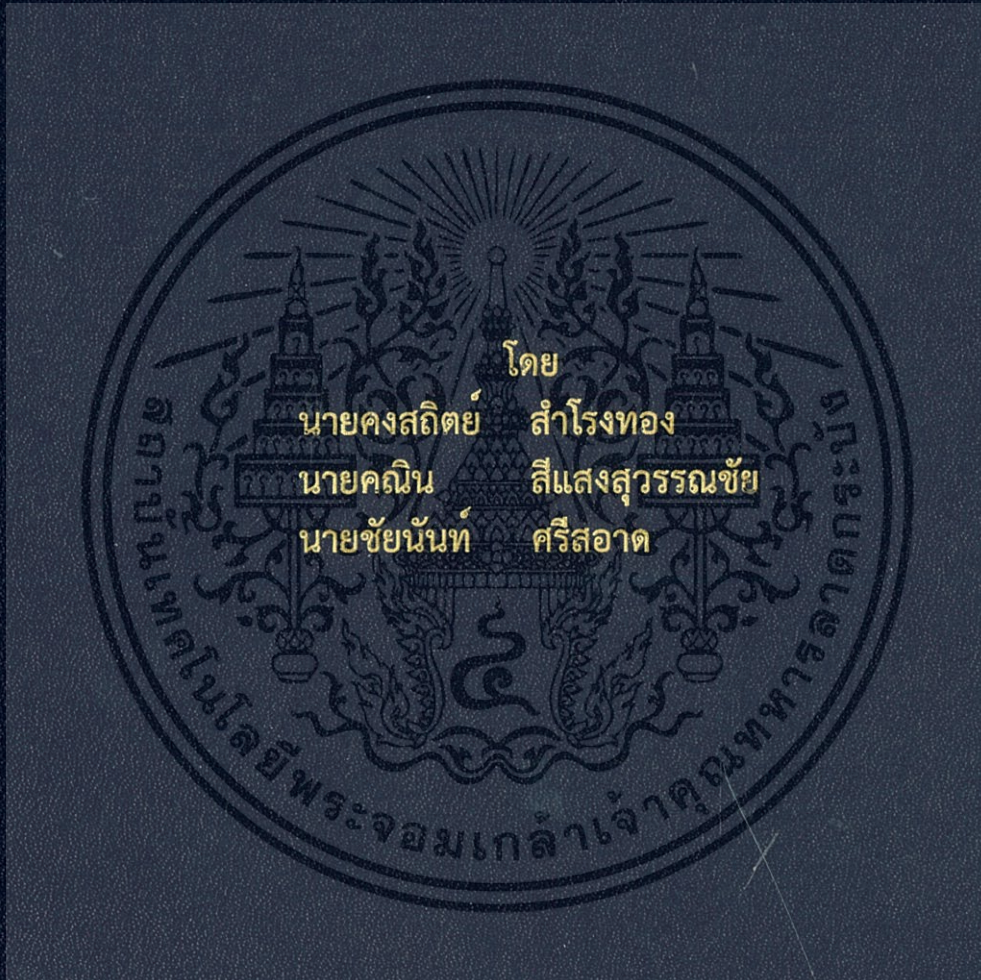


ระบบควบคุมเครื่องปรับอากาศอัจฉริยะผ่านระบบอินเทอร์เน็ต
SMART AIR CONDITIONER CONTROL VIA INTERNET



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ระบบควบคุมเครื่องปรับอากาศอัจฉริยะผ่านระบบอินเทอร์เน็ต
SMART AIR CONDITIONER CONTROL VIA INETERNET



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

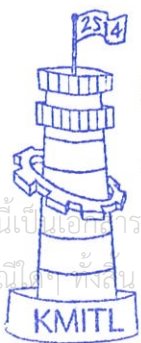
ระบบควบคุมเครื่องปรับอากาศอัจฉริยะผ่านระบบอินเทอร์เน็ต
SMART AIR CONDITIONER CONTROL VIA INETERNET

โดย

นายคงสถิตย์	สำโรงทอง	56010112
นายคณิน	สีแสงสุวรรณชัย	56010118
นายชัยนันท์	ศรีสอาด	56010292

อาจารย์ที่ปรึกษา
ศ.ดร. พรชัย ทรัพย์นินิ
รศ.ดร. พิสิฐ บุญศรีเมือง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559



ผ่านการตรวจรูปเล่มแล้ว

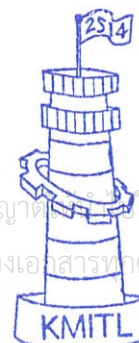
(P.S.)

อาจารย์ที่ปรึกษา

05/11/60

วิศวกรรมโทรคมนาคม

Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

([Signature])

กรรมการผู้ตรวจชิ้นงาน

.....

วิศวกรรมโทรคมนาคม

Telecommunications Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารหากครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมเครื่องปรับอากาศอัจฉริยะผ่านระบบอินเทอร์เน็ต

SMART AIR CONDITIONER CONTROL VIA INTERNET

ผู้จัดทำ

- | | |
|---------------------------|----------|
| 1. นายคงสถิตย์ สำโรงทอง | 56010112 |
| 2. นายคณิน สีแสงสุวรรณชัย | 56010118 |
| 3. นายชัยนันท์ ศรีสอาด | 56010292 |



(ศ.ดร. พรชัย ทรัพย์นิธิ)

อาจารย์ที่ปรึกษา



(รศ.ดร. พิสิฐ บุญศรีเมือง)

อาจารย์ที่ปรึกษาร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จได้ด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษาหลัก ศาสตราจารย์ ดร. พรชัย ทรัพย์นิธิ อาจารย์ที่ปรึกษาร่วม รศ. ดร. พิสิฐ บุญศรีเมือง และ ดร. เวธิต ภาคย์พิสุทธิ์ ที่เสียสละเวลาให้คำปรึกษาและคำแนะนำความรู้ต่างๆ ทั้งทางด้านทฤษฎี และด้านปฏิบัติ การตรวจทานแก้ไขจุดบกพร่องต่างๆ ของโครงการ งบประมาณส่วนเกินจากค่าใช้จ่ายในการจัดซื้อวัสดุ อุปกรณ์ ตลอดจนอุปกรณ์ที่ใช้ในการทำงานโครงการและสถานที่ทั้งในเวลา และนอกเวลาราชการ รวมไปถึงนักศึกษาปริญญาโทและปริญญาเอกห้อง T-113 ภาควิชาวิศวกรรมโทรคมนาคม ที่คอยให้คำแนะนำและให้คำปรึกษาแก้ไขปัญหาต่างๆ ที่เกิดขึ้นในการทำโครงการมาเป็นอย่างดีมาโดยตลอด ทางคณะผู้จัดทำจึงกราบขอบพระคุณมา ณ โอกาสนี้

นายคงสถิตย์ สำโรงทอง
นายคณิน สีแสงสุวรรณชัย
นายชัยนันท์ ศรีสอาด
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมเครื่องปรับอากาศอัจฉริยะผ่านระบบอินเทอร์เน็ต
SMART AIR CONDITIONER CONTROL VIA INTERNET

โดย นายคงสถิตย์ สำโรงทอง 56010112
นายคณิน สีแสงสุวรรณชัย 56010118
นายชัยนันท์ ศรีสอาด 56010292

อาจารย์ที่ปรึกษา ศ. ดร. พรชัย ทรัพย์นิธิ
อาจารย์ที่ปรึกษาร่วม รศ.ดร. พิสิฐ บุญศรีเมือง

บทคัดย่อ

ในการควบคุมเครื่องปรับอากาศในปัจจุบันผู้ใช้งานจะเป็นผู้ควบคุมผ่านรีโมทคอนโทรลเลอร์ โดยการควบคุมนี้ยังอยู่ภายในบริเวณเครื่องปรับอากาศ ทำให้ผู้ใช้งานต้องรอให้อุณหภูมิลดลงตามที่กำหนด ซึ่งยังมีระยะเวลานาน ดังนั้นผู้จัดทำจึงออกแบบโครงการควบคุมเครื่องปรับอากาศอัจฉริยะผ่านระบบอินเทอร์เน็ต ที่สามารถสั่งการควบคุมเครื่องปรับอากาศจากแอนดรอยด์แอปพลิเคชันและเว็บแอปพลิเคชันจากภายนอกที่พกพาได้เพื่อให้สะดวกสบาย โดยภายในวงจรควบคุมนั้นทางผู้จัดทำเลือกใช้ RASPBERRY PI และหลอดไฟโอดเปล่งแสงชนิดอินฟราเรดเพื่อส่งสัญญาณไปยังเครื่องปรับอากาศ และมีเซนเซอร์วัดอุณหภูมิ DHT22 เป็นอุปกรณ์วัดอุณหภูมิภายในห้องเพื่อตรวจสอบว่าอุณหภูมิตรงตามที่ผู้ใช้งานสั่งการจริง การทำงานของระบบจะเริ่มต้นเมื่อผู้ใช้งานสั่งการผ่านทางหน้าแอนดรอยด์แอปพลิเคชันหรือเว็บแอปพลิเคชัน แอนดรอยด์แอปพลิเคชันหรือเว็บแอปพลิเคชันจะส่งค่าเข้าไปเก็บในระบบเซิร์ฟเวอร์ฐานข้อมูลภายใน RASPBERRY PI จากนั้นจะส่งสัญญาณควบคุมไปยังเครื่องปรับอากาศตามที่ใช้กำหนด ยิ่งไปกว่านั้นเซนเซอร์วัดอุณหภูมิ DHT22 จะวัดอุณหภูมิและความชื้น และส่งค่าที่ได้ไปยังระบบเซิร์ฟเวอร์ฐานข้อมูล จากนั้นแอนดรอยด์แอปพลิเคชันหรือเว็บแอปพลิเคชันจะนำค่าที่อยู่ในระบบเซิร์ฟเวอร์ฐานข้อมูลจากเซนเซอร์ DHT22 มาแสดงอุณหภูมิ และความชื้นห้องบนแอนดรอยด์แอปพลิเคชันหรือเว็บแอปพลิเคชัน นอกจากนี้ RASPBERRY PI จะนำข้อมูลที่อยู่ในฐานข้อมูลมาวิเคราะห์เพื่อจัดทำระบบเรียนรู้อัจฉริยะ โดยสามารถทำการ เปิด-ปิดเครื่องปรับอากาศแบบอัตโนมัติ วัดค่าพลังงานที่ใช้กับเครื่องปรับอากาศ มีการแจ้งเตือนและแนะนำอุณหภูมิที่ควรจะเป็นให้ผู้ใช้งาน ซึ่งข้อมูลที่วิเคราะห์ประกอบด้วยพฤติกรรมการใช้งานเครื่องปรับอากาศ, อุณหภูมิ และความชื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Presently, the air conditioner has to be controlled nearby user so it takes a duration of time to decrease temperature. This project is developed the smart air conditioner control system via internet by using the smartphone from everywhere. The Raspberry Pi and Light Emitting Diode are used to transmit the IR bit pattern to control the air conditioner. In addition, the DHT22 sensor is used to measure the temperature and humidity to ensure that the air conditioner receives the user's command. This system starts with the user selecting temperature in the android or web application. Then, it will send the user's command to the database on the Raspberry Pi and the Raspberry Pi will fetch the user's command and send the corresponding IR bit pattern to control the Air-conditioner. Moreover, the temperature and humidity from DHT22 sensor will be sent to the database and displayed on the android and web application. Finally, Raspberry Pi will bring user's data from the database to analyze and create a smart system which on-off the air conditioner automatically, approximate power consumption of the air conditioner and appropriate temperature suggestion.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	IX
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 Raspberry PI 3 Model B	3
2.2 ระบบเครือข่ายอินเทอร์เน็ต (Internet Network System)	5
2.3 การเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย	7
2.4 ฐานข้อมูล	11
2.5 การส่งและรับสัญญาณ Infrared	12
2.6 หลักการมอดูเลตสัญญาณอินฟราเรด	13
2.7 วงจรขยายสัญญาณอินฟราเรด	15
2.8 โครงข่ายประสาทเทียม	16
2.9 แอนดรอยด์แอปพลิเคชัน (Application Android)	19
2.10 เว็บแอปพลิเคชัน (Web Application)	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 3	
การออกแบบและการจัดทำโครงการ	23
3.1 หลักการทำงานของโครงการ	23
3.2 การออกแบบ	30
3.3 เครื่องมือที่ใช้ในการทดลอง	33
3.4 การจัดเก็บผลการทดลอง	42
บทที่ 4	
ผลการทดลอง	50
4.1 การวัดระยะทางในการรับและส่งข้อมูล	50
4.2 การคำนวณความผิดพลาดในการส่งการจากหน้าแอนดรอยด์แอปพลิเคชัน	52
4.3 การคำนวณความผิดพลาดในการส่งการจากหน้าเว็บแอปพลิเคชัน	53
4.4 การวัดค่าอุณหภูมิห้องโดยใช้เซนเซอร์วัดอุณหภูมิ DHT22	54
4.5 การวัดความผิดพลาดของการเปิดเครื่องปรับอากาศตามระยะทางโดยใช้ GPS	56
4.6 การเก็บปิดแพทเทิร์นจากรีโมทคอนโทรลยี่ห้อ SHARP	57
4.7 ทดสอบประสิทธิภาพโมเดลโครงข่ายประสาทเทียม	64
4.8 การประมาณระยะเวลาที่เครื่องปรับอากาศใช้พลังงาน	66
บทที่ 5	
สรุปผลและข้อเสนอแนะ	68
5.1 สรุปผล	68
5.2 ข้อเสนอแนะ	68
บรรณานุกรม	69
ภาคผนวก	
Code สั่งการ Raspberry PI 3 Model B	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	Raspberry PI 3 Model B	3
2.2	แผนภาพแสดงส่วนประกอบของ Raspberry Pi3 Model B	4
2.3	แผนภาพแสดง PIN OUTPUT ของ Raspberry Pi3 Model B	5
2.4	แผนผังแสดงวงจรภายใน Infrared Sensor	13
2.5	การมอดูเลตที่ฝั่งส่งสัญญาณอินฟราเรด	14
2.6	ลักษณะของสัญญาณข้อมูลที่เกิดขึ้นในกระบวนการส่งและรับข้อมูล	15
2.7	วงจรขยายสัญญาณอินฟราเรด	16
2.8	ข่ายงานประสาทเทียมที่มีการเชื่อมต่อกันผ่านกลุ่มโหนด	17
2.9	เพอร์เซ็ปตรอนหลายชั้น (Multi-Layer Perceptron)	18
3.1	แผนภาพบล็อกของโครงการ	23
3.2	แผนผังการทำงานการทำงานส่วนเว็บแอปพลิเคชัน	25
3.3	แผนผังการทำงานหลักของ Raspberry PI	26
3.4	ส่วนวิเคราะห์สถานการณ์ทำงานของเครื่องปรับอากาศโดยใช้โครงข่ายประสาทเทียม	28
3.5	ส่วนแนะนำอุณหภูมิ	29
3.6	วงจรควบคุมเครื่องปรับอากาศ	30
3.7	ฐานข้อมูลเก็บอุณหภูมิที่ผู้ใช้เลือกผ่านทางเว็บแอปพลิเคชัน	31
3.8	ฐานข้อมูลเก็บข้อมูลอุณหภูมิที่วัดได้โดยใช้ DHT22	32
3.9	ฐานข้อมูลเก็บสถานะของเครื่องปรับอากาศที่วิเคราะห์จากโครงข่ายประสาทเทียม	32
3.10	Raspberry PI 3 Model B	33
3.11	เซนเซอร์อินฟราเรด TSOP4838	34
3.12	เซนเซอร์วัดอุณหภูมิ DHT22	34
3.13	ทรานซิสเตอร์ชนิด PNP 2N2222	35
3.14	ไดโอดอินฟราเรด	35
3.15	หน้าแรกของแอนดรอยด์แอปพลิเคชันที่ใช้ในการเลือกยี่ห้อเครื่องปรับอากาศ	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
3.16	สถานะการทำงานเมื่อเลือกอุณหภูมิ 16 °C	37
3.17	สถานะการทำงานเมื่อเลือกโหมด Auto	37
3.18	สถานะการทำงานเมื่อเลือกระดับการแกว่งพัดลมเป็น Auto	38
3.19	สถานะการทำงานเมื่อเลือกระดับความแรงพัดลมเป็นระดับ 1	38
3.20	หน้าเว็บแอปพลิเคชัน	39
3.21	สถานการเลือกอุณหภูมิของหน้าเว็บแอปพลิเคชัน	40
3.22	บรรจุภัณฑ์วงจรสั่งงานเครื่องปรับอากาศ	41
3.23	การตั้งค่าการ Forwarding Port ผ่าน Router ยี่ห้อ Huawei รุ่น HG658 V2	41
3.24	การตั้งค่า DDNS ผ่าน Router ยี่ห้อ Huawei รุ่น HG658 V2	42
3.25	ห้องที่มีแดด	43
3.26	ห้องที่สว่าง	43
3.27	ห้องที่สว่างน้อย	44
3.28	หน้าแอนดรอยด์แอปพลิเคชันแสดงสถานะการใช้งานต่างๆ	45
3.29	หน้าเว็บแอปพลิเคชันแสดงสถานะการใช้งานต่างๆ	45
3.30	แอนดรอยด์แอปพลิเคชัน เมื่ออุณหภูมิที่เลือกมีค่าเท่ากับอุณหภูมิที่วัดได้จาก DHT22	46
3.31	แอนดรอยด์แอปพลิเคชัน เมื่ออุณหภูมิที่เลือกมีค่าเท่ากับอุณหภูมิที่วัดได้จาก DHT22	46
3.32	โมเดลโครงข่ายประสาทเทียมเพื่อวิเคราะห์สถานะการทำงานของเครื่องปรับอากาศ	47
4.1	ค่าอุณหภูมิปัจจุบันในห้องและค่าอุณหภูมิจากการสั่งการเมื่อเริ่มต้นสั่งการ	55
4.2	ค่าอุณหภูมิปัจจุบันในห้องเท่ากับค่าอุณหภูมิจากการสั่งการหลังจากผ่านไป 40 นาที	55
4.3	กราฟลักษณะบิตสัญญาณที่โหมด ON	57
4.4	กราฟลักษณะบิตสัญญาณที่โหมด OFF	57
4.5	กราฟลักษณะบิตสัญญาณที่อุณหภูมิ 16 °C	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
4.6	กราฟลักษณะบิตสัญญาณที่อุณหภูมิ 25 °C	58
4.7	กราฟลักษณะบิตสัญญาณที่โหมด ON	59
4.8	กราฟลักษณะบิตสัญญาณที่โหมด OFF	59
4.9	กราฟลักษณะบิตสัญญาณที่โหมด 16 °C	59
4.10	กราฟลักษณะบิตสัญญาณที่โหมด 25 °C	60
4.11	เฟรมบิตข้อมูลสัญญาณรีโมทเครื่องปรับอากาศยี่ห้อ SHARP	64
4.12	ค่า Cost ที่ได้จากการเรียนรู้ Weight และ Bias	65
4.13	ค่า Accuracy ที่ได้จากการเรียนรู้ Weight และ Bias	65
4.14	เวลาเมื่อเริ่มตรวจจับสถานะเริ่มและสถานะหยุดการทำงานของเครื่องปรับอากาศ	66
4.15	จำนวนชั่วโมงที่ใช้งานเครื่องปรับอากาศ	66
4.16	ประมาณอัตราการใช้งานเครื่องปรับอากาศ	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	มาตรฐาน WI-FI ประเภทต่างๆ	8
4.1	ระยะทางในการรับและส่งข้อมูลในห้องที่มีดทึบ	50
4.2	ระยะทางในการรับและส่งข้อมูลในห้องที่สว่างน้อย	51
4.3	ระยะทางในการรับและส่งข้อมูลในห้องที่สว่าง	51
4.4	การส่งการผ่านแอนดรอยด์แอปพลิเคชัน	52
4.5	การส่งการผ่านเว็บแอปพลิเคชัน	53
4.6	การเปิดของเครื่องปรับอากาศตามระยะทางโดยใช้ GPS	56
4.7	ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0 หรือ 1 ตั้งแต่บิตที่ 1 ถึง 15	61
4.8	ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0 หรือ 1 ตั้งแต่บิตที่ 16 ถึง 32	61
4.9	ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0 หรือ 1 ตั้งแต่บิตที่ 33 ถึง 50	62
4.10	ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0 หรือ 1 ตั้งแต่บิตที่ 51 ถึง 68	62
4.11	ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0 หรือ 1 ตั้งแต่บิตที่ 69 ถึง 86	63
4.12	ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0 หรือ 1 ตั้งแต่บิตที่ 87 ถึง 106	63
4.13	ประมาณระยะเวลาที่เครื่องปรับอากาศใช้พลังงานในรูปสัญลักษณ์	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันประเทศไทยมีอุณหภูมิที่สูงขึ้นส่งผลให้เครื่องปรับอากาศมีความสำคัญกับชีวิตประจำวันมากขึ้น ด้วยเหตุผลดังกล่าว ทางคณะผู้จัดทำจึงจัดทำโครงการ ควบคุมเครื่องปรับอากาศผ่านระบบอินเทอร์เน็ต โดยมีแรงบันดาลใจจากผู้ใช้งานที่ต้องการเปิดเครื่องปรับอากาศก่อนที่จะกลับมาอยู่ที่พักอาศัย และการใช้เครื่องปรับอากาศจะต้องถูกสั่งการด้วยรีโมทคอนโทรลทำให้ต้องเสียเวลารอให้อุณหภูมิในห้องเท่ากับที่เราสั่งการ ซึ่งใช้เวลาพอสมควร แต่ถ้าหากผู้ใช้งานควบคุมโดยสั่งการผ่านเว็บแอปพลิเคชัน จะทำให้ปรับอุณหภูมิในห้องเท่ากับที่เราสั่งการก่อนที่จะกลับมาถึงที่พักอาศัยได้โดยที่ผู้ใช้งานไม่จำเป็นต้องรอให้อุณหภูมิในห้องเท่ากับที่เราสั่งการ

