

# LoPy<sup>4</sup>

**LoPy4 With SigFox**  
**Real life IoT applications**

**06026144**  
**INTERNET OF THINGS**

Second semester, 2019.  
Faculty of Information Technology  
King Mongkut's Institute of Technology Ladkrabang

**Panwit Tuwanut**

60070141 Tanakit Intaniyom  
60070156 Warinthorn Thananporn

Picture by **Tanakit Intaniyom**

ในปัจจุบันอุปกรณ์ IoT ได้เป็นอุปกรณ์ส่วนหนึ่งในชีวิตประจำวันไปแล้ว ตั้งแต่รถยนต์ แอร์ เซนเซอร์ต่างๆ เช่น อุณหภูมิ ความชื้น และอีกมากมาย อุปกรณ์เหล่านี้ก็จะเชื่อมต่อผ่านเครือข่ายที่ให้บริการ ก็เหมือนกับเทคโนโลยี 2G 3G 4G 5G ที่โทรศัพท์มือถือสามารถเชื่อมต่ออินเทอร์เน็ตได้ โดยไม่ต้องใช้สายอีกต่อไป อุปกรณ์ IoT ก็เช่นกัน อุปกรณ์เหล่านี้ก็จะเชื่อมต่อกับเครือข่ายที่ให้บริการ

ผู้ให้บริการ IoT รายใหญ่ที่สำคัญก็จะมี LoRaWAN NP-IoT และ Sigfox โดยทางที่ทีมงานจะเลือก Sigfox มาเป็นเครือข่ายครับ เนื่องจากใช้คลื่นสัญญาณที่ไม่ต้องใช้ใบอนุญาต (Unlicensed) และใช้แบนวิธ (Bandwidth) ที่แคบมาก ทำให้คงทนต่อการรบกวน และยังประหยัดแบตเตอรี่มากอีกด้วย ในตัว Sigfox เองก็มีระบบจัดการอุปกรณ์ที่ใช้งานง่าย และไม่ซับซ้อน



หน้าตา Logo ของ Sigfox

<https://www.sigfox.com/>

Sigfox นั้นเปิดให้บริการในหลายประเทศทั่วโลกแล้ว 70 กว่าประเทศ ในปี 2020 รวมทั้งประเทศไทยด้วย โดยในประเทศไทยนั้นผู้ให้บริการ Sigfox คือ ThingsOnNet ซึ่งเป็นผู้ให้บริการรายเดียว





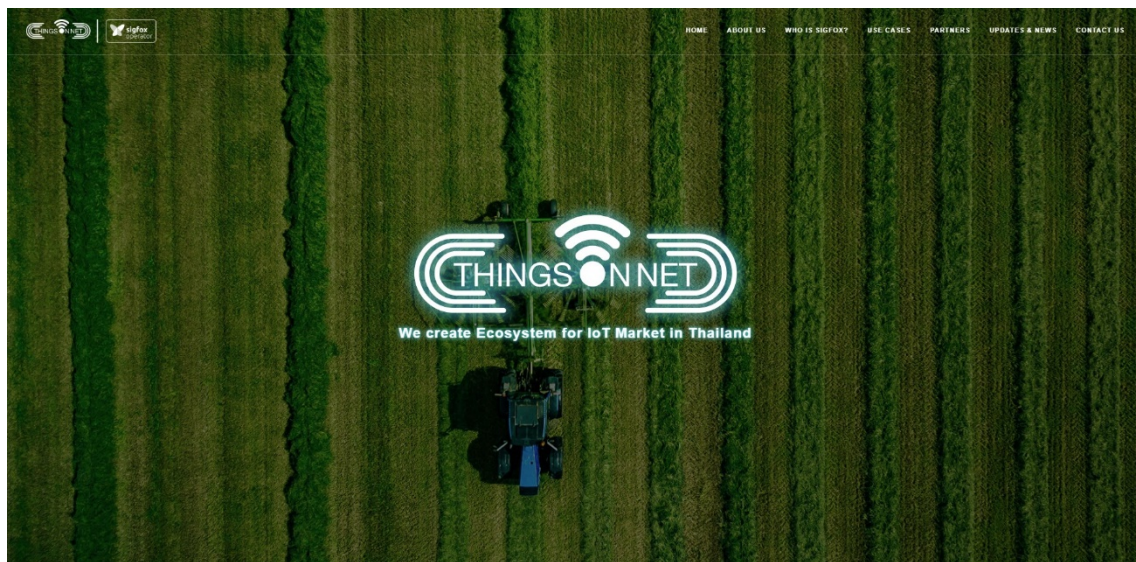
We've got you covered!

Sigfox is already available in over 70 countries and regions and aims to cover 100% of the globe in the next few years...



70 ประเทศทั่วโลกในปี 2020

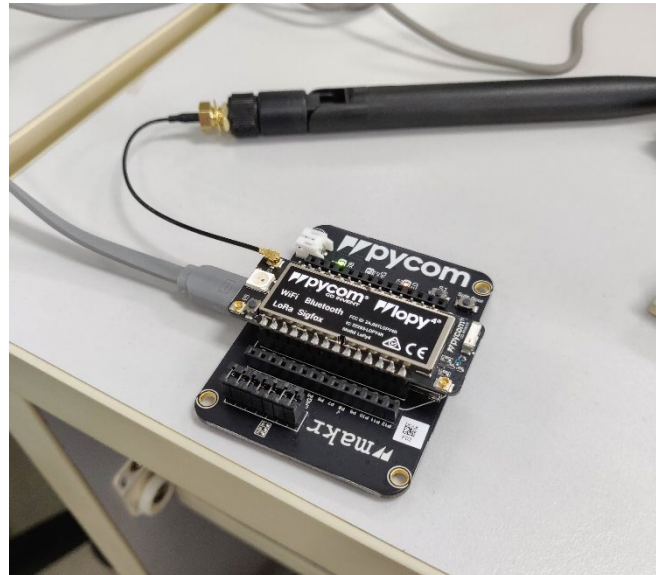
<https://www.sigfox.com/en/coverage>



ThingsOnNet ผู้ให้บริการ Sigfox ในประเทศไทย

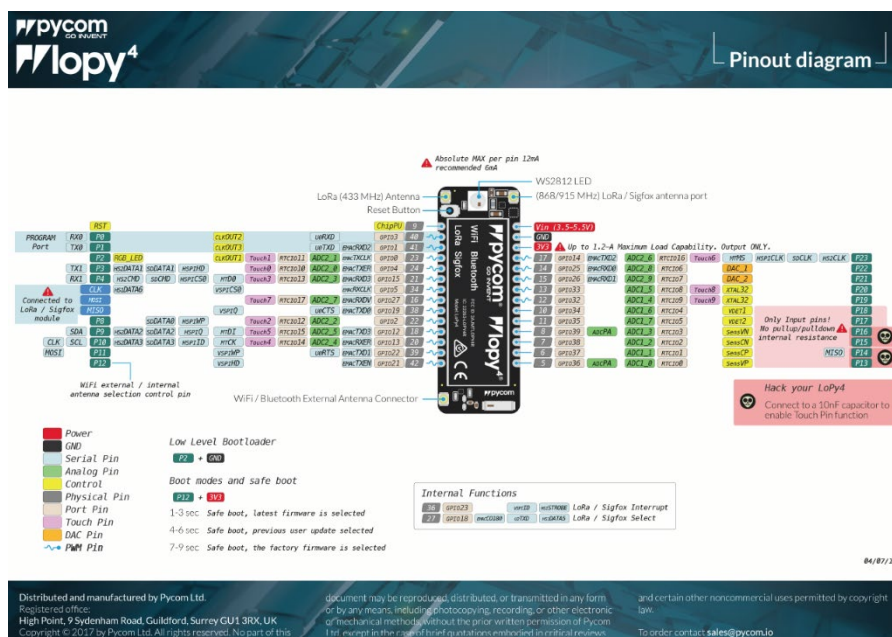
<https://www.thingsonnet.net/>

อุปกรณ์ที่ทางทีมงานจะใช้เชื่อมต่อนั้น เป็นบอร์ด LoPy รุ่นที่ 4 ซึ่งอาจารย์ได้จัดเตรียมไว้ให้แล้ว ผลิตและพัฒนาโดยบริษัท Pycom บอร์ดนี้ใช้ Python ในการสั่งและควบคุมการทำงาน ซึ่งเป็นภาษาที่เขียนง่าย และไม่ซับซ้อนเท่า C++ ที่ใช้ใน Arduino (ทีมงาน #A มีประสบการณ์การใช้ Arduino มาก่อน)

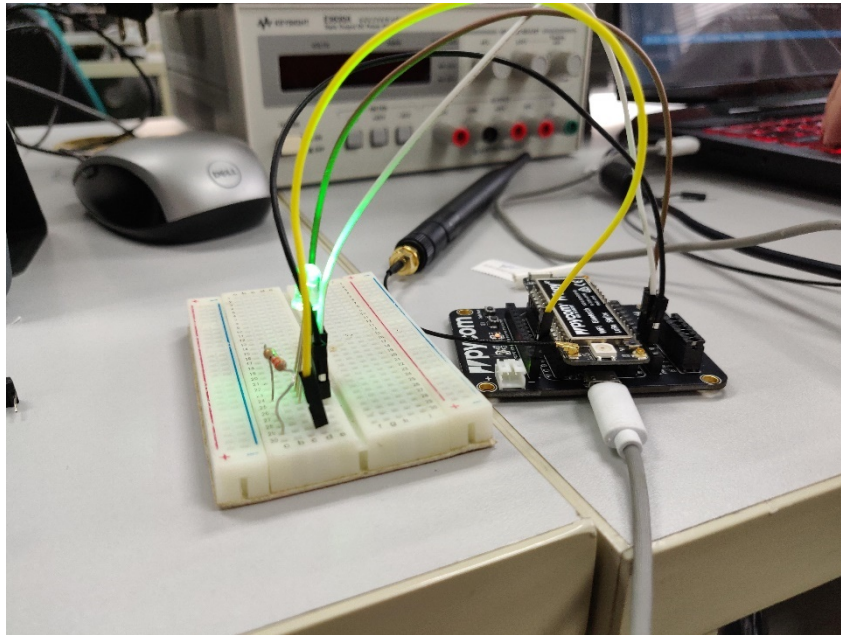


## หน้าตาของบอร์ด LoPy4 และ Athena ของมัน

โดย Schematics (รูปแบบของบอร์ด) ของบอร์ดนี้ก็หน้าชวนปวดหัวเหมือนกัน (เมื่อเทียบกับบอร์ดชนิดอื่นแล้ว น่าปวดหัวกว่า) ทางทีมงาน #A และ #B ก็ไม่รู้เหมือนกันว่าแต่ละขามันทำงานยังไง เสียบให้ถูกรู้ก็เพียงพอแล้ว

