

2102447 ปฏิบัติการวิศวกรรมอิเล็กทรอนิกส์

Electronic Engineering Laboratory

ผู้สอนประจำวิชา ผศ.ดร.สุริย์ พุ่มรินทร์ และ อ.ดร.ณพงศ์ ปณิธานธรรม

ผู้สอนปฏิบัติการ ภัทร เกษมสำราญ (ภาคการศึกษาปลาย 2567)



ภาควิชาวิศวกรรมไฟฟ้า  
คณะวิศวกรรมศาสตร์

ก่อตั้ง ๒๔๗๒

จุฬาลงกรณ์มหาวิทยาลัย

## การควบคุมและทริกเกอร์อุปกรณ์ IoT ด้วย NETPIE

(IoT Device Control and Trigger via NETPIE)

### วัสดุและอุปกรณ์

1. คอมพิวเตอร์ติดตั้งแอปพลิเคชัน Arduino IDE 2.3.3
2. บอร์ด IOXESP32+ และบอร์ดขยายขา BASE32
3. โมดูลจอ OLED 0.96" 128x64 pixel (I<sup>2</sup>C protocol)
4. สาย USB-A to USB-C, สายไฟ Jumper
5. โมดูลวัดอุณหภูมิ ความชื้น และความดัน BME280 sensor (I<sup>2</sup>C protocol)
6. โมดูลวัดความเข้มแสง TSL2561 Lux sensor (I<sup>2</sup>C protocol)
7. โมดูลตรวจจับการเคลื่อนไหว PIR Motion sensor (Digital output)
8. โมดูลโปรβάตอุณหภูมิดิจิทัล DS18B20 (Digital 1-Wire)
9. อุปกรณ์อิเล็กทรอนิกส์อื่นๆ



### "IoT Learning Kit" 2102447 Electronics Engineering Laboratory

ELECTRICAL  
ENGINEERING  
LABORATORY  
CHULALONGKORN UNIVERSITY



IOXESP32+ Detail



ESP32 Repository

#### Components list :

- |                                     |         |
|-------------------------------------|---------|
| 1.) IOXESP32+ V1.0 Embedded Board   | 1 each  |
| 2.) BASE32 Pinout Expansion Board   | 1 each  |
| 3.) OLED Display 128x64 0.96"       | 1 each  |
| 4.) OLED Housing Bracket            | 1 each  |
| 5.) Breadboard / Protoboard         | 1 each  |
| 6.) 4 SPST Switch Module            | 1 each  |
| 7.) 8mm LED (R/Y/G) Module          | 1 each  |
| 8.) Potentiometer Module            | 1 each  |
| 9.) USB-C to USB-A Cable            | 1 each  |
| 10.) Jumper Wire (Male to Male)     | 10 each |
| 11.) Jumper Wire (Male to Female)   | 10 each |
| 12.) Jumper Wire (Female to Female) | 10 each |

#### Sensor selection list :

- |   |   |
|---|---|
| <input type="checkbox"/> TSL2561 Lux sensor | <input type="checkbox"/> BME280 (Temp, Humid, Pressure) |
| <input type="checkbox"/> PIR Motion Sensor  | <input type="checkbox"/> DS18B20 with Module            |

### IOXESP32+



[docs.ioxesp32.com/ioxesp32+](https://docs.ioxesp32.com/ioxesp32+)

