# อินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะ

# SMART GARDEN SYSTEM OF IoT

นายทวีป ตรีหะจินดารัตน์
นายทศพร ปั้นจาด
นายปวรัชฎ์ คชรินทร์

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2559

# อินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะ

# SMART GARDEN SYSTEM OF IoT


นายทวีป ตรีหะจินดารัตน์

นายทศพร ปั้นจาด

นายปวรัชฎ์ คชรินทร์


โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ปีการศึกษา 2559


ลิขสิทธิ์เป็นของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

โครงงานเรื่อง

อินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะ

ของ

นายทวีป ตรีหะจินดารัตน์

นายทศพร ปั้นจาด

นายปวรัชฎ์ คชรินทร์

ได้รับอนุมัติจากคณะวิศวกรรมศาสตร์ให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.เวคิน ปิยรัตน์)

คณะกรรมการการสอบโครงงาน

.....ประธาน

(ผู้ช่วยศาสตราจารย์ ดร.ชาญไชย ไทยเจียม)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.นำคุณ ศรีสนิท)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ศิริพงษ์ ฉายสินธ์)

# อินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะ
# ปีการศึกษา 2559

**โดย**

นายทวีป ตรีหะจินดารัตน์
นายทศพร ปั้นจาด
นายปวรัชฎ์ คชรินทร์

**อาจารย์ที่ปรึกษา**

ผู้ช่วยศาสตราจารย์ศิริพงษ์ ฉายสินธ์

## บทคัดย่อ

เนื่องจากในปัจจุบันเทคโนโลยีสารสนเทศและการสื่อสารต่างๆ เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น และได้พัฒนาการเชื่อมต่อระหว่างอินเตอร์เน็ตกับอุปกรณ์ต่างๆ เรียกว่าอินเตอร์เน็ตกับทุกสิ่ง ในโครงงานนี้เป็นการนำเสนออินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะ เป็นระบบจัดการดูแลสวนที่สามารถเก็บข้อมูลสภาพแวดล้อม นำมาประมวลผลและทำงานตามเงื่อนไขได้อย่างอัตโนมัติ แล้วยังสามารถควบคุมการทำงานของระบบนี้ได้จากระยะไกลด้วยเทคโนโลยีสารสนเทศ สร้างความสะดวกสบายและง่ายต่อการดูแล โดยนำอินเตอร์เน็ตกับทุกสิ่งมาใช้งานร่วมกับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ เป็นการใช้งานโนดเอ็มซียู (Node MCU) เป็นตัวควบคุมการทำงานของระบบ เขียนคำสั่งผ่านซอฟแวร์ที่มีชื่อว่าอาร์ดูไอโน ไอดีอี ที่เป็น open source ด้วยภาษา C/C++ บรรจุอยู่ในตัวโนดเอ็มซียู โดยควบคุมการทำงานของเซนเซอร์วัดค่าอุณหภูมิและความชื้นในอากาศ (DHT22) เซนเซอร์วัดค่าความชื้นในดิน (Soil Moisture Sensor) และ โซลินอยวาวล์ (Solenoid Valve) สามารถบันทึกค่าอุณหภูมิกับความชื้นที่ได้รับจากเซนเซอร์และควบคุมการทำงานของระบบนี้จากระยะไกลได้ผ่านเว็บบราวเซอร์

**คำสำคัญ:** โนดเอ็มซียู อินเตอร์เน็ตกับทุกสิ่ง

# SMART GARDEN SYSTEM OF IoT

## Academic Year 2016

**By**

Mr. Tavip Treehajindarat

Mr. Todsaporn Punjard

Mr. Pawarat Kotcharin

**Advisor**

Asst. Prof. Siripong Chaysin

## Abstract

Nowadays, information technology and communications have been very popular and have developed a connection between the Internet and devices. It is called the internet of things. This project present is about smart garden system of IoT. It is a smart garden system that can store environmental information. That can processing and operation automatically. It also controls the operation of this system anytime and anywhere with information technology. It is made convenient and easy to care for the garden. The Node MCU is a control system. Which commands through software called Arduino IDE. The program is an open source that has C/C++ language for writing command. The Node MCU has controlled temperature humidity sensor, soil moisture sensor and solenoid valve. It also records the temperature and humidity of the sensor too.

**Keyword** : Node MCU, Internet of Thing

# กิตติกรรมประกาศ

โครงงานวิศวกรรมนี้ สามารถสำเร็จได้ด้วยความช่วยเหลือจากคณาจารย์และผู้มีพระคุณหลายท่าน คณะผู้จัดทำขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ศิริพงษ์ ฉายสินธ์ อาจารย์ที่ปรึกษาโครงงานที่ได้กรุณาเสียสละเวลา เพื่อให้คำปรึกษา คำแนะนำ การชี้แนะแนวทางในการหาคำตอบในปัญหาต่างๆ ที่เกิดขึ้นตลอดระยะเวลาการจัดทำโครงงานด้วยความเอาใจใส่อย่างยิ่ง ตลอดจนการสอบถามความคืบหน้า ตรวจสอบ และแก้ไขโครงงานนี้จนสำเร็จโดยสมบูรณ์

ขอขอบคุณ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ องครักษ์ และภาควิชาวิศวกรรมไฟฟ้าให้เอื้ออำนวยสถานที่และอุปกรณ์ที่ใช้ในการจัดทำโครงงานเป็นอย่างดี

ขอบคุณเพื่อนร่วมกลุ่มสำหรับการทุ่มเท มุ่งมั่น ตั้งใจ รวมทั้งการช่วยเหลือกันและกันในการแก้ปัญหาต่างๆ เพื่อให้โครงงานวิศวกรรมนี้ผ่านไปด้วยดี

สุดท้ายนี้ คณะผู้จัดทำโครงงานขอขอบพระคุณครอบครัวที่ให้กำลังใจและสนับสนุนในการศึกษาตลอดมา โดยหวังเป็นอย่างยิ่งว่าโครงงานวิศวกรรมนี้ จะเป็นประโยชน์ต่อผู้ที่สนใจนำไปศึกษาต่อไป ความดีและประโยชน์ใดๆ จากการจัดทำโครงงานวิศวกรรมนี้ ขอมอบให้กับคณาจารย์และผู้มีพระคุณทุกท่านที่ได้กล่าวมาทั้งหมด ขอบคุณครับ

**คณะผู้จัดทำโครงงาน**

# สารบัญ


| | หน้า |
|---|---|
| บทคัดย่อภาษาไทย | ก |
| บทคัดย่อภาษาอังกฤษ | ข |
| กิตติกรรมประกาศ | ค |
| สารบัญ | ง |
| สารบัญตาราง | ฉ |
| สารบัญรูป | ช |
| บทที่ 1 บทนำ | 1 |
| 1.1 ที่มาและความสำคัญของโครงงาน | 1 |
| 1.2 วัตถุประสงค์ของโครงงาน | 1 |
| 1.3 ขอบเขตของโครงงาน | 1 |
| 1.4ประโยชน์ที่คาดว่าจะได้รับ | 2 |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้อง | 3 |
| 2.1 Internet of Thing (IoT) | 3 |
| 2.2 ระบบสวนอัจฉริยะ Smart Garden | 4 |
| 2.3 เครือข่ายและมาตรฐานต่างๆ | 5 |
| 2.3.1 เครือข่ายไร้สายตามมาตรฐาน802.11WLAN | 6 |
| 2.3.2 วิวัฒนาการของมาตรฐาน IEEE 802.11 | 7 |
| 2.4 NETPIE (Network Platform for Internet of Everything) | 9 |
| 2.5 NodeMCU v2 | 11 |
| 2.6 ESP8266 12e | 13 |
| 2.7 Soil Moisture Sensor อุปกรณ์เซ็นเซอร์วัดความชื้นในดิน | 14 |
| 2.8 เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ | 16 |
| 2.9 Relay Module 4 Channels | 18 |
| 2.10 SOLENOID VALVE | 20 |
| 2.11 งานวิจัยที่เกี่ยวข้อง | 22 |

# สารบัญ (ต่อ)


| | หน้า |
|---|---|
| บทที่ 3 วิธีการดำเนินงาน | 23 |
| 3.1 บล็อกไดอะแกรมแสดงภาพรวมการทำงานของระบบ | 23 |
| 3.2 ขั้นตอนและวิธีการดำเนินงาน | 23 |
| 3.3 แผนผังระบบโดยรวม | 24 |
| 3.4 การเชื่อมต่อ NodeMCU v2 เข้ากับ DHT22 | 25 |
| 3.5 การเชื่อมต่อ NodeMCU v2 เข้ากับ Soil Moisture Sensor | 27 |
| 3.6 การเชื่อมต่อ NodeMCU v2 เข้ากับ Relay Module 4 Channels และ SOLENOID VALE | 29 |
| บทที่ 4 ผลการดำเนินงาน | 31 |
| 4.1 ทดสอบวัดค่าอุณหภูมิและความชื้นผ่านเว็บบราวเซอร์ | 32 |
| 4.2 ทดสอบการควบคุมระบบจ่ายน้ำผ่านเว็บบราวเซอร์ | 43 |
| บทที่ 5 สรุปผล และข้อเสนอแนะ | 46 |
| 5.1 ปัญหาที่พบ | 46 |
| 5.2 แนวทางในการพัฒนา | 46 |
| เอกสารอ้างอิง | 47 |
| ภาคผนวก | 48 |

# สารบัญตาราง

**ตารางที่** **หน้า**


4.1 ผลการทดลองการวัดค่าความชื้นในดินประเภทดินแห้ง 37

4.2 ผลการทดลองการวัดค่าความชื้นในดินประเภทดินชื้น 38

4.3 ผลการทดลองวัดค่าความชื้นในดินประเภทดินเปียก 38

4.4 ผลการทดลองการวัดค่าอุณหภูมิภายใน 41

4.5 ผลการทดลองการวัดค่าอุณหภูมิภายนอก 41

4.6 ผลการทดลองการวัดค่าค่าความชื้นภายในห้อง 42

4.7 ผลการทดลองการวัดค่าอุณหภูมิภายในห้อง 42

# สารบัญรูป


| รูปที่ | | หน้า |
|---|---|---|
| 2.1 | Internet of Thing | 4 |
| 2.2 | Smart Garden | 5 |
| 2.3 | Network Platform for Internet of Everything | 9 |
| 2.4 | NodeMCU | 11 |
| 2.5 | Pin Map บนบอร์ด NodeMCU v2 | 12 |
| 2.6 | ESP8266 12e | 13 |
| 2.7 | Soil Moisture Sensor | 14 |
| 2.8 | Circuit Soil Moisture Sensor | 15 |
| 2.9 | อุปกรณ์เวนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ | 16 |
| 2.10 | คุณสมบัติเซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ | 17 |
| 2.11 | แสดงลำดับของข้อมูลบิตในการอ่านค่าจากไอซี | 17 |
| 2.12 | แสดงลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH | 18 |
| 2.13 | Relay Module 4 Channels | 18 |
| 2.14 | แสดงภาพและตารางแสดงขาที่ใช้ในการเชื่อมต่อของ Relay Module 4 Channels | 19 |
| 2.15 | สัญลักษณ์ในวงจรของรีเลย์ | 19 |
| 2.16 | SOLENOID VALVE | 20 |
| 2.17 | โครงสร้างภายในและหลักการทำงานของ SOLENOID VALVE | 21 |
| 3.1 | บล็อกไดอะแกรมแสดงภาพรวมการทำงานของระบบ | 23 |
| 3.2 | แผนผังการทำงานของระบบโดยรวม | 24 |
| 3.3 | แผนผังการทำงานของ Sensor DHT22 | 25 |
| 3.4 | แสดงการต่อวงจรของ NodeMCU กับ DHT22 | 26 |
| 3.5 | แผนผังการทำงานของ Soil Moisture Sensor | 27 |
| 3.6 | แสดงการต่อวงจร NodeMCU กับ Soil Moisture Sensor | 28 |
| 3.7 | แผนผังการทำงานของ Relay Module 4 Channels | 29 |
| 3.8 | แสดงการต่อวงจร NodeMCU กับ Relay Module 4 Channels และ Solenoid Valve | 30 |

# สารบัญรูป (ต่อ)

**รูปที่** **หน้า**

4.1 รูปวงจร Smart Garden (ภายนอก) 31
4.2 รูปวงจร Smart Garden (ภายใน) 32
4.3 เปิด Internet Web Browser เข้าไปที่เว็บไซต์ของ NETPIE 33
4.4 ทำการเข้าสู่ระบบด้วย ID และ Password ของ NETPIE 33
4.5 อ่านค่าที่ได้รับจากเซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ DHT22 แบบเรียลไทม์ 34
4.6 อ่านค่าที่ได้รับจาก Soil Moisture Sensor (เซนเซอร์วัดความชื้นในดิน) เมื่อดินแห้ง 34
4.7 อุปกรณ์แสดงค่าการวัดความชื้นเมื่ออยู่ในดินแห้ง 35
4.8 อ่านค่าที่ได้รับจาก Soil Moisture Sensor (เซนเซอร์วัดความชื้นในดิน) เมื่อดินชื้น 35
4.9 อุปกรณ์แสดงค่าการวัดความชื้นเมื่ออยู่ในดินชื้น 36
4.10 อ่านค่าที่ได้รับจาก Soil Moisture Sensor (เซนเซอร์วัดความชื้นในดิน) เมื่อดินเปียก 36
4.11 อุปกรณ์แสดงค่าการวัดความชื้นเมื่ออยู่ในดินเปียก 37
4.12 อ่านค่าที่ได้รับจาก DHT 22 (เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ) ในอุณหภูมิ ภายนอก 39
4.13 อ่านค่าที่ได้รับจากเครื่องมือวัดอุณหภูมิและความชื้นในอากาศจากภายนอก 39
4.14 อ่านค่าที่ได้รับจาก DHT 22 (เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ) ในอุณหภูมิ ภายในห้อง 40
4.15 อ่านค่าที่ได้รับจากเครื่องมือวัดอุณหภูมิและความชื้นในอากาศภายในห้อง 40
4.16 แสดงค่าสถานะขณะปิด SOLENOID VALVE 43
4.17 แสดงวงจรขณะควบคุม SOLENOID VALVE ขณะปิดวงจร 44
4.18 แสดงค่าสถานะขณะสั่งเปิด SOLENOID VALVE 44
4.19 แสดงวงจรขณะควบคุม SOLENOID VALVE ขณะเปิดวงจร 45

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและที่มาของปัญหาที่ทำโครงงาน

ระบบสื่อสารเครือข่ายแบบไร้สายเป็นระบบที่ได้รับความนิยมมากในปัจจุบัน และได้เข้ามาแทนที่ระบบสื่อสารแบบมีสาย ระบบสื่อสารเครือข่ายแบบไร้สายได้รับความนิยมมากและได้ถูกนำใช้งานอย่างกว้างขวาง ทั้งในหน่วยงาน องค์กรต่างๆ หรือนำไปใช้ในชีวิตประจำวัน เพื่อที่ช่วยอำนวยความสะดวกสบายต่างๆ จึงได้มีการพัฒนานวัตกรรมที่เรียกว่า ระบบการดูแลสวนอัจฉริยะ หรือที่เรียกว่า smart garden ที่ได้นำมาใช้ในการช่วยให้การดูแลสวนให้เป็นเรื่องง่ายซึ่งจะสามารถสร้างความสะดวกสบายในการดูแลสวนได้เป็นอย่างดี โดยมีการใช้โนดเอ็มซียู ที่เป็น open hardware และ open source ที่ทำให้ง่ายต่อการศึกษา การหาข้อมูล และการเรียนรู้ ดังนั้นคณะผู้จัดทำได้มีแนวคิดในการจัดทำระบบการดูแลสวนอัจฉริยะโดยใช้โนดเอ็มซียู ในการควบคุม โดยมีการสร้างเพจเพื่อใช้การควบคุมการสั่งการเปิด-ปิดหัวฉีดน้ำ และนอกจากนั้นยังแสดงค่าการตรวจวัดอุณหภูมิในอากาศ และความชื้นในดินอีกด้วย

## 1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาสภาพแวดล้อมที่เหมาะสมของไม้สวนไม้ประดับ

1.2.2 เพื่อนำบอร์ด ARDUINO มาใช้ในการติดต่อกับอุปกรณ์เซ็นเซอร์

1.2.3 เพื่อสามารถเขียนโปรแกรมควบคุมการทำงานของระบบ ผ่านเครือข่าย

1.2.4 เพื่อการประยุกต์ใช้งานอินเตอร์เน็ตทุกสิ่ง

## 1.3 ขอบเขตโครงงานวิศวกรรม

1.3.1 ตรวจความชื้นในดินและอุณหภูมิในอากาศ

1.3.2 ควบคุมการสั่งงานผ่านเครือข่ายไร้สาย

1.3.3 แสดงผลข้อมูลความชื้นและอุณหภูมิบนเว็บ

## 1.4 ประโยชน์ที่จะได้รับ

1.4.1 ได้รับความรู้ทางด้านโปรแกรม ARDUINO

1.4.2 สามารถประยุกต์ใช้ในการทำสวนโดยเครือข่ายไร้สาย

1.4.3 เกิดความคิดในการแก้ปัญหาเมื่อไม่สามารถลงมือทำได้

1.4.4 เป็นแนวทางสำหรับบุคคลที่สนใจสั่งงานผ่านเครือข่ายไร้สาย

# บทที่ 2
# ทฤษฎีที่เกี่ยวข้อง

## 2.1 Internet of Thing (IoT)

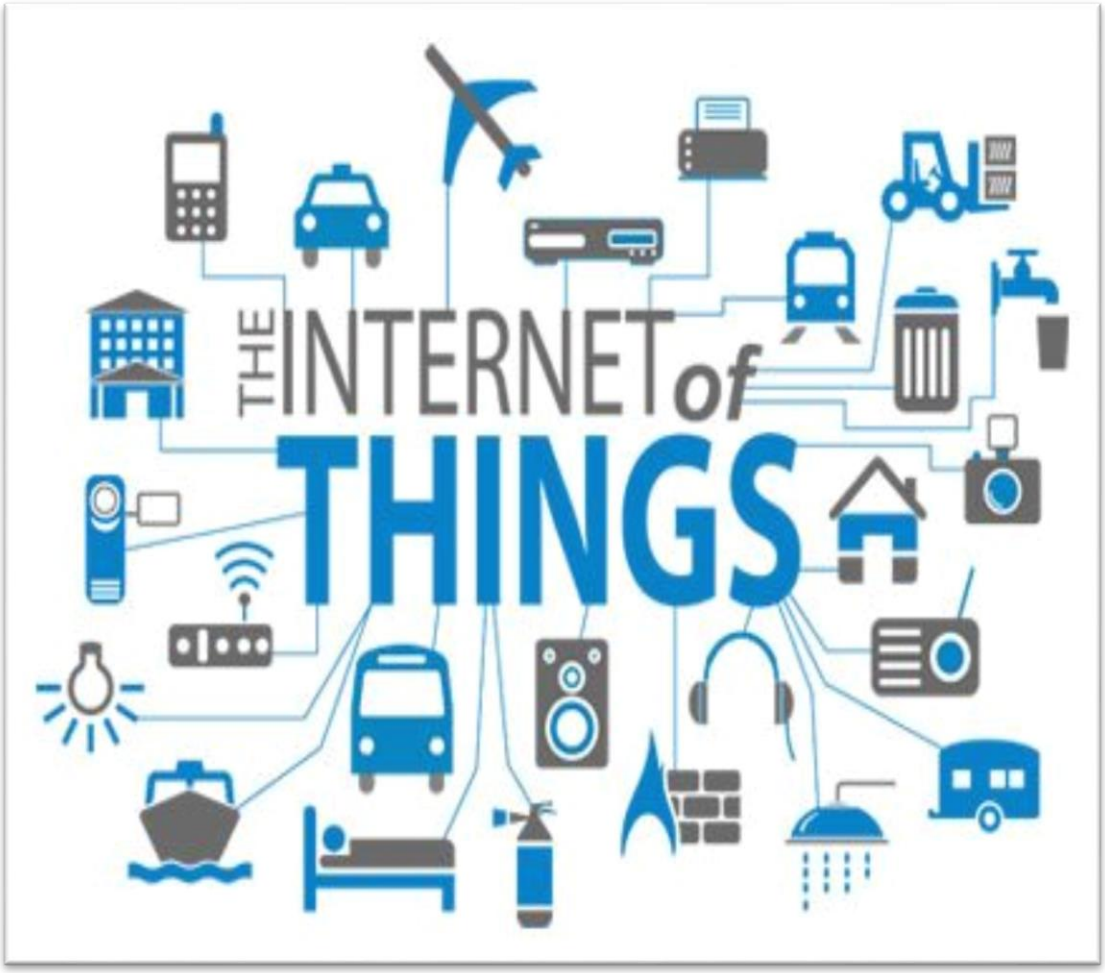
IoT คือ สภาพแวดล้อมอันประกอบด้วยสรรพสิ่งที่สามารถสื่อสารและเชื่อมต่อกันได้ผ่านโพรโทคอลสำหรับการสื่อสารทั้งแบบใช้สายและไร้สาย โดยสรรพสิ่งต่าง ๆ มีวิธีการระบุตัวตนได้ รับรู้บริบทของสภาพแวดล้อมได้ และมีปฏิสัมพันธ์โต้ตอบและทำงานร่วมกันได้ ความสามารถในการสื่อสารของสรรพสิ่งนี้จะนำไปสู่นวัตกรรมและบริการใหม่อีกมากมาย ตัวอย่างเช่น เซ็นเซอร์ภายในบ้านตรวจจับการเคลื่อนไหวของผู้อยู่อาศัย และส่งสัญญาณไปสั่งเปิด/ปิดสวิตช์ไฟตามห้องต่าง ๆ ที่มีคนหรือไม่มีคนอยู่ อุปกรณ์วัดสัญญาณชีพของผู้ป่วย/ผู้สูงอายุและส่งข้อมูลไปยังบุคลากรทางการแพทย์ หรือส่งข้อความเรียกหน่วยกู้ชีพหรือรถฉุกเฉิน เป็นต้น นอกจากนี้ IoT จะเปลี่ยนรูปแบบและกระบวนการผลิตในภาคอุตสาหกรรมไปสู่ยุคใหม่ หรือที่เรียกว่า Industry 4.0 ที่จะอาศัยการเชื่อมต่อสื่อสารและทำงานร่วมกันระหว่างเครื่องจักร มนุษย์ และข้อมูล เพื่อเพิ่มอำนาจในการตัดสินใจที่รวดเร็วและมีความถูกต้องแม่นยำสูง โดยที่ข้อมูลทั้งหลายที่เก็บจากเซ็นเซอร์ที่ใช้ตรวจวัดตัวอุปกรณ์และสภาพแวดล้อมจะถูกนำมาวิเคราะห์ ให้ได้ผลลัพธ์เพื่อนำไปปรับปรุงกระบวนการผลิตได้อย่างทันที นอกจากการข้ามขีดจำกัดเรื่องเวลาแล้ว ระบบควบคุมหรือระบบวิเคราะห์ข้อมูล อาจไม่ได้อยู่ในที่เดียวกันกับเครื่องจักร แต่สามารถควบคุมสั่งการได้โดยไร้ขีดจำกัดเรื่องสถานที่เทคโนโลยีที่ทำให้ IoT เกิดขึ้นได้จริงและสร้างผลกระทบในวงกว้างได้ แบ่งออกเป็นสามกลุ่มได้แก่

**2.1.1 เทคโนโลยีที่ช่วยให้สรรพสิ่งรับรู้ข้อมูลในบริบทที่เกี่ยวข้อง** เช่น เซ็นเซอร์

**2.1.2 เทคโนโลยีที่ช่วยให้สรรพสิ่งมีความสามารถในการสื่อสาร** เช่น ระบบสมองกลฝังตัว รวมถึงการสื่อสารแบบไร้สายที่ใช้พลังงานต่ำ อาทิ Zigbee, 6LowPAN, Low-power Bluetooth