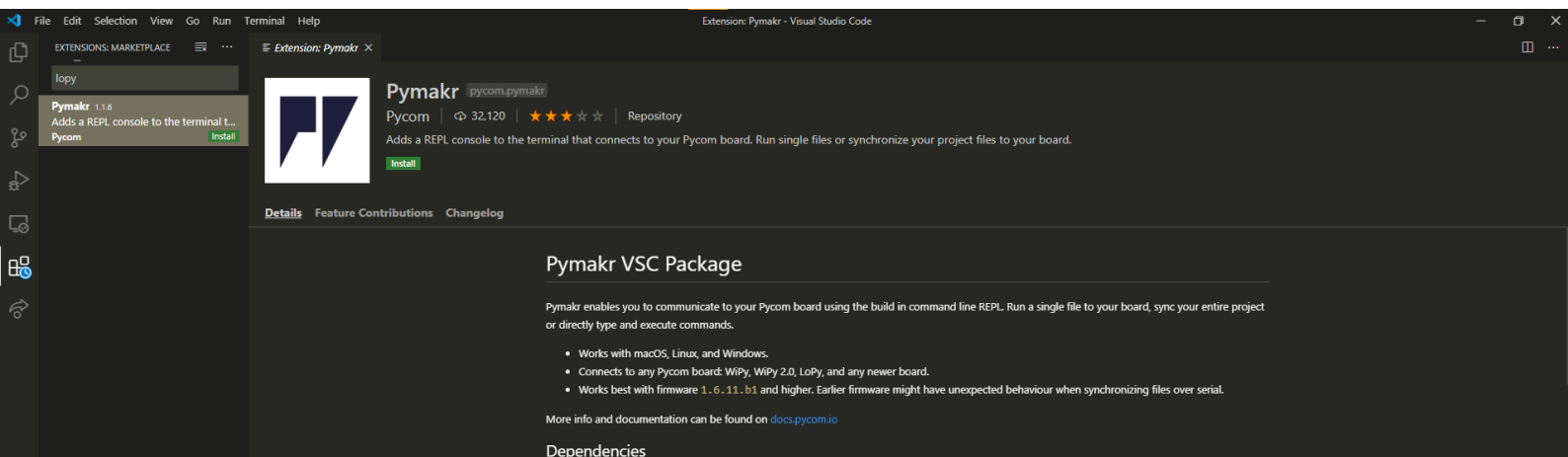


## “Schematics (Pinout Diagram)”



ทดสอบการทำงานของบอร์ดโดยการต่อวงจร LED ง่ายๆ (\*ห้ามลืมตัวต้านทาน R\*)

ในขั้นตอนต่อไปก็จะทำการเชื่อมต่อบอร์ดกับเครื่องคอมพิวเตอร์ โดยโปรแกรมที่ใช้คือ Visual Studio Code ซึ่งเป็น IDE ที่ใช้งานได้หลากหลายและสามารถลง Package เพิ่มเติมสำหรับอุปกรณ์ LoPy ได้

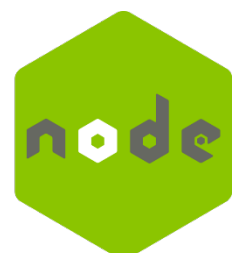


“Pymakr” คือชื่อ Package ที่ต้องลงใน VSCODE

อย่าลืมลง “NodeJS” กันด้วยนะครับ

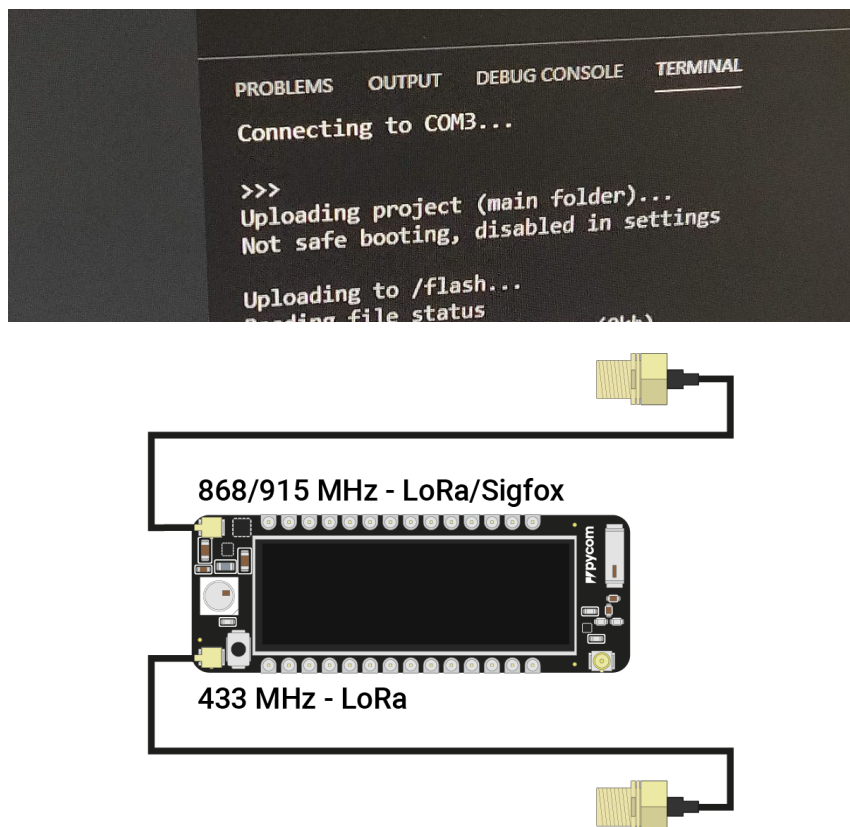
เดี๋ยวจะใช้งานกันไม่ได้ (มันคือ JavaScript Framework ตัวหนึ่ง)

<https://nodejs.org/en/>



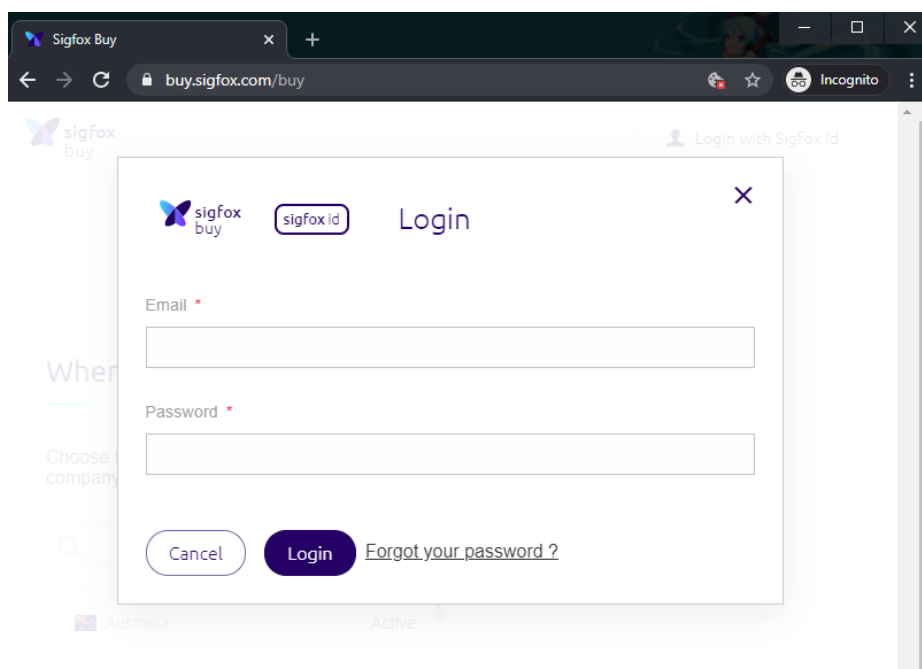


โดยเมื่อเชื่อมต่อด้วยสาย USB แล้ว จะเป็น Port COM X ซึ่งจะต้องปรับเปลี่ยนให้ตรงกับที่ VS CODE



COM3 คือตัวอย่าง และอย่าลืมเสียบเสาอากาศให้ถูกช่องด้วย

ขั้นตอนต่อไปคือ สมัครสมาชิกและเปิดใช้บริการ\*



ID Registration and Subscription Purchase

\*ซึ่งในประเทศไทยนั้น ผู้ให้บริการคือบริษัท ThingsOnNet ซึ่งในขณะนี้ (เมษายน 2020) ไม่มีประเทศไทยในรายชื่อประเทศที่เปิดบริการโดยตรงผ่านเว็บได้ ซึ่งต้องโทรไปขอเปิดบริการกับ บริษัทโดยตรงได้ที่ +66 020911230

\*และทั้งนี้ต้องสมัครสมาชิกบนเว็บ Sigfox ตามปกติด้วยครับ แต่ในกรณีที่ถ้าติดต่อ แล้วสตาฟบอกว่า จะทำการสมัคร Account ให้ แล้วให้เรา Reset password เองขั้นตอนนี้สามารถ เข้าได้เลย

เรียนน้องธนกฤตครับ

อยากทราบข้อมูลว่า

1. ทางน้องได้ใช้บริการ Sigfox Atlas สำหรับการนำข้อมูล Location บริการพิเศษของทาง Sigfox มาใช้หรือไม่ครับ
2. ทาง Sigfox มีแพคเกจตามจำนวน message uplink, downlink อยู่ตามรูปนี้ครับ อยากทราบว่าทางน้องอยากใช้แพคเกจไหนครับ

