

**Week0510.20230502-Grafana and InfluxDB**

กลุ่มคำนวณ \_\_\_\_ รหัส B6326319 ชื่อ-สกุล นายสิรภัทร สังข์ทอง

**1/4 การทำรายงาน**

- เพิ่มเติมเนื้อหาด้านท้าย ด้วยการคัดลอกและจัดเรียงใหม่ (3/4)
- รูปที่เป็นการทดสอบ ESP32 ควรเป็นรูปของตัวเองที่ทดสอบ (4/4)

**2/4. Read More**

- <https://grafana.com/blog/2021/03/08/how-i-built-a-monitoring-system-for-my-avocado-plant-with-arduino-and-grafana-cloud/>
- <https://www.metricfire.com/blog/iot-dashboards-with-grafana-and-prometheus/>
- <https://gabrieltanner.org/blog/grafana-sensor-visualization>

### 3/4. ให้จัดเรียงเรียงข้อมูล

#### ESP8266 / ESP32 & Mesh Network

- ESP Mesh Network 1 - <https://meetjoeblog.com/2018/03/25/esp8266-esp32-mesh-network-ep1/>
- ESP Mesh Network 2 - <https://meetjoeblog.com/2018/03/27/esp8266-esp32-mesh-network-painlessmesh-client-server-ep2/>
- ESP Mesh Network 3 - <https://meetjoeblog.com/2018/03/30/esp8266-esp32-mesh-network-painlessmesh-bridge-ep3/>
- ESP Mesh Network 4 - <https://meetjoeblog.com/2018/04/25/esp8266-esp32-painlessmesh-bridge-with-lora-ep4/>
- ESP Mesh Network 5 - <https://meetjoeblog.com/2018/08/30/esp8266-esp32-mesh-network-ep5-influxdb-grafana/>

#### Arm-Pelion Full Stack IoT Platform

- <https://www.techtalkthai.com/arm-pelion-full-stack-iot-platform/>

#### Grafana Dashboard

- <https://developers.ascendcorp.com/ทำความรู้จักกับ-grafana-dashboard-1a5efe6d170a>

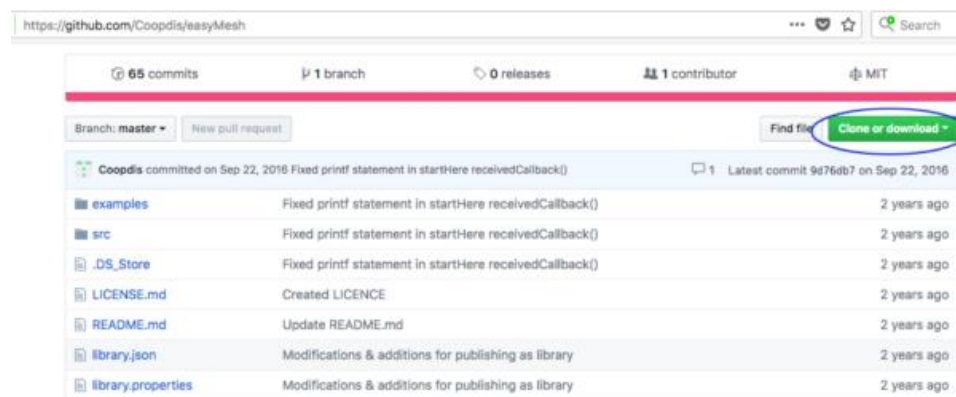
#### 4/4. การทดสอบ

- ให้ทำการทดสอบและเขียนขั้นตอนการทดสอบ โดยใช้ ESP32 ส่งข้อมูลไปยัง MQTT Broker แล้วใช้ Grafana . ในการมอนิเตอร์ข้อมูล โดยปรับแก้การทดสอบจาก <https://gabrielatanner.org/blog/grafana-sensor-visualization>

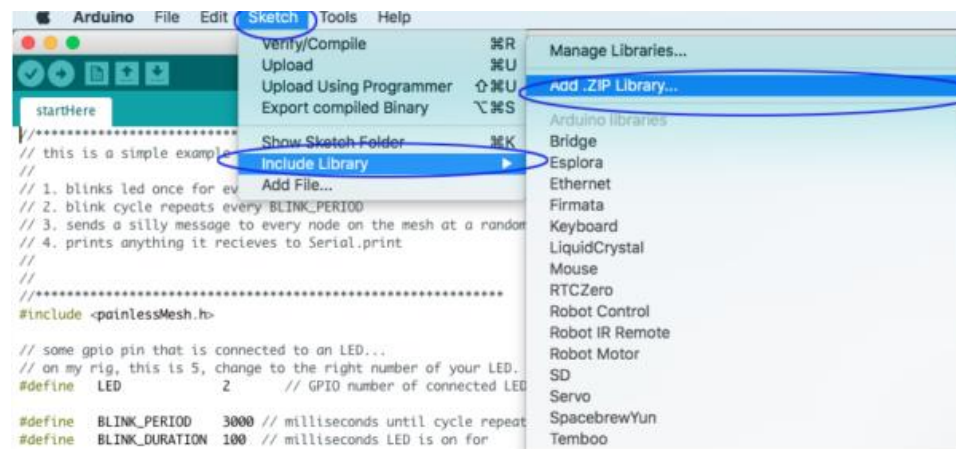
#### Visualize Sensor data using Grafana and InfluxDB

เตรียมความพร้อม

ซึ่งอันนี้สมมติว่าผู้อ่านได้ติดตั้ง board esp32 และ esp8266 ไว้เรียบร้อยแล้วนะครับ ขั้นแรกก็ Download/Clone Library จาก Github มาไว้ที่เครื่องก่อน โดยเข้าไปที่ [Painlessmesh ที่ Github](#)



จากนั้นก็ติดตั้งลงใน Arduino IDE โดยเข้าไปที่ Sketch → Include Library → Add .zip library แล้วเลือก zip file ที่เรา Download มา

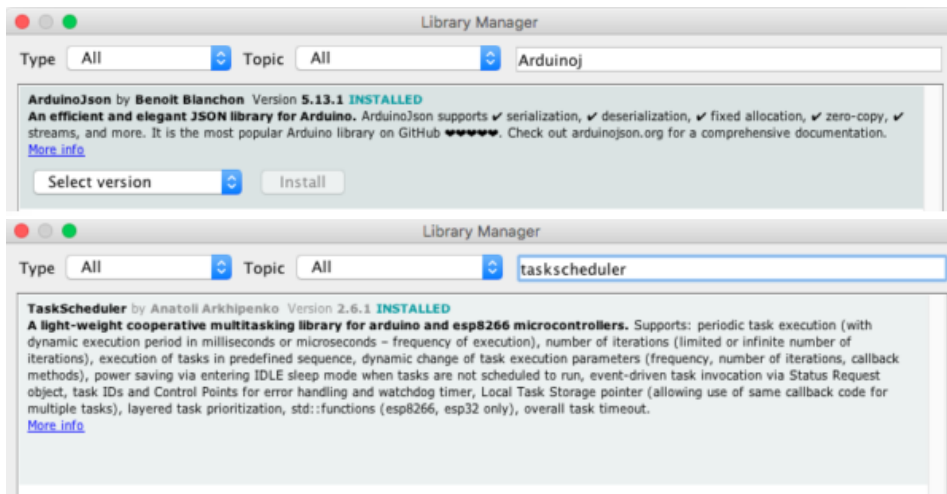


ติดตั้ง **Library Dependencies** ซึ่งจำเป็นต่อการใช้ **Painlessmesh**

ก่อนการใช้งานเรายังต้องใช้ Library อีกสองตัวนั่นก็คือ ArduinoJson และ TaskScheduler ซึ่งการ Install Library ก็สามารถทำได้ไม่ยากเลยสำหรับสองตัวนี้ เข้าไปที่ Sketch → Include Library → Manage Libraries จากนั้นในช่อง Filter Your Search ก็สามารถใส่ชื่อของ Library ทั้งสองตัวนี้ได้เลย