**2.2.3 เทคโนโลยีที่ช่วยให้สรรพสิ่งประมวลผลข้อมูลในบริบทของตน** เช่น เทคโนโลยีการประมวลผลแบบคลาวด์ และเทคโนโลยีการวิเคราะห์ข้อมูลขนาดใหญ่ หรือ Big Data Analytics ในด้านสถานะการพัฒนา เทคโนโลยีในกลุ่มเซ็นเซอร์ในปัจจุบันมีความแม่นยำสูง และราคาถูกมาก ศูนย์เทคโนโลยีไมโครอิเล็กทรอนิกส์ (TMEC) มีความเชี่ยวชาญด้านการผลิตเซ็นเซอร์คุณภาสำหรับงานด้านการเกษตร และอุตสาหกรรม ส่วนเทคโนโลยีระบบสมองกลฝังตัวก็มีความสามารถสูงขึ้นในราคาที่ถูกลง แผงวงจรไมโครคอนโทรลเลอร์ขนาดเล็กที่มีความสามารถสูงเทียบเท่าคอมพิวเตอร์ ปัจจุบันมีราคาตั้งแต่สามร้อยบาท อีกทั้งมีฮาร์ดแวร์แบบโอเพ่นซอร์สมากขึ้น ทำให้ต้นทุนการผลิตอุปกรณ์ IoT ต่ำลงมาก นักพัฒนาชาวไทยสามารถนำฮาร์ดแวร์เปิดเหล่านี้ไปดัดแปลงและขายเป็นบอร์ดเฉพาะทาง หรือสามารถสร้างผลิตภัณฑ์ใหม่ของตนเองได้อย่างรวดเร็ว ส่วนเทคโนโลยีการประมวลผลแบบคลาวด์ และเทคโนโลยีการวิเคราะห์ข้อมูลขนาดใหญ่ ในต่างประเทศผ่านจุดของการวิจัยมาสู่บริการเชิงพาณิชย์แล้ว ในประเทศ

ไทย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) มีบริการคลาวด์แพลตฟอร์ม NETPIE สำหรับให้บริการเชื่อมต่อสื่อสารในรูปแบบ IoT ดังนั้นจึงเป็นโอกาสของผู้พัฒนาชาวไทยและประเทศไทยที่จะเข้ามามีบทบาท ไม่ใช่ในฐานะผู้ใช้เท่านั้น แต่ยังสามารถมีส่วนกำหนดทิศทาง สร้างนวัตกรรม บริการ ผลิตภัณฑ์หรือมาตรฐานใหม่ เพื่อก้าวขึ้นไปเป็นผู้นำด้าน IoT ของโลก



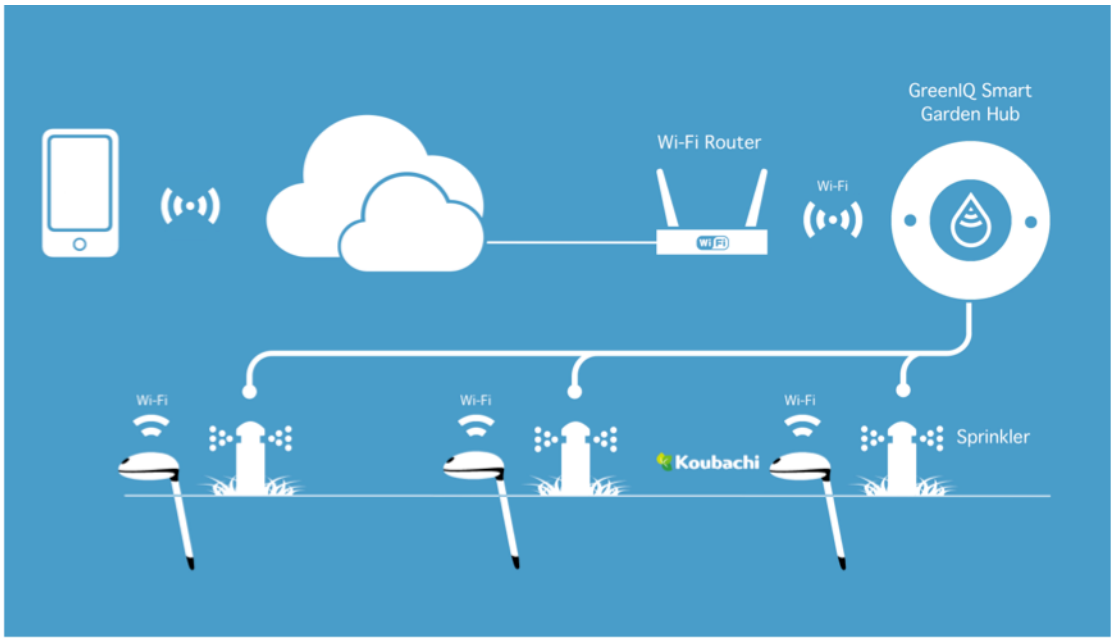

**รูปที่ 2.1** Internet of Thing

**ที่มา :** http://www.huffingtonpost.com/sam-cohen/internet-of-things-as-the_b_10937956.html

## 2.2 ระบบสวนอัจฉริยะ Smart Garden

ระบบสวนอัจฉริยะ (Smart Garden) เป็นการจัดการระบบสวนโดยนำวิทยาศาสตร์เทคโนโลยีสารสนเทศมาใช้เป็นเครื่องมือในการพัฒนาและดูแลจัดการระบบปฏิบัติการต่างๆ โดยสามารถประมวลผลได้อย่างรวดเร็วและถูกต้องแม่นยำ สามารถเก็บข้อมูลสภาพแวดล้อมด้วยอุปกรณ์ต่างๆ เช่น เครือข่ายเซ็นเซอร์ เครื่องสแกนสภาพดิน เป็นต้น แล้วปฏิบัติตามขั้นตอนที่วางไว้ เช่น การติดตั้งโปรแกรมการให้น้ำ

หรือให้ปุ๋ย ซึ่งสามารถทำงานตามเงื่อนไขที่กำหนดด้วยระบบการทำงานอัตโนมัติ แล้วยังสามารถควบคุมการทำงานของระบบนี้ได้ทุกที่ทุกเวลาด้วยเทคโนโลยีสารสนเทศ เพื่อให้เกิดความสะดวกสบายและง่ายต่อการจัดการ




**รูปที่ 2.2** Smart Garden

**ที่มา :** https://www.houselogix.com/shop/greeniq-smart-garden-hub

## 2.3 เครือข่ายและมาตรฐานต่างๆ

มาตรฐาน IEEE 802.11 ซึ่งได้รับการตีพิมพ์ครั้งแรกเมื่อปีพ.ศ. 2540 โดย IEEE (The Institute of Electronics and Electrical Engineers) และเป็นเทคโนโลยีสำหรับ WLAN ที่นิยมใช้กันอย่างแพร่หลาย มากที่สุด คือข้อกำหนด (Specfication) ส าหรับอุปกรณ์ WLAN ในส่วนของ Physical (PHY) Layer และ Media Access Control (MAC) Layer โดยในส่วนของ PHY Layer มาตรฐาน IEEE 802.11 ได้กำหนดให้ อุปกรณ์มีความสามารถในการรับส่งข้อมูลด้วยความเร็ว 1, 2, 5.5, 11 และ 54 Mbps โดยมีสื่อ 3 ประเภท ให้เลือกใช้ได้แก่ คลื่นวิทยุที่ความถี่สาธารณะ 2.4 และ 5 GHz, และ อินฟราเรด (Infarred) (1 และ 2 Mbps เท่านั้น) สำหรับในส่วนของ MAC Layer มาตรฐาน IEEE 802.11 ได้กำหนดให้มีกลไกการทำงานที่ เรียกว่า CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) ซึ่งมีความคล้ายคลึงกับ หลักการ CSMA/CD (Collision Detection) ของมาตรฐาน IEEE 802.3 Ethernet ซึ่งเป็นที่นิยมใช้กัน ทั่วไปในเครือข่าย LAN แบบใช้สายนำสัญญาณ นอกจากนี้ในมาตรฐาน IEEE802.11 ยังกำหนดให้มี ทางเลือกสำหรับความปลอดภัยให้กับเครือข่าย โดยกลไกการเข้ารหัสข้อมูล

**2.3.1 เครือข่ายไร้สายตามมาตรฐาน802.11WLAN**

คือ มาตรฐานการทำงานของระบบเครือข่ายไร้สาย ที่ถูกกำหนดขึ้นโดย Institute of Electrical and Electronics Engineers (IEEE) เป็นมาตรฐานกลาง ที่ได้นำมาปฏิบัติใช้ เพื่อที่จะทำการเชื่อมโยงอุปกรณ์ เครือข่ายไร้สายเข้าด้วยกันบนระบบในทางปกติแล้ว การเชื่อมต่อระบบเครือข่ายไร้สายจำเป็นต้อง ใช้อุปกรณ์สองชิ้น นั่นคือ แอคเซสพอยต์คือตัวกลางที่ช่วยในการติดต่อระหว่าง ตัวรับ-ส่งสัญญาณไวร์เลส ของผู้ใช้กับ สาย นำสัญญาณจากทองแดงที่ได้รับการเชื่อมต่อกับระบบเครือข่ายแล้ว เช่น สายแลน ตัวรับ-ส่งสัญญาณไวร์เลส ทำหน้าที่รับ-ส่ง สัญญาณ ระหว่างตัวรับส่งแต่ละตัวด้วยกัน หลังจากที่เทคโนโลยีเครือข่ายไร้สายนี้ได้เกิดขึ้น ก็ได้เกิดมาตรฐานตามมาอีกมายมาย โดยที่การจะเลือกซื้อหรือเลือกใช้อุปกรณ์เครือข่ายไร้สายเหล่านั้น เราจำเป็นจะต้องคำนึงถึงเทคโนโลยีที่ใช้ในผลิตภัณฑ์ นั้นๆ รวมถึงความเข้ากันได้ของเทคโนโลยีที่ต่างๆด้วย

- มาตรฐาน 802.11a

ทำงานบนย่านความถี่ 5 GHz โดยที่สามารถให้อัตราการส่งถ่ายข้อมูล 54 Mbps และเนื่องด้วยการที่มาตรฐานนี้ ใช้การเชื่อมต่อที่ความถี่สูงๆ ทำให้มาตรฐานนี้ มีระยะการรับส่งที่ ค่อนข้างใกล้ คือประมาณ 35 เมตร ในโครงสร้างปิด เช่น ในตึก ในอาคาร และ 120 เมตรในที่โล่งแจ้งและ ด้วยความที่ส่งข้อมูลด้วยความถี่สูงนี้ทำให้การส่งข้อมูลนั้นไม่สามารถทะลุทะลวงโครงสร้างของตึกได้มากนัก อุปกรณ์ไร้สายที่รองรับเทคโนโลยี IEEE 802.11a นี้ไม่สามารถเข้ากันได้กับอุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11b และ IEEE 802.11g ที่จะอธิบายด้านล่างนี้ได้

- มาตรฐาน802.11b (11 Mbps)

ทำงานบนย่านความถี่ 2.4 GHz โดยที่สามารถให้อัตราการส่งถ่ายข้อมูล 11 Mbps เนื่องจากการใช้คลื่นความถี่ที่ต่ำากว่าอุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11a ทำให้อุปกรณ์ที่ใช้ มาตรฐานนี้จะมีความสามารถในการส่งคลื่นสัญญาณไปได้ไกลกว่าคือประมาณ 38 เมตรในโครงสร้างปิด และ 140 เมตรในที่โล่งแจ้ง รวมถึงสัญญาณสามารถทะลุทะลวงโครงสร้างตึกได้มากกว่าอุปกรณ์ที่รองรับกับ มาตรฐาน IEEE 802.11a ด้วย

- มาตรฐาน802.11g (54 Mbps)

เป็นมาตรฐานเครือข่ายไวร์เลสแลนตัวใหญ่ล่าสุดที่เพิ่งเปิดตัวไปเมื่อปลายปี2003 มีข้อดีเหนือกว่าตัวอื่นๆ คือ มีความเร็วในการทำงานสูงถึง 54 เมกะบิต และมีระยะการทำงานไกลสุดเท่ากับมาตรฐาน 802.11be - มาตรฐาน 802.11n (300 – 600 Mbps)

เป็นไวร์เลสแลนที่มีความเร็วสูงสุดถึง 600 Mbps โดยส่งผ่าน 4 stream ซึ่งจะประกอบไปด้วยเสาอากาศ 4 ชุด ทั้งภาครับและส่ง แต่ละ Stream จะมีความเร็วที่ 150 Mbps สามารถใช้งานได้ทั้งย่านความถี่ 2.4 GHz และ 5 GHs หากใช้งานช่องสัญญาณขนาด 40 MHz จะส่งข้อมูลได้สูงกว่าการใช้ช่องสัญญาณขนาดมาตรฐานคือ 20 MHz ไวร์เลสแลน 802.11n นี้จะเป็นมาตรฐานที่อยู่ในโน้ตบุ๊คใหม่

ตั้งแต่ปี 2009 โดยจะสามารถรับส่องข้อมูล 2-3 Stream ทั้งนี้ขึ้นอยู่กับจำนวนภาครับส่งและเสา หากเป็นรุ่นที่มี 2 เสาจะมีความเร็วสูงสุดที่ 300 Mbps แต่ถ้าเป็นรุ่น 3 เสา จะมีความเร็วสูงสุดที่ 450 Mbps

**2.3.2 วิวัฒนาการของมาตรฐาน IEEE 802.11**

มาตรฐาน IEEE 802.11 ได้รับการตีพิมพ์ครั้งแรกในปี พ.ศ. 2540 ซึ่งอุปกรณ์ตามมาตรฐานดังกล่าว จะมีความสามารถในการรับส่งข้อมูลด้วยความเร็ว 1 และ 2 Mbps ด้วยสื่อ อินฟราเรด (Infarred) หรือ คลื่นวิทยุที่ความถี่ 2.4 GHz และมีกลไก WEP ซึ่งเป็นทางเลือกสำหรับสร้างความปลอดภัยให้กับเครือข่าย WLAN ได้ในระดับหนึ่ง เนื่องจากมาตรฐาน IEEE 802.11 เวอร์ชันแรกเริ่มมีประสิทธิภาพค่อนข้างต่างและไม่ มีการรองรับหลักการ Quality of Service (QoS) ซึ่งเป็นที่ต้องการของตลาด อีกทั้งกลไกรักษาความ ปลอดภัยที่ใช้ยังมีช่องโหว่อยู่มาก IEEE จึงได้จัดตั้งคณะทำงาน (Task Group) ขึ้นมาหลายชุดด้วยกันเพื่อทำ การปรับปรุงเพิ่มเติมมาตรฐานให้มีศักยภาพสูงขึ้น โดยคณะทำงานกลุ่มที่มีผลงานที่น่าสนใจและเป็นที่รู้จักกันดี ได้แก่ IEEE 802.11a, IEEE 802.11b, IEEE 802.11e, IEEE 802.11g, และ IEEE 802.11i

- IEEE 802.11b คณะทำงานชุด IEEE 802.11b ได้ตีพิมพ์มาตรฐานเพิ่มเติมนี้เมื่อปี พ.ศ. 2542 ซึ่งเป็นที่รู้จักกันดี และใช้งานกันอย่างแพร่หลายมากที่สุด มาตรฐาน IEEE 802.11b ใช้เทคโนโลยีที่เรียกว่า CCK (Complimentary Code Keying) ผนวกกับ DSSS (Direct Sequence Spread Spectrum) เพื่อปรับปรุงความสามารถของอุปกรณ์ให้รับส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 11 Mbps ผ่านคลื่นวิทยุความถี่ 2.4 GHz (เป็นย่านความถี่ที่เรียกว่า ISM (Industrial Scientific and Medical) ซึ่งถูกจัดสรรไว้อย่างสากลสำหรับการ ใช้งานอย่างสาธารณะด้านวิทยาศาสตร์ อุตสาหกรรม และการแพทย์ โดยอุปกรณ์ที่ใช้ความถี่ย่านนี้ก็เช่น IEEE 802.11, Bluetooth, โทรศัพท์ไร้สาย, และเตาไมโครเวฟ) ส่วนใหญ่แล้วอุปกรณ์ IEEE 802.11 WLAN ที่ใช้กันอยู่ในปัจจุบันจะเป็นอุปกรณ์ตามมาตรฐาน IEEE 802.11b นี้และใช้เครื่องหมายการค้าที่รู้จักกันดีใน นาม Wi-Fi ซึ่งเครื่องหมายการค้าดังกล่าวถูกกำหนดขึ้นโดยสมาคม WECA (Wireless Ethernet Compatability Alliance) โดยอุปกรณ์ที่ได้รับเครื่องหมายการค้าดังกล่าวได้ผ่านการตรวจสอบแล้วว่า เป็นไปตามมาตรฐาน IEEE 802.11b และสามารถนำไปใช้งานร่วมกับอุปกรณ์ยี่ห้ออื่นๆที่ได้รับเครื่องหมาย Wi-Fi ได้

- IEEE 802.11a คณะทำงานชุด IEEE 802.11a ได้ตีพิมพ์มาตรฐานเพิ่มเติมนี้เมื่อปี พ.ศ. 2542 มาตรฐาน IEEE 802.11a ใช้เทคโนโลยีที่เรียกว่า OFDM (Orthogonal Frequency Division Multiplexing) เพื่อปรับปรุง ความสามารถของอุปกรณ์ให้รับส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 54 Mbps แต่จะใช้คลื่นวิทยุที่ความถี่ 5 GHz ซึ่งเป็นย่านความถี่สาธารณะสำหรับใช้งานในประเทศสหรัฐอเมริกาที่มีสัญญาณรบกวนจากอุปกรณ์อื่น น้อยกว่าในย่านความถี่ 2.4 GHz อย่างไรก็ตามข้อเสียหนึ่งของมาตรฐาน IEEE 802.11a ที่ใช้คลื่นวิทยุที่ ความถี่ 5 GHz ก็คือในบางประเทศย่านความถี่ดังกล่าวไม่สามารถนำมาใช้

งานได้อย่างสาธารณะ ตัวอย่างเช่น ประเทศไทยไม่อนุญาตให้มีการใช้งานอุปกรณ์ IEEE 802.11a เนื่องจากความถี่ย่าน 5 GHz ได้ ถูกจัดสรรสำหรับกิจการอื่นอยู่ก่อนแล้ว นอกจากนี้ข้อเสียอีกอย่างหนึ่งของอุปกรณ์ IEEE 802.11a WLAN ก็ คือรัศมีของสัญญาณมีขนาดค่อนข้างสั้น (ประมาณ 30 เมตร ซึ่งสั้นกว่ารัศมีสัญญาณของอุปกรณ์ IEEE 802.11b WLAN ที่มีขนาดประมาณ 100 เมตร สำหรับการใช้งานภายในอาคาร) อีกทั้งอุปกรณ์ IEEE 802.11a WLAN ยังมีราคาสูงกว่า IEEE 802.11b WLAN ด้วย ดังนั้นอุปกรณ์ IEEE 802.11a WLAN จึง ได้รับความนิยมน้อยกว่า IEEE 802.11b WLAN มาก

- IEEE 802.11g คณะทำงานชุด IEEE 802.11g ได้ใช้นำเทคโนโลยี OFDM มาประยุกต์ใช้ในช่องสัญญาณวิทยุความถี่ 2.4 GHz ซึ่งอุปกรณ์ IEEE 802.11g WLAN มีความสามารถในการรับส่งข้อมูลด้วยความเร็วสูงสุดที่ 54 Mbps ส่วนรัศมีสัญญาณของอุปกรณ์ IEEE 802.11g WLAN จะอยู่ระหว่างรัศมีสัญญาณของอุปกรณ์ IEEE 802.11a และ IEEE 802.11b เนื่องจากความถี่ 2.4 GHz เป็นย่านความถี่สาธารณะสากล อีกทั้งอุปกรณ์ IEEE 802.11g WLAN สามารถทำงานร่วมกับอุปกรณ์ IEEE 802.11b WLAN ได้ (backward-compatible) ดังนั้นจึงมีแนวโน้มสูงว่าอุปกรณ์ IEEE 802.11g WLAN จะได้รับความนิยมอย่างแพร่หลายหากมีราคาไม่แพง จนเกินไปและน่าจะมาแทนที่ IEEE 802.11b ในที่สุด ตามแผนการแล้วมาตรฐาน IEEE 802.11g จะได้รับ การตีพิมพ์ประมาณช่วงกลางปี พ.ศ. 2546

- IEEE 802.11e คณะทำงานชุดนี้ได้รับมอบหมายให้ปรับปรุง MAC Layer ของ IEEE 802.11 เพื่อให้สามารถรองรับ การใช้งานหลักการ Qualitiy of Service ส าหรับ application เกี่ยวกับมัลติมีเดีย (Multimedia) เนื่องจาก IEEE 802.11e เป็นการปรับปรุง MAC Layer ดังนั้นมาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุกเวอร์ชั่นได้ แต่อย่างไรก็ตามการทำงานของคณะทำงานชุดนี้ยังไม่แล้วเสร็จในขณะนี้ (พฤษภาคม พ.ศ. 2546)

- IEEE 802.11i คณะทำงานชุดนี้ได้รับมอบหมายให้ปรับปรุง MAC Layer ของ IEEE 802.11 ในด้านความปลอดภัย เนื่องจากเครือข่าย IEEE 802.11 WLAN มีช่องโหว่อยู่มากโดยเฉพาะอย่างยิ่งการเข้ารหัสข้อมูล (Encryption) ด้วย key ที่ไม่มีการเปลี่ยนแปลง คณะทำงานชุด IEEE 802.11i จะนำเอาเทคนิคขั้นสูงมาใช้ ในการเข้ารหัสข้อมูลด้วย key ที่มีการเปลี่ยนค่าอยู่เสมอและการตรวจสอบผู้ใช้ที่มีความปลอดภัยสูง มาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุกเวอร์ชันได้ แต่อย่างไรก็ตามการ ทำงานของคณะทำงานชุดนี้ยังไม่แล้วเสร็จในขณะนี้ (พฤษภาคม พ.ศ. 2546)

## 2.4 NETPIE (Network Platform for Internet of Everything)

**รูปที่ 2.3** Network Platform for Internet of Everything

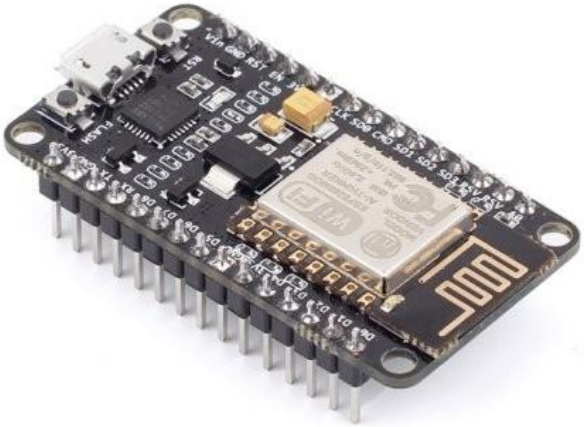
**ที่มา :** https://www.nectec.or.th/innovation/innovation-software/netpie.html

NETPIE แพลตฟอร์ม IoT เพื่อนักพัฒนาและอุตสาหกรรมไทย” ตั้งเป้าเป็นแพลตฟอร์มทางเลือกแรกของนักพัฒนาไทยที่เชื่อมอุปกรณ์และเครื่องมือต่างๆ หรือ The Internet of Things (IoT) ระยะแรกเน้นการสนับสนุนนักพัฒนาและอุตสาหกรรมขนาดย่อม SMEs) เพื่อสร้างขีดความสามารถและความเข้มแข็งให้กับอุตสาหกรรมไทยขนาดใหญ่ของไทย

