



## COMPUTER ENGINEERING PROGRAM FACULTY OF ENGINEERING, THAI-NICHI INSTITUTE OF TECHNOLOGY

---

|                |   |   |
|----------------|---|---|
| ปีการศึกษา     | : | 2/2567                                      |
| หลักสูตร       | : | Special Topic in AI and IoT on Raspberry Pi |
| ผู้สอน         | : | อาจารย์อัตนा เชzn็อติ๊ส                     |
| การทดลองที่ 05 | : | การพัฒนา IoT ด้วย Node-RED บน RPi part II   |

---

### 1. วัตถุประสงค์

- 1.1 เพื่อทำความเข้าใจเกี่ยวกับการพัฒนา IoT
- 1.2 เพื่อทำความเข้าใจเกี่ยวกับการพัฒนา IoT ด้วย Node-RED บน RPi part II

### 2. สมาชิกกลุ่ม

- 2.1 ชื่อJiraprapa Kanavong รหัสนักศึกษา ....2211311564....  
2.2 ชื่อNuttamon Chanseeda รหัสนักศึกษา 2211310764  
2.3 ชื่อKachachan Ruanpetch.รหัสนักศึกษา..2211311721....  
2.4 ชื่อศศิกานต์ โคงรปีติพุฒ รหัสนักศึกษา.2211311960  
2.5 หมายเลขอร์ด.....4.....

การพัฒนา IoT ด้วย Node-RED บน Raspberry Pi (RPi) เป็นกระบวนการที่ง่ายและทรงพลังสำหรับการสร้างระบบ IoT โดยใช้การเขียนโปรแกรมแบบลากและวางในอินเทอร์เฟซที่ใช้งานง่าย Node-RED ซึ่งติดตั้งได้บน RPi ช่วยให้นักพัฒนาสามารถเชื่อมต่ออุปกรณ์ IoT เช่นเซอร์ และระบบต่างๆ เข้าด้วยกันผ่านโนนด (Nodes) ที่ออกแบบมาเฉพาะสำหรับการประมวลผลข้อมูล การสื่อสาร และการควบคุม โดยไม่จำเป็นต้องมีความเชี่ยวชาญด้านการเขียนโค้ดมากนัก ทำให้เหมาะสมสำหรับการพัฒนาโครงการ IoT ทั้งในเชิงทดลองและการใช้งานจริงอย่างมีประสิทธิภาพและประหยัดเวลา

| **รูป (OPTIONAL)**

The diagram illustrates a system for monitoring people entering a building. A person in a red dress stands on a yellow mat labeled "temperature sensor". Another person in a white suit stands near a door. A third person in a black suit is being checked with a thermometer. Red arrows point from the text labels to the respective sensors: "แทนด้วย ultrasonic sensor" points to the thermometer, and "แทนด้วย temperature sensor" points to the mat.

Sqlite Database Dashboard

| index | checking | temp | timestamp           |
|-------|----------|------|---------------------|
| 1     | found1   | 31   | 2024-01-20 10:39:04 |
| 2     | found2   | 31   | 2024-01-20 10:39:13 |
| 3     | found3   | 31   | 2024-01-20 10:39:16 |
| 4     | found4   | 31   | 2024-01-20 10:39:18 |

Recode Data

## วิธีการกรอกคำตอบลงใน worksheet และส่ง worksheet ไปยัง Google Classroom

- 1) เปิดใบงาน Lab04\_starting\_with\_IoT\_using\_NodeReD ใน Google classroom
- 2) ให้นศ. กรอกคำตอบ/Capture หน้าจอ (ให้สังเกตส่วนรูปที่ภาพที่มีคำว่า “ตัวอย่าง”) ลงใต้ตัวอย่างของคำตอบในใบงาน โดยจัดให้สวยงามและสมบูรณ์
- 3) เมื่อตรวจสอบความถูกต้อง ครบถ้วนสมบูรณ์แล้ว ให้นศ. กดส่งงานภายในเวลาที่กำหนด ตามไฟล์ที่ได้จัดเตรียมไว้
- 4) หากมีคำถามท้ายการทดลอง ให้นศ. ตอบให้ครบถ้วนสมบูรณ์

จากใบงานครั้งที่ 04 Lab04\_starting\_with\_IoT\_using\_NodeReD เป็นที่เรียบร้อยแล้ว ให้นักศึกษาทำแบบฝึกหัดเกี่ยวกับการพัฒนางาน IoT ด้วย Node-RED บน RPi ซึ่งมีดังต่อไปนี้

### **โจทย์: ควบคุม LED ให้สว่างด้วย PWM ด้วย NODE-RED**

ขั้นตอนของการควบคุม LED ให้สว่างด้วย Node-RED มีดังนี้

- 1) เตรียมฮาร์ดแวร์บอร์ด RPI + PC computer (เชื่อมต่อด้วย VNC viewer)
- 2) บน VNC viewer ให้เปิดโปรแกรม NODE-RED และบน PC ให้เปิดเว็บбраузเซอร์ และพิมพ์ IP 192.168.50.201 เพื่อเปิดใช้งาน Node-RED
- 3) บนบอร์ด RPI ให้เชื่อมต่อฮาร์ดแวร์ LED
- 4) สร้าง Flow และ Dashboard ควบคุมการส่องสว่าง LED
- 5) Deploy และทดสอบการทำงาน
- 6) ปิดเครื่อง + ปรับ IP address + เก็บอุปกรณ์

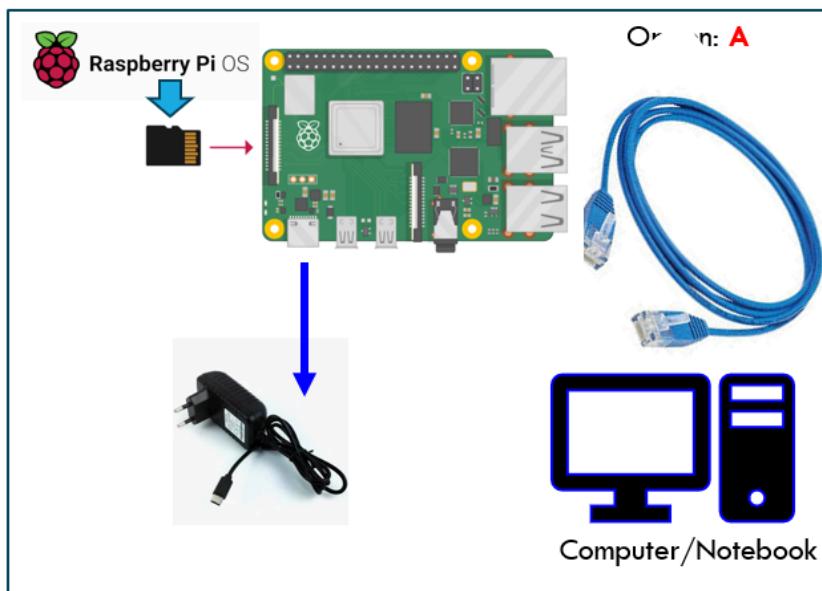
### **แบบฝึกหัดที่ 1: เตรียมฮาร์ดแวร์บอร์ด RPI + PC computer**

