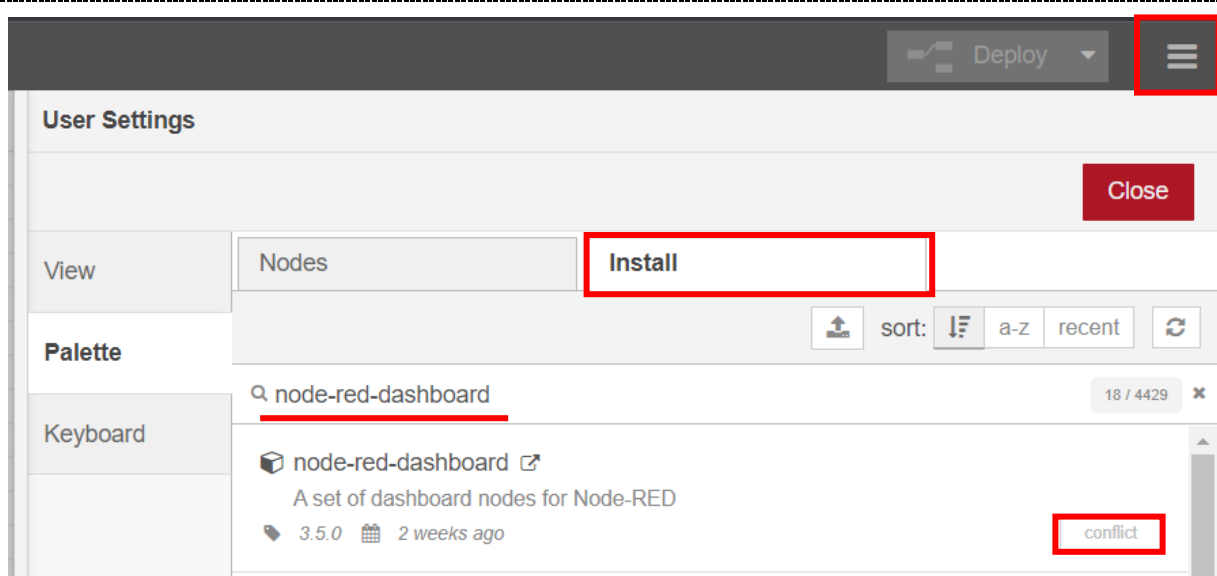


การสร้าง MQTT Server บน Raspberry Pi เพื่อใช้งาน Chatbot LINE ในฟาร์มอัจฉริยะ Chatbot LINE from Raspberry Pi MQTT Server for Smart Farming	
ชื่อ-สกุล : นางสาวขวัญจิรา พันธุ์เกตุ	รหัสนักศึกษา : B6321451
6/6 –คำถามท้ายบทเพื่อทดสอบความเข้าใจ	

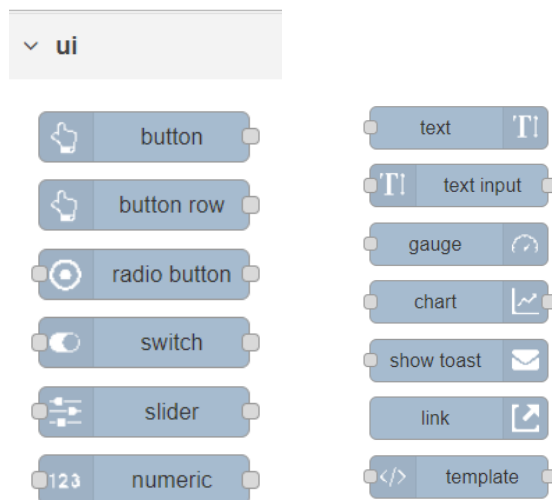
### Quiz\_201 – Node-RED UI

- แสดงข้อมูลที่ได้ทำการทดสอบ

Code nodejs -- ไปที่ Install -> ค้นหา node-red-dashboard แล้วกดติดตั้ง



เมื่อติดตั้งเสร็จที่ tab ด้านซ้ายจะได้ node ใหม่เพิ่มเข้ามา



Code nodejs – เลือก



แล้วตั้งค่าตามนี้จากนั้นกด Done

### Edit function node

Delete

Cancel

Done

#### Properties

Name

set\_high

Setup

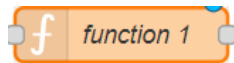
On Start

On Message

On Stop

```
1 msg.payload = 1;
2 return msg;
```

Code nodejs – เลือก



แล้วตั้งค่าตามนี้จากนั้นกด Done

### Edit function node

Delete

Cancel

Done

#### Properties

Name

set\_low

Setup

On Start

On Message

On Stop

```
1 msg.payload = 0;
2 return msg;
3
```

Code nodejs – เลือก



แล้วตั้งค่าตามนี้จากนั้นกด Done

**Edit function node**

Delete Cancel Done

**Properties**

Name

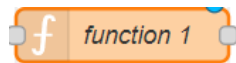
Setup On Start **On Message** On Stop

```

1  if (msg.payload == "true") {
2    |   flow.set("switch_state", "true");
3  }
4  else {
5    |   flow.set("switch_state", "false");
6    |   msg.payload = 0
7    |   return msg;
8  }
9

```

Code nodejs – เลือก



แล้วตั้งค่าตามนี้จากนั้นกด Done

**Edit function node**

Delete Cancel Done

**Properties**

Name

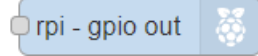
Setup On Start **On Message** On Stop

```

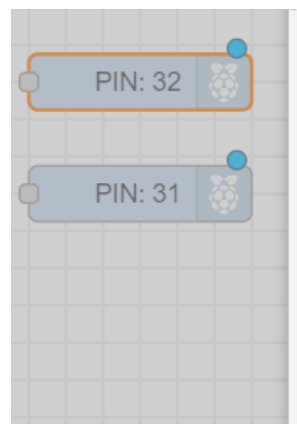
1  if (flow.get("switch_state") == "false") {
2    |   msg.payload = 0;
3    |   return msg;
4  }
5  else {
6    |   return msg;
7  }
8

```

Code nodejs – เลือก มา 2 node



ตั้งค่าตามนี้แล้ว Done



GPIO05 - 29	<input type="radio"/>	30 - Ground	<input type="radio"/>
GPIO06 - 31	<input checked="" type="radio"/>	32 - GPIO12	<input type="radio"/>
GPIO13 - 33	<input type="radio"/>	34 - Ground	<input type="radio"/>
GPIO19 - 35	<input type="radio"/>	36 - GPIO16	<input type="radio"/>
GPIO26 - 37	<input type="radio"/>	38 - GPIO20	<input type="radio"/>
Ground - 39	<input type="radio"/>	40 - GPIO21	<input type="radio"/>

