เครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียง
VOICE-ACTIVATED HOME APPLIANCES

ปณิธาน อิ่มอุสาห์
พงศกร จิรธนาคุณ
สิรภัทร์ อรุณวัต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียง
VOICE-ACTIVATED HOME APPLIANCES

ปณิธาน อิ่มอุส่าห์
พงศกร จิรธนาคุณ
สิรภัทร์ อรุณวัต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VOICE-ACTIVATED HOME APPLIANCES

PANITAN IMUSA
PONGSAKORN JIRATHANAKHUN
SIRAPHAT ARUNWAT

THIS THESIS IS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องใช้ไฟฟ้าสั่งงานด้วยเสียง

VOICE-ACTIVATED HOME APPLIANCES

ผู้จัดทำ นายปณิธาน อิ่มอุส่าห์ 58010719

นายพงศกร จิรธนาคุณ 58010808

นายสิรภัทร์ อรุณวัต 58011311

.....อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร. ชนินทร์ บุญลักษณานุสรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องใช้ไฟฟ้าสั่งงานด้วยเสียง

โดย

ปณิธาน	อิ่มอุส่าห์	58010719
พงศกร	จิรธนาคุณ	58010808
สิรภัทร์	อรุณวัต	58011311

อาจารย์ที่ปรึกษา
รองศาสตราจารย์ ดร.ชนินทร์ บุญลักษณานุสรณ์

ปีการศึกษา 2561

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ศึกษาทฤษฎีและออกแบบเครื่องใช้ไฟฟ้าสั่งงานด้วยเสียง ซึ่งจัดทำเพื่อนำเทคโนโลยีโทรศัพท์มือถือกับ Internet of thing มาต่อยอดกับเครื่องใช้ไฟฟ้าภายในบ้าน เพื่ออำนวยความสะดวกสบายในการดำเนินชีวิตประจำวันของเรา ประหยัดพลังงานไฟฟ้าภายในบ้านได้ และลดค่าใช้จ่ายสำหรับอุปกรณ์ไฟฟ้าที่สั่งงานผ่านอินเทอร์เน็ต ซึ่งในปัจจุบันอุปกรณ์เหล่านี้มีราคาค่อนข้างสูง ดังนั้นงานวิจัยนี้จึงนำเสนอ อุปกรณ์ไฟฟ้าภายในบ้านที่สามารถสั่งงานผ่านโทรศัพท์มือถือและสามารถสั่งงานด้วยเสียงผ่านโทรศัพท์มือถืออีกด้วย ซึ่งอุปกรณ์ที่จัดทำมีดังนี้ คือ ปลั๊กไฟฟ้า สวิสต์ไฟฟ้า กลอนประตูไฟฟ้า ม่านไฟฟ้า และ อุปกรณ์รีโมตที่ส่งสัญญาณผ่านอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VOICE-ACTIVATED HOME APPLIANCES

By

Mr. Panitan	Imusa	58010719
Mr. Pongsakorn	Jirathanakhun	58010808
Mr. Siraphat	Arunwat	58011311

Advisor

Assoc. Prof. Dr. Chanin Bunluksananusorn

Academic Year 2018

ABSTRACT

The purpose of this project is to study and design a system for ordinary appliances to be smarter and could be controlled by voice command. By using smartphone technology and IOT to improve appliances to facilitate humans manners, saving energy, saving money for a smart home product. Nowadays, smart home products are expensive. This thesis presents home appliances that could be controlled by a smartphone or by voice. The products we provided are "Smart Outlet", "Smart Switch", "Smart Door Lock", "Smart Curtain" and "Smart IR Remote"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลืออย่างดียิ่งจากอาจารย์ที่ปรึกษาโครงงาน รองศาสตราจารย์ ดร.ชนินทร์ บุญลักษณานุสรณ์ อาจารย์ภาควิชา วิศวกรรมการวัดและควบคุม รวมถึงเจ้าหน้าที่ คณาจารย์ และคณะผู้บริหาร คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีตลอดเวลาที่ผ่านมา คณะผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณพี่ๆ และเพื่อนๆ ทุกคนที่สนับสนุนอุปกรณ์ และให้คำปรึกษา รวมถึงขอขอบพระคุณครอบครัว ญาติๆ ที่คอยสนับสนุนการทำงาน ให้ความช่วยเหลืออื่นๆ ที่เป็นประโยชน์ต่อโครงงาน และคอยให้กำลังใจเสมอมา

การศึกษาค้นคว้าปริญญานิพนธ์ครั้งนี้สำเร็จลุล่วงด้วยดี และความดีอันเกิดจากการศึกษาค้นคว้าครั้งนี้ คณะผู้จัดทำขอมอบแด่ บิดา มารดา อาจารย์ และผู้มีพระคุณทุกท่าน คณะผู้จัดทำมีความซาบซึ้งในความกรุณาอันดียิ่งจากทุกท่านที่ได้กล่าวนามมา จึงขอขอบพระคุณทุกท่าน และหากผิดพลาดประการใดขออภัยไว้ ณ โอกาสนี้

ผู้จัดทำ
ปณิธาน อิ่มอุส่าห์
พงศกร จิรธนาคุณ
สิรภัทร์ อรุณวัต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์โครงงาน	1
1.3 ขอบเขตของโครงงาน	2
1.4 ขั้นตอนการศึกษาและจัดทำโครงงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	3
2.1 ความสำคัญของโครงงานเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียง	3
2.2 ระบบอินเทอร์เน็ตไร้สาย (WiFi)	3
2.2.1 เทคโนโลยี WiFi	3
2.2.2 ประวัติ WiFi	4
2.2.3 มาตรฐาน WiFi IEEE 802.11	5
2.2.4 มาตรฐานที่อยู่ภายใต้กรอบของเทคโนโลยี IEEE 802.11	5
2.3 Raspberry Pi	9
2.3.1 ระบบปฏิบัติการ Raspbian	9
2.4 Raspberry Pi 3 B+	9
2.5 ภาษา Python	9
2.5.1 คุณลักษณะเด่นของภาษา Python	9
2.6 บอร์ด ESP8266	11
2.6.1 บอร์ด Wemos D1 mini	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.7 รีเลย์ (Relay)	13
2.7.1 ประเภทของรีเลย์	13
2.7.2 ชนิดของรีเลย์	14
2.8 มอเตอร์ไฟฟ้า (Motor)	15
2.8.1 ชนิดของมอเตอร์ไฟฟ้า	16
2.9 อินฟราเรด (Infrared IR)	18
2.9.1 ลักษณะของอินฟราเรด	18
บทที่ 3 วิธีการดำเนินงาน	19
3.1 การเลือกอุปกรณ์	19
3.1.1 ไมโครคอนโทรลเลอร์ (Microcontroller)	19
3.1.2 รีเลย์ (Relay)	20
3.1.3 มอเตอร์ (Motor)	21
3.2 การตั้งค่า Raspberry Pi	22
3.3 ลงโปรแกรม HomeBridge ใน Raspberry Pi	24
3.4 การเชื่อมต่ออุปกรณ์	24
3.5 การลง HomeBridge Plugin	27
3.6 การติดตั้งอุปกรณ์	28
3.6.1 การติดตั้งระบบม่านไฟฟ้าสั่งงานด้วยเสียง	28
3.6.2 การติดตั้งปลั๊กไฟฟ้าสั่งงานด้วยเสียง	29
3.6.3 การติดตั้งสวิตช์ไฟฟ้าสั่งงานด้วยเสียง	30
3.6.4 การติดตั้งกลอนประตูไฟฟ้าสั่งงานด้วยเสียง	31
3.6.5 การติดตั้งอุปกรณ์ตัวส่งสัญญาณอินฟราเรดด้วยเสียง	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 4 ผลการวิเคราะห์ข้อมูล	33
4.1 ผลจากการทดสอบการทำงาน	33
4.1.1 การเชื่อมต่อไร้สาย	33
4.1.2 การทำงานของม่านไฟฟ้าสั่งงานด้วยเสียง	34
4.1.3 การทำงานของปลั๊กไฟฟ้าสั่งงานด้วยเสียง	35
4.1.4 การทำงานของสวิตช์ไฟฟ้าสั่งงานด้วยเสียง	35
4.1.5 การทำงานกลอนประตูไฟฟ้าสั่งงานด้วยเสียง	35
4.1.6 การทำงานของตัวส่งสัญญาณอินฟราเรดด้วยเสียง	35
4.2 การนำไปใช้งาน	36
บทที่ 5 สรุปผลและข้อเสนอแนะ	37
5.1 สรุปผลการทดลอง	37
5.2 ปัญหาและอุปสรรค	37
5.3 แนวทางแก้ไขปัญหา	37
5.4 แนวทางการพัฒนา	38
เอกสารอ้างอิง	39
ภาคผนวก	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การเชื่อมต่อ WiFi	7
2.2 Raspberry Pi 3 B+	8
2.3 WeMos D1 mini	12
2.4 รีเลย์ (Relay)	15
2.5 มอเตอร์ไฟฟ้า	17
2.6 หลอดอินฟราเรด	18
3.1 Raspberry Pi B+	19
3.2 บอร์ด WeMos D1 Mini	20
3.3 Relay Shield for WeMos D1 Mini	20
3.4 28BYJ-48 Stepper Motor และบอร์ดไดร์ฟเวอร์ ULN2003	21
3.5 หน้า Download Raspbian	22
3.6 โปรแกรม Win32 Disk Imager	22
3.7 หน้าต่างตั้งค่า	23
3.8 พอร์ตเชื่อมต่อของ Raspberry Pi 3 B+	25
3.9 หมายเลข GPIO ของ Pin ต่างๆ	25
3.10 รูปแบบการเชื่อมต่อระหว่างอุปกรณ์ต่างๆ	26
3.11 วงจรสั่งการม่านไฟฟ้าสั่งงานด้วยเสียง	28
3.12 การติดตั้งม่าน	28
3.13 วงจรการทำงานของปลั๊กไฟฟ้า	29
3.14 ชุดปลั๊กไฟฟ้า	29
3.15 วงจรสั่งงานสวิตช์ไฟฟ้าสั่งงานด้วยเสียง	30
3.16 ชุดอุปกรณ์สวิตช์ไฟฟ้าสั่งงานด้วยเสียง	30
3.17 วงจรไฟฟ้ากลอนประตูสั่งงานด้วยเสียง	31
3.18 กลอนประตูไฟฟ้าสั่งงานด้วยเสียง	31
3.19 วงจรตัวส่งสัญญาณอินฟราเรดด้วยเสียง	32
3.20 อุปกรณ์ส่งสัญญาณอินฟราเรดด้วยเสียง	32
4.1 แอพพลิเคชัน Home ในระบบปฏิบัติการ iOS	33
4.2 การสั่งการด้วยเสียงผ่าน Siri ในระบบ iOS	33
4.3 ปุ่มควบคุมม่านไฟฟ้าสั่งงานด้วยเสียง	34
4.4 การปรับระดับม่านผ่านแอพพลิเคชัน	34
4.5 อุปกรณ์ไฟฟ้าสั่งงานด้วยเสียง	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในบทนี้จะกล่างถึงความเป็นมาและความสำคัญของปริญญานิพนธ์ วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน ประโยชน์ที่คาดว่าจะได้รับจากโครงงานนี้ โดยมีรายละเอียดดังต่อไปนี้

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

เนื่องจากในปัจจุบันโทรศัพท์มือถือมีการพัฒนาให้สามารถทำหน้าที่ได้หลากหลายอย่าง แล้วยังมีการพัฒนาแอพพลิเคชันให้กับโทรศัพท์มือถือ รวมถึงระบบอินเทอร์เน็ตในปัจจุบัน ทำให้โทรศัพท์มือถือไม่ได้ใช้เฉพาะในการสื่อสารเท่านั้น แต่ยังสามารถนำมาใช้ประโยชน์ได้อีกมากมาย อาทิเช่น การทำธุรกรรมทางการเงินผ่านแอพพลิเคชันในโทรศัพท์มือถือ เป็นต้น รวมถึงโทรศัพท์มือถือในปัจจุบันสามารถรับคำสั่งด้วยเสียงได้อีกด้วย ทำให้การดำเนินชีวิตในปัจจุบันมีความสะดวกสบายมากยิ่งขึ้น

จากข้างต้นทำให้มีแนวคิดที่จะประยุกต์ใช้ประโยชน์จากโทรศัพท์มือถือ อินเทอร์เน็ต และการรับคำสั่งเสียง ในการควบคุมเครื่องใช้ไฟฟ้าภายในบ้านที่สามารถสั่งงานด้วยเสียงผ่านโทรศัพท์มือถือจากระยะไกลผ่านอินเทอร์เน็ต ซึ่งเป็นอีกเทคโนโลยีที่จะอำนวยความสะดวกในการดำเนินชีวิต ทำให้ประหยัดเวลา และประหยัดพลังงานไฟฟ้าได้ แล้วในปัจจุบันยังมีการพัฒนาอุปกรณ์ไฟฟ้าที่สามารถสั่งการผ่านโทรศัพท์มือถืออย่างต่อเนื่อง และมีคนสนใจในเทคโนโลยีนี้อย่างมาก แต่ในปัจจุบันเทคโนโลยีและอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านที่สามารถทำงานได้เช่นนี้มีราคาที่ค่อนข้างสูง และยังไม่มีการรวมเครื่องใช้ไฟฟ้าต่างๆภายในบ้านในแอพพลิเคชันเดียว เช่น ถ้าจะเปิดหลอดไฟ ต้องใช้แอพพลิเคชันหนึ่ง แล้วถ้าอยากตัดไฟฟ้าออกจากปลั๊กต้องใช้อีกแอพพลิเคชันหนึ่ง และไม่สามารถสั่งงานด้วยเสียง เป็นต้น

1.2 วัตถุประสงค์โครงงาน

1. เพื่อเพิ่มความสะดวกสบาย และประหยัดเวลาในการดำเนินชีวิตประจำวัน
2. เพื่อทำให้ประหยัดพลังงานไฟฟ้า
2. เพื่อลดค่าใช้จ่ายค่าอุปกรณ์ไฟฟ้าที่สั่งงานผ่านโทรศัพท์มือถือ
3. เพื่อเพิ่มประสิทธิภาพโทรศัพท์มือถือให้ได้ประโยชน์มากขึ้น
4. เพื่อประยุกต์ใช้เทคโนโลยีในปัจจุบันให้เป็นประโยชน์
5. เพื่อศึกษาและพัฒนาการสั่งงานด้วยเสียงผ่านโทรศัพท์มือถือในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1. สามารถใช้คำสั่งเสียงสั่งงานเครื่องใช้ไฟฟ้าภายในบ้าน อย่างเช่น เปิด-ปิดหลอดไฟ เปิด-ปิดเต้ารับไฟ การเปิด-ปิดม่าน กลอนประตูไฟฟ้า และเปิด-ปิด เครื่องเสียง

2. ใช้โปรแกรม Raspbian, Arduino IDE, Putty และ MQTT Lens ในการเขียนโปรแกรมสั่ง งานเครื่องใช้ไฟฟ้าข้างต้น

3. ใช้ Raspberry Pi3 Model B+, Step Motor, WEMOS D1 mini ESP8266 และ Relay ช่วยในการทำงานของเครื่องใช้ไฟฟ้าภายในบ้านข้างต้น

4. สามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านภายในหนึ่งแอพพลิเคชัน

1.4 ขั้นตอนการศึกษาและจัดทำโครงงาน

1. ศึกษาการทำงานของโปรแกรมและอุปกรณ์ต่างๆ ในการทำโครงงาน
2. เขียนวงจรการทำงานของเครื่องใช้ไฟฟ้า
3. เขียนโปรแกรมการสั่งการเครื่องใช้ไฟฟ้า
4. ทดลองต่อวงจรอุปกรณ์เครื่องใช้ไฟฟ้า
5. ลงโปรแกรมใน WEMOS D1 mini ESP8266 แล้วสั่งงาน
6. แก้ไขและปรับปรุง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มความสะดวกสบายและประหยัดเวลาในการดำเนินชีวิตประจำวัน
2. สามารถลดการใช้พลังงานไฟฟ้า
3. สามารถลดค่าใช้จ่ายอุปกรณ์ไฟฟ้าที่สามารถสั่งงานผ่านโทรศัพท์มือถือ
4. สามารถใช้งานได้จริงและมีประสิทธิภาพ
5. สามารถประยุกต์ใช้เทคโนโลยีในปัจจุบันให้มีประโยชน์มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ความสำคัญของโครงงานเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียง

โครงงานเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียงถูกทำขึ้น เพื่อแก้ไขปัญหาเครื่องใช้ไฟฟ้าที่สามารถสั่งงานผ่านระบบอินเทอร์เน็ต ซึ่งได้เล็งเห็นปัญหา 3 ด้านด้วยกันคือ ในปัจจุบันสินค้าไม่สามารถควบคุมอุปกรณ์ไฟฟ้าภายในแอพพลิเคชันเดียว หรือบางผลิตภัณฑ์สามารถเปิด-ปิดไฟฟ้าได้แค่ผ่านทางแอพลิเคชันเท่านั้น ซึ่งทำให้ยุ่งยากในการใช้งานจริง อุปกรณ์ไฟฟ้าในปัจจุบันไม่สามารถรองรับระบบปฏิบัติการได้ทุกระบบ และที่สำคัญผลิตภัณฑ์จำพวกนี้มีราคาค่อนข้างสูงเนื่องจากยังไม่แพร่หลาย ซึ่งในประเทศไทยในค่อนข้างที่จะหาได้ยาก ดังนั้นโครงงานเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียงนี้สร้างขึ้นเพื่อแก้ไขปัญหาข้างต้นเหล่านี้เพื่อความสะดวกสบายในการใช้งานและลดต้นทุนสำหรับผู้สนใจ

2.2 ระบบอินเทอร์เน็ตไร้สาย (WiFi)

2.2.1 เทคโนโลยี WiFi

เป็นเทคโนโลยีที่ได้รับความนิยมที่ช่วยให้อุปกรณ์อิเล็กทรอนิกส์ในการแลกเปลี่ยนข้อมูล หรือการเชื่อมต่ออินเทอร์เน็ตแบบไร้สายโดยใช้คลื่นวิทยุ คำๆ นี้เป็นเครื่องหมายการค้าของ WiFi Alliance ที่ได้ให้คำนิยามของไวไฟว่าหมายถึง "ชุดผลิตภัณฑ์ใดๆ ที่สามารถทำงานได้ตามมาตรฐานเครือข่ายคอมพิวเตอร์แบบไร้สาย (แลนไร้สาย) ซึ่งอยู่บนมาตรฐาน IEEE 802.11" อย่างไรก็ตามเนื่องจากแลนไร้สายที่ทันสมัยส่วนใหญ่จะขึ้นอยู่กับมาตรฐานเหล่านี้ คำว่า "ไวไฟ" จึงนำมาใช้ในภาษาอังกฤษทั่วไปโดยเป็นคำพ้องสำหรับ "แลนไร้สาย" เดิมทีไวไฟออกแบบมาใช้สำหรับอุปกรณ์พกพาต่างๆ และใช้เครือข่าย LAN เท่านั้น แต่ปัจจุบันนิยมใช้ไวไฟเพื่อต่อกับอินเทอร์เน็ต โดยอุปกรณ์พกพาต่างๆ เช่น คอมพิวเตอร์ส่วนบุคคล เครื่องเล่นเกมส์ โทรศัพท์สมาร์ทโฟน แท็บเล็ต กล้องดิจิทัลและเครื่องเสียงดิจิทัล สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ผ่านอุปกรณ์ที่เรียกว่า แอคเซสพอยต์ หรือฮอตสปอต และบริเวณที่ระยะทำการของแอคเซสพอยต์ครอบคลุมอยู่ที่ประมาณ 20 เมตรในอาคาร แต่ระยะนี้จะไกลกว่าถ้าเป็นที่โล่งแจ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ประวัติ WiFi

ไวไฟ หรือเทคโนโลยีเครือข่ายแบบไร้สาย มาตรฐาน IEEE 802.11 ถือกำเนิดขึ้นในปี ค.ศ. 1997 จัดตั้งโดยองค์การไอทริปเปิ้ลอี (สถาบันวิศวกรรมทางด้านไฟฟ้าและอิเล็กโทรนิคส์) มีความเร็ว 1 Mbps ในยุคเริ่มแรกนั้นให้ประสิทธิภาพการทำงานที่ค่อนข้างต่ำ ทั้งไม่มีการรับรองคุณภาพของการให้บริการที่เรียกว่า QoS (Quality of Service) และมาตรฐานความปลอดภัยต่ำ จากนั้นทาง IEEE จึงจัดตั้งคณะทำงานขึ้นมาปรับปรุงหลายกลุ่มด้วยกัน โดยที่กลุ่มที่มีผลงานเป็นที่น่าพอใจและได้รับการยอมรับอย่างเป็นทางการว่า ได้มาตรฐาน ได้แก่ กลุ่ม 802.11a, 802.11b และ 802.11g

เทคโนโลยี 802.11 มีต้นกำเนิดในปี ค.ศ. 1985 กำหนดขึ้นโดยคณะกรรมการการสื่อสารแห่งชาติสหรัฐอเมริกา (U.S. Federal Communications Commission) หรือ FCC ที่ประกาศช่วงความถี่สำหรับกิจการด้านอุตสาหกรรม วิทยาศาสตร์และการแพทย์ (ISM) สำหรับการใช้งานที่ไม่ต้องมีใบอนุญาต

ในปี ค.ศ. 1991 บริษัท เอ็นซีอาร์/เอทีแอนด์ที (ตอนนี้เป็น Alcatel-Lucent และ LSI คอร์ปอเรชั่น) ได้สร้างชุดตั้งต้นของ 802.11 ในเมือง Nieuwegein, เนเธอร์แลนด์ ตอนแรกนักประดิษฐ์ตั้งใจจะใช้เทคโนโลยีนี้สำหรับระบบเก็บเงิน ผลิตภัณฑ์ไร้สายตัวแรกที่นำออกสู่ตลาดอยู่ภายใต้ชื่อ WaveLAN ที่มีอัตราข้อมูลดิบของ 1 Mbit/s และ 2 Mbit/s

วิก เฮส์ผู้เป็นประธานของ IEEE 802.11 เป็นเวลา 10 ปี และเรียกว่า "บิดาแห่ง WiFi" ได้มีส่วนร่วมในการออกแบบ 802.11b และ 802.11a มาตรฐานเริ่มต้นภายใน IEEE.

นักวิทยุ-ดาราศาสตร์ชาวออสเตรเลียชื่อ จอห์น โอ ซัลลิแวนได้พัฒนาสิทธิบัตรที่สำคัญที่ใช้ใน WiFi ที่เป็นผลพลอยได้ในโครงการวิจัย CSIRO "การทดลองที่ล้มเหลวในการตรวจสอบหาการระเบิดหลุมดำขนาดเล็กที่มีขนาดเท่าหนึ่งอนุภาคอะตอม" ในปี ค.ศ. 1992 และปี ค.ศ. 1996 องค์กร ของออสเตรเลียชื่อ CSIRO (the Australian Commonwealth Scientific and Industrial Research Organisation) ได้รับสิทธิบัตร สำหรับวิธีการที่ในภายหลังใช้ใน WiFi ในการ "กำจั ดรอยเปื้อน"ของสัญญาณ

ในปี ค.ศ. 1999 WiFi Alliance จัดตั้งขึ้นเป็นสมาคมการค้าเจ้าของเครื่องหมายการค้า WiFi ซึ่งผลิตภัณฑ์ส่วนใหญ่ที่ใช้ WiFi จะมีเครื่องหมายนี้

ในเดือนเมษายนปีค.ศ. 2009 14 บริษัทเทคโนโลยีตกลงที่จะจ่าย 250 ล้านดอลลาร์สหรัฐให้กับ CSIRO สำหรับการละเมิดสิทธิบัตรของ CSIRO สิ่งนี้ทำให้ WiFi กลายเป็นสิ่งประดิษฐ์ ของออสเตรเลีย แม้ว่าจะเป็นเรื่องของการโต้เถียงกันอยู่ ในปี ค.ศ. 2012 CSIRO ยังชนะคดีและจะได้รับเงินชดเชยเพิ่มเติม 220 ล้านดอลลาร์ สำหรับการละเมิดสิทธิบัตร WiFi กับบริษัทระดับโลกในประเทศสหรัฐอเมริกา ซึ่งจะต้องจ่ายค่าลิขสิทธิ์แก่ CSIRO ที่คาดว่าจะมีมูลค่าเพิ่มอีก 1 พันล้านดอลลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 มาตรฐาน WiFi IEEE 802.11

IEEE 802.11 คือ มาตรฐานการทำงานของระบบเครือข่ายไร้สายกำหนดขึ้น โดยสถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers : IEEE) เป็นมาตรฐานกลางที่ได้นำมาปฏิบัติใช้ เพื่อที่จะทำการเชื่อมโยงอุปกรณ์เครือข่ายไร้สายเข้าด้วยกันบนระบบ ในทางปกติแล้วการเชื่อมต่อระบบเครือข่ายไร้สาย จำเป็นต้องใช้อุปกรณ์สองชิ้น นั่นคือ

1. แอคเซสพอยต์ คือ ตัวกลางที่ช่วยในการติดต่อระหว่างตัวรับ-ส่งสัญญาณไวเลสของผู้ใช้กับเราต์เตอร์ผ่านทางสายนำสัญญาณ ที่ทำจากทองแดงที่ได้รับการเชื่อมต่อกับระบบเครือข่าย เช่น สายแลนหรือสายโทรศัพท์ ADSL หรือผ่านทางสายใยแก้วนำแสง

2. ตัวรับ-ส่งสัญญาณไวเลส ทำหน้าที่รับ-ส่งสัญญาณ ระหว่างตัวรับส่งแต่ละตัวด้วยกันหรือระหว่างตัวลูกข่ายกับแอคเซสพอยต์

หลังจากที่เทคโนโลยีเครือข่ายไร้สายนี้ได้เกิดขึ้น ก็ได้เกิดมาตรฐานตามมาอีกมายมาย โดยที่การจะเลือกซื้อหรือเลือกใช้อุปกรณ์เครือข่ายไร้สายเหล่านั้น จำเป็นจะต้องคำนึงถึงเทคโนโลยีที่ใช้ในผลิตภัณฑ์นั้นๆ รวมถึงความเข้ากันได้ของเทคโนโลยีที่ต่างๆ

2.2.4 มาตรฐานที่อยู่ภายใต้ กรอบของเทคโนโลยี IEEE 802.11

ปัจจุบันนี้มีมาตรฐานออกมาหลายอย่าง แต่ที่ได้รับความนิยมทั้งในอดีตและปัจจุบันนั้น แบ่งออกเป็น 7 มาตรฐานด้วยกัน ได้แก่

1. IEEE 802.11a - เสร็จสมบูรณ์เมื่อปี ค.ศ. 1999 โดยออกเผยแพร่ช้ากว่าของมาตรฐาน IEEE 802.11b ใช้เทคโนโลยีที่เรียกว่า OFDM (Orthogonal Frequency Division Multiplexing) เพื่อปรับปรุงความเร็วในการส่งข้อมูลให้วิ่งได้สูงถึง 54 Mbps บนความถี่ 5 Ghz ซึ่งจะมีคลื่นรบกวนน้อยกว่าความถี่ 2.4 Ghz ที่มาตรฐานอื่นใช้กัน ที่ความเร็วนี้สามารถทำการแพร่ภาพและข่าวสาร ที่ต้องการความละเอียดสูงได้ อัตราความเร็วในการรับส่งข้อมูลสามารถปรับระดับให้ช้าลงได้ เพื่อเพิ่มระยะทางการเชื่อมต่อให้มากขึ้น แต่ทว่าข้อเสียก็คือ ความถี่ 5 Ghz นั้น หลายๆ ประเทศไม่อนุญาตให้ใช้ เช่น ประเทศไทย เพราะได้จัดสรรให้อุปกรณ์ประเภทอื่นไปแล้ว และเนื่องด้วยการที่มาตรฐานนี้ใช้การเชื่อมต่อที่ความถี่สูงๆ ทำให้มาตรฐานนี้ มีระยะการรับส่งที่ค่อนข้างใกล้คือ ประมาณ 35 เมตรในโครงสร้างปิด (เช่น ในตึก ในอาคาร) และ 120 เมตรในที่โล่งแจ้ง และด้วยความที่ส่งข้อมูลด้วยความถี่สูงนี้ ทำให้การส่งข้อมูลนั้นไม่สามารถทะลุทะลวงโครงสร้างของตึกได้มากนัก อุปกรณ์ไร้สายที่รองรับเทคโนโลยี IEEE 802.11a นี้ไม่สามารถเข้ากันได้กับอุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11b และ IEEE 802.11g ที่จะอธิบายด้านล่างนี้ได้ อีกทั้งอุปกรณ์ของ IEEE 802.11a ยังมีราคาสูงกว่า IEEE 802.11b ด้วย ดังนั้นอุปกรณ์ IEEE 802.11a จึงได้รับความนิยมน้อยกว่า IEEE 802.11b มาก จึงทำให้ไม่ค่อยเป็นที่ได้รับความนิยมเท่าที่ควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. IEEE 802.11b - มาตรฐานนี้เสร็จสมบูรณ์เมื่อปี ค.ศ. 1999 ใช้เทคโนโลยีที่เรียกว่า CCK (Complimentary Code Keying) ผนวกกับ DSSS (Direct Sequence Spread Spectrum) เพื่อปรับปรุงความสามารถของอุปกรณ์ให้รับส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 11 Mbps ผ่านคลื่นวิทยุความถี่ 2.4 GHz เนื่องจากการใช้คลื่นความถี่ที่ต่ำกว่าอุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11a ทำให้อุปกรณ์ที่ใช้มาตรฐานนี้จะมีความสามารถในการส่งคลื่นสัญญาณไปได้ไกลกว่า คือประมาณ 38 เมตรในโครงสร้างปิด และ 140 เมตรในที่โล่งแจ้ง รวมถึงสัญญาณสามารถทะลุทะลวงโครงสร้างตึกได้มากกว่าอุปกรณ์ที่รองรับกับมาตรฐาน IEEE 802.11a ด้วย ปัจจุบันผลิตภัณฑ์อุปกรณ์เครือข่ายไร้สายภายใต้มาตรฐานนี้ได้รับการผลิตออกมาเป็นจำนวนมาก โดยอุปกรณ์ที่ใช้ความถี่ย่านนี้ เช่น IEEE 802.11, Bluetooth, โทรศัพท์ไร้สาย และเตาไมโครเวฟ และที่สำคัญแต่ละผลิตภัณฑ์มีความสามารถทำงานร่วมกันได้ อุปกรณ์ของผู้ผลิตทุกยี่ห้อต้องผ่านการตรวจสอบจากสถาบัน WiFi Alliance เพื่อตรวจสอบมาตรฐานของอุปกรณ์และความเข้ากันได้ของแต่ละผู้ผลิต ปัจจุบันนี้นิยมนำอุปกรณ์ WLAN ที่มาตรฐาน 802.11b ไปใช้ในองค์กรธุรกิจ สถาบันการศึกษา สถานที่สาธารณะ และกำลังแพร่เข้าสู่สถานที่พักอาศัยมากขึ้น มาตรฐานนี้มีระบบเข้ารหัสข้อมูลแบบ WEP ที่ 128 บิต

3. IEEE 802.11g - เสร็จสมบูรณ์ในปี ค.ศ. 2003 ทางคณะทำงาน IEEE 802.11g ได้นำเอาเทคโนโลยี OFDM ของ 802.11a มาพัฒนาบนความถี่ 2.4 Ghz จึงทำให้ใช้ความเร็ว 36-54 Mbps ซึ่งเป็นความเร็วที่สูงกว่ามาตรฐาน 802.11b ซึ่ง 802.11g สามารถปรับระดับความเร็วในการสื่อสารลงเหลือ 2 Mbps ได้ตามสภาพแวดล้อมของเครือข่ายที่ใช้งาน มาตรฐานนี้เป็นที่ยอมรับจากผู้ใช้เป็นจำนวนมากและกำลังจะเข้ามาแทนที่ 802.11b ในอนาคตอันใกล้

นอกจากที่กล่าวมาข้างต้นนี้มีบางผลิตภัณฑ์ใช้เทคโนโลยีเฉพาะตัวเข้ามาเสริม ทำให้ความเร็วเพิ่มขึ้นจาก 54 Mbps เป็น 108 Mbps แต่ต้องทำงานร่วมกันเฉพาะอุปกรณ์ที่ผลิตจากบริษัทเดียวกันเท่านั้น ซึ่งความสามารถนี้เกิดจากชิป (Chip) กระจายสัญญาณของตัวอุปกรณ์ที่ผู้ผลิตบางรายสามารถเพิ่มประสิทธิภาพการรับส่ง สัญญาณเป็น 2 เท่าของการรับส่งสัญญาณได้แต่ปัญหาของการกระจายสัญญาณนี้จะมีผลทำให้อุปกรณ์ ไร้สายในมาตรฐาน 802.11b มีประสิทธิภาพลดลงด้วยเช่นกัน

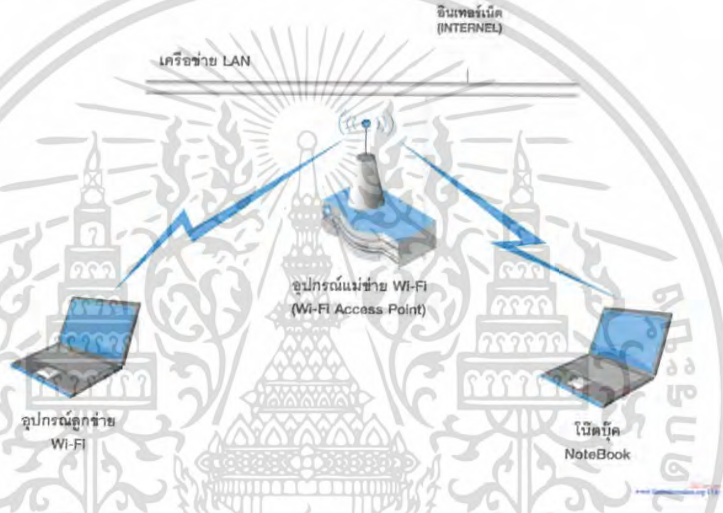
4. IEEE 802.11n - เสร็จสมบูรณ์ในปี ค.ศ. 2009 ทำงานบนย่านความถี่ 2.4 และ 5 GHz โดยที่สามารถให้อัตราการส่งถ่ายข้อมูลสูงสุดถึง 300 Mbps มีความสามารถในการส่งคลื่นสัญญาณได้ระยะประมาณ 70 เมตรในโครงสร้างปิด และ 250 เมตรในที่โล่งแจ้ง เพิ่มความสามารถในการกันสัญญาณกวนจากอุปกรณ์อื่นๆ ที่ใช้ความถี่ 2.4 GHz เหมือนกัน และสามารถรองรับอุปกรณ์มาตรฐาน IEEE 802.11b และ IEEE 802.11g ได้

5. 802.11-2012 - ในปี 2007 กลุ่มงาน TGmb ได้รับการอนุมัติให้รวบรวมการแก้ไขทั้งหมดให้เป็นเวอร์ชันที่เรียกว่า REVmb หรือ 802.11mb ที่ประกอบด้วย 802.11k, r, y, n, w, p, z, v, u, s ตีพิมพ์วันที่ 29 มีนาคม 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. 802.11ac - เป็นมาตรฐานที่ 5 GHz ให้ทรูพุทกับแลนไร้สายแบบหลายสถานีสูงกว่าที่อย่างน้อย 1 Gbps และสำหรับลิงก์เดี่ยวที่อย่างน้อย 500 Mbps โดยการใช้ RF แบนด์วิธที่กว้างกว่า (80 หรือ 160 MHz) สตรีมมากกว่า (สูงถึง 8 สตรีม) และมอดูเลทที่ความจุสูงกว่า (สูงถึง 256 QAM)

7. 802.11ad - หรือ "WiGig" เกิดจากการผลักดันจากผู้ผลิตฮาร์ดแวร์ ในวันที่ 24 กรกฎาคม 2012 Marvell และ Wilocity ได้ประกาศการเป็นคู่ค้าใหม่เพื่อนำ WiFi Solution แบบ Tri-band ใหม่ออกสู่ตลาด โดยการใช้ความถี่ที่ 60 GHz ทรูพุททางทฤษฎีสูงสุดถึง 7 Gbps มาตรฐานนี้จะออกสู่ตลาดได้ราวต้นปี 2014



รูปที่ 2.1 การเชื่อมต่อ WiFi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Raspberry Pi

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กที่จัดอยู่ในกลุ่มคอมพิวเตอร์แบบฝังตัว (Embedded Computer) ถูกพัฒนาขึ้นในสหราชอาณาจักร (UK) โดย Raspberry Pi Foundation รูปแบบเดิมออกแบบเพื่อเป้าหมายในการส่งเสริมการเรียนการสอนวิทยาศาสตร์คอมพิวเตอร์ขั้นพื้นฐาน ในโรงเรียนและในประเทศกำลังพัฒนา ในปัจจุบันเป็นที่นิยมมากขึ้นกว่าเป้าหมายเดิมที่คาดการณ์ไว้ โดยขายนอกตลาดเป้าหมายสำหรับการใช้งาน เช่น หุ่นยนต์, Server, Internet of Thing ฯลฯ

Raspberry Pi ได้รับการเผยแพร่หลายรุ่นแล้ว ทุกรุ่นมีชิพ Broadcom เป็นหน่วยประมวลผลกลาง (CPU) และหน่วยประมวลผลกราฟิก (GPU) ในตัว มีความเร็วตั้งแต่ 700MHz ถึง 1.4 GHz (1.4 GHz สำหรับ Raspberry Pi 3 รุ่น B+ โดยมี RAM ตั้งแต่ 256MB ถึง 1GB มีการ์ด Secure Digital (SD) ใช้เพื่อจัดเก็บระบบปฏิบัติการและหน่วยความจำ ในขนาด SDHC หรือ MicroSDHC บอร์ดมีพอร์ต USB ตั้งแต่ 1 ถึง 4 พอร์ต มีเอาต์พุตวิดีโอ HDMI และวิดีโอคอมโพสิต โดยมีแจ็คขนาด 3.5 มิลลิเมตร สำหรับเอาต์พุตเสียง และ GPIO จำนวนหนึ่งซึ่งสนับสนุนโปรโตคอลทั่วไป เช่น i2C อีกทั้งในรุ่น B มีพอร์ต Ethernet และ Raspberry Pi Zero W มี WiFi 802.11n และ Bluetooth ที่ติดตั้งอยู่ในตัว

Raspberry Pi 3 Model B ได้รับการปล่อยตัวในเดือนกุมภาพันธ์ปี 2017 ด้วยโปรเซสเซอร์ Quad-Core 64 บิทและมี WiFi On-board, Bluetooth และ USB Boot ได้ และล่าสุดเมื่อ 2018 ได้เปิดตัวรุ่น 3 B+ พร้อมกับโปรเซสเซอร์ความเร็ว 1.4 GHz ใช้เครือข่ายความเร็วสูงขึ้น 3 เท่า มี Gigabit (300 Mbit/s) หรือ 2.4/5 GHz dual-band WiFi (100 Mbit/s) ตัวเลือกอื่นๆ ได้แก่ Power over Ethernet (PoE) บูต USB และการบูตระบบเครือข่าย ดังรูปที่ 2.2

รูปที่ 2.2 Raspberry Pi 3 B+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 ระบบปฏิบัติการ Raspbian

Raspbian เป็นหนึ่งในระบบปฏิบัติการที่ไม่เสียค่าใช้จ่าย ถูกปรับแต่งมาให้ใช้กับอุปกรณ์ Raspberry Pi ซึ่งได้ถูกตั้งค่าโปรแกรมพื้นฐาน และโปรแกรมอรรถประโยชน์ที่ใช้ในการทำงานของ Raspberry Pi เปิดตัวอย่างเป็นทางการในเดือน กรกฎาคม 2012 ต่อมาระบบปฏิบัติการ Raspbian มีสมาคม Raspberry Pi เป็นผู้ดูแล ซึ่งระบบปฏิบัติการนี้มีพื้นฐานมาจาก ARM Hard-float (Armfh) กับ LXDE Dest Enviroment แต่ถูกปรับแต่งสำหรับ ARMv6 ตั้งค่าสำหรับ Raspberry Pi มาโดยเฉพาะ

2.4 Raspberry Pi 3 B+

Raspberry Pi 3 Model B+ เป็นรุ่นต่อยอดจาก Pi 3 Model B ก่อนหน้า โดยเพิ่มสเปกหลักๆ อย่างซีพียูและอินเทอร์เน็ต สำหรับตัว Raspberry Pi 3 Model B+ หน่วยประมวลผล Broadcom BCM2837B0 แบบ Quad-Core ความเร็ว 1.4 GHz จากเดิม 1.2 GHz เพิ่มการเชื่อมต่อไร้สายแบบ Dual-Band (2.4 GHz กับ 5 GHz) เพิ่มชิป Bluetooth 4.2 จากเดิมมีเฉพาะ Bluetooth Low Energy รองรับ Gigabit Ethernet ผ่าน USB 2.0 โดยเพิ่มความเร็วเป็น 300 Mbps มากกว่าเดิม 3 เท่า และสุดท้ายมาพร้อม Power Over Ethernet (PoE) ในตัว ช่วยให้จ่ายไฟผ่านพอร์ต LAN ได้เลย Raspberry Pi 3 Model B+ อัพเกรดความเร็วซีพียูจาก 1.2 GHz เป็น 1.4 GHz เพิ่มการเชื่อมต่อไร้สายแบบ Dual-Band (2.4 GHz กับ 5 GHz) รองรับ Gigabit Ethernet ผ่าน USB 2.0 แล้ว และมาพร้อม Power Over Ethernet (PoE) ในตัว

2.5 ภาษา Phython

Python คือ ชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือ สามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัว นี้เป็น OpenSource เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมได้ฟรีโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครบคุมกับทุกลักษณะงาน

2.5.1 คุณลักษณะเด่นของภาษา Python

1. สนับสนุนแนวแบบคิดออปเจกต์โอเรียนเทด หรือ OOP (Object Oriented Programming)

2. เป็น Open Source

3. โค้ดที่เขียนด้วย Python สามารถนำไปรันบนระบบปฏิบัติการได้หลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สนับสนุนเทคโนโลยี COM ของ Ms-windows

5. Python รวมมาตรฐานการอินเตอร์เฟส Tkinter ซึ่งสนับสนุนบนระบบ X Windows, Ms-Windows และ Macintosh การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่นๆ

6. เป็น Dynamic Typing คือ สามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก

7. มี Buil-in Object Types คือ โครงสร้างของข้อมูลที่สามารถใช้ได้ใน Python ประกอบด้วย ลิสต์, ดิกชันนารี, สตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง

8. มีเครื่องมือต่างๆ มากมาย เช่น การประมวลผลเท็กซ์ไฟล์ การเรียงข้อมูล การเชื่อต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนคำ เป็นต้น

9. มีมอดูลสำหรับจัดการ Regular Expresion

10. มีมอดูลที่สร้างขึ้นจากนักพัฒนาสนับสนุนมากมาย ได้แก่ COM, Image, CORBA, ORBs, XML เป็นต้น

11. จัดการหน่วยความจำอย่างอัตโนมัติ สามารถจักการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ

12. อนุญาตให้ฝังชุดคำสั่งของ Python เอาไว้ภายในโค๊ดภาษา C/C++ ได้

13. อนุญาตให้โปรแกรมเมอร์สร้าง Dynamic Link Libray (DLL) เพื่อใช้ร่วมกับ Python

14. มีมอดูลสนับสนุนเกี่ยวกับเน็ตเวิร์ก โปรเซส เธรด Regular, Expression, Xml, GUI และอื่นๆ

15. ประกอบด้วยมอดูลสำหรับสร้าง Internet Script และติดต่อกับอินเทอร์เน็ตผ่าน Sockets, และทำหน้าที่เป็น CGI Script ตรอดจนใช้งานคำสั่ง FTP, Glopher, XML และอื่นๆอีกมาก

16. สามารถประมวลผลทางด้านวิยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ

17. มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น MySQL, Sybase, Oracle , Informix, ODBC และอื่นๆ

18. มีไลบรารีสนับสนุนด้านการสร้างภาพกราฟฟิก เช่น ทำภาพเบลอ หรือภาพชัด หรือเขียนข้อความบนภาพ ตลอดจนบันถึกไฟล์ในรูปแบบต่างๆ ได้อย่างสะดวกและมีประสิทธิภาพ

19. มีไลบรารีสนับสนุนด้านปัญยาประดิษฐ์

20. มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer

21. มีไลบรารีสำหรับสร้าง Shockwaves Flash (SWF) โดยไม่ต้องติดตั้ง Macromedia Flash

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 บอร์ด ESP8266

ESP8266 เป็นชื่อเรียกของชิปของโมดูล ESP8266 สำหรับติดต่อสื่อสารบนมาตรฐาน WiFi ทำงานที่แรงดันไฟฟ้า 3.0-3.6 V ทำงานใช้กระแสโดยเฉลี่ย 80 mA รองรับคำสั่ง Deep Sleep ในการประหยัดพลังงาน ใช้กระแสน้อยกว่า 10 ไมโครแอมป์ สามารถ Wake Up กลับมาส่งข้อมูลใช้เวลาน้อยกกว่า 2 มิลลิวินาที ภายในมี Low Power MCU 32 บิต ทำให้เขียนโปรแกรมสั่งงานได้ มีวงจร Analog Digital Converter ทำให้สามารถอ่านค่าจาก Analog ได้ความละเอียด 10 บิต ทำงานได้ที่อุณหภูมิ -40 ถึง 125 องศาเซลเซียส รายละเอียดเพิ่มเติมจากผู้ผลิตอ้างอิงตามลิงค์นี้ ESP8266 Datasheet

เมื่อนำชิป ESP8266 มาผลิตเป็นโมดูลหลายรุ่น ก็จะขึ้นต้นด้วย ESP866 แล้วตามด้วยรุ่น เช่น ESP-01, ESP-03, ESP-07, ESP-12E, ESP8266 ติดต่อกับ WiFi แบบ Serial สามารถเขียนโปรแกรมลงไปในชิป โดยใช้ Arduino IDE ได้ ทำให้การเขียนโปรแกรมและใช้งานเป็นเรื่องง่าย คล้ายกับการใช้ Arduino แน่นอนว่าสามารถติดต่ออุปกรณ์อื่นๆ เซนเซอร์ต่างๆ แบบสไตล์ Arduino ถ้ามีพื้นฐาน Arduino อยู่แล้ว ก็เข้าใจและใช้งานได้รวดเร็ว

โมดูล ESP8266 มีหลายรุ่น และมีรุ่นใหม่พัฒนาออกมาเรื่อยๆ โดยโครงสร้างและขาที่ใช้งานก็จะมีลักษณะคล้ายกันคือ

- GPIO0 เป็นขาสำหรับเลือกโหมด โดยเมื่อต่อกับ GND จะเข้าโหมดโปรแกรม เมื่อต้องการให้ทำงานปกติก็ไม่ต้องต่อ
- GPIO15 เป็นขาที่ต้องต่อลง GND เพื่อให้โมดูลทำงาน
- CH_PD หรือ EN เป็นขาที่ต้องต่อไฟ VCC เพื่อ Pull Up สัญญาณ ให้โมดูลทำงาน โมดูลบางรุ่นไม่มีขารีเซตมาให้ เมื่อต้องการรีเซต ให้ต่อขา CH_PD กับ GND
- รีเซตต่อกับไฟ VCC เพื่อ Pull Up สัญญาณ โดยเมื่อต้องการรีเซต ให้ต่อกับไฟ GND
- VCC เป็นขาสำหรับจ่ายไฟเลี้ยง ใช้ไฟเลี้ยง 3.0-3.6 V
- GND ต่อกับไฟ 0 V
- GPIO เป็นขาดิจิตอล INPUT/OUTPUT ทำงานที่ไฟ 3.3 V
- ADC เป็นขา Analog INPUT รับแรงดันสูงสุด 1 V ความละเอียด 10 บิต หรือ 1024 ค่า เวลาโปรแกรมเพียงมองหาขาเหล่านี้ แล้วต่อให้ครบเท่าที่มีขาให้ต่อ ก็สามารถโปรแกรม ESP8266 ได้ทุกรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 บอร์ด Wemos D1 mini

เป็นบอร์ดพัฒนา ESP8266 บนบอร์ดมีชิปสำหรับ USB to TTL อยู่บนบอร์ดแล้ว ทำให้สามารถเสียบเข้ากับคอมพิวเตอร์ผ่านพอร์ต USB ได้โดยตรง สามารถใช้ NodeMCU เฟิร์มแวร์ ในการเขียนโปรแกรมสั่งงาน หรือใช้โปรแกรม Arduino IDE ก็ได้ โดยใช้งานได้เหมือนกับบอร์ด NodeMCU ทุกประการ สามารถใช้ Arduino IDE และ Syntax สำหรับ Arduino มาเขียนได้เลย นั้นทำให้ Code ส่วนใหญ่ที่รันบน Arduino มาใส่ใน Wemos d1 ได้เลย และยังสามารถใช้ Libary ของ Arduino ส่วนใหญ่ได้ด้วย

บนตัวบอร์ดได้มีการฝังเสา WiFi มาให้เรียบร้อยโดยจะใช้ความที่ 2.4 Ghz เท่านั้น ทำให้สามารถใช้งานมันเป็นได้ทั้ง Client ซึ่งก็คือ สำหรับเชื่อมสัญญาณจาก Router หรือก็คือ เหมือนมือถือ หรือ Notebook นั้นเอง นอกจากนั้นแล้วยังสามารถทำเป็น Acces Point ได้ซึ่งเหมือน Router ที่ปล่อย WiFi ออกมาให้เชื่อม และเนื่องจากมันเป็นสัญญาณ WiFi อุปกรณ์ทุกเครื่องที่ใช้ WiFi จึง สามารถเชื่อมต่อกันได้

รูปที่ 2.3 WeMos D1 mini

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 รีเลย์ (Relay)

เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับ สวิตช์อิเล็กทรอนิกส์ ซึ่งสามารถนำรีเลย์ไปประยุกต์ใช้ในการควบคุมวงจรต่างๆ ในงานช่างอิเล็กทรอนิกส์มากมาย

รีเลย์ ประกอบด้วยส่วนสำคัญ 2 ส่วนหลัก คือ

1. ส่วนของขดลวด (Coil) เหนี่ยวนำกระแสต่ำ ทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แกนโลหะไปกระทุ้งให้หน้าสัมผัสต่อกัน ทำงานโดยการรับแรงดันจากภายนอกต่อคร่อมที่ขดลวดเหนี่ยวนำนี้ เมื่อขดลวดได้รับแรงดัน (ค่าแรงดันที่รีเลย์ต้องการขึ้นกับชนิดและรุ่นตามที่ผู้ผลิตกำหนด) จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะด้านในไปกระทุ้งให้แผ่นหน้าสัมผัสต่อกัน

2. ส่วนของหน้าสัมผัส (Contact) ทำหน้าที่เหมือนสวิตช์ จ่ายกระแสไฟให้กับ อุปกรณ์ที่ต้องการนั่นเอง

จุดต่อใช้งานมาตรฐาน ประกอบด้วย

1. จุดต่อ NC ย่อมาจาก Normal Close หมายความว่าปกติปิด หรือหากยังไม่มีไฟจ่ายให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปมักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา

2. จุดต่อ NO ย่อมาจาก Normal Open หมายความว่าปกติเปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปมักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิด

3. จุดต่อ C ย่อมากจาก Common คือ จุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ

2.7.1 ประเภทของรีเลย์

เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (Solenoid) รีเลย์ใช้ในการควบคุมวงจรไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power Relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

2. รี เลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุม บางทีเรียกกันง่ายๆ ว่า “รีเลย์"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 ชนิดของรีเลย์

การแบ่งชนิดของรีเลย์สามารถแบ่งได้ 11 แบบ คือ

ชนิดของรีเลย์แบ่งตามลักษณะของคอยล์ หรือแบ่งตามลักษณะการใช้งาน (Application) ได้แก่รีเลย์ดังต่อไปนี้

1. รีเลย์กระแส (Current Relay) คือ รีเลย์ที่ทำงานโดยใช้กระแส มีทั้งชนิดกระแสขาด (Under-Current) และกระแสเกิน (Over-Current)

2. รีเลย์แรงดัน (Voltage Relay) คือ รีเลย์ที่ทำงานโดยใช้แรงดันมีทั้งชนิดแรงดันขาด (Under-voltage) และ แรงดันเกิน (Over Voltage)

3. รีเลย์ช่วย (Auxiliary Relay) คือ รีเลย์ที่เวลาใช้งานจะต้องประกอบเข้ากับรีเลย์ชนิดอื่นจึงจะทำงานได้

4. รีเลย์กำลัง (Power Relay) คือ รีเลย์ที่รวมเอาคุณสมบัติของรีเลย์กระแส และรีเลย์แรงดันเข้าด้วยกัน

5. รีเลย์เวลา (Time Relay) คือ รีเลย์ที่ทำงานโดยมีเวลาเข้ามาเกี่ยวข้องด้วย ซึ่งมีอยู่ด้วยกัน 4 แบบ คือ

5.1 รีเลย์กระแสเกินชนิดเวลาผกผันกับกระแส (Inverse Time Over Current Relay) คือรีเลย์ที่มีเวลาทำงานเป็นส่วนกลับกับกระแส

5.2 รีเลย์กระแสเกินชนิดทำงานทันที (Instantaneous Over Current Relay) คือ รีเลย์ที่ทำงานทันทีทันใดเมื่อมีกระแสไหลผ่านเกินกว่าที่กำหนดที่ตั้งไว้

5.3 รีเลย์แบบดิฟฟินิตไทม์เล็ก (Definite Time Lag Relay) คือ รีเลย์ที่มีเวลาการทำงานไม่ขึ้นอยู่กับความมากน้อยของกระแสหรือค่าไฟฟ้าอื่นๆ ที่ทำให้เกิดงานขึ้น

5.4 รีเลย์แบบอินเวอสดิฟฟินิตมินิมั่มไทม์เล็ก (Inverse Definite Time Lag Relay) คือ รีเลย์ที่ทำงานโดยรวมเอาคุณสมบัติของเวลาผกผันกับกระแส (Inverse Time) และแบบดิฟฟินิตไทม์แล็ก (Definite Time Lag Relay) เข้าด้วยกัน

6. รีเลย์กระแสต่าง (Differential Relay) คือ รีเลย์ที่ทำงานโดยอาศัยผลต่างของกระแส

7. รีเลย์มีทิศ (Directional Relay) คือ รีเลย์ที่ทำงานเมื่อมีกระแสไหลผิดทิศทาง มีทั้งแบบรีเลย์กำลังมีทิศ (Directional Power Relay) และรีเลย์กระแสมีทิศ (Directional Current Relay)

8. รีเลย์ระยะทาง (Distance Relay) คือ รีเลย์ระยะทางมีแบบต่างๆ ดังนี้

- รีแอกแตนซ์รีเลย์ (Reactance Relay)
- อิมพีแดนซ์รีเลย์ (Impedance Relay)
- โมห์รีเลย์ (Mho Relay)
- โอห์มรีเลย์ (Ohm Relay)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โพลาไรซ์โมห์รีเลย์ (Polaized Mho Relay)

- ออฟเซทโมห์รีเลย์ (Off Set Mho Relay)

9. รีเลย์อุณหภูมิ (Temperature Relay) คือ รีเลย์ที่ทำงานตามอุณหภูมิที่ตั้งไว้

10. รีเลย์ความถี่ (Frequency Relay) คือ รีเลย์ที่ทำงานเมื่อความถี่ของระบบต่ำกว่าหรือมากกว่าที่ตั้งไว้

11. บูคโฮลซ์รีเลย์ (Buchholz's Relay) คือ รีเลย์ที่ทำงานด้วยก๊าซ ใช้กับหม้อแปลงที่แช่อยู่ในน้ำมันเมื่อเกิด ฟอลต์ ขึ้นภายในหม้อแปลง จะทำให้น้ำมันแตกตัวและเกิดก๊าซขึ้นภายในไปดันหน้าสัมผัส ให้รีเลย์ทำงาน

รูปที่ 2.4 รีเลย์ (Relay)

2.8 มอเตอร์ไฟฟ้า (Motor)

มอเตอร์ไฟฟ้า (Electric Motor) เป็นอุปกรณ์ไฟฟ้าที่แปลงพลังงานไฟฟ้าเป็นพลังงานกล การทำงานปกติของมอเตอร์ไฟฟ้าส่วนใหญ่เกิดจากการทำงานร่วมกันระหว่างสนามแม่เหล็กของแม่เหล็กในตัวมอเตอร์ และสนามแม่เหล็กที่เกิดจากกระแสในขดลวดทำให้เกิดแรงดูดและแรงผลักของสนามแม่เหล็กทั้งสอง ในการใช้งาน ตัวอย่างเช่น ในอุตสาหกรรมการขนส่งใช้มอเตอร์ฉุดลาก เป็นต้น นอกจากนั้นแล้ว มอเตอร์ไฟฟ้ายังสามารถทำงานได้ถึงสองแบบ ได้แก่ การสร้างพลังงานกล และการผลิตพลังงานไฟฟ้า

มอเตอร์ไฟฟ้าถูกนำไปใช้งานที่หลากหลาย เช่น พัดลมอุตสาหกรรม เครื่องเป่า ปั๊ม เครื่องมือ เครื่องใช้ในครัวเรือน และดิสก์ไดรฟ์ มอเตอร์ไฟฟ้าสามารถขับเคลื่อนโดยแหล่งจ่ายไฟกระแสตรง (DC) เช่น จากแบตเตอรี่, ยานยนต์หรือวงจรเรียงกระแส หรือจากแหล่งจ่ายไฟกระแสสลับ (AC) เช่น จากไฟบ้าน อินเวอร์เตอร์ หรือเครื่องปั่นไฟ มอเตอร์ขนาดเล็กอาจจะพบในนาฬิกาไฟฟ้า มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั่วไปที่มีขนาดและคุณลักษณะมาตรฐานสูงจะให้พลังงานกลที่สะดวกสำหรับใช้ในอุตสาหกรรม มอเตอร์ไฟฟ้าที่ใหญ่ที่สุดใช้สำหรับการใช้งานลากจูงเรือ และการบีบอัดท่อส่งน้ำมันและปั๊มป์สูบจัดเก็บน้ำมันซึ่งมีกำลังถึง 100 เมกะวัตต์ มอเตอร์ไฟฟ้าอาจจำแนกตามประเภทของแหล่งที่มาของพลังงานไฟฟ้า หรือตามโครงสร้างภายในหรือตามการใช้งานหรือตามการเคลื่อนไหวของเอาต์พุต และอื่นๆ อุปกรณ์ เช่น ขดลวดแม่เหล็กไฟฟ้าและลำโพงที่แปลงกระแสไฟฟ้าให้เป็นการเคลื่อนไหว แต่ไม่ได้สร้างพลังงานกลที่ใช้งานได้ จะเรียกถูกว่า Actuator และ Transducer ตามลำดับ คำว่ามอเตอร์ไฟฟ้านั้น ต้องใช้สร้างแรงเชิงเส้น (Linear Force) หรือแรงบิด (Torque) หรือเรียกอีกอย่างว่า หมุน (Rotary) เท่านั้น

2.8.1 ชนิดของมอเตอร์ไฟฟ้า

1. มอเตอร์ไฟฟ้ากระแสสลับ (Alternating Current Motor) หรือเรียกว่า เอ.ซี มอเตอร์ (A.C. Motor) การแบ่งชนิดของมอเตอร์ไฟฟ้าสลับแบ่งออกเป็น 3 ชนิดได้แก่

1.1 มอเตอร์ไฟฟ้ากระแสสลับชนิด 1 เฟส หรือเรียกว่าซิงเกิลเฟสมอเตอร์ (A.C. Sing Phase) จะใช้กับแรงดันไฟฟ้า 220 โวลต์มีสายไฟเข้า 2 สาย มีแรงม้าไม่สูง ส่วนใหญ่ตามบ้านเรือน

- สปลิทเฟส มอเตอร์ (Split-Phase Motor)
- คาปาซิเตอร์ มอเตอร์ (Capacitor Motor)
- รีพัลชั่นมอเตอร์ (Repulsion-type Motor)
- ยูนิเวอร์แวซลมอเตอร์ (Universal Motor)
- เช็ดเดดโพล มอเตอร์ (Shaded-pole Motor)

1.2 มอเตอร์ไฟฟ้าสลับชนิด 2 เฟสหรือเรียกว่าทูเฟสมอเตอร์ (A.C.Two Phase Motor)

1.3 มอเตอร์ไฟฟ้ากระแสสลับชนิด 3 เฟสหรือเรียกว่าทีเฟสมอเตอร์ (A.C. Three Phase Motor) เป็นมอเตอร์ที่ใช้ในงานอุตสาหกรรมต้องใช้ระบบไฟฟ้า 3 เฟส ใช้แรงดัน 380 โวลต์ มีสายไฟเข้ามอเตอร์ 3 สาย

2.มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor) หรือเรียกว่า ดี.ซี มอเอตร์ (D.C. Motor) การแบ่งชนิดของมอเตอร์ไฟฟ้ากระแสตรงแบ่งออกได้ดังนี้

มอเตอร์ไฟฟ้ากระแสตรงแบ่งออกเป็น 3 ชนิดได้แก่

1. มอเตอร์แบบอนุกรมหรือเรียกว่า ซีรีส์มอเตอร์ (Series Motor)
2. มอเตอร์แบบอนุขนานหรือเรียกว่า ชันท์มอเตอร์ (Shunt Motor)
3. มอเตอร์ไฟฟ้าแบบผสมหรือเรียกว่า คอมเปาวด์มอเตอร์ (Compound Motor)

2.8.2 ส่วนประกอบหลักๆ ของมอเตอร์ไฟฟ้า

1. ขดลวดสนามแม่เหล็ก (Field Coil) คือ ขดลวดที่ถูกพันอยู่กับขั้วแม่เหล็กที่ ยึดติดกับโครงมอเตอร์ ทำหน้าที่กำเนิดขั้วแม่เหล็กขั้วเหนือ (N) และขั้วใต้ (S) แทนแม่เหล็กถาวรขดลวดที่ใช้เป็นขดลวดอาบน้ำยาฉนวน สนามแม่เหล็กจะเกิดขึ้นเมื่อจ่ายแรงดันไฟตรงให้มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขั้วแม่เหล็ก (Pole Pieces) คือ แกนสำหรับรองรับขดลวดสนามแม่เหล็กถูกยึดติดกับโครงมอเตอร์ด้านใน ขั้วแม่เหล็กทำมาจากแผ่นเหล็กอ่อนบางๆ อัดซ้อนกัน (Lamination Sheet Steel) เพื่อลดการเกิดกระแสไหลวน (Edy Current) ที่จะทำให้ความเข้าของสนามแม่เหล็กลดลง ขั้วแม่เหล็กทำหน้าที่ให้กำเนิดขั้วสนามแม่เหล็กมีความเข้มสูงสุด แทนขั้วสนามแม่เหล็กถาวร ผิวด้านหน้าของขั้วแม่เหล็กทำให้โค้งรับกับอาร์เมเจอร์พอดี

3. โครงมอเตอร์ (Motor Frame) คือ ส่วนเปลือกหุ้มภายนอกของมอเตอร์ และยึดส่วนอยู่กับที่ (Stator) ของมอเตอร์ไว้ภายในร่วมกับฝาปิดหัวท้ายของมอเตอร์ โครงมอเตอร์ทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กระหว่างขั้วแม่เหล็กให้เกิดสนามแม่เหล็กครบวงจร

4. อาร์เมเจอร์ (Armature) คือ ส่วนเคลื่อนที่ (Rotor) ถูกยึดติดกับเพลา (Shaft) และรองรับการหมุนด้วยที่รองรับการหมุน (Bearing) ตัวอาร์เมเจอร์ทำจากเหล็กแผ่นบางๆ อัดซ้อนกัน ถูกเซาะร่องออกเป็นส่วนๆ เพื่อไว้พันขดลวดอาร์เมเจอร์ (Armature Winding) ขดลวดอาร์เมเจอร์เป็นขดลวดอาบน้ำยาฉนวน ร่องขดลวดอาร์เมเจอร์จะมีขดลวดพันอยู่และมีลิ่มไฟเบอร์อัดแน่นขัดขดลวดอาร์เมเจอร์ไว้ ปลายขดลวดอาร์เมเจอร์ต่อไว้กับคอมมิวเตเตอร์ อาร์เมเจอร์ผลักดันของสนามแม่เหล็กทั้งสอง ทำให้อาร์เมเจอร์หมุนเคลื่อนที่

5. คอมมิวเตเตอร์ (Commutator) คือ ส่วนเคลื่อนที่อีกส่วนหนึ่ง ถูกยึดติดเข้ากับอาร์เมเจอร์และเพลาร่วมกัน คอมมิวเตเตอร์ทำจากแท่งทองแดงแข็งประกอบเข้าด้วยกันเป็นรูปทรงกระบอก แต่ละแท่งทองแดงของคอมมิวเตเตอร์ถูกแยกออกจากกันด้วยฉนวนไมก้า (Mica) อาร์เมเจอร์ คอมมิวเตเตอร์ทำหน้าที่เป็นขั้วรับแรงดันไฟตรงที่จ่ายมาจากแปรงถ่าน เพื่อส่งไปให้ขดลวดอาร์เมเจอร์

