

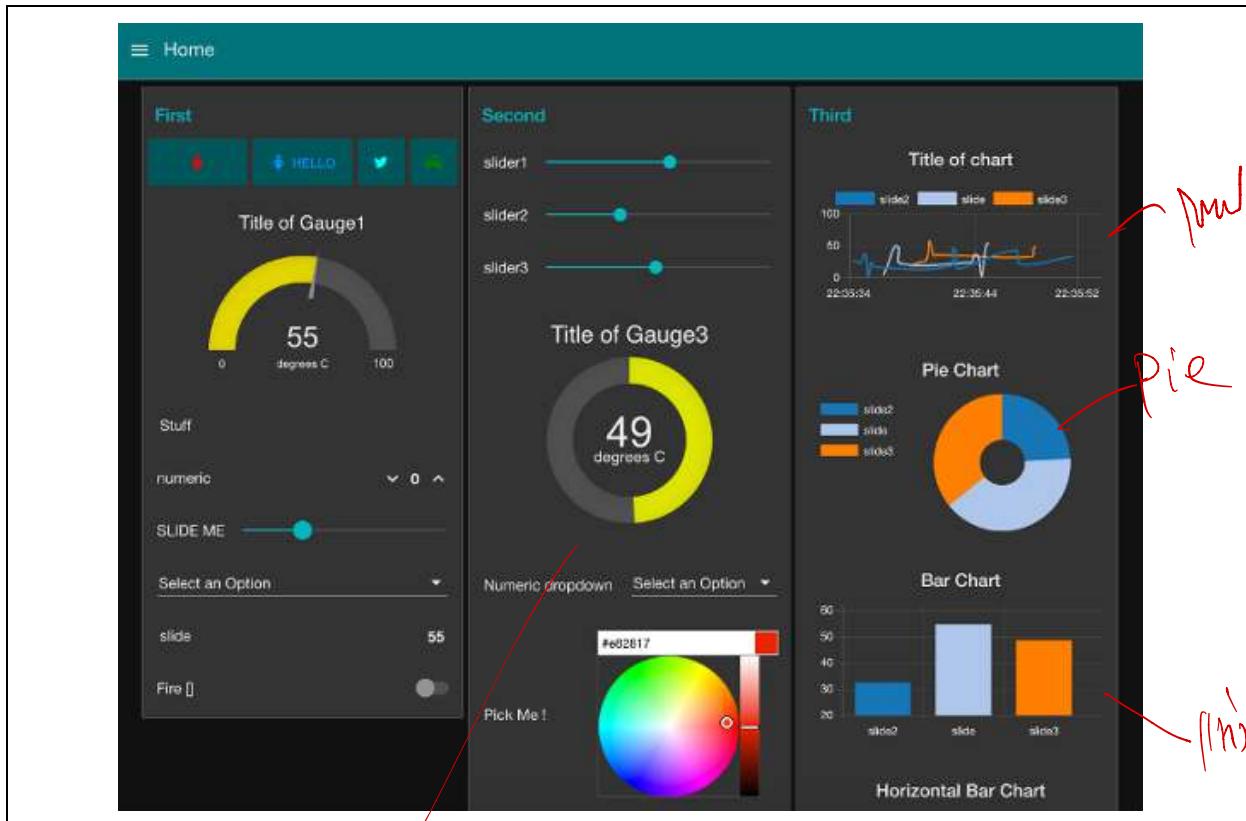
การสร้าง MQTT Server บน Raspberry Pi เพื่อใช้งาน Chatbot LINE ในฟาร์มอัจฉริยะ Chatbot LINE from Raspberry Pi MQTT Server for Smart Farming	
4/4 – LINE Bot on Raspberry PI <ul style="list-style-type: none"> การสร้าง UI ด้วย Node-RED การสร้าง UI ด้วย Node-RED สำหรับฟาร์มอัจฉริยะ การโปรแกรมเพื่อแจ้งเตือนผ่าน LINE ด้วย LINE API Python การโปรแกรมเพื่อแจ้งเตือนผ่าน LINE ด้วย Node-RED การโปรแกรมเพื่อติดต่อกับผู้ใช้งานผ่าน LINE คำถ้ามท้ายบทเพื่อทดสอบความเข้าใจ 	

1/6 - การสร้าง UI ด้วย Node-RED

<https://medium.com/mmp-li/node-red-dashboard-ทำเว็บด้วย-node-red-โดยไม่เสียเงินได้ลักคัว-ฉบับปี-2018-23345af6bf5d>

<https://medium.com/mmp-li/เริ่มต้นใช้งาน-node-red-ฉบับปี-2018-3fc5ed140f9>

<https://nerdiy.de/en/nodered-systemdaten-des-raspberrypis-auf-dem-dashboard-anzeigen/>



Node-RED Dashboard

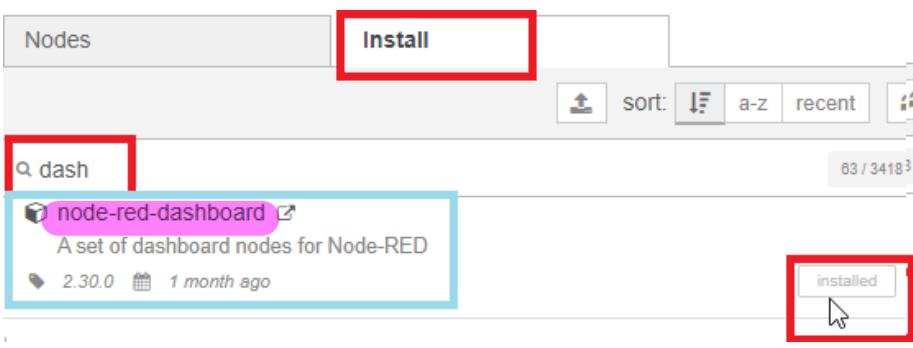
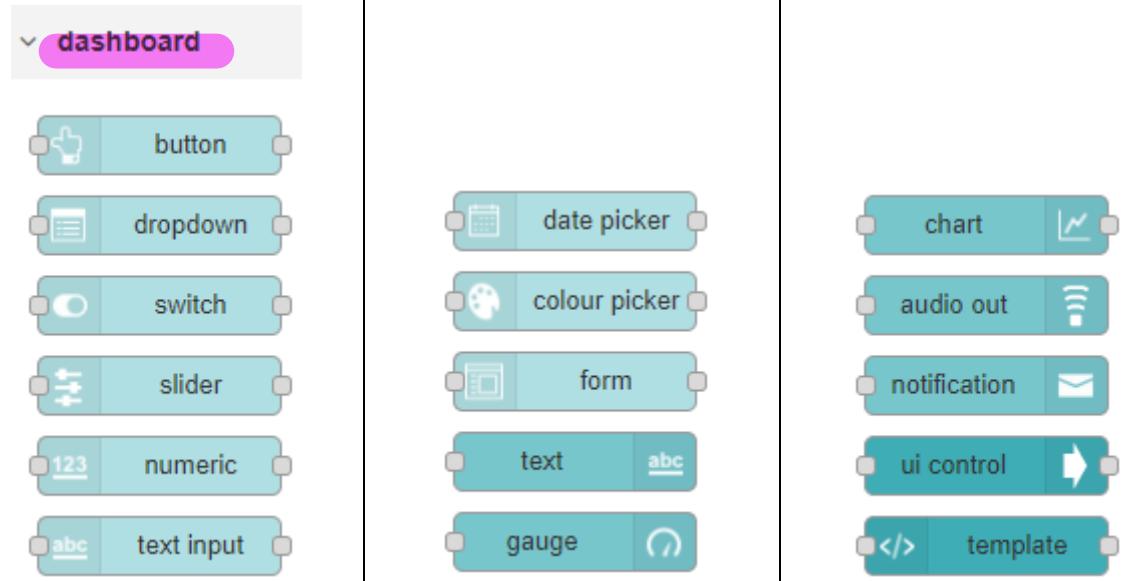
gauge

คือ หน้าเว็บโปรแกรม ที่ใช้สำหรับควบคุม/สั่งเกตุ ค่าต่างๆที่เราสนใจ ไม่ว่าจะเป็นการสั่งงานผ่าน MQTT ,การพื้นต์กราฟเพื่อแสดงให้เห็นถึงความเปลี่ยนแปลง, การแสดงสถานะของอุปกรณ์ต่างๆ ที่สำคัญคือ “Node-RED Dashboard ทำงานแบบ Real Time ไม่ต้องรีเฟรชหน้าเว็บเวลามีการอัพเดท”