ตามที่รัฐบาลได้ประกาศวิสัยทัศน์ประเทศไทย พ.ศ. 2558-2563 มั่นคง มั่งคั่ง ยั่งยืน โดยได้กำหนดเป้าหมายการเพิ่มขีดความสามารถในการแข่งขันของประเทศ เพื่อหลุดพ้นจากประเทศรายได้ปานกลาง ซึ่งเนคเทคมองว่าเศรษฐกิจนวัตกรรม (innovation economy) จะเป็นกลไกสำคัญของประเทศที่เป็นพื้นฐานให้กลไกอื่นในทุกภาคส่วนสามารถบรรลุเป้าหมายดังกล่าวได้ ประเทศไทยต้องพึ่งพาเทคโนโลยีและนวัตกรรมจากภายในประเทศให้มากขึ้น และพึ่งพาเทคโนโลยีและนวัตกรรมจากต่างประเทศอย่างชาญฉลาด โดยการสร้างกลไกการสนับสนุนและจูงใจให้ภาคเอกชนไทยเป็นผู้นำในการพัฒนางานวิจัย การนำวิทยาศาสตร์ เทคโนโลยีและนวัตกรรมมาเพิ่มขีดความสามารถในการสร้างคุณค่าให้กับผลิตภัณฑ์ ด้วยการนำเทคโนโลยีด้านอิเล็กทรอนิกส์ผสานพลังเข้ากับเทคโนโลยีคอมพิวเตอร์ เป็นทางเชื่อมต่อให้กับระบบเศรษฐกิจและสังคมในยุคหน้า ผ่านนวัตกรรมดิจิทัล (digital innovation) สำหรับรองรับอุตสาหกรรมเชิงอนาคตต่อไป ซึ่งการเปิดตัว NETPIE แพลตฟอร์ม IoT เพื่อนักพัฒนาและอุตสาหกรรมไทยในวันที่ 16 กันยายน 2558 นี้ เนคเทคมีกลุ่มเป้าหมายในระยะแรกคือนักพัฒนา เมคเกอร์ หรือผู้ประกอบการ SME ให้เข้ามาร่วมใช้ NETPIE แพลตฟอร์ม IoT ในการพัฒนานวัตกรรมเพื่อนำไปสู่การเกิดธุรกิจ บริการและผลิตภัณฑ์ใหม่ๆ ส่วนเป้าหมายในระยะยาวคือนักพัฒนาและผู้ประกอบการกลุ่มนี้จะเป็นแรงขับเคลื่อนให้

เกิดการนำ IoT ไปประยุกต์ใช้และสร้างมูลค่าเพิ่มในภาคอุตสาหกรรมมากขึ้น ทั้งนี้เนคเทคตั้งเป้าที่จะส่งเสริมให้เกิดอุปกรณ์ IoT ที่พัฒนาโดยคนไทยอย่างน้อย 1 ล้านอุปกรณ์ภายใน 3 ปี เพื่อวางรากฐานที่แข็งแกร่งให้กับภาคธุรกิจและอุตสาหกรรมที่เกี่ยวข้องต่อไปในอนาคต เพราะ IoT คือปัจจุบันและอนาคตของโลก และทุกคนซึ่งหมายรวมถึงบริษัทเทคโนโลยียักษ์ใหญ่ทั้งหลายกำลังแข่งขันกันเพื่อช่วงชิงความได้เปรียบในเชิงธุรกิจในด้าน IoT NETPIE ซึ่งเป็นผลงานวิจัยและพัฒนาของเนคเทค จะเป็นเครื่องมือหนึ่งในการส่งเสริมให้ธุรกิจ/อุตสาหกรรมที่เกี่ยวข้องของไทย เจริญเติบโตรองรับ Innovation Economy สามารถแข่งขันได้ในตลาดโลกดร.พนิตา พงษ์ไพบูลย์ นักวิจัยจากห้องปฏิบัติการวิจัยเทคโนโลยีเครือข่ายเนคเทค หัวหน้าทีมพัฒนา NETPIE ได้ NETPIE (Network Platform for Internet of Everything) cloud platform ที่ถูกออกแบบและพัฒนาขึ้นเพื่ออำนวยให้เกิดการสื่อสารระหว่างอุปกรณ์หรือ things ในเครือข่าย IoT โดยมีประโยชน์ต่อนักพัฒนาและอุตสาหกรรมไทย อาทิ NETPIE ช่วยให้อุปกรณ์สามารถคุยกันได้โดยผู้พัฒนาไม่ต้องกังวลว่า อุปกรณ์นั้นจะอยู่ที่ใด ทั้งในแง่ physical และ logical เพียงนำ NETPIE library ไปติดตั้งในอุปกรณ์ NETPIE จะรับหน้าที่ดูแลการเชื่อมต่อให้ทั้งหมด ไม่ว่าอุปกรณ์นั้นจะอยู่ในเครือข่ายชนิดใด ลักษณะใด หรือแม้กระทั่งเคลื่อนย้ายไปอยู่ที่ใด ผู้พัฒนาสามารถตัดปัญหากวนใจในการที่จะต้องมาออกแบบการเข้าถึงอุปกรณ์จากระยะไกล (remote access) ด้วยวิธีแบบเดิมๆ เช่น การใช้ fixed public IP หรือการตั้ง port forwarding ในเราท์เตอร์ หรือการต้องไปลงทะเบียนกับผู้ให้บริการ dynamic DNS ซึ่งทั้งหมดล้วนมีความยุ่งยากและลดความยืดหยุ่นของระบบ ไม่เพียงเท่านั้น NETPIE ยังช่วยให้การเริ่มต้นใช้งานเป็นไปโดยง่ายโดยการออกแบบให้อุปกรณ์ถูกค้นพบและเข้า (automatic discovery, plug and play) NETPIE ถูกออกแบบให้มี authorization/access control ในระดับ fine grain กล่าวคือผู้พัฒนาสามารถออกแบบได้เองทั้งหมด เช่น สิ่งใดมีสิทธิคุยกับสิ่งใด สิ่งใดมีสิทธิหรือไม่-เพียงใดในการอ่านหรือเขียนข้อมูล และสิทธิเหล่านี้จะมีอายุเท่าใดหรือถูกเพิกถอนภายใต้เงื่อนไขใด เป็นต้น NETPIE มีสถาปัตยกรรมเป็น cloud อย่างแท้จริงในทุกๆ ระดับของระบบ ทำให้เกิดความยืดหยุ่น และคล่องตัวสูงในการขยายตัว นอกจากนี้ โมดูลต่างๆ ยังถูกออกแบบให้ทำงานแยกจากกันเพื่อให้เกิดสภาวะ loose coupling และสื่อสารกันด้วยวิธีการ asynchronous messaging ช่วยให้แพลตฟอร์มมี reliability สูง สามารถนำไปใช้ซ้ำ และพัฒนาต่อเติมได้ง่าย ดังนั้นผู้พัฒนาไม่จำเป็นต้องกังวลกับการขยายตัวเพื่อรับโหลดที่เพิ่มขึ้นในระบบอีกต่อไป นอกจากนี้ทางเนคเทคจะเปิด NETPIE library ในรูปแบบ open-source ให้นักพัฒนาสามารถนำไปปรับปรุงต่อให้ตรงกับความต้องการใช้งานโดยเปิดโอกาสให้นำไปใช้ในเชิงพาณิชย์ได้โดยเนคเทคหวังที่จะให้เกิด community ที่จะมาร่วมกันพัฒนาต่อยอดสร้างความเข้มแข็งให้กับวงการ IoT ของไทย

## 2.5 NodeMCU v2



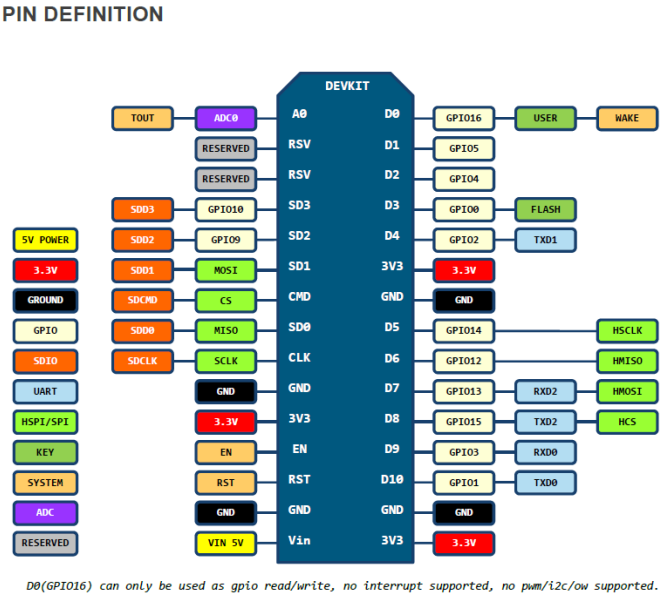
**รูปที่ 2.4** NodeMCU v2

**ที่มา :** http://www.elec-za.com/wp-content/uploads/2015/07/113990105-1.jpg

NodeMCU Development Kit V2 เป็นตัวที่พัฒนาจาก NodeMCU Version เดิม โดยเป็นโมดูลที่ประกอบด้วย ESP8266-12 E มีเสาอากาศแบบ PCB Antenna เชื่อมต่อเฮดเดอร์สำหรับ

ขาสัญญาณต่างๆ ได้แก่ GPIO, PWM, I2C, 1-wire, ADC และ มี SPI เพิ่มขึ้นมาจาก Version เดิม มีส่วนของ USB-to-TTL และพอร์ต micro USB ซึ่งใช้ชิพ USB to Serial ของ silicon lab cp2102 เชื่อมต่อ

เข้ากับเครื่องคอมพิวเตอร์เพื่อพัฒนาโปรแกรม สามารถติดตั้งเฟิร์มแวร์ NodeMCU ได้ และยังมีขนาดของ PCB ที่เล็กลง สามารถใช้งานกับ breadboard ได้ผู้ใช้สามารถเลือกพัฒนาด้วยสคริปต์ Lau โดยใช้เฟิร์มแวร์ NodeMCU หรือใช้เป็นชุดพัฒนาด้วยโมดูล ESP8266 ก็ได้ซึ่งสามารถเขียนด้วย Arduino IDE ได้ โมดูลมี GPIO ให้ใช้ถึง 10 พอร์ต สามารถนำมาพัฒนาโปรเจ็คทางด้าน Internet of Things (IoTs) เชื่อมต่อกับอุปกรณ์อื่นๆ ตามต้องการ




**รูปที่ 2.5** (Pin Map) บนบอร์ด NodeMCU v2

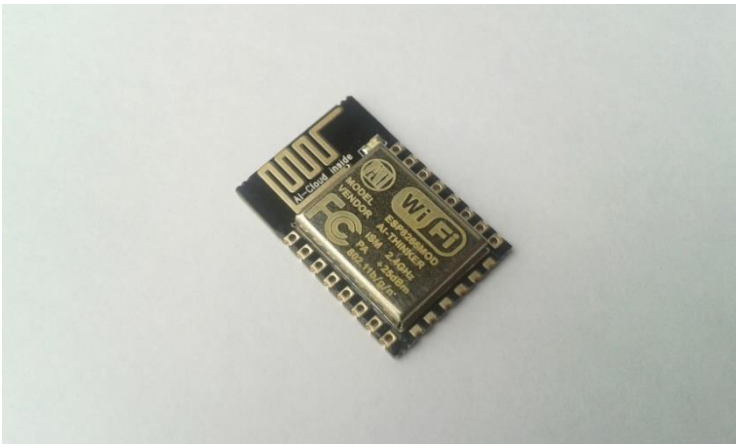
**ที่มา :** http://thaiopensource.org/มาเล่น-nodemcu-devkit-v2-กัน/

ข้อมูลสำคัญเชิงเทคนิคของบอร์ด NodeMCU v2

- ใช้โมดูล ESP-12E (ESP8266 SoC chip) ของบริษัท AI Thinker (ในขณะที่ NodeMCU v1 ใช้โมดูล ESP12) มีขาเพิ่มมาอีก 6 ขา เมื่อเปรียบเทียบกับ ESP-12
- ใช้ชิป Flash ความจุ 32Mbits (4MBytes)
- มีขนาดแคบกว่า NodeMCU v1 ดังนั้นเมื่อเสียบขาลงบนเบรดบอร์ด จะมีช่องเหลือด้านข้าง ทำให้สะดวกในการต่อวงจรบนเบรดบอร์ด
- มีวงจรควบคุมแรงดัน 3.3V (@800mA max.) บนบอร์ด ใช้ไอซีที่จ่ายกระแสได้มากกว่าบอร์ด NodeMCU v1
- ใช้ชิป CP2102 ของ Silabs ทำหน้าที่เป็นส่วนเชื่อมต่อ USB-to-Serial (แต่ NodeMCU v1 ใช้ชิป CH340G)
- มีขาสำหรับ SPI สำหรับต่อกับการ์ด SD (เพิ่มจากเดิมที่มีขาสำหรับ HSPI)
- มีขา GPIO3/RXD0 และ GPIO1/TXD0 ที่ต่อกับขา TXD และ RXD ของชิป CP2102 ตามลำดับ
- มีขา GPIO13/RXD2 และ GPIO15/TXD2 (ใช้เป็นพอร์ต Serial เพิ่มอีกหนึ่งชุด)
- ใช้คอนเนกเตอร์แบบ micro-USB สำหรับจ่ายแรงดันไฟเลี้ยง (VUSB) เท่ากับ +5V
- สามารถจ่ายแรงดันไฟเลี้ยง +5V จากภายนอกได้ (ต่อเข้าที่ขา VDD5V)
- มีปุ่มกด RST (รีเซตการทำงาน) และ Flash (สำหรับโปรแกรมเฟิร์มแวร์ใหม่)

- มีขา A0 รับอินพุตแรงดันแบบแอนะล็อกสำหรับวงจร ADC (ขนาด 10 บิต) ที่อยู่ภายในชิป ผ่านวงจรแบ่งแรงดันด้วยตัวต้านทาน 100k / 220k (ลดแรงดันอินพุตจาก 0..3.3V ลงมา ให้อยู่ในช่วง 0V-1V)

## 2.6 ESP8266 12e



**รูปที่ 2.6** ESP8266 12e
**ที่มา :** https://diygeeks.org/shop/wireless/esp8266-12e/

Wi-Fi คือ Wireless LAN หรือระบบ Network แบบไร้สาย ด้วยเทคโนโลยีการสื่อสารภายใต้มาตรฐาน IEEE 802.11 ซึ่งอุปกรณ์ทุกตัวที่ต่างยี่ห้อกันนั้นจะสามารถติดต่อสื่อสารกันได้โดยไม่ประสบปัญหา หากอุปกรณ์นั้นผ่านตามเกณฑ์มาตรฐานก็จะมีการประทับตรา Wi-Fi Certified ซึ่งหมายความว่าอุปกรณ์ตัวนี้สามารถเชื่อมต่อแบบไร้สายกับ อุปกรณ์อื่นที่มีตรา Wi-Fi Certified ได้ แล้วจึงกลายมาเป็นคำศัพท์ของอุปกรณ์ LAN ไร้สาย ระบบเครือข่ายไร้สาย (WLAN = Wireless Local Area Network) คือระบบการสื่อสารข้อมูลที่มีความคล่องตัวมาก ซึ่งอาจจะนำมาใช้ทดแทนหรือเพิ่มต่อกับระบบเครือข่ายแลนใช้สายแบบดั้งเดิม โดยใช้การส่งคลื่นความถี่วิทยุในย่านวิทยุ RF และ คลื่นอินฟราเรด ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่อง ผ่านอากาศ ทะลุกำแพง เพดานหรือสิ่งก่อสร้างอื่นๆ โดยปราศจากความต้องการของการเดินสาย นอกจากนั้นระบบเครือข่ายไร้สายก็ยังมีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบ LAN แบบใช้สาย ที่สำคัญก็คือ การที่มันไม่ต้องใช้สายทำให้การเคลื่อนย้ายการใช้งานทำได้โดยสะดวก ไม่เหมือนระบบ LAN แบบใช้สาย ที่ต้องใช้เวลาและการลงทุนในการปรับเปลี่ยนตำแหน่งการใช้งานเครื่องคอมพิวเตอร์ สำหรับเลข 802.11 นั้นเป็น เทคโนโลยีมาตรฐานแบบเปิดซึ่งกำหนดโดย Institute of Electrical and Electronics Engineers : IEEE โดยเลขหลักตัวหน้าจะเหมือน ๆ กัน แต่ความแตกต่างของเทคโนโลยีจะกำหนดด้วยตัวอักษรด้านหลัง เช่น 802.11b , 802.11a , 802.11g มาตรฐาน 802.11b ถือเป็นมาตรฐาน Wi-Fi ตัวแรก ที่ได้รับการพัฒนาขึ้นมา สามารถส่งข้อมูล

ได้ด้วยความเร็ว 11 เมกะบิตต่อวินาทีโดยใช้ช่วงความถี่ 2.4 กิกะเฮิรตซ์ ครอบคลุมพื้นที่ทำการในระยะ 150 เมตร

## 2.7 Soil Moisture Sensor อุปกรณ์เซ็นเซอร์วัดความชื้นในดิน

**รูปที่ 2.7** Soil Moisture Sensor
**ที่มา :** https://hobbytronics.com.pk/product/soil-moisture-sensor/

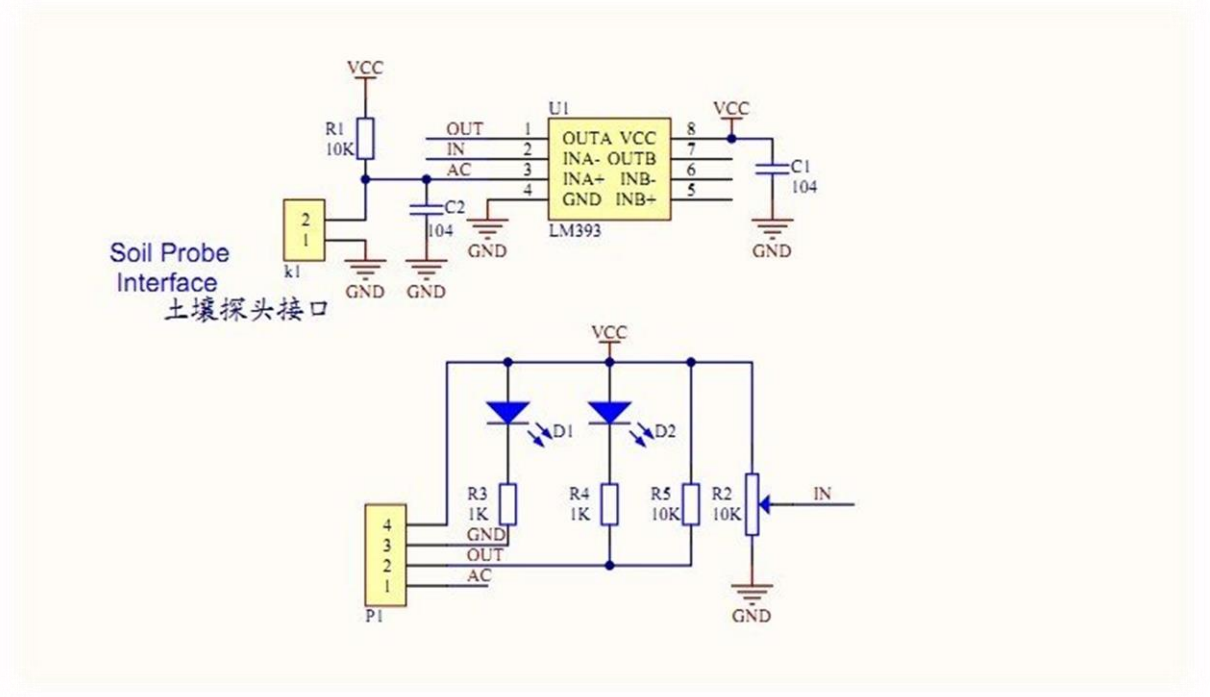
ใช้วัดความชื้นในดิน หรือใช้เป็นเซ็นเซอร์น้ำ สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์โดยใช้อนาล็อกอินพุตอ่านค่าความชื้น หรือเลือกใช้สัญญาณดิจิตอลที่ส่งมาจากโมดูล

การนำไปใช้งานหากนำไปใช้งานด้านการวัดความชื้นแบบละเอียด แนะนำให้ใช้งานขา A0 ต่อเข้ากับไมโครคอนโทรลเลอร์เพื่อวัดค่าแรงดันที่ได้ ซึ่งจะได้ออกมาใช้เปรียบเทียบค่าความชื้นได้ หากมีความชื้นน้อย แรงดันจะใกล้ 5V มาก หากความชื้นมาก แรงดันก็จะลดต่ำลงหากต้องการนำไปใช้ในโปรเจคที่ไม่ต้องใช้วัดละเอียด เช่น โปรเจครดน้ำต้นไม้ ใช้ควบคุมปั้มน้ำให้รดน้ำต้นไม้อัตโนมัติ สามารถนำขา D0 ต่อเข้ากับทรานซิสเตอร์กำลังเพื่อสั่งให้ปั้มน้ำ หรือโซลินอยให้ทำงานเพื่อให้มีน้ำไหลมารดต้นไม้ได้เลย เมื่อความชื้นในดินมีมากพอ จะปล่อยลอจิก 0 แล้วทรานซิสเตอร์จะหยุดนำกระแส ทำให้ปั้มน้ำหยุดปล่อยน้ำ

**Soil Moisture Sensor** สามารถอ่านค่าได้ 2 แบบ

- อ่านค่าแบบ Analog คือ อ่านค่าความชื้นและให้ค่าตั้งแต่ 0 – 1024
- อ่านค่าแบบ Digital คือ มีการเปรียบเทียบค่าที่ตั้งไว้และให้ค่าเป็น HIGH หรือ LOW
- ถ้าค่าที่อ่านได้มากกว่า 1000 แสดงว่าเซนเซอร์อยู่ในอากาศ (“Sensor in the Air”)
- ถ้าค่าที่อ่านได้มากกว่า 800 และน้อยกว่า 1000 แสดงว่าเซนเซอร์อยู่ในดินที่แห้ง

- ถ้าค่าที่อ่านได้มากกว่า 300 และน้อยกว่า 800 แสดงว่าเซนเซอร์อยู่ในดินที่ชื้น
- ถ้าค่าที่อ่านได้น้อยกว่า 300 แสดงว่าเซนเซอร์อยู่ในดินที่เปียก




**รูปที่ 2.8** Circuit Soil Moisture Sensor

**ที่มา :** https://www.ioxhop.com/product/87/เซ็นเซอร์วัดความชื้นในดิน-soil-moisture-sensor

Description:

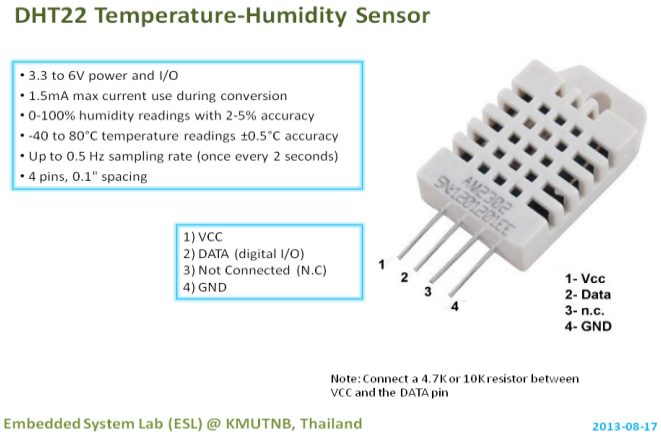
- This is a simple water sensor can be used to detect soil moisture when the soil moisture deficit module outputs a high level and vice versa output low. Use this sensor producedan automatic plant water device so that the plants in your garden without people to manage
- Sensitivity adjustable the blue digital potentiometer adjustment
- Operating voltage 3.3V-5V
- Module dual output mode digital output analog output more accurate
- With fixed bolt hole for easy installation
- PCB size: 3cm * 1.6cm
- Power indicator  and digital switching output indicator

Interface Description

- VCC: 3V-5V

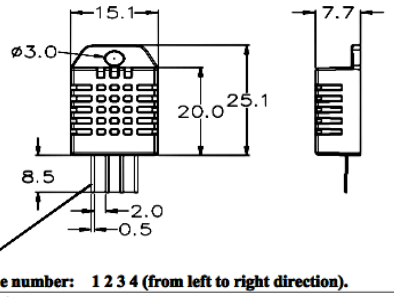
- GND: GND ของ Arduino
- D0: ต่อเข้ากับขา Digital ของ Arduino
- A0: ต่อเข้ากับขา Analog ของ Arduino

## 2.8 DHT22 เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์




**รูปที่ 2.9** อุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์
**ที่มา :** www.evselectro.com/digital-humidity-and-temperature-sensor-dht22-am2302-6475

อุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ (Temperature & Relative Humidity Sensor) เป็นอุปกรณ์ที่สามารถนำมาประยุกต์ใช้งานทางด้านระบบสมองกลฝังตัวได้หลากหลาย เช่น การวัดและควบคุมอุณหภูมิและความชื้น ระบบบันทึกข้อมูลเกี่ยวกับอุณหภูมิและความชื้นในห้อง เป็นต้น อุปกรณ์ประเภทนี้แตกต่างกันตามผู้ผลิต ราคา ความแม่นยำ ความละเอียดในการวัด การให้ค่าแบบดิจิทัลหรือแบบ แอนะล็อก เป็นต้น บทความนี้กล่าวถึง การทดลองใช้งานโมดูล DHT22/AM2302 ซึ่งมีราคาถูก ให้ค่าเป็นแบบดิจิทัล ใช้ขาสัญญาณดิจิทัลเพียงเส้นเดียวในการเชื่อมต่อแบบบิตอนุกรมสองทิศทาง (serial data, bi-directional) โดยนำมาเชื่อมต่อกับ Arduino เพื่ออ่านค่าจากเซนเซอร์




Pin sequence number: 1 2 3 4 (from left to right direction).

| Pin | Function |
|---|---|
| 1 | VDD----power supply |
| 2 | DATA--signal |
| 3 | NULL |
| 4 | GND |

**รูปที่ 2.10** คุณสมบัติเซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์

**ที่มา :** https://www.arduinoall.com/product/825/dht22-am2302-module

Description:

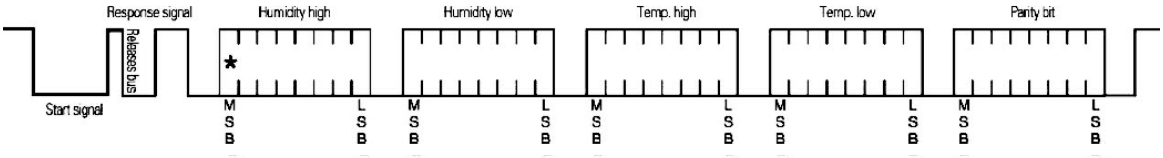
- ใช้แรงดันไฟเลี้ยงได้ในช่วง: 3.3V ถึง 5.5V DC (ดังนั้นจึงใช้ได้กับ 3.3V และ 5V)
- วัดอุณหภูมิได้ในช่วง: -40 to 80 °C (±0.5 °C accuracy)
- วัดความชื้นสัมพัทธ์ได้ในช่วง: 0 - 100 RH% (2 - 5% accuracy)
- อัตราการวัดสูงสุด: 0.5Hz
- คอนเนกเตอร์แบบ 4 ขา ( 0.1" / 2.54mm spacing)

  Pin 1 = VCC

  Pin 2 = SDA (Serial data, bidirectional)

  Pin 3 = N.C. (Not Connect)

ในการอ่านข้อมูลจากไอซีนั้น จะใช้ขาสัญญาณเพียงเส้นเดียวคือ DATA (หรือ SDA) แบบสองทิศทาง และในสถานะปรกติสัญญาณ DATA จะเป็น HIGH ในการอ่านข้อมูลแต่ละครั้ง

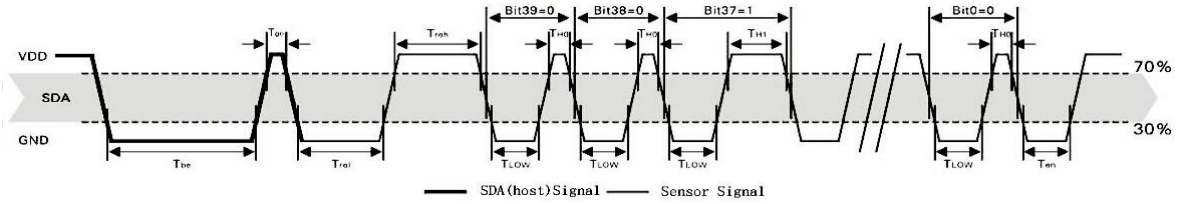



**รูปที่ 2.11** แสดงลำดับของข้อมูลบิตในการอ่านค่าจากไอซี

**ที่มา :** http://cpre.kmutnb.ac.th/esl/learning/index.php?article=dht22_am2302

ไมโครคอนโทรลเลอร์จะต้องกำหนดให้ขา DATA เป็นเอาต์พุต และสร้างบิต START ซึ่งจะต้องเป็น LOW อย่างน้อย 800 µsec จากนั้นจึงให้เป็น HIGH อย่างน้อย 20 µsec หลังจากนั้นเป็นการรอการตอบกลับ (response) และจากไอซี ขา DATA จะถูกต้องเปลี่ยนเป็นอินพุต เริ่มต้นของการตอบกลับไอซี

จะดึงสัญญาณลงเป็น LOW และปล่อยให้เป็น HIGH ช่วงละ 80 µsec โดยประมาณ (เรียกว่า Response Bit) จากนั้นจึงจะเป็นการส่งข้อมูลทีละบิต รวม 40 บิต (ช่วง LOW ตามด้วยช่วง HIGH) ช่วง LOW ของแต่ละบิต จะกว้างเท่ากัน แต่จะต่างกันในช่วง HIGH สำหรับบิตที่มีค่าเป็น 0 หรือ 1 (ใช้ความกว้างช่วง HIGH ในการจำแนกค่าของบิต)




**รูปที่ 2.12** แสดงลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH

**ที่มา :** http://cpre.kmutnb.ac.th/esl/learning/index.php?article=dht22_am2302

## 2.9 Relay Module 4 Channels

**รูปที่ 2.13** Relay Module 4 Channels

**ที่มา :** http://www.nexuscyber.com/4-channel-12v-relay-module

มีเอาต์พุตคอนเน็คเตอร์ที่ Relay เป็น NO/COM/NC สามารถใช้กับโหลดได้ทั้งแรงดันไฟฟ้า DC และ AC โดยใช้สัญญาณในการควบคุมการทำงานด้วยสัญญาณโลจิก TTL

คุณสมบัติ (Features)

รีเลย์เอาต์พุตแบบ SPDT จำจวน 4 ช่อง

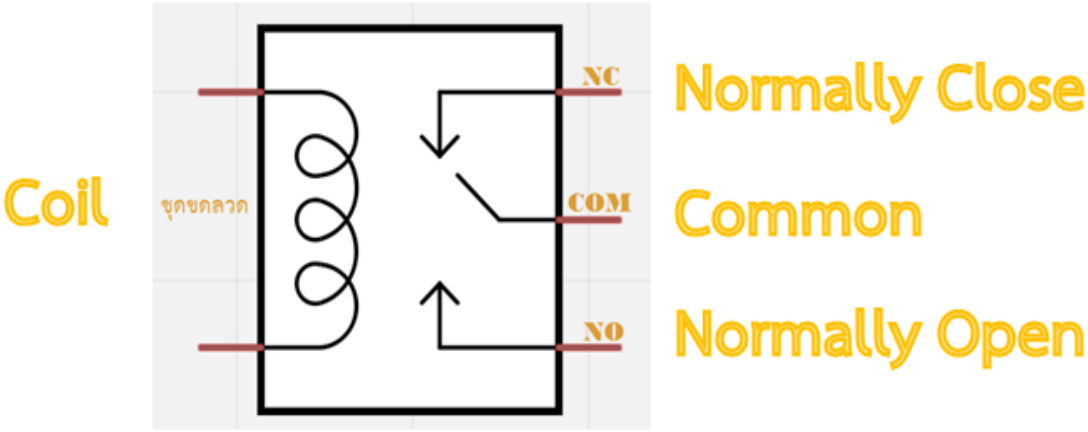
สั่งงานด้วยระดับแรงดัน TTL

- CONTACT OUTPUT ของรีเลย์รับแรงดันได้สูงสุด 250 VAC 10 A, 30 VDC 10 A
- มี LED แสดงสถานะ การทำงานของรีเลย์และแสดงสถานะของบอร์ด
- มีจัมพ์เปอร์สำหรับเลือกว่าจะใช้กราวด์ร่วมหรือแยก
- มี OPTO-ISOLATED เพื่อแยกกราวด์ส่วนของสัญญาณควบคุมกับไฟฟ้าที่ขับรีเลย์ออกจากกัน

| ขาที่ | คำอธิบาย |
|---|---|
| 1 | +VCC ขาไฟ 5VDC |
| 2 | GND |
| 3 | ขาสัญญาณอินพุต Relay 1 ( IN1 ) |
| 4 | ขาสัญญาณอินพุต Relay 2 ( IN2 ) |
| 5 | ขาสัญญาณอินพุต Relay 3 ( IN3 ) |
| 6 | ขาสัญญาณอินพุต Relay 4 ( IN4 ) |
| 7 | COM (คอมมอนของ OPTO) |
| 8 | GND (กราวด์ของบอร์ดเป็นกราวด์เดียวกันกับขาที่ 2) |
| 9 | NC (Normal Close) ซึ่งหมายถึงหน้าสัมผัสแบบปกติปิด |
| 10 | COM (Common) ที่จะตัดหรือต่อวงจรจากขา NC, NO |
| 11 | NO (Normal Open) ซึ่งหมายถึงหน้าสัมผัสแบบปกติเปิด |

**รูปที่ 2.14** แสดงภาพและตารางแสดงขาที่ใช้ในการเชื่อมต่อของ Relay Module 4 Channels
**ที่มา :** http://thaieasyelec.com/article-wiki/review-product-article/ตัวอย่างการใช้งาน

สัญลักษณ์ในวงจรไฟฟ้าของรีเลย์




**รูปที่ 2.15** สัญลักษณ์ในวงจราของรีเลย์
**ที่มา :** http://thaieasyelec.com/article-wiki/review-product-article/ตัวอย่างการใช้งาน-arduino-relay-module-ควบคุมการปิดเปิดเครื่องใช้ไฟฟ้า.html

ภายใน Relay จะประกอบไปด้วยขดลวดและหน้าสัมผัส

หน้าสัมผัส NC (Normally Close) เป็นหน้าสัมผัสปกติปิด โดยในสภาวะปกติหน้าสัมผัสนี้จะต่อเข้ากับขา COM (Common) และจะลอยหรือไม่สัมผัสกันเมื่อมีกระแสไฟฟ้าไหลผ่านขดลวด

หน้าสัมผัส NO (Normally Open) เป็นหน้าสัมผัสปกติเปิด โดยในสภาวะปกติ จะลอยอยู่ ไม่ถูกต่อกับขา COM (Common) แต่จะเชื่อมต่อกันเมื่อมีกระแสไฟไหลผ่านขดลวด

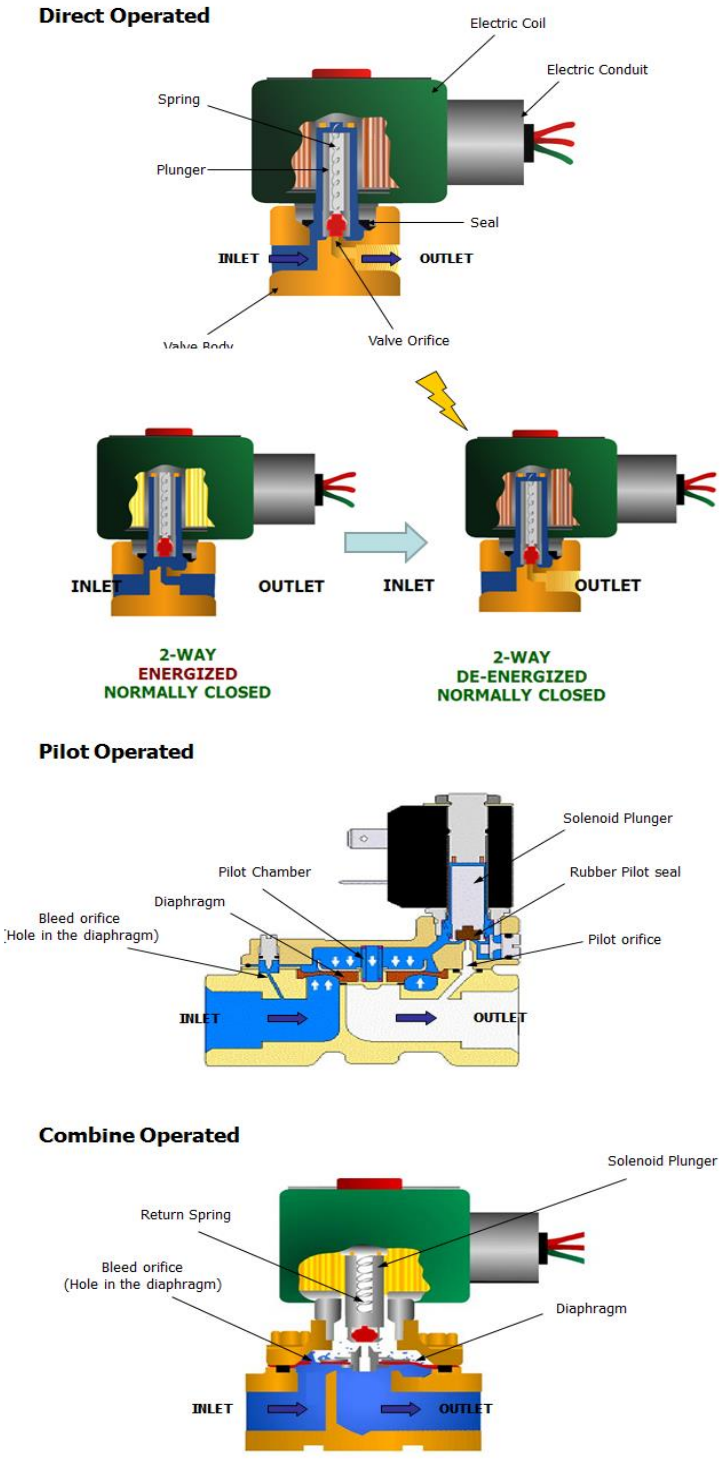
ขา COM (Common) เป็นขาที่ถูกใช้งานร่วมกันระหว่าง NC และ NO ขึ้นอยู่กับว่า ขณะนั้นมกระแสไฟฟ้าไหลผ่านขดลวดหรือไม่ หน้าสัมผัสใน Relay 1 ตัวอาจมีมากกว่า 1 ชุด ขึ้นอยู่กับผู้ผลิตและ

สวิตช์จะถูกแยกประเภทตามจำนวน Pole และจำนวน Throw ซึ่งจำนวน Pole (SP-Single Pole, DP-Double Pole, 3P-Triple Pole, etc.) จะบอกถึงจำนวนวงจรที่ทำการเปิด-ปิด หรือ จำนวนของขา COM นั่นเอง และจำนวน Throw (ST, DT) จะบอกถึงจำนวนของตัวเลือกของ Pole ตัวอย่างเช่น SPST- Single Pole Single Throw สวิตช์จะสามารถเลือกได้เพียงอย่างเดียวโดยจะเป็นปกติเปิด (NO-Normally Open) หรือปกติปิด (NC-Normally Close) แต่ถ้าเป็น SPDT- Single Pole Double Throw สวิตช์จะมีหนึ่งคู่เป็นปกติเปิด (NO) และอีกหนึ่งคู่เป็นปกติปิดเสมอ (NC)

## 2.10 SOLENOID VALVE

**รูปที่ 2.16** SOLENOID VALVE
**ที่มา :** http://www.asco.com/en-us/Pages/solenoid-valves.aspx

โซลินอยด์ (Solenoid) เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่ง ที่มีหลักการทำงานคล้ายกับรีเลย์ (Relay) ภายในโครงสร้างของโซลินอยด์จะประกอบด้วยขดลวดที่พันอยู่รอบแท่งเหล็กที่ภายในประกอบด้วยแม่เหล็กชุดบนกับชุดล่าง เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็ก ทำให้แท่งเหล็กชุดล่างมีอำนาจแม่เหล็กดึงแท่งเหล็กชุดบนลงมาสัมผัสกันทำให้ครบวงจรทำงาน เมื่อวงจรถูกตัดกระแสไฟฟ้าทำให้แท่งเหล็กส่วนล่างหมดอำนาจแม่เหล็ก สปริงก็จะดันแท่งเหล็กส่วนบนกลับสู่ตำแหน่งปกติ จากหลักการดังกล่าวของโซลินอยด์ก็จะนำมาใช้ในการเลื่อนลิ้นวาล์วของระบบนิวแมติกส์การปิด-เปิดการจ่ายน้ำหรือของเหลวอื่นๆ โครงสร้างของ Solenoid โดยทั่วไปแบ่งออกเป็น 2 ชนิดคือ เลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยสปริง (Single Solenoid Valve) และเลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยโซลินอยด์วาล์ว (Double Solenoid Valve) ในที่นี้ใช้แบบ เลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับ




**รูปที่ 2.17** โครงสร้างภายในและหลักการทำงานของ SOLENOID VALE

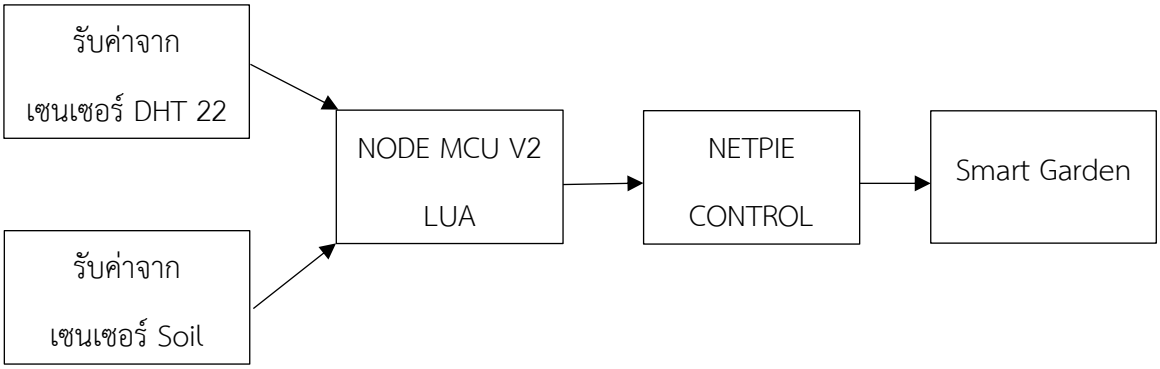
**ที่มา :** https://www.factomart.com/th/factomartblog/principle-of-solenoid-valve/

## 2.11 งานวิจัยที่เกี่ยวข้อง

2.11.1 กมลชนก ลอมโฮม, วรรณิศา ผางต๊ะ. 2555.การควบคุมหุ่นยนต์ผ่านเครือข่ายไร้สายโดยใช้อาร์ดูไอโน. ปริญญานิพนธ์, ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒองครักษ์ โครงงานนี้นำเสนอการควบคุมหุ่นยนต์ผ่านเครือข่ายไร้สายโดยใช้อาร์ดูโน โดยงานวิจัยนี้มีจุดประสงค์เพื่อทำการศึกษาและใช้งานโปรแกรม ARDUINO ในการสร้างหุ่นยนต์ โดยควบคุมผ่านเครือข่ายไร้สาย ซึ่งได้ทำการประดิษฐ์หุ่นยนต์ที่สามารถควบคุมผ่านเครือข่ายไร้สายโดยมีการสร้างเว็บเพจที่ใช้สำหรับควบคุมการเคลื่อนที่ของหุ่นยนต์พร้อมทั้งแสดงค่าระยะห่างของวัตถุด้วยอัลตร้าโซนิค โดยผลลัพธ์ที่ได้จากการที่ผู้ทำโครงงานได้ทดสอบการควบคุมหุ่นยนต์ผ่านเครือข่ายไร้สายโดยใช้อาร์ดูโน หุ่นยนต์สามารถเคลื่อนที่ได้ตามคำสั่งที่ควบคุมผ่านทางหน้าเว็บเพจและระบุระยะห่างของวัตถุจากบริเวณด้านหน้าและด้านหลังของหุ่นยนต์ได้ แต่การวัดระยะของ อัลตร้าโซนิคมีความคลาดเคลื่อนอยู่บ้างไม่เกิน 2 % จากที่ได้ศึกษาโครงงานนี้ทำให้ได้ความรู้ในการใช้งาน ARDUINO และหลักการในคำสั่งการควบคุมผ่านเว็บบราวเซอร์

# บทที่ 3
# หลักการทำงานและดำเนินงาน

## 3.1 บล็อกไดอะแกรมแสดงภาพรวมการทำงานของระบบ




**รูปที่ 3.1** บล็อกไดอะแกรมแสดงภาพรวมการทำงานของระบบ

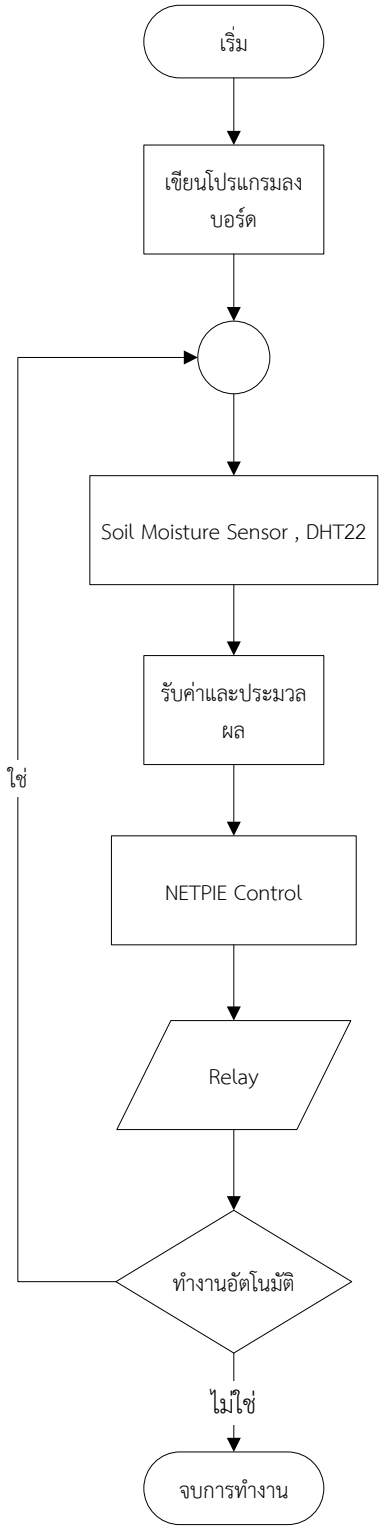
## 3.2 ขั้นตอนและวิธีการดำเนินงาน

ขั้นตอนที่1 ศึกษาข้อมูลต่างๆศึกษาข้อมูลเกี่ยวกับการเขียนภาษา C++ ในโปรแกรม NODEMCU V2 โครงสร้างและการทำงานของบอร์ดทดลอง การทำงานของเซ็นเซอร์ DHT22 และ Soil Moisture และระบบเครือข่ายไร้สาย เพื่อนำมาเขียนภาษาการทำงานของวงจร Smart Garden ในแต่ละส่วน

ขั้นตอนที่2 วางแผนการออกแบบระบบ Smart Garden เพื่อกำหนดลักษณะการวางอุปกรณ์ภายในของวงจร เพื่อให้วงจรและอุปกรณ์เข้าด้วยกัน รวมทั้งออกแบบโครงสร้างภายในและภายนอกของวงจร เพื่อให้อุปกรณ์ในภายมีการจัดวางแบบมีระเบียบและมีรูปลักษณะที่สวยงาม

