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

5.2.4 การเลือกใช้อุปกรณ์รีเลย์ เนื่องจากรีเลย์มีหลายแบบหลายขนาด จึงต้องเลือกใช้ให้เหมาะสมตามปัจจัยต่าง ๆ เช่น กระแสไฟฟ้า

แนวทางการแก้ปัญหา คือ ศึกษาคุณสมบัติของรีเลย์ว่าสามารถรองรับการทำงานที่แรงดันและกระแสไฟฟ้าเท่าไร ใช้แรงดันไฟฟ้าในการทำงานอย่างไร โดยอาจจะต้องดูจากคุณสมบัติเบื้องต้นดังนี้

1. ยี่ห้อ รุ่นของผู้ผลิต (แบรนด์) รวมถึงสัญลักษณ์มาตรฐานต่างๆ
2. รายละเอียดของไฟฟ้ากระแสสลับที่รองรับการทำงานได้ (VAC)
3. รายละเอียดของไฟฟ้ากระแสตรงที่รองรับการทำงานได้ (VDC)
4. โมเดล ระดับแรงดันฝั่งขดลวด ชนิดและโครงสร้าง และข้อมูลด้าน Coil Sensitivity

โดยในขั้นตอนที่ทำชุดทดลองได้เลือกใช้ [Relay Module 4 Channels](http://thaieasyelec.com/products/components-th/relay/4-channels-relay-module-detail.html) มีเอาต์พุต  
คอนเน็คเตอร์ที่ Relay เป็น NO/COM/NC สามารถใช้กับโหลดได้ทั้งแรงดันไฟฟ้า DC และ AC   
โดยใช้สัญญาณในการควบคุมการทํางานด้วยสัญญาณโลจิก TTL

**คุณสมบัติ**

- รีเลย์เอาต์พุตแบบ SPDT จํานวน 4 ช่อง

- สั่งงานด้วยระดับแรงดัน TTL

- CONTACT OUTPUT ของรีเลย์รับแรงดันได้สูงสุด 250 VAC 10 A, 30 VDC 10A

- มี LED แสดงสถานะ การทํางานของรีเลย์และแสดงสถานะของบอร์ด

- มีจัมพ์เปอร์สําหรับเลือกว่าจะใช้กราวด์ร่วมหรือแยก

- มี OPTO-ISOLATED เพื่อแยกกราวด์ส่วนของสัญญาณควบคุมกับไฟฟ้าที่ขับรีเลย์

ออกจากกัน

และในขั้นตอนที่สองเลือกใช้ บอร์ด Relay 16 ช่อง 12 โวลต์ 10A 250V เนื่องจากต้องการใช้ควบคุมไฟฟ้าจำนวน 10 ห้องจึงต้องใช่รีเลย์ขนาด 16 ช่อง และแต่ละห้องมีไฟผ่านไม่เกิน 10A

คุณสมบัติ

- การเชื่อมต่อมาตรฐานที่สามารถใช้ควบคุมได้โดยตรงจากไมโครคอนโทรนเลอร์ เช่น  Arduino , 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic

- มีวงจรเรกูเลเตอร์ 5V ในตัวบอร์ด สามารถจ่ายไฟให้ Arduino หรือวงจรอื่นที่ต้องการได้อย่างสะดวก

- ใช้ควบคุมไฟฟ้าแรงสูงได้ที่ DC30V 10A , AC250V 10A

- ใช้กระแสขับ relay แต่ละตัวที่ 15-20 mA

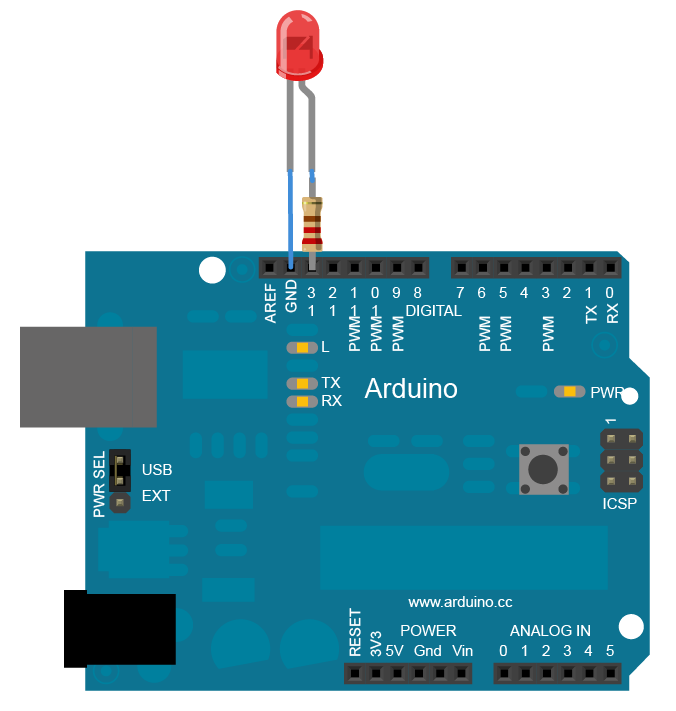
- วงจรขับรีเลย์เป็นแบบแยกกราวด์ Opto isolated Relay ปลอดภัยต่อวงจรไมโครคอนโทรเลอร์

**5.3 ข้อเสนอแนะ**

5.3.1 ควรศึกษาและเลือกอุปกรณ์ที่จะนำมาใช้ทำระบบให้เหมาะสมก่อนที่จะจัดซื้อ   
เพื่อลดความเสี่ยงทั้งในเรื่องค่าใช้จ่ายและความปลอดภัย

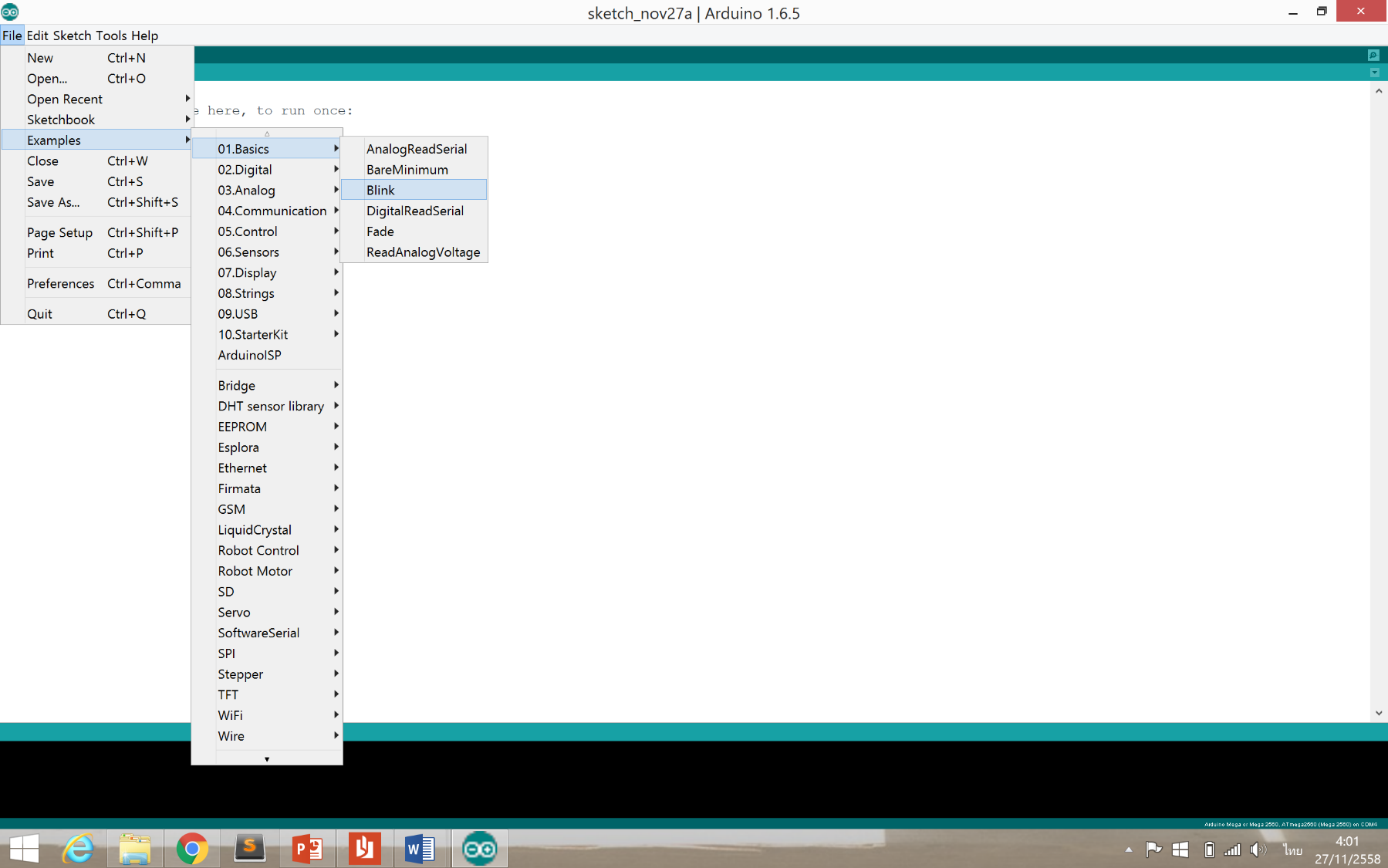
5.3.2 ควรศึกษาการต่อวงจรอิเล็กทรอนิกส์ให้เข้าใจ เพื่อป้องความเสียหายที่จะเกิดขึ้นทั้งกับอุปกรณ์และตัวผู้ทำ

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