BCM GPIO

12

Type

Digital output



GPIO05 - 29	<input type="radio"/>	30 - Ground	<input type="radio"/>
GPIO06 - 31	<input checked="" type="radio"/>	32 - GPIO12	<input type="radio"/>
GPIO13 - 33	<input type="radio"/>	34 - Ground	<input type="radio"/>
GPIO19 - 35	<input type="radio"/>	36 - GPIO16	<input type="radio"/>
GPIO26 - 37	<input type="radio"/>	38 - GPIO20	<input type="radio"/>
Ground - 39	<input type="radio"/>	40 - GPIO21	<input type="radio"/>

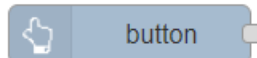
BCM GPIO

6

Type

PWM output

Code nodejs – เลือก



แล้วตั้งค่าตามนี้

## Edit button node

Delete

Cancel

Done

## Properties

Tab

LED

Name

Digital Switch

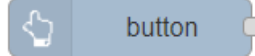
Payload

Topic

Group

Order 1

Code nodejs – เลือก



แล้วตั้งค่าตามนี้

Edit button node

Delete
Cancel
Done

Properties

Tab
LED

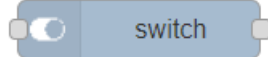
Name
OFF

Payload

Topic

Group
Order 1

Code nodejs – เลือก



แล้วตั้งค่าตามนี้

Edit switch node

Delete
Cancel
Done

Properties

Tab
LED

Name
Analog Switch

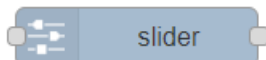
Topic

Group
Order 1

On Value
true

Off Value
false

Code nodejs – เลือก



แล้วตั้งค่าตามนี้

**Edit slider node**

Delete Cancel Done

**Properties**

Tab LED

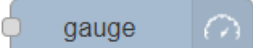
Name Slider

Topic

Group Order 1

Min 0

Max 10

Code nodejs – เลือก  แล้วตั้งค่าตามนี้

**Edit gauge node**

Delete Cancel Done

**Properties**

Tab LED

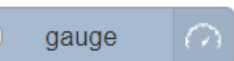
Name Gauge

Group Order 1

Template {{value}}

Min 0

Max 10

Code nodejs – เลือก  แล้วตั้งค่าตามนี้

Edit gauge node
Delete
Cancel
Done

Properties

Tab
LED

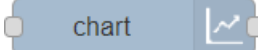
Name
LED status

Group
Order 1

Template
{{value}}

Min
0

Max
1

Code nodejs – เลือก  แล้วตั้งค่าตามนี้

Edit chart node
Delete
Cancel
Done

Properties

Tab
LED

Name

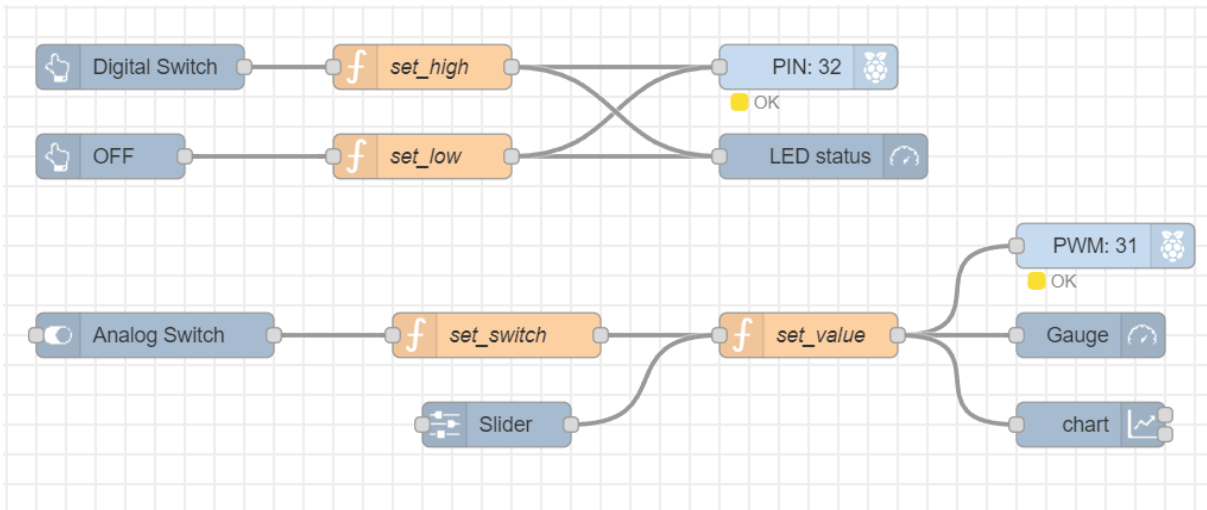
Group
Order 1

Old after
1
day(s)