1) อุปกรณ์ที่ต้องเตรียมเพื่อใช้งาน

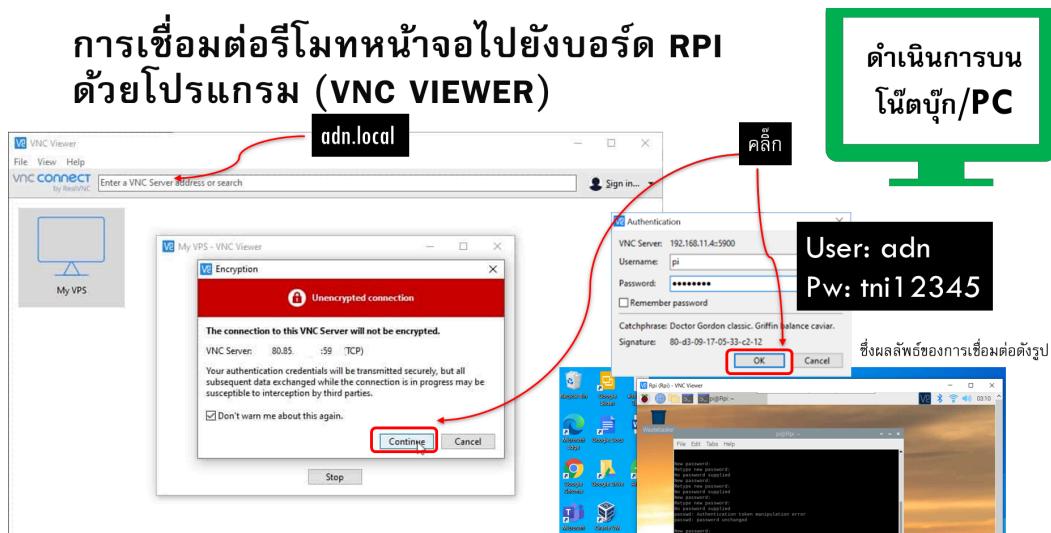
- คอมพิวเตอร์ตั้งโต๊ะ/หรือโน๊ตบุ๊ค
- บอร์ดร้าสเบอร์รีพาย

เมื่อล็อกอินคอมพิวเตอร์แล้วให้ตรวจสอบโปรแกรมที่ต้องใช้สำหรับการเชื่อมต่อดังนี้

- a) ตรวจสอบ Putty ได้ติดตั้งหรือยัง? หากยัง ให้ดาวน์โหลด <https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.82-installer.msi> และติดตั้งให้เรียบร้อย (หากมีการถาม user,pw สำหรับแอดมินเพื่อติดตั้งให้ป้อน (user: labinstall, pw: Tni12345))
- b) ตรวจสอบ RealVNC viewer ได้ติดตั้งหรือยัง? หากยัง ให้ดาวน์โหลด <https://downloads.realvnc.com/download/file/viewer.files/VNC-Viewer-7.1.3.0-Windows.exe> และติดตั้งให้เรียบร้อย (หากมีการถาม user,pw สำหรับแอดมินเพื่อติดตั้งให้ป้อน (user: labinstall, pw: Tni12345))
- c) เมื่อติดตั้งโปรแกรมทั้งสองเป็นที่เรียบร้อยแล้ว ให้ถอนสาย LAN ออกจากอุปกรณ์ที่เสียบของໂต๊ะ เลวนำไปเสียบที่บอร์ดraspberrypithen

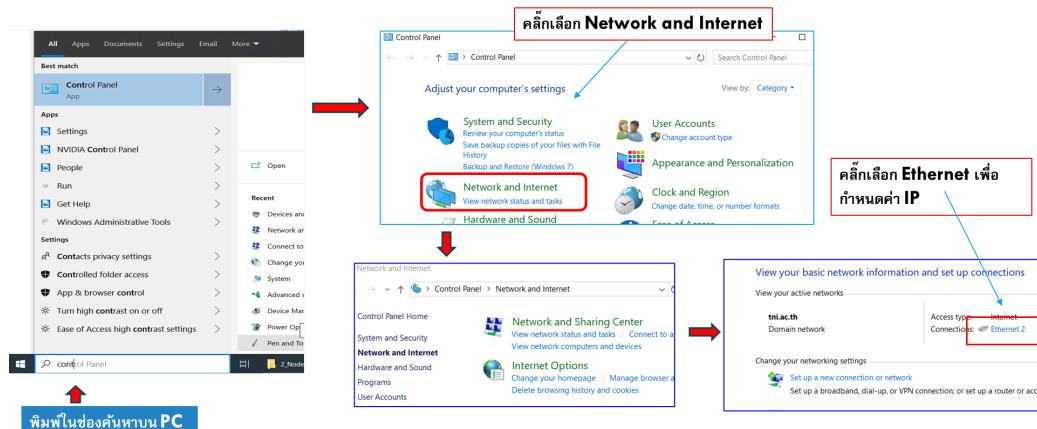


- 2) เมื่อเตรียมฮาร์ดแวร์ดังนี้รูปข้างต้นแล้ว ไปที่ PC ให้ดำเนินการเชื่อมต่อดังรูปที่ 2.1 ด้วย โปรแกรม RealVNC viewer

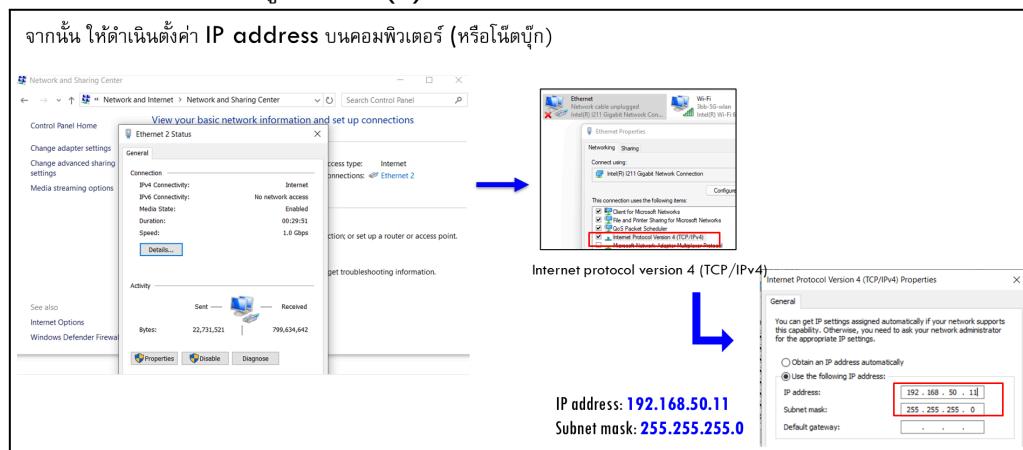


รูปที่ 2.1 ฮาร์ดแวร์และการเชื่อมต่อด้วย PC หรือ Notebook

- 3) (**หากเชื่อมต่อได้ให้ข้ามไปทำข้อที่ 4**) หากดำเนินการตามข้อ 2.1 แล้วเชื่อมต่อไม่ได้ ให้เตรียมฮาร์ดแวร์ดังนี้รูป ดังรูปที่ 2.2 จากนั้นบน PC ให้ดำเนินการตั้งค่า Fix IP address ดังรูปที่ 2.2 (a), (b)