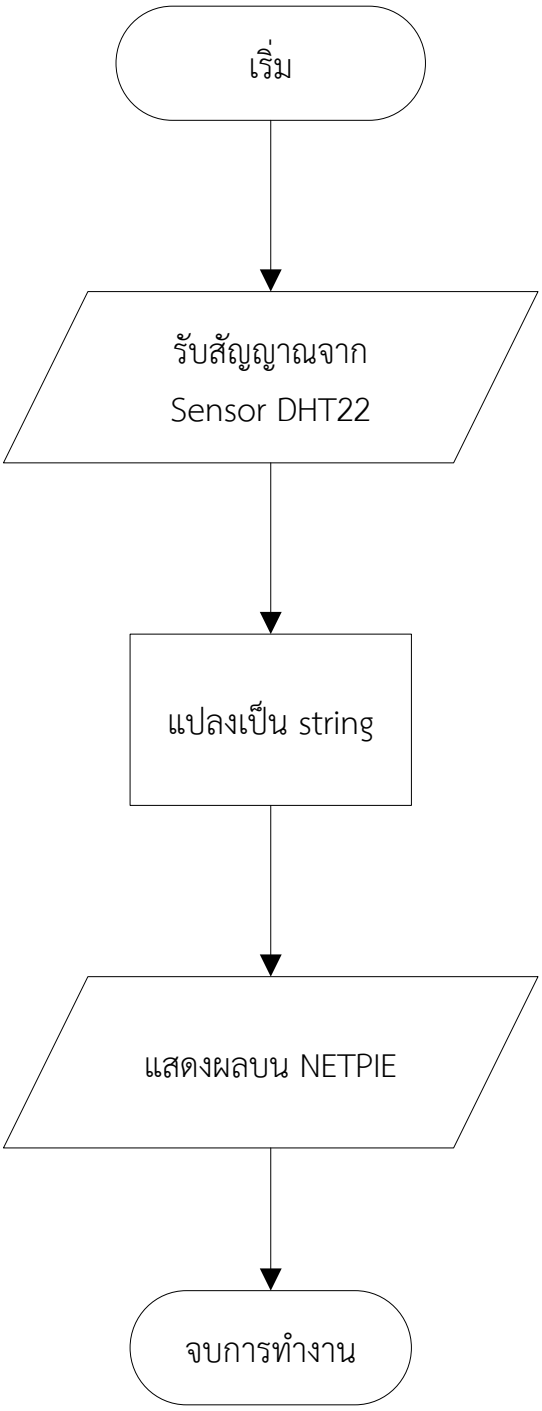
ขั้นตอนที่3 การทดสอบการทำงานของวงจรทำการทดสอบ ระบบสั่งงาน วัดความชื้นในดินและอุณหภูมิในอากาศและทำการเชื่อมต่อกับเครือข่ายไร้สาย โดยมีการควบคุมผ่าน NETPIE เพื่อให้วงจร Smart Garden แสดงผลการทำงานของระบบได้

## 3.3 แผนผังระบบโดยรวม



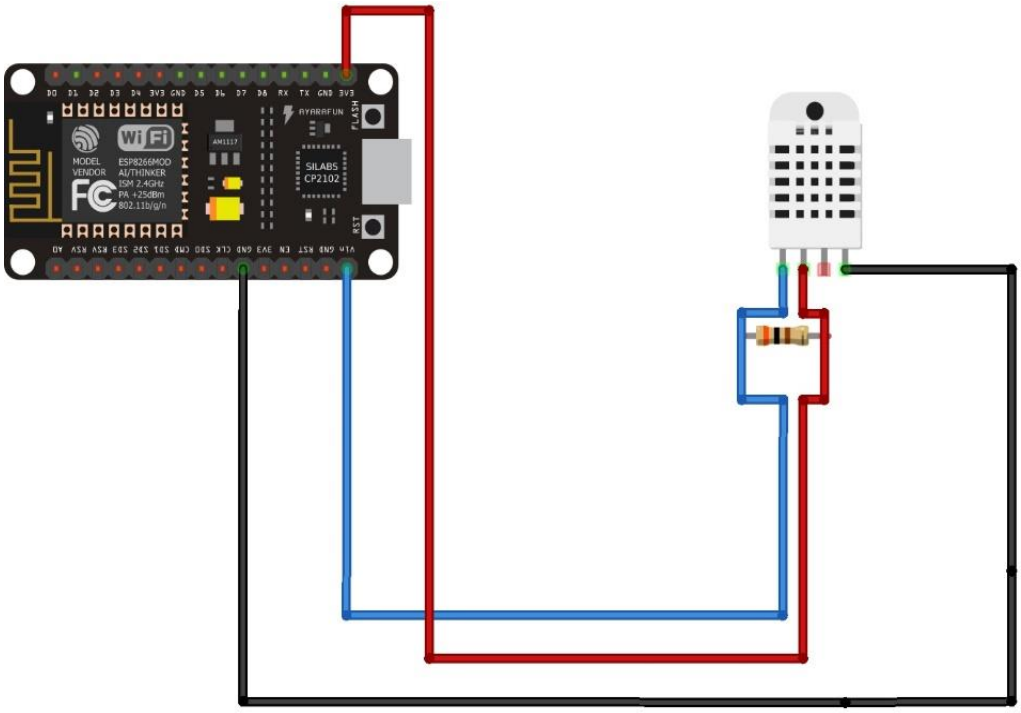

**รูปที่ 3.2** แผนผังการทำงานของระบบโดยรวม

**3.4 การเชื่อมต่อ NodeMCU v2 เข้ากับ DHT22 เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์**



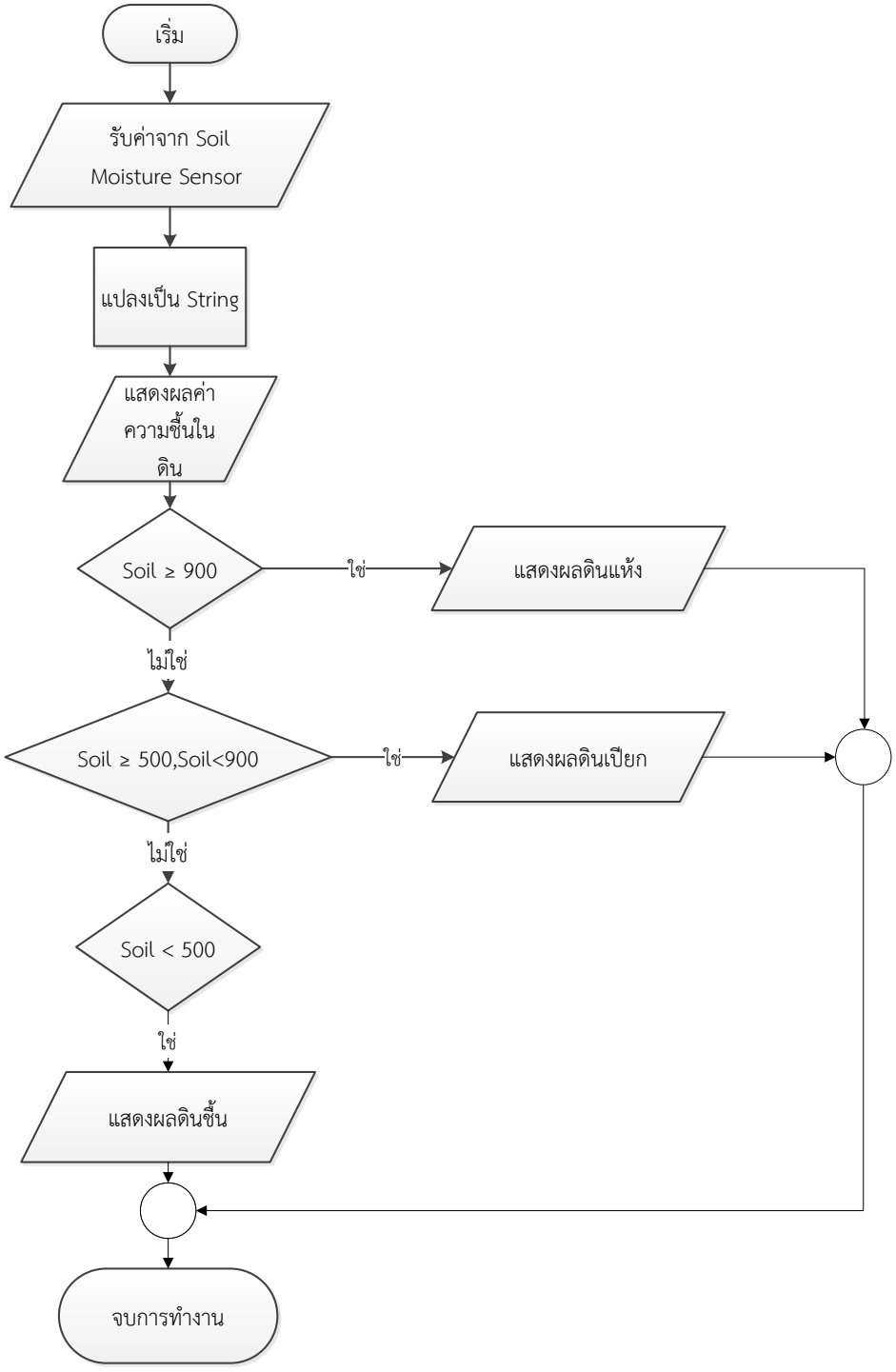

**รูปที่ 3.3** แผนผังการทำงานของ Sensor DHT22

- ขา $V_{cc}$ ใช้ไฟขนาด 3V ถึง 5.5V ต่อเข้าบอร์ด NodeMCU v2 ที่ขา $V_{in}$ 5V
- ขา Data ต่อเข้าบอร์ด NodeMCU v2 ในส่วนของแถวช่องสัญญาณดิจิตอลที่ขา GPIO4
- ขา Ground ต่อเข้ากับบอร์ด Node MCU v2 ในส่วนของแถวสัญญาณแอนะล็อกที่ขา GND



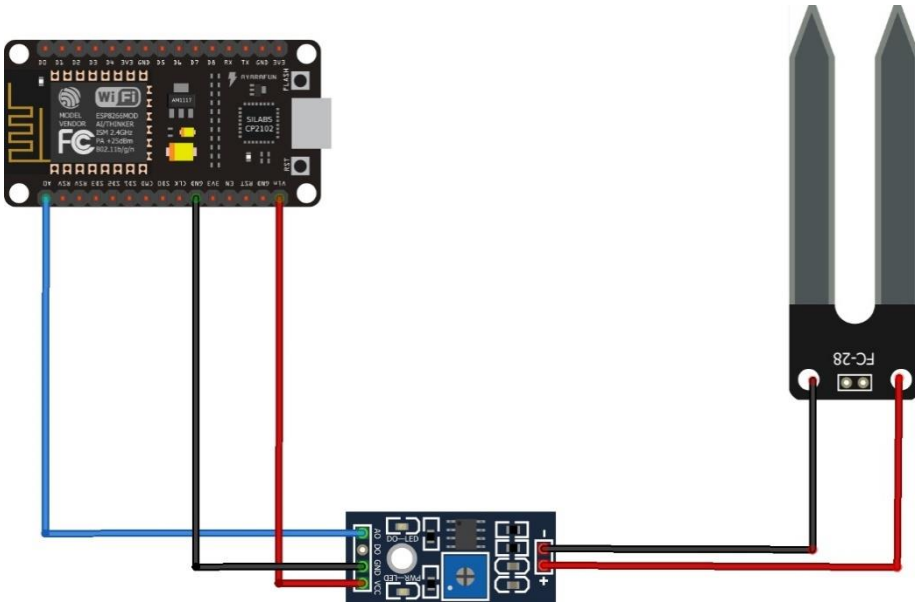

**รูปที่ 3.4** แสดงการต่อวงจรของ NodeMCU กับ DHT22

## 3.5 การเชื่อมต่อ NodeMCU v2 เข้ากับ Soil Moisture Sensor อุปกรณ์เซ็นเซอร์วัดความชื้นในดิน



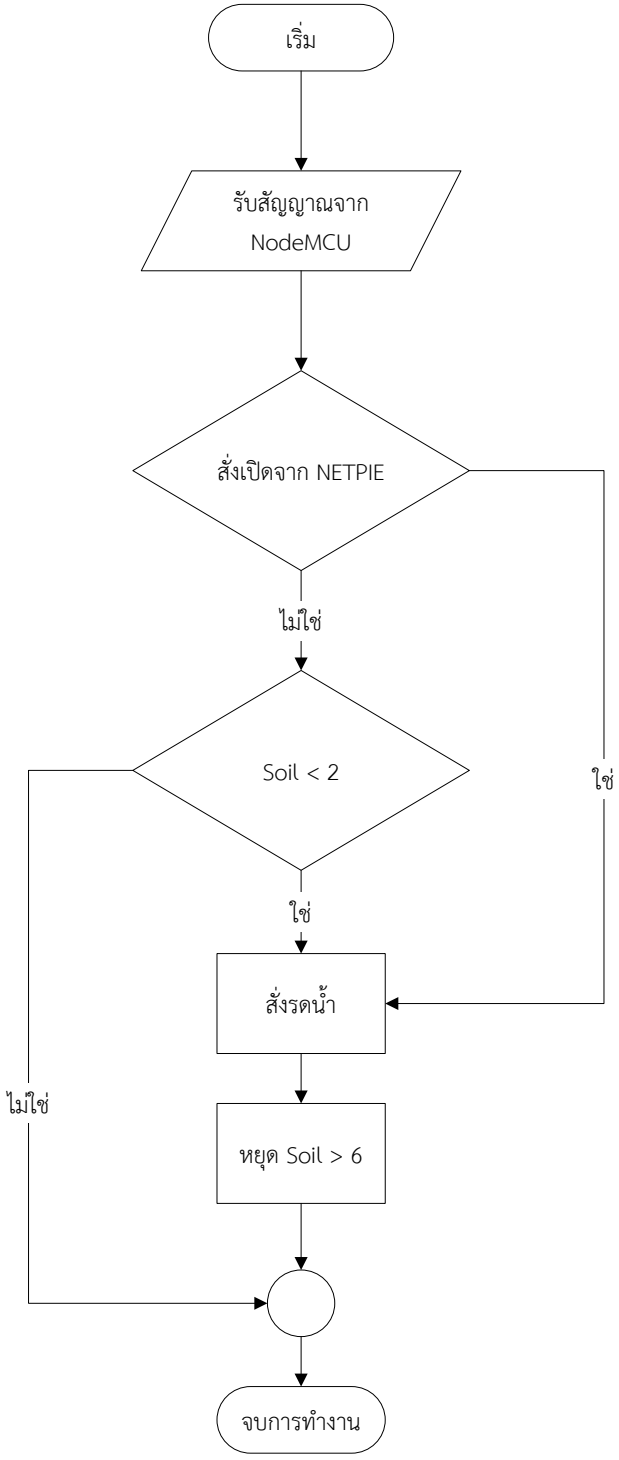

**รูปที่ 3.5** แผนผังการทำงานของ Soil Moisture Sensor

- ขา $V_{cc}$ ใช้ไฟขนาด 3V ถึง 5.5V ต่อเข้าบอร์ด NodeMCU v2 ที่ขา $V_{in}$ 5V
- ขา Data ต่อเข้าบอร์ด NodeMCU v2 ในส่วนของแถวช่องสัญญาณแอนะล็อกที่ขา GPIO4
- ขา GND ต่อเข้าบอร์ด NodeMCU v2 ในส่วนของแถวสัญญาณแอนะล็อกที่ขา GND




**รูปที่ 3.6** แสดงการต่อวงจร NodeMCU กับ Soil Moisture Sensor

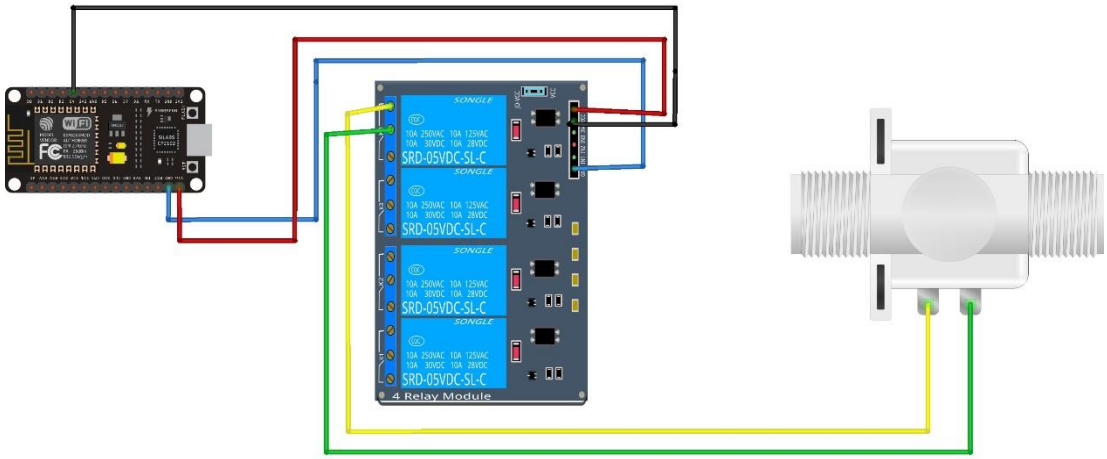
## 3.6 การเชื่อมต่อ NodeMCU v2 เข้ากับ Relay Module 4 Channels และ SOLENOID VALVE




**รูปที่ 3.7** แผนผังการทำงานของ Relay Module 4 Channels

- ขา $V_{cc}$ ใช้ไฟขนาด 5V ต่อเข้าบอร์ด NodeMCU v2 ที่ขา $V_{in}$ 5V

- ขา IN4 ต่อเข้าบอร์ด NodeMCU v2 ในส่วนของแถวช่องสัญญาณดิจิตอลที่ขา GPIO2
- ขา GND ต่อเข้าบอร์ด NodeMCU v2 ในส่วนของแถวสัญญาณแอนะล็อกที่ขา GND
- ขา ON และ ขา OFF จาก Relay Module 4 Channels ต่อเข้าโซลินอยวาวล์

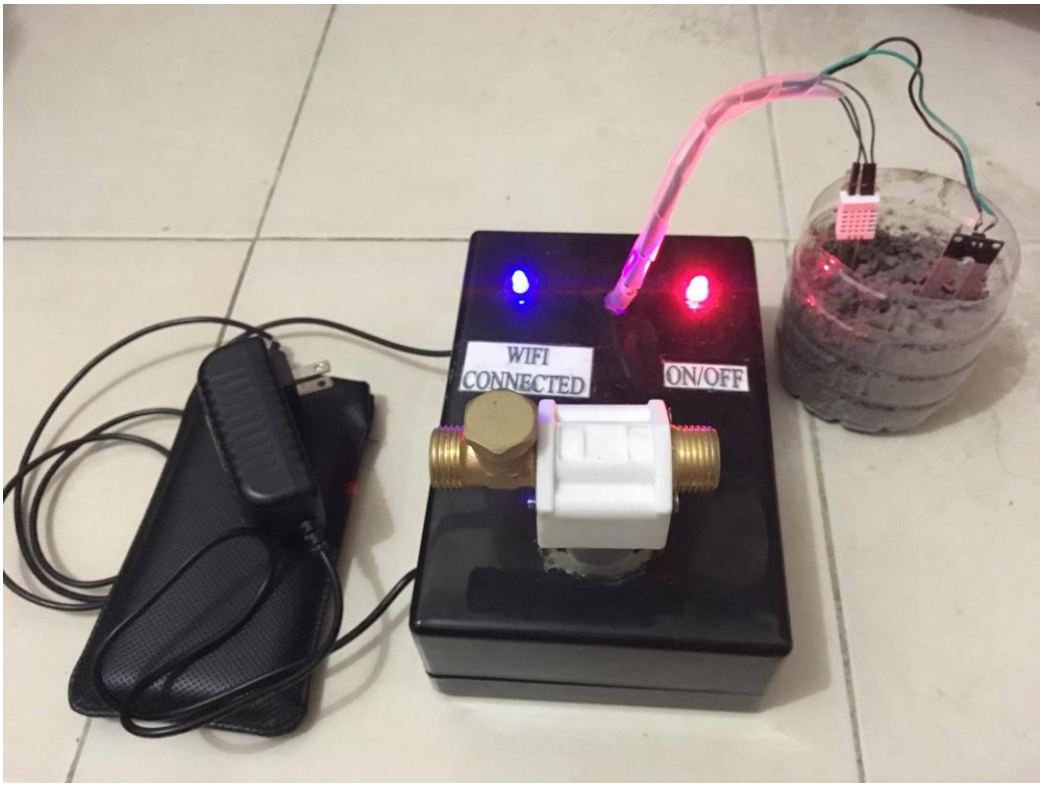



**รูปที่ 3.8** แสดงการต่อวงจร NodeMCU กับ Relay Module 4 Channels และ Solenoid Valve

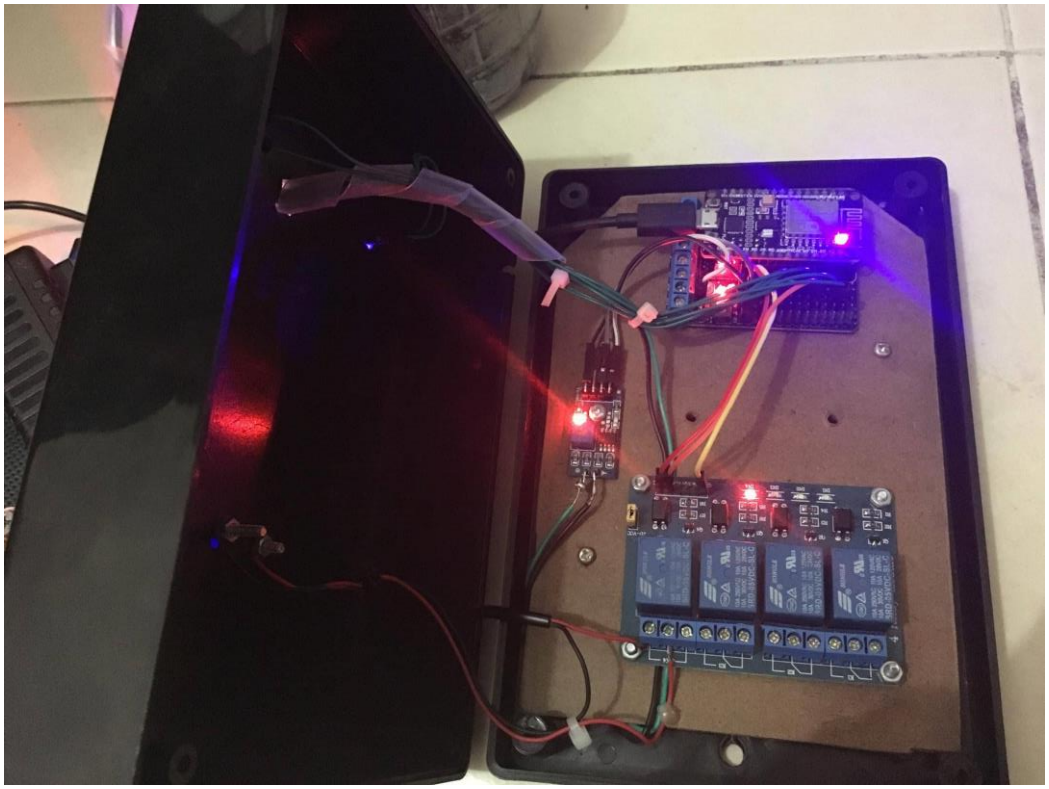
# บทที่ 4
# ผลการดำเนินงาน

เริ่มแรกในส่วนจองการทดลองนี้ จะทำการทดสอบการทำงานในส่วนต่างๆของวงจร Smart Garden นี้ทั้งเซนเซอร์ตรวจวัดอุณหภูมิกับความชื้นในอากาศ และเซนเซอร์วัดความชื้นในดิน

1. การเชื่อมต่ออุปกรณ์กับ NETPIE
2. ทดสอบวัดค่าอุณหภูมิและความชื้นในอากาศและวัดค่าความชื้นในดินผ่านเว็บบราวเซอร์
3. ทดสอบการควบคุมระบบจ่ายน้ำผ่านเว็บบราวเซอร์