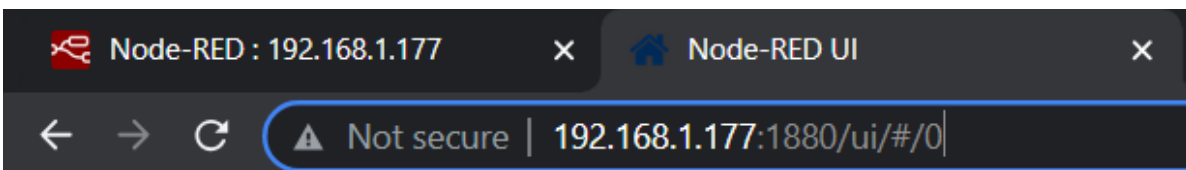
No data
No Data

Interpolate
linear

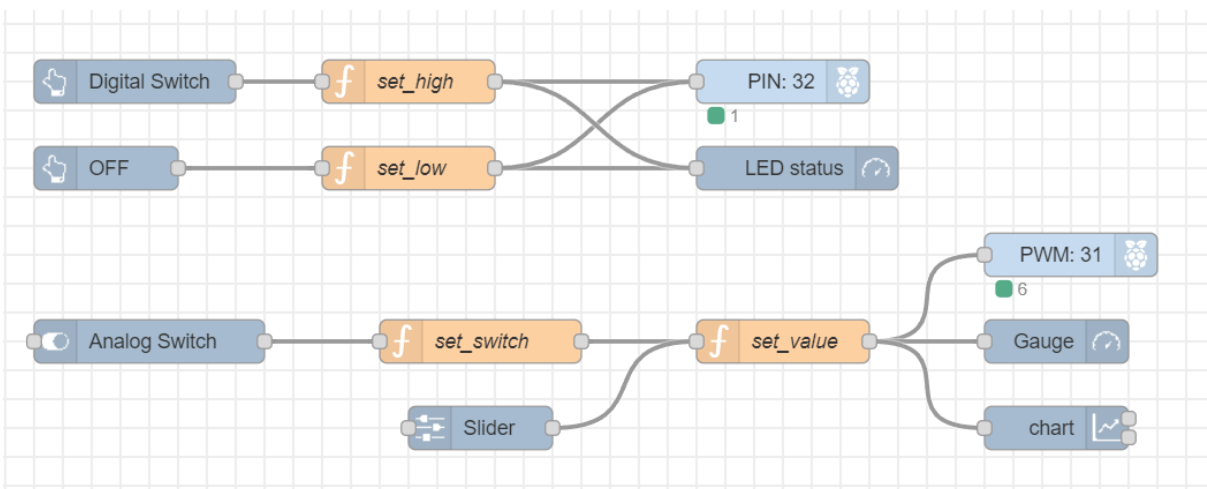
นำทุก node มาต่อรวมกันดังรูป



สามารถเข้าไปดูUI ได้โดยการเข้าไปที่ link ที่ใช้เข้า Node-Red แล้วตัวด้วย /ui