หมายเหตุ : ในกล่องไม่มีรายการที่ 5.) Breadboard / Protoboard ให้

## การทดลองที่ 1 | การควบคุมและทริกเกอร์อุปกรณ์ด้วย NETPIE

### สิ่งที่ต้องส่ง

มีสิ่งที่ต้องส่ง 1 รายการ คือ

1. วิดีโอแสดงการทำงานของอุปกรณ์ โดยส่ง URL ของวิดีโอใน textbox บน mycourseville

### ขั้นตอนปฏิบัติ

1. ดาวน์โหลดไฟล์ IoT-3.zip โดยประกอบด้วย 7 ไฟล์ ได้แก่
  - a. Lab447-IoT-3.pdf ชีทแล็บนี้
  - b. IoT-3.ino โค้ดหลัก
  - c. credentials.h เก็บรหัสผ่าน WiFi และ NETPIE
  - d. iot\_iconset\_16x16.h เก็บไอคอน
  - e. schema.txt โค้ด Schema
  - f. trigger.txt โค้ด Trigger
  - g. eventhooks.txt โค้ด Event Hooks
2. ต่อโมดูล OLED และโมดูล BME280 เข้ากับบอร์ด ESP32 ผ่านทาง I2C และแก้ไข I2C Address ในไฟล์ IoT-3.ino หากไม่ทราบให้ใช้ I2C Scanner เพื่อหา Address ของอุปกรณ์ก่อน
3. แก้ไขรหัส WiFi และ NETPIE ในไฟล์ credentials.h แนะนำให้ใช้ WiFi Hotspot เพื่อความสะดวกในการทดลองเปิดปิด WiFi เพื่อตรวจสอบว่า Source code สามารถ Reconnect ได้หรือไม่
4. ให้นิสิตไปที่เว็บไซต์ <https://netpie.io> จากนั้นเลือก Login เพื่อเข้าสู่ระบบเหมือนการทดลองก่อนหน้านี้
5. เลือกที่ Device และกดเข้าไปที่ Device ที่สร้างไว้แล้ว เลือก Schema ของ Device ใน NETPIE เปลี่ยนมุมมองจาก Tree เป็น Code จากนั้น Copy โค้ดในไฟล์ schema.txt ใส่ลงไปแล้วกด Save
6. Upload โค้ดในไฟล์ IoT-3.ino ไปยังบอร์ด ESP32 แล้วดูผลลัพธ์ที่จอ OLED และส่วน Shadow ของ Device ใน NETPIE หากบอร์ด ESP32 สามารถเชื่อมต่อ WiFi & NETPIE จะแสดงค่าใน Shadow
7. ให้นิสิตเข้าที่ NETPIE 2020 กดเลือก Project ที่นิสิตสร้างไว้จากนั้นไปที่ Console และกดเลือก Dashboard จากนั้นกดปุ่ม “+ Create” เพื่อสร้าง Dashboard ตั้งชื่อ Dashboard Name (และกำหนด Description ถ้ามี) กดปุ่ม Save แล้วกดเลือก Dashboard ที่สร้างไว้
8. เมื่อนิสิตเข้ามาในหน้า Dashboard แล้วให้เลือก Setting จากนั้นกดปุ่ม “+ Add device” และเลือก Device ที่ได้สร้างไว้ก่อนหน้านี้ ส่วนของ Privileges ให้เลือก Subscribe Message, Publish Message, Read Shadow, Write Shadow, Read Feed, Write Feed และ Push แล้วจึงกด Save