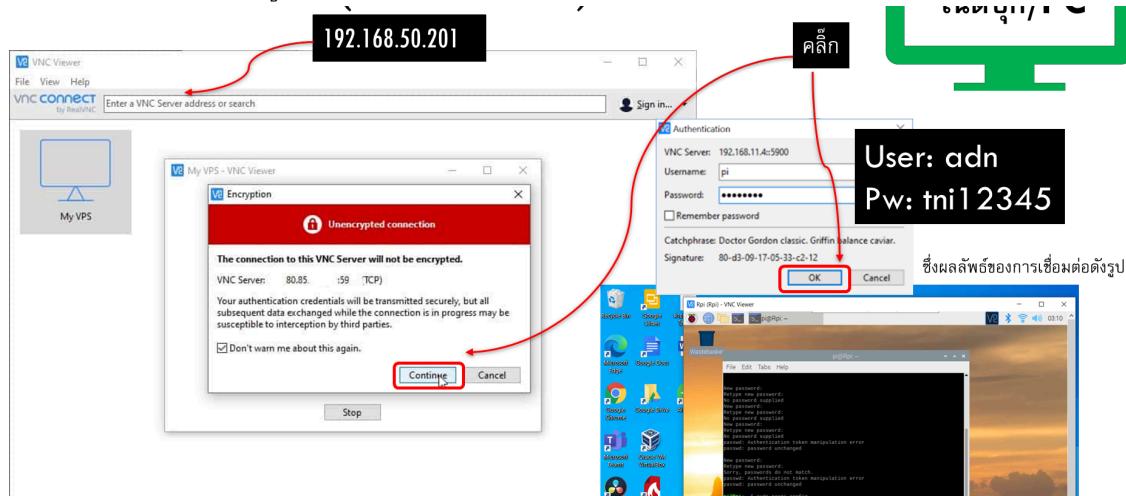


รูปที่ 2.2 (a) การตั้งค่า Fix IP address



รูปที่ 2.2 (b) การตั้งค่า Fix IP address

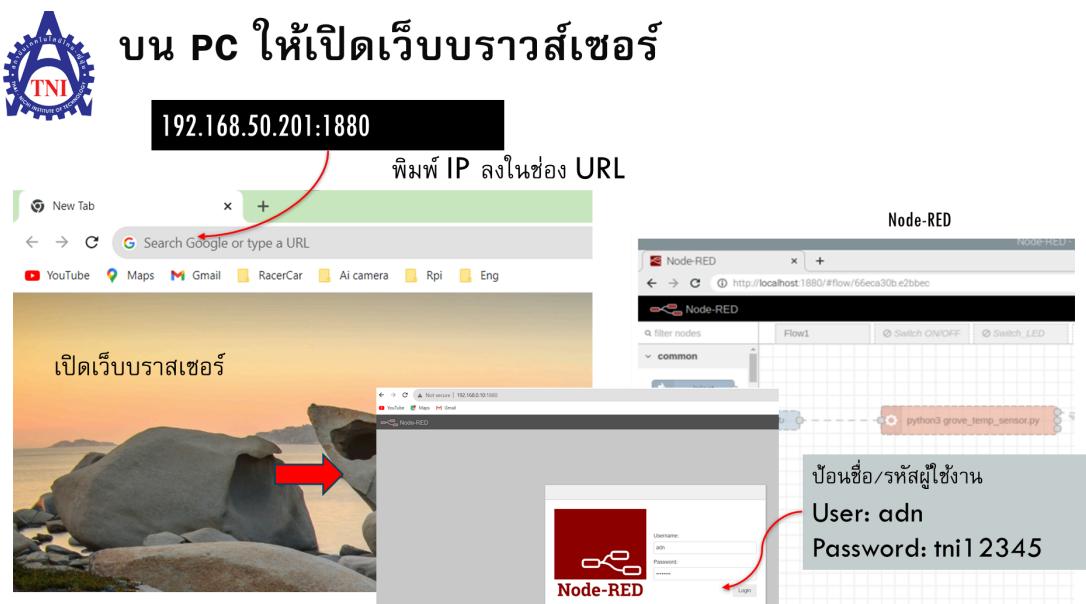
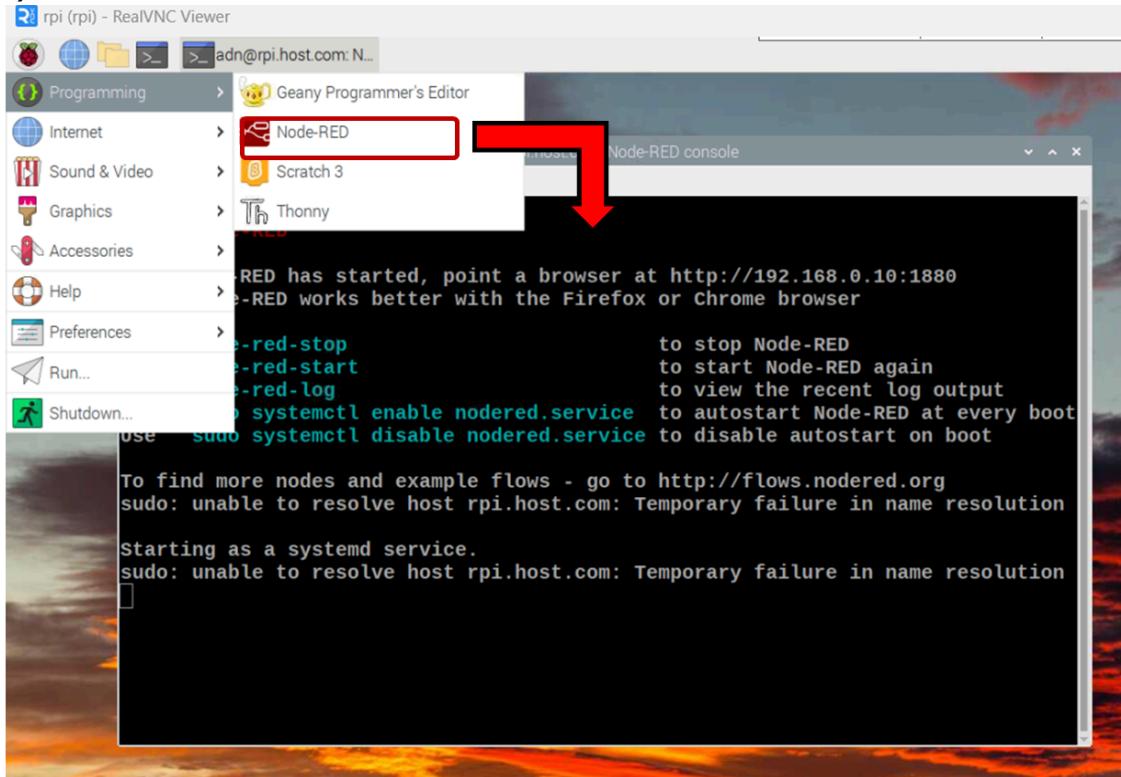
เปิดโปรแกรม RealVNC viewer (บน PC) โดยป้อน IP address: 192.168.50.201 ลงในช่อง Connection ดังรูป



รูปที่ 2.3 โปรแกรม RealVNC viewer (บน PC)

## แบบฝึกหัดที่ 2: บน VNC viewer ให้เปิดโปรแกรม NODE-RED และ พิมพ์ IP

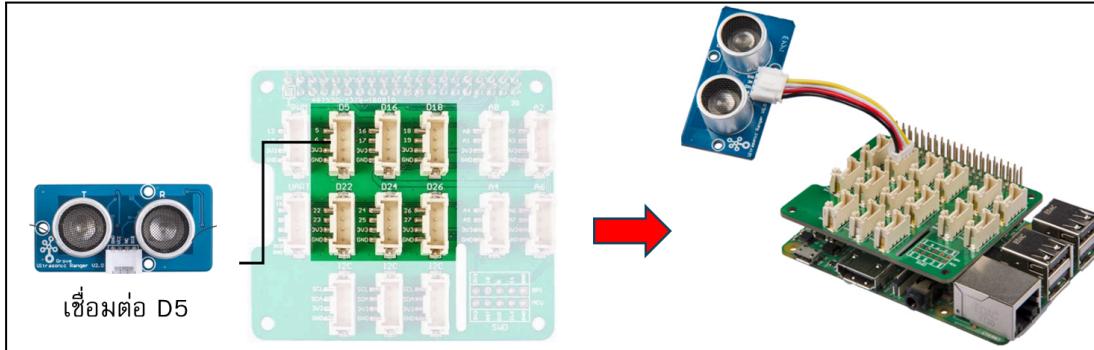
### 1) จับภาพหน้าจอผลลัพธ์ของการเชื่อมต่อพร้อมเปิดโปรแกรม NODE-RED



### แบบฝึกหัดที่ 3: การใช้เซนเซอร์ด้วยโหนดจาก node-red-contrib-grove-base-hat

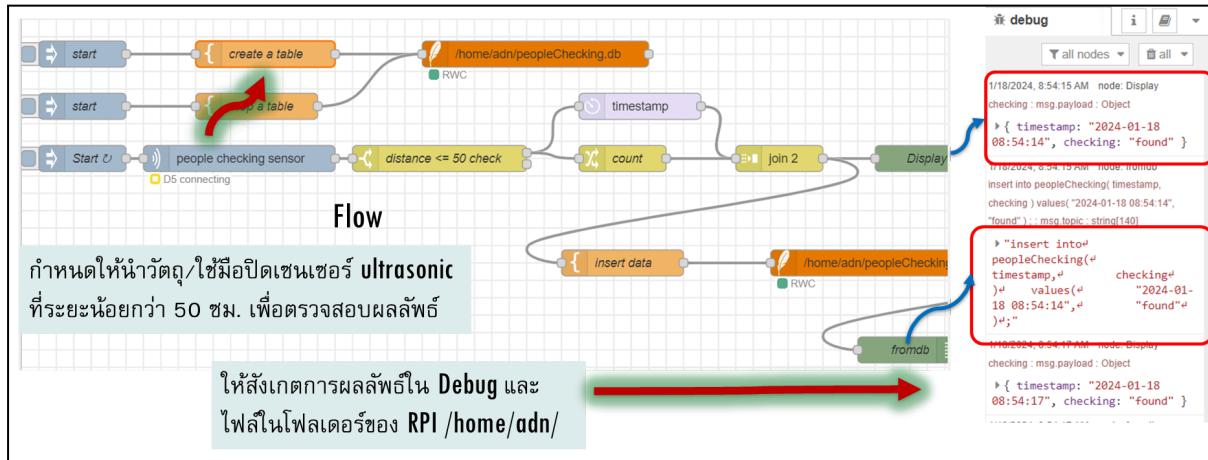
#### 1. สร้าง Flow แสดงผล Ultrasonic – ด้วย Debug

- เชื่อมต่อฮาร์ดแวร์ดังรูป



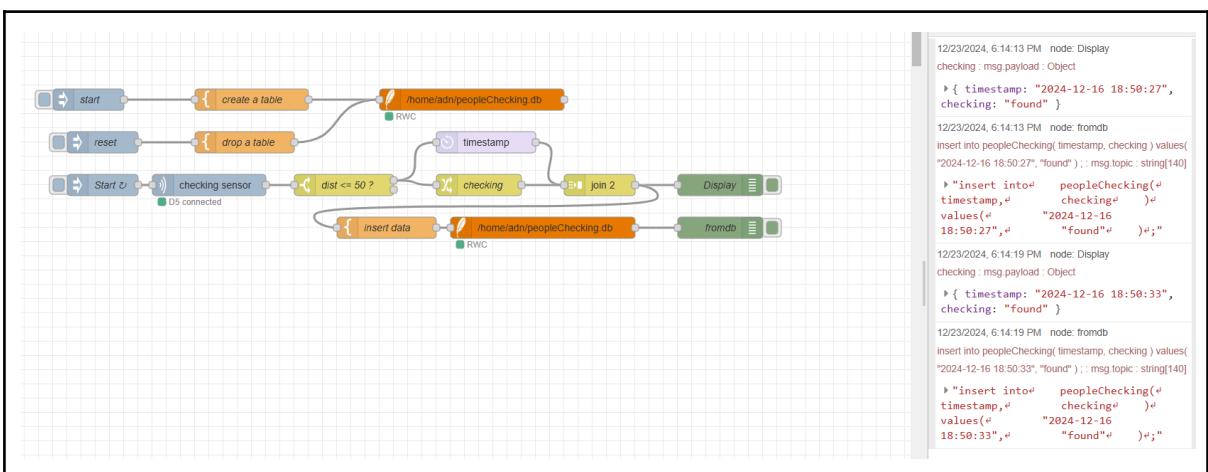
## แบบฝึกหัดที่ 4: การใช้ Grove เซนเซอร์เพื่อสร้างโปรเจคเก็บข้อมูล

- สร้าง Flow เพื่อตรวจสอบคนเข้า-ออกด้วย Ultrasonic บันทึกเวลาเพื่อจัดเก็บเป็นฐานข้อมูล (ให้ทำตามสไลด์บทที่ 5) โดยใช้รูปต่อไปนี้

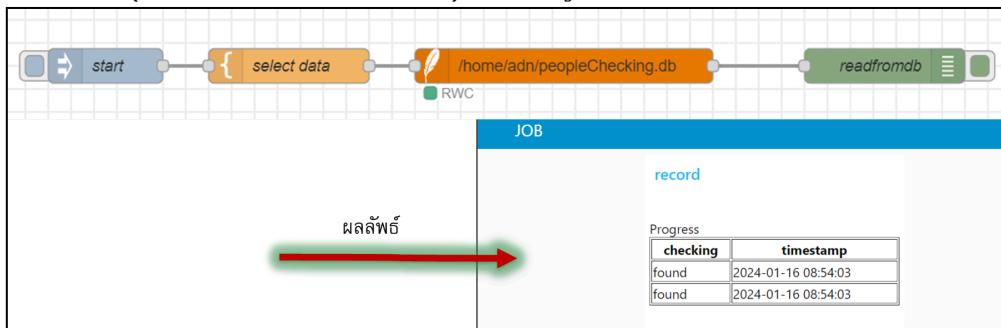


- เมื่อเสร็จสิ้นขั้นตอนการร้าด Flow จากนั้นให้คลิกที่ปุ่ม Deploy สีแดงขนาดใหญ่ที่ด้านบน ข่าวของหน้าจอ ข้อความควรปรากฏขึ้นที่ด้านบนว่า “Successfully deployed” (ติดตั้งสำเร็จแล้ว)

-----จับภาพหน้าจอของคำสั่งและผลลัพธ์-----



- สร้าง Flow เพื่ออ่านค่าจากฐานข้อมูล “peopleChekcing.db” และแสดงผลบน Dashboard (ให้ทำตามสไลด์บทที่ 5) โดยใช้รูปต่อไปนี้



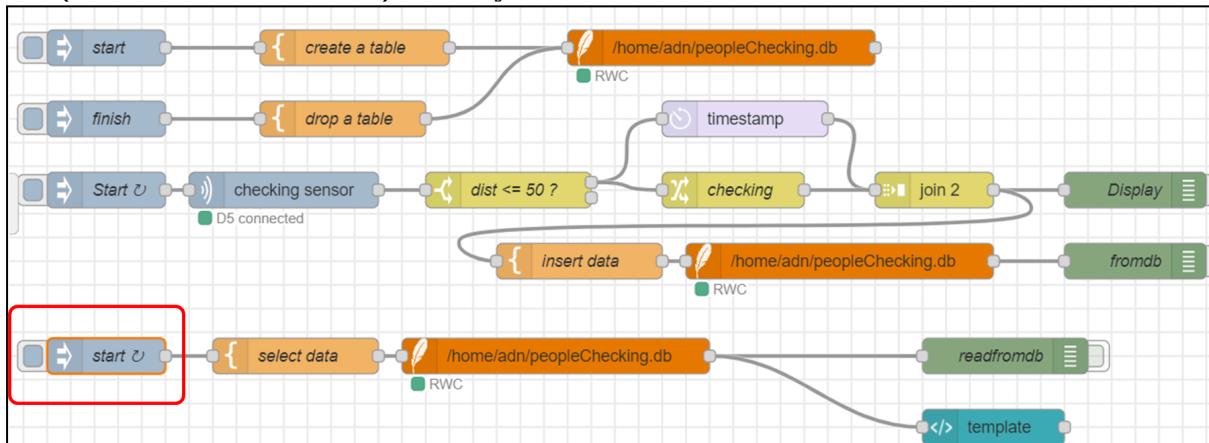
- เมื่อเสร็จสิ้นขั้นตอนการร้าด Flow จากนั้นให้คลิกที่ปุ่ม Deploy  
-----จับภาพหน้าจอของคำสั่งและผลลัพธ์-----

## Recode Data

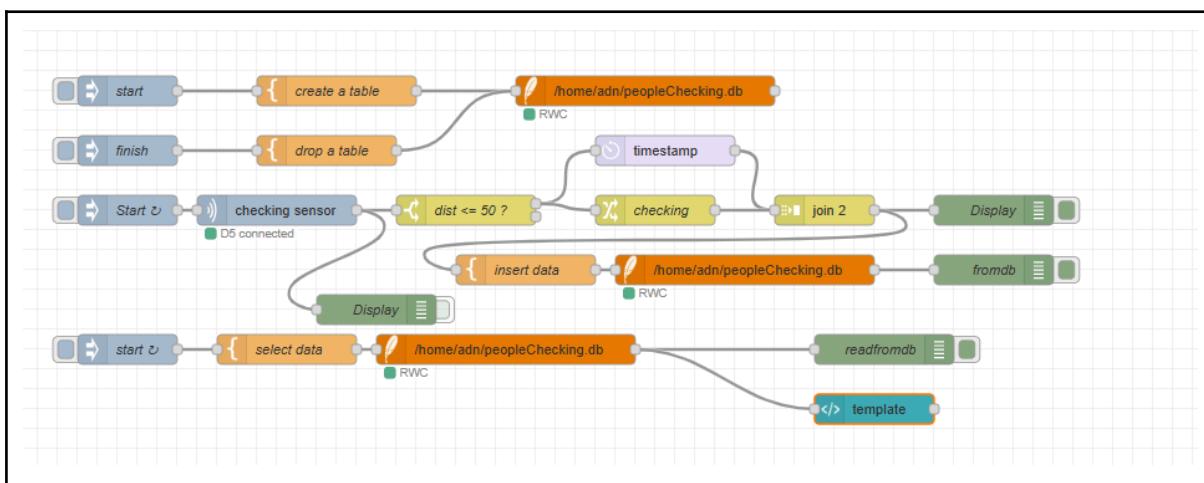
| index | checking | timestamp           |
|-------|----------|---------------------|
| 1     | found1   | 2024-12-23 18:24:42 |
| 2     | found2   | 2024-12-23 18:25:27 |
| 3     | found3   | 2024-12-23 18:25:28 |
| 4     | found4   | 2024-12-23 18:25:28 |
| 5     | found5   | 2024-12-23 18:25:31 |
| 6     | found6   | 2024-12-23 18:25:32 |

### แบบฝึกหัดที่ 5: การใช้ Grove เขียนเซอร์ฟิเวต์เพื่อสร้างโปรเจคเก็บข้อมูล

1. สร้าง Flow เพื่ออ่านค่าจากฐานข้อมูล และอัปเดตค่าอัตโนมัติ (ให้ทำงานสไลด์บบที่ 5) โดยใช้รูปด้านไปนี้



-----จับภาพหน้าจอของคำสั่งและผลลัพธ์-----



The screenshot shows two windows side-by-side. On the left is the Node-RED interface with a flow consisting of an inject node, a switch node, a function node, and a catch node. The function node contains a script that logs the timestamp and checking value. On the right is a browser window displaying a table titled 'record' with columns 'checking' and 'timestamp'. The table contains five rows of data.

| checking | timestamp           |
|----------|---------------------|
| found    | 2024-01-16 08:54:03 |
| found    | 2024-01-16 08:54:03 |
| found    | 2024-01-16 10:18:17 |
| found    | 2024-01-16 10:18:29 |
| found    | 2024-01-16 10:18:46 |

## 2. เมื่อเสร็จสิ้นขั้นตอนการร้าด Flow จากนั้นให้คลิกที่ปุ่ม Deploy -----จับภาพหน้าจอของคำสั่งและผลลัพธ์-----

The screenshot shows a terminal window with a command-line interface. On the left is a table titled 'Recode Data' with columns 'index', 'checking', and 'timestamp'. On the right is the output of a Node.js script. The script inserts data into a database table named 'peopleChecking' with columns 'timestamp' and 'checking'. The data is inserted in batches of 20 rows each, with a timestamp of '2024-12-23 18:26:09' and a checking value of 'found'.

| index | checking | timestamp           |
|-------|----------|---------------------|
| 1     | found1   | 2024-12-23 18:24:42 |
| 2     | found2   | 2024-12-23 18:25:27 |
| 3     | found3   | 2024-12-23 18:25:28 |
| 4     | found4   | 2024-12-23 18:25:28 |
| 5     | found5   | 2024-12-23 18:25:31 |
| 6     | found6   | 2024-12-23 18:25:32 |
| 7     | found7   | 2024-12-23 18:25:33 |
| 8     | found8   | 2024-12-23 18:25:37 |
| 9     | found9   | 2024-12-23 18:25:38 |
| 10    | found10  | 2024-12-23 18:25:39 |
| 11    | found11  | 2024-12-23 18:25:40 |
| 12    | found12  | 2024-12-23 18:25:41 |
| 13    | found13  | 2024-12-23 18:25:42 |
| 14    | found14  | 2024-12-23 18:25:43 |
| 15    | found15  | 2024-12-23 18:25:44 |
| 16    | found16  | 2024-12-23 18:25:45 |
| 17    | found17  | 2024-12-23 18:25:46 |
| 18    | found18  | 2024-12-23 18:25:48 |
| 19    | found19  | 2024-12-23 18:25:49 |
| 20    | found20  | 2024-12-23 18:25:50 |

```

object, object, object, object,
object, object ... ]

12/23/2024, 6:25:54 PM node: Display
checking : msg.payload : Object
▶ { timestamp: "2024-12-23
18:26:09", checking: "found" }

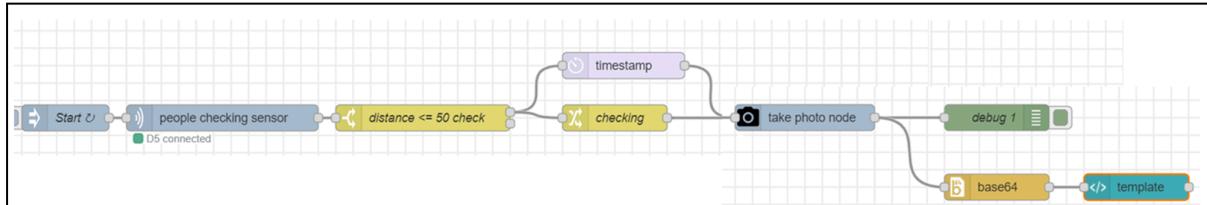
12/23/2024, 6:25:54 PM node: fromdb
insert into peopleChecking( timestamp, checking )
values( "2024-12-23 18:26:09", "found" ); :: msg.topic :
string[140]
▶ "insert into` peopleChecking(` timestamp,` checking` )` values(` 2024-12-23
18:26:09`, ` found` )";"

12/23/2024, 6:25:55 PM node: readfromdb

```

## แบบฝึกหัดที่ 6: การใช้ Grove เซนเซอร์เพื่อสร้างโปรเจคเก็บข้อมูล (Optional)

- สร้าง Flow เพื่ออ่านค่าจากฐานข้อมูล และแสดงผลบน Dashboard และอัปเดตค่าอัตโนมัติ (ให้ทำตามสไลด์บทที่ 5) โดยใช้รูปต่อไปนี้



-----จับภาพหน้าจอของคำสั่งและผลลัพธ์-----



- เมื่อเสร็จสิ้นขั้นตอนการร่าง Flow จากนั้นให้คลิกที่ปุ่ม Deploy

The screenshot shows the Node-RED interface with a deployed flow and an open SQLite Database Dashboard. The flow consists of a 'start' node, a 'reset' node, and a 'Start' node connected to a 'take photo node'. The database dashboard displays a log of events and a table with two entries.

SQLite Database Dashboard

Recode Data

ผลลัพธ์

| index | checking | temp | timestamp           |
|-------|----------|------|---------------------|
| 1     | found1   | 28   | 2024-01-26 22:20:06 |
| 2     | found2   | 28   | 2024-01-26 22:20:07 |

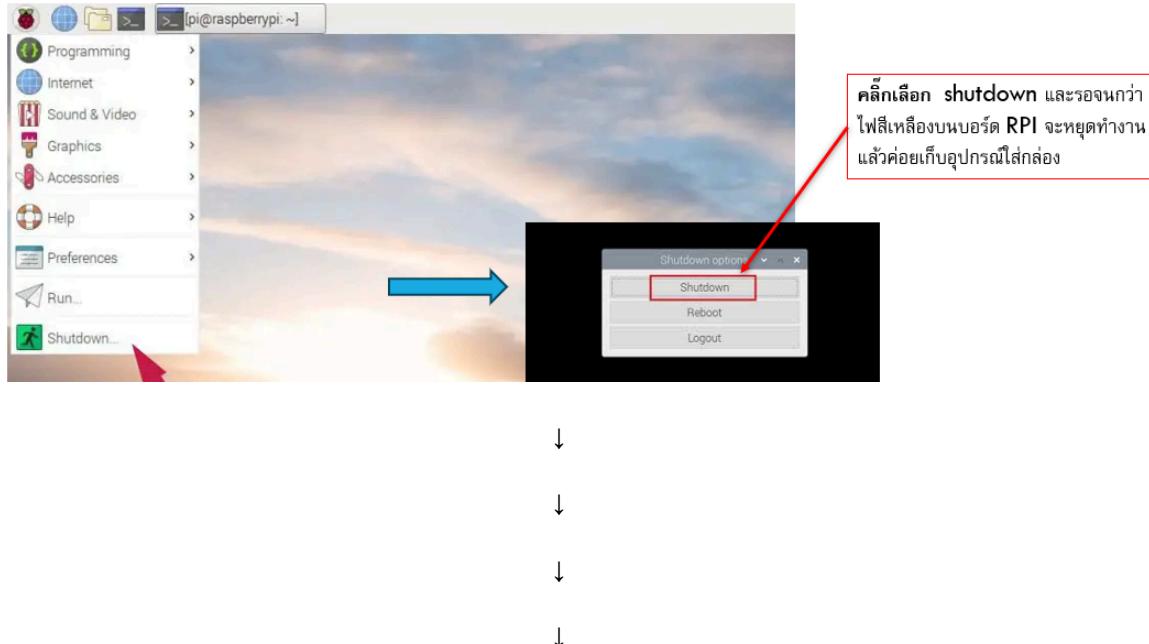
-----จับภาพหน้าจอของคำสั่งและผลลัพธ์-----



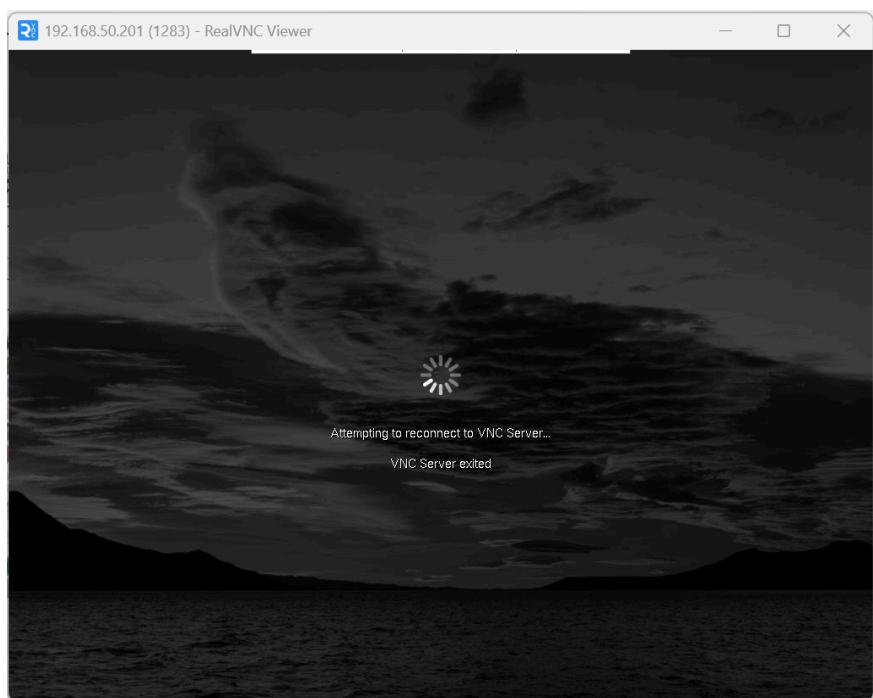
## แบบฝึกหัดที่ 7: เมื่อใช้งานเสร็จขอให้หน้าจอปิดเครื่อง + ปรับ IP address + เก็บอุปกรณ์