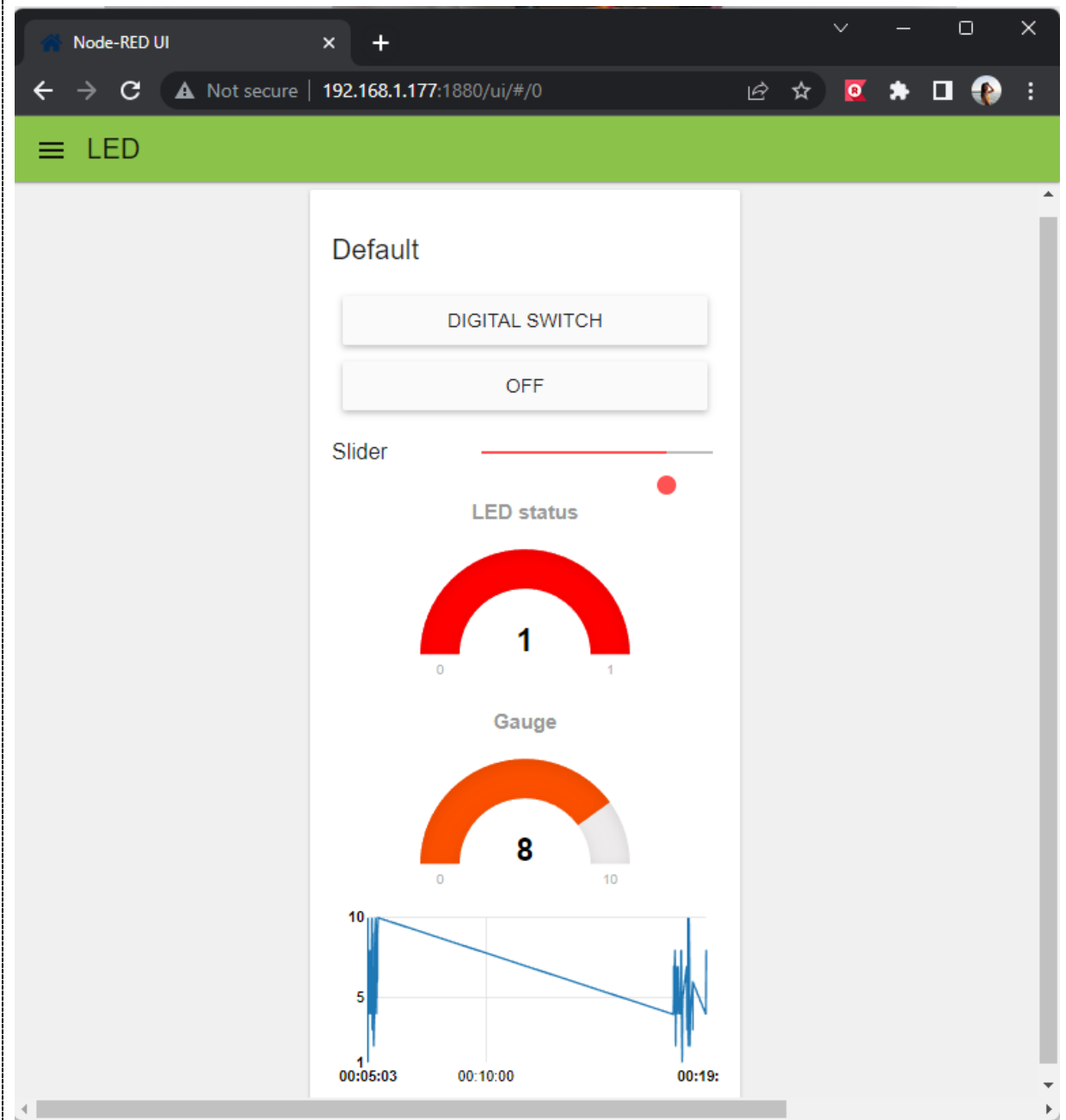


หน้าจอ node flow ที่ได้ทดสอบ

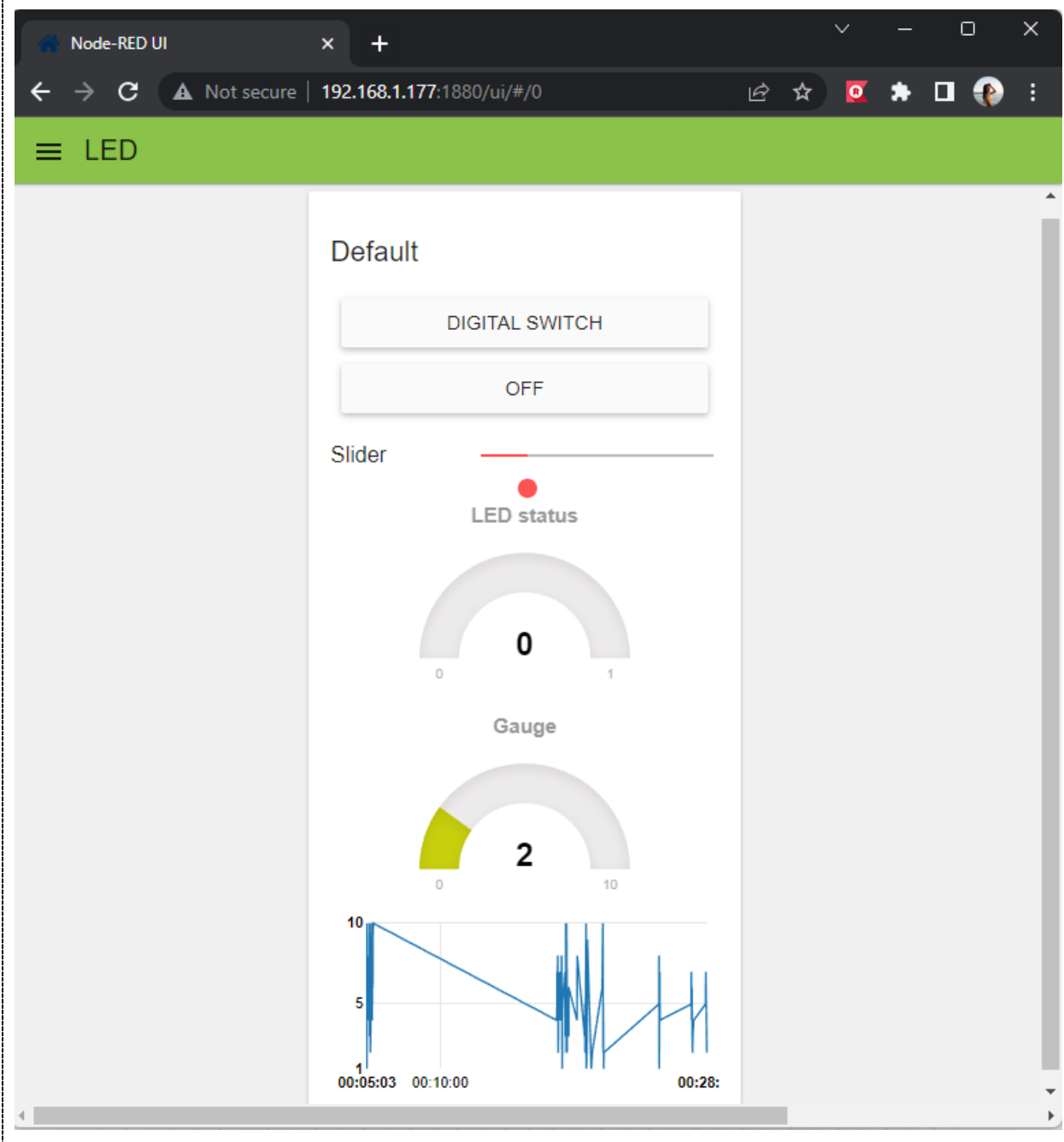




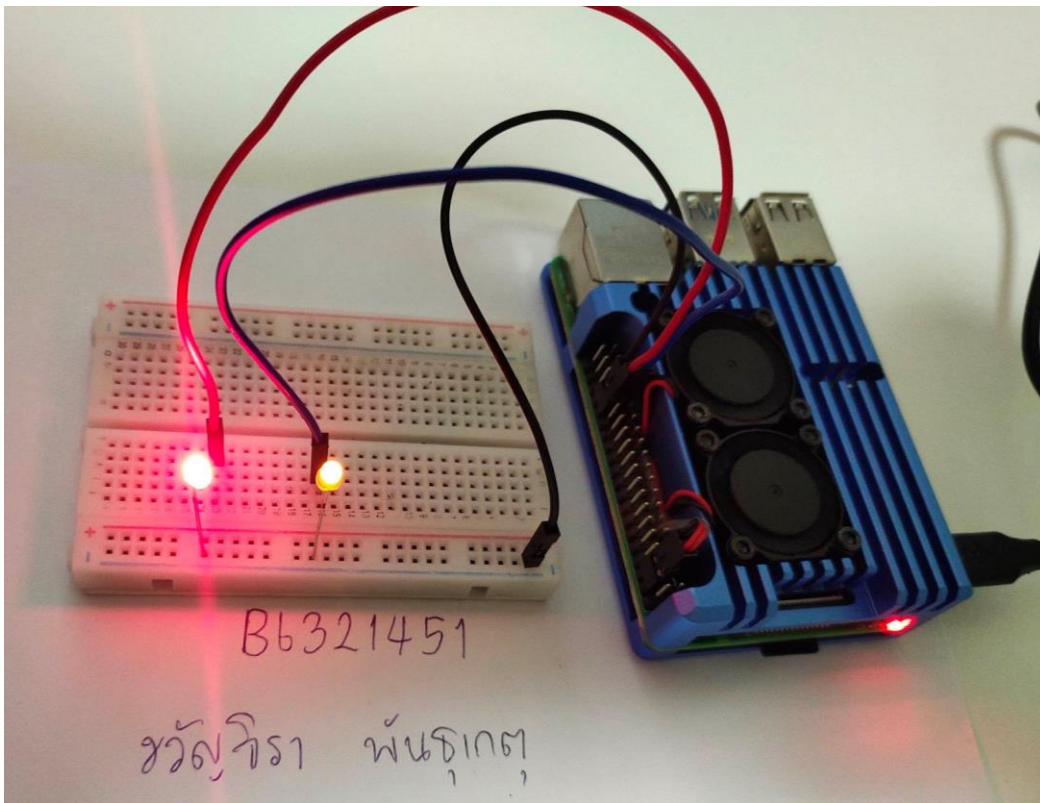
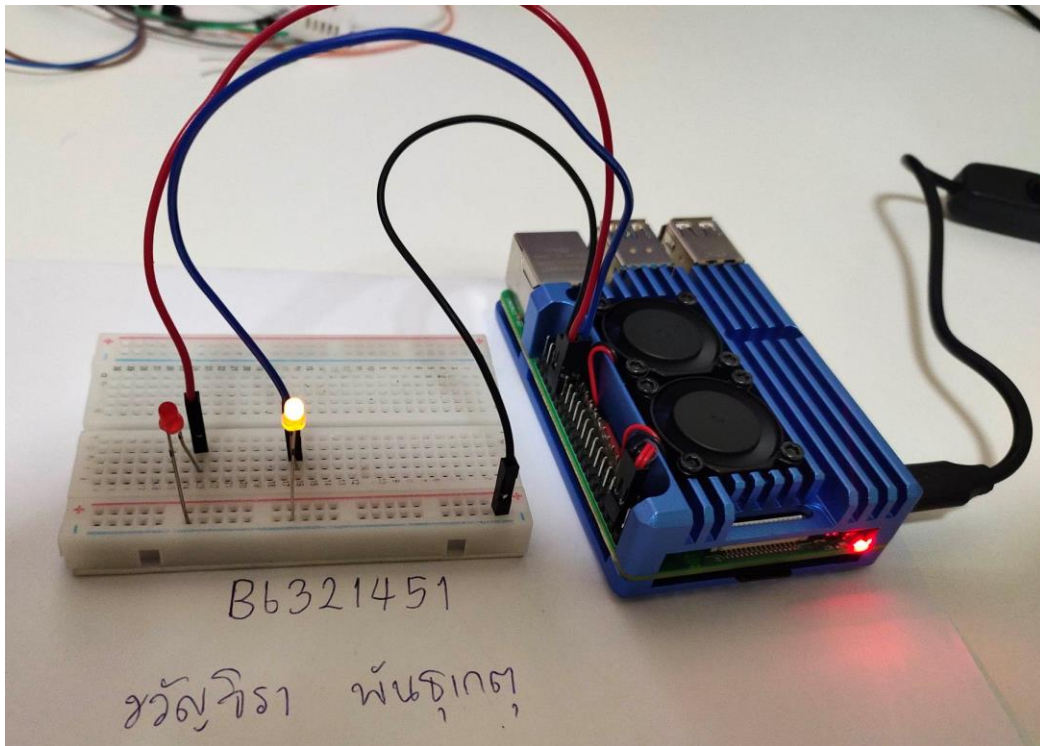
รูปการทดสอบ 1 – UI