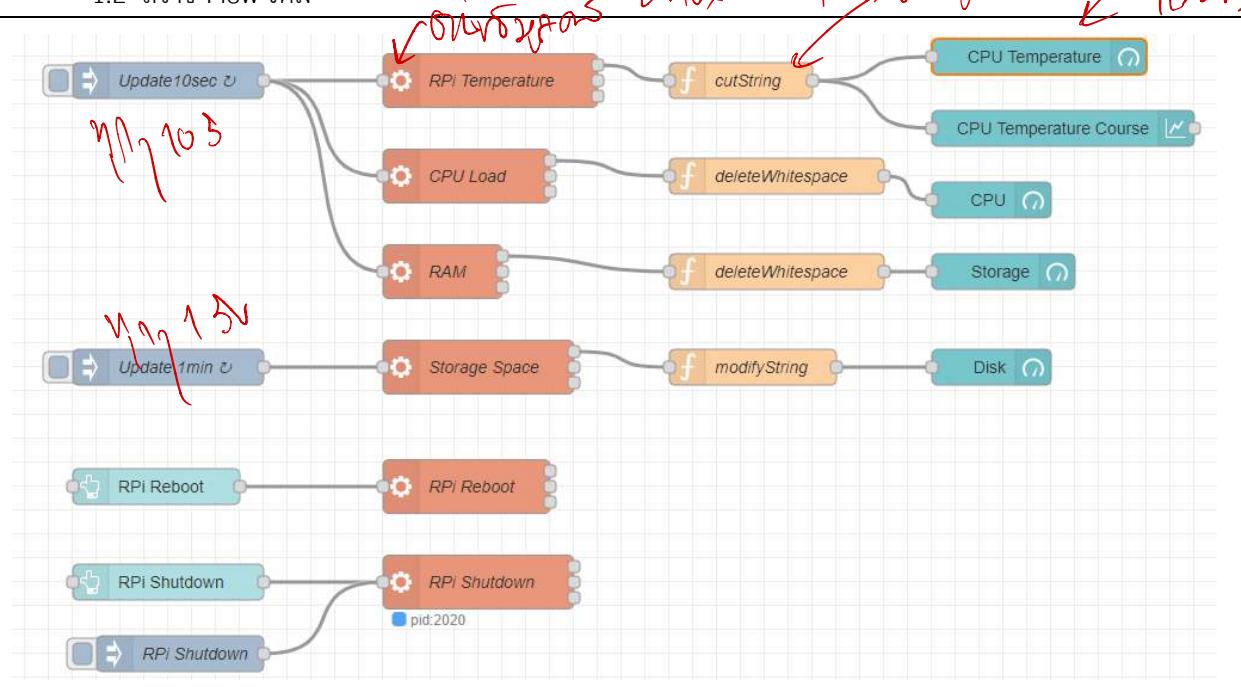
Lab401 – Node-RED Dashboard

1. แสดงค่าข้อมูลของ Raspberry Pi บน Node-RED Dashboard

1.1 เพิ่มหนด กลุ่ม Node-RED UI

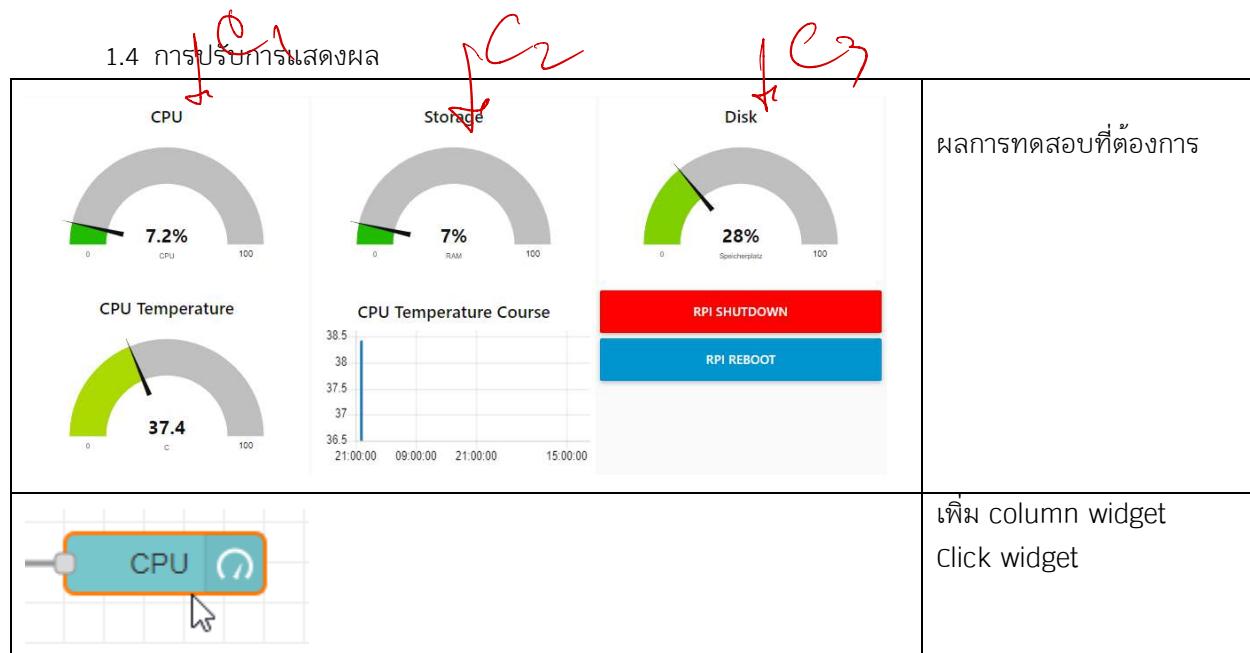
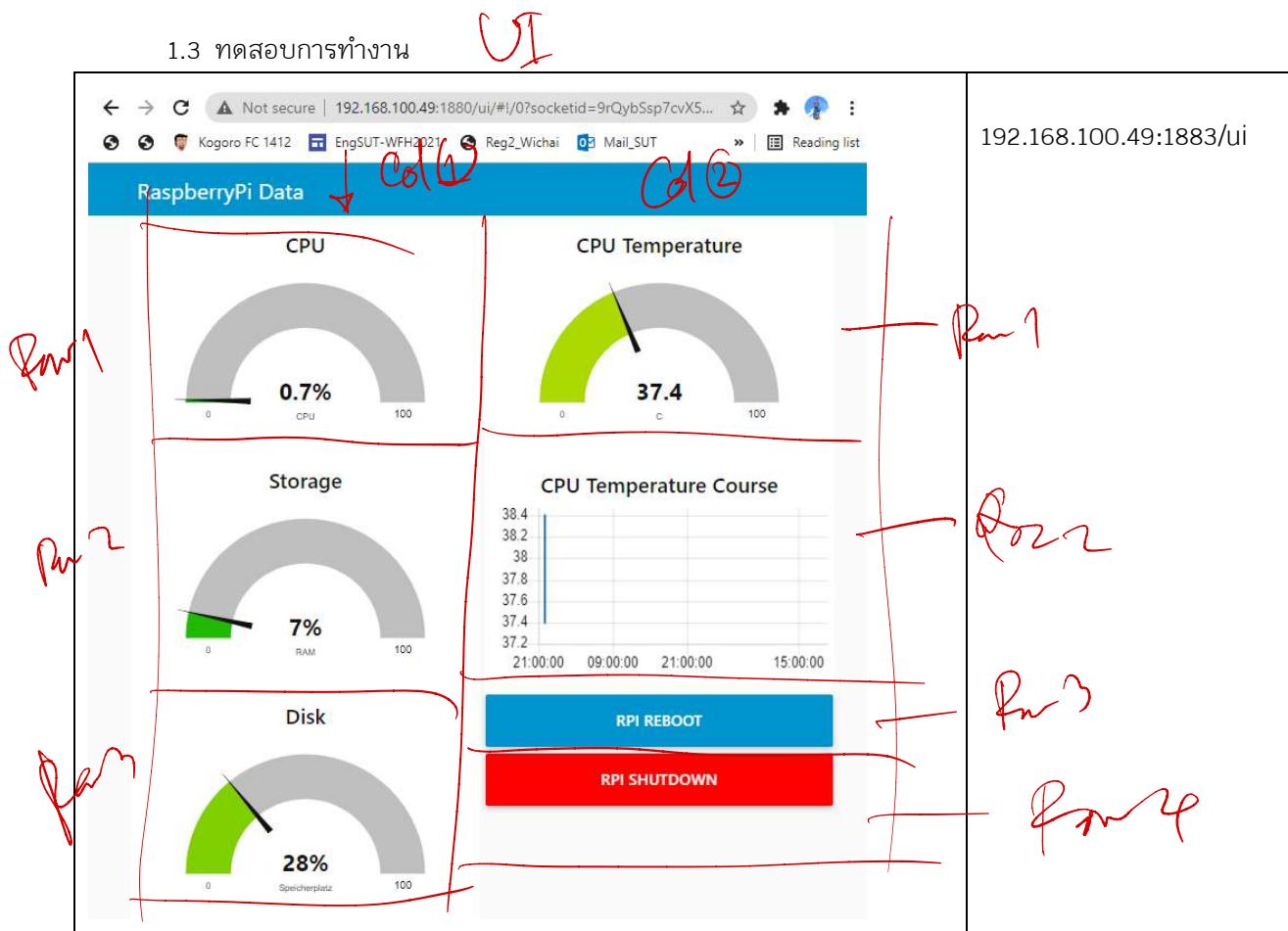
	Manage palette Tab = Install Filter = dash ตรวจสอบ ชื่อ node-red-dashboard V2.30.0 Install
	
	

1.2 สร้าง Flow ใหม่



< Code ตาม Lab401a_RPi_monitor >

```
[{"id": "8d8ede65.bc90c", "type": "tab", "label": "Flow 1", "disabled": false, "info": ""}, {"id": "8aa024f3.7775b8", "type": "exec", "z": "8d8ede65.bc90c", "command": "vcgencmd measure_temp", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "RPi Temperature"}, {"x": "510", "y": "200", "wires": [[[{"z": "75f6e734.61bd28"}, {}, {}]]]}, {"id": "66bd6fde.6aa98", "type": "inject", "z": "8d8ede65.bc90c", "name": "Update10sec", "props": [{"p": "topic", "vt": "str"}], "repeat": "10", "crontab": "", "once": false, "onceDelay": "", "topic": "", "payload": "", "payloadType": "date", "x": "240", "y": "200", "wires": [{"z": "8aa024f3.7775b8", "x": "37d6ed88.286ca2", "y": "200"}, {"z": "s21184bd.4907ac", "x": "37d6ed88.286ca2", "y": "220"}]}, {"id": "75f6e734.61bd28", "type": "function", "z": "8d8ede65.bc90c", "name": "cutString", "func": "str = msg.payload\nmsg = str.substring(5,9)\nreturn msg", "outputs": 1, "noerr": 0, "x": "720", "y": "200", "wires": [{"z": "bda18f6c.a0d7c", "x": "1542ed.df499", "y": "200"}]}, {"id": "a6e3e16b.a9dc9", "type": "exec", "z": "8d8ede65.bc90c", "command": "sudo reboot", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "RPi Reboot"}, {"x": "490", "y": "450", "wires": [{"z": "52855703.cdec98", "x": "8d8ede65.bc90c", "y": "450"}]}, {"id": "37d6ed88.286ca2", "type": "exec", "z": "8d8ede65.bc90c", "command": "top -d 1 -b -n2 | grep \Cpu(s)\|tail -n 1 | awk \{print $2 + $4\}", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "CPU Load"}, {"x": "490", "y": "280", "wires": [{"z": "32da55ec.24d81a", "x": "1542ed.df499", "y": "280"}]}, {"id": "s21184bd.4907ac", "type": "exec", "z": "8d8ede65.bc90c", "command": "free | grep Mem | awk \{print 100*($3+$5)/$2\}", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "RAM"}, {"x": "470", "y": "360", "wires": [{"z": "6864ad99.e64814", "x": "1542ed.df499", "y": "360"}]}, {"id": "94a7e731.2c68d8", "type": "exec", "z": "8d8ede65.bc90c", "command": "df -h", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "Storage Space"}, {"x": "500", "y": "440", "wires": [{"z": "d609d0f3.a9bf3", "x": "1542ed.df499", "y": "440"}]}, {"id": "d609d0f3.a9bf3", "type": "function", "z": "8d8ede65.bc90c", "name": "modifyString", "func": "var re = /([0-9]{2})%/\nvar idx = msg.payload.search(re);\nvar str = msg.payload;\nif (idx > 0) {\n    str = msg.payload.substring(idx, idx + 2);}\nmsg.payload = str;\nreturn msg", "outputs": 1, "noerr": 0, "x": "730", "y": "440", "wires": [{"z": "b0a0f8", "x": "1542ed.df499", "y": "440"}]}, {"id": "52855703.cdec98", "type": "exec", "z": "8d8ede65.bc90c", "command": "sudo shutdown -h now", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "RPi Shutdown"}, {"x": "490", "y": "540", "wires": [{"z": "52855703.cdec98", "x": "8d8ede65.bc90c", "y": "540"}]}, {"id": "pid:2020", "type": "ui_button", "z": "8d8ede65.bc90c", "name": "RPi Shutdown", "label": "Shutdown", "group": "5fe5a9b.a8a6158", "order": 3, "width": 0, "height": 0, "passthru": false, "label": "RPI Reboot"}, {"id": "cfd5a087.e95a8", "type": "ui_button", "z": "8d8ede65.bc90c", "name": "RPi Reboot", "label": "Reboot", "group": "5fe5a9b.a8a6158", "order": 4, "width": 0, "height": 0, "passthru": false, "label": "RPI Shutdown"}, {"id": "77393c0b.231ca4", "type": "ui_gauge", "z": "8d8ede65.bc90c", "name": "CPU Temperature", "label": "CPU", "format": "\{(value)\}%", "min": "0", "max": "100", "colors": "#00b500", "#e6e600", "#ca3838", "seg1": "0", "seg2": "100", "x": "230", "y": "540", "wires": [{"z": "a6e3e16b.a9dc9", "x": "1542ed.df499", "y": "540"}]}, {"id": "66bd6fde.6aa98", "type": "inject", "z": "8d8ede65.bc90c", "name": "Update1min", "props": [{"p": "topic", "vt": "str"}], "repeat": "60", "crontab": "", "once": false, "onceDelay": "0.1", "topic": "", "payload": "", "payloadType": "date", "x": "240", "y": "620", "wires": [{"z": "52855703.cdec98", "x": "1542ed.df499", "y": "620"}]}, {"id": "32da55ec.24d81a", "type": "exec", "z": "8d8ede65.bc90c", "command": "top -d 1 -b -n2 | grep \Cpu(s)\|tail -n 1 | awk \{print $2 + $4\}", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "CPU Load"}, {"x": "490", "y": "300", "wires": [{"z": "166432a4a.06ad54", "x": "1542ed.df499", "y": "300"}]}, {"id": "d204ea31.667318", "type": "ui_gauge", "z": "8d8ede65.bc90c", "name": "Memory Usage", "label": "Memory", "format": "\{(value)\}%", "min": "0", "max": "100", "colors": "#00b500", "#e6e600", "#ca3838", "seg1": "0", "seg2": "100", "x": "940", "y": "300", "wires": [{"z": "b0a0f8", "x": "1542ed.df499", "y": "300"}]}, {"id": "6864ad99.e64814", "type": "exec", "z": "8d8ede65.bc90c", "command": "free | grep Mem | awk \{print 100*($3+$5)/$2\}", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "RAM"}, {"x": "940", "y": "360", "wires": [{"z": "166432a4a.06ad54", "x": "1542ed.df499", "y": "360"}]}, {"id": "b0a0f8", "type": "ui_gauge", "z": "8d8ede65.bc90c", "name": "Disk Usage", "label": "Disk", "format": "\{(value)\}%", "min": "0", "max": "100", "colors": "#00b500", "#e6e600", "#ca3838", "seg1": "0", "seg2": "100", "x": "940", "y": "440", "wires": [{"z": "166432a4a.06ad54", "x": "1542ed.df499", "y": "440"}]}, {"id": "166432a4a.06ad54", "type": "exec", "z": "8d8ede65.bc90c", "command": "df -h", "addpay": false, "append": "", "useSpawn": "", "timer": "", "oldrc": false, "name": "Disk"}, {"x": "940", "y": "540", "wires": [{"z": "166432a4a.06ad54", "x": "1542ed.df499", "y": "540"}]}]
```

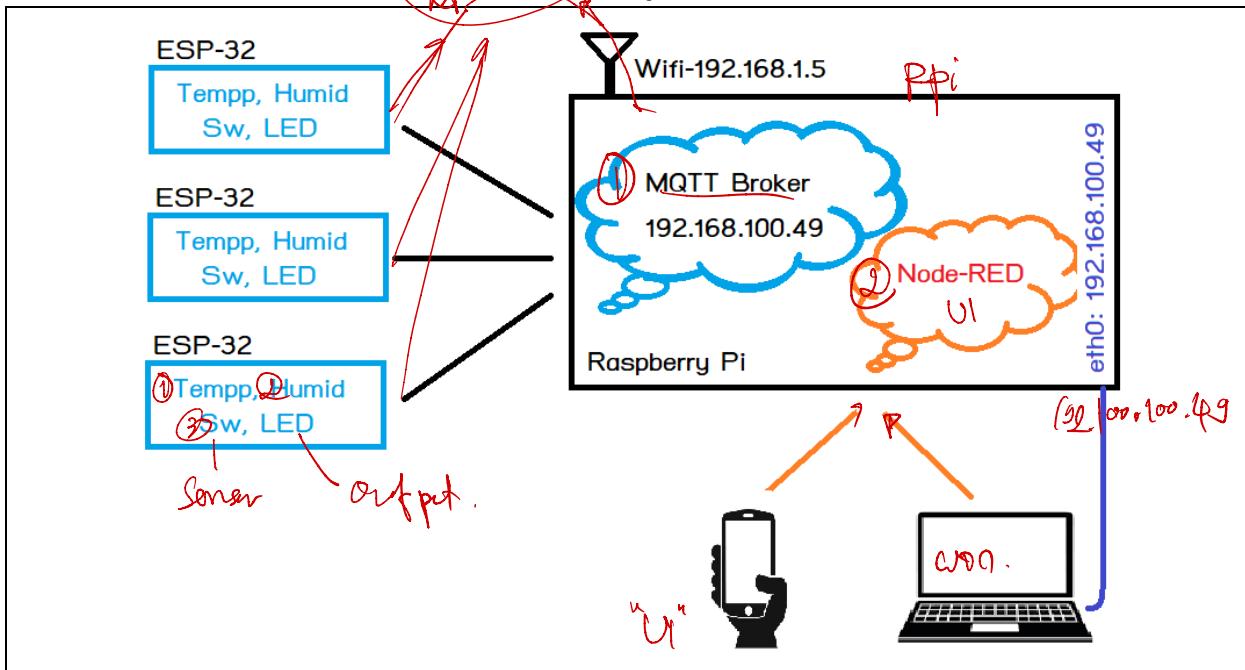


<p>Properties</p> <p>Group: [RaspberryPi Data] 3</p> <p>Size:</p> <p>Type: Add new ui_group...</p>	Add new ui_group
<p>Properties</p> <p>Name: 3</p> <p>Tab: RaspberryPi Data</p> <p>Width: 6</p> <p><input type="checkbox"/> Display group name</p>	Add Colum 3 In dashboard name Not display group
<p>dashboard</p> <p>Layout Site Theme Dashboard</p> <p>Rows: Row 1, Row 2, Row 3, Row 4</p> <p>Groups:</p> <ul style="list-style-type: none"> RaspberryPi Data <ul style="list-style-type: none"> 1 <ul style="list-style-type: none"> CPU CPU Temperature 2 <ul style="list-style-type: none"> Storage CPU Temperature Course 3 <ul style="list-style-type: none"> Disk RPi Shutdown RPi Reboot 	การย้ายตำแหน่ง Dashboard Layout ขยับตำแหน่ง บันลัง

♥ mom mom mom U no nodeRED

2/5 - การสร้าง UI ด้วย Node-RED สำหรับฟาร์มอัจฉริยะ

เป็นตัวอย่างโครงการเพื่อแสดงระบบดูแลควบคุมล่วงของโรงเรือนเลี้ยงไก่ บน Node-RED Dashboard การทำงานจะใช้ ESP32 สร้างข้อมูลที่สื่อสารกับ RPi Server ผ่าน MQTT Protocol ระบบห้องหมุดจะทำงานเป็นระบบปิดภายในฟาร์ม ระบบที่ใช้เทคโนโลยีนี้ดังรูป



Lab402 – Node-RED Dashboard for Smart Farm

1. Raspberry Pi MQTT Server Start

- User Name = mymqtt
- User Password = myraspi



2. ESP-32 Library and Code

ตรวจสอบ PubSubClient by Nick O'Leary V2.8.0

PubSubClient

by Nick O'Leary Version 2.8.0 **INSTALLED**

A client library for MQTT messaging. MQTT is a lightweight messaging protocol ideal for small devices. This library allows you to send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol and can be configured to use the older MQTT 3.1 if needed. It supports all Arduino Ethernet Client compatible hardware, including the Intel Galileo/Edison, ESP8266 and TI CC3000.