**รูปที่ 4.1** รูปวงจร Smart Garden (ภายนอก)



**รูปที่ 4.2** รูปวงจร Smart Garden (ภายใน)

## การทดลองที่ 1

ทดสอบวัดค่าอุณหภูมิและความชื้นในอากาศผ่านเว็บบราวเซอร์

อุปกรณ์

1.NodeMCU

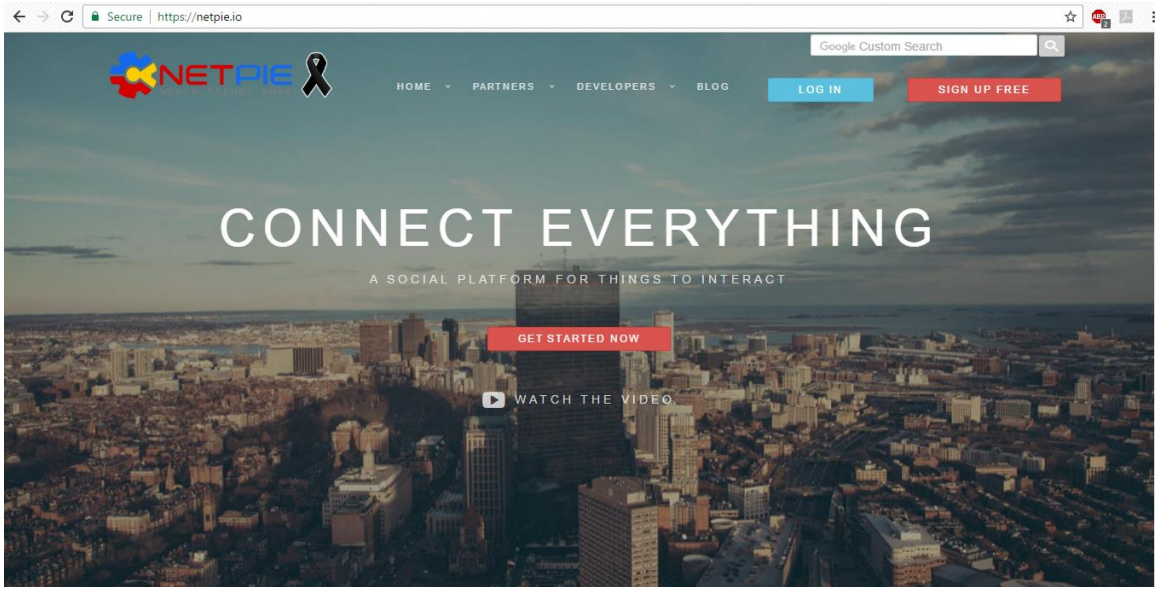
2. เครื่องคอมพิวเตอร์ที่ทำการติดตั้งโปรแกรมอาร์ดูไอโนเรียบร้อยแล้ว

3. เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ

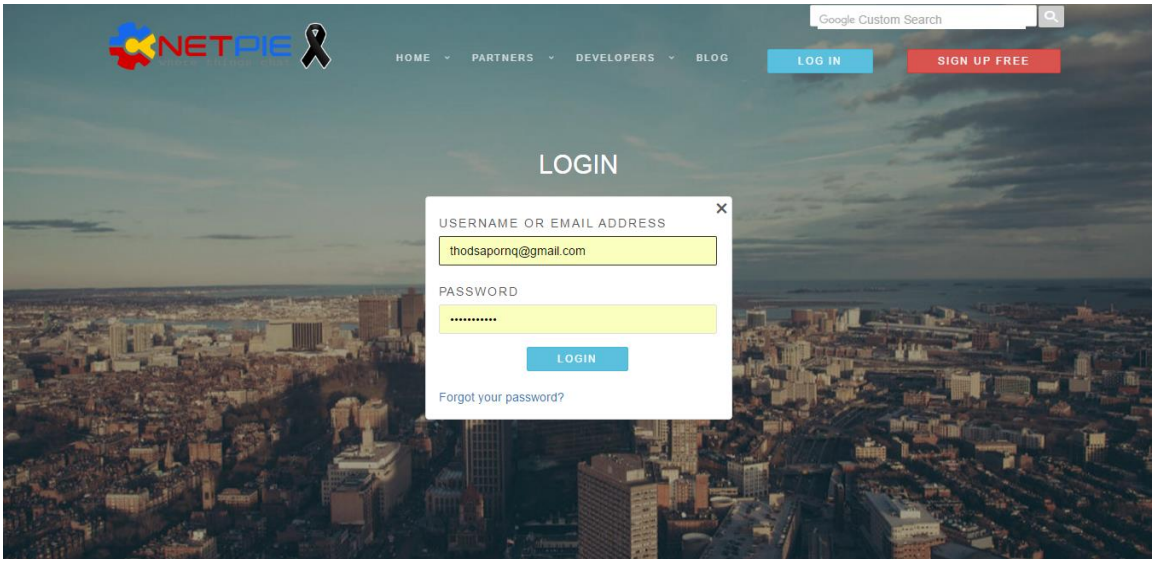
4. Internet Web Browser (Google Chrome, Internet Explorer, Firefox)

วิธีการทดลอง

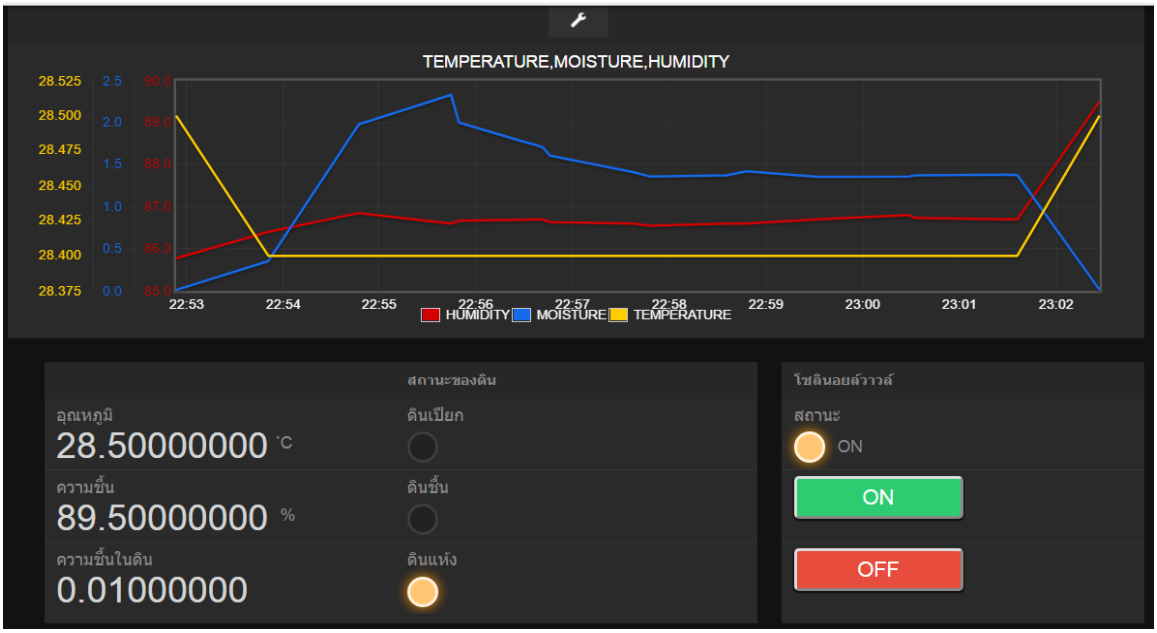
1. เปิด Internet Web Browser (Google Chrome, Internet Explorer, Firefox) โดยเข้าไป ที่เว็บไซต์ของ NETPIE
2. ทำการเข้าสู่ระบบด้วย ID และ Password ของ NETPIE
3. เข้าไปในส่วนของ FREEBOARDS
4. อ่านค่าที่ได้รับจากเซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ
5. อ่านค่าที่ได้รับจากเซนเซอร์วัดความชื้นในดิน
6. นำมาเปรียบเทียบกับอุปกรณ์วัดอุณหภูมิและความชื้น



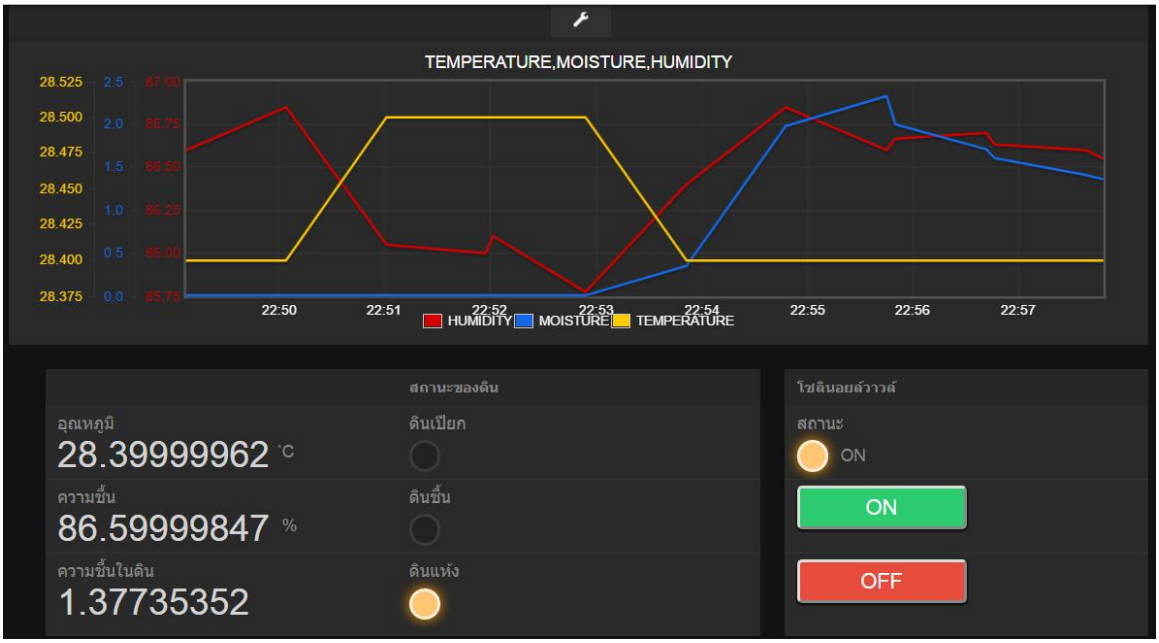

**รูปที่ 4.3** เปิด Internet Web Browser เข้าไปที่เว็บไซต์ของ NETPIE




**รูปที่ 4.4** ทำการเข้าสู่ระบบด้วย ID และ Password ของ NETPIE

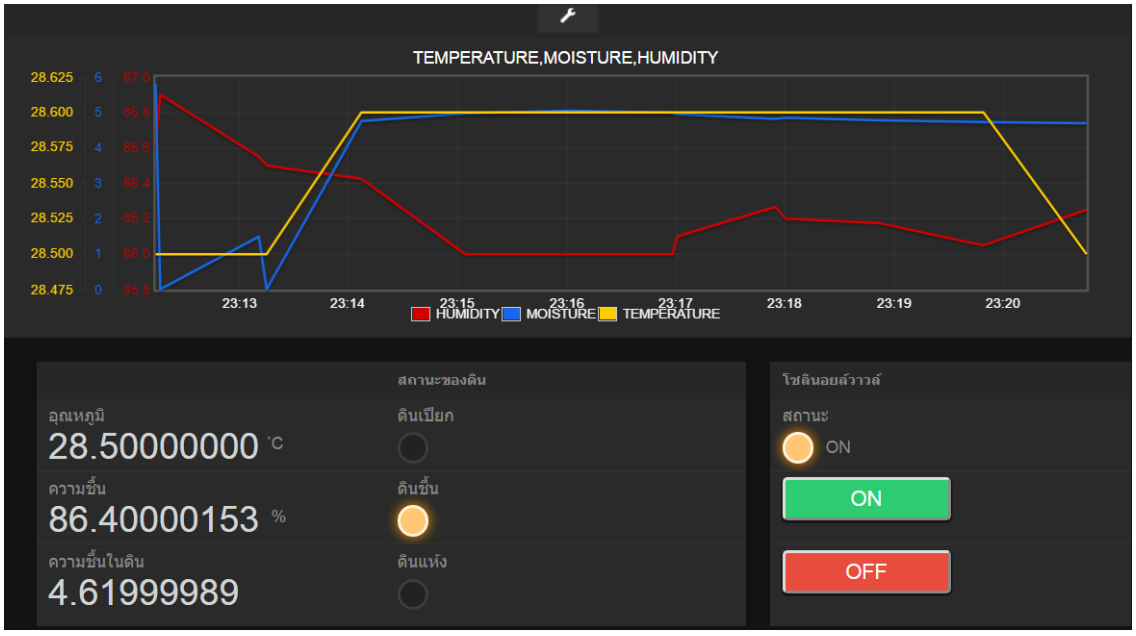



**รูปที่ 4.5** อ่านค่าที่ได้รับจากเซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ DHT22 แบบเรียลไทม์



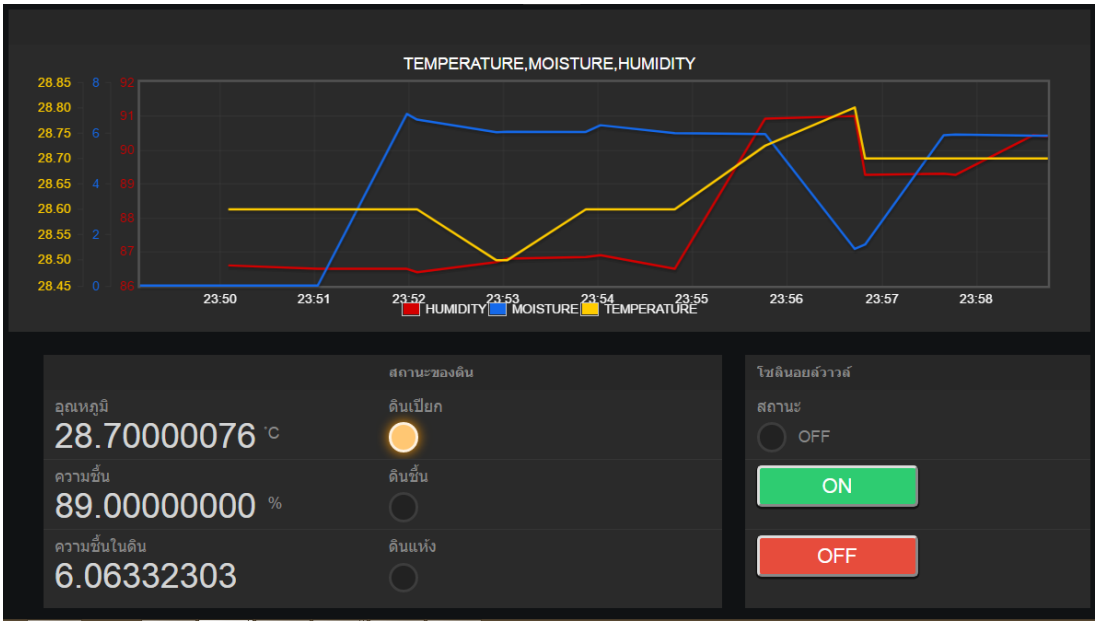

**รูปที่ 4.6** อ่านค่าที่ได้รับจาก Soil Moisture Sensor (เซนเซอร์วัดความชื้นในดิน) เมื่อดินแห้ง

**รูปที่ 4.7** อุปกรณ์แสดงค่าการวัดความชื้นเมื่ออยู่ในดินแห้ง




**รูปที่ 4.8** อ่านค่าที่ได้รับจาก Soil Moisture Sensor (เซนเซอร์วัดความชื้นในดิน) เมื่อดินชื้น

**รูปที่ 4.9** อุปกรณ์แสดงค่าการวัดความชื้นเมื่ออยู่ในดินชื้น




**รูปที่ 4.10** อ่านค่าที่ได้รับจาก Soil Moisture Sensor (เซนเซอร์วัดความชื้นในดิน) เมื่อดินเปียก

**รูปที่ 4.11** อุปกรณ์แสดงค่าการวัดความชื้นเมื่ออยู่ในดินเปียก

**ตารางที่ 4.1** ผลการทดลองการวัดค่าความชื้นในดินประเภทดินแห้ง

| ค่าความชื้นในดินที่วัดได้จ | ค่าความชื้นในดินที่วัดได้จากเซ็นเซอร์ | ค่าความคลาดเคลื่อน (%) |
|---|---|---|
| 1.5 | 1.38 | 1.2 |
| 1.5 | 1.37 | 1.3 |
| 1.4 | 1.37 | 0.3 |
| 1.4 | 1.35 | 0.5 |
| 1.4 | 1.31 | 0.9 |
| 1.4 | 1.31 | 0.9 |
| 1.4 | 1.31 | 0.9 |
| 1.4 | 1.27 | 1.3 |
| 1.3 | 1.25 | 0.5 |
| 1.3 | 1.25 | 0.5 |

การทดลองการวัดค่าความชื้นในดินประเภทดินแห้ง ค่าความคลาดเคลื่อนเฉลี่ย = 0.83%

**ตารางที่ 4.2** ผลการทดลองการวัดค่าความชื้นในดินประเภทดินชื้น

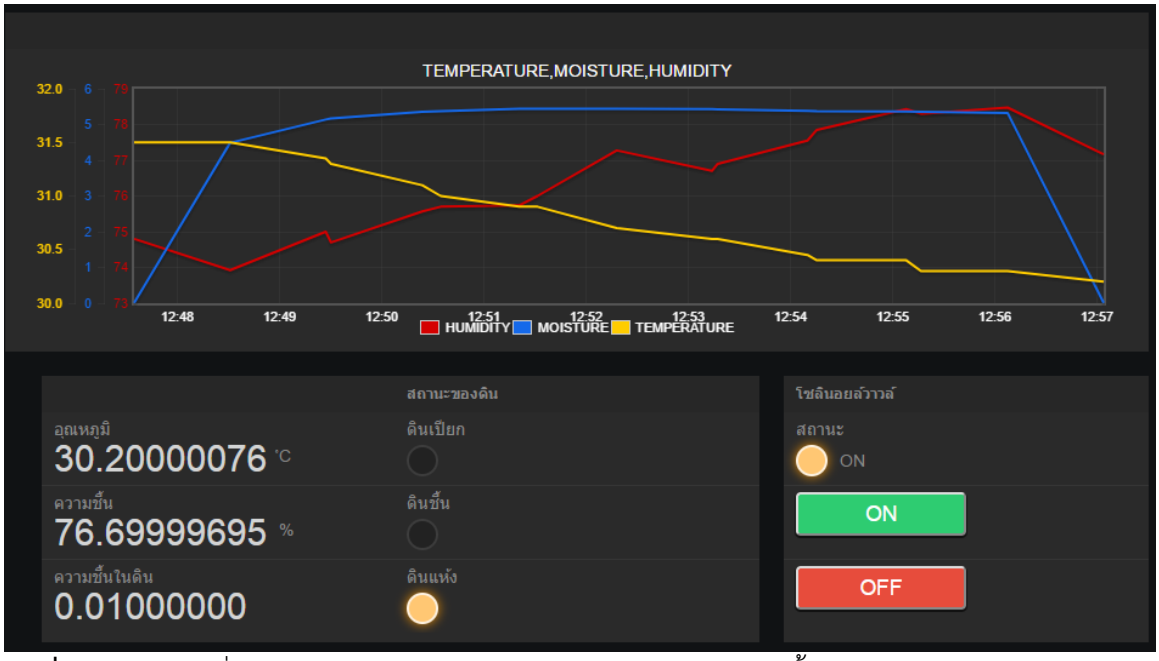
| ค่าความชื้นในดินที่วัดได้จริง | ค่าความชื้นในดินที่วัดได้จากเซ็นเซอร์ | ค่าความคลาดเคลื่อน (%) |
|---|---|---|
| 5.1 | 4.94 | 1.6 |
| 5.1 | 4.85 | 2.5 |
| 4.9 | 4.84 | 0.6 |
| 4.9 | 4.76 | 1.4 |
| 4.9 | 4.71 | 1.9 |
| 4.9 | 4.71 | 1.9 |
| 4.9 | 4.71 | 1.9 |
| 4.9 | 4.63 | 2.7 |
| 4.8 | 4.63 | 1.7 |
| 4.7 | 4.51 | 1.9 |

การทดลองการวัดค่าความชื้นในดินประเภทดินชื้น ค่าความคลาดเคลื่อนเฉลี่ย = 1.81%

**ตารางที่ 4.3** ผลการทดลองการวัดค่าความชื้นในดินประเภทดินเปียก

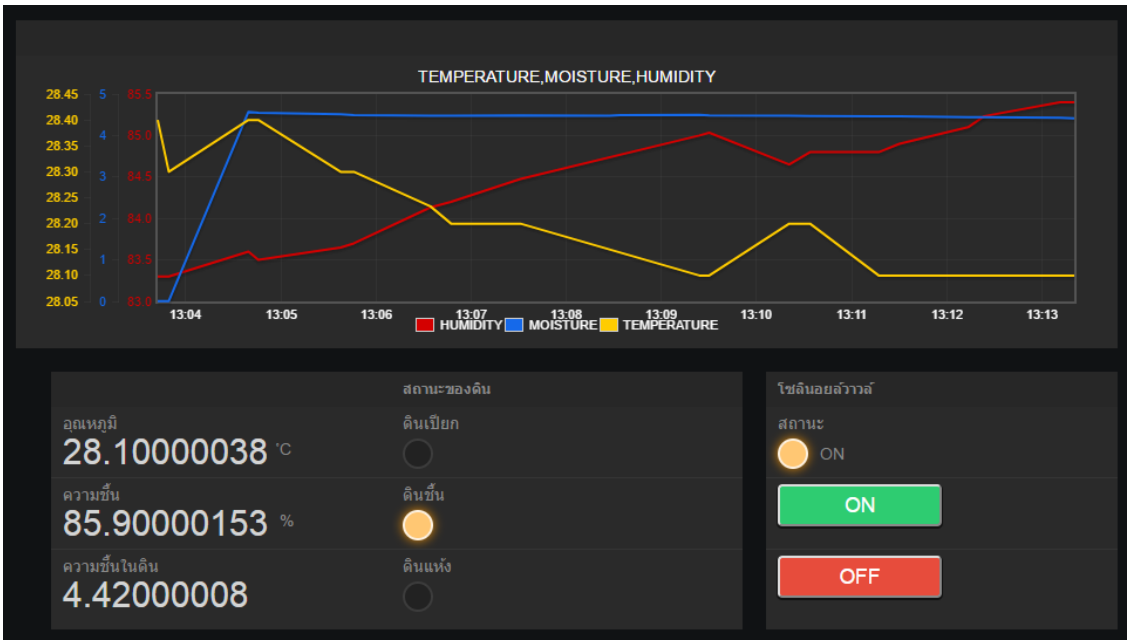
| ค่าความชื้นในดินที่วัดได้จริง | ค่าความชื้นในดินที่วัดได้จากเซ็นเซอร์ | ค่าความคลาดเคลื่อน (%) |
|---|---|---|
| 6.2 | 6.06 | 1.4 |
| 6.2 | 6.06 | 1.4 |
| 6.1 | 6.04 | 0.6 |
| 6.1 | 6.03 | 0.7 |
| 6.1 | 6.03 | 0.3 |
| 6.1 | 6.00 | 1.0 |
| 6.1 | 5.96 | 1.4 |
| 6.0 | 5.93 | 0.7 |
| 6.0 | 5.93 | 0.7 |
| 6.0 | 5.93 | 0.7 |

การทดลองการวัดค่าความชื้นในดินเปียก ค่าความคลาดเคลื่อนเฉลี่ย = 0.89%




**รูปที่ 4.12** อ่านค่าที่ได้รับจาก DHT 22 (เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ) ในอุณหภูมิภายนอก

**รูปที่ 4.13** อ่านค่าที่ได้รับจากเครื่องมือวัดอุณหภูมิและความชื้นในอากาศจากภายนอก




**รูปที่ 4.14** อ่านค่าที่ได้รับจาก DHT 22 (เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ) ในอุณหภูมิภายในห้อง

**รูปที่ 4.15** อ่านค่าที่ได้รับจากเครื่องมือวัดอุณหภูมิและความชื้นในอากาศภายในห้อง

**ตารางที่ 4.4** ผลการทดลองการวัดค่าอุณหภูมิภายนอก

| ค่าอุณหภูมิที่วัดได้จริง (℃) | ค่าอุณหภูมิที่วัดได้จากเซ็นเซอร์(℃) | ค่าความคลาดเคลื่อน (%) |
|---|---|---|
| 32 | 31.50 | 1.56 |
| 32 | 31.29 | 2.22 |
| 32 | 31.39 | 1.90 |
| 32 | 31.29 | 2.22 |
| 32 | 31.39 | 1.90 |
| 32 | 31.19 | 2.53 |
| 31 | 31.00 | 0.00 |
| 31 | 30.90 | 0.32 |
| 31 | 30.69 | 1.00 |
| 31 | 30.20 | 2.58 |