## รูปการทดสอบ 2



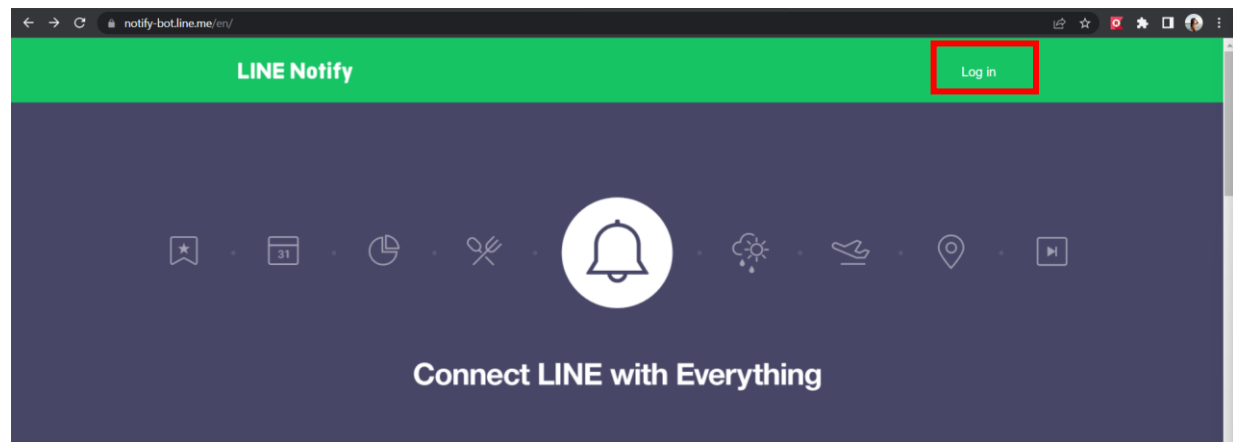
รูปวงจร



## Quiz\_202 – Node-RED to LINE notify

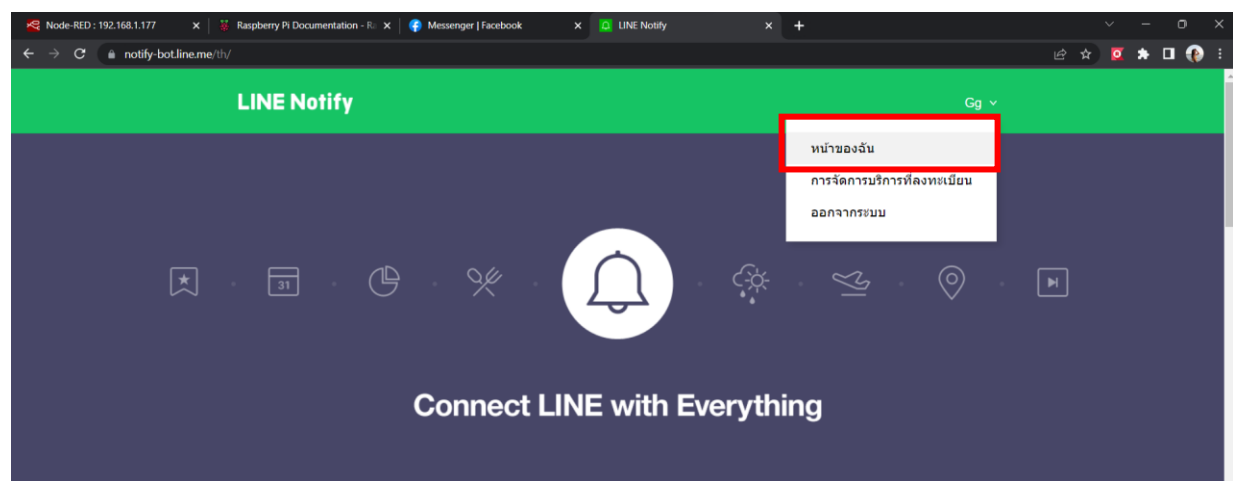
- แสดงข้อมูลที่ได้ทำการทดสอบ

โปรแกรมที่ใช้ทดสอบ – รับ token ของ LINE Notify ไปที่ <https://notify-bot.line.me/en/>



ทำการ Login ด้วยบัญชี LINE ที่ใช้งานอยู่

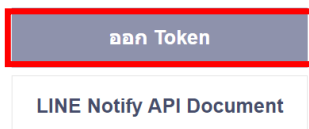
กดที่ชื่อบัญชี LINE แล้วกด My page



เลื่อนหน้า Web ลงมาด้านล่างสุด แล้วกด Generate token

### ออก Access Token (สำหรับผู้พัฒนา)

เมื่อใช้ Access Token แบบบุคคล จะสามารถตั้งค่าการแจ้งเตือนได้โดยไม่ต้องลงทะเบียนกับเว็บเซิร์ฟเวอร์



ตั้งชื่อเป็น Node-Red แล้วเลือก รับการแจ้งเตือนแบบตัวต่อตัวจาก LINE Notify (1-on1 chat with LINE Notify)  
จากนั้นกด ออก Token ( Generate token)

**ออก Token**
✕

โปรดใส่ชื่อ Token (จะแสดงเมื่อมีการแจ้งเตือน)

Node-Red

โปรดเลือกห้องแชทที่ต้องการส่งข้อความแจ้งเตือน

🔍 Search by group name

**รับการแจ้งเตือนแบบตัวต่อตัวจาก LINE Notify**

3/4/65 รูปภาพนายบุญทัน

4B

523201-641-4B

CPE 28#

CPE 28#

FNG

\* เมื่อเปิดเผย Personal Access Token จะทำให้บุคคลที่สามารถรับข้อมูลชื่อห้องแชทที่เชื่อมต่อและชื่อโปรไฟล์ได้

**ออก Token**

จากนั้นจะได้ Token

**ออก Access**

เมื่อใช้ Access Token แบบ...

**Token ที่ออก**


**xAhmrX2ySEBkqXqj97A3F1H7Gve067v87qVWjz**

ถ้าออกจากหน้านี้ ระบบจะไม่แสดง Token ที่ออกใหม่อีกต่อไป โปรดคัดลอก Token ก่อนออกจากหน้านี้

คัดลอก

ปิด

xAhmrX2ySEBkqXqj97A3F1H7Gve067v87qVWjz2gpkHx

Code nodejs – เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done

## Edit function node

Delete

Cancel

Done

## Properties

Name

function 1

Setup

On Start

On Message

On Stop

```

1  msg.token = "xAhmrX2y5EDkqXqI37A8PH7G.ec007v87qVWjZ2gpkHx";
2  msg.message = "Hello from Node-Red";
3  msg.stickerPackageId = 1;
4  msg.stickerId = 106;
5  msg.headers = {
6    "content-type": "application/x-www-form-urlencoded",
7    "Authorization": "Bearer " + msg.token
8  };
9  msg.payload = {
10    "message": msg.message,
11    "stickerPackageId": msg.stickerPackageId,
12    "stickerId": msg.stickerId
13  };
14  return msg;
15

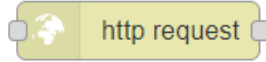
```

```

msg.token = "xAhmrX2y5EDkqXqI37A8PH7G.ec007v87qVWjZ2gpkHx";
msg.message = "Hello from Node-Red";
msg.stickerPackageId = 1;
msg.stickerId = 106;
msg.headers = {
  "content-type": "application/x-www-form-urlencoded",
  "Authorization": "Bearer " + msg.token
};
msg.payload = {
  "message": msg.message,
  "stickerPackageId": msg.stickerPackageId,
  "stickerId": msg.stickerId
};
return msg;

```

Code nodejs – เลือก



แล้วตั้งค่าตามนี้จากนั้นกด Done

Edit http request node

Delete Cancel Done

⚙ Properties

Method POST

URL <https://notify-api.line.me/api/notify>

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

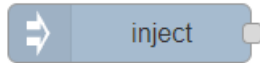
☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

Return a UTF-8 string

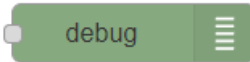
Headers

Code nodejs -- เลือก



มาโดยไม่ต้องตั้งค่าอะไรเพิ่มเติม

Code nodejs – เลือก



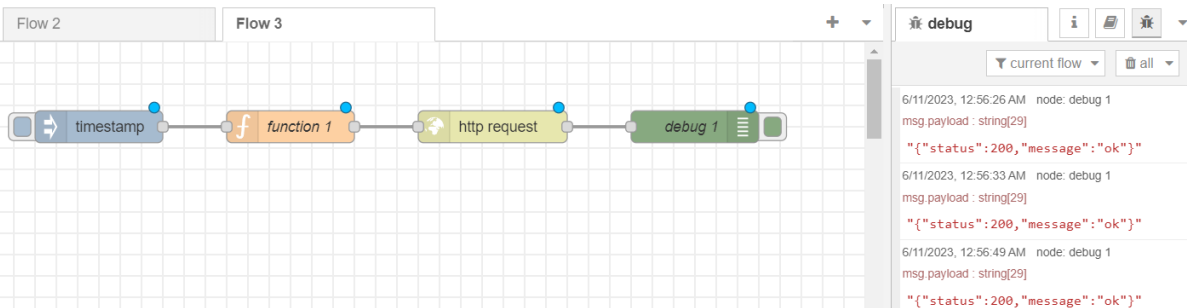
แล้วนำทุก node มาต่อตามนี้



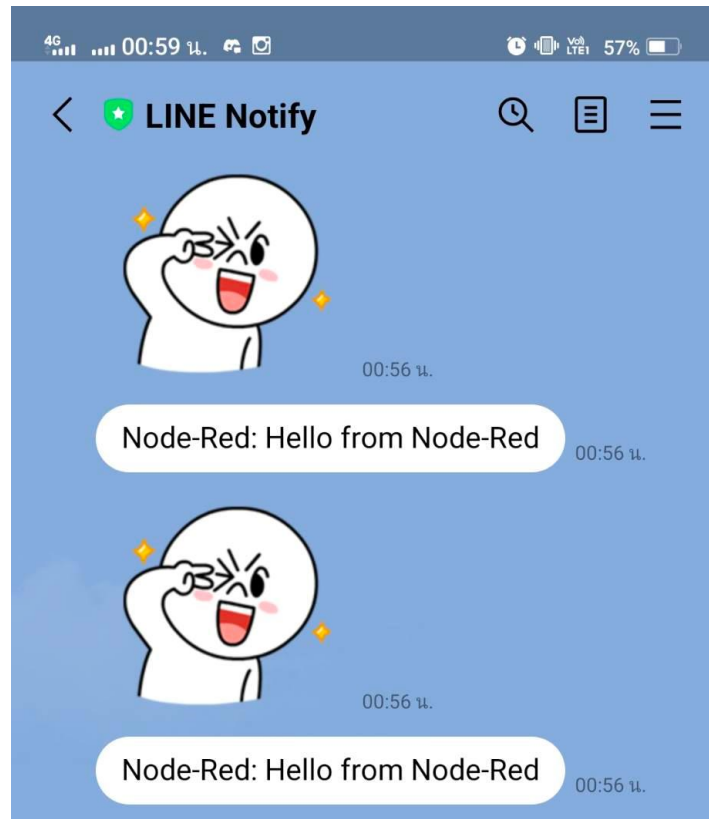
จากนั้นกด



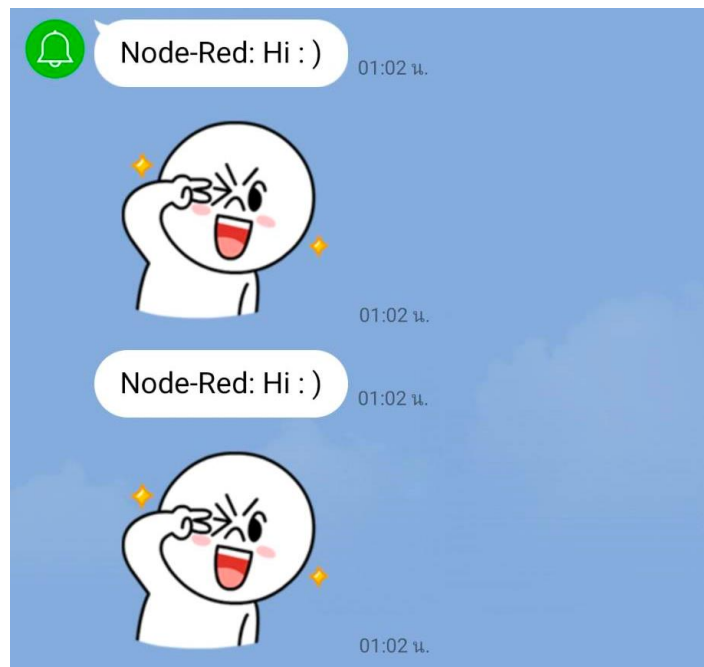
หน้าจอ node flow ที่ได้ทดสอบ - เมื่อกด timestamp จะส่งข้อความใน build\_message ไป



รูปการทดสอบ 1 – LINE



รูปการทดสอบ 2 --





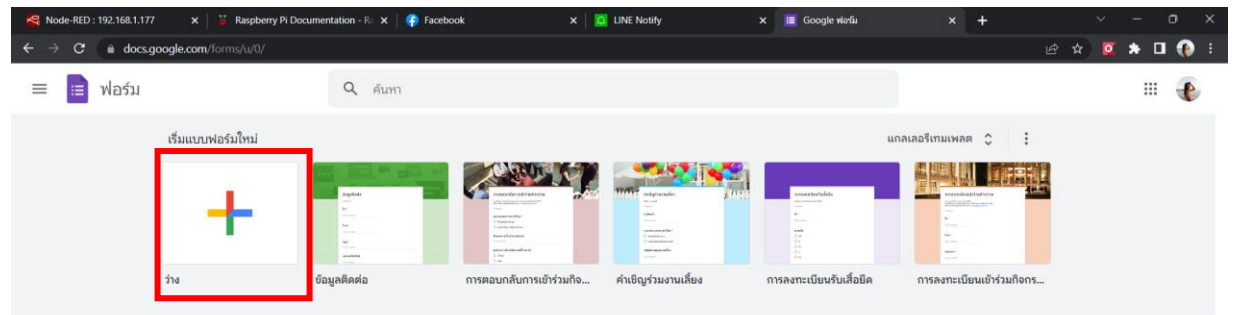
### Quiz\_203 – Node-RED to Google Sheet

- แสดงข้อมูลที่ได้ทำการทดสอบ

โปรแกรมที่ใช้ทดสอบ – รับ link Google Form ไปที่ <https://docs.google.com/forms/u/0/>

Login ด้วยบัญชี gmail ที่ใช้งานอยู่

กดที่ + เพื่อสร้าง Google Form



สร้าง Form ตามรูปแบบนี้

ที่ด้านบนขวาให้กดที่ มุมมอง เพื่อทดสอบกรอกข้อมูล



ทดสอบส่งข้อมูลเข้า Google Form 3 ถึง 4 รายการ

กลับไปหน้าจอสร้าง Google Form ไปที่ การตอบกลับ

เลือกสร้างสเปรดชีตใหม่ แล้วใส่ชื่อเดียวกับชื่อของ Google Form ที่สร้าง และกด สร้าง

**Data Access Point** ☆ 📁 ☁

ไฟล์ แก้ไข ดู แทรก รูปแบบ ข้อมูล เครื่องมือ ส่วนขยาย ความช่วยเหลือ

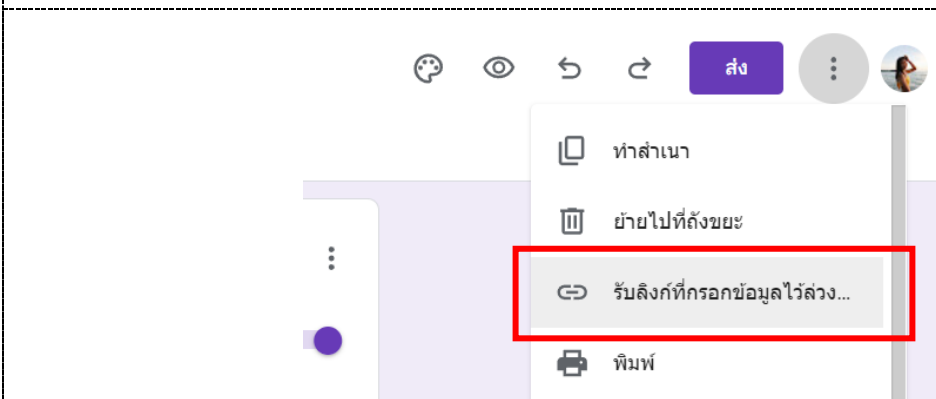
↶ ↷ 🖨 📄 100% ▾ | B % .0 .00 123 | คำเริ่มต้น... ▾ | - 10 + | B I 🔗 A 📎 🏠 📄 ▾ | ≡

A1 ▾ | 📄 ประทับเวลา

	A	B	C	D	E	
1	ประทับเวลา	Temperature(C)	Humidity(%)	Light(%)	RSSI(dB)	
2	11/6/2023, 2:06:21	35	20	10	55	
3	11/6/2023, 2:06:41	40	11	55	33	
4	11/6/2023, 2:10:35	32	15	49	53	
5						

กลับไปหน้าสร้าง Google Form

ที่ด้านบนขวาให้กดที่ -> รับลิงก์ที่กรอกข้อมูลไว้ล่วงหน้า



กรอกข้อมูลตามนี้จากนั้นกด รับลิงก์

เมื่อกด รับลิงก์ จะมีกล่องข้อความขึ้นมาที่ด้านซ้ายล่าง ให้กด คัดลอกลิงก์

แชร์ลิงก์นี้เพื่อรวมคำตอบที่กรอกไว้ล่วงหน้า

คัดลอกลิงก์

นำลิงก์ที่คัดลอกมาจะมีรูปแบบดังนี้ โดยตัวหนังสือสีแดง คือส่วนที่ต้องคัดลอกเก็บไว้

1	<a href="https://docs.google.com/forms/d/e/">https://docs.google.com/forms/d/e/</a>
2 access token	1FAIpQLS <u>gLFmLh-ONE_OPEQF18U1XvtdzgCDF1</u> ,33G6s3I7UUNwQ
3	/viewform?usp=pp_url
4	&entry.1117552749=aaaaaa
5	&entry.139512928=bbbbbb
6	&entry.1414442786=ccccc
7	&entry.993283756=dddddd

นำลิงก์ที่คัดลอกมาปรับรูปแบบดังนี้แทนที่ส่วนสีแดงด้วยส่วนที่คัดลอกมาจากลิงก์ก่อนหน้า

1	<a href="https://docs.google.com/forms/u/0/d/e/">https://docs.google.com/forms/u/0/d/e/</a>
2 access token	1FAIpQLS <u>gLFmLh-ONE_OPEQF18U1XvtdzgCDF1</u> ,33G6s3I7UUNwQ
3	/formResponse?
4	&entry.1117552749={{payload.Temperature}}
5	&entry.139512928={{payload.Humidity}}
6	&entry.1414442786={{payload.Light}}
7	&entry.993283756={{payload.RSSI}}

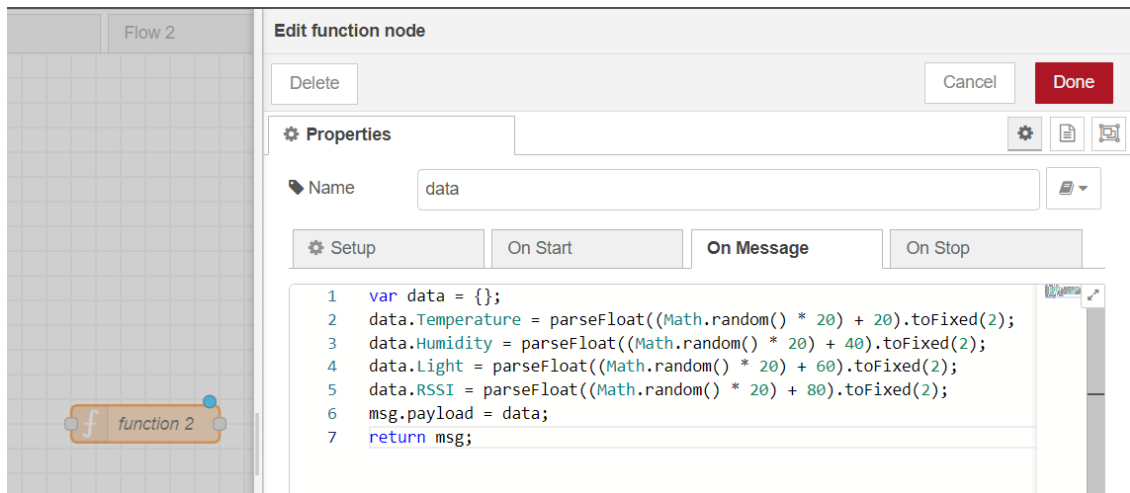
คัดลอกลิงก์ที่ปรับรูปแบบแล้วไว้เพื่อนำไปใช้งานต่อกับ node http request

Code nodejs -- เลือก



function

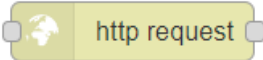
แล้วตั้งค่าตามนี้จากนั้นกด Done



```
var data = {};
data.Temperature = parseFloat((Math.random() * 20) + 20).toFixed(2);
data.Humidity = parseFloat((Math.random() * 20) + 40).toFixed(2);
data.Light = parseFloat((Math.random() * 20) + 60).toFixed(2);
data.RSSI = parseFloat((Math.random() * 20) + 80).toFixed(2);
msg.payload = data;
return msg;
```

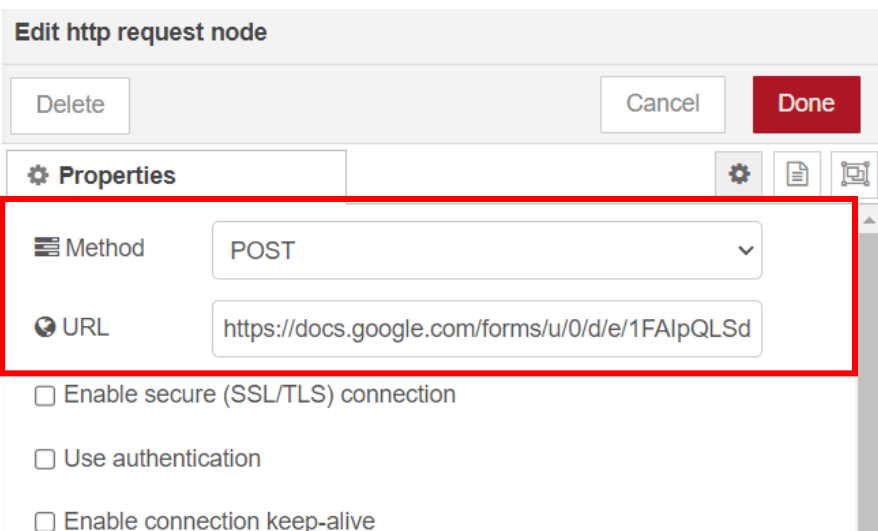
ใช้เป็น generator ในสร้างข้อมูล สำหรับส่งให้ google sheet

Code nodejs -- เลือก



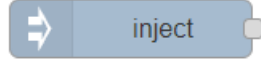
http request

แล้วตั้งค่าตามนี้จากนั้นกด Done



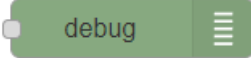
URL: ลิงก์ที่ปรับรูปแบบแล้ว

Code nodejs -- เลือก



มาโดยไม่ต้องตั้งค่าอะไรเพิ่มเติม

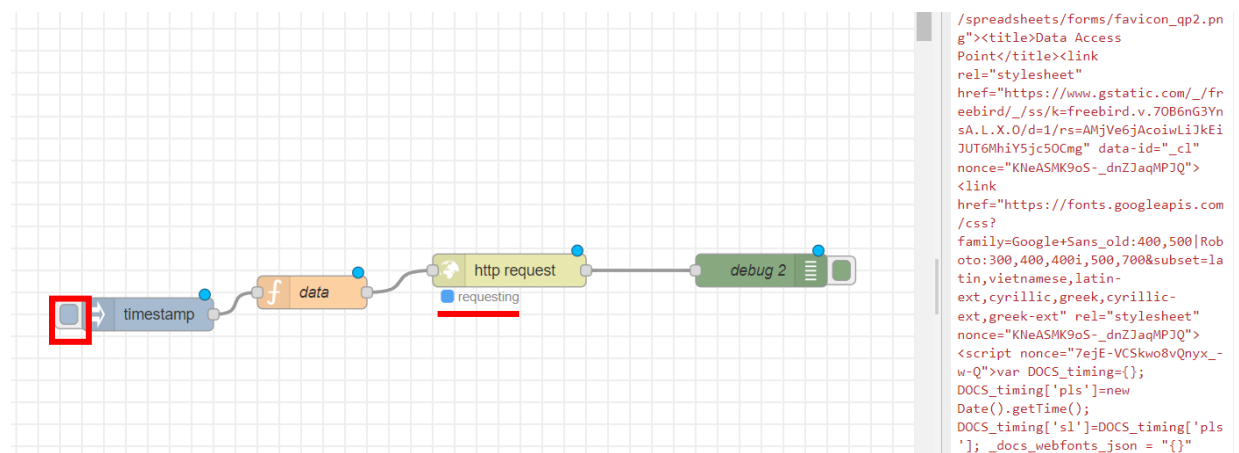
Code nodejs -- <sup>4</sup>เลือก



ให้นำทุก node มาต่อตามนี้




หน้าจอ node flow ที่ได้ทดสอบ – เมื่อกดส่งข้อมูลไป Google Sheet



ถ้าส่งได้สำเร็จจะเห็นชื่อของ Google Form ใน tag <title> ที่ตอบกลับมาจากการส่งข้อมูล

## รูปการทดสอบ 1 – Google Sheet



Data Access Point

☆

📁

☁

ไฟล์ แก้ไข ดู แทรก รูปแบบ ข้อมูล เครื่องมือ ส่วนขยาย ความช่วยเหลือ

↶ ↷ 🖨 📄 100% ▾

B

%

0.00 123

ค่าเริ่มต้น...

10

+

B

I

↕

A

🔍

📐

A1

🔍

🕒 ประทับเวลา

	A	B	C	D	E
1	ประทับเวลา	Temperature(C)	Humidity(%)	Light(%)	RSSI(dB)
2	11/6/2023, 2:06:21	35	20	10	55
3	11/6/2023, 2:06:41	40	11	55	33
4	11/6/2023, 2:10:35	32	15	49	53
5	11/6/2023, 2:42:12	27	47.6	74.13	81.51
6	11/6/2023, 2:45:45	31.12	47.19	62.04	80.58
7	11/6/2023, 2:45:47	34.99	53.06	74.93	89.4
8	11/6/2023, 2:45:53	21.32	53.4	77.43	87.85
9	11/6/2023, 2:45:53	22.49	40.25	66.78	98.65
10					

รูปการทดสอบ 2 – Google Form ที่ tab ข้อมูลสรุป

คำถาม
การตอบกลับ 8
การตั้งค่า

คำตอบ 8 ข้อ

ดูในชีต

เปิดรับคำตอบ

ข้อมูลสรุป

คำถาม

แยกรายการ

Temperature(C)

คำตอบ 8 ข้อ

35

40

32

27.00

31.12

34.99

21.32

22.49

### Humidity(%)

ค่าตอบ 8 ข้อ

20

11

15

47.60

47.19

53.06

53.40

40.25

### Light(%)

ค่าตอบ 8 ข้อ

10

55

49

74.13

62.04

74.93

77.43

66.78



RSSI(dB)

ค่าตอบ 8 ข้อ

55

33

53

81.51

80.58

89.40

87.85

98.65