ArduinoJson

TaskScheduler



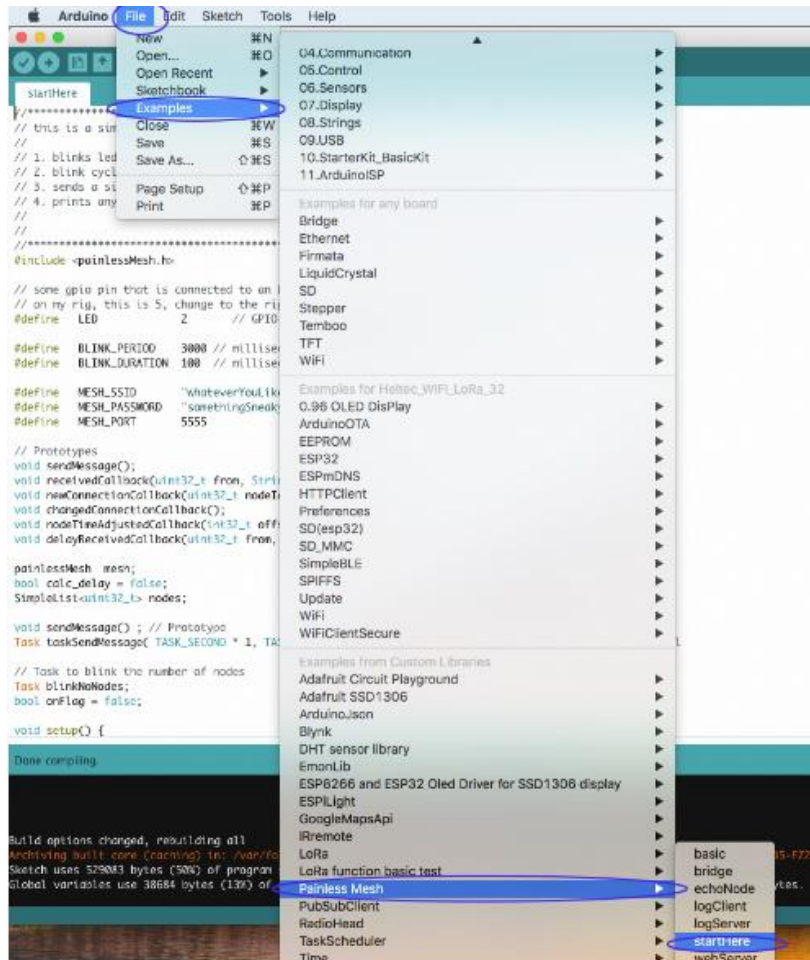
## Hello Mesh Network

หลังจากเตรียม environment ในการทำงานเรียบร้อยแล้วเราก็จะมาเริ่มทดลองสร้าง Mesh Network แรกของเรากันนะขอเจ้า เขียนร่ายมาซะยาว เริ่มซัก 5 node ก่อนละกัน โดยที่ Node ของเราจะประกอบด้วย



Wemos D1 Mini, ESP32 Wroom, ESP-01, แล้วก็ Nodemcu อีกสองตัว

ขั้นแรกเลยก็เปิดไฟล์ startHere กันก่อนเลยครับ ที่ Arduino IDE ก็เข้าไปที่ File->Examples->Painless Mesh->startHere



มาดูการตั้งค่าคร่าว ๆ ของ Mesh Network เรากัน

```
#define MESH_SSID "HelloMyMesh"
#define MESH_PASSWORD "hellomymeshnetwork"
#define MESH_PORT 5555
```

จากไฟล์ startHere ที่มากับตัวอย่าง เราสามารถที่จะแยกงานของ Mesh รวมถึงเพิ่ม Security ในส่วนของ Mesh Network เราได้จาก 3 บรรทัดบนนี้เลยครับ ซึ่ง help ของ painlessmesh รวมถึงคำอธิบายที่อยู่ใน code นั้นค่อนข้างดีเลยทีเดียว

ซึ่งการทำงาน Mesh Network โดยใช้ Painlessmesh นั้นจะแบ่งการทำงานเป็นลักษณะของ task โดยการกำหนดจาก task scheduler อย่างเช่น task ที่ใช้ในการส่ง message  
 mesh.scheduler.addTask( taskSendMessage );  
 taskSendMessage.enable();

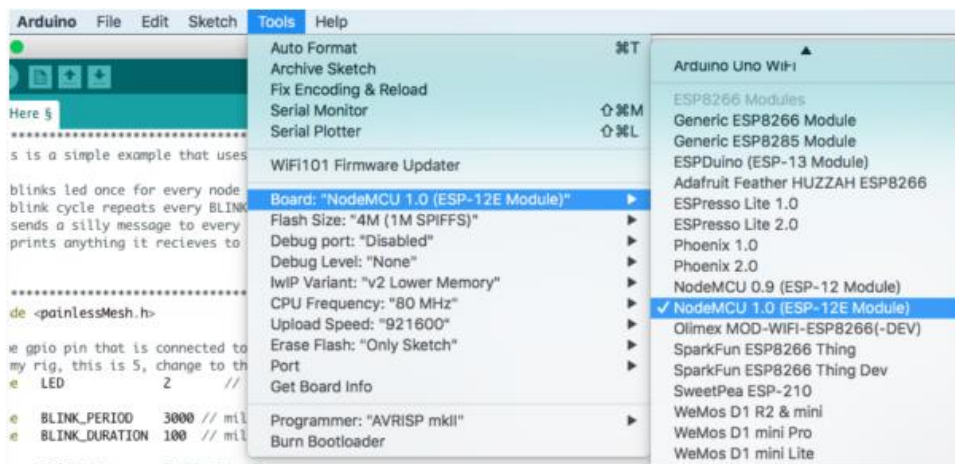
และในส่วนของ loop ก็จะมีแค่ การ update mesh เป็นหลัก พวก logic ต่างๆจะไปอยู่ที่ task scheduler หมด

```
mesh.update();
```

ในส่วนของ Callback นั้นหลักที่ใช้งานจริง ๆ เลยก็คือ receivedCallback เพื่อที่จะดูว่า message ที่เราได้รับมา ไม่ว่าจะเป็น Broadcast หรือมาแบบระบุ NodeID นั้นเราจะนำไปทำอะไรต่อซึ่งในตัวอย่างนี้ก็คือ Print ออกมาผ่านทาง Serial Port

```
void receivedCallback(uint32_t from, String & msg) {
  Serial.printf("startHere: Received from %u msg=%s\n", from, msg.c_str());
}
```

ซึ่งที่ผมแก้หลัก ๆ เลยก็คือตัว SSID และ Password แค่นั้นจากนั้นก็ทำการ Flash เลยครับ เลือก Board ให้ตรงรุ่นแค่นั้น



ซึ่งจุดประสงค์หลักของตัวอย่างนี้ก็ตามด้านล่างนี้เลย กระพริบ led ตามช่วงเวลาที่กำหนด แล้วก็ส่งข้อความแบบ broadcast ไปยังทุกๆ node ต้องลองทำตามครับตัวอย่างนี้ตัวอย่างเดียวได้อะไรเยอะ

```
// 1. blinks led once for every node on the mesh
// 2. blink cycle repeats every BLINK_PERIOD
// 3. sends a silly message to every node on the mesh at a random time between 1 and 5
seconds
// 4. prints anything it recieves to Serial.print
```

เมื่อทำการ flash nodemcu ตัวแรกไปแล้วเปิดที่ Serial Monitor ดู ข้อมูลมันก็จะเยอะๆ หน่อย เพราะมีการ enable debug ไว้ในตอน setup mesh.setDebugMsgTypes(ERROR | DEBUG | CONNECTION); ซึ่งระดับของการ debug ก็มีหลายแบบให้ลองเลือกใช้งานตามที่เหมาะสมดูครับ



```

/dev/cu.SLAB_USBtoUART
0x8 stationScan(): HelloMyMesh
0x8 espWifiEventCb(): SYSTEM_EVENT_SCAN_DONE
0x8 scanComplete():-- > scan finished @ 11078038 < --
0x8 scanComplete():-- > Cleared old aps.
0x8 scanComplete(): num=0, err=0
0x8 scanComplete(): After getting records, num=0, err=0
0x8 Found 0 nodes
0x8 connectToAP():0x8 connectToAP(): No unknown nodes found scan rate set to normal
Sending message: Hello from node 3896055851 myFreeMemory: 35728
Sending message: Hello from node 3896055851 myFreeMemory: 35728
Sending message: Hello from node 3896055851 myFreeMemory: 35728
Sending message: Hello from node 3896055851 myFreeMemory: 35728
0x8 stationScan(): HelloMyMesh
0x8 espWifiEventCb(): SYSTEM_EVENT_SCAN_DONE
0x8 scanComplete():-- > scan finished @ 21155879 < --
0x8 scanComplete():-- > Cleared old aps.
0x8 scanComplete(): num=0, err=0
0x8 scanComplete(): After getting records, num=0, err=0
0x8 Found 0 nodes
0x8 connectToAP():0x8 connectToAP(): No unknown nodes found scan rate set to normal
Sending message: Hello from node 3896055851 myFreeMemory: 35728
Sending message: Hello from node 3896055851 myFreeMemory: 35728
Autoscroll No line ending 115200 baud Clear output

```

ซึ่งเมื่อ flash เสร็จและเริ่มทำงาน nodemcu ตัวแรกในวง Mesh Network (มีตัวเดียวจะเรียก Mesh ทำมายยย) ก็จะเริ่มหา AP ที่ชื่อเดียวกัน บ้านใกล้เรือนเคียงของ node อื่นๆละ เพื่อที่จะดึงมาเข้าสู่ Mesh นี้โดยข้อมูลจากการ Debug ก็จะได้ nodeid รวมถึง memory ที่เหลืออยู่ละ

คราวนี้เรามา flash nodemcu อีกสองตัวเข้าไปดูบ้างว่าผลลัพธ์ที่ได้เป็นอย่างไร (note ไว้ก่อนว่า nodeid ของตัวแรกคือ 5851 เอาแค่ 4 ตัวสุดท้ายพอจะได้ไม่ต้องจำเยอะ และ memory ที่เหลือคือ 35728 byte)

```

/dev/cu.SLAB_USBtoUART
0x8 Found 2 nodes
0x8 connectToAP():0x8 connectToAP(): No unknown nodes found scan rate set to normal
startHere: Received from 3896664414 msg=Hello from node 3896664414 myFreeMemory: 31880
startHere: Received from 3896055851 msg=Hello from node 3896055851 myFreeMemory: 33872
Sending message: Hello from node 3893419003 myFreeMemory: 27088
startHere: Received from 3896664414 msg=Hello from node 3896664414 myFreeMemory: 31880
startHere: Received from 3896055851 msg=Hello from node 3896055851 myFreeMemory: 33872
startHere: Received from 3896664414 msg=Hello from node 3896664414 myFreeMemory: 31208
startHere: Received from 3896055851 msg=Hello from node 3896055851 myFreeMemory: 33872
Sending message: Hello from node 3893419003 myFreeMemory: 27088
startHere: Received from 3896664414 msg=Hello from node 3896664414 myFreeMemory: 31880
0x8 stationScan(): HelloMyMesh
0x8 espWifiEventCb(): SYSTEM_EVENT_SCAN_DONE
0x8 scanComplete():-- > scan finished @ 260926004 < --
0x8 scanComplete():-- > Cleared old aps.
0x8 scanComplete(): num=2, err=0
0x8 scanComplete(): After getting records, num=2, err=0
0x8 found : HelloMyMesh, -29dBm
0x8 found : HelloMyMesh, -15dBm
0x8 Found 2 nodes
0x8 connectToAP():0x8 connectToAP(): No unknown nodes found scan rate set to normal
startHere: Received from 3896055851 msg=Hello from node 3896055851 myFreeMemory: 33872
Autoscroll No line ending 115200 baud Clear output

```

คราวนี้เรามี Node ที่อยู่ใน Mesh นี้ 3 ตัวละนั่นคือ 5851, 4414 และ 9003 ซึ่งแต่ละตัวก็จะ Broadcast Message ออกมาตามเวลาที่สุ่มได้ในแต่ละตัว แต่ในขั้นตอนการกระพริบไฟ Led นั้นจะมีการตั้ง time synchronize เมื่อมี node ใหม่เพิ่มเข้ามาใน mesh ซึ่งจะมี Callback ไว้รองรับและทำงานร่วมกับ MeshUpdate ที่วน loop หาวามี Node ไหน drop connection ไปหรือมี node ไหน join เข้ามาและมีค่าความแรงของสัญญาณเท่าไร เพื่อใช้ในการจัดเส้นทางในการส่งข้อมูล

```

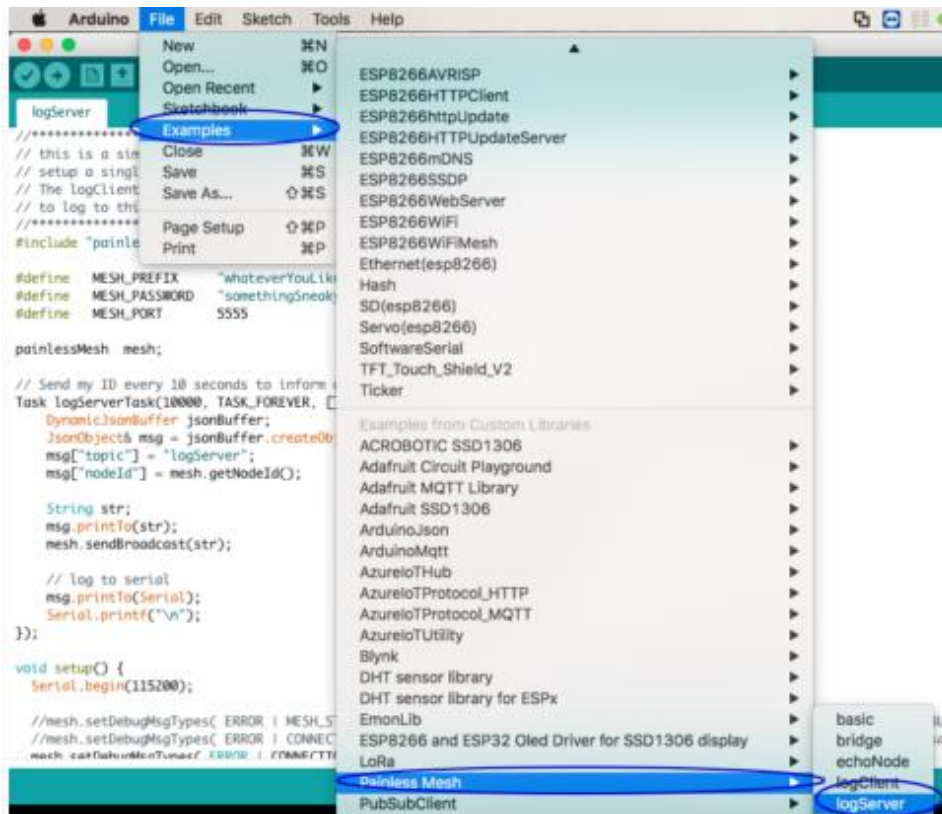
/dev/cu.SLAB_USBtoUART
0x8  espWifiEventCb(): SYSTEM_EVENT_AP_STACONNECTED
startHere: Received from 3896664414 msg=Hello from node 3896664414 myFreeMemory: 28656
0x8  stationScan(): HelloMyMesh
0x8  espWifiEventCb(): SYSTEM_EVENT_SCAN_DONE
0x8  scanComplete():-- > scan finished @ 583618487 < --
0x8  scanComplete():-- > Cleared old aps.
0x8  scanComplete(): num=1, err=0
0x8  scanComplete(): After getting records, num=1, err=0
0x8      found : HelloMyMesh, ~28d8m
0x8      Found 1 nodes
0x8  connectToAP():0x8      connectToAP(): No unknown nodes found scan rate set to normal
startHere: Received from 3896055851 msg=Hello from node 3896055851 myFreeMemory: 33336
Sending message: Hello from node 3893419003 myFreeMemory: 27872
Changed connections [{"nodeId":3896664414,"subs":[{"nodeId":3896055851,"subs":[]}]}]
Num nodes: 2
Connection list: 3896664414 3896055851
Sending message: Hello from node 3893419003 myFreeMemory: 27856
Delay to node 3896664414 is 8120 us
Delay to node 3896055851 is 12122 us
startHere: Received from 3896055851 msg=Hello from node 3896055851 myFreeMemory: 33824
startHere: Received from 3896664414 msg=Hello from node 3896664414 myFreeMemory: 31872
Sending message: Hello from node 3893419003 myFreeMemory: 27856
startHere: Received from 3896055851 msg=Hello from node 3896055851 myFreeMemory: 33808

☐ Autoscroll
No line ending 115200 baud Clear output

```

## เริ่มใช้งาน Painlessmesh แบบ Client/Server

ตัวอย่างประกอบสำหรับการใช้งานแบบ Client/Server จะใช้อยู่สองตัวด้วยกันคือ logServer และ logClient โดยสามารถเลือกเปิดได้จาก File->Examples->Painless Mesh-> logServer หรือ logClient





```
#define MESH_PREFIX "HelloMyMesh"
#define MESH_PASSWORD "hellomymeshnetwork"
#define MESH_PORT 5555
```

โดยเราจะเริ่มที่ logServer กันก่อนครับ ซึ่ง concept ก็ยังคงเหมือนเดิม ให้แก้ไขในส่วนของการตั้งค่า Mesh Network ของเรานี้ซึ่งก็คือ 3 บรรทัดด้านบนนี้ครับ หรือใครจะคงเดิมไว้เหมือนตัวอย่างที่ให้มาก็ได้แล้วก็ได้ Flash ลง ESP8266/ESP32 ของเราได้เลย

แต่ในส่วนของคนที่ใช้ ESP32 ต้องแก้หนึ่งครับ เพราะการ Declare ในส่วนของ Authentication Mode นั้นไม่เหมือนกัน แต่ ESP8266 จะแก้ให้เป็นแบบ ESP32 ก็สามารถใช้งานได้เหมือนกันครับ

```
mesh.init( MESH_PREFIX, MESH_PASSWORD, MESH_PORT, STA_AP, WIFI_AUTH_WPA2_PSK,
6 ); // <-- ต้องเพิ่มคำว่า WIFI_ เข้าไปด้วย
```

Concept ของตัวอย่าง logServer นั้นคล้ายๆกับตอนที่ 1 ครับเพียงแต่ว่า Broadcast Message ที่ส่งออกไปนั้น มีจุดประสงค์ในการส่งออกไปเพื่อบอกให้ node ที่อยู่ใน Mesh Network นี้รู้ว่า ฉัน NodeID นี้จะเป็น Server Node นะ ซึ่งก็จะทำงานด้วย task scheduler ทุกๆ 10 วิ ฉะนั้นเมื่อมี node ใหม่เข้ามา join ใน Mesh Network ก็ารู้แล้วว่าต้องส่งข้อมูลให้ใครซึ่งรูปแบบ Message ที่ส่งออกไปก็อยู่ในฟอร์แมทของ JSON ครับ โดยในส่วนของ receivedCallback ก็ยังเหมือนเดิมคือ ได้รับ message มากี่ Print ออกทาง Serial Port ไม่มีอะไรเปลี่ยนแปลง

```
Task logServerTask(10000, TASK_FOREVER, []() { DynamicJsonBuffer jsonBuffer;
JsonObject& msg = jsonBuffer.createObject(); msg["topic"] = "logServer";
//กำหนด Topic เพื่อที่ Node ที่ join เข้ามาหาข้อมูลของ Server เจอ msg["nodeId"] =
mesh.getNodeId(); // ใช้คำสั่ง mesh.getNodeID(); เพื่ออ่านค่า NodeID String str;
msg.printTo(str); mesh.sendBroadcast(str); // ส่งข้อความ Broadcast ไปยังทุก Node ว่าฉันคือ
Server Node นะ // log to serial msg.printTo(Serial); Serial.printf("\n"); });
```

```

/dev/cu.SLAB_USBtoUART
Send

0x8 stationScan(): HelloMyMesh
0x8 espWifiEventCb(): SYSTEM_EVENT_SCAN_DONE
0x8 scanComplete():-- > scan finished @ 21130500 < --
0x8 scanComplete():-- > Cleared old aps.
0x8 scanComplete(): num=0, err=0
0x8 scanComplete(): After getting records, num=0, err=0
0x8 Found 0 nodes
0x8 connectToAP():0x8 connectToAP(): No unknown nodes found scan rate set to normal
0x8 espWifiEventCb(): SYSTEM_EVENT_AP_STACONNECTED
0x8 New AP connection incoming
0x8 meshConnectedCb(): we are AP
0x8 findConnection(2487009679): did not find connection
0x8 newConnectionTask():
0x8 newConnectionTask(): adding 2487009679 now= 24504763
New Connection 2487009679
0x20 handleTimeSync(): Response sent {"type":2,"t0":2134652,"t1":24514068,"t2":24514302}
0x20 handleTimeSync(): timeSyncStatus with 2487009679 completed
0x20 handleTimeSync():
0x20 handleTimeSync(): Response sent {"type":2,"t0":24719063,"t1":24722058,"t2":24722307}
0x20 handleTimeSync(): timeSyncStatus with 2487009679 completed
0x20 handleTimeSync():
{"topic":"logServer","nodeId":2752206217}
0x8 stationScan(): HelloMyMesh
0x8 espWifiEventCb(): SYSTEM_EVENT_SCAN_DONE
0x8 scanComplete():-- > scan finished @ 31268843 < --
0x8 scanComplete():-- > Cleared old aps.
0x8 scanComplete(): num=1, err=0
0x8 scanComplete(): After getting records, num=1, err=0
0x8 Found : HelloMyMesh, -46dBm
0x8 Found 1 nodes
0x8 connectToAP():0x8 connectToAP(): No unknown nodes found scan rate set to normal
logServer: Received from 2487009679 msg={"topic":"sensor","value":1}

Autoscroll No line ending 115200 baud Clear output

```

หลังจาก Flash เสร็จการทำงานก็เป็นอย่างที่เราเห็นข้างบนครับ จะเริ่มทำการเช็คข้างเคียงว่ามี Node ไหนที่สามารถ Join ได้บ้างจากนั้นก็ปรับ Offset Time Synchronize ให้เท่ากับเพื่อนในวงชะ แล้วก็เข้าสู่ Task ที่ใช้ในการ Broadcast Message ออกไปที่เราคือ Server เพื่อให้ Node ทุกตัวที่อยู่ใน Mesh Network เดียวกันรู้ว่าจะส่งข้อมูลให้ Server ให้ส่งไปที่ NodeID หมายเลขอะไร ซึ่งในที่นี้ก็คือ 6217

มาดูในส่วนของ logClient กันบ้าง ซึ่งในส่วนของ receivedCallback เนี่ยแหละที่จะเปลี่ยนไป เพราะจะมีการเช็ค Broadcast Message จาก Node อื่นๆ ซึ่งก็คือ Server Node ว่า เหยี่ย นีแหละ Server Node นะ มี ID นี้ละ จากนั้นถ้าจะส่งข้อความไปที่ ServerNode ให้ส่งไปที่ NodeID นี้ละ

```

1. void receivedCallback( uint32_t from, String &msg ) {
2.     Serial.printf("logClient: Received from %u msg=%s\n", from, msg.c_str());
3.
4.     // Saving logServer
5.     DynamicJsonBuffer jsonBuffer;
6.     JsonObject& root = jsonBuffer.parseObject(msg);
7.     if (root.containsKey("topic")) {
8.         if (String("logServer").equals(root["topic"].as<String>())) { //เช็ค Topic ว่า
เป็น logServer หรือเปล่า
9.             // check for on: true or false
10.            logServerId = root["nodeId"]; //อ่านและบันทึกค่า nodeID ของ Server เพื่อใช้เป็น
ปลายทางในการส่ง
11.            Serial.printf("logServer detected!!!\n");
12.        }
13.        Serial.printf("Handled from %u msg=%s\n", from, msg.c_str());
14.    }
15. }

```

สำหรับส่วนของ Task ที่ใช้ในการส่ง Message ก็จะทำนองเดียวกัน คือถ้ายังไม่เจอว่า ใน Mesh Network ที่เพิ่ง join เข้ามา มี server หรือเปล่า ฉะนั้นก็ broadcast ข้อมูลของตนเลยละกัน เนื่องจาก Server Node มีระยะเวลา 10 วินาทีถึงจะ Broadcast

Message Topic “logserver” ออกไป ฉะนั้นการทำเช่นนี้ Server Node ที่รอข้อมูลจาก Client Node ก็จะได้ข้อมูลแน่ๆ (รวมถึงตัวอื่นๆด้วย)

```
Task myLoggingTask(10000, TASK_FOREVER, []) {
    DynamicJsonBuffer jsonBuffer;
    JsonObject& msg = jsonBuffer.createObject();
    msg["topic"] = "sensor";
    msg["value"] = random(0, 180); //สำหรับไฟล์ตัวอย่างนี้ยังไม่ได้ต่อเข้ากับเซ็นเซอร์

    String str;
    msg.printTo(str);
    if (logServerId == 0) // If we don't know the logServer yet
        mesh.sendBroadcast(str);
    else
        mesh.sendSingle(logServerId, str);

    // log to serial
    msg.printTo(Serial);
    Serial.printf("\n");
};
```

```
0x2 AP tcp server established on port 5555
0x8 stationScan(): HelloMyMesh
{"topic":"sensor2","value":173}
0x8 espWIFIEventCb(): SYSTEM_EVENT_SCAN_DONE
0x8 scanComplete():-- > scan finished @ 21204468 < --
0x8 scanComplete():-- > Cleared old aps.
0x8 scanComplete(): num=2, err=0
0x8 scanComplete(): After getting records, num=2, err=0
0x8 found: HelloMyMesh, -49dBm
0x8 found: HelloMyMesh, -46dBm
0x8 Found 2 nodes
0x8 findConnection(2752206217): did not find connection
0x8 findConnection(2487009679): did not find connection
0x8 connectToAP():0x8 connectToAP(): Best AP is 2487009679<---
0x8 connectToAP(): Trying to connect, scan rate set to 4*normal
0x8 espWIFIEventCb(): SYSTEM_EVENT_STA_AUTHMODE_CHANGE
0x8 espWIFIEventCb(): SYSTEM_EVENT_STA_CONNECTED
0x8 espWIFIEventCb(): SYSTEM_EVENT_STA_GOT_IP
0x8 New STA connection incoming
0x8 meshConnectedCb(): we are STA
0x8 findConnection(2487009679): did not find connection
0x8 newConnectionTask():
0x8 newConnectionTask(): adding 2487009679 now= 22385371
{"topic":"sensor2","value":63}
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":2752206217} Broadcast Message ที่ได้จ้กแ้ก้
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":2752206217} เพื่อจ้กแ้ก้มี Topic Server ที่ทำข้้นน้ก Serve NodeID
{"topic":"sensor2","value":2}
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":2752206217} ข้้นน้กส่งออก ไปในร้บแ้บ JSON
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":2752206217}
{"topic":"sensor2","value":149} ข้้นน้กส่งออก ไปในร้บแ้บ JSON
```

คราวนี้แทนที่เราจะจำลองข้อมูลเหมือนในตัวอย่าง ก็ได้เวลาเปิดกระป๋องและ เอา DHT22 มาพ่วงเข้าไป คราวนี้ Temp/Humid ที่ได้ก็จะเป็นค่าจริงแล้วโดยการปรับ Code ในส่วนของ Task ที่ใช้สำหรับการส่งข้อความ ซึ่งจะมีการอ่านค่าจาก DHT22 แล้วจัดรูปแบบให้อยู่ในฟอร์แมทของ JSON โดยที่ Mesh Network นี้จะประกอบไปด้วย 12 Node ด้วยกันดังนี้



- ESP32 Wroom ทำตัวเป็น Server
- ESP-01 เป็น Client ที่ส่งค่าส่งเข้ามา
- Wemos D1 Mini 4 ตัว เป็น Client ที่ส่งค่าส่งเข้ามา (Wemos-01 ถึง Wemos-04)
- Wemos D1 Mini 1 ตัว เป็น Client ที่ส่งค่า Temp/Humid ที่อ่านได้จาก DHT22
- Nodemcu 4 ตัว เป็น Client ที่ส่งค่าส่งเข้ามา (mcu-01 ถึง mcu-04)
- Nodemcu 1 ตัว เป็น Client ที่ส่งค่า Temp/Humid ที่อ่านได้จาก DHT22

Code ที่ทำการแก้ไขสำหรับ Node ตัวที่ส่งค่า (ESP-01, Wemos-01 ถึง 04, mcu-01 ถึง 04)

```
// Send message to the logServer every 10 seconds
Task myLoggingTask(5000, TASK_FOREVER, []) {
    DynamicJsonBuffer jsonBuffer;
    JsonObject& msg = jsonBuffer.createObject();
    msg["nodename"] = "mcu3"; //เพิ่ม nodeName เข้ามาพอทราบ
    msg["NodeID"] = mesh.getNodeId(); //เพิ่ม NodeID เอาไว้อ้างอิง
    msg["random value"] = random(0, 180); //เปลี่ยนเป็น random value

    String str;
    msg.printTo(str);
    if (logServerId == 0) // If we don't know the logServer yet
        mesh.sendBroadcast(str);
    else
        mesh.sendSingle(logServerId, str);

    // log to serial
    msg.printTo(Serial);
    Serial.printf("\n");
};
```

Code ที่ทำการแก้ไขสำหรับ Node ตัวที่ส่งค่า Temp/Humid จาก DHT22 (Wemos-t1 และ mcu-t1)

```
#include "painlessMesh.h"
#include "DHT.h"
```

```
#define DHTPIN D4
#define DHTTYPE DHT22
```

```

#define MESH_PREFIX      "HelloMyMesh"
#define MESH_PASSWORD    "hellomymeshnetwork"
#define MESH_PORT        5555

DHT dht(DHTPIN, DHTTYPE);

void receivedCallback( uint32_t from, String &msg );

painlessMesh mesh;

size_t logServerId = 0;

// Send message to the logServer every 5 seconds
Task myLoggingTask(5000, TASK_FOREVER, []() {

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    DynamicJsonBuffer jsonBuffer;
    JsonObject& msg = jsonBuffer.createObject();
    msg["nodename"] = "Wemos-t1";
    msg["NodeID"] = mesh.getNodeId();
    msg["Temp"] = String(t)+"C";
    msg["Humidity"] = String(h)+"%";

    String str;
    msg.printTo(str);
    if (logServerId == 0) // If we don't know the logServer yet
        mesh.sendBroadcast(str);
    else
        mesh.sendSingle(logServerId, str);

    // log to serial
    msg.printTo(Serial);
    Serial.printf("\n");
});

void setup() {

    Serial.begin(115200);
    Serial.println("Begin DHT22 Mesh Network test!");

    dht.begin();

```

```

    mesh.setDebugMsgTypes( ERROR | STARTUP | CONNECTION ); // set before
    init() so that you can see startup messages

    mesh.init( MESH_PREFIX, MESH_PASSWORD, MESH_PORT, STA_AP, AUTH_WPA2_PSK, 6
);
    mesh.onReceive(&receivedCallback);

    // Add the task to the mesh scheduler
    mesh.scheduler.addTask(myLoggingTask);
    myLoggingTask.enable();

}

void loop() {
    // put your main code here, to run repeatedly:

    mesh.update();

}

void receivedCallback( uint32_t from, String &msg ) {
    Serial.printf("logClient: Received from %u msg=%s\n", from, msg.c_str());

    // Saving logServer
    DynamicJsonBuffer jsonBuffer;
    JsonObject& root = jsonBuffer.parseObject(msg);
    if (root.containsKey("topic")) {
        if (String("logServer").equals(root["topic"].as<String>())) {
            // check for on: true or false
            logServerId = root["nodeId"];
            Serial.printf("logServer detected!!!\n");
        }
        Serial.printf("Handled from %u msg=%s\n", from, msg.c_str());
    }
}

```

หลังจาก Flash ครบตัวสุดท้ายก็เปิด Serial Monitor ขึ้นมาจะเห็นได้จากรูปข้างล่างเลยครับว่า เจอ Node ข้างเคียงอยู่ทั้งหมด 10 Node พร้อมกับความแรงของสัญญาณจากแต่ละ Node ซึ่งจะใช้ในการจัดการการเชื่อมต่อของ Mesh Network ของเรา



```
~/dev/cy.SLAB_USBtoUART
Send

0x8 scanComplete():-- > scan finished @ 5750896 < --
0x8 scanComplete():-- > Cleared old aps.
0x8 scanComplete(): num=10, err=0
0x8 scanComplete(): After getting records, num=10, err=0
0x8         found : HelloMyMesh, -41dBm
0x8         found : HelloMyMesh, -42dBm
0x8         found : HelloMyMesh, -35dBm
0x8         found : HelloMyMesh, -37dBm
0x8         found : HelloMyMesh, -50dBm
0x8         found : HelloMyMesh, -43dBm
0x8         found : HelloMyMesh, -61dBm
0x8         found : HelloMyMesh, -59dBm
0x8         found : HelloMyMesh, -56dBm
0x8         found : HelloMyMesh, -57dBm
0x8         Found 10 nodes
0x8 findConnection(983688460): did not find connection
0x8 findConnection(2488971084): did not find connection
0x8 findConnection(2487009679): did not find connection
0x8 findConnection(2136600749): did not find connection
0x8 findConnection(3163409689): did not find connection
0x8 findConnection(572025877): did not find connection
0x8 findConnection(3893419003): did not find connection
0x8 findConnection(3896713776): did not find connection
0x8 findConnection(3163408206): did not find connection
0x8 findConnection(3896055851): did not find connection
0x8 connectToAP():0x8 connectToAP(): Best AP is 2487009679<--
Autoscroll No line ending 115200 baud Clear output
```

อย่างที่ได้บอกไปข้างต้นครับ เมื่อเริ่มการทำงาน Server จะ Broadcast Message เพื่อบอกเพื่อนๆ ใน Mesh Network นี้ว่า NodeID ของ Server นั้นคือหมายเลขอะไร เพื่อที่ว่าเมื่อบางครั้ง ได้มีการเพิ่ม Node เข้าไปใน Mesh Network นี้ก็สามารถที่จะรู้ได้ทันทีว่าจะต้องส่งข้อมูลไปที่ Server NodeID ไหน แต่ระหว่างที่ยังไม่รู้ว่าเป็น Server NodeID คือหมายเลขอะไร Client Node ก็จะต้อง Broadcast Message ออกไปเพื่อแจ้งไปก่อน อย่างน้อยมันก็ต้องเข้า Server Node แหะละ คราวนี้เรามาลองทดสอบกันดู ระหว่างที่ยังไม่เจอ Server Node และเมื่อ Server Node Online และทุกๆ Node รับทราบกันหมดแล้วและส่ง Message ออกไปแบบระบุ NodeID

### Scenario ที่ 1: No Server Node

ตามรูปด้านล่างนี้เลยครับ ข้อมูลที่ได้รับผ่าน `receivedCallback` ได้รับข้อมูลจาก Node ที่ Online ใน Mesh Network นี้ทั้งหมด เพราะะายังไม่ได้เปิด Server

```
[{"nodename": "mcu-04", "NodeID": 389664414, "random value": 87}
logClient: Received from 3163449689 msg= {"nodename": "Wemos-02", "NodeID": 3163449689, "random value": 95}
logClient: Received from 3896055851 msg= {"nodename": "mcu-02", "NodeID": 3896055851, "random value": 55}
logClient: Received from 3896713776 msg= {"nodename": "mcu-01", "NodeID": 3896713776, "random value": 156}
logClient: Received from 3163448026 msg= {"nodename": "Wemos-04", "NodeID": 3163448026, "random value": 175}
logClient: Received from 3893413903 msg= {"nodename": "mcu-03", "NodeID": 3893413903, "random value": 12}
logClient: Received from 572025877 msg= {"nodename": "Wemos-01", "NodeID": 572025877, "random value": 148}
logClient: Received from 2487090679 msg= {"nodename": "Wemos-t1", "NodeID": 2487090679, "Temp": 25.18C, "Humidity": 44.78%}
logClient: Received from 2136600749 msg= {"nodename": "Wemos-03", "NodeID": 2136600749, "random value": 16}
logClient: Received from 2488971884 msg= {"nodename": "mcu-t1", "NodeID": 2488971884, "Temp": 25.90C, "Humidity": 42.10%}
logClient: Received from 983688460 msg= {"nodename": "esp-01", "NodeID": 983688460, "random value": 96}
[{"nodename": "mcu-04", "NodeID": 389664414, "random value": 142}
logClient: Received from 3163449689 msg= {"nodename": "Wemos-02", "NodeID": 3163449689, "random value": 149}
logClient: Received from 3896055851 msg= {"nodename": "mcu-02", "NodeID": 3896055851, "random value": 37}
logClient: Received from 3896713776 msg= {"nodename": "mcu-01", "NodeID": 3896713776, "random value": 133}
logClient: Received from 3163448026 msg= {"nodename": "Wemos-04", "NodeID": 3163448026, "random value": 176}
logClient: Received from 3893413903 msg= {"nodename": "mcu-03", "NodeID": 3893413903, "random value": 13}
logClient: Received from 572025877 msg= {"nodename": "Wemos-01", "NodeID": 572025877, "random value": 91}
logClient: Received from 2136600749 msg= {"nodename": "Wemos-03", "NodeID": 2136600749, "random value": 13}
logClient: Received from 2487090679 msg= {"nodename": "Wemos-t1", "NodeID": 2487090679, "Temp": 25.40C, "Humidity": 44.80%}
logClient: Received from 2488971884 msg= {"nodename": "mcu-t1", "NodeID": 2488971884, "Temp": 25.90C, "Humidity": 42.10%}
logClient: Received from 983688460 msg= {"nodename": "esp-01", "NodeID": 983688460, "random value": 106}
[{"nodename": "mcu-04", "NodeID": 389664414, "random value": 26}
logClient: Received from 3163449689 msg= {"nodename": "Wemos-02", "NodeID": 3163449689, "random value": 27}
logClient: Received from 3896055851 msg= {"nodename": "mcu-02", "NodeID": 3896055851, "random value": 61}]
```

## Scenario ที่ 2: Server Node join Mesh Network

เมื่อเปิด Server Node เข้ามาก็จะเป็นเหมือนรูปข้างล่างนี้ครับ ได้รับ Broadcast Message จาก Server Node และในส่วนของ receivedCallback ก็จะมี logic เพื่อใช้ในการอ่านค่าว่ามี Topic ที่เป็น logServer มั้ย ถ้ามีก็บันทึกเก็บไว้ เพื่อในการส่งรอบต่อไปจะไม่ Broadcast ไปยังทุก Node ละ แต่จะระบุปลายทางเป็น NodeID ของ Server แทน

```

/dev/cu.SLAB_USBtoUART
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
0x8 espWifiEventCb(): SYSTEM_EVENT_AP_STA_CONNECTED
0x8 New AP connection incoming
0x8 meshConnectedCb(): we are AP
0x8 findConnection(2752206217): did not find connection
0x8 newConnectionTask(): adding 2752206217 now= 3889582447
logClient: Received from 3893419003 msg={"nodeName":"mco-03","NodeID":3893419003,"random value":163}
logClient: Received from 572025877 msg={"nodeName":"Wemos-01","NodeID":572025877,"random value":1}
logClient: Received from 2136600749 msg={"nodeName":"mco-03","NodeID":2136600749,"random value":49}
logClient: Received from 2487009679 msg={"nodeName":"Wemos-t1","NodeID":2487009679,"Temp":"24.10C","Humidity":"38.10%"}
logClient: Received from 983688460 msg={"nodeName":"esp-01","NodeID":983688460,"random value":20}
{"nodeName":"mco-04","NodeID":3896664414,"random value":77}
logClient: Received from 3163409689 msg={"nodeName":"Wemos-02","NodeID":3163409689,"random value":30}
logClient: Received from 3896055851 msg={"nodeName":"mco-02","NodeID":3896055851,"random value":170}
logClient: Received from 3896713776 msg={"nodeName":"mco-01","NodeID":3896713776,"random value":128}
logClient: Received from 2488971084 msg={"nodeName":"mco-11","NodeID":2488971084,"Temp":"24.60C","Humidity":"36.60%"}
logClient: Received from 3163408206 msg={"nodeName":"Wemos-04","NodeID":3163408206,"random value":37}
logClient: Received from 3893419003 msg={"nodeName":"mco-03","NodeID":3893419003,"random value":174}
logClient: Received from 572025877 msg={"nodeName":"Wemos-01","NodeID":572025877,"random value":8}
logClient: Received from 2136600749 msg={"nodeName":"Wemos-03","NodeID":2136600749,"random value":162}
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
{"nodeName":"mco-04","NodeID":3896664414,"random value":81}

```

หลังจากที่ทุก Node ได้รับ Server NodeID จาก Broadcast Message ที่ส่งมาจาก Server Node แล้ว แทนที่ทุก Node จะส่งข้อมูลออกไปยังทุกๆ Node ก็จะส่งไปยัง Server Node แทนดังจะเห็นได้จากรูปข้างล่างที่ได้รับ Broadcast Message เข้ามาจะมีเฉพาะข้อความที่มาจาก Server Node เท่านั้น เพราะตัวอื่นๆส่งไปที่ Server Node หมดแล้ว

```

/dev/cu.SLAB_USBtoUART
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
{"nodeName":"mco-04","NodeID":3896664414,"random value":123}
{"nodeName":"mco-04","NodeID":3896664414,"random value":62}
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
{"nodeName":"mco-04","NodeID":3896664414,"random value":124}
{"nodeName":"mco-04","NodeID":3896664414,"random value":76}
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
{"nodeName":"mco-04","NodeID":3896664414,"random value":150}
{"nodeName":"mco-04","NodeID":3896664414,"random value":55}
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
{"nodeName":"mco-04","NodeID":3896664414,"random value":105}
{"nodeName":"mco-04","NodeID":3896664414,"random value":88}
logClient: Received from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
logServer detected!!!
Handled from 2752206217 msg={"topic":"logServer","nodeId":"2752206217"}
{"nodeName":"mco-04","NodeID":3896664414,"random value":24}
{"nodeName":"mco-04","NodeID":3896664414,"random value":46}

```

คราวนี้เรามาลองดูในฝั่งของ Server กันบ้างว่าเมื่อทุก Node ส่งข้อมูลมาที่ Server Node แล้วข้อมูลที่ได้จะเป็นลักษณะไหนกัน จากรูปจะเห็นว่า ข้อมูลจะได้ครบเหมือนตอนที่เราดูของ Node ใด Node หนึ่งตอนที่ยังไม่มี Server Node เนื่องจากคราวนี้ทุก Node ใน Mesh Network ส่งมาที่จุดๆเดียว ก็เลยจะได้ข้อมูลครบ และตัว Server Node เองก็ยังคง Broadcast Message ออกไปเรื่อยๆเพื่อให้ทุกคนรู้ว่า ฉันทอยู่ที่ไหนะ ถ้าจะส่งข้อมูลมาหา ให้ส่งมาที่ไหน

```

/dev/cu.SLAB_USBtoUART
logServer: Received from 2487009679 msg={"nodename":"Wemos-t1","NodeID":2487009679,"Temp":"23.60C","Humidity":"38.20%"}
logServer: Received from 3163408206 msg={"nodename":"Wemos-04","NodeID":3163408206,"random value":139}
logServer: Received from 3893419003 msg={"nodename":"mcu-03","NodeID":3893419003,"random value":126}
logServer: Received from 572025877 msg={"nodename":"Wemos-01","NodeID":572025877,"random value":82}
logServer: Received from 3896664414 msg={"nodename":"mcu-04","NodeID":3896664414,"random value":162}
{"topic":"logServer","nodeId":2752206217}
logServer: Received from 2488971084 msg={"nodename":"mcu-t1","NodeID":2488971084,"Temp":"24.30C","Humidity":"36.50%"}
logServer: Received from 2136600749 msg={"nodename":"Wemos-03","NodeID":2136600749,"random value":92}
logServer: Received from 983688460 msg={"nodename":"esp-01","NodeID":983688460,"random value":154}
logServer: Received from 3163409689 msg={"nodename":"Wemos-02","NodeID":3163409689,"random value":45}
logServer: Received from 3896055851 msg={"nodename":"mcu-02","NodeID":3896055851,"random value":100}
logServer: Received from 3896713776 msg={"nodename":"mcu-01","NodeID":3896713776,"random value":126}
logServer: Received from 3163408206 msg={"nodename":"Wemos-04","NodeID":3163408206,"random value":20}
logServer: Received from 3893419003 msg={"nodename":"mcu-03","NodeID":3893419003,"random value":41}
logServer: Received from 572025877 msg={"nodename":"Wemos-01","NodeID":572025877,"random value":152}
logServer: Received from 2487009679 msg={"nodename":"Wemos-t1","NodeID":2487009679,"Temp":"23.60C","Humidity":"38.20%"}
logServer: Received from 3896664414 msg={"nodename":"mcu-04","NodeID":3896664414,"random value":97}
logServer: Received from 2136600749 msg={"nodename":"Wemos-03","NodeID":2136600749,"random value":67}
logServer: Received from 2488971084 msg={"nodename":"mcu-t1","NodeID":2488971084,"Temp":"24.40C","Humidity":"36.50%"}
logServer: Received from 983688460 msg={"nodename":"esp-01","NodeID":983688460,"random value":153}
logServer: Received from 3163409689 msg={"nodename":"Wemos-02","NodeID":3163409689,"random value":45}
logServer: Received from 3896055851 msg={"nodename":"mcu-02","NodeID":3896055851,"random value":104}
logServer: Received from 3896713776 msg={"nodename":"mcu-01","NodeID":3896713776,"random value":33}
logServer: Received from 3163408206 msg={"nodename":"Wemos-04","NodeID":3163408206,"random value":50}
logServer: Received from 3893419003 msg={"nodename":"mcu-03","NodeID":3893419003,"random value":142}
logServer: Received from 572025877 msg={"nodename":"Wemos-01","NodeID":572025877,"random value":179}
logServer: Received from 3896664414 msg={"nodename":"mcu-04","NodeID":3896664414,"random value":8}
{"topic":"logServer","nodeId":2752206217}
Server เริ่ม Broadcast NodeID ของตัวเองได้สำเร็จ
logServer: Received from 2487009679 msg={"nodename":"Wemos-t1","NodeID":2487009679,"Temp":"23.60C","Humidity":"38.20%"}
logServer: Received from 2136600749 msg={"nodename":"Wemos-03","NodeID":2136600749,"random value":21}

```

ส่งค่าได้สมบูรณ์ 11 Node

☐ Autocroll    No line ending    115200 baud    Clear output