การทดลองการวัดค่าอุณหภูมิภายนอก ค่าความคลาดเคลื่อนเฉลี่ย = 1.62%

**ตารางที่ 4.5** ผลการทดลองการวัดค่าอุณหภูมิภายในห้อง

| ค่าอุณหภูมิที่วัดได้จริง (℃) | ค่าอุณหภูมิที่วัดได้จากเซ็นเซอร์ (℃) | ค่าความคลาดเคลื่อน (%) |
|---|---|---|
| 29.5 | 28.89 | 2.06 |
| 29.5 | 28.60 | 3.05 |
| 29 | 28.49 | 1.76 |
| 29 | 28.30 | 2.41 |
| 29.5 | 28.49 | 3.42 |
| 29 | 28.39 | 2.10 |
| 28.5 | 28.19 | 1.08 |
| 29 | 28.19 | 2.79 |
| 28.5 | 28.10 | 1.40 |
| 28.5 | 28.00 | 1.75 |

การทดลองการวัดค่าอุณหภูมิภายในห้อง ค่าความคลาดเคลื่อนเฉลี่ย = 2.182%

**ตารางที่ 4.6** ผลการทดลองการวัดค่าความชื้นภายนอก

| ค่าความชื้นที่วัดได้จริง (%) | ค่าความชื้นที่วัดได้จากเซ็นเซอร์ (%) | ค่าความคลาดเคลื่อน (%) |
|---|---|---|
| 72.5 | 75.19 | 2.69 |
| 72 | 74.80 | 2.80 |
| 72.5 | 76.09 | 3.59 |
| 73 | 77 | 4 |
| 74 | 78.4 | 4.4 |
| 74 | 78.4 | 4.4 |
| 75 | 79 | 4 |
| 77 | 79.4 | 2.4 |
| 77 | 79.5 | 2.5 |
| 78 | 80.9 | 2.9 |

การทดลองการวัดค่าความชื้นภายนอก ค่าความคลาดเคลื่อนเฉลี่ย = 3.37%

**ตารางที่ 4.7** ผลการทดลองการวัดค่าความชื้นภายในห้อง

| ค่าความชื้นที่วัดได้จริง(%) | ค่าความชื้นที่วัดได้จากเซ็นเซอร์ (%) | ค่าความคลาดเคลื่อน (%) |
|---|---|---|
| 84 | 83.20 | 0.8 |
| 84 | 83.59 | 0.41 |
| 84.5 | 83.20 | 1.3 |
| 84.5 | 83.89 | 0.61 |
| 84.5 | 84.19 | 0.31 |
| 84.5 | 84.80 | 0.3 |
| 85 | 85.29 | 0.29 |
| 85 | 85.49 | 0.49 |
| 85 | 85.90 | 0.9 |
| 85.5 | 86.10 | 0.60 |

ผลการทดลองการวัดค่าความชื้นภายในห้อง ค่าความคลาดเคลื่อนเฉลี่ย = 0.6%

## การทดลองที่ 2

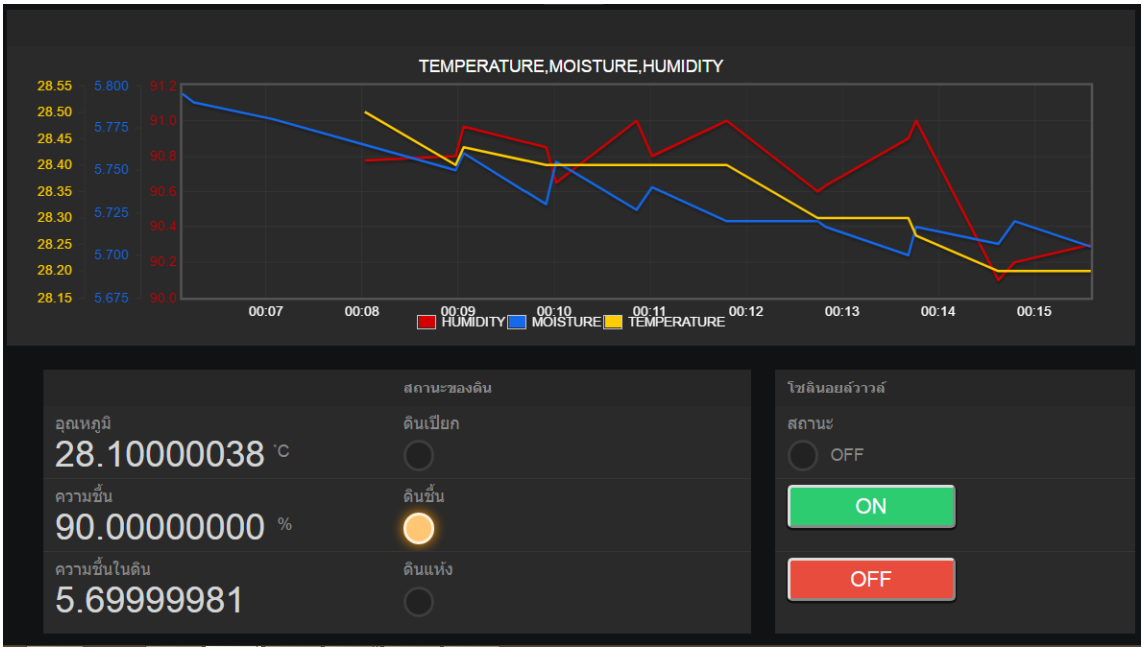
ทดสอบการควบคุมระบบจ่ายน้ำผ่านเว็บบราวเซอร์

อุปกรณ์

1. NodeMCU
2. เครื่องคอมพิวเตอร์ที่ทำการติดตั้งโปรแกรมอาร์ดูไอโนเรียบร้อยแล้ว
3. Relay Module 4 Channels
4. SOLENOID VALVE
5. Internet Web Browser (Google Chrome, Internet Explorer, Firefox)

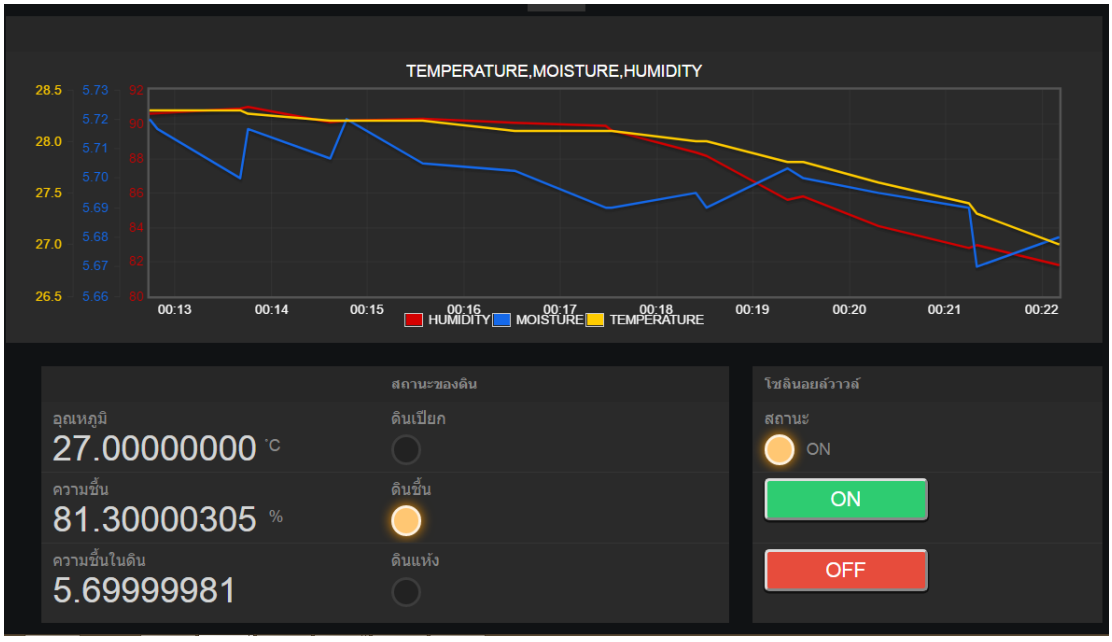
วิธีการทดลอง

1. เปิด Internet Web Browser (Google Chrome, Internet Explorer, Firefox) โดยเข้าที่ NETPIE
2. กดปุ่ม ON เพื่อเปิดการทำงาน โซลินอยด์วาล์ว
3. กดปุ่ม OFF เพื่อเปิดการทำงาน โซลินอยด์วาล์ว




**รูปที่ 4.16** แสดงค่าสถานะขณะปิด SOLENOID VALVE

**รูปที่ 4.17** แสดงวงจรขณะควบคุม SOLENOID VALVE ขณะปิดวงจร




**รูปที่ 4.18** แสดงค่าสถานะขณะสั่งเปิด SOLENOID VALVE

**รูปที่ 4.19** แสดงวงจรขณะควบคุม SOLENOID VALVE ขณะเปิดวงจร

ผลการทดลอง

เมื่อเปิดโปรแกรม Internet Web Browser และทำการเข้าสู้ระบบของ NETPIE แล้ว ได้ไปที่ APPICATION เข้าไปที่ FREEBROADS จะปรากฏหน้าเว็บดังในรูปที่ 4.13 จะมีปุ่ม ที่ใช้ในการควบคุมการทำงานโซลินอยด์วาล์ว โดยที่จะปุ่ม 2 ปุ่ม คือ เปิดไฟ กับ ปิดไฟ แล้วเมื่อทำการกดปุ่มเปิดไฟ โซลินอยด์จะทำงานเปิดให้น้ำไหลผ่านได้ แล้วเมื่อทำการกดปุ่มปิดไฟ โซลินอยด์ จะปิดการทำงานไม่ให้น้ำไหลผ่านไปได้ และจะมีไฟแสดงสถานการณ์ทำงานของโซลินอยด์ แสดงว่าอยู่ในโหมดการทำงาน หรือ ปิดการทำงาน

# บทที่ 5
# สรุปผลและข้อเสนอแนะ

## 5.1 สรุปผล

จากการทดสอบการทำงานของโครงงานอินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะ ซึ่งกระบวนการในการพัฒนานั้น ประกอบด้วย 3 กระบวนการ คือ 1. กระบวนการเขียนโปรแกรม 2. กระบวนการในการเชื่อมต่อเครือข่าย 3. กระบวนในการควบคุมอุปกรณ์เซนเซอร์และระบบจ่ายน้ำ ซึ่งปัญหาที่พบของการทำงานในระยะแรกคือการเชื่อมต่อของระบบเครือข่ายมีปัญหาไม่สามารถเชื่อมต่อได้ และ NodeMCU เกิดการรีเซ็ตตัวเองจากกระแสไฟเลี้ยงไม่เพียงพอ โดยในท้ายที่สุดการทำงานของโครงงานอินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะสามารถควบคุมการทำงานได้ตามวัตถุประสงค์ โดยสามารถแสดงอุณหภูมิและความชื้นบนเว็บบราวเซอร์และควบคุมการจ่ายน้ำบนเว็บบราวเซอร์

## 5.2 ข้อเสนอแนะ

5.2.1 เพิ่มคำสั่งของ Smart Garden ให้สามารถสั่งงานผ่าน NETPIE ได้หลากหลายรูปแบบมากขึ้นในอนาคต

5.2.2 เพิ่มจำนวน Sensor เพื่อให้ผลการวัดค่าในพื้นที่ ที่มีความหลากหลายในพรรณไม้เพื่อช่วยในการตัดสินใจปิด เปิดการจ่ายน้ำในระบบ Smart Garden เพื่อให้ระบบได้ทำงานมีประสิทธิภาพสูงสุด

# เอกสารอ้างอิง

กมลชนก ลอมโฮม, วรรณิศา ผางต๊ะ. 2555.การควบคุมหุ่นยนต์ผ่านเครือข่ายไร้สายโดยใช้อาร์ดูไอโน. ปริญญานิพนธ์, ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ องครักษ์.

http://library.stou.ac.th/blog/2014/03/06/internet-of-things- (21/3/2560)

http://www.ioxhop.com/product /87/เซ็นเซอร์วัดความชื้นในดิน-soil-moisture-sensor

http://www.myarduino.net/product/697/dht22-am2302-moduleโมดูลวัดอุณหภูมและควา temperature-and-humidity-sensor-module-พร้อมสายไฟ

http://cpre.kmutnb.ac.th/esl/learning/index.php?article=esp8266-modules

http://www.thaieasyelec.com/article-wiki/review-product- article/ตัวอย่างการใช้งาน-arduino- relay-module-ควบคุมการปิดเปิดเครื่องใช้ไฟฟ้า.html

http://www.tic.co.th/index.php?op=tips-detail&id=110

https://medium.com/sathittham/galileo-gen2-getting-started-6การอ่านค่าความชื้นในดิน-soil-moisture-sensor-6f54e0dd92d0

https://www.facebook.com/notes /อธิษฐาน-โป๊ยเซียน/ปัจจัยที่จำเป็นต่อการเจริญเติบโตของพืช/1412557258976660/

http://www.ioxhop.com/article/13/esp8266 -ตอนที่-1-รู้จักกับ-esp-และรุ่นที่นิยมใช้งาน

https://www.nectec.or.th/innovation/innovation-software/netpie.html

https://ict.tu.ac.th/document/manual-internet/manual_wifi.pdf

# ภาคผนวก

# ภาคผนวก

## 1.โค้ดโปรแกรมการทำงานทั้งหมด

```
#include <AuthClient.h>
#include <MicroGear.h>
#include <ESP8266WiFi.h>
#include "DHT.h"

const char* ssid     = "iPhone"; (ชื่อ Wifi)
const char* password = "12345678910"; (รหัส Wifi)

#define RELAY D4
#define LEDPin D8
#define APPID       "SWUSmartGarden"
#define GEARKEY     "xxxxxxxxxxxxxxxxxxxx"
#define GEARSECRET  "xxxxxxxxxxxxxxxxxxxxxxxx"
#define ALIAS       "NodeMCU"
#define SCOPE       ""
#define FEEDID      "newfeed"
#define DHTPIN D2
#define DHTTYPE DHT22
#define SensorMoist A0
#define red D7
#define yellow D6
#define green D5

int varMoist;
int moist;
DHT dht(DHTPIN, DHTTYPE);
WiFiClient client;
AuthClient *authclient;
```

```
 MicroGear microgear(client);

void onConnected(char *attribute, uint8_t* msg, unsigned int msglen) {
  Serial.println("Connected to NETPIE...");
  microgear.setAlias(ALIAS);
}

void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen) {
  Serial.print("Incoming message --> ");
  Serial.print(topic);
  Serial.print(" : ");
  char strState[msglen];
  for (int i = 0; i < msglen; i++) {
    strState[i] = (char)msg[i];
    Serial.print((char)msg[i]);
  }
  Serial.println();

  String stateStr = String(strState).substring(0, msglen);
  if (stateStr == "ON") {
    digitalWrite(D4, LOW);
    microgear.chat("Sensor/LED", "ON");
  }
  else if (stateStr == "OFF") {
    digitalWrite(D4, HIGH);
    microgear.chat("Sensor/LED", "OFF");
  }
}

void setup() {
  digitalWrite(D4, HIGH);
```

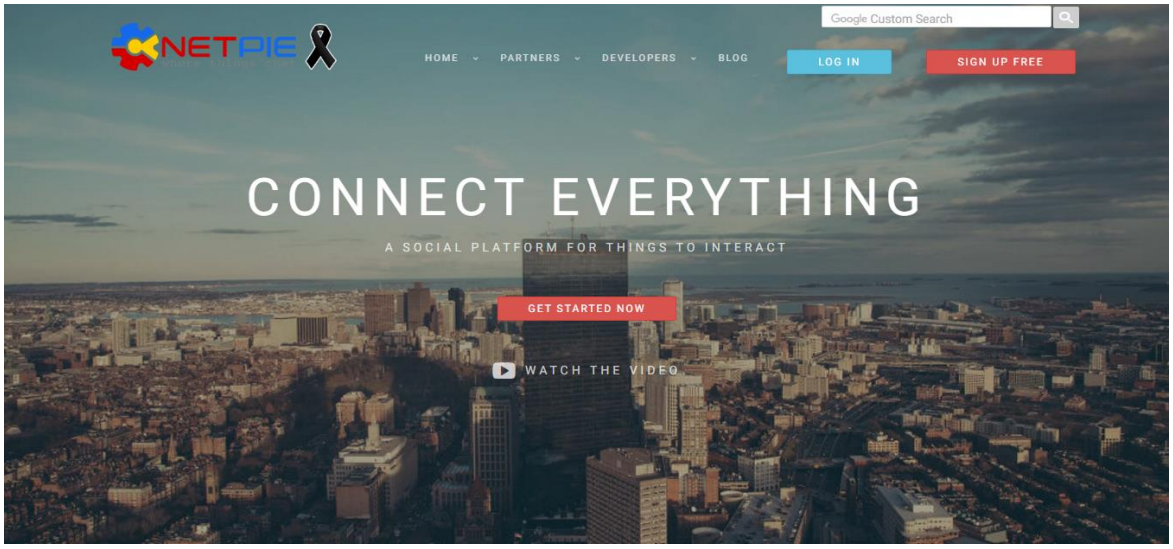
```
  Serial.begin(115200);
  Serial.println("Starting...");
  pinMode(SensorMoist, INPUT);
  pinMode(RELAY, OUTPUT);
  pinMode(LEDPin, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);

  microgear.on(MESSAGE, onMsghandler);
  microgear.on(CONNECTED, onConnected);

  if (WiFi.begin(ssid, password)) {
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    microgear.resetToken();
    microgear.init(GEARKEY, GEARSECRET, SCOPE);
    microgear.connect(APPID);
  }
}
void loop() {
  if (microgear.connected()) {
    microgear.loop();
    Serial.println("connect...");
    float h = dht.readHumidity();
```

```
    float t = dht.readTemperature();
    varMoist = analogRead(SensorMoist);
    moist = 1023-varMoist;
    if (varMoist >= 900) {
      digitalWrite(D7, HIGH);
      digitalWrite(D4, LOW);
      microgear.chat("Sensor/LED", "ON");
      microgear.chat("Sensor/red", "ON");
      microgear.chat("Sensor/yellow", "OFF");
      microgear.chat("Sensor/green", "OFF");
    }
    else if (varMoist >= 500 && varMoist < 900) {
      digitalWrite(D6, HIGH);
      microgear.chat("Sensor/red", "OFF");
      microgear.chat("Sensor/yellow", "ON");
      microgear.chat("Sensor/green", "OFF");
    }
    else if (varMoist < 500) {
      digitalWrite(D5, HIGH);
      digitalWrite(D4, HIGH);
      microgear.chat("Sensor/LED", "OFF");
      microgear.chat("Sensor/red", "OFF");
      microgear.chat("Sensor/yellow", "OFF");
      microgear.chat("Sensor/green", "ON");
    }
    String Data = "{\"humidity\":"; Data += h; Data += ",\"temperature\":"; Data += t;
Data += ",\"moisture\":"; Data += varMoist; Data += "}";
    microgear.chat("Sensor/Temperature",dht.readTemperature()); // ส่งข้อความถึง
server
    microgear.chat("Sensor/Humidity",dht.readHumidity());
    microgear.chat("Sensor/Moist",varMoist);
```

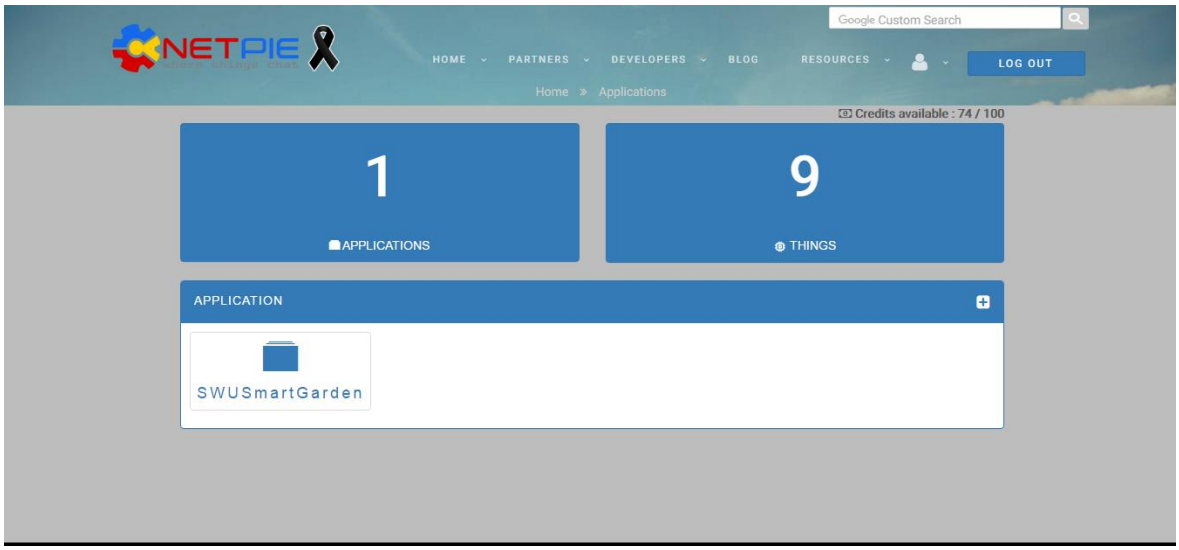
```
    microgear.writeFeed("newfeed",Data,"7f9xZmFUROnNqsahniZPaytiQ6z7ITLQ");
    digitalWrite(D8, HIGH);
    delay(500);
  }
  else {
    Serial.println("connection lost, reconnect...");
    digitalWrite(D8, LOW);
    microgear.connect(APPID);
  }
  delay(100);
}
```

## 2.การเชื่อมต่อ NodeMCU กับ NETPIE ด้วย ESP8266

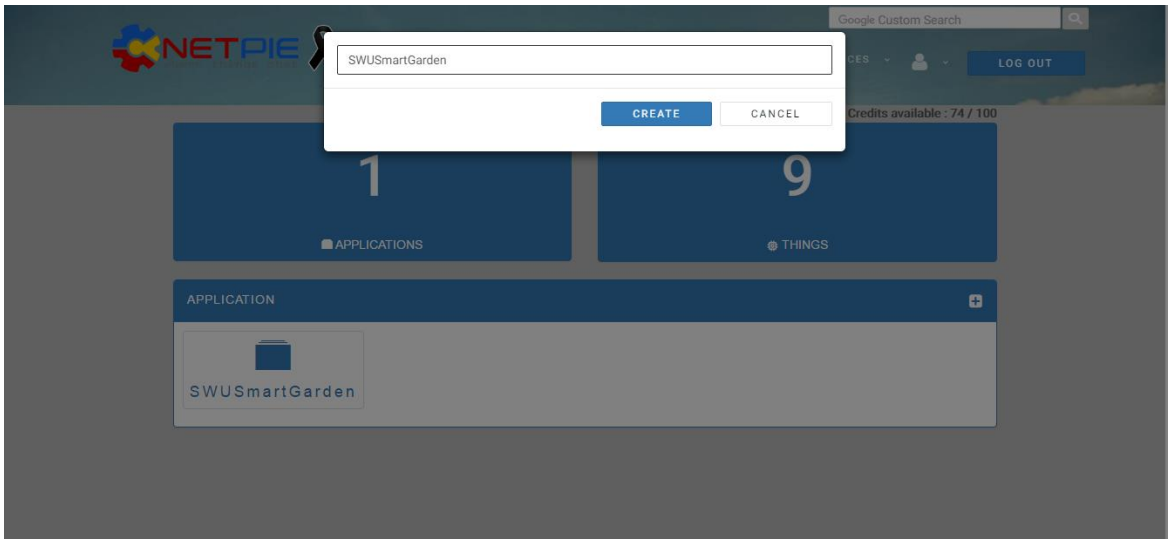
เข้าเว็บ NETPIE.io แล้วสมัครสมาชิกของ NETPIE หลังจากที่เราสมัครเรียบร้อย ให้เรา login แล้วเข้า application