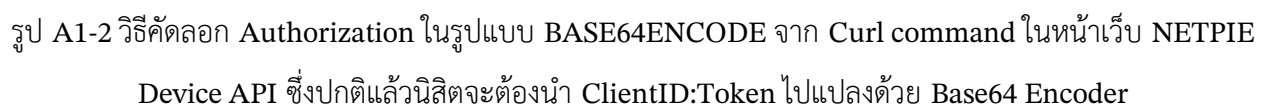
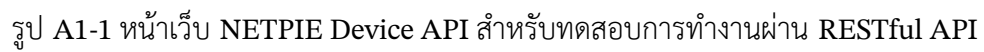
9. กลับมาที่หน้า Dashboard ให้กดปุ่ม Edit แล้วจึงกดปุ่ม “+ Add Panel” นิสิตสามารถปรับขนาดความกว้างของ Panel ได้โดยการกดไอคอนรูปประแจ (Config) และกำหนดชื่อ Panel ใน Title และกำหนดความกว้างด้วยการเลื่อน Columns จากนั้นกดปุ่ม Done และจึงกดปุ่ม Save (คำเตือน ให้กดปุ่ม Save ทุกครั้งที่มีการแก้ไข Dashboard ไม่เช่นนั้นแล้วสิ่งที่ทำจะไม่ถูกบันทึก)
10. จากนั้นทำการเพิ่ม Widget ให้นิสิตกดปุ่ม “+” (ปุ่มลำดับแรกใน Panel) จะปรากฏหน้าต่าง Widget ให้เลือก Type = Indicator Light แล้วกำหนดค่าต่างๆ ดังนี้ Title กำหนดชื่อว่า LED Status ส่วนของ LIGHT ON COLOR กำหนดเป็น #[ชื่อ Device ของนิสิต]["msg"]["led"]=="on" และ LIGHT OFF COLOR กำหนดเป็น #[ชื่อ Device ของนิสิต]["msg"]["led"]=="off" ถัดมา ON TEXT กำหนดว่า ON และ OFF TEXT กำหนด OFF จากนั้นจึงกดปุ่ม Done และ Save ตามลำดับ
11. จากนั้นให้สร้างปุ่มกดเพิ่มเติม ให้นิสิตกดปุ่ม “+” (ปุ่มลำดับแรกใน Panel) จะปรากฏหน้าต่าง Widget ให้เลือก Type = Button แล้วกำหนดค่าต่างๆ ดังนี้ BUTTON CAPTION เป็น ON ถัดมา LABEL เป็น LED ON สีของปุ่ม BUTTON COLOR เป็น Green และ ONCLICK ACTION กำหนดเป็น #[ชื่อ Device ของนิสิต].publishMsg("led","on") จากนั้นจึงกดปุ่ม Done และ Save ตามลำดับ
12. ให้นิสิตกดปุ่ม “+” (ปุ่มลำดับแรกใน Panel) จะปรากฏหน้าต่าง Widget ให้เลือก Type เป็น Button แล้วกำหนดค่าต่างๆ ดังนี้ BUTTON CAPTION เป็น OFF ถัดมา LABEL เป็น LED OFF สีของปุ่ม BUTTON COLOR เป็น Red และ ONCLICK ACTION กำหนดเป็น #[ชื่อ Device ของนิสิต].publishMsg("led","off") จากนั้นจึงกดปุ่ม Done และ Save ตามลำดับ
13. เมื่อกด Save ที่ Dashboard และทดสอบกดปุ่ม ON และ OFF สังเกตสัญลักษณ์หลอดไฟบนหน้าจอ OLED และจะมีข้อความปรากฏเสมือนสถานะของหลอด LED on board หากนิสิตกำหนดการตั้งค่าต่างๆ ถูกต้องสมบูรณ์ ทั้งหน้าการแสดงผลบน Dashboard, หน้าจอ OLED และ LED on board จะทำงานสัมพันธ์กัน

**ข้อควรระวัง!** ใน Source Code IoT-3.ino สำหรับ LED\_BUILTIN 5 ที่ติดมากับบอร์ด IOXESP32+ จะเป็นอุปกรณ์แบบ Active LOW นั้นหมายความว่าความถึามีการจ่าย Logic High ที่ขา 5 หลอดไฟจะดับ หากเป็น LOW หลอดไฟจะติดสว่าง ถ้านิสิตต้องการต่อหลอด LED ที่ GPIO อื่นๆ และอยากทำเป็น Active HIGH คือจ่าย HIGH หลอดไฟติด และ LOW หลอดไฟดับ ตรงคำสั่ง digitalWrite(LED\_BUILTIN, 0); แก้เป็น digitalWrite(27, 1); กรณีที่นิสิตต่อหลอด LED เข้ากับขาที่ 27 และ Common ground

14. ไปที่ <https://trial-api.netpie.io/> กด Authorize ใส่ Username (ClientID) และ Password (Token) แล้วกด Authorize แล้วกด Close ขั้นตอนนี้จะเป็นการทดสอบ RESTful API ของ NETPIE นิสิตสามารถศึกษาเพิ่มเติมจาก <https://docs.netpie.io/device-api.html>

