

บทที่ 4

การพัฒนาและสร้างระบบจัดการพลังงานไฟฟ้าแสงสว่าง

ในบทนี้เป็นการนำเสนอรายละเอียดของการพัฒนาและสร้างระบบจัดการพลังงานไฟฟ้าแสงสว่างจากการเลือกใช้ 6 ฟังก์ชันการทำงาน ซึ่งได้อธิบายถึงวัสดุอุปกรณ์ วงจร และโปรแกรมที่ใช้ในการทดลอง นอกจากนี้จะเป็นการนำเสนอรายละเอียดของโครงสร้าง Hardware และ Software ที่ได้พัฒนาขึ้นมาในระบบจัดการพลังงานไฟฟ้าแสงสว่างนี้

4.1 อุปกรณ์ที่ใช้ในการพัฒนาและสร้างระบบจัดการพลังงานไฟฟ้าแสงสว่าง

ระบบจัดการพลังงานไฟฟ้าแสงสว่างที่ได้ออกแบบมานั้นได้ถูกนำไปใช้ในหลอดไฟ 2 ชนิด ได้แก่ หลอดฟลูออเรสเซนต์และหลอด LED โดยมีอุปกรณ์ที่ใช้ในการพัฒนาและสร้าง ดังนี้ต่อไป

4.1.1 อุปกรณ์ทั่วไป

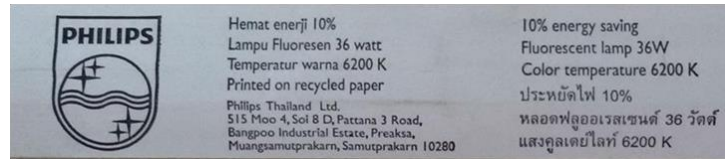
1) หลอดไฟ 2 ชนิด ประกอบด้วย

1.1) หลอดฟลูออเรสเซนต์

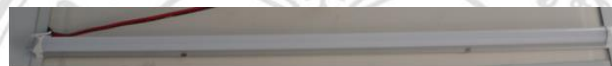
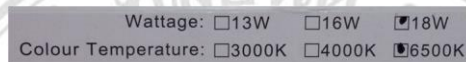
อุปกรณ์ให้แสงสว่างพื้นฐานที่ใช้ในงานวิจัยนี้ได้เลือกใช้หลอดฟลูออเรสเซนต์ขนาดฟลักซ์ 36 Watt ค่าแรงดัน 230 Vac ประสิทธิภาพการส่องสว่าง 72 lm/Watt เป็นแสงชนิด Cool Daylight 6,200 K อายุการใช้งาน 13,000 ชั่วโมง ดังแสดงในรูปที่ 4.1

1.2) หลอด LED

อุปกรณ์ให้แสงสว่างพื้นฐานอีกชนิดที่ได้เลือกใช้ในงานวิจัยนี้คือ หลอด LED ขนาดฟลักซ์ 18 Watt ใช้แรงดันกระแสตรง 12 Vdc เป็นแสงชนิด Cool Daylight 6,500 K อายุการใช้งาน 50,000 ชั่วโมง ดังแสดงในรูปที่ 4.2



รูปที่ 4.1 หลอดไฟชนิดหลอดฟลูออเรสเซนต์



รูปที่ 4.2 หลอดไฟชนิดหลอด LED

2) ไมโครคอนโทรลเลอร์ (Microcontroller)

ในงานวิจัยนี้ได้ใช้ไมโครคอนโทรลเลอร์ในการรับค่าหรือสัญญาณที่วัดได้จาก เซนเซอร์ตรวจจับการเคลื่อนไหวและเซนเซอร์วัดความเข้มแสง อีกทั้งควบคุม การเปิด/ปิดและปรับหรี่แสงสว่างจากหลอดไฟผ่านวงจรอิเล็กทรอนิกส์ จึง เลือกใช้ไมโครคอนโทรลเลอร์ Arduino Atmega 328 รุ่น UNO R3 ดังแสดงใน รูปที่ 4.3 ซึ่งมีคุณสมบัติที่เพียงพอในการนำไปใช้งาน เพราะสามารถเชื่อมต่อ สัญญาณขาเข้า/ออกได้ทั้งแบบอนาล็อกและดิจิตอลจำนวน 20 ขา (สามารถสร้าง สัญญาณ PWM ได้ 6 ขา) อีกทั้งยังสามารถประยุกต์ใช้ร่วมกับเซนเซอร์ชนิด ต่าง ๆ ได้ โดยใช้ภาษา C++ ในการโปรแกรมด้วยรูปแบบคำสั่งที่ง่ายไม่ซับซ้อน และมีราคาที่เหมาะสม



รูปที่ 4.3 บอร์ดไมโครคอนโทรลเลอร์ Arduino Atmega 328 รุ่น UNO

3) บอร์ดคอมพิวเตอร์ขนาดเล็ก (Raspberry Pi) และ SD Card

ในการประมวลผลและจัดการข้อมูลต่าง ๆ ไม่ว่าจะเป็นคำสั่งการควบคุมแสงสว่างจากผู้ใช้หรือข้อมูลการวัดค่าปริมาณทางไฟฟ้าที่ระบบใช้งานที่วัดได้จากมิเตอร์ อีกทั้งการให้ผู้ใช้จะสามารถควบคุมการใช้แสงสว่างและดูข้อมูลผ่านทางโปรแกรมจัดการพลังงานไฟฟ้าแสงสว่าง โดยทำหน้าที่เป็นทั้ง Web Server และ Database Server จึงได้เลือกใช้บอร์ด Raspberry Pi ดังแสดงในรูปที่ 4.4 ที่มีความสามารถเหมือนคอมพิวเตอร์ทั่วไป เพียงแต่อาจจะมีความเร็วที่น้อยกว่าสามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ บอร์ดนี้จะใช้ระบบปฏิบัติการลินุกซ์ (Linux Operating System) ซึ่งต้องทำการติดตั้งลง SD Card ซึ่งถือได้ว่ามีคุณสมบัติเพียงพอในการทำงานนี้ อีกทั้งยังมีขนาดเล็กและราคาไม่แพง

4) จอแบบสัมผัส (Touch Screen)

ในการควบคุมปริมาณแสงสว่างรวมถึงแสดงผลข้อมูลที่ได้นับที่ก/จัดเก็บไว้ผ่านโปรแกรมจัดการพลังงานไฟฟ้าแสงสว่างที่ได้สร้างขึ้น ในงานวิจัยนี้ได้เลือกใช้จอแบบสัมผัสชนิด TFT ขนาด 3.5 นิ้ว ความละเอียด 320 x 480 พิกเซล เนื่องจากสามารถเชื่อมต่อกับบอร์ด Raspberry Pi ผ่าน SPI บน GPIO ได้และมีขนาดที่เหมาะสม ดังแสดงในรูปที่ 4.5

5) แหล่งจ่ายไฟ 5 Vdc (Adapter 5 Vdc)

เนื่องจาก Raspberry Pi ทำงานในระดับแรงดัน 5 Vdc และมีการกินกระแสเพียง 100-300 mA จึงเลือกใช้แหล่งจ่ายไฟขนาด 5 Vdc 3 A ซึ่งเพียงพอต่อการทำงาน ดังแสดงในรูปที่ 4.6



รูปที่ 4.4 บอร์ดคอมพิวเตอร์ขนาดเล็ก (Raspberry Pi)



รูปที่ 4.5 จอแบบสัมผัส รุ่น TFT



รูปที่ 4.6 แหล่งจ่ายไฟให้ Raspberry Pi

6) เซนเซอร์วัดความเข้มแสง (Light Sensor)

ในการวัดค่าความสว่างเฉลี่ยภายในห้อง ในงานวิจัยนี้ได้เลือกใช้เซนเซอร์วัดความเข้มแสง BH1750FVI รุ่น GY-302 ดังแสดงในรูปที่ 4.7 ที่สามารถใช้วัดแสงจำพวกแสงจากหลอดไส้ หลอดฟลูออเรสเซนต์ หลอด LED หลอดฮาโลเจน และแสงอาทิตย์ได้ ที่สำคัญมีความคลาดเคลื่อนเพียง $\pm 20\%$ และสามารถใช้งานร่วมกับบอร์ดไมโครคอนโทรเลอร์ได้ ผ่านการสื่อสารผ่านบัส I2C โดยมีช่วงการวัดอยู่ที่ 1-65535 lux ซึ่งเพียงพอต่อการทำงานวิจัยนี้

7) เซนเซอร์ตรวจจับการเคลื่อนไหว (Motion Sensor)

ในการตรวจสอบผู้ใช้งานในบริเวณต่าง ๆ จากการตรวจจับการเคลื่อนไหวในบริเวณนั้น ๆ ว่ามีการเคลื่อนไหวอยู่หรือไม่ เพื่อสั่งการเปิดไฟเมื่อมีการเคลื่อนไหวหรือมีผู้ใช้งานและปิดไฟเมื่อไม่มีการใช้งานในบริเวณนั้น ในงานวิจัยนี้ได้เลือกใช้เซนเซอร์ตรวจจับการเคลื่อนไหวชนิด PIR Motion Sensor Module HC-SR501 ซึ่งเป็นโมดูลตรวจจับความเคลื่อนไหวที่ใช้การตรวจจับรังสีอินฟราเรดที่แผ่ออกมาจากมนุษย์ อีกทั้งยังสามารถใช้งานร่วมกับบอร์ดไมโครคอนโทรเลอร์ได้ด้วย โดยทำงานได้ในระยะไม่เกิน 8-10 เมตร และตรวจจับการเคลื่อนไหว 55 องศาจากแนวระนาบ ที่สำคัญมีขนาดเล็กใช้พลังงานต่ำหรือใช้กระแสประมาณ 65 ไมโครแอมป์ดังแสดงในรูปที่ 4.8



รูปที่ 4.7 เซนเซอร์วัดความเข้มแสง



รูปที่ 4.8 เซนเซอร์ตรวจจับการเคลื่อนไหว

8) มิเตอร์วัดพลังงาน

ในการวัดค่าปริมาณไฟฟ้าที่มีการใช้ในระบบแสงสว่าง เช่น ค่ากระแส แรงดัน กำลังไฟฟ้า ฯลฯ โดยจะวัดกำลังไฟฟ้าทั้งหมดที่จ่ายให้กับกล่องควบคุมหรือวัดที่บริเวณกล่องควบคุม จึงเลือกใช้มิเตอร์วัดพลังงานรุ่น Elnet LT ที่สามารถสื่อสารผ่านพอร์ต RS 485 เพื่อส่งข้อมูลให้บอร์ด Raspberry Pi ดังแสดงในรูปที่ 4.9

9) USB to RS-485 Converter

ในการใช้เชื่อมต่อสื่อสารระหว่างบอร์ด Raspberry Pi กับมิเตอร์วัดพลังงาน เพื่อรับข้อมูลปริมาณทางไฟฟ้าจากมิเตอร์วัดพลังงาน ซึ่งใช้รูปแบบการสื่อสารผ่านพอร์ต RS 485 ของมิเตอร์มายังพอร์ต USB บน Raspberry Pi จึงได้เลือกใช้ USB to RS-485 Converter เป็นตัวเชื่อมต่อสื่อสารข้อมูลระหว่างสองพอร์ตดังแสดงในรูปที่ 4.10

จากอุปกรณ์ที่ได้กล่าวมาข้างต้นได้แสดงรายละเอียดของอุปกรณ์ไว้ในภาคผนวก ข



รูปที่ 4.9 มิเตอร์วัดพลังงาน



รูปที่ 4.10 USB to RS-485 Converter

4.1.2 อุปกรณ์ที่ใช้สำหรับระบบควบคุมที่ไม่สามารถปรับระดับแสงสว่างได้ ซึ่งใช้ควบคุมหลอดไฟชนิดหลอดฟลูออเรสเซนต์

1) แหล่งจ่าย 7 Vdc (Adapter)

บอร์ด Arduino ใช้ไฟเลี้ยงขนาด 7-12 Vdc กินกระแสประมาณ 100-200 mA โดยขึ้นกับพอร์ตที่ใช้งาน ได้เลือกใช้แหล่งจ่ายไฟขนาด 7 Vdc 2A เป็นแหล่งจ่ายไฟให้กับบอร์ดดังแสดงในรูปที่ 4.11

2) อุปกรณ์ที่ใช้ในวงจรควบคุมการเปิด/ปิดแสงสว่าง (Switching Circuit)