6. แปรงถ่าน (Brush) คือ ตัวสัมผัสกับคอมมิวเตเตอร์ ทำเป็นแท่งสี่ เหลี่ยมผลิตมาจากคาร์บอนหรือแกรไฟต์ผสมผงทองแดง เพื่อให้แข็งและนำไฟฟ้าได้ดี มีสายตัวนำต่อร่วมกับแปรงถ่านเพื่อไปรับแรงดันไฟตรงที่จ่ายเข้ามา แปรงถ่านทำหน้าที่รับแรงดันไฟตรงจากแหล่งจ่าย จ่ายผ่านไปให้คอมมิวเตเตอร์

รูปที่ 2.5 มอเตอร์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 อินฟราเรด (Infrared IR)

อินฟราเรด คือ ลำแสงอินฟราเรดหรือ รังสีใต้แสงสีแดงหรือรังสีความร้อนมีลักษณะเป็นคลื่นวิทยุความถี่สูง ที่มีความยาวคลื่นต่ำกว่าความยาวคลื่นแสงสีแดงทำให้สายตามนุษย์ไม่สามารถมองเห็น มีการเดินทางของแสงเป็นแนวตรง ระดับสายตา ไม่สามารถทะลุผ่านวัตถุ ได้ถูกนำมาใช้ในการสื่อสารนานแล้วหลายทศวรรษ

2.9.1 ลักษณะของอินฟราเรด

ลื่นสัญญาณอินฟราเรด (Infrared IR) เป็นคลื่นแม่เหล็กไฟฟ้ามีความถี่ระหว่าง 300 GHz ถึง 400 THz (TeraHertz) มีความยาวคลื่นตั้งแต่ 1 มิลลิเมตร ถึง 770 นาโนเมตร เป็นความถี่ใต้แสงสีแดงที่ตามนุษย์ไม่สามารถมองเห็น เนื่องจากเป็นคลื่นสั้นจึงส่งสัญญาณได้เป็นแนวตรงระยะทางไม่มาก 30 ถึง 80 ฟุต ไม่สามารถเดินทางทะลุวัตถุได้

ข้อดีของการใช้งานคลื่นอินฟราเรดคือ สามารถใช้กับอุปกรณไร้สายระยะใกล้ๆ ได้ อุปกรณ์มีขนาดเล็ก ใช้พลังงานน้อย ราคาถูก เชื่อมต่ออุปกรณ์ได้ง่าย สัญญาณมีความปลอดภัยสูงไม่ต้องขออนุญาตการใช้งานคลื่น ไม่มีสัญญาณแทรก

ข้อเสียของการใช้งานคลื่นอินฟราเรดคือ ใช้ในระยะทางไกลๆ ไม่ได้ ใช้ภายนอกอาคารที่มีแสงแดดไม่ได้ เนื่องจากแสงแดดเป็นคลื่นอินฟราเรดเหมือนกันทำให้รบกวนกัน ถ้ามีการบังทางเดินของแสงจะทำให้สัญญาณขาดหาย

รูปที่ 2.6 หลอดอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

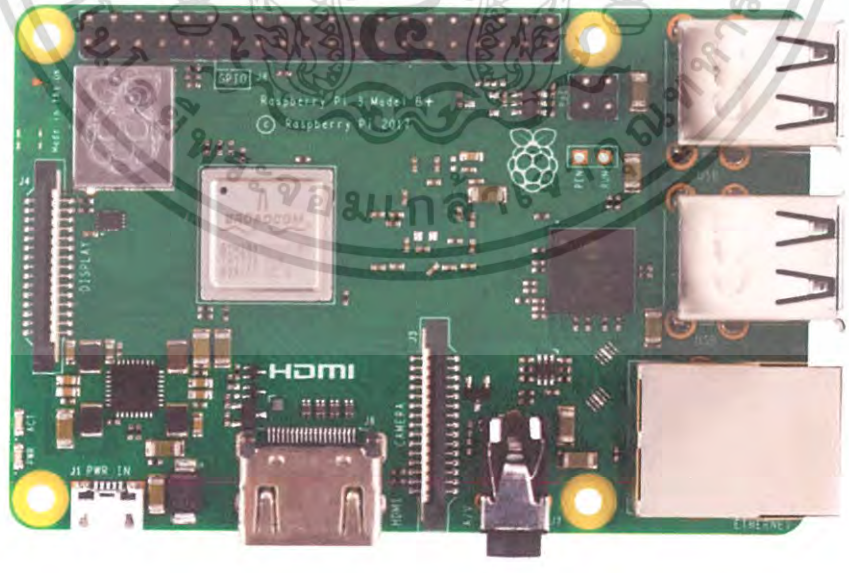
บทที่ 3
วิธีการดำเนินงาน

3.1 การเลือกอุปกรณ์

3.1.1 ไมโครคอนโทรลเลอร์

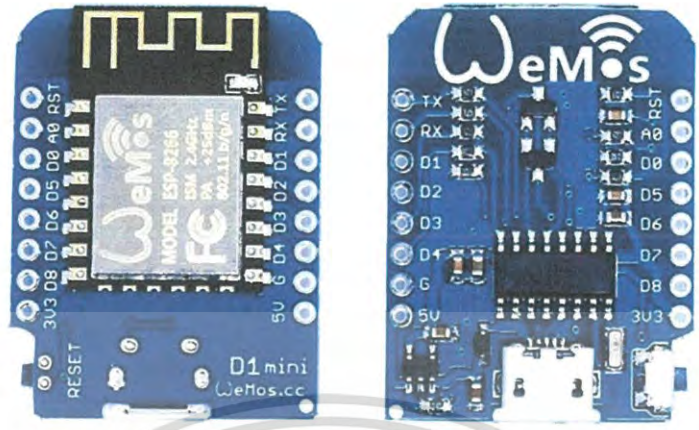
ไมโครคอนโทรลเลอร์ในปัจจุบันนั้นมีให้เลือกใช้หลากหลาย ตามการใช้งานของแต่ละคน ดังนั้นผู้ใช้งานจึงเลือกคุณสมบัติของไมโครคอนโทรลเลอร์ให้เหมาะสมกับลักษณะการใช้งาน สำหรับโครงงานเครื่องใช้ไฟฟ้าภายในบ้านนั้น ใช้ไมโครคอนโทรลเลอร์ 2 ตัว ซึ่งตัวหนึ่งทำหน้าที่เป็นตัวกลางในการประมวลผล รับส่งข้อมูลระหว่างโทรศัพท์มือถือกับเครื่องใช้ไฟฟ้าภายในบ้าน และไมโครคอนโทรลเลอร์อีกตัวทำหน้าที่ควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน โดยไมโครคอนโทรลเลอร์ทั้ง 2 อย่างนั้นต้องสามารถสื่อสารกันด้วยสัญญาณอินเทอร์เน็ต นอกจากนี้ ไมโครคอนโทรลเลอร์ที่ใช้ควบคุมอุปกรณ์ไฟฟ้านั้นควรมีขนาดเล็ก เนื่องจากต้องติดตั้งในพื้นที่ที่ขนาดจำกัด และเพื่อความสวยงามในการติดตั้ง

ดังนั้นจึงได้ข้อสรุปในการเลือกไมโครคอนโทรลเลอร์ โดยไมโครคอนโทรลเลอร์ที่ทำหน้าที่เป็นตัวกลางในการประมวลผลคือ Rasberry Pi 3 b+ และไมโครคอนโทรลเลอร์ที่ใช้ควบคุมอุปกรณ์ไฟฟ้านั้น คือ WeMos D1 Mini ซึ่งทั้ง 2 ตัวนี้มีคุณสมบัติตามที่ผู้ใช้ต้องการ และสามารถประยุกต์ใช้ได้หลากหลาย และสามารถรับส่งข้อมูลผ่านระบบอินเทอร์เน็ตไร้สายได้



รูปที่ 3.1 Raspberry Pi B+

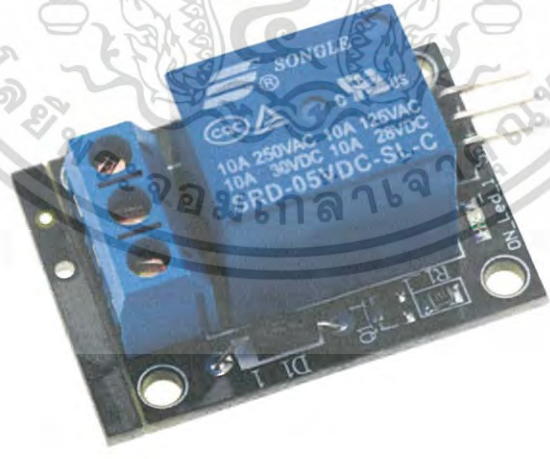
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 บอร์ด WeMos D1 Mini

3.1.2 รีเลย์ (Relay)

จากโครงงานเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียง อุปกรณ์ควบคุมไฟฟ้าภายในบ้านนั้นควรมีขนาดไม่ใหญ่ และให้เหมาะสมกับการใช้งาน และมีพื้นที่จำกัดในการติดตั้ง ดังนั้นจึงเลือกใช้ Relay Sheild for WeMos D1 Mini เพราะ Relay นี้สามารถเชื่อมต่อด้านบนของ WeMos D1 Mini ทำให้ประหยัดพื้นที่ และเหมาะสมกับการใช้งาน

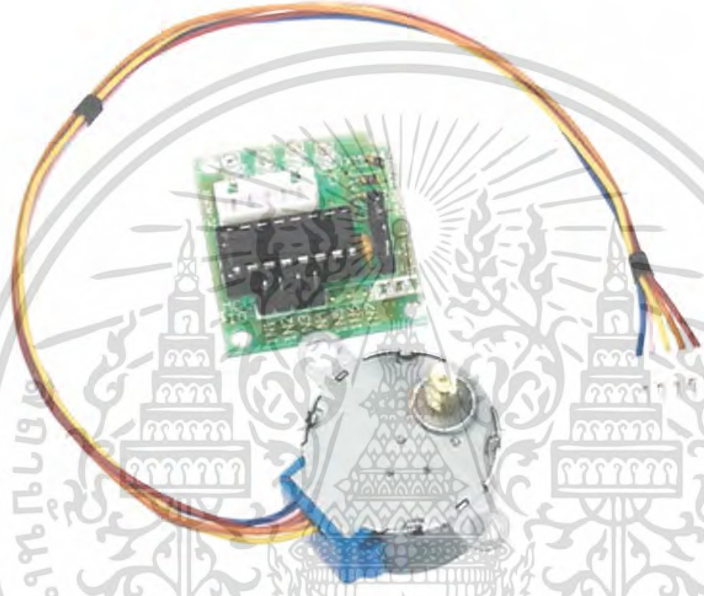


รูปที่ 3.3 Relay Shield for WeMos D1 Mini

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 มอเตอร์

มอเตอร์ในโครงงานเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียงนั้น ใช้สำหรับเปิดปิดม่าน ดังนั้นมอเตอร์ไม่ต้องการความเร็ว น้ำหนักเบา และให้มีขนาดเล็กเพื่อความสวยงามในการติดตั้ง จึงเลือกใช้เสต็ปมอเตอร์ ซึ่งเป็นมอเตอร์สำหรับงานที่ไม่ต้องใช้ความเร็วสูง นอกจากนี้ยังมีราคาที่ถูกและขนาดที่เล็ก และเลือกใช้รุ่น 28BYJ-48 Stepper Motor และบอร์ดไดรว์เวอร์ ULN2003 เนื่องจากรุ่นนี้สามารถหาซื้อได้ทั่วไปตามท้องตลาดและมีขนาดและกำลังที่เหมาะสมในการติดตั้งและใช้งาน ดังรูปที่ 3.4



รูปที่ 3.4 28BYJ-48 Stepper Motor และบอร์ดไดรว์เวอร์ ULN2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

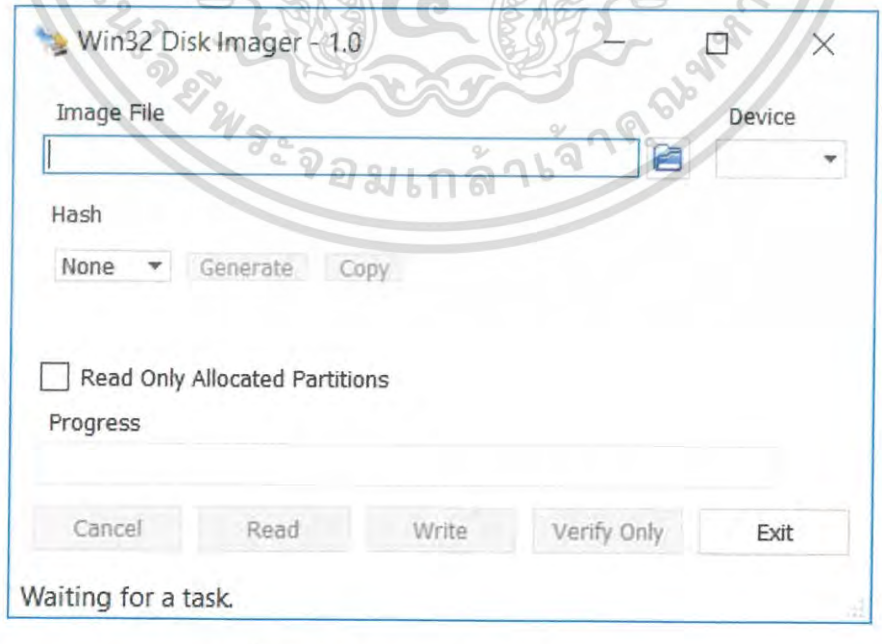
3.2 การตั้งค่า Raspberry Pi

Raspberry pi ก็เปรียบเหมือน Computer ทั่วไป จำเป็นต้องลง OS ก่อนการใช้งาน ในที่นี้จะลง Raspbian OS

3.2.1 Download Raspbian OS ได้จาก https://www.raspberrypi.org/downloads/ และทำการแตกไฟล์ zip ดังรูปที่ 3.5

รูปที่ 3.5 หน้า Download Raspbian

3.2.2 ใช้โปรแกรม Win32 Disk Imager ในการลง Raspbian OS ลง SD การ์ด ดังรูปที่ 3.6



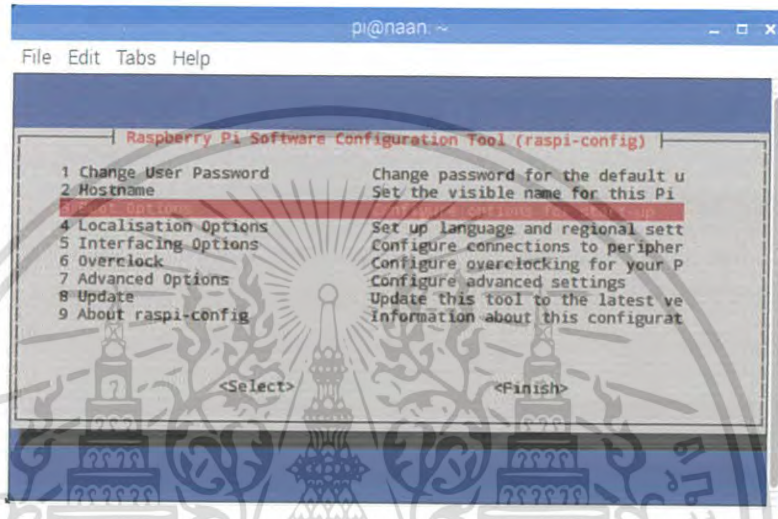
รูปที่ 3.6 โปรแกรม Win32 Disk Imager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 นำ Micro SD Card กลับไปเสียบที่ Pi หลังจากนั้น เสียบ Mouse, Keyboard, หน้าจอ และแหล่งจ่ายไฟ

3.2.4 เปิดเครื่องมา ต่อ WiFi หลังจากนั้นเปิดหน้า Terminal ขึ้นมา

3.2.5 พิมพ์คำสั่ง sudo raspi-config ลงไป จะเห็นหน้าดังรูปที่ 3.7 ใช้ลูกครเลื่อนเพื่อเปลี่ยนคำสั่งต่างๆ



รูปที่ 3.7 หน้าต่างตั้งค่า

1. เข้าไปที่ Interfacing Option เข้าไปที่ Interfacing Option แล้ว Enable SSH
2. เข้าไปที่ Change User Password และตั้งรหัสตามที่ต้องการ
3. เข้าไปที่ Localisation Option แล้วกด Change Timezone เลือก Asia > Bangkok
4. เข้าไปที่ Advance Options แล้วกด Expand filesystem รอสักพักเครื่องจะ Reboot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ลงโปรแกรม HomeBridge ใน Raspberry Pi