15. กด PUT /device/message แล้วกด Try it out กำหนด topic = led, Request body = on (or off) แล้วกด Execute แล้วดูผลเช่นเดิม หากส่งคำสั่งสำเร็จจะขึ้น 200 OK และควบคุม LED on board ได้ ดังรูป A1-1
16. กดเลือก Event Hooks จากนั้นกดปุ่ม “+ Create” เพื่อสร้าง Event Hook ตั้งชื่อว่า LED\_ON กดปุ่ม Create แล้วกดเลือก Event Hook ที่สร้างไว้ จากนั้น Copy โค้ดในไฟล์ eventhooks.txt ส่วน LED\_ON ไปส่งไป
17. นำ Authorization ใน Responses => Curl ดังรูป A1-2 จากการทดลองในข้อที่ 15 มาใช้แทนแล้วกด Save
18. ให้นิสิตสร้าง Device ตัวที่ 2 ตั้งชื่อว่า MQTTbox ในหน้า Device ของ NETPIE และให้ Copy ทั้ง Client ID, Token และ Secret เก็บไว้ เพื่อนำมาใช้ในการทดลองผ่านโปรแกรม MQTTbox และเลือกที่ Group ทำการจับกลุ่ม Device ทั้ง ESP32 และ MQTTbox เข้าด้วยกันโดยการกดปุ่ม “+ Create” จากนั้นเข้าไปด้านในเลือก Manage Device แล้วทำการติ๊กถูก Device ทั้ง 2 ตัวให้อยู่กลุ่มเดียวกัน แล้วจึงกด Save
19. จากนั้นไปที่ <https://netpie.io/guide> และเลื่อนหาหัวข้อ Must Download => Useful Programs => MQTTbox สามารถใช้ได้ทั้ง Windows และ MacOS เมื่อดาวน์โหลดเสร็จแล้วให้เปิดโปรแกรกดังกล่าว
20. เมื่อโปรแกรม MQTTbox ทำงานแล้วให้นิสิตเลือก Create MQTT Client จะปรากฏรูปที่ A1-3 ให้ทำการกำหนดค่าดังนี้ MQTT Client Name = “MQTTbox2ESP32” MQTT Client Id = “NETPIE Client ID”, Protocol = “mqtt/tcp”, Host = “broker.netpie.io”, Username = “NETPIE Token”, Password = “NETPIE Secret”, Append timestamp to MQTT client id? = No แล้วกด Save
21. ไปที่หน้าต่างฝั่งซ้ายมือ Topic to subscribe กำหนดค่าดังนี้ Topic to publish = “@msg/led”, Payload Type = “Strings/JSON/XML/Characters”, Payload = on (หรือ off) แล้วกด Publish ดูผลหลอดไฟที่บอร์ด ESP32 บนหน้าจอ OLED และ Dashboard จะต้องทำงานสัมพันธ์กัน
22. ไปที่หน้าต่างฝั่งขวามือ Topic to subscribe = “@msg/led” แล้วกด Subscribe แล้วกดเปิดปิด LED จากทั้ง Dashboard และ MQTTBox ดูผลที่ Subscriber ใน MQTTBox
23. ไปที่ส่วน Trigger ของ Device ใน NETPIE จากนั้น Copy โค้ดในไฟล์ trigger.txt ไปส่งแล้วกด Save (สำหรับ NETPIE 2020 ก่อนปี 2565 แต่ถ้าหลังปี 2566 จะต้องสร้าง Trigger เอง โดยการกด + Add trigger จากนั้นตั้งชื่อ Trigger Title เช่น LED\_ON กำหนด Event เป็น SHADOW.UPDATED ในส่วนของ Under conditions ให้กำหนดตามในไฟล์ trigger.txt เช่น \$NEW.altitude > \$PREV.altitude ส่วน Action to ให้กำหนดไปยัง Event hook ที่สร้างขึ้น เช่น LED\_ON แล้วกด Add เมื่อนิสิตสร้างทุก Trigger ครบแล้วให้กด Enabled all นิสิตสามารถศึกษาเพิ่มเติมจาก <https://docs.netpie.io/device-config.html>
24. ทำเช่นข้อ 15-16 แต่เปลี่ยนจาก LED\_ON เป็น LED\_OFF จากนั้นดูผลเช่นเดิม