1.2 วัตถุประสงค์

- 1) ศึกษาการออกแบบแอปพลิเคชันแอนดรอยด์ พัฒนาเว็บแอปพลิเคชันแอนดรอยด์เพื่อสั่งการควบคุมเครื่องปรับอากาศ และเพื่อความสะดวกสบายในการควบคุมเครื่องปรับอากาศผ่านระบบอินเทอร์เน็ตแทนการสั่งการจากรีโมทคอนโทรล
- 2) เพื่อศึกษาโครงสร้างการทำงานของการควบคุมอุปกรณ์ผ่านระบบอินเทอร์เน็ตโดยใช้ Raspberry Pi ในการควบคุมอุปกรณ์อื่น ๆ ผ่านระบบอินเทอร์เน็ต

1.3 ขอบเขตของโครงการ

ขอบเขตโครงการตลอดปีการศึกษาที่ 1 ประกอบไปด้วยการตรวจสอบสถานะเครื่องปรับอากาศ การวัดอุณหภูมิและความชื้นโดยใช้เซนเซอร์ คำสั่งควบคุมเครื่องปรับอากาศพื้นฐานซึ่งประกอบไปด้วยการเปิด-ปิด การเลือกอุณหภูมิโดยสามารถควบคุมเครื่องปรับอากาศที่นิยมใช้ในปัจจุบันได้เช่น Daikin, Sharp และ Panasonic ซึ่งใช้ Raspberry Pi สั่งการฟังก์ชันที่ได้กล่าวมา โดยผู้ใช้งานสามารถควบคุม Raspberry Pi ผ่านแอปพลิเคชันในระบบปฏิบัติการแอนดรอยด์ นอกจากนี้ยังมีฟังก์ชันที่ผู้ใช้งานสามารถตัดลอคสัญญาณรีโมทปรับอากาศยี่ห้ออื่นๆ ที่ไม่ได้กล่าวมาข้างต้น หรือแม้กระทั่งอุปกรณ์อื่นๆ ที่สามารถควบคุมได้ผ่านสัญญาณอินฟราเรด (Infrared) และสามารถสั่งการได้เหมือนรีโมทตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตโครงการตลอดปีการศึกษาที่ 2 ประกอบไปด้วย การทำระบบเรียนรู้อัจฉริยะของเครื่องปรับอากาศ โดยระบบจะสามารถวิเคราะห์พฤติกรรมการใช้งานเครื่องปรับอากาศ และทำการเปิด-ปิดเครื่องปรับอากาศแบบอัตโนมัติตามระยะทางของผู้ใช้งานกับที่อยู่อาศัย มีการแจ้งเตือนและแนะนำอุณหภูมิที่ควรจะเป็นให้แก่ผู้ใช้งาน ซึ่งสามารถควบคุมผ่านแอปพลิเคชันแอนดรอยด์และหน้าเว็บแอปพลิเคชันได้



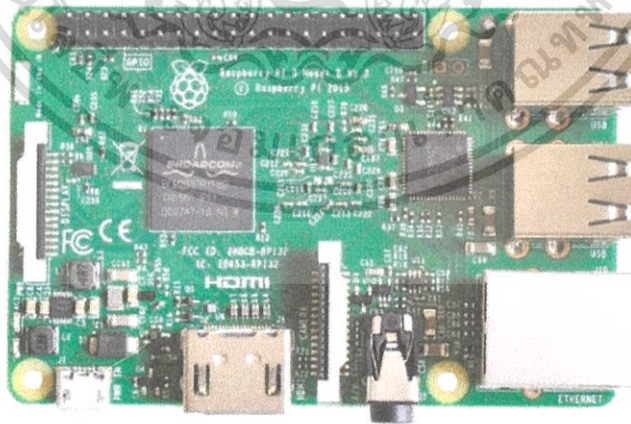
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Raspberry Pi 3 Model B

Raspberry Pi3 Model B เป็นแพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโครงงานอินเทอร์เน็ตในทุกสิ่งหรือ Internet of Things (IoT) Raspberry Pi 3 Model B คือบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงานด้านการวิเคราะห์ข้อมูลตัวเลขหรือข้อความ ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย เช่น GPIO22 ของบอร์ด Raspberry Pi 3 Model B ต่อกับ อินฟราเรด LED (IR LED) ใช้เป็นอุปกรณ์ในการส่งสัญญาณ ดังรูปที่ 2.1

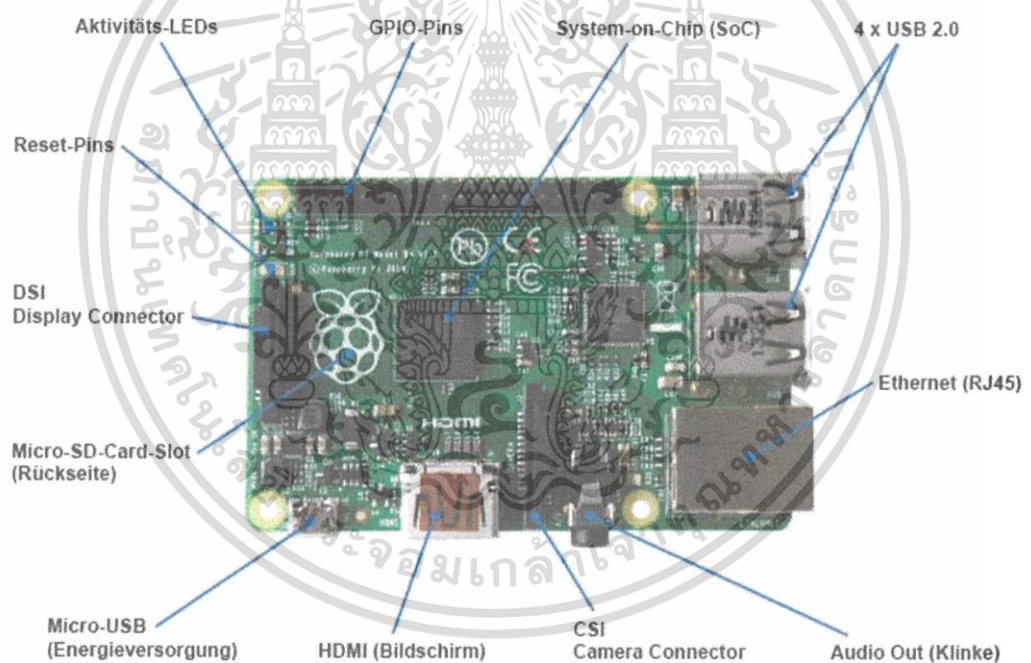


รูปที่ 2.1 Raspberry Pi3 Model B [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 คุณสมบัติของ Raspberry Pi3 Model B

บอร์ด Raspberry Pi 3 Model B ใช้ชิพ Broadcom BCM2837 64-bit Quad-Core ARM Cortex-A53 ARMv8 ความเร็ว 1.2 GHz มีหน่วยความจำชนิด LPDDR 2 SDRAM ขนาด 1 GB ชิป Broadcom BCM43438 เป็น Wi-Fi 802.11 b/g/n และ Bluetooth 4.1 พร้อมสายอากาศแบบ Chip Antenna บนบอร์ดพอร์ตอื่นๆ ประกอบด้วย USB 2.0 Host x 4, Fast Ethernet RJ-45 ความเร็ว 10/100 Mbps x 1, HDMI x 1, Audio and Composite Video 3.5 mm 4 -pole x 1CSI, Camera x 1, Micro SD Card Slot x 1

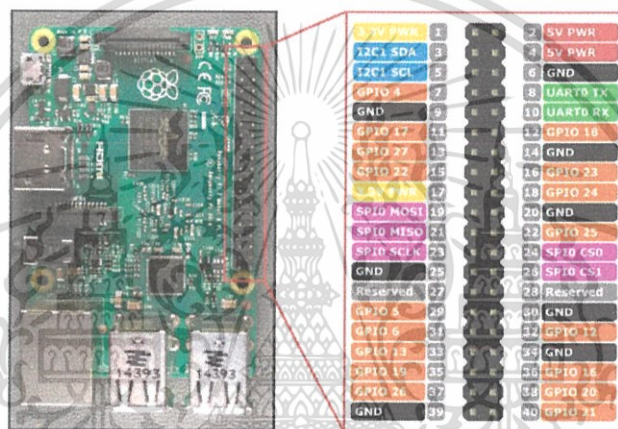


รูปที่ 2.2 แผนภาพแสดงส่วนประกอบของ Raspberry Pi3 Model B [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ช่องทางในการติดต่อสื่อสารของ Raspberry Pi 3 Model B

Raspberry Pi3 Model B มีช่องทางในการติดต่อสื่อสารเพื่อติดต่อ Hardware ภายนอก ซึ่งเรียกว่า General Purpose Input Output (GPIO) มีทั้งหมด 40 ขา โดยออกแบบให้จัดเรียงเป็นแถวเพื่อให้สามารถต่อใช้งานกับอุปกรณ์เชื่อมต่อได้ หรือหากต้องแก้ไขก็ทำเพียงเล็กน้อยโดยรายละเอียดขาสัญญาณแสดงดังรูปที่ 2.3



รูปที่ 2.3 แผนภาพแสดง pin output ของ Raspberry Pi3 Model B [3]

2.2 ระบบเครือข่ายอินเทอร์เน็ต (Internet Network System)

อินเทอร์เน็ต เป็นเครือข่ายคอมพิวเตอร์ที่มีขนาดใหญ่ เกิดจากการเชื่อมต่อเครือข่ายคอมพิวเตอร์จำนวนมากเข้าด้วยกัน ไม่ว่าจะเป็นเครือข่ายขนาดเล็ก เช่นระบบเครือข่ายแบบไกล์ หรือระบบขนาดเครือข่ายขนาดใหญ่ เช่น ระบบเครือข่ายของมินิหรือเมนเฟรมคอมพิวเตอร์ ซึ่งแต่ละเครือข่ายก็จะมีเครื่องคอมพิวเตอร์ที่เป็นเครื่องแม่ข่ายหรือโฮสต์ (Host) ซึ่งมีอยู่หลายชนิดหลายยี่ห้อโดยจะมีการกำหนดข้อตกลงในการสื่อสารที่เรียกว่า โพรโตคอล (Protocol) ขึ้นมาเพื่อให้คอมพิวเตอร์แต่ละชนิดสามารถติดต่อสื่อสารกันได้ โพรโตคอลมาตรฐานที่ใช้ในการสื่อสารบนอินเทอร์เน็ตจะมีชื่อเรียกว่า TCP/IP (Transmission Control Protocol / Internet Protocol) ในส่วนของ Raspberry Pi 3 Model B สามารถเชื่อมต่ออินเทอร์เน็ตผ่านทาง wifi หรือสาย LAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ไอพี แอดเดรส (IP Address)

IP Address เป็นหมายเลขประจำตัวของเครื่องคอมพิวเตอร์ทุกเครื่องที่มีการเชื่อมต่อกับระบบเครือข่ายประกอบด้วยตัวเลข 4 ชุดที่คั่นกันด้วยเครื่องหมายจุด (.) เช่น 202.44.194.6 ตัวเลขในแต่ละชุดจะมีขนาด 8 บิต แต่ละชุดจึงมีค่าตัวเลขได้ตั้งแต่ 0 ถึง 255 เท่านั้น โสสต์คอมพิวเตอร์ทุกเครื่องต้องขอหมายเลข IP นี้จากหน่วยงาน Internet Network Information Center (InterNIC) ขององค์กร Network Solution Incorporated (NSI) สหรัฐอเมริกา ผู้ใช้ธรรมดาทั่วไปสามารถสมัครเป็นสมาชิกกับหน่วยงานที่ให้บริการอินเทอร์เน็ต (Internet Service Provider หรือ ISP) เพื่อรับหมายเลข IP ได้

2.2.2 โพรโตคอลทีซีพี / ไอพี (TCP / IP Protocol)

ในหัวข้อนี้จะกล่าวถึงการทำงานของอินเทอร์เน็ต ซึ่งประกอบไปด้วยเรื่องเกี่ยวกับ TCP / IP, การกำหนดชื่อ และเลข IP อินเทอร์เน็ตนับได้ว่าเป็นเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นๆ เชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์ แต่ปัญหาที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันคือ แต่ละเครือข่ายใช้คอมพิวเตอร์คนละชนิด คนละยี่ห้อ คนละระบบปฏิบัติการกัน มาตรฐาน TCP / IP จึงถูกนำมาใช้เป็นกฎเกณฑ์สำคัญในการแก้ปัญหาเหล่านี้ โดยจะกลายเป็นระบบเปิดที่สมบูรณ์และที่มีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่พีซีจนถึงเมนเฟรม และไม่จำกัดระบบปฏิบัติการที่ใช้ TCP / IP จึงเป็นมาตรฐานที่ทั่วโลกยอมรับ มีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสนับสนุน TCP / IP มากมาย ดังนั้นจึงนับได้ว่า TCP / IP เป็นสิ่งสำคัญมากสำหรับเครือข่ายอินเทอร์เน็ต

TCP มาจากคำว่า Transmission Control Protocol TCP เป็นชุดของโปรโตคอลที่ใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้ TCP เป็นโปรโตคอลที่ได้รับความนิยมที่สุดในโลกของอินเทอร์เน็ต มีแอปพลิเคชันจำนวนมากที่ใช้โปรโตคอล TCP เป็นสื่อกลางในการเชื่อมต่อ เช่น เวิลด์ไวด์เว็บ (www) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP เป็นโปรโตคอลในระดับที่ 4 เมื่อเทียบกับ OSI Model มีลักษณะการทำงานเป็น Virtual Circuit คือจะมีการทำวงจรเสมือนขึ้นมาก่อนที่จะรับส่งข้อมูลกัน นั่นคือแต่ละโหนดต้องมีตารางของ address และ destination route เพื่อให้รู้ว่าจะต้องกับใครจึงจะได้วงจรเสมือนตามที่ต้องการ เมื่อทำ connection setup เสร็จแล้วก็จะรับส่งข้อมูลกัน โดยใช้เส้นทางนี้ตลอด ดังนั้นจะไม่มีปัญหาเรื่องการเรียงลำดับของชุดข้อมูลผิดพลาด หรือเกิดการซ้ำซ้อนของข้อมูล การส่งผ่านข้อมูลบน TCP เป็น byte stream-oriented สำหรับหน้าที่ของ TCP ประกอบด้วย การตรวจสอบ error, ทำ flow control, ทำการรวมสัญญาณ (multiplex) หรือทำการแยกสัญญาณ (demultiplex) application layer connection นอกจากนี้ก็ยังทำหน้าที่ควบคุมแลกเปลี่ยนสถานะและทำการประสานเวลา (Synchronization) ด้วย

2.3 การเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย [4]

การเชื่อมต่ออินเทอร์เน็ตแบบไร้สายเป็นสิ่งที่มักได้เห็นได้ทั่วๆ ไปในปัจจุบัน ซึ่งบางคนก็เรียกการเชื่อมต่อนี้ว่า Wi-Fi แต่บางคนก็เรียกการเชื่อมต่อแบบนี้ในอีกชื่อว่า Wireless

2.3.1 Wi-Fi (Wireless Fidelity)

องค์กรที่ทำหน้าที่ทดสอบผลิตภัณฑ์ระบบเครือข่ายแบบไร้สาย (Wireless Lan) ภายใต้เทคโนโลยีการสื่อสารมาตรฐาน IEEE 802.11 ว่าอุปกรณ์แต่ละยี่ห้อสามารถติดต่อสื่อสารกันได้โดยไม่เกิดข้อผิดพลาด ซึ่งเมื่ออุปกรณ์เหล่านั้นผ่านการตรวจสอบที่เรียบร้อยแล้ว อุปกรณ์ก็จะถูกประทับตราสัญลักษณ์ Wi-Fi certified รับรองว่าอุปกรณ์ชิ้นนั้นสามารถติดต่อสื่อสารกับอุปกรณ์ตัวอื่นๆ ที่มีตราสัญลักษณ์ Wi-Fi certified ได้

2.3.2 มาตรฐานของ Wi-Fi IEEE 802.11

IEEE 802.11 คือมาตรฐานการทำงานของระบบเครือข่ายไร้สายกำหนดขึ้นโดย Institute of Electrical and Electronics Engineers (IEEE) เป็นมาตรฐานกลาง ที่ได้นำมาปฏิบัติใช้ เพื่อที่จะทำการเชื่อมโยงอุปกรณ์เครือข่ายไร้สายเข้าด้วยกันบนระบบ ดังตารางที่ 2.1