HomeBridge เป็น Server ที่เขียนด้วย NodeJS ซึ่งทำหน้าที่จำลองเป็น HomeKit API ให้คุณสามารถสร้างเครื่องใช้ไฟฟ้าที่รองรับ HomeKit ได้ โดย HomeBridge มี Plugin ให้ใช้เชื่อมต่อเครื่องใช้ไฟฟ้าหลายแบบ ไม่ว่าจะทำเอง เช่น การต่อ Relay หรือเชื่อมต่อกับ ESP8266 นอกจากนั้นยังทำให้อุปกรณ์ iOT ที่ไม่รองรับ HomeKit สามารถใช้งานผ่าน HomeKit ได้ เช่น Sonoff

3.3.1 ทำการลง NodeJS โดยการพิมพ์คำสั่งด้านล่างลงไปใน Command Line (ทีละบรรทัด)

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
sudo apt-get install -y nodejs
```

3.3.2 ลง Avahi และ Dependencies อื่นๆ

```
sudo apt-get install libavahi-compat-libdnssd-dev
```

3.3.3 ลง HomeBridge

```
sudo npm install -g --unsafe-perm homebridge
```

3.3.4 เมื่อพิมพ์คำสั่งด้านบนเสร็จ สามารถทดลองรัน HomeBridge ได้โดยการพิมพ์ Homebridge ลงใน Command Line หลังจากนั้นให้เปิดแอป Home บน iPhone ขึ้นมาแล้วกด + จะเจอ HomeBridge แล้วใส่รหัส

เมื่อเพิ่ม HomeBridge ลงไปใน iPhone ได้แล้ว Setup และลง HomeBridge ใน Raspberry Pi สำเร็จแล้ว และตอนนี้สามารถเพิ่มเครื่องใช้ไฟฟ้าไปยัง HomeBridge ได้โดยการลง Plugin

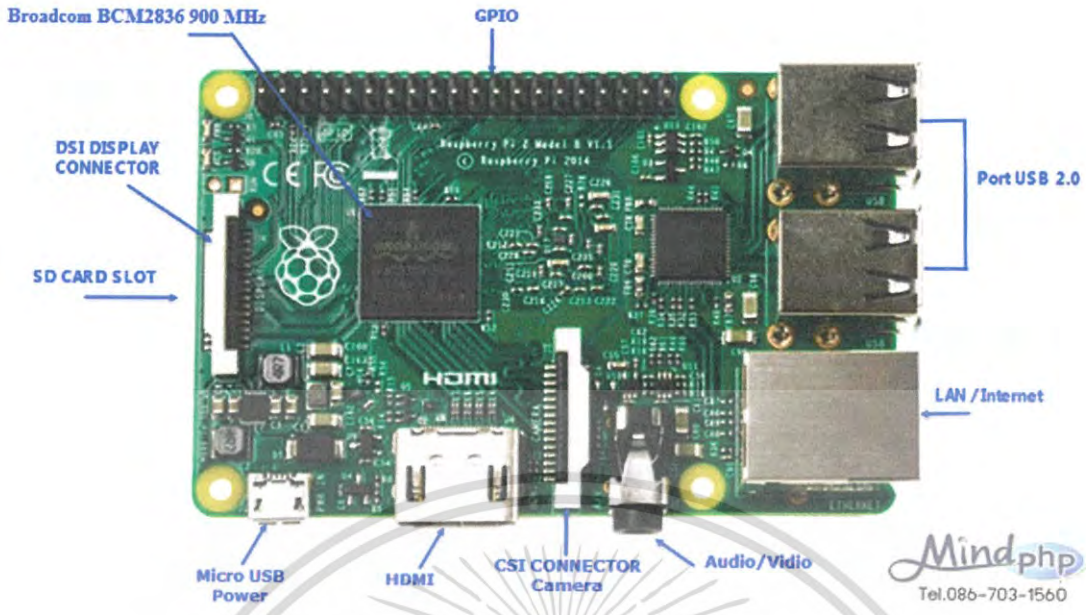
3.4 การเชื่อมต่ออุปกรณ์

Port ต่างๆ ของบอร์ด Raspberry Pi 3 B+ ประกอบด้วย

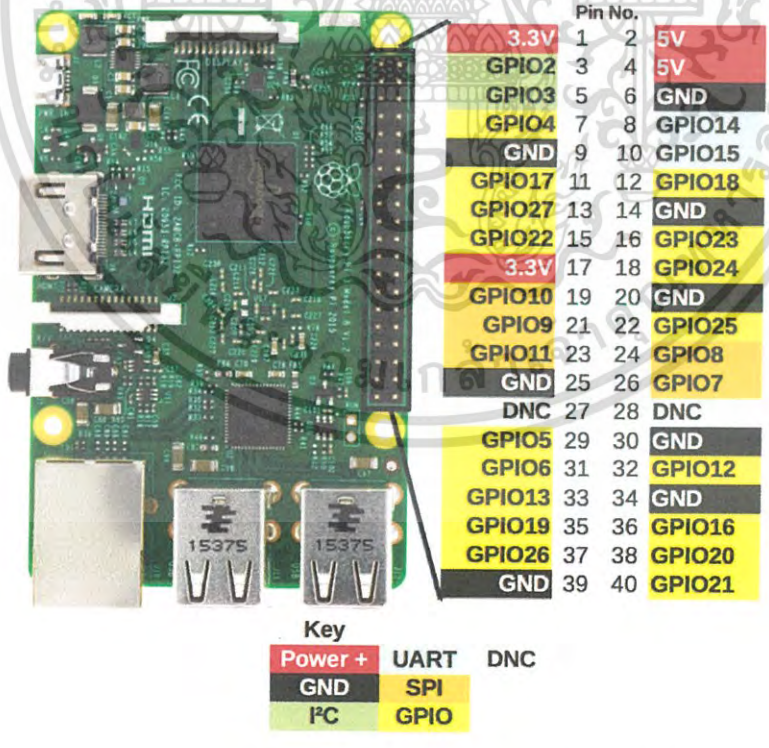
1. HDMI Port
2. Micro USB 5V Power Input
3. SD Card Slot

ดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

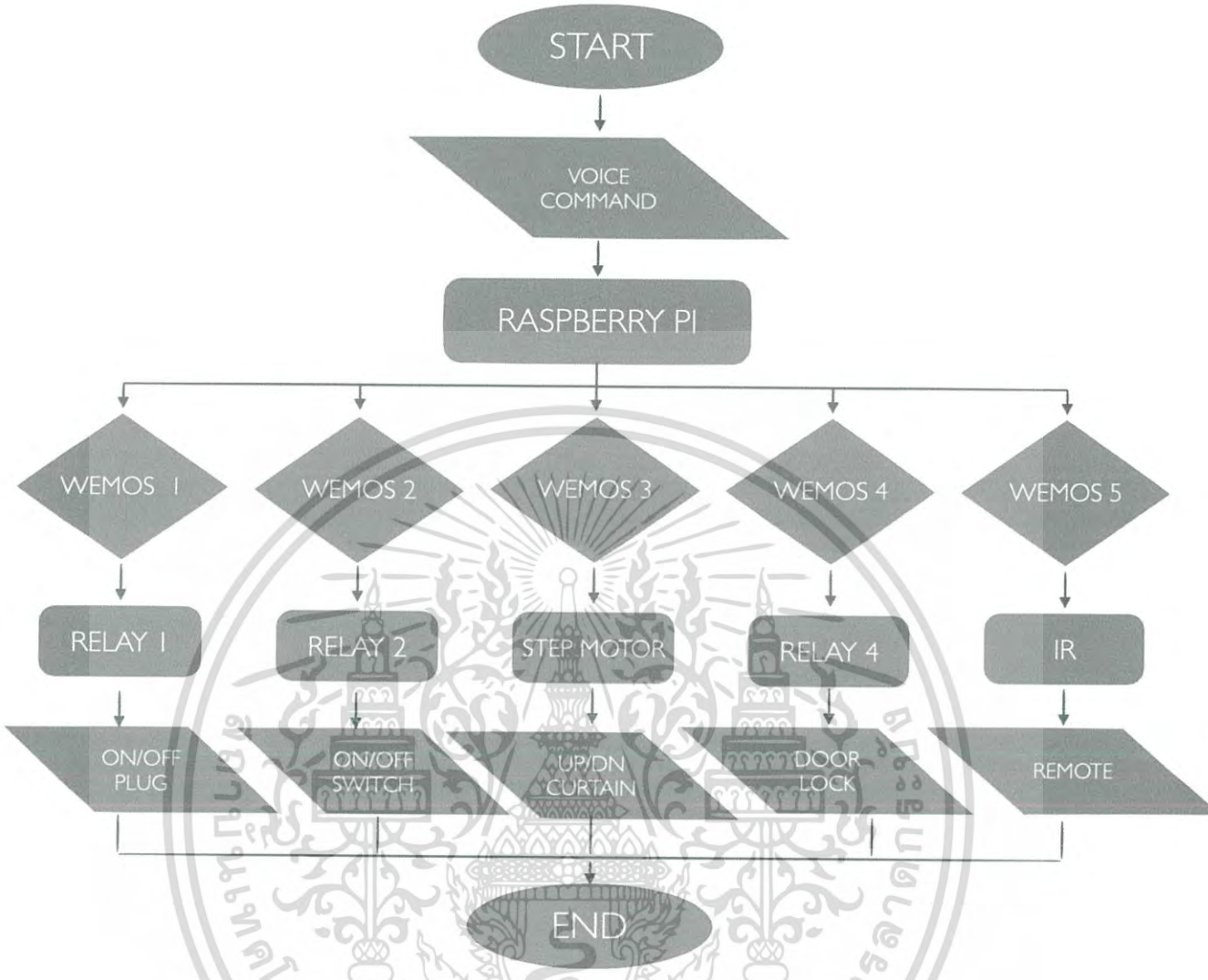


รูปที่ 3.8 พอร์ตเชื่อมต่อของ Raspberry Pi 3 B+



รูปที่ 3.9 หมายเลข GPIO ของ Pin ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 รูปแบบการเชื่อมต่อระหว่างอุปกรณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การลง HomeBridge Plugin

การลง HomeBrdge Plugin เพื่อเพิ่มความสามารถการควบคุม Relay ให้ระบบ HomeKit Plugin ที่จะทำการลงชื่อ homebridge-gpio-relay สามารถทำการลงได้ด้วยคำสั่งดังนี้

```
sudo npm install -g homebridge-gpio-relay
git clone git://github.com/jamesblanksby/quick2wire-gpio-admin.git
cd quick2wire-gpio-admin
make
sudo make install
sudo adduser $USER gpio
```

เมื่อลงเสร็จ ให้แก้ไขไฟล์ config เพื่อตั้งค่าให้กับ Relay

```
cd
sudo nano config.json
```

จะเห็นหน้าสำหรับแก้ไขไฟล์ ให้แก้ไขไฟล์ให้อยู่ในรูปแบบนี้ (สามารถดูความหมายของพารามีเตอร์เพิ่มเติมได้ที่หน้า NPM ของ homebridge-gpio-relay

```
{
"bridge": {
"name": "Homebridge",
"username": "CC:22:3D:E3:CE:30",
"port": 51826,
"pin": "031-45-154"
},
 "accessories": [{
"accessory": "relay",
"name": "My relay light 1",
"pin": 7
}]
}
```

กด <code>ctrl+x </code> ตามด้วย <code>y</code> เพื่อออกและเซฟไฟล์

ทดลองรัน Homebridge โดยการพิมพ์ <code>homebridge</code>

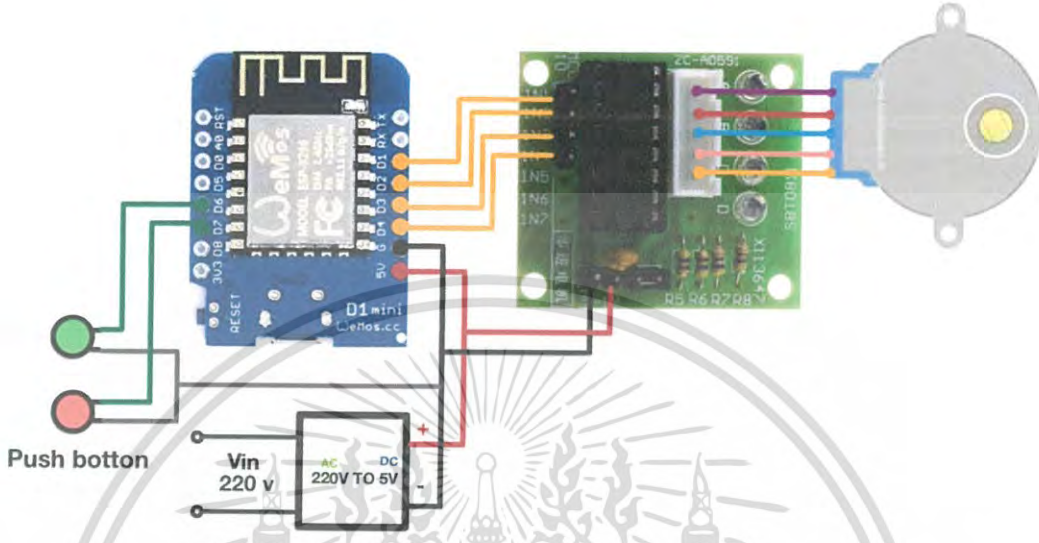
หากทุกอย่างถูกต้อง ตอนนี้จะสามารถควบคุม Relay ได้ด้วย HomeBridge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การติดตั้งอุปกรณ์

3.6.1 การติดตั้งระบบม่านไฟฟ้าสั่งงานด้วยเสียง

1. ต่อวงจรทำงานของระบบม่านไฟฟ้าสั่งงานด้วยเสียง ดังรูปที่ 3.11

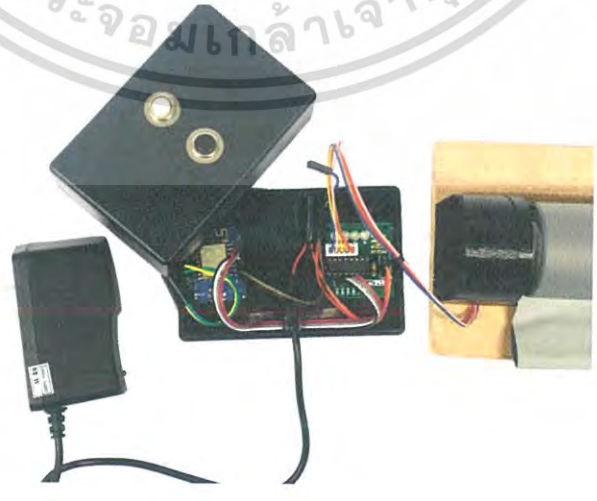


รูปที่ 3.11 วงจรสั่งการม่านไฟฟ้าสั่งงานด้วยเสียง

2. นำ Code ที่เขียนเพื่อสั่งการทำงานของม่านไฟฟ้าสั่งการด้วยเสียงลงในบอร์ด Arduino Wemos D1 Mini

3. ทำการสั่งงานเพื่อทดสอบการทำงานและจัดตำแหน่งของม่านไฟฟ้าสั่งงานด้วยเสียง

4. นำอุปกรณ์ควบคุมการม่านไฟฟ้าสั่งการด้วยเสียงทั้งหมดประกอบเข้าด้วยกันรวมถึงประกอบร่วมกับม่าน ดังรูปที่ 3.12

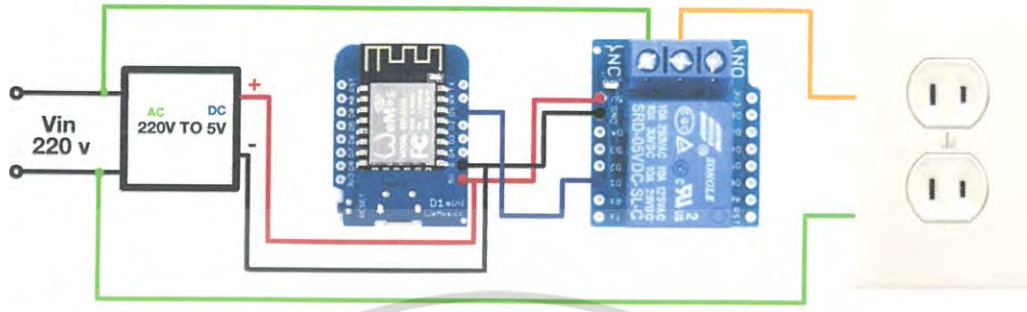


รูปที่ 3.12 การติดตั้งม่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 การติดตั้งปลั๊กไฟฟ้าสั่งงานด้วยเสียง

1. ต่อวงจรควบคุมการทำงานของปลั๊กไฟฟ้าสั่งงานด้วยเสียง ดังรูปที่ 3.13



รูปที่ 3.13 วงจรการทำงานของปลั๊กไฟฟ้า

2. นำ Code ที่เขียนสั่งการทำงานของปลั๊กไฟฟ้าสั่งงานด้วยเสียงลงในบอร์ด Arduino Wemos D1 Mini

3. ทดลองสั่งงานปลักไฟฟ้าสั่งงานด้วยเสียงและแก้ไขปัญหา

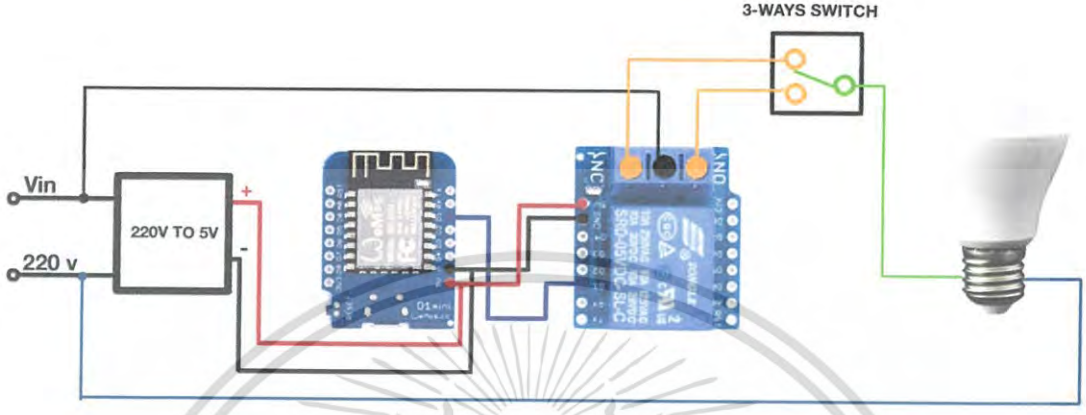
4. นำอุปกรณ์ควบคุมปลั๊กไฟฟ้าสั่งงานด้วยเสียงทั้งหมดประกอบเข้าด้วยกัน ดังรูปที่ 3.14

รูปที่ 3.14 ชุดปลั๊กไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3 การติดตั้งสวิตช์ไฟฟ้าสั่งงานด้วยเสียง

1. ต่อวงจรควบคุมสวิตช์ไฟฟ้าสั่งงานด้วยเสียง ดังรูปที่ 3.15



รูปที่ 3.15 วงจรสั่งงานสวิตช์ไฟฟ้าสั่งงานด้วยเสียง

2. นำ Code ที่เขียนเพื่อควบคุมการสั่งงานสวิตช์ไฟฟ้าสั่งงานด้วยเสียงลงในบอร์ด Arduino Wemos D1 Mini

3. ทดสอบการทำงานของสวิตช์ไฟฟ้าทั้งการสั่งงานด้วยเสียง และการกดสวิตช์ไฟฟ้าโดยตรง

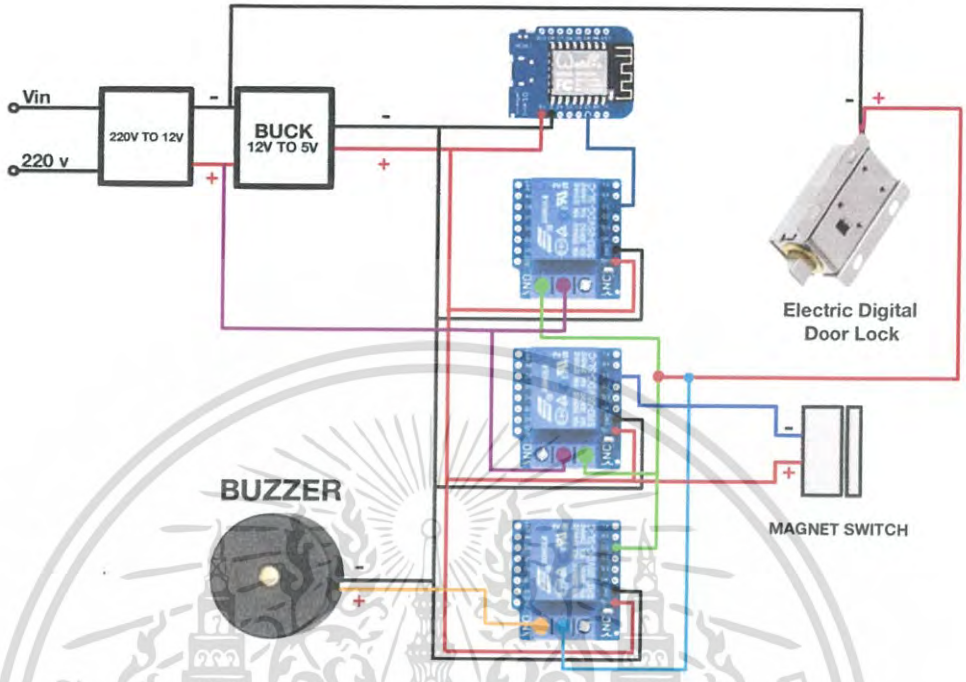
4. นำอุปกรณ์ควบคุมทั้งหมดประกอบเข้าด้วยกัน ดังรูปที่ 3.16

รูปที่ 3.16 ชุดอุปกรณ์สวิตช์ไฟฟ้าสั่งงานด้วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.4 การติดตั้งกลอนประตูไฟฟ้าสั่งงานด้วยเสียง

1. ต่อวงจรกลอนประตูไฟฟ้าสั่งงานด้วยเสียงดังรูปที่ 3.17

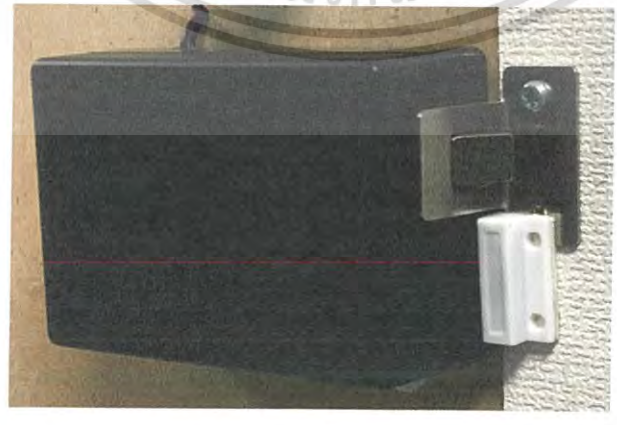


รูปที่ 3.17 วงจรไฟฟ้ากลอนประตูสั่งงานด้วยเสียง

2. นำ Code ที่เขียนเพื่อสั่งงานกลอนประตูไฟฟ้าสั่งงานด้วยเสียงลงในบอร์ด Arduino Wemos D1 Mini

3. ทดสอบการทำงานของระบบกลอนประตูไฟฟ้าสั่งงานด้วยเสียงทั้งระบบ และแก้ไขปัญหา

4. ประกอบอุปกรณ์ควบคุมกลอนประตูไฟฟ้าสั่งงานด้วยเสียงเข้าด้วยกัน ดังรูปที่ 3.18

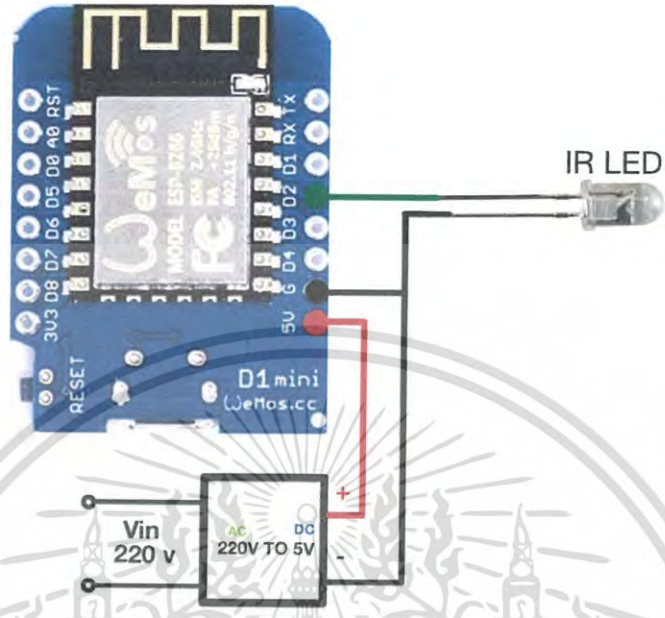


รูปที่ 3.18 กลอนประตูไฟฟ้าสั่งงานด้วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.5 การติดตั้งอุปกรณ์ตัวส่งสัญญาณอินฟราเรดด้วยเสียง

1. ต่อวงจรตัวส่งสัญญาณอินฟราเรดด้วยเสียงดังรูปที่ 3.19



รูปที่ 3.19 จงจรตัวส่งสัญญาณอินฟราเรดด้วยเสียง

2. นำ Code ที่เขียนเพื่อควบคุมอุปกรณ์ส่งสัญญาณอินฟราเรดด้วยเสียงลงในบอร์ด Arduino Wemos D1 Mini

3. ทดสอบการทำงานของอุปกรณ์ส่งสัญญาณอินฟราเรดด้วยเสียง และแก้ไขปัญหา

4. ประกอบอุปกรณ์การทำงานตัวส่งสัญญาณอินฟราเรดด้วยเสียงเข้าด้วยกัน ดังรูปที่ 3.20

รูปที่ 3.20 อุปกรณ์ส่งสัญญาณอินฟราเรดด้วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการวิเคราะห์ข้อมูล

4.1 ผลจากการทดสอบการทำงาน

4.1.1 การเชื่อมต่อไร้สาย

การเชื่อมต่อระหว่างโทรศัพท์มือถือกับอุปกรณ์ไฟฟ้าภายในบ้านผ่านโปรแกรม Homebridge ใน Raspberry Pi สามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านต่างๆ ผ่านแอพลิเคชัน Home ในระบบ iOS หรือ Google Assistant ในระบบ Android ซึ่งสามารถสั่งการด้วยเสียงผ่านโทรศัพท์มือถือ ซึ่งทำงานรวมกับระบบปฏิบัติการในโทรศัพท์มือถืออยู่แล้ว ซึ่งในการทดลองผู้ทำได้ทดสอบในระบบ iOS บนโทรศัพท์มือถือ ดังรูปที่ 4.1 และรูปที่ 4.2

รูปที่ 4.1 แอพพลิเคชัน Home ในระบบ iOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 การสั่งการด้วยเสียงผ่าน Siri ในระบบ iOS

4.1.2 การทำงานของม่านไฟฟ้าสั่งงานด้วยเสียง

ม่านไฟฟ้าสั่งงานด้วยเสียงนั้นสามารถเปิดปิดผ่านแอพพลิเคชัน HOME และสามารถสั่งงานผ่านเสียงผ่าน Siri ได้ ซึ่งม่านไฟฟ้าสั่งงานด้วยเสียงนั้นสามารถปรับระดับความสูงต่ำ ได้ผ่านการกดปุ่มเปิด-ปิดม่านได้ดังรูปที่ 4.3 และม่านนี้สามารถสั่งการผ่านแอพพลิเคชันดังรูปที่ 4.4

รูปที่ 4.3 ปุ่มควบคุมม่านไฟฟ้าสั่งงานด้วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 การปรับระดับม่านผ่านแอพพลิเคชัน

4.1.3 การทำงานของปลั๊กไฟฟ้าสั่งงานด้วยเสียง

ปลั๊กไฟฟ้าสั่งงานด้วยเสียงนั้นทำงานโดยรีเลย์ในการตัดต่อไฟฟ้า ซึ่งรีเลย์นี้ถูกควบคุมโดยบอร์ด Arduino Wemos D1 Mini ซึ่งรับคำสั่งจากการสั่งงานผ่านการกดในโทรศัพท์มือถือในแอพพลิเคชัน Home และสามารถสั่งงานด้วยเสียงผ่านโทรศัพท์มือถือได้เช่นกัน

4.1.4 การทำงานของสวิตช์ไฟฟ้าสั่งงานด้วยเสียง

สวิตช์ไฟฟ้าสั่งงานด้วยเสียงนั้นทำงานโดยรีเลย์ในการตัดต่อไฟฟ้า ซึ่งรีเลย์นี้ถูกควบคุมโดยบอร์ด Arduino Wemos D1 Mini ซึ่งรับคำสั่งจากการสั่งงานผ่านการกดในโทรศัพท์มือถือในแอพพลิเคชัน Home รวมถึงสามารถกดผ่านสวิตช์ 3 ทางปกติได้อีกด้วย และสามารถสั่งงานด้วยเสียงผ่านโทรศัพท์มือถือได้เช่นกัน

4.1.5 การทำงานกลอนประตูไฟฟ้าสั่งงานด้วยเสียง

การทำงานของกลอนประตูไฟฟ้านั้นทำงานด้วยรีเลย์ ซึ่งถูกควบคุมโดยบอร์ด Arduino Wemos D1 Mini ซึ่งรับคำสั่งจากการสั่งงานผ่านการกดในโทรศัพท์มือถือในแอพพลิเคชัน Home และสามารถสั่งงานด้วยเสียงผ่านโทรศัพท์มือถือได้เช่นกัน

กลอนประตูไฟฟ้าสั่งงานด้วยเสียงนี้มีเซนเซอร์แม่เหล็กติดตั้งอยู่ด้วย ซึ่งเมื่อผู้ใช้งานนั้นทำการปลดล็อคผ่านโทรศัพท์มือถือแต่ไม่ได้ทำการเปิดประตูภายใน 5 วินาทีนั้นกลอนประตูไฟฟ้าจะกลับมาล็อคเช่นเดิม และถ้าเปิดประตูทิ้งไว้นานเกินเวลาที่กำหนดจะมีเสียงเตือน โดยเสียงเตือนนั้นจะหยุดเมื่อทำการปิดประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 การทำงานของตัวส่งสัญญาณอินฟราเรดด้วยเสียง

การทำงานของตัวส่งสัญญาณอินฟราเรดด้วยเสียงนั้นทำงานด้วยรีเลย์ ซึ่งถูกควบคุมโดยบอร์ด Arduino Wemos D1 Mini ซึ่งรับคำสั่งจากการสั่งงานผ่านการกดในโทรศัพท์มือถือในแอพพลิเคชัน Home และสามารถสั่งงานด้วยเสียงผ่านโทรศัพท์มือถือได้เช่นกัน โดยการที่จะการสั่งการเครื่องใช้ไฟฟ้าที่รับสัญญาณอินฟราเรดได้นั้น ต้องทำการนำค่าสัญญาณจากรีโมตนั้นมาลงในบอร์ด Arduino Wemos D1 Mini ก่อนถึงจะสามารถสั่งการผ่านโทรศัพท์ได้

4.2 การนำไปใช้งาน

จากการทดลองใช้งานเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียงนั้น ทุกอุปกรณ์สามารถทำงานได้ดี โดยสามารถควบคุมได้ทั้งผ่านแอพพลิเคชัน Home ผ่านการกดสวิตช์ของอุปกรณ์นั้นๆ และสามารถสั่ง การด้วยเสียงผ่านโทรศัพท์มือถือ แต่ปัญหาคือ ตัวส่งสัญญาณอินฟราเรดนั้นส่งได้ระยะไม่ไกล ทำให้ต้องนำตัวส่งสัญญาณอินฟราเรดไว้ใกล้ๆ กับอุปกรณ์ไฟฟ้านั้น ซึ่งเกิดจากหลอดส่งสัญญาณนั้นส่งได้ไม่ไกล และหากระบบอินเทอร์เน็ตล่มนั้นก็ไม่สามารถใช้อุปกรณ์ไฟฟ้าต่างๆ ได้ โดยจัดแสดงอุปกรณ์ทั้งหมดไว้ในบ้านจำลอง ดังรูปที่ 4.5

รูปที่ 4.5 อุปกรณ์ไฟฟ้าสั่งงานด้วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

การทำปริญญานิพนธ์เรื่องเครื่องใช้ไฟฟ้าภายในบ้านสั่งงานด้วยเสียง ต้องมีความรู้หลายด้านประกอบเข้าด้วยกัน ได้แก่ การใช้โปรแกรม Raspbian ลง ไมโครคอนโทรลเลอร์ Rasberry Pi การใช้โปรแกรม Arduino ในการควบคุมอุปกรณ์ไฟฟ้าต่างๆ การใช้โปรแกรม Homebridge และความรู้ด้านอิเล็กทรอนิกส์เบื้องต้น เช่น Relay มอเตอร์ เป็นต้น โดยขั้นตอนแรกของการศึกษาและค้นคว้าข้อมูล มีการทำงานออกเป็น 2 ส่วนหลักๆ คือ ส่วนของฮาร์ดแวร์และส่วนของซอฟต์แวร์ ซึ่งเป็นการศึกษาความรู้ในทางทฤษฎีมาประยุกต์ใช้ในการออกแบบระบบการทำงานในอุปกรณ์ไฟฟ้าต่างๆ สำหรับการติดตั้งในการใช้งานจริง โดยจากการทดลองจะเห็นว่าอุปกรณ์ต่างๆ สามารถทำงานได้ตามต้องการ โดยสามารถสั่งงานด้วยเสียงผ่านโทรศัพท์มือถือบนระบบอินเทอร์เน็ต และสามารถควบคุมโดยตรงได้เช่นกัน

5.2 ปัญหาและอุปสรรค

1. ในช่วงแรกอุปกรณ์อิเล็กทรอนิกส์ภายในตัวควบคุมอุปกณ์มีขนาดไม่เหมาะสม
2. ในช่วงการทำสวิตช์ไฟสามารถทำเปิด-ปิดผ่านโทรศัพท์แต่ไม่สามารถเปิด-ปิด ผ่านสวิตช์ไฟได้
3. ในช่วงแรกของการทำตัวอุปกรณ์ส่งสัญญาณอินฟราเรด ตัวอ่านสัญญาณอินฟราเรดจากรีโมตโดยตรงทำให้ค่าสัญญาณที่ได้มามีค่าไม่เสถียร ทำให้ไม่สามารถเขียนโปรแกรมที่ส่งสัญญาณได้เสถียรมากพอ

5.3 แนวทางแก้ไขปัญหา

1. เลือกเปลี่ยนอุปกรณ์อิเล็กทรอนิกส์ให้มีขนาดเล็กลงกว่าเดิม โดยมีคุณสมบัติการใช้งานที่เหมาะสมกับการใช้งาน อย่างเปลี่ยนจาก ESP8266 ตัวธรรมดาเป็น Wemos D1 Mini ซึ่งสามารถต่อกับ Relay Shield for Wemos d1 mini ซึ่งทำให้ขนาดอุปกรณ์มีขนาดที่เหมาะสมกับการใช้งาน
2. ทำการแก้ไขปัญหาการเปิดปิดสวิตช์ให้สามารถเปิด ปิดได้ผ่านทั้งโทรศัพท์มือถือ และสวิตช์ไฟโดยการแก้ไขโปรแกรมการทำงานใน Arduino Wemos D1 Mini
3. ในการอ่านค่าสัญญาณอินฟราเรดควรใช้โปรแกรมในการอ่านค่า ที่มีความเสถียรทำให้ค่าที่ออกมานั้นมีความเสถียร ทำให้สามารถเขียนโปรแกรมส่งสัญญาณอินฟราเรดที่มีความเสถียรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางการพัฒนา

1. พัฒนาให้อุปกรณ์ไฟฟ้าต่างๆ สามารถทำงานได้ตามปกติ หากระบบอินเทอร์เน็ตไร้สายมีปัญหา

2. พัฒนาโปรแกรมให้สามารถตรวจสอบสถานะอุปกรณ์ไฟฟ้าต่างๆ ได้ว่ามีสถานะเป็นเช่นไร

3. พัฒนาตัวส่งสัญญาณอินฟราเรดให้สามารถส่งสัญญาณได้ระยะไกล และสามารถควบคุมทุกอุปกรณ์ในตัวส่งสัญญาณตัวเดียว

4. พัฒนาให้อุปกรณ์ไฟฟ้าสั่งงานด้วยเสียงสามารถแสดงสถานะผ่านแอพพลิเคชันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Homebridge Configuration

```
{
    "bridge": {
    "name": "Homebridge",
    "username": "CC:22:3D:E3:CE:30",
   "port": 51826,
    "pin": "031-45-154"
},
"accessories": [{
    "accessory": "EspLock",
    "name": "Door Lock",
    "url":  "http://192.168.43.75/",
    "lock-id": "1"
},
{
   "accessory": "Http",
   "name": "Outlet",
   "on_url": "http://192.168.43.49/gpio/1",
   "off_url": "http://192.168.43.49/gpio/0",
   "http_method": "GET"
},
{
   "accessory": "Http",
   "name": "IR",
   "on_url": "http://192.168.43.235/gpio/1",
   "off_url": "http://192.168.43.235/gpio/0",
   "http_method": "GET"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  },
  {

  "accessory": "Http",
      "name": "Switch",
      "on_url": "http://192.168.43.211/gpio/1",
      "off_url": "http://192.168.43.211/gpio/0",
      "http_method": "GET"
  }],
  "platforms": [{
       "platform": "mqtt",
       "name": "mqtt",
       "url": "mqtt://127.0.0.1",
       "port": "1883",
       "topic_type": "multiple",
       "topic_prefix": "homebridge",
       "qos": 1
     },
     {
       "platform": "Alexa",
       "name": "Alexa",
       "username": "******@hotmail.com",
       "password": "**********"
     }]
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมสั่งงานควบคุมการทำงานของปลั๊กไฟฟ้าสั่งงานด้วยเสียง

```
#include <ESP8266WiFi.h>

const char* ssid = "Fluke's iPhone";
const char* password = "00000000";

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(5, OUTPUT);
  digitalWrite(5, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

// Static IP
  IPAddress ip(192, 168, 43, 49);
  IPAddress gateway(192, 162, 43, 49);
  IPAddress subnet(255, 255, 255, 0);
  IPAddress dns(192, 168, 43, 49);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  // Static IP Setup
  WiFi.config(ip, gateway, dns);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 while(!client.available()){
   delay(1);
 }

 // Read the first line of the request
 String req = client.readStringUntil('\r');
 Serial.println(req);
 client.flush();

 // Match the request
 int val;
 if (req.indexOf("/gpio/0") != -1)
   val = 0;
 else if (req.indexOf("/gpio/1") != -1)
   val = 1;
 else {
   Serial.println("invalid request");
   client.stop();
   return;
 }

 // Set GPIO2 according to the request
 digitalWrite(5, val);
  digitalWrite(LED_BUILTIN, val);   // Turn the LED on (Note that LOW is the
voltage level
 client.flush();

 // Prepare the response
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nGPIO is now ";
s += (val)?"high":"low";
s += "</html>\n";

// Send the response to the client
client.print(s);
delay(10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมสั่งงานควบคุมการทำงานของสวิตช์ไฟฟ้าสั่งงานด้วยเสียง

```
#include <ESP8266WiFi.h>
const char* ssid = "Fluke's iPhone";
const char* password = "00000000";
#define button D6
#define pressed LOW

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(button,INPUT_PULLUP);
  pinMode(5, OUTPUT);
  digitalWrite(5, 0);

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

// Static IP
  IPAddress ip(192, 168, 43, 211);
  IPAddress gateway(192, 168, 43, 211);
  IPAddress subnet(255, 255, 255, 0);
  IPAddress dns(192, 168, 43, 211);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  // Static IP Setup
  WiFi.config(ip, gateway, dns);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop() {
  bool ReadSwitch = digitalRead(button);
  if(ReadSwitch == pressed)
  {
    if(digitalRead(5) == HIGH){
      digitalWrite(5, 0);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else{
    digitalWrite(5, 1);
  }
  return;
}
// Check if a client has connected
WiFiClient client = server.available();
if (!client) {
  return;
}

// Wait until the client sends some data
Serial.println("new client");
while(!client.available()){
  delay(1);
}

// Read the first line of the request
String req = client.readStringUntil('\r');
Serial.println(req);
client.flush();

// Match the request
int val;
if (req.indexOf("/gpio/0") != -1)
  val = 0;
else if (req.indexOf("/gpio/1") != -1)
  val = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  // Set GPIO2 according to the request
  digitalWrite(5, val);
   digitalWrite(LED_BUILTIN, val);   // Turn the LED on (Note that LOW is the
voltage level
  client.flush();

  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nGPIO is now ";
  s += (val)?"high":"low";
  s += "</html>\n";

  // Send the response to the client
  client.print(s);
  delay(10);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

โปรแกรมสั่งงานควบคุมการทำงานกลอนประตูไฟฟ้าสั่งงานด้วยเสียง

```
#include <ESP8266WebServer.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266mDNS.h>
#define RELAY_PIN 5

const char* ssid        = "Fluke's iPhone";
const char* password    = "00000000";
const long  lockTimeout = 5000;

ESP8266WebServer server(80);
long unlockedTime = 0;

void handleQuery() {
   char rsp[255];
   sprintf(rsp, "{\"state\":\"%s\",\"statusCode\":200,\"battery\":100}",
digitalRead(RELAY_PIN) ? "unlocked" : "locked");

   server.send(200, "text/plain", rsp);
}

void handleLock() {
   String state = server.arg("state");
   if (state == "locked") {
      digitalWrite(RELAY_PIN, 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    } else {
      // There's a bug for ESP8266 Arduino core version 2.3.0 that it can't get
  param state
      digitalWrite(RELAY_PIN, 1);
      // Lock again after 3 seconds
      unlockedTime = millis();
    }
    server.send(200, "text/plain", "{\"battery\":100,\"statusCode\":200}");
}

void setup() {
    Serial.begin(115200);
    pinMode(RELAY_PIN, OUTPUT);
    WiFi.mode(WiFi_STA);
   // WiFi.begin(ssid, password);

// Static IP
  IPAddress ip(192, 168, 43, 75);
  IPAddress gateway(192, 168, 43, 75);
  IPAddress subnet(255, 255, 255, 0);
  IPAddress dns(192, 168, 43, 75);

  // Static IP Setup
  WiFi.config(ip, gateway, subnet);
  WiFi.begin(ssid, password);

    Serial.println("");
    while (WiFi.status() != WL_CONNECTED) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());

    server.on("/", HTTP_GET, handleQuery);      // Query lock status
    server.on("/", HTTP_POST, handleLock);      // Set lock status
    server.begin();

}

void loop() {
  // Serial.print("Open  : ");
   //Serial.println(digitalRead(RELAY_PIN));
   server.handleClient();

   // Restore lock status
   if ((unlockedTime > 0) && (millis() - unlockedTime > lockTimeout)) {
      unlockedTime = 0;
      digitalWrite(RELAY_PIN, 0);
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

โปรแกรมสั่งงานควบคุมการทำงานของตัวส่งสัญญาณอินฟราเรดด้วยเสียง

```
#include <ESP8266WiFi.h>
#include <IRremoteESP8266.h>
#include <Arduino.h>
#include <IRsend.h>

const char* ssid = "Fluke's iPhone";
const char* password = "00000000";
const uint16_t kIrLed = 4;
IRsend irsend(kIrLed);
// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
 Serial.begin(115200);
 delay(10);
irsend.begin();
 //Serial.begin(9600);

 // Connect to WiFi network
 Serial.println();
 Serial.println();
 Serial.print("Connecting to ");
 Serial.println(ssid);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  // Static IP
   IPAddress ip(192, 168, 43, 235);
   IPAddress gateway(192, 168, 43, 235);
   IPAddress subnet(255, 255, 255, 0);
   IPAddress dns(192, 168, 43, 235);

   // Static IP Setup
   WiFi.config(ip, gateway, subnet);
   WiFi.begin(ssid, password);
   while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
   }
   Serial.println("");
   Serial.println("WiFi connected");

   // Start the server
   server.begin();
   Serial.println("Server started");

   // Print the IP address
   Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(req);
  client.flush();

  // Match the request
  int val;
  if (req.indexOf("/gpio/0") != -1){
    Serial.println("send ir signal");
    irsend.sendNEC(0x20DF10EF, 32);}
  else if (req.indexOf("/gpio/1") != -1){
    Serial.println("send ir signal");
    irsend.sendNEC(0x20DF10EF, 32);}
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  // Set GPIO2 according to the request
   digitalWrite(LED_BUILTIN, val);
  client.flush();

  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nGPIO is now ";
  s += (val)?"high":"low";
  s += "</html>\n";

  // Send the response to the client
  client.print(s);
  delay(10);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

โปรแกรมอ่าน Code Infrared Remote

```
#include <IRremote.h>

int recvPin = 11;
IRrecv irrecv(recvPin);

void  setup ( )
{
  Serial.begin(9600);   // Status message will be sent to PC at 9600 baud
  irrecv.enableIRIn();  // Start the receiver
}
void  ircode (decode_results *results)
{
  // Panasonic has an Address
  if (results->decode_type == PANASONIC) {
    Serial.print(results->address, HEX);
    Serial.print(":");
  }
  Serial.print(results->value, HEX);
}
void  encoding (decode_results *results)
{
  switch (results->decode_type) {
    default:
    case UNKNOWN:      Serial.print("UNKNOWN");       break ;
    case NEC:          Serial.print("NEC");           break ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case SONY:      Serial.print("SONY");      break ;
    case RC5:       Serial.print("RC5");       break ;
    case RC6:       Serial.print("RC6");       break ;
    case DISH:      Serial.print("DISH");      break ;
    case SHARP:     Serial.print("SHARP");     break ;
    case JVC:       Serial.print("JVC");       break ;
    case SANYO:     Serial.print("SANYO");     break ;
    case MITSUBISHI: Serial.print("MITSUBISHI"); break ;
    case SAMSUNG:   Serial.print("SAMSUNG");   break ;
    case LG:        Serial.print("LG");        break ;
    case WHYNTER:   Serial.print("WHYNTER");   break ;
    case AIWA_RC_T501: Serial.print("AIWA_RC_T501"); break ;
    case PANASONIC: Serial.print("PANASONIC"); break ;
    case DENON:     Serial.print("Denon");     break ;
  }
}
void  dumpInfo (decode_results *results)
{
 if (results->overflow) {
   Serial.println("IR code too long. Edit IRremoteInt.h and increase RAWLEN");
   return;
 }
 Serial.print("Encoding  : ");
 encoding(results);
 Serial.println("");
 Serial.print("Code      : ");
 ircode(results);
 Serial.print(" (");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.print(results->bits, DEC);
  Serial.println(" bits)");
}
void  dumpRaw (decode_results *results)
{
 Serial.print("Timing[");
  Serial.print(results->rawlen-1, DEC);
  Serial.println("]: ");

  for (int i = 1;  i < results->rawlen;  i++) {
    unsigned long  x = results->rawbuf[i] * USECPERTICK;
    if (!(i & 1)) {  // even
      Serial.print("-");
      if (x < 1000)  Serial.print(" ") ;
      if (x < 100)   Serial.print(" ") ;
      Serial.print(x, DEC);
    } else {  // odd
      Serial.print("     ");
      Serial.print("+");
      if (x < 1000)  Serial.print(" ") ;
      if (x < 100)   Serial.print(" ") ;
      Serial.print(x, DEC);
      if (i < results->rawlen-1) Serial.print(", "); //',' not needed for last one
    }
    if (!(i % 8))  Serial.println("");
  }
  Serial.println("");                    // Newline
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void  dumpCode (decode_results *results)
{
  Serial.print("unsigned int  ");          // variable type
  Serial.print("rawData[");                // array name
  Serial.print(results->rawlen - 1, DEC);  // array size
  Serial.print("] = {");                   // Start declaration

  for (int i = 1;  i < results->rawlen;  i++) {
    Serial.print(results->rawbuf[i] * USECPERTICK, DEC);
    if ( i < results->rawlen-1 ) Serial.print(","); // ',' not needed on last one
    if (!(i & 1))  Serial.print(" ");
  }

  // End declaration
  Serial.print("};");  //

  // Comment
  Serial.print("  // ");
  encoding(results);
  Serial.print(" ");
  ircode(results);

  // Newline
  Serial.println("");

  // Now dump "known" codes
  if (results->decode_type != UNKNOWN) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // Some protocols have an address
    if (results->decode_type == PANASONIC) {
      Serial.print("unsigned int  addr = 0x");
      Serial.print(results->address, HEX);
      Serial.println(";");
    }
    Serial.print("unsigned int  data = 0x");
    Serial.print(results->value, HEX);
    Serial.println(";");
  }
}
void  loop ( )
{
  decode_results  results;        // Somewhere to store the results

  if (irrecv.decode(&results)) {  // Grab an IR code
    dumpInfo(&results);           // Output the results
    dumpRaw(&results);            // Output the results in RAW format
    dumpCode(&results);           // Output the results as source code
    Serial.println("");           // Blank line between entries
    irrecv.resume();              // Prepare for the next value
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้