1) ที่โปรแกรม VNC viewer ให้ปิดบอร์ดราสเบอร์รี่พาย (ให้รอสักประมาณ 1 นาทีเพื่อให้ปิดเครื่องสมบูรณ์)

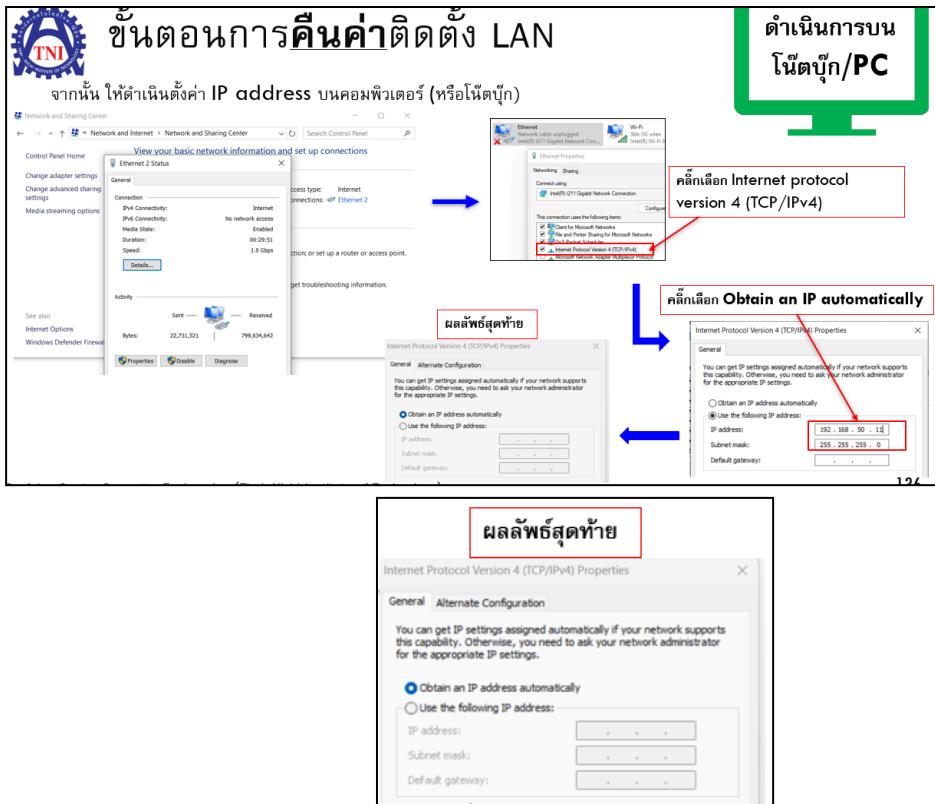
หลังจากที่มีการใช้งานบอร์ดราสเบอร์รี่พายเสร็จเป็นที่เรียบร้อยแล้วให้ปิดเครื่อง (Shutdown) ทุกครั้งก่อน ถอดสายอะแดปเตอร์



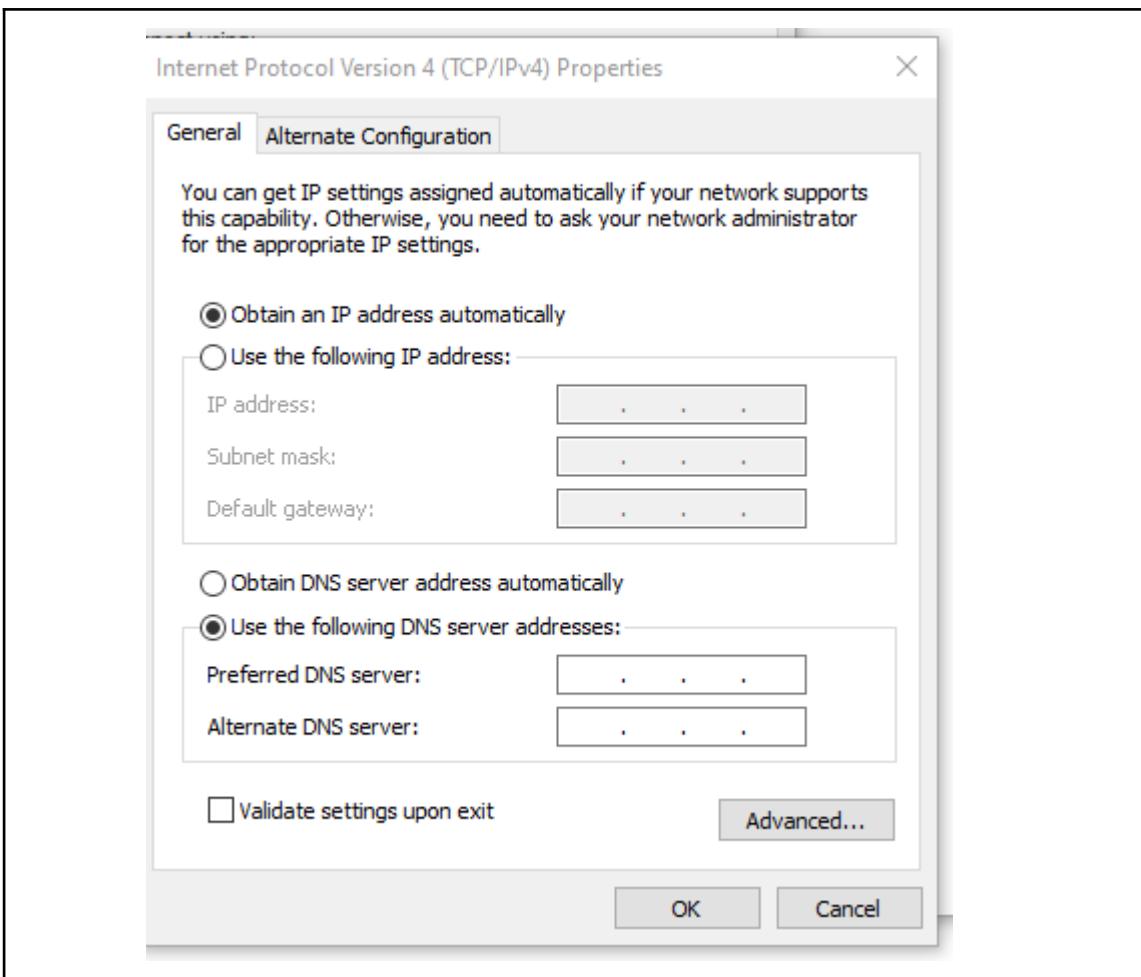
รอให้ไฟสถานะ LAN ดับ ถึงแม้ว่าหน้าจอคอมพิวเตอร์จะขึ้นดังรูปด้านบน



## 2) บันคอมพิวเตอร์ให้ดำเนินการปรับ IP address ให้เป็นแบบ Obtain Automatically



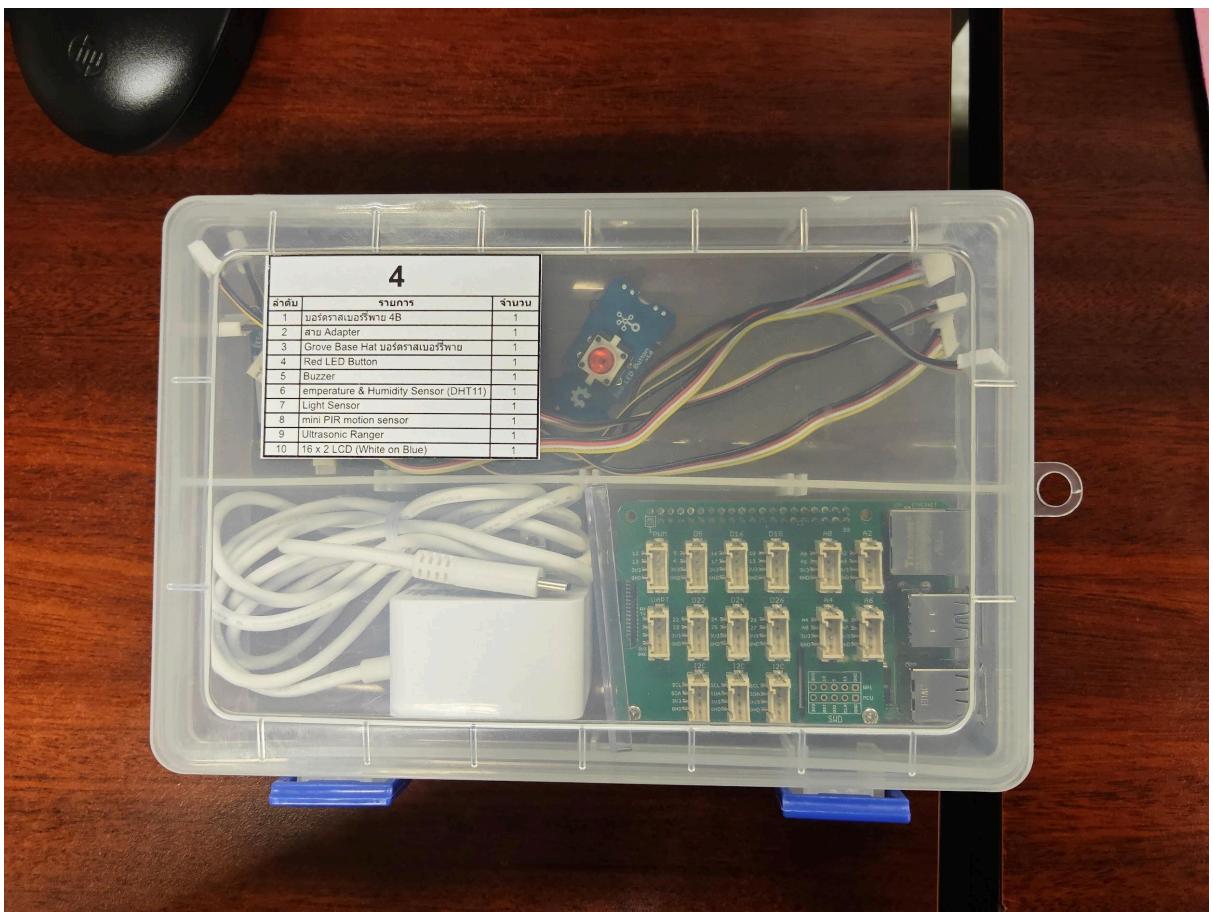
-----จับภาพหน้าของคำสั่งและผลลัพธ์-----



3) นำอุปกรณ์ต่างๆ ใส่ลงกล่อง (ถ่ายรูป) และคืนอุปกรณ์



## จับภาพหน้าของคำสั่งและผลลัพธ์



### คำถาม

#### 1. โหนด msg.payload คืออะไร? และทำหน้าที่อะไร

msg.payload เป็นคุณสมบัติ (property) หลักของข้อความ (message) ใน Node-RED ที่ใช้สำหรับจัดเก็บข้อมูลหรือค่าที่ส่งผ่านไปยังโหนดต่าง ๆ ใน flow หน้าที่: เป็นตัวเก็บข้อมูลหลักที่ใช้ในการประมวลผล เช่น ข้อความ ข้อมูล JSON หรือค่าต่าง ๆ ที่ต้องส่งต่อไปยังโหนดถัดไป

#### 2. โหนด msg.topic คืออะไร? และทำหน้าที่อะไร

msg.topic เป็นอีกคุณสมบัติหนึ่งของข้อความใน Node-RED ที่มักใช้ระบุหัวข้อ (topic) หรือคำอธิบายของข้อมูลใน msg.payload หน้าที่: ช่วยแยกหรือจัดประเภทข้อมูล เช่น ระบุประเภทข้อมูลใน payload หรือใช้ในเงื่อนไขการทำงานของโหนด

#### 3. โหนด template ทำหน้าที่อะไร

โหนด template ใช้สำหรับสร้างข้อความหรือข้อมูลที่มีโครงสร้าง โดยใช้เทมเพลต (template) ที่กำหนดเอง เช่น HTML, JSON, หรือข้อความธรรมดा หน้าที่: ช่วยจัดรูปแบบหรือสร้างข้อความใหม่โดยอ้างอิงค่าจากข้อความ (message) เช่น msg.payload

#### 4. โหนด sqlite ทำหน้าที่อะไร

โหนด sqlite เป็นโหนดที่ใช้สำหรับเชื่อมต่อและจัดการฐานข้อมูล SQLite ใน Node-RED

หน้าที่: ทำการ Query, Insert, Update, หรือ Delete ข้อมูลในฐานข้อมูล SQLite ใช้สำหรับจัดการข้อมูลใน flow ที่ต้องการฐานข้อมูล.

## 5. โหนด join ทำหน้าที่อะไร

โหนด join ใช้รวมข้อความหลายข้อความ (messages) หรือข้อมูลหลายส่วนเข้าด้วยกัน

หน้าที่:

รวมข้อมูลจากโหนดที่ส่งมาหลายทาง

สร้าง array หรือ object จากข้อมูลที่แยกส่งมาหลายส่วน

## 6. เมื่อต้องการเพิ่มข้อมูลลงในฐานข้อมูลต้องใช้โหนดอะไร?

ต้องใช้ โหนด sqlite หรือ โหนด mysql (ขึ้นอยู่กับประเภทฐานข้อมูล) เพื่อทำการเพิ่มข้อมูล (Insert) ลงในฐานข้อมูล  
ข้อมูลที่ต้องการเพิ่มมักจะต้องอยู่ใน msg.payload และมีคำสั่ง SQL ที่ถูกต้องสำหรับการเพิ่มข้อมูล