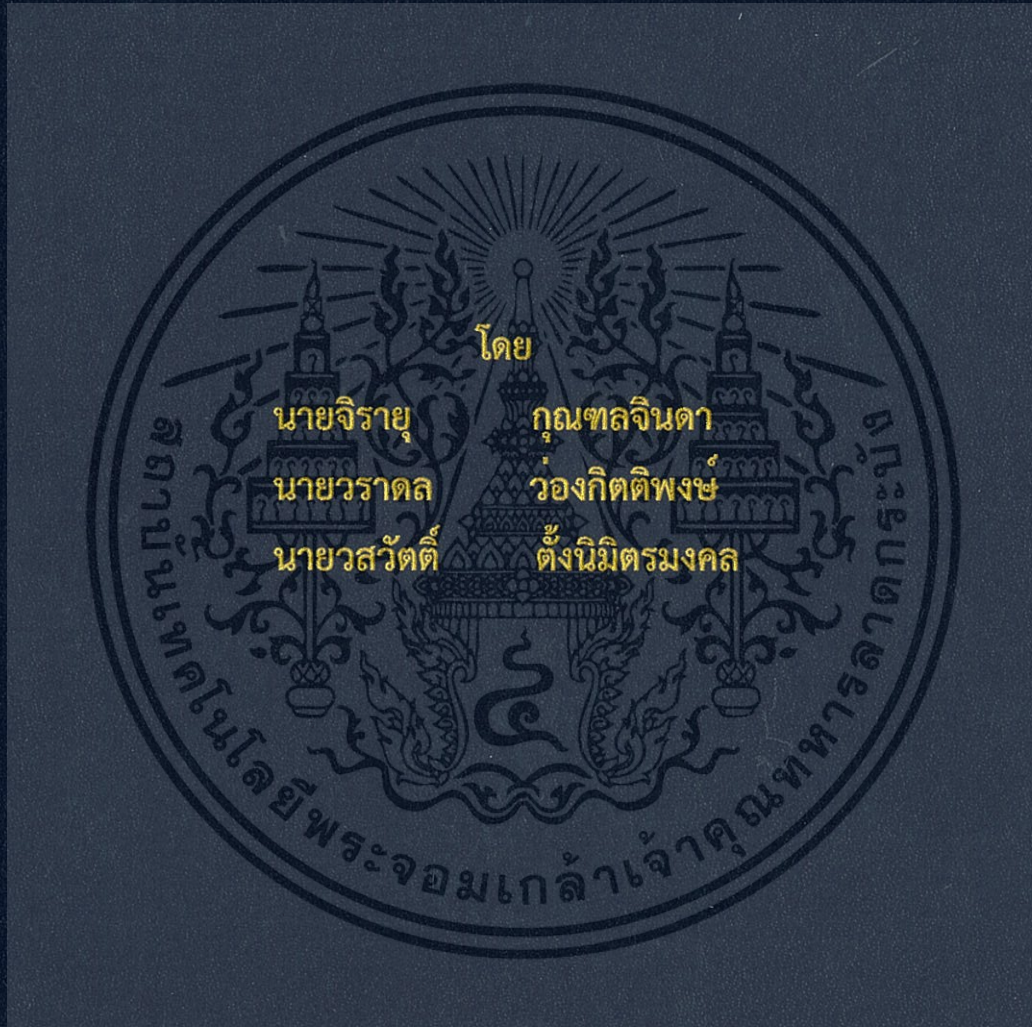


ระบบควบคุมแสงภายในห้อง
Controlled light system in room



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

ระบบควบคุมแสงภายในห้อง
Controlled light system in room



โดย

นายจิรายุ	กฤษณทลจินดา	55010182
นายวราดล	ว่องกิตติพงษ์	55011099
นายวสวัตดี	ตั้งนิมิตรมงคล	55011116

สาขา.....
เลขทะเบียน 144341
รับเดือนที่ 4 พ.ย. 2559


อาจารย์ที่ปรึกษา

ผศ.ดร. สิริภาพ ตู่ประกาย

b. 12820568
i.

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

ผ่านการตรวจรูปเล่มแล้ว
()
อาจารย์ที่ปรึกษา
24/05/59
วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว
()
กรรมการผู้ตรวจชิ้นงาน
24/พค, 59
วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาโทปีการศึกษา 2558

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมแสง

Controlled light system in room

ผู้จัดทำ

- | | | |
|---------------|---------------|----------|
| 1. นายจิรายุ | กฤษณจินดา | 55010182 |
| 2. นายวรากล | ว่องกิตติพงษ์ | 55011099 |
| 3. นายสวัสดิ์ | ตั้งนิมิตมงคล | 55011116 |

.....
(ผศ.ดร. สิริภาพ ผู้ประภาย)

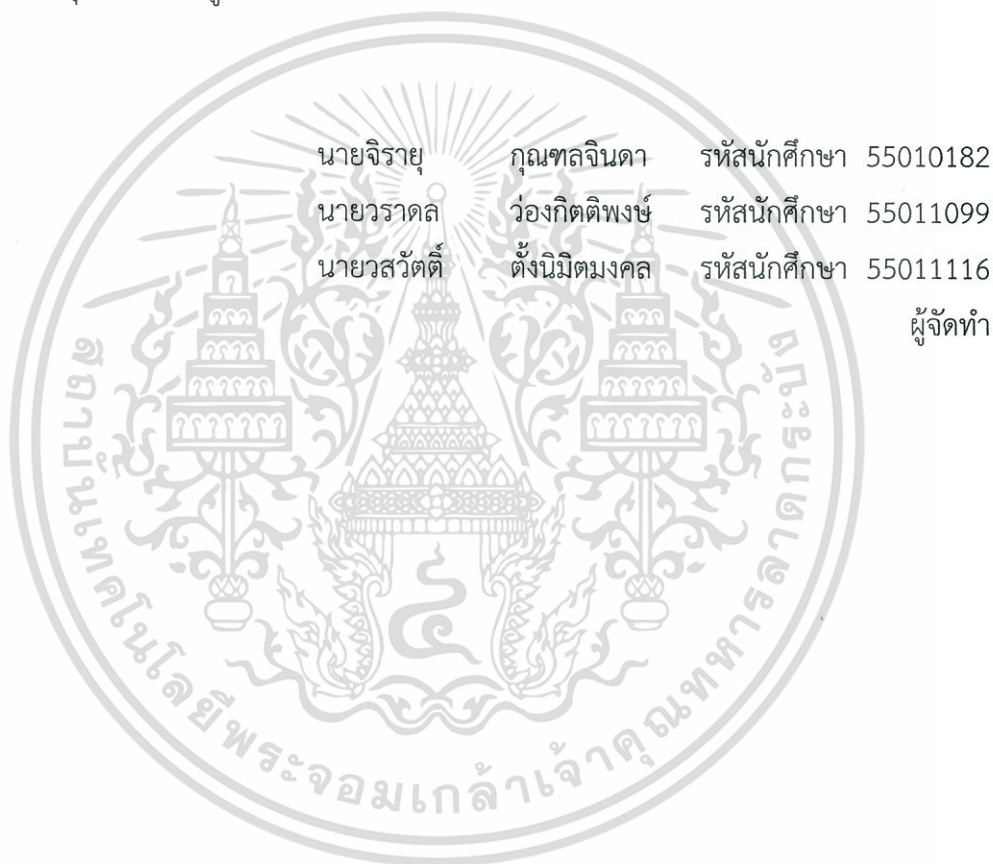
อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอกราบขอบพระคุณบิดา มารดา ที่ให้การสนับสนุนและให้กำลังใจตลอดการศึกษา
ขอขอบพระคุณ ผศ.ดร. สิริภพ ตู๊ประกาย ที่ให้คำแนะนำและช่วยเหลือสนับสนุนที่เป็นประโยชน์ต่อ
การทำโครงการนี้ให้สำเร็จลุล่วง และขอบคุณรุ่นพี่และเพื่อนๆทุกคนที่ให้การช่วยเหลือ ให้คำปรึกษา
ต่างๆ เกี่ยวกับการทำโครงการนี้ ผู้จัดทำโครงการรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอ
กราบขอบพระคุณเป็นอย่างสูง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมแสงภายในห้อง

CONTROLLED LIGHT SYSTEM IN ROOM

โดย นายจิรายุ กุณฑลจินดา 55010182

นายวรากล ว่องกิตติพงษ์ 55011099

นายสวัสดิ์ ตั้งนิมิตรมงคล 55011116

อาจารย์ที่ปรึกษา ผศ.ดร.สิรภพ ตู้ประกาย

บทคัดย่อ

โครงการนี้แสดงระบบควบคุมแสงภายในห้อง ประกอบเป็นระบบรักษาความปลอดภัย สามารถควบคุมผ่านทางโทรศัพท์มือถือ ผ่านทางอินเทอร์เน็ต จะแบ่งเป็น 3 ระบบคือ ระบบควบคุม ม่าน โดยมีอุปกรณ์ตรวจจับสถานะของม่าน ระบบควบคุมไฟ จะใช้ Relay เป็นตัวสั่งการ โดยมี อุปกรณ์ตรวจจับแสงภายนอกบ้าน เพื่อให้ไฟทำงานได้อย่างอัตโนมัติ และระบบตรวจจับความ เคลื่อนไหวโดยใช้ PIR โดยจะมีสัญญาณแจ้งเตือนให้เจ้าของบ้านทราบ โดยทั้ง 3 ระบบนี้ จะควบคุม ผ่าน Microcontroller Arduino ควบคุมผ่านทาง Android และสามารถ Login ข้อมูลของผู้ใช้งาน โดยใช้ IP Address ของ Wifi Module เพื่อให้เกิดความปลอดภัยในการใช้งาน

ABSTRACT

This project is established for the purpose of learning about how to design an Android application to let Arduino UNO r3 control the electronic system automatically, how to construct the dimmer and motor system, and most importantly, how to design an application by using Android Studio program to control illuminance with Arduino module.

All in all, we have learned many new and interesting things from doing this project which are the way how to write the program on an Android device, how to make it practical in real life, and how the dimmer and motor system are working. Then, we apply all the knowledge and our skills to build a controllable lighting system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	4
2.1 ARDUINO UNO R3	4
2.2 รายละเอียดอุปกรณ์ใน วงจร RELAY วงจรมอเตอร์และ MODULE ต่างๆของ ARDUINO	11
2.3 โปรแกรม ANDROID STUDIO 1.3.2	20
บทที่ 3	
การออกแบบและการจัดทำปริญญานิพนธ์	25
3.1 การออกแบบ	25
3.2 เครื่องมือที่ใช้ในการทดลอง	30
3.3 การจัดเก็บผลการทดลอง	31

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	32
4.1 การเข้าสู่ระบบ	32
4.2 การแสดงผลของระบบไฟ	33
4.3 การแสดงผลของระบบมาน	34
4.4 การแสดงผลของระบบตรวจจับความเคลื่อนไหว	35
บทที่ 5 สรุปผลและข้อเสนอแนะ	36
5.1 สรุปผล	36
5.2 ข้อเสนอแนะ	36
บรรณานุกรม	37
ภาคผนวก ก	38
ภาคผนวก ข	43
ภาคผนวก ค	49

สารบัญรูป

รูปที่	หน้า
1.1 BLOCK DIAGRAM ของโครงงาน	2
2.1 โครงสร้างของ ARDUINO UNO R3	5
2.2 แสดงหน้าจอ SERIAL MONITOR	8
2.3 เลือกบอร์ด ARDUINO ที่ต้องการ UPLOAD	9
2.4 เลือกบอร์ด ARDUINO ที่ต้องการ UPLOAD	9
2.5 เลือกหมายเลข COMPORT ของบอร์ด	10
2.6 กดปุ่ม VERIFY เพื่อตรวจสอบความถูกต้อง	10
2.7 ทำการ COMPILE และ UPLOAD โค้ดโปรแกรม	10
2.8 RELAY	11
2.9 DIODE	11
2.10 DIAC	12
2.11 โครงสร้าง สัญลักษณ์ และวงจรสมมูลของ TRIAC	13
2.12 โครงสร้าง RESISTOR	13
2.13 หลอดไฟ LED	14
2.14 โครงสร้าง L293D อุปกรณ์เสริม ในโปรแกรม	15
2.15 โครงสร้าง มอเตอร์กระแสตรง	16
2.16 โปรแกรม ALTIUM DESIGNER เป็นโปรแกรมออกแบบวงจร	16
2.17 PIR MODULE	17
2.18 SPEAKER MODULE	17
2.19 BH1750 MODULE	18
2.20 ULTRASONIC MODULE	19
2.21 ESP8266 MODULE	20
2.22 หน้า HOMEPAGE ของ ANDROID STUDIO	22
2.23 หน้าต่าง แรกในการติดตั้งโปรแกรม	22
2.24 หน้าต่างการเลือกติดตั้งอุปกรณ์เสริม ในโปรแกรม	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.25	หน้าต่างเลือก PATH ในกรณีเลือกติดตั้ง ทั้ง ANDROID SDK และ ANDROID STUDIO	23
2.26	หน้าต่างการเลือกติดตั้งเฉพาะ ANDROID STUDIO	24
2.27	หน้าต่างการแสดงว่าการติดตั้งเสร็จสิ้น โปรแกรมพร้อมเข้าใช้งาน	24
3.1	BLOCK DIAGRAM ของโครงงาน	25
3.2	ผังการทำงานของระบบเปิด/ปิดไฟ	26
3.3	ผังการทำงานของระบบเปิด/ปิดม่าน	27
3.4	ผังการทำงานของระบบตรวจจับการเคลื่อนไหว	28
3.5	หน้าแอปพลิเคชัน บน ANDROID	29
3.6	ภายในแอปพลิเคชัน	29
4.1	แอปพลิเคชัน	32
4.2	ภายในแอปพลิเคชัน	32
4.3	แสดงการเปิดไฟผ่านแอปพลิเคชัน	33
4.4	แสดงการปิดไฟผ่านแอปพลิเคชัน	33
4.5	แสดงการเปิดม่านผ่านแอปพลิเคชัน	34
4.6	แสดงการปิดม่านผ่านแอปพลิเคชัน	34
4.7	แสดงการหยุดมอเตอร์ผ่านทางแอปพลิเคชันโดยมี ULTRASONIC เพื่อตรวจจับสถานะของม่าน	35
4.8	ระบบตรวจจับการเคลื่อนไหว และ SPEAKER เพื่อแจ้งเตือน	35
5.1	SCHEMATIC ของวงจร RELAY	40
5.2	SCHEMATIC ของวงจรมอเตอร์	40
5.3	การ Update PCB	41
5.4	PCB วงจร RELAY	41
5.5	PCB วงจร มอเตอร์	42

สารบัญตาราง

ตารางที่

2.1 โครงสร้าง L293D

หน้า

16



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการพัฒนาทางด้านเทคโนโลยีนั้นก้าวหน้าไปอย่างรวดเร็ว ทั้งในด้านภายนอกหรือ Hardware และภายในหรือ Software และมีเทคโนโลยีใหม่ๆเกิดขึ้นทุกวัน ซึ่งส่งผลกระทบต่อ การดำเนินชีวิตประจำวันของผู้คนเป็นอย่างมาก ด้วยเหตุผลเหล่านี้เราจึงมีความคิดที่จะสร้างระบบเพื่อ จะอำนวยความสะดวกและรักษาความปลอดภัย ชื่อว่า “ระบบควบคุมแสงภายในห้อง” โดยระบบนี้ จะเป็นการควบคุมการเปิด/ปิดไฟอย่างอัตโนมัติ ควบคุมการเคลื่อนที่ของผ้าม่านโดยมีตัวตรวจจับ สถานะ และการตรวจจับความเคลื่อนไหวโดยจะมีสัญญาณแจ้งเตือน โดยควบคุมผ่านทาง แอปพลิเคชันบนโทรศัพท์มือถือ ซึ่งทำงานร่วมกับไมโครคอนโทรลเลอร์ เพื่อเป็นทางเลือกให้แก่คน ทั่วไปในการใช้เทคโนโลยี

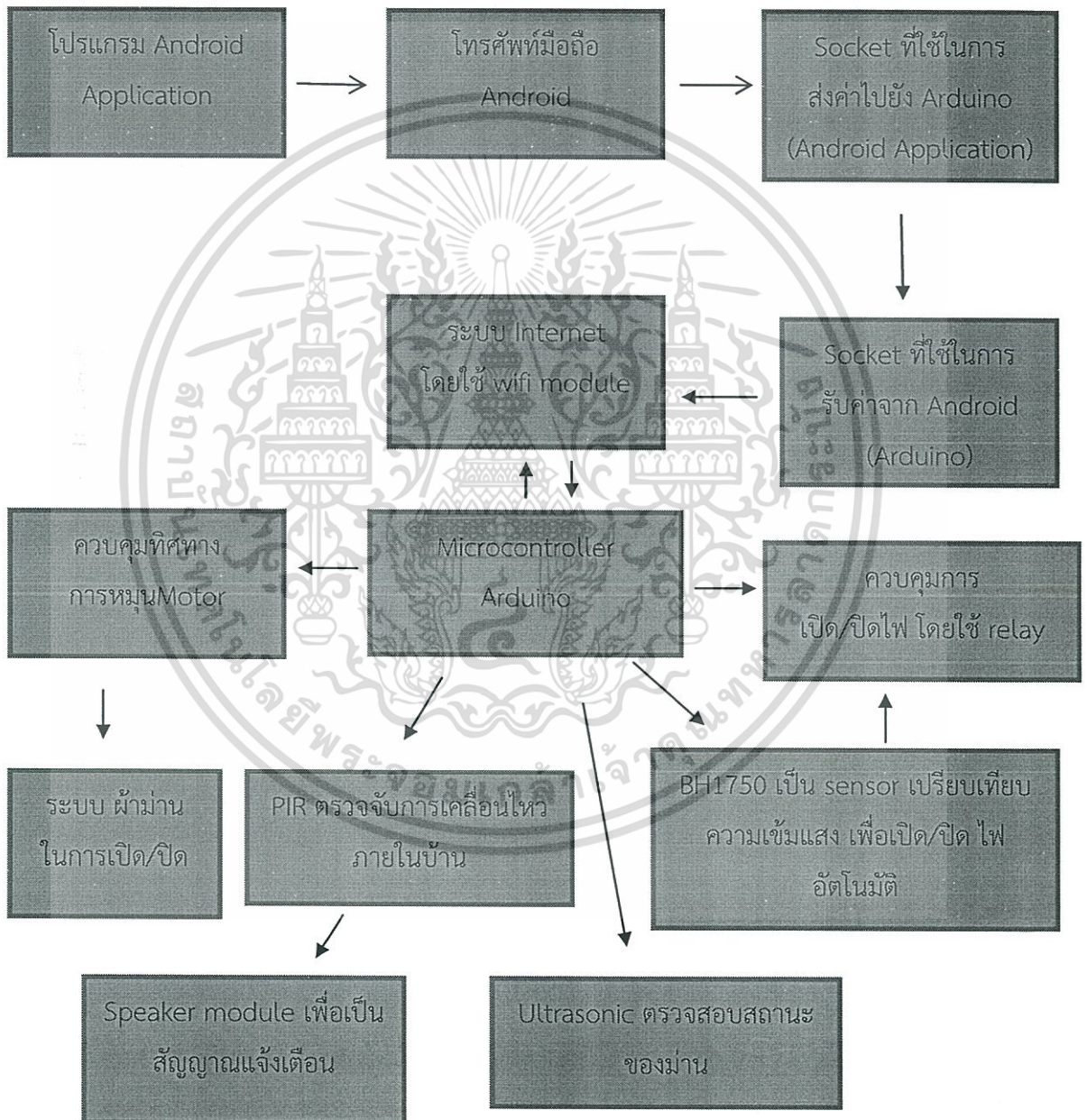
1.2 วัตถุประสงค์

- 1) ศึกษาการทำงานและการใช้โปรแกรม arduino altium designer และ android studio
- 2) ศึกษาการทำงานของวงจรมอเตอร์ Relay และวงจรมอเตอร์
- 3) ศึกษาการทำงานร่วมกันของ hardware และ software เพื่อควบคุมระบบแสง ภายในห้องอย่างอัตโนมัติ โดยผ่านทางแอปพลิเคชันโทรศัพท์มือถือ
- 4) เพื่อเป็นแนวทางให้ผู้สนใจในการทำโปรเจกต์ได้ศึกษาต่อ และสามารถให้ความรู้กับ น้องๆในการต่อยอดโปรเจกต์ได้

1.3 ขอบเขตของปริญญาานิพนธ์

โครงการนี้เป็นการสร้างอุปกรณ์ เพื่อควบคุม การเปิด/ปิด ของผ้าม่าน การเปิด/ปิด ของ หลอดไฟ และ ระบบตรวจจับการเคลื่อนไหว ประกอบเป็นระบบแจ้งเตือนเพื่อรักษาความปลอดภัย โดยใช้ Microcontroller Arduino เป็นตัวควบคุมการทำงาน และสามารถควบคุมได้ผ่านทาง โทรศัพท์มือถือ Android โดยใช้ระบบ Internet ระบบแรก การเปิด/ปิด ผ้าม่าน จะให้ Arduino ควบคุมมอเตอร์ เพื่อหมุนเปิด/ปิด/หยุด ม่าน และจะใช้ Ultrasonic สถานะของม่านว่าเปิดหรือปิดอยู่ ระบบที่สอง การเปิด/ปิด ของหลอดไฟ ระบบนี้จะมีการทำงานเป็นสองส่วน ส่วนแรกคือ ทำงานเปิด/ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปิด โดยผ่านทางโทรศัพท์มือถือ และส่วนที่สอง เมื่อเราไม่อยู่บ้านสามารถกดเปิดระบบทำงานอัตโนมัติ โดยใช้ BH1750 เป็น sensor เพื่อควบคุมการเปิด/ปิด เมื่อมืด ไฟจะเปิดโดยอัตโนมัติ และระบบที่ 3 คือ ระบบตรวจจับการเคลื่อนไหว โดยใช้ PIR เมื่อมีสิ่งต่างๆผ่าน จะมีสัญญาณแจ้งเตือนเพื่อให้เจ้าของบ้านทราบ โดยทั้ง 3 ระบบนี้ จะสามารถสั่งการผ่าน โทรศัพท์มือถือ Android



รูปที่ 1.1 Block Diagram ของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถเขียน Arduino code และ Android code ได้ จากที่ไม่เคยเขียนcode มาก่อนเลย
- 2) ได้รับความรู้จากการทำ project มากขึ้น เพราะต้องศึกษาจากแหล่งข้อมูลต่างๆ ไม่ว่าจะเป็น internet หรือนิตยสารต่างๆ
- 3) สามารถเป็นแนวทางสำหรับผู้ที่มีสนใจในการทำ project นำไปศึกษาและนำไปใช้จริงได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

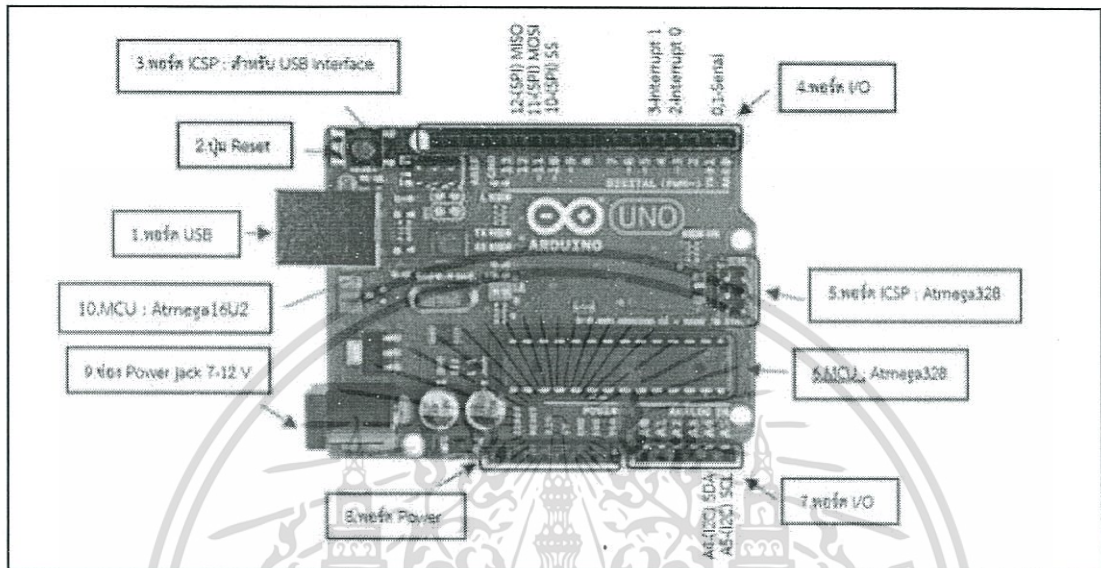
อธิบายถึงทฤษฎีและหลักการพื้นฐานที่เกี่ยวข้องในการทำโครงงาน และอธิบายเกี่ยวกับโปรแกรมที่ใช้ในการทำโครงงานต่างๆ รวมถึงอุปกรณ์ Hardware ที่ใช้ในการสร้างชิ้นงาน ตลอดจนข้อมูลอื่นๆที่เป็นประโยชน์ต่อผู้อ่าน

2.1 Arduino UNO R3

2.1.1 ทฤษฎีพื้นฐานเกี่ยวกับ Arduino

เป็นภาษาอิตาลี เป็นชื่อที่ใช้เพื่อโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูลของ AVR (Alf (Egil Bogen) and Vegard (Wollan)'s RISC processor) แบบรหัสเปิด (open source) ที่ได้รับการปรับปรุงมาจากอีกโครงการ “Wiring” ที่ใช้ ATmega128 ที่มีหน่วยความจำและ I/O ค่อนข้างมากและตัวถังของ ATmega128 เป็นอุปสรรคกับผู้ใช้งานในเบื้องต้น จึงไม่เป็นที่นิยมจากนั้นทีมงานของ Arduino ได้นำมาพัฒนาต่อยอดโดยให้สามารถใช้งาน AVR ขนาดเล็กได้จึงทำให้วงจรของบอร์ดมีขนาดที่เล็กลง อีกทั้งยังใช้อุปกรณ์จำนวนน้อยอีกด้วย ทำให้ง่ายในการต่อวงจรและประหยัดในการสร้าง บอร์ดมาก จึงทำให้ Arduino ได้รับความนิยมอย่างมาก Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้ AVR ขนาดเล็กเป็นตัวประมวลผล เหมาะสำหรับการศึกษาเรียนรู้ไมโครคอนโทรลเลอร์ และการนำไปประยุกต์ใช้งานในการควบคุมอุปกรณ์ทั้ง อินพุทและเอาต์พุทต่างๆได้ โปรแกรม ภาษาของ Arduino จะใช้รูปแบบของภาษา C++ โดยในการทดลองนี้จะเป็นการเริ่มต้นศึกษาการใช้งานบอร์ด ไมโครคอนโทรลเลอร์ Arduino Model: Arduino UNO R3 ใช้ชิพ ATmega328 รั้นที่ความถี่ 16 MHz หน่วยความจำ แฟลช 32 KB แรม 2 KB บอร์ดใช้ไฟเลี้ยง 7 ถึง 12 V มีระดับแรงดันไฟฟ้าในการทำงานและขาสัญญาณอยู่ที่ 5 V (TTL) มี Digital Input / Output 14 ขา (เป็น PWM ได้ 6 ขา) มี Analog Input 6 ขา Serial UART 1 ชุด I²C 1 ชุด SPI 1 ชุด เขียนโปรแกรม บนซอฟต์แวร์ Arduino IDE และโปรแกรมผ่านพอร์ต USB [1]

โครงสร้างของ Arduino สามารถแสดงได้ในรูป 2.1



รูปที่ 2.1 โครงสร้างของ arduino UNO R3

2.1.2 ส่วนประกอบของ arduino มีดังนี้

2.1.2.1. USBPort: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด

2.1.2.2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

2.1.2.3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

2.1.2.4. I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM

2.1.2.5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

2.1.2.6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

2.1.2.7. I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

2.1.2.8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

2.1.2.10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

ถึงปัจจุบัน Arduino มีบอร์ดหลายแบบให้เลือกใช้งานตามความถนัดและความเหมาะสมมากกว่า 20 รุ่น แต่ละรุ่นก็มีขนาด ความจุ ความเร็ว จำนวนขาพอร์ตอินพุต เอาท์พุต แตกต่างกันไป มีตั้งแต่ราคาหลักสองสามร้อยบาท ไปจนกระทั่งพันกว่าบาท นอกจากนี้ยังมีอุปกรณ์ต่อพ่วง (Shield) ให้อีกสารพัด ราคาก็เป็นไปตาม concept เดิมครับ คือ สมเหตุสมผลสุดๆ ทำให้เป็นที่นิยมมากขึ้นเรื่อยๆ

2.1.3 ชนิดข้อมูลพื้นฐานของ arduino

Byte ใช้สำหรับข้อมูลที่เป็นเลขจำนวนเต็มได้ตั้งแต่ 0 ถึง 255

Int ใช้สำหรับข้อมูลที่เป็นเลขจำนวนเต็มได้ตั้งแต่ -32768 ถึง +32767

Float ใช้สำหรับข้อมูลที่เป็นเลขทศนิยม เป็นค่าที่เป็นบวกหรือลบในช่วง ที่กว้างกว่าชนิดข้อมูลแบบ byte หรือ int และเป็นเลขทศนิยมได้ด้วย แต่มีความละเอียดเพียง 6-7 ตำแหน่งหลังจุดทศนิยมในเลขฐานสิบ

Boolean ใช้สำหรับข้อมูลที่เป็นค่าลอจิก true(จริง) หรือ false(ผิด) เท่านั้น

2.1.4 โปรแกรมพื้นฐานของ arduino

ในการเขียนโปรแกรม Arduino นั้นจะมีส่วนประกอบอยู่ 2 ส่วนคือ ส่วนสำหรับการกำหนดค่าต่างๆและส่วนของการสั่งโปรแกรมทำงาน โดย แบ่งออกได้ดังนี้

```
void setup ()
```

```
{
```

```
}
```

ในส่วนนี้จะเป็นการกำหนดค่าต่างๆ เช่น จะใช้กำหนดขาในการใช้งานให้เป็น INPUT หรือ OUTPUT ซึ่งการกำหนดส่วนนี้จะกำหนดเพียงครั้งเดียวเท่านั้น เช่น

```
void setup()
```

```
{
```

```
pinMode(led1, OUTPUT);
```

```
}
```

```
Void loop ()
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนนี้จะเป็นส่วนการเขียนโปรแกรมและสั่งให้โปรแกรมทำงาน ซึ่งในการทำงานจะเป็นแบบวนลูปไปเรื่อยๆ เช่น

```
void loop()
{
digitalWrite(led1,HIGH); )
delay(1000);
digitalWrite(led1,LOW); );
delay(1000);
}
```

2.1.5 คำสั่งพื้นฐาน

Int คือการประกาศตัวแปร เมื่อเราประกาศตัวแปรเช่น int a = 13 แปลว่าเราจะต้องใช้ตัวแปร a แทน port 13 ตลอดทั้งการเขียนโปรแกรม

pinMode() ใช้กำหนดทิศทางสัญญาณ input/output ของขา digital

digitalRead() ใช้อ่านค่าจากขา digital ที่ถูกกำหนดให้เป็นอินพุต

digitalWrite() ใช้เขียนค่า (LOW หรือ HIGH) ให้ขา digital ที่ถูกกำหนดให้เป็น

เอาต์พุต

analogWrite() ใช้สร้างสัญญาณ PWM เป็นเอาต์พุต

analogRead() ใช้อ่านค่าจากขา analog-input

delay() รอเวลาให้ผ่านไปตามระยะเวลาที่กำหนด(มิลลิวินาที) ก่อนที่จะทำขั้นตอน

ต่อไป

delayMicroseconds() รอเวลาให้ผ่านไปตามระยะเวลาที่กำหนด (ไมโครวินาที)

random() ให้ค่าเป็นเลขสุ่ม

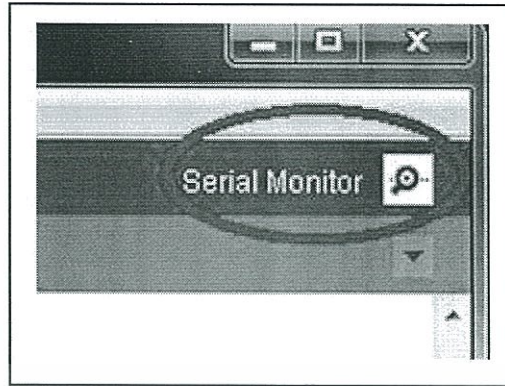
min() ให้ค่าน้อยสุดระหว่างตัวเลข 2 ค่าที่นำมาเปรียบเทียบกัน

max() ให้ค่ามากที่สุดระหว่างตัวเลข 2 ค่าที่นำมาเปรียบเทียบกัน

2.1.6 Serial Monitor

ในตัวโปรแกรม Arduino จะมี Serial Monitor ซึ่งมีประโยชน์ในการเขียนโปรแกรมที่ต้องการให้เห็นว่าตัวแปรนั้นๆ เปลี่ยนค่า หรือมีค่าตามที่กำหนดหรือไม่ สามารถแสดงได้ในรูปที่ 2.2

[2]



รูปที่ 2.2 แสดงหน้าจอ Serial Monitor

```
Serial.begin(9600);
```

โดยตัวเลขที่อยู่ในวงเล็บคือค่าความเร็วในการรับ-ส่งข้อมูล ในการทดลองนี้ใช้ 9600 ครั้ง และคำสั่งสำหรับสั่งให้ บอร์ดส่งข้อความมาที่คอมพิวเตอร์ของเรา คือ

```
Serial.print();
```

```
Serial.println();
```

โดยข้อความที่ Arduino จะส่งมาให้กับคอมพิวเตอร์คือข้อความที่อยู่ในเครื่องหมายคำพูด ในที่นี้คือในวงเล็บ โดยข้อแตกต่างระหว่างสองคำสั่งนี้คือ print จะเป็นการส่งข้อความต่อกันไปเรื่อยๆไม่มีการขึ้นบรรทัดใหม่ ส่วนคำสั่ง print ln จะขึ้นบรรทัดใหม่ให้เราทุกครั้งหลังจากส่งข้อความในเครื่องหมายคำพูด

2.1.7 คำสั่ง Else if

ในกรณีที่มีการตรวจสอบหลายๆ เงื่อนไขทำให้ไม่สะดวกต่อการใช้ if หลายๆ ครั้ง เราสามารถใช้ Else if เข้ามาช่วยตรวจสอบในแต่ละเงื่อนไขได้ เช่น

```
if เงื่อนไข 1 Then
```

```
ทำในนี้เมื่อตรงตามเงื่อนไข1
```

```
Else if เงื่อนไข 2 Then
```

```
ทำในนี้เมื่อตรงตามเงื่อนไข2
```

```
Else if เงื่อนไข 3 Then
```

```
ทำในนี้เมื่อตรงตามเงื่อนไข3
```

```
Else
```

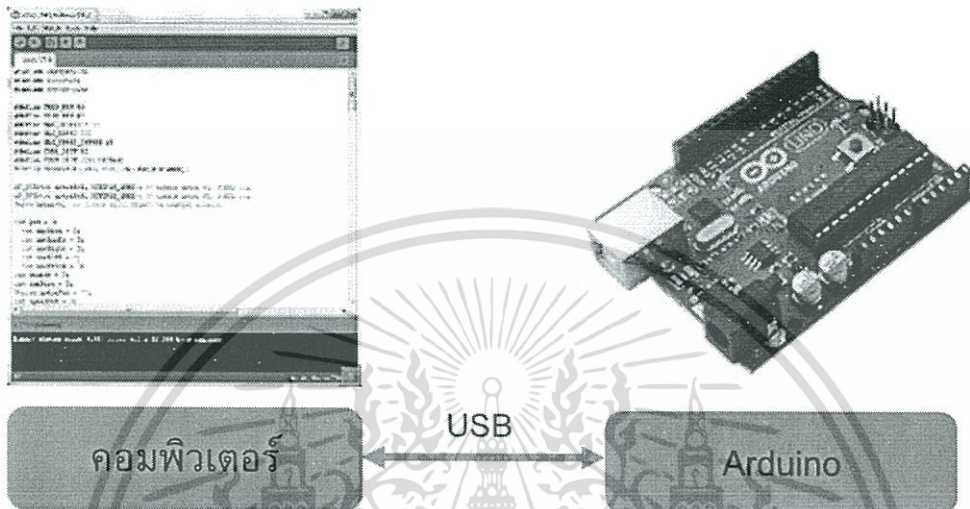
```
ทำในนี้เมื่อไม่ตรงตามเงื่อนไขใดเลย
```

```
end if
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

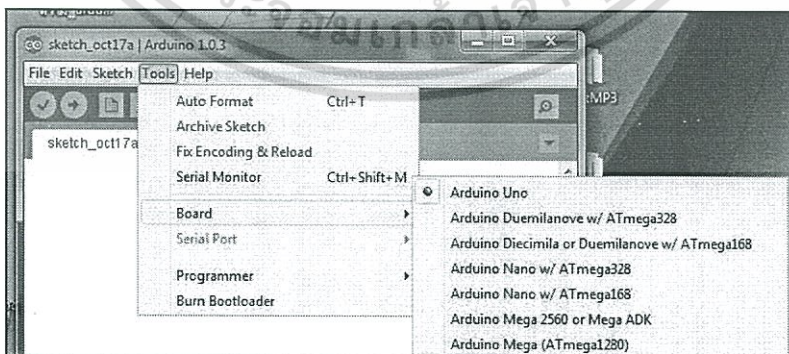
2.1.8 รูปแบบการเขียนโปรแกรมบน Arduino

ขั้นตอนแรกเราต้องเลือก บอร์ด arduino ที่จะทำโครงการนี้ ในที่นี้เราเลือกใช้ Arduino UNO R3 สามารถแสดงได้ดังรูปที่ 2.3



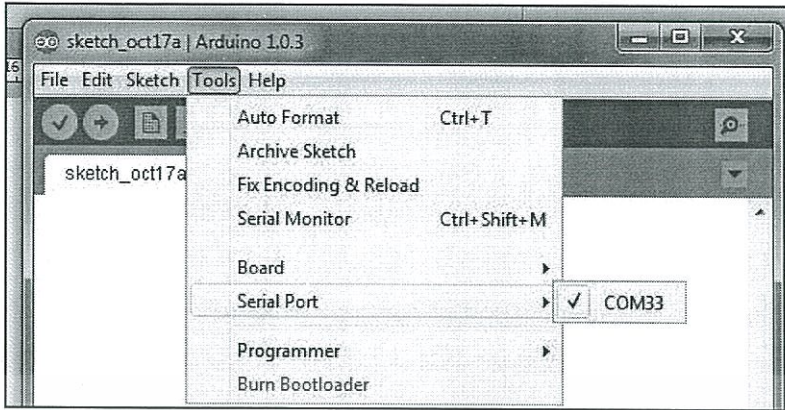
รูปที่ 2.3 เลือกบอร์ด Arduino ที่ต้องการ upload

1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino IDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software
2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกบอร์ด Arduino ที่ใช้และหมายเลข Com port สามารถแสดงได้ดังรูปที่ 2.4 และ 2.5



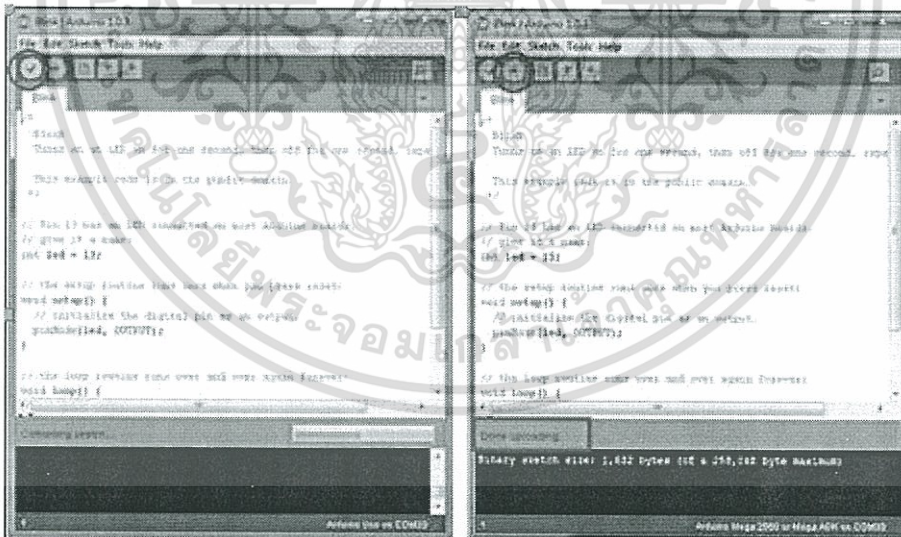
รูปที่ 2.4 เลือกบอร์ด Arduino ที่ต้องการ upload

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 เลือกหมายเลข Comport ของบอร์ด

3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที แสดงได้ดังรูปที่ 2.6 และ รูปที่ 2.7



รูปที่ 2.6 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง
รูปที่ 2.7 ทำการ Compile และ Upload โค้ดโปรแกรม

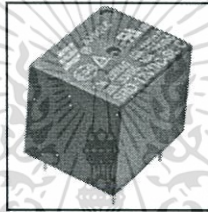
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 รายละเอียดอุปกรณ์ใน วงจร Relay วงจรมอเตอร์และ Module ต่างๆของ Arduino

2.2.1 อุปกรณ์ใน วงจร Relay

2.2.1.1 Relay

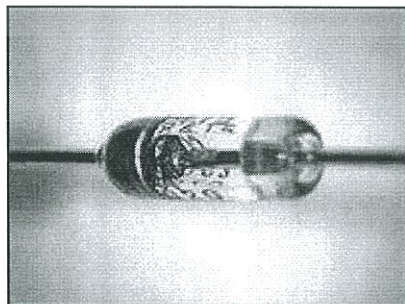
รีเลย์ (Relay) เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสถานะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย [3]



รูปที่ 2.8 Relay

2.2.1.2 Diode

ไดโอด เป็นชิ้นส่วนอิเล็กทรอนิกส์ชนิดสองขั้วคือขั้ว p และขั้ว n ที่ออกแบบและควบคุมทิศทางการไหลของประจุไฟฟ้า มันจะยอมให้กระแสไฟฟ้าไหลในทิศทางเดียว และกั้นการไหลในทิศทางตรงกันข้าม เมื่อก้าวถึงไดโอด มักจะหมายถึงไดโอดที่ทำมาจากสารกึ่งตัวนำ (Semiconductor diode) ซึ่งก็คือผลึกของสารกึ่งตัวนำที่ต่อกันได้ขั้วทางไฟฟ้าสองขั้ว ส่วนไดโอดแบบหลอดสุญญากาศ ถูกใช้เฉพาะทางในเทคโนโลยีไฟฟ้าแรงสูงบางประเภท เป็นหลอดสุญญากาศที่ประกอบด้วยขั้วอิเล็กโทรดสองขั้ว ซึ่งก็คือแผ่นตัวนำ และแคโทด [4]



รูปที่ 2.9 Diode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3 Diac

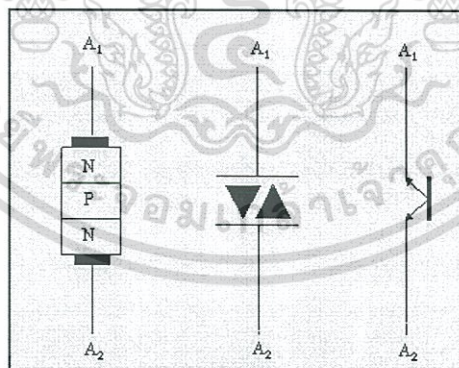
ไดแอก(Diac) เป็นอุปกรณ์สารกึ่งตัวนำที่อยู่ในกลุ่มของของไตรสเตอร์ มี 2 ขั้วคือ ขั้วแอโนด 1 (A1) และขั้วแอโนด 2 (A2) เพราะไดแอกสามารถนำกระแสได้สองด้าน ไดแอกสามารถนำไปใช้กับแรงดันไฟฟ้าสลับและแรงดันไฟฟ้ากระแสตรงได้

โครงสร้าง

เป็นอุปกรณ์สารกึ่งตัวนำมี 3 ตอนใหญ่ชนิดสาร PNP และยังประกอบด้วยสารกึ่งตัวนำ 2 ตอนย่อย ชนิด N ต่อร่วมในสารกึ่งตัวนำชนิด P ทั้ง 2 ตอนด้านนอก มีขาต่อออกมาใช้งานเพียง 2 ขา แต่ละขาที่ต่อใช้งานจะต่อร่วมกับสารกึ่งตัวนำทั้งชนิด N และชนิด P จึงทำให้ไดแอก สามารถทำงานได้ทั้งแรงดันไฟบวกและแรงดันไฟลบ ขาแอโนด 1 (A1) เรียกว่า ขาเทอมินอล 1 ใช้ตัวย่อ MT1 และขาแอโนด 2 (A2) เรียกว่า ขาเทอมินอล 2 ใช้ตัวย่อ MT2 แต่ละขาสามารถต่อสลับกันได้

การทำงาน

ไดแอกมี 2 ขา แต่มีคุณสมบัติสามารถทำงานได้กับแรงดันช่วงบวก และแรงดันช่วงลบคือกระแสได้ 2 ทิศทาง ดังนั้นในการใช้งานจึงไม่จำเพาะเจาะจงในการต่อวงจร ใช้ขาด้านใดด้านหนึ่งต่อเข้าวงจรก็ได้คุณสมบัติเหมือนกัน การทำงานของไดแอกเปรียบเหมือนกับชอคเลย์ไดโอด 2 ตัวต่อกลับหัวกัน มีวงจรสมมูลของไดแอกแทนด้วยทรานซิสเตอร์ 4 ตัว

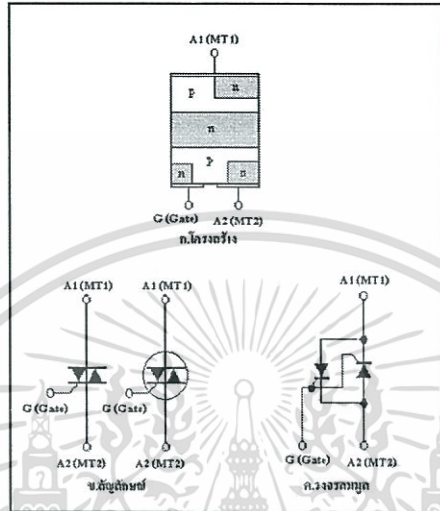


รูปที่ 2.10 Diac

2.2.1.4 Triac

เป็นอุปกรณ์นำพวกระบบสารกึ่งตัวนำในกลุ่มของไตรสเตอร์ มีลักษณะ โครงสร้างภายในคล้ายกับไดแอก แต่มีขาเกตเพิ่มขึ้นมาอีก 1 ขา ไตรแอก ถูกสร้างขึ้นเพื่อแก้ไข ข้อบกพร่องของ SCR ซึ่งไม่สามารถนำกระแสในซีกลบของไฟฟ้าสลับได้ การนำ ไตรแอก ไปใช้งานส่วนใหญ่จะใช้ทำเป็น เอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรควบคุมการทำงานเป็นสวิตช์ต่อแรงดันไฟสลับ ไตรแอก ถูกสร้าง ขึ้นมาให้ใช้งานกับกระแสสูง ๆ ดังนั้นต้องระวังเรื่องของการระบายความร้อน สัญลักษณ์,โครงสร้างและวงจรสมมูลของไตรแอก ดังรูปที่ 2.11

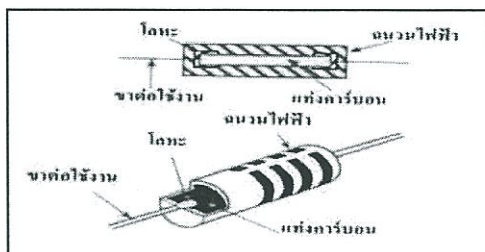


รูปที่ 2.11 โครงสร้าง สัญลักษณ์ และวงจรสมมูลของ Triac

2.2.1.5 Resister

เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่งที่มีคุณสมบัติในการต้านทานการไหลผ่านของกระแสไฟฟ้า แสดงได้ดังรูปที่ 2.12 ทำด้วยลวดต้านทานหรือถ่านคาร์บอน เป็นต้น นั่นคือ ถ้าอุปกรณ์นั้นมีความต้านทานมาก กระแสไฟฟ้าที่ไหลผ่านจะน้อยลง เป็นอุปกรณ์ไฟฟ้าชนิดพาสซีฟ สองขั้ว ที่สร้างจากความต่างศักย์ไฟฟ้าคร่อมขั้วทั้งสอง (V) โดยมีสัดส่วนมากน้อยตามปริมาณกระแสไฟฟ้าที่ไหลผ่าน (I) อัตราส่วนระหว่างความต่างศักย์ และปริมาณกระแสไฟฟ้า ก็คือ ค่าความต้านทานทางไฟฟ้า หรือค่าความต้านทานของตัวนำมีหน่วยเป็นโอห์ม (สัญลักษณ์ : Ω) เขียนเป็นสมการตามกฎของโอห์ม ดังนี้

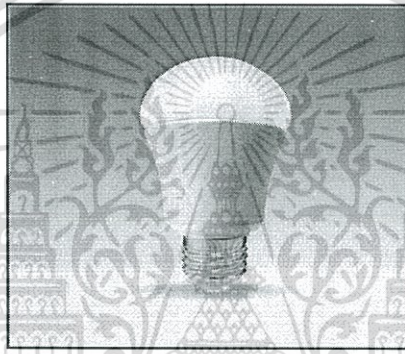
$$R = V/I$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการนำไปใช้ประโยชน์ด้านการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาตให้ นำไปใช้ประโยชน์ด้านการศึกษา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.6 หลอดไฟ LED

หลอดไฟ LED เป็นเทคโนโลยีที่ประหยัดการใช้ไฟฟ้ากว่าหลอดแบบสมัยเก่า ได้มากกว่า 90% การใช้ไฟฟ้าที่น้อยลงไปไม่ได้แค่ช่วยประหยัดเงินในธุรกิจ ยังเป็นการช่วยสิ่งแวดล้อมของโลกได้อีกอย่างหนึ่งด้วย อย่างลึ้มว่าค่าไฟฟ้า 30-40% จะต้องจ่ายไปเมื่อระบบแสงสว่างในร้านค้า อยู่แล้ว หลอดไฟ LED จะสามารถเติมเต็มในทุกส่วนเพื่อแทนหลอดไฟแบบเก่าที่จะหริ่แสงได้อย่างสิ้น ไหลไม่กระตุนั่นก็คือ หลอดไส้ ให้ได้ในที่สุด เนื่องด้วยความเป็นจริงแล้วหลอดไส้ที่หริ่แสงได้นั้น คือ หลอดไฟประเภทที่กินพลังงานไฟฟ้าสูงสุด แม้ว่าจะมีการหริ่แสงให้เบาลงมาแล้วก็ตาม เพราะ มีความร้อนสูงจากตัวหลอด และเกิดปัญหาเรื่องหลอดไส้ขาดได้ง่าย



รูปที่ 2.13 หลอดไฟ LED

2.2.2 อุปกรณ์ใน วงจรมอเตอร์

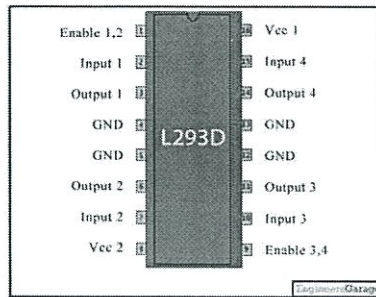
2.2.2.1 IC L293D

L293D เป็น IC ชนิด dual H-bridge มอเตอร์ โดยทำหน้าที่เป็นตัวขยาย กระแส โดยที่ทำการควบคุมกระแสที่มีค่าต่ำ และผลิตกระแสที่ค่าสูงๆ โดยใช้กระแสสูงในการ ขับเคลื่อนมอเตอร์

L293D ประกอบด้วย h-bridge driver 2 ตัว โดยมอเตอร์กระแสตรง สามารถทำงานพร้อมกันในทิศทาง forward และ reverseซึ่งมอเตอร์2ตัวจะควบคุมด้วยค่าลอจิก โดย อินพุตลอจิกจะอยู่ที่ขา 2&7 10&15 อินพุต 00 หรือ 11 จะเป็นคำสั่งหยุดมอเตอร์ ลอจิก01และ10 จะเป็นคำสั่งหมุนตามเข็มนาฬิกาและทวนเข็มนาฬิกา ตามลำดับ

Enable pins 1 และ 9 ต้องมีค่าที่สูงสำหรับการสั่งการให้มอเตอร์เริ่มทำงาน เมื่อenable input มีค่าสูง มอเตอร์ก็จะสามารถทำงานได้ และเอาท์พุตก็จะเป็น active และ ทำงานในเฟสเดียวกันกับอินพุต และถ้า enable input มีค่าต่ำ มอเตอร์ก็จะไม่ทำงาน output จะเป็น off และเป็นสถานะ impedance สูงด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 โครงสร้าง L293D

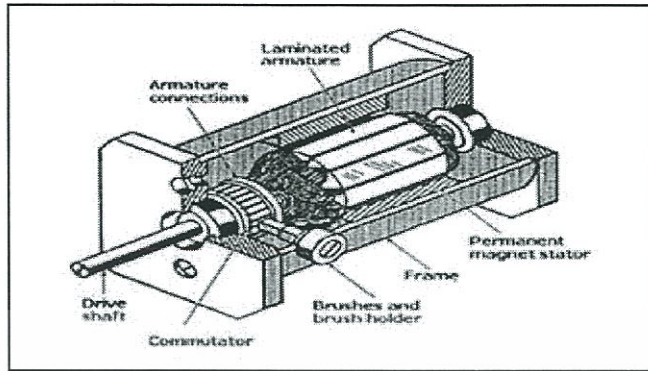
Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	Vcc ₂
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 2	Input 3
11	Output 1 for Motor 2	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 2	Output 4
15	Input 2 for Motor 2	Input 4
16	Supply voltage; 5V (up to 36V)	Vcc ₁

ตารางที่ 2.1 โครงสร้าง L293D

2.2.2.2 มอเตอร์กระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง เป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรมเพราะมีคุณสมบัติที่โดดเด่นในด้านการปรับความเร็วได้ตั้งแต่ความเร็วต่ำสุดจนถึงสูงสุด นิยมใช้กันมากในโรงงานอุตสาหกรรม เช่น โรงงานทอผ้า โรงงานเส้นใยโพลีเอสเตอร์ โรงงานถลุงโลหะหรือให้ เป็นต้นกำลังในการขับเคลื่อนรถไฟฟ้า เป็นต้นในการศึกษาเกี่ยวกับมอเตอร์ไฟฟ้ากระแสตรงจึงควรรู้จัก อุปกรณ์ต่าง ๆ ของมอเตอร์ไฟฟ้ากระแสตรงและเข้าใจถึงหลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบต่าง ๆ ดังรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 โครงสร้าง มอเตอร์กระแสตรง

2.2.3 โปรแกรม Altium Designer



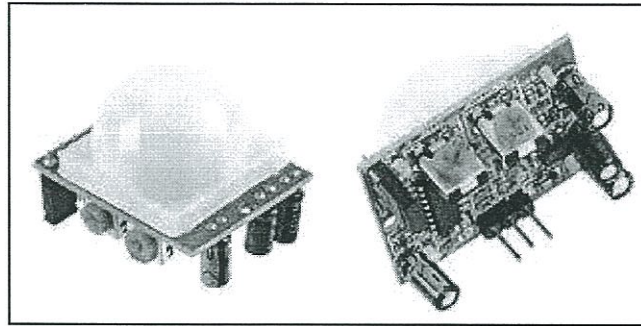
รูปที่ 2.16 โปรแกรม Altium Designer เป็นโปรแกรมออกแบบวงจร

2.2.4 Module ต่างๆ ใน Arduino

2.2.4.1 PIR Module (เซนเซอร์ตรวจจับความเคลื่อนไหว)

ตรวจจับความเคลื่อนไหวจากความร้อน เมื่อมีคนเดินผ่านก็จะจับค่าความร้อนที่เปลี่ยนแปลง แล้วส่งค่าสัญญาณมีไฟ ออกมา ในกรณีที่ไม่มี การเปลี่ยนแปลงส่งค่าไม่มีไฟ ออกมา เราสามารถนำค่านี้ไปส่งควบคุม Arduino ได้ สามารถปรับเวลาหน่วงเวลาในการตรวจจับครั้งต่อไปได้ ปรับระยะทางการตรวจจับได้ 3-7 เมตร มีช่องให้ต่อ LDR เพิ่ม เพื่อให้ทำงานตรวจจับแค่ตอนกลางคืน ดังรูปที่ 2.17 [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 PIR Module

2.2.4.2 Speaker Module

โมดูล Active Buzzer ใช้ไฟเลี้ยง 3.3 - 5V สามารถสร้างเสียงเตือนได้อย่างง่าย ๆ เพียงแค่จ่ายไฟ เข้าไปที่ ขา I/O โมดูลนี้มีทรานซิสเตอร์เบอร์ 9012 ช่วยขยายสัญญาณจึงมีความดังเป็นพิเศษ [6]

module interface specifications (3 wire):

1. VCC external 3.3 V-5 V voltage (can be directly and 5 V single-chip microcomputer and 3.3 V single chip connected)
2. GND external GND
3. I/O external microcontroller IO mouth



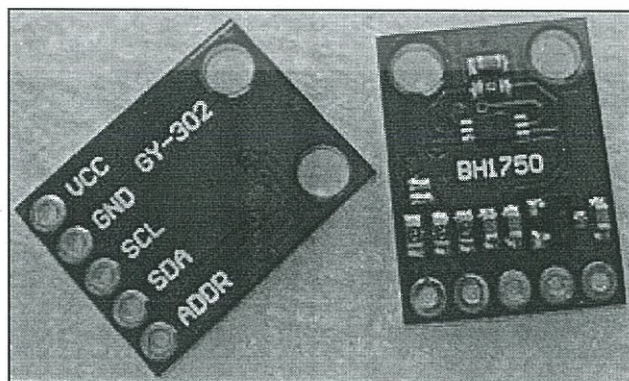
รูปที่ 2.18 Speaker Module

2.2.4.3 BH1750 Module

ในงานที่ต้องการวัดและเปรียบเทียบความเข้มแสงหรือตรวจจับการเปลี่ยนแปลงของปริมาณแสงนั้น จำเป็นต้องมีอุปกรณ์หรือเซนเซอร์ที่ใช้สำหรับวัดคุณสมบัติดังกล่าว อาจเลือกใช้ตัวต้านทานไวแสง (LDR: Light Dependent Resistor) ที่นำมาต่อกับตัวต้านทาน เพื่อใช้เป็นวงจร Voltage Divider แล้วใช้ ADC (Analog-to-Digital Converter) ภายนอก หรือ ภายใน ไมโครคอนโทรลเลอร์ (MCU) วัดระดับแรงดันของสัญญาณที่ได้ หรือจะเลือกใช้ไอซีสำหรับวัดความเข้มแสงโดยเฉพาะ ซึ่งจะได้เอาต์พุตแบบแอนาล็อกหรือดิจิทัลแล้วแต่อุปกรณ์ [7]

รายละเอียดของไอซี BH1750FVI

- เชื่อมต่อแบบ I²C (มีขา SCL และ SDA)
- ใช้งานและเชื่อมต่อแบบ I²C slave device
- ความเร็วสำหรับบัส I²C ได้ถึง 400kHz
- มีขา ADDR สำหรับต่อกับ LOW หรือ HIGH เพื่อใช้กำหนดเลขที่อยู่ของ I²C Slave (สามารถกำหนดเลขที่อยู่ได้สองค่า)
- แรงดันไฟเลี้ยงในช่วง 2.4V - 3.6V
- เลือกโหมดการวัดได้ (Measurement Mode) เช่น H-resolution Mode, H-resolution Mode2, L-resolution Mode ซึ่งเป็นตัวกำหนดความละเอียดในการวัด รวมถึงระยะเวลาในการวัดและอ่านค่า (measurement time)
- ความละเอียด: 16 บิต ได้ค่า 1-65536 หน่วยเป็น Lux (step: 0.5 Lux, 1 Lux, หรือ 4 Lux ขึ้นอยู่กับโหมดการวัดที่เลือก)
- ระยะเวลาในการวัดแต่ละครั้ง: ประมาณ 120 msec (สำหรับ 0.5 Lux หรือ 1 Lux), 16 msec (สำหรับ 4 Lux) ขึ้นอยู่กับโหมดการวัดที่เลือก



รูปที่ 2.19 BH1750 Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4.4 Ultrasonic Module

โมดูลอัลตราโซนิกนี้เป็นอุปกรณ์ใช้วัดระยะทางโดยไม่ต้องมีการสัมผัสกับตำแหน่งที่ต้องการวัด วัดได้ตั้งแต่ 2 cm ถึง 400 cm โดยส่งสัญญาณอัลตราโซนิก ความถี่ 40 kHz ไปที่วัตถุที่ต้องการวัดและรับสัญญาณที่สะท้อนกลับมา พร้อมทั้งจับเวลาเพื่อนำมาใช้ในการคำนวณระยะทาง [8]

การต่อเพื่อใช้งานทำได้โดยต่อ PIN ดังนี้

- PIN ที่ 1 -- 5V Supply
- PIN ที่ 2 -- Trigger Pulse Input
- PIN ที่ 3 -- Echo Pulse Output
- PIN ที่ 4 -- GND



รูปที่ 2.20 Ultrasonic Module

2.2.4.5 ESP8266 Module

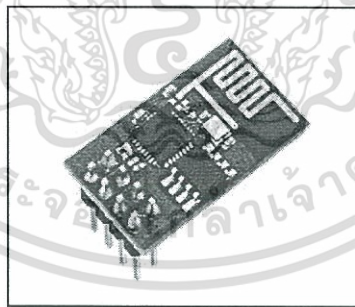
ESP8266 คือโมดูล Wifi ที่มีความพิเศษตรงที่ตัวมันสามารถโปรแกรมลงไปได้ ทำให้สามารถนำไปใช้งานแทนไมโครคอนโทรลเลอร์ได้เลย และมีพื้นที่โปรแกรมที่มากถึง 4MB ทำให้มีพื้นที่เหลือมากในการเขียนโปรแกรมลงไป ESP8266 เป็นชื่อของชิปไอซีบนบอร์ดของโมดูล ซึ่งไอซี ESP8266 ไม่มีพื้นที่โปรแกรม (flash memory) ในตัว ทำให้ต้องใช้ไอซีภายนอก (external flash memory) ในการเก็บโปรแกรม ที่ใช้การเชื่อมต่อผ่านโปรโตคอล SPI ซึ่งสาเหตุนี้เองทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่าไอซีไมโครคอนโทรลเลอร์เบอร์อื่นๆ ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3V - 3.6V การนำไปใช้งานร่วมกับเซ็นเซอร์อื่นๆที่ใช้แรงดัน 5V ต้องใช้วงจรแบ่งแรงดันมาช่วย เพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200mA ความถี่คริสตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

40MHz ทำให้เมื่อนำไปใช้งานอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การแสดงผลข้อมูลรวดเร็วกว่าไมโครคอนโทรลเลอร์ Arduino มาก

ขาของโมดูล ESP8266 แบ่งได้ดังนี้

- VCC เป็นขาสำหรับจ่ายไปเข้าเพื่อให้โมดูลทำงานได้ ซึ่งแรงดันที่ใช้งานได้คือ 3.3 - 3.6V
- GND
- Reset และ CH_PD (หรือ EN) เป็นขาที่ต้องต่อเข้าไฟ + เพื่อให้โมดูลสามารถทำงานได้ ทั้ง 2 ขานี้สามารถนำมาใช้รีเซ็ตโมดูลได้เหมือนกัน แตกต่างตรงที่ขา Reset สามารถลอยไว้ได้ แต่ขา CH_PD (หรือ EN) จำเป็นต้องต่อเข้าไฟ + เท่านั้น เมื่อขานี้ไม่ต่อเข้าไฟ + โมดูลจะไม่ทำงานทันที
- GPIO เป็นขาดิจิตอลอินพุต / เอาต์พุต ทำงานที่แรงดัน 3.3V
- GPIO15 เป็นขาที่ต้องต่อลง GND เท่านั้น เพื่อให้โมดูลทำงานได้
- GPIO0 เป็นขาสำหรับการเลือกโหมดทำงาน หากนำขานี้ลง GND จะเข้าโหมดโปรแกรม หากลอยไว้ หรือนำเข้าไฟ + จะเข้าโหมดการทำงานปกติ
- ADC เป็นขานาล็อกอินพุต รับแรงดันได้สูงสุดที่ 1V ขนาด 10 บิต การนำไปใช้งานกับแรงดันที่สูงกว่าต้องใช้วงจรแบ่งแรงดันเข้าช่วย [9]



รูปที่ 2.21 ESP8266 Module

2.3 โปรแกรม Android studio 1.3.2

2.3.1 การติดตั้งโปรแกรมในการทำงาน

2.3.1.1 System requirement

สำหรับอุปกรณ์พื้นฐานที่สามารถดาวน์โหลดและติดตั้งตัวโปรแกรม

Android Studio ได้ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.1.1 สำหรับอุปกรณ์ที่ทำงานบน Windows ควรมีอุปกรณ์ขั้นพื้นฐาน
ดังนี้

Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit), 2 GB RAM minimum, 4 GB RAM recommended, 400 MB hard disk space + At least 1 G for Android SDK, emulator system images, and caches, 1280 x 800 minimum screen resolution, Java Development Kit (JDK) 7, Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

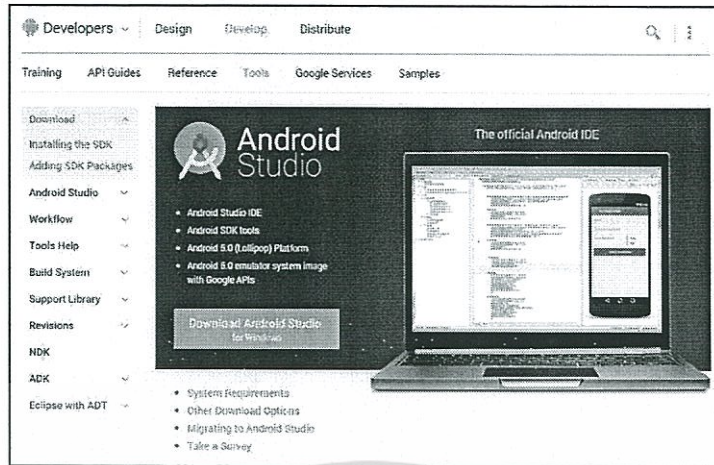
2.3.1.1.2 สำหรับอุปกรณ์ที่ทำงานบน Mac OS X ควรมีอุปกรณ์ขั้นพื้นฐาน
ดังนี้

Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks), 2 GB RAM minimum, 4 GB RAM recommended, 400 MB hard disk space + At least 1 GB for Android SDK, emulator system images, and caches, 1280 x 800 minimum screen resolution, Java Runtime Environment (JRE) 6, Java Development Kit (JDK) 7, Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

2.3.1.1.3 สำหรับอุปกรณ์ที่ทำงานบน Linux ควรมีอุปกรณ์ขั้นพื้นฐานดังนี้
GNOME or KDE desktop, GNU C Library (glibc) 2.11 or later, 2 GB RAM minimum, 4 GB RAM recommended, 00 MB hard disk space + At least 1 GB for Android SDK, emulator system images, and caches, 1280 x 800 minimum screen resolution, Oracle® Java Development Kit (JDK) 7

2.3.1.2 การติดตั้งอุปกรณ์

ในการติดตั้งควรเชื่อมต่อกับอินเทอร์เน็ตด้วย เพราะจะมีบางขั้นตอนที่ต้องดาวน์โหลดข้อมูลเพิ่มจากอินเทอร์เน็ต โดยสามารถดาวน์โหลดโปรแกรมได้จาก <https://developer.android.com/intl/ja/sdk/index.html> ดังรูปที่ 2.22



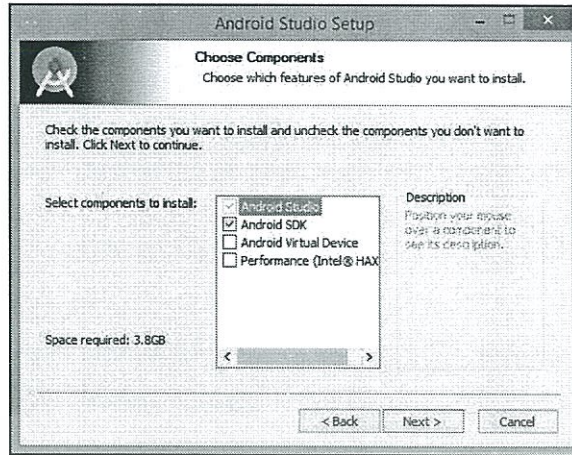
รูปที่ 2.22 หน้า Homepage ของ Android studio

เมื่อทำการดาวน์โหลดเสร็จเรียบร้อยแล้วให้เปิดไฟล์ขึ้นมา ก็จะพบกับหน้า Welcome ให้กดปุ่ม Next เพื่อไปขั้นตอนต่อไป ดังแสดงในรูป 2.23



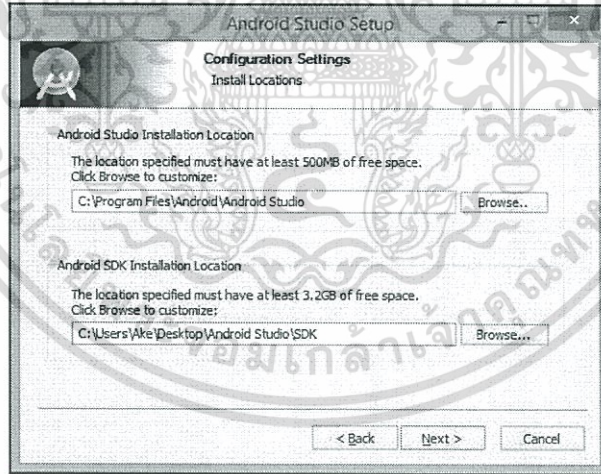
รูปที่ 2.23 หน้าต่างแรกในการติดตั้งโปรแกรม

หน้าต่อมาจะให้เลือกว่าต้องการติดตั้งอะไรบ้าง ส่วน Android Virtual Device (AVD) กับ Performance (Intel®HAXM) ไม่จำเป็นต้องติดตั้ง เนื่องจากเป็นตัว Emulator และตัวช่วยที่ทำให้ Emulator ทำงานได้ไวขึ้น ดังรูปที่ 2.24



รูปที่ 2.24 หน้าต่างการเลือกติดตั้งอุปกรณ์เสริม ในโปรแกรม

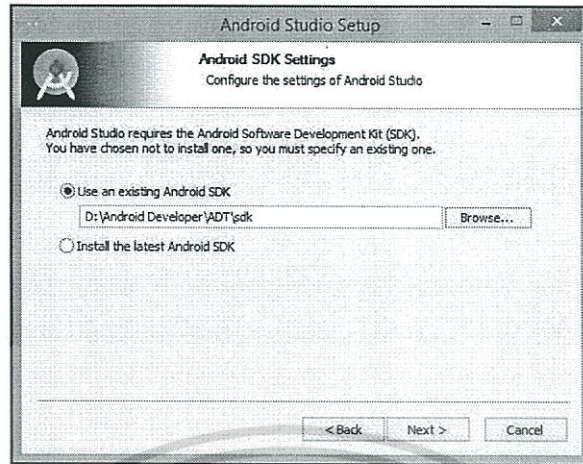
ต่อมาจะเป็นหน้ากำหนด Path ที่ต้องการติดตั้ง โดยกรณีที่เลือกติดตั้ง Android Studio และ Android SDK ซึ่งในการกำหนด Path ของ Android Studio และ Android SDK ห้ามเลือก Path ของ Android SDK ให้อยู่ใน Android Studio เพราะจะทำให้ไม่สามารถอัปเดต Android Studio ได้ (โปรแกรมจะมีแจ้งเตือนอยู่) ดังนั้นควรแยกโฟลเดอร์ทั้งสองไว้เลยดังรูปที่ 2.25 [10]



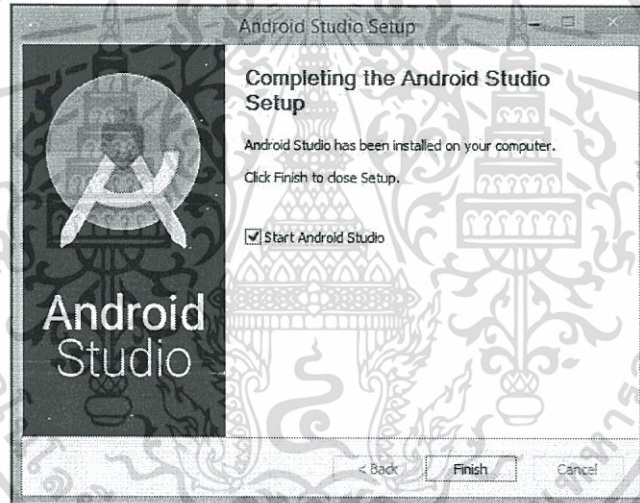
รูปที่ 2.25 หน้าต่างเลือก path ในกรณีเลือกติดตั้ง ทั้ง Android SDK และ Android studio

และในกรณีที่ติดตั้งเฉพาะ Android Studio โปรแกรมจะถาม Path ของ Android SDK ที่มีอยู่ในเครื่องแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 หน้าต่างการเลือกติดตั้งเฉพาะ Android studio



รูปที่ 2.27 หน้าต่างการแสดงว่าการติดตั้งเสร็จสิ้น โปรแกรมพร้อมเข้าใช้งาน

ส่วนในรูปที่ 2.26 และ 2.27 จะเป็นการให้เราติดตั้งโปรแกรม Android studio ไว้ที่ไหน จากนั้นก็เสร็จสิ้นการ install เรียบร้อย พร้อมใช้งาน

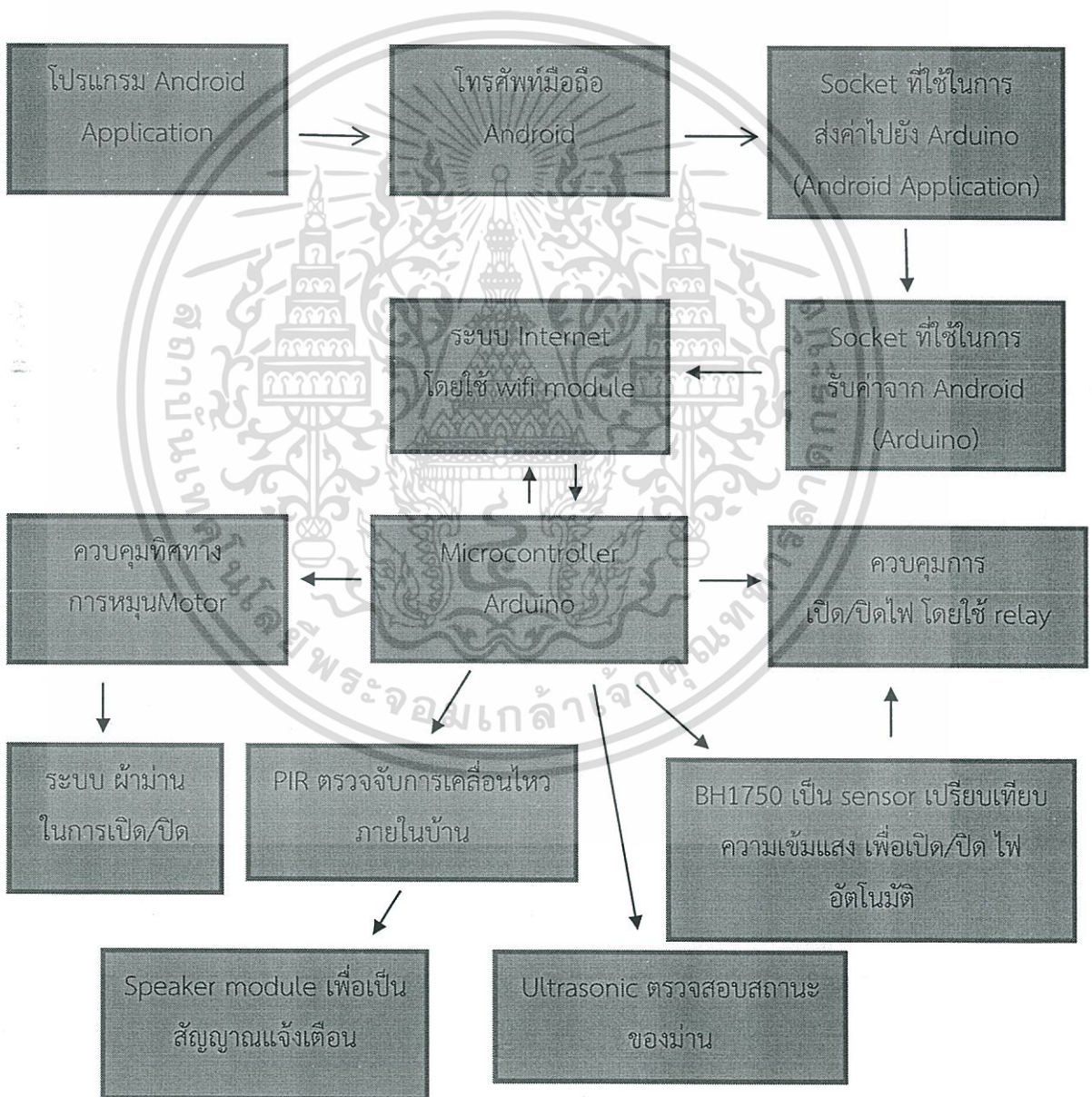
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

อธิบาย การทำงานของ Block Diagram และ Flow Chart

3.1 การออกแบบ



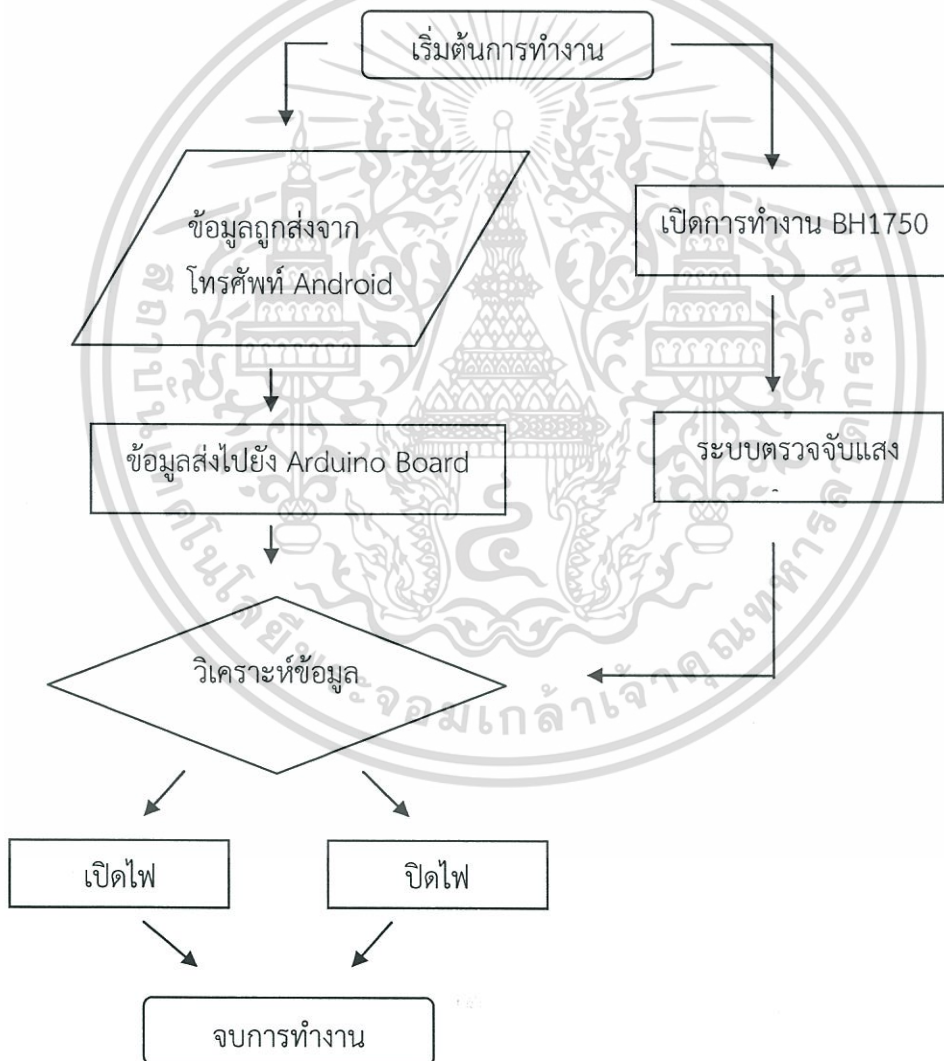
รูปที่ 3.1 Block Diagram ของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 แสดงภาพรวมของโครงการนี้ก็คือเราจะใช้โทรศัพท์มือถือในการควบคุมในการเปิด/ปิดไฟในระบบควบคุมแสง และสามารถเลื่อนผ่านมันได้ โดยเชื่อมกับ module wifi esp8266 ในการ connect ระหว่าง Arduino และ Android Application และเมื่อควบคุมแสงได้ ระบบก็จะสามารถ detect สิ่งแปลกปลอมเข้ามาในบ้านได้ เช่น จะมีเสียงดังจาก BUZZER เพื่อป้องกันการบุกรุก และถ้าอากาศมืดไฟจะสว่าง เป็นต้น

3.1.1 ผังการทำงานต่างๆในระบบ

3.1.1.1 การทำงานของระบบเปิด/ปิดไฟ

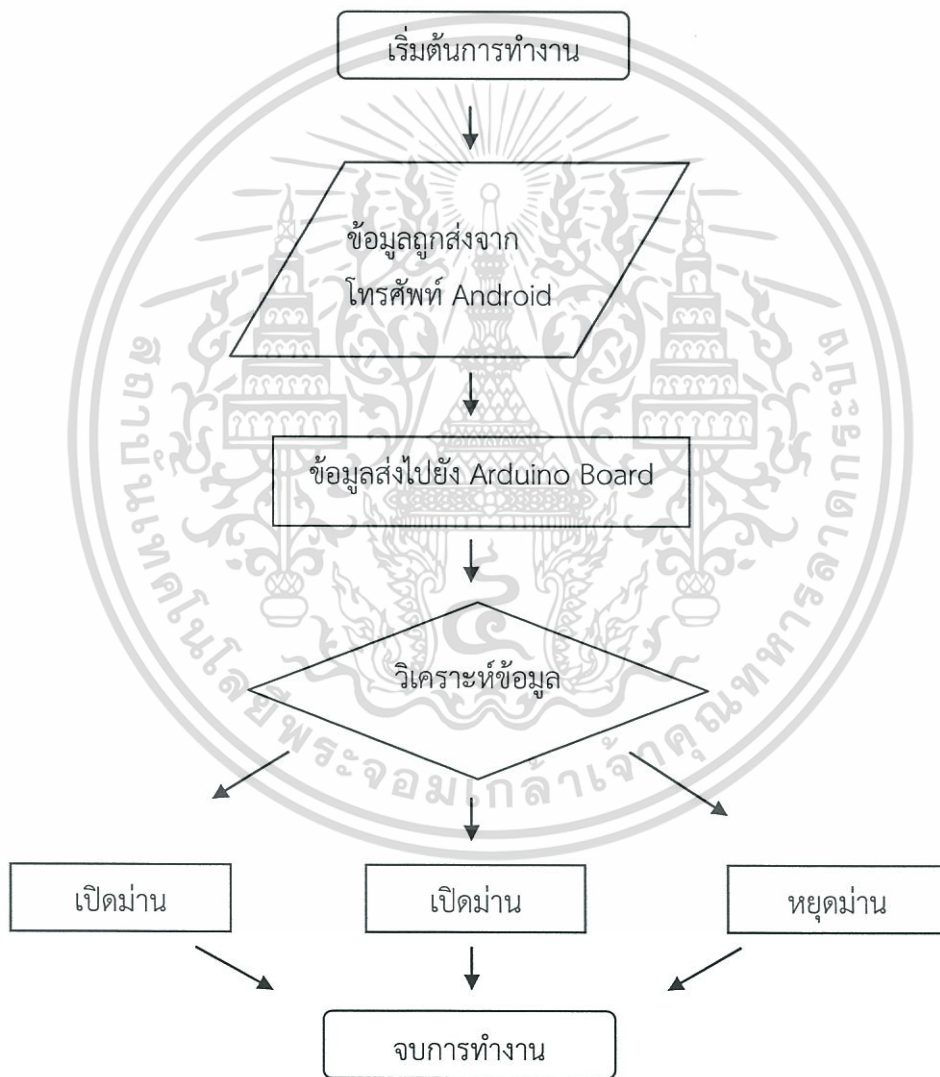


รูปที่ 3.2 ผังการทำงานของระบบเปิด/ปิดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 แสดงการทำงานจะแบ่งเป็น 2 ส่วน ส่วนแรก ข้อมูลจะถูกส่งจาก โทรศัพท์ Android ไป Arduino Board และจะทำการวิเคราะห์ข้อมูลที่ส่งมา เพื่อมาแสดงผล เปิด/ปิดไฟ ส่วนที่สอง ระบบตรวจจับแสงภายนอกบ้าน การทำงาน เมื่อเปิดสวิตช์ระบบจะทำงานอัตโนมัติ โดยจะตรวจจับความสว่างภายนอกบ้าน ถ้ามีแสงไฟจะปิด ถ้าไม่มีแสง ไฟจะเปิดเองโดยอัตโนมัติ

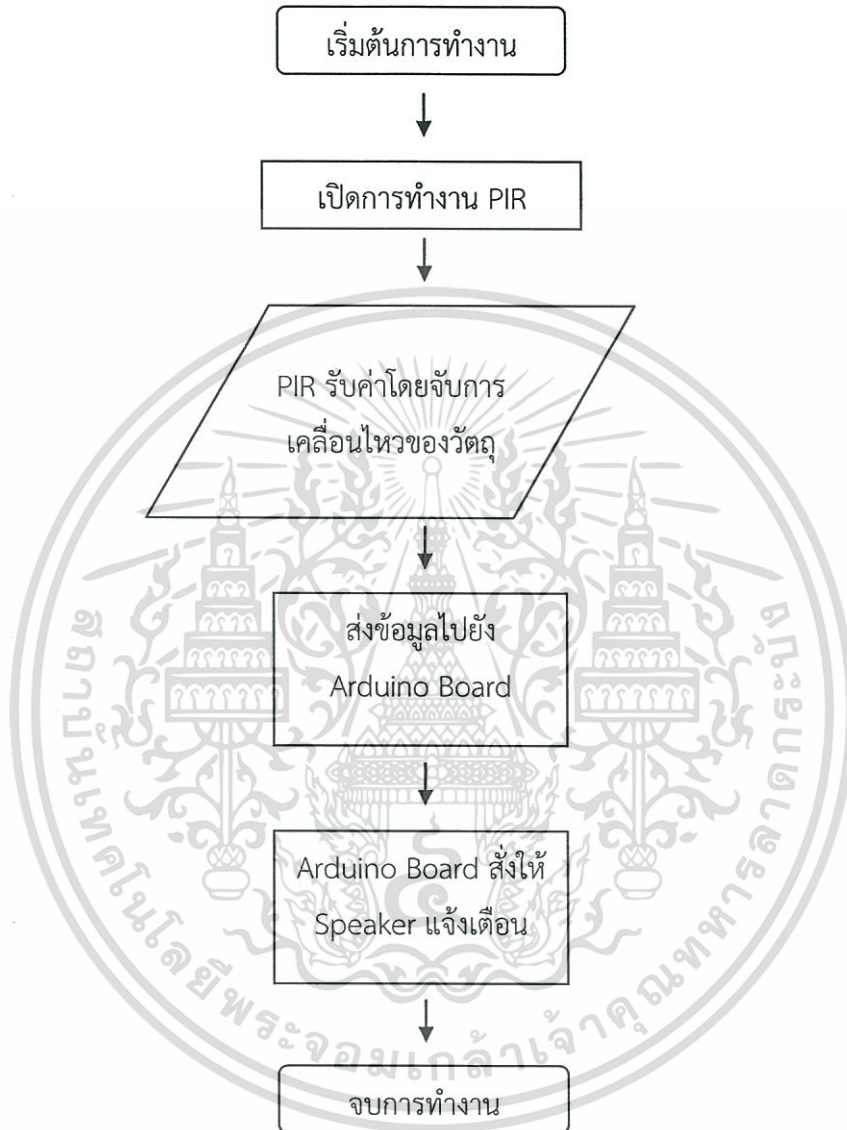
3.1.1.2 การทำงานของระบบม่าน



รูปที่ 3.3 ผังการทำงานของระบบเปิด/ปิดม่าน

รูปที่ 3.3 แสดงการทำงาน ข้อมูลจะถูกส่งจากโทรศัพท์ Android ไป Arduino Board และจะทำการวิเคราะห์ข้อมูลที่ส่งมา เพื่อมาแสดงผล เปิด/ปิด/หยุดม่าน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

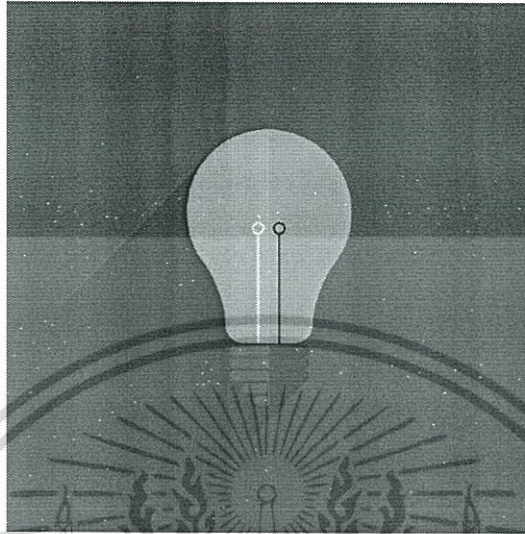
3.1.1.3 การทำงานของ ระบบตรวจจับความเคลื่อนไหว



รูปที่ 3.4 ผังการทำงานของระบบตรวจจับการเคลื่อนไหว

รูปที่ 3.4 แสดงการทำงาน เมื่อมีวัตถุผ่านเข้ามาในระยะตรวจจับของ PIR จากนั้น Arduino Board จะประมวลผล และสั่งให้ Speaker ทำงาน

3.1.2 การออกแบบ แอปพลิเคชันใน Android



รูปที่ 3.5 หน้าแอปพลิเคชัน บน Android



รูปที่ 3.6 ภายในแอปพลิเคชัน

โดยการทำงาน เราต้องพิมพ์ IP เพื่อให้ระบบเชื่อมต่อการทำงาน จากนั้นจึงสามารถควบคุมระบบผ่าน และไฟ ผ่านทาง แอปพลิเคชันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 โปรแกรม Android Studio

3.2.2 โปรแกรม Altium Designer

3.2.3 โปรแกรม Arduino

3.2.4 Arduino UNO r3

3.2.5 วงจรเปิด/ปิด ไฟ

- 1) Relay
- 2) Diode 1N4148
- 3) Triac 2N3904
- 4) Resister
- 5) Terminal
- 6) หลอดไฟ LED

3.2.6 วงจรมอเตอร์

- 1) DC motor 12 V rpm:150
- 2) L293D IC
- 3) Terminal

3.2.7 Arduino Module

- 1) PIR Module
- 2) Speaker Module
- 3) BH1750 Module
- 4) Ultrasonic Module
- 5) ESP8266 Wifi Module

3.2.8 อุปกรณ์ทำ model

3.3 การจัดเก็บผลการทดลอง

- 1.ระบบสามารถ เปิด/ปิด ไฟได้ ผ่านแอปพลิเคชันบนโทรศัพท์ Android และสามารถเปิด/ปิด ไฟ อย่างอัตโนมัติ โดยใช้ระบบตรวจจับแสง
- 2.ระบบสามารถเลื่อน เปิด/ปิด/หยุด ม่านได้ ผ่านแอปพลิเคชันบนโทรศัพท์ Android และสามารถตรวจจับได้ว่า ม่านนั้น เปิดหรือปิดอยู่
- 3.ระบบตรวจจับความเคลื่อนไหว โดยใช้ PIR มีการแจ้งเตือนเมื่อมีวัตถุผ่านระบบตรวจจับให้ ผู้ใช้งานรู้



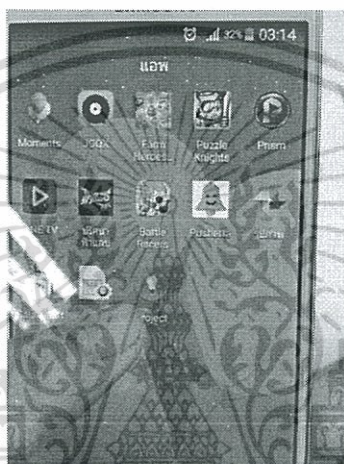
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

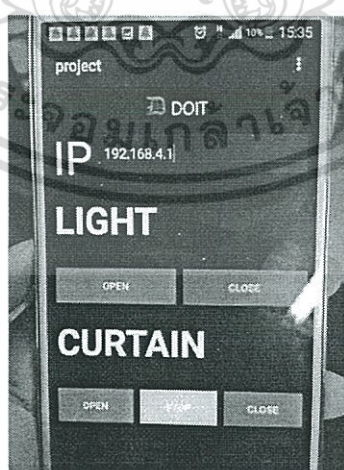
4.1 การเข้าสู่ระบบ

เมื่อทำการอัปเดตแอปพลิเคชัน ลงโทรศัพท์มือถือ Android แล้ว ให้กดเข้าไปภายในแอปพลิเคชัน ดังรูปที่ 4.1



รูปที่ 4.1 แอปพลิเคชัน

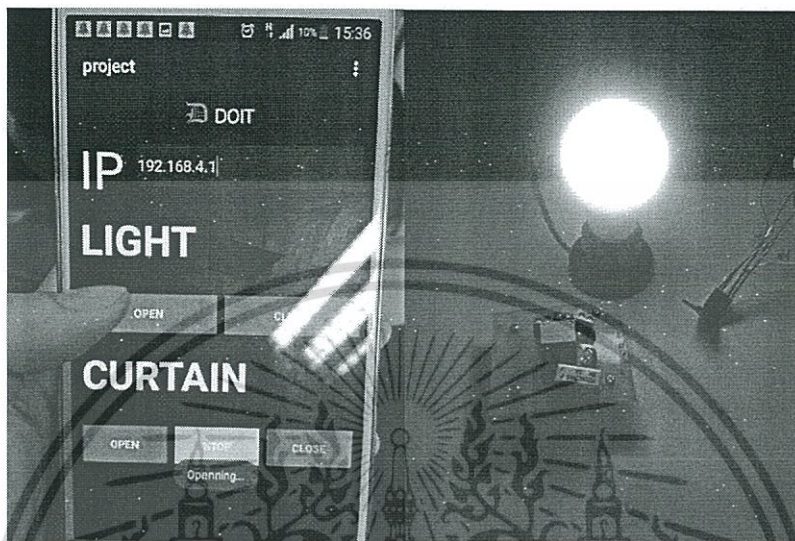
หลังจากเข้าไปใน แอปพลิเคชัน พิมพ์ IP เพื่อเชื่อมต่อกับ Wifi Module โดย IP ที่เชื่อมต่อคือ 192.168.4.1 ดังรูปที่ 4.2



รูปที่ 4.2 ภายในแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การแสดงผลของระบบไฟ



รูปที่ 4.3 แสดงการเปิดไฟผ่านแอปพลิเคชัน

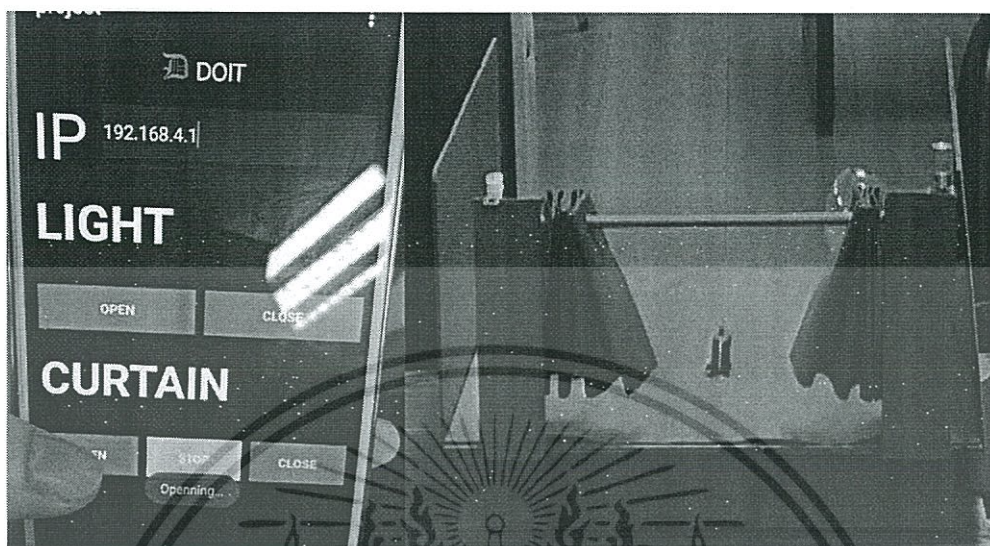


รูปที่ 4.4 แสดงการปิดไฟผ่านแอปพลิเคชัน

โดยจะมี อุปกรณ์ตรวจจับแสง ซึ่งทำให้สามารถหลอดไฟ LED สามารถเปิด/ปิดไฟได้อย่างอัตโนมัติ โดยจะควบคุมผ่านใน App Android Studio ผ่านทางโทรศัพท์มือถือหรือทาง Arduino ก็ได้เช่นกัน ดังรูปที่ 4.3 และ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การแสดงผลของระบบม่าน



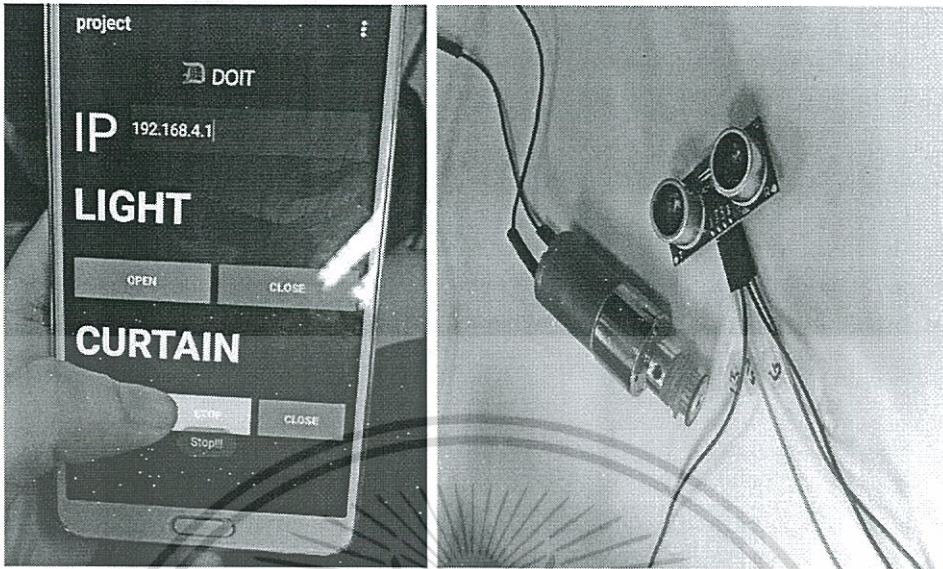
รูปที่ 4.5 แสดงการเปิดม่านผ่านแอปพลิเคชัน



รูปที่ 4.6 แสดงการปิดม่านผ่านแอปพลิเคชัน

สำหรับระบบเปิด/ปิดผ้าม่านเราจะใช้ มอเตอร์ทรอปเป็นตัวควบคุมในการเปิด/ปิดผ้าม่าน โดยจะควบคุมผ่านใน App Android Studio ผ่านทางโทรศัพท์มือถือหรือทาง Arduino ก็ได้เช่นกัน ดังรูปที่ 4.5 และ 4.6

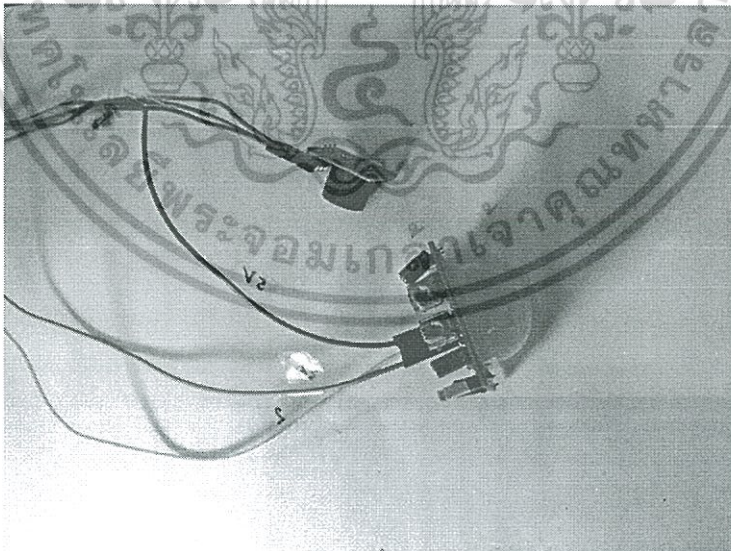
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงการหยุดมอเตอร์ผ่านทางแอปพลิเคชันโดยมี Ultrasonic เพื่อตรวจจับสถานะของม่าน

4.4 การแสดงผลของระบบตรวจจับความเคลื่อนไหว

ระบบนี้จะไม่เชื่อมต่อกับ โทรศัพท์มือถือ Android โดยเมื่อมีวัตถุผ่านตัวตรวจจับการเคลื่อนไหว จะมีเสียงสัญญาณแจ้งเตือนเพื่อป้องกันภัย ดังรูปที่ 4.8



รูปที่ 4.8 ระบบตรวจจับการเคลื่อนไหว และ Speaker เพื่อแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงการ ระบบควบคุมแสงในห้อง เป็นการสร้างระบบเพื่อควบคุม การเปิด/ปิด ของผ้าม่าน การเปิด/ปิด ของหลอดไฟ LED และ ระบบตรวจจับการเคลื่อนไหว ประกอบเป็นระบบแจ้งเตือนเพื่อรักษาความปลอดภัย โดยใช้ Microcontroller Arduino เป็นตัวควบคุมการทำงาน และสามารถควบคุมได้ผ่านทางโทรศัพท์มือถือ Android โดยใช้ระบบ Internet ระบบแรก การเปิด/ปิด ผ้าม่าน จะให้ Arduino ควบคุมมอเตอร์ เพื่อหมุนเปิด/ปิด/หยุด ม่าน และจะใช้ Ultrasonic สถานะของม่าน ว่าเปิดหรือปิดอยู่ ระบบที่สอง การเปิด/ปิด ของหลอดไฟ ระบบนี้จะมีการทำงานเป็นสองส่วน ส่วนแรกคือ ทำงานเปิด/ปิด โดยผ่านทางโทรศัพท์มือถือ และส่วนที่สอง เมื่อเราไม่อยู่บ้านสามารถกดเปิดระบบทำงานอัตโนมัติ โดยใช้ BH1750 เป็น sensor เพื่อควบคุมการเปิด/ปิด เมื่อมีด ไฟจะเปิดโดยอัตโนมัติ และระบบที่ 3 คือ ระบบตรวจจับการเคลื่อนไหว โดยใช้ PIR เมื่อมีสิ่งต่างๆผ่าน จะมีสัญญาณแจ้งเตือนเพื่อให้เจ้าของบ้านทราบ และทั้ง 3 ระบบนี้ จะสามารถสั่งการผ่าน โทรศัพท์มือถือ Android โดยผลที่ได้คือ สามารถควบคุมระบบทั้ง 3 ผ่านทางแอปพลิเคชันที่สร้างขึ้นบนโทรศัพท์มือถือ Android ได้ โดยต้องเข้ารหัส IP Address ของตัวอุปกรณ์ผ่านทาง Wifi ก่อนจึงสามารถใช้งานได้

5.2 ข้อเสนอแนะ

โครงการนี้เป็นโครงการจำลองโมเดล เพื่อสร้างระบบภายในบ้าน ซึ่งยังไม่ได้ลองใช้ในสถานที่จริง และผู้ที่รับผิดชอบโครงการก็ได้เริ่มจัดทำโครงการนี้โดยไม่เคยทำมาก่อนเลย แต่โครงการนี้ผู้จัดทำโครงการได้ตั้งใจทำออกมาเพื่อที่จะเป็นประสบการณ์ในการทำงานต่อไปและจบภาคการศึกษาโครงการนี้สามารถเปิด/ปิดไฟได้อัตโนมัติ, ควบคุมการเปิดปิดผ้าม่านและสามารถตรวจจับสิ่งแปลกปลอมที่เข้ามาในโมเดลแบบบ้านได้ เป็นการสร้างบ้านจำลองนั่นเอง หากผิดพลาดประการใดขออภัยไว้ ณ ที่นี้ด้วย และขอขอบคุณเพื่อนๆ อาจารย์และคนที่ให้คำปรึกษามา ณ โอกาสนี้

บรรณานุกรม

- [1] Administrator Arduino. <http://www.thaieasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html>.
- [2] Gravitechthaicom. SerialMonitor. <https://www.gravitechthai.com/guru2.php?p=271>.
- [3] Wikipediaorg. Relay. <https://th.wikipedia.org/wiki/รีเลย์>.
- [4] Wikipediaorg. Diode. <https://th.wikipedia.org/wiki/ไดโอด>.
- [5] Thaisensormodulecom. PIR Module. <http://thaisensormodule.com/index.php/light/pirsens1>.
- [6] Arduinoallcom. Buzzer Module. <http://www.arduinoall.com/product/536/active-buzzer-module-3-3-5v>.
- [7] Kmutnbac.th. Light sensor <http://cpre.kmutnb.ac.th/esl/learning/index.php?article=bh1750-light-sensor>.
- [8] Arduitrronicscom. Ultrasonic Module. <http://www.arduitronics.com/article/ultrasonic-ranging-module-hc-sr04>.
- [9] Ioxhopcom. ESP8266 Module. <http://www.ioxhop.com/article/esp8266-ตอนที่-1-รู้จักกับ-esp-และรุ่นที่นิยมใช้งาน>.
- [10] Android Method. Android Studio. <https://www.youtube.com/channel/UCkgnZDuS9fM951TpyBTkNPg?nohtml5=False>.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกัดปรินต์

วัสดุที่ใช้

1. กระดาษโฟโต้ (ขนาด A4)
2. เครื่องพิมพ์เลเซอร์
3. เตารีดแบบพื้นเรียบ (ไม่ใช่แบบไอน้ำ)
4. ภาชนะใส่น้ำ
5. ปากกาเขียนแผ่นซีดี
6. กรดกัดปรินต์
7. ทินเนอร์
8. กระดาษชำระ

ขั้นตอนการกัดปรินต์

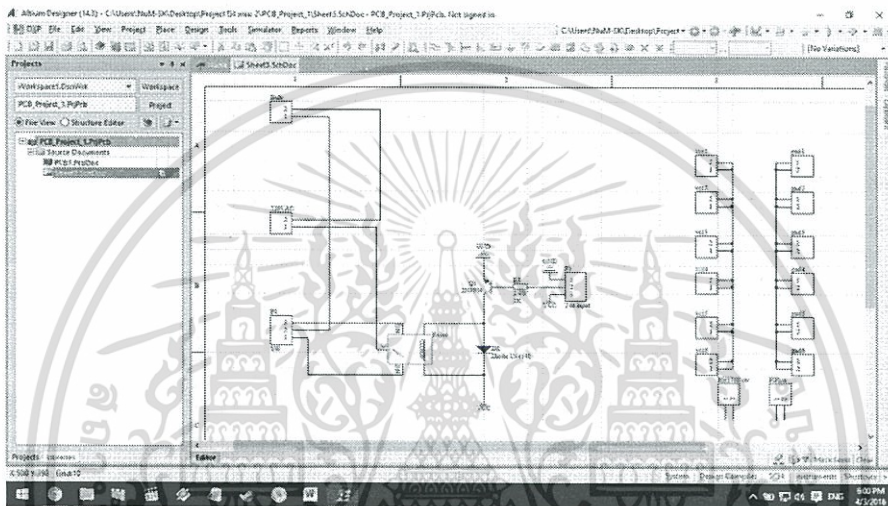
1. เริ่มจากออกแบบลายวงจรด้วยโปรแกรมออกแบบลายวงจรตามถนัดโดยใช้โปรแกรม Altium Designer
2. copy ลายวงจรทำให้ได้หลายๆแผ่นโดยใช้กระดาษแผ่นเดียว
3. พิมพ์ด้วยกระดาษโฟโต้(ด้านมัน) ด้วยเครื่องพิมพ์เลเซอร์ สิ่งที่ได้จะต้องเป็นลายตรงข้ามกับลายจริง
4. ตัดกระดาษต้นแบบเอาในส่วนของที่ต้องการนำไปทำงาน
5. ชัดปรินต์(ด้านทองแดง) ด้วยฝอยขัดหม้อหรือสก็อตไบต์กับน้ำ ใช้น้ำยาล้างจานช่วยด้วย นำแผ่นปรินต์วางคว่ำบนแผ่นต้นแบบโดยเอาด้านทองแดงสัมผัสกับแบบ
6. ตัดกระดาษกาวกันการเลื่อนขยับ
7. ตั้งความร้อนเตารีดค่อนข้างสูงหน่อย
8. ตัดกระดาษกาวกับพื้นกันเลื่อน
9. กดเตารีด(กดเฉยๆและหนักๆหน่อย) ใช้เวลาประมาณ 30 วินาที
10. รีดไปทั่วๆแผ่นโดยใช้แรงกดพอสมควร (ประมาณ 1.30 นาที) ตะแคงเตารีดเอาขอบด้านหัวเตารีด รีดกดไปทั่วๆแผ่น ย่ำๆ ซ้ำๆ (ประมาณ 1.30 นาที)
11. นำลงแช่น้ำโดยใช้เวลาแช่ 10 นาที
12. ใช้เข็มหรือขาไอซีเขี่ยรูโดนัทเนื่องจากชั้นตอนที่ผ่านมามันไม่ค่อยออก ใช้ปากกาทันน้ำ(เขียนซีดี)แต่เส้นที่ขาด
13. นำไปกัดด้วยกรดกัดปรินต์
14. เมื่อกัดปรินต์เสร็จ ใช้กระดาษชำระ(ทิชชู) 1 แผ่นพับให้เหลือสัก 1

ตารางนี้ขุบทินเนอร์เข็ดหมึก และเข็ดให้สะอาด เป็นอันว่าสำเร็จ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Altium Designer

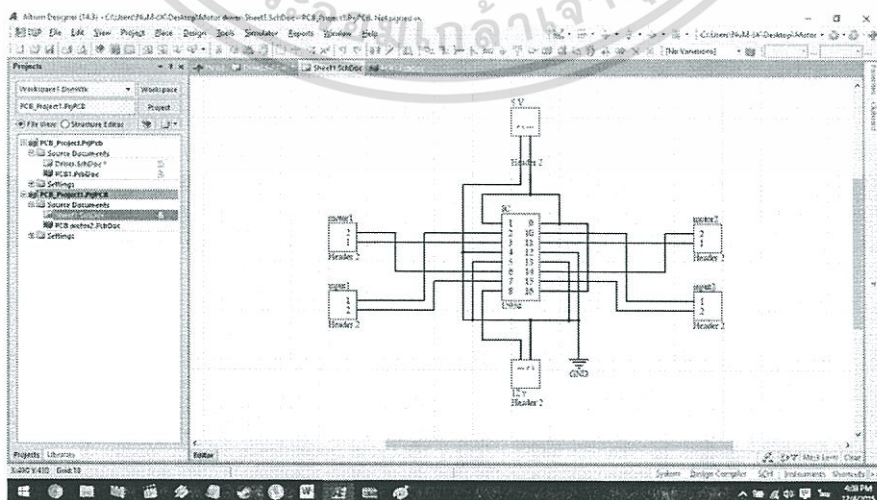
การออกแบบวงจร Relay และวงจรมอเตอร์ โดยใช้โปรแกรม Altium

เมื่อนำความรู้ที่ศึกษามาเกี่ยวกับวงจร Relay และวงจรมอเตอร์ มาสร้างเป็นวงจรที่สามารถใช้งานได้จริง อันดับแรก คือ เมื่อออกแบบวงจร Relay เพื่อใช้เปิด ปิดไฟ และการขับเคลื่อนของมอเตอร์สำเร็จ เขียนลายวงจรเป็น Schematic ใน Altium ดังรูปที่ 5.1



รูปที่ 5.1 Schematic ของวงจร Relay

วงจร Relay เปรียบเสมือน สวิตช์ตัดต่อวงจรโดยใช้แม่เหล็กไฟฟ้า วิธีการทำให้วงจร Relay ทำงานได้ ต้องจ่ายไฟให้มันตามที่กำหนด เมื่อจ่ายไฟจะทำให้เป็นวงจรปิด ซึ่งจะทำให้ทำงาน โดยเราจะนำวงจร Relay นี้มาสร้างเป็นระบบเปิดปิดไฟ ดังแสดงในรูปที่ 5.2

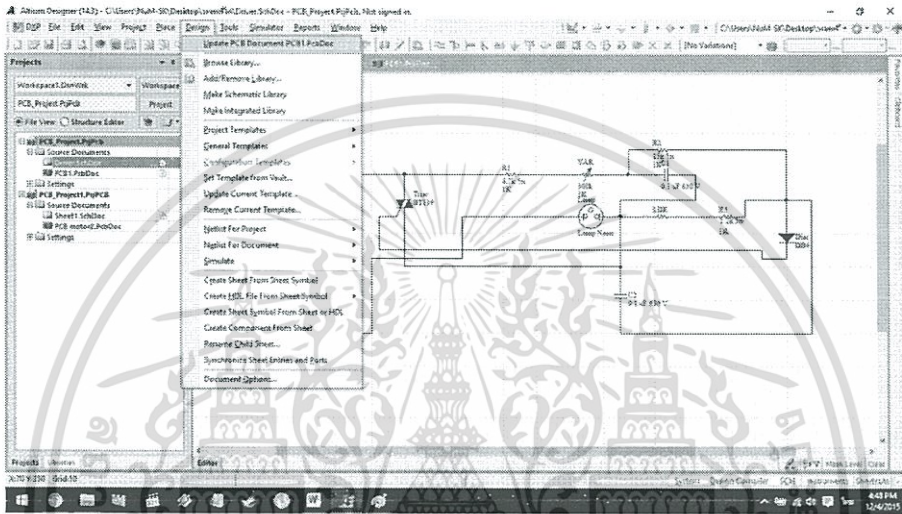


รูปที่ 5.2 Schematic ของวงจรมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

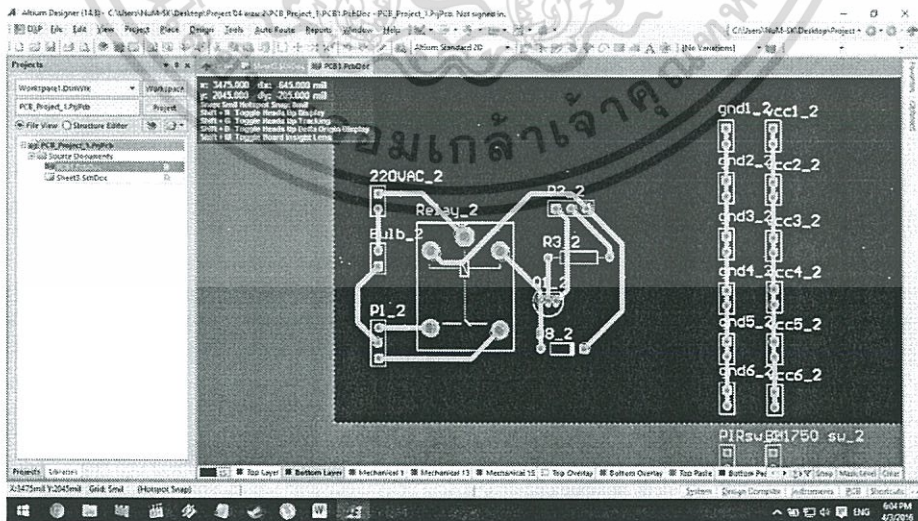
จะใช้ L293D IC เป็น ไดรเวอร์ในการควบคุมการหมุนของมอเตอร์ โดยจะใช้เป็น DC motor แบบทอรอบ ที่ใช้ไฟ 12 V ในวงจรสามารถ ใช้กับมอเตอร์ได้ 2 ตัว ซึ่งในทิศทางการหมุน นั้น ต้องใช้ Arduino เป็นตัวควบคุม

เมื่อทำการเขียนลายวงจร Schematic สำเร็จ ต้องทำการสร้างเป็นลายวงจร PCB โดยกด Update PCB ดังรูปที่ 5.3



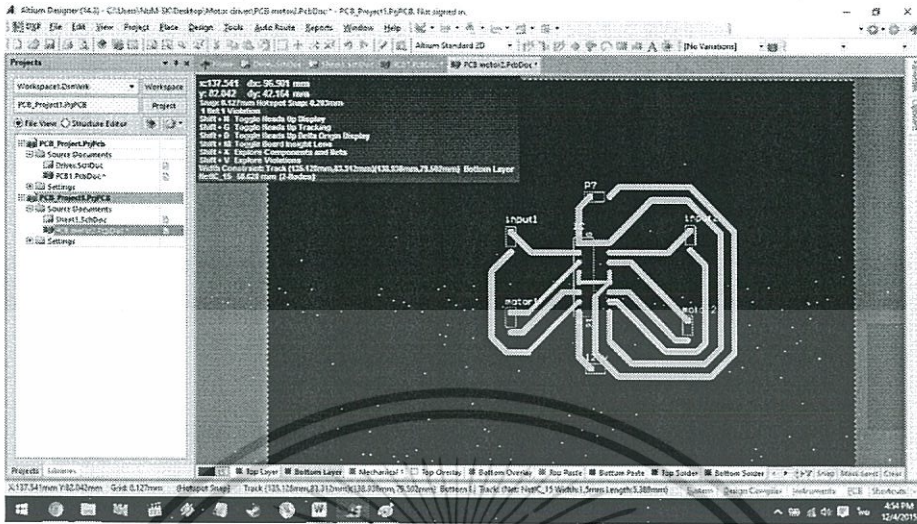
รูปที่ 5.3 การ Update PCB

เมื่อกด Update PCB ลายวงจรจะถูกจัดเรียงอย่างกระจัดกระจาย เราต้องจัดเรียง วงจรใหม่ให้เป็นรูปแบบตามที่ต้องการ ดังรูปที่ 5.4 และ 5.5



รูปที่ 5.4 PCB วงจร Relay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 PCB วงจร มอเตอร์

เมื่อทำการลากลายวงจรใน PCB สำเร็จ ขั้นตอนถัดไปจะเป็นกระบวนการกัดปริ้น
ลงแผ่นทองแดง สร้างเป็นวงจรหรีไฟแล้ววงจรมอเตอร์ เพื่อสะดวกในการใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Code

```

#include <Wire.h>
#include <BH1750.h>
#define trigPin 13
#define echoPin 12
int inputPin = 2;           // choose the input pin (for PIR sensor)
int pirState = LOW;        // we start, assuming no motion detected
int val = 0;               // variable for reading the pin status
BH1750 lightMeter;
const int ledPin = 11;     // the number of the LED pin
int motorPin1 = 7;         // One motor wire connected to digital pin 7
int motorPin2 = 4;         // One motor wire connected to digital pin 4
int incomingByte = 0;
int n = 0;
int value;
int k;
int buzzer = 8;           // setting controls the digital IO foot buzzer
int i;
void setup() {
  Serial.begin(9600);
  pinMode(inputPin, INPUT); // declare sensor as input
  lightMeter.begin();
  pinMode(ledPin, OUTPUT);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.println("start");
  pinMode (buzzer, OUTPUT) ;// set the digital IO pin mode, OUTPUT out

```

of Wen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void loop() {
  uint16_t lux = lightMeter.readLightLevel();
  //value = atoi(lux);
  val = digitalRead(inputPin); // read input value
  long duration, distance;
  digitalWrite(trigPin, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin, HIGH);
  // delayMicroseconds(1000); - Removed this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  if (val == HIGH) { // check if the input is HIGH
    if (pirState == LOW) {
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
    for (i = 0; i <10; i++) // Wen a frequency sound
    {
      digitalWrite (buzzer, HIGH) ;// send voice
      delay (200) ;// Delay 1ms
      digitalWrite (buzzer, LOW) ;// do not send voice
      delay (200) ;// delay ms
    }
    // delay(10000);
  }
  } else {
    if (pirState == HIGH){
      Serial.println("Motion ended!");

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// We only want to print on the output change, not state
pirState = LOW;

}
}

if (distance >= 30 && n == 0){
  Serial.println("curtain open");
  n = 1;
}
else if (distance < 10 && n==1){
  Serial.println("curtain close");
  n = 0;
}
if (lux < 5 && k==0){
  digitalWrite(ledPin, HIGH);
  Serial.println("auto light");
  k=1;
}
else if (lux > 5 && k==1){
  digitalWrite(ledPin, LOW);
  Serial.println("stop light");
  k=0;
}

if (Serial.available() > 0)
{
  incomingByte = Serial.read();
  // set the direction of pin 9:
  if (incomingByte == 'o') {
    Serial.println("OPEN");
    digitalWrite(motorPin2, LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(motorPin1, LOW); //reset motor
delay(1);
digitalWrite(motorPin1, LOW); // set the Pin motorPin1 LOW
digitalWrite(motorPin2, HIGH); //rotates motor

}
else if(incomingByte == 'c') {
Serial.println("CLOSE");
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin1, LOW); //rotates motor
delay(1);
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);
}
else if(incomingByte == 's') {
Serial.println("STOP");
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin1, LOW);
}
else if(incomingByte == 'a') {
Serial.println("open lamp");
digitalWrite(ledPin, HIGH);
}
else if(incomingByte == 'b') {
Serial.println("close lamp");
digitalWrite(ledPin, LOW);
}
else if(incomingByte == 'l') {
Serial.print("Light: ");

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.print(lux);  
Serial.println(" lx");  
delay(1000);  
}  
else if(incomingByte == 'd') {  
Serial.println("distance");  
Serial.print(distance);  
Serial.println(" cm");  
}  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XML

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background1"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/banner"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:background="#ff000000"
        android:src="@drawable/banner" />
    <LinearLayout
        android:id="@+id/getIP"
        android:layout_below="@id/banner"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:layout_marginLeft="10dp"
            android:text="IP"
            android:textColor="#FFFFFF"
            android:textSize="60dp"
        />
        <EditText
            android:id="@+id/ipNetwork"
            android:layout_width="270dp"
            android:layout_height="50dp"

```

```

        android:background="#333333"
        android:layout_gravity="center"
    />
</LinearLayout>

<RelativeLayout
    android:id="@+id/Lightdimmer1"
    android:layout_below="@+id/getIP"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/BrightnessText"
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:text="LIGHT"
        android:textColor="#FFFFFF"
        android:textStyle="bold"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="10dp"
        android:textSize="50dp"
    />
</RelativeLayout>

```

```

<LinearLayout
    android:layout_marginTop="30dp"
    android:id="@+id/ButtonText1"
    android:layout_below="@id/Lightdimmer1"
    android:orientation="horizontal"
    android:layout_width="match_parent"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

android:layout_height="wrap_content">
<Button
    android:id="@+id/openlight"
    android:layout_width="167dp"
    android:layout_height="wrap_content"
    android:background="#339999"
    android:text="OPEN"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:layout_marginLeft="10dp"

/>
<Button
    android:id="@+id/closetlight"
    android:layout_marginLeft="10dp"
    android:layout_width="167dp"
    android:layout_height="wrap_content"
    android:text="CLOSE"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:background="#FF3030"

/>

```

```
</LinearLayout>
```

```
<LinearLayout
```

```

    android:id="@+id/Curtain"
    android:layout_below="@id/ButtonText1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

```

```
<TextView
```

```
    android:layout_width="220dp"
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

android:layout_height="70dp"
android:text="CURTAIN"
android:textColor="#FFFFFF"
android:textStyle="bold"
android:textSize="50dp"
android:layout_marginLeft="10dp"
android:layout_marginTop="20dp" />

```

```
</LinearLayout>
```

```
<LinearLayout
```

```

android:layout_marginTop="30dp"
android:id="@+id/ButtonText2"
android:layout_below="@id/Curtain"
android:orientation="horizontal"
android:layout_width="match_parent"
android:layout_height="wrap_content">

```

```
<Button
```

```

android:id="@+id/opencurtain"
android:layout_width="107dp"
android:layout_height="wrap_content"
android:background="#339999"
android:text="OPEN"
android:textColor="#FFFFFF"
android:textStyle="bold"
android:layout_marginLeft="10dp"
/>

```

```
<Button
```

```

android:id="@+id/stopcurtain"
android:layout_marginLeft="10dp"
android:layout_width="107dp"
android:layout_height="wrap_content"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        android:text="STOP"
        android:textColor="#FFFFFF"
        android:textStyle="bold"
        android:background="#FFCC33"
    />
<Button
    android:id="@+id/closecurtain"
    android:layout_width="107dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:text="CLOSE"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:background="#FF3030"
/>
</LinearLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
</LinearLayout>

```

```
</RelativeLayout>
```

Java Code

```

package com.example.dang.project;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเข้าถึงเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import android.widget.Toast;
import java.io.BufferedWriter;
import java.io.OutputStreamWriter;
import java.net.Socket;
import java.net.UnknownHostException;

;import static android.view.View.*;

public class Showcase extends AppCompatActivity {
    EditText ipNetwork;
    Button openLight, closeLight, openCurtain, stopCurtain, closeCurtain;
    int port = 80;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_showcase);

        openLight = (Button)findViewById(R.id.openlight);
        closeLight = (Button)findViewById(R.id.closelight);
        openCurtain = (Button)findViewById(R.id.opencurtain);
        stopCurtain = (Button)findViewById(R.id.stopcurtain);
        closeCurtain = (Button)findViewById(R.id.closecurtain);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ipNetwork =(EditText)findViewById(R.id.ipNetwork);

openLight.setOnClickListener(new OnClickListener(){
    @Override
    public void onClick(View v){
        setText("Openning...");
        connectArduino();
    }
});

closeLight.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        setText("Closing...");
        connectArduino();
    }
});

openCurtain.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        setText("Openning...");
        connectArduino();
    }
});

closeCurtain.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        setText("Closing...");
        connectArduino();
    }
});

stopCurtain.setOnClickListener(new OnClickListener() {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ขออนุญาต
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@Override
public void onClick(View v) {
    setText("Stop!!!");
    connectArduino();
}
});

}

public void connectArduino() {
    Runnable runSend = new Runnable() {
        public void run() {
            try {
                Socket s = new Socket(ipNetwork.getText().toString()
                    , port);
                BufferedWriter out = new BufferedWriter
                    (new OutputStreamWriter(s.getOutputStream()));
                out.flush();
                Log.i("ip", ipNetwork.getText().toString());
                s.close();
            } catch (UnknownHostException e) {
                e.printStackTrace();
                setText("No device on this IP address.");
            } catch (Exception e) {
                e.printStackTrace();
                setText("Connection failed. Please try again.");
            }
        }
    };
}
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void setText(String str){
    final String status = str;
    Toast toast = Toast.makeText(getApplicationContext(), status,
    Toast.LENGTH_SHORT);
    toast.show();
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_showcase, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้