

## การสร้าง MQTT Server บน Raspberry Pi เพื่อใช้งาน Chatbot LINE ในฟาร์มอัจฉริยะ

### Chatbot LINE from Raspberry Pi MQTT Server for Smart Farming

ชื่อ-สกุล : นางสาวขวัญจิรา พันธุ์เกตุ

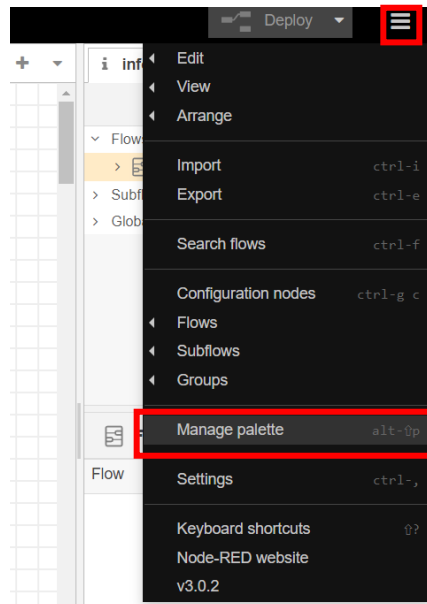
รหัสนักศึกษา : B6321451

6/6 - คำถามท้ายบทเพื่อทดสอบความเข้าใจ

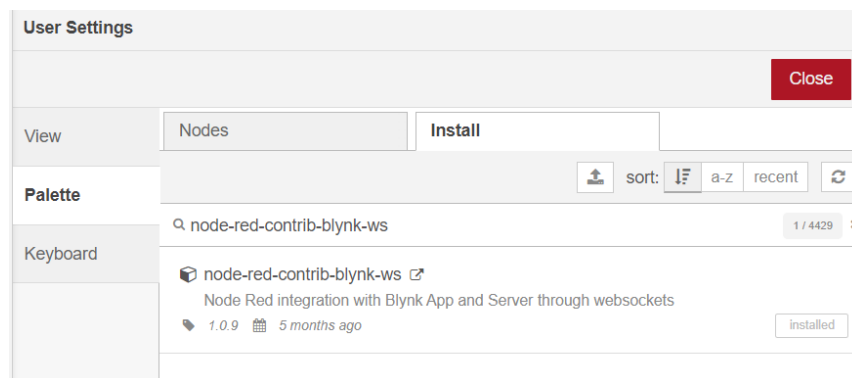
#### Quiz\_301 – Blynk with Node-RED

- ทดสอบควบคุม 4 LED ด้วย Raspberry Pi ผ่าน Node-RED
- เพิ่มเติมให้ส่งข้อมูล Temperature ไปแสดงที่ Gauge บน Blynk

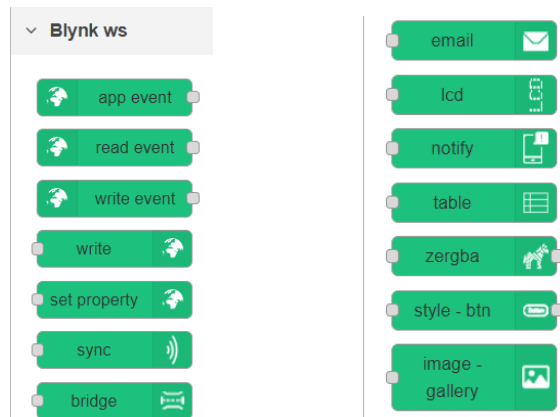
โปรแกรมที่ใช้ทดสอบ – ติดตั้ง palette สำหรับ DHT22 กดไปที่ ด้านบนขวา กดไปที่ Manage palette



ไปที่ Install -> ค้นหา node-red-contrib-blynk-ws แล้วกดติดตั้ง



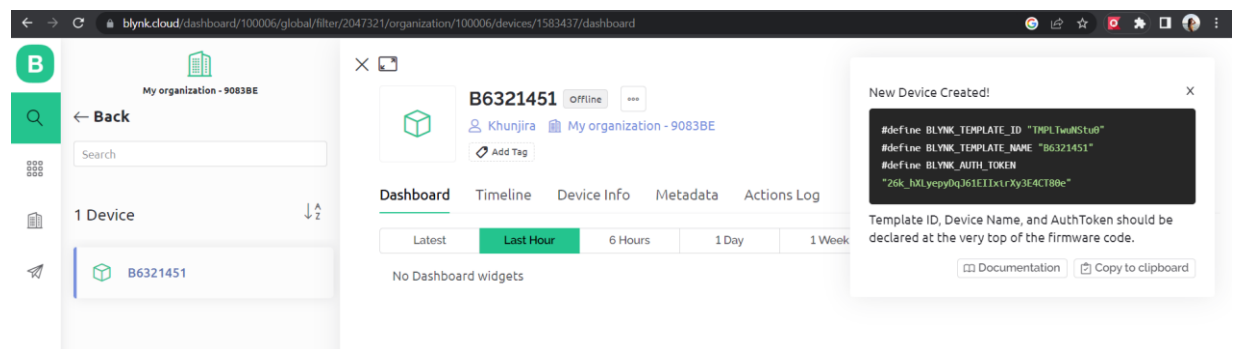
เมื่อติดตั้งเสร็จที่ tab ด้านซ้ายจะได้ node ใหม่เพิ่มเข้ามา



โปรแกรมที่ใช้ทดสอบ – ติดตั้ง Blynk บนโทรศัพท์ได้จาก Google Play

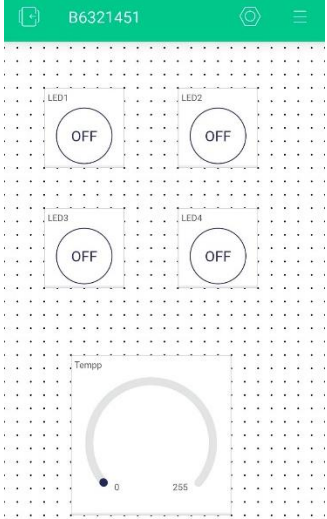


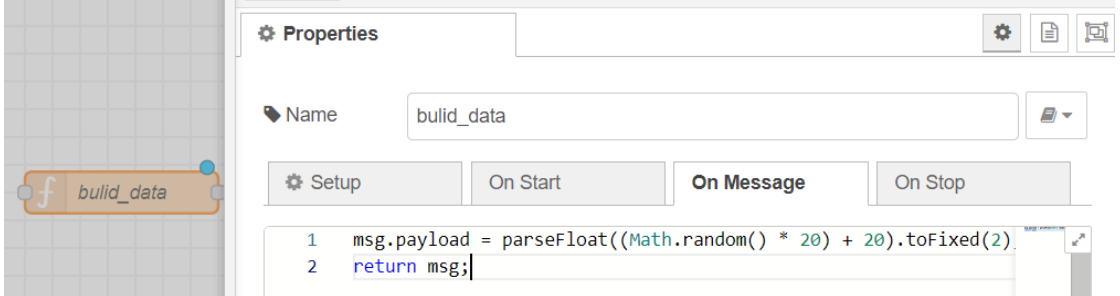

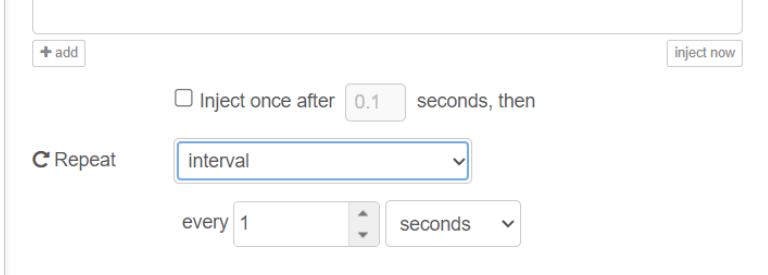


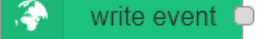
ไปที่ เว็บไซต์ของ Blynk -> เข้าสู่ระบบ -> สร้าง Devices -> copy Auth Token เก็บไว้ใช้กับ node ของ Blynk ใน Node-Red ต่อไป




สร้าง dashboard บน Blynk โดยใช้ widget ต่อไปนี้

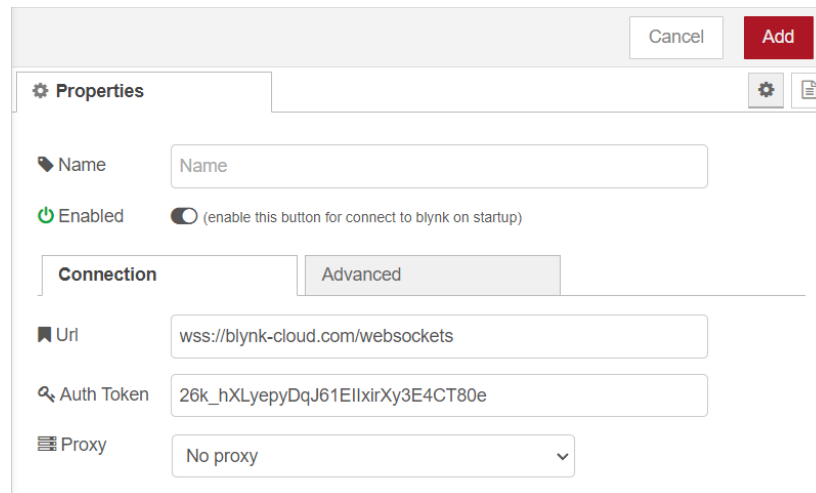


<p>Button x 4 , Gauge x 1</p>	<p>กดย้อนกลับ จะได้หน้าตาแบบนี้</p>
	
<p>Node-RED Code -- เลือก  <b>function</b> แล้วตั้งค่าตามนี้จากนั้นกด Done</p>	
	
<pre>msg.payload = parseFloat((Math.random() * 20) + 20).toFixed(2); return msg;</pre>	
<p>ใช้เป็น generator ในสร้างข้อมูล สำหรับส่งให้ Blynk</p>	
<p>Node-RED Flow -- เลือก  <b>inject</b> แล้วตั้งค่าตามนี้จากนั้นกด Done</p>	
	