Select version

```
#include <WiFi.h>
#include <PubSubClient.h>

const char *ssid = "Mue.Home";
const char *password = "pk1212312121";
const char *mqtt_server = "192.168.1.5";
const int mqtt_port = 1883;
const char *myName = "mymqtt";
const char *myPass = "myraspi";
const char *myTopic = "Test7749";
```

Handwritten annotations:

- A red arrow points from the word 'WiFi' to the line '#include <WiFi.h>'.
- A red arrow points from the text 'IP in MQTT Broker' to the line 'const char *mqtt_server = "192.168.1.5";'.
- A red arrow points from the word 'Topic' to the line 'const char *myTopic = "Test7749";'.

```

#define MSG_BUFFER_SIZE (50)
#define LED0_Pin 2
#define SW0_Pin 0

WiFiClient espClient;
PubSubClient client(espClient);

int SW0_Status = 99;
unsigned long lastMsg = 0;
char msg[MSG_BUFFER_SIZE];

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32_Cient-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str(), myName, myPass)) {
      Serial.println("connected");
      client.subscribe(myTopic);
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void callback(char* topic, byte* payload, unsigned int length)
{ char myPayLoad[50];
  Serial.print("Message arrived [");
  Serial.print(myTopic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
    myPayLoad[i] = payload[i];
    myPayLoad[i + 1] = '\0'; // End of String
  }
  Serial.print("\n--> ");
  Serial.println(myPayLoad);
  myPayLoad[4] = '\0'; // String lessthan 4 Charector
  if ((String)myPayLoad == "ON1") digitalWrite(LED0_Pin, HIGH);
  if ((String)myPayLoad == "OFF1") digitalWrite(LED0_Pin, LOW);
}

void setup() {
  Serial.begin(115200);
  pinMode(LED0_Pin, OUTPUT);
  pinMode(SW0_Pin, INPUT_PULLUP);
  Serial.print("\n Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("\n WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  if (digitalRead(SW0_Pin) != SW0_Status) {
    SW0_Status = digitalRead(SW0_Pin);
    Serial.println(SW0_Status == HIGH ? "Status Switch = OFF" : "Status Switch = ON");
    client.publish(myTopic, (SW0_Status == HIGH ? "SW_OFF" : "SW_ON"));
    delay(100);
  }

  unsigned long now = millis();
  if (now - lastMsg > 5000) {
    lastMsg = now;
    float Temp = random(3000, 5000) / 100.0;
    float Humid = random(6000, 8000) / 100.0;
    sprintf(msg, MSG_BUFFER_SIZE, "Temp=%0.2f,Humid=%0.2f", Temp, Humid);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(myTopic, msg);
  }
}

```

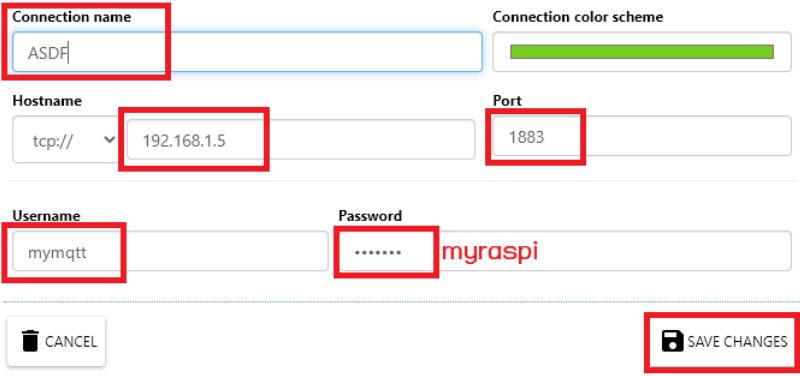
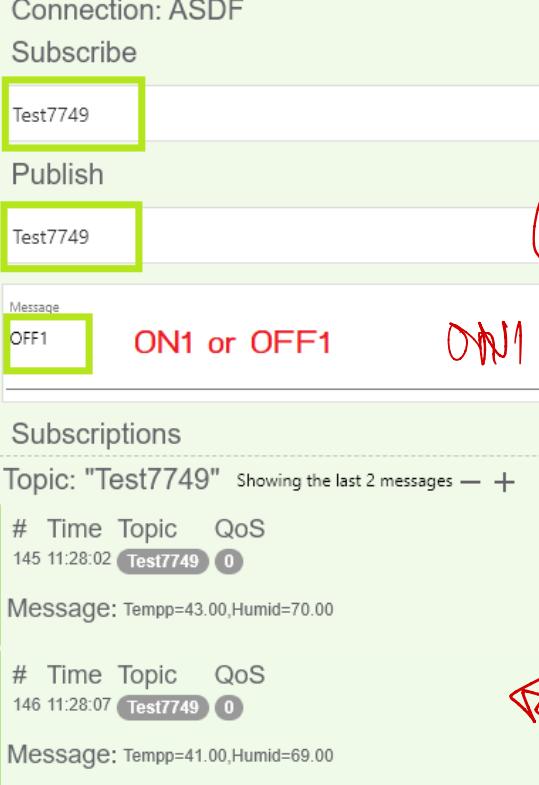
④ Output

① In No switch ON

② Status Switch OFF

① Random

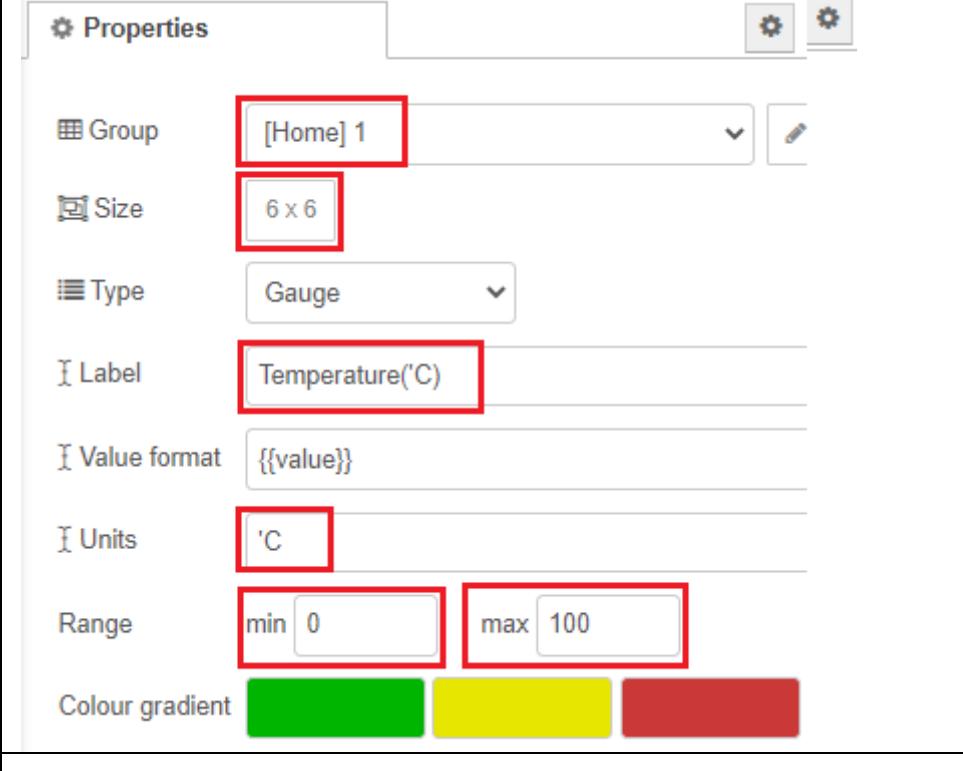
② Temp Humid

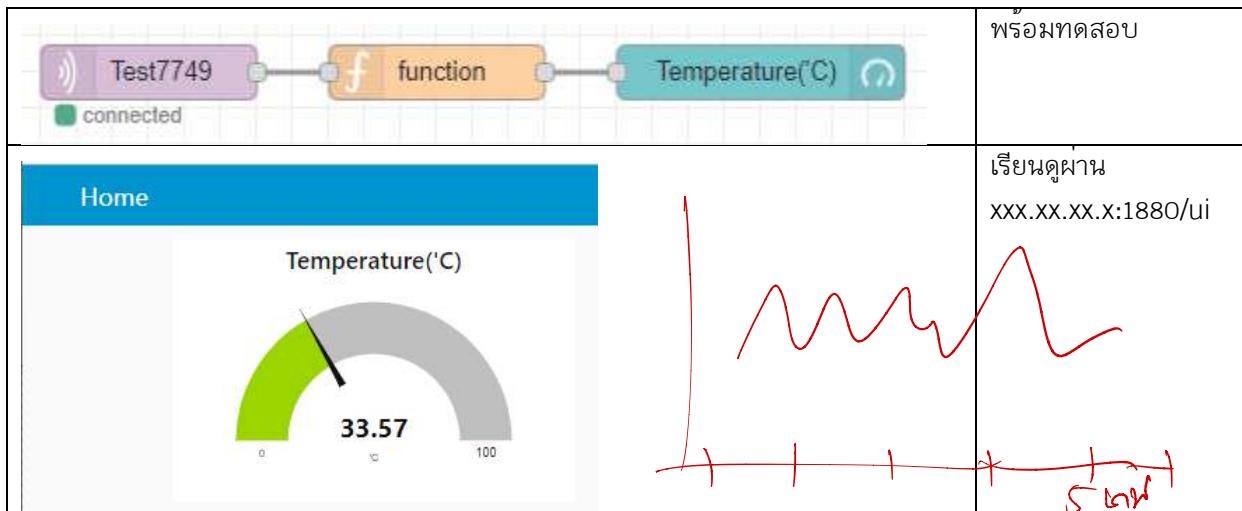
	Name = อะไรก็ได้ Host = Wifi-IP:1883 UName = mymqtt UPass = myraspi Save
	สีแดง ไม่ทำงาน สีเขียว พร้อมทำงาน
	การทดสอบ Topic = Test7749

0013

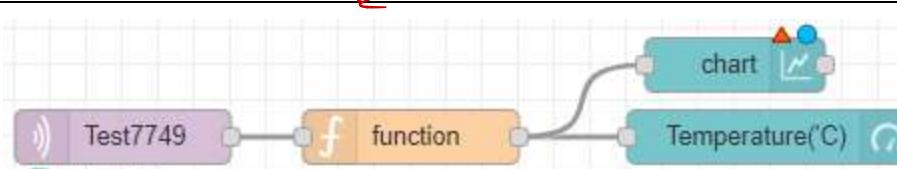
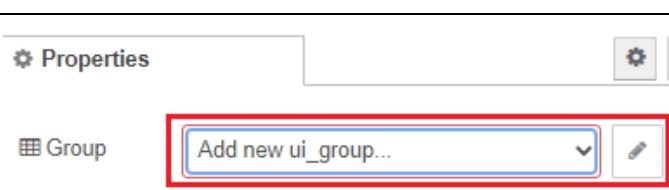
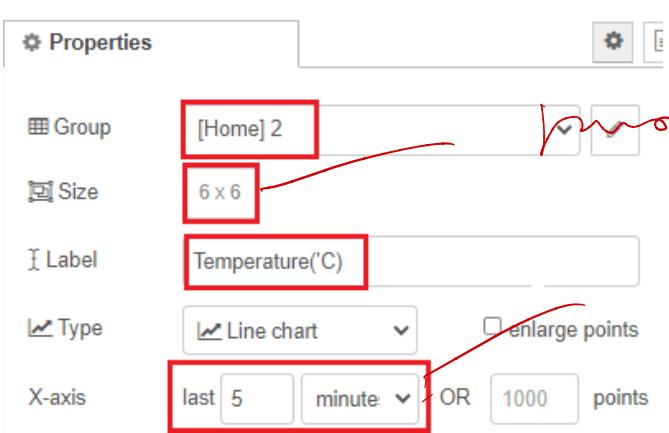
3. Node-RED UI Mission 1/4 – Temperature Gauge

<p>◆ Name MQTT_priv</p> <p>Connection Security Messages</p> <p>● Server 127.0.0.1 Port 1883</p>	Mqtt node
<p>Connection Security</p> <p>● Username mymqtt</p> <p>● Password myraspi</p>	Mqtt node
<p>⚙ Properties</p> <p>● Server MQTT_priv</p> <p>■ Topic Test7749</p>	Mqtt node
<p>⚙ Properties</p> <p>■ Group Add new ui_group... Edit</p>	Gauge node
<p>⚙ Properties</p> <p>◆ Name 1</p> <p>■ Tab Home Edit</p> <p>↔ Width 6</p>	Gauge node