2.1) รีเลย์ 5 โวลต์ ชนิด 5 Vdc ดังแสดงในรูปที่ 4.12

2.2) ทรานซิสเตอร์ ชนิด NPN เบอร์ BC547 ดังแสดงในรูปที่ 4.13

2.3) ไดโอด

2.4) ความต้านทาน 1 กิโลโอห์ม



รูปที่ 4.11 แหล่งจ่ายไฟให้ไมโครคอนโทรลเลอร์

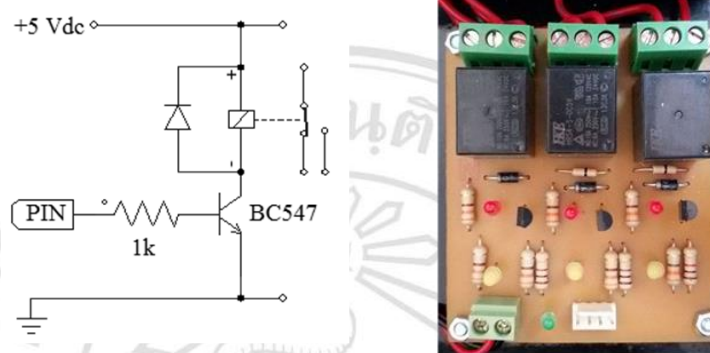


รูปที่ 4.12 รีเลย์ 5 โวลต์



รูปที่ 4.13 ทรานซิสเตอร์ ชนิด NPN เบอร์ BC54

วงจร Switching Circuit ใช้รีเลย์ในการควบคุมการเปิด/ปิดหลอดไฟ โดยรีเลย์จะทำงานเมื่อได้รับคำสั่ง/สัญญาณจากไมโครคอนโทรลเลอร์ เช่น เมื่อมีการสั่งให้เปิดไฟทรานซิสเตอร์จะช่วยขยายกระแสจากไมโครคอนโทรลเลอร์ให้สูงขึ้นจนจับขดลวดของรีเลย์ได้ ทำให้รีเลย์ต่อเข้ากับขา NO เพื่อปิดวงจรหลอดไฟก็จะสว่าง โดยรีเลย์จะการทำงานเหมือนสวิตช์ทั่วไปซึ่งมีวงจรไฟฟ้าดังแสดงในรูปที่ 4.14



รูปที่ 4.14 วงจรเปิด/ปิดหลอดไฟ

4.1.3 อุปกรณ์ที่ใช้สำหรับระบบควบคุมที่สามารถเปิด/ปิดและปรับระดับแสงสว่างได้ ซึ่งใช้ควบคุมหลอดไฟชนิดหลอด LED

1) สวิตซ์ชิงเพาเวอร์ซัพพลาย (Switching Power Supply)

ในการจ่ายแรงดัน 12 Vdc ให้กับหลอดไฟ LED ทั้งหมดในระบบได้เลือกสวิตซ์ชิงเพาเวอร์ซัพพลายที่สามารถจ่ายกระแสได้ 29 A เพื่อจ่ายแรงดันและกระแสให้ระบบ อีกทั้งยังปรับลดแรงดันลงเหลือ 5 Vdc เพื่อนำมาจ่ายแรงดันให้บอร์ด Arduino และบอร์ด Raspberry Pi อีกด้วยดังแสดงในรูปที่ 4.15



รูปที่ 4.15 สวิตซ์ชิงเพาเวอร์ซัพพลาย

2) บอร์ดเรกูเลเตอร์แบบลดแรงดัน (DC-to-DC Step Down Module)

เพื่อลดแรงดันไฟฟ้ากระแสตรงจากสวิทช์ซึ่งพาวเวอร์ซัพพลายขนาด 12 โวลต์ ให้เหลือ 5 โวลต์ก่อนจ่ายให้แก่บอร์ด Arduino และบอร์ด Raspberry Pi ได้เลือกใช้บอร์ดเรกูเลเตอร์แบบลดแรงดันที่สามารถปรับแรงดันได้ตั้งแต่ 0-12 Vdc ดังแสดงในรูปที่ 4.16

3) USB Hub

ในการเชื่อมจาก DC-to-DC Step Down Module เพื่อจ่ายแรงดันไฟฟ้า 5 Vdc ให้บอร์ด Arduino และบอร์ด Raspberry Pi ผ่านทางพอร์ต USB ได้เลือกใช้ USB Hub ที่รองรับการเชื่อมต่อพอร์ต USB มากถึง 4 พอร์ตทำให้เพียงพอต่อการใช้งานดังแสดงในรูปที่ 4.17

4) อุปกรณ์ที่ใช้ในวงจรควบคุมการเปิด/ปิดและปรับหรือแสงสว่าง (Dimming Circuit)

4.1) รีเลย์ 12 โวลต์ ชนิด 12 Vdc ดังแสดงในรูปที่ 4.18

4.2) MOSFET เบอร์ IRL 540 และแผ่นระบายความร้อน (Heat Sink) ดังแสดงในรูปที่ 4.19

4.3) ความต้านทาน 1 กิโลโอห์ม และ 220 กิโลโอห์ม



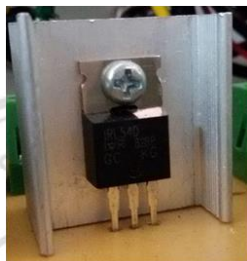
รูปที่ 4.16 บอร์ดเรกูเลเตอร์แบบลดแรงดัน



รูปที่ 4.17 USB Hub

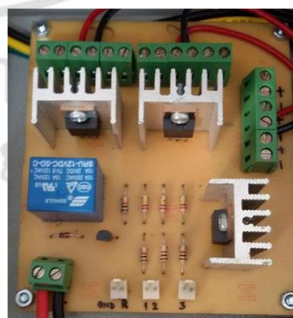
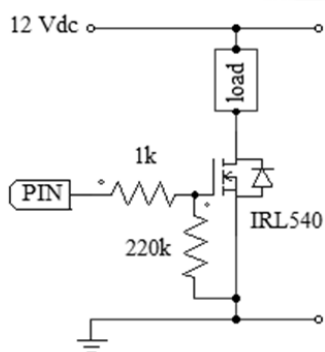


รูปที่ 4.18 รีเลย์ 12 โวลต์



รูปที่ 4.19 MOSFET เบอร์ IRL 540 พร้อมแผ่นระบายความร้อน

ในการควบคุมเปิด/ปิดและปรับหรือหลอดไฟที่ใช้สำหรับแรงดันไฟฟ้ากระแสตรงได้เลือกใช้ MOSFET (IRL540) ในควบคุมแรงดันที่ตกคร่อมหลอดไฟด้วยวิธีการ PWM ที่สามารถควบคุมระดับแรงดันไฟฟ้าเฉลี่ยได้ถึง 0-100% ผ่านการสร้างสัญญาณจากไมโครคอนโทรลเลอร์ (ระดับความถี่ในการส่งสัญญาณมีสองระดับคือ 490 Hz และ 980 Hz) โดยในการพัฒนาได้ควบคุมระดับแสงสว่างเป็น 4 ระดับ จากการควบคุมแรงดันเฉลี่ยที่จะถูกส่งออกมาจากไมโครคอนโทรลเลอร์ 25%, 50%, 75% และ 100% ซึ่งมีวงจรดังแสดงในรูปที่ 4.20



รูปที่ 4.20 วงจรควบคุมการเปิด/ปิด และปรับหรือแสงสว่าง

4.2 โปรแกรมคอมพิวเตอร์ที่ใช้ในการควบคุมระบบ

ในการพัฒนาระบบจัดการพลังงานไฟฟ้าแสงสว่างมีการใช้โปรแกรมในการพัฒนาทั้งในส่วนควบคุมการทำงานของไมโครคอนโทรลเลอร์, การสร้าง Web Server บน Raspberry Pi และการสร้างโปรแกรมจัดการพลังงานไฟฟ้าแสงสว่างที่ให้ผู้ใช้งานสามารถใช้งานผ่าน Touch Screen และหน้า Webpage ซึ่งมีโปรแกรมที่ใช้งานดังต่อไปนี้

4.2.1 โปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ (Microcontroller Program)

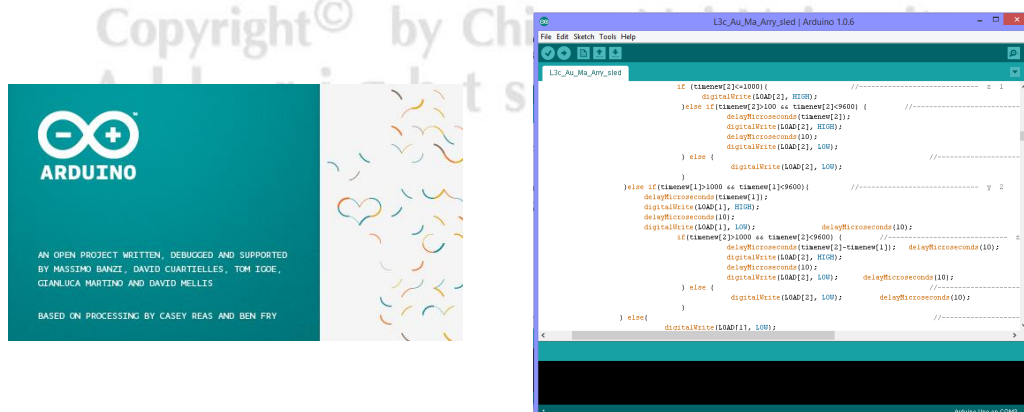
1) โปรแกรม ArduinoIDE

ในการเขียนชุดคำสั่งให้กับไมโครคอนโทรลเลอร์เพื่อทำการควบคุมระบบ จะใช้โปรแกรม ArduinoIDE ดังแสดงในรูปที่ 4.21 ซึ่งเป็นโปรแกรมที่พัฒนาขึ้นมาเพื่อการเขียนชุดคำสั่งให้กับบอร์ด Arduino โดยเฉพาะสามารถติดตั้งใช้งานได้กับทุกระบบปฏิบัติการและใช้ภาษา C/C++ ในการเขียนโปรแกรม ที่สำคัญไม่ต้องเสียค่าใช้จ่ายในการใช้งาน

4.2.2 โปรแกรมที่ใช้ในการสร้าง Web Server บน Raspberry Pi (Web Server Program)

1) โปรแกรม Apache2

ในการสร้าง Web Server บน Raspberry Pi หรือ การแปลง Raspberry Pi ให้สามารถบริการ Web Site ที่เรียกใช้จากเครื่องคอมพิวเตอร์เครื่องอื่น ๆ ผ่าน Web Browser ซึ่งจะคอยจัดเก็บและส่ง Homepage ไปยัง Browser ที่มีการเรียก



รูปที่ 4.21 โปรแกรม ArduinoIDE

เข้าใช้งาน ในงานวิจัยนี้เลือกใช้โปรแกรม Apache2 เพราะเป็นซอฟต์แวร์แบบ Open Source ที่เปิดให้บุคคลทั่วไปสามารถเข้ามาร่วมพัฒนาโดยไม่เสียค่าใช้จ่าย มี Packages พร้อมใช้งานครบถ้วนและสามารถเลือกใช้ภาษาในการพัฒนาได้หลากหลาย ที่สำคัญ คือ สามารถทำงานบนระบบลินุกซ์ (Linux) ซึ่งเป็นระบบปฏิบัติการที่ใช้ใน Raspberry Pi

2) โปรแกรม MySQL

ในการสร้าง Raspberry Pi เป็น Database Server เพื่อใช้ในการเก็บข้อมูลคำสั่งควบคุมระบบแสงสว่างจากผู้ใช้ ไม่ว่าจะสั่งเปิด/ปิด และปรับหรี่ จากผู้ใช้โดยตรง หรือการตั้งเวลา อีกทั้งจัดเก็บข้อมูลที่ได้จากมิเตอร์วัดพลังงาน ได้เลือกใช้โปรแกรม MySQL ซึ่งเป็น Freeware ที่สามารถจัดเก็บ ค้นหา เรียบเรียง หรือดึงข้อมูลได้ง่าย อีกทั้งผู้ใช้งานสามารถเข้าถึงข้อมูลได้อย่างรวดเร็วจากหลาย ๆ ที่ในเวลาเดียวกัน โปรแกรมยังช่วยเก็บข้อมูลอย่างเป็นระบบ ซึ่งใช้คำสั่ง SQL ในการเชื่อมต่อระหว่างผู้ใช้กับฐานข้อมูล ที่สำคัญสามารถทำงานบนระบบลินุกซ์ (Linux) ได้

3) โปรแกรม PHP5

ในการแปลภาษาเพื่อติดต่อกับ Database Server หรือ ประมวลผลและส่งข้อมูลให้กับเครื่องผู้ใช้ (Client) ได้เลือกใช้โปรแกรม PHP5 ที่สามารถดาวน์โหลดได้ฟรี โปรแกรมนี้สามารถทำงานบนระบบลินุกซ์ (Linux) ได้และโปรแกรมยังสามารถเขียนภาษาสคริปต์เพื่อสั่งการต่าง ๆ ได้ด้วยเช่นกัน

4) โปรแกรม phpMyAdmin

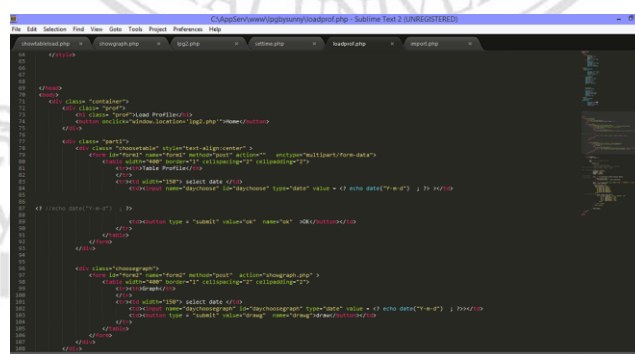
ในการจัดการข้อมูลใน MySQL ไม่ว่าจะเป็นการสร้าง/ลบฐานข้อมูล การจัดข้อมูลลงในตารางฐานข้อมูลนั้น ๆ และการหาผลสรุป เพื่อส่งผลไปยังโปรแกรมอื่นที่ร้องขอมา ได้เลือกใช้โปรแกรม phpMyAdmin มาช่วยให้สามารถเข้าถึงฐานข้อมูลผ่าน Web Browser ได้โดยตรงดังแสดงในรูปที่ 4.22 เพื่อให้สามารถจัดการฐานข้อมูล MySQL ได้สะดวกยิ่งขึ้นแทนการเขียนคำสั่งด้วยภาษา SQL