Basic	
Uplinks: ↑ 2 Daily	Downlinks: ↓ 4 Monthly
Plus	
Uplinks: ↑ 70 Daily	Downlinks: ↓ 2 Daily
Ultra	
Uplinks: ↑ 140 Daily	Downlinks: ↓ 4 Daily

ด้วยความนับถือ

[Redacted]

Tel: [Redacted]

บริษัท ดิงส์ ออน เน็ต จำกัด

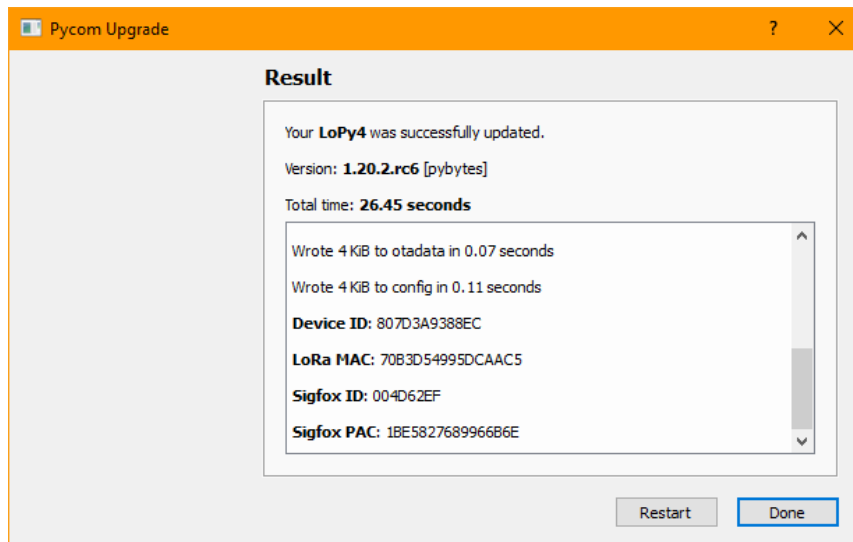
170/63 อาคารโอเชียน ทาวเวอร์ 1 ชั้น 20

ซอยสุขุมวิท 16 (สามมิตร) ถนน รัชดาภิเษก

แขวงคลองเตย เขตคลองเตย

กรุงเทพฯ 10110

อีเมลติดต่อกลับจากสตาฟของบริษัท ThingsOnNet พร้อม Package ในการใช้งาน



Sigfox ID และ Sigfox PAC ที่ได้จากโปรแกรม Firmware upgrade

Hardware : LoPy4

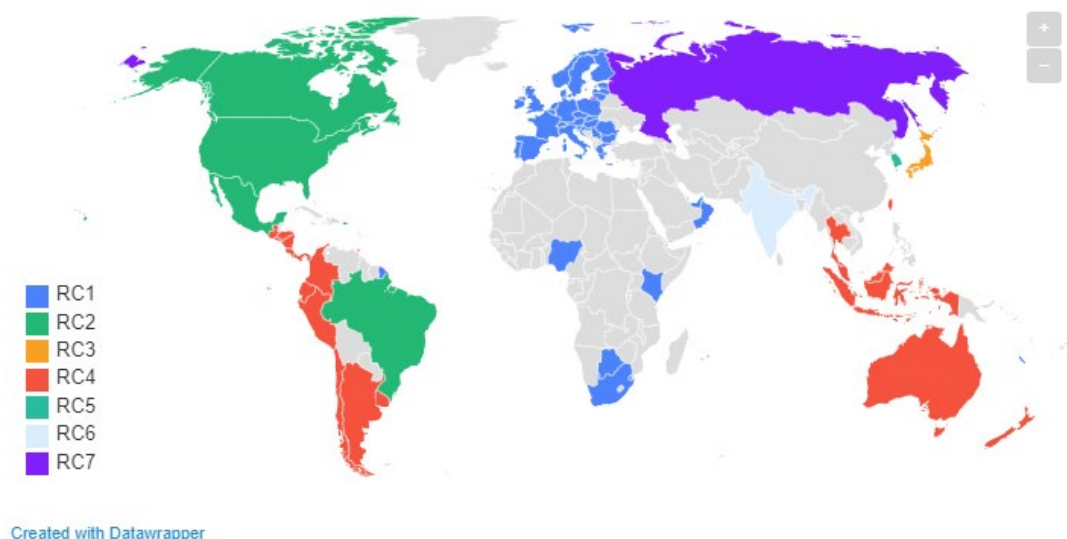
Device ID : 807D3A9388EC

LoRa MAC : 70B3D54995DCAAC5

LoPy4 Sigfox ID : 004D62EF

LoPy4 Sigfox PAC : 1BE5827689966B6E

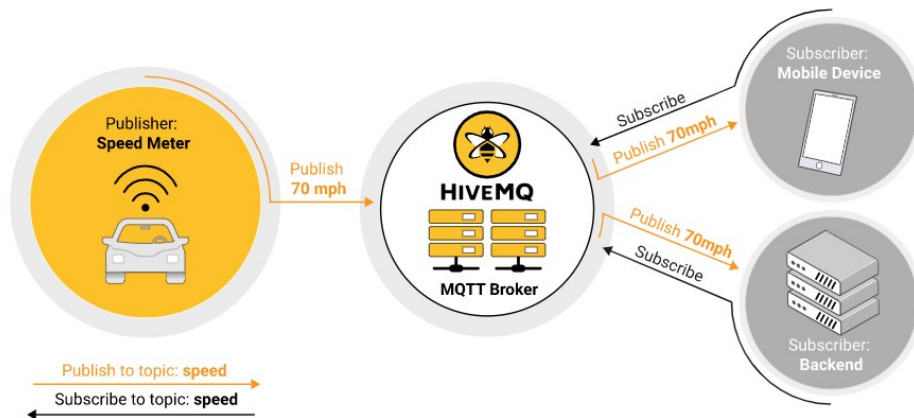
### Sigfox Geographical Availability



อย่าลืมดูพื้นที่ของคลื่นวิทยุ (Radio Region) ก่อนลงโค้ดด้วยครับ เดี่ยวจะใส่โค้ดผิดเขา  
ในประเทศไทยจะเป็น RC4



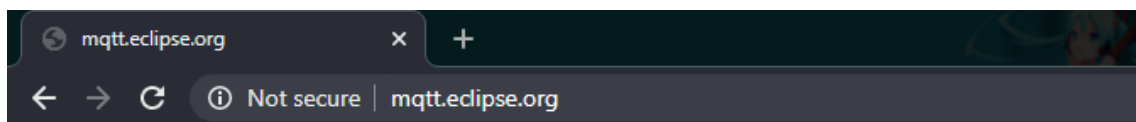
ก่อนที่จะไปใช้งาน Sigfox กันจริงๆ เรามาดูการทำ Publish และ Subscribe ของ MQTT กันก่อนครับ ซึ่งการ Publish คือการส่งข้อความไปยัง Broker เพื่อรอให้มีผู้ Subscribe คือรับข้อความนั่นเอง



MQTT Publish/Subscribe Architecture

## HiveMQ เป็น MQTT Broker รายหนึ่ง

และเมื่อทราบการทำงานของ การ Publish และ Subscribe แล้ว เราก็จะทำการเขียนโค้ดทำการ Publish และ Subscribe โดย Broker ที่ใช้จะเป็นของ Eclipse ซึ่งเปิดเป็นสาธารณะและฟรี



## mqtt.eclipse.org

This is a public test MQTT broker service. It currently listens on the following ports:

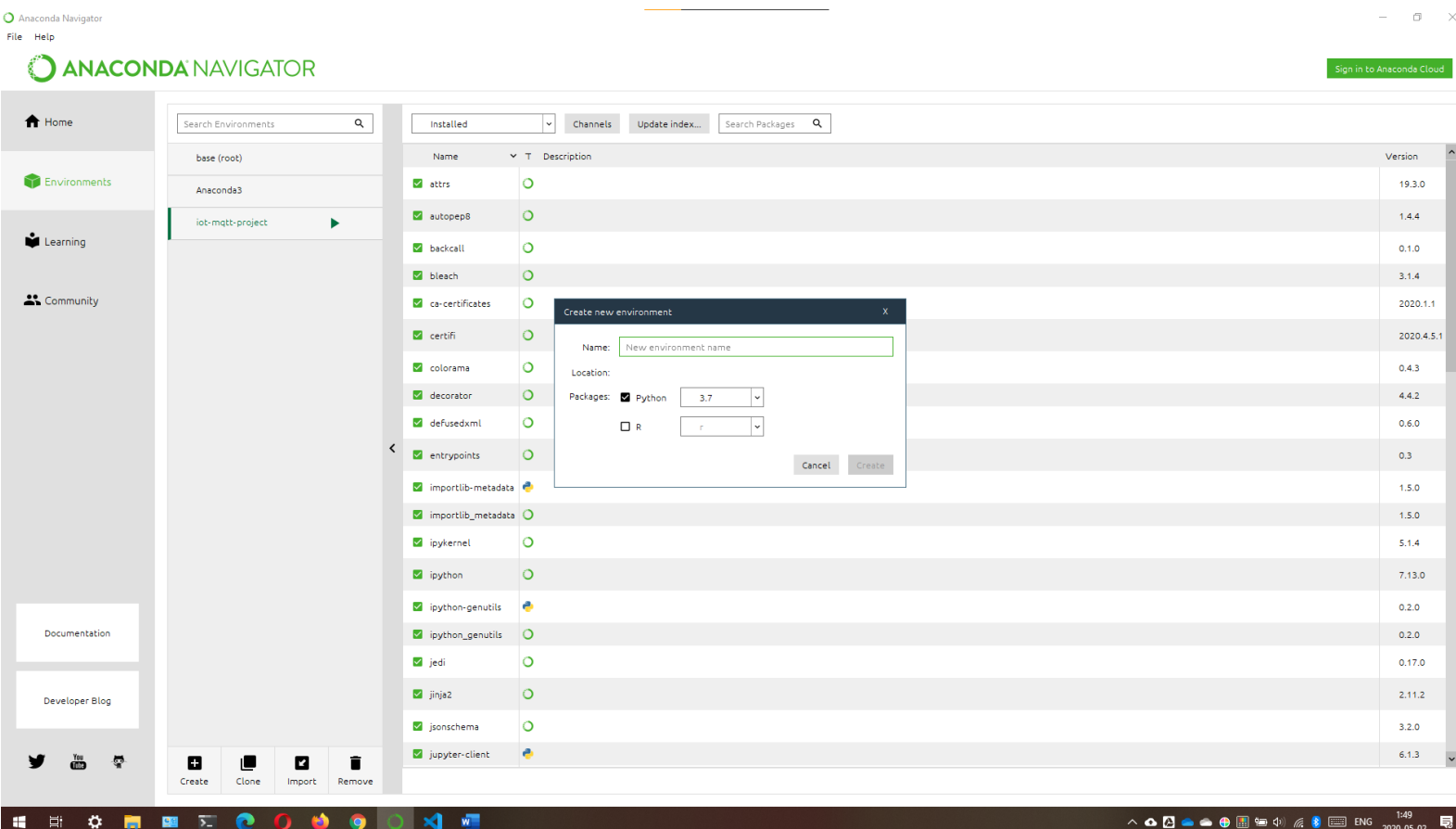
- 1883 : MQTT over unencrypted TCP
- 8883 : MQTT over encrypted TCP
- 80 : MQTT over unencrypted WebSockets (note: URL must be **/mqtt** )
- 443 : MQTT over encrypted WebSockets (note: URL must be **/mqtt** )