	Gauge node
<pre> 1 // Tempp=38.20,Humid=73.52 2 var output = msg.payload.split(","); 3 var sTempp = output[0].split("="); 4 msg.payload = sTempp[1] 5 6 7 return msg; </pre> <p>// Tempp=38.20,Humid=73.52 var output = msg.payload.split(","); var sTempp = output[0].split("=". msg.payload = sTempp[1] return msg;</p>	Function Node

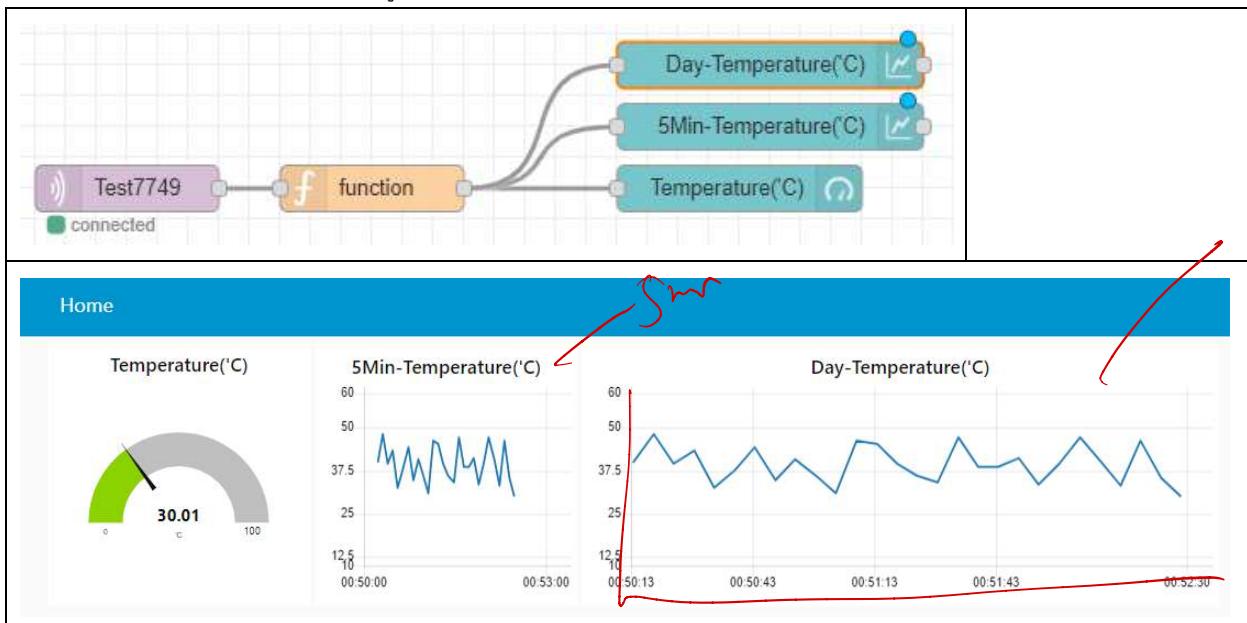


4. Node-RED UI Mission 2/4 – Temperature Gauge and line chart

	Add Chart
	Click chart Add ui group
	Column 2 in Home
	<i>move down</i> <i>Size</i> <i>Temperature('C)</i> <i>Line chart</i> <i>X-axis</i>

<pre> graph LR T[Test7749] --> F(function) F --> T1[Temperature(C)] F --> T2[Temperature(C)] </pre>	พร้อมทดสอบ
<p>Home</p> <p>Temperature('C)</p> <p>Temperature('C)</p> <p>42.1</p> <p>0 100</p> <p>50 45 40 35 30</p> <p>00:41:00 00:47:00</p>	ผลการทำงาน

5. เพิ่มอีกราฟเพื่อบันทึกข้อมูลที่ယาวขึ้น



```

[{"id": "8d8ede65.bc90c", "type": "tab", "label": "Flow 1", "disabled": false, "info": ""}, {"id": "61861bb9.64de44", "type": "blynk-ws-client", "name": "myBlynk", "path": "wss://blynk-cloud.com/websockets", "key": "CQwbMwHyedfb0s28DmHMAkj2IDbul7", "dbg_all": false, "dbg_read": false, "dbg_write": false, "dbg_notify": false, "dbg_mail": false, "dbg_prop": false, "dbg_sync": false, "dbg_bridge": false, "dbg_low": false, "dbg_pins": "", "multi_cmd": false, "proxy_type": "no", "proxy_url": "", "enabled": true}, {"id": "7436b594.5bbf3c", "type": "mqtt-broker", "name": "MQTT_priv", "broker": "127.0.0.1", "port": "1883", "clientid": "", "usetls": false, "protocolVersion": "4", "keepalive": "60", "cleansession": true, "birthTopic": "", "birthQos": "0", "birthPayload": "", "birthMsg": {}, "closeTopic": "", "closeQos": "0", "closePayload": "", "closeMsg": {}, "willTopic": "", "willQos": "0", "willPayload": "", "willMsg": {}, "sessionExpiry": ""}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": false}, "customTheme": {"name": "Untitled Theme 1", "default": "#4B7930", "baseColor": "#4B7930", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}, "themeState": {"baseColor": {"default": "#0094CE", "value": "#0094CE", "edited": false}, "page-titlebarBackgroundColor": {"value": "#0094CE", "edited": false}, "page-backgroundColor": {"value": "#fafafa", "edited": false}, "page-sidebarBackgroundColor": {"value": "#ffffff", "edited": false}, "group-textColor": {"value": "#1bbfff", "edited": false}, "group-borderColor": {"value": "#ffffff", "edited": false}, "group-backgroundColor": {"value": "#ffffff", "edited": false}, "widget-textColor": {"value": "#111111", "edited": false}, "widget-backgroundColor": {"value": "#0094ce", "edited": false}, "widget-borderColor": {"value": "#ffffff", "edited": false}, "base-font": {"value": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}}
```

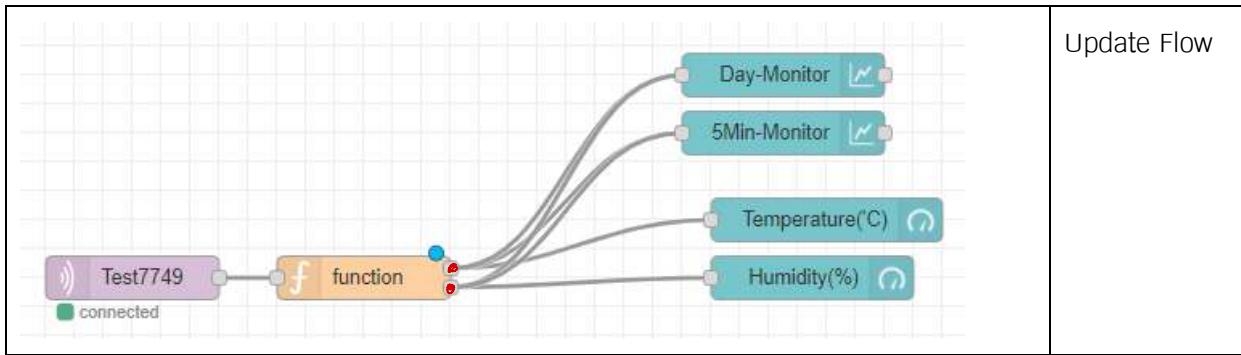
```

serif"}}, "angularTheme": {"primary": "indigo", "accents": "blue", "warn": "red", "background": "grey", "palette": "light"}}, "site": {"name": "Node-RED Dashboard", "hideToolbar": "false", "allowSwipe": "false", "lockMenu": "false", "allowTempTheme": "true", "dateFormat": "DD/MM/YYYY", "sizes": [{"sx": 48, "sy": 48, "gx": 6, "gy": 6, "cx": 6, "cy": 6, "px": 0, "py": 0}], "id": "67e2957.e504f6c", "type": "ui_tab", "name": "Home", "icon": "dashboard", "disabled": false, "hidden": false}, {"id": "29de1dd2.698e12", "type": "ui_group", "name": "me", "tab": "67e2957.e504f6c", "order": 1, "disp": false, "width": "5%", "collapse": false}, {"id": "67163726.764658", "type": "ui_group", "name": "2", "tab": "67e2957.e504f6c", "order": 2, "disp": false, "width": "5%", "collapse": false}, {"id": "70b4a6cc.14fd78", "type": "ui_group", "name": "3", "tab": "67e2957.e504f6c", "order": 3, "disp": false, "width": "12%", "collapse": false}, {"id": "23e1297.90f16e", "type": "mqtt_in", "x": "2", "y": "8d8ede65.bc90c", "name": "", "topic": "Test7749", "qos": "2", "datatype": "auto", "broker": "7436b594.5bbf3c", "nl": false, "rap": true, "rh": 0, "x": 160, "y": 340, "wires": [{"id": "ddc518f7.44c468"}]}, {"id": "ddc518f7.44c468", "type": "function", "x": "2", "y": "8d8ede65.bc90c", "name": "", "func": "\n// Tempp=38.20, Humid=73.52\nvar output = msg.payload.split(',')\nvar sTempp = output[0].split('=')\nmsg1.payload = sTempp[1]\nmsg1.topic = 'Temperature'\nvar sHumid = output[1].split('=')\nmsg2.payload = sHumid[1]\nmsg2.topic = 'Humidity'\nreturn [msg1, msg2];", "width": "5%", "height": "5%", "gtype": "gauge", "title": "Temperature(C)", "label": "C", "format": "{{value}}", "min": 0, "max": 100, "colors": [{"#00b500", "#e6e600", "#ca3838"}], "seg1": "", "seg2": "", "x": 560, "y": 340, "wires": []}, {"id": "c31e4824.aebf28", "type": "ui_group", "name": "4", "tab": "67163726.764658", "order": 4, "width": "5%", "height": "5%", "label": "5Min-Temperature(C)", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "10", "ymax": "60", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColor": false, "useUTC": false, "colors": [{"#1f77b4", "#aecd8b", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"}], "outputs": 1, "useDifferentColor": false, "x": 580, "y": 300, "wires": []}, {"id": "8eff3667.95dcf8", "type": "ui_group", "name": "5", "tab": "67163726.764658", "order": 5, "width": "5%", "height": "5%", "label": "5Min-Temperature(C)", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "10", "ymax": "60", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "useUTC": false, "colors": [{"#1f77b4", "#aecd8b", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"}], "outputs": 1, "useDifferentColor": false, "x": 580, "y": 260, "wires": []}], [{"id": "29de1dd2.698e12", "x": 1, "y": 1}, {"id": "67163726.764658", "x": 2, "y": 1}, {"id": "70b4a6cc.14fd78", "x": 3, "y": 1}, {"id": "23e1297.90f16e", "x": 4, "y": 1}, {"id": "ddc518f7.44c468", "x": 2, "y": 2}, {"id": "c31e4824.aebf28", "x": 4, "y": 2}, {"id": "8eff3667.95dcf8", "x": 4, "y": 3}]}

```

6. Node-RED UI Mission 3/4 – Temperature Gauge and line chart from 2 Data

<pre> 1 // Tempp=38.20, Humid=73.52 2 [o] [1] 3 var msg1 = {}; 4 var msg2 = {}; 5 6 var output = msg.payload.split(","); 7 8 var sTempp = output[0].split("="); 9 msg1.payload = sTempp[1]; 10 msg1.topic = 'Temperature'; 11 12 var sHumid = output[1].split("="); 13 msg2.payload = sHumid[1]; 14 msg2.topic = 'Humidity'; 15 16 return [msg1, msg2]; == == </pre>	Update Function
<pre> // Tempp=38.20, Humid=73.52 var msg1 = {}; var msg2 = {}; var output = msg.payload.split(","); var sTempp = output[0].split("="); msg1.payload = sTempp[1]; msg1.topic = 'Temperature'; var sHumid = output[1].split("="); msg2.payload = sHumid[1]; msg2.topic = 'Humidity'; return [msg1, msg2]; </pre>	



7. Node-RED UI Mission 4/4 – with monitor and control

node-red-contrib-ui-led

A simple LED status indicator for the Node-RED Dashboard

0.4.9 5 months ago

install

The Node-RED flow consists of the following components:

- Sensors:** Temperature (C) and Humidity (%)
- Monitors:** Day-Monitor, 5Min-Monitor, Input Monitor
- Functions:** Two function nodes connected to the sensors.
- Topics:** Test7749 (connected)
- Control:** Output Control node connected to Test7749.
- Annotations:**
 - Red circle labeled ① Top Task
 - Red circle labeled ② Start Sensor
 - Red arrow labeled publish pointing to the Output Control node.
 - Red arrows labeled Input sink and Control pointing to the Input Monitor node.
 - Red arrow labeled Day-Monitor pointing to the Day-Monitor node.