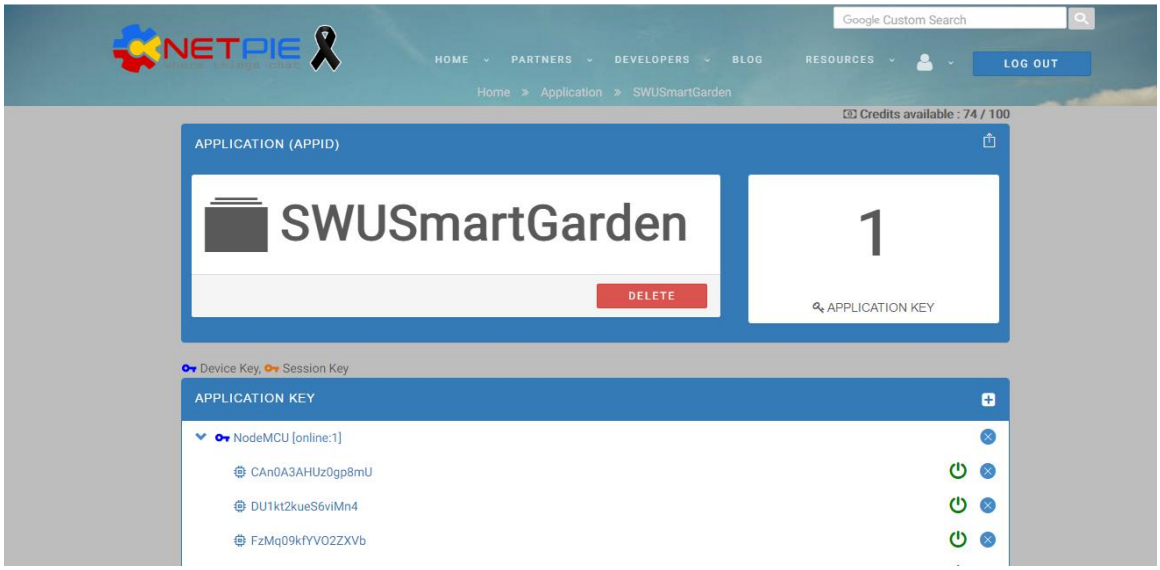

**รูปที่ 2.1** เมื่อเข้ามาแล้วจะเจอหน้านี้



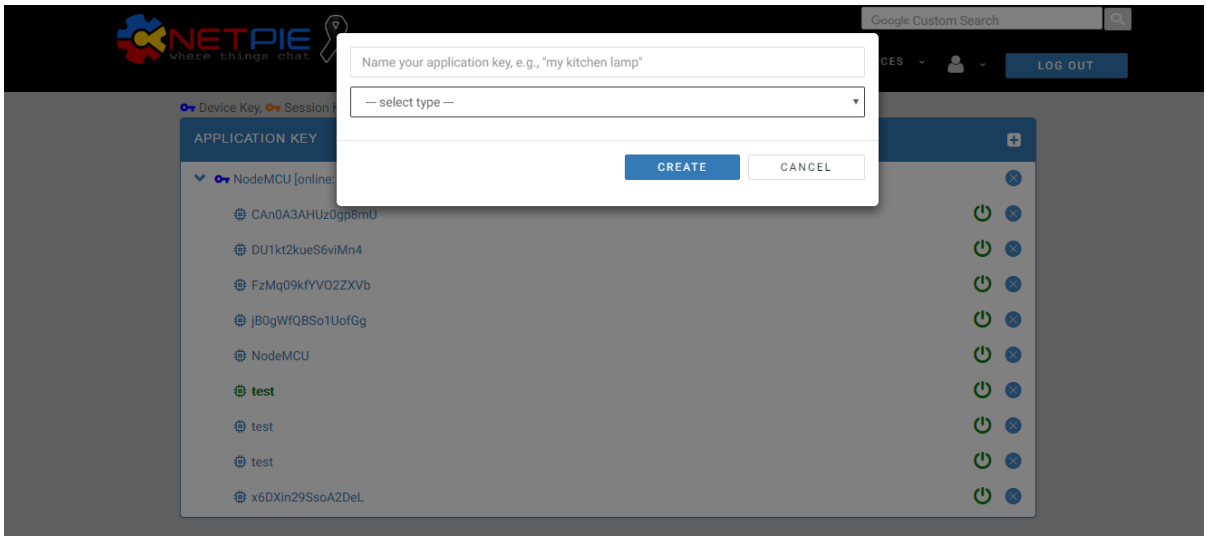

**รูปที่ 2.2** Application คือ Application ID ที่เราสร้างไว้




**รูปที่ 2.3** เมื่อเราสร้าง Application ID แล้วจะปรากฏหน้านี้ขึ้น

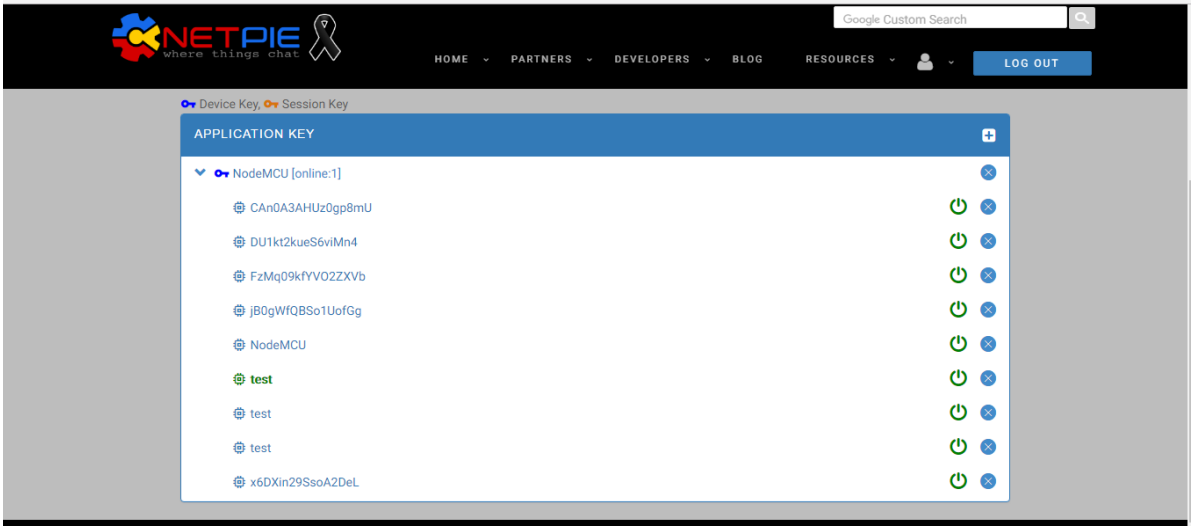



**รูปที่ 2.4** ในขั้นตอนต่อไปให้เราทำการสร้าง Application Key และ Application Secret ซึ่งทั้งสองค่านี้จะ unique ไม่ซ้ำกับใคร โดยในการสร้างให้กด + ที่ Application Key จะมีหน้าต่างปรากฏ

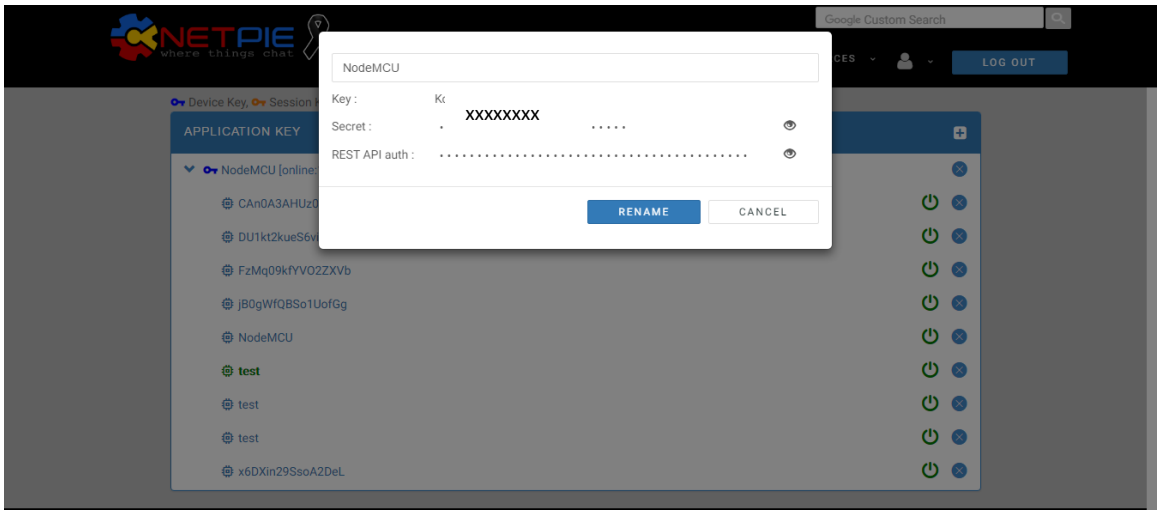



**รูปที่ 2.5** ในช่องแรกคือชื่อ อุปกรณ์ หรือ gear หรือ ALIAS ช่องถัดมาคือเลือกประเภทของ Key ซึ่งมีสองประเภท

- Device Key เป็น key ที่ไม่มีวันหมดอายุ ใช้กับอะไรก็ได้
- Session Key เป็น key ที่ไว้ใช้สำหรับ html5 มีอายุขัยการใช้งาน
- การ key นั้นให้เลือกตามความต้องการของเราได้เลยครับ อย่างในตัวอย่างผมจะเลือก Device Key




**รูปที่ 2.6** ตั้งชื่อ ALIAS ว่า NodeMCU ถ้าเราคลิกที่ตัวนี้จะปรากฏ Key และ Secret ให้ดู




**รูปที่ 2.7** ซึ่งค่าต่างๆเหล่ามีไว้ใช้เชื่อมต่อ Things ต่างๆของเรากับ NETPIE.io Cloud Service ในรูปแบบของการ Authentication ซึ่งค่าต่างๆเหล่านี้จะมีความเป็น unique ที่จะไม่ซ้ำใคร

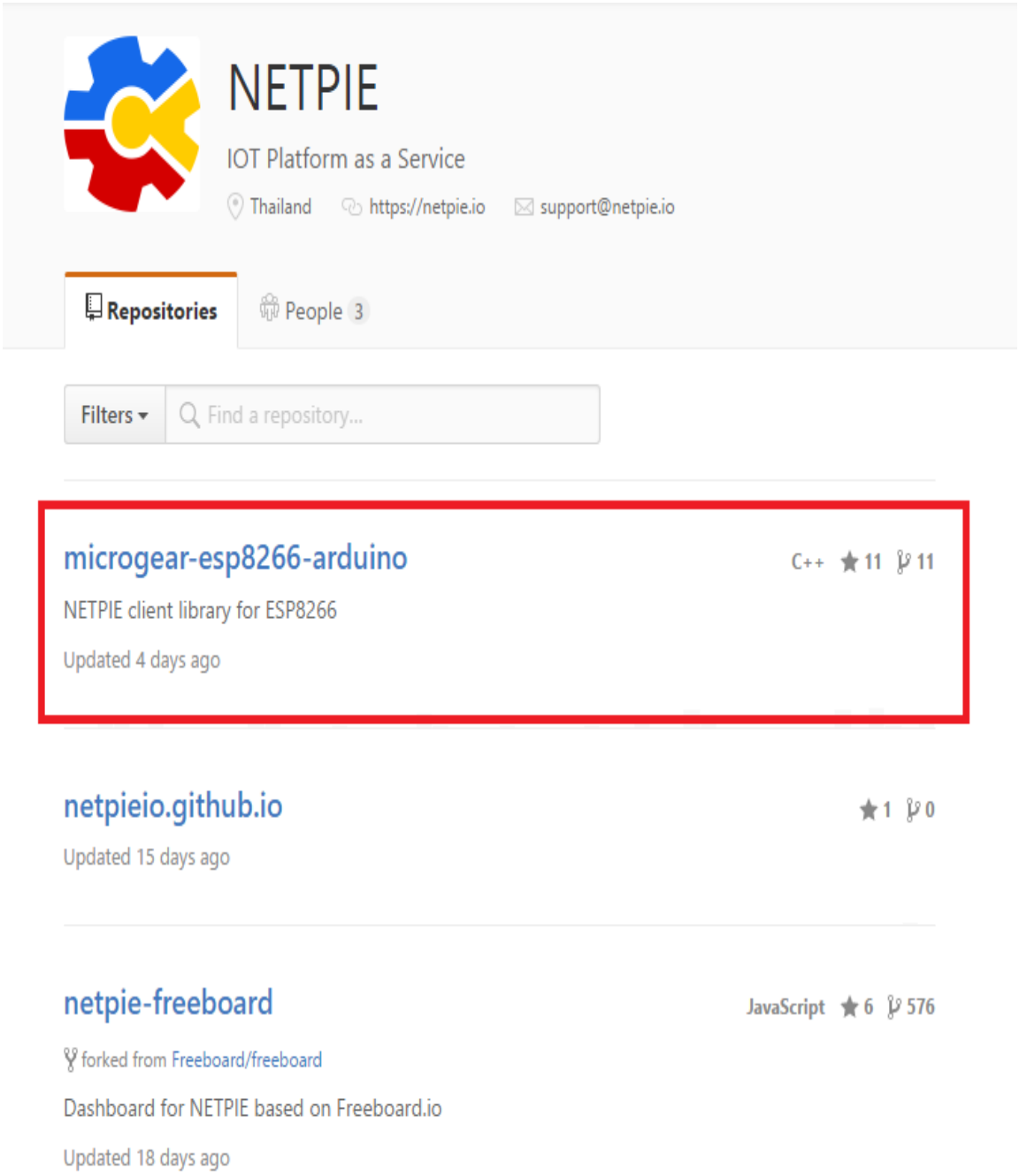
ข้อมูลต่างๆทั้ง App ID, App Key และ App Secret ห้ามนำไปเปิดเผยให้ใครโดยเด็ดขาดเพื่อความปลอดภัยในข้อมูล เพราะอาจมีผู้ประสงค์ร้ายนำไปใช้ได้
ซึ่งในตัวอย่างตอนนี้จะมีค่าต่างๆดังนี้

- App ID = NETPIEBasic
- App Key = jlIoBeurGfr6ddH
- App Secret = qqXdMEoTMSKrzx3jRxl2EEls
- Alias = NetpieAlias1

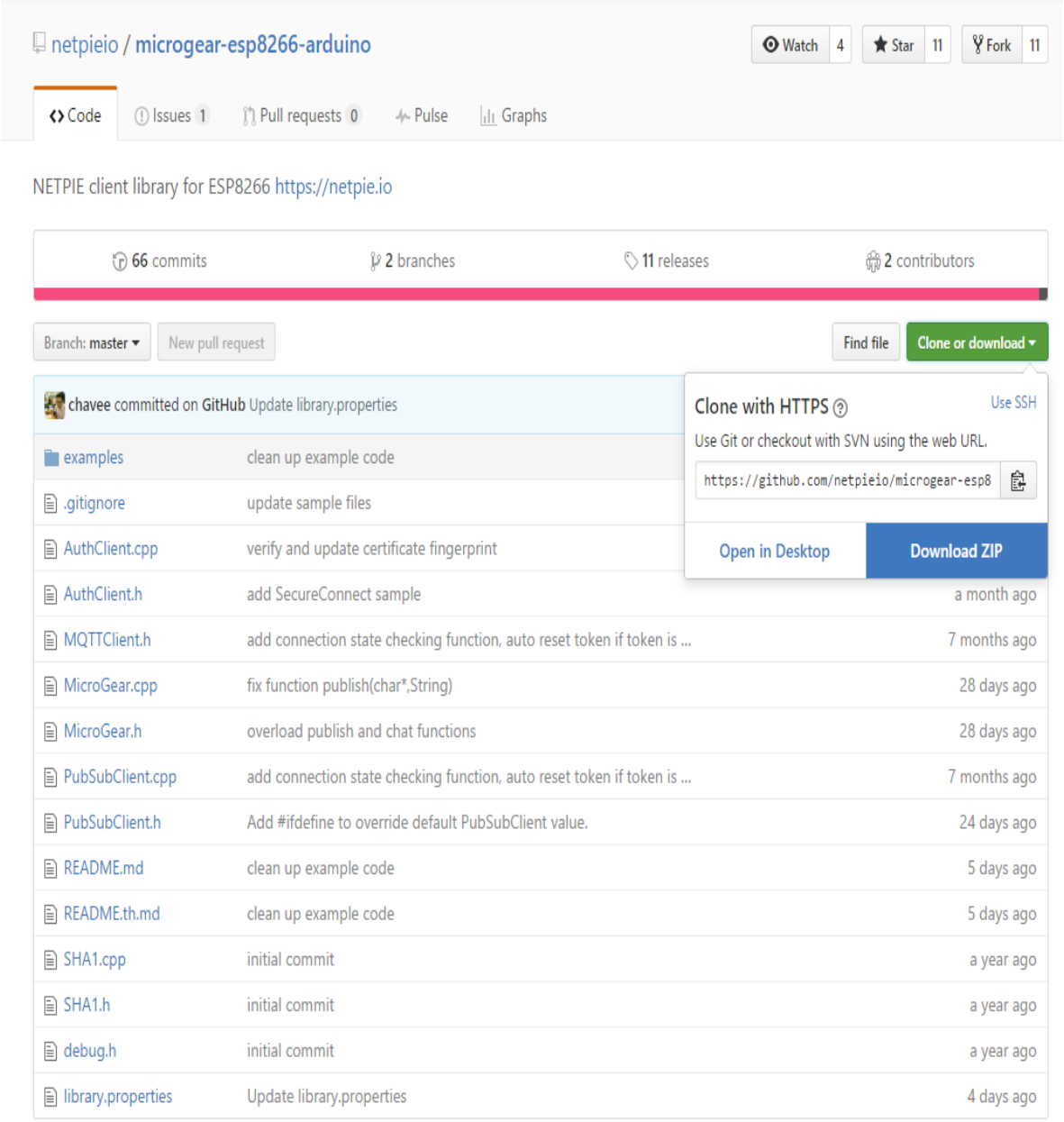
ค่าเหล่านี้จะถูกใช้เวลาเราเขียนโค้ดต่างๆทั้งในฝั่ง ESP8266, HTML5 หรือ Mobile Application

## 3. โปรแกรมArduino IDE กับ NETPIE

ก่อนที่เราจะใช้ NETPIE.io เราจำเป็นต้องที่มี Library ของ Microgear สำหรับ Arduino ก่อน จาก https://github.com/netpieio แล้วเลือก microgear-esp8266-arduino




**รูปที่ 3.1** หลังจากคลิกเข้าไปให้เลือก Clone or download ต่อด้วย Download ZIP

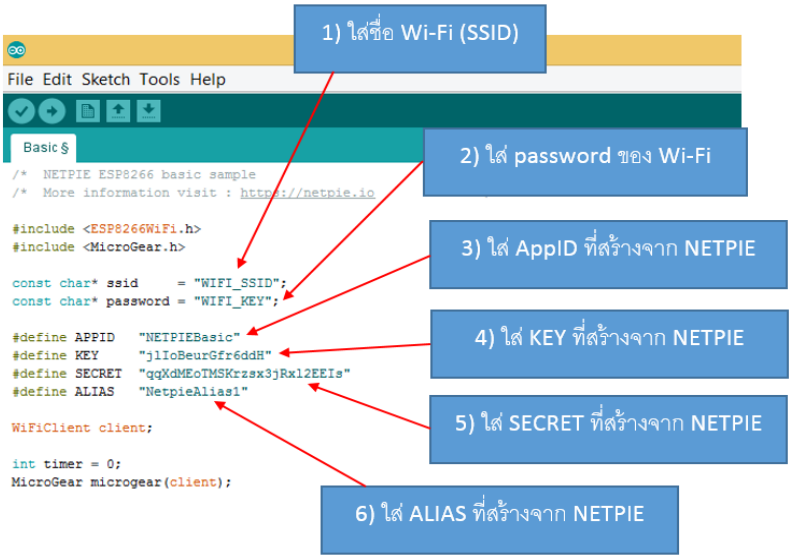



**รูปที่ 3.2** จากนั้นให้เราแตกไฟล์ microgear-esp8266-arduino-master ไปไว้ในตำแหน่ง libraries ของโปรแกรม Arduino IDE C:\Users\OS\Documents\Arduino\libraries

- หลังจากติดตั้ง microgear library เสร็จแล้ว เรามาลองเชื่อมต่อ ESP8266 กับ NETPIE.io
- ให้เราทำการนำ FTDI เชื่อมต่อกันกับ USB และ ESP8266 แล้วเปิด Arduino IDE
- จากนั้นให้เข้า Files > Examples > ESP8266 Microgear > Basic
- โค้ดที่โชว์ขึ้นมาคือโค้ดสำหรับทดลองต่อ NETPIE.io ให้เราใส่ค่าต่างๆ

```
Basic

/*  NETPIE ESP8266 basic sample                              */
/*  More information visit : https://netpie.io               */

#include <ESP8266WiFi.h>
#include <MicroGear.h>

const char* ssid     = <WIFI_SSID>;
const char* password = <WIFI_KEY>;

#define APPID   <APPID>
#define KEY     <APPKEY>
#define SECRET  <APPSECRET>
#define ALIAS   "esp8266"

WiFiClient client;

int timer = 0;
MicroGear microgear(client);

/* If a new message arrives, do this */
void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen) {
    Serial.print("Incoming message --> ");
    msg[msglen] = '\0';
    Serial.println((char *)msg);
}
```

**รูปที่ 3.3** ทำการใส่ข้อมูลต่างๆ




**รูปที่ 3.4** จากนั้นให้เรากด verify เพื่อตรวจความถูกต้องของโค้ด

- จากนั้นกด upload เพื่ออัพโค้ดไปยัง ESP8266เมื่อเราอัพโหลดเสร็จ ให้เปิด Serial Monitor ของ ESP8266
- กรณีไม่สำเร็จเช่น ESP8266 ทำการ reset ตัวเองอัตโนมัติ อาจจะเกิดจาก port ที่ใช้อยู่ไม่สามารถเชื่อมต่อกับได้หรือใส่ค่าต่างๆไม่ถูกต้อง

# ประวัติย่อผู้ทำโครงงาน

# ประวัติย่อผู้ทำโครงงาน

| | |
|---|---|
| ชื่อ ชื่อสกุล | นายทวีป ตรีหะจินดารัตน์ |
| วันเดือนปีเกิด | 4 เมษายน 2538 |
| สถานที่เกิด | อำเภอเมือง จังหวัดสระแก้ว |
| สถานที่อยู่ปัจจุบัน | 265 ถ.สุวรรณสร |
| | จ.สระแก้ว 27000 |
| หมายเลขโทรศัพท์ติดต่อ | 081-435-4095 |

ประวัติการศึกษา

| | |
|---|---|
| พ.ศ. 2556 | มัธยมศึกษาปีที่ 6 |
| | จากโรงเรียนมารีวิทยากบินทร์บุรี |
| พ.ศ. 2560 | กำลังศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า |
| | คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ |

# ประวัติย่อผู้ทำโครงงาน

| | |
|---|---|
| ชื่อ ชื่อสกุล | นายทศพร ปั้นจาด |
| วันเดือนปีเกิด | 14 กันยายน 2537 |
| สถานที่เกิด | อำเภอองครักษ์ จังหวัดนครนายก |
| สถานที่อยู่ปัจจุบัน | 250/111 หมู่ 1 ต.ลำผักกูด อ.ธัญบุรี |
| | จ.ปทุมธานี 12110 |
| หมายเลขโทรศัพท์ติดต่อ | 083-924-0802 |

ประวัติการศึกษา

| | |
|---|---|
| พ.ศ. 2556 | มัธยมศึกษาปีที่ 6 |
| | จากโรงเรียนธัญรัตน์ |
| พ.ศ. 2560 | กำลังศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า |
| | คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ |

# ประวัติย่อผู้ทำโครงงาน

| | |
|---|---|
| ชื่อ ชื่อสกุล | นายปวรัชฎ์ คชรินทร์ |
| วันเดือนปีเกิด | 18 กรกฎาคม 2537 |
| สถานที่เกิด | อำเภอเมือง จังหวัดระยอง |
| สถานที่อยู่ปัจจุบัน | 53 หมู่ 6 ต.ทับมา อ.เมือง |
| | จ.ระยอง 21000 |
| หมายเลขโทรศัพท์ติดต่อ | 081-411-2237 |

ประวัติการศึกษา

| | |
|---|---|
| พ.ศ. 2556 | มัธยมศึกษาปีที่ 6 |
| | จากโรงเรียนอัสสัมชัญระยอง |
| พ.ศ. 2560 | กำลังศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า |
| | คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ |