**บทที่ 1**

**บทนำ**

**1.1 ความเป็นมาและความสำคัญของปัญหา**

เนื่องจากปัจจุบันทางคณะเทคโนโลยีและการจัดการอุตสาหกรรมยังไม่มีระบบที่ช่วย  
ในการควบคุมการใช้ไฟฟ้า ทำให้ประสบปัญหา เช่น การเปิดไฟฟ้าทิ้งไว้ขณะที่ไม่มีการเรียนการสอนหรือไม่มีผู้ใช้ห้อง มีการใช้ไฟฟ้าหลังเวลาทำการโดยไม่ได้รับอนุญาต

ดังนั้นเพื่อเป็นการแก้ไขเบื้องต้น ทางผู้จัดทำจึงจะนำเทคโนโลยี Internet of Thing (IoT) ซึ่งเป็นเทคโนโลยีที่ได้รับความสนใจเป็นอย่างมากในปัจจุบัน โดยกล่าวคร่าว ๆ คือ Internet of Thing เป็นเทคโนโลยีอินเตอร์เน็ตที่เชื่อมอุปกรณ์และเครื่องมือต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ ตู้เย็น โทรทัศน์ และอื่น ๆ เข้าไว้ด้วยกัน โดยเครื่องมือต่าง ๆ จะสามารถเชื่อมโยงและสื่อสารกันได้โดยผ่านระบบอินเตอร์เน็ต โดยทางผู้จัดทำจะนำเทคโนโลยี Internet of Thing นี้มาช่วยในการตรวจสอบ เพื่อดูว่ามีห้องใดบ้างที่ตอนนี้กำลังใช้ไฟฟ้าอยู่บ้าง ทั้งในเวลาทำการและนอกเวลาทำการ และยังสามารถทำการควบคุมการใช้งานไฟฟ้าในห้องดังกล่าวนั้นได้ เช่น หากห้องดังกล่าวที่กำลังใช้ไฟฟ้าอยู่นั้นอยู่นอกเวลาทำการและไม่ได้ทำการขออนุญาตก็อาจจะทำการสั่งปิดได้ เป็นต้น

**1.2 วัตถุประสงค์ของการจัดทำโครงงานพิเศษ**

1.2.1 เพื่อเป็นแนวทางในการนำ Internet of Thing มาใช้ในการควบคุมระบบไฟฟ้าภายในตึกคณะเทคโนโลยีและการจัดการอุตสาหกรรม

1.2.2 เพื่อตรวจสอบว่าห้องไหนกำลังใช้ไฟฟ้านอกเวลาโดยที่ไม่ได้รับอนุญาต

1.2.3 เพื่อสะดวกต่อการควบคุมการใช้ไฟฟ้าของห้องเรียน

1.2.4 เพื่อช่วยประหยัดการใช้ไฟฟ้าของตึกคณะ

**1.3 ขอบเขตของการทำโครงงานพิเศษ ภาคการศึกษา 1/2558**

1.3.1 ศึกษาวิธีการเชื่อมต่อวงจรระหว่างบอร์ดกับเบรคเกอร์

1.3.2 จัดทำชุดบอร์ดทดลองจำนวน 2 ชุดเพื่อทำการเปรียบเทียบ โดยแบ่งเป็น 2 ระบบ คือ

1.3.2.1 เชื่อมต่ออินเตอร์เน็ตแบบแลนใช้ Arduino กับ Ethernet Shield

1.3.2.2 เชื่อมต่ออินเตอร์เน็ตแบบไร้สายโดยใช้ NodeMCU

1.3.3 ระบบสามารถแบ่งการทำงานหลัก ๆ ได้เป็น 2 ส่วน คือ

1.3.3.1 ตัวบอร์ด

1.3.3.1.1 Arduino

ก) เขียนโปรแกรมเพื่อตรวจสอบการเปิดปิดของไฟฟ้า

ข) เขียนโปรแกรมเพื่อควบคุมการเปิดปิดไฟฟ้า

ค) เขียนโปรแกรมเพื่อตรวจสอบการปริมาณการไหลของกระแสไฟฟ้า

1.3.3.1.2 Ethernet Shield เชื่อมต่อตัวบอร์ด Arduino กับ switch เพื่อให้เชื่อมต่อกับอินเตอร์เน็ต

1.3.3.1.3 Relay ต่อวงจรของระบบไฟเข้ากับตัวอุปกรณ์ Arduino เพื่อ  
รับคำสั่งในการควบคุมระบบไฟฟ้า

1.3.3.1.4 ACS712 ใช้วัดปริมาณกระแสไฟฟ้าที่ไหลเข้าอุปกรณ์

1.3.3.1.5 NodeMCU

ก) เขียนโปรแกรมเพื่อตรวจสอบการเปิดปิดของไฟฟ้า

ข) เขียนโปรแกรมเพื่อควบคุมการเปิดปิดไฟฟ้า

ค) เขียนโปรแกรมเพื่อตรวจสอบการปริมาณการไหลของกระแสไฟฟ้า

ง) เขียนโปรแกรมเพื่อเชื่อมต่อ Wi-Fi

1.3.3.2 หน้าเว็บ

1.3.3.2.1 สามารถแสดงสถานะทางการใช้ไฟฟ้าได้ โดยแบ่งสถานะของ การแสดงผลดังนี้

ก) เขียว : มีการใช้ไฟฟ้าในเวลา

ข) เทา : ไม่มีการใช้งาน

1.3.3.2.2 แสดงปริมาณไฟฟ้าที่ไหลผ่านอุปกรณ์

1.3.3.2.3 สามารถตั้งเวลาการปิดการใช้งานไฟฟ้าได้

1.3.3.2.4 สามารถสั่งควบคุมการเปิดปิดไฟฟ้าผ่านทางบอร์ด Arduino และ ESP8266 ได้

1.3.4 สามารถควบคุมการเปิด-ปิดไฟฟ้าผ่าน Internet ได้

**1.4 วิธีดำเนินการจัดทำโครงงานพิเศษ**

1.4.1 ศึกษาปัญหาและความเป็นไปได้ในการแก้ไขปัญหาของการที่จะนำเทคโนโลยี มาช่วยควบคุมการทำงานของระบบไฟฟ้าภายในตึก

1.4.2 ศึกษาและสืบค้นข้อมูลการใช้งานส่วนต่าง ๆ ของอุปกรณ์

1.4.3 ศึกษาวิธีการเชื่อมต่อวงจรระหว่างบอร์ดกับเบรคเกอร์

1.4.4 เขียนโปรแกรมเพื่อตรวจสอบการไหลของกระแสไฟฟ้าและเขียนโปรแกรมเพื่อควบคุม การเปิดปิดไฟฟ้าให้กับตัวบอร์ด Arduino และ NodeMCU

1.4.5 เพิ่ม Ethernet Shield เพื่อเชื่อมต่อตัวบอร์ด Arduino กับ Switch เพื่อให้เชื่อมต่อกับ อินเตอร์เน็ต

1.4.6 เพิ่ม ACS712 เพื่อใช้วัดปริมาณกระแสไฟฟ้าที่ไหลเข้าอุปกรณ์

1.4.7 เขียนหน้าเว็บเพื่อใช้สำหรับแสดงผลและควบคุมบอร์ด

1.4.7.1 แสดงสถานการณ์ใช้ไฟฟ้า โดยแบ่งเป็น

1.4.7.1.1 เขียว : มีการใช้ไฟฟ้าในเวลา

1.4.7.1.2 เทา : ไม่มีการใช้งาน

1.4.7.2 แสดงผลสรุประยะเวลาการใช้ไฟฟ้า

1.4.7.3 ควบคุมการตั้งเวลาการปิดไฟฟ้า

1.4.7.4 ควบคุมการเปิดปิดไฟฟ้า

1.4.8 ทำการทดสอบระบบโดยการทดลองกับระบบไฟฟ้าเพื่อเปรียบเทียบความแตกต่างระหว่างบอร์ด Arduino และ NodeMCU และปรับปรุงแก้ไข

1.4.9 ดำเนินงานและประเมินผล

1.4.10 จัดทำเอกสารประกอบขั้นตอนการทำงาน

**1.5 กิจกรรมและตารางเวลาในการจัดทำโครงงานพิเศษ**

**ตารางที่ 1-1** แสดงแผนกิจกรรมและตารางเวลาในการจัดทำภาคการศึกษาที่ 1/2558

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ขั้นตอนการดำเนินงาน | ภาคการศึกษาที่ 1/2558 | | | | | | | | | | | | | | | |
| สิงหาคม  1/2558 | | | | กันยายน  1/2558 | | | | ตุลาคม  1/2558 | | | | พฤศจิกายน  1/2558 | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1. ศึกษาปัญหาและความเป็นไปได้ในการแก้ไขปัญหาของการที่จะนำเทคโนโลยีมาช่วยควบคุมการทำงานของระบบไฟฟ้าภายในตึก |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. ศึกษาและสืบค้นข้อมูลการใช้งานส่วนต่างๆ ของอุปกรณ์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. ศึกษาวิธีการเชื่อมต่อวงจรระหว่างบอร์ดกับ เบรคเกอร์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. เขียนโปรแกรมเพื่อตรวจสอบการไหลของกระแสไฟฟ้าควบคุมการเปิดปิดไฟฟ้า |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. เขียนหน้าเว็บเพื่อใช้สำหรับแสดงผลและควบคุมบอร์ด |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6. ทำการทดสอบระบบโดยการทดลองกับระบบไฟฟ้าเพื่อเปรียบเทียบความแตกต่างระหว่างบอร์ด Arduino และ NodeMCU และปรับปรุงแก้ไข |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ตารางที่ 1-1 (ต่อ)** | | | | | | | | | | | | | | | | |
| ขั้นตอนการดำเนินงาน | ภาคการศึกษาที่ 1/2558 | | | | | | | | | | | | | | | |
| สิงหาคม  1/2558 | | | | กันยายน  1/2558 | | | | ตุลาคม  1/2558 | | | | พฤศจิกายน  1/2558 | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 7. ดำเนินงานและประเมินผล |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8. จัดทำเอกสารประกอบขั้นตอนการทำงาน |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**1.6 ทรัพยากรที่ต้องใช้ในการจัดทำโครงงานพิเศษ**

1. Hardware

- Arduino Mega R3 จำนวน 1 บอร์ด

- NodeMCU จำนวน 1 ตัว

- อุปกรณ์วัดกระแสไฟฟ้า Sensor ASC712 จำนวน 6 ตัว

- อุปกรณ์บอร์ดรีเลย์ จำนวน 1 บอร์ด

- หลอดไฟ จำนวน 6 หลอด

2. Software

- Sublime Text 2

- Arduino IDE

**1.7 ผลที่คาดว่าจะได้รับ**

1.7.1 สามารถควบคุมการทำงานของระบบไฟฟ้าภายในตึกในช่วงนอกเวลาทำการได้

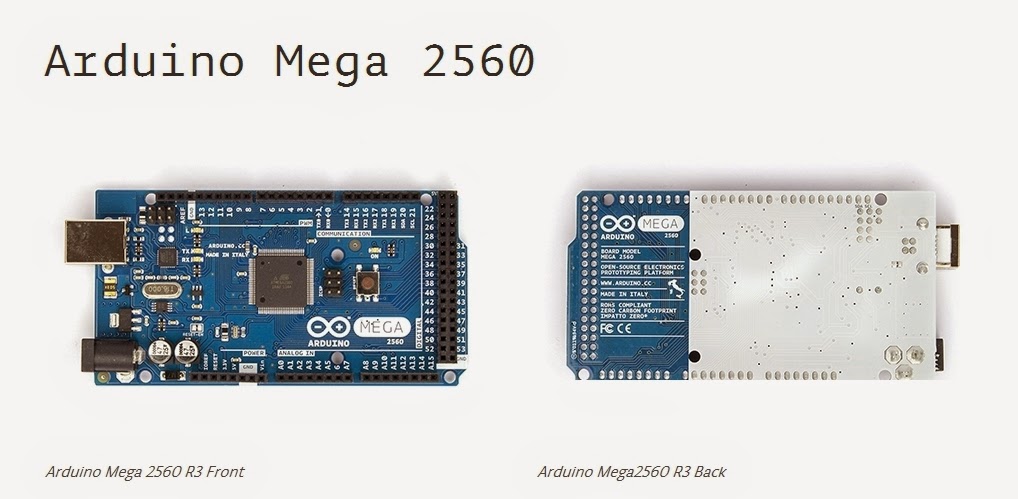
1.7.2 ช่วยประหยัดการใช้ไฟฟ้าที่เกินความจำเป็นได้

1.7.3 สามารถนำไปพัฒนาการใช้งานเพิ่มเติมในอนาคตได้

1.7.4 ใช้เป็นการทดลองและกรณีศึกษาในการใช้งาน Internet of Thing

**บทที่ 2**

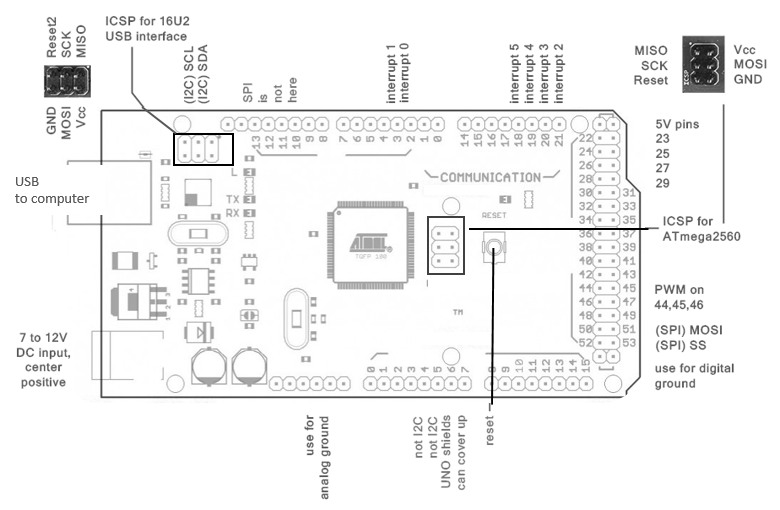
**ทฤษฎีที่เกี่ยวข้อง**

**2.1 Arduino**

**ภาพที่ 2-1** แสดงตัวอย่าง Arduino Mega 2560 R3

[Arduino](http://en.wikipedia.org/wiki/Arduino)เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด [Arduino](http://en.wikipedia.org/wiki/Arduino" \t "_blank) ถูกออกแบบมา  
ให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติม  
และพัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

        ความง่ายของบอร์ด [Arduino](http://en.wikipedia.org/wiki/Arduino" \t "_blank)ในการต่ออุปกรณ์เสริมต่าง ๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถ เลือกต่อกับบอร์ดเสริม ([Arduino Shield](http://www.thaieasyelec.com/products/development-boards/arduino/unofficial-shields.html" \t "_blank)) ประเภทต่าง ๆ เช่น [Arduino XBee Shield](http://www.thaieasyelec.com/products/development-boards/xbee-shield-v2-0-detail.html" \t "_blank), [Arduino Music Shield](http://www.thaieasyelec.com/products/development-boards/music-shield-v2-0-detail.html" \t "_blank), [Arduino Relay Shield](http://www.thaieasyelec.com/products/development-boards/relay-shield-v2-0-detail.html" \t "_blank), [Arduino Wireless Shield](http://thaieasyelec.com/products/development-boards/arduino/official-shields-made-in-italy/wireless-sd-shield-detail.html" \t "_blank) เป็นต้น มาเสียบเข้ากับบอร์ด [Arduino](http://en.wikipedia.org/wiki/Arduino)แล้วสามารถเขียนโปรแกรมพัฒนาต่อได้



**ภาพที่ 2-2** แสดงโครงสร้างบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3

**คุณสมบัติ Arduino Mega 2560 R3**

- Microcontroller ATmega2560

- ใช้แรงดันไฟฟ้า 5 V

- รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ) 7 – 12 V

- รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด) 6 – 20 V

- Digital I/O Pins 54

- Analog Input Pins 16

- กระแสไฟที่จ่ายได้ในแต่ละพอร์ต 40 mA

- กระแสไฟที่จ่ายได้ในพอร์ต 3.3V 50 mA

- พื้นที่โปรแกรมภายใน 256 KB

- SRAM 8 KB

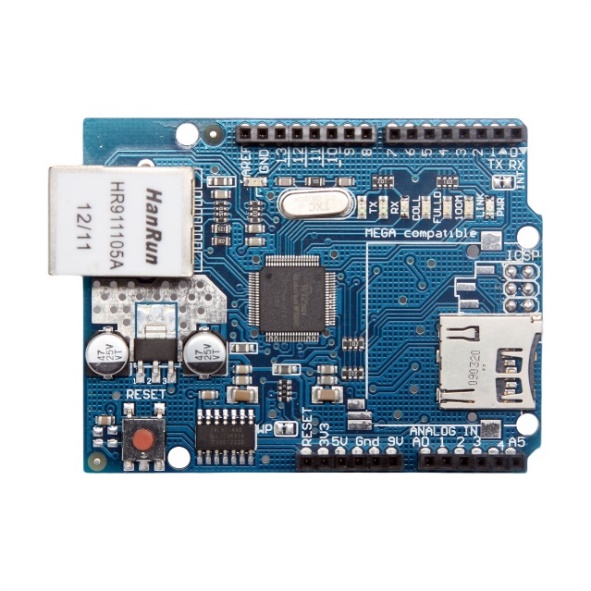
- พื้นที่หน่วยความจำถาวร (EEPROM) 4 KB

- Clock Speed 16 MHz

**การนำมาใช้กับระบบ**

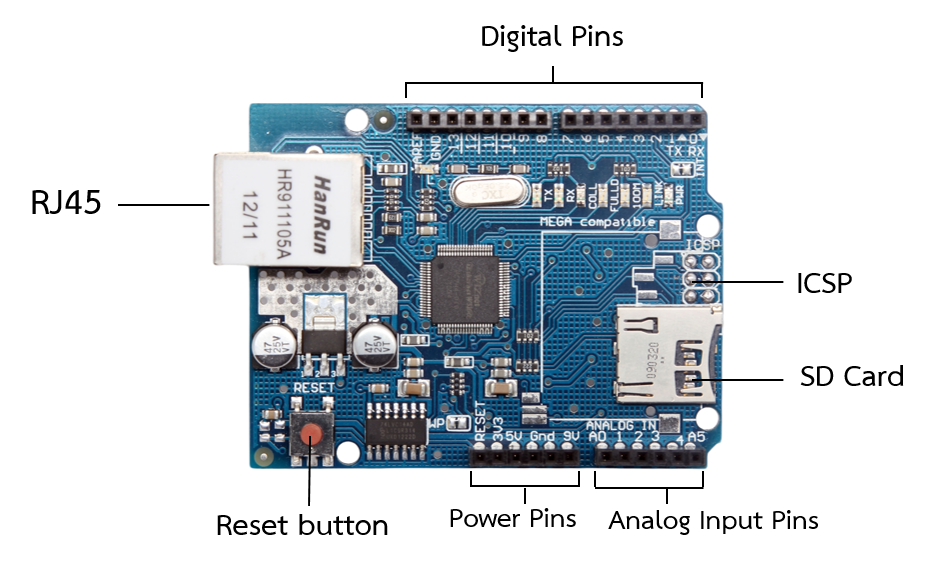
ใช้สำหรับควบคุมการทำงานของระบบ เช่น การควบคุมคำสั่งการเปิดปิดไฟฟ้าของรีเลย์

**2.2 Ethernet Shield**



**ภาพที่ 2-3** แสดงตัวอย่าง Ethernet Shield

Ethernet Shield นี้จะใช้ประกอบกับ Arduino UNO เพื่อที่จะทำให้สามารถติดต่อกับระบบเครือข่ายได้ โดยใช้ Ethernet Library ซึ่งเวอร์ชันล่าสุดจะมีช่องอ่าน Micro SD Card ติดมาด้วย สามารถใช้กับ SD Library ของ Arduino ได้ โดยการเชื่อมต่อกับ Ethernet Shield นี้จะใช้สาย RJ45 อาจจะใช้ CAT5 หรือ CAT6 โดยสามารถใช้ DHCP ได้ แต่การเชื่อมต่อระหว่างสองจุดยังต้อง  
ใช้สาย Cross Over อยู่ เพราะไม่มีวงจร Cross Over ภายใน ความเร็วในการสื่อสารของบอร์ดจะอยู่ที่ 10/50 Mbps หรือ 10/100 Mbps แล้วแต่แหล่งผลิต



**ภาพที่ 2-4** แสดงรายละเอียดของ Ethernet Shield 5100

**คุณสมบัติ Ethernet Shield W5100**

- ใช้ประกอบกับ Arduino เพื่อติดต่อกับระบบอินเทอร์เน็ตผ่านสายแลนได้

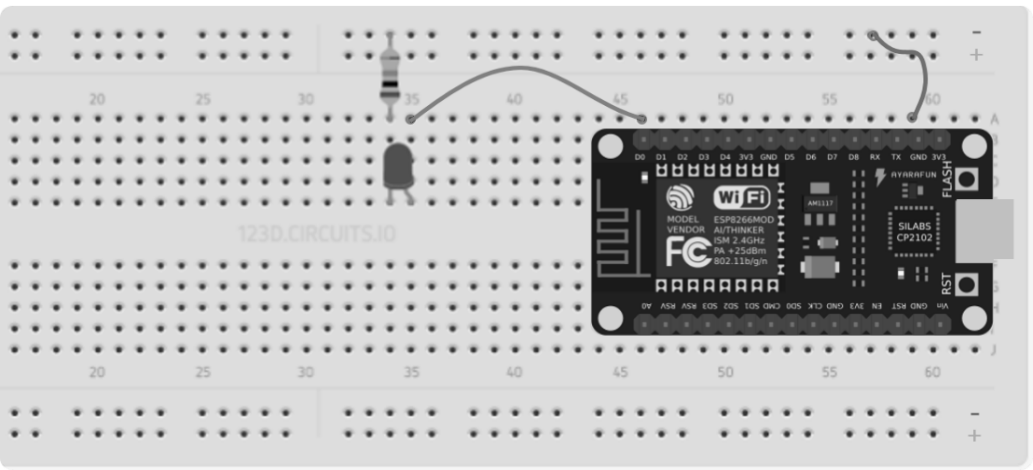
- รองรับ TCP/IP Protocol

- รองรับการเชื่อมต่อแบบ ADSL

**การนำมาใช้กับระบบ**

เชื่อมต่อกับกับตัวบอร์ด Arduino เพื่อที่จะทำให้สามารถติดต่อกับระบบเครือข่าย

**2.3 NodeMCU**



**ภาพที่ 2-5**  แสดงตัวอย่าง NodeMCU

โมดูล ESP8266 มาพร้อมกับเฟิร์มแวร์ที่ทำงานในลักษณะเป็น Serial-to-Wi-Fi ช่วยให้อุปกรณ์อื่นอย่างเช่น MCU สามารถต่อเข้ากับระบบอินเทอร์เน็ตผ่านระบบ Wi-Fi โดยสื่อสารผ่านพอร์ต Serial (ขา Tx และ Rx) และใช้ชุดคำสั่งแบบ AT ในการควบคุมการทำงาน ปัจจุบันมีการพัฒนา  
เฟิร์มแวร์ที่มีชื่อว่า NodeMCU และจะช่วยในนักพัฒนาสามารถเขียนโค้ดโดยใช้ภาษา Lua เพื่อควบคุมการทำงานของโมดูล ESP8266 ได้

**คุณสมบัติ NodeMCU**

- ใช้ชิป Flash ความจุ 32 Mbits (4MBytes)

- มีวงจรควบคุมแรงดัน 3.3 V (@800mA max.) บนบอร์ด

- สามารถจ่ายแรงดันไฟเลี้ยง +5 V จากภายนอกได้

- มีขา A0 รับอินพุตแรงดันแบบ Analog สำหรับวงจร ADC

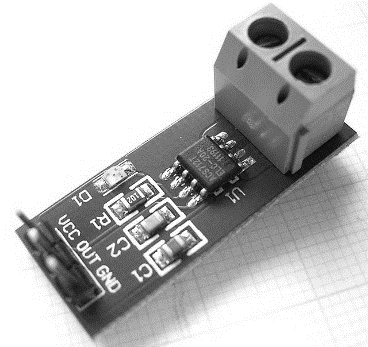
- ใช้ Connector แบบ Micro-USB สำหรับจ่ายแรงดันไฟฟ้าเลี้ยง (VUSB) เท่ากับ +5 V และสำหรับดาวน์โหลด Firmware

- มีขาสำหรับ SPI สำหรับต่อกับการ์ด SD

**การนำมาใช้กับระบบ**

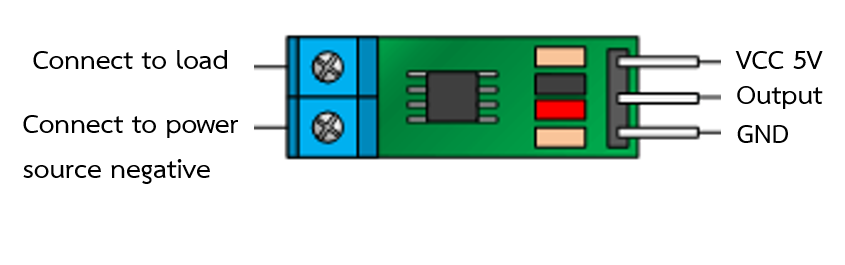
ใช้สำหรับควบคุมระบบไฟฟ้าผ่านทาง Wi-Fi

**2.4 Sensor วัดกระแสไฟฟ้า ACS712**



**ภาพที่ 2-6** แสดงตัวอย่าง Sensor ACS712

Current Sensor เป็นตัวตรวจจับสัญญาณให้กับไมโครคอนโทรลเลอร์ซึ่ง Current Sensor ที่ใช้เป็น ACS712 ซึ่งใช้หลักการของ Hall Effect Sensor ในการตรวจจับกระแสไฟฟ้า เมื่อวัดไฟฟ้ากระแสสลับ Output ที่ได้ออกมาจะเป็น VAC ที่วิ่งอยู่บน 2.50V DC ซึ่งจะใช้ Controller ในการตรวจจับค่า VAC ที่ได้แล้วแปลงกลับเป็นค่ากระแสไฟฟ้าที่วัดได้



**ภาพที่ 2-7**  แสดงรายละเอียด Sensor ACS712

**คุณสมบัติ Sensor วัดกระแสไฟฟ้า ACS712 – 30A**

- วัดกระแสไฟฟ้าได้ในย่าน -30 ถึง +40 A (วัดได้ทั้งกระแส AC และ DC)

- ใช้แหล่งจ่ายแรงดัน 5V

- Sensitivity 66 mV/A

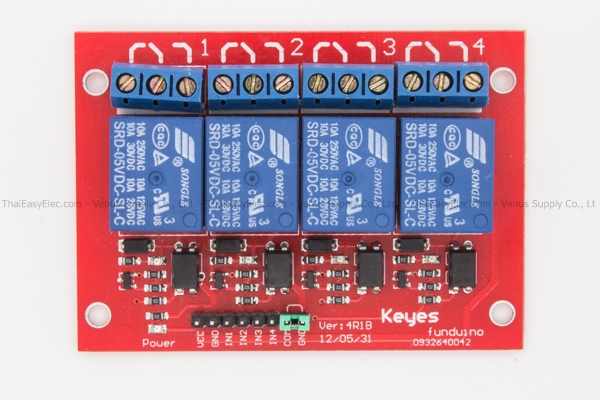
- 5 μs Output rise time in response to step input current

- 80 kHz Bandwidth

**การนำมาใช้กับระบบ**

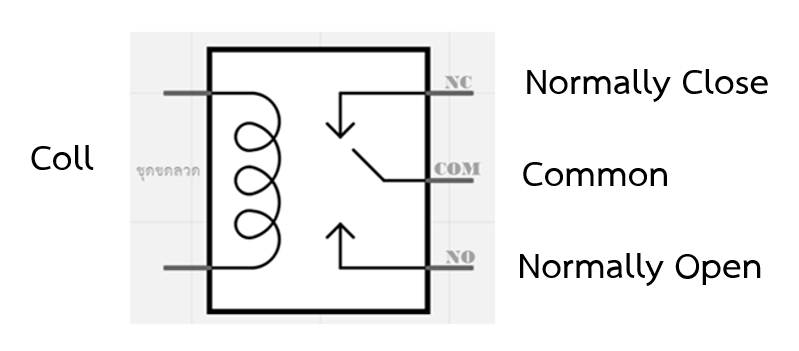
ใช้ในการวัดกระแสไฟฟ้า เพื่อที่จะนำมาเก็บข้อมูลทางสถิติใน Database

**2.5 รีเลย์ (Relay)**



**ภาพที่ 2-8** แสดงตัวอย่างรีเลย์

เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็กเพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย



**ภาพที่ 2-9** แสดงสัญลักษณ์ในวงจรไฟฟ้าของรีเลย์

**ภายใน**[Relay](http://thaieasyelec.com/products-components/relay.html)**จะประกอบไปด้วยขดลวดและหน้าสัมผัส**ดังนี้

1. **หน้าสัมผัส NC (Normally Close)**เป็นหน้าสัมผัสปกติปิด โดยในสภาวะปกติหน้าสัมผัสนี้จะต่อเข้ากับขา COM (Common) และจะลอยหรือไม่สัมผัสกันเมื่อมีกระแสไฟฟ้าไหลผ่านขดลวด

2**. หน้าสัมผัส NO (Normally Open)**เป็นหน้าสัมผัสปกติเปิด โดยในสภาวะปกติจะลอยอยู่ ไม่ถูกต่อกับขา COM (Common) แต่จะเชื่อมต่อกันเมื่อมีกระแสไฟไหลผ่านขดลวด  
        3. **ขา COM (Common)** เป็นขาที่ถูกใช้งานร่วมกันระหว่าง NC และ NO ขึ้นอยู่กับว่าขณะนั้นมีกระแสไฟฟ้าไหลผ่านขดลวดหรือไม่ หน้าสัมผัสใน Relay 1 ตัวอาจมีมากกว่า 1 ชุด ขึ้นอยู่กับผู้ผลิตและลักษณะของงานที่ถูกนำไปใช้ จำนวนหน้าสัมผัสถูกแบ่งออกดังนี้

- หน้าสัมผัสแบบ A (Form A) หมายถึง หน้าสัมผัสของ [Relay](http://thaieasyelec.com/products-components/relay.html) ในสภาพปกติจะเปิดอยู่ (Normally Open) และหน้าสัมผัสเป็นแบบ SPST ถ้าจะเขียนเป็นสัญลักษณ์ได้คือ

relay A

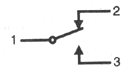
**ภาพที่ 2-10** แสดงหน้าสัมผัสแบบ A

- หน้าสัมผัสแบบ B (Form B) หมายถึง หน้าสัมผัสของ [Relay](http://thaieasyelec.com/products-components/relay.html) ในสภาพปกติจะปิด (Normally Close) และเป็นแบบ SPST เขียนเป็นสัญลักษณ์ได้คือ

relay B

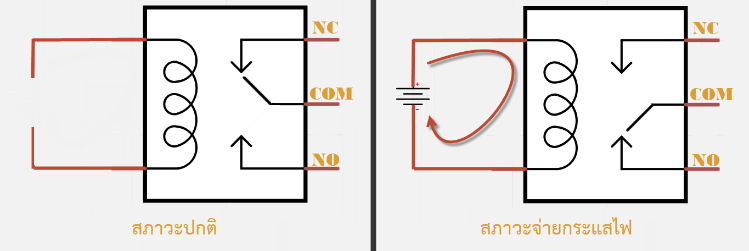
**ภาพที่ 2-11** แสดงหน้าสัมผัสแบบ B

- หน้าสัมผัสแบบ C (Form C) แบบนี้เรียกว่า "break, make หรือ transfer" เป็นหน้าสัมผัสแบบ SPDT เขียนสัญลักษณ์ได้ดังนี้