Dashboard View:

The dashboard displays the following data:

- Temperature (C): 35.11
- Humidity (%): 76.76
- 5Min-Monitor: Shows Temperature and Humidity over time (15:17:00 to 15:23:00).
- Day-Monitor: Shows Temperature and Humidity over time (13:29:00 to 15:23:00).
- Input Monitor: Shows a green button labeled "Input sink" and a blue switch labeled "control".

```
[{"id": "8d8ede65.bc90c", "type": "tab", "label": "Flow 1", "disabled": false, "info": ""}, {"id": "7436b594.5bbf3c", "type": "mqtt-broker", "name": "MQTT_priv", "broker": "127.0.0.1", "port": "1883", "clientId": "", "useTLS": false, "protocolVersion": "4", "keepalive": "60", "cleanSession": true, "birthTopic": "", "birthQos": "0", "birthPayload": "", "birthMsg": {}, "closeTopic": "", "closeQos": "0", "closePayload": "", "closeMsg": {}, "willTopic": "", "willQos": "0", "willPayload": "", "willMsg": {}, "sessionExpiry": ""}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Output Control", "x": 100, "y": 100, "w": 150, "h": 50}, {"id": "7436b594.5bbf3c", "type": "mqtt-broker", "name": "MQTT_priv", "broker": "127.0.0.1", "port": "1883", "clientId": "", "useTLS": false, "protocolVersion": "4", "keepalive": "60", "cleanSession": true, "birthTopic": "", "birthQos": "0", "birthPayload": "", "birthMsg": {}, "closeTopic": "", "closeQos": "0", "closePayload": "", "closeMsg": {}, "willTopic": "", "willQos": "0", "willPayload": "", "willMsg": {}, "sessionExpiry": ""}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Test7749", "x": 100, "y": 150, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Input Monitor", "x": 300, "y": 150, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Day-Monitor", "x": 300, "y": 200, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "5Min-Monitor", "x": 300, "y": 250, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Temperature (C)", "x": 300, "y": 300, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Humidity (%)", "x": 300, "y": 350, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "function", "x": 200, "y": 400, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "function", "x": 200, "y": 450, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Test7749", "x": 100, "y": 500, "w": 150, "h": 50}, {"id": "61871ae5.830a64", "type": "ui_base", "theme": {"name": "theme-light", "lightTheme": {"default": "#0094CE", "baseColor": "#0094CE", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif", "edited": true, "reset": false}, "darkTheme": {"default": "#097479", "baseColor": "#097479", "baseFont": "-apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif"}}, "label": "Output Control", "x": 100, "y": 550, "w": 150, "h": 50}]


```

```

serif","edited":false},"customTheme":{"name":"Untitled Theme 1","default":"#4B7930","baseColor":"#4B7930","baseFont":'-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"},"themeState":{"baseColor":{"default":"#0094CE","value":"#0094CE","edited":false},"page-titlebar-backgroundColor":{"value":"#0094CE","edited":false},"page-backgroundColor":{"value":"#fafafa","edited":false},"page-sidebar-backgroundColor":{"value":"#ffffff","edited":false},"group-textColor":{"value":"#1bbfff","edited":false},"group-borderColor":{"value":"#ffffff","edited":false},"group-backgroundColor":{"value":"#ffffff","edited":false},"widget-textColor":{"value":"#111111","edited":false},"widget-backgroundColor":{"value":"#0094ce","edited":false},"widget-borderColor":{"value":"#ffffff","edited":false},"base-font":{ "value": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif" }}, "angularTheme":{ "primary": "indigo", "accents": "blue", "warn": "red", "background": "grey", "palette": "light" }}, "site":{ "name": "No de-RED
Dashboard", "hideToolbar": "false", "allowSwipe": "false", "lockMenu": "false", "allowTempTheme": "true", "dateFormat": "DD/MM/YYYY", "sizes": { "sx": 48, "sy": 48, "gx": 6, "cy": 6, "px": 0, "py": 0 } }, {"id": "67e2957.e504f6c", "type": "ui_tab", "name": "Home", "icon": "dashedboard", "disabled": false, "hidden": false}, {"id": "29de1dd2.698e12", "type": "ui_group", "name": "1", "tab": "67e2957.e504f6c", "order": 1, "display": false, "width": "5", "collapse": false}, {"id": "67163726.764658", "type": "ui_group", "name": "2", "tab": "67e2957.e504f6c", "order": 2, "display": false, "width": "5", "collapse": false}, {"id": "70b4a6cc.14fd78", "type": "ui_group", "name": "3", "tab": "67e2957.e504f6c", "order": 3, "display": false, "width": "12", "collapse": false}, {"id": "223e1297.90f16e", "type": "mqtt_in", "z": "8d8ede65.bc90c", "name": "", "topic": "Test7749", "qos": "2", "datatype": "auto", "broker": "7436b594.5bbf3c", "nl": false, "rap": true, "rh": "0", "x": 80, "y": 380, "wires": [ [{"id": "dcd518f7.44c468", "z": "d74c2697.e45cb8"} ] ], {"id": "dcd518f7.44c468", "type": "function", "z": "8d8ede65.bc90c", "name": "", "func": "// Tempp=38.20, Humid=73.52 \n\nvar msg1 = {};\nvar msg2 = {};\n\nvar output = msg.payload.split('\\n');\n\nvar sTempp = output[0].split('=')[1];\nmsg1.payload = sTempp[1];\nmsg1.topic = 'Temperature';\n\nvar sHumid = output[1].split('=')[1];\nmsg2.payload = sHumid[1];\nmsg2.topic = 'Humidity';\n\nreturn [msg1, msg2];"}, {"id": "8eff3667.95dcf8", "type": "ui_chart", "z": "8d8ede65.bc90c", "name": "", "group": "67163726.764658", "order": 1, "width": "5", "height": "5", "label": "5Min-Monitor", "chartType": "line", "legend": "true", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "0", "ymax": "100", "removeOlder": "5", "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColor": false, "useUTC": false, "colors": ["#1f77b4", "#ff7f0e", "#0fffef", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "outputs": 1, "useDifferentColor": false, "x": 590, "y": 220, "wires": [ [] ] }, {"id": "77a5ac72.e33644", "type": "ui_chart", "z": "8d8ede65.bc90c", "name": "", "group": "70b4a6cc.14fd78", "order": 0, "width": "12", "height": "5", "label": "Day-Monitor", "chartType": "line", "legend": "true", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "0", "ymax": "100", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "useUTC": false, "colors": ["#1f77b4", "#ff7f0e", "#0fffef", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "outputs": 1, "useDifferentColor": false, "x": 590, "y": 180, "wires": [ [] ] }, {"id": "11c2d779.976e69", "type": "ui_gauge", "z": "8d8ede65.bc90c", "name": "", "group": "29de1dd2.698e12", "order": 4, "width": "5", "height": "4", "gtype": "gage", "title": "Temperature(C)", "label": "%", "format": "{value}", "min": 0, "max": 100, "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 620, "y": 280, "wires": [ [] ] }, {"id": "9cf31456.952a38", "type": "ui_led", "z": "8d8ede65.bc90c", "name": "", "group": "70b4a6cc.14fd78", "order": 2, "width": "5", "height": "2", "label": "Input-Monitor", "chartType": "line", "legend": "true", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "0", "ymax": "100", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "useUTC": false, "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 610, "y": 320, "wires": [ [] ] }, {"id": "148b7d53.08ad13", "type": "ui_switch", "z": "8d8ede65.bc90c", "name": "", "label": "Output Control", "tooltip": "", "group": "67163726.764658", "order": 2, "width": "5", "height": "1", "passthru": true, "decouple": false, "topic": "topic", "topicType": "msg", "style": "", "onvalue": "ON1", "onvalueType": "str", "onicon": "", "oncolor": "", "offvalue": "OFF1", "offvalueType": "str", "officon": "", "offcolor": "", "animate": false, "x": 100, "y": 500, "wires": [ [{"id": "d9c08c8.771a57"} ] ], {"id": "d9c08c8.771a57", "type": "mqtt_out", "z": "8d8ede65.bc90c", "name": "", "topic": "Test7749", "qos": "", "retain": "", "respTopic": "", "contentType": "", "userProps": "", "correl": "", "expiry": "", "broker": "7436b594.5bbf3c", "x": 320, "y": 500, "wires": [ [] ] }, {"id": "d74c2697.e45cb8", "type": "function", "z": "8d8ede65.bc90c", "name": "", "func": "\nvar stsNow = context.global.sts;\nif (msg.payload == 'SW_ON') {\n    stsNow = true;\n}\nif (msg.payload == 'SW_OFF') {\n    stsNow = false;\n}\ncontext.global.sts = stsNow;\nnmsg.payload = stsNow;\n\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 300, "y": 380, "wires": [ [{"id": "9cf31456.952a38"} ] ] }

```

3/6 - การโปรแกรมเพื่อแจ้งเตือนผ่าน LINE ด้วย LINE API Python

<https://www.youtube.com/watch?v=rwkvgtXgCzs>

<https://github.com/carpedm20/LINE>

<https://www.on-fix.com/2020/01/chapter-10-line-notify.html>

Lab403 – LINE Notify with Python

- ทำการอัปเดต packet ของ raspbian ก่อน

sudo apt-get install update
sudo apt-get install upgrade

Updat , Upgrade

- ติดตั้ง requests Library

pip install requests

client 8 (Install)

```
pi@raspberrypi:~ $ pip install requests
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: requests in /usr/lib/python2.7/dist-packages (2.21.0)
pi@raspberrypi:~ $
```

- เข้าไปที่ <https://notify-bot.line.me/th/> เพื่อสร้าง Token Key

Wichai Srisuruk

My page

Manage registered services

Generate token

Please enter a token name to be displayed before each notification.

Test7749

Select a chat to send notifications to.

Search by group name

1-on-1 chat with LINE Notify

Bluechara

IS1 M 4/3

Opt.Chem.

The nanoster

Note: Revealing your personal access token can allow a third party to obtain the names of your connected chats as well as your profile name.

Generate token

4. ทดสอบส่งข้อความ “Hello Test-7749”

```

import requests
token = '7h9eZuTCwdTRw0FTctHTCsq6VCKhMWRTi6IzVHIVPPG' Hg Token
url = 'https://notify-api.line.me/api/notify'

LHeaders = {'content-type':'application/x-www-form-urlencoded','Authorization':'Bearer '+token}
msg = 'Hello Test7749'
r = requests.post(url, headers=LHeaders, data = {'message':msg})
print (r.text)
    
```

Msg

test_LINE.py

```

1 import requests
2 token = '7h9eZuTCwdTRw0FTctHTCsq6VCKhMWRTi6IzVHIVPPG'
3 url = 'https://notify-api.line.me/api/notify'
4
5 LHeaders = {
6     'content-type': 'application/x-www-form-urlencoded',
7     'Authorization': 'Bearer ' + token
8 }
9
10 msg = 'Hello Test7749'
11 r = requests.post(url, headers=LHeaders, data = {'message':msg})
12 print (r.text)
13
14
    
```

```
{"status":200, "message":"ok"}
```

200 OK

```
-- 
(program exited with code: 0)
Press return to continue
```

5. ทดสอบส่งข้อความ รูปภาพ รูปภาพผ่านลิงค์ และ Sticker

```

import requests
import time

L_TOKEN = '7h9eZuTCwdTRwOFTctHTCsq6VCXhMWRTi6lzVHIVPPG'
LINE_URL = 'https://notify-api.line.me/api/notify'

LHeaders = {
    'Content-Type': 'application/x-www-form-urlencoded',
    'Authorization': 'Bearer ' + L_TOKEN
}

#function send text
def messageNotify(message):
    payload = {'message': message}
    session = requests.Session()
    r = requests.post(LINE_URL, headers=LHeaders, data=payload)
    return r

#function send Picture(png,jpg)
def fileNotify(filename):
    file = {'imageFile':open(filename,'rb')}
    payload = {'message': "Picture"}
    XHeaders = {'Authorization': 'Bearer ' + L_TOKEN}
    session = requests.Session()
    r = requests.post(LINE_URL, headers=XHeaders, files=file, data=payload)
    return r

#function send picture-url
def urlImageNotify(url):
    payload = {
        'message': "URL-Img",
        'imageThumbnail': url,
        'imageFullsize': url
    }
    XHeaders = {'Authorization': 'Bearer ' + L_TOKEN}
    session = requests.Session()
    r = requests.post(LINE_URL, headers=XHeaders, data=payload)
    return r

#function send sticker Ref>https://devdocs.line.me/files/sticker_list.pdf
def stickerNotify(stickerID,stickerPackageID):
    payload = {
        'message': "Sticker",
        'stickerPackageId':stickerPackageID,
        'stickerId':stickerID
    }
    session = requests.Session()
    r = requests.post(LINE_URL, headers=LHeaders, data = payload)
    return r

### Main Test Run ###
print('1/4-Test Message')
print(messageNotify("Hello Test-1212312121"))
time.sleep(2)

print('2/4-Test Picture')
print(fileNotify('/home/pi/Pictures/Test2.png'))
time.sleep(2)

print('3/4-Test URL-Img')
print(urlImageNotify('https://s1.ibtimes.com/sites/www.ibtimes.com/files/styles/full/public/2017/05/26/ladybug-7435621920.jpg'))
time.sleep(2)

print('3/4-Test URL-Img')
print(urlImageNotify('https://hddesktopwallpapers.in/wp-content/uploads/2015/09/insect-picture.jpg'))
time.sleep(2)

print('4/4-Test Sticker')
print(stickerNotify(179,2))
time.sleep(2)

```

```
1 import requests
2 import time
3
4 L-Token = '7h9eZuTCwdTRwOFTctHTCsq6VChMWRTi6IzVHlVPPG'
5 LNE_URL = 'https://notify-api.line.me/api/notify'
6
7 LHeaders = {
8     'content-type': 'application/x-www-form-urlencoded',
9     'Authorization': 'Bearer ' + L-Token
10    }
11
12 #function send text
13 def messageNotify(message):
14     payload = {'message': message}
15     session = requests.Session()
16     r = requests.post(LNE_URL, headers=LHeaders, data=payload)
17     return r
18
19 #function send Picture(png,jpg)
20 def fileNotify(filename):
21     file = {'imageFile': open(filename, 'rb')}
22     payload = {'message': "Picture"}
23     XHeaders = {'Authorization': 'Bearer ' + L-Token}
24     session = requests.Session()
25     r = requests.post(LNE_URL, headers=XHeaders, files=file, data=payload)
26     return r
27
28 #function send picture-url
29 def urlImageNotify(url):
30     payload = { 'message': "URL-Img",
31                 'imageThumbnail': url,
32                 'imageFullsize': url
33                 }
34     XHeaders = {'Authorization': 'Bearer ' + L-Token}
35     session = requests.Session()
36     r = requests.post(LNE_URL, headers=XHeaders, data=payload)
37     return r
38
39 #function send sticker Ref>https://devdocs.line.me/files/sticker_list.p
40 def stickerNotify(stickerID,stickerPackageID):
41     payload = { 'message': "Sticker",
42                 'stickerPackageId': stickerPackageID,
43                 'stickerId': stickerID
44                 }
45     session = requests.Session()
46     r = requests.post(LNE_URL, headers=LHeaders, data = payload)
47     return r
48
49
```

```

50  ### Main Test Run ###
51  print('1/4-Test Massage')
52  print(messageNotify("Hello Test-1212312121"))
53  time.sleep(2)
54
55  print('2/4-Test Picture')
56  print(fileNotify('/home/pi/Pictures/Test2.png'))
57  time.sleep(2)
58
59  print('3/4-Test URL-Img')
60  print(urlImageNotify('https://s1.ibtimes.com/sites/www.ibtimes.com/files'))
61  time.sleep(2)
62
63  print('3/4-Test URL-Img')
64  print(urlImageNotify('https://hddesktopwallpapers.in/wp-content/uploads/'))
65  time.sleep(2)
66
67  print('4/4-Test Stiker')
68  print(stickerNotify(179,2))
69  time.sleep(2)
70

```

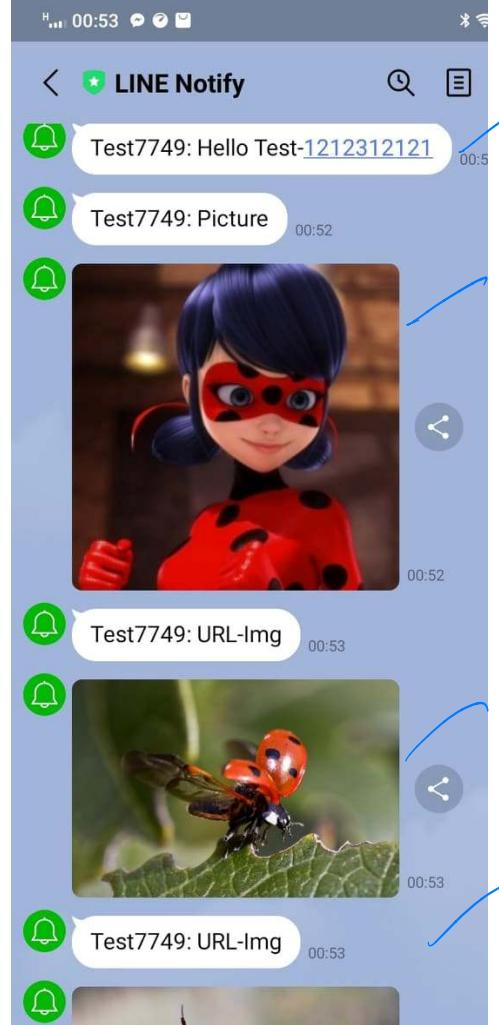
SSH is enabled and the default password is set to 'raspberry'. This is a security risk - please change it to a new password.

```

1/4-Test Massage
<Response [200]>
2/4-Test Picture
<Response [200]>
3/4-Test URL-Img
<Response [200]>
3/4-Test URL-Img
<Response [200]>
4/4-Test Stiker
<Response [200]>

-----

```

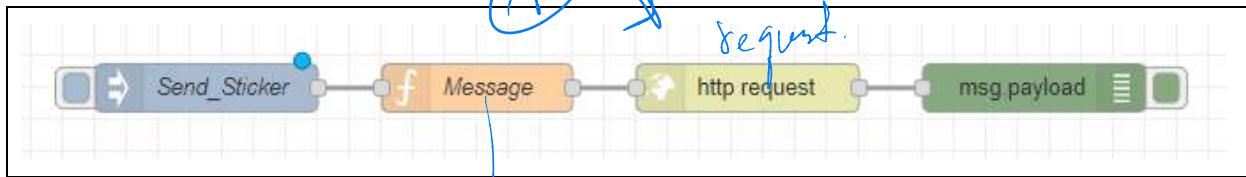


4/6 - การโปรแกรมเพื่อแจ้งเตือนผ่าน LINE ด้วย Node-RED

<https://jackrobotics.me/จับ-line-notify-ยด-node-red-โดยใชความสามารถเดิมของ-node-red-1730d83c067c>

Lab404 – LINE Notify with NODE-RED

1. Send Sticker with Node-RED



```

[{"id": "50d97bcf.d02684", "type": "tab", "label": "Flow LINE", "disabled": false, "info": ""}, {"id": "8cb66889.d96b28", "type": "http request", "z": "50d97bcf.d02684", "name": "", "method": "POST", "ret": "txt", "paytoqs": "ignore", "url": "https://notify-api.line.me/api/notify", "ts": "", "persist": false, "proxy": "", "authType": "", "x": 530, "y": 200, "wires": [{"id": "6d47b0ce.054b2"}]}, {"id": "6d47b0ce.054b2", "type": "debug", "z": "50d97bcf.d02684", "name": "", "active": true, "console": false, "statusVal": "", "statusType": "auto", "x": 710, "y": 200, "wires": []}, {"id": "51ef834a.3d95ec", "type": "function", "z": "50d97bcf.d02684", "name": "Message", "func": "msg.token = '7h9eZuTCwdTRw0FTctHTCsq6VChMWRTi6IzVHlVPPG';\nmsg.message = 'Hello';\nmsg.stickerPackageId = 1;\nmsg.stickerId = 106;\n\nmsg.headers = {\n    'content-type': 'application/x-www-form-urlencoded',\n    'Authorization': 'Bearer ' + msg.token\n};\n\nmsg.payload = {\n    'message': msg.message,\n    'stickerPackageId':msg.stickerPackageId,\n    'stickerId':msg.stickerId\n};\n\nreturn msg;", "x": 360, "y": 200, "wires": [{"id": "75e765f7.d089fc"}]}, {"id": "75e765f7.d089fc", "type": "inject", "z": "50d97bcf.d02684", "name": "", "x": 710, "y": 200, "wires": []}], {"id": "75e765f7.d089fc", "type": "inject", "z": "50d97bcf.d02684", "name": "", "x": 710, "y": 200, "wires": []}
  
```

Setup	On Start	On Message	On Stop	Change
		<pre> 1 msg.token = '7h9eZuTCwdTRw0FTctHTCsq6VChMWRTi6IzVHlVPPG'; 2 msg.message = 'Hello'; 3 msg.stickerPackageId = 1; 4 msg.stickerId = 106; 5 6 msg.headers = { 7 'content-type': 'application/x-www-form-urlencoded', 8 'Authorization': 'Bearer ' + msg.token 9 }; 10 11 msg.payload = { 12 'message': msg.message, 13 'stickerPackageId':msg.stickerPackageId, 14 'stickerId':msg.stickerId 15 }; 16 return msg; 17 </pre>	# Token Key in function # Message # Sticker Packed ID # Sticker ID	

Sticker List: https://devdocs.line.me/files/sticker_list.pdf

2. Send Message, Image, URL-Image, Sticker with Node-RED

node-red-contrib-notifyimage ↗
sending text,image via line notify
0.0.1 10 months ago

(Handwritten notes: B Myp)

Add Node =
node-red-contrib-notifyimage
VO.0.1

```
[{"id": "50d97bcf.d02684", "type": "tab", "label": "Flow LINE", "disabled": false, "info": ""}, {"id": "8cb66889.d96b28", "type": "http request", "z": "50d97bcf.d02684", "name": "", "method": "POST", "ret": "txt", "paytoqs": "ignore", "url": "https://notify-api.line.me/api/notify", "tls": "", "persist": false, "proxy": "", "authType": "", "x": 200, "y": 200, "wires": [{"id": "6d47b0ce.054b2"}]}, {"id": "6d47b0ce.054b2", "type": "debug", "z": "50d97bcf.d02684", "name": "", "active": true, "tosticker": true, "console": false, "tostatus": false, "complete": "payload", "targetType": "msg", "statusVal": "", "statusType": "auto", "x": 270, "y": 200, "wires": []}, {"id": "51ef834a.3d95ec", "type": "function", "z": "50d97bcf.d02684", "name": "Message1", "func": "var token = '7h9eZuTCwdTrwOFTctHTCsq6VChhMWRTi6izVHIVPPG';\nvar message = 'LINE-Massage Test-12123';\n\nnmsg.headers = {\n  'Content-Type': 'application/x-www-form-urlencoded',
  'Authorization': 'Bearer ' + token
};\n\nnmsg.payload = {\n  'message': message
};\n\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 370, "y": 200, "wires": [{"id": "8cb66889.d96b28"}]}, {"id": "75e765f7.d089fc", "type": "inject", "z": "50d97bcf.d02684", "name": "Send_Massage", "props": [{"p": "payload"}, {"p": "topic", "vt": "str"}], "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "topic": "", "payload": "", "payloadType": "date", "x": 180, "y": 200, "wires": [{"id": "51ef834a.3d95ec"}]}, {"id": "80860978.fca738", "type": "inject", "z": "50d97bcf.d02684", "name": "Send_Sticker", "props": [{"p": "payload"}, {"p": "topic", "vt": "str"}], "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "topic": "", "payload": "", "payloadType": "date", "x": 170, "y": 260, "wires": [{"id": "d94f6383.f36cc"}]}, {"id": "3ae61caf.c237f4", "type": "function", "z": "50d97bcf.d02684", "name": "Message3", "func": "var url_img = 'https://s1.libtimes.com/sites/www.libtimes.com/files/styles/full/public/2017/05/26/ladybug-7435621920.jpg';\n\nnmsg.headers = {\n  'Content-Type': 'application/x-www-form-urlencoded',
  'Authorization': 'Bearer ' + token
};\n\nnmsg.payload = {\n  'imageThumbnailUrl': url_img,
  'imageFullsizeUrl': url_img
};\n\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 370, "y": 320, "wires": [{"id": "8cb66889.d96b28"}]}, {"id": "77d0259a.90a3c", "type": "inject", "z": "50d97bcf.d02684", "name": "Send_URL-Image", "props": [{"p": "payload"}, {"p": "topic", "vt": "str"}], "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "topic": "", "payload": "", "payloadType": "date", "x": 190, "y": 320, "wires": [{"id": "8cb66889.d96b28"}]}, {"id": "d94f6383.f36cc", "type": "function", "z": "50d97bcf.d02684", "name": "Message2", "func": "var token = '7h9eZuTCwdTrwOFTctHTCsq6VChhMWRTi6izVHIVPPG';\n\nnmsg.headers = {\n  'Content-Type': 'application/x-www-form-urlencoded',
  'Authorization': 'Bearer ' + token
};\n\nnmsg.payload = {\n  'stickerPackageld': stickerPackageld
};\n\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 370, "y": 380, "wires": [{"id": "8cb66889.d96b28"}]}, {"id": "7edcb5f1.7e8d1c", "type": "NotifyImage", "z": "50d97bcf.d02684", "name": "LINE-Image", "msgfile": "/home/pi/Pictures/Test2.png", "AccToken": "7h9eZuTCwdTrwOFTctHTCsq6VChhMWRTi6izVHIVPPG", "x": 550, "y": 380, "wires": [{"id": "8cb66889.d96b28"}]}, {"id": "997fd0d3.27d59", "type": "function", "z": "50d97bcf.d02684", "name": "Message4", "func": "var message = 'Send Image Data';\nnmsg.payload = message;\n\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 370, "y": 440, "wires": [{"id": "7edcb5f1.7e8d1c"}]}]
```

```

graph TD
    S1[Send_Massage] --> M1[Message1]
    S2[Send_Sticker] --> M2[Message2]
    S3[Send_URL-Image] --> M3[Message3]
    S4[Send_Image] --> M4[Message4]

    M1 --> H1[http request]
    M2 --> H2[http request]
    M3 --> H3[http request]
    M4 --> H4[http request]

    H1 --> P1[msg.payload]
    H2 --> P1
    H3 --> P1
    H4 --> P1

```

Function Change
Token Key in function
Massage
Sticker Packed ID, Sticker ID
Image URL

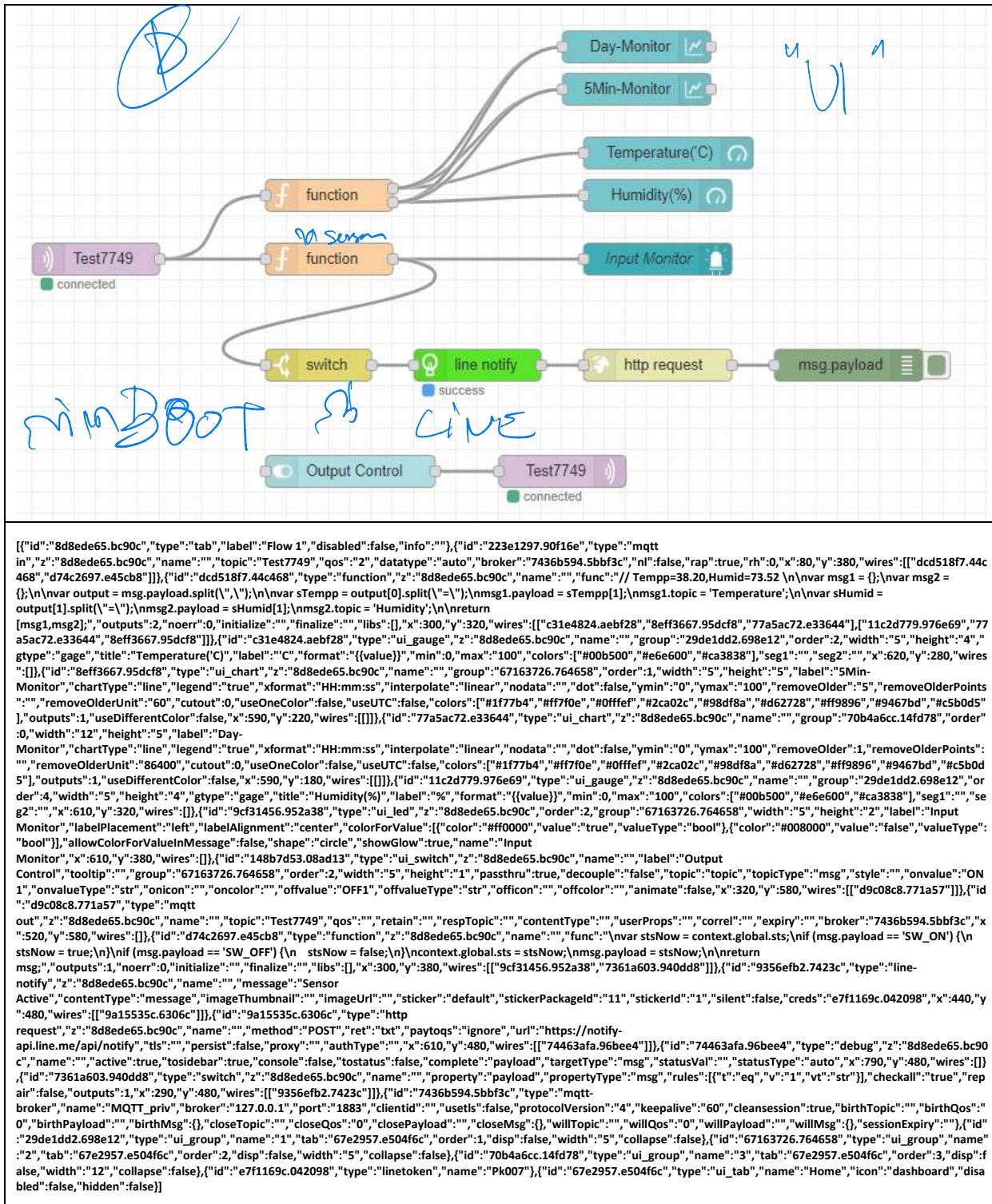
LINE Image Change
Token Key in function
Massage
Image File