★ จบการทดลองที่ 1 ★



MQTTBox

MQTTBox Edit Help

Menu MQTT CLIENT SETTINGS

<b>MQTT Client Name</b>	<b>MQTT Client Id</b>
<input type="text" value="MQTT Client Name"/>	<input type="text" value="MQTT Client Id"/>
<b>Protocol</b>	<b>Host</b>
<input type="text" value="mqtt / tcp"/>	<input type="text" value="Host"/>
<b>Username</b>	<b>Password</b>
<input type="text" value="Username"/>	<input type="text" value="Password"/>
<b>Reconnect Period (milliseconds)</b>	<b>Connect Timeout (milliseconds)</b>
<input type="text" value="1000"/>	<input type="text" value="30000"/>
<b>Will - Topic</b>	<b>Will - QoS</b>
<input type="text" value="Will - Topic"/>	<input type="text" value="0 - Almost Once"/>

Save

Client Settings Help

<b>Append timestamp to MQTT client id?</b>	<b>Broker is MQTT v3.1.1 compliant?</b>
<input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes
<b>Clean Session?</b>	<b>Auto connect on app launch?</b>
<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<b>Reschedule Pings?</b>	<b>Queue outgoing QoS zero messages?</b>
<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<b>KeepAlive (seconds)</b>	
<input type="text" value="10"/>	
<b>Will - Retain</b>	<b>Will - Payload</b>
<input type="checkbox"/> No	<input type="text"/>

รูป A1-3 หน้าต่างในโปรแกรม MQTTbox ในการกำหนดค่าต่างๆ เพื่อให้เชื่อมต่อกับ NETPIE

## การทดลองที่ 2 | การประยุกต์ใช้งาน NETPIE

## สิ่งที่ต้องส่ง

มีสิ่งที่ต้องส่ง 2 รายการ คือ

1. ไฟล์ IoT-3\_Report.zip โดยประกอบด้วยไฟล์รายงานและไฟล์ต่าง ๆ ที่แก้ไขยกเว้น credentials.h และ iot\_iconset\_16x16.h โดยส่งใน attachment slot บน mycourseville
2. วิดีโอแสดงการทำงานของอุปกรณ์ โดยส่ง URL ของวิดีโอใน textbox บน mycourseville

## ขั้นตอนปฏิบัติ

1. ให้นิสิตทดลองประยุกต์ใช้งาน NETPIE จากอุปกรณ์ที่ได้รับ เช่น
  - (ตัวอย่าง) ใช้ตัวรับรู้ 1 ตัวเก็บข้อมูลเพื่อนำมาแปลความหมาย (data interpretation) เช่น หาเวลาการตอบสนองของอุณหภูมิเมื่อเปิดปิดเครื่องปรับอากาศในห้อง หาความถี่การกินอาหารของสัตว์เลี้ยง โดยตรวจจับการเคลื่อนไหว
  - (ตัวอย่าง) ใช้ตัวรับรู้ตั้งแต่ 2 ตัวที่ต่างชนิดกันเก็บข้อมูลเพื่อหาความสัมพันธ์ (data correlation) เช่น หาความสัมพันธ์ระหว่างความสว่างของแสงกับเปอร์เซ็นต์ความชื้น
  - (ตัวอย่าง) ใช้ตัวรับรู้ตั้งแต่ 2 ตัวที่ชนิดเดียวกันเก็บข้อมูลเพื่อเปรียบเทียบ (data comparison) เช่น เปรียบเทียบอุณหภูมิจุดที่โดนแดดและจุดที่ไม่โดนแดด เปรียบเทียบความสว่างในแต่ละพื้นที่ ในกรณีนี้ นิสิตอาจสร้าง Device อีกตัวหนึ่ง ส่ง Key ของ Device และโค้ดให้เพื่อนอีกกลุ่มที่อยู่คนละที่ช่วยใส่รหัส WiFi แล้ว Upload ลงบอร์ด ESP32 เพื่อเก็บข้อมูลมาเปรียบเทียบได้
2. เขียนรายงานการทดลอง ประกอบด้วย
  - ที่มาและความสำคัญ
  - วัตถุประสงค์ อย่างน้อย 3 ข้อ
  - ประโยชน์ที่คาดว่าจะได้รับ
  - ขั้นตอนการทำงาน, Flow chart/Blackbox และ Schematic diagram
  - อภิปรายผล และสรุปผล
  - แหล่งอ้างอิง
  - ภาคผนวก อธิบาย Source code

★ จบการทดลองที่ 2 ★