Node-RED Flow -- เลือก  แล้วตั้งค่าตามนี้จากนั้นกด Done

เลือก Connection ให้เป็น Add new blynk-ws-client จากนั้นกด 

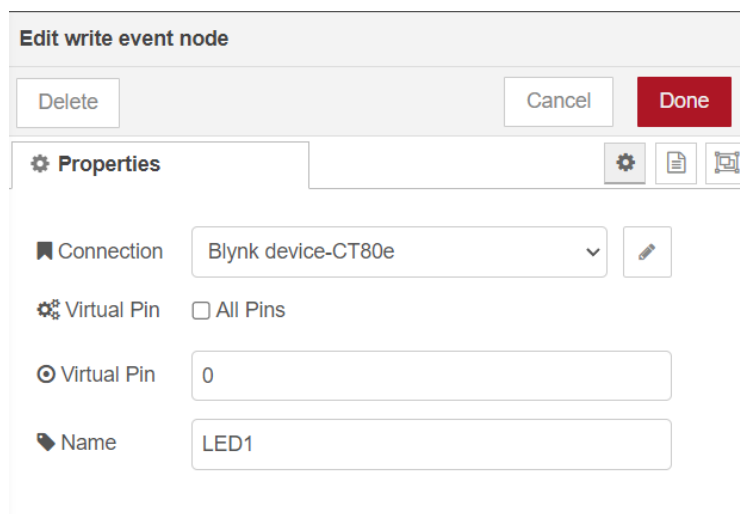
ตั้งค่าตามนี้จากนั้นกด Add

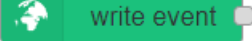


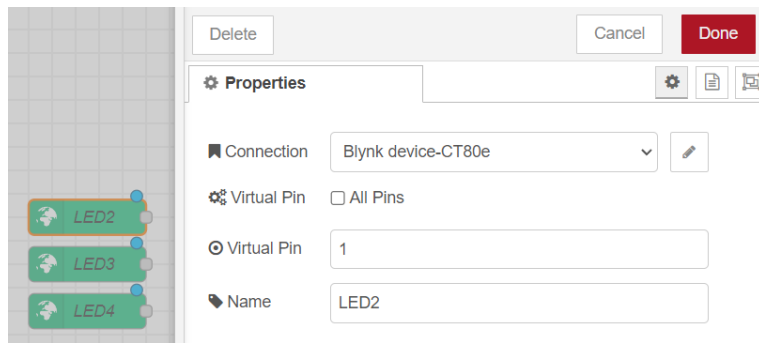
Url: wss://blynk-cloud.com/websockets

Auth Token: token ที่ได้มาตอนสร้าง project ใหม่ใน blynk

หลังจากกด Add จะกลับมาหน้านี้ ให้ตั้งค่าตามนี้



Node-RED Flow -- เลือก  มา 3 node ตั้งค่าตามนี้แล้วกด Done



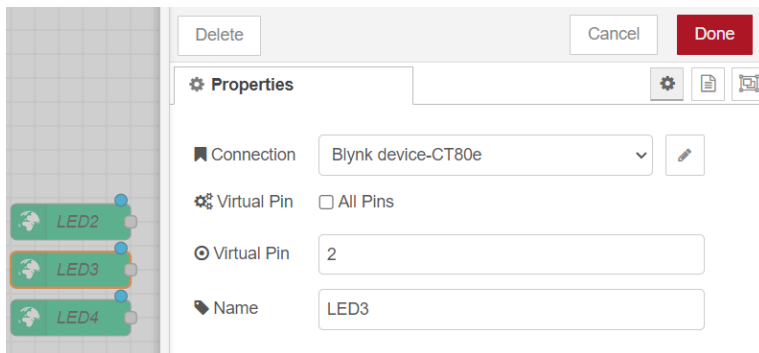
Properties

Connection: Blynk device-CT80e

Virtual Pin: ☐ All Pins

Virtual Pin: 1

Name: LED2



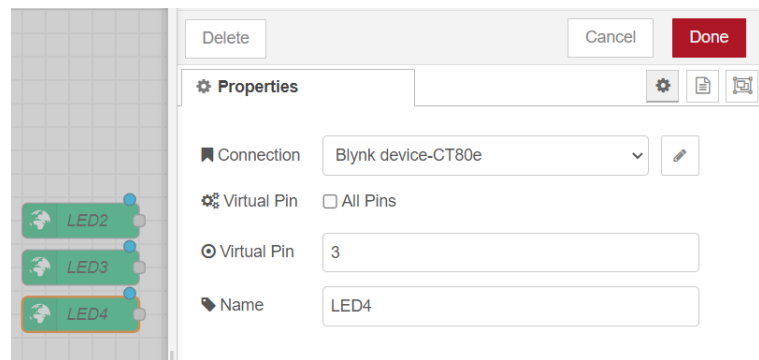
Properties

Connection: Blynk device-CT80e

Virtual Pin: ☐ All Pins

Virtual Pin: 2

Name: LED3



Properties

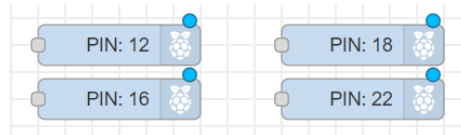
Connection: Blynk device-CT80e

Virtual Pin: ☐ All Pins

Virtual Pin: 3

Name: LED4

Node-RED Flow -- เลือก  มา 4 node ตั้งค่าตามนี้แล้วกด Done



PIN: 12

PIN: 16

PIN: 18

PIN: 22

Type: Digital output

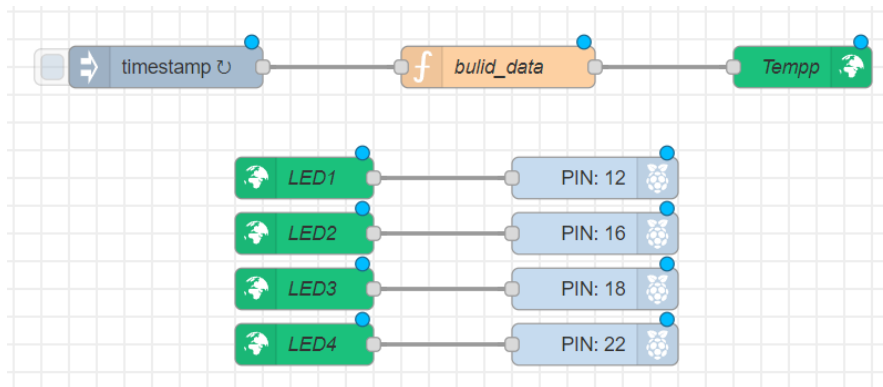
Node-RED Flow -- เลือก **write** ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

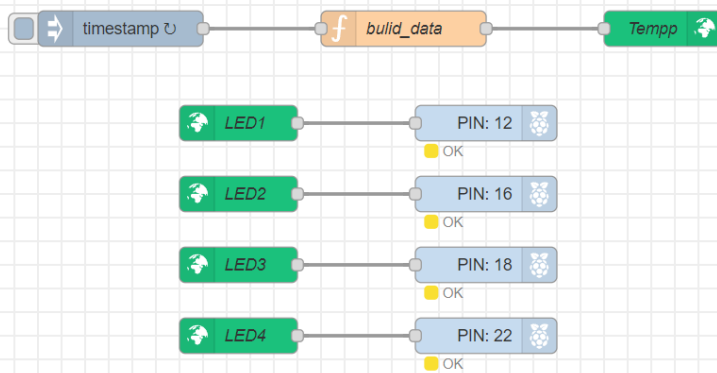
\* Properties

Connection Blynk device-CT80e
Pin Mode Fixed
Virtual Pin 4
Name Tempp

นำทุก node มาต่อรวมกันดังรูป



ติดปัญหา



```

Warning: You are trying to connect
to the Blynk Cloud Server (blynk-
cloud.com) which has not been shut
down permanently as of year end 2022"

6/11/2023, 3:23:07 PM node: M3D3
msg: string[139]

Warning: You are trying to connect
to the Blynk Cloud Server (blynk-
cloud.com) which has not been shut
down permanently as of year end 2022"

6/11/2023, 3:23:07 PM node: 8f8b4ffe66b6c83
msg: string[54]

"Websocket Error: getaddrinfo
ENOTFOUND blynk-cloud.com"

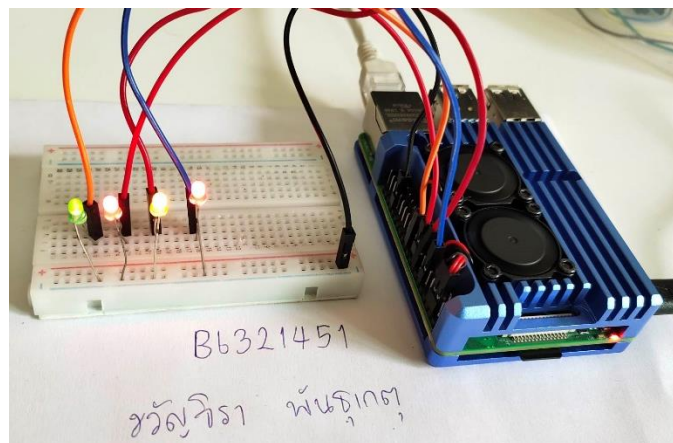
6/11/2023, 3:23:07 PM node: M3D3
msg: string[54]

"Websocket Error: getaddrinfo
ENOTFOUND blynk-cloud.com"
    
```