3. ทดสอบการใช้งานด้วย LINE-Notify node

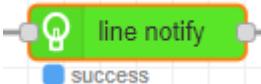
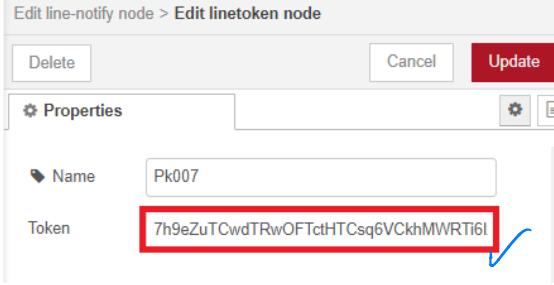
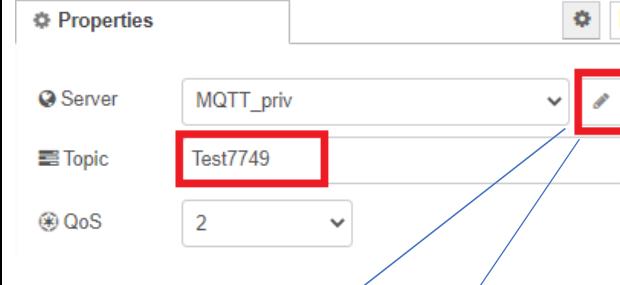
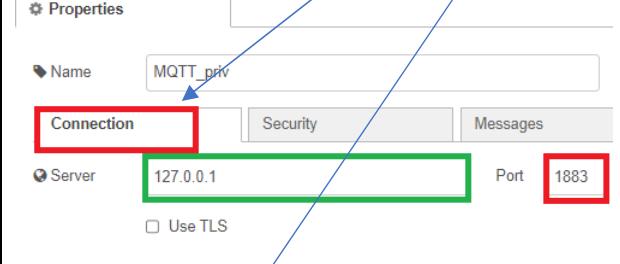
Add Node =
node-red-contrib-line-notify
V03.1.1

```
[{"id": "6790d352.2280ac", "type": "tab", "label": "Flow 2", "disabled": false, "info": ""}, {"id": "ca2e8cd2.5bd98", "type": "line-notify", "z": "6790d352.2280ac", "name": "", "message": "Test-wichai", "1212312121": "contentType": "message", "imageThumbnail": "", "imageUrl": "", "sticker": "default", "stickerPackageId": "11", "stickerId": "1", "silent": false, "creds": "e7f1169c.042098", "x": 420, "y": 180, "wires": [{"15fc8190.012b2e"}]}, {"id": "15fc8190.012b2e", "type": "http request", "z": "6790d352.2280ac", "name": "", "method": "POST", "ret": "txt", "paytoqs": "ignore", "url": "https://notify-api.line.me/api/notify", "tls": "", "persist": false, "proxy": "", "authType": "", "x": 630, "y": 180, "wires": [{"b23ae221.44489"}]}, {"id": "b23ae221.44489", "type": "debug", "z": "6790d352.2280ac", "name": "", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "targetType": "msg", "statusVal": "", "statusType": "auto", "x": 810, "y": 180, "wires": []}, {"id": "88d3a65e.a258c8", "type": "inject", "z": "6790d352.2280ac", "name": "", "props": [{"p": "payload"}, {"p": "topic"}, {"p": "vt"}], "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "topic": "", "payload": "", "payloadType": "date", "x": 220, "y": 180, "wires": [{"ca2e8cd2.5bd98"}]}, {"id": "e7f1169c.042098", "type": "linetoken", "name": "Pk007"}]
```

4. If Press Switch 0 on ESP32 then LINE Alert



4.1 ແກ່ໄຂ

1. LINE Token Key	2. MQTT Server {IP, Topic}
	
	
	
	

4.2 ผลการทดสอบ

Home

Temperature('C)

34.74

5Min-Monitor

Temperature Humidity

Day-Monitor

Temperature Humidity

Humidity(%)

62.55 %

Input Monitor

Output Control

V1
LINE

Test7749: Test Action 20:49

Test7749: Sensor Active 21:32

Test7749: Sensor Active 21:32

LIVE

+
camera icon
map icon
...
home icon
refresh icon

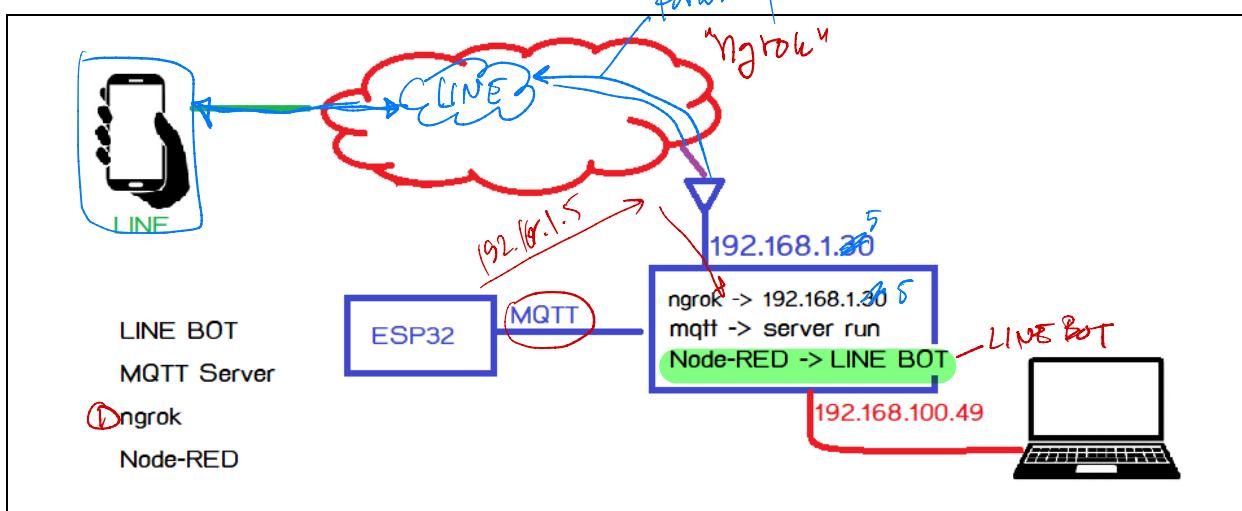
...
home icon
refresh icon

5/6 - การโปรแกรมเพื่อติดต่อกับผู้ใช้งานผ่าน Chatbot LINE

https://www.youtube.com/watch?v=P0_5fD47HIE<https://www.youtube.com/watch?v=4KKIEmWYCGQ><https://thisdavej.com/how-to-host-a-raspberry-pi-web-server-on-the-internet-with-ngrok/><https://medium.com/@nattaponsirikamonnet/สร้าง-bot-ด้วย-line-messaging-api-d7de644ac892>

Lab405 – LINE Chatbot with NODE-RED

1. ระบบที่ทดสอบ



2. ติดตั้ง ngrok เพื่อทำ port forwarding

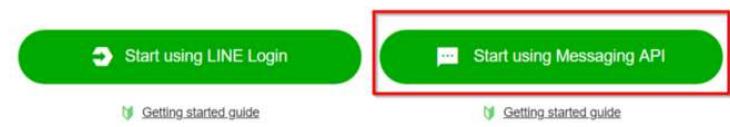
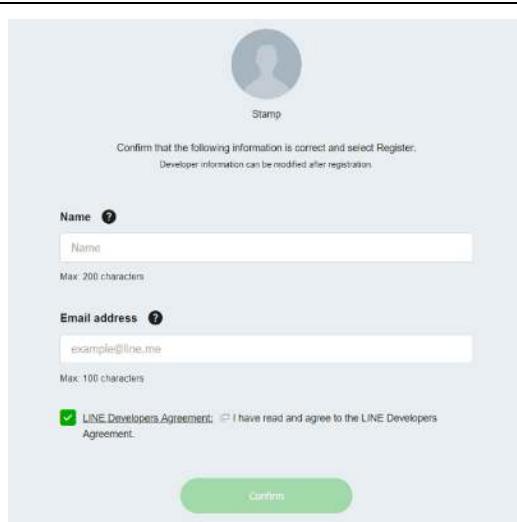
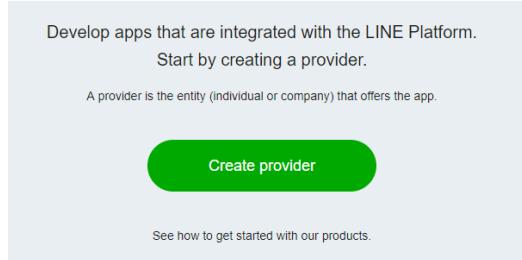
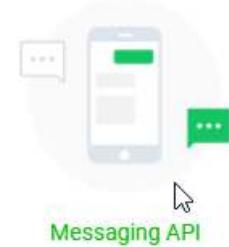
npm -v	ตรวจสอบ npm Version
pi@raspberrypi:~ \$ npm -v 5.8.0	
sudo npm install -g npm@latest	Update npm
pi@raspberrypi:~ \$ sudo npm install -g npm@latest changed 16 packages, and audited 257 packages in 2s 10 packages are looking for funding run `npm fund` for details found 0 vulnerabilities pi@raspberrypi:~ \$ npm -v 7.21.0	ล่าสุด V 7.21.0
sudo npm install --unsafe-perm -g ngrok	Install ngrok

ดูแล้ว ✓

<pre> found 0 vulnerabilities pi@raspberrypi:~ \$ sudo npm install --unsafe-perm -g npm WARN deprecated uuid@3.4.0: Please upgrade to version 8 which ain circumstances, which is known to be problematic. changed 51 packages, and audited 52 packages in 12s 7 packages are looking for funding run `npm fund` for details found 0 vulnerabilities </pre>	
ngrok http 192.168.1.30:1880	Start ngrok
<pre> ngrok by @inconshreveable Session Status online Session Expires 1 hour, 59 minutes Version 2.3.40 Region United States (us) Web Interface http://127.0.0.1:4040 Forwarding http://f121-2403-6200-8820-d649-5647-32dd-88eb-9c10.ngrok.io -> http://192.168.1.30:1880 Forwarding https://f121-2403-6200-8820-d649-5647-32dd-88eb-9c10.ngrok.io -> https://192.168.1.30:1880 Connections ttl opn rt1 rt5 p50 p90 0 0 0.00 0.00 0.00 0.00 </pre>	
	<p>start node-red เพื่อ ทดสอบ ngrok ทดสอบเข้าใช้งานผ่าน web browser</p>

Smart Farmer Test port

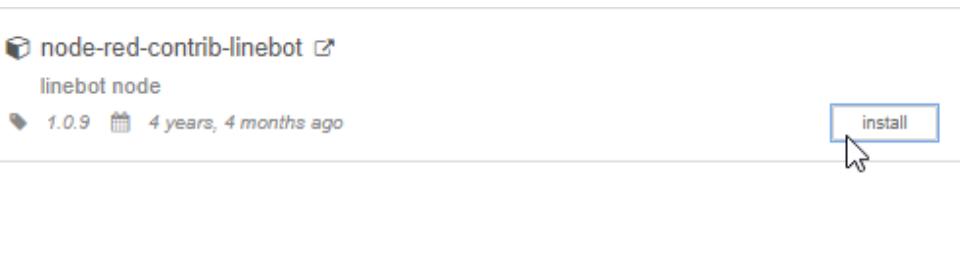
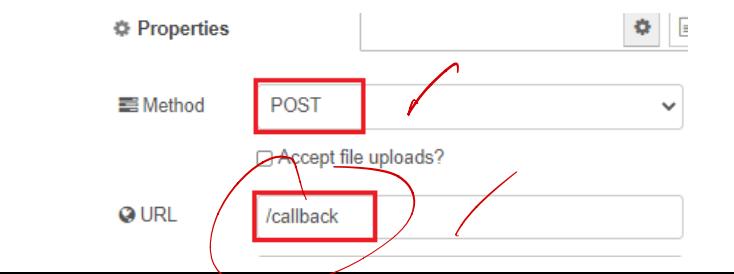
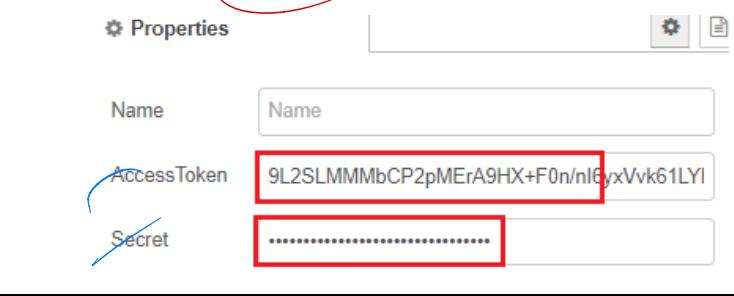
3. เข้าใช้งานและตั้งค่า LINE Bot

