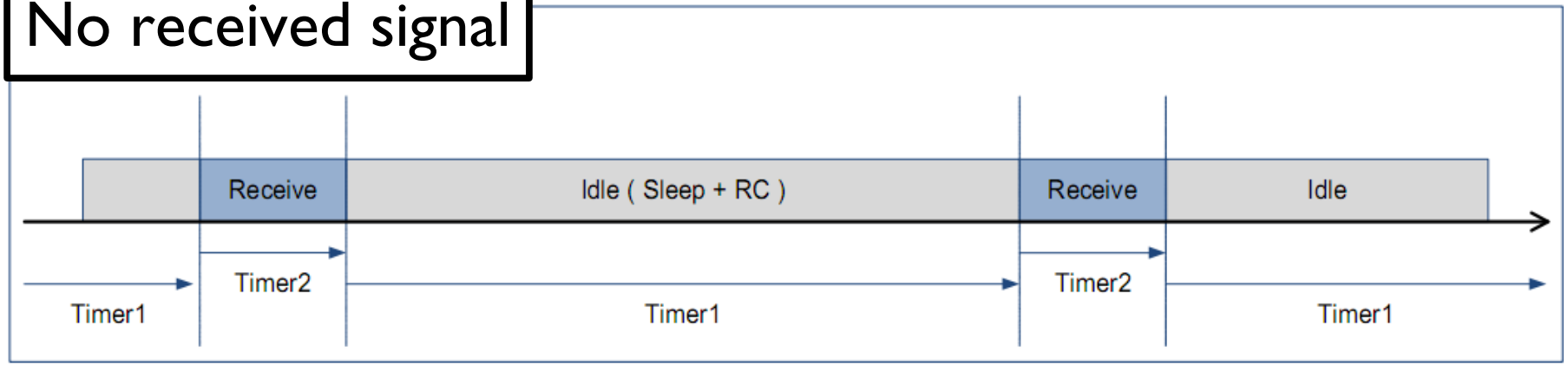


Figure 47. Wake On PreambleDetect State Machine

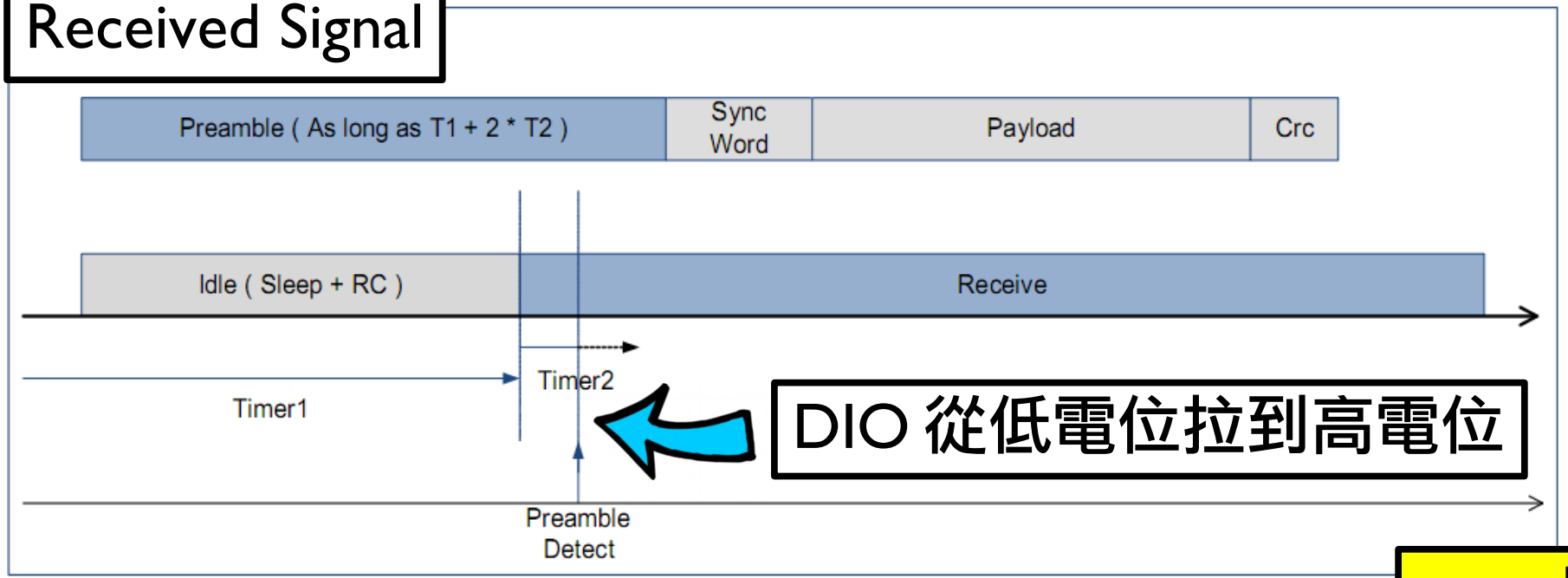
http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

觸發 (DIO mapping)

No received signal



Received Signal



http://www.semtech.com/images/datasheet/sx1276_77_78_79.pdf

資料搬移 (傳送與接收)

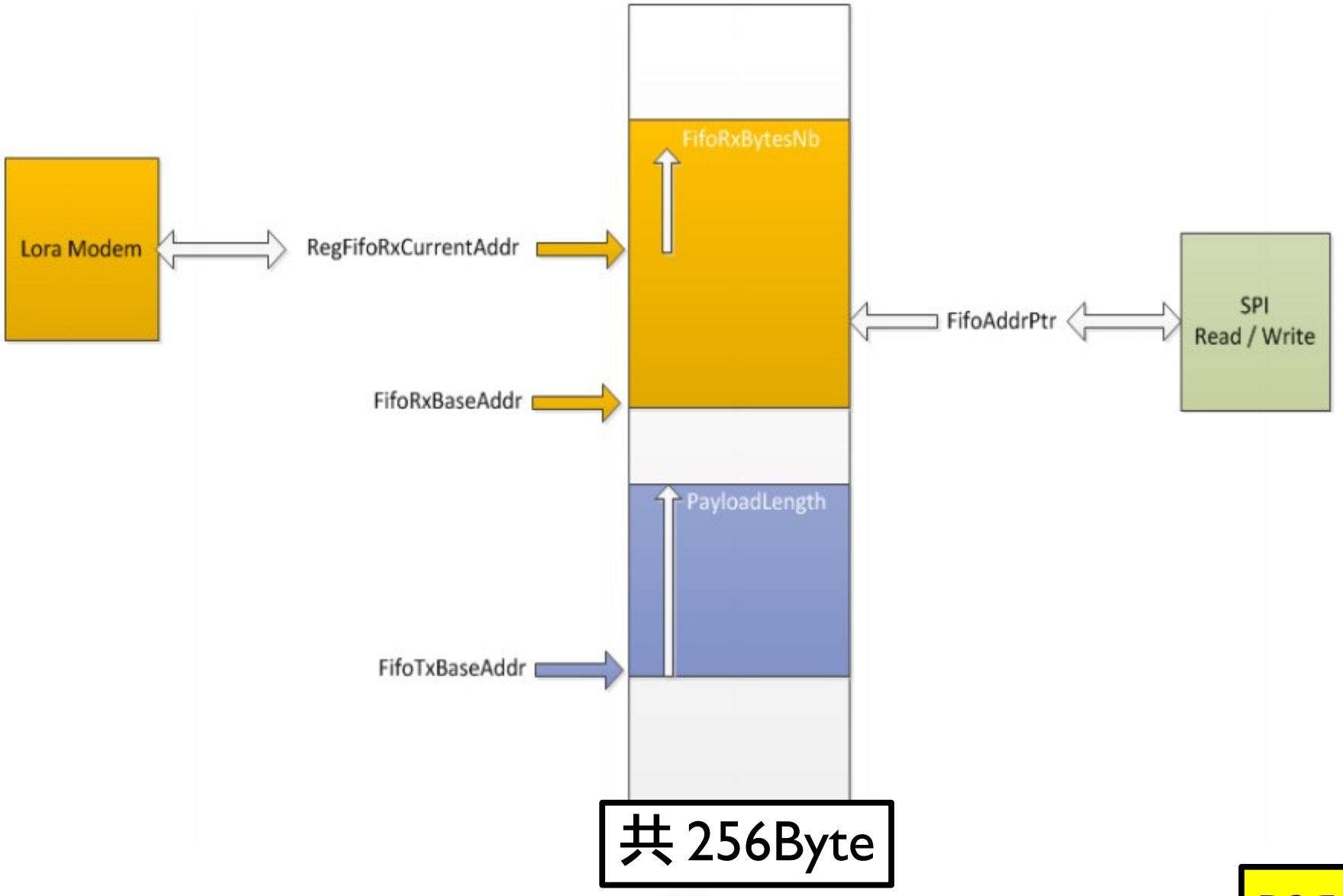


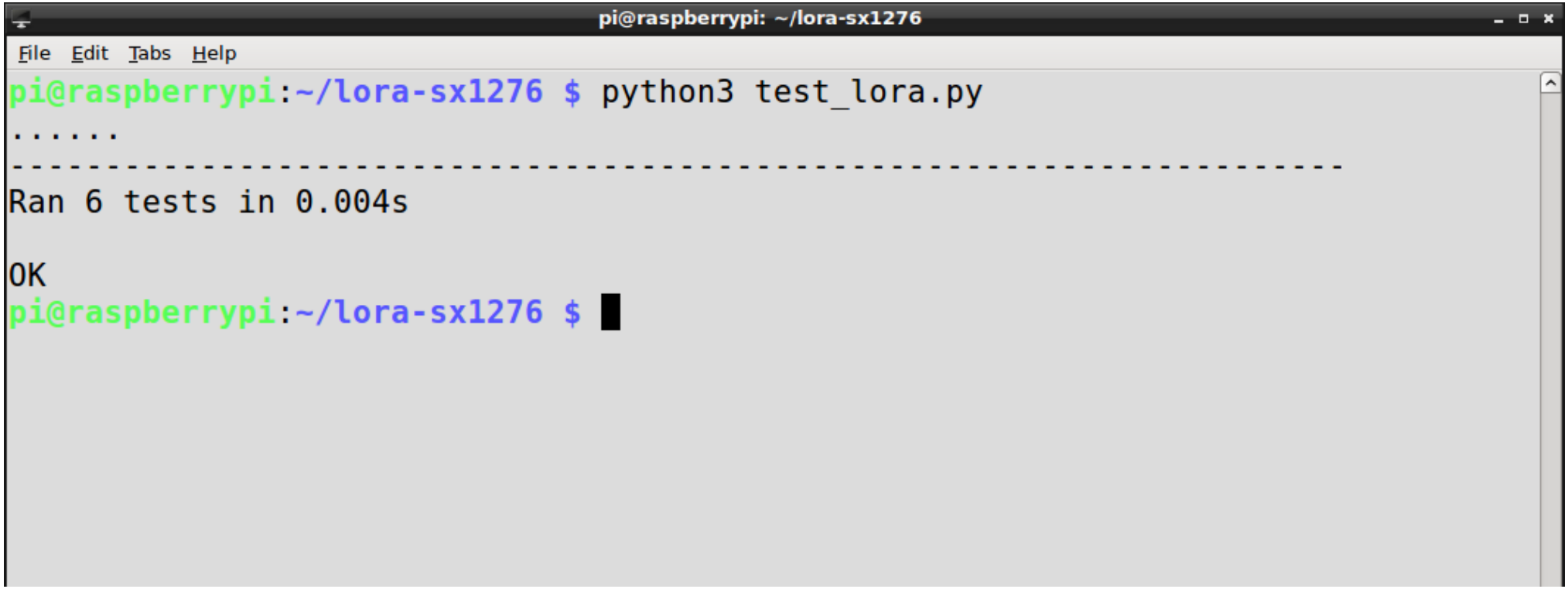
Figure 8. LoRa™ Data Buffer

還好有人幫我們都做好了

A Python Interface to Semtech SX127x

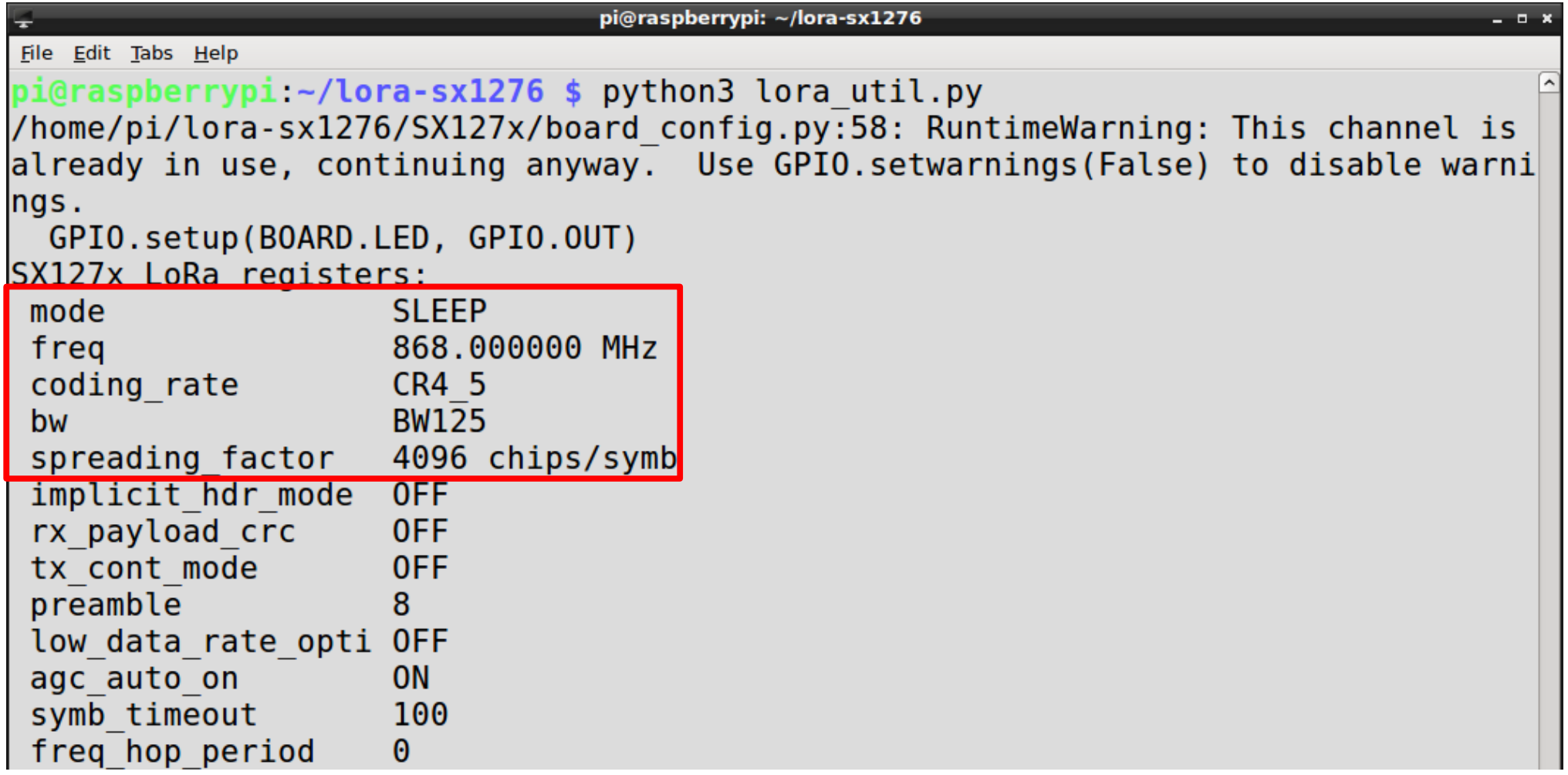
- 先做 unit test 確認接腳正確

```
$ cd ~/lora-sx1276
$ python3 test_lora.py
```



讀取預設參數

```
$ cd ~/lora-sx1276
$ python3 lora_util.py
```



LoRa 成功傳送接收必要條件

1. 相同頻率 (Frequency)
2. 相同頻寬 (Bandwidth)
3. 相同展頻因子 (Spreading Factor)

- 注：不同 BW 和 SF 也可能因為不正交而收到

測試連續接收 (用 Python2 看結果)

```
$ cd ~/lora-sx1276
$ python rx_cont.py -f 868 -b BW125 -s 12
```

```
usage: rx_cont.py [-h] [--ocp OCP] [--sf SF] [--freq FREQ] [--bw BW]
                  [--cr CODING_RATE] [--preamble PREAMBLE]

Continous LoRa receiver

optional arguments:
  -h, --help            show this help message and exit
  --ocp OCP, -c OCP     Over current protection in mA (45 .. 240 mA)
  --sf SF, -s SF        Spreading factor (6...12). Default is 7.
  --freq FREQ, -f FREQ  Frequency
  --bw BW, -b BW        Bandwidth (one of BW7_8 BW10_4 BW15_6 BW20_8 BW31_25
                        BW41_7 BW62_5 BW125 BW250 BW500). Default is BW125.
  --cr CODING_RATE, -r CODING_RATE
                        Coding rate (one of CR4_5 CR4_6 CR4_7 CR4_8). Default
                        is CR4_5.
  --preamble PREAMBLE, -p PREAMBLE
                        Preamble length. Default is 8.
```

https://github.com/mayeranalytics/pySX127x

測試連續發送 (用 Python2 看結果)

```
$ cd ~/lora-sx1276
$ python tx_beacon.py -f 868 -b BW125 -s 12
```

```
usage: tx_beacon.py [-h] [--ocp OCP] [--sf SF] [--freq FREQ] [--bw BW]
                    [--cr CODING_RATE] [--preamble PREAMBLE] [--single]
                    [--wait WAIT]

A simple LoRa beacon

optional arguments:
  -h, --help            show this help message and exit
  --ocp OCP, -c OCP     Over current protection in mA (45 .. 240 mA)
  --sf SF, -s SF        Spreading factor (6...12). Default is 7.
  --freq FREQ, -f FREQ  Frequency
  --bw BW, -b BW        Bandwidth (one of BW7_8 BW10_4 BW15_6 BW20_8 BW31_25
                        BW41_7 BW62_5 BW125 BW250 BW500). Default is BW125.
  --cr CODING_RATE, -r CODING_RATE
                        Coding rate (one of CR4_5 CR4_6 CR4_7 CR4_8). Default
                        is CR4_5.
  --preamble PREAMBLE, -p PREAMBLE
                        Preamble length. Default is 8.
  --single, -S          Single transmission
  --wait WAIT, -w WAIT  Waiting time between transmissions (default is 0s)
```

https://github.com/mayeranalytics/pySX127x

一端接收另一端發送

rx_cont.py

```
pi@raspberrypi: ~/lo
File  Edit  Tabs  Help
 fifo_rx_base_addr  0x0
 fifo_rx_curr_addr  0x0
 fifo_rx_byte_addr  0x0
 status             {'signal_sync': 0, 'sig
: 0, 'rx_ongoing': 0, 'modem_clear': 1, 'rx
 version            0x12

Press enter to start...
-96 1 00
RxDon
[15]
-93 1 00
RxDone
[15]
-93 1 00
RxDone
[15]
-92   00
RxDo
[15]
-93 1 00
RxDone
[15]
```

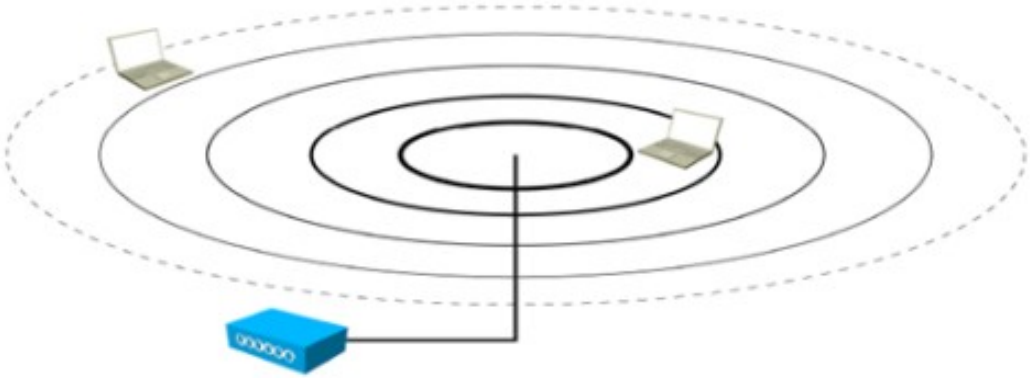
-96 I 00
RSSI RX_STATUS MODEM_CLR

[15]

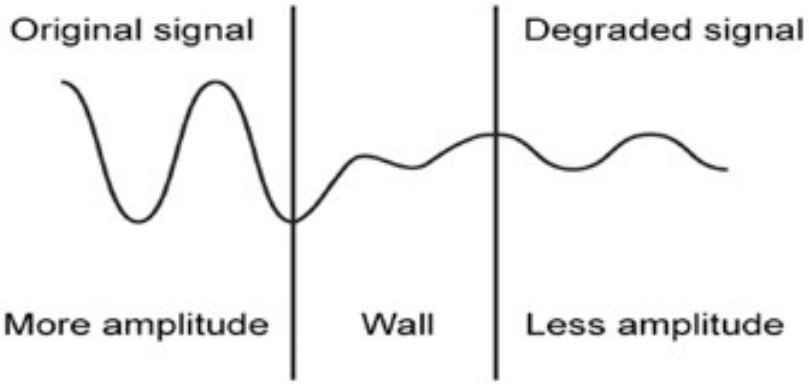
tx_beacon.py

```
pi@raspberrypi: ~/lo
File  Edit  Tabs  Help
 lna_gain           NOT_USED
 lna_boost_lf       0b0
 lna_boost_hf       0b0
 detect_optimize    0x3
 detection_thresh   0xa
 sync_word          0x12
 dio_mapping 0..5   [1, 0, 0, 0, 0, 0]
 tcxo               XTAL
 pa_dac             default
 fifo_addr_ptr      0x81
                    0x80
                    0x0
                    0x0
                    0x0
 status             {'signal_sync': 0, 'si
: 0, 'rx_ongoing': 0, 'modem_clear': 0, 'r
 version            0x12

Beacon config:
  Wait 1.000000 s
  Single tx = False

Press enter to start...
tx #5
```

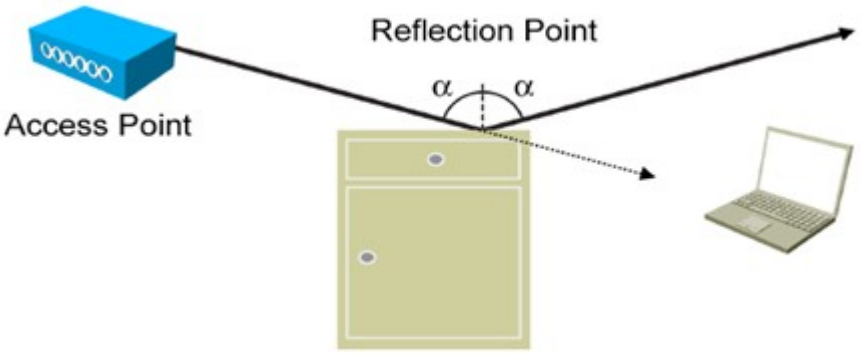
訊號傳送和損失



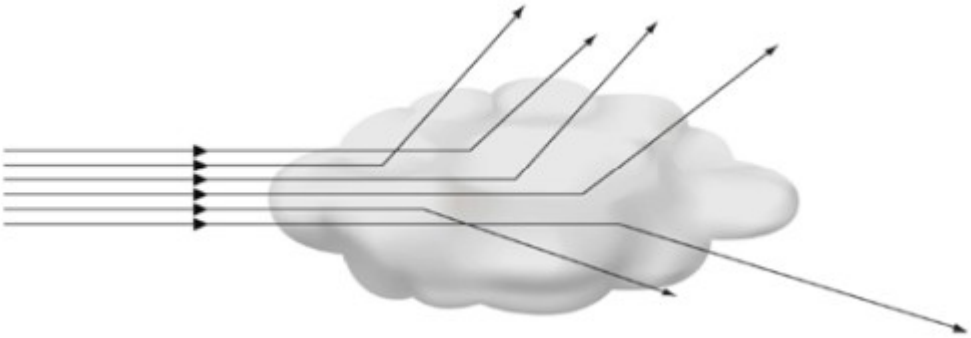
自由空間路徑損耗



吸收 (Absorption)



反射 (Reflection)



散射 (Scattering)

https://rscciew.wordpress.com/2013/11/24/rf-principle-in-wireless-network/

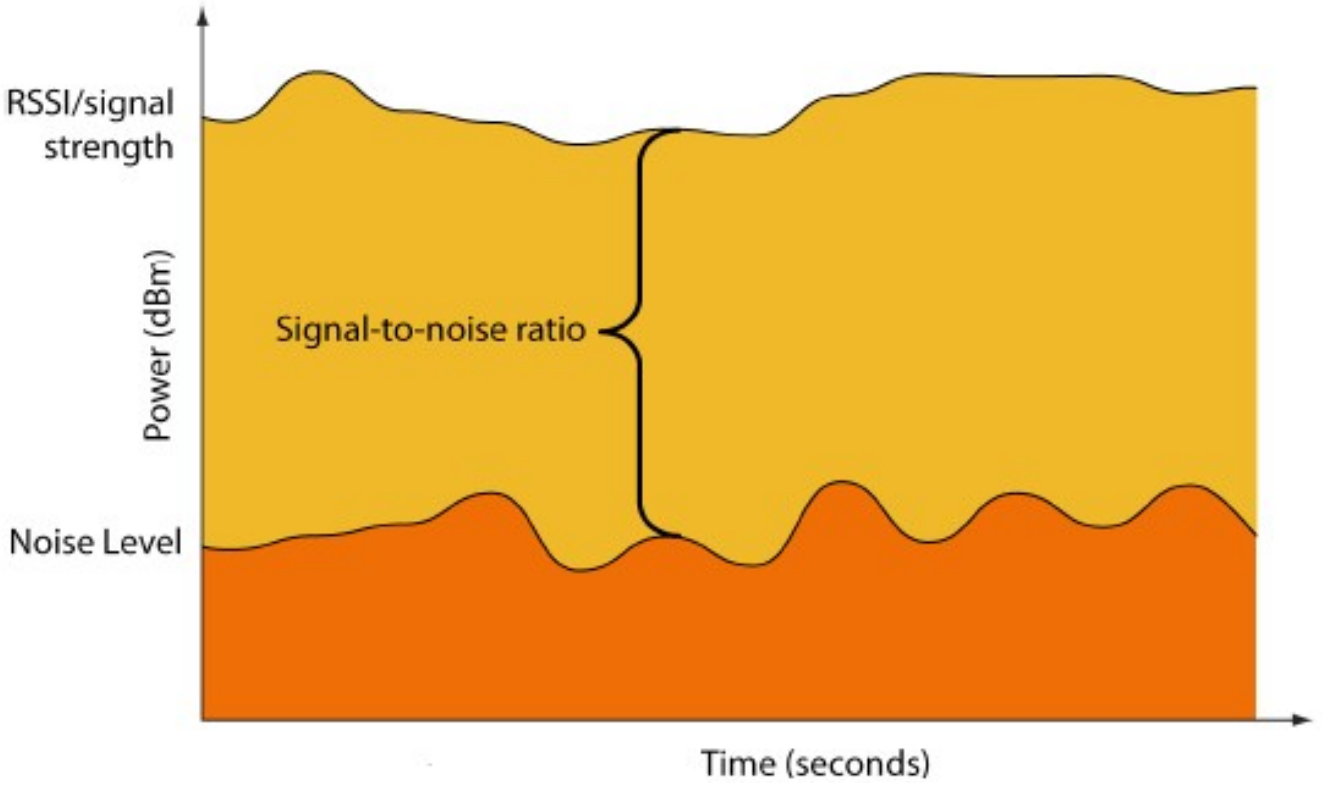
先備知識

- 通訊領域常用 dB,dBm 表示相對強度或功率
- 對數相加減 = 常數相乘除

- dB ≡ 10 × log10(ratio), 純量無單位
  - 因此 10 倍增加標示為 +10dB, 而 1/100 標示為 -20dB
  - 如果是 2 倍增加，標示為 +3dB(應為 +3.01dB)

- dBm ≡ 10 × log10(Power)1mW
  - 因此 0.02 Watt 是 20mW, 也是 13dBm
  - 可以從 10dBm(10mW)+3dBm(2mW) 計算得來

RSSI and SNR



- RSSI(Received Signal Strength Indicator)
- SNR(Signal to Noise Ratio)
- RSSI 是負數(dBm),SNR 通常是正數(dBm)
- RSSI 或是 SNR 數值越大表示訊號強度越好

https://rscciew.wordpress.com/2013/11/24/rf-principle-in-wireless-network/

tx_beacon.py

```
    def on_tx_done(self):
        global args
        self.set_mode(MODE.STDBY)
        self.clear_irq_flags(TxDone=1)
        sys.stdout.flush()
        self.tx_counter += 1
        sys.stdout.write("\rtx #%d" % self.tx_counter)
        if args.single:
            print
            sys.exit(0)
        sleep(args.wait)
        self.write_payload([0x0f])
        BOARD.led_on()
        self.set_mode(MODE.TX)
```

傳送 HEX 的 0x0F

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	|
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

asciichars.com

http://www.asciichars.com/

自由輸入字串發送 (p2p_send.py)

```
    def on_tx_done(self):
        global args
        self.set_mode(MODE.STDBY)
        self.clear_irq_flags(TxDone=1)
        sys.stdout.flush()
        self.tx_counter += 1
        sys.stdout.write("\rtx #%d" % self.tx_counter)
        if args.single:
            print
            sys.exit(0)
        sleep(args.wait)
        rawinput = input(">>> ")
        data = [int(hex(ord(c)), 0) for c in rawinput]
        self.write_payload(data)
        self.set_mode(MODE.TX)
```

將每個字元轉成 ASCII 再轉成 HEX

DEMO
p2p_send.py

```
$ cd ~/lora-sx1276/02-p2p
$ python p2p_send.py -f 868
```

接收字串後解碼 (p2p_recv.py)

```
def on_rx_done(self):
    print("\nRxDone")
    self.clear_irq_flags(RxDone=1)
    payload = self.read_payload(nocheck=True)
    data = ''.join([chr(c) for c in payload])
    print(data)

    self.set_mode(MODE.SLEEP)
    self.reset_ptr_rx()
    self.set_mode(MODE.RXCONT)
```

收到 HEX 後轉成 ASCII 字元

DEMO
p2p_recv.py

```
$ cd ~/lora-sx1276/02-p2p
$ python p2p_recv.py -f 868
```

從 LoRa 到 LoRaWAN

LoRaWAN

- 由 LoRa Alliance 訂定 LoRaWAN 協定
- LoRaWAN: 1.0.0 > 1.0.1 > 1.0.2 > 1.1
- 版本 1.1 在 2017 定稿

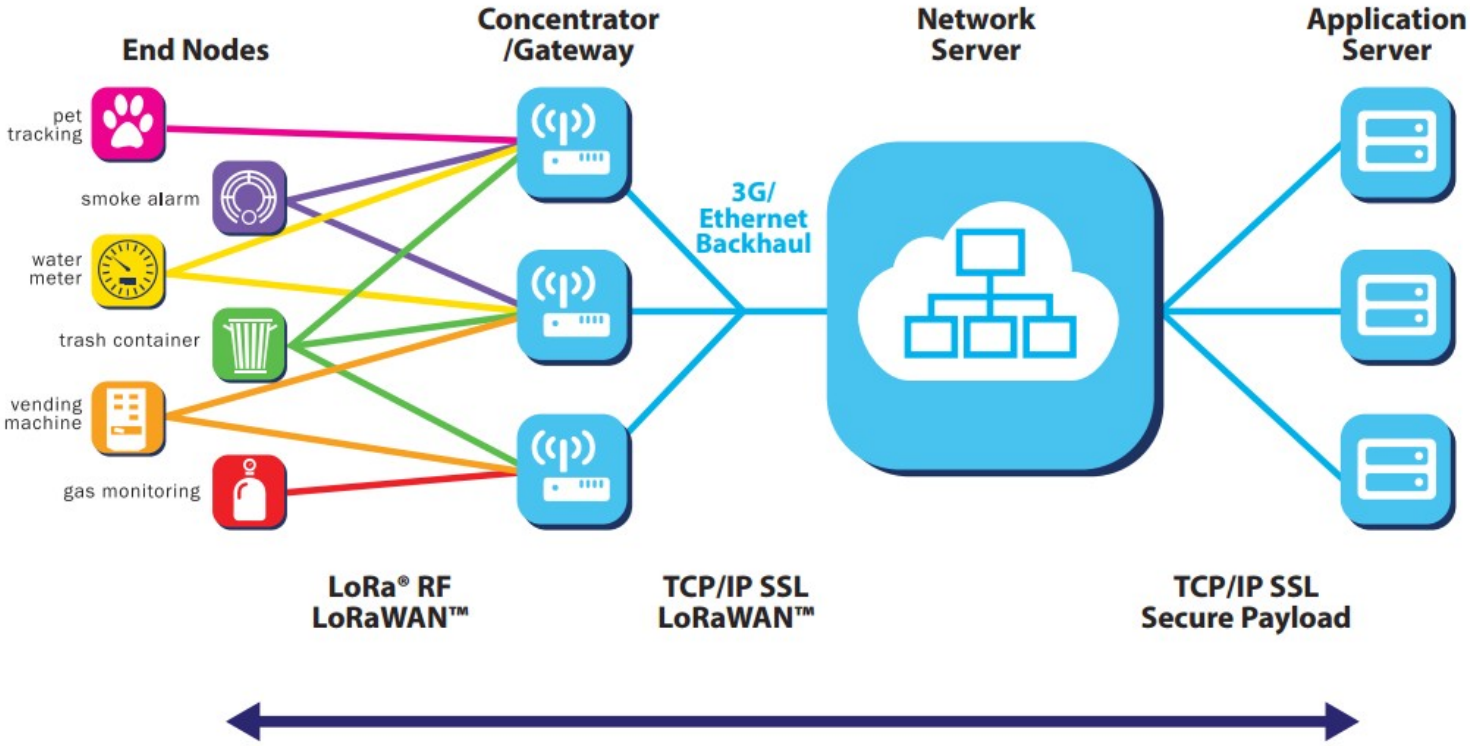
https://www.lora-alliance.org/member-list