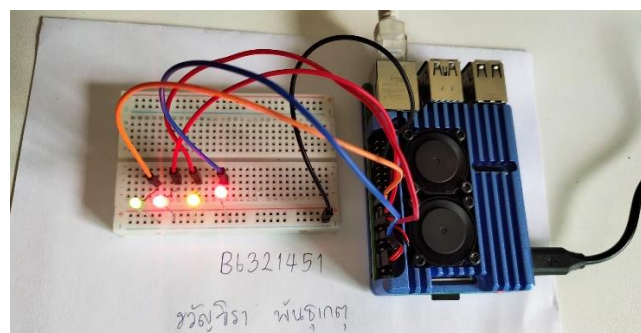
หน้าจอ Blynk



รูปการทดสอบ 1




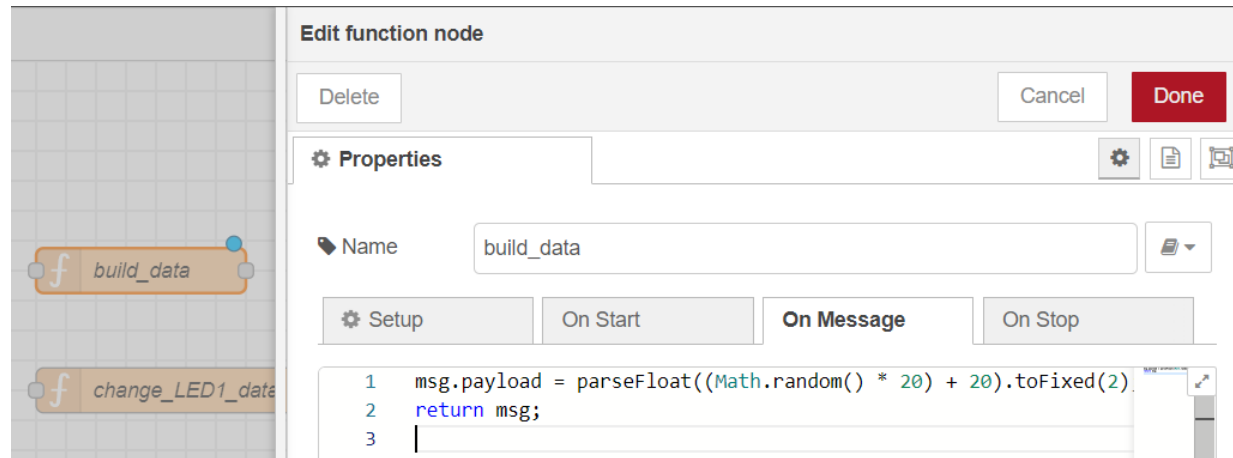
รูปการทดสอบ 2



**Quiz\_302 – MQTT Client with Node-RED**

- ทดสอบควบคุม 4 LED ด้วย Raspberry Pi ผ่าน Node-RED
- เพิ่มเติมให้ส่งข้อมูล Temperature ไปแสดงที่ บน Server

Node-RED Code -- เลือก  function แล้วตั้งค่านี้นี้จากนั้นกด Done



**Edit function node**

Delete Cancel Done

**Properties**

Name: build\_data

Setup On Start **On Message** On Stop

```


1 msg.payload = parseFloat((Math.random() * 20) + 20).toFixed(2);
2 return msg;
3

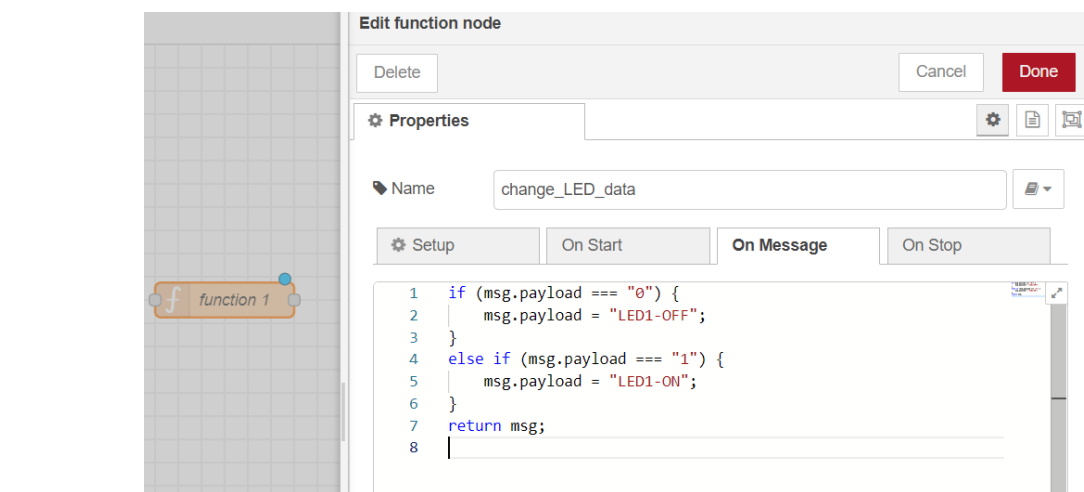
```

```

msg.payload = parseFloat((Math.random() * 20) + 20).toFixed(2);
return msg;

```

Node-RED Code -- เลือก  function แล้วตั้งค่านี้นี้จากนั้นกด Done



**Edit function node**

Delete Cancel Done

**Properties**

Name: change\_LED\_data

Setup On Start On Message **On Stop**

```

1 if (msg.payload === "0") {
2   msg.payload = "LED1-OFF";
3 }
4 else if (msg.payload === "1") {
5   msg.payload = "LED1-ON";
6 }
7 return msg;
8


```

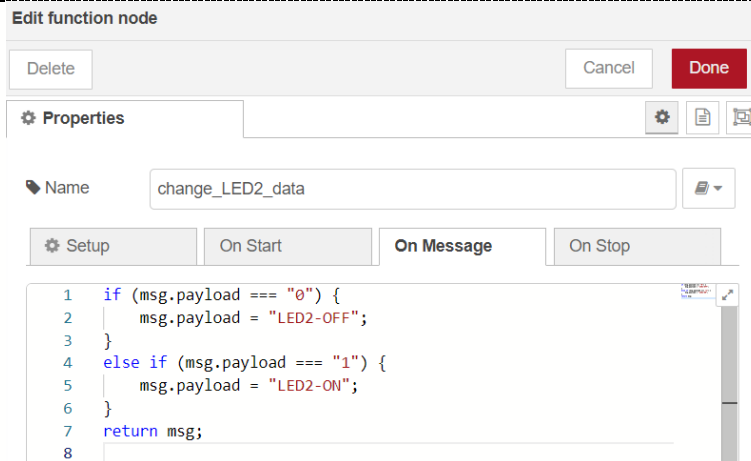
```

if (msg.payload === "0") {
  msg.payload = "LED1-OFF";
}
else if (msg.payload === "1") {
  msg.payload = "LED1-ON";
}
return msg;

```





Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done

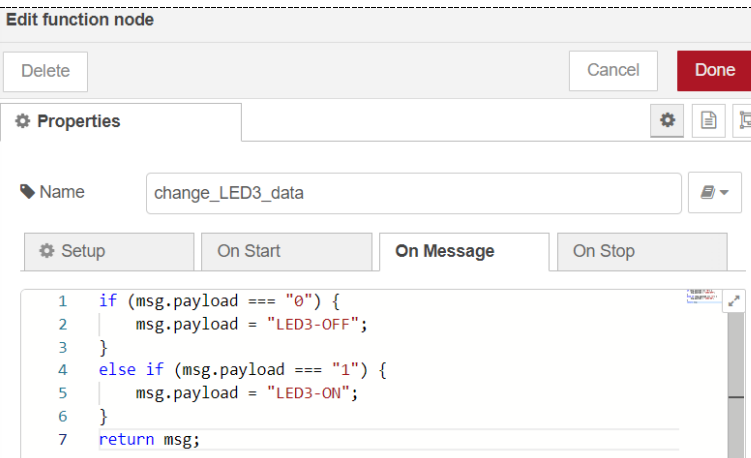


```

if (msg.payload === "0") {
    msg.payload = "LED2-OFF";
}
else if (msg.payload === "1") {
    msg.payload = "LED2-ON";
}
return msg;

```



Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done

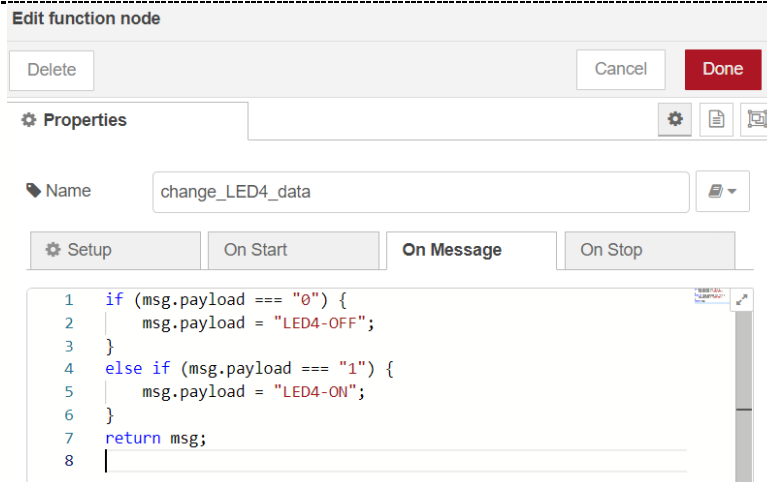


```



if (msg.payload === "0") {
    msg.payload = "LED3-OFF";
}
else if (msg.payload === "1") {
    msg.payload = "LED3-ON";
}
return msg;

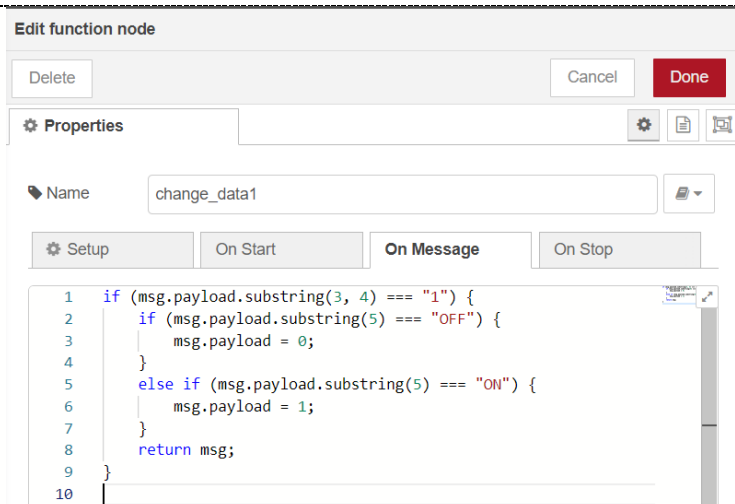
```

Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done



```
if (msg.payload === "0") {
    msg.payload = "LED4-OFF";
}
else if (msg.payload === "1") {
    msg.payload = "LED4-ON";
}
return msg;
```

Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done




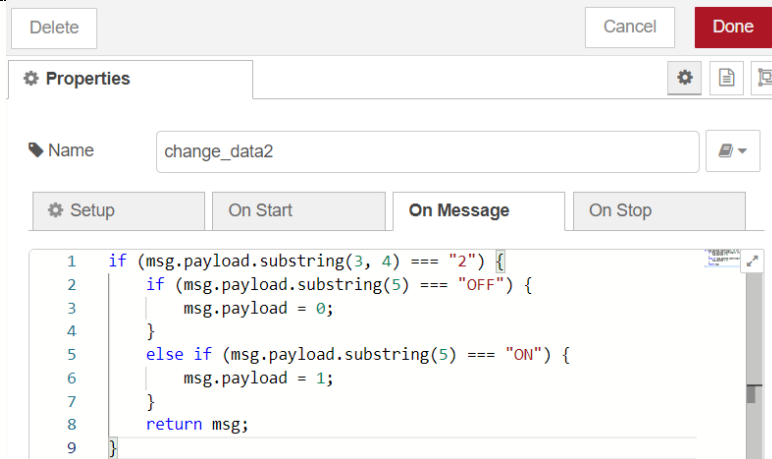
```
if (msg.payload.substring(3, 4) === "1") {
    if (msg.payload.substring(5) === "OFF") {
        msg.payload = 0;
    }
    else if (msg.payload.substring(5) === "ON") {
        msg.payload = 1;
    }
}
```

```

    }
    return msg;
}

```


Node-RED Code -- เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done

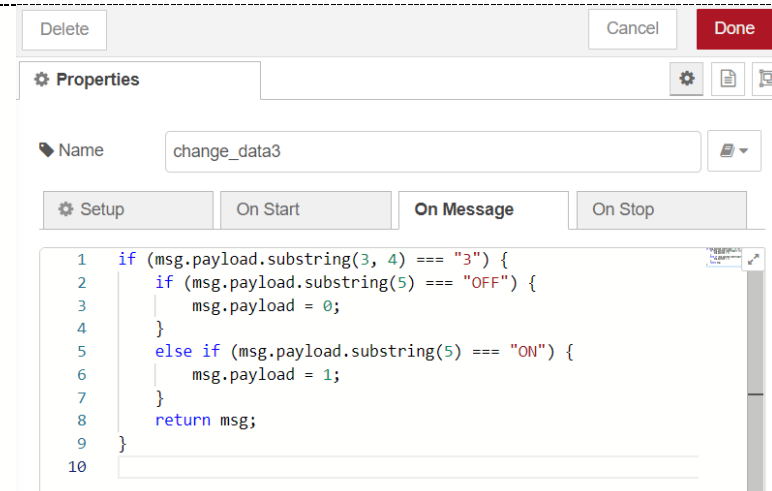


```

if (msg.payload.substring(3, 4) === "2") {
    if (msg.payload.substring(5) === "OFF") {
        msg.payload = 0;
    }
    else if (msg.payload.substring(5) === "ON") {
        msg.payload = 1;
    }
    return msg;
}

```

Node-RED Code -- เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done



```


if (msg.payload.substring(3, 4) === "3") {
    if (msg.payload.substring(5) === "OFF") {

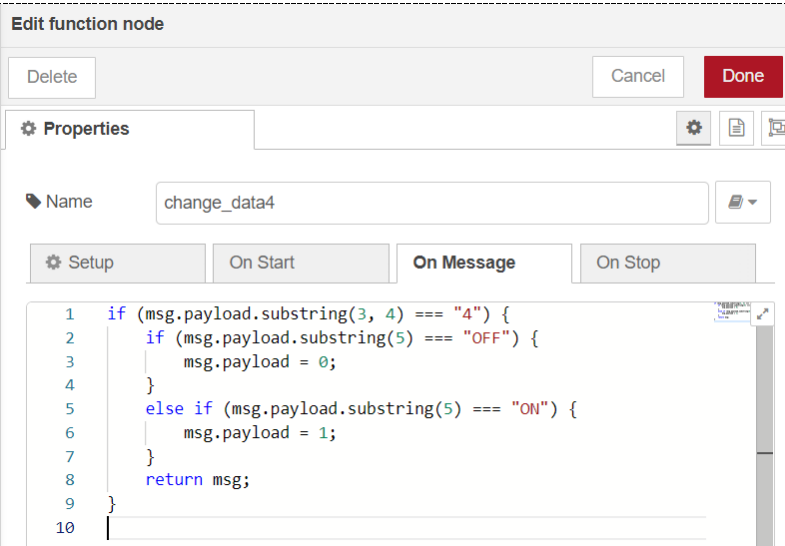
```

```

    msg.payload = 0;
  }
  else if (msg.payload.substring(5) === "ON") {
    msg.payload = 1;
  }
  return msg;
}

```


Node-RED Code -- เลือก  **function** แล้วตั้งค่าตามนี้จากนั้นกด Done

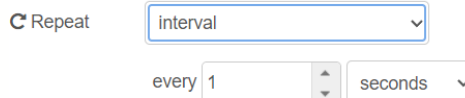


```

if (msg.payload.substring(3, 4) === "4") {
  if (msg.payload.substring(5) === "OFF") {
    msg.payload = 0;
  }
  else if (msg.payload.substring(5) === "ON") {
    msg.payload = 1;
  }
  return msg;
}

```

Node-RED Code -- เลือก  **inject** แล้วตั้งค่าตามนี้จากนั้นกด Done



Node-RED Flow -- เลือก  **rpi - gpio out** มา 4 node ตั้งค่าตามนี้แล้วกด Done

PIN: 12

PIN: 18

PIN: 16

PIN: 22

Type

Digital output

Node-RED Flow -- เลือก

mqtt out

ตั้งค่าตามนี้แล้วกด Done

⚙ Properties

⚙
📄
🖨

🌐 Server

Add new mqtt-broker...

✎

Edit mqtt out node > Edit mqtt-broker node

Delete
Cancel
Update

⚙ Properties

⚙
📄

📁 Name

m3d3

Connection
Security
Messages

🌐 Server

test.mosquitto.org

Port

1883

☒ Connect automatically  
☐ Use TLS

⚙ Protocol

MQTT V3.1.1

📁 Client ID

mqtx\_d16c9e6c

💓 Keep Alive

60

📁 Session

☒ Use clean session

☐ Enabled
📄 4 nodes use this config
On all flows

Edit mqtt out node

Delete
Cancel
Done

⚙ Properties

⚙
📄
🖨

🌐 Server

m3d3

✎

📁 Topic

temp

📁 QoS

▼

🔄 Retain

▼

📁 Name

Tempp-Data-to-MQTT

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit mqtt out node**

Delete Cancel Done

**Properties**

Server m3d3

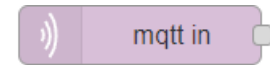
Topic led

QoS 0 Retain true

Name LED-Data-to-MQTT

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server m3d3

Action Subscribe to single topic

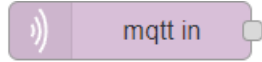
Topic temp

QoS 0

Output auto-detect (parsed JSON object, string or buffer)

Name Tempp-Data-from-MQTT

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server m3d3

Action Subscribe to single topic

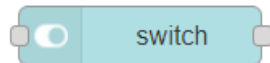
Topic led

QoS 0

Output auto-detect (parsed JSON object, string or buffer)

Name LED-Data-from-MQTT

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit switch node**

Delete Cancel Done

**Properties**

Group [MQTT Server] Status1

Size auto

Label LED1

Tooltip optional tooltip

Icon Default

→ Pass through msg if payload matches valid state: ☒

When clicked, send:

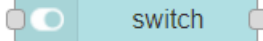
On Payload true

Off Payload false

Topic msg. topic

Class Optional CSS class name(s) for widget

Name LED1

Node-RED Flow -- เลือก  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group
[MQTT Server] Status2

Size
auto

Label
LED2

Tooltip
optional tooltip

Icon
Default

Pass through msg if payload matches valid state: ☒

When clicked, send:

On Payload
true

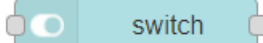
Off Payload
false

Topic
msg. topic

Class
Optional CSS class name(s) for widget

Name
LED2

☐ Enabled

Node-RED Flow -- เลือก  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group
[MQTT Server] Status1

Size
auto

Label
LED3

Tooltip
optional tooltip

Icon
Default

Pass through msg if payload matches valid state: ☒

When clicked, send:

On Payload
true

Off Payload
false

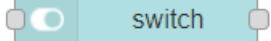
Topic
msg. topic

Class
Optional CSS class name(s) for widget

Name
LED3

☐ Enabled



Node-RED Flow -- เลือก  ตั้งค่าตามนี้แล้วกด Done

**Edit switch node**

Delete Cancel Done

**Properties**

Group [MQTT Server] Status2

Size auto

Label LED4

Tooltip optional tooltip

Icon Default

Pass through msg if payload matches valid state: ☒

When clicked, send:

On Payload true

Off Payload false

Topic msg. topic

Class Optional CSS class name(s) for widget

Name LED4

Enabled

Node-RED Flow -- เลือก  ตั้งค่าตามนี้แล้วกด Done

**Edit gauge node**

Delete Cancel Done

**Properties**

Group [MQTT Server] Status1

Size auto

Type Level

Label gauge



Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED1

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group [MQTT Server] Status2

Size auto

Type Level

Label gauge



Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED2

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group [MQTT Server] Status1

Size auto

Type Level

Label gauge


Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED3

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group [MQTT Server] Status2

Size auto

Type Level

Label gauge



Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED4

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Edit gauge node

Delete
Cancel
Done

Properties

Group [MQTT Server] Temperature

Size auto

Type Gauge

Label

Value format {{value}}

Units optional sub-label

Range min 0 max 100

Colour gradient

Sectors 0 ... optional ... optional ... 100

Fill gauge from centre. ☐

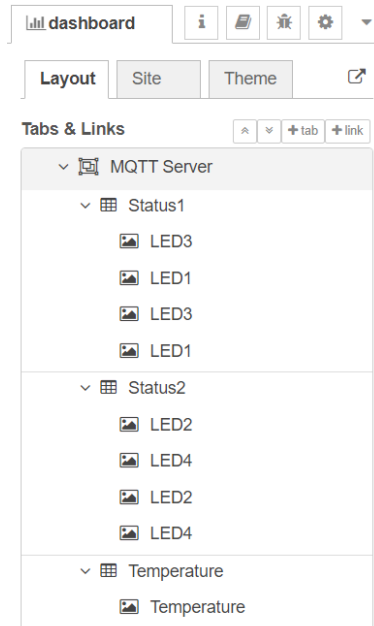
Class Optional CSS class name(s) for widget