3.1 login LINE Bot >> https://developers.line.biz/en/	
3.2 เลือก Start Using Massing API	
	
3.3 จากนั้น Login ด้วย email และ password ที่ได้ผูกไว้กับแอปพลิเคชัน line	
3.4 หลังจาก Login แล้ว หน้าตัดไปให้เลือกที่จะต้องแสดงบนหน้า console และใส่ email ที่ต้องการให้เป็น admin (แนะนำให้ใส่เป็น email ที่ login เข้ามาเมื่อสักครู่)	
	
3.5 จากนั้นกด Create Provider --> กำหนดชื่อ → เลือก Massaging API	>> Pk007_Provider
	
3.6 สร้าง Channel ใน Provider เลือก Massage API	>> Pk007_Channel
	

3.7 กำหนดชื่อ Channel Name - - , Create	
<p><input checked="" type="checkbox"/> I have read and agree to the LINE Official Account Terms of Use <small>[?]</small></p> <p><input checked="" type="checkbox"/> I have read and agree to the LINE Official Account API Terms of Use <small>[?]</small></p> <p><small>✓ Select the checkbox after reading the related document</small></p> <p>Create</p>	
3.8 Add Friend	
<p>Basic settings Messaging API LIFF Security</p> <p>Messaging API settings</p> <p>Bot information</p> <p>Bot basic ID @769teukz <small>[?]</small></p> <p>QR code</p> 	

3.9	<p>Set Web Hook</p> <p><i>+ IP on ngrok</i></p> <p>Webhook settings</p> <p>Webhook URL ⓘ <input type="text" value="https://f4ff-2403-6200-8820-d649-5647-32dd-88eb-9c10.ngrok.io/callback"/> <i>อย่าลืม /callback</i></p> <ul style="list-style-type: none"> ✓ Don't leave this empty ✓ Enter a valid HTTPS URL ✓ Enter no more than 500 characters <p>Update Cancel</p> <p>Use webhook ⓘ <input checked="" type="checkbox"/> <i>Callback</i></p>	
3.10	<p>Get Channel Access Token</p> <p>Channel access token</p> <p>Channel access token (long-lived) ⓘ</p> <p>9L2SLMMMbCP2pMER9HX+F0n/nl6yxVvk61LYbV9⁹8ErE0cW6sW/7g8lhKgI5OgJr06 AdB04t89/10/w1cDnyiIFU=</p>	
3.11	<p>Get Chanel Secret</p> <p>Basic settings Messaging API LIFF Security</p> <p>Basic settings</p> <p>Channel secret ⓘ eb45dc70fd0b6c448⁹d5a65d5db85d718 </p>	
	<p>Create a Messaging API channel with the following details?</p> <p>Channel name : Pk007_Channel Official Account name : Pk007_Channel Provider : Pk007_Provider</p> <p> <ul style="list-style-type: none"> • If you proceed, an official account will be created with the same name as the messaging API channel above. • You cannot change the channel provider after the channel is created. Make sure that the provider and official account owner are the same individual developer, company or organization. • For the handling of LINE user information, please refer to User Data Policy. </p> <p>Cancel OK</p>	

4. ตั้งค่า Node-RED

4.1 Add Node	
	Line <ul style="list-style-type: none"> line - notify linebot - reply linebot - reply - sticker linebot - client
4.2 place 4 node and config	
	
 Properties <ul style="list-style-type: none"> Method: POST URL: /callback 	http
 Properties <ul style="list-style-type: none"> Name: Name AccessToken: 9L2SLMMMBCP2pMErA9HX+F0n/nl6yxVvk61LYI Secret: (redacted) 	Linebot-client
<pre> var receive_msg = msg.payload; if(receive_msg.events[0].message.type=="text"){ if(receive_msg.events[0].message.text=="ສະບັບ"){ reply = {type:'text', text:"ສະບັບ"}; } } else if(receive_msg.events[0].message.text=="sticker"){ reply = { type: "sticker", packageId: 11537, stickerId: 52002744 } } else if(receive_msg.events[0].message.text=="sound"){ reply = { type: "audio", originalContentUrl: "https://mokmoon.com/audios/line.mp3", duration: 1000 } } else if(receive_msg.events[0].message.text=="video"){ reply = { type: "video", } } </pre> <p style="text-align: center;">blue X/r</p>	function

```
originalContentUrl: "https://mokmoon.com/videos/Brown.mp4",
previewImageURL: "https://linefriends.com/img/bangolufs..."
}

}

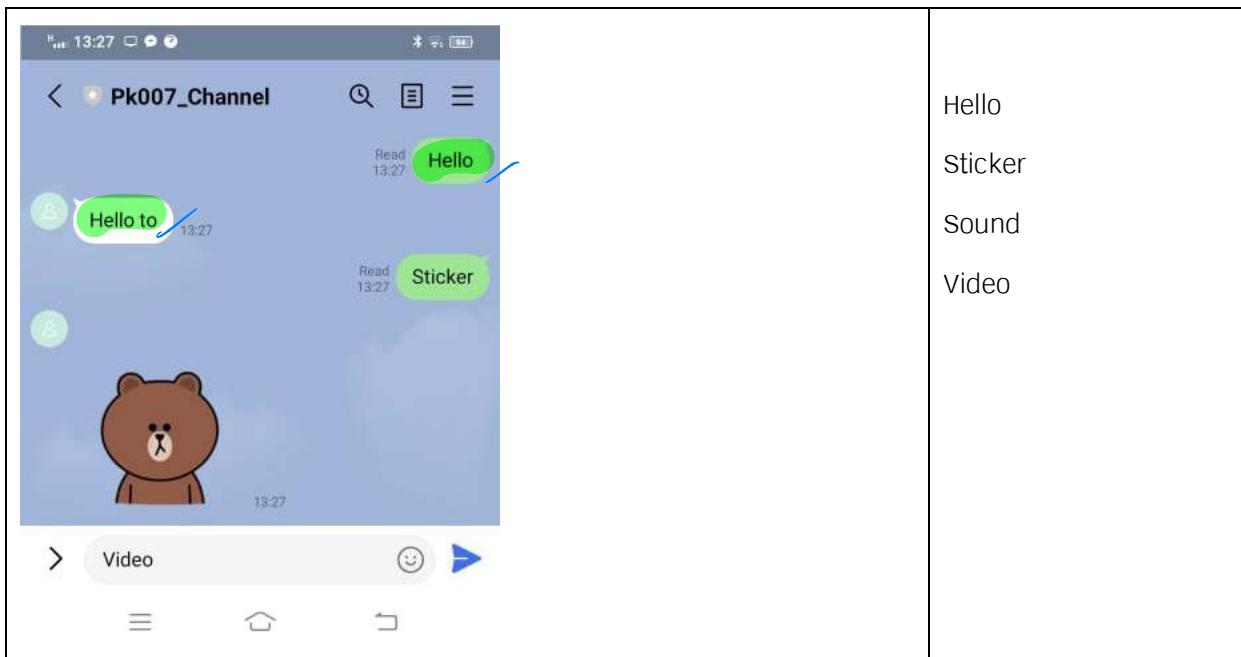
else{ reply = {type:'text', text:"ลิ้วจิ้วลาบวิ้ง"}
}

var value = [receive_msg,reply];
msg.payload = value;
return msg;
```



```
[[{"id": "f051a57f.edfdb8", "type": "tab", "label": "Flow 3", "disabled": "false", "info": ""}, {"id": "ca26da3.0998128", "type": "linebot-client", "z": "f051a57f.edfdb8", "name": "", "channelAccessToken": "9L2SLMMBcP2pMErA9HX+F0n/nl6yxVv61LybP+8ErEcW6sW/7g8lhKgI50gJrO6+3MwInhLIGdU/70CFYal5qNCBCBWHms7MyMw7zybuw591B7Weo21gMky+LVSr3z2rTbCqwujwmNVwsSsAdB04t89/10/w1cDnylfU=", "channelSecret": "eb45dc70fd0b6c449ds5a65d5db85d718", "x": "510", "y": "300", "wires": [{"id": "d16c0df5.b432e2"}, {"id": "34e8c7c7.4295a8"}, {"type": "http-in", "z": "f051a57f.edfdb8", "name": "", "url": "/callback", "method": "post", "upload": false, "swaggerDoc": "", "x": "160", "y": "300", "wires": [{"id": "9c59f23e.f0484"}]}, {"id": "9c59f23e.f0484", "type": "function", "z": "f051a57f.edfdb8", "name": "", "func": "var receive_msg = msg.payload;\nif(receive_msg.events[0].message.type=='text'){\n    if(receive_msg.events[0].message.text=='Hello')\n        reply = {type:'text', text:'Hello to '}\n    else\n        reply = {type: 'sticker',\n            packageId: 11537,\n            stickerId: 52002744\n        }\n\n    else if(receive_msg.events[0].message.text=='Sound')\n        reply = {type: 'audio',\n            originalContentUrl:\n                'https://mokmoon.com/audio/line.mp3',\n            duration: 1000\n        }\n\n    else if(receive_msg.events[0].message.text=='Video')\n        reply = {\n            type: 'video',\n            originalContentUrl: 'https://mokmoon.com/videos/Brown.mp4',\n            previewImageURL: 'https://linefriends.com/img/bangolufs...',\n            w\n        }\n\n    else\n        reply = {\n            type: 'text',\n            text: 'I don't know'\n        }\n\n}]\n\nvar value = [receive_msg,reply];\nmsg.payload = value;\nreturn msg;\n"}, {"id": "d16c0df5.b432e2", "type": "debug", "z": "f051a57f.edfdb8", "name": "", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": false, "statusVal": "", "statusType": "auto", "x": "690", "y": "300", "wires": []}]]
```

5. ทดสอบการทำงาน



6. ปรับปรุง Node-RED เพื่อตอบ { Tempp, Humid, Sensor, On, Off }

The Node-RED flow diagram illustrates the architecture for a smart farming application. It starts with an MQTT connection to a broker named "Test7749". This connection triggers a function node, which then branches into two parallel monitoring paths: "Day-Monitor" and "5Min-Monitor". These paths read "Temperature(C)" and "Humidity(%)". The data from these sensors is collected by an "Input Monitor" node. A "Notify" node is triggered by this monitor, which then sends a "line notify" message via an "http request" to an external service. The response is stored in "msg.payload". Below this, an "Output Control" node is connected to a "function" node, which then connects back to the "Test7749" connection. A timestamp node is also present in the flow. Handwritten annotations include "Temp", "Hum", "Sen", "OS", "Notify", "my func", and "bot".

<p>Pk007_Channel</p> <ul style="list-style-type: none"> Hello to 43.27 61.06 Ok On Ok Off Sensor Off Sensor On <p>Responses:</p> <ul style="list-style-type: none"> Read 14:23 Temp Read 14:23 Humid Read 14:23 On Read 14:23 Off Read 14:23 Sensor Read 14:23 Sensor 	<p>แก้ไข</p> <p>MQTT Server -> { IP, Topic} LINE Notify -> Token LINE Bot → {Token Key, Secret} Ngrok → IP Forward</p> <p>ทดสอบได้ต่อไปด้วย</p> <p>Tempp Humid Sensor On Off</p>
--	---

การสร้าง MQTT Server บน Raspberry Pi เพื่อใช้งาน Chatbot LINE ในฟาร์มอัจฉริยะ
Chatbot LINE from Raspberry Pi MQTT Server for Smart Farming

ข้อ-สกุล :

6/6 - คำถ้ามท้ายบทเพื่อทดสอบความเข้าใจ

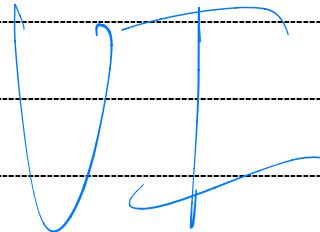
Quiz_401 – RPi Smart Farm

- แสดงรูป โปรแกรม ของผลการทำงานตามหัวข้อ การสร้าง UI ด้วย Node-RED สำหรับฟาร์มอัจฉริยะ

Capture Node-RED Flow

Node-RED Code

รูปการทดสอบ 1: UI Result



รูปการทดสอบ 2:

รูปการทดสอบ 3

รูปการทดสอบ 4

รูปการทดสอบ 5

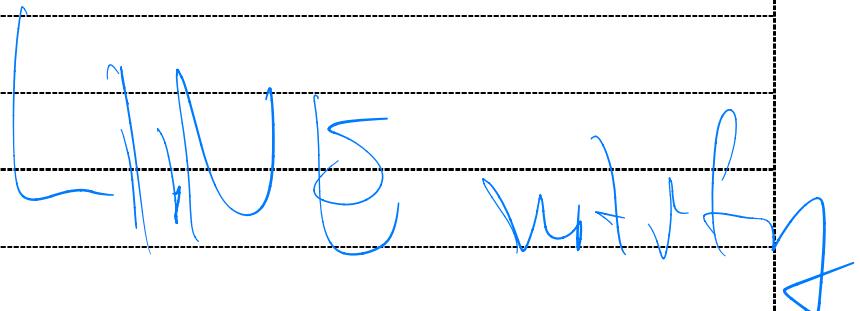
Quiz_402 – LINE Notify

- แสดงรูป โปรแกรม ของผลการทำงานตามหัวข้อ การส่งข้อความด้วย Node-RED สำหรับฟาร์มอัจฉริยะ

Capture Node-RED Flow

Node-RED Code

รูปการทดสอบ 1



รูปการทดสอบ 2:

รูปการทดสอบ 3

รูปการทดสอบ 4

รูปการทดสอบ 5

Quiz_403 – LINE Chatbot

- แสดงรูป โปรแกรม ของผลการทำงานตามหัวข้อ การติดต่อสื่อสารด้วยข้อความสำหรับฟาร์มอัจฉริยะ

Capture Node-RED Flow	
Node-RED Code	
รูปการทดสอบ 1	
รูปการทดสอบ 2:	
รูปการทดสอบ 3	
รูปการทดสอบ 4	
รูปการทดสอบ 5	
รูปการทดสอบ 6	