2) ผ่านหน้า Webpage

ในการสร้างโปรแกรมควบคุมพลังงานไฟฟ้าแสงสว่างให้ผู้ใช้งานสามารถควบคุมแสงสว่าง และดูข้อมูลต่าง ๆ ผ่านหน้า Webpage ได้ โดยในงานวิจัยนี้ได้เลือกใช้โปรแกรม AppServ ในการพัฒนา เนื่องจากโปรแกรมมีการรวบรวม Packages หลัก ได้แก่ Apache, PHP, MySQL และ phpMyAdmin ที่ช่วยจำลองเครื่องคอมพิวเตอร์ เป็น Web Server ได้ ซึ่งทำให้ง่ายต่อการเชื่อมต่อระหว่างฐานข้อมูล MySql เพื่อที่จะส่งข้อมูลคำสั่งเปิด/ปิดหลอดไฟไปเก็บในตารางและนำข้อมูลจากตารางกลับมาแสดงได้ ดังแสดงในรูปที่ 4.24



รูปที่ 4.23 โปรแกรม Gambas



รูปที่ 4.24 โปรแกรม AppServ

4.3 การพัฒนาและสร้างระบบจัดการพลังงานไฟฟ้าแสงสว่าง

ในการพัฒนาและออกแบบระบบจัดการพลังงานไฟฟ้าแสงสว่างสำหรับหลอดไฟ 2 ชนิด ได้แก่ ชนิดหลอดฟลูออเรสเซนต์และชนิดหลอด LED จะมีโครงสร้างหลัก 2 ส่วน

ส่วนแรก คือ Hardware จะเป็นส่วนที่มีความแตกต่างกัน เนื่องจากหลอดไฟชนิดหลอดฟลูออเรสเซนต์จะเป็นระบบควบคุมที่ใช้การควบคุมด้วยวิธีการเปิด/ปิดหลอดไฟเท่านั้นไม่สามารถปรับระดับแสงสว่างได้ (Switching Control: SCS) จึงใช้การควบคุมผ่านกล่องควบคุมการเปิด/ปิดหลอดไฟ (Switching Control Box) แต่หลอดไฟชนิดหลอด LED จะเป็นระบบที่สามารถเปิด/ปิดและปรับระดับแสงสว่างได้ (Dimming Control: DCS) ซึ่งจะใช้การควบคุมผ่านกล่องควบคุมการเปิด/ปิดและปรับหรี่หลอดไฟ (Dimming Control Box) นอกจากนี้ในส่วน Hardware ยังมีกล่องควบคุมระบบผ่านหน้าจอแบบสัมผัส (Touch Screen Box) ที่ให้ผู้ใช้สามารถใช้ในการควบคุมแสงสว่างจากหลอดไฟตามความต้องการของผู้ใช้อีกด้วย

เนื่องจากระบบ DCS ไม่เหมาะสมในระบบหลอดฟลูออเรสเซนต์ เพราะจะส่งผลให้อายุการใช้งานของหลอดไฟสั้นลง อีกทั้งบัลลาสต์อิเล็กทรอนิกส์ที่สามารถปรับระดับแสงได้ และมีประสิทธิภาพในการประหยัดพลังงานมีราคาที่สูง ทำให้ในงานวิจัยนี้ไม่ได้ทำการทดลองระบบหรี่ไฟในหลอดฟลูออเรสเซนต์ แต่อย่างไรก็ตามระบบ DCS เหมาะสมกับหลอดชนิด LED เนื่องจากหลอดชนิดนี้มีความสามารถในการปรับระดับแสงได้อย่างเต็มที่ ซึ่งปริมาณแสงที่ปรับส่งผลต่อการบริโภคพลังงานของหลอดไฟ (ปริมาณแสงมากใช้กำลังไฟฟ้ามาก เมื่อปริมาณแสงลดลงจะใช้กำลังไฟฟ้าลดลง) และหลอดชนิดนี้ยังมีอายุการใช้งานที่ยาวนาน ไม่มีผลกระทบต่อการทำงานการเปิด/ปิดหลอดไฟบ่อยครั้ง อีกทั้งหลอด LED ยังช่วยลดปริมาณแก๊สคาร์บอนไดออกไซด์ได้อีกด้วย

ส่วนที่ 2 ได้แก่ Software ในส่วนนี้ของหลอดไฟทั้ง 2 ชนิด จะมีการออกแบบ พัฒนา และสร้างที่คล้ายคลึงกัน ซึ่งจะแบ่งออกเป็น 4 ส่วน คือ ส่วนโปรแกรมที่พัฒนาขึ้นมาเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ (Microcontroller Program) ส่วนโปรแกรมที่พัฒนาขึ้นเพื่อสร้าง Web Server บน Raspberry Pi (Web Server Program) ส่วนโปรแกรมที่พัฒนาขึ้นเพื่อสร้างฐานข้อมูลของระบบ (Database Program) และส่วนที่พัฒนาขึ้นเป็นโปรแกรมควบคุมการใช้พลังงานไฟฟ้าแสงสว่าง (Lighting Control Program) ให้ผู้ใช้งานสามารถควบคุม/สั่งการและแสดงผลข้อมูลปริมาณทางไฟฟ้าของระบบผ่านจอ Touch Screen และหน้า Webpage โดยระบบมีการใช้โปรแกรม Gambus เป็นศูนย์กลางในการประมวลผลข้อมูลของระบบ (Operating Station) เพื่อส่งคำสั่งไปสั่งการให้

ไมโครคอนโทรลเลอร์ทำงานต่อไป โดยสามารถแสดงโครงสร้างทั้ง 2 ส่วนของระบบดังแสดงในรูปที่ 4.25 และ ในแต่ละส่วนมีรายละเอียดในการพัฒนาดังต่อไปนี้

4.3.1 การพัฒนาด้าน Hardware

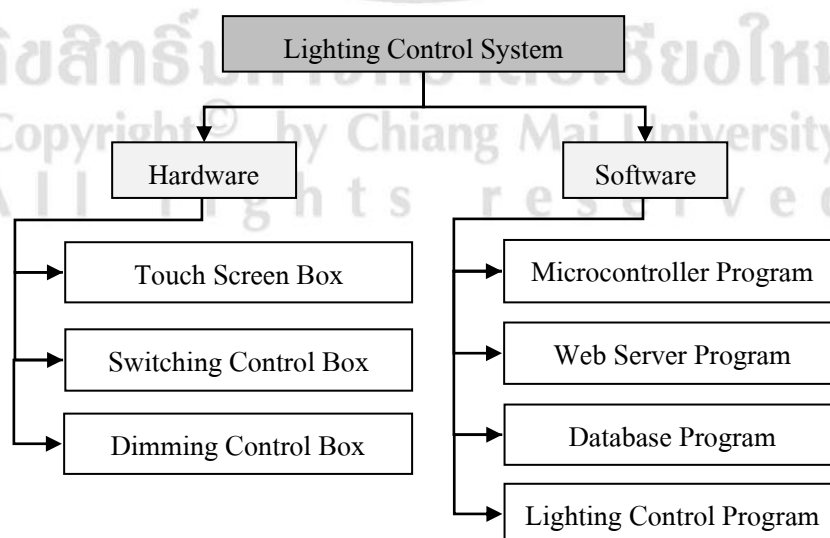
Hardware จะประกอบด้วย 3 ส่วนหลัก ได้แก่ กล้องควบคุมระบบผ่านหน้าจอแบบสัมผัส กล้องควบคุมการเปิด/ปิดหลอดไฟ และกล้องควบคุมการเปิด/ปิดและปรับหรี่หลอดไฟ ซึ่งแต่ละส่วนมีโครงสร้างและหลักการทำงาน ดังนี้

1) กล้องควบคุมระบบผ่านหน้าจอแบบสัมผัส (Touch Screen Box)

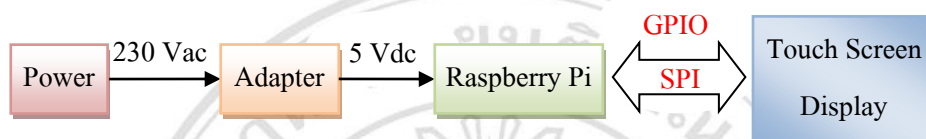
ส่วนประกอบของอุปกรณ์ที่ใช้ในกล้อง Touch Screen Box มีดังนี้

- 1) สาย Adapter 5 Vdc
- 2) Raspberry Pi
- 3) SD Card
- 4) จอ Touch Screen

โดย Raspberry Pi จะรับแรงดันไฟฟ้า 5 Vdc ที่แปลงออกมาจาก Adapter และจะเชื่อมต่อกับจอ Touch Screen ผ่านพอร์ตเนกประสงค์หรือพอร์ต GPIO (General Purpose Input/Output) บน Raspberry Pi ด้วยการสื่อสารรูปแบบ SPI (Serial Peripheral Interface) ซึ่งรูปที่ 4.26 แสดงถึงโครงสร้างของ Touch Screen Box

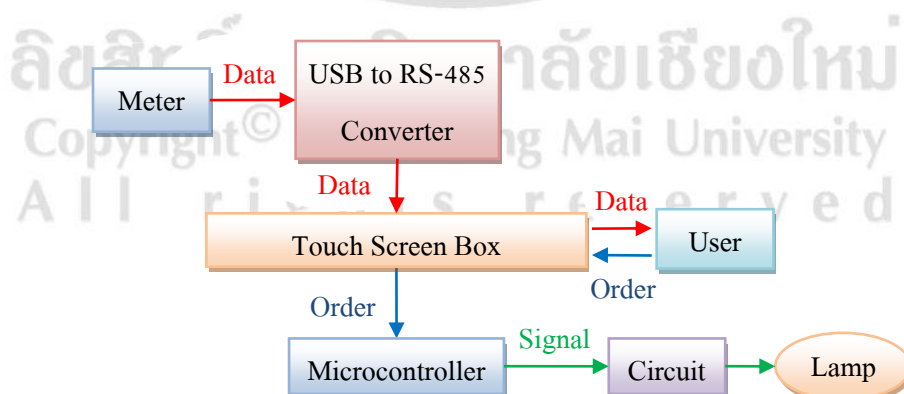


รูปที่ 4.25 โครงสร้างระบบจัดการพลังงานไฟฟ้าแสงสว่าง



รูปที่ 4.26 โครงสร้างส่วนประกอบกล่องควบคุมระบบผ่านหน้าจอแบบสัมผัส

กล่อง Touch Screen Box มีหลักการทำงานดังแสดงในรูปที่ 4.27 โดย Raspberry Pi จะรองรับคำสั่งควบคุมระบบจากผู้ใช้ที่สั่งผ่านจอ Touch Screen ที่อยู่บนกล่อง หรือทางหน้า Web Page จากคอมพิวเตอร์เครื่องอื่นที่เชื่อมต่ออยู่ในระบบ อีกทั้งยังเชื่อมต่อกับมิเตอร์วัดพลังงานเพื่อรับข้อมูลแรงดัน กระแส และกำลังไฟฟ้าผ่านการสื่อสารแบบ RS-485 ด้วยอุปกรณ์ USB to RS-485 Converter โดย Raspberry Pi จะจัดเก็บและประมวลผลข้อมูลทั้งหมด เพื่อสั่งการให้ไมโครคอนโทรลเลอร์ทำการส่งสัญญาณไปยังวงจรไฟฟ้าเพื่อควบคุมหลอดไฟ หรือแสดงผลข้อมูลที่ได้นับที่กไว้ในฐานข้อมูลผ่านทางโปรแกรมที่ได้พัฒนาขึ้น ซึ่งใช้งานได้กับทั้ง Switching Control Box และ Dimming Control Box



รูปที่ 4.27 หลักการทำงานของกล่องควบคุมระบบผ่านหน้าจอแบบสัมผัส

2.) กล่องควบคุมการเปิด/ปิดหลอดไฟ (Switching Control Box)

ส่วนประกอบของอุปกรณ์ที่ใช้ในกล่อง Switching Control Box มีดังนี้

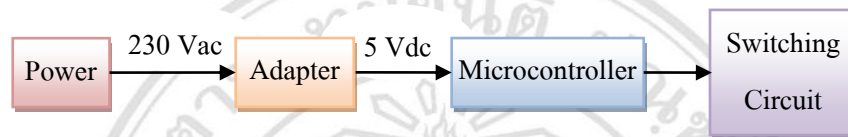
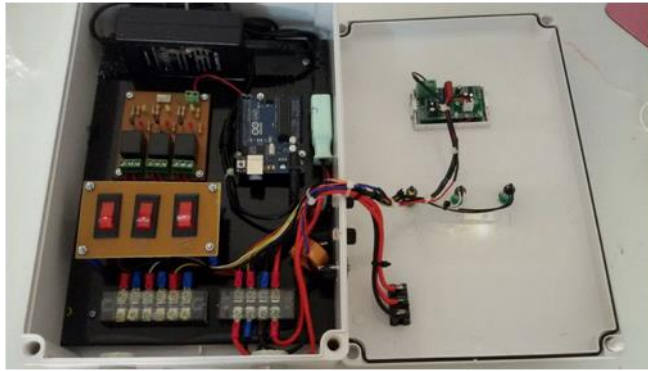
- 1) Adapter 5 Vdc
- 2) ไมโครคอนโทรลเลอร์
- 3) Switching Circuit

โดยไมโครคอนโทรลเลอร์จะทำงานจากการรับแรงดันไฟฟ้า 5 Vdc ที่แปลงออกมาจาก Adapter และจะควบคุมการเปิด/ปิดหลอดไฟด้วยพอร์ต Input/Output ผ่าน Switching Circuit ซึ่งสามารถแสดงโครงสร้างการทำงานได้ดังรูปที่ 4.28

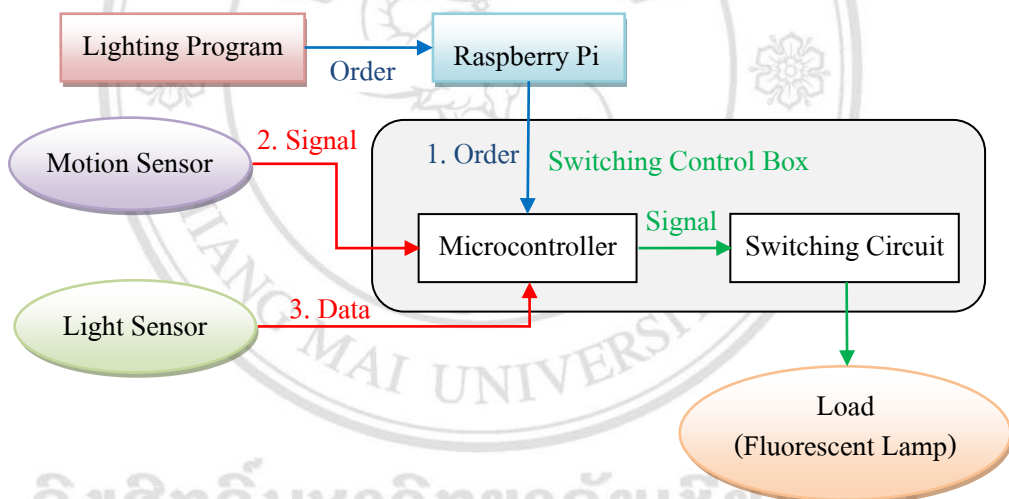
นอกจากนั้นจากรูปที่ 4.29 แสดงหลักการทำงานโดยรวมของ Switching Control Box โดยไมโครคอนโทรลเลอร์จะมีการรับข้อมูลจาก 3 ส่วน ได้แก่

- 1) รับคำสั่งการเปิด/ปิดหลอดไฟจากผู้ใช้โดยตรงหรือการกำหนดตารางเวลาทางโปรแกรมจัดการพลังงานไฟฟ้าแสงสว่างที่พัฒนาขึ้น (Lighting Program) จากนั้น Raspberry Pi ทำการประมวลผลแล้วส่งคำสั่งไปยังไมโครคอนโทรลเลอร์ผ่านการสื่อสารแบบอนุกรม (Serial Communication)
- 2) รับสัญญาณการตรวจจับการเคลื่อนไหวผ่านทางเซนเซอร์ตรวจจับการเคลื่อนไหวผ่านพอร์ต Input/Output บนไมโครคอนโทรลเลอร์
- 3) รับค่าปริมาณแสงสว่างผ่านทางเซนเซอร์วัดความเข้มแสงที่เชื่อมต่อแบบ I2C ผ่านพอร์ต Input/Output บนไมโครคอนโทรลเลอร์

จากนั้นไมโครคอนโทรลเลอร์จะส่งสัญญาณไปควบคุมการเปิด/ปิดหลอดไฟผ่าน Switching Circuit ตามชุดคำสั่งที่ได้โปรแกรมไว้



รูปที่ 4.28 โครงสร้างส่วนประกอบกล่องควบคุมการเปิด/ปิดไฟ



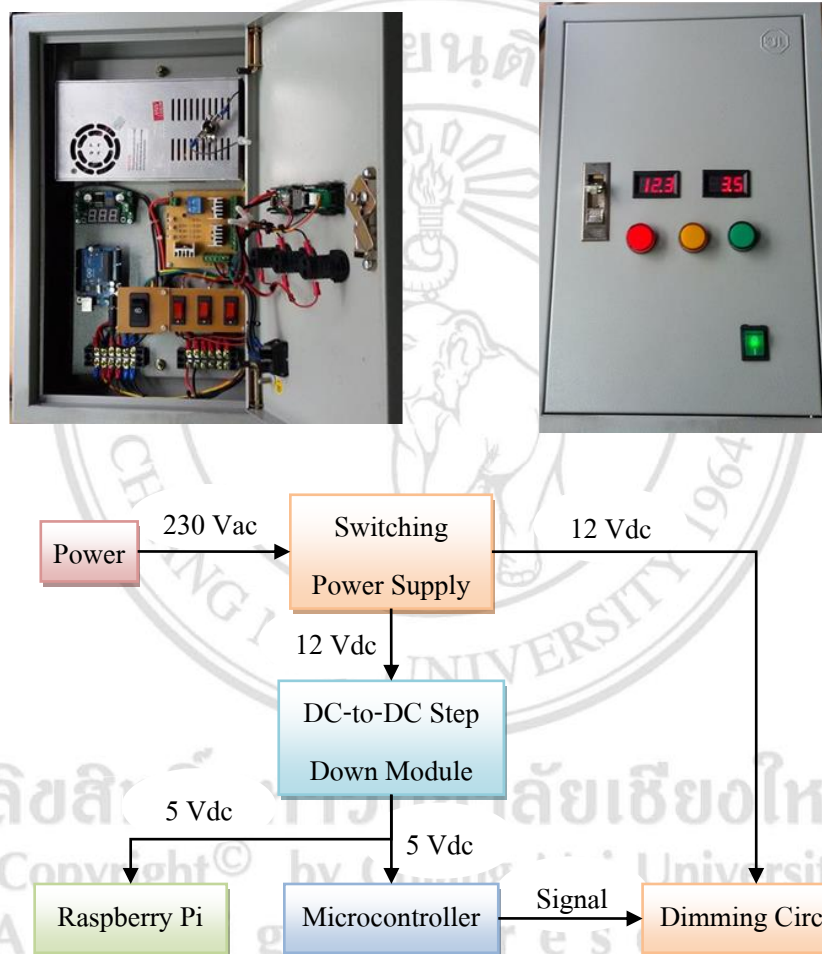
รูปที่ 4.29 หลักการทำงานของกล่องควบคุมการเปิด/ปิดไฟ

3.) กล่องควบคุมการเปิด/ปิดและปรับหรี่หลอดไฟ (Dimming Control Box)

ส่วนประกอบของอุปกรณ์ที่ใช้ในกล่อง Dimming Control Box มีดังนี้

- 1) Switching Power Supply 12 Vdc
- 2) DC-to-DC Step Down Module
- 3) ไมโครคอนโทรลเลอร์
- 4) Dimming Circuit

โดย Switching Power Supply ทำการจ่ายแรงดันให้กับหลอดไฟทั้งระบบ อีกทั้งยังจ่ายแรงดันให้กับไมโครคอนโทรเลอร์ และ Raspberry Pi โดยต้องผ่านการลดแรงดันลงด้วย DC-to-DC Step Down Module ให้เหลือ 5 Vdc ก่อน นอกจากนี้ไมโครคอนโทรเลอร์จะควบคุมการเปิด/ปิดและปรับหรี่หลอดไฟด้วยพอร์ต Input/Output บนไมโครคอนโทรเลอร์ผ่าน Dimming Circuit ซึ่งสามารถแสดงโครงสร้างการทำงานโดยรวมได้ดังรูปที่ 4.30

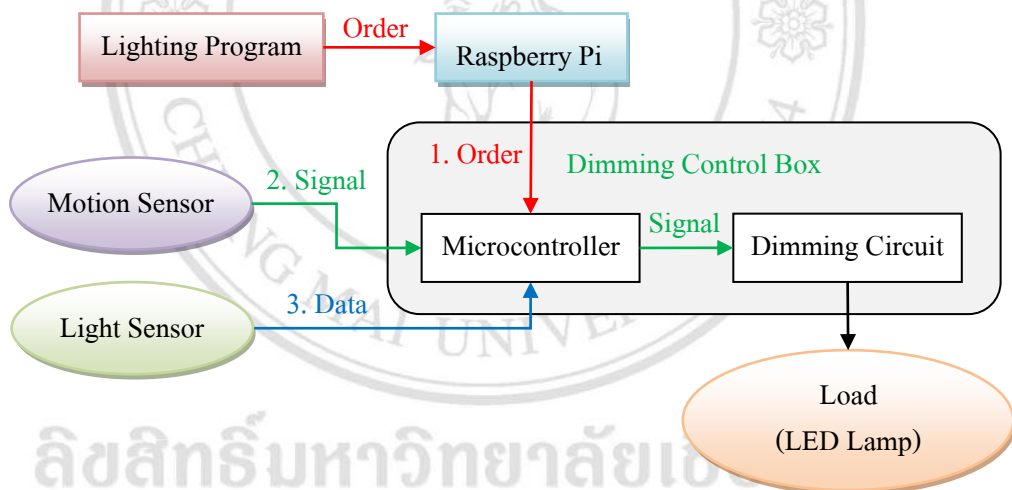


รูปที่ 4.30 โครงสร้างส่วนประกอบกล่องควบคุมการเปิด/ปิดและปรับหรี่หลอดไฟ

โดย Dimming Control Box มีหลักการทำงานรวมดังแสดงในรูปที่ 4.31 ซึ่งไมโครคอนโทรลเลอร์จะมีการรับข้อมูลจาก 3 ส่วน ได้แก่

- 1) รับคำสั่งการเปิด/ปิดหลอดไฟจากผู้ใช้งาน Lighting Program จากนั้น Raspberry Pi ทำการประมวลผลแล้วส่งคำสั่งไปยังไมโครคอนโทรลเลอร์ผ่านการสื่อสารแบบอนุกรม
- 2) รับสัญญาณการตรวจจับการเคลื่อนไหวผ่านทางเซนเซอร์ตรวจจับการเคลื่อนไหวผ่านพอร์ต Input/Output บนไมโครคอนโทรลเลอร์
- 3) รับค่าปริมาณแสงสว่างผ่านทางเซนเซอร์วัดความเข้มแสง (Light Sensor) ที่เชื่อมต่อแบบ I2C ผ่านพอร์ต Input/Output บนไมโครคอนโทรลเลอร์

จากนั้นไมโครคอนโทรลเลอร์จะส่งสัญญาณไปควบคุมการเปิด/ปิดหลอดไฟผ่านวงจร Dimming Circuit ตามชุดคำสั่งที่ได้โปรแกรมไว้



รูปที่ 4.31 หลักการทำงานกล่องควบคุมการเปิด/ปิดและปรับหรี่หลอดไฟ

4.3.2 การพัฒนาด้าน Software

ด้าน Software ของทั้งระบบ SCS และ DCS จะมีการพัฒนาและออกแบบที่คล้ายคลึงกัน โดยจะแบ่งออกเป็น 4 ส่วนหลัก ประกอบด้วย

- 1) ส่วนที่พัฒนาขึ้นเพื่อควบคุมการทำงานของส่วนฮาร์ดแวร์ หรือ ควบคุมการทำงานของไมโครคอนโทรลเลอร์ (Microcontroller Program)
- 2) ส่วนที่พัฒนาขึ้นเพื่อสร้าง Web Server บน Raspberry Pi (Web Server Program)
- 3) ส่วนที่พัฒนาขึ้นเป็นโปรแกรมควบคุมพลังงานไฟฟ้าแสงสว่าง (Lighting Control Program)
- 4) ส่วนที่พัฒนาขึ้นเพื่อเก็บข้อมูลต่าง ๆ บนฐานข้อมูล MySQL (Database Program)

ซึ่งทั้ง 4 ส่วนมีรายละเอียดของวิธีและโครงสร้างของการพัฒนารวมถึงหลักการทำงาน ดังนี้

- วิธีและโครงสร้างการพัฒนาซอฟต์แวร์

การพัฒนาด้านซอฟต์แวร์ของทั้ง 4 ส่วนจะมีโปรแกรมที่ใช้ในการพัฒนาที่แตกต่างกัน ดังแสดงในรูปที่ 4.33 ทำให้ภาษาที่ใช้ในการพัฒนาก็จะแตกต่างกันไปด้วย โดยโปรแกรมที่เลือกใช้ในงานวิจัยนี้จะเป็นโปรแกรมที่สำเร็จรูป ซึ่งมีการรวม packages ที่จำเป็นอยู่ในโปรแกรมนั้นแล้ว ทำให้ไม่ยุ่งยากต่อการติดตั้งใช้งาน และที่สำคัญโปรแกรมเหล่านี้ไม่มีค่าใช้จ่ายในการใช้งาน ซึ่งมีรายละเอียดของแต่ละส่วน ดังต่อไปนี้

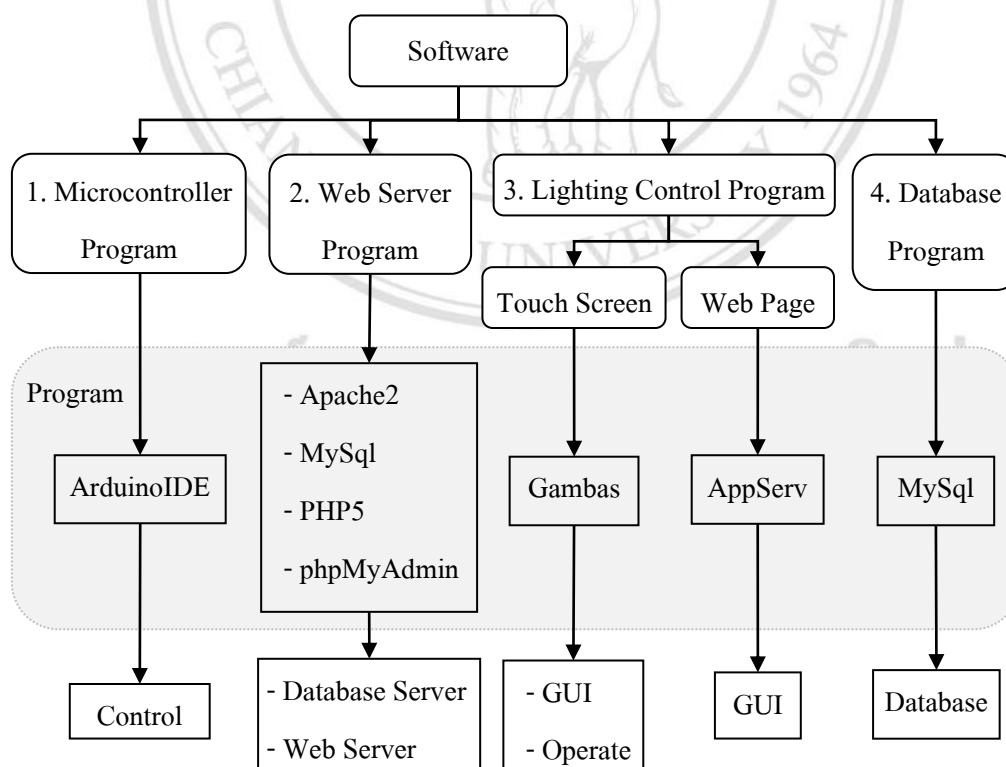
1) Microcontroller Program

ในการโปรแกรมเพื่อควบคุมการทำงานให้ไมโครคอนโทรลเลอร์ หรือการสร้างชุดคำสั่งเพื่อสั่งการให้วงจรอิเล็กทรอนิกส์ทำงาน จะใช้โปรแกรม ArduinoIDE ในการโปรแกรมชุดคำสั่งด้วยภาษา C หรือ C++

2) Web Server Program

ในส่วนนี้เป็นการสร้าง Web Server บน Raspberry Pi ซึ่งมีขั้นตอนดังต่อไปนี้

- 2.1) ติดตั้งโปรแกรม Apache ที่ใช้สำหรับระบบปฏิบัติการ Linux เพื่อให้ Raspberry Pi ทำหน้าที่เป็น Web Server
- 2.2) ติดตั้งโปรแกรม PHP5 เพื่อให้ Raspberry Pi สามารถแปลคำสั่ง Script PHP ที่เป็นชุดคำสั่งในการติดต่อกับ Database Server
- 2.3) ติดตั้งโปรแกรม MySQL เพื่อให้ Raspberry pi สามารถให้บริการหรือรองรับฐานข้อมูล MySQL ได้ กล่าวได้ว่าทำหน้าที่เป็น Database Server
- 2.4) ติดตั้งโปรแกรม PHPMyAdmin เพื่อให้สามารถเข้าไปจัดการฐานข้อมูลใน MySQL ผ่าน Web Browser ได้



รูปที่ 4.32 โครงสร้างการพัฒนาด้าน Software

3) Lighting Control Program

ผู้ใช้งานสามารถควบคุมการใช้ไฟฟ้าด้วย Lighting Program ได้ 2 รูปแบบ คือ ผ่านทางจอ Touch Screen และ Webpage ซึ่งทั้งสองส่วนมีการพัฒนา ดังนี้

3.1) โปรแกรมจัดการพลังงานไฟฟ้าแสงสว่างที่ควบคุมผ่านทางจอ Touch Screen (Touch Screen Lighting Program)

โปรแกรมจะถูกพัฒนามาจากโปรแกรม Gambas ด้วยภาษา Basic ที่ติดตั้งบน Raspberry Pi ซึ่งใช้ในการพัฒนาอยู่ 2 ส่วน คือ การออกแบบหน้าจอโปรแกรม (Graphical User Interface, GUI) เพื่อให้ผู้ใช้งานโปรแกรมได้ง่ายขึ้น และส่วนประมวลผลการทำงานของระบบ

3.2) โปรแกรมจัดการพลังงานไฟฟ้าแสงสว่างที่ควบคุมผ่านทางหน้า Webpage (Web Lighting Program)

ในการออกแบบหน้าจอเพื่อให้ผู้ใช้งานสามารถใช้งานโปรแกรมผ่านทาง Webpage ได้เลือกใช้โปรแกรม AppServ ซึ่งเขียนด้วยภาษา HTML, CSS, PHP และ JavaScript ในการออกแบบรูปแบบโปรแกรมเพื่อให้ผู้ใช้งานเข้าใช้งาน

4) Database Program

ในการเก็บข้อมูลในงานวิจัยทั้งหมดจะเก็บบนฐานข้อมูลของโปรแกรม MySQL ซึ่งมีวิธีการสร้างฐานข้อมูลดังนี้

- 4.1) สร้างฐานข้อมูลเพื่อเก็บตารางข้อมูลที่ใช้นับทิกข้อมูลในแต่ละส่วน ดังแสดงในรูปที่ 4.33



ตาราง	ประเภทการ	ระเบียน	ชนิด	การเรียงลำดับ	ขนาด	เก็บความจำเป็น
lamp		1	MyISAM	utf8_general_ci	1.0 กิโลไบต์	-
loadprofile		104	MyISAM	utf8_general_ci	9.0 กิโลไบต์	-
member		2	MyISAM	utf8_general_ci	2.1 กิโลไบต์	-
settime		7	MyISAM	utf8_unicode_ci	2.3 กิโลไบต์	-
4 ตาราง	ผลรวม	114	MyISAM	utf8_general_ci	14.4 กิโลไบต์	0 ไบต์

รูปที่ 4.33 ฐานข้อมูลที่ใช้เก็บตารางข้อมูล

- 4.2) สร้างตาราง 4 ตารางที่ใช้เก็บข้อมูลของระบบแต่ละส่วนซึ่งในงานวิจัยนี้มีการเก็บข้อมูลอยู่ 4 ส่วน ได้แก่

- 1) ตารางในการเก็บข้อมูลของคำสั่งเปิด/ปิดไฟจากผู้ใช้ผ่านทางหน้า Webpage และคำสั่งการใช้กำลังไฟฟ้าเกินขีดจำกัด ดังแสดงในรูปที่ 4.34 ซึ่งมีฟิลด์ในตาราง ดังนี้

- ฟิลด์ id ใช้ระบุจำนวนข้อมูล เพื่อง่ายต่อการเรียกใช้ข้อมูล
- ฟิลด์ zone ต่าง ๆ โดยจะมีจำนวนเท่ากับบริเวณที่ควบคุม ทำหน้าที่รองรับคำสั่งเปิด/ปิดหลอดไฟจากผู้ใช้
- ฟิลด์ alarmoff ใช้เก็บข้อมูลการใช้กำลังไฟฟ้าเกินขีดจำกัด

2) ตารางที่ใช้เก็บข้อมูลคำสั่งเปิด/ปิดไฟตามตารางเวลา มีหน้าที่เก็บข้อมูลที่ใช้กำหนดการเปิด/ปิดหลอดไฟล่วงหน้า ดังแสดงในรูปที่ 4.35 ซึ่งมีฟิลด์ในตาราง 4 ฟิลด์ ได้แก่

- ฟิลด์ id ใช้ระบุจำนวนข้อมูล
- ฟิลด์ที่ใช้เก็บ วัน/เดือน/ปี ที่ต้องการสั่งการเปิด/ปิดหลอดไฟ
- ฟิลด์ที่ใช้เก็บเวลาที่ต้องการสั่งการเปิด/ปิดหลอดไฟ
- ฟิลด์ที่ใช้เก็บข้อมูลคำสั่งเปิด/ปิดหลอดไฟ

3) ตารางที่ใช้เก็บข้อมูลสมาชิก เพื่อเก็บข้อมูลสมาชิกที่สามารถเข้าใช้โปรแกรมได้ ดังแสดงในรูปที่ 4.36 ซึ่งมีฟิลด์ในตาราง 4 ฟิลด์ ได้แก่

- ฟิลด์ id ใช้ระบุจำนวนข้อมูล
- ฟิลด์ที่ใช้เก็บชื่อสมาชิก
- ฟิลด์ที่ใช้เก็บรหัสผ่านของสมาชิก
- ฟิลด์ที่ใช้เก็บ E-mail ของสมาชิก

4) ตารางที่ใช้เก็บข้อมูล แรงดัน กระแส และกำลังไฟฟ้า ที่วัดได้จากมิเตอร์ มีหน้าที่เก็บข้อมูลการใช้ไฟฟ้าย้อนหลังได้ ดังแสดงในรูปที่ 4.37 ซึ่งมีฟิลด์ในตาราง 6 ฟิลด์ ได้แก่

- ฟิลด์ id ใช้ระบุจำนวนข้อมูล
- ฟิลด์ที่ใช้เก็บ วัน/เดือน/ปี ของการบันทึกข้อมูล
- ฟิลด์ที่ใช้เก็บเวลาในการบันทึกข้อมูล
- ฟิลด์ที่ใช้เก็บค่าแรงดันไฟฟ้า
- ฟิลด์ที่ใช้เก็บค่ากระแสไฟฟ้า
- ฟิลด์ที่ใช้เก็บค่ากำลังไฟฟ้า

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: lighting ▶ ตาราง : lamp

	id	zone1	zone2	zone3	alartoff
☐	1	OFF	ON	OFF	0

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก:   

รูปที่ 4.34 ตารางที่ใช้เก็บข้อมูลคำสั่งเปิด/ปิดหลอดไฟจากผู้ใช้ผ่านทางหน้า Webpage

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: lighting ▶ ตาราง : settime

เปิดดู โครงสร้าง SQL ค้นหา ว่างแรก ส่งออก Import กระบวนการ ลบข้อมูล โยนทิ้ง

	ฟิลด์	ชนิด	การเรียงลำดับ	แอตทริบิวต์	ว่างเปล่า (null)	ค่าปริยาย	เพิ่มเติม	กระทำการ
☐	id	int(11)			ไม่		auto_increment	    
☐	dateset	varchar(20)	utf8_unicode_ci		ไม่			    
☐	timeset	varchar(20)	utf8_unicode_ci		ไม่			    
☐	doing	varchar(20)	utf8_unicode_ci		ไม่			    

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก:     

รูปที่ 4.35 ตารางที่ใช้เก็บข้อมูลคำสั่งเปิด/ปิดหลอดไฟตามตารางเวลา

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: lighting ▶ ตาราง : member

เปิดดู โครงสร้าง SQL ค้นหา ว่างแรก ส่งออก Import กระบวนการ ลบข้อมูล โยนทิ้ง

	ฟิลด์	ชนิด	การเรียงลำดับ	แอตทริบิวต์	ว่างเปล่า (null)	ค่าปริยาย	เพิ่มเติม	กระทำการ
☐	id	int(11)		UNSIGNED ZEROFILL	ไม่		auto_increment	    
☐	password	varchar(10)	utf8_general_ci		ไม่			    
☐	username	varchar(20)	utf8_general_ci		ไม่			    
☐	emailaddress	varchar(50)	utf8_general_ci		ไม่			    

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก:     

รูปที่ 4.36 ตารางที่ใช้เก็บข้อมูลสมาชิก

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: lighting ▶ ตาราง : loadprofile

เปิดดู โครงสร้าง SQL ค้นหา ว่างแรก ส่งออก Import กระบวนการ ลบข้อมูล โยนทิ้ง

	ฟิลด์	ชนิด	การเรียงลำดับ	แอตทริบิวต์	ว่างเปล่า (null)	ค่าปริยาย	เพิ่มเติม	กระทำการ
☐	id	int(11)			ไม่		auto_increment	    
☐	dayin	varchar(30)	utf8_general_ci		ไม่			    
☐	timein	varchar(30)	utf8_general_ci		ไม่			    
☐	power	varchar(30)	utf8_general_ci		ไม่			    
☐	voltage	varchar(30)	utf8_general_ci		ไม่			    
☐	current	varchar(30)	utf8_general_ci		ไม่			    

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก:     

รูปที่ 4.37 ตารางที่ใช้เก็บข้อมูล แรงดัน กระแส และ กำลังไฟฟ้า

- หลักการทำงานในส่วนซอฟต์แวร์

การทำงานของระบบจัดการพลังงานไฟฟ้าแสงสว่างจะควบคุมด้วยโปรแกรม Gambas เป็นโปรแกรมหลักหรือเปรียบได้เป็น Operating Station ซึ่งมีหน้าที่ดังนี้

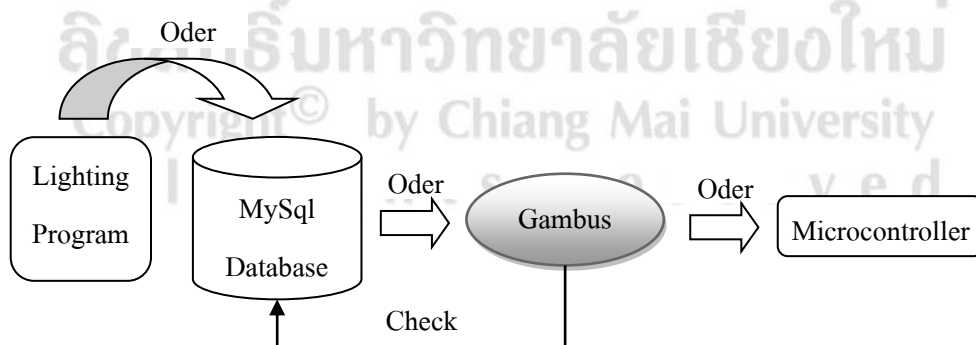
1) โปรแกรม Gambus จะเชื่อมต่อกับฐานข้อมูล Mysql เพื่อทำหน้าที่หลัก 3 ส่วน ได้แก่

1.1) ตรวจเช็คคำสั่งการเปิด/ปิดหลอดไฟจากผู้ใช้ผ่านหน้า Webpage

โดยเมื่อผู้ใช้สั่งเปิด/ปิดหลอดไฟด้วยโปรแกรมผ่านทางหน้า Webpage คำสั่งจะส่งไปเก็บบนฐานข้อมูล MySQL ในขณะเดียวกัน โปรแกรม Gambus จะทำการตรวจเช็คคำสั่งบนฐานข้อมูลตลอดเวลา เพื่อรอที่จะส่งให้ไมโครคอนโทรลเลอร์ทำงานต่อไปดังแสดงในรูปที่ 4.38

1.2) ตรวจเช็คสั่งการเปิด/ปิดหลอดไฟที่ได้กำหนด วัน/เดือน/ปี และเวลาไว้ล่วงหน้า

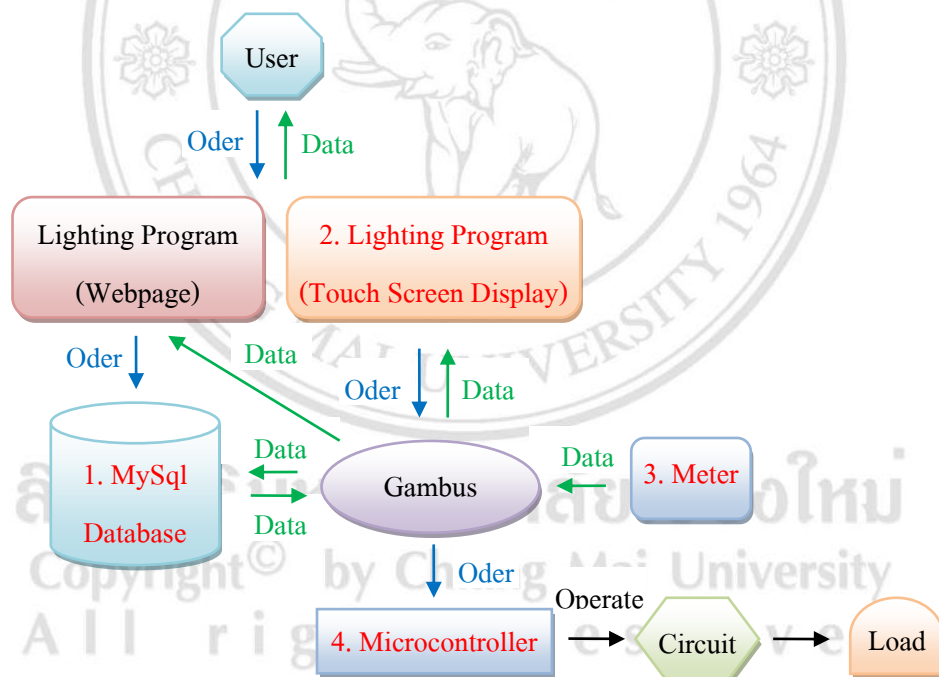
1.3) เพื่อที่จะดึงข้อมูลการใช้ไฟฟ้าย้อนหลังถูกที่ใช้ในระบบมาแสดงผ่านโปรแกรมจัดการพลังงานไฟฟ้าแสงสว่าง



รูปที่ 4.38 การทำงานเมื่อมีการสั่งการผ่านหน้า Webpage

- 2) โปรแกรม Gambus จะรับคำสั่งเปิด/ปิดหลอดไฟจากผู้ใช้โดยตรงด้วยโปรแกรมที่พัฒนาขึ้นผ่านทางจอ Touch Screen จากนั้นจะประมวลผลและส่งคำสั่งไปสั่งการให้ไมโครคอนโทรลเลอร์ทำงาน
- 3) โปรแกรม Gambus จะรับข้อมูลของปริมาณทางไฟฟ้าจากมิเตอร์ ผ่านการสื่อสารแบบ RS 485 ด้วยอุปกรณ์ USB to RS-485 Converter จากนั้นจะส่งข้อมูลไปเก็บไว้ที่ตารางที่สร้างได้ไว้บนฐานข้อมูล Mysql
- 4) โปรแกรม Gambus จะคอยส่งคำสั่งในการควบคุมหลอดไฟไปให้ไมโครคอนโทรลเลอร์ทำงาน

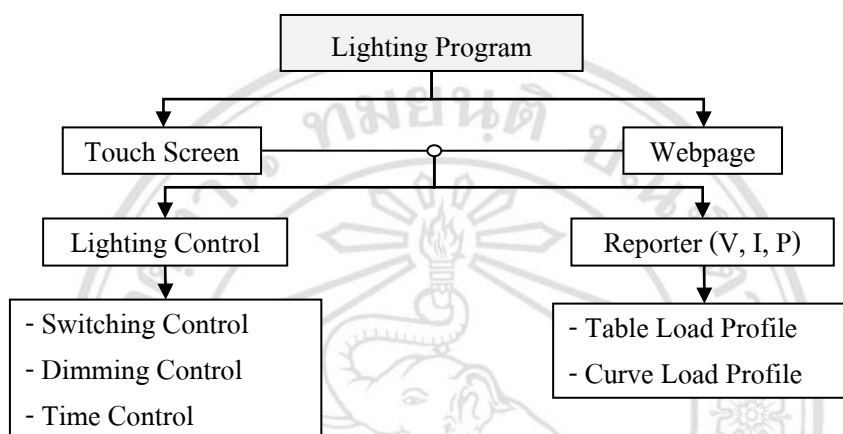
โดยสามารถแสดงหลักการทำงานของ Software ทั้งระบบซึ่งมีโปรแกรม Gambus ทำหน้าที่เป็น Operation Station ได้ดังรูปที่ 4.39



รูปที่ 4.39 หลักการทำงานของซอฟต์แวร์ของระบบ

4.3.3 โปรแกรมจัดการพลังงานในระบบแสงสว่าง (Lighting Program)

โปรแกรมจัดการพลังงานในระบบแสงสว่างที่ได้พัฒนาขึ้นในงานวิจัยนี้จะมีอยู่ 2 รูปแบบ คือ โปรแกรมจัดการพลังงานในระบบแสงสว่างที่ใช้งานผ่านจอ Touch Screen และใช้งานผ่านหน้า Webpage โดยแบ่งการทำงานเป็น 2 ส่วน ได้แก่ ส่วนควบคุม (Lighting Control) และส่วนรายงานผล (Reporter) ดังแสดงในรูปที่ 4.40



รูปที่ 4.40 โครงสร้างการออกแบบโปรแกรมจัดการพลังงานในระบบแสงสว่าง

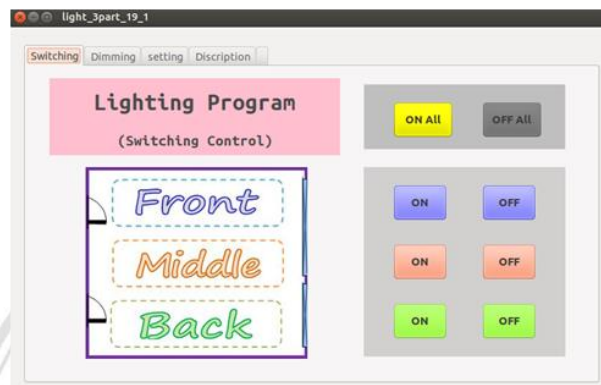
ในส่วนแรก คือ ส่วนควบคุมการเปิด/ปิดระบบแสงสว่าง ในส่วนนี้ช่วยให้ผู้ใช้สามารถบริหารจัดการ ควบคุม และวางแผนการใช้ไฟฟ้าแสงสว่างได้อย่างมีประสิทธิภาพ ซึ่งมีคุณสมบัติ ดังนี้

- ผู้ใช้สามารถควบคุมเปิด/ปิดและปรับระดับแสงสว่างได้ตามความต้องการผ่านโปรแกรมโดยตรง
- ผู้ใช้สามารถกำหนดตารางเวลาในการสั่งเปิด/ปิดหลอดไฟในระบบแสงสว่างได้
- โปรแกรมสามารถแจ้งเตือนเมื่อระบบมีการใช้พลังงานไฟฟ้าเกินที่กำหนดไว้

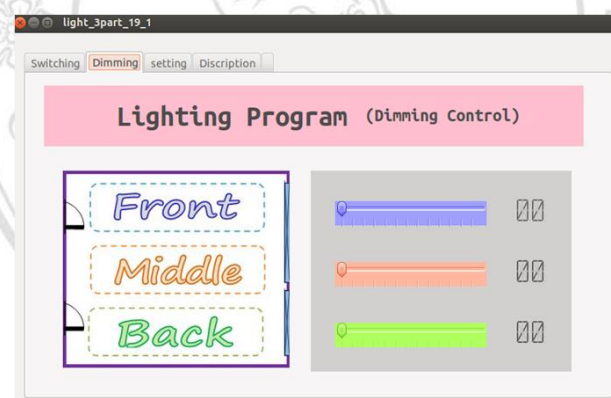
ในอีกส่วน คือ ส่วนแสดงผล ในส่วนนี้ใช้แสดงผลข้อมูลทางไฟฟ้าที่ระบบใช้งาน เช่น แรงดัน (V), กระแส (I) และกำลังไฟฟ้า (P) ซึ่งสามารถแสดงข้อมูลย้อนหลังได้ทั้งในรูปแบบตารางข้อมูล (Table Load Profile) และรูปแบบกราฟ (Curve Load Profile)

1) โปรแกรมจัดการพลังงานไฟฟ้าแสงสว่างที่ใช้งานผ่านจอ Touch Screen

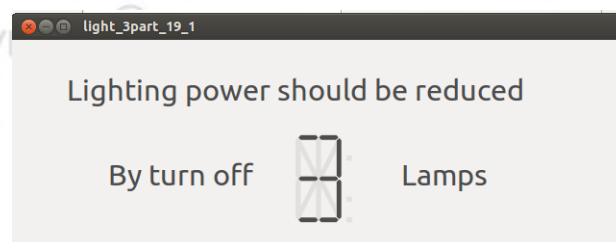
1.1) ส่วนควบคุมการเปิด/ปิดและปรับระบบแสงสว่าง (Lighting Control)
ดังแสดงในรูปที่ 4.41, 4.42, 4.43 และ 4.44



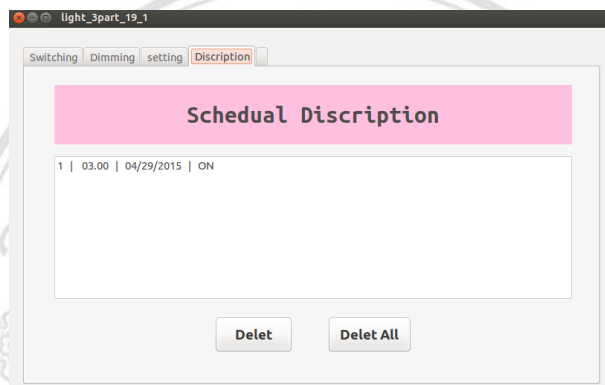
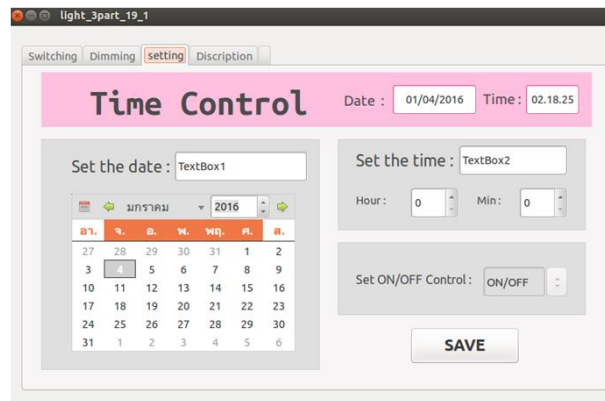
รูปที่ 4.41 ส่วนโปรแกรมที่ใช้ควบคุมการเปิด/ปิดระบบแสงสว่างผ่านหน้าจอแบบสัมผัส



รูปที่ 4.42 ส่วนโปรแกรมที่ใช้ควบคุมการเปิด/ปิดและปรับระบบแสงสว่างผ่านหน้าจอแบบสัมผัส



รูปที่ 4.43 ส่วนโปรแกรมที่ใช้แจ้งเตือนให้ผู้จัดการใช้พลังงานลงผ่านหน้าจอแบบสัมผัส



รูปที่ 4.44 ส่วนโปรแกรมที่ใช้ควบคุมการเปิด/ปิดแสงสว่างตามตารางเวลาผ่านหน้าจอแบบสัมผัส

- 1.2) ส่วนแสดงผลซึ่งใช้แสดงผลข้อมูลทางไฟฟ้าที่ระบบใช้งาน (Reporter)
 ดังแสดงในรูปที่ 4.45, 4.46 และ 4.47



รูปที่ 4.45 ส่วนโปรแกรมที่ใช้ดูข้อมูลการใช้ไฟฟ้าผ่านหน้าจอแบบสัมผัส

light_3part_19_1

Main Select

Load Profile Data

Select Date : MM/DD/YY

Show Table Data : View

Select Electrical Data : P, I, V, pf, View

Show curve : View

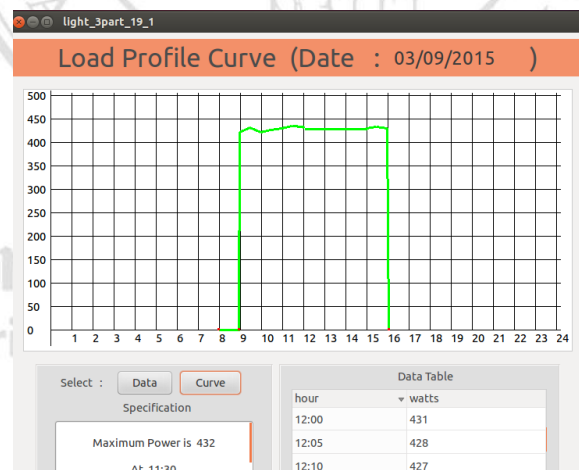
Load Profile Table (Date : 03/09/2015)

hour	voltage	current	frequency	pf	watts	appear
00:15	230	3	50	1	8	690
00:30	230	2	50	1	6	670
00:45	230	1	50	1	6	600
01:00	230	1	50	1	6	600
02:00	230	7	50	0.85	4	550
02:45	230	2	50	0.8	9	540

Specification

Max	voltage	current	frequency	pf	watts	appear
	230	9	50	1	9	800

รูปที่ 4.46 ส่วน โปรแกรมที่ใช้ดูข้อมูลการใช้ไฟฟ้าในรูปแบบตารางผ่านจอแบบสัมผัส



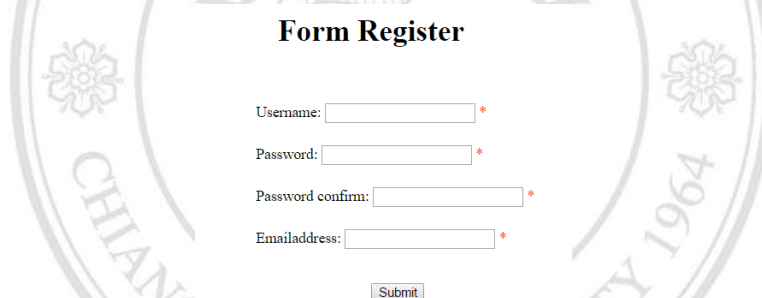
รูปที่ 4.47 ส่วน โปรแกรมที่ใช้ดูข้อมูลการใช้ไฟฟ้าในรูปแบบกราฟผ่านจอแบบสัมผัส

2) โปรแกรมจัดการพลังงานไฟฟ้าแสงสว่างที่ใช้งานผ่านหน้า Webpage

ในการเข้าใช้งานผ่านหน้า Webpage ผู้ใช้ต้องมีการ Login ด้วยชื่อและรหัสผ่านก่อนเข้าใช้งานทุกครั้ง ดังนั้นผู้ใช้จะต้องมีการสมัครสมาชิก (Registration) ก่อนเข้าใช้โปรแกรมดังแสดงในรูปที่ 4.48 และ 4.49



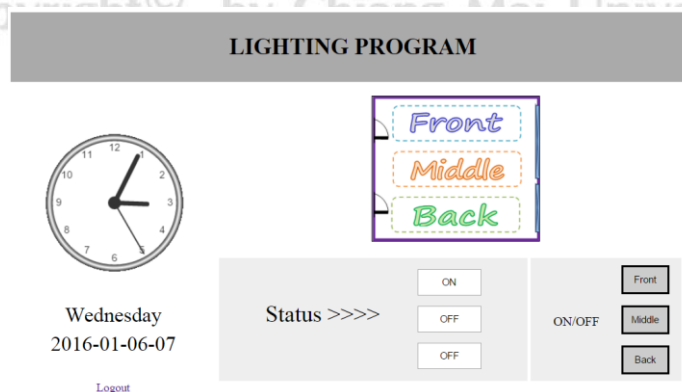
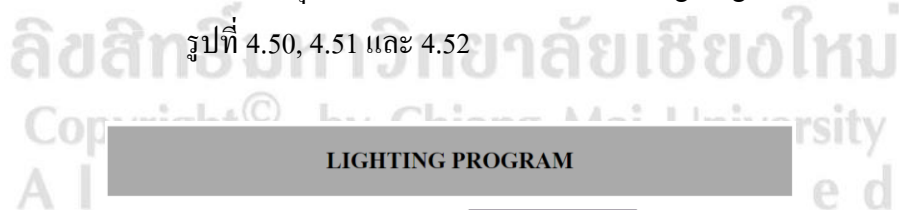
รูปที่ 4.48 ส่วนโปรแกรมที่ใช้ Login เข้าใช้งานผ่านหน้า Webpage



รูปที่ 4.49 ส่วนโปรแกรมที่ใช้เพื่อสมัครสมาชิกผ่านหน้า Webpage

2.1) ส่วนควบคุมการเปิด/ปิดระบบแสงสว่าง (Lighting Control) ดังแสดงใน

รูปที่ 4.50, 4.51 และ 4.52



รูปที่ 4.50 ส่วนโปรแกรมที่ใช้แสดงควบคุมการเปิด/ปิดระบบแสงสว่างผ่านหน้า Webpage

Time Control

Schedual Discription					
ID	DATE	TIME	DOING	Delete	Edit
2	2015-12-07	03:02	ON	Delet	Edit
3	2015-12-07	03:02	OFF	Delet	Edit
4	2015-12-23	03:02	ON	Delet	Edit
5	2015-12-08	14:01	OFF	Delet	Edit
6	2015-12-14	14:01	OFF	Delet	Edit
8	2015-12-25	15:01	OFF	Delet	Edit

Setting

Date (yyyy-mm-dd)	ดด/ว/ปปปป
time (00.00)	--:-- --
ON/OFF	ON/OFF
Save Back	

Edit data 10

Date (yyyy-mm-dd)	ดด/ว/ปปปป
time (00.00)	--:-- --
ON/OFF	
Add Back	

รูปที่ 4.51 ส่วนโปรแกรมที่ใช้ควบคุมการเปิด/ปิดแสงสว่างตามตารางเวลาผ่านหน้า Webpage

Lighting power should be reduced

By turn off 3 lamps

รูปที่ 4.52 ส่วนโปรแกรมที่ใช้แจ้งเตือนให้ผู้จัดการใช้พลังงานลงผ่านหน้า Webpage

2.2) ส่วนแสดงผลซึ่งใช้แสดงผลข้อมูลทางไฟฟ้าที่ระบบใช้งาน (Reporter)

ดังแสดงในรูปที่ 4.53, 4.54 และ 4.55

Lighting Program (Reporter)

Load Profile (Data)	Load Profile (Curve)
Select the date 01/06/2016 OK	Select the date 01/06/2016 Curve

รูปที่ 4.53 ส่วนโปรแกรมที่ใช้ดูข้อมูลการใช้ไฟฟ้าผ่าน Webpage

Lighting Program (Reporter)						
Home						
Load Profile (Data)				Load Profile (Curve)		
Select the date		01/06/2016		Select the date		01/06/2016
		OK				Curve
1244	2015-03-29	1:50:00 PM	280.0	230.4	2.3	
1261	2015-03-29	1:55:00 PM	275.4	230.4	2.3	
1278	2015-03-29	2:00:00 PM	278.5	230.4	2.3	
1295	2015-03-29	2:05:00 PM	132.3	230.4	2.3	
1312	2015-03-29	2:10:00 PM	131.7	230.4	1.3	
1329	2015-03-29	2:15:00 PM	432.4	230.4	1.3	
1346	2015-03-29	2:20:00 PM	279.0	230.4	1.3	

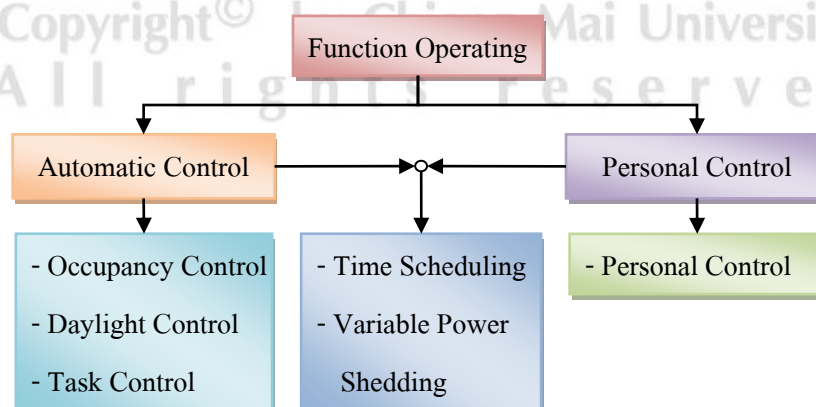
รูปที่ 4.54 ส่วนโปรแกรมที่ใช้ดูข้อมูลการใช้ไฟฟ้าในรูปแบบตารางผ่าน Webpage



รูปที่ 4.55 ส่วนโปรแกรมที่ใช้ดูข้อมูลการใช้ไฟฟ้าในรูปแบบกราฟผ่าน Webpage

4.4 การพัฒนาและสร้างฟังก์ชันการทำงาน

ฟังก์ชันการทำงานหลักทั้ง 6 ฟังก์ชัน จะมีรูปแบบการทำงาน (Function Operating) อยู่ 2 รูปแบบ ได้แก่ การทำงานแบบอัตโนมัติ (Automatic Control) และการทำงานด้วยการควบคุมสั่งการผ่านผู้ใช้ (Personal Control) โดยแต่ละฟังก์ชันสามารถแยกรูปแบบการทำงานได้ดังแสดงในรูปที่ 4.56 ซึ่งจะพบว่าระบบโดยส่วนใหญ่จะทำงานแบบอัตโนมัติผ่านทางเซนเซอร์และไมโครคอนโทรลเลอร์ โดยแต่ละฟังก์ชันมีรายละเอียดของการพัฒนาและสร้างระบบ ดังต่อไปนี้



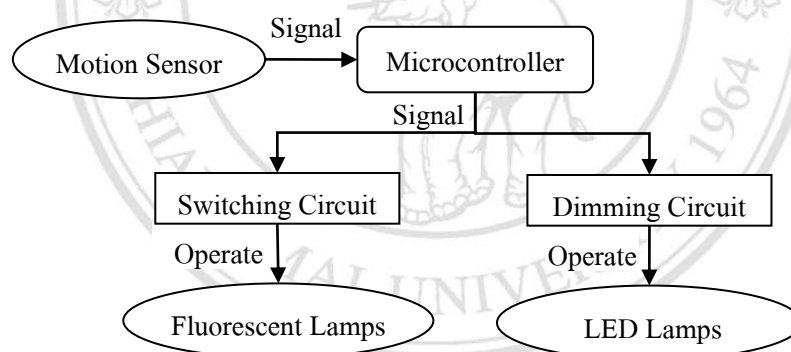
รูปที่ 4.56 รูปแบบการทำงานของแต่ละฟังก์ชัน

4.4.1. ฟังก์ชันตรวจจับการเคลื่อนไหว (OCT)

วัสดุอุปกรณ์ที่ใช้ในฟังก์ชันมีดังต่อไปนี้

- 1) เซนเซอร์ตรวจจับการเคลื่อนไหว
- 2) ไมโครคอนโทรลเลอร์
- 3) วงจรควบคุมหลอดไฟ
 - Switching Circuit
 - Dimming Circuit

ระบบของฟังก์ชันนี้จะเป็นการทำงานแบบอัตโนมัติจากการตรวจจับการเคลื่อนไหวด้วยเซนเซอร์ตรวจจับการเคลื่อนไหว โดยจะมีไมโครคอนโทรลเลอร์รับสัญญาณจากเซนเซอร์ เพื่อส่งสัญญาณไปยังวงจรอิเล็กทรอนิกส์ให้เปิด/ปิดหลอดไฟตามที่ได้โปรแกรมไว้ ซึ่งมีหลักการทำงานดังแสดงในรูปที่ 4.57



รูปที่ 4.57 การทำงานของฟังก์ชันตรวจจับการเคลื่อนไหว

4.4.2. ฟังก์ชันการควบคุมการเปิด/ปิดไฟอย่างอัตโนมัติตามตารางเวลาที่กำหนดไว้ (TSD)

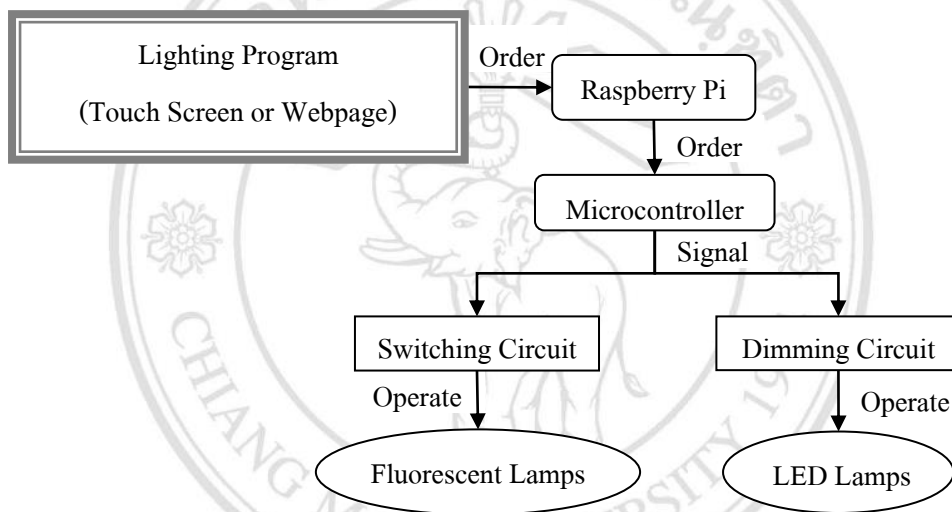
วัสดุอุปกรณ์ที่ใช้ในฟังก์ชันมีดังต่อไปนี้

- 1) Touch Screen
- 2) Raspberry Pi
- 3) ไมโครคอนโทรลเลอร์

4) วงจรควบคุมหลอดไฟ

- Switching Circuit
- Dimming Circuit

ในฟังก์ชันนี้ผู้ใช้จะกำหนด วัน/เดือน/ปี และเวลาที่ต้องการเปิด/ปิดหลอดไฟด้วยโปรแกรมที่พัฒนาขึ้นผ่านทางจอ Touch Screen หรือหน้า Webpage จากนั้น Raspberry Pi จะเป็นตัวประมวลผลข้อมูลด้วยโปรแกรม Gambus เพื่อส่งการต่อไปยังไมโครคอนโทรลเลอร์ให้ส่งสัญญาณไปควบคุมหลอดไฟผ่านวงจรอิเล็กทรอนิกส์ต่อไป ซึ่งมีหลักการทำงานดังแสดงในรูปที่ 4.58



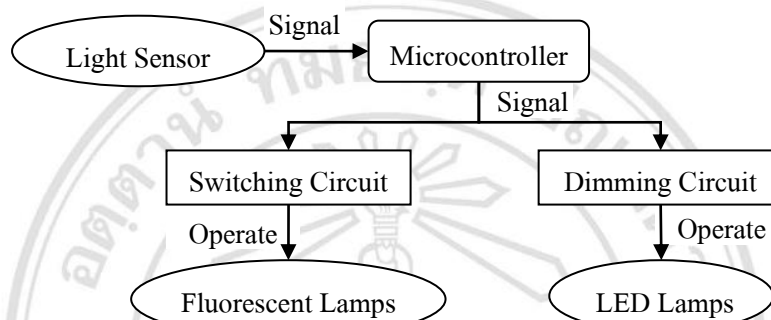
รูปที่ 4.58 การทำงานของฟังก์ชันการควบคุมการเปิด/ปิดไฟอย่างอัตโนมัติ
ตามตารางเวลาทำงานที่กำหนดไว้

4.4.3 ฟังก์ชันการใช้ประโยชน์จากแสงสว่างภายนอก (DCT)

วัสดุอุปกรณ์ที่ใช้ในฟังก์ชันมีดังต่อไปนี้

- 1) เซนเซอร์วัดความเข้มแสง (Light Sensor)
- 2) ไมโครคอนโทรลเลอร์
- 3) วงจรควบคุมหลอดไฟ
 - Switching Circuit
 - Dimming Circuit

ระบบของฟังก์ชันนี้จะมีการควบคุมให้แสงภายในห้องมีปริมาณที่เหมาะสมตามมาตรฐานการใช้งานแบบอัตโนมัติ จากการใช้ประโยชน์จากแสงสว่างจากธรรมชาติให้ได้มากที่สุด โดยการตรวจเช็คค่าแสงสว่างภายในห้องผ่านทางเซนเซอร์วัดความเข้มแสง จากนั้นส่งค่าที่ได้ไปยังไมโครคอนโทรลเลอร์เพื่อสั่งการให้ระบบเปิด/ปิดหรือปรับระดับแสงสว่างจากหลอดไฟให้เหมาะสมกับปริมาณแสงที่วัดได้ขณะนั้นตามที่ได้โปรแกรมไว้ ซึ่งมีหลักการทำงานดังแสดงในรูปที่ 4.59



รูปที่ 4.59 การทำงานของฟังก์ชันการใช้ประโยชน์จากแสงสว่างภายนอก

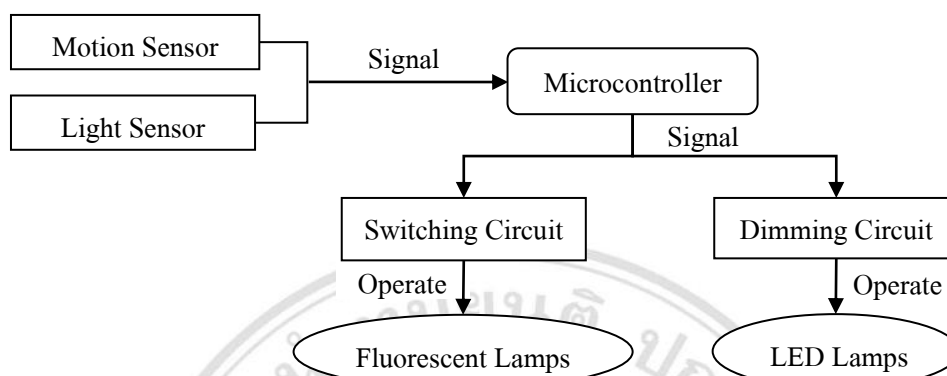
4.4.4. ฟังก์ชันการใช้แสงสว่างเฉพาะจุดหรือเฉพาะบริเวณที่มีผู้ใช้งาน (TCT)

วัสดุอุปกรณ์ที่ใช้ในฟังก์ชันมีดังต่อไปนี้

- 1) เซนเซอร์
 - เซนเซอร์ตรวจจับการเคลื่อนไหว
 - เซนเซอร์วัดความเข้มแสง
- 2) ไมโครคอนโทรลเลอร์
- 3) วงจรควบคุมหลอดไฟ
 - Switching Circuit
 - Dimming Circuit

ฟังก์ชันนี้จะแบ่งเป็น 2 กรณี คือ กรณีการแบ่งพื้นที่ในการใช้งานเพื่อควบคุมแสงสว่างเฉพาะบริเวณที่มีผู้ใช้งานแบบอัตโนมัติ จากการใช้งานของเซนเซอร์ตรวจจับการเคลื่อนไหวรวมกับการทำงานของไมโครคอนโทรลเลอร์ หรือในอีกกรณีจะเป็นการแบ่งพื้นที่ห้องตามระยะห่างจากหน้าต่าง โดยบริเวณที่ใกล้หน้าต่างจะใช้แสงสว่างจาก

หลอดไฟน้อยกว่าบริเวณที่ไถ่ลอกออกมา ในส่วนนี้จะเป็นการใช้เซนเซอร์วัดความเข้มแสง ร่วมกับการทำงานของไมโครคอนโทรลเลอร์ ซึ่งมีหลักการทำงานดังแสดงในรูปที่ 4.60



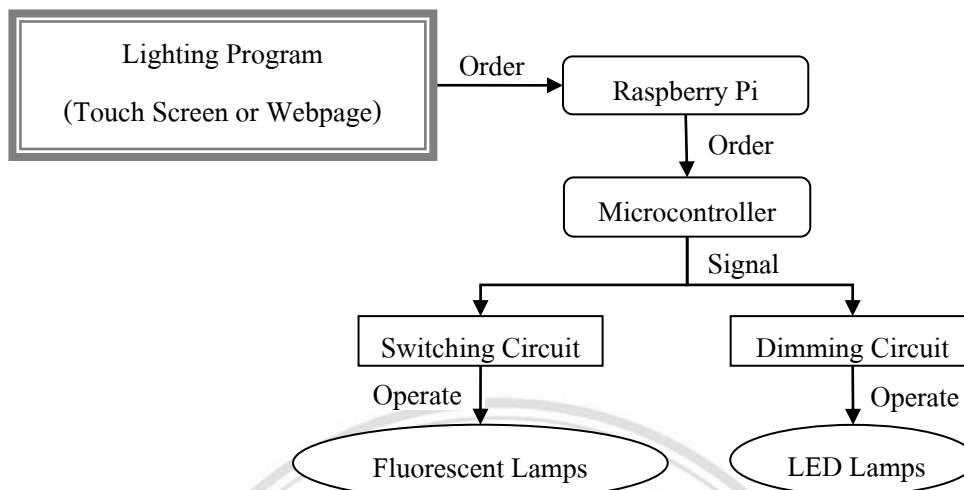
รูปที่ 4.60 การทำงานของฟังก์ชันการใช้แสงสว่างเฉพาะจุดหรือเฉพาะบริเวณที่มีผู้ใช้งาน

4.4.5. ฟังก์ชันการควบคุมความสว่างหลอดไฟด้วยผู้ใช้ไฟเอง (PCT)

วัสดุอุปกรณ์ที่ใช้ในฟังก์ชันมีดังต่อไปนี้

- 1) Touch Screen
- 2) Raspberry Pi
- 3) ไมโครคอนโทรลเลอร์
- 4) วงจรควบคุมหลอดไฟ
 - Switching Circuit
 - Dimming Circuit

ระบบของฟังก์ชันนี้จะทำงานเมื่อมีการสั่งการจากผู้ใช้งานโดยตรง ไม่ว่าจะผ่านทางสวิตช์ ควบคุมด้วยมือ (Manual Switch) รวมถึงการควบคุมด้วยโปรแกรมจัดการพลังงานไฟฟ้า แสงสว่างผ่านอุปกรณ์อื่น ๆ เช่น จอ Touch screen หรือหน้า Webpage เป็นต้น ซึ่งมี หลักการทำงานดังแสดงในรูปที่ 4.61



รูปที่ 4.61 การทำงานของฟังก์ชันการควบคุมความสว่างหลอดไฟด้วยผู้ใช้ไฟเอง

4.4.6. ฟังก์ชันการควบคุมการใช้พลังงานโดยรวมทั้งระบบจากปริมาณค่ากำลังไฟฟ้าสูงสุด (VPS)

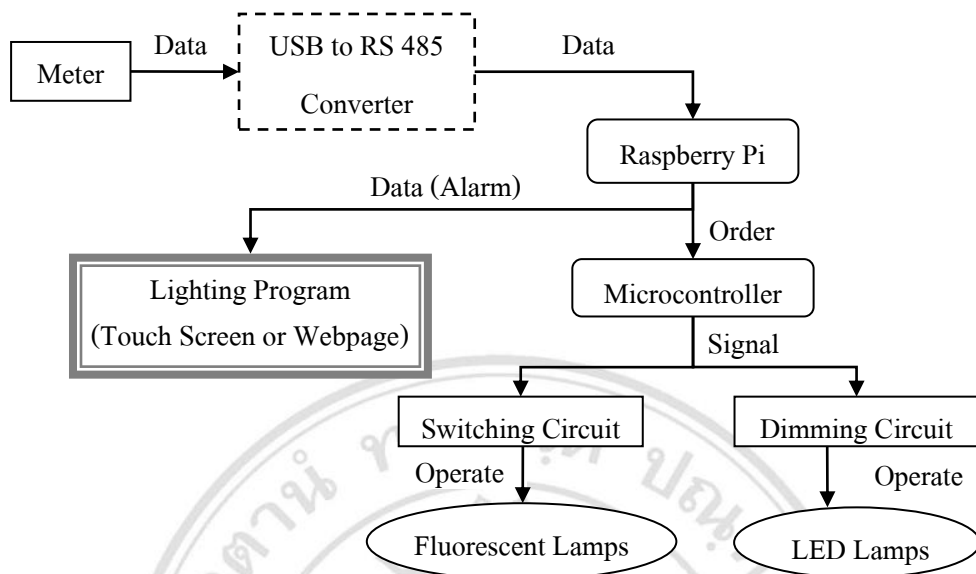
วัสดุอุปกรณ์ที่ใช้ในฟังก์ชันมีดังต่อไปนี้

- 1) มิเตอร์วัดพลังงาน (Meter)
- 2) USB to RS 485 Converter
- 3) Touch Screen
- 4) Raspberry Pi
- 5) ไมโครคอนโทรลเลอร์
- 6) วงจรควบคุมหลอดไฟ

- Switching Circuit

- Dimming Circuit

ในฟังก์ชันนี้มิเตอร์จะวัดค่าการใช้กำลังไฟฟ้าจากนั้นจะส่งผลการใช้กำลังไฟฟ้าผ่าน USB to RS 485 Converter มายัง Raspberry Pi ให้ประมวลผลว่ามีการใช้กำลังไฟฟ้าเกินค่าที่กำหนดไว้หรือไม่ เพื่อทำการเตือนให้ผู้ใช้ลดการใช้แสงสว่างลงเมื่อมีการใช้กำลังไฟฟ้าถึงค่าที่กำหนด หรือสั่งการให้ไมโครคอนโทรลเลอร์ปิดไฟลงแบบอัตโนมัติ ในบริเวณที่สำคัญน้อยที่สุดลงก่อน ซึ่งมีหลักการทำงานดังแสดงในรูปที่ 4.62



รูปที่ 4.62 การทำงานของฟังก์ชันการควบคุมการใช้พลังงานโดยรวมทั้งระบบจากปริมาณค่ากำลังไฟฟ้าสูงสุด

ในการเลือกใช้นิค หรือจำนวนของไมโครคอนโทรลเลอร์ต้องพิจารณาถึงจำนวน Input/Output ทั้งหมดจากทั้งเซนเซอร์และวงจรที่ไมโครคอนโทรลเลอร์ทำการควบคุม หรือ ต้องการส่ง/รับสัญญาณไปยังเซนเซอร์และวงจรนั้นๆ