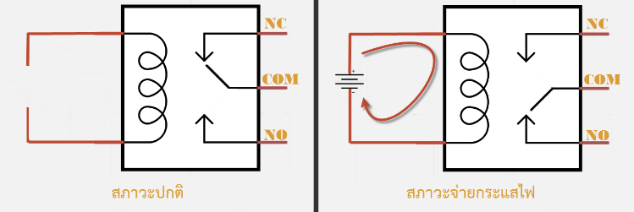


**ภาพที่ 2-12** แสดงหน้าสัมผัสแบบ C

หน้าสัมผัสแบบ C จะมีอยู่ด้วยกัน 3 ขา ในขณะที่ [Relay](http://thaieasyelec.com/products-components/relay.html) ยังไม่ทำงาน หน้าสัมผัส 1 และ 2 จะต่อกันอยู่ เมื่อ [Relay](http://thaieasyelec.com/products-components/relay.html) ทำงานหน้าสัมผัส 1 และ 2 จะแยกกัน จากนั้นหน้าสัมผัส 1 จะมาต่อกับหน้าสัมผัส 3 แทน เมื่อ [Relay](http://thaieasyelec.com/products-components/relay.html) หยุดทำงานหน้าสัมผัส 1 กับ 2 ก็จะกลับมาต่อกันตามเดิม



**ภาพที่ 2-13** แสดงRelay สภาวะปกติ



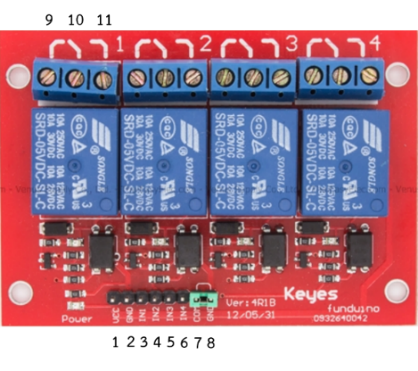
**ภาพที่ 2-14** แสดง Relay สภาวะจ่ายกระแสไฟ

ประเภทของรีเลย์ แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power Relay) หรือมักเรียกกันว่า คอนแทกเตอร์ (Contactor or Magnaticcontactor) ใช้ในการควบคุมไฟฟ้ากำลัง ซึ่งมีขนาดใหญ่กว่ารีเลย์ธรรมดา

2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุมบาง  
ที่เรียกกันง่าย ๆ ว่า "รีเลย์"

**ขาสัญญาณ (Pin Definition)**



**ภาพที่ 2-15** แสดงขาสัญญาณรีเลย์

จากภาพที่ 2-15 สามารถอธิบายรายละเอียดขาสัญญาณต่าง ๆ ที่ใช้ในการเชื่อมต่อของรีเลย์  
ได้ดังตารางที่ 2-1

**ตารางที่ 2-1** แสดงการอธิบายขาสัญญาณรีเลย์

|  |  |
| --- | --- |
| 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 |

**คุณสมบัติรีเลย์**

- ควบคุมไฟ DC ได้สูงสุด 30 VDC 10A และไฟ AC สูงสุด 250VAC 10A

- ระดับสัญญาณอินพุตควบคุมแบบ TTL ทำงานด้วยสัญญาณแบบ Active High

- มี OPTO-ISOLATED เพื่อแยก GND ส่วนสัญญาณควบคุมกับไฟที่ขับรีเลย์ออกจากกัน

- มี Jumper สำหรับเลือกว่าจะใช้ GND ร่วมหรือแยก

- มี LED แสดงสถานะการทำงานของรีเลย์และแสดงสถานะของบอร์ด

**การนำมาใช้กับระบบ**

ใช้ในการต่อวงจรของระบบไฟเข้ากับตัวอุปกรณ์ Arduino เพื่อรับคำสั่งในการตัดระบบไฟฟ้า

**2.6 กำลังไฟฟ้า**

กำลังไฟฟ้า หมายถึง พลังงานไฟฟ้าที่เครื่องใช้ไฟฟ้า ใช้ในเวลา 1 วินาที

**การคำนวณหาค่ากำลังไฟฟ้า**

กำลังไฟฟ้าของเครื่องใช้ไฟฟ้า คำนวณได้จากปริมาณกระแสไฟฟ้าที่ไหลผ่านเครื่องใช้ไฟฟ้า   
ถ้ามีกระไฟฟ้าไหลผ่านมากแสดงว่าเครื่องใช้ไฟฟ้านั้นใช้พลังงานไฟฟ้ามากแสดงว่าใช้กำลังไฟฟ้ามาก

P = VI

เมื่อ P = กำลังไฟฟ้า มีหน่วยเป็นวัตต์ (W)

V = ความต่างศักย์ มีหน่วยเป็นโวลต์ (V)

I = กระแสไฟฟ้า มีหน่วยเป็นแอมแปร์ (A)

**2.7 พลังงานไฟฟ้า**

พลังงานไฟฟ้า (Electrical Energy)  คือพลังงานที่ใช้ไปหรือสร้างขึ้นมาใหม่จากกำลังไฟฟ้า  
ที่ส่งเข้ามาหรือส่งออกไปโดยมีความสัมพันธ์กับเวลา มีหน่วยใช้แสดงพลังงานเป็นจุล (J) พลังงานไฟฟ้าใช้สัญลักษณ์ตัว "W" สามารถเขียนสมการได้ดังนี้

W = P x t

เมื่อ W  =  พลังงานไฟฟ้า  มีหน่วยเป็นจูล (J)  
                 P  =  กำลังไฟฟ้า       มีหน่วยเป็นวัตต์ (W)  
                 t   =  เวลา              มีหน่วยเป็นวินาที (s)

ไฟฟ้ากระแสสลับที่ถูกนำมาใช้งานในชีวิตประจำวัน เราต้องซื้อมาจากหน่วยงานที่ผลิตกระแสไฟฟ้าออกจำหน่าย เช่น การไฟฟ้าส่วนภูมิภาค และการไฟฟ้านครหลวง เป็นต้น พลังงานไฟฟ้าเหล่านี้มิได้ถูกคิดออกมาเป็นจูล (J) แต่จะคิดออกมาเป็นกิโลวัตต์-ชั่วโมง (Kilowatt-hour, kWh) ไม่ได้จัดเป็นหน่วย SI แต่มีความสัมพันธ์กับหน่วยระบบ SI โดยคิดค่ากำลังไฟฟ้าที่ใช้  
เป็นกิโลวัตต์ (kW) คิดในเวลาเป็นชั่วโมง (h) เขียนสมการออกมาได้ดังนี้

W(kWh)  =  P(kW)  x  t(h)

**การคำนวณหาจำนวนหน่วยไฟฟ้าที่ใช้**

ก่อนอื่นต้องทราบจำนวนเครื่องใช้ไฟฟ้าที่มีอยู่ในบ้านก่อนว่ามีจำนวนเท่าใดและเครื่องใช้ไฟฟ้าแต่ละชนิดกินไฟเท่าไร สามารถสังเกตได้จากคู่มือการใช้งานหรือแถบป้ายที่ติดกับเครื่องใช้ไฟฟ้า ที่เขียนว่า “กำลังไฟฟ้า” ซึ่งมีหน่วยเป็นวัตต์ (Watt) หลังจากนั้นลองคำนวณดูว่าเครื่องใช้ไฟฟ้าแต่ละชนิดที่ใช้งานในแต่ละวันกินไฟวันละกี่ยูนิต โดยสามารถคำนวณได้จากสูตรต่อไปนี้

จำนวนยูนิต = (กำลังไฟฟ้า x จำนวนเครื่องใช้ไฟฟ้า) x จำนวนชั่วโมง

1000

**ตัวเลขที่กำกับไว้บนเครื่องใช้ไฟฟ้า**

เครื่องใช้ไฟฟ้าแต่ละชนิดจะใช้พลังงานไฟฟ้าต่างกัน ทั้งนี้ขึ้นอยู่กับชนิดและขนาดของเครื่องใช้ไฟฟ้า ซึ่งทราบได้จากตัวเลขที่กำกับไว้บนเครื่องใช้ไฟฟ้าที่ระบุไว้ทั้งความต่างศักย์ (V) และกำลังไฟฟ้า (W)

**ตัวเลขที่กำหนดความต่างศักย์และกำลังไฟฟ้าบนเครื่องใช้ไฟฟ้า**

เครื่องใช้ไฟฟ้าแต่ละชนิด เช่น หลอดไฟฟ้า หม้อหุงข้าว เตารีดไฟฟ้า มีตัวเลขกำกับไว้ บนเครื่องใช้ไฟฟ้า เช่น หลอดไฟฟ้ามีตัวเลขกำกับไว้ว่า 220 V 60 W ตัวเลข 220 V หมายถึง หลอดไฟฟ้านี้ใช้กับความต่างศักย์ 220 โวลต์ ซึ่งเราต้องใช้ให้ตรงกับค่าความต่างศักย์ที่กำหนดมาส่วนตัวเลข 60 W หมายถึงค่าของพลังงานไฟฟ้าที่หลอดไฟฟ้าใช้ไปในเวลา 1 วินาที ซึ่งเรียกว่า “กำลังไฟฟ้า” การวัดพลังงานไฟฟ้าใช้หน่วยเป็นจูล (J) ตัวเลข 60 W จึงหมายความว่าขณะเปิดไฟ หลอดไฟฟ้านี้จะใช้พลังงานไฟฟ้า 60 จูล ในเวลา 1 วินาที

**กำลังไฟฟ้าที่ใช้ในเครื่องใช้ไฟฟ้า**

เครื่องใช้ไฟฟ้าแต่ละชนิดใช้กำลังไฟฟ้าต่างกันดังนี้

**ตารางที่ 2-2** แสดงตัวอย่างการใช้กำลังไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละชนิด

|  |  |
| --- | --- |
| **เครื่องใช้ไฟฟ้า** | **กำลังไฟฟ้า (วัตต์)** |
| เตารีดไฟฟ้า | 700 – 1,600 |
| หม้อหุงข้าวไฟฟ้า | 500 – 1,400 |
| ตู้เย็น | 70 – 260 |
| เครื่องปรับอากาศ | 1,150 ขึ้นไป |

**2.8 ภาษา C**

ภาษา C เป็นภาษาระดับสูง (High-Level-Language) และภาษาโปรแกรมที่นักเขียนโปรแกรมหรือที่เรียกว่าโปรแกรมเมอร์ นิยมใช้กันมาก เนื่องจากเป็นภาษาที่มีความเร็วในการทำงานสูงใกล้เคียงกับภาษาเครื่อง มีโครงสร้างที่ชัดเจน เข้าใจง่าย สามารถเขียนโปรแกรมเพื่อติดต่อกับฮาร์ดแวร์ ของเครื่องคอมพิวเตอร์ได้อย่างดี ภาษา C เกิดขึ้นในปี ค.ศ.1972 ผู้คิดค้นคือนายเดนนีส ริทชี (Dennis Ritchi) การศึกษาภาษา C ถือว่าเป็นพื้นฐานในการศึกษาภาษาใหม่ ๆ ได้

ภาษา C เป็นภาษาที่ใช้สำหรับเขียนโปรแกรมประยุกต์ต่าง ๆ เช่นเดียวกับ ภาษาปาสคาล ภาษาเบสิก และภาษาฟอร์แทรน เป็นต้น นอกจากนี้ภาษา C ยังใช้สำหรับเขียนโปรแกรมระบบและโปรแกรมสำหรับควบคุมฮาร์ดแวร์บางส่วนที่ภาษาโปรแกรมระดับสูงหลายภาษาไม่สามารถทำได้ ภาษา C จึงจัดเป็นภาษาระดับกลางด้วย

**ข้อดีของภาษา C**

เป็นภาษาที่มีลักษณะเป็นโครงสร้างจึงเขียนโปรแกรมง่าย โปรแกรมที่เขียนขึ้นจะทำงานได้อย่าง มีประสิทธิภาพสูง สั่งงานคอมพิวเตอร์ได้รวดเร็วกว่าภาษาระดับสูงอื่น ๆ สั่งงานอุปกรณ์ในระบบคอมพิวเตอร์ได้เกือบทุกส่วนของฮาร์ดแวร์ ซึ่งภาษาระดับสูงภาษาอื่นทำงานดังกล่าวได้น้อยกว่า คอมไพเลอร์ภาษา C ทุกโปรแกรมในท้องตลาดจะทำงานอ้างอิงมาตรฐาน (ANSI = American National Standards Institute) เกือบทั้งหมด จึงทำให้โปรแกรมที่เขียนขึ้นด้วยภาษา C สามารถนำไปใช้กับคอมพิวเตอร์ได้ทุกรุ่นที่มาตรฐาน ANSI รับรอง โปรแกรมที่เขียนขึ้นด้วยภาษา C สามารถนำไปใช้กับเครื่องคอมพิวเตอร์ที่ใช้ CPU ต่างเบอร์กันได้ หรือกล่าวได้ว่าโปรแกรมมีความยืดหยุ่น (Portability) สูง

สามารถนำภาษา C ไปใช้ในการเขียนโปรแกรมประยุกต์ได้หลายระดับ เช่น เขียนโปรแกรม จัดระบบงาน (OS) คอมไพเลอร์ของภาษาอื่น โปรแกรมสื่อสารข้อมูลโปรแกรมจัดฐานข้อมูล โปรแกรมปัญญาประดิษฐ์ (AI = Artificial Intelligent) รวมทั้งโปรแกรมคำนวณงานทางด้านวิทยาศาสตร์และวิศวกรรมศาสตร์ เป็นต้น มีโปรแกรมช่วย (tool box) ที่ช่วยในการเขียนโปรแกรมมากและราคาไม่แพงหาซื้อได้ง่าย สามารถประกาศข้อมูลได้หลายชนิดและหลายรูปแบบ ทำให้สะดวกรวดเร็วต่อการพัฒนาโปรแกรมตามวัตถุประสงค์ของผู้ใช้ ประยุกต์ใช้ในงานสื่อสารข้อมูล   
และงานควบคุมที่ต้องการความแม่นยำในเรื่องเวลาได้ดีกว่าภาษาระดับสูงอื่น ๆ หลาย ๆ ภาษา

**ข้อเสียของภาษา C**

ภาษา C ไม่มีตัวจัดการจองหน่วยความจำในตัวเอง เมื่อต้องการจองหน่วยความจำแบบ Dynamic ภาษา C ทำ Wrapper เพื่อติดต่อกับระบบปฏิบัติการเพื่อขอจองหน่วยความจำโดยตรง ปัญหาก็คือ การติดต่อกันระหว่างโปรแกรมของเรากับระบบปฏิบัติการเป็นไปอย่างหลวม ๆ   
ถ้าโปรแกรมลืมบอกระบบปฏิบัติการว่าเลิกจองหน่วยความจำดังกล่าว หน่วยความจำนั้นก็จะถูกจองไปเรื่อย ๆ สาเหตุหลักของปัญหานี้คือสิ่งที่เรียกว่าหน่วยความจำรั่ว หรือ Memory Leak

**ภาษา C กับ Arduino**

เมื่อเริ่มต้นทำงาน Arduino จะทำตามคำสั่งต่าง ๆ ที่อยู่ในฟังก์ชัน Setup เป็นจำนวน 1 รอบ โดยคำสั่งต่าง ๆ ที่จะเขียนในฟังก์ชันนี้ ส่วนมากจะเป็นการกำหนดค่าเริ่มต้น การกำหนดหน้าที่ ของแต่ละขา หรือคำสั่งต่าง ๆ ที่ต้องการเรียกใช้เพียงแค่ครั้งแรกครั้งเดียว หลังจากที่จบฟังก์ชัน Setup จะไม่มีการย้อนกลับมาทำคำสั่งในนี้อีก ส่วนฟังก์ชัน Loop จะทำงานต่อจาก Setup   
โดยใน Loop นี้ จะเป็นการทำตามคำสั่งแบบวนซ้ำ คือ ทำงานตามคำสั่งบรรทัดแรกไปเรื่อย ๆ จนถึงบรรทัดสุดท้าย แล้ววนกลับมาเริ่มทำที่บรรทัดแรกใหม่อีกครั้ง เป็นเช่นนี้ไปเรื่อย ๆ

**2.9 ภาษา HTML**

HTML หรือ Hyper Text Markup Language เป็นภาษาคอมพิวเตอร์รูปแบบหนึ่ง ที่มีโครงสร้างการเขียนโดยอาศัยตัวกำกับ (Tag) ควบคุมการแสดงผลข้อความ, รูปภาพ หรือวัตถุอื่น ๆ   
ผ่านโปรแกรมบราวเซอร์ แต่ละ Tag อาจจะมีส่วนขยายที่เรียกว่า Attribute สำหรับระบุ หรือควบคุมการแสดงผลของเว็บได้ด้วย HTML เป็นภาษาที่ถูกพัฒนาโดย World Wide Web Consortium (W3C) จากแม่แบบของภาษา SGML (Standard Generalized Markup Language) โดยตัดความสามารถบางส่วนออกไป เพื่อให้สามารถทำความเข้าใจและเรียนรู้ได้ง่าย และด้วยประเด็นดังกล่าว ทำให้บริการ WWW เติบโตขยายตัวอย่างกว้างขวางตามไปด้วย Tag

**บทที่ 3**

**ออกแบบและพัฒนา**

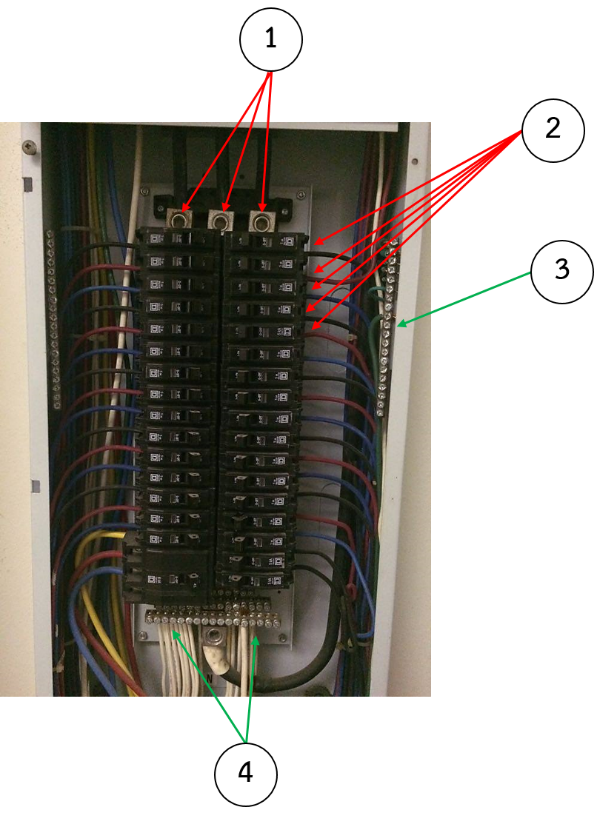
แนวคิดเกี่ยวกับการออกแบบระบบควบคุมไฟฟ้าด้วยเทคโนโลยี IoT นี้ เนื่องมาจากการ  
ต้องการทราบและจัดการเวลาการใช้ไฟฟ้าในแต่ละห้องเรียนของทางคณะเทคโนโลยี  
และการจัดการอุตสาหกรรม จึงมีแนวคิดในการประดิษฐ์เครื่องควบคุมการใช้ไฟฟ้าขึ้น โดยจะอาศัยไมโครคอนโทรลเลอร์มาใช้ในการควบคุมการทำงานในการควบคุมการเปิดปิดไฟฟ้าและส่วนต่าง ๆ เพราะไมโครคอนโทรลเลอร์มีกลไกที่รวบรวมการทำงานต่าง ๆ ไว้ภายในตัวเอง มีขนาดเล็ก  
และสามารถเขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์ต่าง ๆ ที่เชื่อมต่อกับตัวมันได้   
ซึ่งทำให้ง่ายต่อการนำไปใช้งานหรือแก้ไขในอนาคต

ในระบบควบคุมไฟฟ้าด้วยเทคโนโลยี IoT นี้ จำเป็นต้องมีการวางแผนการทำงานให้ทราบถึง การทำงานของระบบ โดยทำการศึกษาหาความรู้เกี่ยวกับอุปกรณ์ที่จะนำมาใช้งาน หลักการ  
และรายละเอียดต่าง ๆ ที่เกี่ยวข้อง เพื่อให้สามารถดำเนินการพัฒนาระบบให้บรรลุตามวัตถุประสงค์ มากที่สุด

**3.1 ภาพรวมของระบบ**

ระบบควบคุมไฟฟ้าด้วยเทคโนโลยี IoT เป็นระบบที่ใช้สำหรับควบคุมการใช้ไฟฟ้าผ่านอินเตอร์เน็ตโดยใช้ไมโครคอนโทรลเลอร์ในการรับคำสั่งจากหน้าเว็บบราวเซอร์แล้วดำเนินการทำงานต่าง ๆ ของระบบตามคำสั่งที่ได้รับ โดยระบบควบคุมไฟฟ้าด้วยเทคโนโลยี IoT นี้สามารถตรวจสอบได้ว่ามีการใช้ไฟฟ้าอยู่หรือไม่ และสามารถควบคุมการใช้งานของไฟฟ้าของอุปกรณ์ที่เชื่อมต่ออยู่รวมถึงสามารถตรวจสอบกระแสไฟฟ้าที่ใช้งานอยู่ได้

**3.2 การศึกษาค้นคว้าเกี่ยวกับเบรกเกอร์**



**ภาพที่ 3-1** แสดงตัวอย่างเบรกเกอร์ของระบบปัจจุบัน

จากภาพที่ 3-1 เป็นภาพของวงจรเบรกเกอร์ในปัจจุบัน ซึ่งสามารถอธิบายส่วนต่าง ๆ ได้ ดังนี้

1. Main : สายไฟหลักที่ต่อจากแหล่งกำเนิดไฟฟ้าเข้ากับเบรกเกอร์

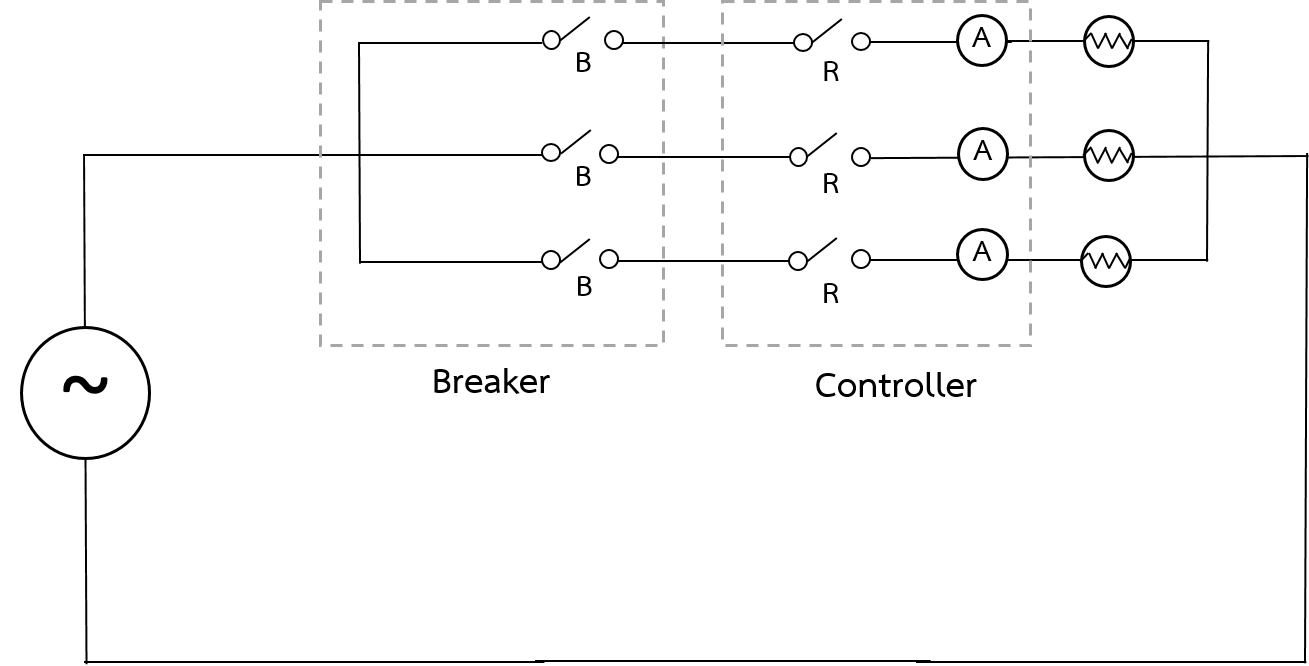
2. Output : สายไฟที่นำไฟฟ้าออกจากเบรกเกอร์เพื่อไปยังเครื่องใช้ไฟฟ้า

3. Ground : สายไฟที่นำกระแสไฟฟ้าที่รั่วออกมาจากเบรกเกอร์ลงดิน เพื่อป้องกันไฟรั่ว

4. Neutal : สายไฟที่ไม่มีไฟฟ้า โดยจะต่อออกมาจากเครื่องใช้ไฟฟ้ามายังเบรกเกอร์



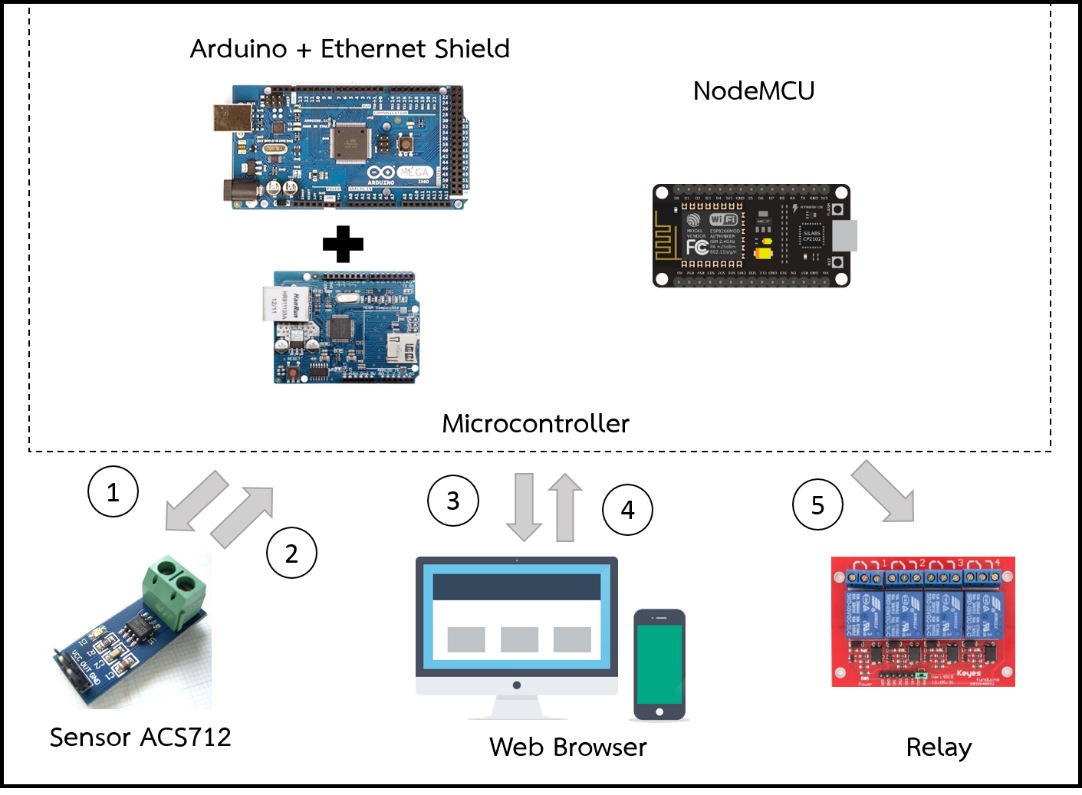
**ภาพที่ 3-2** แสดงตัวอย่างวงจรเบรกเกอร์ของระบบปัจจุบัน



**ภาพที่ 3-3** แสดงตัวอย่างวงจรเบรกเกอร์ของระบบใหม่

จากภาพที่ 3-2 จะแสดงภาพวงจรเบรกเกอร์ของระบบที่ใช้อยู่ในปัจจุบัน และภาพที่ 3-3   
แสดงภาพเบรกเกอร์ของระบบที่จะทำการดัดแปลง โดยในระบบใหม่นั้นจะทำการเพิ่มส่วนของไมโครคอนโทรลเลอร์เพื่อทำการควบคุมระบบโดยผู้ใช้จะการสั่งการจากหน้าเว็บบราวเซอร์ไปยัง บอร์ดไมโครคอนโทรลเลอร์เพื่อทำการควบคุมการใช้ไฟฟ้าต่อไป

**3.3 เชื่อมต่ออุปกรณ์ทุกอย่างเข้าด้วยกันและขั้นตอนการทำงานในส่วนต่าง ๆ ของระบบ**



**ภาพที่ 3-4** แสดงขั้นตอนการทำงานของระบบ

จากภาพที่ 3-4 สามารถอธิบายเกี่ยวกับขั้นตอนการทำงานของระบบได้ดังนี้

1. บอร์ดไมโครคอนโทรลเลอร์ติดต่อกับ Sensor ACS712 เพื่อสั่งให้เริ่มการทำงานในการ วัดค่ากระแสไฟฟ้า

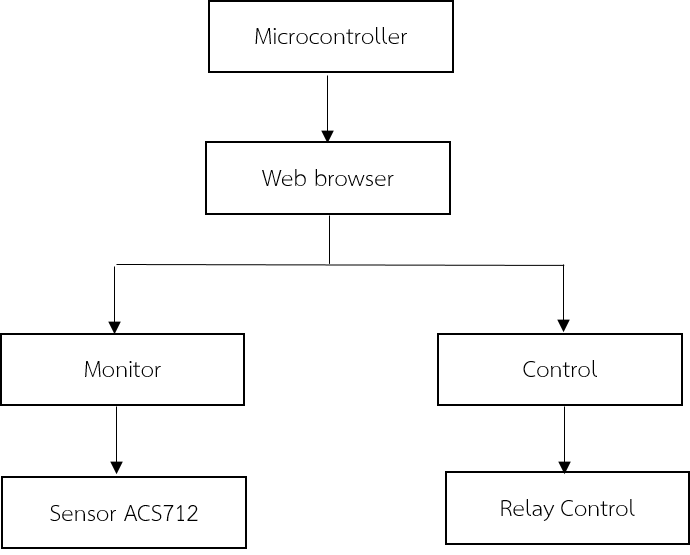
2. บอร์ดไมโครคอนโทรลเลอร์รับค่ากระแสไฟฟ้าที่วัดได้จาก Sensor ACS712

3. บอร์ดไมโครคอนโทรลเลอร์ติดต่อกับกับหน้าเว็บบราวเซอร์เพื่อแสดงสถานะการใช้งานไฟฟ้าและค่ากระแสไฟฟ้าที่วัดได้จากตัว Sensor ACS712

4. บอร์ดไมโครคอนโทรลเลอร์รับคำสั่งการควบคุมการเปิดปิดไฟฟ้าจากหน้าเว็บบราวเซอร์

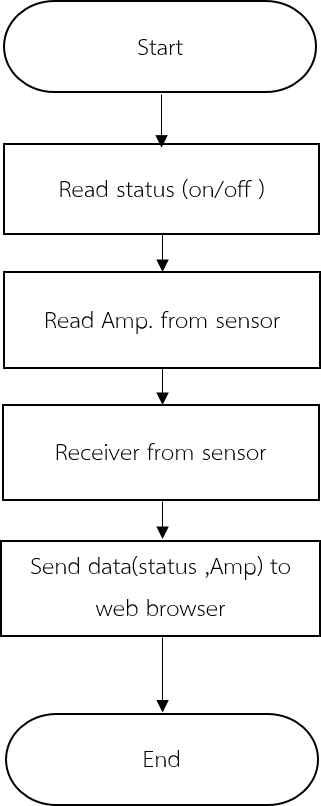
5. บอร์ดไมโครคอนโทรลเลอร์ติดต่อกับ Relay เพื่อควบคุมการจ่ายไฟฟ้าให้กับอุปกรณ์

**3.4 แผนภาพ Flowchart และ Diagram แสดงขั้นตอนกระบวนการทำงานของระบบ**



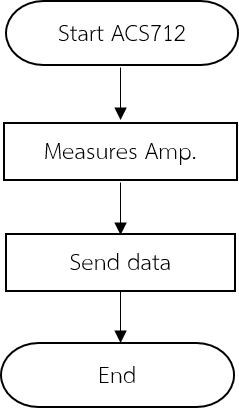
**ภาพที่ 3-5** แสดงลำดับขั้นตอนการทำงานของระบบโดยรวม

จากภาพที่ 3-5 แสดงลำดับขั้นตอนการทำงานของระบบ ซึ่งสามารถอธิบายได้ดังนี้ เริ่มจาก การจ่ายไฟให้กับตัวบอร์ดเพื่อเริ่มการทำงานของระบบ จากนั้นเปิดใช้งานหน้าเว็บบราวเซอร์  
โดยการระบุ IP Address ของตัวบอร์ด โดยหน้าเว็บบราวเซอร์จะมีการแสดงสถานการณ์ใช้งานไฟฟ้า  
ที่มอนิเตอร์อยู่และสามารถควบคุมได้ด้วยว่ามีจะเปิดหรือปิดไฟฟ้านั้น ๆ รวมถึงแสดงค่ากระแสไฟฟ้า  
ที่วัดได้จาก Sensor ACS712



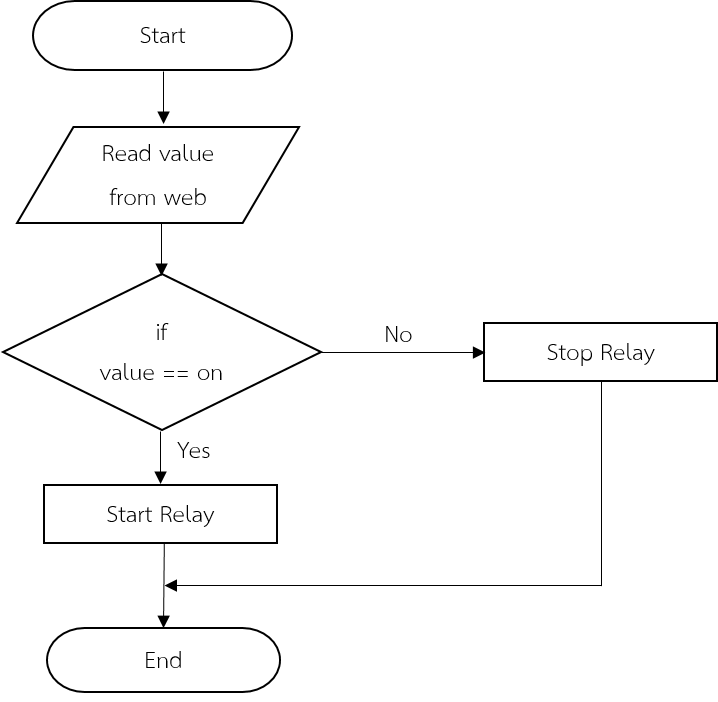
**ภาพที่ 3-6** แสดง Flowchart การเริ่มต้นการทำงานของระบบ

จากภาพที่ 3-6 เป็นกระบวนการการเริ่มต้นการทำงานของระบบ โดยเริ่มจากการตรวจสอบสถานะของไฟฟ้าที่ต้องการจะตรวจสอบว่าเปิดหรือปิดอยู่ และเริ่มอ่านค่าจาก Sensor ACS712 ที่เชื่อมต่อเข้ากับตัวบอร์ด และเริ่มการทำงานของ Web Server ถัดมาเมื่อทำการเปิดหน้า   
Web Browser จะเป็นการทำงานในส่วนของ Web Client ในการร้องขอข้อมูลค่าที่ได้รับจาก Sensor มาแสดงผลและสามารถควบคุมอุปกรณ์ผ่านทางหน้าเว็บบราวเซอร์



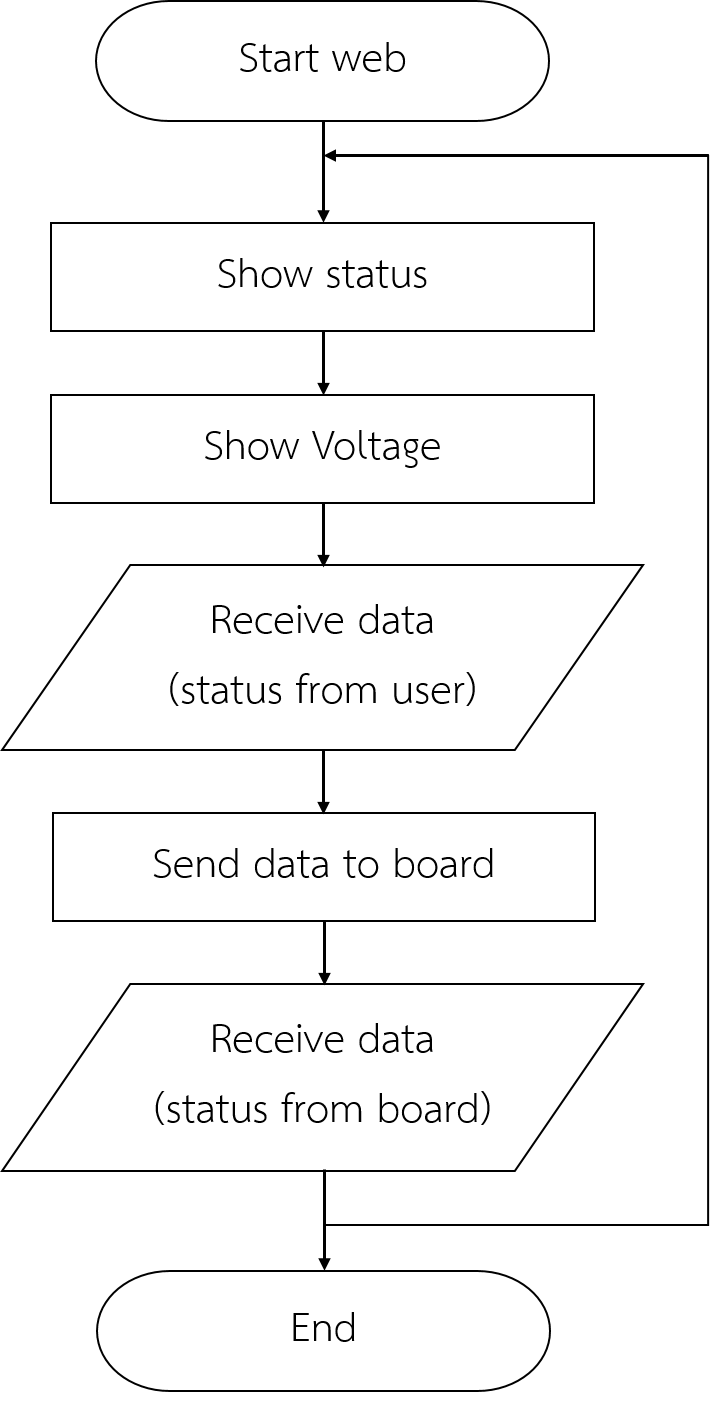
**ภาพที่ 3-7** แสดง Flow Chart การทำงานของ Sensor ACS712

จากภาพที่ 3-7 เป็นการแสดงการทำงานของ Sensor ACS712 โดยจะเริ่มต้นเมื่อมีอุปกรณ์  
มาต่อเข้ากับระบบเพื่อใช้งานกระแสไฟฟ้า Sensor จะทำการวัดค่าของกระแสไฟฟ้าที่อุปกรณ์นั้น ๆ ใช้งานไป และทำการส่งข้อมูลไปให้กับอุปกรณ์ไมโครคอนโทรลเลอร์เพื่อทำการหาค่ากำลังไฟฟ้า



**ภาพที่ 3-8** แสดง Flowchart การทำงานของ Relay

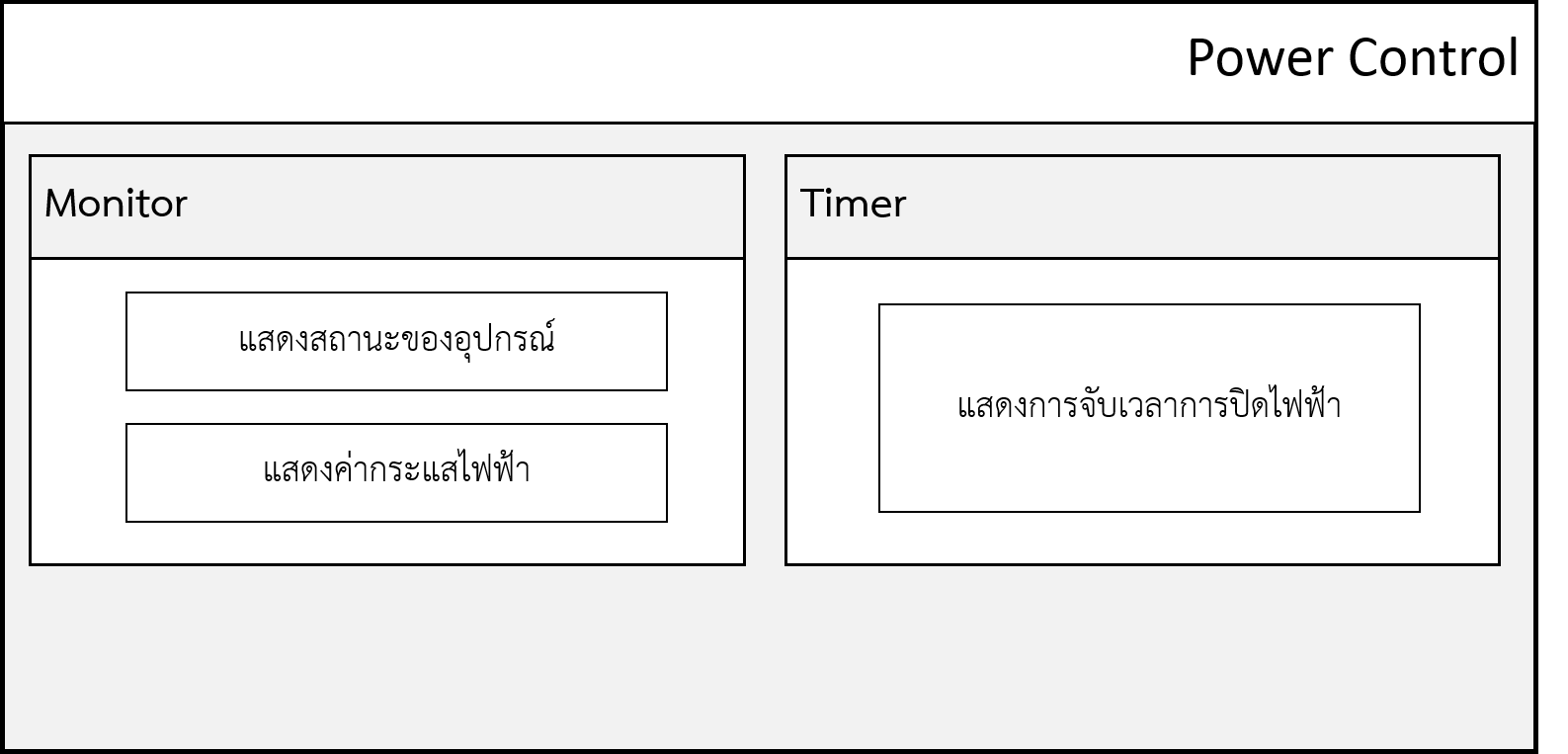
ภาพที่ 3-8 เป็นการแสดงการทำงานของ Relay จะ Start bit Port และ Data Port   
เพื่อทำการติดต่อรับข้อมูล เมื่อมีการรับข้อมูลก็จะทำการเปลี่ยนสถานะของอุปกรณ์ให้เป็นตาม  
ที่ต้องการ แล้วส่งกลับข้อมูลของอุปกรณ์ว่าได้เปลี่ยนสถานะใดไปแล้วบ้าง และทำการรอรับคำสั่งต่อไป โดยถ้าไม่มีการติดต่อจะเป็นการ Stop thread แต่ถ้ามีการเปลี่ยนสถานะก็จะเปลี่ยนสถานะตามอุปกรณ์ที่ส่งข้อมูลมาและเก็บข้อมูลแล้วกลับไปรอรับการทำงานต่อไป



**ภาพที่ 3-9** แสดง Flowchart การทำงานของหน้าเว็บ

ภาพที่ 3-9 เป็นการแสดงการทำงานของหน้าเว็บบราวเซอร์ โดยเริ่มจากการเปิดหน้าเว็บขึ้นมาผ่านทาง IP Address ของบอร์ดไมโครคอนโทรลเลอร์ โดยเริ่มแรกจะมีการแสดงสถานะของอุปกรณ์ไฟฟ้าที่เชื่อมต่ออยู่กับระบบและแสดงค่ากระแสไฟฟ้าที่วัดได้จาก Sensor ACS712 จากนั้น  
เมื่อมีการสั่งการควบคุม เช่น สั่งให้เปิดไฟโดยการกดปุ่มจากบนหน้าเว็บ ทางหน้าเว็บก็จะทำการ  
ส่งค่าที่ได้ไปที่บอร์ดไมโครคอนโทรลเลอร์เพื่อทำการควบคุมระบบ หลังจากที่บอร์ดได้รับคำสั่ง  
การควบคุมไปแล้ว ก็จะทำการประมวลผลและส่งค่าสถานะใหม่กลับมาที่หน้าเว็บ

**3.5 การออกแบบหน้าจอ**



**ภาพที่ 3-10** แสดงหน้าจอของหน้าเว็บบราวเซอร์

จากภาพที่ 3-10 สามารถอธิบายรายละเอียดได้ว่าเมื่อเข้าหน้าเว็บบราวเซอร์มา ทางด้านซ้ายจะเป็นการแสดงสถานะของอุปกรณ์ไฟฟ้าที่เชื่อมต่ออยู่กับระบบว่ามีการใช้งานอยู่หรือไม่   
และเป็นส่วนที่ใช้สำหรับควบคุมการเปิดปิดไฟ รวมถึงแสดงค่าปริมาณกระแสไฟฟ้าที่ไหลผ่านที่อ่านได้จาก Sensor ACS712 ทางด้านขวาจะเป็นการควบคุมโดยตั้งเวลาถอยหลังเพื่อปิดไฟฟ้า

**บทที่ 4**

**ผลการดำเนินงาน**

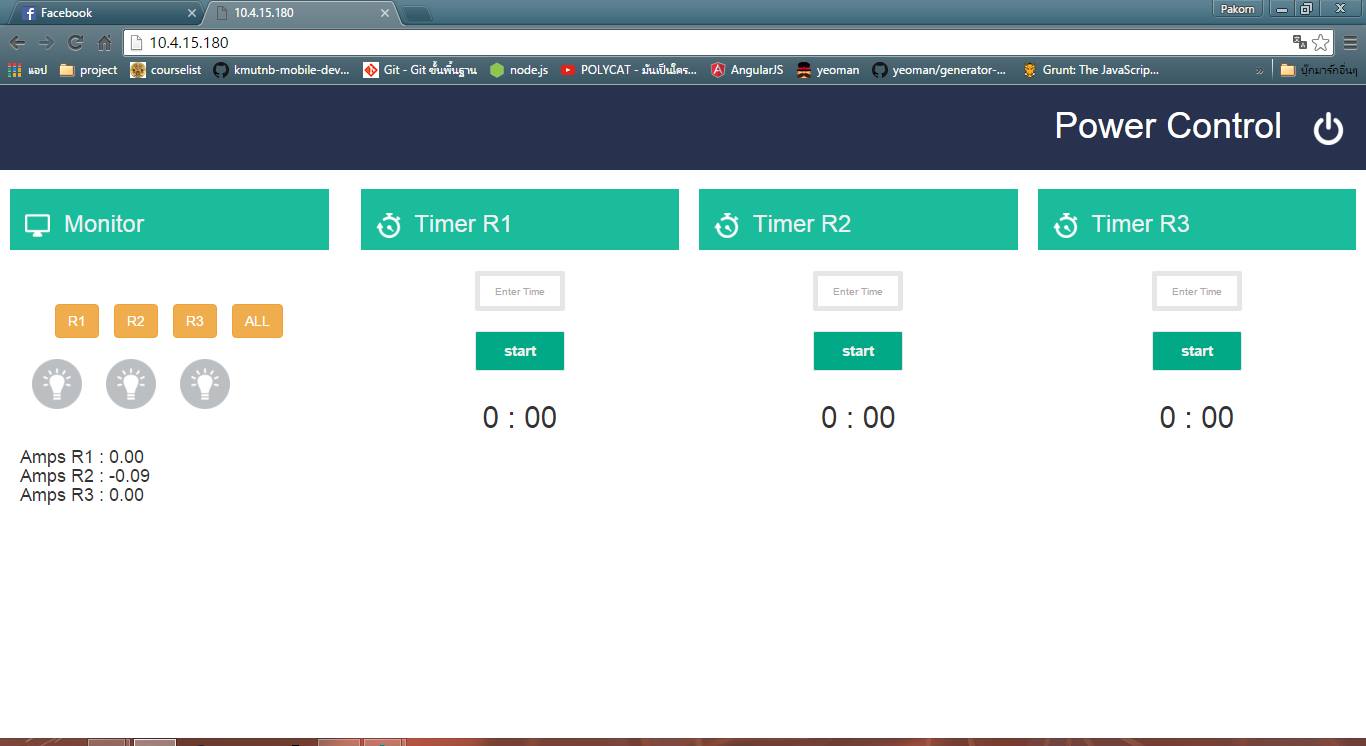
ระบบการเชื่อมต่อของอุปกรณ์ในส่วนต่าง ๆ ของระบบควบคุมไฟฟ้าด้วยเทคโนโลยี IoT ขณะนี้ได้เสร็จสมบูรณ์ตามระยะเวลาที่กำหนดไว้ และตรงตามวัตถุประสงค์ของโครงงานที่  
ได้กำหนดไว้ โดยระบบที่จัดขึ้นสามารถแบ่งออกเป็น 3 ส่วน ได้ดังนี้

1. ส่วนที่หนึ่ง คือ ส่วนของการแสดงสถานะและการควบคุมการเปิดปิดไฟฟ้าผ่านทาง  
หน้าเว็บบราวเซอร์

2. ส่วนที่สอง คือ ส่วนของการแสดงผลกระแสไฟฟ้าที่วัดได้จาก Sensor ACS712   
และค่ากำลังไฟฟ้าที่คำนวณจากกระแสไฟฟ้าที่วัดได้บนหน้าเว็บบราวเซอร์

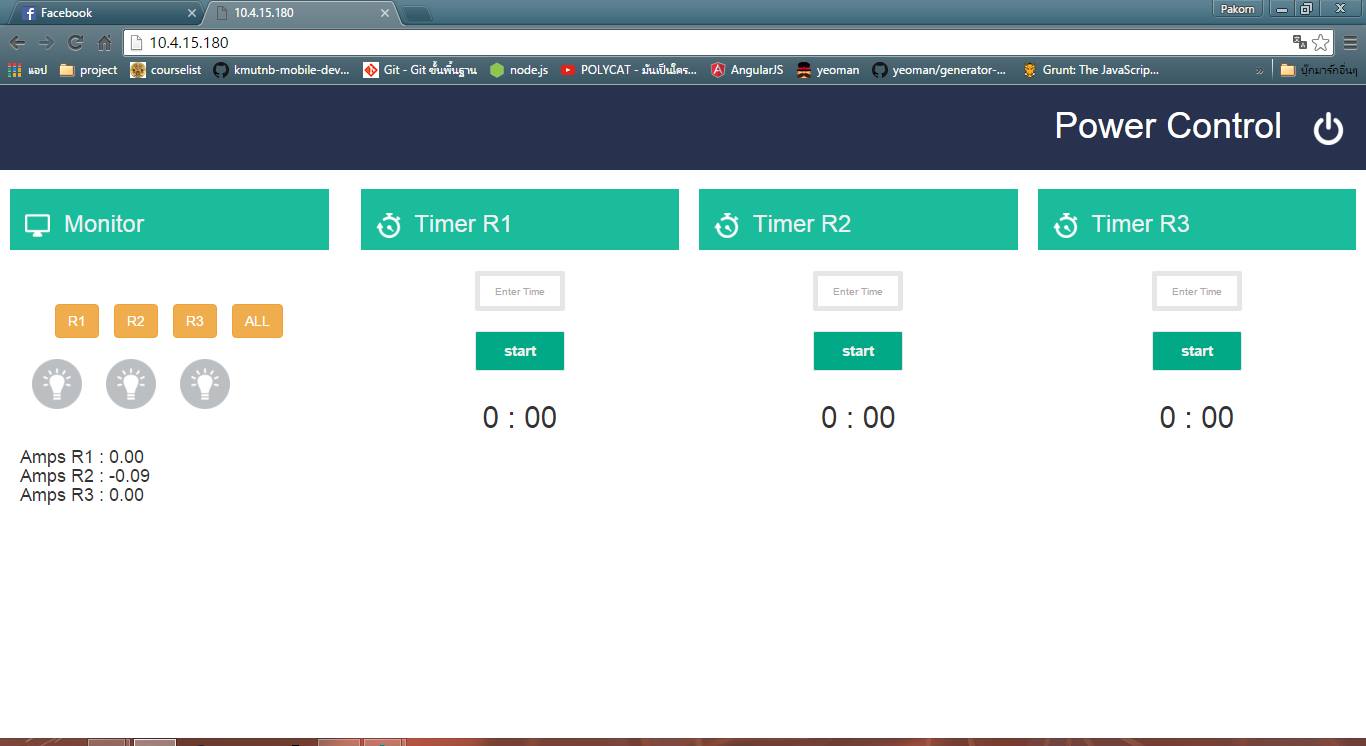
3. ส่วนที่สาม คือ ส่วนของการประกอบวงจรของชุดอุปกรณ์สำหรับควบคุมการเปิดปิดไฟฟ้าและ Sensor ACS712 สำหรับวัดกระแสไฟฟ้าที่ไหลผ่าน

**4.1 หน้าเว็บบราวเซอร์สำหรับควบคุมระบบและแสดงผล**



**ภาพที่ 4-1** แสดงหน้าเว็บบราวเซอร์สำหรับควบคุมและแสดงผล

จากภาพที่ 4-1 เป็นหน้าจอที่ใช้สำหรับแสดงสถานะการใช้ไฟฟ้าของอุปกรณ์ที่เชื่อมต่อเข้ากับระบบและควบคุมการเปิดปิดไฟฟ้า รวมทั้งแสดงค่ากระแสไฟฟ้าที่วัดได้จาก Sensor ACS712 และแสดงกำลังไฟฟ้า



**ภาพที่ 4-2** แสดงหน้าจอส่วนของการ Monitor

จากภาพที่ 4-2 เป็นส่วนที่แสดงค่าการ Monitor ต่าง ๆ ที่ต้องการ ซึ่งประกอบไปด้วย

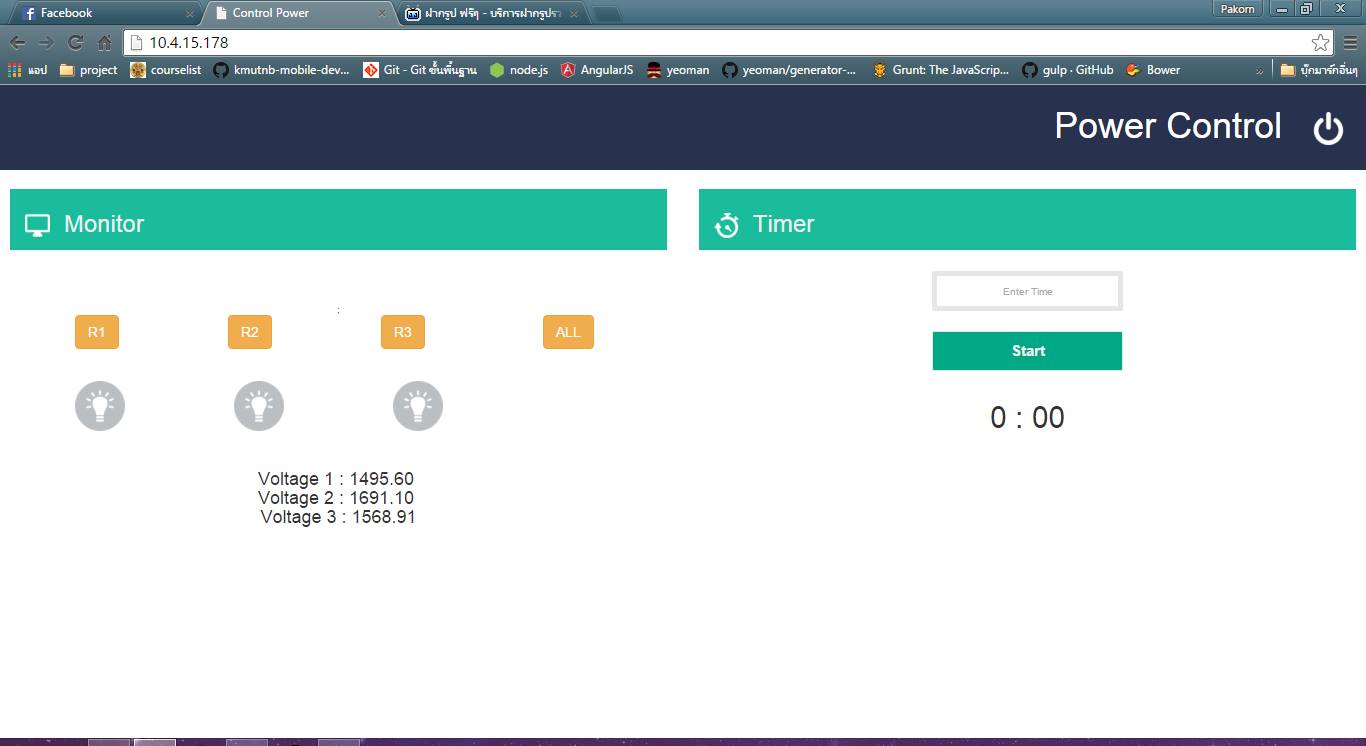
1. ค่าการแสดงสถานะการใช้งานของหลอดไฟที่เชื่อมต่ออยู่กับระบบ โดยจะแบ่งเป็น  
 2 สถานะ คือ

- ไอคอนสีเขียว คือ สถานะเปิด

- ไอคอนสีเทา คือ สถานะปิด

และหากต้องการจะทำการควบคุมการเปิดปิดไฟฟ้าสามารถได้ที่ไอคอนรูปไฟที่แสดงสถานะ ซึ่งจะทำการส่งคำสั่งไปที่ตัวไมโครคอนโทรลเลอร์เพื่อที่จะควบคุมการทำงานของรีเลย์ในการควบคุมการเปิดปิดไฟฟ้า

2. ค่ากระแสไฟฟ้าที่วัดได้จาก Sensor ACS712 และค่ากำลังไฟฟ้าที่คำนวณได้จาก  
ค่ากระแสไฟฟ้าที่อ่านได้

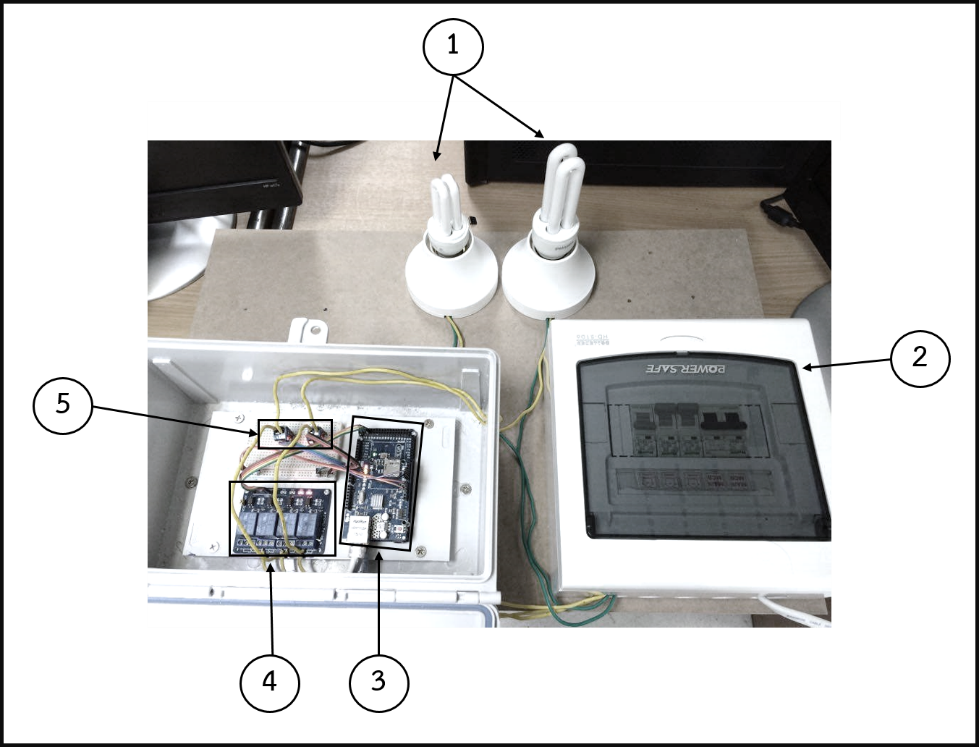


**ภาพที่ 4-3** แสดงหน้าจอส่วนของ Timer

จากภาพที่ 4-3 เป็นส่วนของการตั้งเวลาเพื่อเปิดปิดไฟฟ้า โดยจะสามารถตั้งเวลาได้  
โดยการกำหนดเวลาเป็นนาที จากนั้นกดที่ปุ่ม Start เพื่อทำการจับเวลาถอยหลัง