名詞解釋

- LoRaWAN
- LoRa Node
- Gateway
- Network Server

LoRaWAN

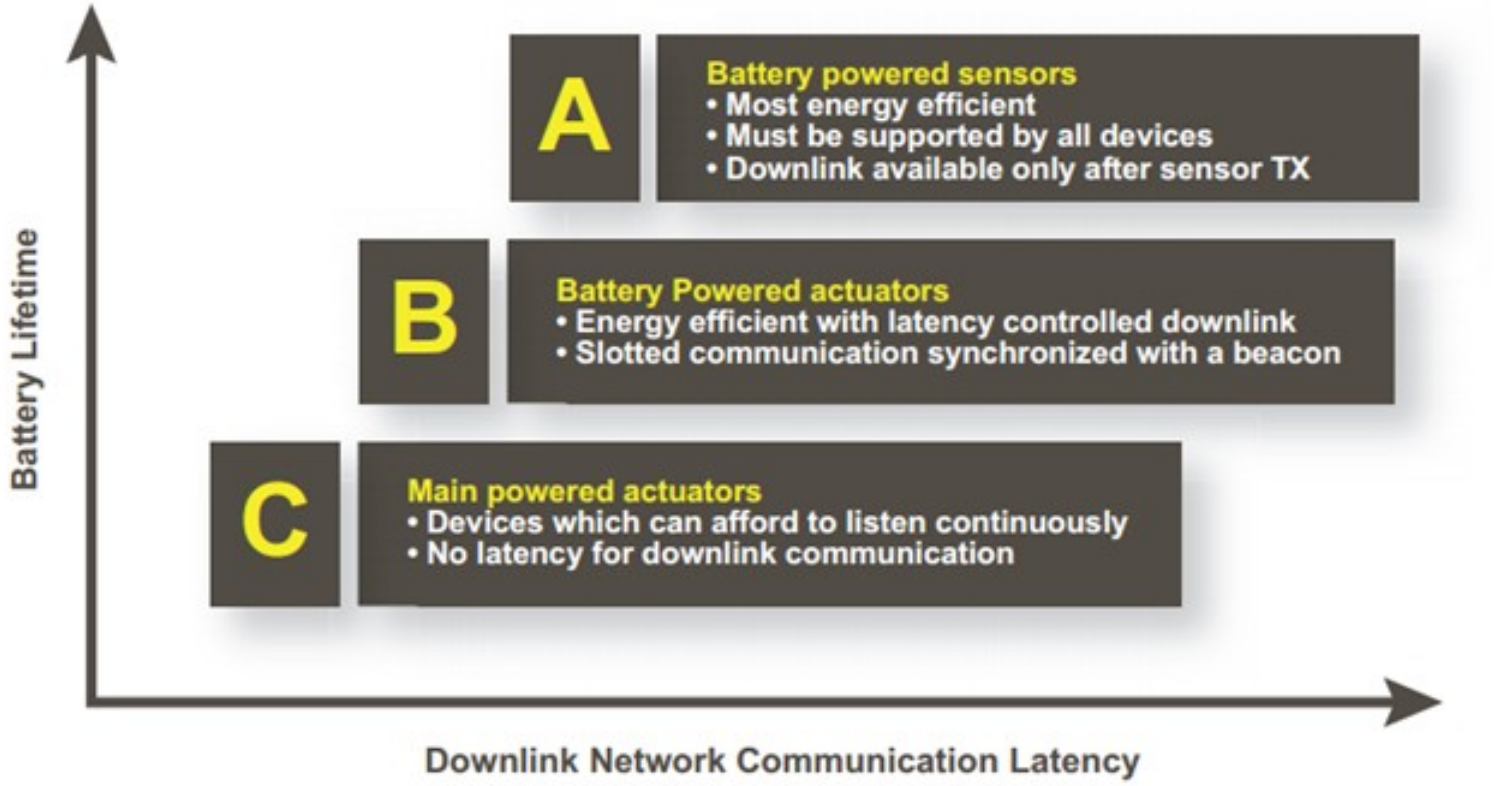
- 定義網路系統架構與通訊協定
- 終端點採 LoRa 做長距離通訊（星狀拓樸）
- 和終端點的是通訊雙向
- AES 加密



http://www.atim.com/en/technologies-2/lorawan/

LoRa Node

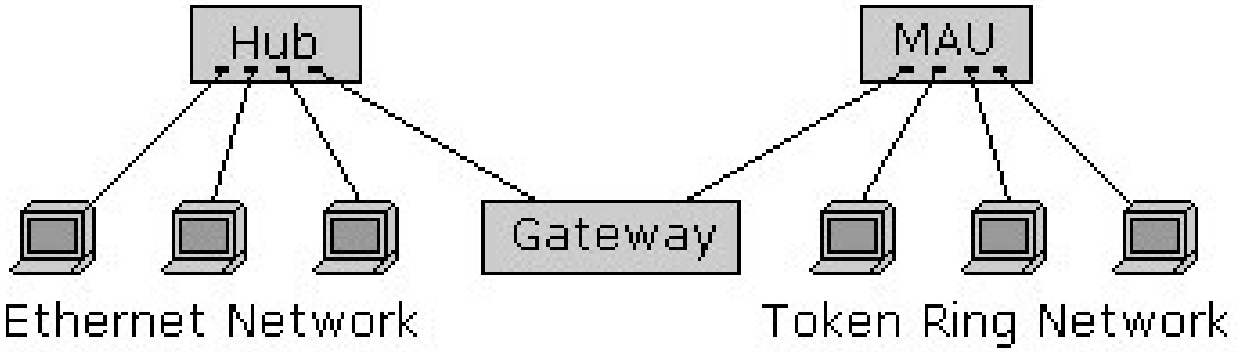
- 定址：DevEUI, AppEUI
- 使用 ALOHA，沒有 CSMA 機制，三種 class



https://www.thethingsnetwork.org/wiki/LoRaWAN/Home

Gateway

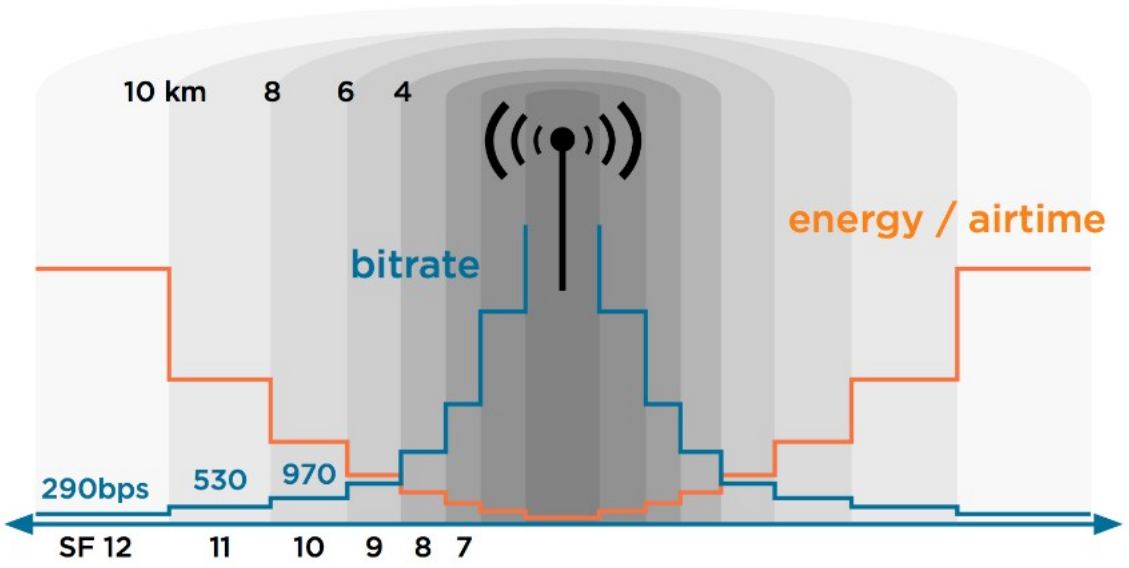
- 傳輸協定轉換 (ex:LoRa + WiFi)
- 接收所有頻道的所有資訊
- 封包直接轉發到後端 (Network Server)



https://en.wikipedia.org/wiki/Gateway_(telecommunications)

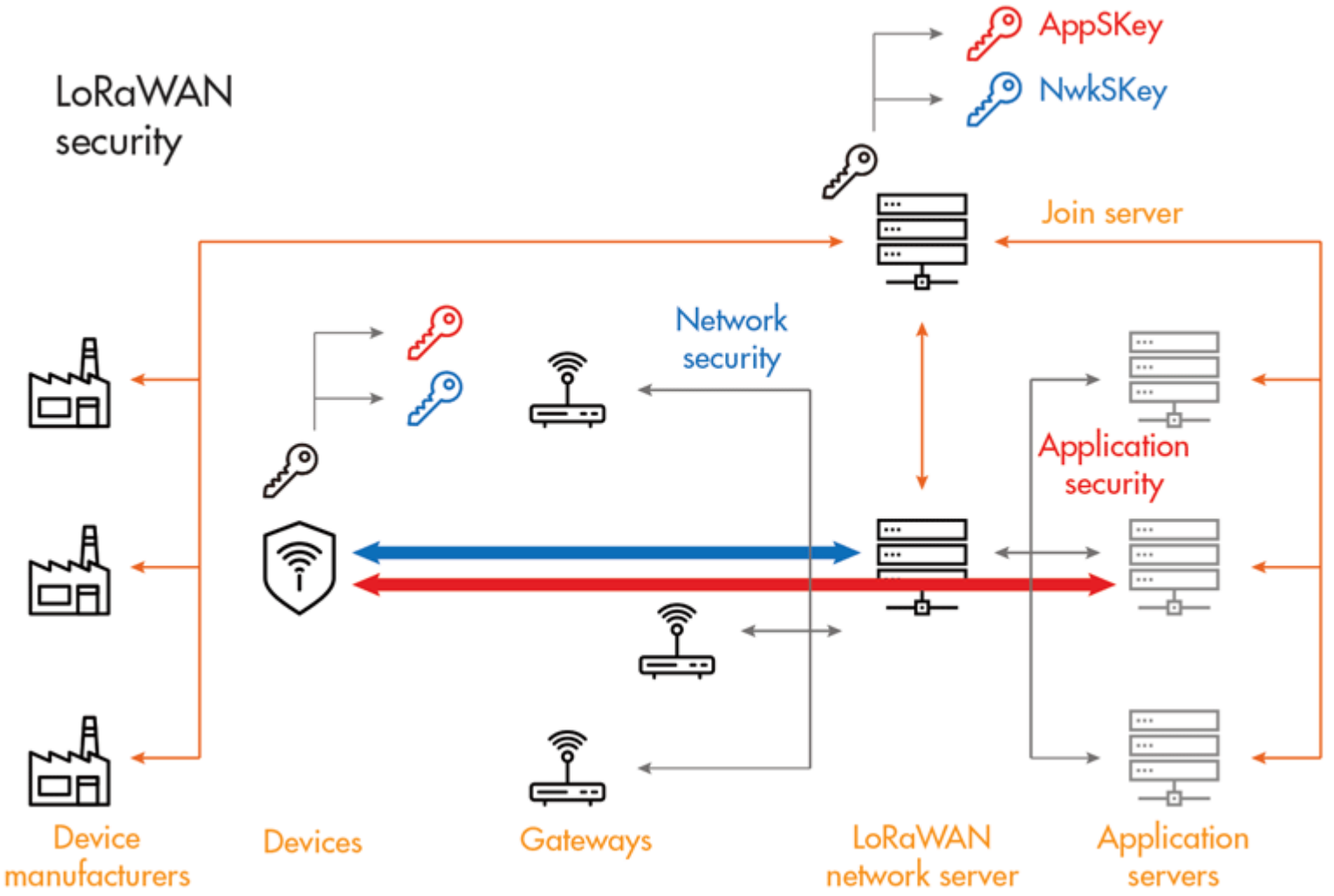
Network Server

- 冗餘封包濾除 (CRC)
- 安全性查驗 (Authentication/Authorization
- 適應性資料率 (Adaptive Data Rate, ADR)



http://docs.wixstatic.com/ugd/eccc1a_20fe760334f84a9788c5b11820281bd0.pdf

LoRaWAN Security

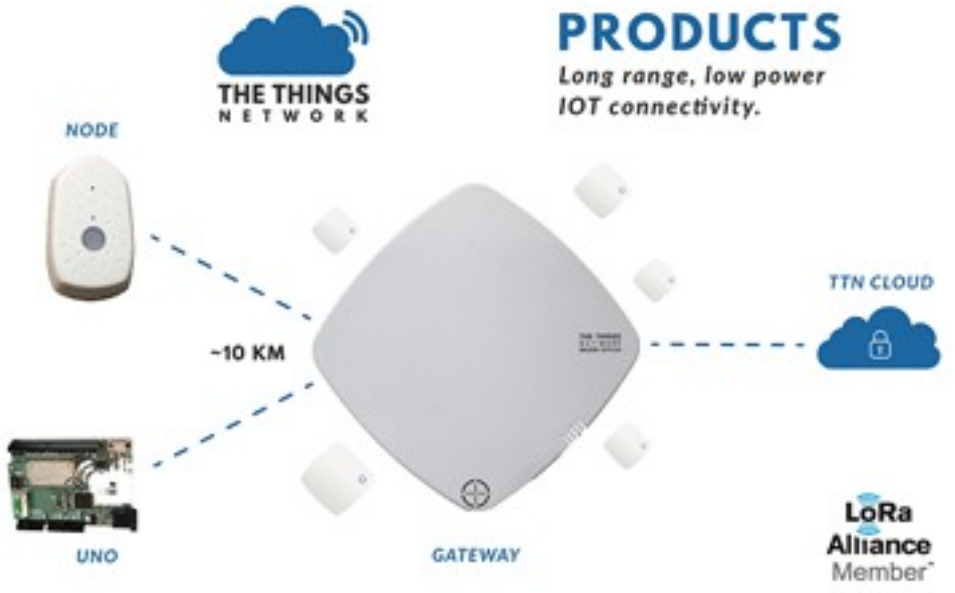


http://www.electronicdesign.com/embedded-revolution/lora-rolls-philly

使用公用的 LoRaWAN

The Things Network

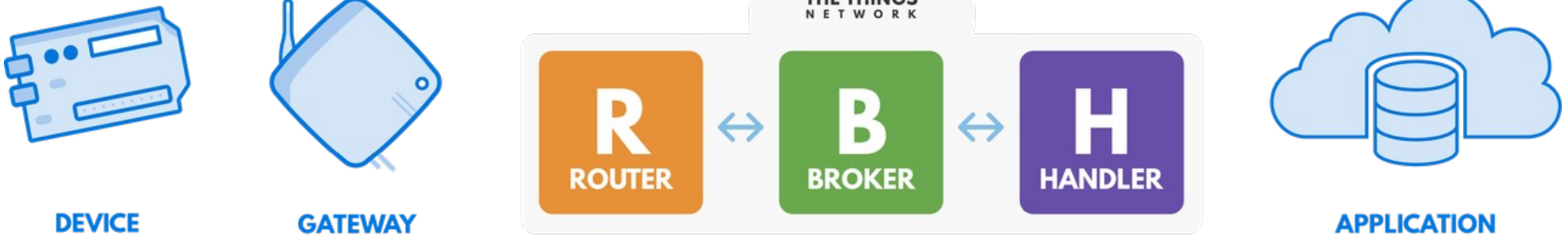
- 公用 LoRaWAN 營運商，提供設備與服務
- 使用者可註冊 TTN，提供公開的閘道器，或是使用 TTN 的閘道器產品
- 使用者可建立自己的閘道器和應用程式



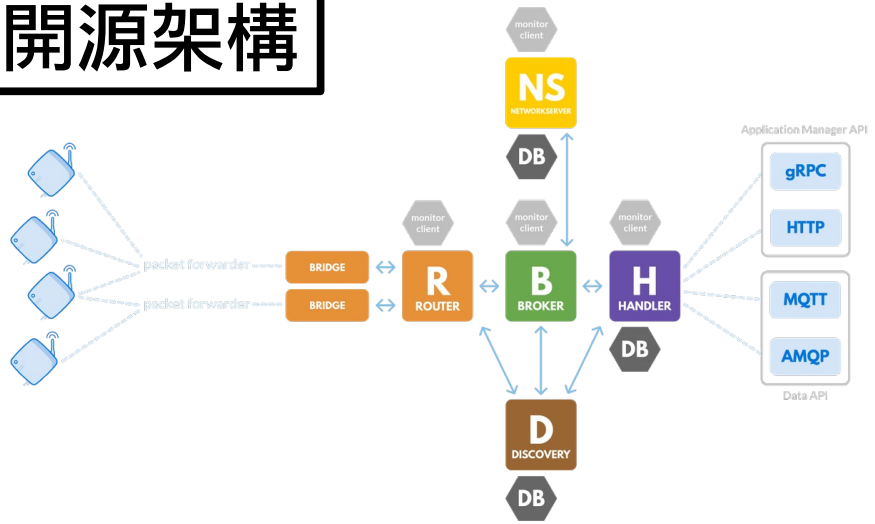
https://www.thethingsnetwork.org/

TTN 網路架構

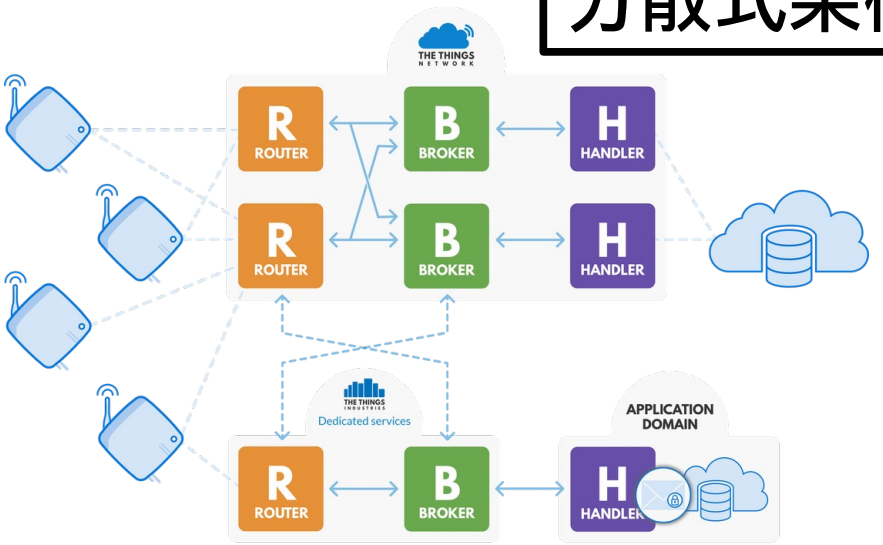
基本元件



開源架構

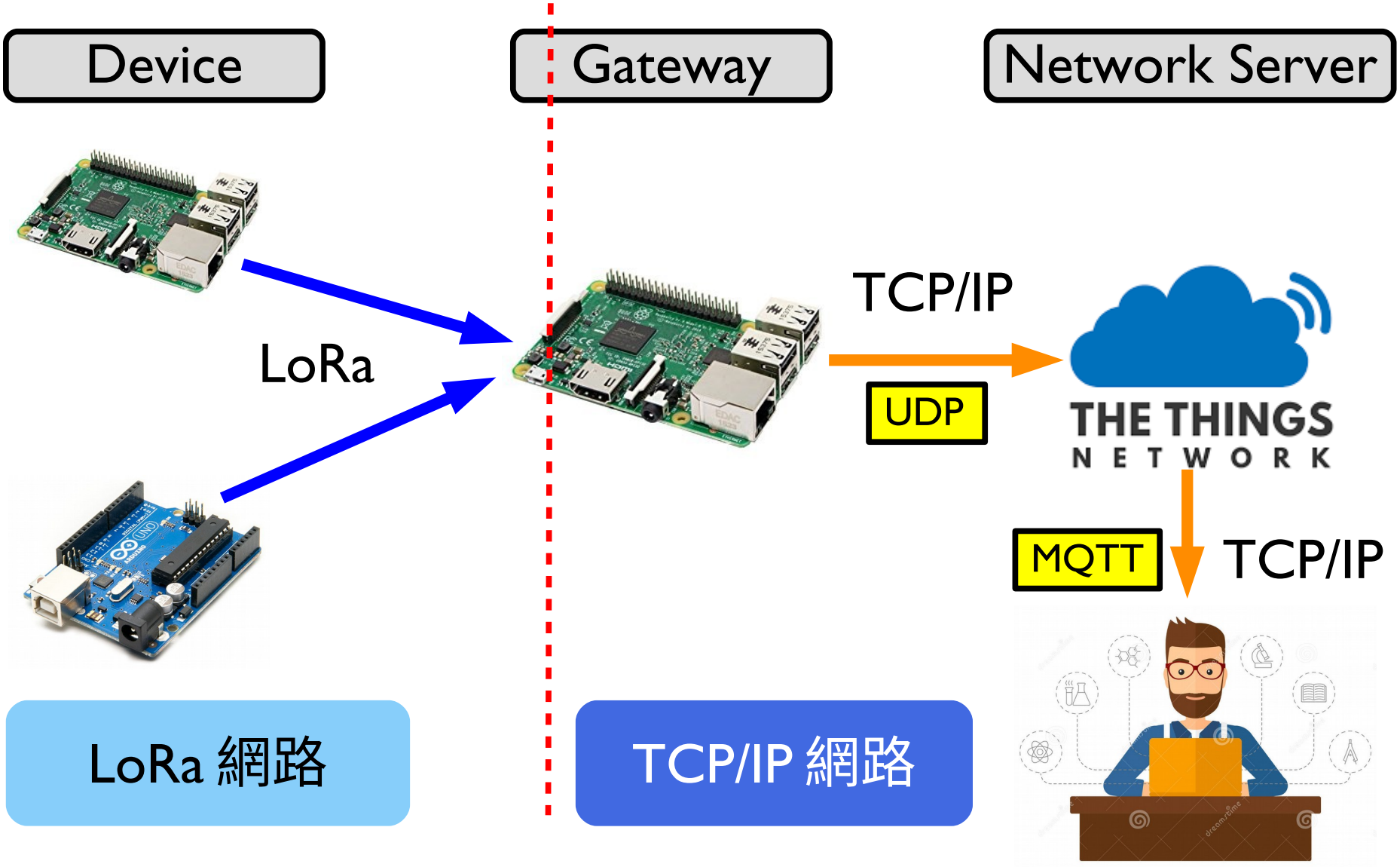


分散式架構



https://www.thethingsnetwork.org/article/the-things-network-architecture-1

串接 TTN



https://www.dreamstime.com/63482375

如何從 Gateway 轉送資料到 TTN?

使用 Semtech UDP protocol, Port 1700

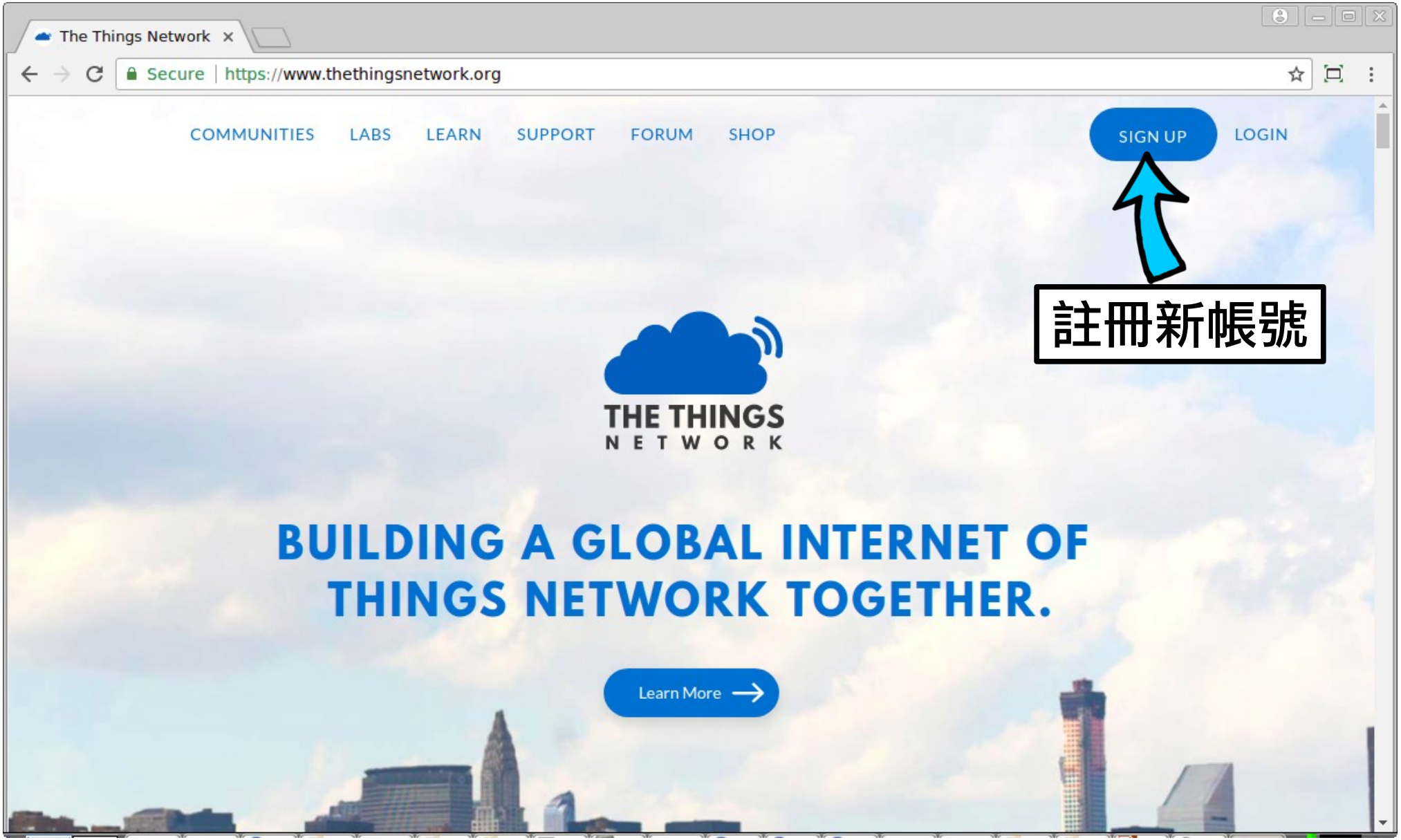

```
{
  "rxpk": {
    "tmst":20900514000,
    "chan":2,
    "rfch":0,
    "freq":866.349812,
    "stat":1,
    "modu":"LORA",
    "datr":"SF7BW125",
    "codr":"4/6",
    "rssi":-35,
    "lsnr":5,
    "size":23,
    "data":"AMy7qgAAAAAATYMmmnj6AADl6YP1Jrw"
  }
}
```


https://www.thethingsnetwork.org/docs/gateways/start/connection.html

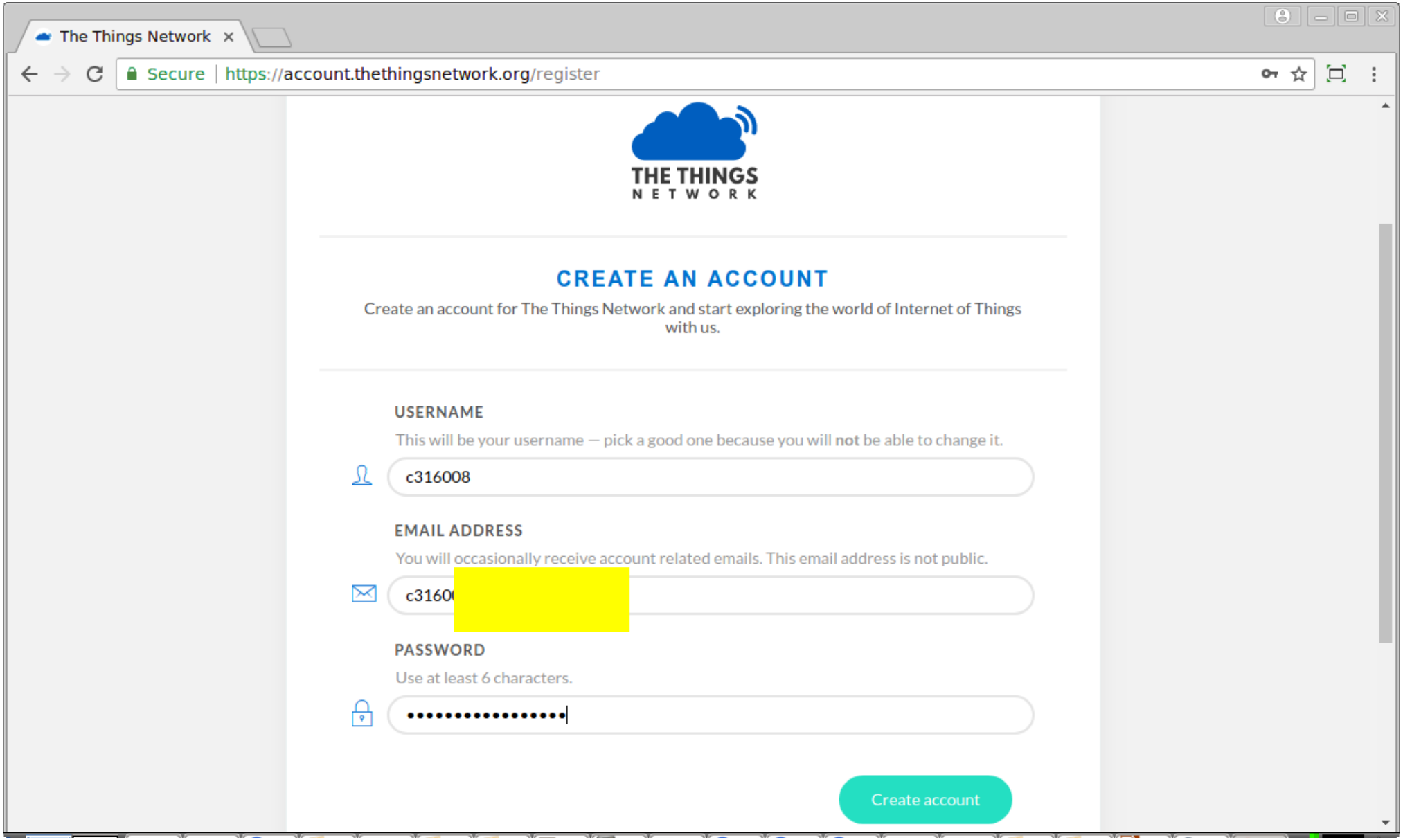
TTN 操作流程說明

- 註冊一個帳號
- 建立 gateway > 新增 app > 新增 device

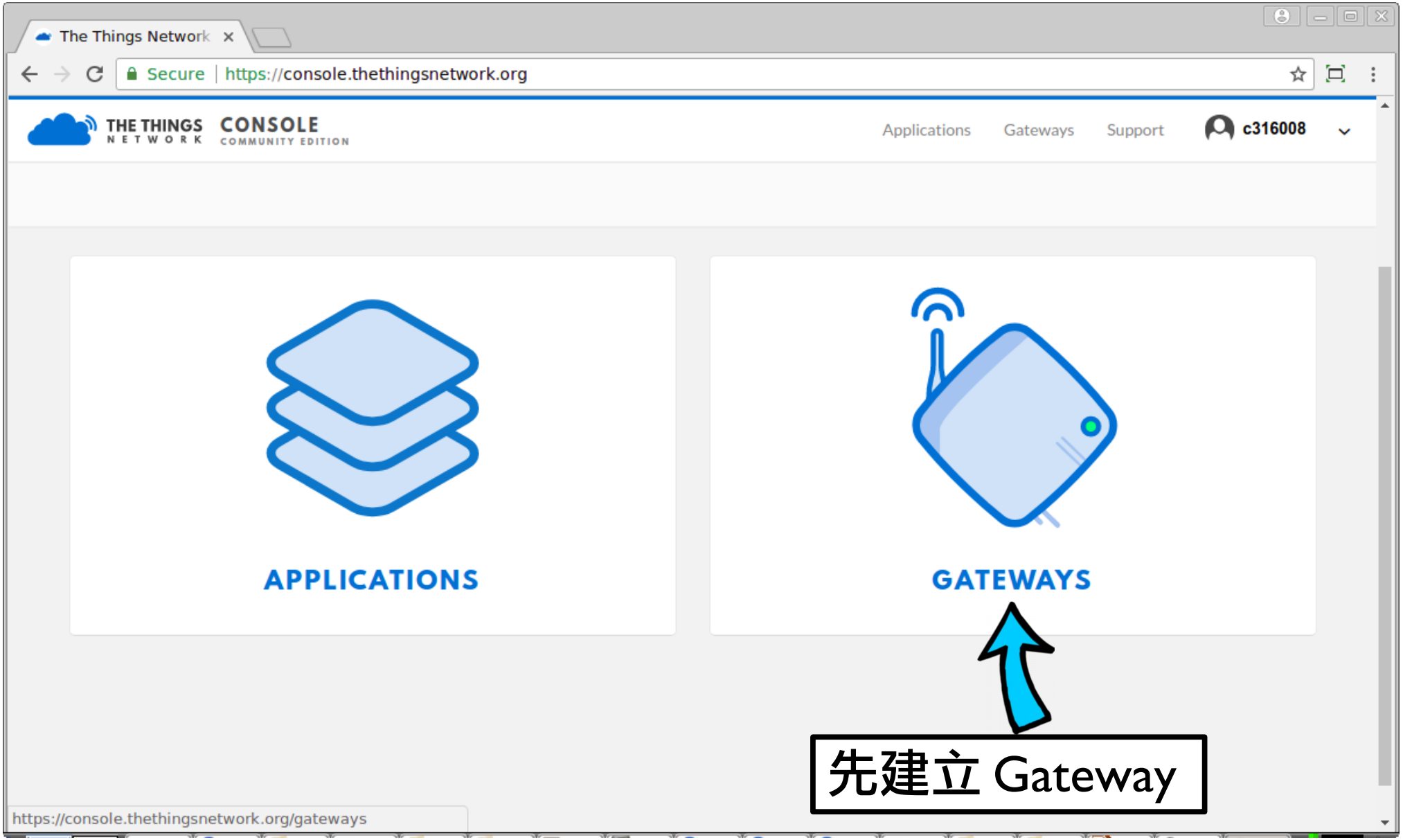
https://www.thethingsnetwork.org



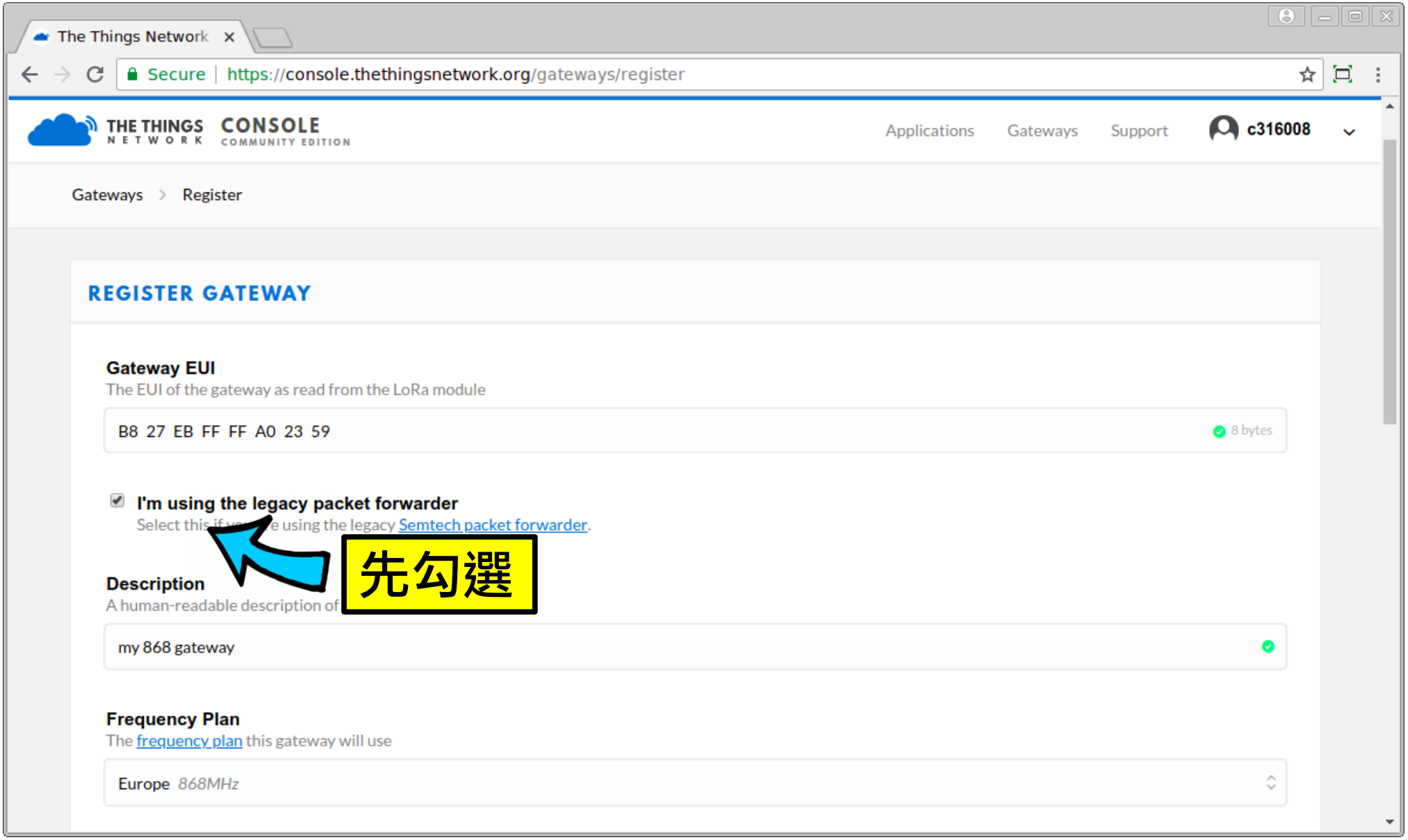
註冊新帳號


The Things Network
Secure | https://account.thethingsnetwork.org/register
THE THINGS NETWORK
CREATE AN ACCOUNT
Create an account for The Things Network and start exploring the world of Internet of Things with us.
USERNAME
This will be your username — pick a good one because you will not be able to change it.
c316008
EMAIL ADDRESS
You will occasionally receive account related emails. This email address is not public.
c3160
PASSWORD
Use at least 6 characters.
Create account

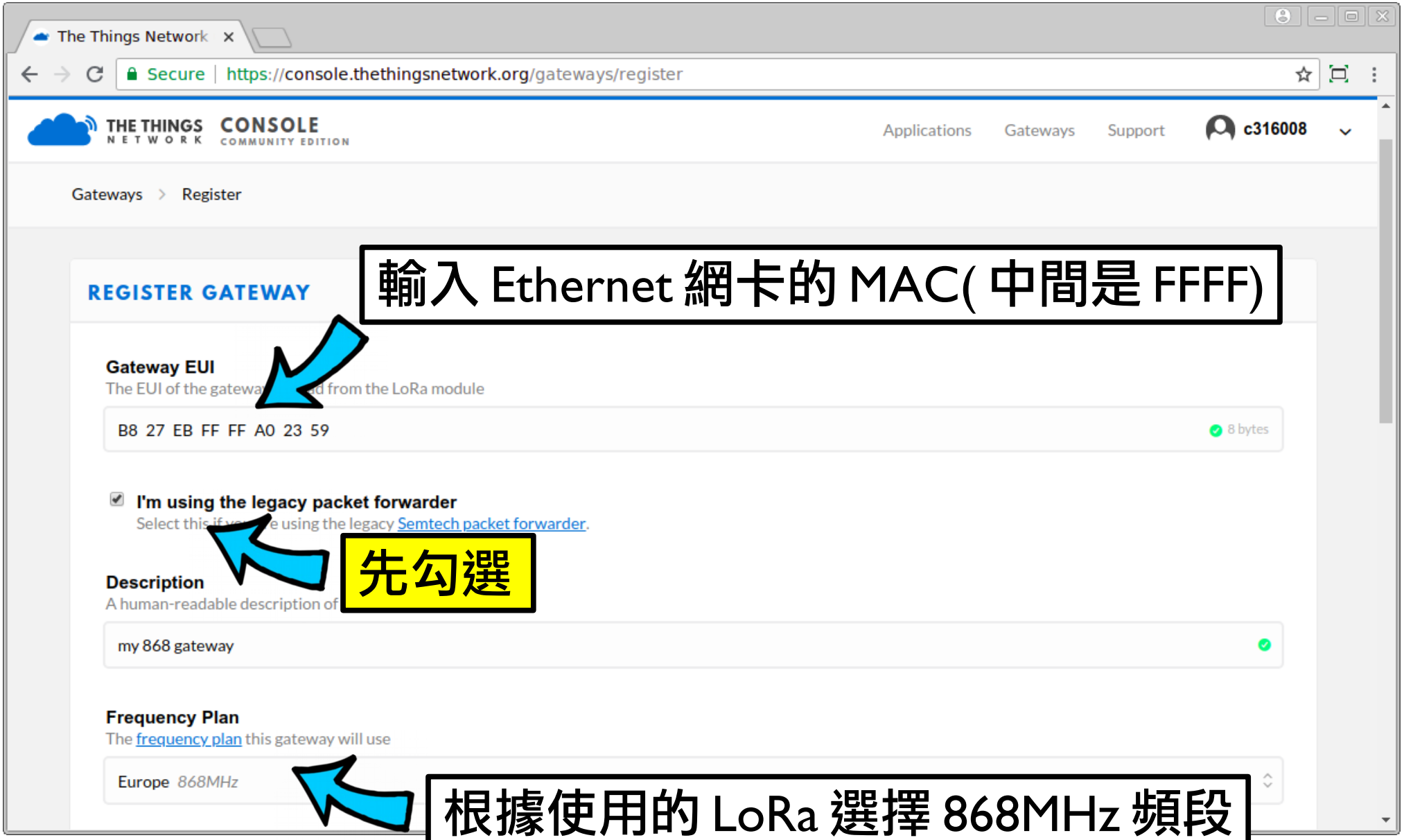
後端管理界面 (Console)



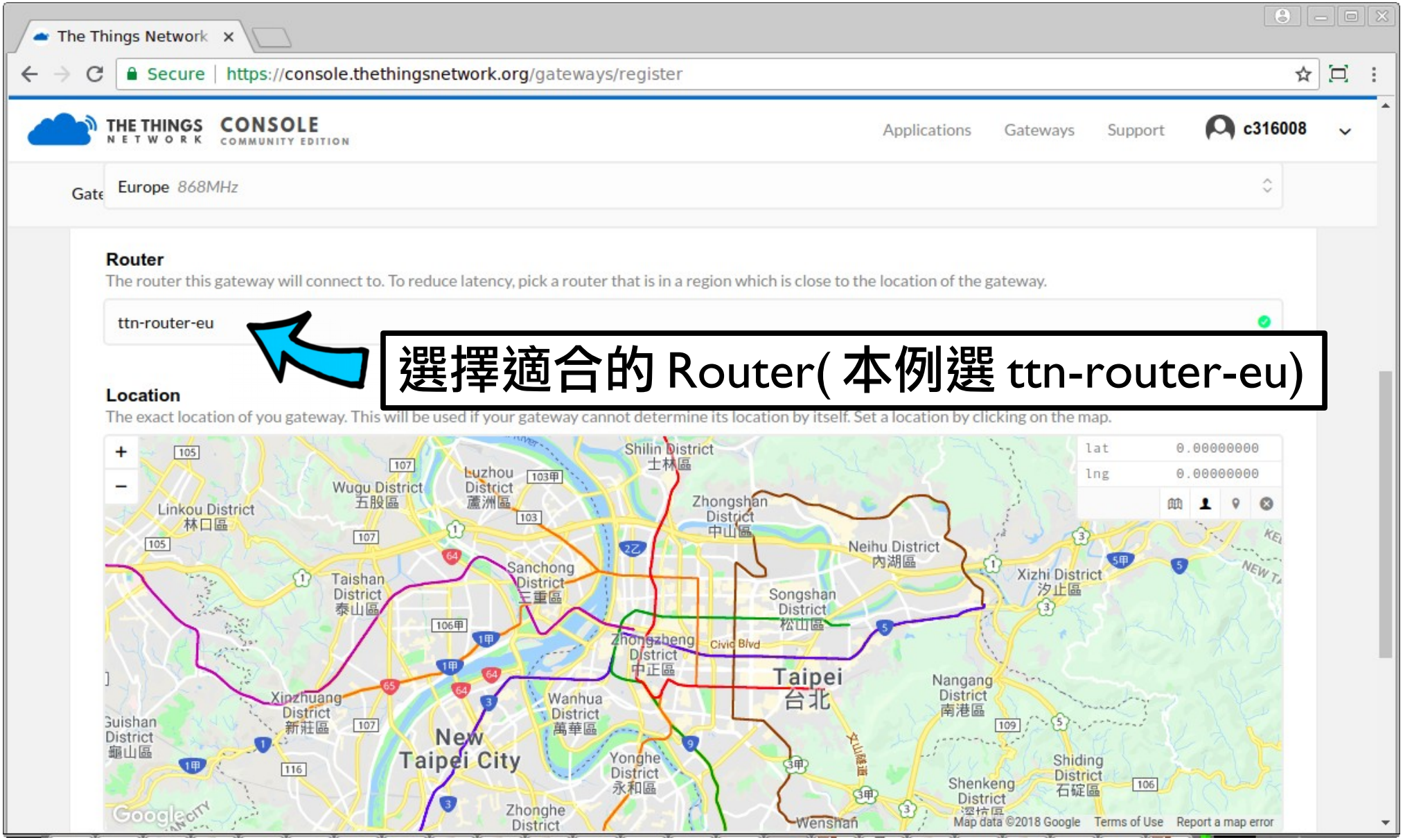
註冊一個新的閘道器 (Gateway)



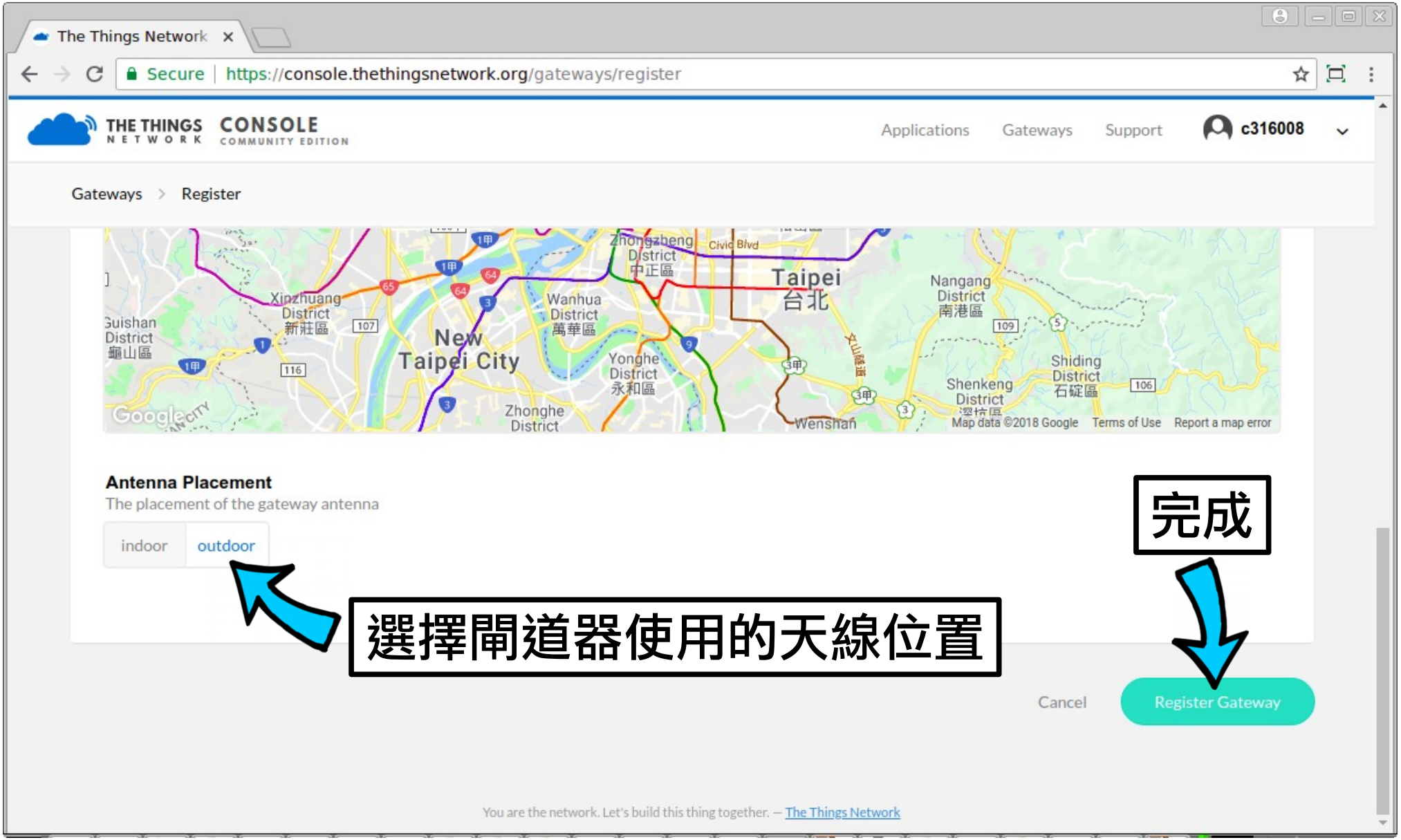
註冊一個新的閘道器 (Gateway)



選擇路由器 (Router)



完成註冊


The Things Network
Secure | https://console.thethingsnetwork.org/gateways/register
THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
c316008
Gateways > Register
Zhongzheng District 中正區
Civic Blvd
Taipei 台北
Nangang District 南港區
Xinzhuang District 新莊區
Wanhua District 萬華區
New Taipei City
Yonghe District 永和區
Zhonghe District
Wenshan
Shiding District 石碇區
Shenkeng District
Map data ©2018 Google
Terms of Use
Report a map error
Antenna Placement
The placement of the gateway antenna
indoor
outdoor
選擇閘道器使用的天線位置
完成
Cancel
Register Gateway
You are the network. Let's build this thing together. — The Things Network

查詢 Ethernet 網卡的 MAC

- `$ ifconfig eth0`

```
pi@raspberrypi:~ $ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.113  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::abdf:a26e:183f:1a5  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:a0:23:59  txqueuelen 1000  (Ethernet)
        RX packets 178  bytes 16071 (15.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 174  bytes 26964 (26.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- 本例 `eth0` 的 MAC 為 `b8:27:eb:a0:23:59`

測試是否連上 Gateway

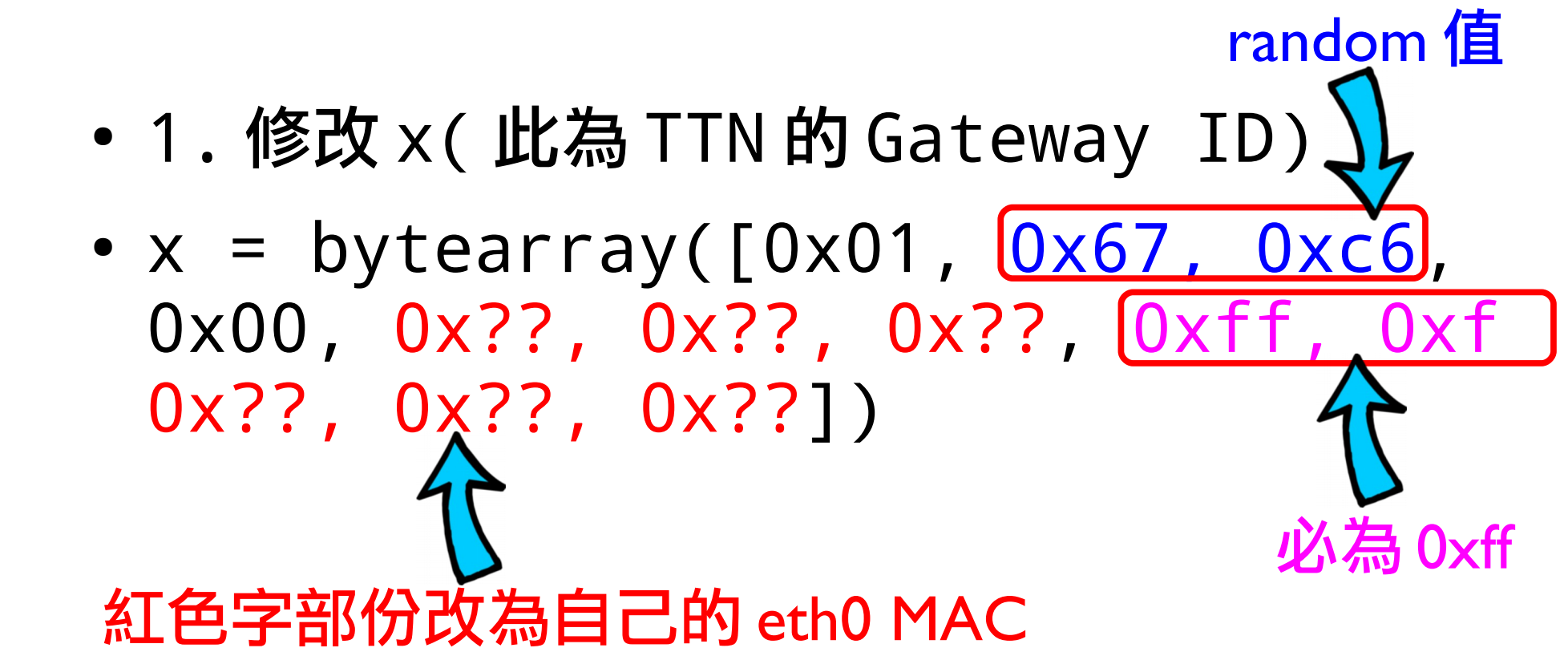
- 如果 Router 選擇 ttn-router-eu, 先 ping 機器 router.eu.thethings.network 取得正確 IP(本例為 52.169.76.203)
- $ ping router.eu.thethings.network

```
pi@raspberrypi:~ $ ping router.eu.thethings.network
PING bridge.eu.thethings.network (52.169.76.203) 56(84) bytes of data.
```

- router.eu.thethings.network # EU 433 MHz and EU 863-870 MHz
- router.us.thethings.network # US 902-928 MHz
- router.cn.thethings.network # China 470-510 MHz and 779-787 MHz
- router.au.thethings.network # Australia 915-928 MHz

http://www.instructables.com/id/Use-Lora-Shield-and-RPi-to-Build-a-LoRaWAN-Gateway/

修改 03-ttn/ttn_var.py 程式

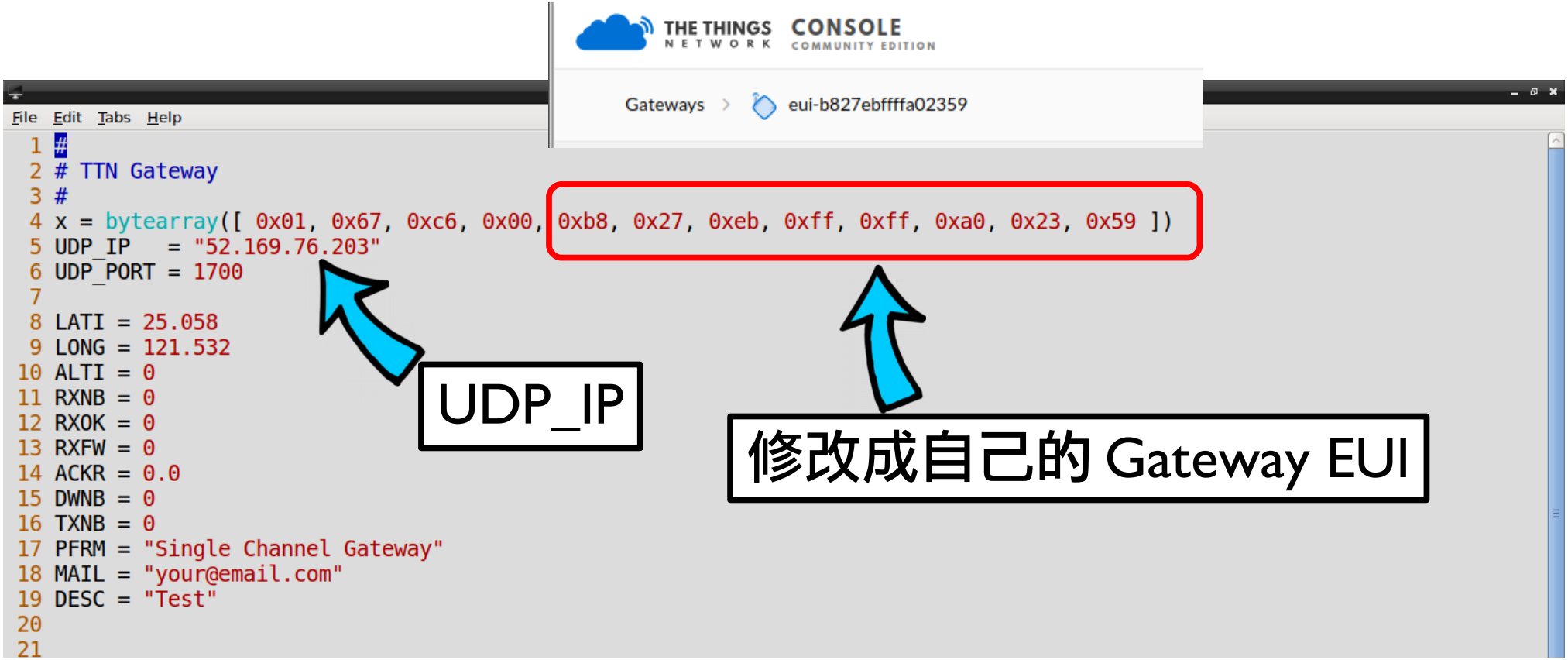


- 1. 修改 x(此為 TTN 的 Gateway ID)
- x = bytearray([0x01, 0x67, 0xc6,
  0x00, 0x??, 0x??, 0x??, 0xff, 0xf
  0x??, 0x??, 0x??])
- 2. 修改 UDP_IP(此為 Router IP)

  UDP_IP = "52.169.76.203"

修改 03-ttn/ttn_var.py 程式

- 傳送 UDP 封包到 TTN，連線埠號為 1700
- 和 ttn_gateway.py 組合 LoRaWAN 的傳輸內容

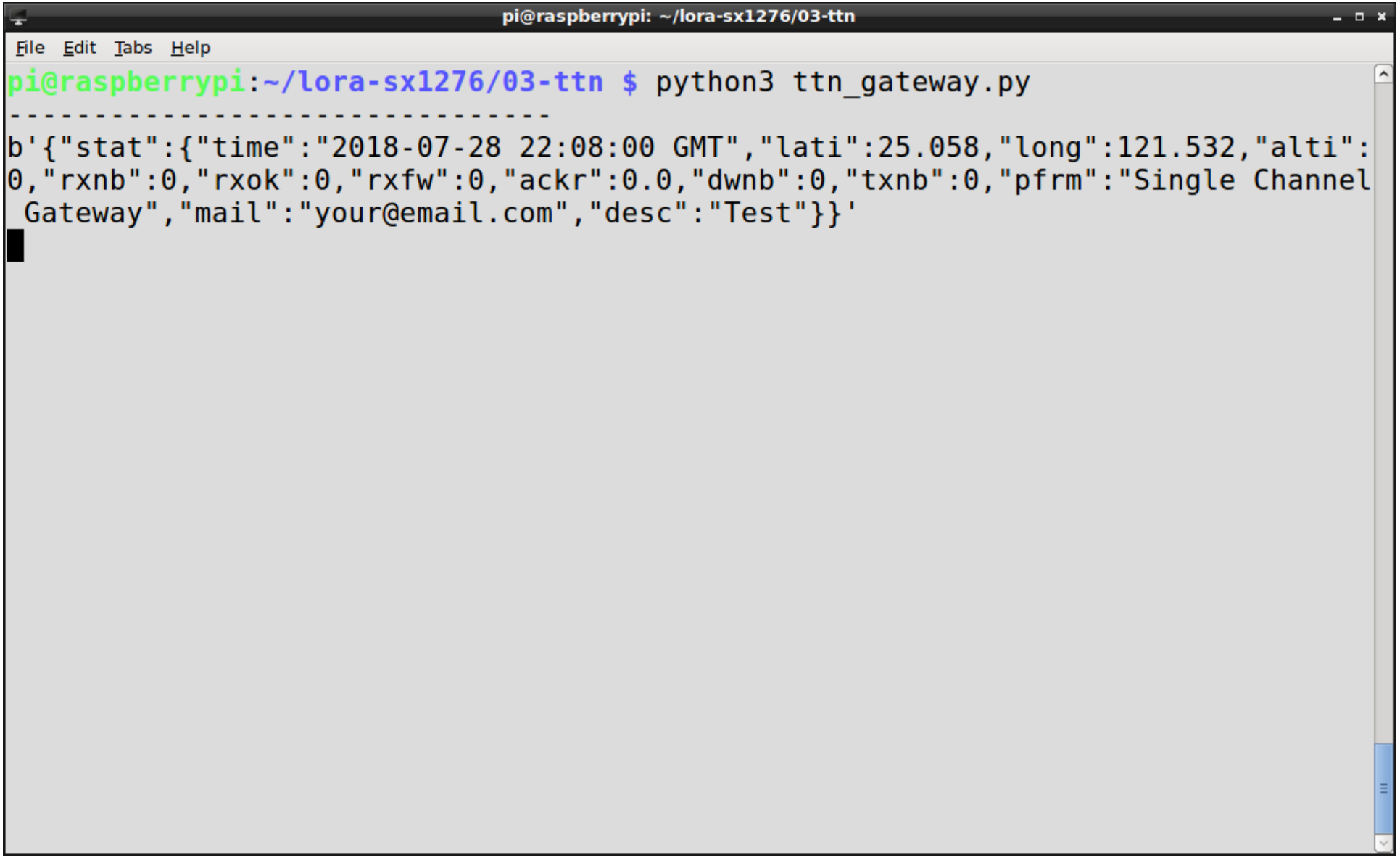


欄位意義

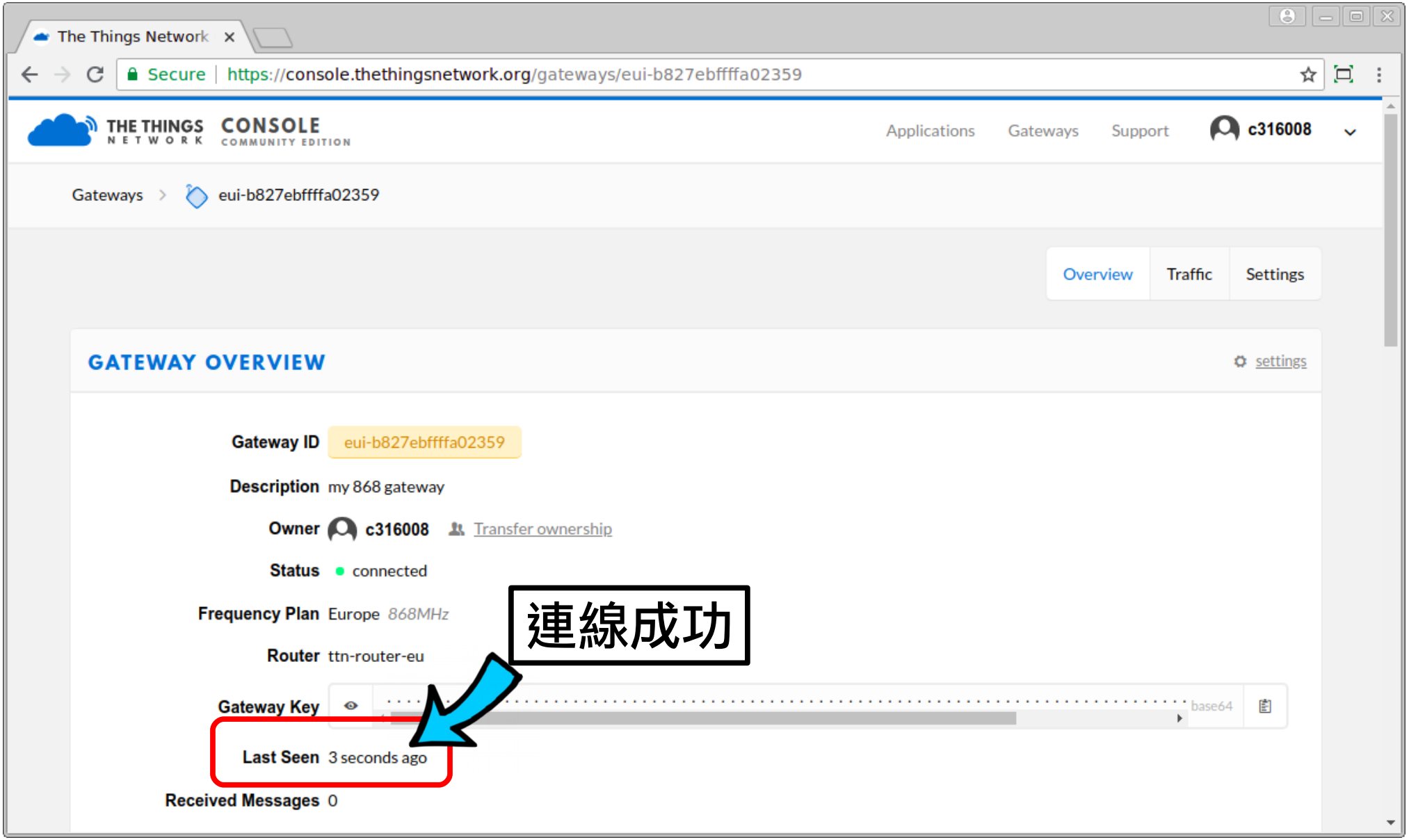
Name	Type	Function
time	string	UTC 'system' time of the gateway, ISO 8601 'expanded' format
lati	number	GPS latitude of the gateway in degree (float, N is +)
long	number	GPS latitude of the gateway in degree (float, E is +)
alti	number	GPS altitude of the gateway in meter RX (integer)
rxnb	number	Number of radio packets received (unsigned integer)
rxok	number	Number of radio packets received with a valid PHY CRC
rxfw	number	Number of radio packets forwarded (unsigned integer)
ackr	number	Percentage of upstream datagrams that were acknowledged
dwnb	number	Number of downlink datagrams received (unsigned integer)
txnb	number	Number of packets emitted (unsigned integer)

https://github.com/Lora-net/packet_forwarder

執行 ttn_gateway.py 範例程式



在 TTN 後台看到連線成功



要先修改 ttn_var.py 程式

DEMO
ttn_gateway.py

```
$ cd ~/lora-sx1276/03-ttn
$ python3 ttn_gateway.py
```

串接 TTN

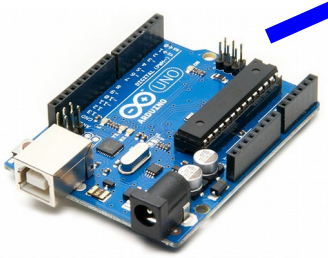
Device

Gateway

Network Server

連線成功

THE THINGS NETWORK



LoRa 網路

TCP/IP 網路

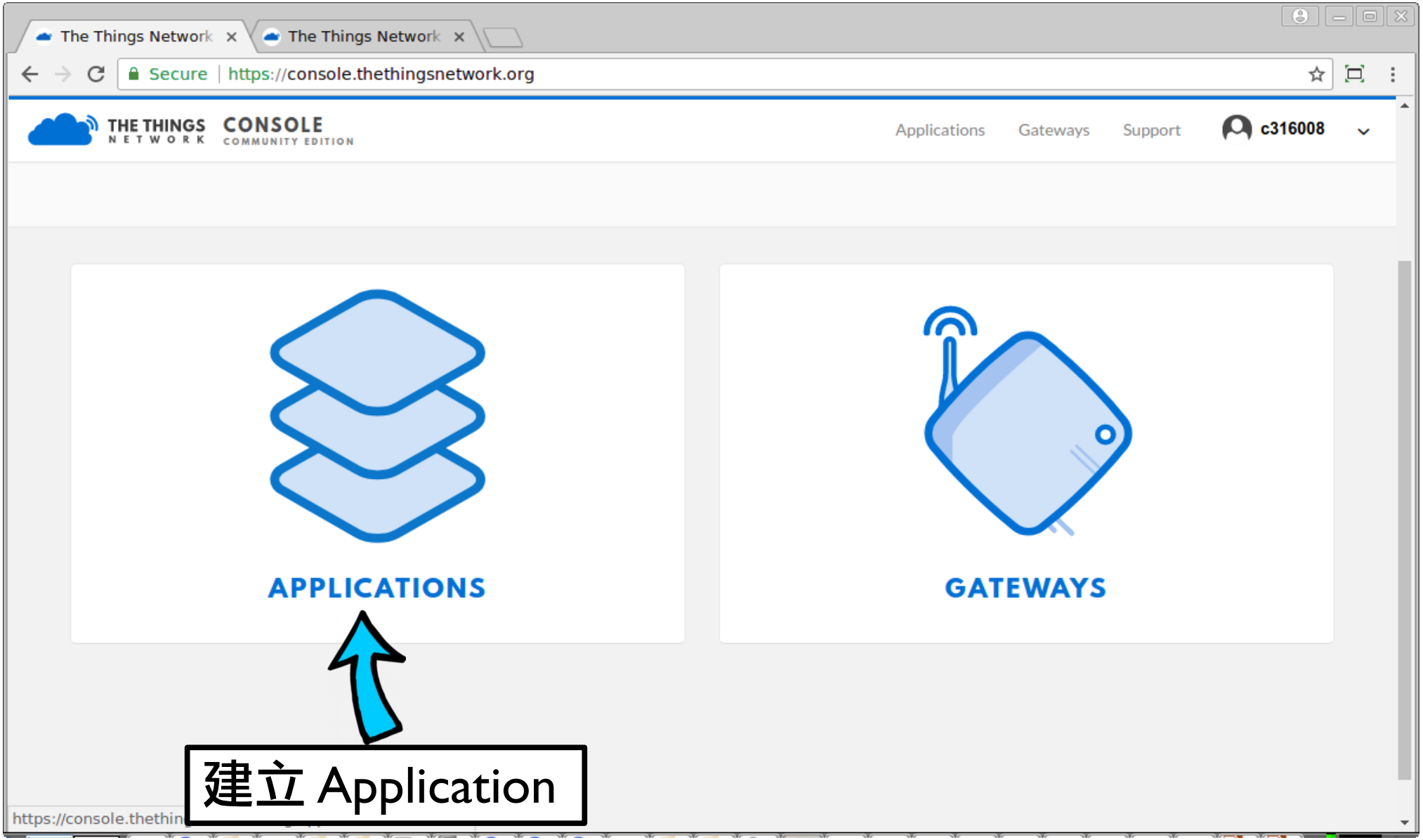
https://www.dreamstime.com/63482375

每個裝置都可以送資料，誰是合法的？

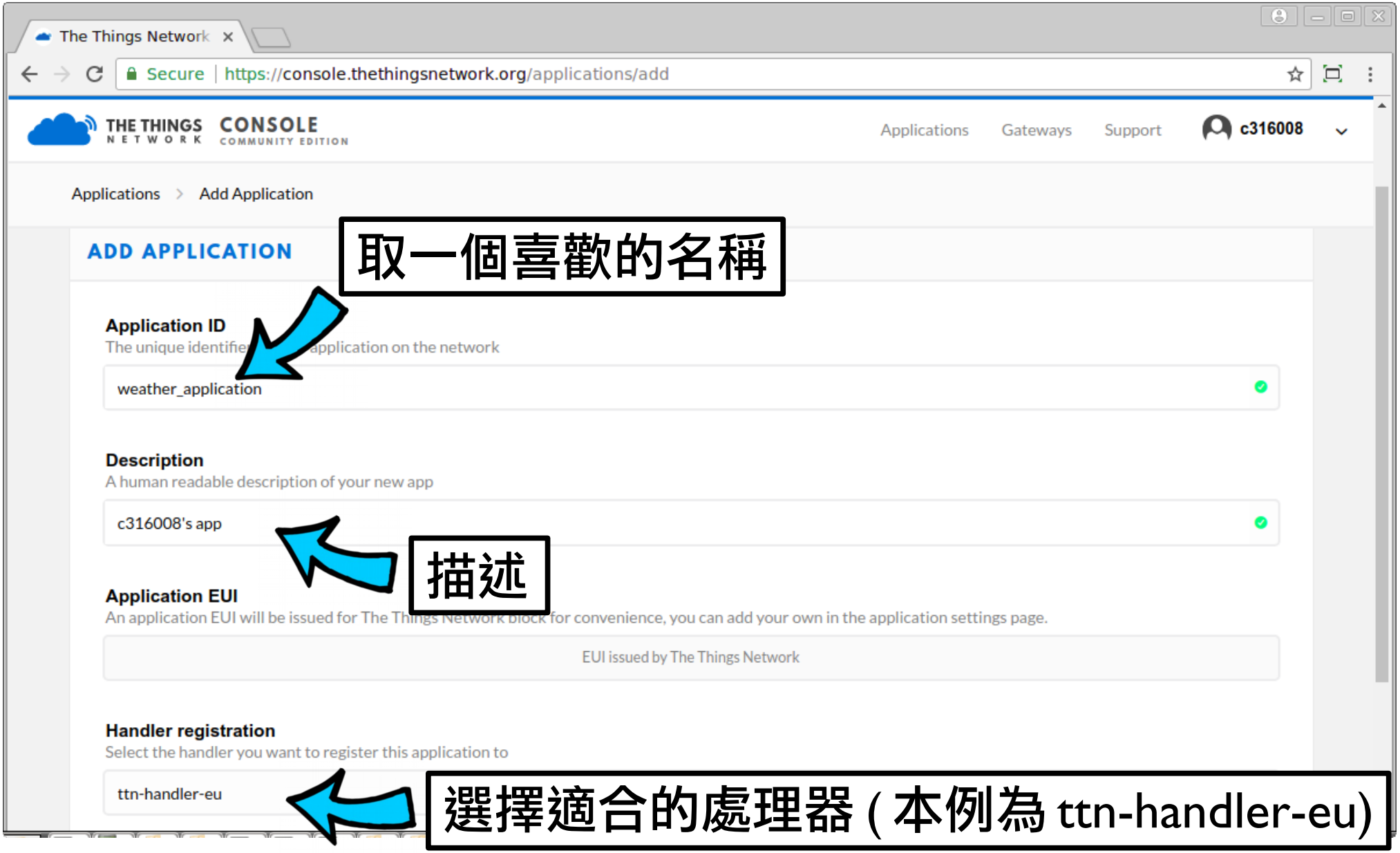
在 TTN 註冊裝置

- 註冊完成後取得 DevAddr
- 後續資料傳輸時需要一併傳送 DevAddr 資訊
- 由 Network Server(TTN) 確認裝置是否合法

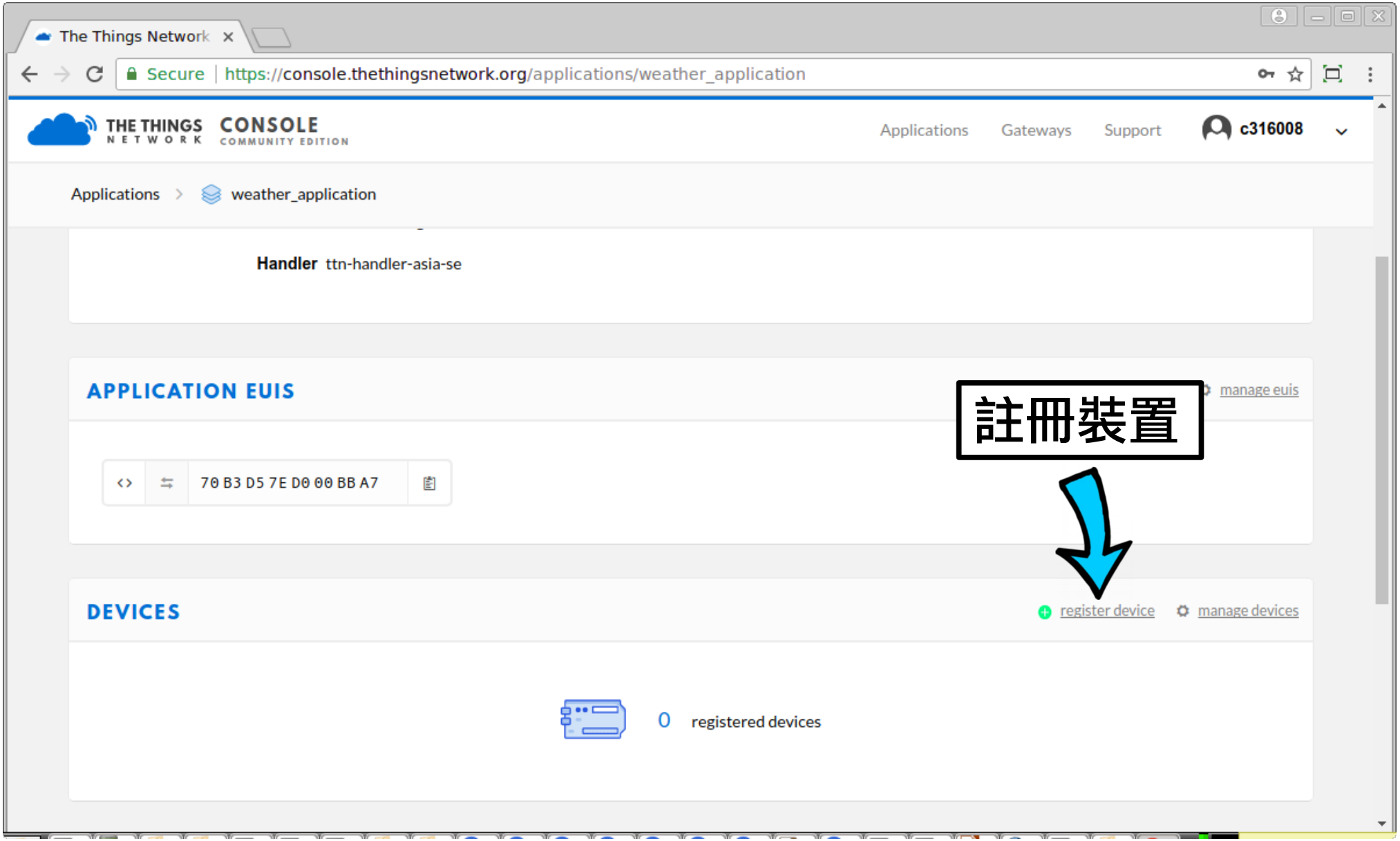
TTN 後端管理界面 (Console)



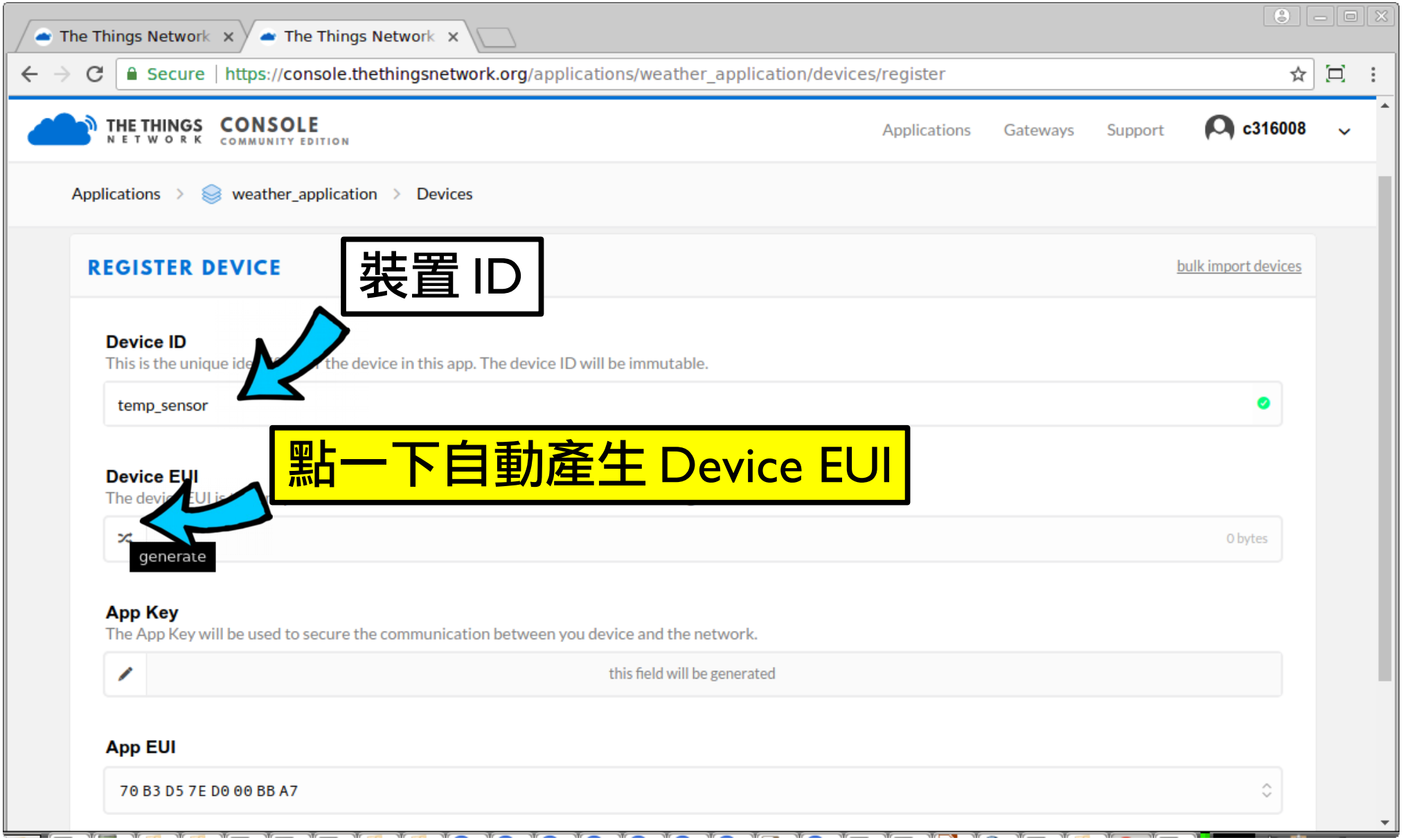
建立新的應用程式


The Things Network
Secure | https://console.thethingsnetwork.org/applications/add
THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
c316008
Applications > Add Application
ADD APPLICATION
取一個喜歡的名稱
Application ID
The unique identifier application on the network
weather_application
Description
A human readable description of your new app
c316008's app
描述
Application EUI
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.
EUI issued by The Things Network
Handler registration
Select the handler you want to register this application to
ttn-handler-eu
選擇適合的處理器 (本例為 ttn-handler-eu)

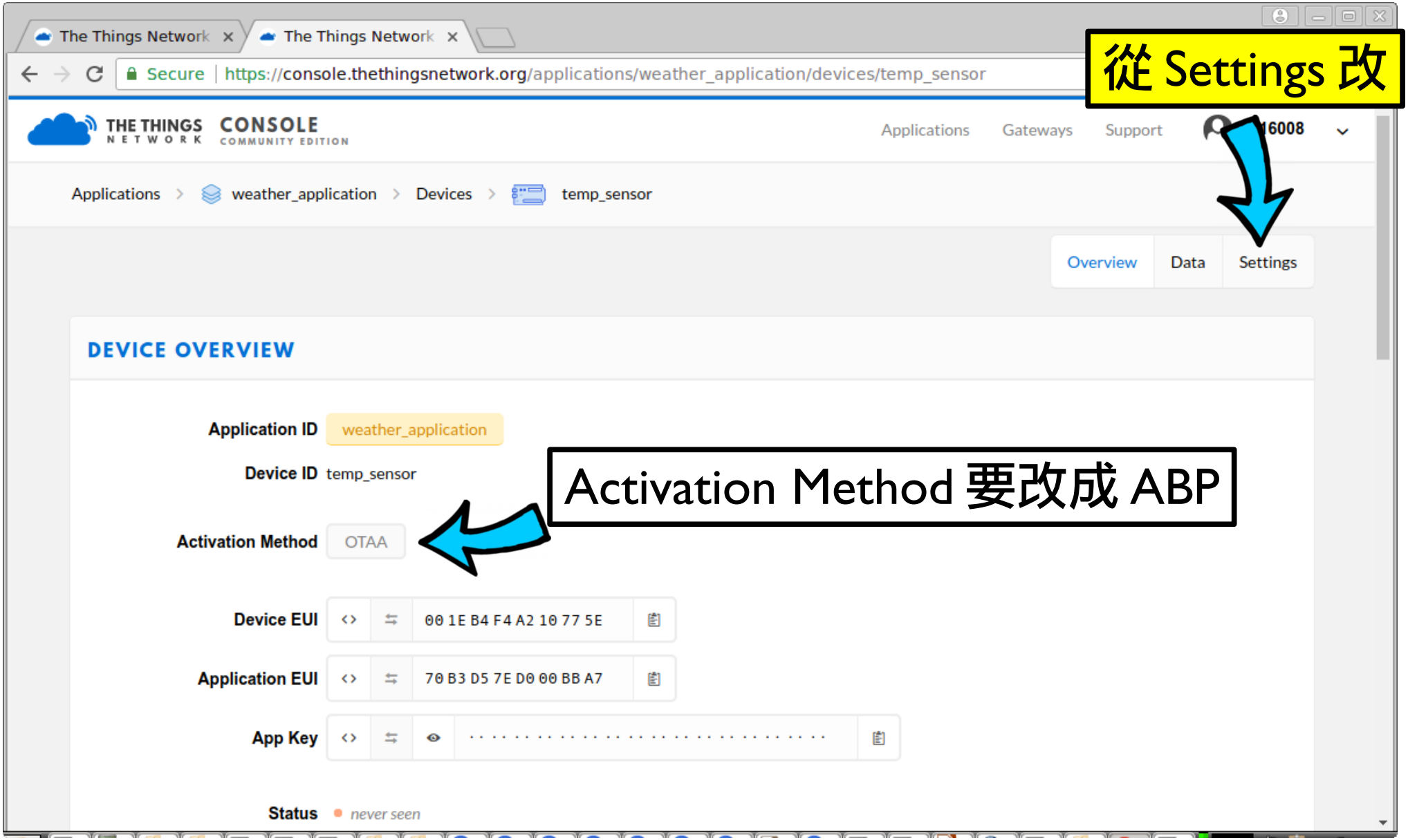
新增裝置


The Things Network
Secure | https://console.thethingsnetwork.org/applications/weather_application
THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
c316008
Applications > weather_application
Handler ttn-handler-asia-se
APPLICATION EUIS
manage euis
70 B3 D5 7E D0 00 BB A7
註冊裝置
DEVICES
register device
manage devices
0 registered devices

註冊新的裝置


The Things Network
The Things Network
Secure
https://console.thethingsnetwork.org/applications/weather_application/devices/register
THE THINGS NETWORK
CONSOLE
COMMUNITY EDITION
Applications
Gateways
Support
c316008
Applications
weather_application
Devices
REGISTER DEVICE
bulk import devices
裝置 ID
Device ID
This is the unique ide the device in this app. The device ID will be immutable.
temp_sensor
點一下自動產生 Device EUI
Device EUI
The devi EUI is
generate
0 bytes
App Key
The App Key will be used to secure the communication between you device and the network.
this field will be generated
App EUI
70 B3 D5 7E D0 00 BB A7

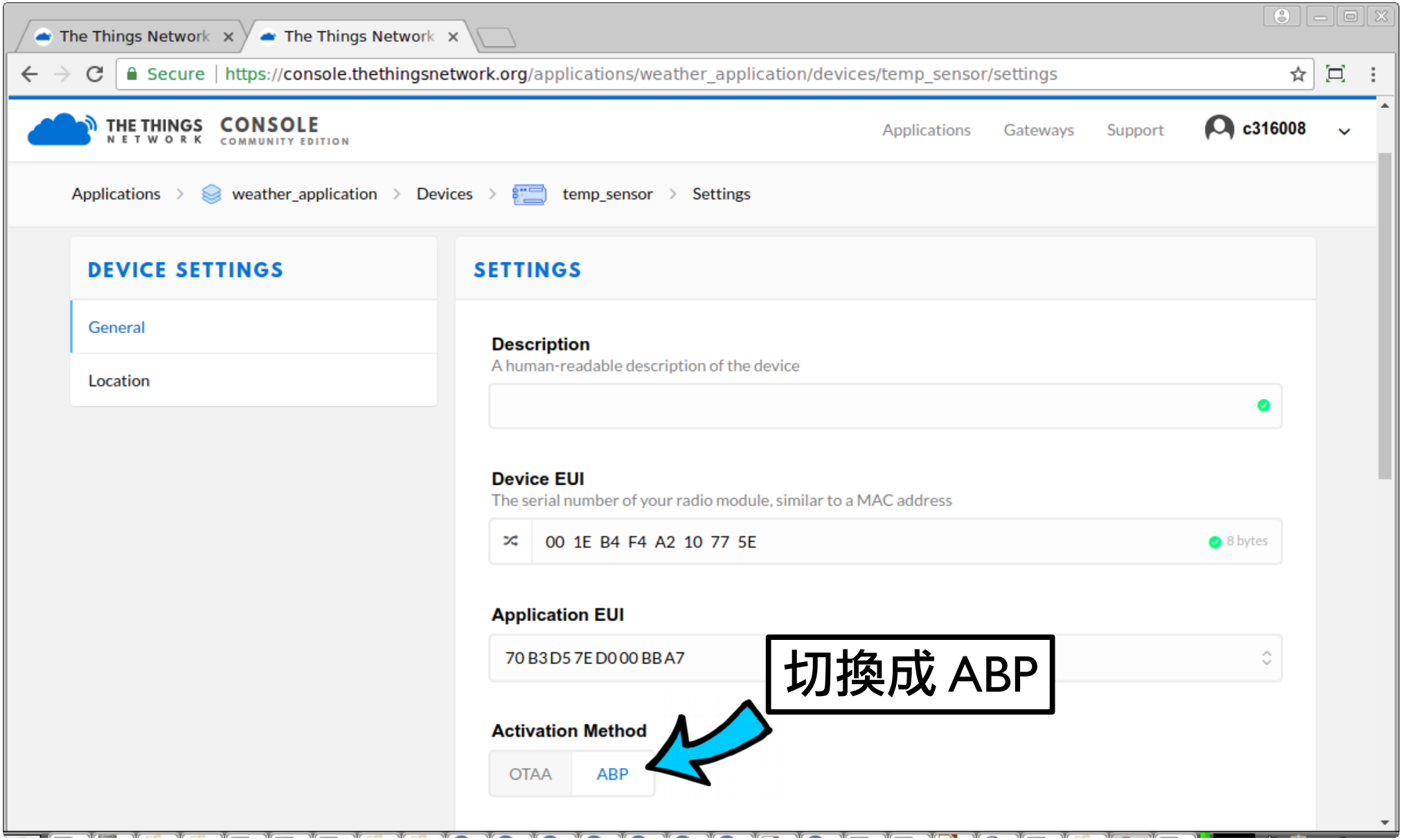
裝置概況


The Things Network
The Things Network
Secure
https://console.thethingsnetwork.org/applications/weather_application/devices/temp_sensor
從 Settings 改
THE THINGS NETWORK
CONSOLE
COMMUNITY EDITION
Applications
Gateways
Support
16008
Applications
weather_application
Devices
temp_sensor
Overview
Data
Settings
DEVICE OVERVIEW
Application ID
weather_application
Device ID
temp_sensor
Activation Method
OTAA
Activation Method 要改成 ABP
Device EUI
00 1E B4 F4 A2 10 77 5E
Application EUI
70 B3 D5 7E D0 00 BB A7
App Key
Status
never seen

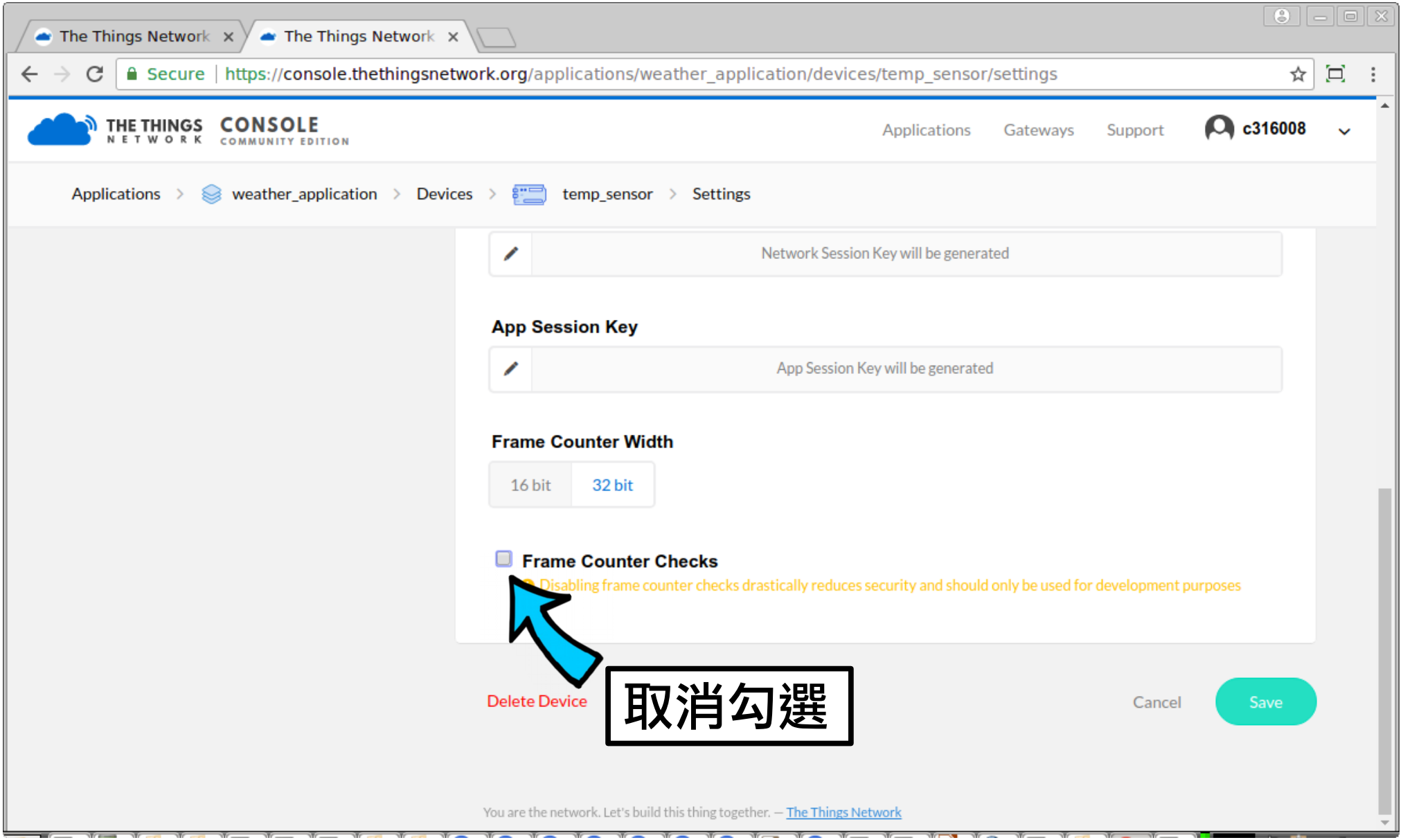
Activation Method

- ABP(Activation By Personalization)
  - 將接上網路的資訊寫在設備上，亦即上電即上網
- OTAA(Over-The-Air Activation)
  - 設備原本不接上網路，需要線上註冊與啟動
    - node 發送 join_request message
    - gateway 收到 node 資訊後，上傳到伺服器
    - 伺服器收到上網請求，將設備註冊後把金鑰回傳給 gateway，表示 join_accept_message
    - gateway 轉發到 node
    - node 取得金鑰，包括 DevAddr,APPSKEY,NWKSKEY

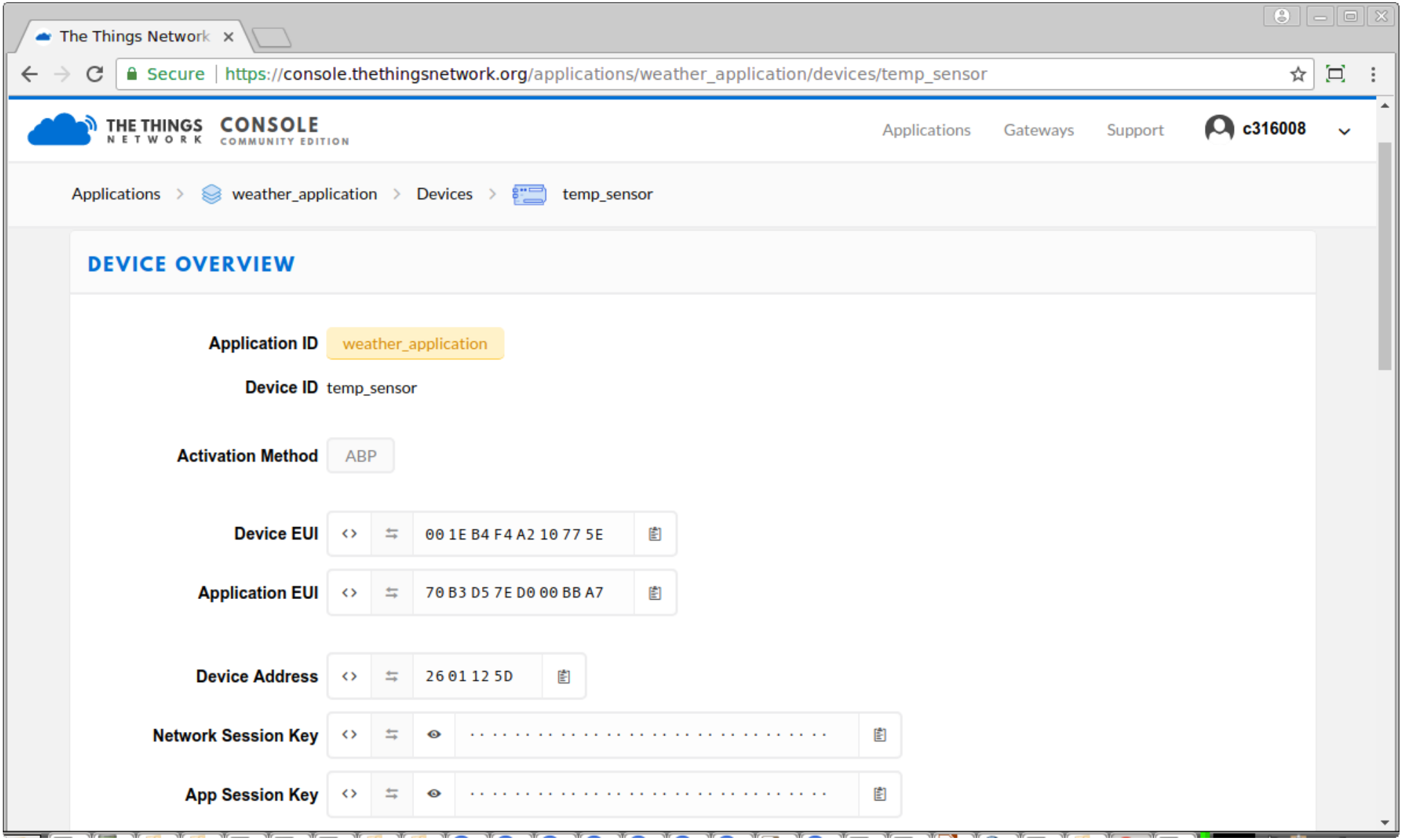
到設定 (Settings) 修改啟動方式



取消 Frame Counter

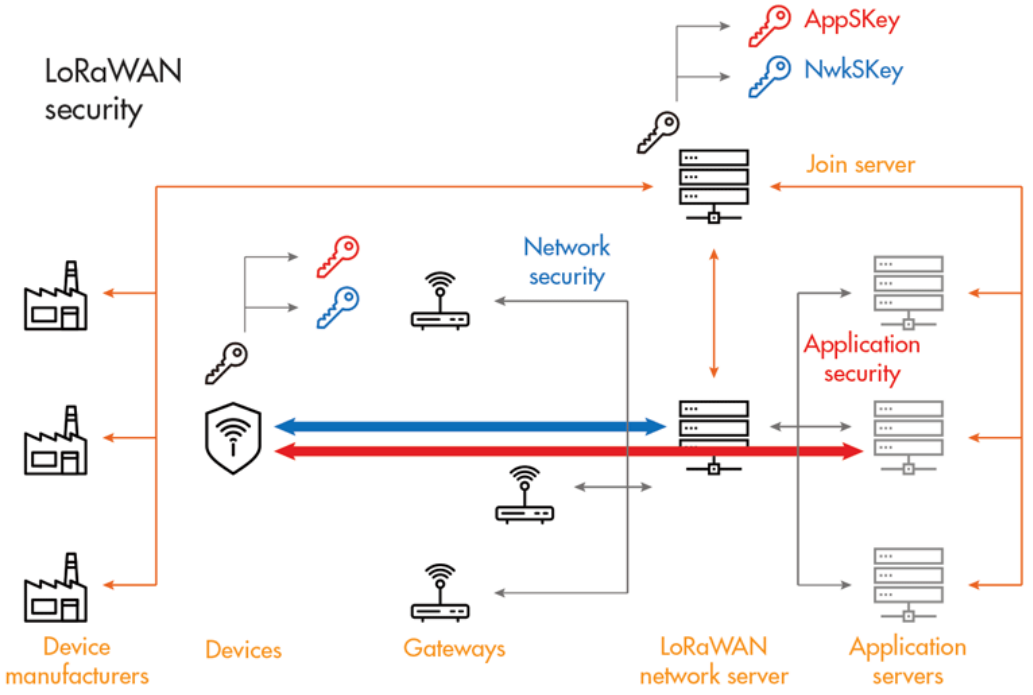


裝置設定


The Things Network
Secure
https://console.thethingsnetwork.org/applications/weather_application/devices/temp_sensor
THE THINGS NETWORK
CONSOLE
COMMUNITY EDITION
Applications
Gateways
Support
c316008
Applications
weather_application
Devices
temp_sensor
DEVICE OVERVIEW
Application ID
weather_application
Device ID
temp_sensor
Activation Method
ABP
Device EUI
00 1E B4 F4 A2 10 77 5E
Application EUI
70 B3 D5 7E D0 00 BB A7
Device Address
26 01 12 5D
Network Session Key
App Session Key

LoRaWAN Security

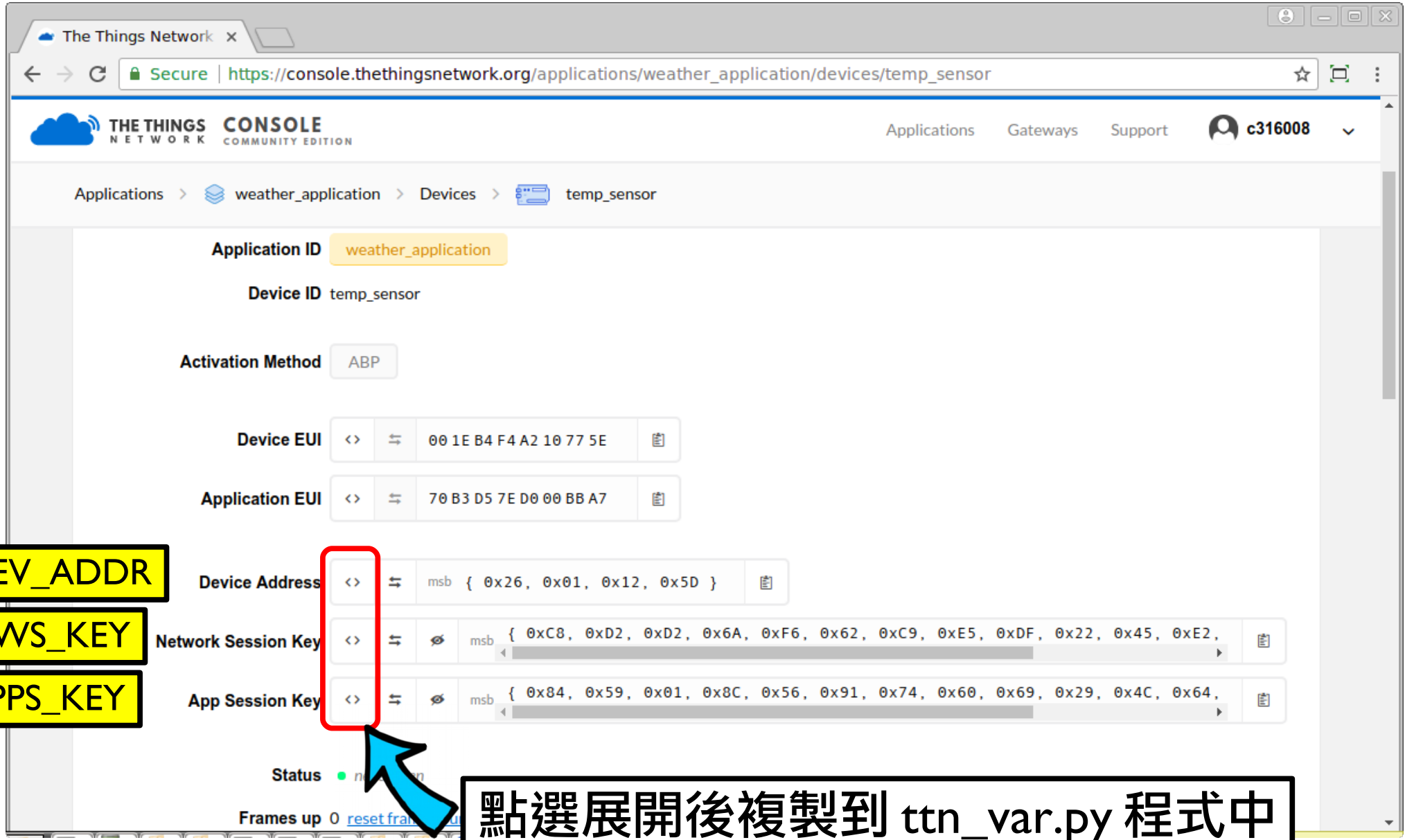
- 傳輸過程使用 AES128 加密，使用 AppSKey 和 NwkSKey 兩把金鑰
- 認證使用 DevAddr 確認裝置是否合法
- 透過 UDP 傳送使用 Base64 編碼方式的封包內容



修改 ttn_var.py 程式

- 1. 修改 DEV_ADDR(裝置的 Device Address)
- 2. 修改 NWS_KEY(裝置的 Network Session Key)
- 3. 修改 APPS_KEY(裝置的 App Session Key)
- 4. 修改 x(為 TTN 的 Gateway ID)
- 5. 修改 UDP_IP(此為 Router IP)
- 6. 修改 LoRa 參數內容 (選擇性)

展開欄位設定



DEV_ADDR

NWS_KEY

APPS_KEY

點選展開後複製到 ttn_var.py 程式中

修改 ttn_var.py 程式

DEV_ADDR

NWS_KEY

APPS_KEY

pi@raspberrypi: ~/lora-sx1276/03-ttn

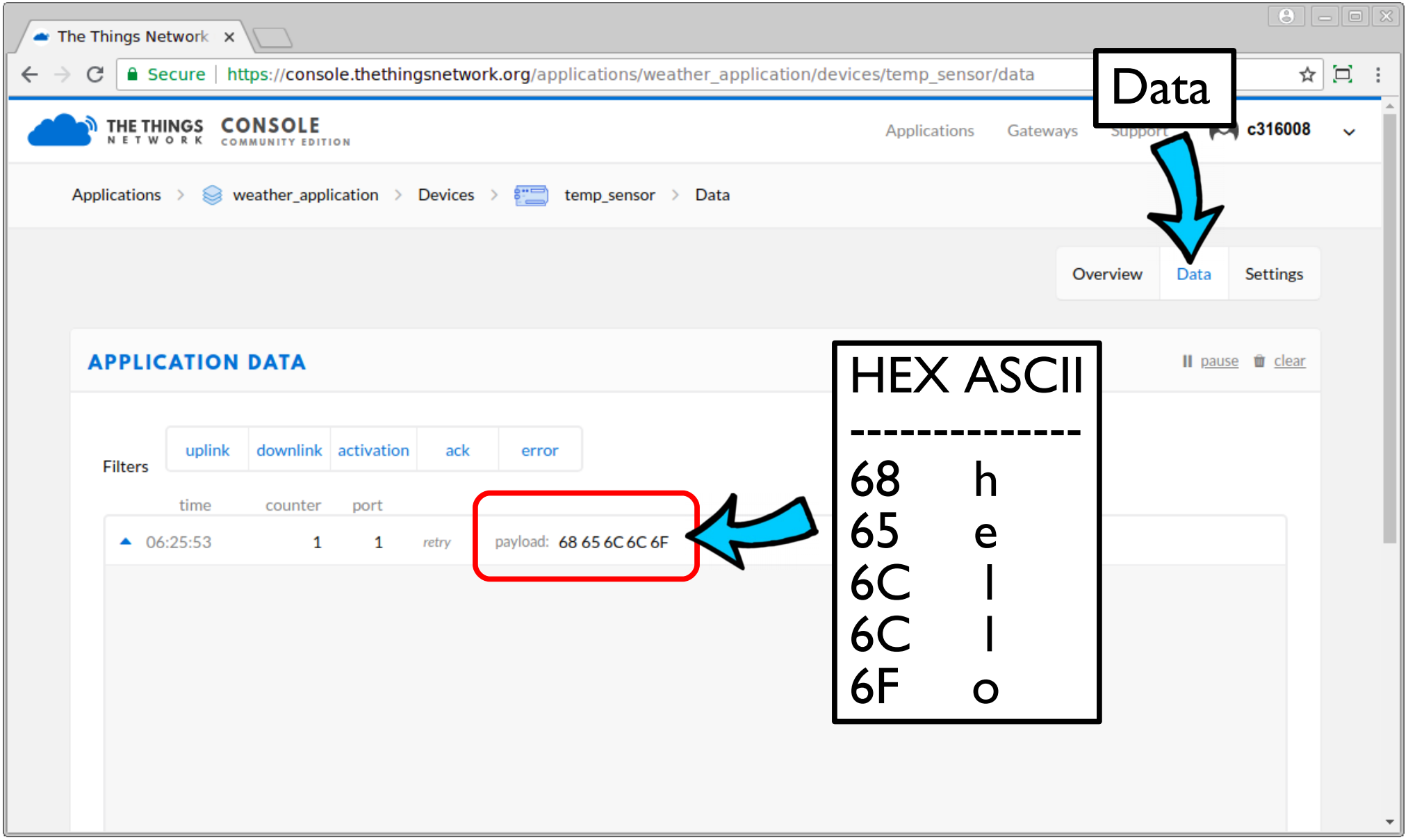
File Edit Tabs Help

```
20
21
#
# TTN Application
#
DEV_ADDR = [ 0x26, 0x01, 0x12, 0x5D ]
NWS_KEY  = [ 0xC8, 0xD2, 0xD2, 0x6A, 0xF6, 0x62, 0xC9, 0xE5, 0xDF, 0x22, 0x45, 0xE2, 0xA6, 0x98, 0xF2, 0x8A ]
APPS_KEY = [ 0x84, 0x59, 0x01, 0x8C, 0x56, 0x91, 0x74, 0x60, 0x69, 0x29, 0x4C, 0x64, 0x10, 0x18, 0x6C, 0x17 ]

CHAN = 0
RFCH = 0
FREQ = 868.0
STAT = 1
MODU = "LORA"
DATR = "SF7BW125"
CODR = "4/5"
LSNR = 9
RSSI = -32
SIZE = 19

40
```

可根據實際狀況修改，但不影響資料傳送

執行 ttn_app.py 程式

在 TTN 後台看到傳送資料成功



Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	|
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

asciichars.com

http://www.asciichars.com/

要先修改 ttn_var.py 程式

DEMO
ttn_app.py

```
$ cd ~/lora-sx1276/03-ttn
$ python3 ttn_app.py
```

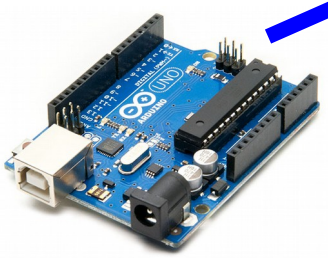
串接 TTN

Device

Gateway

Network Server

傳送資料成功



LoRa 網路

TCP/IP 網路

https://www.dreamstime.com/63482375

資料上傳到 TTN 以後怎麼下載回來？

(前言) 世界愈來愈聰明

Smarter Vehicles

- realtime telemetry
- predictive maintenance
- look-ahead alerting
- pay-as-you-drive

Smarter Homes

- energy tracking
- automation
- remote monitoring
- smart appliances

Smarter Logistics

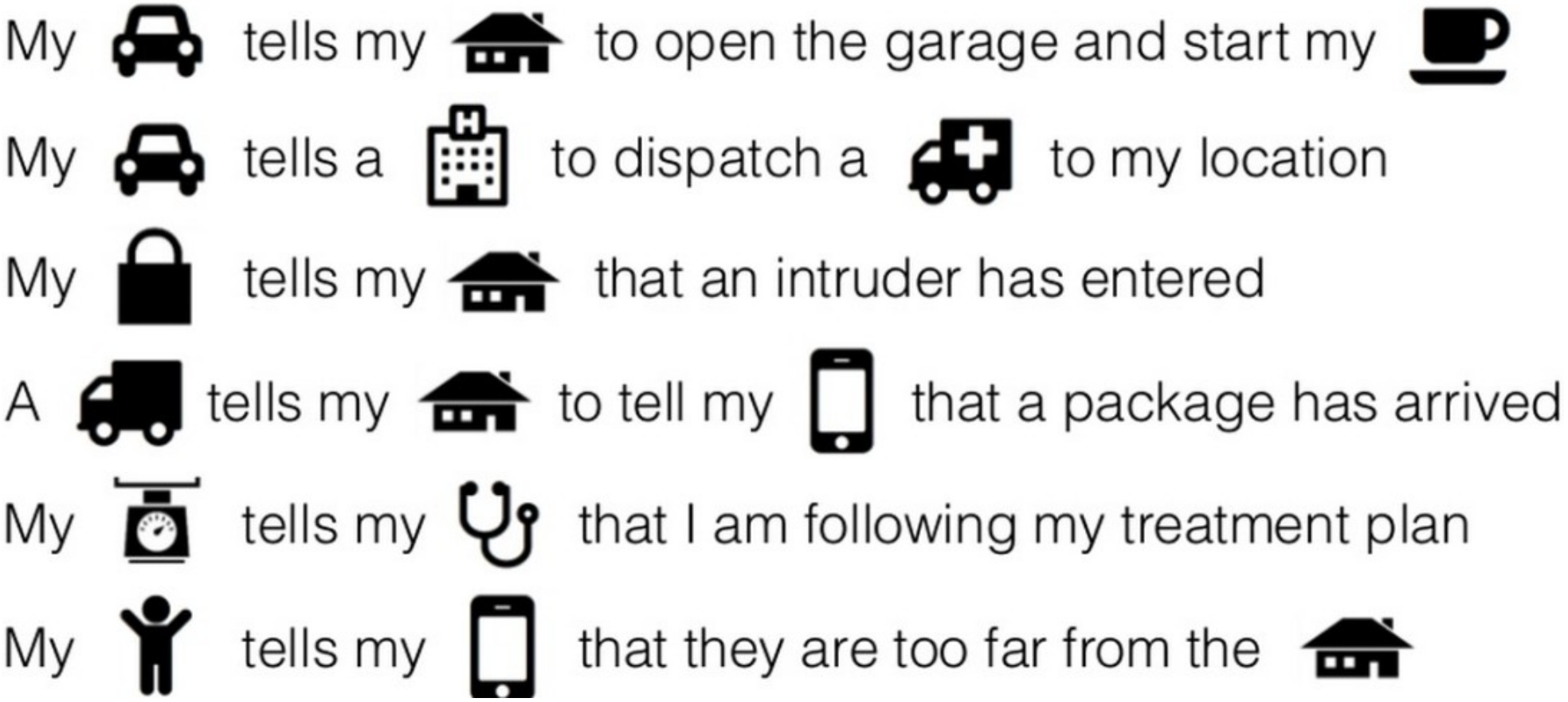
- end-to-end tracking
- theft prevention
- real-time updates
- fleet monitoring

Smarter Healthcare

- smart scales
- in-home monitoring
- assisted living
- physician messaging

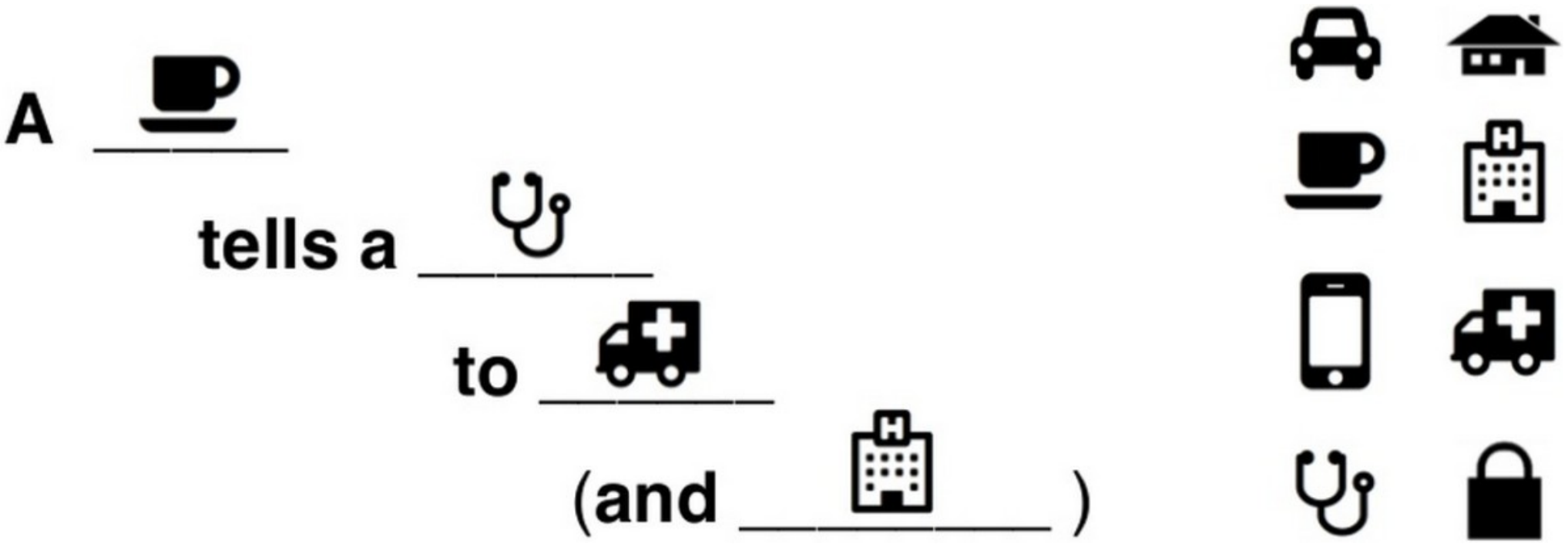
https://www.slideshare.net/BryanBoyd/mqtt-austin-api

(前言) 萬物將會相連



https://www.slideshare.net/BryanBoyd/mqtt-austin-api

(前言) 訊息的傳遞是個大問題



情境：
我的咖啡杯 > 傳訊息給醫生 > 請他叫救護車送我去醫院 > 因為我攝取太多咖啡因

https://www.slideshare.net/BryanBoyd/mqtt-austin-api

MQTT

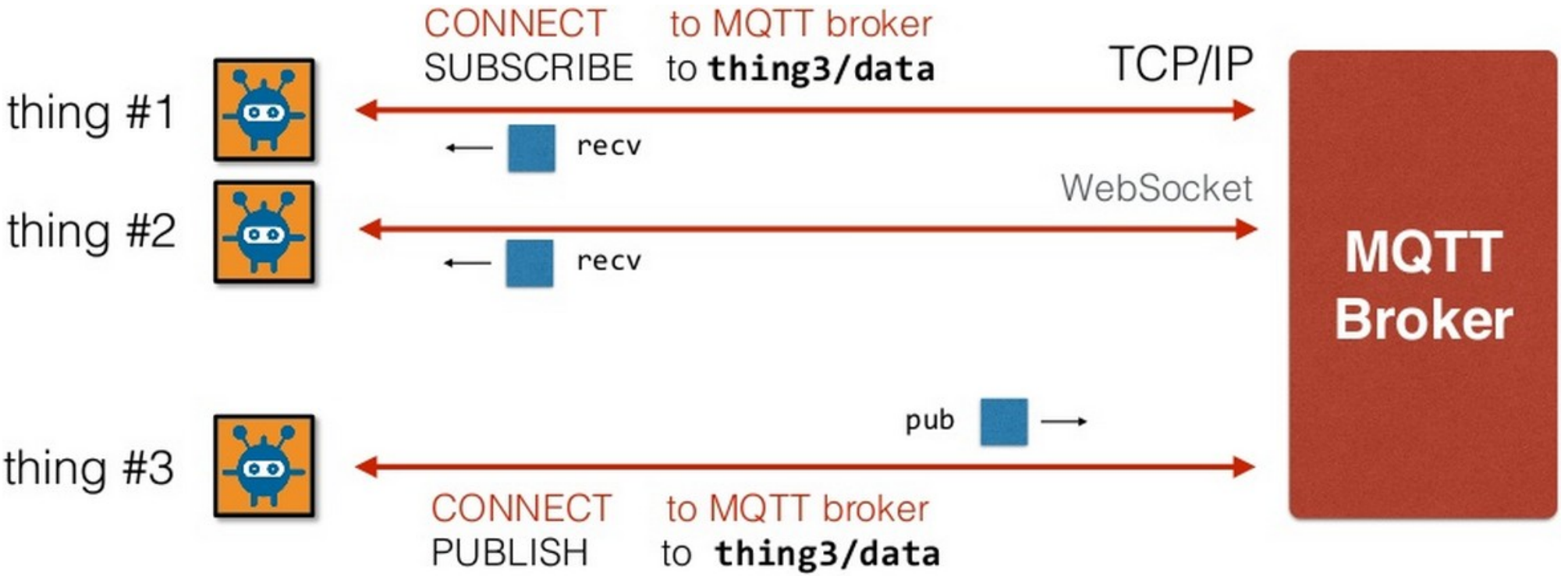
- 輕量級的 IoT 訊息傳輸協定
  - 開放：公開的標準，有 40+ 以上的用戶端實做
  - 輕量：最小資料傳輸 + 小型用戶端 (kb 等級)
  - 可靠：提供 QoS 確保服務品質
  - 簡單 :43 頁的規格書 ,connect + publish + subscribe

https://www.slideshare.net/BryanBoyd/mqtt-austin-api

MQTT Brokers

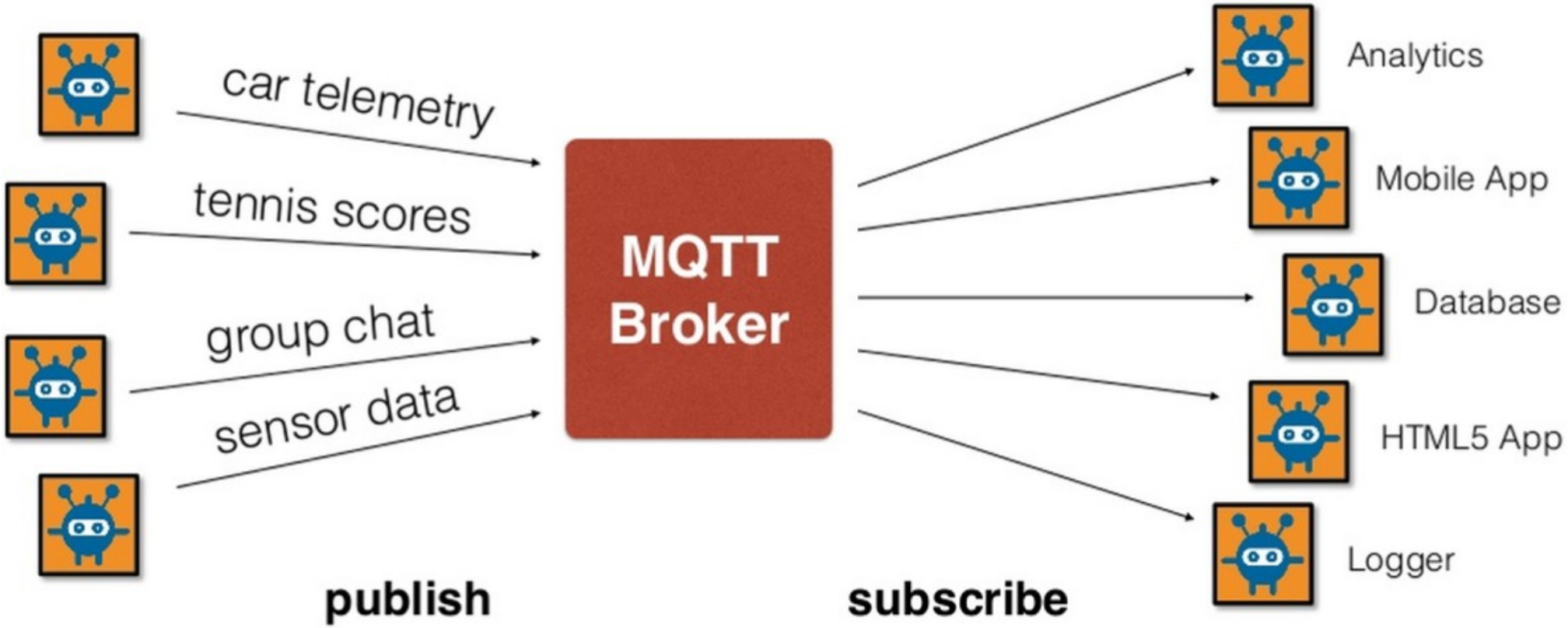
https://www.slideshare.net/BryanBoyd/mqtt-austin-api

bi-direction, async “push” 通訊



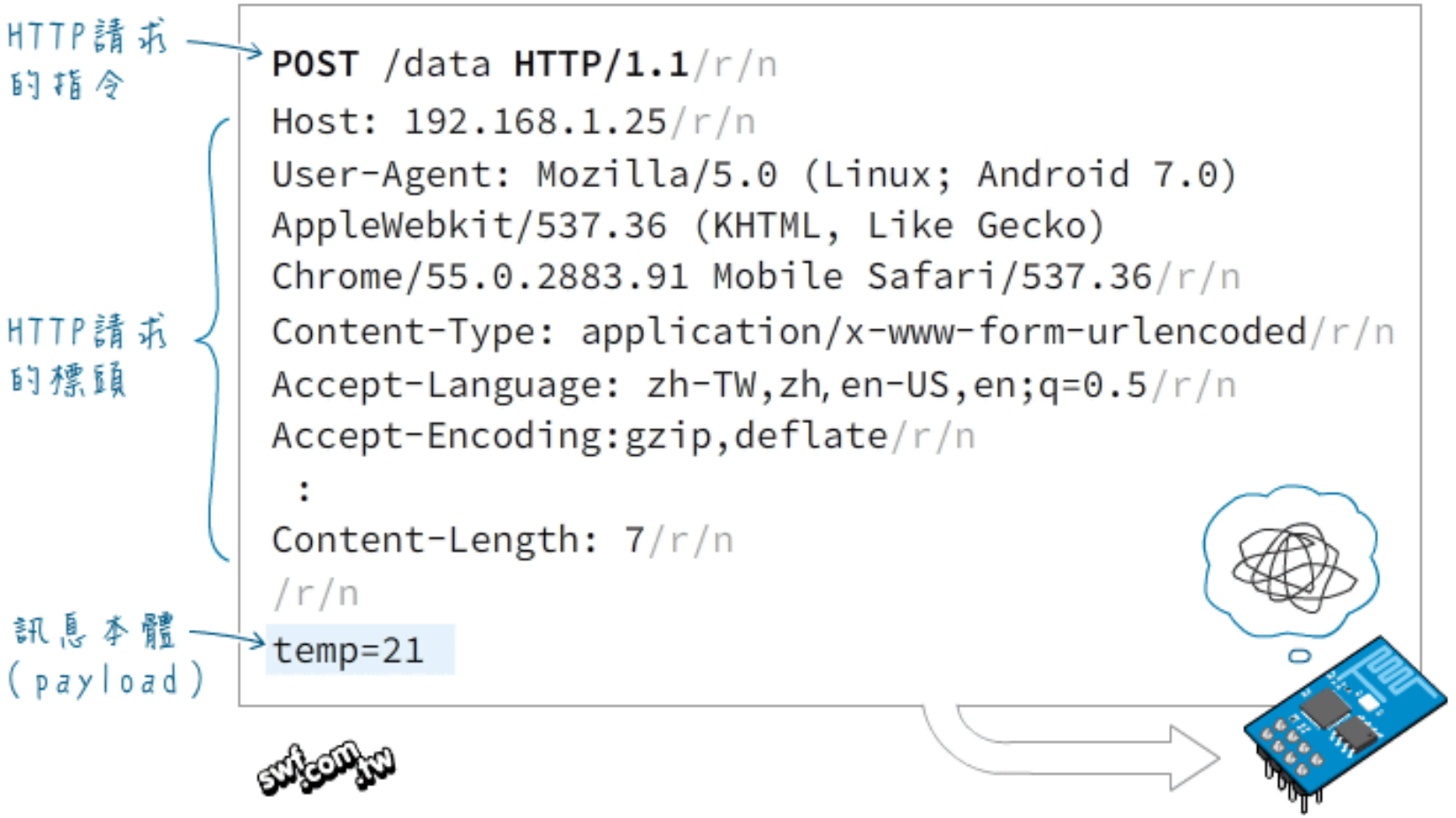
https://www.slideshare.net/BryanBoyd/mqtt-austin-api

發布 / 訂閱 , 降低發送端和接收端的耦合



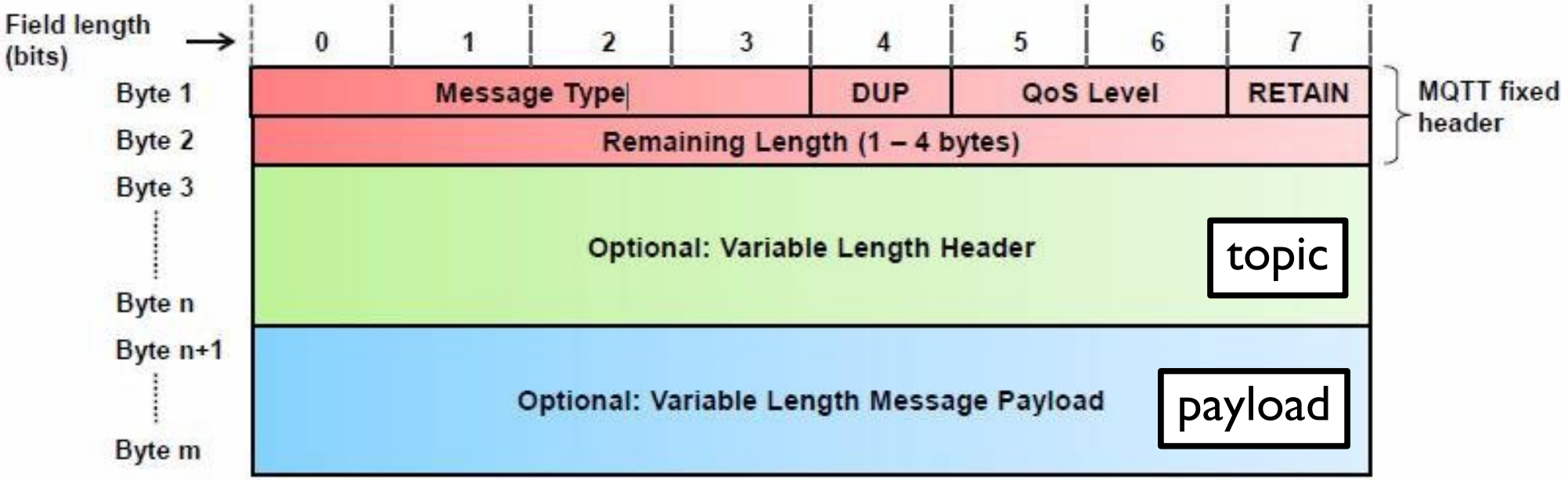
https://www.slideshare.net/BryanBoyd/mqtt-austin-api

HTTP 封包中，標頭的環境變數



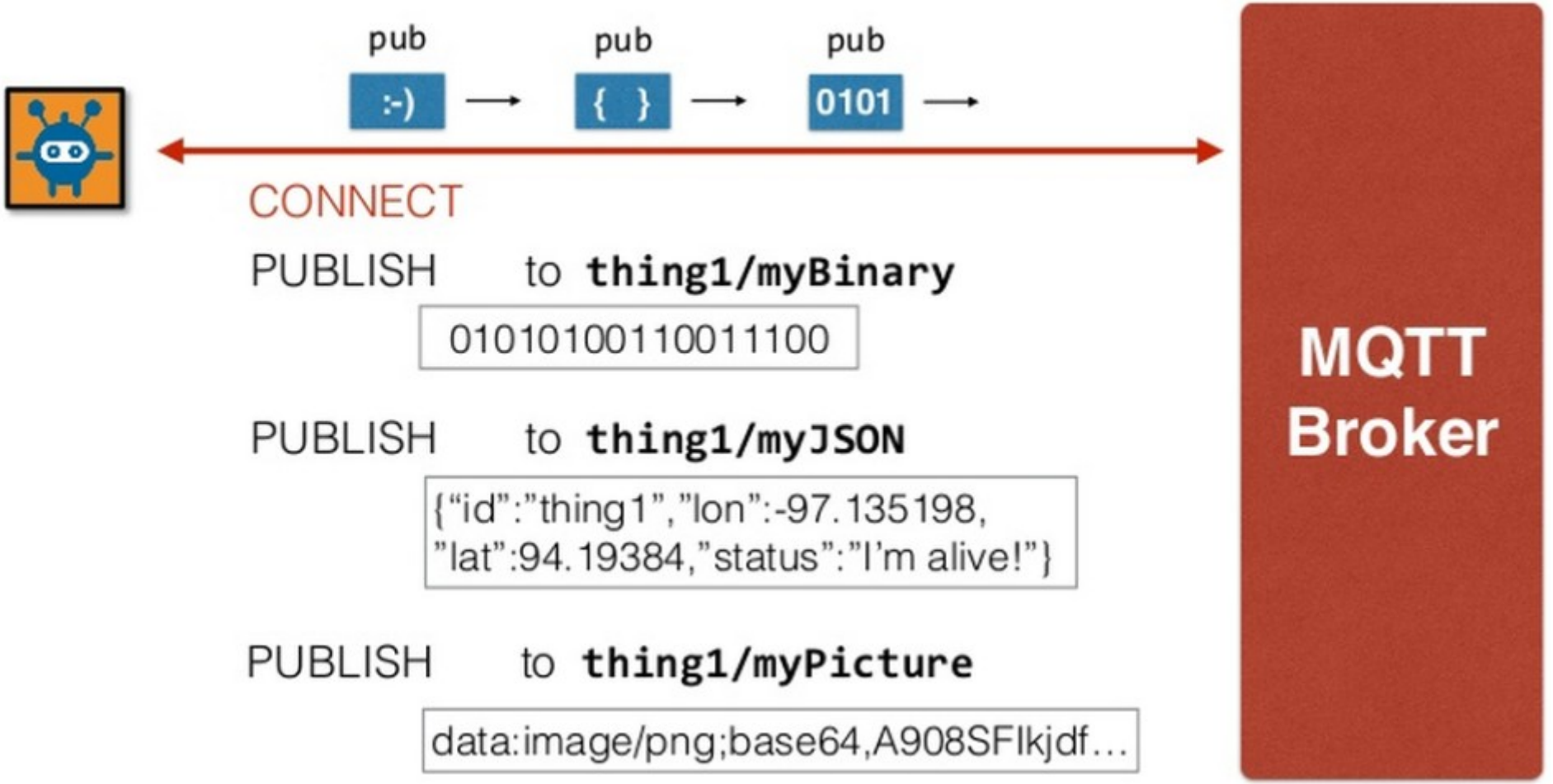
https://swf.com.tw/?p=1002

MQTT 封包的標頭只佔 2 位元



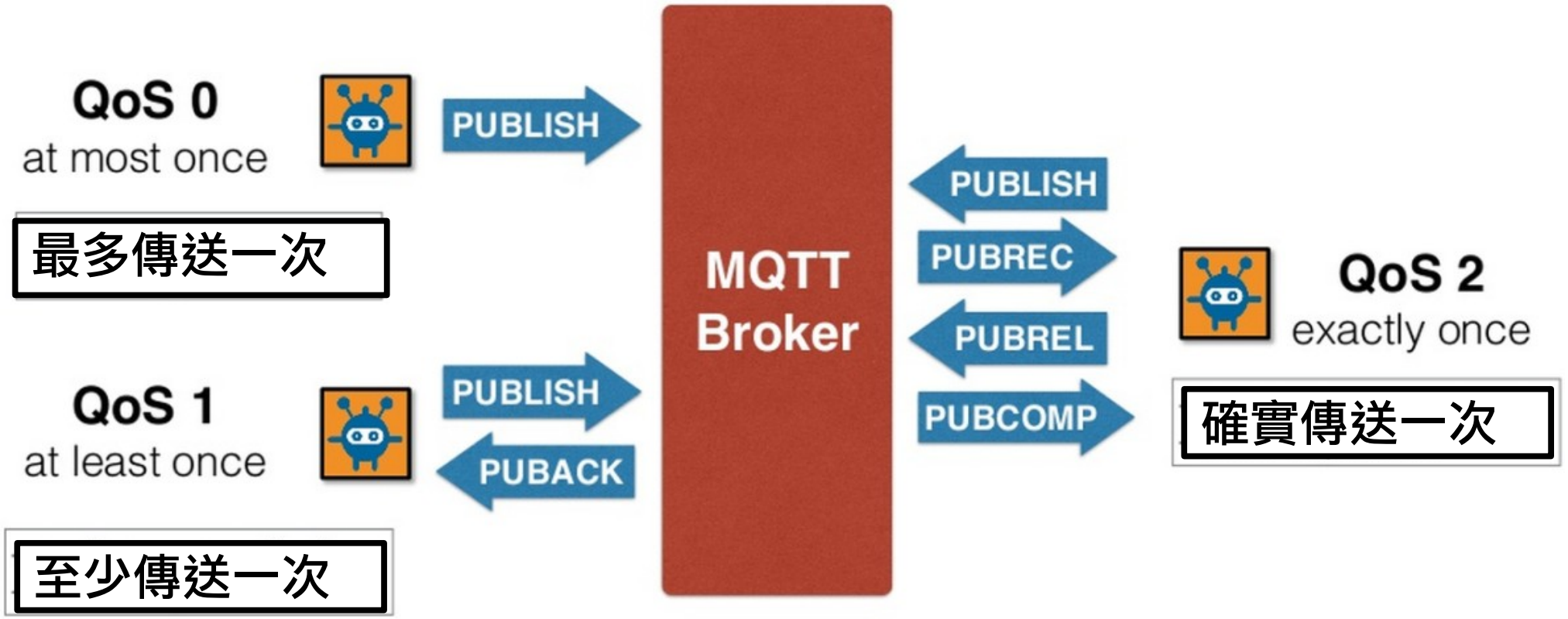
http://www.rfwireless-world.com/Tutorials/MQTT-tutorial.html

弹性的封包内容



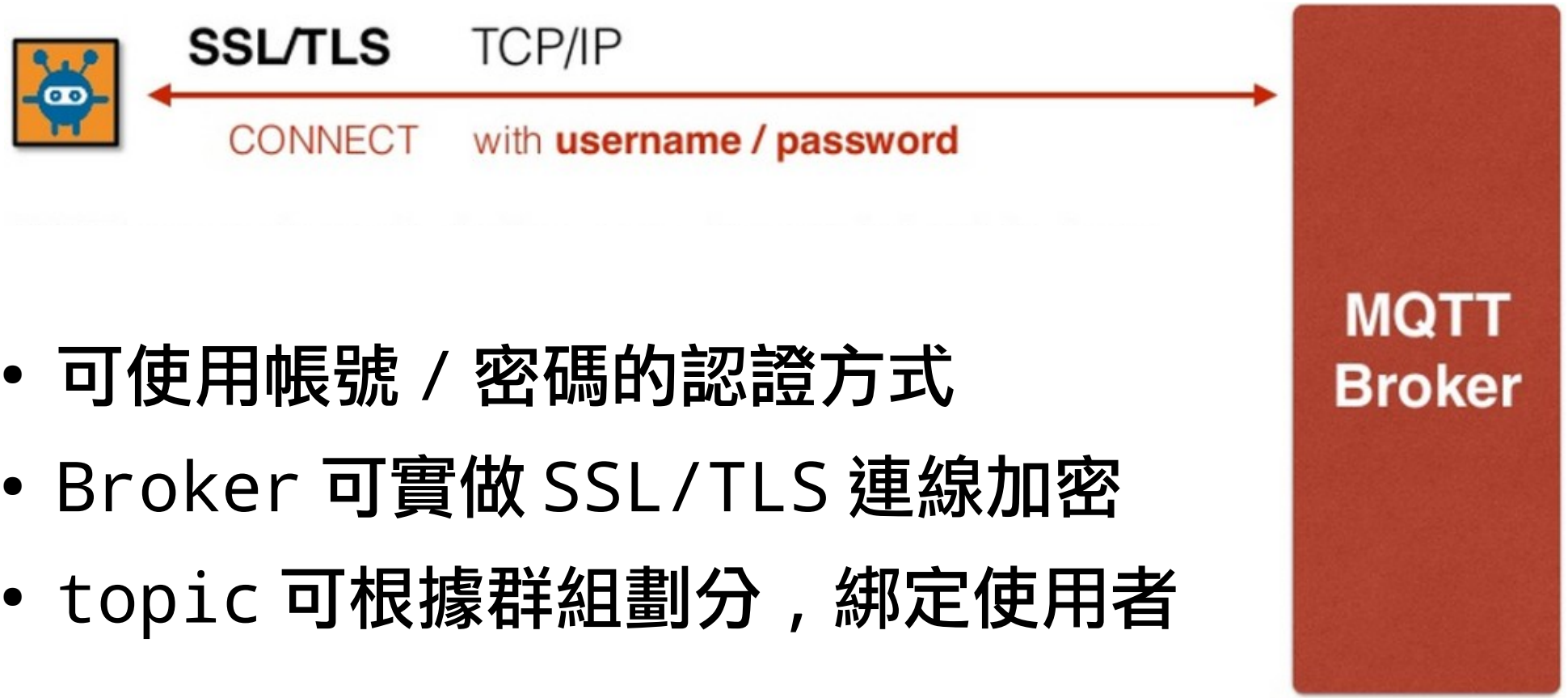
https://www.slideshare.net/BryanBoyd/mqtt-austin-api

服務品質 (QoS)



https://www.slideshare.net/BryanBoyd/mqtt-austin-api

安全性



- 可使用帳號 / 密碼的認證方式
- Broker 可實做 SSL/TLS 連線加密
- topic 可根據群組劃分，綁定使用者

https://www.slideshare.net/BryanBoyd/mqtt-austin-api

在 Raspberry Pi 上使用 MQTT

使用 Python

- `$ sudo pip3 install paho-mqtt`

- `paho.mqtt` 模組提供三個類別
  - `Publish`(一次性的發送)
  - `Subscribe`(一次性的接收)
  - `Client`(新建客戶端、連接、訂閱、發送、回呼函數)

訂閱 TTN 需驗證的 Broker

```
def on_connect(client, userdata, flags, rc):
    print("Connected with result code: {}".format(str(rc)))
    client.subscribe("+/devices/#")

def on_message(client, userdata, msg):
    print("topic: {}, message: {}".format(msg.topic, str(msg.payload)))

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

user, password = "<AppID>", "<AppKey>"
client.username_pw_set(user, password)
client.connect("<TtnRouter>", 1883, 60)

client.loop_forever()
```

修改 ttn_var.py 程式

```
40
41 #
42 # TTN Application
43 #
44 user, password = "weather_application", "ttn-account-v2.zPnOrzsTVP-Lk-CDGvzF
   Ky15NjRL0YVgCN-Ay0QR7y4"
45 ttn_router = "eu.thethings.network"
46
47
48
```

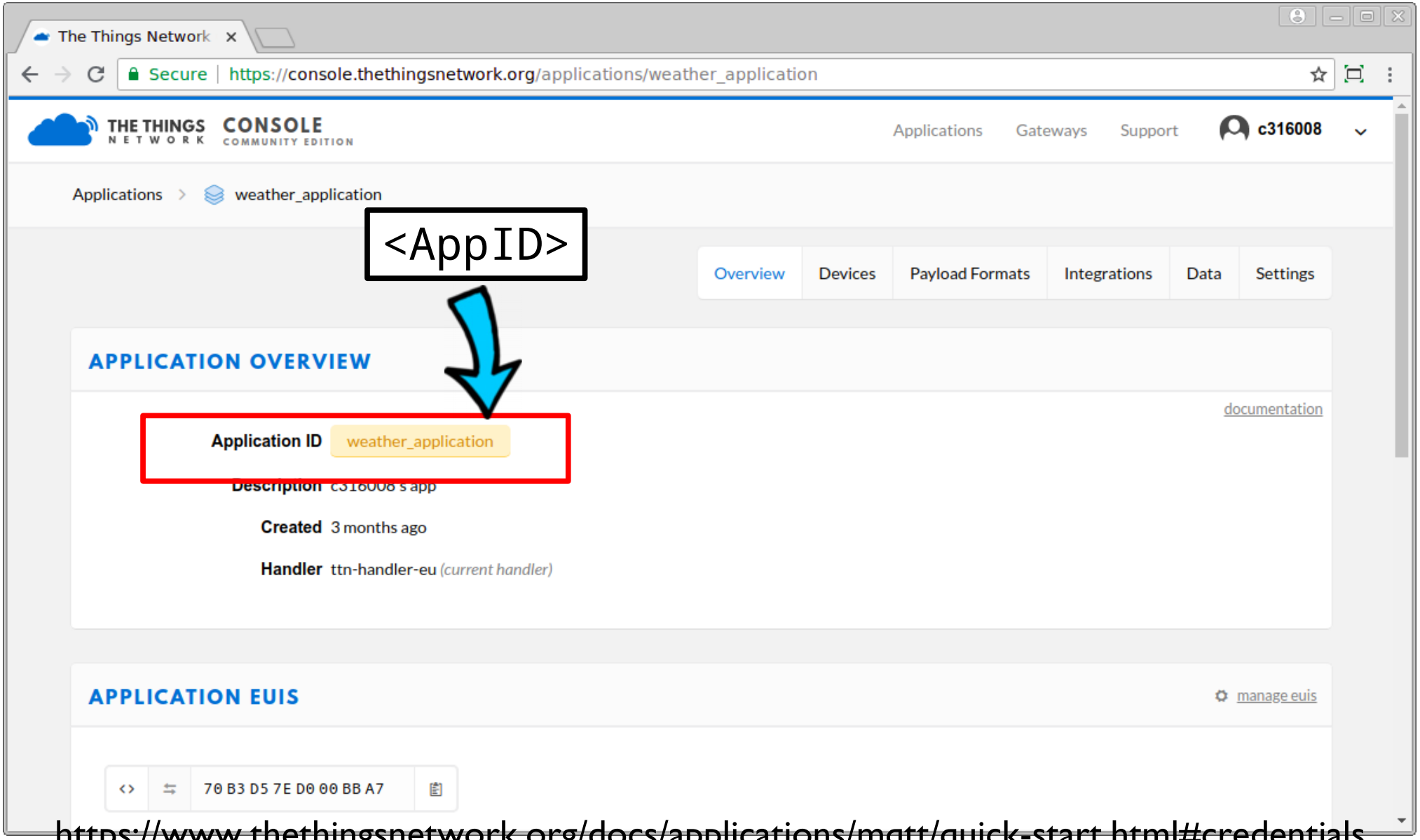
<AppID>

<AppKey>

<TtnRouter>

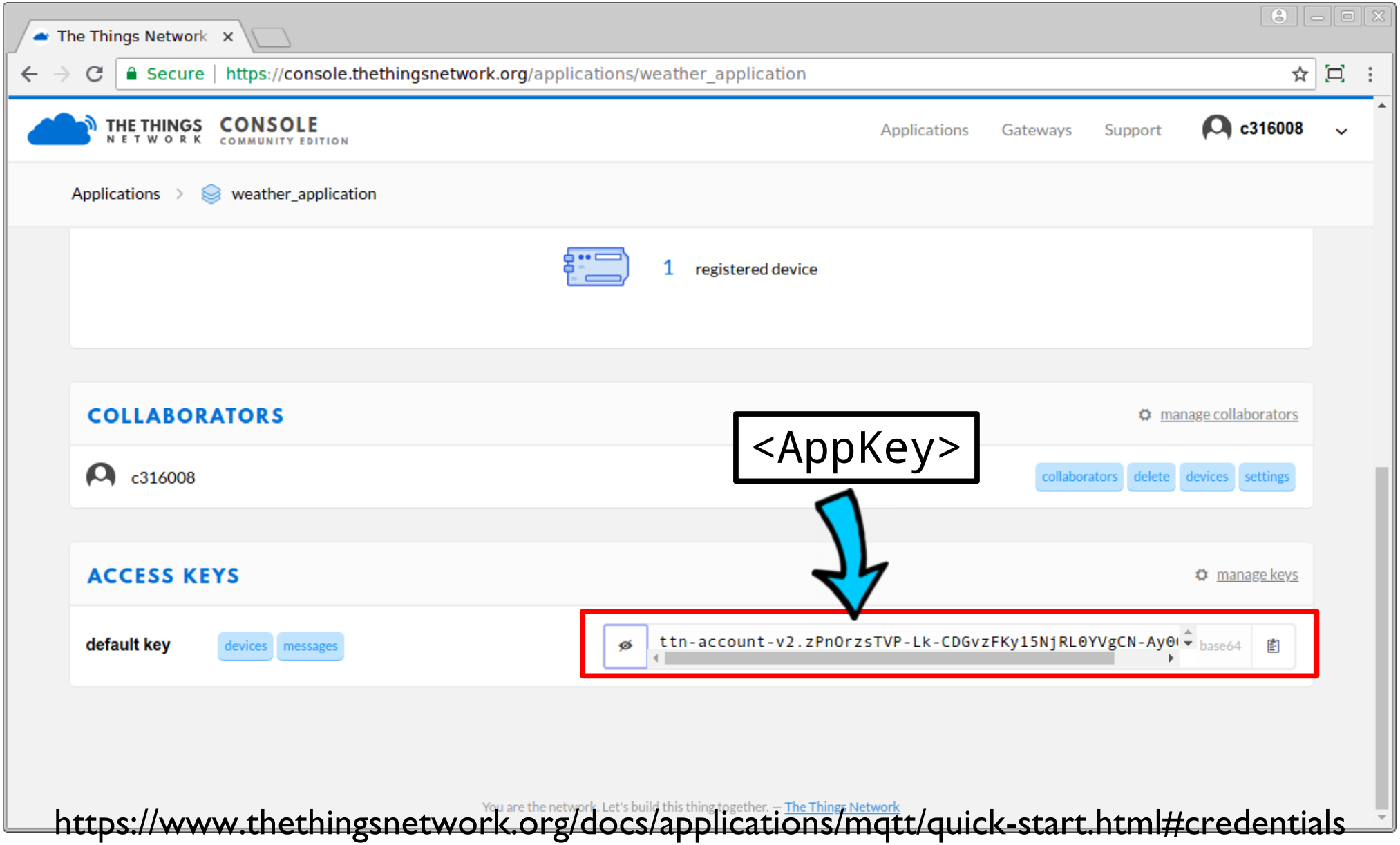
<AppID>, <AppKey>, <TtnRouter> 換成自己的設定

<AppID>



https://www.thethingsnetwork.org/docs/applications/mqtt/quick-start.html#credentials

<AppKey>

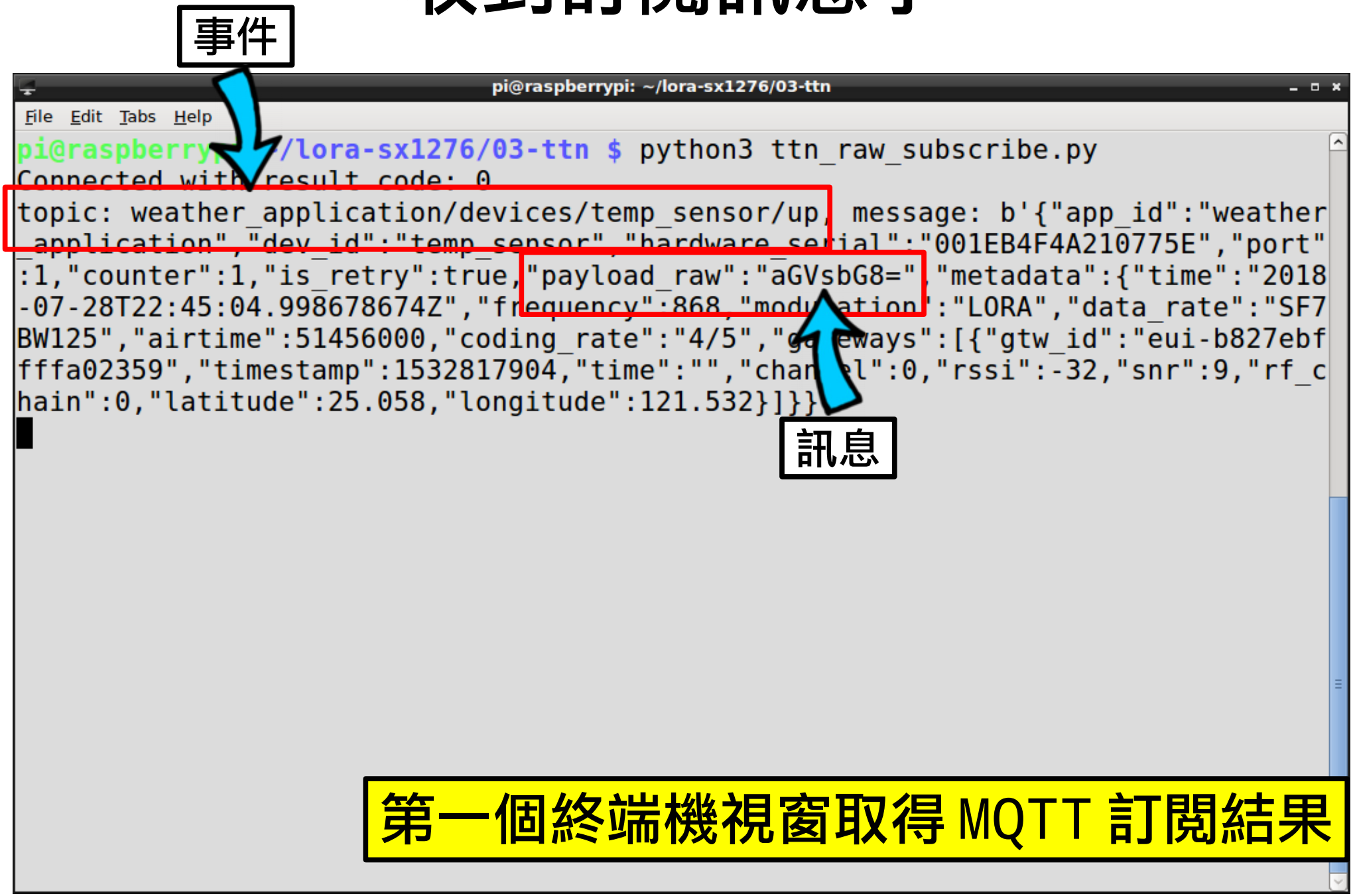


https://www.thethingsnetwork.org/docs/applications/mqtt/quick-start.html#credentials

訂閱 TTN 的 APP 資訊

再執行一次 ttn_app.py 程式

收到訂閱訊息了


事件
pi@raspberrypi: ~/lora-sx1276/03-ttn
File Edit Tabs Help
pi@raspberrypi:~/lora-sx1276/03-ttn $ python3 ttn_raw_subscribe.py
Connected with result code: 0
topic: weather_application/devices/temp_sensor/up, message: b'{"app_id":"weather_application","dev_id":"temp_sensor","hardware_serial":"001EB4F4A210775E","port":1,"counter":1,"is_retry":true,"payload_raw":"aGVsbG8=","metadata":{"time":"2018-07-28T22:45:04.998678674Z","frequency":868,"modulation":"LORA","data_rate":"SF7BW125","airtime":51456000,"coding_rate":"4/5","gateways":[{"gtw_id":"eui-b827ebfffffa02359","timestamp":1532817904,"time":"","channel":0,"rssi":-32,"snr":9,"rf_chain":0,"latitude":25.058,"longitude":121.532}]}}
訊息
第一個終端機視窗取得 MQTT 訂閱結果

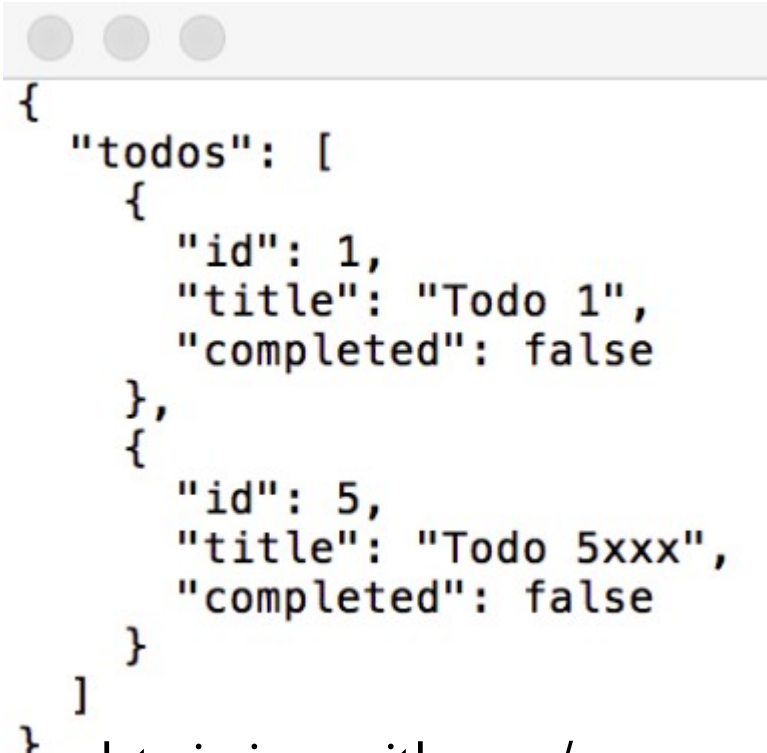
要先修改 ttn_var.py 程式

DEMO
ttn_raw_subscribe.py

```
$ cd ~/lora-sx1276/03-ttn
$ python3 ttn_raw_subscribe.py
```

使用 JSON 打包訊息

- JSON(JavaScript Object Notation)是一種資料結構
  - 物件(object)以 { } 表示
  - 鍵 / 值(collection)以 : 表示
  - 陣列(array)以 [ ] 表示
  - 最外層用 {} 包起來



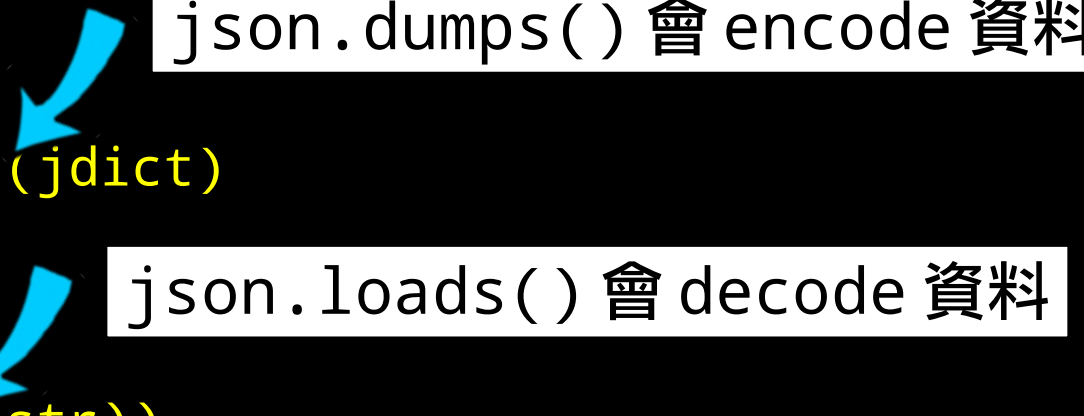
```
{
  "todos": [
    {
      "id": 1,
      "title": "Todo 1",
      "completed": false
    },
    {
      "id": 5,
      "title": "Todo 5xxx",
      "completed": false
    }
  ]
}
```

https://www.ssaurel.com/blog/parse-and-write-json-data-in-java-with-gson/

Python 和 JSON

- $ python3

```
>>> import json
>>> jdict = {"dev_id": "temp_sensor","payload_raw":"aGVsbG8="}
>>> type(jdict)
<type 'dict'>
>>> jstr = json.dumps(jdict)
>>> type(jstr)
<type 'str'>
>>> type(json.loads(jstr))
<type 'dict'>
>>> print(jdict)
{'dev_id': 'temp_sensor', 'payload_raw': 'aGVsbG8='}
>>> print(json.loads(jstr))
{'dev_id': 'temp_sensor', 'payload_raw': 'aGVsbG8='}
```



使用 JSON+Base64 得到 Plain Text

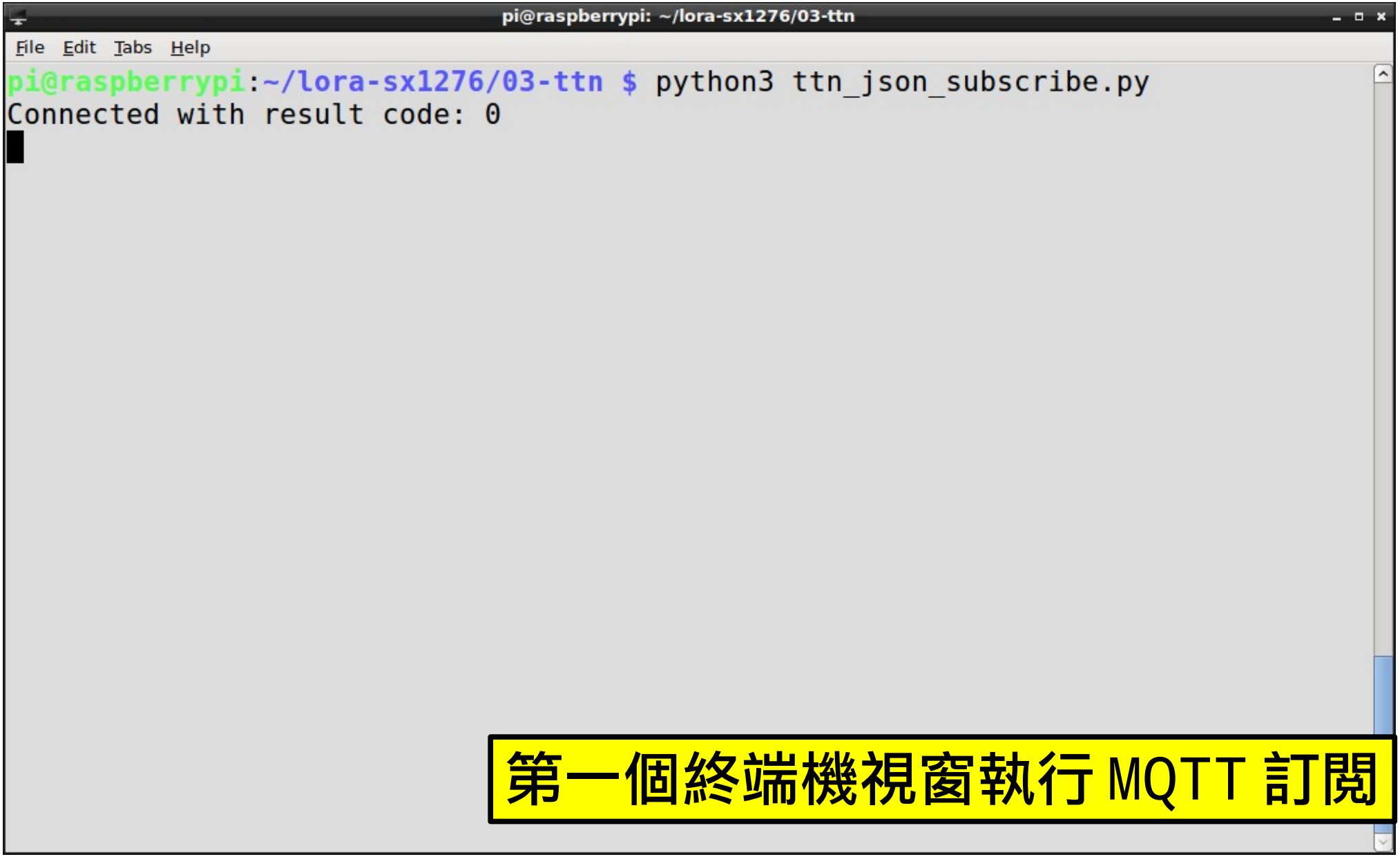
```
import base64

def on_message(client, userdata, msg):
    #print("topic: {}, message: {}".format(msg.topic, str(msg.payload)))
    jdict = json.loads( str(msg.payload, 'utf-8') )
    mtime = jdict['metadata']['time']
    dev_id = jdict['dev_id']
    payload_raw = jdict['payload_raw']
    payload_plain = base64.b64decode(jdict['payload_raw'])
    print("payload_decode:{}".format(payload_plain))

client = mqtt.Client()
client.on_message = on_message
user, password = "<AppID>", "<AppKey>"
client.username_pw_set(user, password)
client.connect("<TtnRouter>", 1883, 60)

client.loop_forever()
```

使用 Base64 Decode 內容

DEMO
ttn_json_subscribe.py

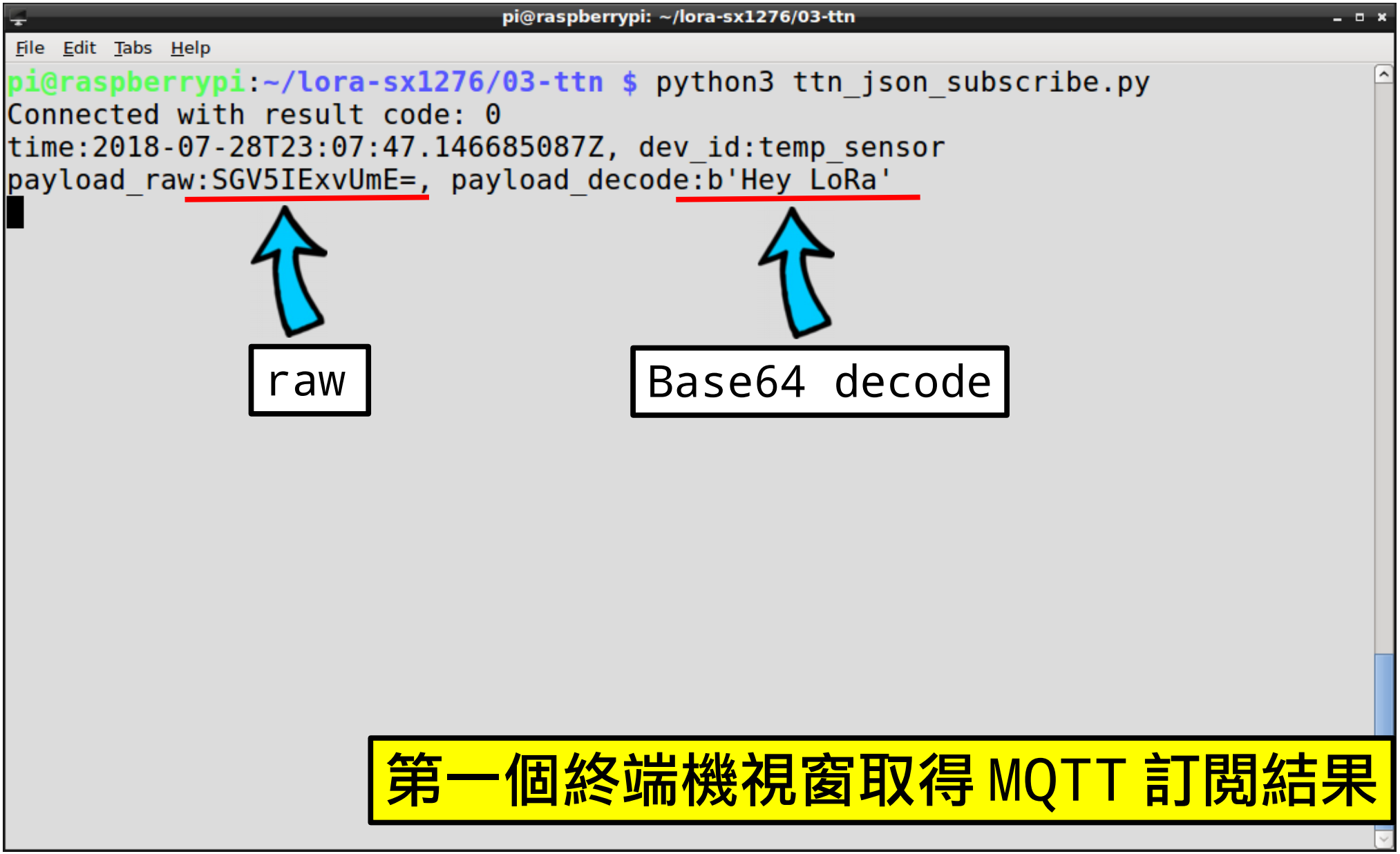
```
$ cd ~/lora-sx1276/03-ttn
$ python3 ttn_json_subscribe.py
```

訂閱 TTN 的 APP 資訊

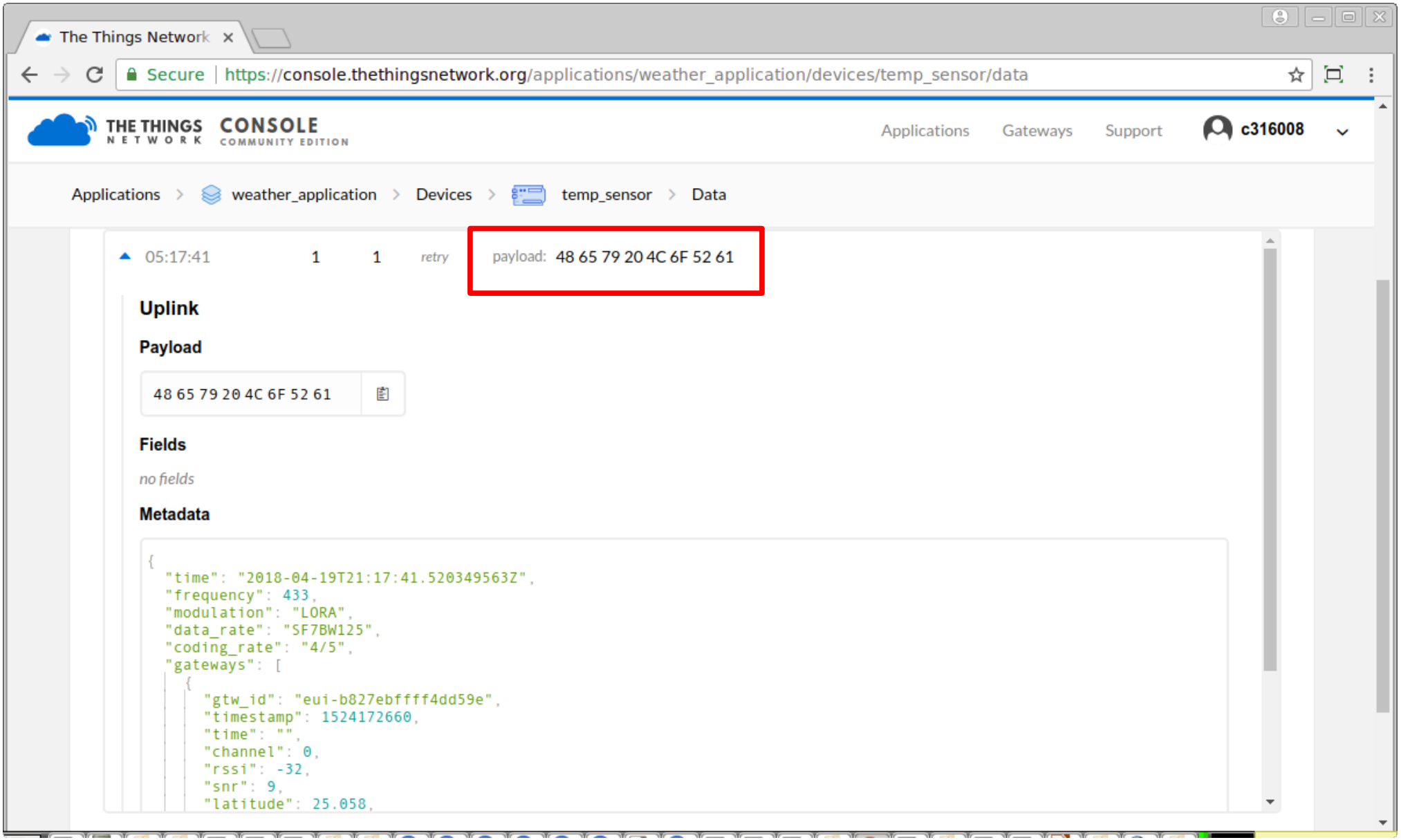


再執行一次 ttn_app.py 程式

收到明文的訂閱訊息了


pi@raspberrypi: ~/lora-sx1276/03-ttn
File Edit Tabs Help
pi@raspberrypi:~/lora-sx1276/03-ttn $ python3 ttn_json_subscribe.py
Connected with result code: 0
time:2018-07-28T23:07:47.146685087Z, dev_id:temp_sensor
payload_raw:SGV5IExvUmE=, payload_decode:b'Hey LoRa'
raw
Base64 decode
第一個終端機視窗取得 MQTT 訂閱結果

TTN 的 APP 資訊

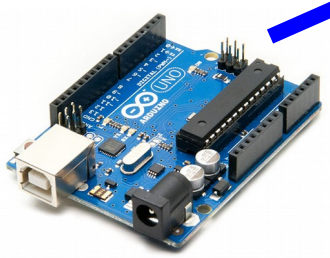

The Things Network
Secure | https://console.thethingsnetwork.org/applications/weather_application/devices/temp_sensor/data
THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
c316008
Applications > weather_application > Devices > temp_sensor > Data
05:17:41 1 1 retry payload: 48 65 79 20 4C 6F 52 61
Uplink
Payload
48 65 79 20 4C 6F 52 61
Fields
no fields
Metadata
{
"time": "2018-04-19T21:17:41.520349563Z",
"frequency": 433,
"modulation": "LORA",
"data_rate": "SF7BW125",
"coding_rate": "4/5",
"gateways": [
{
"gtw_id": "eui-b827ebffff4dd59e",
"timestamp": 1524172660,
"time": "",
"channel": 0,
"rssi": -32,
"snr": 9,
"latitude": 25.058,

串接 TTN

Device

Gateway

Network Server



THE THINGS NETWORK

收到訂閱資料成功

MQTT

LoRa 網路

TCP/IP 網路

https://www.dreamstime.com/63482375

使用 TTN 注意事項

上傳下載限制

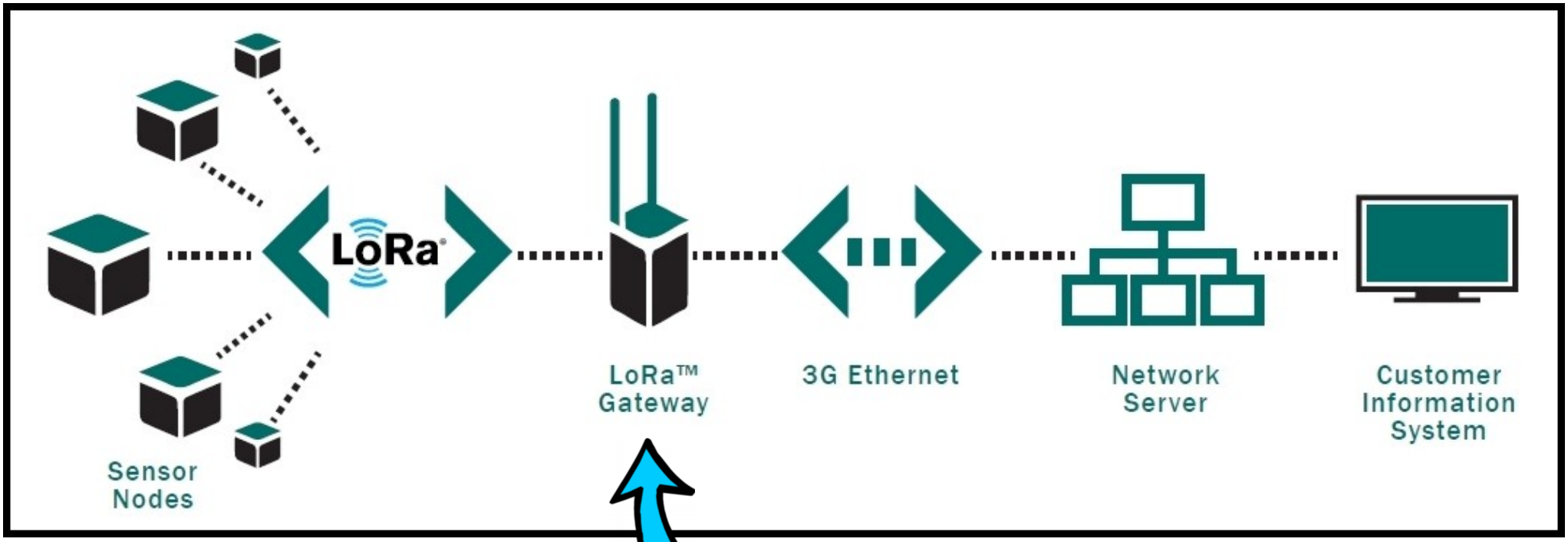
- data rate, packet size, 30 seconds uplink and 10 messages downlink per day Fair Access Policy

- 計算條件：
  - 8 frequencies
  - 5% receive duty cycle on the gateway
  - 86,400 seconds in a day
  - 1,000 nodes
  - => 30 = (8 × 86,400 × 0.05) / 1,000

http://bit.ly/2qEeqKc

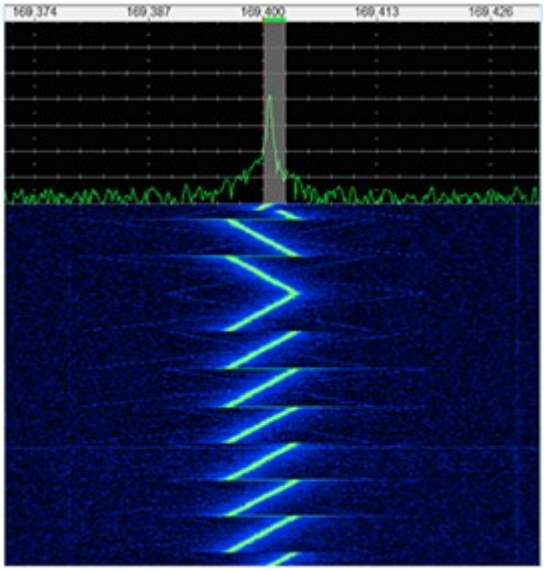
用 Pi+LoRa 做微型物聯網閘道器

實做 LoRa Gateway 與 LoRa Node



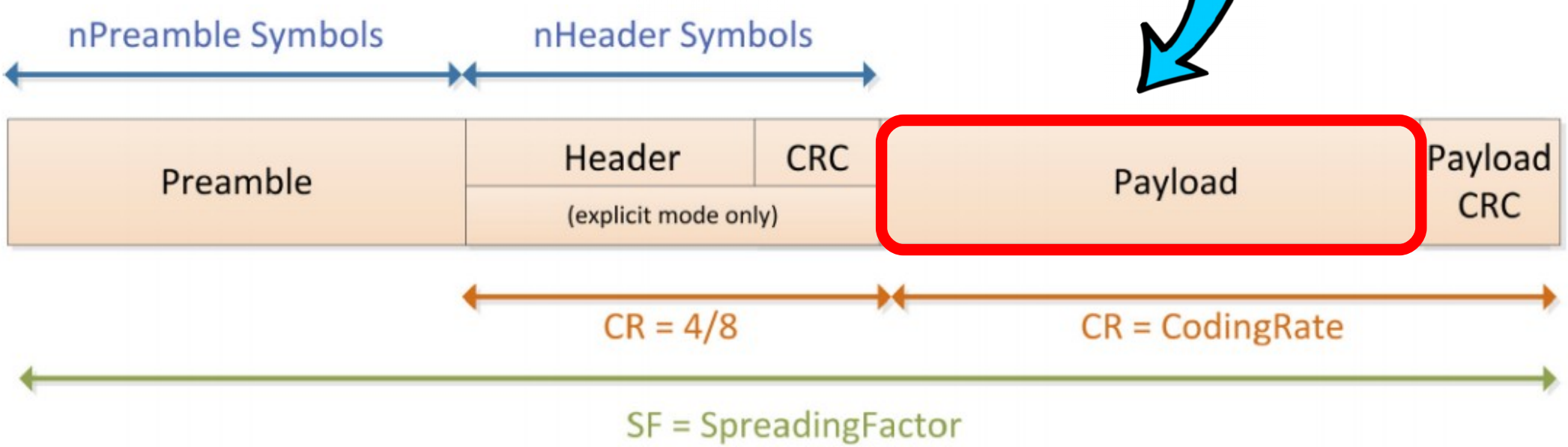
https://www.iot-now.com/2015/12/02/39616-lora-looks-good-to-go/

LoRa 封包格式



非必要

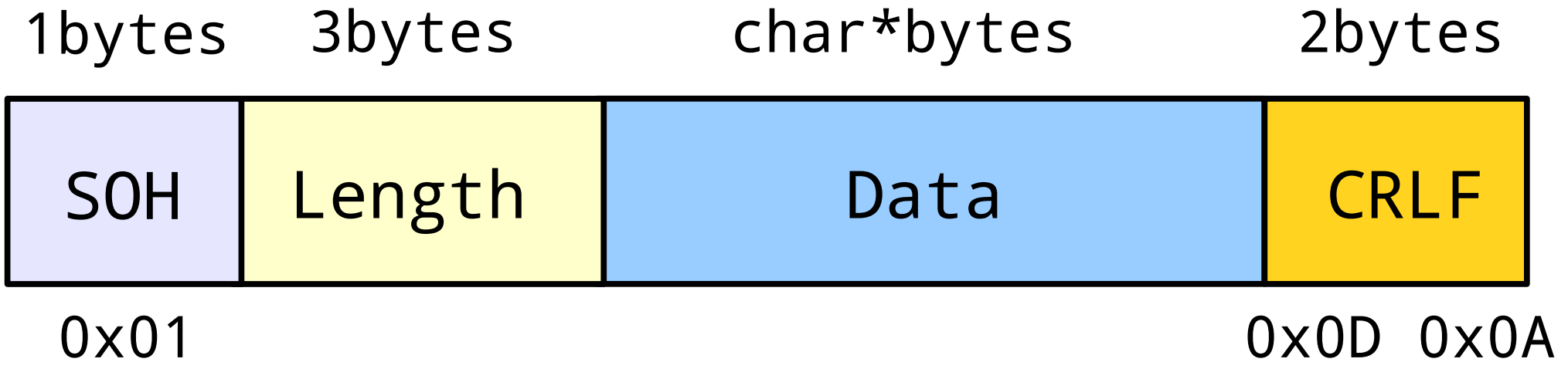
傳送資料 1234



https://www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf

設計 Payload

- 限制：
  - 只能傳送 0-F 資料 (Hex)
  - 每個符號以 ASCII 方式傳送 (1bytes)
- Data 資料：
  - 包含 id 與 content, 以 JSON 格式封裝



設計 ACK 與重送機制

- 接收端 ACK 機制與傳送時間
  - 當接收端收到封包後，回傳 ACK(0x06) 與 id
  - 回傳時間與封包長度有關

- 發送端重送機制類似 class A
  - 傳送端發送完畢後，將等待一段時間等待 ACK
  - 如果沒有收到 ACK，將用 ALOHA 方式重送
  - 最大重送次數 <3

打包封包 (packer.py)

```
SOH  = "01"    # 0x01
ACK  = "06"    # 0x06
CRLF = "43524C46"  # CRLF \r\n

def Pack_Str(string):
    data = string.encode('utf-8')
    length = len(data)

    if length < 10:
        length = str(0) + str(0) + str(length)
    elif length >= 10 and length < 100:
        length = str(0)  + str(length)
    else:
        length = str(length)

    payload = bytes(SOH, 'utf-8') + str(length).encode(encoding='utf-8') + data +
bytes(CRLF, 'utf-8')
    return [length, payload]
```

CRLF SOH length Data

Gateway 接收資料 (gw_rx.py)

```
def on_rx_done(self):
    payload = self.read_payload(nocheck=True)
    data = ''.join([chr(c) for c in payload])
    _length, _data = packer.Unpack_Str(data)

    self.set_dio_mapping([1,0,0,0,0
    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(c), 0) for c in _ack]
    self.write_payload(ack)
    self.set_mode(MODE.TX)
```

解開封包

發送 ACK(Tx)

DEMO
gw_rx.py

```
$ cd ~/lora-sx1278/04-gateway
$ python3 gw_rx.py -f 868
```

Sensor Node 發送資料 (gw_tx.py)

```
def start(self):
    while True:
        if len(rawinput) < 200:
            data = {"id":self._id, "data":rawinput}
            _length, _payload = packer.Pack_Str( json.dumps(data) )
            data = [int(hex(c), 0) for c in _payload]
            self.set_mode(MODE.SLEEP)
            self.set_dio_mapping([1,0,0,0,0,0])    # TX
            self.clear_irq_flags(TxDone=1)
            self.set_mode(MODE.STDBY)
            sleep(.5)
            self.write_payload(data)
            self.set_mode(MODE.TX)

def on_rx_done(self):
    self.clear_irq_flags(RxDone=1)
    payload = self.read_payload(nocheck=True)
    data = ''.join([chr(c) for c in payload])
    self.rx_done = True
    self.set_dio_mapping([1,0,0,0,0,0])    # TX
    self.set_mode(MODE.STDBY
    self.clear_irq_flags(TxDone=1)
```

打包封包

發送封包 (Tx)

收完封包後切回 Tx 模式

DEMO
gw_tx.py

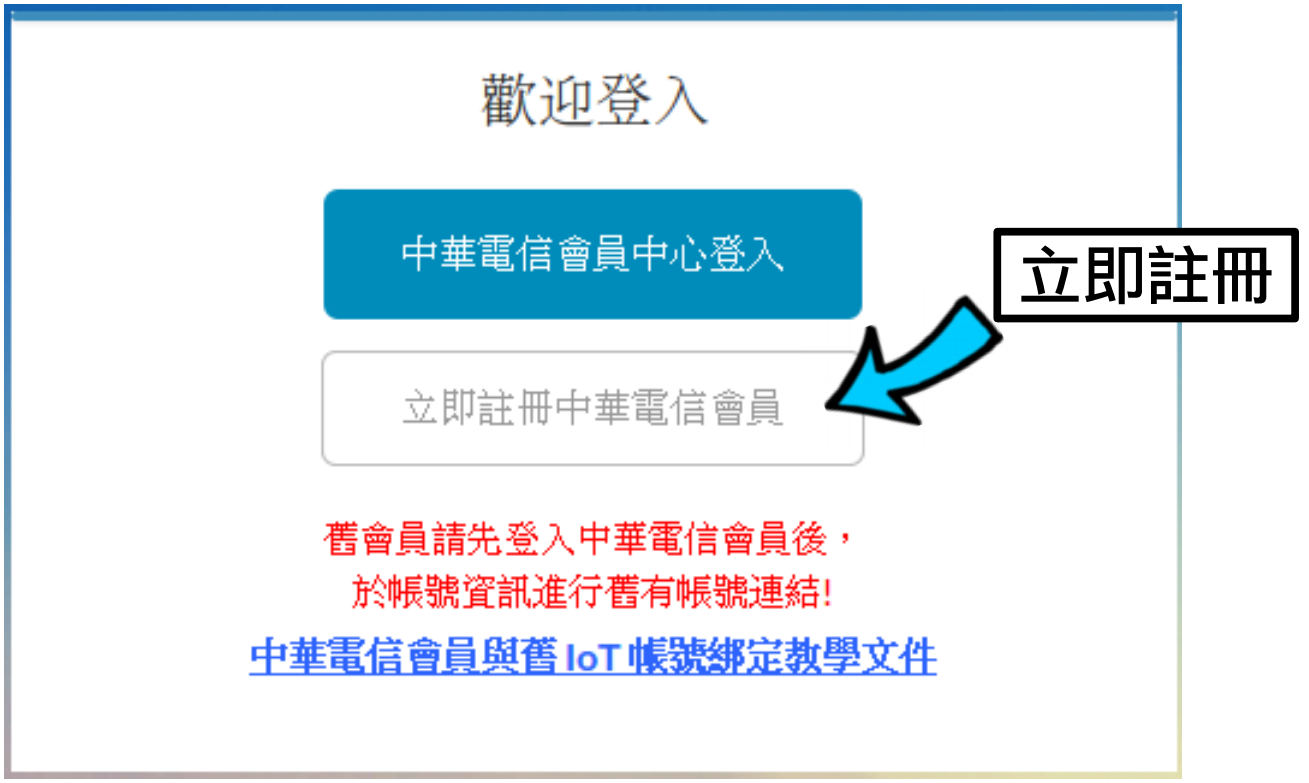
```
$ cd ~/lora-sx1278/04-gateway
$ python3 gw_tx.py -f 868
```

如何串接到後端應用程式伺服器？

中華電信物聯網大平台

https://iot.cht.com.tw

註冊



1．送簡訊給驗證碼（手機）
2．送簡訊給密碼（手機）

https://iot.cht.com.tw/iot/

專案 > 設備 > 感測器


智慧聯網平台
SMART PLATFORM
開發者中心
專案管理
應用服務
帳號資訊
登出
Select Language
Powered by Google Translate
首頁 / 專案管理
專案管理
增加專案
增加新專案
pulse
聯絡我們
中華電信股份有限公司版權所有 © 2017 Chunghwa Telecom Co., Ltd. All Rights Reserved.

專案名稱和 API KEY

設定專案內容


智慧聯網平台
SMART PLATFORM
開發者中心
專案管理
應用服務
帳號資訊
登出
Select Language
Powered by Google Translate
首頁 / 專案管理
專案管理
增加專案
pulse
lora_demo
點選可設定專案內容
中華電信股份有限公司版權所有 © 2017 Chunghwa Telecom Co., Ltd. All Rights Reserved.

對專案增加新的設備


智慧聯網平台
SMART PLATFORM
開發者中心
專案管理
應用服務
帳號資訊
登出
Select Language
Powered by Google Translate
首頁 / 專案管理 / 設備管理
設備管理
專案名稱: lora_demo
增加設備
返回
聯絡我們
中華電信股份有限公司版權所有 © 2017 Chunghwa Telecom Co., Ltd. All Rights Reserved.
新增設備

設備名稱和金鑰


建立設備名稱
設備管理
基本資料
設備名稱
pi3
設備描述
設備描述...
類型
通用設備
Modbus工業設備
ONVIF影像設備
經度
經度 ...
緯度
緯度 ...
URI
URI ...
設備金鑰
DKSW472G
B
取消
下一頁
設備金鑰(API_KEY)


設備管理
基本資料
擴充屬性資訊
提供客製化擴充屬性設置，滿足額外所需的設備屬性資訊
屬性名稱(key)
屬性數值(value)
Key
Value
+
取消
儲存

對設備增加新的感測器


智慧聯網平台
SMART PLATFORM
開發者中心
專案管理
應用服務
帳號資訊
登出
Select Language
Powered by Google Translate
首頁 / 專案管理 / 設備管理
設備管理
新增感測器
專案名稱: lora_demo
+ 增加感測器
+ 增加設備
返回
快速搜尋:
設備編號
設備名稱
設備描述
設備類型
功能
4883523150
pi3
general
顯示第 1 至 1 項結果，共 1 項
上頁
1
下頁
設備: pi3 (編號:4883523150) 感測項目
設備編號 (DEVICE_NUMBER)
中華電信股份有限公司版權所有 © 2017 Chunghwa Telecom Co., Ltd. All Rights Reserved.

建立感測器名稱


感測器 id(SID)
感測器管理
基本資料
其他資料
識別編號(ID)
random_data_01
識別編號只允許輸入英文或數字或底線符號
顯示名稱
random_data
描述
感測器描述...
感測器名稱
類型
數值
文字
開關
圖像
單位
單位 ...
取消
下一頁


感測器管理
基本資料
其他資料
個人客製化屬性設置，您可以於底下增加感測器屬性資訊
屬性名稱(key)
屬性數值(value)
Key
Value
+
取消
儲存

基本設定完成了


IOT 智慧聯網平台
SMART PLATFORM
開發者中心
專案管理
應用服務
帳號資訊
登出
Select Language
Powered by Google Translate
設備管理
專案名稱: lora_demo
增加感測器
增加設備
返回
快速搜尋:
設備編號
設備名稱
設備描述
設備類型
功能
4883523150
pi3
general
顯示第 1 至 1 項結果，共 1 項
上頁
1
下頁
設備: pi3 (編號:4883523150) 感測項目
IOT
random_data
設定

開始送訊息到物聯網大平台吧

```
host = "iot.cht.com.tw"
topic = "/v1/device/<DEVICE_NUMBER>/rawdata"
user, password = "<PROJECT_API_KEY>", "<PROJECT_API_KEY>"

client = mqtt.Client()
client.username_pw_set(user, password)
client.connect(host, 1883, 60)

for i in range(100):
    v = str(int(np.random.random() *100))
    t = str(time.strftime("%Y-%m-%dT%H:%M:%S"))
    payload = [{"id":"random_data_01","value":[v],
"time":t}]
    client.publish(topic, "%s" % ( json.dumps(payload) ))
    time.sleep(1)
```

範例程式重點

- rawdata 的主題包含設備編號
  - topic = "/v1/device/<DEVICE_NUMBER>/rawdata"
- 使用者帳號 / 密碼使用”專案金鑰”或是”帳號金鑰”
  - user, password = "<API_KEY>", "<API_KEY>"
- 封包內要有 id 欄位表示感測器 id, 要有 value 欄位表示感測器值 (list),time 欄位非必要
  - payload = [{"id":"<SID>","value":[v], "time":t}]

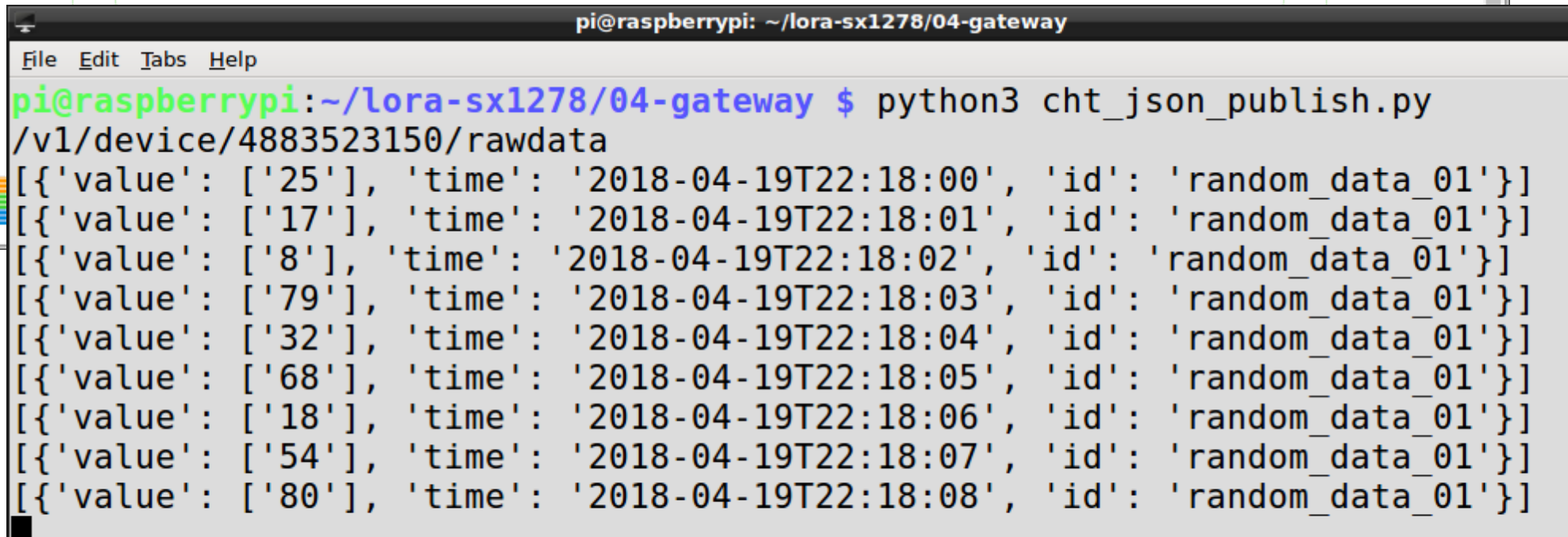
把紅色部份換成專案裡的實際內容

https://iot.cht.com.tw/iot/developer/quick

DEMO
cht_json_publish.py

```
$ cd ~/lora-sx1276/04-gateway
$ python3 cht_json_publish.py
```

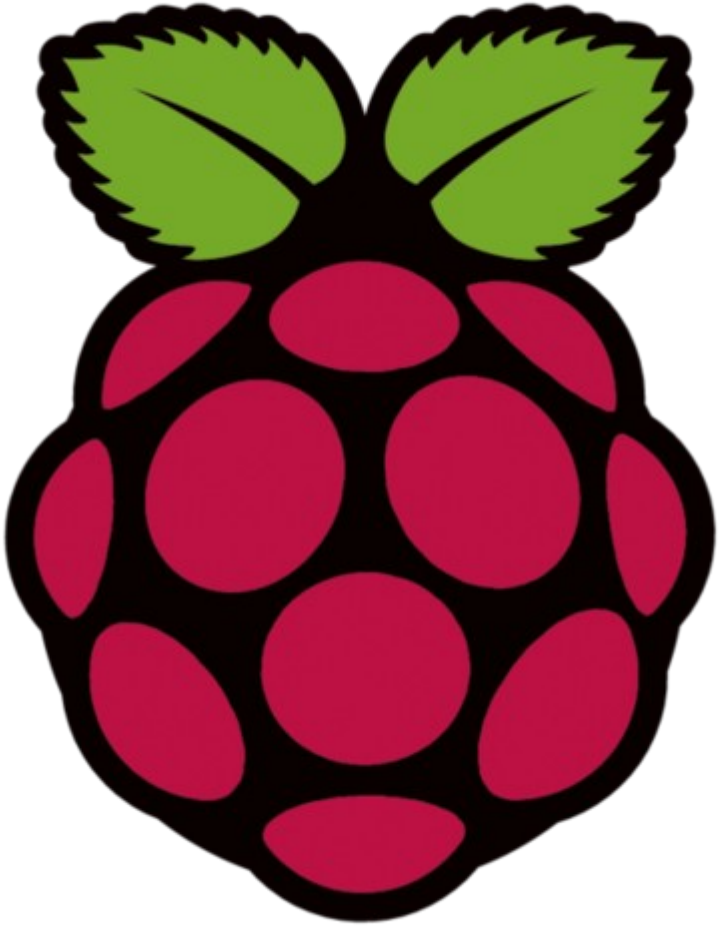

中華電信IoT智慧聯網
iot.cht.com.tw/iot/project/12983/device/4883523150
IOT 智慧聯網大平台
SMART PLATFORM
開發者中心
專案管理
應用服務
聯絡我們
繁體中文
『pi3』設備資訊 (編號:4883523150)
感測器
設備內容
事件驅動
憑證申請
存取統計
共有 1 個感測器
增加感測器
80
random_data


pi@raspberrypi: ~/lora-sx1278/04-gateway
File Edit Tabs Help
pi@raspberrypi:~/lora-sx1278/04-gateway $ python3 cht_json_publish.py
/v1/device/4883523150/rawdata
[{'value': ['25'], 'time': '2018-04-19T22:18:00', 'id': 'random_data_01'}]
[{'value': ['17'], 'time': '2018-04-19T22:18:01', 'id': 'random_data_01'}]
[{'value': ['8'], 'time': '2018-04-19T22:18:02', 'id': 'random_data_01'}]
[{'value': ['79'], 'time': '2018-04-19T22:18:03', 'id': 'random_data_01'}]
[{'value': ['32'], 'time': '2018-04-19T22:18:04', 'id': 'random_data_01'}]
[{'value': ['68'], 'time': '2018-04-19T22:18:05', 'id': 'random_data_01'}]
[{'value': ['18'], 'time': '2018-04-19T22:18:06', 'id': 'random_data_01'}]
[{'value': ['54'], 'time': '2018-04-19T22:18:07', 'id': 'random_data_01'}]
[{'value': ['80'], 'time': '2018-04-19T22:18:08', 'id': 'random_data_01'}]

開發者中心 > 快速開始


智慧聯網平台
SMART PLATFORM
開發者中心
專案管理
應用服務
帳號資訊
登出
Select Language
Powered by Google Translate
首頁 / 開發者中心 / 快速開始
快速開始
平臺介紹
快速開始
API 文件
應用說明
行動裝置
簡介
快速開始將提供簡易的開發指南，使新手開發者能夠迅速上手，只要遵循下圖之開發流程，您即可一步步了解本平臺之架構，並成為中華電信 IoT 平臺開發者的一員，將智慧設備終端聯網，完成一個簡易的 IoT 應用程式。
Step.1
註冊用戶
Step.2
新增專案
Step.3
新增設備
Step.4
新增感測器
Step.5
上傳數據
Step.6
取回數據
步驟一、註冊會員
想要成為中華電信 IoT 平臺會員，您必須透過一個 CHT 會員中心帳號，才可以於自己的帳號之下，新增專案、裝置、感測

Raspberry Pi Rocks the World



Thanks

補充

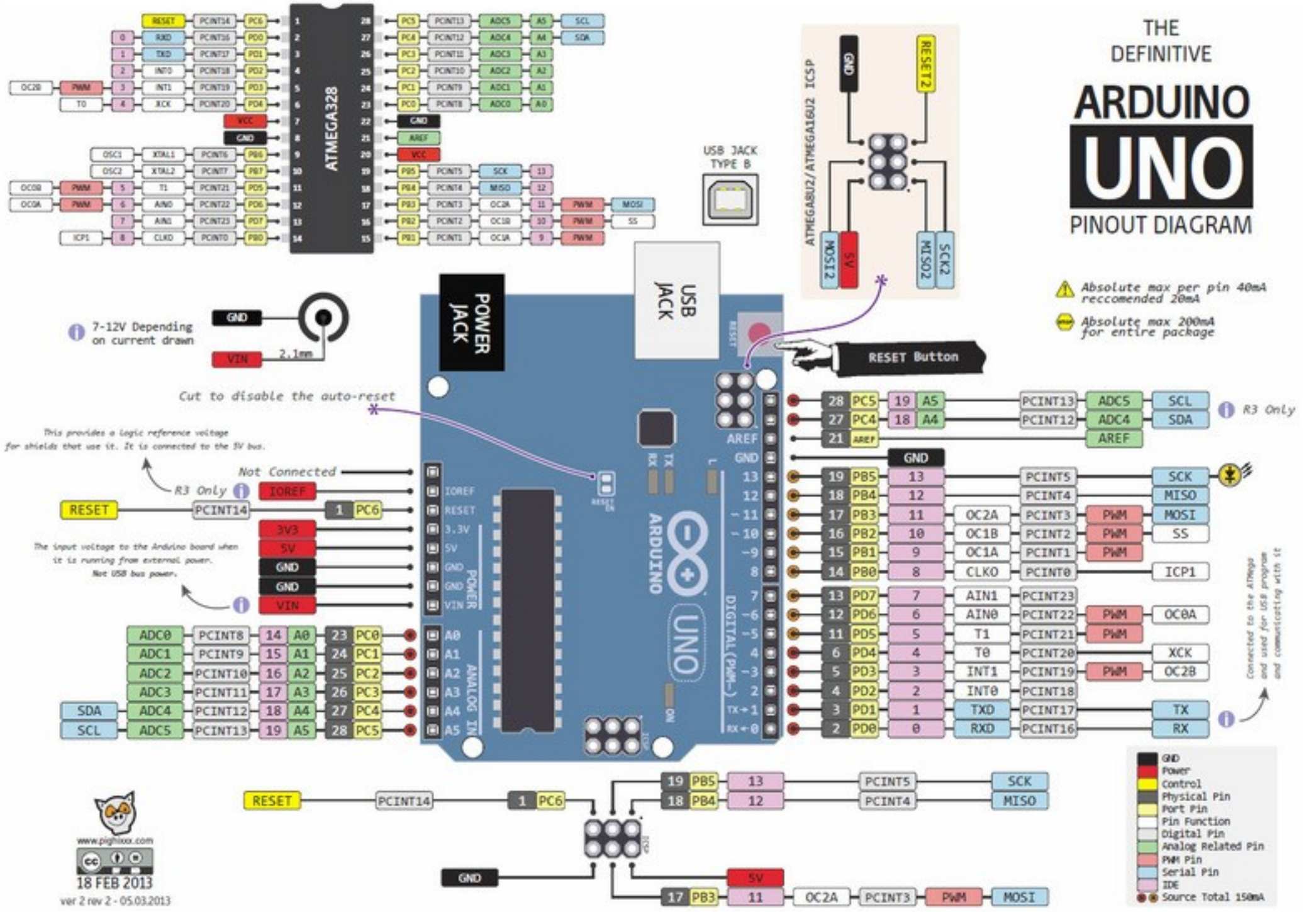
計算 Power Consumption

- 長達 10 年的低功耗（使用鈕扣電池）怎麼計算？

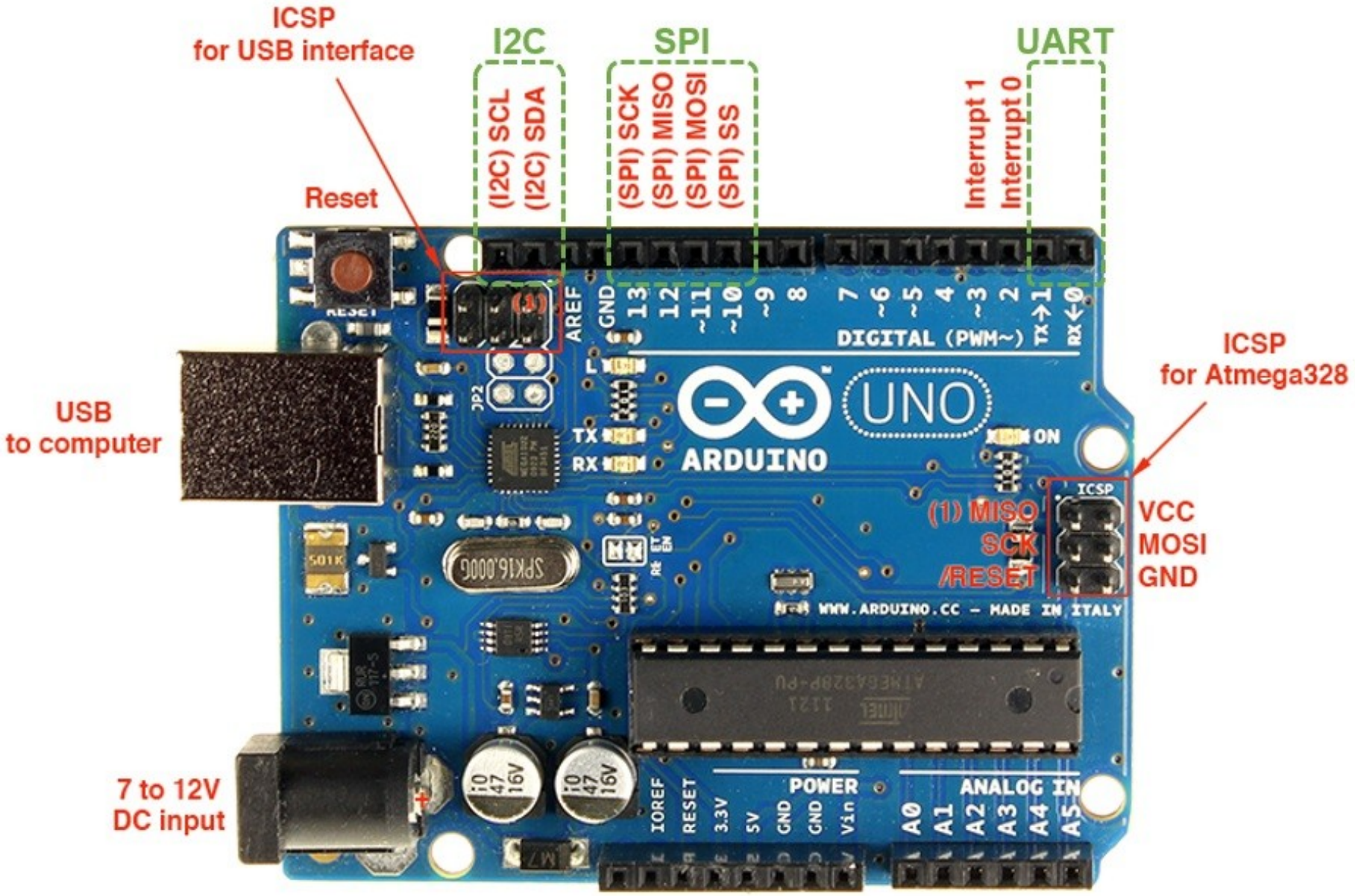
- 條件：
  - 在 SF=6,BW=500KHz, 每次 RX 完整時間為 10ms
  - 鈕扣電池 2032 大約為 210mAh

- 計算：
  - RX 電流為 10mA, 因此 210mAh/10mA=21 小時 =75600 秒
  - RX 時間為 10ms, 因此 75600 秒可以做 7560000 次
  - 每分鐘一次 RX,7560000/60 分 /24 小時 /365 天 =14 年
  - 實際電容量為額定的 70%,14 年 x70%=9.8 年

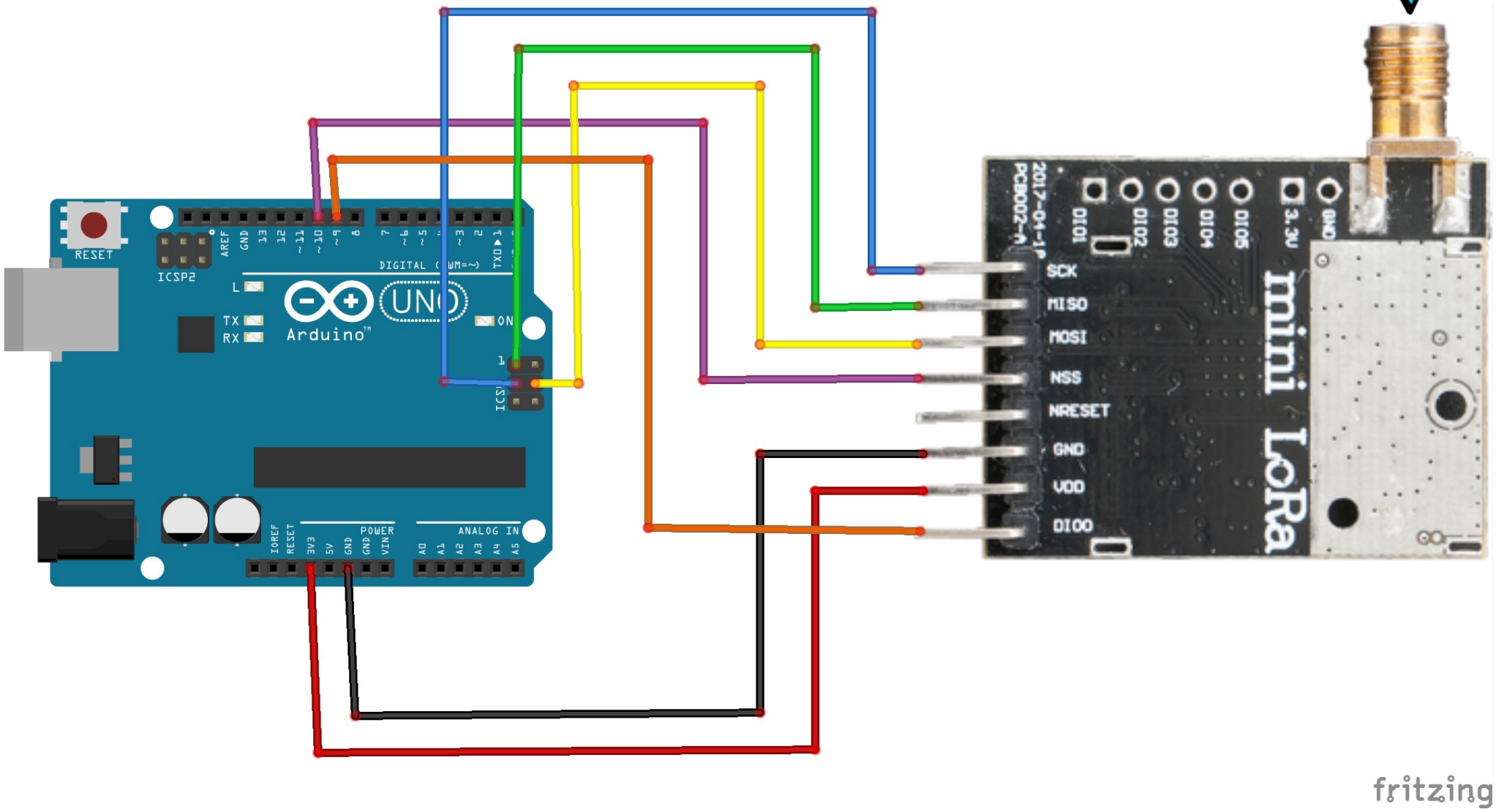
使用 Arduino


THE DEFINITIVE
ARDUINO UNO
PINOUT DIAGRAM
ATMEGA328
ATMEGA8U2/ATMEGA16U2 ICSP
USB JACK TYPE B
POWER JACK
USB JACK
RESET Button
Absolute max per pin 40mA reccomended 20mA
Absolute max 200mA for entire package
7-12V Depending on current drawn
2.1mm
Cut to disable the auto-reset
This provides a logic reference voltage for shields that use it. It is connected to the 5V bus.
Not Connected
R3 Only
The input voltage to the Arduino board when it is running from external power. Not USB bus power.
Connected to the ATMegs and used for USB program and communicating with it
ARDUINO
UNO
DIGITAL (PWM~)
ANALOG IN
POWER
GND
Power
Control
Physical Pin
Port Pin
Pin Function
Digital Pin
Analog Related Pin
PWM Pin
Serial Pin
IDE
Source Total 150mA
www.pighixxx.com
18 FEB 2013
ver 2 rev 2 - 05.03.2013

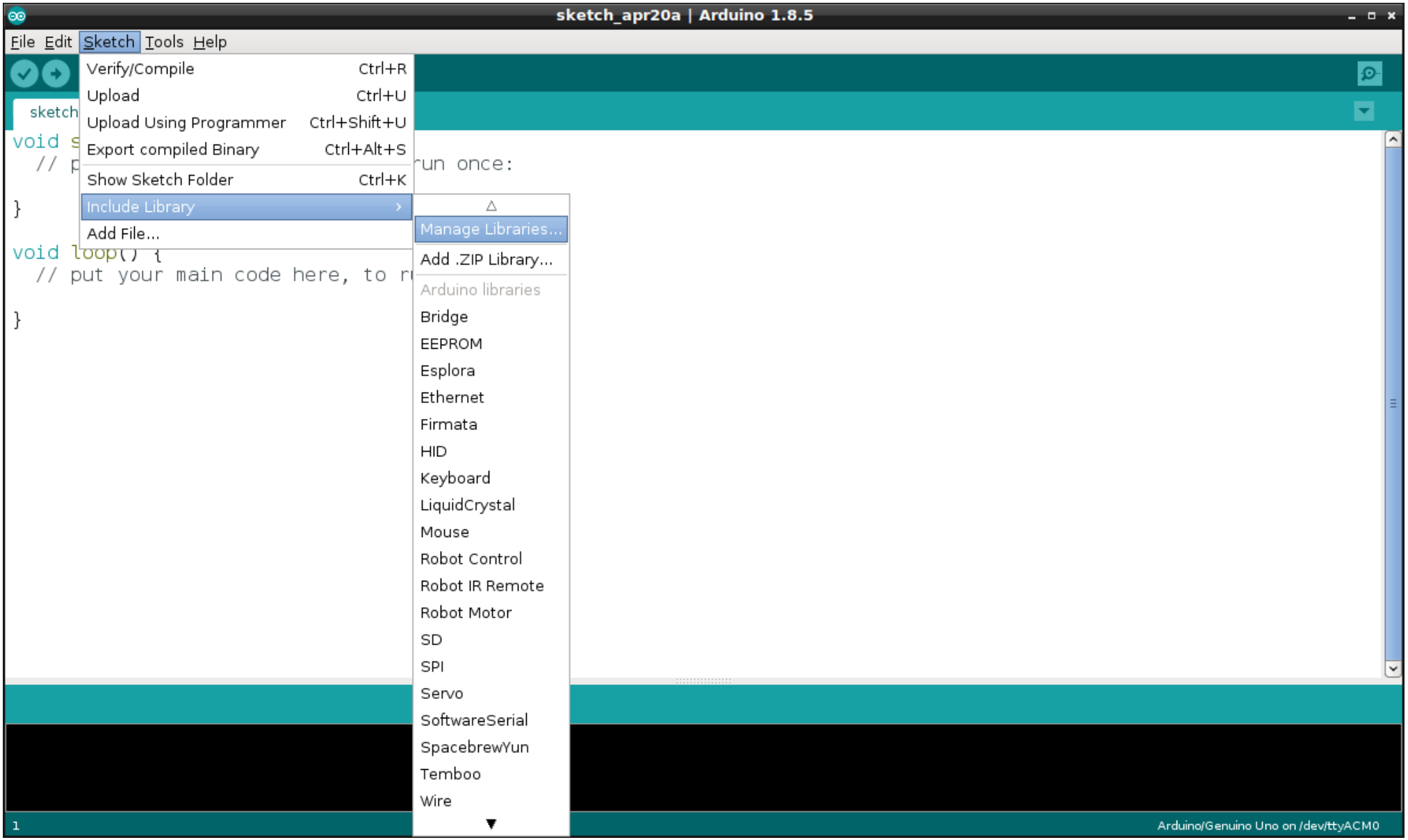
一樣使用 SPI 界面


ICSP
for USB interface
I2C
SPI
UART
(I2C) SCL
(I2C) SDA
(SPI) SCK
(SPI) MISO
(SPI) MOSI
(SPI) SS
Interrupt 1
Interrupt 0
Reset
USB
to computer
ICSP
for Atmega328
VCC
MOSI
GND
(1) MISO
SCK
/RESET
7 to 12V
DC input
DIGITAL (PWM~)
POWER
ANALOG IN
UNO
ARDUINO
WWW.ARDUINO.CC – MADE IN ITALY

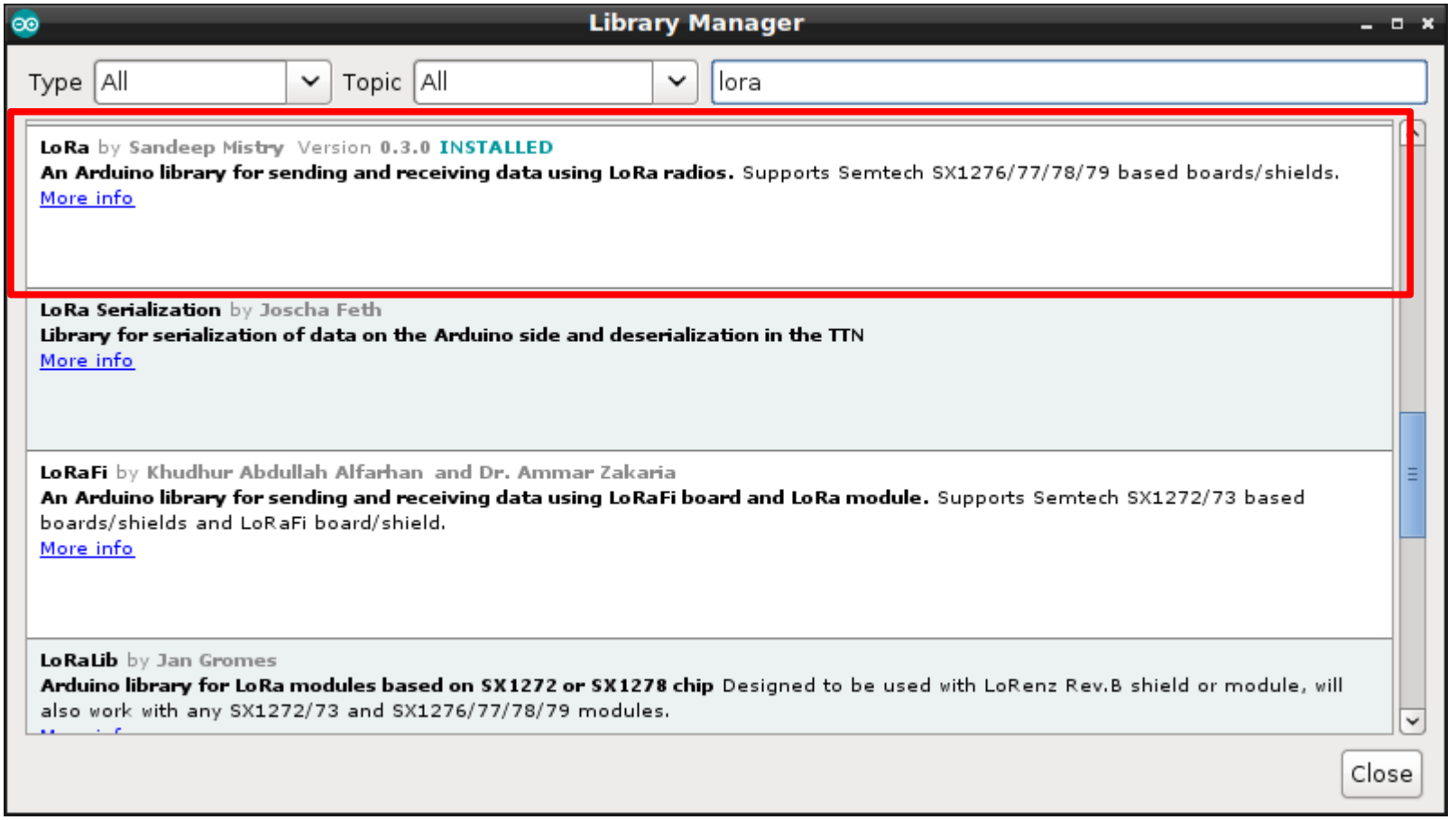
接線圖


裝天線
RESET
ICSP2
Arduino
UNO
POWER
ANALOG IN
SCK
MISO
MOSI
NSS
NRESET
GND
VDD
DIO0
mini LoRa
fritzing

安裝 LoRa 函式庫


sketch_apr20a | Arduino 1.8.5
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries...
Add .ZIP Library...
Arduino libraries
Bridge
EEPROM
Esplora
Ethernet
Firmata
HID
Keyboard
LiquidCrystal
Mouse
Robot Control
Robot IR Remote
Robot Motor
SD
SPI
Servo
SoftwareSerial
SpacebrewYun
Temboo
Wire
sketch
void loop() {
// put your main code here, to r
}
1
Arduino/Genuino Uno on /dev/ttyACM0

搜尋 LoRa 並選擇 Sandeep Mistry



LoRaSender

```
int counter = 0;

void setup() {
  Serial.begin(9600);
  while (!Serial);
  if (!LoRa.begin(868E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.setSpreadingFactor(12);
  LoRa.setSignalBandwidth(125E3);
}

void loop() {
  // send packet
  LoRa.beginPacket();
  LoRa.print(counter);
  LoRa.endPacket();
  counter++;
  delay(5000);
}
```

Frequency

Spreading Factor

Bandwidth

範例請參考：

https://robotzero.one/arduino-ai-thinker-ra-02/

如何優化？

LoRa Modulation Basics

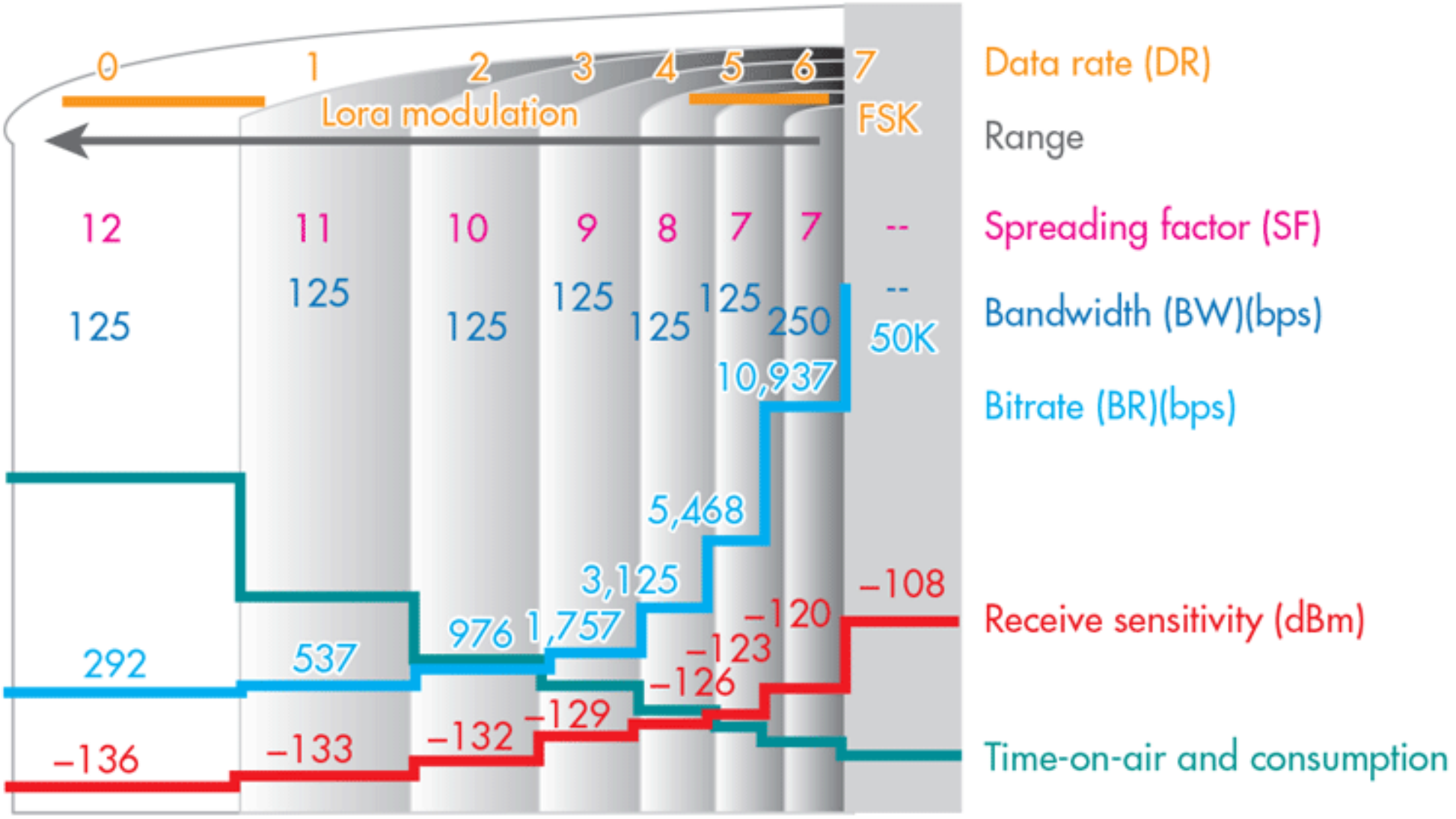
BW SF Data Rate Sensitivity

Bandwidth (kHz)	Spreading Factor	Coding rate	Nominal Rb (bps)	Sensitivity indication (dBm)
10.4	6	4/5	782	-131
	12	4/5	24	-147
20.8	6	4/5	1562	-128
	12	4/5	49	-144
62.5	6	4/5	4688	-121
	12	4/5	146	-139
125	6	4/5	9380	-118
	12	4/5	293	-136

固定 BW 而改變 SF 會讓 Data Rate 和 Sensitivity 改變

https://electronics.stackexchange.com/questions/309424

Bandwidth, Spreading Factor and Data Rate

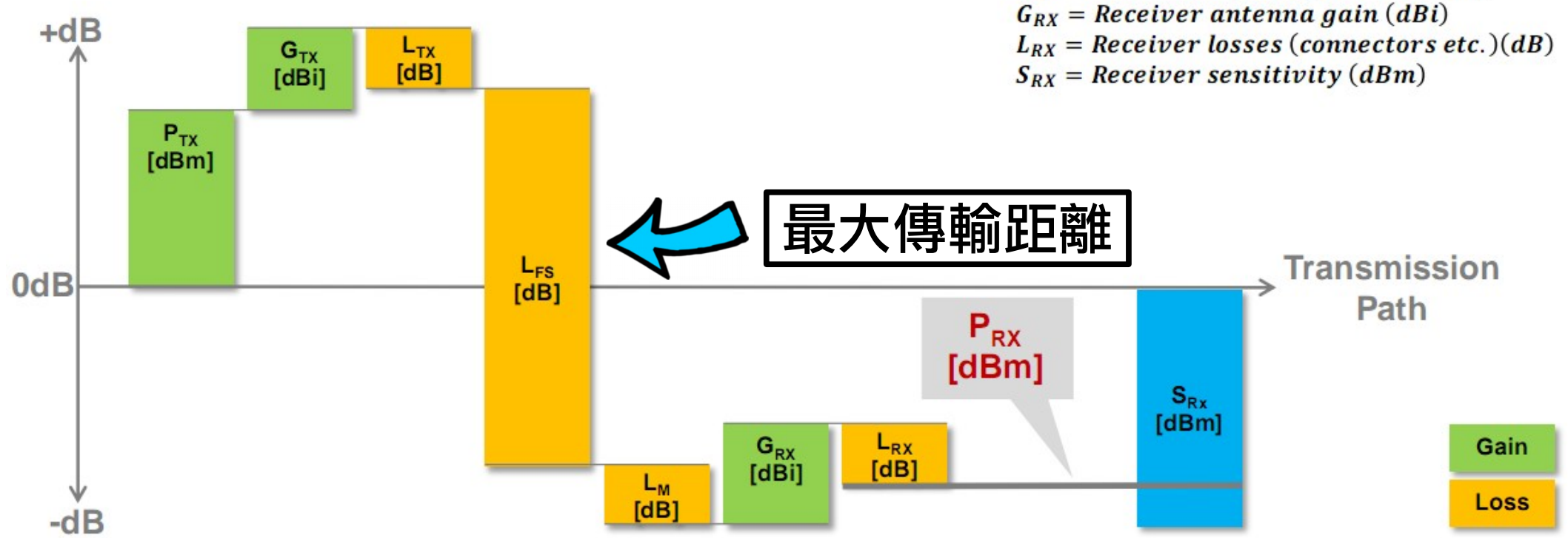


http://www.electronicdesign.com/embedded-revolution/lora-rolls-philly

Link Budget 計算

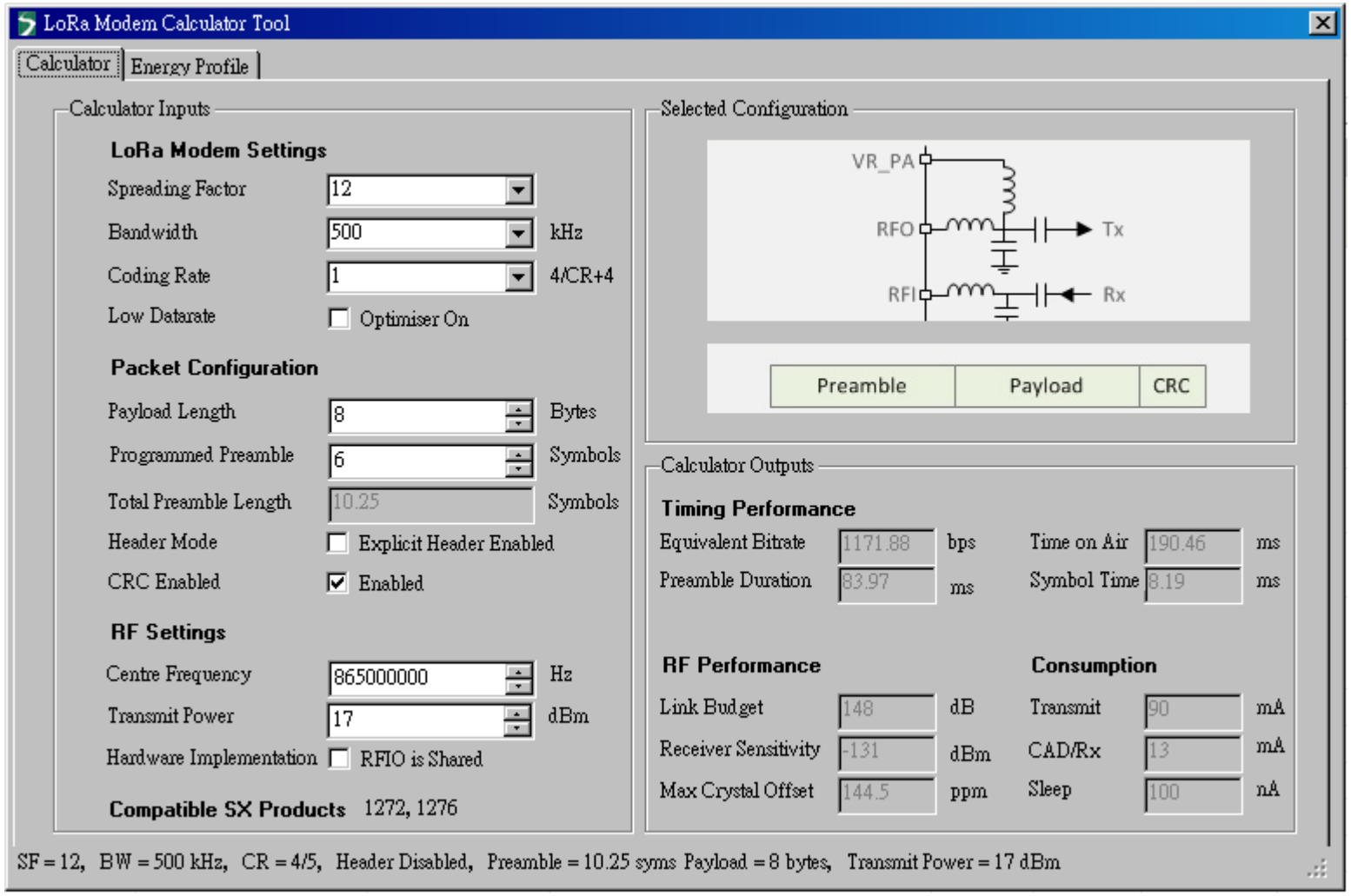
$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{FS} - L_M + G_{RX} - L_{RX}$$

P_{RX} = Received power (dBm)
P_{TX} = Sender output power (dBm)
G_{TX} = Sender antenna gain (dBi)
L_{TX} = Sender losses (connectors etc.)(dB)
L_{FS} = Free space loss (dB)
L_M = Misc. losses (multipath etc.)(dB)
G_{RX} = Receiver antenna gain (dBi)
L_{RX} = Receiver losses (connectors etc.)(dB)
S_{RX} = Receiver sensitivity (dBm)



https://www.slideshare.net/PeterREgli/lpwan

如何估計 Link Budget ？



http://www.semtech.com/wireless-rf/rf-transceivers/sx1272/

頻譜分析儀 (Spectrum Analyzer)

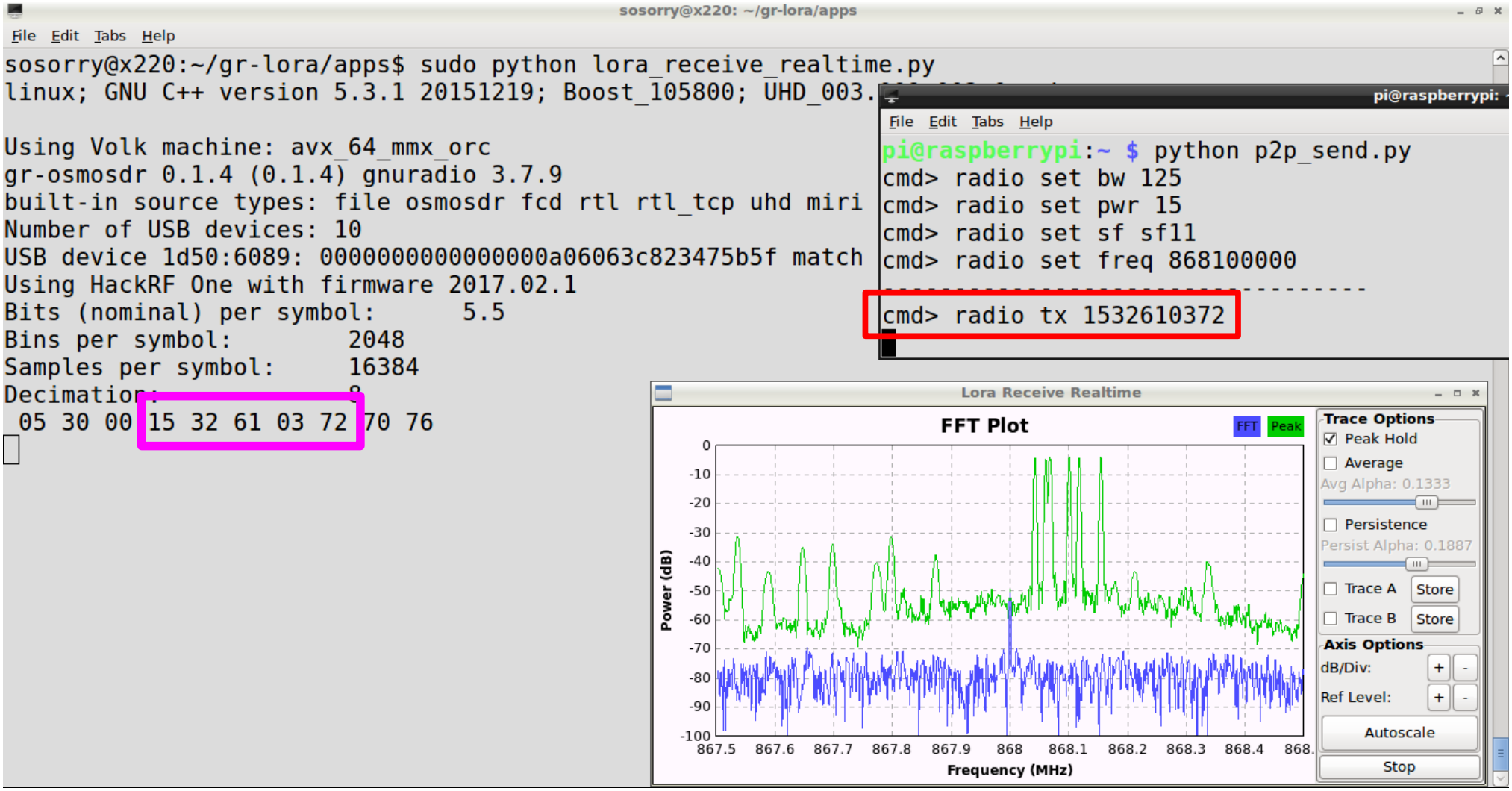
- 量測隨頻率變化的輸入信號強度
- 硬體
  - SDR(Software Defined Radio)
  - HackRF

- 軟體
  - GNU Radio
  - FreqShow(Python)

GNU Radio
THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

http://jensd.be/755/network/lorawan-simply-explained

你的 LoRa 是明文傳送嗎？

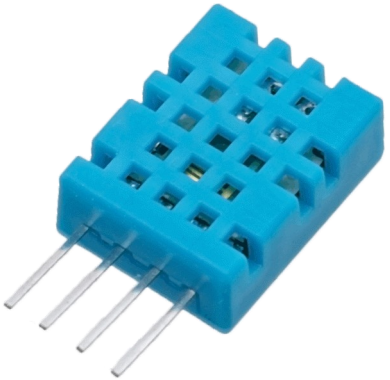
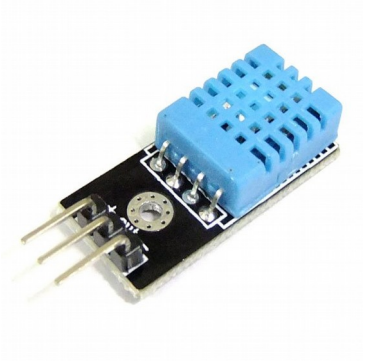


https://github.com/rpp0/gr-lora

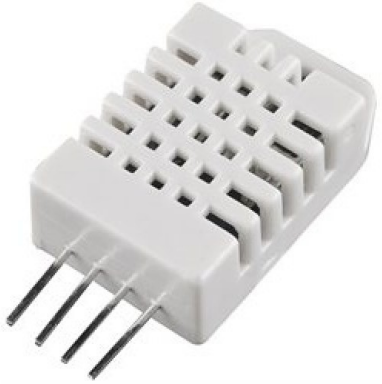
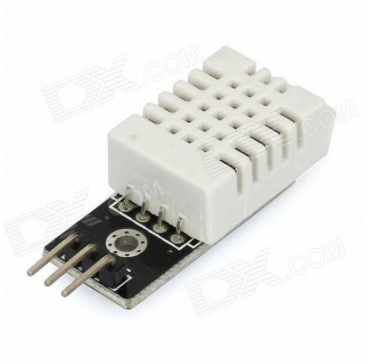
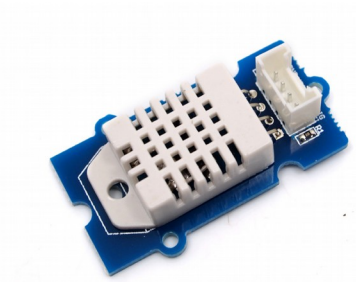
搭配感測器

DHTxx 溫濕度感測器系列

- DHT11

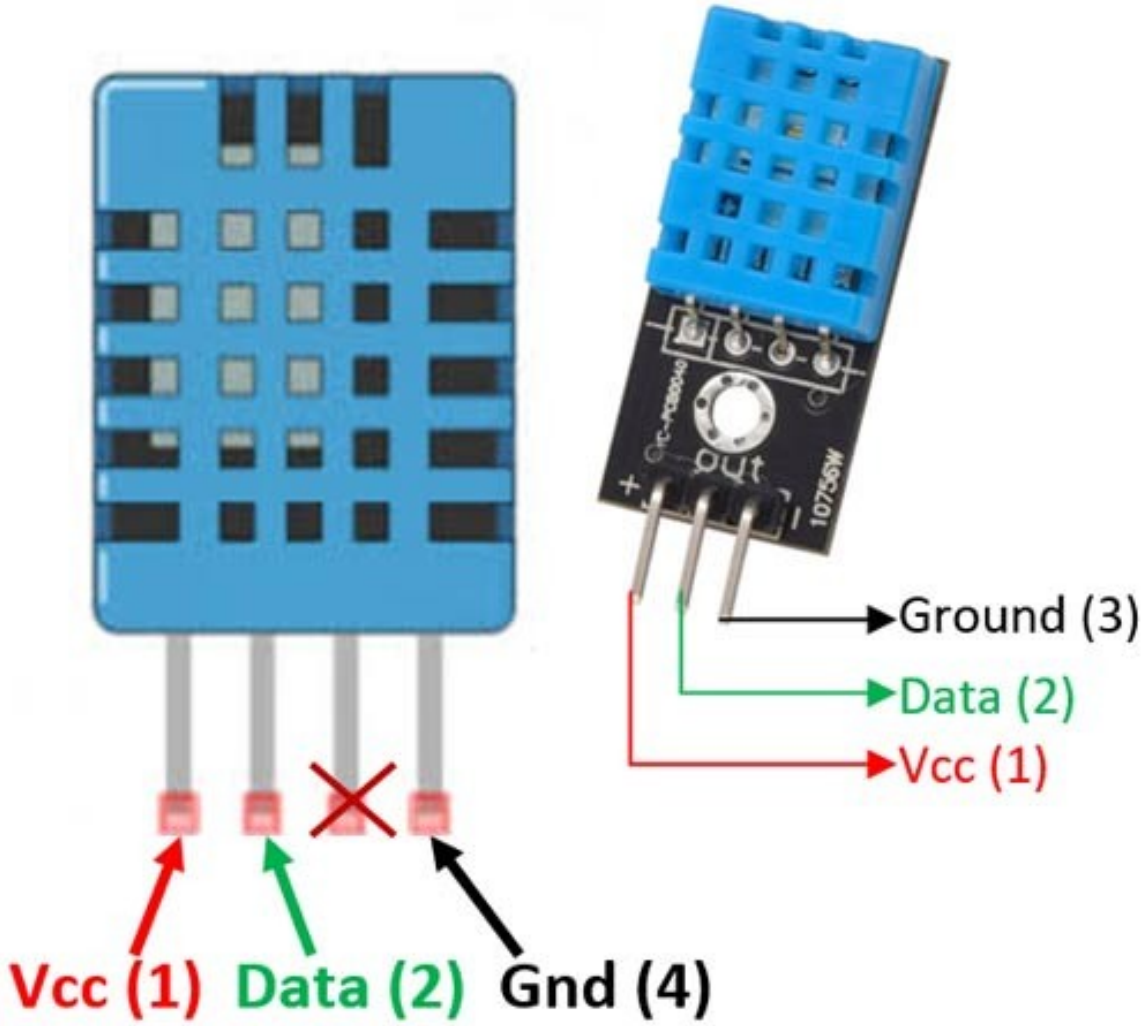



- DHT22


https://www.google.com.tw/search?q=dht11&tbm=isch
https://www.google.com.tw/search?q=dht22&tbm=isch

外觀



https://components101.com/dht11-temperature-sensor

DHT11 規格

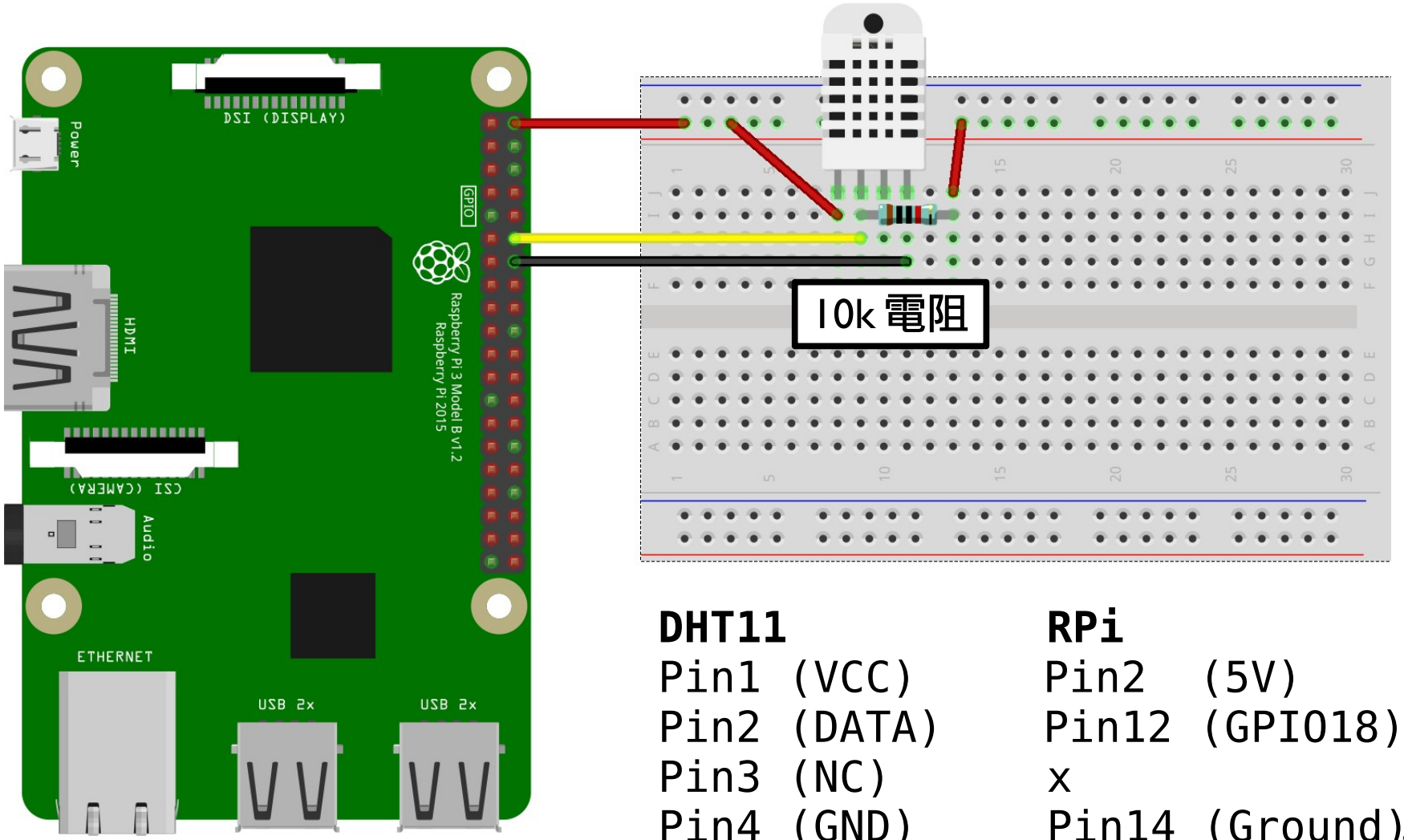
- 3 - 5.5V 工作電壓
- 2.5mA 最高工作電流
- 20 - 90% 相對濕度 (RH) 量測範圍 , ± 5%RH 誤差
- -0 - 50 C 溫度量測範圍 , ± 2 C 誤差
- 取樣頻率為 1 Hz

http://www.adafruit.com/datasheets/DHT11-chinese.pdf

可應用範圍

- 冷暖空調
- 汽車
- 氣象站
- 除濕機
- 測試及檢測設備
- 自動控制
- 醫療

線路圖



DHT11	RPi
Pin1 (VCC)	Pin2 (5V)
Pin2 (DATA)	Pin12 (GPIO18)
Pin3 (NC)	x
Pin4 (GND)	Pin14 (Ground)

1-Wire Protocol

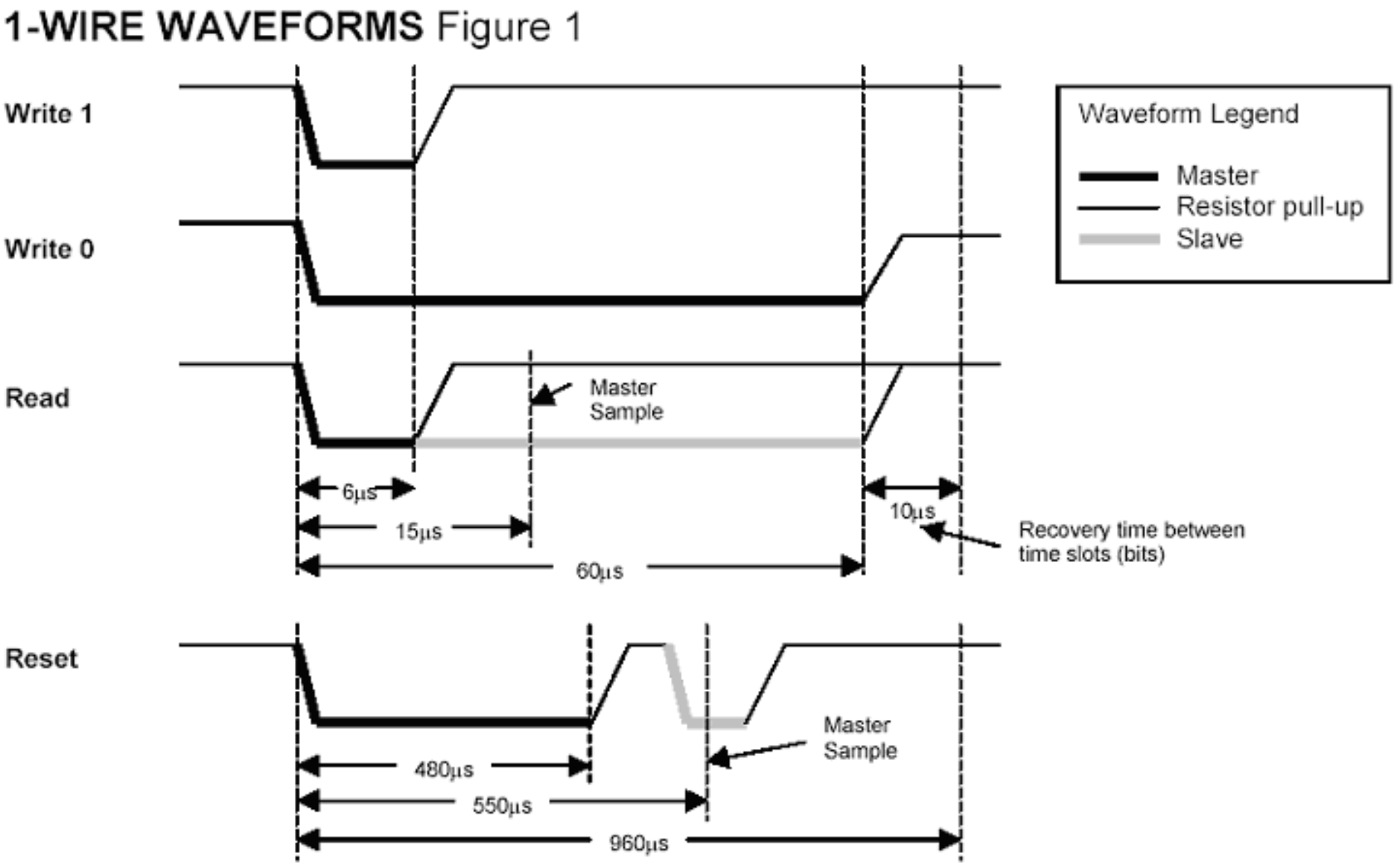
1-Wire Operation Table (範例)

1-WIRE OPERATIONS Table 1

Operation	Description	Implementation
Write 1 bit	Send a '1' bit to the 1-Wire slaves (Write 1 time slot)	Drive bus low, delay 6µs Release bus, delay 64µs
Write 0 bit	send a '0' bit to the 1-Wire slaves (Write 0 time slot)	Drive bus low, delay 60µs Release bus, delay 10µs
Read bit	Read a bit from the 1-Wire slaves (Read time slot)	Drive bus low, delay 6µs Release bus, delay 9µs Sample bus to read bit from slave Delay 55µs
Reset	Reset the 1-Wire bus slave devices and ready them for a command	Drive bus low, delay 480µs Release bus, delay 70µs Sample bus, 0 = device(s) present, 1 = no device present Delay 410µs

http://coecsl.ece.illinois.edu/ge423/sensorprojects/1-wire_full.doc

1-Wire Waveform



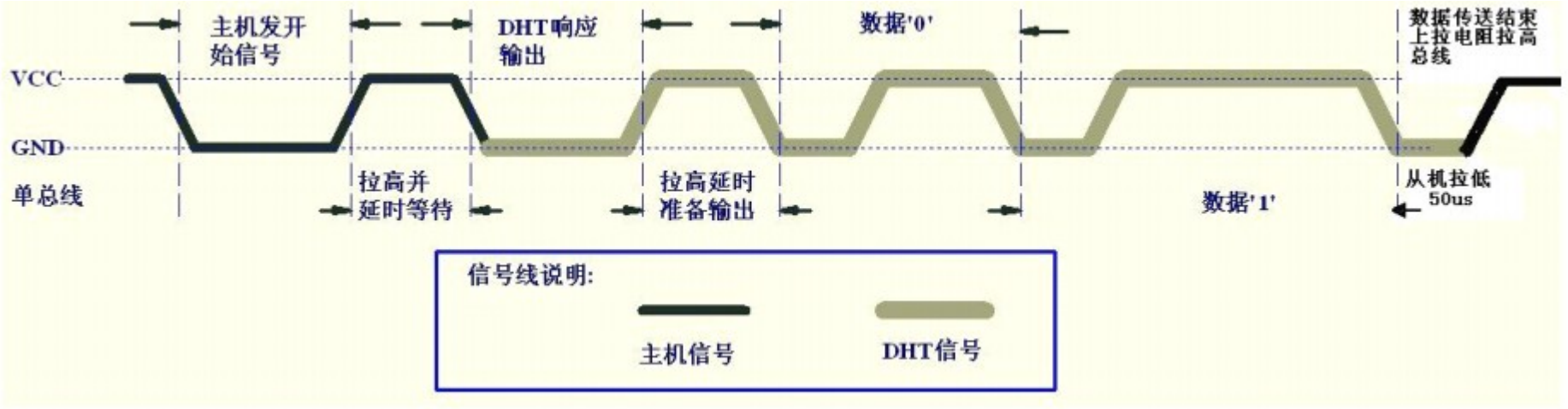
http://coecsl.ece.illinois.edu/ge423/sensorprojects/1-wire_full.doc

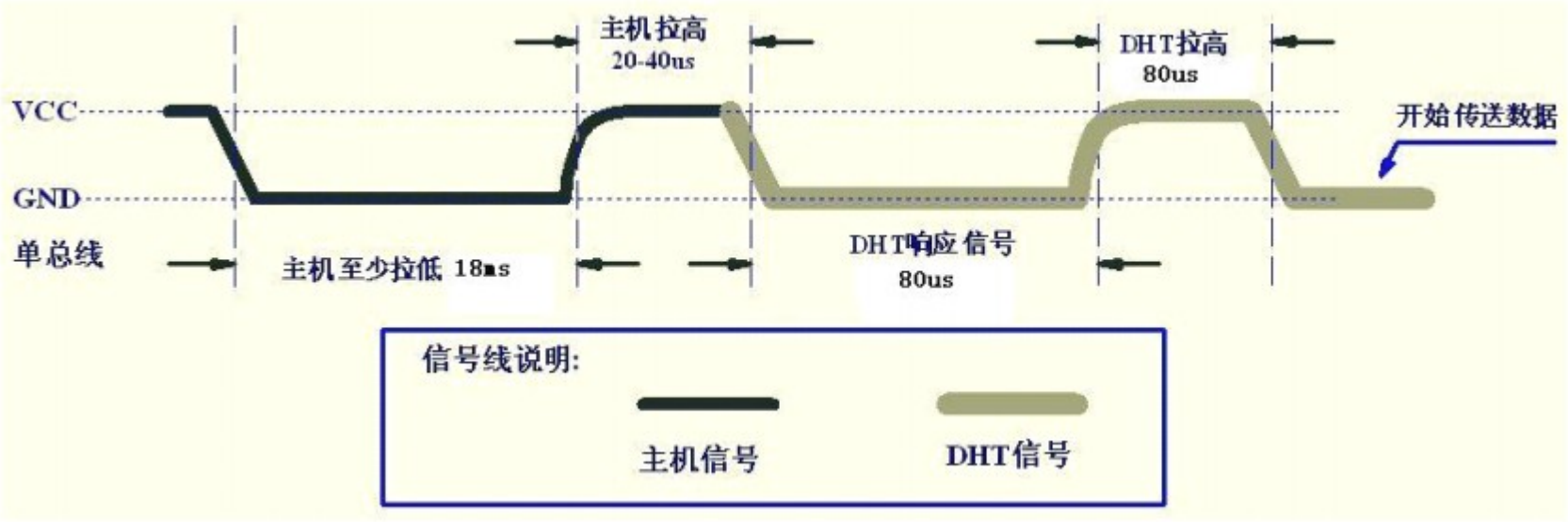
DHTxx Communication Protocol

- 1-wire protocol
- 每次通訊時間：4ms
- 數據格式：40bit, MSB
  - 8bit 濕度整數 +8bit 濕度小數
  - 8bit 溫度整數 +8bit 溫度小數
  - 8bit CRC

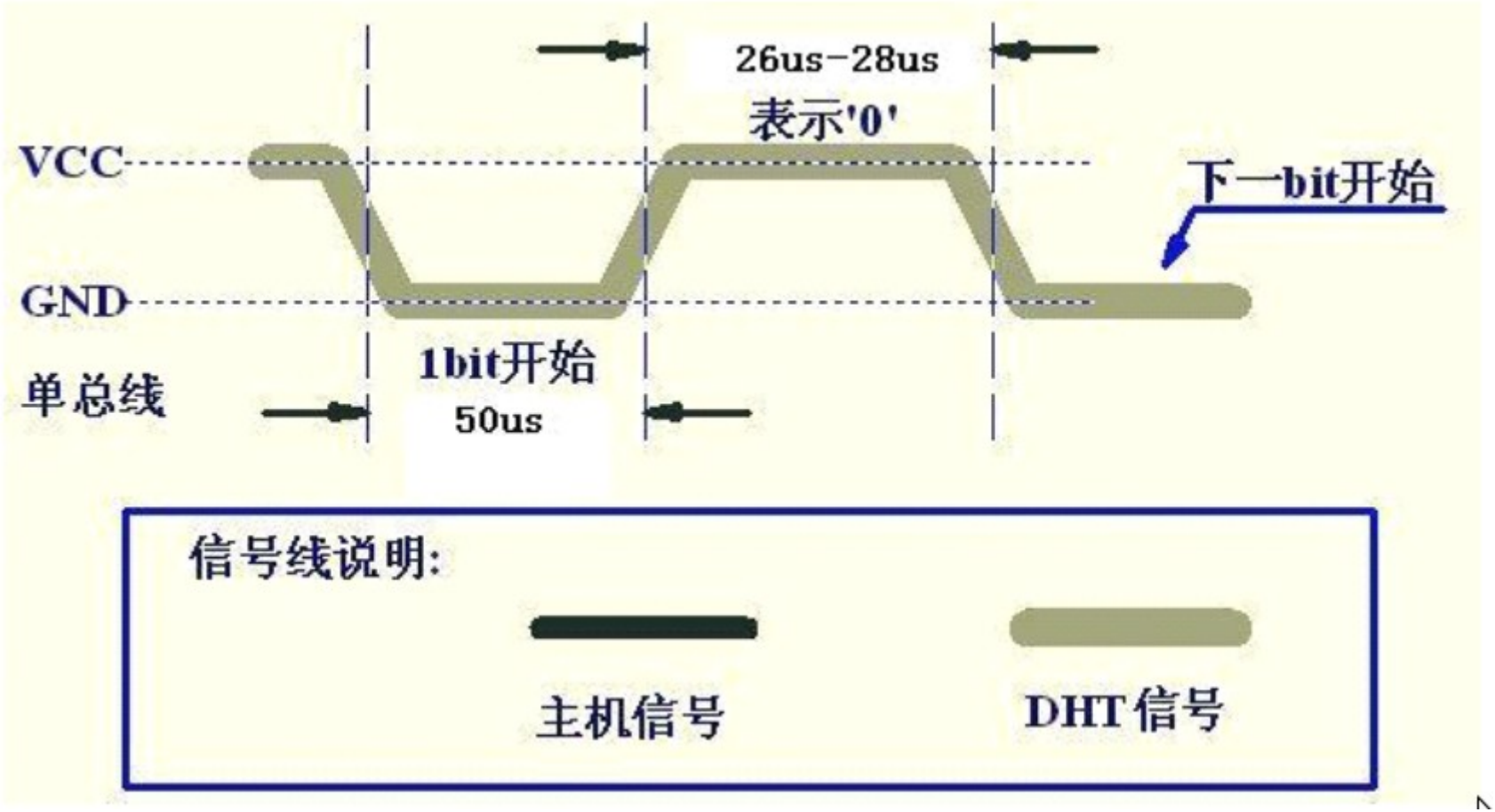
http://www.adafruit.com/datasheets/DHT11-chinese.pdf

DHTxx Communication Protocol

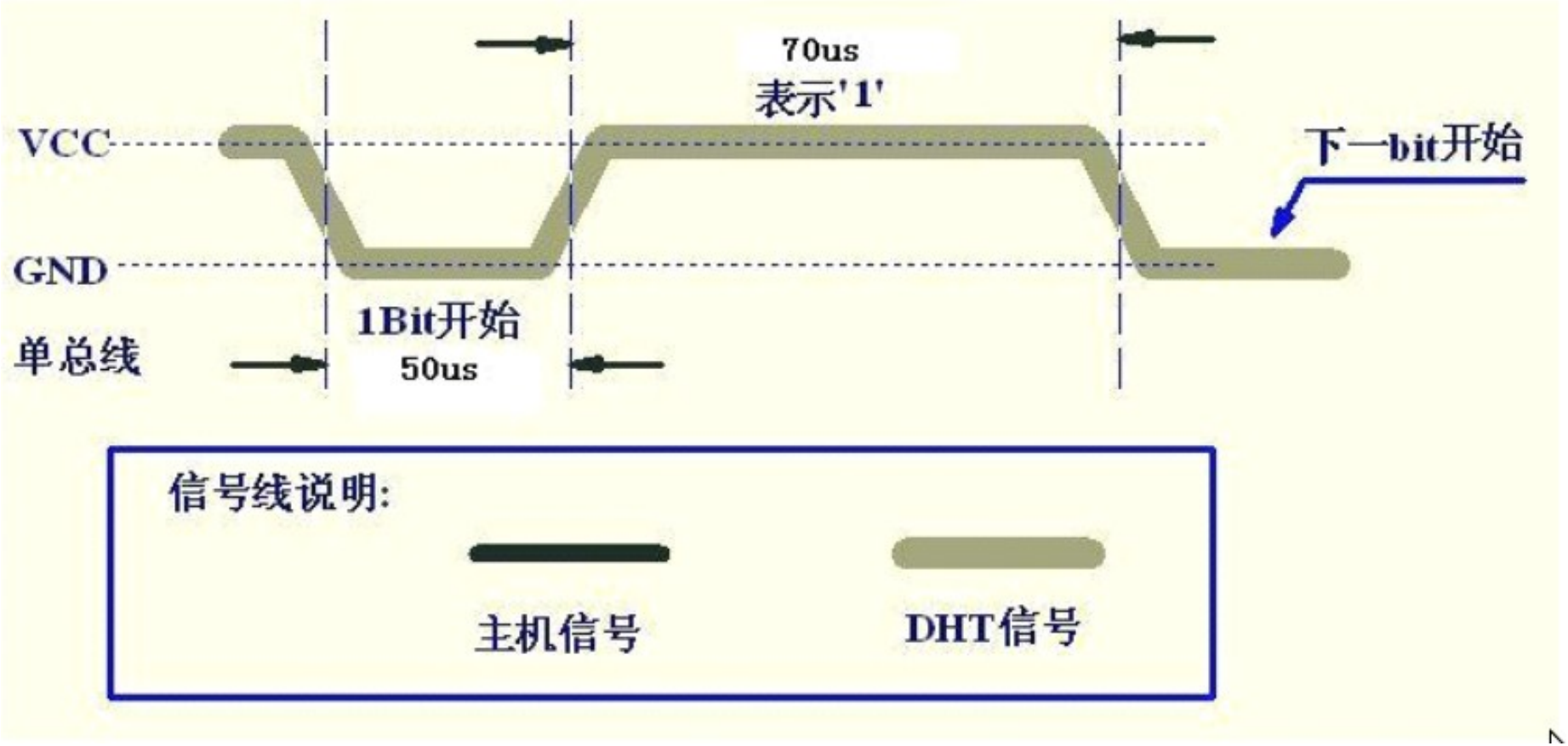




Send 0


26us-28us
表示'0'
VCC
下一bit开始
GND
1bit开始
50us
单总线
信号线说明:
主机信号
DHT信号

Send 1



下載安裝 Adafruit Python DHT 套件

- $ cd ~
- $ git clone https://github.com/adafruit/Adafruit_Python_DHT.git
- $ cd Adafruit_Python_DHT
- $ sudo python setup.py install
- $ cd examples
- $ sudo python AdafruitDHT.py 11 18

```
Temp=28.9*  Humidity=67.9%
```

AdafruitDHT.py

```
import Adafruit_DHT

sensor_args = { '11': Adafruit_DHT.DHT11,
                '22': Adafruit_DHT.DHT22,
                '2302': Adafruit_DHT.AM2302 }
sensor = sensor_args[sys.argv[1]]
pin = sys.argv[2]

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

if humidity is not None and temperature is not None:
    print('Temp={0:0.1f}*  Humidity={1:0.1f}
%'.format(temperature, humidity))
else:
    print('Failed to get reading. Try again!')
    sys.exit(1)
```