หน้าเว็บมันก็จะโล่งๆ แต่เราไปดูที่เบื้องหลังการทำงานดีกว่า

<http://mqtt.eclipse.org/>

จากนั้นเราก็ตั้งค่า Environment ของ Python โดยใช้ Anaconda กันครับ

การสร้าง Environment จำลองนั้น เป็นการป้องกันไม่ให้ตัว Environment หลักถูกรบกวน เพื่อการทำงานของโค้ดเราผิดพลาด มีการลบ Package หรือการขัดแย้งกันเองของ Package (Package Conflict) ในส่วนนี้ถ้าไม่ต้องการ สามารถข้ามได้



และเมื่อทำการลง Package ที่จำเป็นแล้ว เช่น autopep8 jupyter networkx ผ่าน Anaconda ก็จะทำาการเปิดโปรแกรม VSCODE กัน ส่วนด้านล่างซ้ายของรูปภาพด้านล่าง จะเป็น Environment ที่เราสร้าง และเราต้องทำการลง Package ที่ชื่อว่า paho-mqtt ผ่าน Terminal ของ VSCODE โดยใช้ `pip install paho-mqtt`

```
Copyright (C) Microsoft Corporation. All rights reserved.

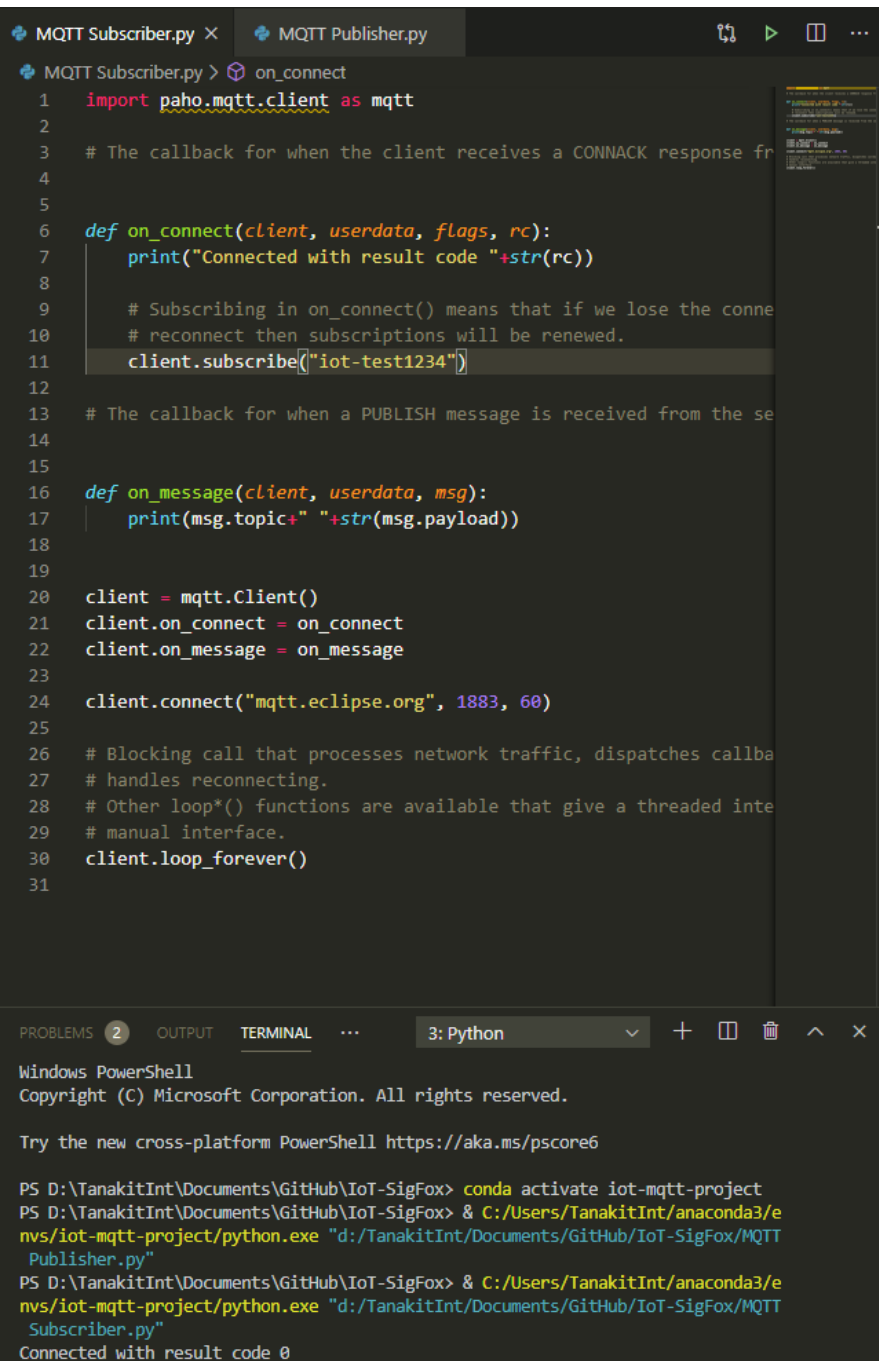
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> conda activate iot-mqtt-project
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> pip install paho-mqtt
Collecting paho-mqtt
  Downloading paho-mqtt-1.5.0.tar.gz (99 kB)
    | 99 kB 838 kB/s
Building wheels for collected packages: paho-mqtt
  Building wheel for paho-mqtt (setup.py) ... done
  Created wheel for paho-mqtt: filename=paho_mqtt-1.5.0-py3-none-any.whl size=6
  4726 sha256=e4088b543978ce830f60290f4a12f378f986916c9fddeb585ad309913a32175b
  Stored in directory: c:\users\tanakitint\appdata\local\pip\cache\wheels\0d\7c
  \fb\05123381bd60c57ffcdc6fcc1c26e585dedee85b8c1625e2c1
Successfully built paho-mqtt
Installing collected packages: paho-mqtt
```