ตารางที่ 2.1 มาตรฐาน Wi-Fi ประเภทต่างๆ [4]

มาตรฐาน	ความถี่ (GHz)	แบนด์วิดท์ (MHz)	มอดูเลชัน	อัตราข้อมูล (Mb/s)
802.11-				
802.11	2.4 GHz	20 MHz	DSSS/FHSS	2 Mb/s
802.11a	5 GHz	20 MHz	OFDM	54 Mb/s
802.11b	2.4 GHz	20 MHz	DSSS	11 Mb/s
802.11g	2.4 GHz	20 MHz	DSSS/OFDM	54 Mb/s
802.11n	2.4 / 5 GHz	20 / 40 MHz	OFDM	600 Mb/s
802.11ad	2.4/5/6 GHz	20 MHz	SC-OFDM	6.76 Gb/s
802.11ac	5 GHz	20 MHz	OFDM	6.93 Gb/s

มาตรฐาน IEEE แต่ละแบบมีข้อดี ข้อเสียดังต่อไปนี้

2.3.2.1 มาตรฐาน IEEE 802.11a

เครือข่ายไวเลสแลนที่ทำงานย่านความถี่ 5 GHz มีความเร็วในการรับส่งข้อมูล 54 Mbps สามารถทำการแพร่ภาพวิดีโอและข้อมูลที่ต้องการความละเอียดสูงได้ โดยอัตราความเร็วในการรับส่งข้อมูลสามารถปรับระดับให้ช้าลงได้ เพื่อเพิ่มระยะทางการเชื่อมต่อให้มากขึ้น เช่น 54, 48, 36, 24 และ 11 Mbps เป็นต้น ขณะที่คลื่นความถี่ 5 GHz ไม่ได้ใช้งานอย่างแพร่หลาย เพราะบางประเทศไม่อนุญาตให้ใช้คลื่นความถี่นี้ ดังนั้นปัญหาการรบกวนคลื่นความถี่จึงมีน้อย ต่างจากคลื่นความถี่ 2.4 GHz ที่มีการใช้งานอย่างแพร่หลายทำให้สัญญาณของคลื่นความถี่ 2.4 GHz ถูกรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากอุปกรณ์ประเภทอื่นที่ใช้คลื่นความถี่เดียวกันได้ ระยะทางการเชื่อมต่อประมาณ 300 ฟุตจากจุดกระจายสัญญาณ Access Point หากเทียบกับมาตรฐาน 802.11b แล้ว ระยะทางจะได้น้อยกว่า 802.11b ที่คลื่นความถี่ต่ำกว่า และทั้ง 2 มาตรฐานนี้ไม่สามารถทำงานร่วมกันได้ แต่ความเร็ว 54 Mbps สามารถใช้งานได้ที่มาตรฐาน 802.11b

2.3.2.2 มาตรฐาน IEEE 802.11b

มาตรฐาน IEEE 802.11b เป็นมาตรฐานที่ได้รับความนิยมอย่างแพร่หลายรวมทั้งประเทศไทยด้วยเช่นกัน ทำงานที่คลื่นความถี่ 2.4 GHz (คลื่นความถี่นี้สามารถใช้งานแบบสาธารณะในประเทศไทยได้) มีความสามารถในการรับส่งข้อมูลที่มีความเร็ว 11 Mbps ผลิตภัณฑ์อุปกรณ์เครือข่ายไวเลสแลนมาตรฐานนี้ได้รับความนิยมจำนวนมาก โดยทุกผลิตภัณฑ์ต้องสามารถทำงานร่วมกันได้ อุปกรณ์ทุกยี่ห้อต้องผ่านการตรวจสอบจากสถาบัน Wi-Fi Alliance เพื่อตรวจสอบมาตรฐานของอุปกรณ์และความเข้ากันได้ของแต่ละผู้ผลิต อุปกรณ์ไวเลสแลนที่มาตรฐาน 802.11b ไปใช้ในองค์กรธุรกิจ สถาบันการศึกษา สถานที่สาธารณะ และกำลังแพร่เข้าสู่สถานที่พักอาศัยมากขึ้น และมาตรฐานนี้มีระบบเข้ารหัสข้อมูลแบบ WEP ที่ 128 บิต

2.3.2.3 มาตรฐาน IEEE 802.11g

มาตรฐาน 802.11g ใช้ความถี่ 2.4 GHz สามารถรับส่งข้อมูลที่มีความเร็ว 36 - 54 Mbps ซึ่งเป็นความเร็วที่สูงกว่ามาตรฐาน 802.11b โดยมาตรฐาน 802.11g สามารถปรับระดับความเร็วในการสื่อสารลงเหลือ 2 Mbps ได้ (ตามสภาพแวดล้อมของเครือข่ายที่ใช้งาน) มาตรฐานนี้เป็นที่นิยมของผู้ใช้เป็นจำนวนมากและเข้ามาแทนที่ 802.11b ที่ความเร็วต่ำกว่า

2.3.2.4 มาตรฐาน IEEE 802.11n

802.11n เป็นมาตรฐานที่สามารถทำงานบนคลื่นความถี่ 2.4 และ 5 GHz ได้ รองรับความเร็วตั้งแต่ 300-450 Mbps โดยมีเสาสัญญาณตั้งแต่ 2 - 4 เสา บนตัวอุปกรณ์กระจายสัญญาณไวเลสแลน และหากผู้ใช้ต้องการใช้งานที่ความเร็วสูงสุด เครื่องคอมพิวเตอร์พกพาหรืออุปกรณ์เคลื่อนที่ที่ต้องรองรับมาตรฐาน 802.11n ด้วยเช่นกัน มาตรฐาน 802.11n สามารถทำงานร่วมกับ 802.11b, g ได้ โดยไม่ทำให้ประสิทธิภาพทั้งระบบลดลงเหมือนมาตรฐาน 802.11g เมื่อมีอุปกรณ์ 802.11b เข้ามาใช้งานร่วมกัน

2.3.2.5 มาตรฐาน IEEE 802.11ad

มาตรฐาน IEEE 802.11ad หรือ “WiGig” เกิดจากการผลักดันจากผู้ผลิตฮาร์ดแวร์ ในวันที่ 24 กรกฎาคม 2012 Marvell และ Wilocity ได้ประกาศการเป็นคู่ค้าใหม่เพื่อนำ Wi-Fi solution แบบ tri-band ใหม่ออกสู่ตลาด โดยการใช้ความถี่ที่ 60 GHz ทรูกุททางทฤษฎีสูงสุดถึง 7 Gbps

2.3.2.6 มาตรฐาน IEEE 802.11ac

มาตรฐาน IEEE 802.11ac เป็นมาตรฐานที่ 5 GHz ให้ทรูพูกุกับแลนไร้สายแบบหลายสถานีสูงกว่าที่อย่างน้อย 1 Gbps และสำหรับลิงก์เดี่ยวที่อย่างน้อย 500 Mbps โดยการใช้ RF แบนด์วิดท์ที่กว้างกว่า (80 หรือ 160 MHz) สตรีมมากกว่า (สูงถึง 8 สตรีม) และมอดูเลตที่ความจุสูงกว่า (สูงถึง 256 QAM)

2.4 ฐานข้อมูล

ฐานข้อมูลเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เกี่ยวข้องในระบบงานต่างๆ ร่วมกันได้ โดยที่จะไม่เกิดความซ้ำซ้อนของข้อมูล และยังสามารถหลีกเลี่ยงความขัดแย้งของข้อมูลด้วย อีกทั้งข้อมูลในระบบก็จะต้องเชื่อถือได้ และเป็นมาตรฐานเดียวกัน โดยจะมีการกำหนดระบบความปลอดภัยของข้อมูลขึ้นฐานข้อมูลที่นิยมใช้ในปัจจุบันได้แก่

2.4.1 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นๆ เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา php ภาษา asp.net หรือภาษาเจเอสพี เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษาวิซวลเบสิกดอทเน็ต ภาษาจาวา หรือภาษาซีชาร์ป เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์ซ (Open Source) ที่ถูกนำไปใช้งานมากที่สุด

2.4.2 Oracle

Oracle คือ โปรแกรมจัดการฐานข้อมูล ผลิตโดยบริษัทออราเคิล ซึ่งเป็นโปรแกรมจัดการฐานข้อมูลเชิงสัมพันธ์ หรือ DBMS (Relational Database Management System) ตัวโปรแกรมนี้จะทำหน้าที่เป็นตัวกลางคอยติดต่อ ประสาน ระหว่างผู้ใช้และฐานข้อมูล ทำให้ผู้ใช้งานสามารถใช้งานฐานข้อมูลได้สะดวกขึ้น เช่นการค้นหาข้อมูลต่างๆ ภายในฐานข้อมูลที่ง่ายและสะดวก โดยผู้ใช้ไม่จำเป็นต้องทราบถึงโครงสร้างภายในของฐานข้อมูลก็สามารถเข้าใช้ฐานข้อมูลนั้นได้

ฐานข้อมูลที่เลือกใช้ได้แก่ MySQL เนื่องจากเป็นระบบฐานข้อมูลโอเพนซอร์ซ ที่ใช้งานง่าย และสามารถติดตั้งภายใน Raspberry PI ได้

2.5 การส่งและรับสัญญาณ INFRARED

2.5.1 รังสีอินฟราเรด (infrared rays)

แสงอินฟราเรดคือแสงโดยปกติทั่วไป ตาของมนุษย์เราไม่สามารถมองเห็นอินฟราเรดได้เพราะอินฟราเรดมีความยาวคลื่น 950 nm ซึ่งต่ำกว่าสเปกตรัมที่ตามนุษย์มองเห็น นั่นเป็นเหตุผลหนึ่งที่อินฟราเรดมักจะถูกเลือกใช้สำหรับการควบคุมระยะไกล และอีกเหตุผลหนึ่งเป็นเพราะไดโอดอินฟราเรดสามารถสร้างได้ง่ายและหาซื้อได้ในราคาถูก แม้ว่ามนุษย์เราไม่สามารถมองเห็นแสงอินฟราเรดที่ปล่อยออกมาจากการควบคุมระยะไกลไม่ได้หมายความว่าเราไม่สามารถทำให้มองเห็นได้ กล้องวิดีโอหรือกล้องถ่ายภาพดิจิทัลสามารถ "เห็น" แสงอินฟราเรดได้ แม้แต่กล้องถ่ายภาพในโทรศัพท์มือถือในทุกวันนี้ เพียงกดปุ่มใดๆ ในรีโมทคอนโทรลซึ่งมายังกล้องถ่ายภาพจากโทรศัพท์ เราสามารถมองเห็นการกะพริบของอินฟราเรดได้ รังสีอินฟราเรดมีแหล่งกำเนิดมากมาย เช่น แสงอินฟราเรดจากดวงอาทิตย์ หลอดไฟ เทียน ระบบทำความร้อน แม้กระทั่งร่างกายของเราสามารถแผ่กระจายรังสีอินฟราเรดได้ ดังนั้นในการสื่อสารควบคุมด้วยอินฟราเรดจึงต้องใช้ความระมัดระวังเพื่อให้มั่นใจได้ว่าข้อมูลอินฟราเรดที่ถูกส่งออกไป เดินทางไปถึงฝั่งรับอย่างไม่เกิดข้อผิดพลาด

2.5.2 การส่งและการรับสัญญาณจาก Infrared Remote Control

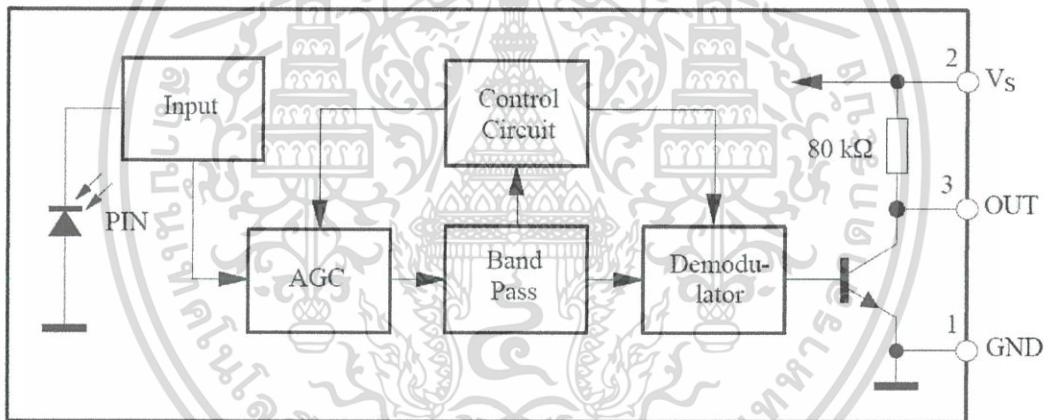
2.5.2.1 การส่งสัญญาณ Infrared จากเครื่องมืออิเล็กทรอนิกส์ที่ใช้ส่งการส่งแสงอินฟราเรด (Infrared Remote Control)

การประยุกต์ใช้คลื่นอินฟราเรดจะเป็นการประยุกต์ใช้ในการสื่อสารแบบไร้สาย (Wireless communication) ในการควบคุมอุปกรณ์หรือเครื่องใช้ไฟฟ้า โดยการส่งสัญญาณผ่านทางหลอดLED (Light emitting diode) ผ่านตัวส่ง (Transmitter) หรือ laser diode และจะมีตัวรับ (receiver) ทำหน้าที่แปลงสัญญาณข้อมูล กลับไปเป็นข้อมูลเริ่มแรกหรือข้อมูลต้นทางนั่นเอง โดยในการส่งสัญญาณนั้นเพื่อที่จะหลีกเลี่ยงการถูกรบกวนจากแสงต่างๆ หรือเพื่อไม่ให้เกิดการซ้ำกันในการส่งสัญญาณ Infrared จึงมีการกำหนด หรือ สร้างรูปแบบของสัญญาณที่จะส่งออกไป

เรียกว่า การ Coding ซึ่งสัญญาณที่ได้จะถูกส่งออกไปในรูปของสัญญาณดิจิทัล (Digital Pulse) ซึ่งมีความถี่ที่แน่นอนอยู่ค่าหนึ่งในช่วง 30 – 60 kHz ซึ่ง IR ของยี่ห้อ LG จะมีความถี่อยู่ที่ 38 kHz

2.5.2.2 การรับสัญญาณ Infrared จาก Infrared Remote Control

ในการรับสัญญาณจะใช้เซนเซอร์อินฟราเรดโดยในโครงงานนี้เราจะใช้ TSOP4838 เนื่องจากสัญญาณที่ออกจาก IR มีความถี่อยู่ที่ 38kHz ซึ่ง TSOP4838 นี้เป็น receiver ที่รับสัญญาณ Infrared แบบมาตรฐานซึ่งภายใน ประกอบด้วยวงจรต่างๆ เช่น วงจรแอม (Amplifier) (แอมพลิฟาย), วงจรกรองแถบความถี่ผ่าน (Band pass), วงจรควบคุม (Control Circuit) และ วงจรดีมอดูเลต (Demodulator) ดังแสดงในบล็อกไดอะแกรม รูปที่ 2.4

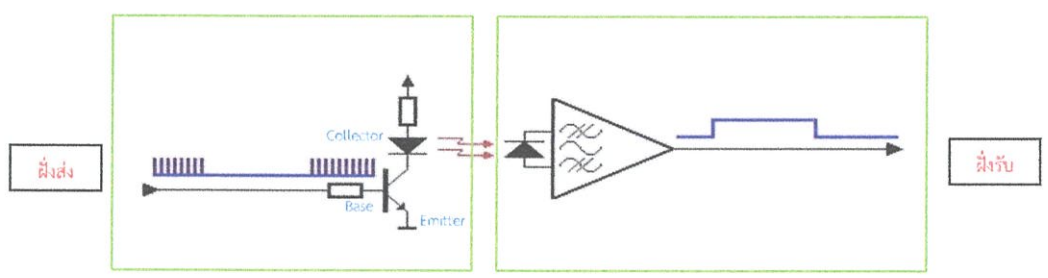


รูปที่ 2.4 แผนผังแสดงวงจรภายใน Infrared Sensor [5]

2.6 หลักการมอดูเลตสัญญาณอินฟราเรด

การมอดูเลตทำให้อินฟราเรดสามารถกระพริบส่งสัญญาณอินฟราเรดแฝงไปกับความถี่ที่เฉพาะความถี่หนึ่งจากแหล่งกำเนิดไปยังปลายทาง เมื่อด่านรับได้รับสัญญาณที่ส่งมาจะมีการกรองเอาความถี่ที่แฝงมาออกไปเหลือแต่สัญญาณข้อมูล แสดงดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



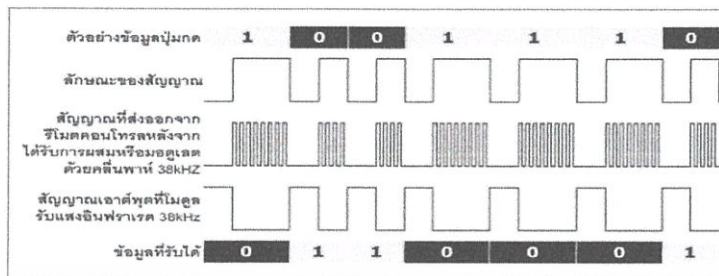
รูปที่ 2.5 การมอดูเลตที่ฝั่งส่งสัญญาณอินฟราเรด [6]

จากรูปที่ 2.5 สัญญาณอินฟราเรดจะผ่านทางขาเบสของทรานซิสเตอร์ชนิด PNP ซึ่งขาคอลเลคเตอร์จะต่อกับ IR LED โดยขาอีมิเตอร์ต่อกราวด์ เมื่อสัญญาณจะถูกส่งออกจากตัว IR LED ฝั่งรับหลังจากนั้นสัญญาณที่ถูกส่งมาจะผ่านวงจรกรองแถบความถี่ผ่าน Band Pass Filter (BPF) จะได้สัญญาณรูปสี่เหลี่ยมออกมา หรืออธิบายได้ว่าสัญญาณที่ถูกมอดูเลตและส่งออกจากอินฟราเรดไดโอดไปยังฝั่งรับ เมื่อฝั่งรับได้รับสัญญาณอินฟราเรดจะเหลือเฉพาะสัญญาณข้อมูลข่าวสารเท่านั้น

ในการสื่อสารแบบอนุกรมเรามักจะพูดถึง “marks” และ “spaces” ซึ่ง spaces เป็นสัญญาณเริ่มต้นซึ่งเป็นสถานะที่ออกจากเครื่องส่งสัญญาณ จะไม่มีแสงถูกปล่อยออกมาในช่วง spaces ช่วง “marks” สถานะของสัญญาณอินฟราเรดจะเป็นพัลส์เปิดและปิดที่ความถี่เฉพาะช่วงความถี่ระหว่าง 30 kHz และ 60 kHz ซึ่งโดยทั่วไปจะนิยมใช้ในอุปกรณ์อิเล็กทรอนิกส์

ที่ด้านฝั่งรับสัญญาณ “spaces” ที่รับได้จะแสดงเป็นระดับ high ที่ด้านเอาต์พุตของฝั่งรับ และสัญญาณ 'mark' ที่รับได้จะแสดงเป็นระดับ low ที่ด้านเอาต์พุตของฝั่งรับ “marks” และ “spaces” ไม่ได้หมายถึง 1 และ 0 เราต้องการที่จะส่งความสัมพันธ์ที่แท้จริงระหว่าง “marks” และ “spaces” เพื่อเชื่อมโยงความสัมพันธ์ของ 1 และ 0 ขึ้นอยู่กับโปรโตคอลที่ถูกนำมาใช้ ลักษณะของสัญญาณข้อมูลที่เกิดขึ้นในกระบวนการส่งและรับข้อมูลอนุกรมของรีโมทคอนโทรลอินฟราเรดแสดงในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

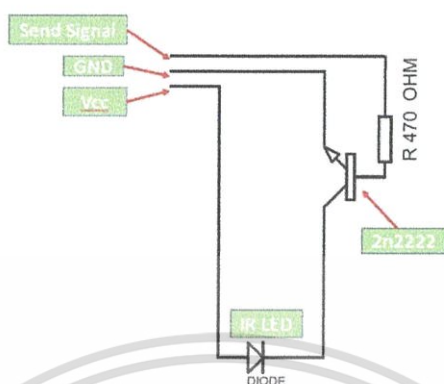


รูปที่ 2.6 ลักษณะของสัญญาณข้อมูลที่เกิดขึ้นในกระบวนการส่งและรับข้อมูล [6]

จากรูปที่ 2.6 จะเห็นได้ว่า อินฟราเรดจากรีโมทคอนโทรลส่งสัญญาณที่ถูกมอดูเลตด้วยคลื่นพาห้ 38 kHz เพื่อป้องกันการรบกวนของสัญญาณ ทางด้านตัวรับอินฟราเรดที่เครื่องปรับอากาศจะทำการดีมอดูเลตสัญญาณที่รับได้ โดยจะทำการสลับระดับของ “marks” และ “spaces” จาก low ให้เป็น high และจาก high ให้เป็น low ตามลำดับ และทำการแปลงบิตแพทเทิร์นสัญญาณที่รับได้ เป็นเลขบิตฐาน 2 โดยวิธีการแปลงเป็นเลขบิตฐาน 2 นั้นสามารถดูได้ตามรูปที่ 2.6 นอกจากนี้ข้อมูลบิตฐาน 2 ที่รับ และส่งจะตรงกันข้ามกัน

2.7 วงจรขยายสัญญาณอินฟราเรด

วงจรขยายสัญญาณอินฟราเรด มีหลักการทำงานคือเมื่อ Raspberry Pi ต้องการจะส่งสัญญาณผ่าน IR LED สัญญาณจะเริ่มผ่านทางขา Base ของตัวทรานซิสเตอร์ชนิด PNP เบอร์ 2N2222 ขา Collector ต่อกับ IR LED ขั้วแคโทด ส่วนขั้วแอนโนดต่อกับแหล่งจ่ายไฟขนาด 3V และขา Emitter ต่อกับ GND จากเดิมที่การส่งสัญญาณโดยไม่ผ่านวงจรขยายสัญญาณอินฟราเรดจะมีค่ากระแสอยู่ประมาณ 0.5 mA เมื่อทำการต่อวงจรขยายสัญญาณอินฟราเรด ค่ากระแสจะเพิ่มขึ้นเป็น 0.7 mA ส่งผลให้ระยะของการส่งสัญญาณมีระยะทางเพิ่มขึ้น โดยมีวงจรดังรูปที่ 2.7



รูปที่ 2.7 วงจรขยายสัญญาณอินฟราเรด

2.8 โครงข่ายประสาทเทียม [7]

โครงข่ายประสาทเทียม หรือ ช่างงานประสาทเทียม (อังกฤษ: artificial neural network) คือ โมเดลทางคณิตศาสตร์หรือโมเดลทางคอมพิวเตอร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (connectionist) แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (bioelectric network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท (neurons) และจุดประสานประสาท (synapses) ช่างงานประสาทเกิดจากการเชื่อมต่อระหว่างเซลล์ประสาทจนเป็นเครือข่ายที่ทำงานร่วมกัน

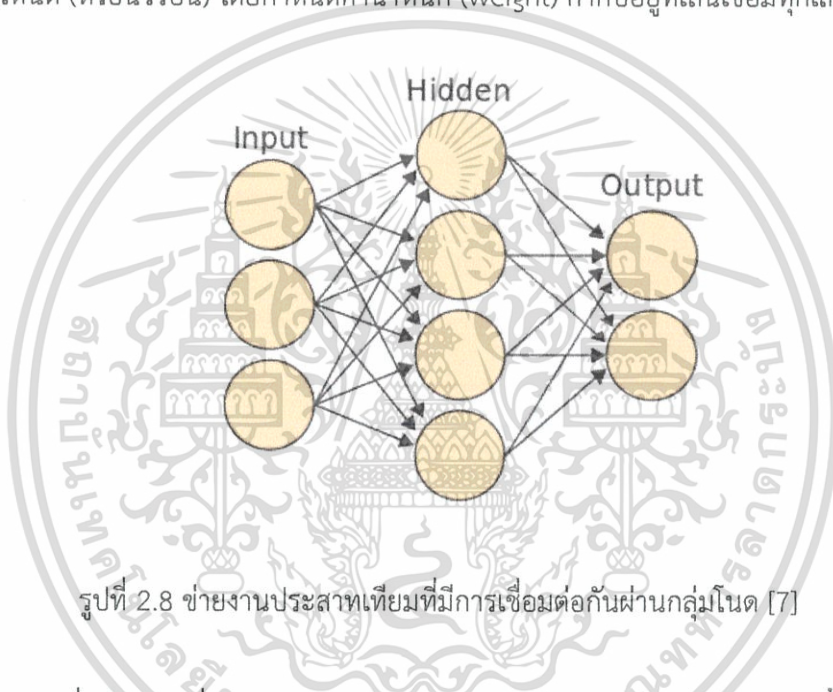
2.8.1 โครงสร้าง

ในปัจจุบันว่าช่างงานประสาทเทียมมีโครงสร้างแตกต่างจากช่างงานในสมอง แต่ก็ยังเหมือนสมองในแง่ที่ว่าช่างงานประสาทเทียม คือการรวมกลุ่มแบบขนานของหน่วยประมวลผลย่อยๆ และการเชื่อมต่อนี้เป็นส่วนสำคัญที่ทำให้เกิดสติปัญญาของช่างงาน เมื่อพิจารณาขนาดแล้ว สมองมีขนาดใหญ่กว่าช่างงานประสาทเทียมอย่างมาก รวมทั้งเซลล์ประสาทยังมีความซับซ้อนกว่าหน่วยย่อยของช่างงาน อย่างไรก็ตามหน้าที่สำคัญของสมอง เช่นการเรียนรู้ ยังคงสามารถจำลองขึ้นด้วยช่างงานประสาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 โมเดล

ข่ายงานประสาทแบบป้อนไปหน้า (feedforward) ดังรูปที่ 2.8 ประกอบด้วยเซตของ โหนด (node) ซึ่งอาจจะถูกกำหนดให้เป็นโหนดอินพุต (input nodes) โหนดเอาต์พุต (output nodes) หรือโหนดอยู่ระหว่างกลางซึ่งเรียกว่า โหนดฮิดเดน (hidden nodes) มีการเชื่อมต่อระหว่างโหนด (หรือนิวรอน) โดยกำหนดค่าน้ำหนัก (weight) กำกับอยู่ที่เส้นเชื่อมทุกเส้น



รูปที่ 2.8 ข่ายงานประสาทเทียมที่มีการเชื่อมต่อกันผ่านกลุ่มโหนด [7]

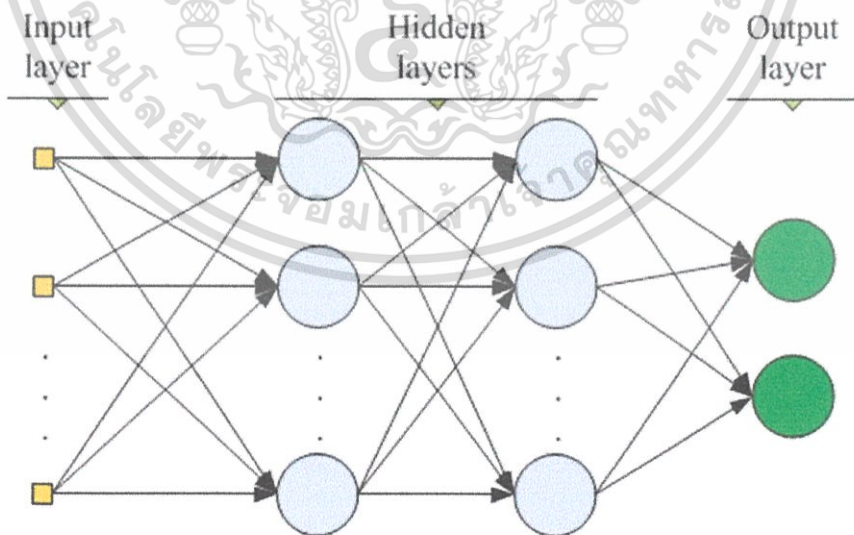
เมื่อข่ายงานเริ่มทำงานจะมีการกำหนดค่าให้แก่โหนดอินพุต โดยค่าเหล่านี้อาจจะได้มาจากการกำหนดโดยมนุษย์ จากเซนเซอร์ที่วัดค่าต่างๆ หรือผลจากโปรแกรมอื่นๆ จากนั้นโหนดอินพุตจะส่งค่าที่ได้รับไปตามเส้นเชื่อมขาออก โดยที่ค่าที่ส่งออกไปจะถูกคูณกับค่าน้ำหนักของเส้นเชื่อม โหนดในชั้นถัดไปจะรับค่า ซึ่งเป็นผลรวมจากโหนดต่างๆ แล้วจึงคำนวณผล โดยทั่วไปจะใช้ฟังก์ชันซิกมอยด์ (sigmoid function) แล้วส่งค่าไปยังชั้นถัดไป การคำนวณเช่นนี้จะเกิดขึ้นไปเรื่อยๆ ทีละชั้น จนถึงโหนดเอาต์พุต โดยในยุคแรก (ราว ค.ศ. 1970) จำนวนชั้นจะถูกกำหนดไว้เป็นค่าคงที่ แต่ในปัจจุบันมีการนำขั้นตอนวิธีเชิงพันธุกรรมมาช่วยออกแบบโครงสร้างของข่ายงานประสาทซึ่งชื่อว่า ดูนีวโรอีโวลูชัน (Neuroevolution)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3 ประเภทของข่ายงานประสาทเทียม

2.8.3.1 เพอร์เซ็ปตรอนหลายชั้น (multi-layer perceptron)

โครงข่ายประสาทเทียมแบบ MLP เป็นรูปแบบหนึ่งของโครงข่ายประสาทเทียมที่มีโครงสร้างเป็นแบบหลายๆชั้น ดังรูปที่ 2.9 ใช้สำหรับงานที่มีความซับซ้อนได้ผลเป็นอย่างดี โดยมีกระบวนการฝึกฝนเป็นแบบมีผู้สอน (Supervised) และใช้ขั้นตอนการส่งค่าย้อนกลับ (Backpropagation) สำหรับการฝึกฝนกระบวนการส่งค่าย้อนกลับ ประกอบด้วย 2 ส่วนย่อยคือการส่งผ่านไปข้างหน้า (Forward Pass) การส่งผ่านย้อนกลับ (Backward Pass) สำหรับการส่งผ่านไปข้างหน้า ข้อมูลจะผ่านเข้าโครงข่ายประสาทเทียมที่ชั้นข้อมูลเข้า และจะส่งผ่านจากอีกชั้นหนึ่งไปสู่อีกชั้นหนึ่งจนกระทั่งถึงชั้นข้อมูลออก ส่วนการส่งผ่านย้อนกลับค่าน้ำหนักการเชื่อมต่อจะถูกปรับเปลี่ยนให้สอดคล้องกับกฎการแก้ข้อผิดพลาด (Error-Correction) คือผลต่างของผลตอบที่แท้จริง (Actual Response) กับผลตอบเป้าหมาย (Target Response) เกิดเป็นสัญญาณผิดพลาด (Error Signal) ซึ่งสัญญาณผิดพลาดนี้จะถูกส่งย้อนกลับเข้าสู่โครงข่ายประสาทเทียมในทิศทางตรงกันข้ามกับการเชื่อมต่อ และค่าน้ำหนักของการเชื่อมต่อจะถูกปรับจนกระทั่งผลตอบที่แท้จริงเข้าใกล้ผลตอบเป้าหมาย



รูปที่ 2.9 เพอร์เซ็ปตรอนหลายชั้น (multi-layer perceptron) [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการการทำงานของ MLP คือในแต่ละชั้นของชั้น Hidden Layer จะมีฟังก์ชันสำหรับคำนวณเมื่อได้รับสัญญาณ (Output) จากโหนดในชั้นก่อนหน้านี้ เรียกว่า Activation Function โดยในแต่ละชั้นไม่จำเป็นต้องเป็นฟังก์ชันเดียวกันก็ได้ ชั้นซ่อนตัวนั้นมีหน้าที่สำคัญคือ จะพยายามแปลงข้อมูลที่เข้ามาในชั้น (Layer) นั้นๆ ให้สามารถแยกแยะความแตกต่างโดยใช้เส้นตรงเส้นเดียว (Linearly Separable) และก่อนที่ข้อมูลจะถูกส่งไปถึงชั้นข้อมูลออก (Output Layer) ในบางครั้งอาจจำเป็นต้องใช้ชั้นซ่อนตัวมากกว่า 1 ชั้นในการแปลงข้อมูลให้อยู่ในรูป Linearly Separable

2.9 แอนดรอยด์แอปพลิเคชัน (Android Application)

แอปพลิเคชันต่างๆ ที่สามารถติดตั้งใช้งานได้กับสมาร์ตโฟนที่ใช้ระบบปฏิบัติการ Android ซึ่งในปัจจุบันได้มีการพัฒนา Android Application กันมากเนื่องจาก Android เป็น OS ที่เป็น Open Source สามารถใช้งานได้ฟรี และติดตั้งได้กับสมาร์ตโฟนหรืออุปกรณ์ที่หลากหลาย และนักพัฒนาสามารถพัฒนา Android Application ได้ด้วย Notebook หรือ PC Desktop ธรรมดาได้ เรียกได้ว่าความสามารถของ Android ที่สามารถทำงานร่วมกับ Hardware อย่างเป็นอิสระได้เกือบทุกอย่าง จึงทำให้ได้รับความนิยมจากองค์กรธุรกิจจำนวนมากและมีการนำ Android Application มาใช้งานร่วมกับธุรกิจหลายๆ ประเภท ทั้ง Application ที่สามารถโหลดมาใช้งานได้เลย หรือ Application ที่ต้องซื้อหรือต้องเสียค่าบริการก็ตาม ถ้า Application นั้นมีประโยชน์ต่อผู้ใช้

2.9.1 Android Studio

Android Studio เป็น IDE Tools ล่าสุดจาก Google ไว้พัฒนาโปรแกรม Android โดยเฉพาะ โดยพัฒนาจากแนวคิดพื้นฐานมาจาก IntelliJ IDEA คล้ายๆ กับการทำงานของ Eclipse และ Android ADT Plugin โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา App บน Android ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัว App มุมมองที่แตกต่างกันบน สมาร์ตโฟน (Smart Phone) แต่ละรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแสดงผลบางอย่างได้ทันทีโดยไม่ต้องทำการรัน App บน Emulator รวมทั้งยังแก้ไขปรับปรุงในเรื่องของความเร็วของ Emulator ที่ยังเจอปัญหากันอยู่ในปัจจุบัน

2.9.2 ภาษา Java

Java หรือ Java programming language คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทน ภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนด้วย Java ได้

ภาษา Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

2.10 เว็บแอปพลิเคชัน (Web Application)

การพัฒนาระบบงานบนเว็บ ซึ่งมีระบบมีการไหลเวียนในรูปแบบ Online (ออนไลน์) ทั้งแบบ Local (โลคอล) ภายในวง LAN (แลน) และ Global (โกลบอล) ออกไปยังเครือข่ายอินเทอร์เน็ต ทำให้เหมาะสำหรับงานที่ต้องการข้อมูลแบบ Real Time (เรียลไทม์)

การทำงานของ Web Application นั้นโปรแกรมส่วนหนึ่งจะวางตัวอยู่บนเร็นเดอร์ริงเอนจิน (Rendering Engine) ซึ่งตัว Rendering Engine จะทำหน้าที่หลักๆ คือนำเอาชุดคำสั่งหรือรูปแบบโครงสร้างข้อมูลที่ใช้ในการแสดงผล นำมาแสดงผลบนพื้นที่ส่วนหนึ่งในจอภาพ โปรแกรมส่วนที่วางตัวอยู่บน Rendering Engine จะทำหน้าที่หลักๆ คือการเปลี่ยนแปลงแก้ไขสิ่งที่แสดงผล จัดการตรวจสอบข้อมูลที่รับเข้ามาเบื้องต้นและการประมวลผลบางส่วนแต่ส่วนการทำงานหลักๆ จะวางตัวอยู่บนเซิร์ฟเวอร์ ในลักษณะ Web Application แบบเบื้องต้น ฟังก์ชันเซิร์ฟเวอร์จะประกอบไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเว็บเซิร์ฟเวอร์ซึ่งทำหน้าที่เชื่อมต่อกับไคลเอนต์ตามโพรโตคอล HTTP/ HTTPS โดยนอกจากเว็บเซิร์ฟเวอร์จะทำหน้าที่ส่งไฟล์ที่เกี่ยวข้องเนื่องกับการแสดงผลตามมาตรฐาน HTTP ตามปกติทั่วไปแล้ว เว็บเซิร์ฟเวอร์จะมีส่วนประมวลผลซึ่งอาจจะเป็นตัวแปลภาษา เช่น Script Engine ของภาษา PHP หรืออาจจะมีการติดตั้งดอทเน็ต เฟรมเวิร์ก (.NET Framework) ซึ่งมีส่วนแปลภาษา CLR ที่ใช้แปลภาษาอินเทอร์เมดิอิต (intermediate) จากโค้ดที่เขียนด้วย VB.NET หรือ C#.NET หรืออาจจะเป็น J2EE ที่มีส่วนแปลไบต์โค้ดของคลาสที่ได้จากโปรแกรมภาษาจาวา เป็นต้น

2.10.1 HTML

HTML ย่อมาจากคำว่า Hypertext Markup Language พัฒนามาจากภาษา sgml (standard generalized markup language) โดย นาย tim berners - lee เป็นภาษามาตรฐานที่ใช้พัฒนาเอกสารในรูปแบบของเว็บเพจบนระบบเครือข่ายอินเทอร์เน็ต การเรียกใช้เอกสารเหล่านี้ทำได้โดยการใช้โปรแกรมเว็บเบราว์เซอร์ (web browser) เช่น mosaic, opera, nescape navigator, internet explorer ฯลฯ เรียกดูแฟ้มที่สร้างด้วยภาษา html ข้อดีของ html คือสามารถใช้ได้กับเครื่องคอมพิวเตอร์ และระบบปฏิบัติการได้หลากหลายชนิด

2.10.2 CSS

CSS ย่อมาจาก Cascading Style Sheet มักเรียกโดยย่อว่า "สไตลชีต" คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ "Style") ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสารไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพท์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดย องค์กร World Wide Web Consortium หรือ W3C

CSS กับ HTML / XHTML นั้นทำหน้าที่คนละอย่างกัน โดย HTML / XHTML จะทำหน้าที่ในการวางโครงร่างเอกสารอย่างเป็นทางการเป็นรูปแบบถูกต้องเข้าใจง่ายไม่เกี่ยวข้องกับการแสดงผล ส่วน CSS จะทำหน้าที่ในการตกแต่งเอกสารให้สวยงาม เรียกได้ว่า HTML /XHTML คือส่วน coding ส่วน CSS คือส่วน design

2.10.3 PHP

PHP ย่อมาจาก PHP Hypertext Preprocessor แต่เดิมย่อมาจาก Personal Home Page Tools PHP คือภาษาคอมพิวเตอร์จำพวก scripting language ภาษาจำพวกนี้คำสั่งต่างๆ จะเก็บอยู่ในไฟล์ที่เรียกว่า script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ตัวอย่างของภาษา สคริปต์เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถ สอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language นั่นคือในทุกๆ ครั้งก่อนที่เครื่องคอมพิวเตอร์ซึ่ง ให้บริการเป็น Web server จะส่งหน้าเว็บเพจที่เขียนด้วย PHP ให้เรา มันจะทำการประมวลผลตาม คำสั่งที่มีอยู่ให้เสร็จเสียก่อน แล้วจึงค่อยส่งผลลัพธ์ที่ได้ให้เรา ผลลัพธ์ที่ได้นั้นก็คือเว็บเพจที่เราเห็น ดังนั้น ถือได้ว่า PHP เป็นเครื่องมือที่สำคัญชนิดหนึ่งซึ่งช่วยให้เราสามารถสร้างเว็บเพจที่มีการโต้ตอบ กับผู้ใช้ (Dynamic Web pages) ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

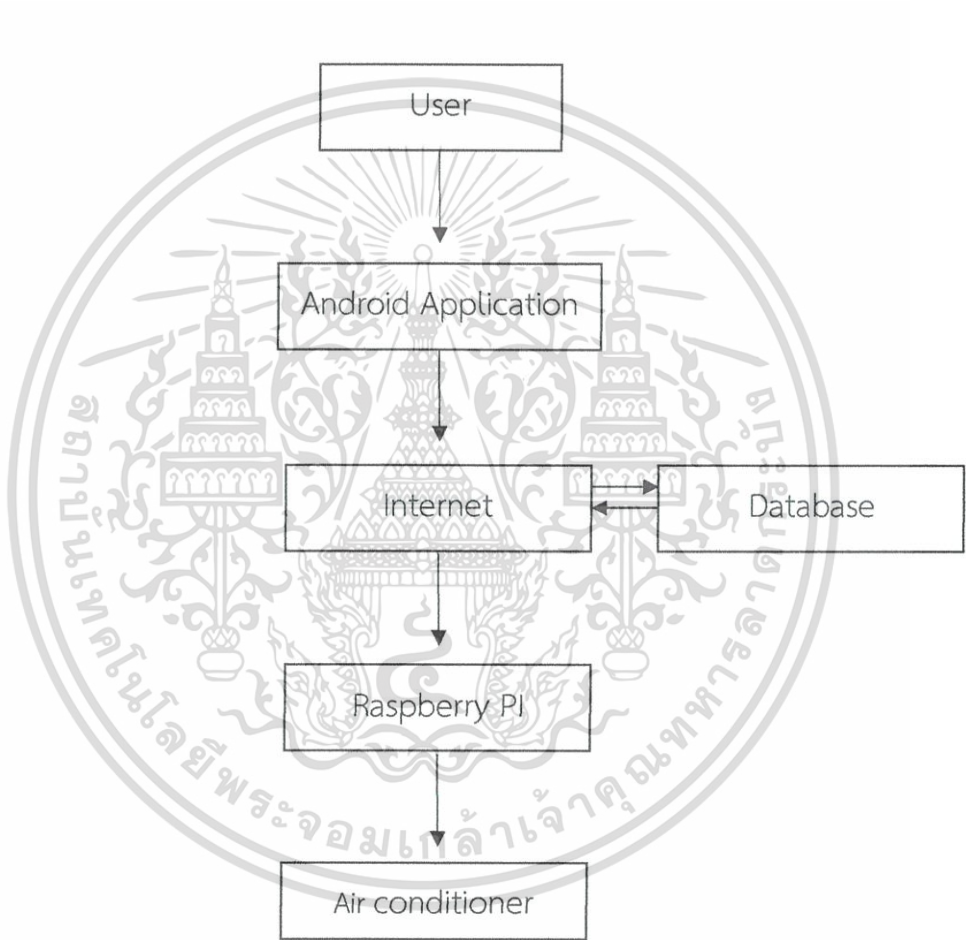
PHP เป็นผลงานที่เติบโตมาจากกลุ่มของนักพัฒนาในเชิงเปิดเผยแพร่ที่สตันฉบับ หรือ OpenSource ดังนั้น PHP จึงมีการพัฒนาไปอย่างรวดเร็ว และแพร่หลายโดยเฉพาะอย่างยิ่งเมื่อใช้ ร่วมกับ Apache Web server ระบบปฏิบัติการอย่างเช่น Linux หรือ FreeBSD เป็นต้น ในปัจจุบัน PHP สามารถใช้ร่วมกับ Web Server หลายๆ ตัวบนระบบปฏิบัติการอย่างเช่น Windows 95/98/NT เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำโครงการ

3.1 หลักการทำงานของโครงการ



รูปที่ 3.1 แผนภาพบล็อกของโครงการ

โครงการนี้จะเริ่มจากผู้ใช้เลือกโหมดการทำงาน หรือเลือกอุณหภูมิจากแอนดรอยด์แอปพลิเคชัน ข้อมูลจะถูกเก็บไว้ในฐานข้อมูลภายใน Raspberry PI จากนั้น Raspberry PI จะทำ

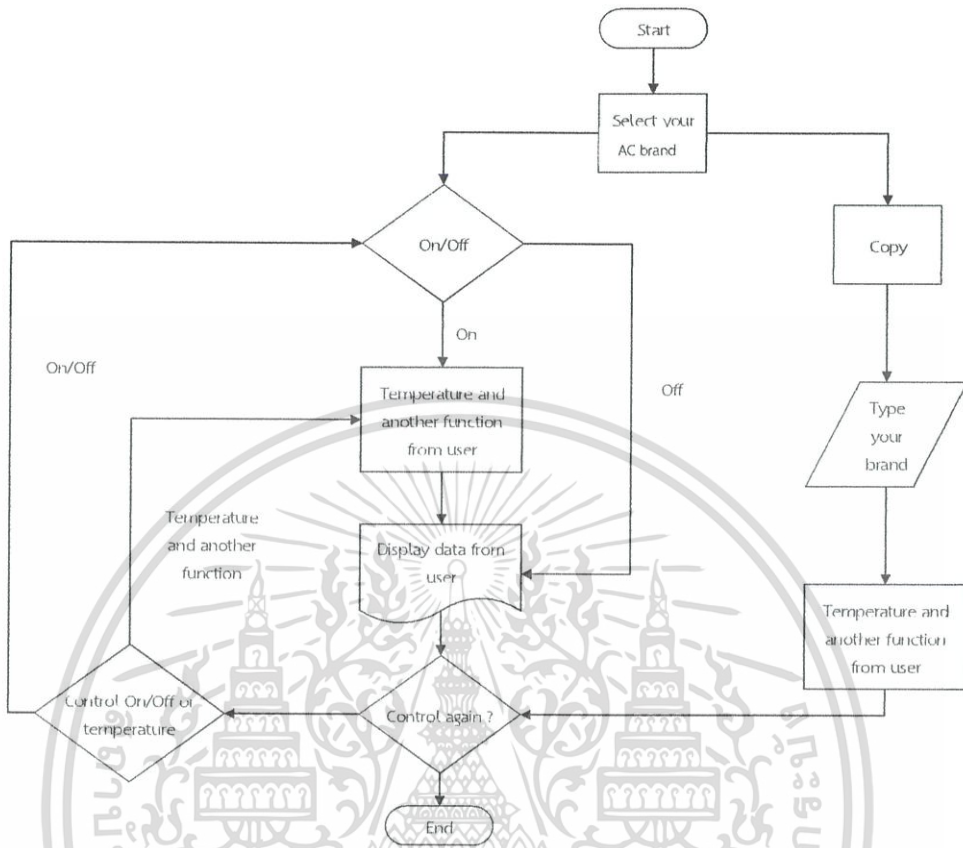
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อฐานข้อมูล และสั่งการเลือกโหมดการทำงานตามที่ใช้กำหนด หรือเข้าสู่โหมดคัดลอกสัญญาณอินฟราเรด รวมไปถึงความสามารถที่ Raspberry PI จะวัด และนำข้อมูลที่วัดได้เก็บไว้ในฐานข้อมูล และแอนดรอยด์แอปพลิเคชันจะทำการเชื่อมต่อฐานข้อมูล เพื่อแสดงอุณหภูมิที่วัดได้

นอกจากนี้ยังมีฟังก์ชันอัจฉริยะซึ่งประกอบไปด้วย วิเคราะห์พฤติกรรมการใช้งานเครื่องปรับอากาศ, ทำการเปิด-ปิดเครื่องปรับอากาศแบบอัตโนมัติตามระยะทางของผู้ใช้งานกับที่อยู่อาศัย, ประมาณค่าการใช้พลังงานไฟฟ้าในแต่ละเดือน และมีการแนะนำอุณหภูมิที่ควรจะเป็นให้แก่ผู้ใช้งาน

3.1.1. ส่วนสั่งการจากหน้าเว็บแอปพลิเคชัน

จากแผนผังการทำงานแสดงการทำงานของโครงการควบคุมเครื่องปรับอากาศผ่านอินเทอร์เน็ตโดยรับคำสั่งจากแอนดรอยด์แอปพลิเคชัน (Android Application) ผู้ใช้สามารถเลือกอุณหภูมิ โหมดการทำงาน ลักษณะทิศทางการไหลของน้ำ ระดับความเร็วของพัดลม และโหมดการคัดลอกสัญญาณจากแอนดรอยด์แอปพลิเคชัน (Android Application) ที่ออกแบบแสดงในรูปที่ 3.2

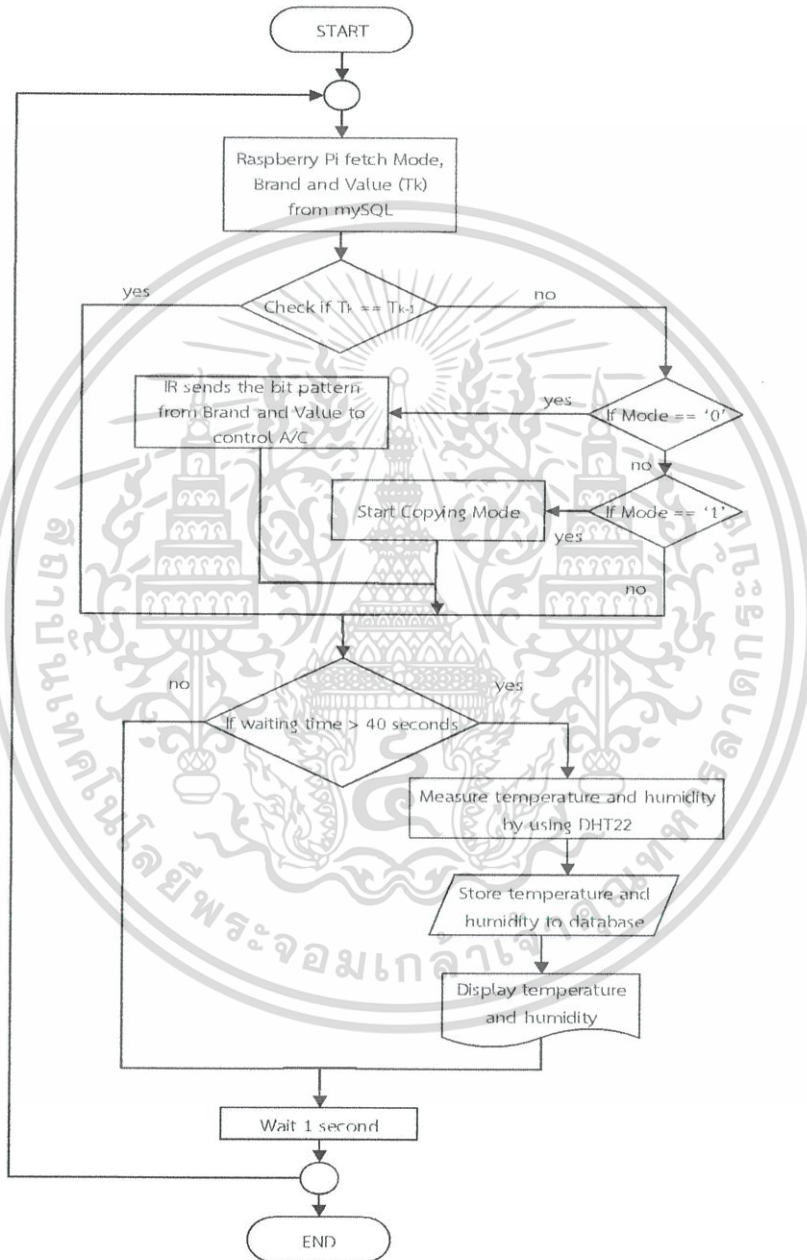


รูปที่ 3.2 แผนผังการทำงานการทำงานส่วนเว็บแอปพลิเคชัน

เมื่อผู้ใช้งานเปิดแอปพลิเคชันต้องทำการเลือกยี่ห้อของเครื่องปรับอากาศที่ต้องการควบคุม สามารถควบคุมการเปิดปิดของเครื่องปรับอากาศ การเลือกอุณหภูมิ รวมถึงฟังก์ชันพื้นฐานของเครื่องปรับอากาศ อาทิ โหมดการทำงาน ลักษณะทิศทางการลมจากเครื่องปรับอากาศ ระดับความเร็วของพัดลม ยกเว้นโหมดการตั้งเวลาปิดของเครื่องปรับอากาศ และมีการ Copy ของสัญญาณอินฟราเรดชนิดอื่น ที่นอกเหนือจากยี่ห้อทั้ง 3 ชนิด

3.1.2. ส่วน Raspberry Pi

หลักการทำงานส่วน Raspberry Pi แสดงในรูปที่ 3.3



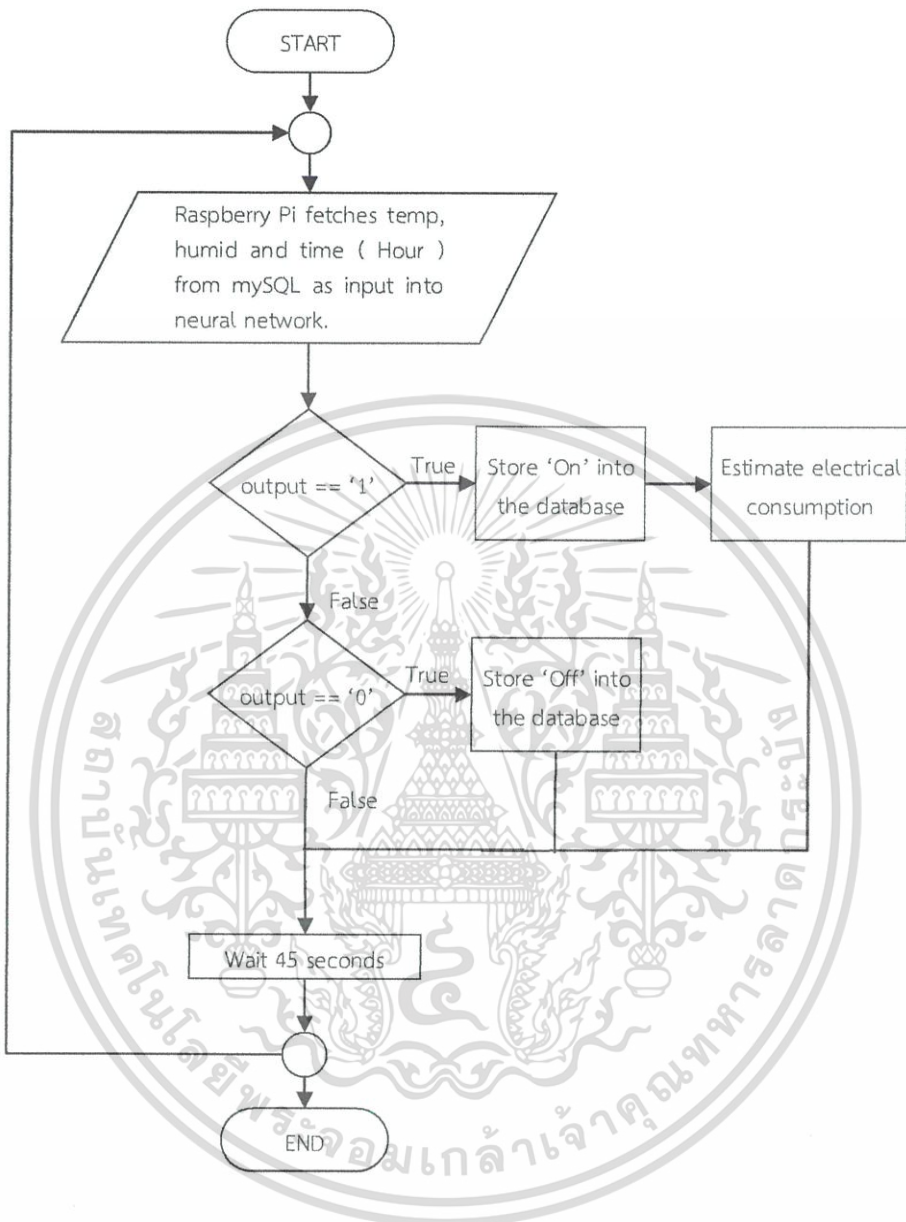
รูปที่ 3.3 แผนผังการทำงานหลักของ Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนผังการทำงานแสดงการทำงานส่วนการนำข้อมูลจากฐานข้อมูลที่ 1 ด้วย Raspberry PI ดังรูปที่ 3.5 จะเริ่มต้นหลังจากที่ผู้ใช้สั่งงานผ่านแอนดรอยด์แอปพลิเคชัน Raspberry PI จะดึงข้อมูลล่าสุดจากตัวแปร Mode, Brand, Value ซึ่งตัวแปร Mode เป็นส่วนกำหนดโหมดการทำงาน ถ้าตัวแปร Mode มีค่าเท่ากับ 0 จะเริ่มระบบสั่งการเครื่องปรับอากาศ แต่ถ้าตัวแปร Mode มีค่าเท่ากับ 1 จะเริ่มฟังก์ชันคัดลอกสัญญาณอินฟราเรดของเครื่องปรับอากาศ ส่วนต่อมาก็คือตัวแปร Brand ใช้สำหรับบอกยี่ห้อของเครื่องปรับอากาศที่จะสั่งการ หรือคัดลอกสัญญาณ ส่วนสุดท้ายคือตัวแปร Value บ่งบอกว่าจะทำการสั่งโหมดไหนเครื่องปรับอากาศ ซึ่งประกอบไปด้วยอุณหภูมิ 16 ถึง 30 องศาเซลเซียส เปิด-ปิดเครื่องปรับอากาศ การเลือกปรับทิศทางลมอัตโนมัติและความแรงลม โหมดการทำงาน หลังจากนั้นจะเปรียบเทียบค่าที่ได้ล่าสุด (Present data) กับค่าที่ได้ก่อนหน้านี้ (Previous data) ซึ่งถ้ามีค่าไม่เท่ากันจะสั่งงานให้อินฟราเรดส่งบิตแพทเทิร์นที่ถูกมอดูเลตด้วยคลื่นพาห์ 38 kHz เพื่อป้องกันการรบกวนของสัญญาณภายนอกไปยังเครื่องปรับอากาศ ตัวรับอินฟราเรดที่เครื่องปรับอากาศจะทำการตีมอดูเลตบิตแพทเทิร์นที่รับได้ นอกจากนี้จะนำอุณหภูมิที่วัดได้โดยใช้ DHT22 เก็บในฐานข้อมูลที่ 2 ดังรูปที่ 3.6

3.1.2. ส่วน smart ฟังก์ชัน

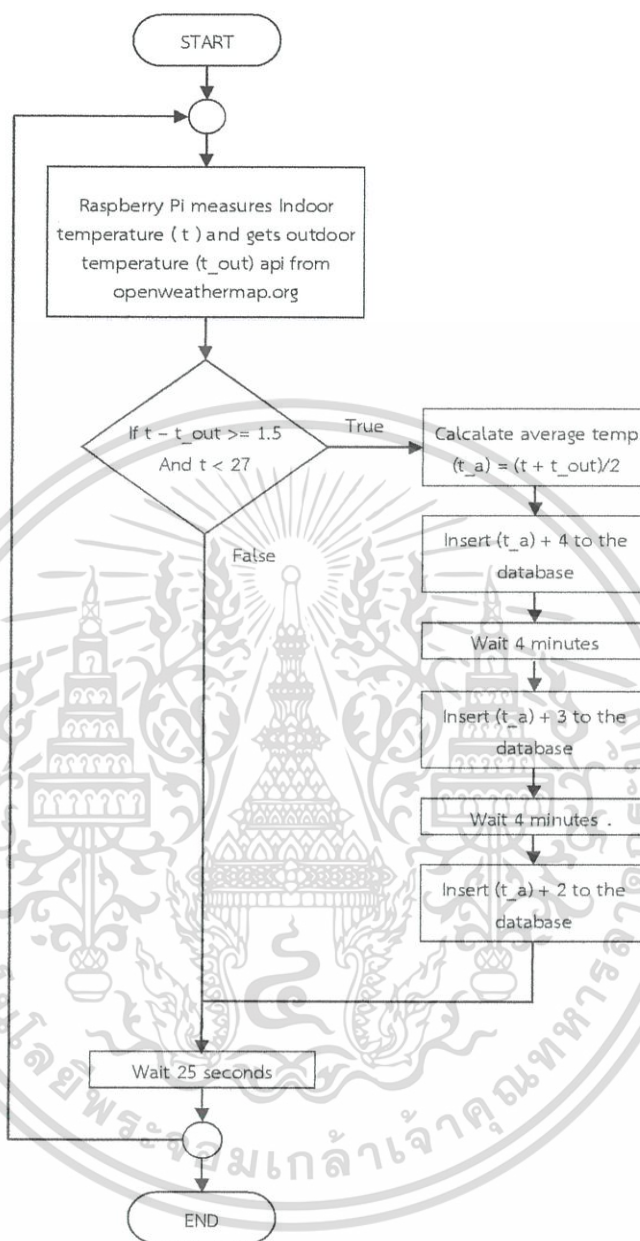
หลักการทำงานส่วน smart จะแบ่งเป็น 2 ส่วนได้แก่ วิเคราะห์สถานะการทำงานของเครื่องปรับอากาศโดยใช้โครงข่ายประสาทเทียม (neural network) และแนะนำอุณหภูมิ แสดงดังรูปที่ 3.4 และ 3.5 ตามลำดับ



รูปที่ 3.4 ส่วนวิเคราะห์สถานการณ์ทำงานของเครื่องปรับอากาศโดยใช้โครงข่ายประสาทเทียม

การทำงานส่วนนี้จะเริ่มทำหลังจากที่โครงข่ายประสาทเทียมเริ่มวิเคราะห์สถานะเครื่องปรับอากาศว่ากำลังทำงาน หรือไม่ได้กำลังทำงาน โดยแสดงเป็นบิต 1 และ 0 ตามลำดับ โดยถ้าเครื่องปรับอากาศกำลังทำงาน (บิต 1) จึงจะเริ่มประมาณค่าพลังงานที่ใช้เครื่องปรับอากาศในแต่ละเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



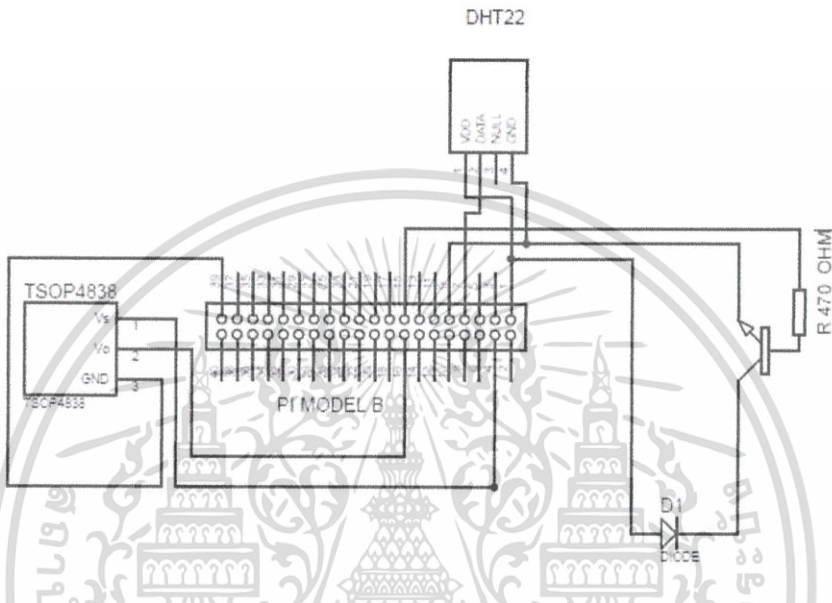
รูปที่ 3.5 ส่วนแนะนำอุณหภูมิ

การทำงานส่วนนี้จะวัดอุณหภูมิภายในที่อยู่อาศัย เทียบกับอุณหภูมิภายนอก โดยใช้เซนเซอร์วัดอุณหภูมิ DHT22 และ API ของอุณหภูมิภายนอกจาก openweathermap.org ตามลำดับ ซึ่งจะแนะนำอุณหภูมิแค้ในกรณีที่ฝนตก (อุณหภูมิภายนอกต่ำกว่าภายใน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การออกแบบ

3.2.1 วงจรควบคุมเครื่องปรับอากาศ



รูปที่ 3.6 วงจรควบคุมเครื่องปรับอากาศ

ในการทำงานของวงจรควบคุมเครื่องปรับอากาศผ่านระบบอินเทอร์เน็ตสามารถอธิบายตามรูปที่ 3.6 ได้ดังนี้

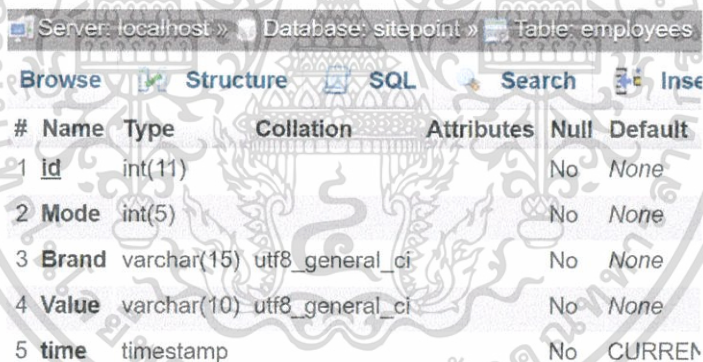
อินฟราเรด LED (IR LED) ใช้เป็นอุปกรณ์ในการส่งสัญญาณ ซึ่งหลักการทำงานคือรอคำสั่งจาก Raspberry Pi3 Model B โดยใช้ตัวทรานซิสเตอร์ชนิด PNP เบอร์ 2N2222 เป็นตัวเพิ่มกระแสให้กับอินฟราเรด LED (IR LED) มีผลทำให้ระยะทางในการส่งสัญญาณของอินฟราเรด LED (IR LED) เพิ่มขึ้น โดยต่อขาเบสของทรานซิสเตอร์เข้ากับขา GPIO 22 ของบอร์ด Raspberry Pi3 Model B โดยผ่านตัวต้านทาน 470 โอห์ม ขาคอลเลคเตอร์ของทรานซิสเตอร์ต่อกับอินฟราเรด LED (IR LED) ซึ่งขาอีกด้านและอินฟราเรด LED (IR LED) จะต่อเข้ากับไฟเลี้ยงในการทำงานขนาด 3 โวลต์

ตัวรับสัญญาณอินฟราเรด TSOP4838 ที่ต่อเข้ากับขา GPIO 23 ของ Raspberry Pi3 Model B จะทำงานเป็นตัวรับสัญญาณอินฟราเรดที่ส่งมา โดยตัวรับสัญญาณอินฟราเรด TSOP4838 จะใช้ไฟเลี้ยงในการทำงานขนาด 5 โวลต์

เซนเซอร์วัดอุณหภูมิ DHT22 ที่ต่อเข้ากับขา GPIO 4 ของ Raspberry Pi3 Model B จะทำงานเป็นเซนเซอร์วัดอุณหภูมิและจะส่งค่าอุณหภูมิที่วัดได้ผ่านขา GPIO 4 ของบอร์ด Raspberry Pi3 Model B โดยเซนเซอร์วัดอุณหภูมิจะใช้ไฟเลี้ยงในการทำงานขนาด 3 โวลต์

3.2.2 โปรแกรมระบบฐานข้อมูล

ฐานข้อมูลที่ได้ออกแบบนั้นประกอบไปด้วย 3 ส่วน คือ ส่วนเก็บอุณหภูมิที่ผู้ใช้เลือกผ่านทางเว็บแอปพลิเคชัน ส่วนเก็บข้อมูลอุณหภูมิที่วัดได้โดยใช้ DHT22 และ ส่วนเก็บสถานะของเครื่องปรับอากาศที่วิเคราะห์จากโครงข่ายประสาทเทียม ดังแสดงในรูปที่ 3.7 3.8 และ 3.9 ตามลำดับ



#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			No	None
2	Mode	int(5)			No	None
3	Brand	varchar(15)	utf8_general_ci		No	None
4	Value	varchar(10)	utf8_general_ci		No	None
5	time	timestamp			No	CURREN

รูปที่ 3.7 ฐานข้อมูลที่เก็บอุณหภูมิที่ผู้ใช้เลือกผ่านทางเว็บแอปพลิเคชัน

#	Name	Type	Collation	Attributes	Null	Defau
1	<u>id</u>	int(11)			No	None
2	Temperature	int(10)			No	None
3	Humidity	int(10)			No	None
4	Time	timestamp			No	CURF

รูปที่ 3.8 ฐานข้อมูลเก็บข้อมูลอุณหภูมิที่วัดได้โดยใช้ DHT22

#	Name	Type	Collation	Attributes	Null	I
1	<u>id</u>	int(10)			No	l
2	status	varchar(7)	latin1_swedish_ci		No	l
3	Time	timestamp		on update CURRENT_TIMESTAMP	No	C

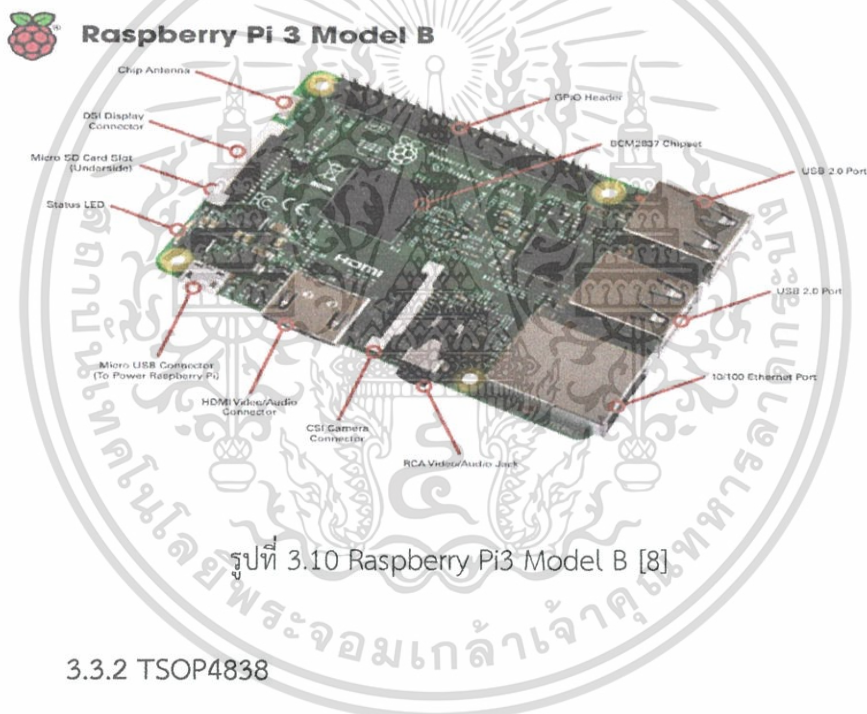
รูปที่ 3.9 ฐานข้อมูลเก็บสถานะของเครื่องปรับอากาศที่วิเคราะห์จากโครงข่ายประสาทเทียม

ฐานข้อมูลที่เก็บอุณหภูมิที่ผู้ใช้เลือกผ่านทางเว็บแอปพลิเคชันดังรูปที่ 3.7 จะเก็บอุณหภูมิที่ผู้ใช้เลือกในตัวแปรชื่อ Mode, Brand และ Value ฐานข้อมูลที่เก็บอุณหภูมิที่วัดได้โดยใช้ DHT22 ดังรูปที่ 3.8 จะเก็บไว้ในตัวแปรชื่อ Temperature และฐานข้อมูลที่ 3.9 จะเก็บสถานะของเครื่องปรับอากาศที่วิเคราะห์จากโครงข่ายประสาทเทียม ในตัวแปร status ในส่วนตัวแปร Time และ id จากฐานข้อมูลในรูปที่ 3.7, 3.8 และ 3.9 เป็นตัวแปรที่บอกเวลาที่ข้อมูลถูกเก็บในฐานข้อมูล และบอกจำนวนข้อมูลที่ถูกเก็บในฐานข้อมูลตามลำดับ

3.3 เครื่องมือที่ใช้ในการทดลอง

3.3.1 Raspberry Pi3 Model B

บอร์ด Raspberry Pi3 Model B รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU, GPU และ RAM อยู่ในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย โดยมีการแสดงภาพอุปกรณ์ดังรูปที่ 3.10

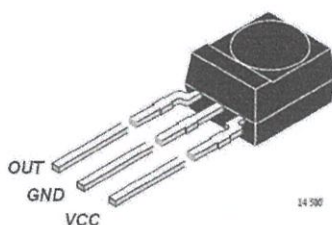


รูปที่ 3.10 Raspberry Pi3 Model B [8]

3.3.2 TSOP4838

TSOP4838 เป็นตัวรับสัญญาณอินฟราเรดที่ถูกย่อส่วนให้เล็กลง และสามารถตีมอดดูเลตสัญญาณอินฟราเรดที่รับได้ด้วยไมโครโปรเซสเซอร์ รูปที่ 3.11 ค่าสูงสุดของแรงดันขาออกคือ 100 mV การจ่ายแรงดันที่ใช้งานอยู่ระหว่าง 2.7-5.5 V ค่าสูงสุดของอุณหภูมิในการใช้งานคือ 85 ถึง -25 องศา โดยใช้ความถี่ 38 KHz ในโครงงานนี้นำมาใช้ในการรับสัญญาณอินฟราเรด เพื่อลองรับโหมด Copy สัญญาณ

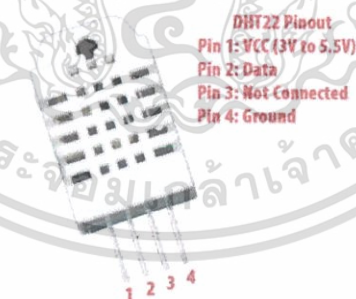
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 เซนเซอร์อินฟราเรด TSOP4838 [5]

3.3.3 DHT22

DHT22 ได้ใช้เทคนิคการเก็บข้อมูลสัญญาณแบบดิจิทัล และใช้เทคโนโลยีการวัดความชื้น ซึ่งทำให้การวัดอุณหภูมิมีความแม่นยำ และน่าเชื่อถือ วิธีวัดอุณหภูมิต้องนำตัว DHT22 ไปไว้ในที่เรากำลังต้องการต่อไฟเลี้ยงขนาด 3 V ที่ขา 1 ต่อกราวด์ที่ขา 4 และต่อขาดำดำ เข้าไปประมวลผลใน Raspberry Pi3 Model B ที่ขา 2 อุณหภูมิที่สามารถวัดได้ - 40 ถึง 80 องศาเซลเซียสมีความถูกต้องประมาณ 0.5 องศา ในโครงงานนี้นำมาใช้วัดอุณหภูมิภายในบริเวณที่ติดตั้งอุปกรณ์เพื่อเปรียบเทียบกับอุณหภูมิภายนอกซึ่งนำค่ามาจาก GPS แล้วประเมินว่าเครื่องปรับอากาศอยู่ในสถานะที่เปิดใช้งานอยู่หรือไม่ ดังรูปที่ 3.12



รูปที่ 3.12 เซนเซอร์วัดอุณหภูมิ DHT22 [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 Transistor PNP 2N2222

ทรานซิสเตอร์ 2N2222 เป็นทรานซิสเตอร์ประเภท Bipolar Junction Transistor (BJT) ใช้งานทั่วไปในวงจรขยายแรงดันต่ำ ทรานซิสเตอร์ตัวนี้ถูกออกแบบให้กระแสที่ต่ำมีค่า 0.8 A และกำลังน้อย ดังรูปที่ 3.13

รูปที่ 3.13 ทรานซิสเตอร์ชนิด PNP 2N2222 [10]

3.3.5 Infrared LED (IR LED)

IR LED มีความยาวคลื่นระหว่าง 940-950 nm สามารถนำไปใช้งานเป็นตัวส่งสัญญาณอินฟราเรด โดยสามารถรับแรงดันได้มากที่สุดที่ 3 VDC และทนกระแสได้มากที่สุด 50mA ดังรูปที่ 3.14 โดยการใช้งาน IR LED สามารถใช้งานโดยตรงกับ Raspberry Pi3 Model B ได้โดยการเชื่อมต่อก็คือ นำขาแอนโอด ต่อเข้ากับขา GPIO ของ Raspberry Pi3 Model B และขาแคโทดต่อกับกราวด์ แต่ระยะทางในการส่งจะได้น้อยมาก (แสดงการทดลองในหัวข้อ 3.4) เพราะเหตุผลนี้จึงได้จัดทำวงจรขยายสัญญาณ เพื่อเพิ่มระยะทางในการส่งสัญญาณ

รูปที่ 3.14 ไดโอดอินฟราเรด [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.6 แอนดรอยด์แอปพลิเคชัน (Android Application)

ในส่วนของแอนดรอยด์แอปพลิเคชัน (Application Android) จะรองรับเวอร์ชันแอนดรอยด์ต่ำสุดที่ Android 4.1 (Jelly Bean) ถ้าเวอร์ชันแอนดรอยด์ต่ำกว่านี้จะไม่สามารถใช้งานแอนดรอยด์แอปพลิเคชันนี้ได้

เมื่อทำการเปิดแอปพลิเคชัน ก็พบกับหน้าแรกของแอปพลิเคชัน ที่ใช้ในการเลือกยี่ห้อของเครื่องปรับอากาศ ดังแสดงในรูปที่ 3.15



รูปที่ 3.15 หน้าแรกของแอนดรอยด์แอปพลิเคชันที่ใช้ในการเลือกยี่ห้อเครื่องปรับอากาศ

การแสดงผลสถานะการทำงานของเครื่องปรับอากาศเมื่อทำการเลือกการทำงานที่อุณหภูมิต่างๆ กัน ดังแสดงในรูปที่ 3.16



รูปที่ 3.16 สถานะการทำงานเมื่อเลือกอุณหภูมิ 16 °C

การแสดงผลสถานะเมื่อทำการเลือกอุณหภูมิที่ 16 องศาเซลเซียส ดังรูปที่ 3.16 และเมื่อทำการเลือกอุณหภูมิที่แตกต่างออกไป หน้าแอปพลิเคชันก็จะแสดงตัวเลขตามอุณหภูมิที่เลือกคล้ายกับการเลือกอุณหภูมิที่ 16 องศาเซลเซียส

การแสดงผลสถานะการทำงานของเครื่องปรับอากาศเมื่อทำการเลือกโหมดการทำงานของเครื่องปรับอากาศ ดังแสดงในรูปที่ 3.17



รูปที่ 3.17 สถานะการทำงานเมื่อเลือกโหมด Auto

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีเลือกโหมด Auto ของเครื่องปรับอากาศ แสดงในรูปที่ 3.17 และเมื่อทำการเลือกโหมดการทำงาน Dry หรือ Cool หน้าแอปพลิเคชันก็จะแสดงสถานะการเลือกโหมดของเครื่องปรับอากาศคล้ายกับการเลือกโหมด Auto

การแสดงผลสถานะการทำงานของเครื่องปรับอากาศเมื่อทำการเลือกระดับการแกว่งพัดลมของเครื่องปรับอากาศ ดังแสดงในรูปที่ 3.18



รูปที่ 3.18 สถานะการทำงานเมื่อเลือกระดับการแกว่งพัดลมเป็น Auto

ในกรณีเลือกระดับการแกว่งพัดลมของเครื่องปรับอากาศเป็นระดับ Auto แสดงในรูปที่ 3.18 และเมื่อทำการเลือกระดับการแกว่งพัดลมของเครื่องปรับอากาศเป็นระดับ 1, 2 หรือ 3 หน้าแอปพลิเคชันก็จะแสดงสถานะการเลือกระดับการแกว่งพัดลมของเครื่องปรับอากาศคล้ายกับการเลือกระดับการแกว่งพัดลมของเครื่องปรับอากาศเป็นระดับ Auto

การแสดงผลสถานะการทำงานของเครื่องปรับอากาศเมื่อทำการเลือกระดับความเร็วพัดลมของเครื่องปรับอากาศ ดังแสดงในรูปที่ 3.19



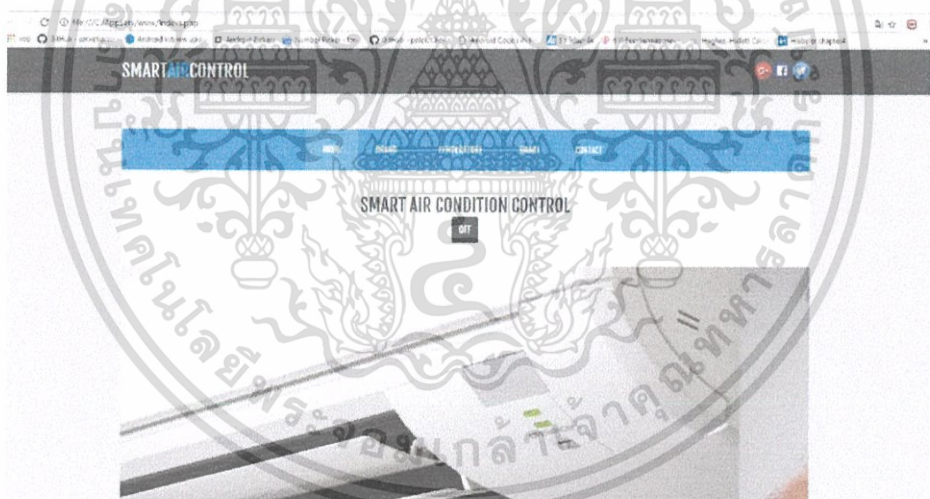
รูปที่ 3.19 สถานะการทำงานเมื่อเลือกระดับความเร็วพัดลมเป็นระดับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีเลือกระดับความแรงพัดลมของเครื่องปรับอากาศเป็นระดับ 1 แสดงในรูปที่ 3.19 และเมื่อทำการเลือกระดับความแรงพัดลมของเครื่องปรับอากาศเป็นระดับ 2 หรือ 3 หน้าแอปพลิเคชันก็จะแสดงสถานะการเลือกระดับความแรงพัดลมของเครื่องปรับอากาศคล้ายกับการเลือกระดับความแรงพัดลมของเครื่องปรับอากาศเป็นระดับ 1

3.3.7 เว็บแอปพลิเคชัน (Web Application)

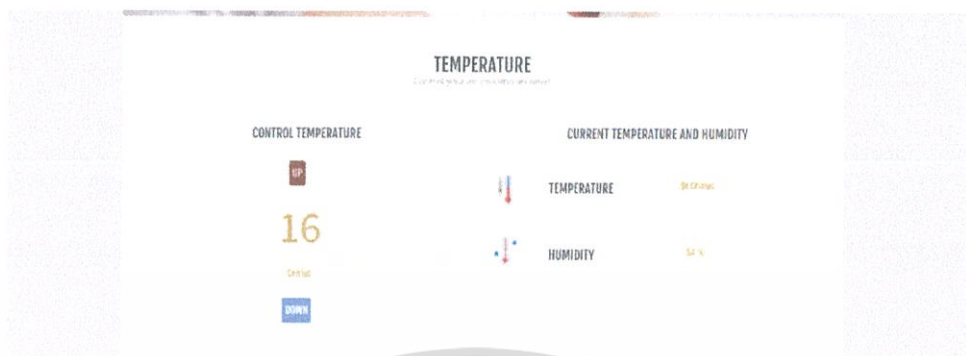
ในส่วนของเว็บแอปพลิเคชัน (Application Android) ได้ทำการออกแบบให้มีอินเทอร์เน็ตที่ใช้งานง่าย ไม่ยุ่งยาก และได้ทำการตัดฟังก์ชันการใช้งานนอกเหนือจากการสั่งการอุณหภูมิ การแสดงอุณหภูมิปัจจุบันของห้อง เนื่องจากผู้ใช้งานส่วนใหญ่ใช้งานโทรศัพท์ ระบบปฏิบัติการแอนดรอยด์ทำให้ทำให้การออกแบบเว็บแอปพลิเคชันแค่การสั่งการอุณหภูมิเท่านั้นก็เพียงพอแล้ว



รูปที่ 3.20 หน้าเว็บแอปพลิเคชัน

การแสดงผลสถานะการเลือกอุณหภูมิของเครื่องปรับอากาศในฝั่งของ Control Temperature เมื่อทำการปรับค่าอุณหภูมิขึ้นหรือลง ซึ่งสามารถปรับได้ในช่วงระหว่าง 16-30 ส่วนฝั่งของ Current Temperature and Humidity เป็นการแสดงอุณหภูมิและความชื้นของห้องในช่วงเวลาปัจจุบัน ดังแสดงในรูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 สถานะการเลือกอุณหภูมิของหน้าเว็บแอปพลิเคชัน

เมื่อผู้ใช้งานต้องการควบคุมเครื่องปรับอากาศผ่านระบบอินเทอร์เน็ตโดยใช้เบราว์เซอร์ โดยพิมพ์ URL เป็น localhost/sacc.php โดยการควบคุมผ่านเบราว์เซอร์นี้สามารถใช้งานได้ทั้งโทรศัพท์ระบบ Android และ IOS รวมถึงคอมพิวเตอร์ แล็ปท็อป ซึ่งหลักการใช้งานจะคล้ายกับแอนดรอยด์แอปพลิเคชัน

3.3.8 บรรจุภัณฑ์

ทำการออกแบบบรรจุภัณฑ์สำหรับบรรจุวงจรควบคุมเครื่องปรับอากาศผ่านระบบอินเทอร์เน็ตโดยมีแหล่งจ่ายไฟมาจากวงจรแปลงไฟจากขนาด 220 V เป็นขนาด 5V กระแส 2.1 A มีช่องสำหรับสายยูเอสบี 1 ช่อง ช่องสำหรับกดปุ่มรีเซ็ตบอร์ด 1 ช่อง และช่องสำหรับวัดอุณหภูมิห้อง 1 ช่อง รวมทั้งหมด 3 ช่อง และเสาส่งสัญญาณอินฟราเรดที่สามารถปรับทิศทางได้เพื่อความสะดวกสบายในการจัดหาที่วางบรรจุภัณฑ์ โดยที่ตัวกล่องทำจากวัสดุอะคริลิกที่มีความแข็งแรงและสวยงาม ดังแสดงในรูปที่ 3.22 ในการเชื่อมต่อ Raspberry Pi สามารถ remote ผ่านโปรแกรม Putty โดยใช้ SSH (Secure Shell) ซึ่งเป็น Network Protocol ที่สามารถแลกเปลี่ยนข้อมูลผ่านช่องทางที่ปลอดภัย เพราะ SSH ใช้วิธีเข้ารหัสข้อมูลที่ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 บรรจุภัณฑ์วงจรส่งงานเครื่องปรับอากาศ

3.3.9 สั่งการจากภายนอกวงแลน

ในการควบคุมเครื่องปรับอากาศผ่านแอนดรอยด์แอปพลิเคชัน หรือเว็บแอปพลิเคชัน จากภายนอกวงแลนได้ใช้วิธีการ forwarding port และ DDNS ผ่าน router ยี่ห้อ Huawei รุ่น HG658 V2 โดยตั้งค่าดังรูปที่ 3.23 และ 3.24 ตามลำดับ

Name:	pi	<input checked="" type="checkbox"/> Edit	<input type="checkbox"/> Delete
Type:	Customization	Application	Choose...
Interface:	INTERNET_TR069_R_VID_33		
Protocol:	TCP/UDP		
Remote host:			
External start port:	999		
External end port:	999		
Internal host:	192.168.1.2		
Internal port:	80		
Mapping name:	pi		
Validdate Code:			

รูปที่ 3.23 การตั้งค่าการ forwarding port ผ่าน router ยี่ห้อ Huawei รุ่น HG658 V2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Enable DDNS:	<input checked="" type="checkbox"/>
Status:	Synchronized
Provider:	No-IP.com ▼
	http://www.nolp.com
Host name:	acproj
	(1-256 characters)
Domain name:	ddns.net
	(eg dyn.com)
Username:	acproj
	(1-256 characters)
Password:	*****
	(1-256 characters)
Validate Code:	
	<input type="button" value="Save"/>

รูปที่ 3.24 การตั้งค่า DDNS ผ่าน router ยี่ห้อ Huawei รุ่น HG658 V2

จากรูปที่ 3.23 ได้กำหนด external port เท่ากับ 999 และ internal port เท่ากับ 80 โดยให้ส่งข้อมูลมาที่ IP 192.168.1.2 ซึ่งเป็น IP ของ Raspberry Pi ในกรณีนี้ และจากรูปที่ 3.24 ได้ทำการสมัคร DDNS จาก www.No-IP.com โดยมี url ที่สมัครชื่อว่า acproj.ddns.net และใช้กับ router ยี่ห้อ Huawei รุ่น HG658 V2

3.4 การจัดเก็บผลการทดลอง

3.4.1 การวัดระยะทางในการรับและส่งข้อมูล

3.4.1.1 การวัดระยะทางในการรับและส่งข้อมูลในห้องที่มีมืด

ทำการวัดระยะทางในการรับและส่งข้อมูลภายในห้องที่มีมืด มีค่าความสว่างเท่ากับ 4 lux ดังแสดงดังรูปที่ 3.25



รูปที่ 3.25 ห้องที่มืด

โดยระยะทางที่เริ่มการวัดระยะทางในการรับและส่งข้อมูลเริ่มต้นที่ 1 เมตร และเพิ่มระยะทางครั้งละ 1 เมตร จนกระทั่งรับสัญญาณไม่ได้

3.4.1.2 การวัดระยะทางในการรับและส่งข้อมูลในห้องที่สว่าง

ทำการวัดระยะทางในการรับและส่งข้อมูลภายในห้องสว่างโดยค่าความสว่างเท่ากับ 410 lux ดังแสดงในรูปที่ 3.26



รูปที่ 3.26 ห้องที่สว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทำการวัดระยะทางในการรับและส่งข้อมูลเริ่มต้นที่ 1 เมตร และเพิ่มระยะทางครั้งละ 1 เมตร จนกระทั่งรับสัญญาณไม่ได้

3.4.1.3 การวัดระยะทางในการรับและส่งข้อมูลในห้องที่สว่างน้อย

ทำการวัดระยะทางในการรับและส่งข้อมูลภายในห้องสว่างที่มีหลอดไฟฟลูออเรสเซนต์ 1 ดวง มีค่าความสว่างเท่ากับ 76 lux ดังแสดงในรูปที่ 3.27

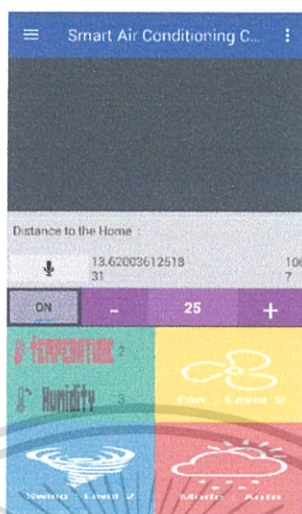


รูปที่ 3.27 ห้องที่สว่างน้อย

โดยทำการวัดระยะทางในการรับและส่งข้อมูลเริ่มต้นที่ 1 เมตร และเพิ่มระยะทางครั้งละ 1 เมตร จนกระทั่งรับสัญญาณไม่ได้

3.4.2 การคำนวณความผิดพลาดในการสั่งการจากแอนดรอยด์แอปพลิเคชัน

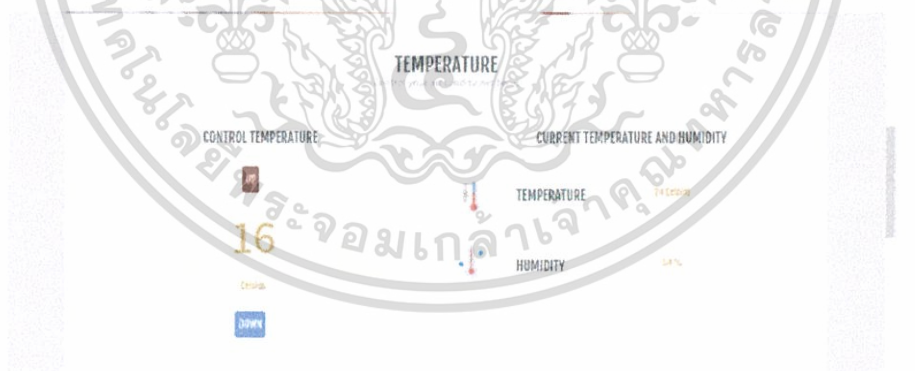
ทำการสั่งการจากแอนดรอยด์แอปพลิเคชัน และสังเกตเครื่องปรับอากาศว่าทำงานตามที่สั่งการจากแอนดรอยด์แอปพลิเคชันหรือไม่ โดยทำการสั่งการทั้งหมด 10 ครั้ง และทำการหาค่าความผิดพลาดในการสั่งการ ดังแสดงรูปที่ 3.28



รูปที่ 3.28 หน้าแอนดรอยด์แอปพลิเคชันแสดงสถานะการใช้งานต่างๆ

3.4.3 การคำนวณความผิดพลาดในการสั่งการจากเว็บแอปพลิเคชัน

ทำการสั่งการจากเว็บแอปพลิเคชัน และสังเกตเครื่องปรับอากาศว่าทำงานตามที่สั่งการจากเว็บแอปพลิเคชันหรือไม่ โดยทำการสั่งการทั้งหมด 10 ครั้ง และทำการหาค่าความผิดพลาดในการสั่งการ ดังแสดงรูปที่ 3.29



รูปที่ 3.29 หน้าเว็บแอปพลิเคชันแสดงสถานะการใช้งานต่างๆ

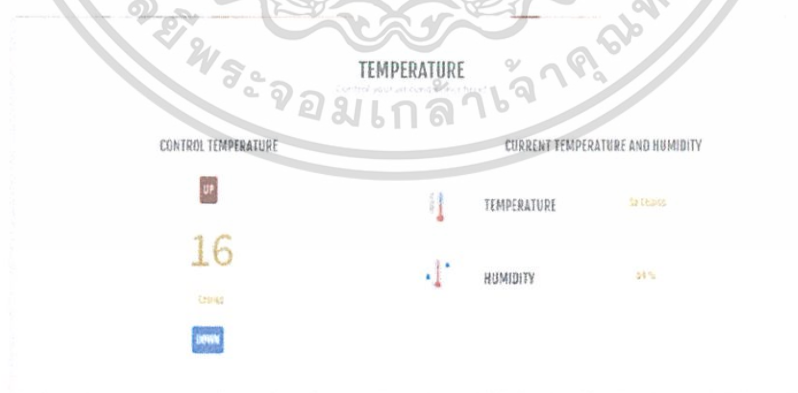
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 การวัดค่าอุณหภูมิห้องโดยใช้เซนเซอร์วัดอุณหภูมิ DHT22

การวัดอุณหภูมิของห้องโดยใช้เซนเซอร์วัดอุณหภูมิ DHT22 ส่งเกตุอุณหภูมิปัจจุบันที่แสดงบนแอนดรอยด์แอปพลิเคชันและหน้าเว็บแอปพลิเคชันจะมีค่าเท่ากับอุณหภูมิที่สั่งการ โดยทำการบันทึกเวลาเมื่อค่าอุณหภูมิปัจจุบันเท่ากับค่าอุณหภูมิที่สั่งการ ในส่วนของแอนดรอยด์แอปพลิเคชันจะแสดงในรูปที่ 3.30 และส่วนของเว็บแอปพลิเคชันจะแสดงในรูปที่ 3.31