**4.2 การเชื่อมต่ออุปกรณ์**

4.2.1 ชุดทดลองการควบคุมระบบโดยใช้แลน



**ภาพที่ 4-4** แสดงตัวอย่างชุดทดลองการควบคุมระบบโดยใช้แลน

จากภาพที่ 4-4 เป็นภาพการเชื่อมต่ออุปกรณ์ต่าง ๆ เข้ากับอุปกรณ์บอร์ด Arduino Mega 2560 R3 เพื่อใช้ในการทำงานของระบบ โดยบอร์ด Arduino Mega 2560 R3 เป็นตัวควบคุมการทำงานของระบบทั้งหมด ซึ่งประกอบไปด้วยอุปกรณ์ดังนี้

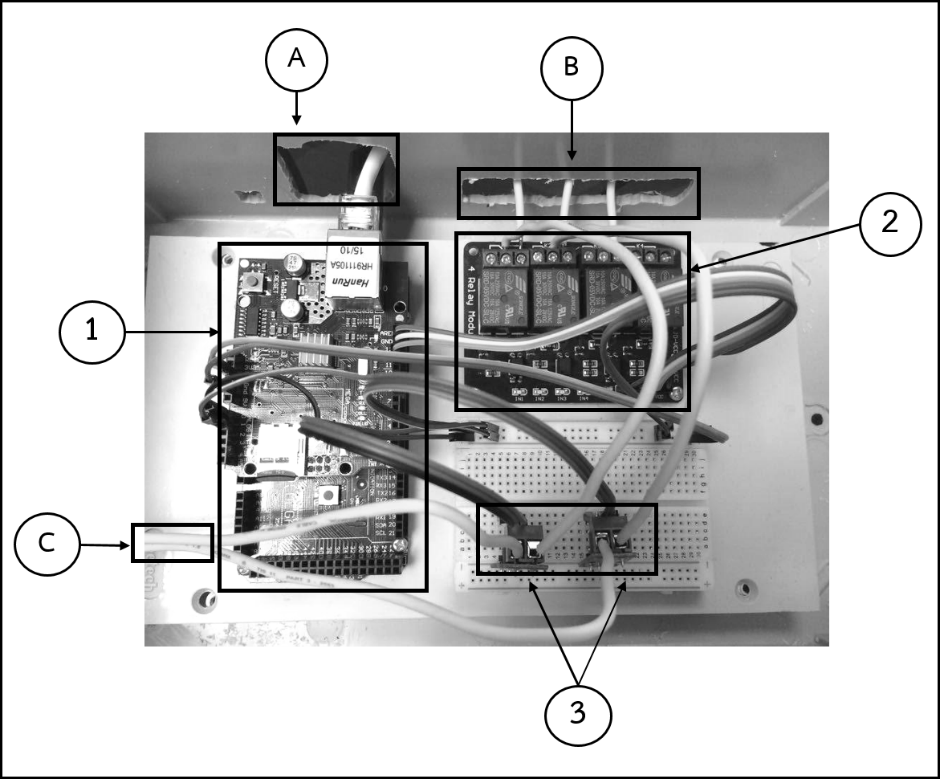
1. หลอดไฟ : สำหรับแทนไฟฟ้าในแต่ละห้อง

2. เบรกเกอร์ : ทำหน้าที่เป็นสะพานไฟรับไฟฟ้าจากข้างนอก

3. Arduino Mega : ทำหน้าที่ควบคุมระบบทั้งหมด

4. รีเลย์ : ทำหน้าที่รับคำสั่งการตัดไฟฟ้าจากไมโครคอนโทรลเลอร์

5. Sensor ACS712 : ใช้สำหรับวัดค่ากระแสไฟฟ้า



**ภาพที่ 4-5** แสดงการวางตัวอุปกรณ์บรรจุภายในกล่อง

จากเป็นภาพที่ 4-5 เป็นภาพการติดตั้งอุปกรณ์ในส่วนของไมโครคอนโทรลเลอร์และวงจรใหม่ที่จะนำมาต่อเข้ากับเบรกเกอร์เพื่อทำการควบคุมระบบ โดยภายในกล่องจะประกอบไปด้วย

1. Arduino Mega : ทำหน้าที่ควบคุมระบบทั้งหมด

2. รีเลย์ : ทำหน้าที่รับคำสั่งการตัดไฟฟ้าจากไมโครคอนโทรลเลอร์

3. Sensor ACS712 : ทำหน้าที่วัดค่ากระแสไฟฟ้าที่ไหลผ่าน

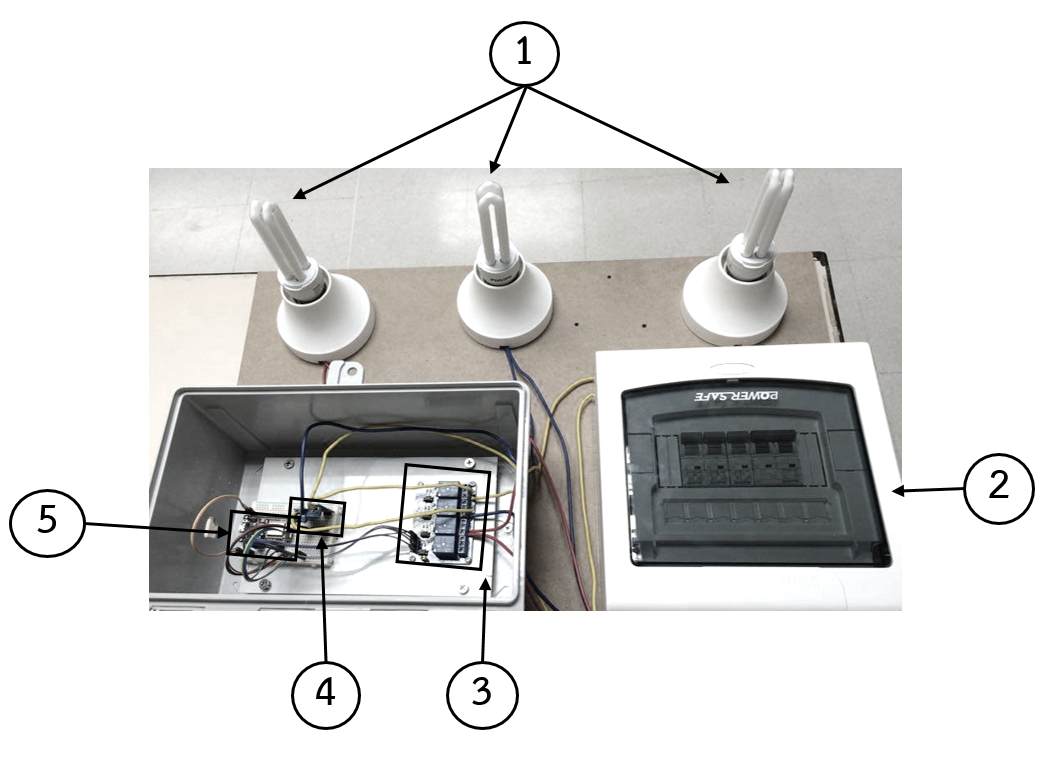
และมีการเชื่อมต่อจากอุปกรณ์ภายในกล่องเข้ากับระบบเดิม โดยมีการเชื่อมต่อ ดังนี้

A. เป็นการเชื่อมต่อแลนจาก Ethernet Shield ไปยัง Switch เพื่อเชื่อมต่อกับอินเตอร์เน็ต และเชื่อมต่อบอร์ด Arduino Mega 2560 R3 เข้ากับตัวจ่ายไฟ

B. เป็นการเชื่อมต่อรีเลย์เข้ากับเบรกเกอร์ สำหรับการควบคุมการจ่ายไฟ

C. เป็นการเชื่อมต่อ Sensor ACS712 เข้ากับหลอดไฟเพื่ออ่านค่ากระแสไฟฟ้า

4.2.2 ชุดทดลองการควบคุมระบบโดยใช้ WiFi



**ภาพที่ 4-6** แสดงตัวอย่างชุดทดลองการควบคุมระบบโดยใช้ WiFi

จากภาพที่ 4-6 เป็นภาพการเชื่อมต่ออุปกรณ์ต่าง ๆ เข้ากับตัว NodeMCU เพื่อใช้  
ในการทำงานของระบบ โดย NodeMCU จะเป็นตัวควบคุมการทำงานของระบบทั้งหมด ซึ่งประกอบไปด้วยอุปกรณ์ดังนี้

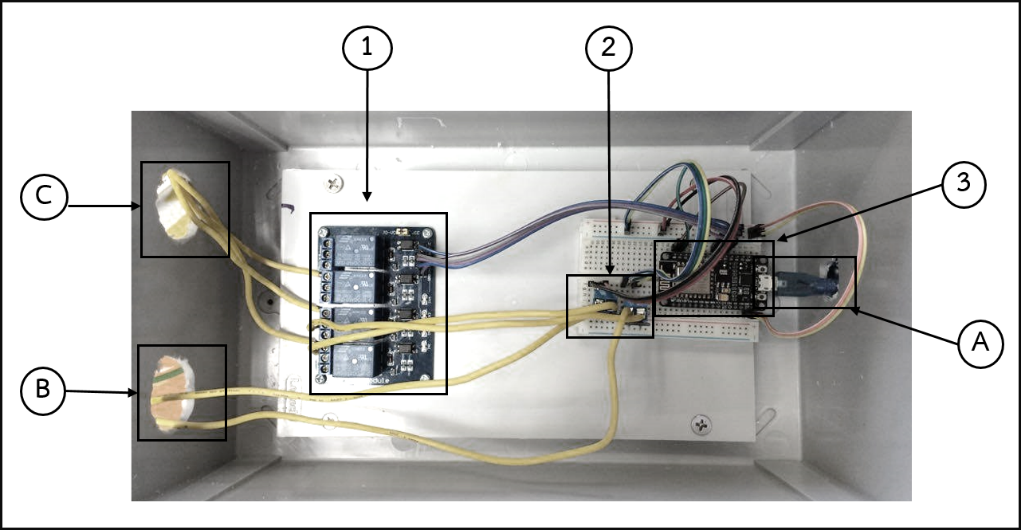
1. หลอดไฟ : สำหรับแทนไฟฟ้าในแต่ละห้อง

2. เบรกเกอร์ : ทำหน้าที่เป็นสะพานไฟรับไฟฟ้าจากข้างนอก

3. รีเลย์ : ทำหน้าที่รับคำสั่งการตัดไฟฟ้าจากไมโครคอนโทรลเลอร์

4. Sensor ACS712 : ทำหน้าที่วัดค่ากระแสไฟฟ้าที่ไหลผ่าน

5. NodeMCU : ทำหน้าที่ควบคุมระบบทั้งหมด



**ภาพที่ 4-7** แสดงการวางตัวอุปกรณ์บรรจุภายในกล่อง

จากเป็นภาพที่ 4-7 เป็นภาพการติดตั้งอุปกรณ์ในส่วนของไมโครคอนโทรลเลอร์และ  
วงจรใหม่ที่จะนำมาต่อเข้ากับเบรกเกอร์เพื่อทำการควบคุมระบบ โดยภายในกล่องจะประกอบไปด้วย

1. รีเลย์ : ทำหน้าที่รับคำสั่งการตัดไฟฟ้าจากไมโครคอนโทรลเลอร์

2. Sensor ACS712 : ทำหน้าที่วัดค่ากระแสไฟฟ้าที่ไหลผ่าน

3. NodeMCU : ทำหน้าที่ควบคุมระบบทั้งหมด

และมีการเชื่อมต่อจากอุปกรณ์ภายในกล่องเข้ากับระบบเดิม โดยมีการเชื่อมต่อ ดังนี้

A. เป็นการเชื่อมต่อ NodeMCU เข้ากับตัวจ่ายไฟฟ้า

B. เป็นการเชื่อมต่อ Sensor ACS712 เข้ากับหลอดไฟเพื่ออ่านค่ากระแสไฟฟ้า

C. เป็นการเชื่อมต่อรีเลย์เข้ากับเบรกเกอร์เพื่อควบคุมการจ่ายไฟฟ้าให้กับหลอดไฟ

**บทที่ 5**

**สรุปผลการดำเนินงาน**

**5.1 สรุปผลการดำเนินงาน**

ผลการดำเนินงานจะแยกออกเป็น 3 ส่วน ดังนี้

5.1.1 การออกแบบโครงสร้างของอุปกรณ์ในการเชื่อมต่อของอุปกรณ์บอร์ดไมโครคอนโทรลเลอร์กับส่วนประกอบต่าง ๆ

5.1.1.1 Arduino Mega R3

- สามารถควบคุมการทำงานของ Sensor ACS712 โดยจะทำหน้าที่สั่งให้ Sensor ACS712 ทำการวัดกระแสไฟฟ้าที่ไหลผ่าน แล้วนำค่าที่วัดได้มาผ่านกระบวนการ  
แปลงสัญญาณ Analog ให้เป็นสัญญาณ Digital จากนั้นนำค่าที่ได้มาแสดงผลบนหน้าเว็บบราวเซอร์

- สามารถควบคุมการทำงานของบอร์ดรีเลย์ในการจ่ายไฟฟ้าให้กับตัวอุปกรณ์

- สามารถรับคำสั่งการควบคุมจากหน้าเว็บได้

5.1.1.2 NodeMCU

- สามารถควบคุมการทำงานของ Sensor ACS712 โดยจะทำหน้าที่สั่งให้ Sensor ACS712 ทำการวัดกระแสไฟฟ้าที่ไหลผ่าน แล้วนำค่าที่วัดได้มาผ่านกระบวนการ แปลงสัญญาณ Analog ให้เป็นสัญญาณ Digital จากนั้นนำค่าที่ได้มาแสดงผลบนหน้าเว็บบราวเซอร์

- สามารถควบคุมการทำงานของบอร์ดรีเลย์ในการจ่ายไฟฟ้าให้กับตัวอุปกรณ์

- สามารถรับคำสั่งการควบคุมจากหน้าเว็บได้

5.1.2 การออกแบบและเขียนโปรแกรมดึงค่าที่ Sensor ACS712 วัดได้

5.1.2.1 ทำการเขียนโปรแกรมภาษา C ให้สามารถดึงค่าสัญญาณจาก Sensor ACS712 แล้วนำค่าที่ได้มาผ่านกระบวนการแปลงสัญญาณจาก Analog เป็นสัญญาณ Digital เพื่อให้ได้ผลลัพธ์ออกมาเป็นค่าตัวเลขที่ผู้ใช้ระบบสามารถเข้าใจได้ แล้วนำค่าที่ได้มาคำนวณหาค่ากระแสไฟฟ้า  
และค่ากำลังไฟฟ้า

5.1.2.2 ทำการเขียนโปรแกรมภาษา C ให้สามารถคำนวณหาค่าของกระแสไฟฟ้า  
และกำลังไฟฟ้าได้ โดยจะคำนวณจากค่า ADC ที่ Sensor ACS712 วัดได้

5.1.3 การออกแบบและเขียนโปรแกรมเพื่อแสดงค่าต่าง ๆ บนหน้าเว็บบราวเซอร์

5.1.3.1 ทำการเขียนโปรแกรมภาษา C เพื่อให้ส่งค่ากระแสไฟฟ้าและค่ากำลังไฟฟ้า ที่คำนวณได้ไปที่หน้าเว็บบราวเซอร์เพื่อแสดงผล

5.1.3.2 ทำการเขียนหน้าเว็บด้วยภาษา HTML และ JavaScript สำหรับรับค่าสถานะการเปิดปิดไฟฟ้า ค่ากระแสไฟฟ้า และกำลังไฟฟ้าที่ส่งมาจากบอร์ดไมโครคอนโทรลเลอร์เพื่อแสดงบนหน้าเว็บบราวเซอร์

5.1.3.3 ทำการเขียนหน้าเว็บด้วยภาษา HTML และ JavaScript สำหรับรับค่า การสั่งงานควบคุมสถานการเปิดปิดไฟฟ้าจากผู้ใช้ผ่านหน้าเว็บบราวเซอร์ แล้วส่งคำสั่งที่ได้รับไปที่บอร์ดไมโครคอนโทรลเลอร์เพื่อทำการควบคุมระบบต่อไป

**ตารางที่ 5-1** แสดงผลการเปรียบเทียบการทำงาน

|  |  |  |
| --- | --- | --- |
| **การทำงาน** | **Arduino Mega 2560 R3** | **NodeMCU** |
| 1. ความเร็วในการเปิดหน้าเว็บ | ช้ากว่า | เร็วกว่า |
| 2. ความเร็วการควบคุมการทำงาน | เร็วกว่า | ช้ากว่า |
| 3. ความเสถียรของระบบ | เสถียรมากกว่า | เสถียรน้อยกว่า |

จากตาราง 5-1 อธิบายได้ ดังนี้

1. ความเร็วในการเปิดหน้าเว็บ เนื่องจากว่าทั้ง 2 บอร์ด มีความเร็วในการเปิดหน้าเว็บที่ ใกล้เคียงกัน ขึ้นอยู่กับความเสถียรของอินเตอร์เน็ตในขณะนั้น และขนาดของไฟล์ รวมทั้งจำนวนของ Process ที่แต่ละบอร์ดต้องทำ ยิ่งมี Process เยอะก็จะยิ่งช้าลง

2. การควบคุมการทำงาน เป็นการควบคุมโดยใช้หลักการ Server Client ทั้ง 2 บอร์ด ซึ่งเป็นการทำงานที่ Server ผ่านทางหน้าเว็บ เพื่อควบคุมการทำงานของ Server แล้ว Server ก็จะส่งคำสั่งนั้น ๆ ไปยัง Client ซึ่งความเร็วในการควบคุมการทำงานนั้นก็จะขึ้นอยู่กับความเสถียรของอินเตอร์เน็ตในขณะนั้นด้วย เนื่องจากระบบนี้เป็นการควบคุมระบบแบบออนไลน์

3. ความเสถียรของระบบ Arduino มีความเสถียรมากกว่า เนื่องจากมีการเชื่อมต่อกับสายแลน ส่วนตัว NodeMCU เป็นการเชื่อมต่อกับ WiFi หากติดตั้งอุปกรณ์ไว้ในที่ที่อับสัญญาณอาจจะทำให้ระบบทำงานได้ช้ากว่า

**5.2 ปัญหาและอุปสรรคและแนวทางแก้ปัญหา**

จากการที่เริ่มพัฒนาระบบระบบควบคุมไฟฟ้าด้วยเทคโนโลยี IoTตั้งแต่ขั้นตอนการศึกษาค้นคว้าจนกระทั่งเสร็จสมบูรณ์ พบปัญหาที่เกิดขึ้นและมีแนวทางการแก้ไขปัญหา ดังนี้

5.2.1 การเชื่อมต่อวงจร เนื่องจากผู้พัฒนาไม่มีความเชี่ยวชาญในเรื่องของวงจรไฟฟ้า และการเชื่อมต่อวงจรของบางอุปกรณ์มีความซับซ้อน

แนวทางการแก้ปัญหา คือต้องศึกษาเกี่ยวกับการเชื่อมต่อวงจรทั้งในตัวของเบรกเกอร์ระบบเดิม และการเชื่อมต่อวงจรเดิมเข้าไมโครคอนโทรลเลอร์ให้เข้าใจ เพราะหากไม่มีความเข้าใจ  
ที่ถูกต้องอาจจะเป็นอันตรายได้

5.2.2 บอร์ดไมโครคอนโทรลเลอร์

ในช่วงเริ่มแรกผู้พัฒนาได้เลือกใช้บอร์ด Arduino รุ่น Uno R3 เป็นบอร์ดสำหรับควบคุมระบบแบบแลนเนื่องจากเป็นบอร์ดพื้นฐาน มีราคาที่ไม่แพง และง่ายต่อการศึกษาเนื่องจากมีตัวอย่างการทำงานอย่างแพร่หลาย แต่เนื่องด้วยคุณสมบัติที่ไม่เพียงพอกับความต้องการของระบบจึงทำให้  
มีปัญหาในการ Compile และ Run Code

แนวทางการแก้ปัญหา คือ เปลี่ยนไปใช้บอร์ด Arduino รุ่น Mega 2560 R3 เนื่องจาก มี Flash Memory มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า   
ในความเร็วของ MCU ที่เท่ากัน

**ตารางที่ 5-2** แสดงการเปรียบเทียบคุณสมบัติระหว่าง Arduino Uno R3 และ Arduino Mega 2560 R3

|  |  |  |
| --- | --- | --- |
| **คุณสมบัติ** | **Arduino Uno R3** | **Arduino Mega 2560 R3** |
| 1. ชิปไอซีไมโครคอนโทรเลอร์ | ATmega328 | ATmega2560 |
| 2. ใช้แรงดันไฟฟ้า | 5V | 5V |
| 3. รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ) | 7 – 12V | 7 – 12V |
| 4. รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด) | 6 – 20V | 6 – 20V |
| 5. Port Digital I/O | 14 port | 14 port |
| 6. กระแสไฟที่จ่ายได้ในแต่ละพอร์ต | 40 mA | 40 mA |
| 7. กระแสไฟที่จ่ายได้ในพอร์ต 3.3V | 50 mA | 50 mA |
| 8. พื้นที่โปรแกรมภายใน (Flash Memory) | 32 KB | 256 KB |
| 9. พื้นที่แรม | 2 KB | 8 KB |
| 10. พื้นที่หน่วยความจำถาวร (EEPROM) | 1 KB | 4 KB |
| 11. Clock rate | 16 MHz | 16 MHz |

5.2.3 การเขียนโปรแกรมควบคุมบอร์ดไมโครคอนโทรลเลอร์ เนื่องจากผู้พัฒนาไม่เชี่ยวชาญ  
ในการเขียนโปรแกรมควบคุมบอร์ดไมโครคอนโทรลเลอร์เพราะเป็นเรื่องใหม่ที่ผู้พัฒนาเพิ่งเริ่มศึกษา จึงจำเป็นต้องใช้เวลาในการศึกษาให้เข้าใจ

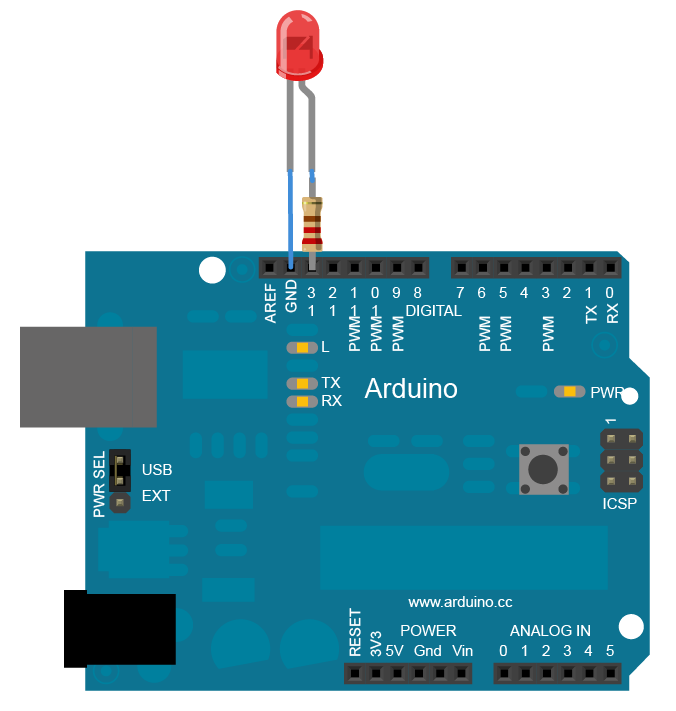
แนวทางการแก้ปัญหา คือ เนื่องจากบอร์ด Arduino และ NodeMCU ใช้ภาษา C   
ในการเขียนการควบคุมการทำงาน จึงทำการศึกษาโครงสร้างของภาษา C และวิธีการเขียนโปรแกรม เช่น คำสั่งการทำงานต่าง ๆ และทดลองเขียนโปรแกรมเพื่อควบคุมบอร์ดเบื้องต้น

**5.3 ข้อเสนอแนะ**

5.3.1 ควรศึกษาและเลือกอุปกรณ์ที่จะนำมาใช้ทำระบบให้เหมาะสมก่อนที่จะจัดซื้อ   
เพื่อลดความเสี่ยงทั้งในเรื่องค่าใช้จ่ายและความปลอดภัย

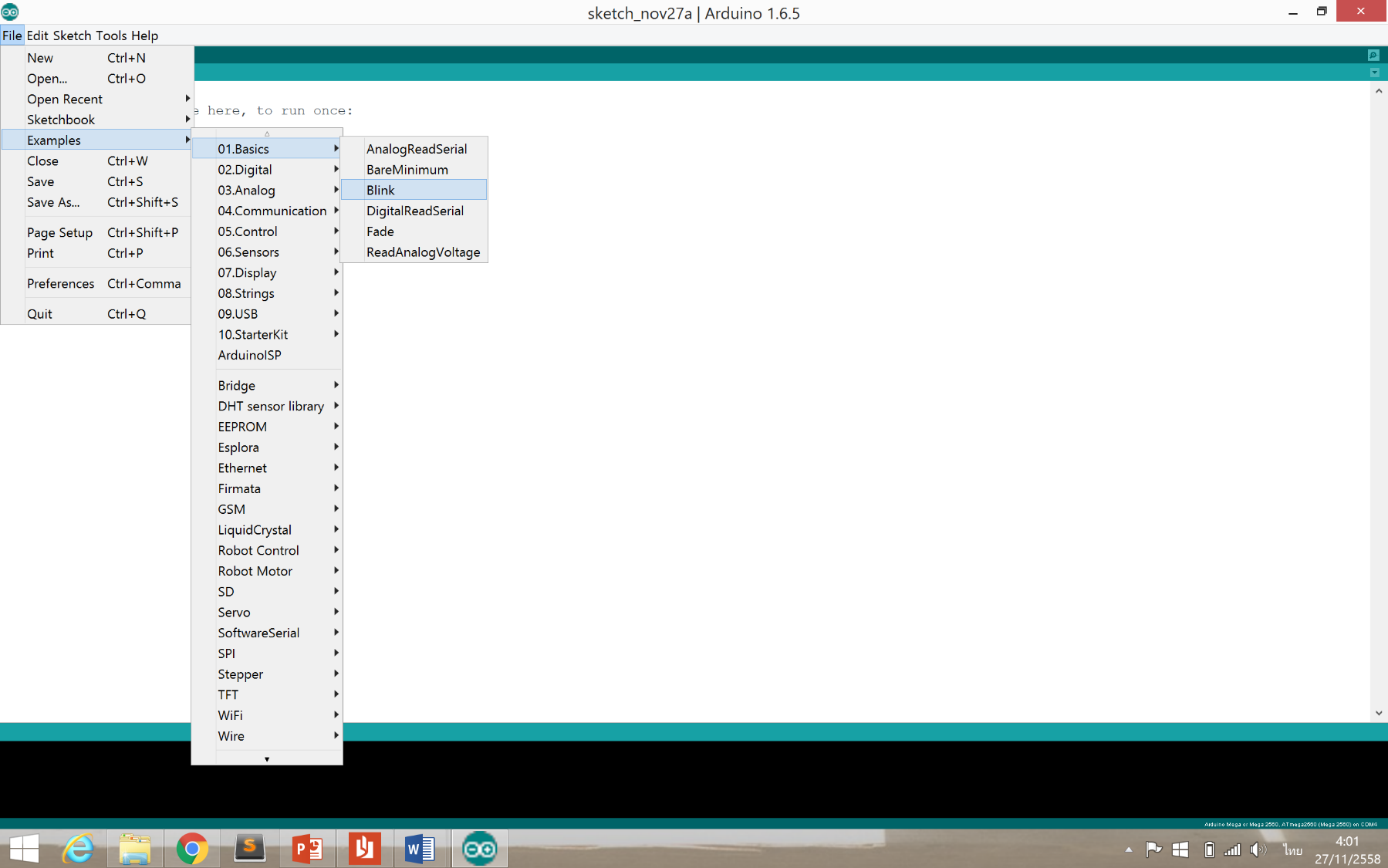
5.3.2 ควรศึกษาการต่อวงจรอิเล็กทรอนิกส์ให้เข้าใจ เพื่อป้องความเสียหายที่จะเกิดขึ้นทั้งกับอุปกรณ์และตัวผู้ทำ

5.3.3 ควรศึกษาวิธีการเขียนโปรแกรมด้วยภาษาที่เหมาะสมกับอุปกรณ์ที่ต้องการพัฒนาระบบ เช่น ในระบบนี้ใช้บอร์ด Arduino และ NodeMCU ซึ่งใช้ภาษา C ในการเขียนการทำงาน   
โดยในเริ่มแรกอาจจะใช้Code ตัวอย่างที่มีมาให้กับโปรแกรม Arduino IDE เพื่อทดลองการทำงานอย่างง่ายก่อน ยกตัวอย่างการต่อวงจรและ Code ตัวอย่างอย่างง่าย เช่น



**ภาพที่ 5-1** แสดงการทดลองการต่อวงจรอย่างง่าย

Code ตัวอย่างนี้สามารถเปิดได้จากโปรแกรม Arduino IDE โดยเข้าไปที่ File > Examples > 01.Basics > Blink



**ภาพที่ 5-2** แสดงตัวอย่างการเขียนโปรแกรมอย่างง่าย

Code ตัวอย่าง

void setup() {

// initialize digital pin 13 as an output.

pinMode(13, OUTPUT);

}

void loop() {

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(13, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

จาก Code ตัวอย่างสามารถอธิบายได้ดังนี้

1. ในฟังก์ชัน void setup() จะเป็นการทำงานในครั้งแรกเมื่อเปิดโปรแกรม จาก Code ตัวอย่างข้างต้น จะเป็นการกำหนด Pin ที่ 13 เป็นขา Output เพื่อแสดงผล ซึ่งตามรูปในวงจร   
Pin 13 นี้ต่ออยู่กับหลอด LED

2. ในฟังก์ชัน void loop() จะทำงานหลังจากฟังก์ชัน setup โดยจะทำงานไปเรื่อย ๆ จนกว่าจะปิดการทำงาน จาก Code ตัวอย่างข้างต้นสามารถอธิบายได้ คือ

- digitalWrite(13, HIGH); เป็นคำสั่งสำหรับสั่งให้ไฟที่ต่ออยู่กับขาที่ 13 ติด

- digitalWrite(13, LOW); เป็นคำสั่งสำหรับสั่งให้ไฟที่ต่ออยู่กับขาที่ 13 ดับ

- delay(1000); เป็นคำสั่งสำหรับการหน่วงเวลา

**5.4 สิ่งที่ได้รับจากการทำโครงงานพิเศษ**

หลังจากที่ได้ลงมือปฏิบัติงานโครงงานพิเศษครั้งนี้ ได้รับความรู้ต่าง ๆ ดังนี้

5.4.1 ได้เรียนเกี่ยวกับวงจรอิเล็กทรอนิกส์

5.4.2 ได้เรียนรู้เกี่ยวกับข้อมูลต่าง ๆ เกี่ยวกับไมโครคอนโทรลเลอร์

5.4.3 ได้เรียนรู้การวางแผนการดำเนินงานเป็นโครงงาน

5.4.4 ได้เรียนรู้การออกแบบและทำงานอย่างเหมาะสม

5.4.5 ได้เรียนรู้ประสบการณ์การทำงานด้านวงจรไฟฟ้า