โดย Source code ทั้งหมดจะอยู่ที่

<https://github.com/TanakitInt/IoT-SigFox>

ซึ่งสามารถเข้าไป Download มาเป็นตัวอย่างการใช้งานได้



```
MQTT Subscriber.py x MQTT Publisher.py
MQTT Subscriber.py > on_connect
1 import paho.mqtt.client as mqtt
2
3 # The callback for when the client receives a CONNACK response fr
4
5
6 def on_connect(client, userdata, flags, rc):
7     print("Connected with result code "+str(rc))
8
9     # Subscribing in on_connect() means that if we lose the conne
10    # reconnect then subscriptions will be renewed.
11    client.subscribe(["iot-test1234"])
12
13 # The callback for when a PUBLISH message is received from the se
14
15
16 def on_message(client, userdata, msg):
17     print(msg.topic+" "+str(msg.payload))
18
19
20 client = mqtt.Client()
21 client.on_connect = on_connect
22 client.on_message = on_message
23
24 client.connect("mqtt.eclipse.org", 1883, 60)
25
26 # Blocking call that processes network traffic, dispatches callba
27 # handles reconnecting.
28 # Other loop*() functions are available that give a threaded inte
29 # manual interface.
30 client.loop_forever()
31
```

```
PROBLEMS 2 OUTPUT TERMINAL ... 3: Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

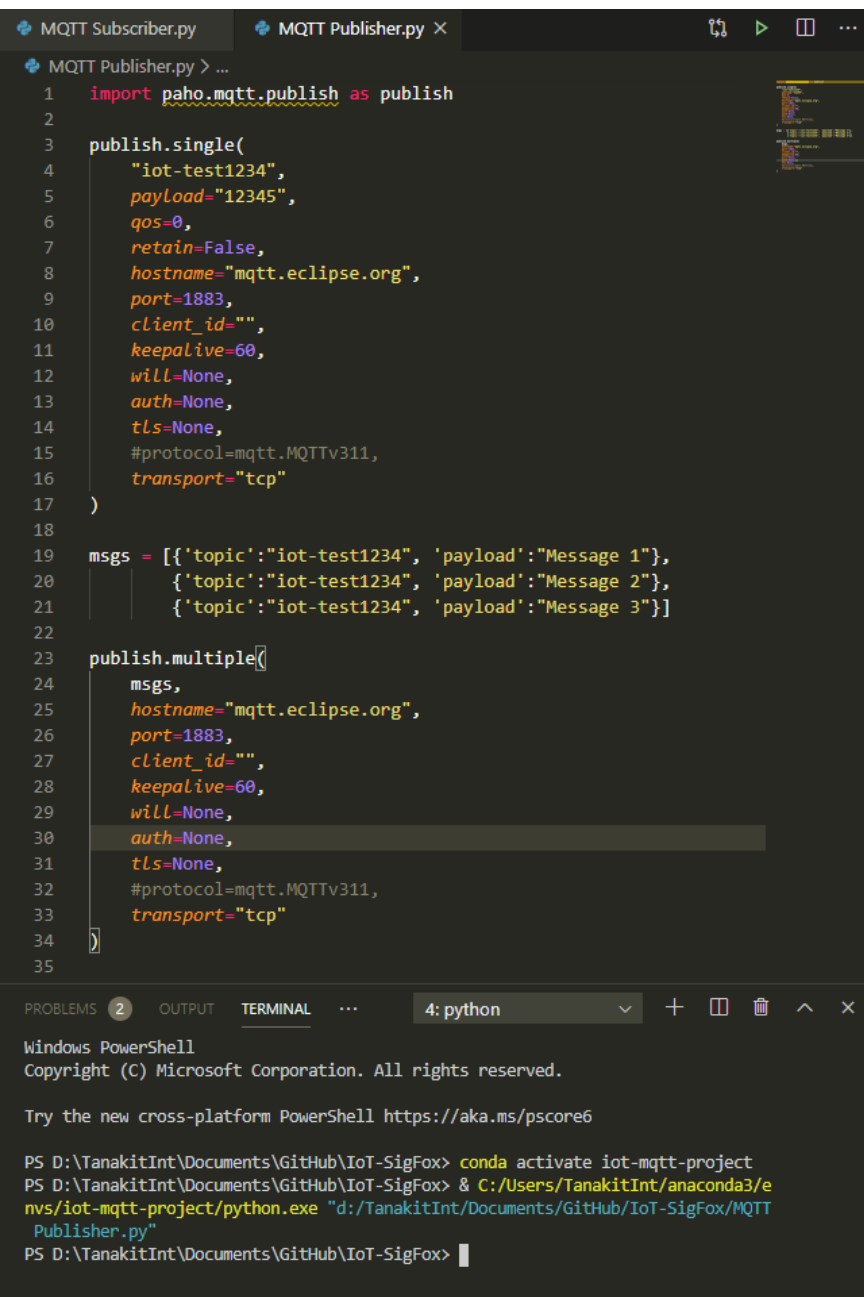
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> conda activate iot-mqtt-project
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> & C:/Users/TanakitInt/anaconda3/e
nvs/iot-mqtt-project/python.exe "d:/TanakitInt/Documents/GitHub/IoT-SigFox/MQTT
Publisher.py"
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> & C:/Users/TanakitInt/anaconda3/e
nvs/iot-mqtt-project/python.exe "d:/TanakitInt/Documents/GitHub/IoT-SigFox/MQTT
Subscriber.py"
Connected with result code 0
```

## MQTT Subscriber

จากนั้นก็จะทำการเขียนโค้ด เพื่อสร้าง Subscriber โดยระบุ Topic ที่ต้องการเป็น “iot-test1234” และระบุ Broker ที่ต้องการโดยจะเป็น “mqtt.eclipse.org” ที่ port 1883 และ keep alive 60

จากนั้นให้ Run โปรแกรมจะอยู่ด้านบนขวา ที่เห็นได้จากด้านล่างสุดที่เป็น Terminal ได้ว่าถ้ามันแสดงค่าออกมาเป็น “Connected with status code 0” ก็แสดงว่าทำการเชื่อมต่อกับ Broker สำเร็จแล้ว



```
MQTT Subscriber.py  MQTT Publisher.py X
MQTT Publisher.py > ...
1  import paho.mqtt.publish as publish
2
3  publish.single(
4      "iot-test1234",
5      payload="12345",
6      qos=0,
7      retain=False,
8      hostname="mqtt.eclipse.org",
9      port=1883,
10     client_id="",
11     keepalive=60,
12     will=None,
13     auth=None,
14     tls=None,
15     #protocol=mqtt.MQTTv311,
16     transport="tcp"
17 )
18
19 msgs = [{ 'topic': "iot-test1234", 'payload': "Message 1" },
20         { 'topic': "iot-test1234", 'payload': "Message 2" },
21         { 'topic': "iot-test1234", 'payload': "Message 3" }]
22
23 publish.multiple(
24     msgs,
25     hostname="mqtt.eclipse.org",
26     port=1883,
27     client_id="",
28     keepalive=60,
29     will=None,
30     auth=None,
31     tls=None,
32     #protocol=mqtt.MQTTv311,
33     transport="tcp"
34 )
35
PROBLEMS 2  OUTPUT  TERMINAL  ...  4: python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

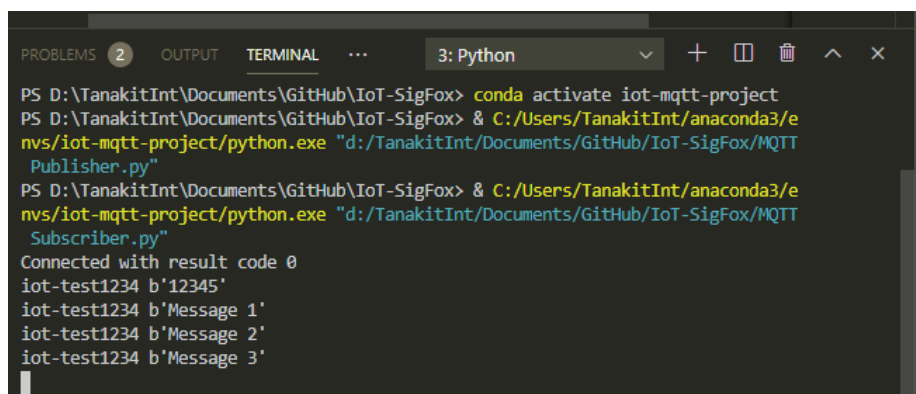
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> conda activate iot-mqtt-project
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> & C:/Users/TanakitInt/anaconda3/envs/iot-mqtt-project/python.exe "d:/TanakitInt/Documents/GitHub/IoT-SigFox/MQTT
Publisher.py"
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox>
```

## MQTT Publisher

และเมื่อเขียน Subscriber ไป  
แล้วก็จะมาเขียน Publisher กันโดยการ  
Publish จะมี 2 แบบคือ Single ที่ส่งครั้ง  
เดียว และ Multiple ที่ส่งหลายครั้ง ใน  
ตัวอย่างจะลองส่งทั้ง 2 แบบเลย

โดยทั้งหมดจะต้องระบุ Broker,  
Topic name, port และ keep alive  
เป็น Parameter ที่จำเป็น ส่วนตัวอื่นๆ  
สามารถรับค่าได้ตามความต้องการ ที่  
สำคัญอย่าลืมใส่ payload ด้วยเพราะ  
นั่นคือข้อความที่เราจะส่ง

จากนั้นให้ Run เพื่อ Publish  
ขึ้นไปยัง Topic ที่เราตั้งไว้



```
PROBLEMS 2  OUTPUT  TERMINAL  ...  3: Python
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> conda activate iot-mqtt-project
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> & C:/Users/TanakitInt/anaconda3/envs/iot-mqtt-project/python.exe "d:/TanakitInt/Documents/GitHub/IoT-SigFox/MQTT
Publisher.py"
PS D:\TanakitInt\Documents\GitHub\IoT-SigFox> & C:/Users/TanakitInt/anaconda3/envs/iot-mqtt-project/python.exe "d:/TanakitInt/Documents/GitHub/IoT-SigFox/MQTT
Subscriber.py"
Connected with result code 0
iot-test1234 b'12345'
iot-test1234 b'Message 1'
iot-test1234 b'Message 2'
iot-test1234 b'Message 3'
```

ข้อความขึ้นแล้ว ทำการ Publish and Subscribe สำเร็จ!



ในขั้นตอนต่อไปนี้ ก็จะเป็นการเริ่มกับอุปกรณ์ Sigfox กันจริงๆ ซึ่งทางทีมงานได้ทำการยืมมาจากบริษัท ThingsOnNet ซึ่งทางบริษัทแจ้งว่าในช่วงวิกฤต COVID-19 ทำให้เสาที่ปล่อยสัญญาณนั้นใช้การไม่ได้ (ช่วงเวลาที่ทีมงานทำงานอยู่ในช่วงเดือน เมษายน - พฤษภาคม 2020) เนื่องจากไม่สามารถไปบำรุงดูแลรักษาจากการไฟฟ้าได้ ซึ่งตัวกล่องตัวปล่อยสัญญาณ Sigfox จะมีหน้าตาเป็นแบบนี้



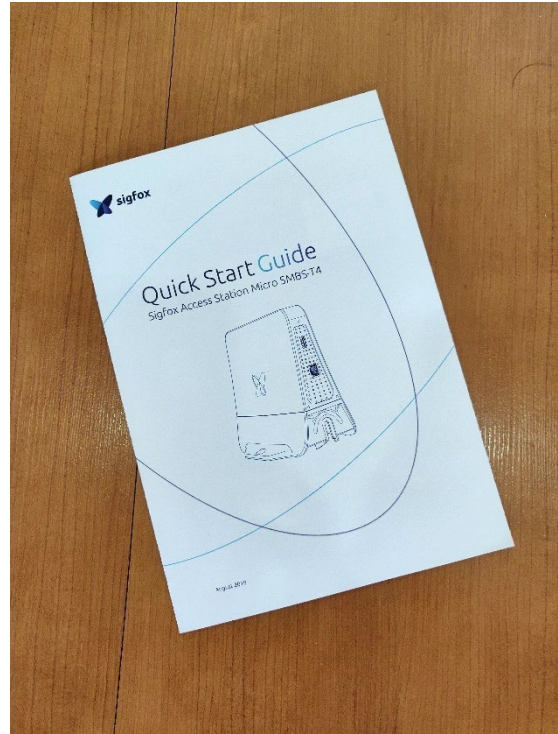
Sigfox Access Station Micro Kit

หลักจากนี้ก็เป็นตั้งค่าอุปกรณ์ให้สามารถปล่อยสัญญาณได้ โดยหลักการเบื้องต้นคือจะทำการเชื่อมต่อผ่านสาย LAN โดยเป็น **Power over Ethernet (PoE)** ไม่ต้องเสียบสาย Power แยก เหมาะสำหรับการใช้งาน In-Door และไฟที่ขึ้นนั้นต้องเป็นสีเขียว เพื่อยืนยันว่าปล่อยสัญญาณแล้ว

**\*โปรดทำตามจากคู่มือที่มีให้ในกล่อง**



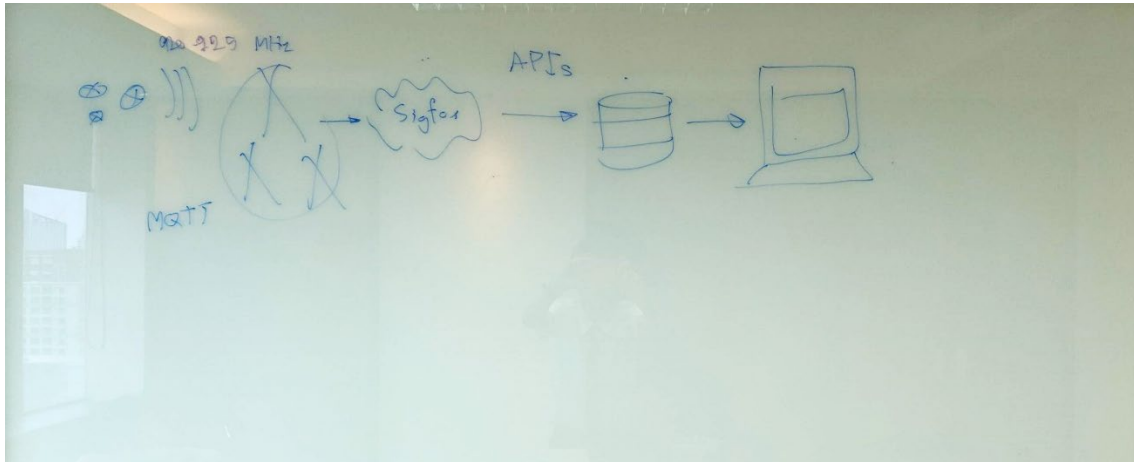
ซ้าย – Power over Ethernet (PoE)



ขวา – Manual (คู่มือ)

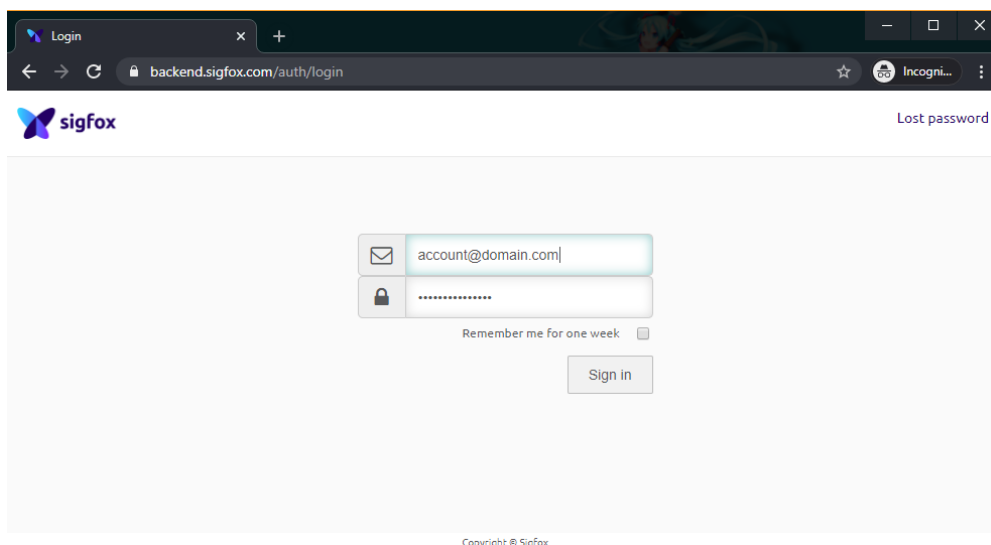


ไฟสีเขียว หมายถึงกำลังปล่อยสัญญาณ



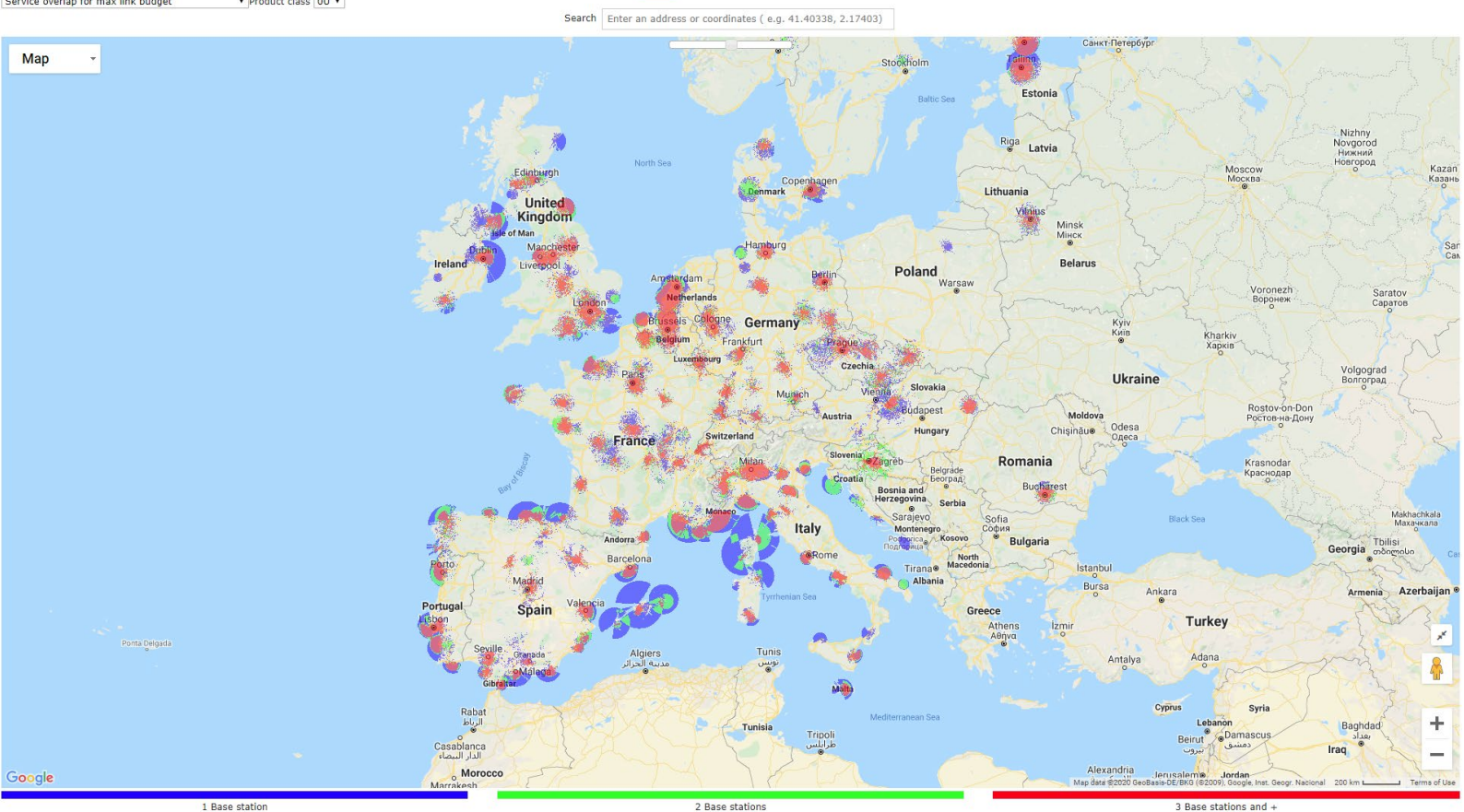
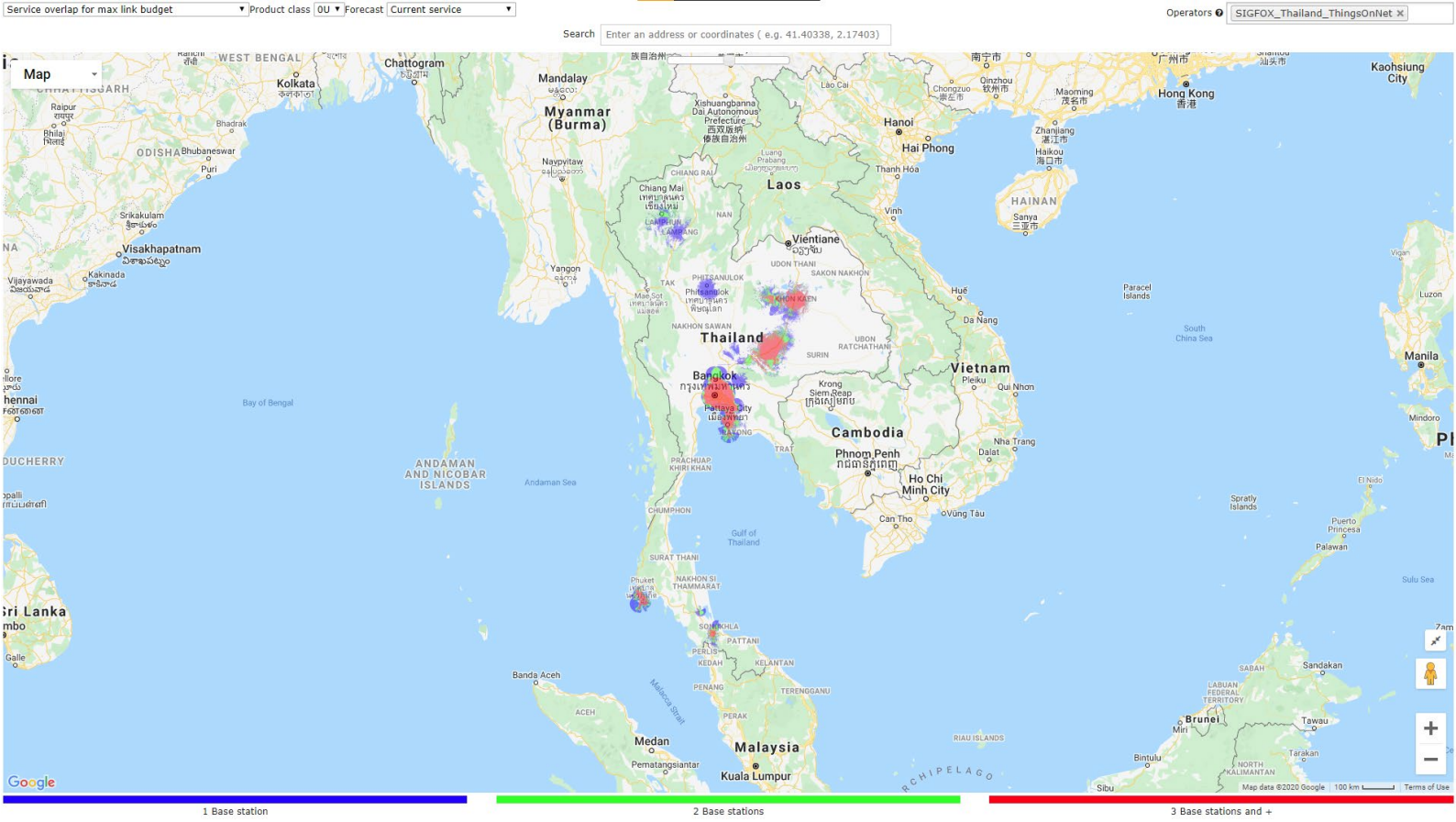
### Bonus : ทางสต๊าฟ ThingsOnNet ได้แนะนำ Diagram ของ Sigfox แบบคร่าวๆ ให้ดู

กลับมาที่ LoPy ของเรากันครับ ตอนนี้เรายังไม่สามารถใช้งานอุปกรณ์ได้โดยตรงนะครับ (ใจเย็นๆ ใกล้เคียง) เมื่อเราได้ทำการสมัคร Account และได้ทำการซื้อสัญญาการเชื่อมต่อเรียบร้อยแล้ว (กลับไปขั้นตอนแรกๆ) ให้ไปที่หน้าเว็บ Sigfox Control Panel



<https://backend.sigfox.com/>



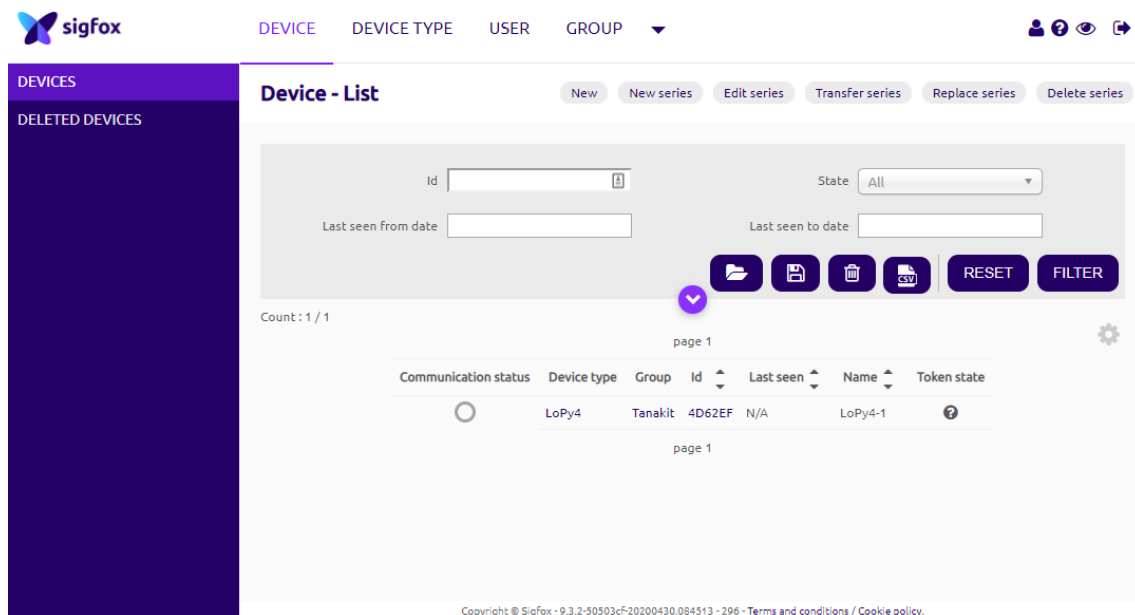


Bonus : Coverage map ของ Sigfox ในประเทศไทย ซึ่งตอนนี้ (พฤษภาคม 2020) จะมีเฉพาะในกรุงเทพและหัวเมืองใหญ่ๆ และ Coverage map ของทวีปยุโรป

<https://backend.sigfox.com/welcome/coverage>



หากเป็นการสมัครจากเว็บโดยตรง เราจะต้องกรอกข้อมูลอุปกรณ์ลงไป อย่างเช่น Hardware Type, Device ID, Sigfox ID, Sigfox PAC แต่หากสต๊าฟได้ทำการสร้าง Account ให้เราแล้ว เราก็จะเห็นอุปกรณ์ใน Control Panel เลยครับ



## Device ใน Sigfox Backend

จากนั้นให้เราเชื่อมต่อไปยังเครือข่าย Sigfox สามารถทำตามคู่มือบนเว็บ Pycom ได้ที่



<https://docs.pycom.io/tutorials/sigfox/>



```
Sigfox Sample > Sigfox_connect.py > ...
1  from network import Sigfox
2  import socket
3
4  # init Sigfox for RCZ4 (Thailand)
5  sigfox = Sigfox(mode=Sigfox.SIGFOX, rcz=Sigfox.RCZ4)
6
7  # create a Sigfox socket
8  s = socket.socket(socket.AF_SIGFOX, socket.SOCK_RAW)
9
10 # make the socket blocking
11 s.setblocking(True)
12
13 # configure it as uplink only
14 s.setsockopt(socket.SOL_SIGFOX, socket.SO_RX, False)
15
16 # send some bytes
17 s.send(bytes([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]))
18
19
```

PROBLEMS 12 OUTPUT TERMINAL ... 2: Pymakr Console

```
Type "help()" for more information.
>>>
>>> Running d:\TanakitInt\Documents\GitHub\IoT-SigFox\Sigfox Sample\Sigfox_connect.py
>>>
>>>
>>>
Pycom MicroPython 1.20.2.rc6 [v1.11-01f49f7] on 2020-02-28; LoPy4 with ESP32
Pybytes Version: 1.3.1
Type "help()" for more information.
>>>
>>>
```

ซึ่งจากโค้ดการเชื่อมต่อนั้น เรา  
ก็ได้ทำการเชื่อมต่ออุปกรณ์ LoPy4 กับ  
เครือข่าย Sigfox สำเร็จ

Communication status	Device type	Group	Id	Last seen	Name	Token state
	LoPy4	Tanakit	4D62EF	N/A	LoPy4-1	

Communication status	Device type	Group	Id	Last seen	Name	Token state
	LoPy4	Tanakit	4D62EF	2020-05-17 15:40:03	LoPy4-1	

บน – ก่อนเชื่อมต่อ Sigfox

ล่าง – หลังเชื่อมต่อ Sigfox

เมื่อเชื่อมต่อสำเร็จ เราก็จะเห็นค่าของอุปกรณ์ เป็นหน้าตาแบบนี้

INFORMATION

LOCATION

MESSAGES

EVENTS

STATISTICS

EVENT CONFIGURATION

### Device 4D62EF - Information

Name: LoPy4-1

Protocol: V1

Activable state: ☒ ⓘ

Sequence number: 2 (2020-05-17 15:40:03)

Trash sequence number: N/A (N/A)

Last seen: 2020-05-17 15:40:03

PAC: 46BEA235819BE944


Product certificate: P\_00CA\_EE5A\_01


Latitude: 0.000 (degrees)

Longitude: 0.000 (degrees)

Device type: LoPy4

State: OK

Link Quality Indicator:  ⓘ

Communication status: 

Contract: Free Contract for ToN Pilot test

Activation date: 2020-05-17 15:37:58

Token validity: 2021-05-17

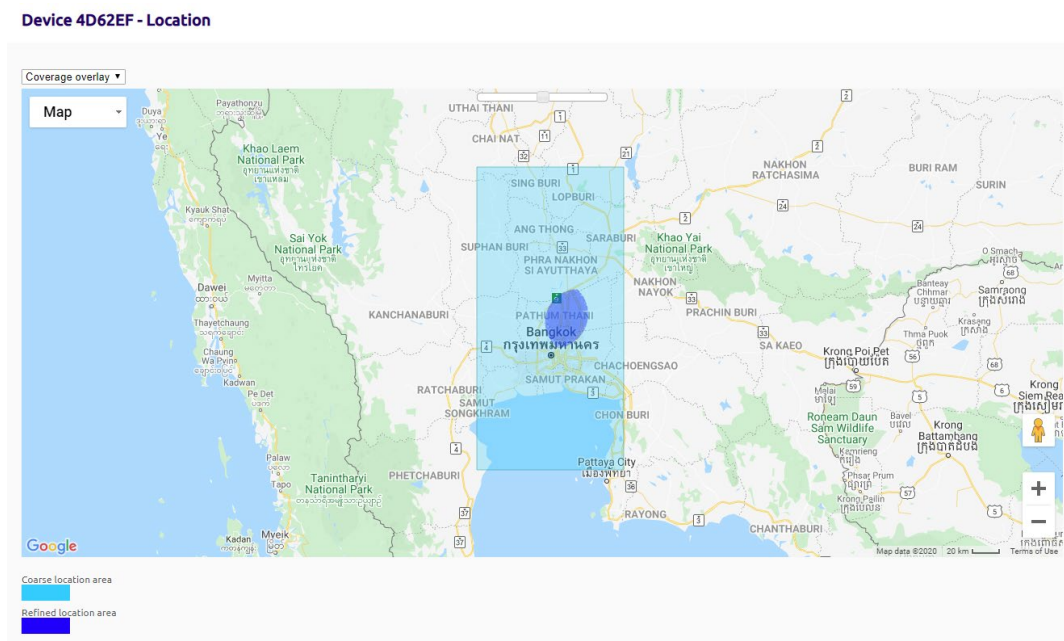
Unsubscription date: N/A  ⓘ

Subscription automatic renewal status: Allowed

Subscription automatic renewal: ☒ ⓘ

Creation date: 2020-05-07 10:56:41

## Device Information บน Sigfox Backend



รู้ตัวว่าบ้านที่ทีมงานอยู่กรุงเทพตอนเหนือ

Device 4D62EF - Messages

From date

To date

RESETFILTERCSV

page 1

Time	Seq Num	Data / Decoding	LQI	Callbacks	Location
2020-05-17 15:40:03	2	0102030405060708090a0b0c	<div></div>	<div></div>	<div></div>
2020-05-17 15:38:18	1	0102030405060708090a0b0c	<div></div>	<div></div>	<div></div>
2020-05-17 15:37:55	0	0102030405060708090a0b0c	<div></div>	<div></div>	<div></div>

page 1

A maximum of fifteen base stations can be displayed in this page. Please refer to APIs if you need to collect information about 15+ base stations.

Message ที่ทีมงานได้ทดลองส่งไป

Device 4D62EF - Events

From date

To date

Type

All

Severity

All

RESETFILTER

page 1

Time	Type	Severity	Source id	Description	Status
2020-05-17 15:37:58	Receive first payable message	INFO	4D62EF	Device #4D62EF has sent its first payable message	<div></div>
2020-05-07 10:56:41	On contract	INFO	4D62EF	Device #4D62EF has been associated to a contract	<div></div>

page 1

ส่วนของ Event ที่เวลา 2020-05-17 15:37:58 ได้รับ Message ครั้งแรก



The image shows a development environment with a code editor on the left and a web interface on the right. The code editor displays a Python script named `Sigfox_connect.py` that uses the `Sigfox` library to connect to a Sigfox network and send data. The terminal shows the execution of the script, which successfully connects to the network and sends a message. The web interface, titled "Device 4D62EF - Messages", displays a table of messages sent by the device. The table includes columns for Time, Seq Num, Data / Decoding, LQI, Callbacks, and Location. The messages are listed in descending order of time, with the most recent message at the top.

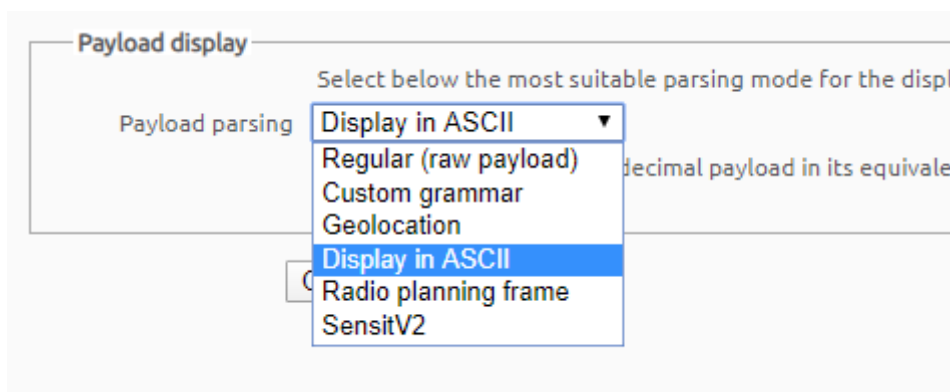
Time	Seq Num	Data / Decoding	LQI	Callbacks	Location
2020-05-17 16:05:09	7	49276d204c6f507934	...	...	...
2020-05-17 16:04:05	6	49276d204c6f507934	...	...	...
2020-05-17 16:02:27	5	49276d204c6f507934	...	...	...
2020-05-17 16:02:23	4	49276d204c6f507934	...	...	...
2020-05-17 16:02:02	3	0102030405060708090a0b0c	...	...	...
2020-05-17 15:40:03	2	0102030405060708090a0b0c	...	...	...
2020-05-17 15:38:18	1	0102030405060708090a0b0c	...	...	...
2020-05-17 15:37:55	0	0102030405060708090a0b0c	...	...	...

เทียบโค้ดที่เขียนและรันส่งไปที่ Lopy4 และ Message ถูกส่งขึ้นไปยัง Sigfox Network  
ขั้นตอนที่เรา Publish ได้สำเร็จแล้ว เย้!

ข้อความที่เห็นในเว็บนั้นสามารถนำไปใช้งานต่อได้โดยการใช้งาน API และ โค้ดที่เขียนเป็น  
ตัวอย่างนั้นสามารถนำไป Embedded กับอุปกรณ์ IoT ได้เพื่อใช้สำหรับการ Publish ข้อมูลจริงๆ ได้

เพิ่มเติมสำหรับการดู Message ที่ส่งมา

ใน Tab “Device Type” ให้กดเลือกอุปกรณ์ แล้วกดเลือก “Edit” ไปที่ “Payload  
Display” แล้วเลือก “Display in ASCII” เพื่อแสดงให้เห็นตัวอักษรโดยตรงจากหน้า Message



Time	Seq Num	Data / Decoding	LQI	Callbacks	Location
2020-05-17 16:05:09	7	49276d204c6f507934 ASCII: I'm LoPy4			

เห็นข้อความที่ส่งมาแล้ว! “I'm LoPy4”

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Radius (Computed location);"Longitude (Computed location);"Latitude (Computed location);"Source (Computed location);"Country";"Data";"Device ID";"Data1 (Down															
2	;;;"49276d204c6f507934";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"7";"2020-05-17 16:05:09"															
3	;;;"49276d204c6f507934";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"6";"2020-05-17 16:04:05"															
4	;;;"49276d204c6f507934";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"5";"2020-05-17 16:02:27"															
5	;;;"49276d204c6f507934";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"4";"2020-05-17 16:02:23"															
6	;;;"0102030405060708090a0b0c";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"3";"2020-05-17 16:02:02"															
7	;;;"0102030405060708090a0b0c";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"2";"2020-05-17 15:40:03"															
8	;;;"0102030405060708090a0b0c";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"1";"2020-05-17 15:38:18"															
9	;;;"0102030405060708090a0b0c";"004D62EF";;;;"Limit";"Good";"N/A";"SIGFOX_Thailand_ThingsOnNet";"0";"0";"2020-05-17 15:37:55"															
10																
11																
12																

เรายังสามารถ Export CSV เพื่อเอาข้อมูลที่ LoPy4 ส่งมา นำไปใช้งานต่อได้ด้วย หากถ้าเป็นข้อมูลฉุกเฉิน เราก็สามารถนำข้อมูลนี้ไป Visualize เพื่อให้เห็นกราฟอีกด้วย

น่าเสียดายที่ทีมงานส่งได้เพียง 8 Message เพราะหลังจากที่ได้ทดสอบส่ง Message เป็นครั้งที่ 9 มันก็ส่งไม่ได้อีกเลย อาจเป็นข้อจำกัดที่ Sigfox ให้ส่งได้แค่ 8 Message หรืออาจจะเป็นปัญหาอื่นๆ อย่างเช่น Hardware ทางทีมงานก็ไม่ทราบเหมือนกัน

สรุปโดยรวมแล้ว Sigfox เป็นเครือข่าย IoT ที่น่าสนใจมาก ทั้งมีราคาถูก ใช้พลังงานต่ำและง่ายต่อการ Setup เพียงแต่ต้องลงทุนด้านอุปกรณ์และเครือข่ายในขั้นตอนแรกๆ ก่อน สำหรับการใช้ในครัวเรือน (Residential use) ยังเป็นสิ่งที่ใหม่และท้าทายมาก ซึ่งเป็นสิ่งที่ดีในการเรียนรู้ เพราะในอนาคต เมืองจะเต็มไปด้วยอุปกรณ์ที่เชื่อมต่ออินเทอร์เน็ตเต็มไปหมด ทั้งรถเมล์ รถไฟ สนามบิน เราได้ข้อมูลจากเครือข่าย IoT มาใช้งานต่อเพื่อประโยชน์ต่างๆ มากมาย

## Useful Link :

Creator's source code :

<https://github.com/TanakitInt/IoT-SigFox>

## Pycom

<https://docs.pycom.io/gitbook/assets/lopy4-pinout.pdf>

[https://docs.pycom.io/gitbook/assets/specsheets/Pycom\\_002\\_Specsheets\\_LoPy4\\_v2.pdf](https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_LoPy4_v2.pdf)

<https://docs.pycom.io/datasheets/development/lopy4/>

<https://docs.pycom.io/gettingstarted/connection/lopy4/>

<https://docs.pycom.io/tutorials/sigfox/>

<https://pycom.io/downloads/>

<https://forum.pycom.io/topic/3029/solved-unable-to-import-from-network>

## Sigfox and ThingsOnNet

<https://buy.sigfox.com/>

<https://id.sigfox.com/sign-up>

<https://sigfoxsurvey.typeform.com/to/NNPvzU?source=sigfox.com>

<https://build.sigfox.com/sigfox-radio-configurations-rc>

[https://www.thingsonnet.net/contact\\_us.php](https://www.thingsonnet.net/contact_us.php)

<https://support.sigfox.com/questions/128/how-to-decode-the-frame-in-sigfox-backend.html>

\*\* (Google Web cache)

## Paho-MQTT

<https://pypi.org/project/paho-mqtt/>

## Eclipse MQTT

<http://mqtt.eclipse.org/>

## HiveMQ

<https://www.hivemq.com/blog/mqtt-client-library-paho-python/>

<http://www.hivemq.com/demos/websocket-client/>

## Tutorial Book :

<https://books.google.com/books?id=1BGLDwAAQBAJ&printsec=frontcover#v=onepage&q&f=false>