Name Temperature

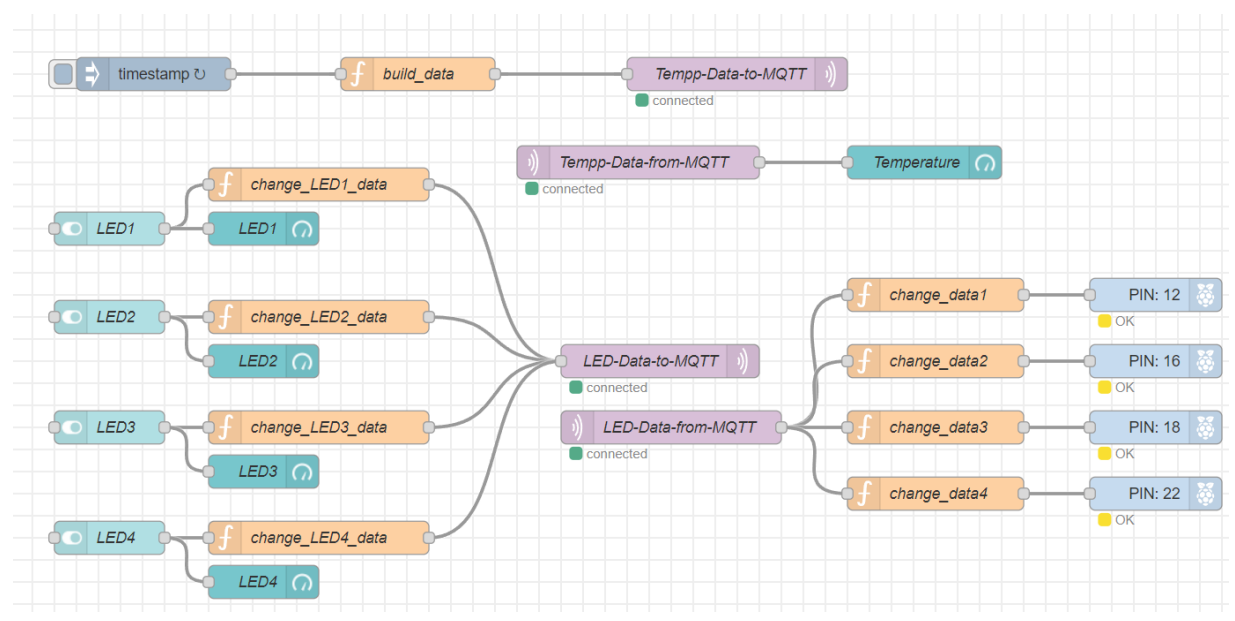
☐ Enabled

จัดเรียง dashboard โดยไปที่ -> Dashboard ที่ด้านขวบน

ที่ tab LED ให้จัดเรียงตามรูปนี้ โดยสามารถใช้mouse ลากเพื่อสลับลำดับได้เลย



นำทุก node มาต่อรวมกันดังรูป



## MQTT x

The image shows two screenshots of the MQTTX application interface.

**Top Screenshot: 'New' Connection Configuration**

The 'New' window is open, showing the 'General' tab. The configuration fields are as follows:

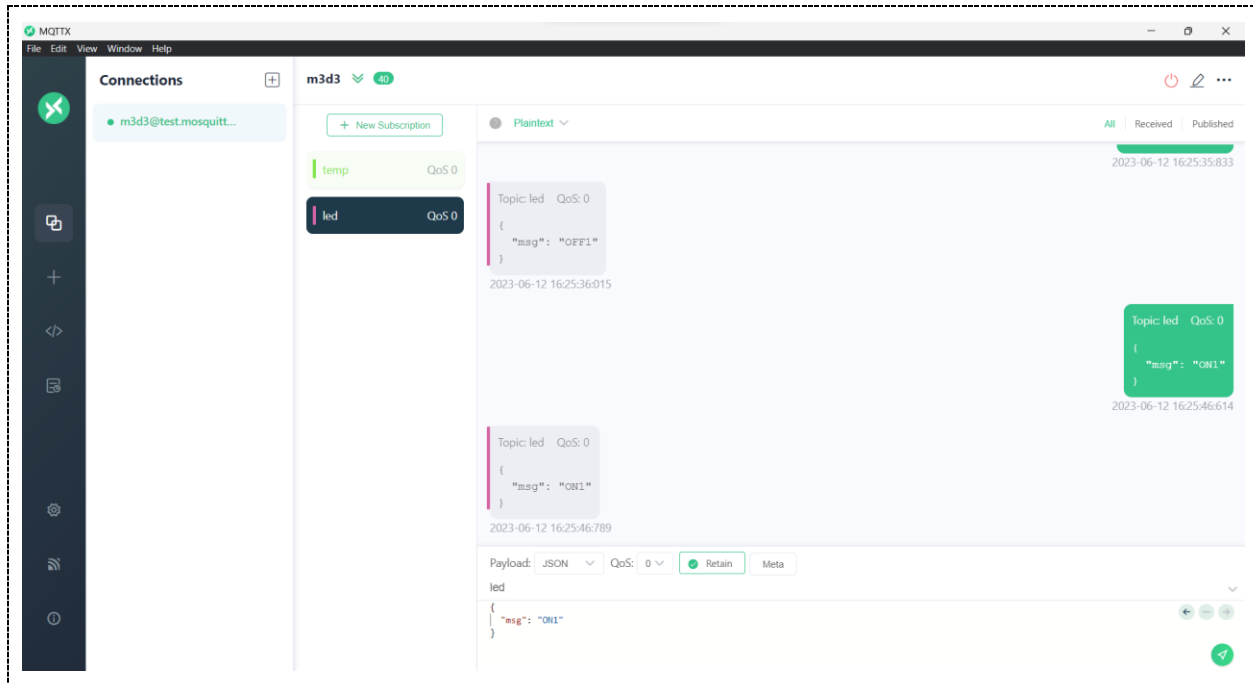
- Name:** m3d3
- Client ID:** mqttx\_d16c9e6c
- Host:** mqtt:// test.mosquitto.org
- Port:** 1883
- Username:** (empty)
- Password:** (empty)
- SSL/TLS:** (disabled)
- Advanced:**
  - MQTT Version:** 5.0
  - Connect Timeout:** 10 (s)
  - Keep Alive:** 60 (s)
  - Auto Reconnect:** (enabled)

**Bottom Screenshot: 'Connections' Screen**

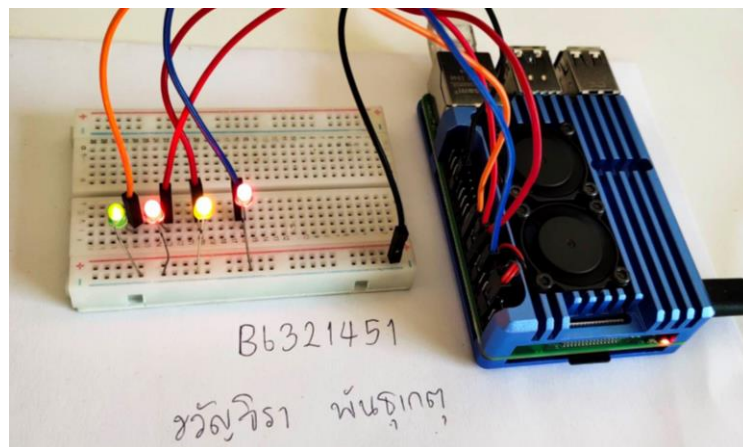
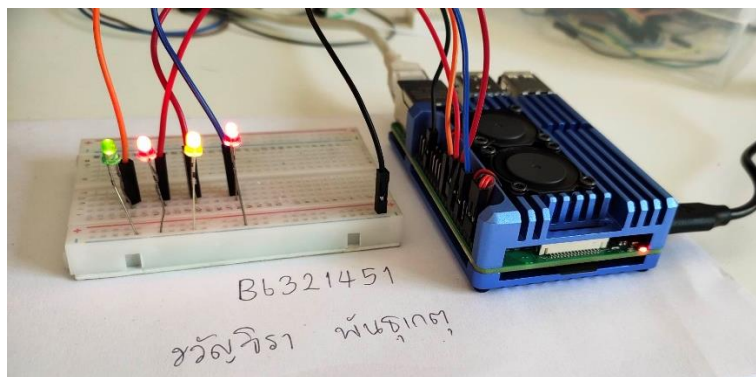
The 'Connections' window shows a list of connections. The connection 'm3d3@test.mosquitto...' is selected. The 'temp' topic is highlighted, showing a list of messages received on the 'temp' topic. The messages are as follows:

Topic	QoS	Message	Timestamp
temp	0	15.25	2023-06-12 16:15:54:404
temp	0	14.95	2023-06-12 16:21:54:404
temp	0	14.99	2023-06-12 16:27:54:554
temp	0	14.9	2023-06-12 16:33:54:411
temp	0	14.93	2023-06-12 16:39:54:566

The 'Payload' field shows the message content: `{ "msg": "ON1" }`.



รูปวงจร



**Quiz\_303 – MQTT Server on Raspberry Pi**

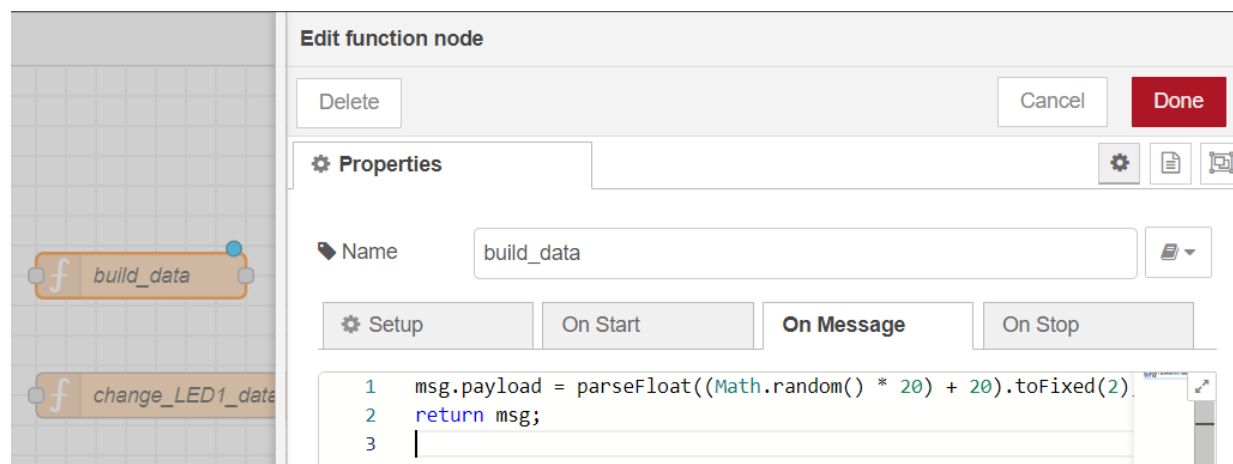
- ทดสอบควบคุม 4 LED ด้วย Raspberry Pi ผ่าน Node-RED ไปยัง private server
- เพิ่มเติมให้ส่งข้อมูล Temperature ไปแสดงที่ บน private Server

โปรแกรมที่ใช้ทดสอบ - ติดตั้ง MQTT Server บน Raspberry Pi
<code>sudo apt-get update</code>
<code>sudo apt-get upgrade</code>
หลังจากนั้น Reboot Raspberry Pi
<code>sudo reboot</code>
หลังจาก Reboot Raspberry Pi ใช้คำสั่งต่อไปนี้บน terminal ของ Raspberry Pi
<code>mkdir mosquitto</code>
<code>cd mosquitto</code>
<code>wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key</code>
<code>sudo apt-key add mosquitto-repo.gpg.key</code>
<code>cd /etc/apt/sources.list.d/</code>
<code>sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list</code>
<code>sudo apt-get update</code>
<code>sudo apt-cache search mosquito</code>
<code>sudo apt-get install mosquitto</code>
<code>sudo apt-get install mosquitto-clients</code>


ตรวจสอบสถานะการทำงานของ MQTT Server
<code>service mosquitto status</code>
<code>ps -ef   grep mosq</code>
<code>netstat -tln   grep 1883</code>
ถ้าเห็นมี service ของ Mosquitto ทำงานอยู่แสดงว่าใช้งานได้
ใช้คำสั่งต่อไปนี้เพื่อเข้าไปแก้ไขไฟล์การตั้งค่าให้กับ MQTT Server
<code>sudo nano /etc/mosquitto/mosquitto.conf</code>
เพิ่ม 3 บรรทัดนี้ต่อท้ายเข้าไปด้านล่าง
<code>allow_anonymous false</code> <code>password_file /etc/mosquitto/pwfile</code> <code>listener 1883</code>
กด Ctrl + o แล้วกด Enter เพื่อ save
กด Ctrl + x เพื่อออกจากการทำงานแก้ไขไฟล์
ตั้ง username และ password ที่เอาไว้ใช้เข้าถึง MQTT Server
Username: mymqtt
<code>sudo mosquitto_passwd -c /etc/mosquitto/pwfile mymqtt</code>
Password: myraspi
Reenter password: myraspi
หลังจากนั้น Reboot Raspberry Pi เพื่อให้ MQTT Server เริ่มทำงาน
<code>sudo reboot</code>

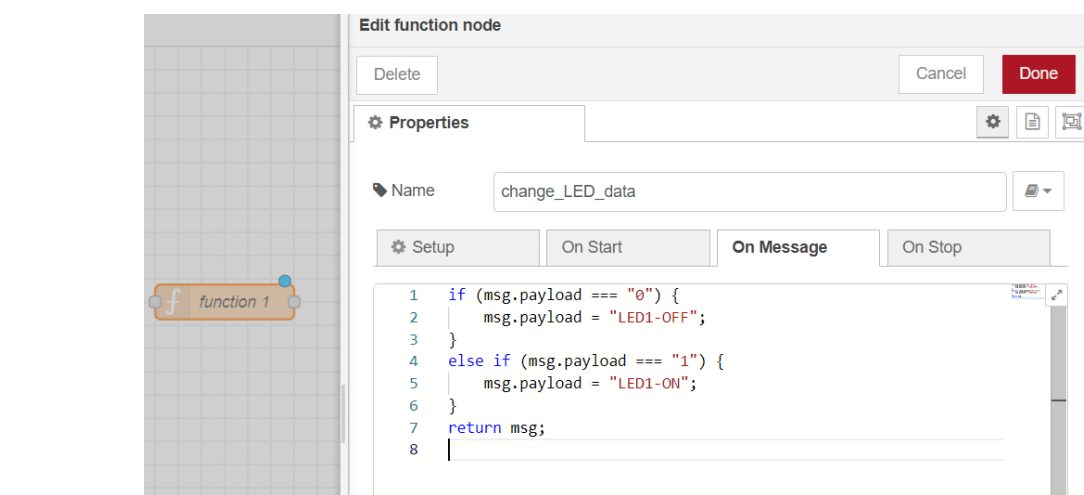


Node-RED Code -- เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done




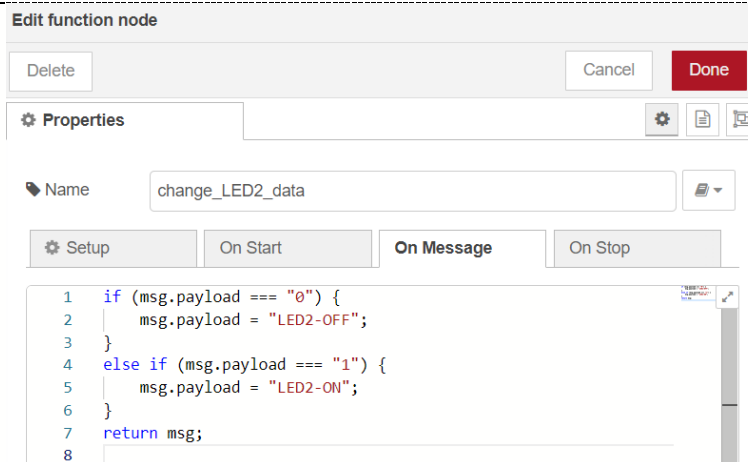
```
msg.payload = parseFloat((Math.random() * 20) + 20).toFixed(2);
return msg;
```

Node-RED Code -- เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done



```
if (msg.payload === "0") {
    msg.payload = "LED1-OFF";
}
else if (msg.payload === "1") {
    msg.payload = "LED1-ON";
}
return msg;
```



Node-RED Code -- เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done

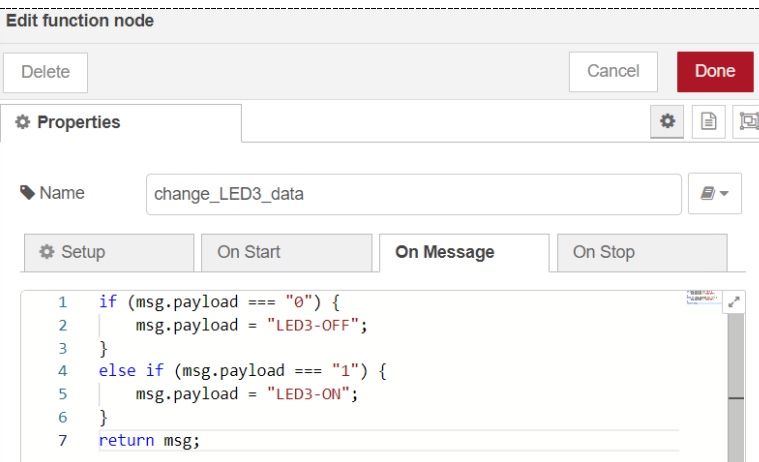


```

if (msg.payload === "0") {
    msg.payload = "LED2-OFF";
}
else if (msg.payload === "1") {
    msg.payload = "LED2-ON";
}
return msg;

```



Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done

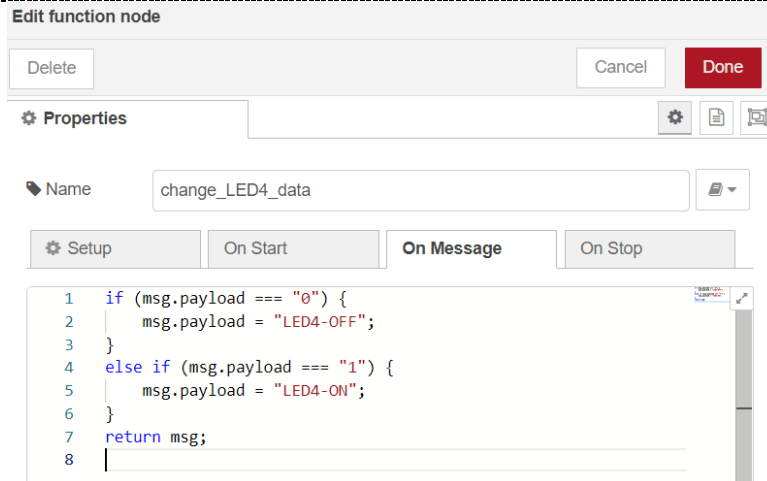


```

if (msg.payload === "0") {
    msg.payload = "LED3-OFF";
}
else if (msg.payload === "1") {
    msg.payload = "LED3-ON";
}
return msg;

```


Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done

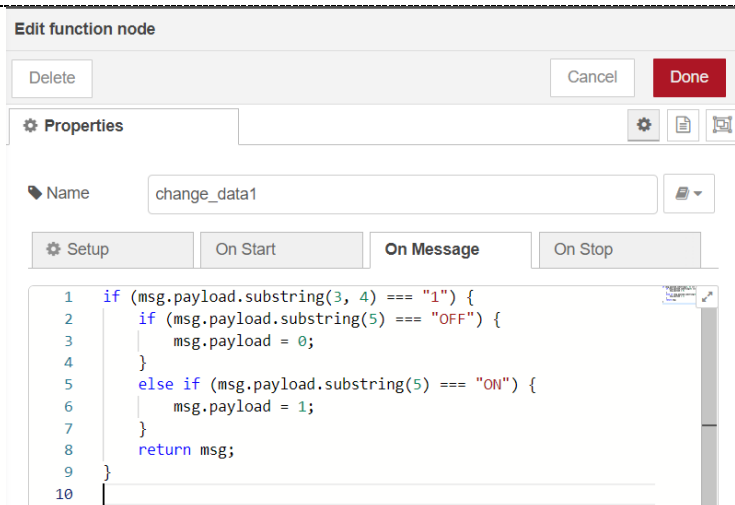


```

if (msg.payload === "0") {
    msg.payload = "LED4-OFF";
}
else if (msg.payload === "1") {
    msg.payload = "LED4-ON";
}
return msg;

```


Node-RED Code -- เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done



```

if (msg.payload.substring(3, 4) === "1") {
    if (msg.payload.substring(5) === "OFF") {
        msg.payload = 0;
    }
    else if (msg.payload.substring(5) === "ON") {
        msg.payload = 1;
    }
    return msg;
}

```

Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done

Delete Cancel Done

Properties

Name change\_data2

Setup On Start On Message On Stop

```



1 if (msg.payload.substring(3, 4) === "2") {
2   if (msg.payload.substring(5) === "OFF") {
3     msg.payload = 0;
4   }
5   else if (msg.payload.substring(5) === "ON") {
6     msg.payload = 1;
7   }
8   return msg;
9 }

```

```

if (msg.payload.substring(3, 4) === "2") {
  if (msg.payload.substring(5) === "OFF") {
    msg.payload = 0;
  }
  else if (msg.payload.substring(5) === "ON") {
    msg.payload = 1;
  }
  return msg;
}

```

Node-RED Code -- เลือก  function  แล้วตั้งค่าตามนี้จากนั้นกด Done

Delete Cancel Done

Properties

Name change\_data3

Setup On Start On Message On Stop

```

1 if (msg.payload.substring(3, 4) === "3") {
2   if (msg.payload.substring(5) === "OFF") {
3     msg.payload = 0;
4   }
5   else if (msg.payload.substring(5) === "ON") {
6     msg.payload = 1;
7   }
8   return msg;
9 }
10

```

```


if (msg.payload.substring(3, 4) === "3") {
  if (msg.payload.substring(5) === "OFF") {
    msg.payload = 0;
  }
  else if (msg.payload.substring(5) === "ON") {

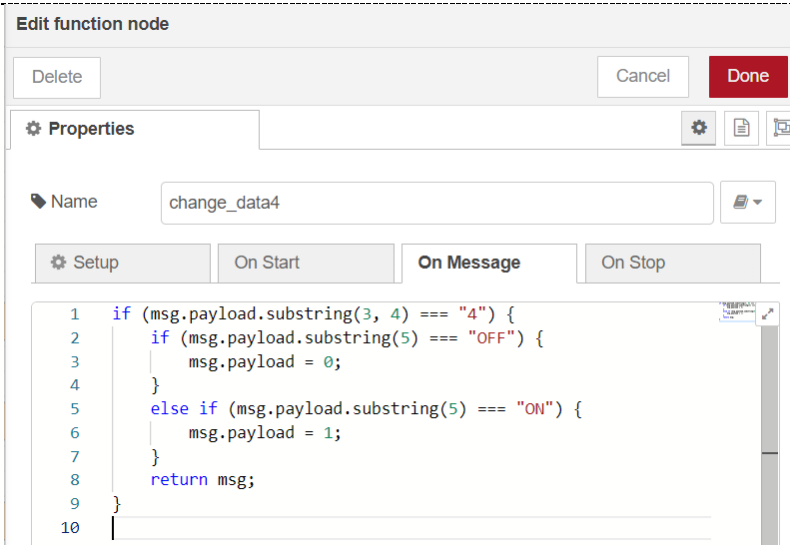
```

```

    msg.payload = 1;
  }
  return msg;
}

```


Node-RED Code -- เลือก  function แล้วตั้งค่าตามนี้จากนั้นกด Done

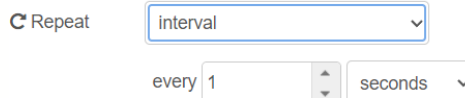


```

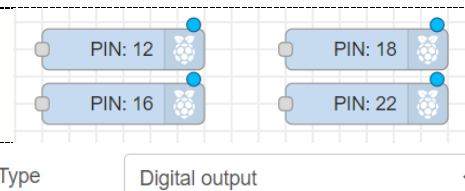
if (msg.payload.substring(3, 4) === "4") {
  if (msg.payload.substring(5) === "OFF") {
    msg.payload = 0;
  }
  else if (msg.payload.substring(5) === "ON") {
    msg.payload = 1;
  }
  return msg;
}

```

Node-RED Code -- เลือก  inject แล้วตั้งค่าตามนี้จากนั้นกด Done




Node-RED Flow -- เลือก  rpi - gpio out มา 4 node ตั้งค่าตามนี้แล้วกด Done



Node-RED Flow -- เลือก **mqtt out** ตั้งค่าตามนี้แล้วกด Done

---

**Properties**

Server  

---

Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

**Properties**

Name

Connection Security Messages

Server  Port

☒ Connect automatically  
☐ Use TLS

Protocol

Client ID

Keep Alive

Session ☒ Use clean session

---

Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

**Properties**

Name

Connection Security Messages

Username

Password


Username: mymqtt  
Password: myraspi

---

Edit mqtt out node

Delete Cancel Done

**Properties**

Server  

Topic

QoS  Retain

Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit mqtt out node**

Delete Cancel Done

**Properties**

Server m3d3

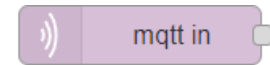
Topic led

QoS 0 Retain true

Name LED-Data-to-MQTT

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server m3d3

Action Subscribe to single topic

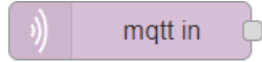
Topic temp

QoS 0

Output auto-detect (parsed JSON object, string or buffer)

Name Tempp-Data-from-MQTT

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server m3d3

Action Subscribe to single topic

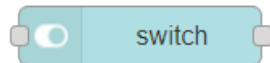
Topic led

QoS 0

Output auto-detect (parsed JSON object, string or buffer)

Name LED-Data-from-MQTT

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit switch node**

Delete Cancel Done

**Properties**

Group [MQTT Server] Status1

Size auto

Label LED1

Tooltip optional tooltip

Icon Default

→ Pass through msg if payload matches valid state: ☒

When clicked, send:

On Payload true

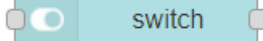
Off Payload false

Topic msg. topic

Class Optional CSS class name(s) for widget

Name LED1



Node-RED Flow -- เลือก  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group
[MQTT Server] Status2

Size
auto

Label
LED2

Tooltip
optional tooltip

Icon
Default

Pass through msg if payload matches valid state:
☒

When clicked, send:

On Payload
true

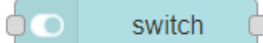
Off Payload
false

Topic
msg. topic

Class
Optional CSS class name(s) for widget

Name
LED2

☐ Enabled

Node-RED Flow -- เลือก  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group
[MQTT Server] Status1

Size
auto

Label
LED3

Tooltip
optional tooltip

Icon
Default

Pass through msg if payload matches valid state:
☒

When clicked, send:

On Payload
true

Off Payload
false

Topic
msg. topic

Class
Optional CSS class name(s) for widget

Name
LED3

☐ Enabled

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit switch node**

Delete Cancel Done

**Properties**

Group [MQTT Server] Status2

Size auto

Label LED4

Tooltip optional tooltip

Icon Default

Pass through msg if payload matches valid state: ☒

When clicked, send:

On Payload true

Off Payload false

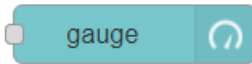
Topic msg. topic

Class Optional CSS class name(s) for widget

Name LED4

Enabled

Node-RED Flow -- เลือก



ตั้งค่าตามนี้แล้วกด Done

**Edit gauge node**

Delete Cancel Done

**Properties**

Group [MQTT Server] Status1

Size auto

Type Level

Label gauge


Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED1

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group [MQTT Server] Status2

Size auto

Type Level

Label gauge



Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED2

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group [MQTT Server] Status1

Size auto

Type Level

Label gauge



Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED3

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Delete
Cancel
Done

Properties

Group [MQTT Server] Status2

Size auto

Type Level

Label gauge



Value format {{value}}

Units units

Range min 0 max 1

Class Optional CSS class name(s) for widget

Name LED4

Node-RED Flow -- เลือก  gauge  ตั้งค่าตามนี้แล้วกด Done

Edit gauge node

Delete
Cancel
Done

Properties

Group [MQTT Server] Temperature

Size auto

Type Gauge

Label

Value format {{value}}

Units optional sub-label

Range min 0 max 100

Colour gradient

Sectors 0 ... optional ... optional ... 100

Fill gauge from centre. ☐

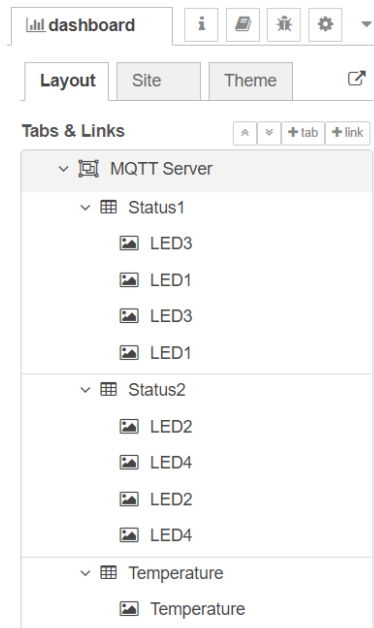
Class Optional CSS class name(s) for widget

Name Temperature

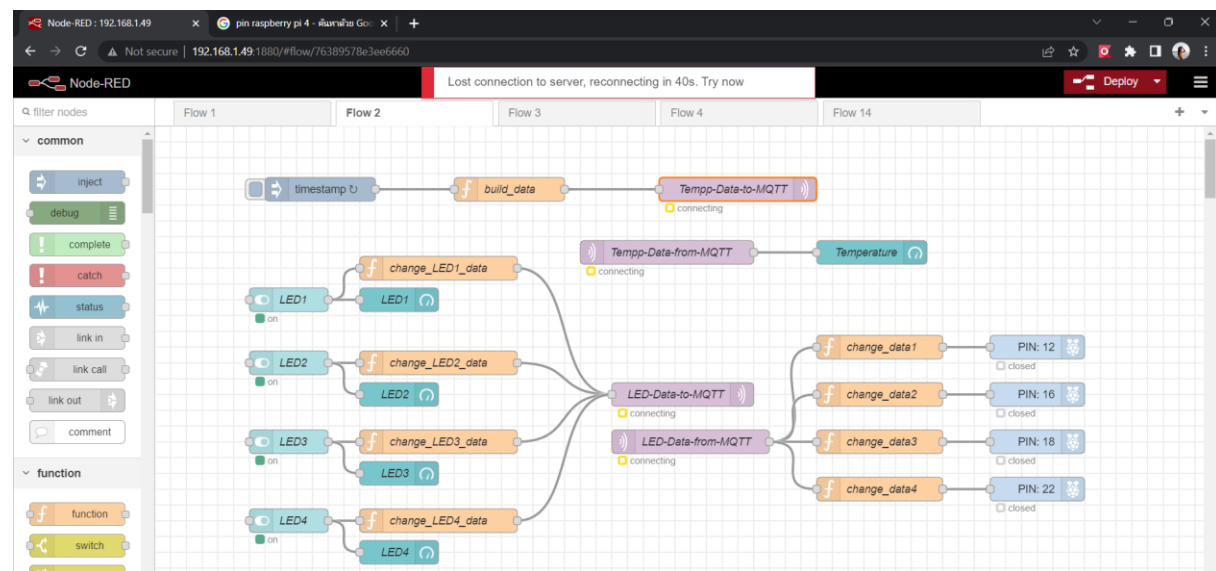
☐ Enabled

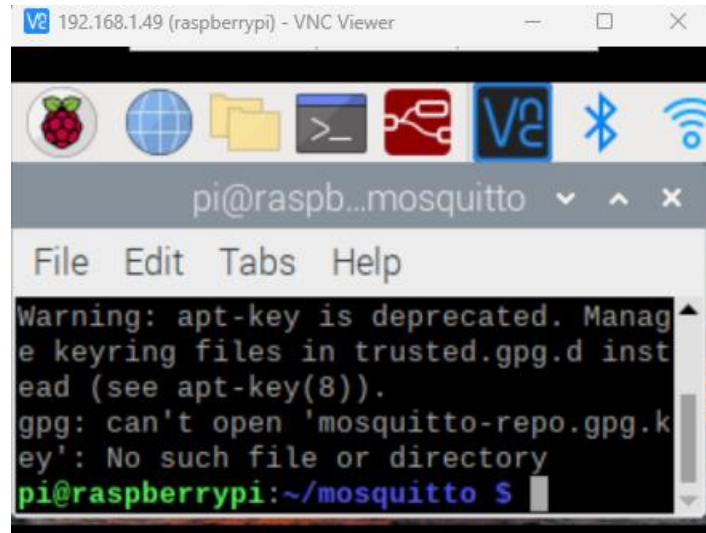
จัดเรียง dashboard โดยไปที่ -> Dashboard ที่ด้านขวามือ

ที่ tab LED ให้จัดเรียงตามรูปนี้ โดยสามารถใช้ mouse ลากเพื่อสลับลำดับได้เลย



นำทุก node มาต่อรวมกันดังรูป

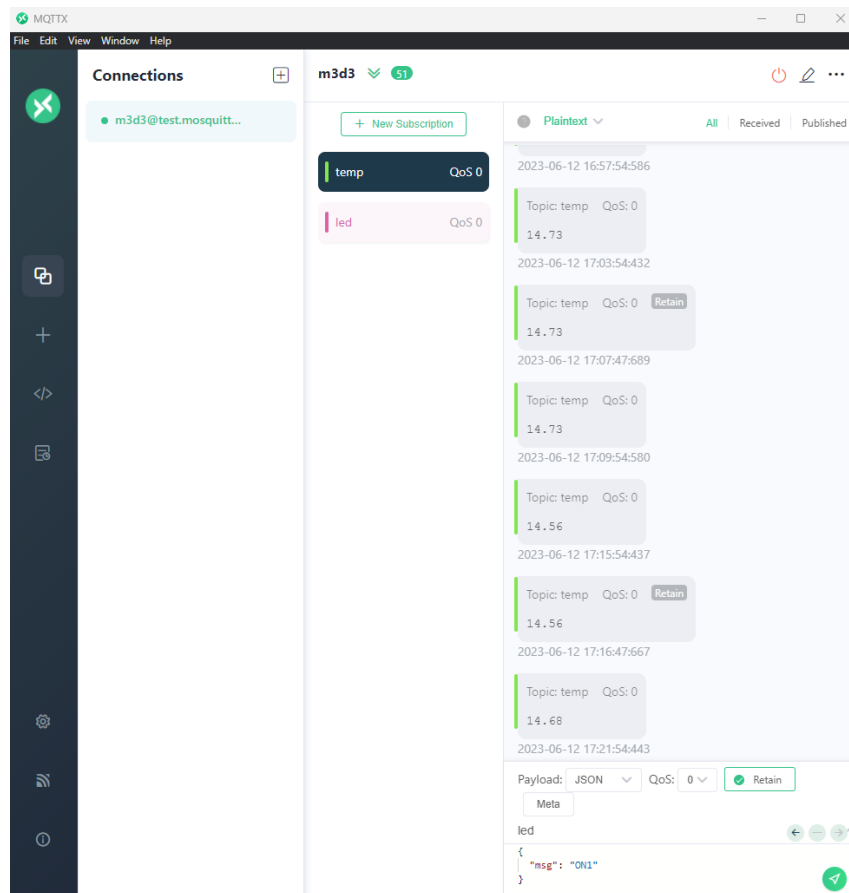


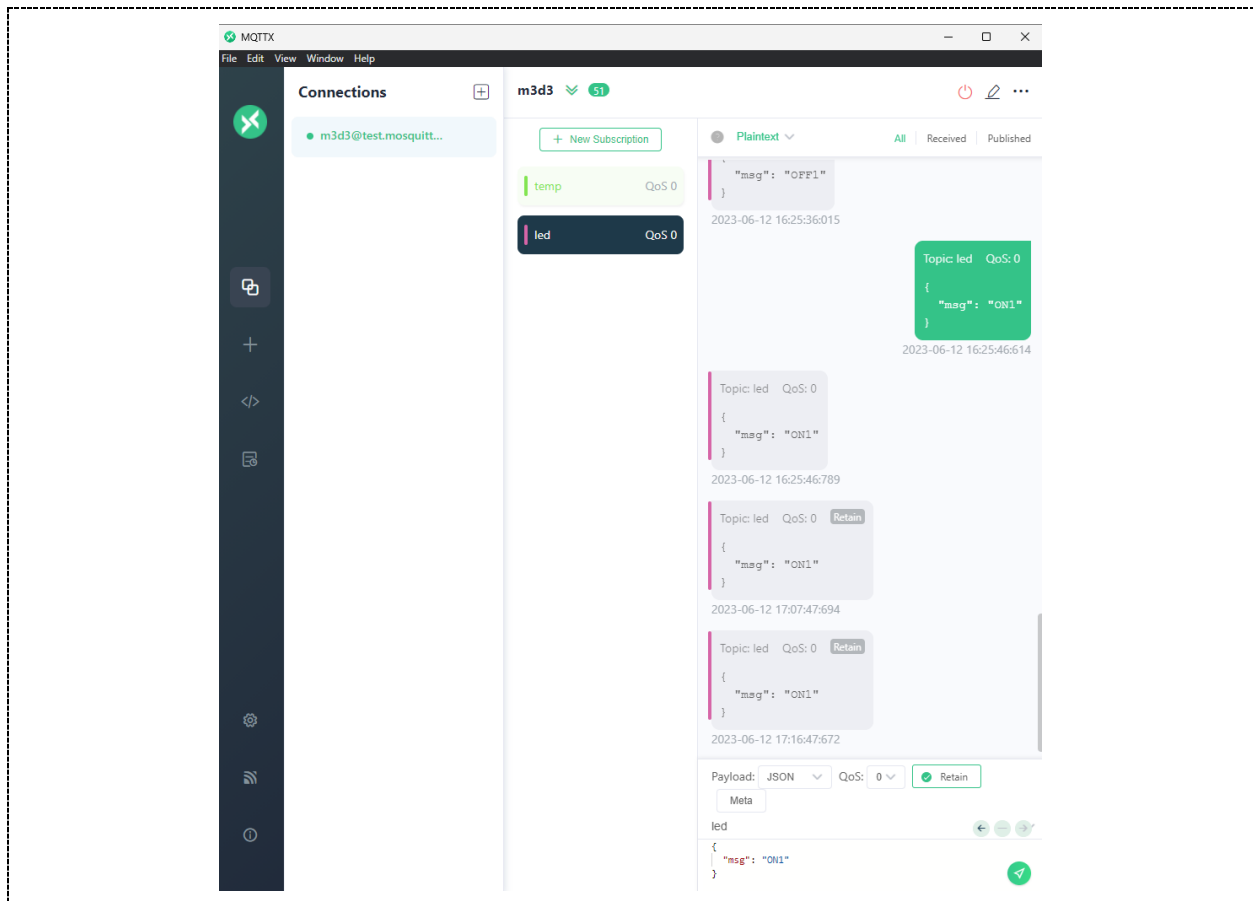


ปัญหาที่เกิดขึ้น

เกิดปัญหาที่ขั้นตอน **sudo apt-key add mosquitto-repo.gpg.key**

หน้าจอ MQTT





รูปวงจร