รูปที่ 3.30 แอนดรอยด์แอปพลิเคชัน เมื่ออุณหภูมิที่เลือกมีค่าเท่ากับอุณหภูมิที่วัดได้จาก DHT22



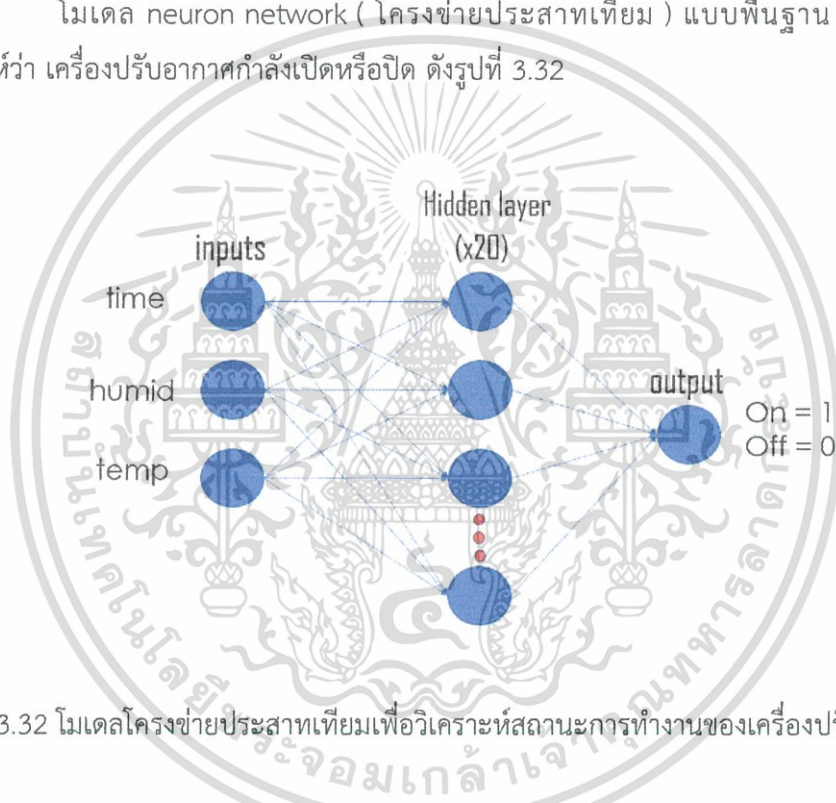
รูปที่ 3.31 แอนดรอยด์แอปพลิเคชัน เมื่ออุณหภูมิที่เลือกมีค่าเท่ากับอุณหภูมิที่วัดได้จาก DHT22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ผู้ใช้สั่งการจากแอนดรอยด์แอปพลิเคชันหรือเว็บแอปพลิเคชันแล้ว เครื่องปรับอากาศไม่ทำงาน สามารถดูได้จากอุณหภูมิปัจจุบันที่แสดงอยู่ในหน้าเว็บแอปพลิเคชัน ภายใน 10 นาที ถ้าอุณหภูมิปัจจุบันมีค่าไม่เข้าใกล้ อุณหภูมิที่เลือก แสดงว่าต้องสั่งการเลือก อุณหภูมิ หรือเลือกโหมดการทำงาน On ทางแอนดรอยด์แอปพลิเคชันหรือเว็บแอปพลิเคชันอีกครั้ง

3.4.5 โมเดลโครงข่ายประสาทเทียม

โมเดล neuron network (โครงข่ายประสาทเทียม) แบบพื้นฐาน ซึ่งสามารถวิเคราะห์ว่า เครื่องปรับอากาศกำลังเปิดหรือปิด ดังรูปที่ 3.32



รูปที่ 3.32 โมเดลโครงข่ายประสาทเทียมเพื่อวิเคราะห์สถานะการทำงานของเครื่องปรับอากาศ

อินพุตจะประกอบไปด้วยเวลา (ในหน่วยชั่วโมง), อุณหภูมิ (องศาเซลเซียส) และความชื้นสัมบูรณ์ (เปอร์เซ็นต์) มี hidden layer จำนวน 20 layer เทรนจำนวน 12 รอบ และมีเอาต์พุตเป็นบิต 0 และ 1 ซึ่งหมายถึง ไม่ได้กำลังใช้งานเครื่องปรับอากาศ และกำลังใช้งานเครื่องปรับอากาศตามลำดับ ข้อมูลที่ถูกโครงข่ายประสาทเทียมเรียนรู้ค่า weight และ bias มาจากรูปร่างข้อมูลในรูปที่ 6 จำนวน 20,000 ข้อมูลซึ่งเก็บข้อมูลเป็นระยะเวลา 7 วัน โดยมีทั้งกรณีที่เครื่องปรับอากาศกำลังใช้งาน และไม่ได้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่โครงข่ายประสาทเทียมเรียนรู้ค่า weight และ bias แล้ว จะทำการทดสอบประสิทธิภาพ โดยใช้สมการ Cross-entropy cost function (C) ดังสมการที่ 3.1

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] \quad (1)$$

โดย n คือจำนวนข้อมูลที่นำมาเรียนรู้

a คือเอาต์พุตที่ออกจากโครงข่ายประสาทเทียม

y คือ เอาต์พุตที่ต้องการ

จากสมการที่ 3.1 จะเห็นได้ว่าถ้าค่า C มีค่าน้อย (ประมาณ 0) จะหมายถึงค่าที่ออกจากโครงข่ายประสาทเทียมนั้นมีค่าใกล้เคียงกับเอาต์พุตที่เราต้องการ และค่า C จะมีค่ามากกว่าศูนย์เสมอ

นอกจากใช้สมการ Cross-entropy cost function แล้วยังใช้การเปรียบเทียบความถูกต้อง (accuracy) ของสถานะเครื่องปรับอากาศที่โมเดลโครงข่ายประสาทเทียมวิเคราะห์ กับสถานะเครื่องปรับอากาศที่กำลังทำงาน โดยข้อมูลที่เข้ามาทดสอบจะมีทั้งชุดที่โครงข่ายประสาทเทียมได้ทำการเรียนรู้แล้ว (training data) และข้อมูลใหม่ที่ยังไม่ได้ทำการเรียนรู้ (evaluation data)

3.4.6 การใช้เวลาในการใช้พลังงานของเครื่องปรับอากาศ

การใช้เวลาในการใช้พลังงานเครื่องปรับอากาศจะประมาณในรูปของเวลา โดยแสดงอัตราการใช้พลังงานเครื่องปรับอากาศที่ใช้ในสัญลักษณ์ 3 สี ได้แก่ สีแดง, สีเหลือง และเขียว หมายถึงใช้งานเครื่องปรับอากาศมากกว่า 20 ชั่วโมงต่อเดือนขึ้นไป, ใช้งานเครื่องปรับอากาศระหว่าง 10 ถึง 20 ชั่วโมงต่อเดือน และใช้งานเครื่องปรับอากาศน้อยกว่า 10 ชั่วโมงต่อเดือน ตามลำดับ โดยทำการทดลองเวลาที่ใช้งานเครื่องปรับอากาศที่ประมาณโดยโปรแกรมเทียบกับเวลาที่ใช้งานเครื่องปรับอากาศจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.7 การใช้ระยะเวลาที่เครื่องปรับอากาศใช้พลังงาน

ทำการทดลองเทียบเวลาเปิดใช้งานเครื่องปรับอากาศที่ประมาณโดยโปรแกรมกับเวลาที่ใช้งานเครื่องปรับอากาศจริง โดยแสดงอัตราการใช้พลังงานเครื่องปรับอากาศที่ใช้ในรูปสัญลักษณ์ 3 สี ได้แก่ สีแดง, สีเหลือง และเขียว หมายถึงใช้งานเครื่องปรับอากาศมากกว่า 20 ชั่วโมงต่อวันขึ้นไป, ใช้งานเครื่องปรับอากาศระหว่าง 10 ถึง 20 ชั่วโมงต่อวัน และใช้งานเครื่องปรับอากาศน้อยกว่า 10 ชั่วโมงต่อวันตามลำดับโดยจะรีเซ็ตค่าเวลานี้ทุกๆ 1 เดือน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

การทดสอบควบคุมเครื่องปรับอากาศผ่านอินเทอร์เน็ต ได้จัดเก็บผลการทำงานของระบบตามบทที่ 3 และสามารถแสดงได้ดังหัวข้อต่อไปนี้

4.1 การวัดระยะทางในการรับและส่งข้อมูล

4.1.1 การวัดระยะทางในการรับและส่งข้อมูลในห้องที่มีมืด

ทำการวัดระยะทางในการรับและส่งข้อมูลภายในห้องที่มีมืดทึบ มีค่าความสว่างเท่ากับ 4 lux โดยทำการวัดระยะทางในการรับและส่งข้อมูลเริ่มต้นที่ 1 เมตร และเพิ่มระยะทางครั้งละ 1 เมตร โดยที่ระยะการทดลองสูงสุดอยู่ที่ 6 เมตร ผลลัพธ์ที่ได้แสดงดังตารางที่ 4.1

ตารางที่ 4.1 ระยะทางในการรับและส่งข้อมูลในห้องที่มีมืด

ระยะทาง	สัญญาณ
1 เมตร	รับสัญญาณได้ตามปกติ
2 เมตร	รับสัญญาณได้ตามปกติ
3 เมตร	รับสัญญาณได้ตามปกติ
4 เมตร	รับสัญญาณได้ตามปกติ
5 เมตร	รับสัญญาณได้ตามปกติ
6 เมตร	ไม่สามารถรับสัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การวัดระยะทางในการรับและส่งข้อมูลในห้องที่สว่างน้อย

ทำการวัดระยะทางในการรับและส่งข้อมูลภายในห้องที่สว่างน้อย มีค่าความสว่างเท่ากับ 76 lux โดยทำการวัดระยะทางในการรับและส่งข้อมูลเริ่มต้นที่ 1 เมตร และเพิ่มระยะทางครั้งละ 1 เมตร โดยที่ระยะการทดลองสูงสุดอยู่ที่ 6 เมตร ผลลัพธ์ที่ได้แสดงดังตารางที่ 4.2

ตารางที่ 4.2 ระยะทางในการรับและส่งข้อมูลในห้องที่สว่างน้อย

ระยะทาง	สัญญาณ
1 เมตร	รับสัญญาณได้ตามปกติ
2 เมตร	รับสัญญาณได้ตามปกติ
3 เมตร	รับสัญญาณได้ตามปกติ
4 เมตร	รับสัญญาณได้ตามปกติ
5 เมตร	รับสัญญาณได้ตามปกติ
6 เมตร	ไม่สามารถรับสัญญาณได้

4.1.3 การวัดระยะทางในการรับและส่งข้อมูลในห้องที่สว่าง

ทำการวัดระยะทางในการรับและส่งข้อมูลภายในห้องที่สว่าง มีค่าความสว่างเท่ากับ 440 lux โดยทำการวัดระยะทางในการรับและส่งข้อมูลเริ่มต้นที่ 1 เมตร และเพิ่มระยะทางครั้งละ 1 เมตร โดยที่ระยะการทดลองสูงสุดอยู่ที่ 5 เมตร ผลลัพธ์ที่ได้แสดงดังตารางที่ 4.3

ตารางที่ 4.3 ระยะทางในการรับและส่งข้อมูลในห้องที่สว่าง

ระยะทาง	สัญญาณ
1 เมตร	รับสัญญาณได้ตามปกติ
2 เมตร	รับสัญญาณได้ตามปกติ
3 เมตร	รับสัญญาณได้ตามปกติ
4 เมตร	รับสัญญาณได้ตามปกติ
5 เมตร	ไม่สามารถรับสัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการวัดระยะทางในการรับและส่งสัญญาณอินฟราเรดสามารถสรุปได้ว่า ถ้าแสงสว่างในสภาวะแวดล้อมนั้นมาก จะมีผลทำให้ระยะทางในการรับสัญญาณอินฟราเรดลดน้อยลงดังตารางที่ 4.1, 4.2 และ 4.3 ภายในห้องที่มีค่าความสว่างเท่ากับ 4 lux, 76 lux และ 440 lux สามารถรับสัญญาณได้ตามปกติภายในระยะ 5 เมตร, 5 เมตร และ 4 เมตร ตามลำดับ นอกจากนี้ผลการทดลองสามารถสรุปได้ว่าระยะทางในการรับและส่งสัญญาณอินฟราเรดเพียงพอต่อระยะทางการใช้งาน โดยเปรียบเทียบกับระยะการสั่งการจากระยะการปรับอากาศทุกๆ ไป ซึ่งมีระยะการสั่งการที่ใกล้เคียงกัน

4.2 การคำนวณความผิดพลาดในการสั่งการจากหน้าแอนดรอยด์แอปพลิเคชัน

ทำการสั่งการจากหน้าแอนดรอยด์แอปพลิเคชัน และสังเกตเครื่องปรับอากาศว่าสามารถทำงานตามที่สั่งการจากหน้าแอนดรอยด์แอปพลิเคชันได้หรือไม่ โดยทำการสั่งการทั้งหมด 10 ครั้ง และทำการหาค่าความผิดพลาดในการสั่งการ ผลลัพธ์ที่ได้แสดงดังตารางที่ 4.4

ตารางที่ 4.4 การสั่งการผ่านแอนดรอยด์แอปพลิเคชัน

อุณหภูมิ	จำนวนครั้งที่สั่งการ	จำนวนครั้งที่เครื่องปรับอากาศทำงาน	เปอร์เซ็นต์ความผิดพลาด
16 °C	10	10	0
17 °C	10	10	0
18 °C	10	10	0
19 °C	10	10	0
20 °C	10	10	0
21 °C	10	10	0
22 °C	10	10	0
23 °C	10	10	0
24 °C	10	10	0
25 °C	10	10	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุณหภูมิ	จำนวนครั้งที่สั่งการ	จำนวนครั้งที่ เครื่องปรับอากาศ ทำงาน	เปอร์เซ็นต์ความ ผิดพลาด
26 °C	10	10	0
27 °C	10	10	0
28 °C	10	10	0
29 °C	10	10	0
30 °C	10	10	0

จากตารางที่ 4.4 สรุปผลได้ว่า ผู้ใช้งานสั่งการเลือกอุณหภูมิ โหมดการทำงาน On Off การเลือกปรับทิศทางลมอัตโนมัติ ความแรงลม และโหมดการทำงาน ผ่านแอนดรอยด์แอปพลิเคชัน ไม่เกิดความผิดพลาด

4.3 การคำนวณความผิดพลาดในการสั่งการจากหน้าเว็บแอปพลิเคชัน

ทำการสั่งการจากหน้าเว็บแอปพลิเคชัน และสังเกตเครื่องปรับอากาศว่าทำงานตามที่สั่งการจากหน้าเว็บแอปพลิเคชันได้หรือไม่ โดยทำการสั่งการทั้งหมด 10 ครั้ง และทำการหาค่าความผิดพลาดในการสั่งการ ผลลัพธ์ที่ได้แสดงดังตารางที่ 4.5

ตารางที่ 4.5 การสั่งการผ่านเว็บแอปพลิเคชัน

อุณหภูมิ	จำนวนครั้งที่สั่งการ	จำนวนครั้งที่ เครื่องปรับอากาศ ทำงาน	เปอร์เซ็นต์ความ ผิดพลาด
16 °C	10	10	0
17 °C	10	10	0
18 °C	10	10	0
19 °C	10	10	0
20 °C	10	10	0

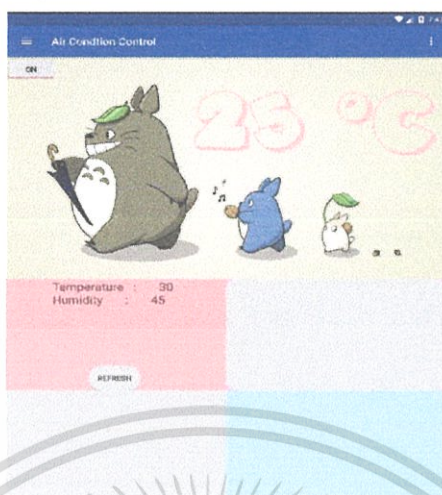
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุณหภูมิ	จำนวนครั้งที่สั่งการ	จำนวนครั้งที่ เครื่องปรับอากาศ ทำงาน	เปอร์เซ็นต์ความ ผิดพลาด
21 °C	10	10	0
22 °C	10	10	0
23 °C	10	10	0
24 °C	10	10	0
25 °C	10	10	0
26 °C	10	10	0
27 °C	10	10	0
28 °C	10	10	0
29 °C	10	10	0
30 °C	10	10	0

จากตารางที่ 4.5 สรุปผลได้ว่า ผู้ใช้งานสั่งการเลือกอุณหภูมิ โหมดการทำงาน On Off การเลือกปรับทิศทางลมอัตโนมัติ ความแรงลม และโหมดการทำงาน ผ่านเว็บแอปพลิเคชันไม่เกิดความผิดพลาด

4.4 การวัดค่าอุณหภูมิห้องโดยใช้เซนเซอร์วัดอุณหภูมิ DHT22

ทำการวัดอุณหภูมิของห้องโดยใช้เซนเซอร์วัดอุณหภูมิ DHT22 โดยค่าอุณหภูมิที่สั่งการจากหน้าแอนดรอยด์แอปพลิเคชันเท่ากับ 25 °C และค่าอุณหภูมิปัจจุบันในห้องเท่ากับ 30 °C ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 ค่าอุณหภูมิปัจจุบันในห้อง และค่าอุณหภูมิจากการสั่งการเมื่อเริ่มต้นสั่งการ
เมื่อเวลาผ่านไป 40 นาที ค่าอุณหภูมิปัจจุบันในห้องจึงเท่ากับค่าอุณหภูมิจากการสั่ง
การ ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 ค่าอุณหภูมิปัจจุบันในห้องเท่ากับค่าอุณหภูมิจากการสั่งการหลังจากผ่านไป 40 นาที

จากรูปที่ 4.1 และ 4.2 สามารถสรุปได้ว่า การที่จะทำให้อุณหภูมิที่วัดได้โดยใช้ DHT22 นั้นต้องใช้เวลาอย่างน้อย 40 นาทีเพื่อที่จะให้อุณหภูมิที่วัดได้ กับอุณหภูมิที่สั่งการมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การวัดความผิดพลาดของการเปิดเครื่องปรับอากาศตามระยะทางโดยใช้ GPS

ทำการเลือกโหมดการเปิดของเครื่องปรับอากาศตามระยะทางโดยใช้ GPS จากหน้าแอนดรอยด์แอปพลิเคชัน และสังเกตเครื่องปรับอากาศว่าทำงานตามที่สั่งการจากแอนดรอยด์แอปพลิเคชันหรือไม่ โดยทำการตั้งค่าระยะทางตั้งแต่ 200 เมตร 500 เมตร 1 กิโลเมตร 5 กิโลเมตร และ 10 กิโลเมตร โดยทำการสั่งการทั้งหมด 10 ครั้ง และทำการหาค่าความผิดพลาดในการสั่งการผลลัพธ์ที่ได้แสดงดังตารางที่ 4.6

ตารางที่ 4.6 การเปิดของเครื่องปรับอากาศตามระยะทางโดยใช้ GPS

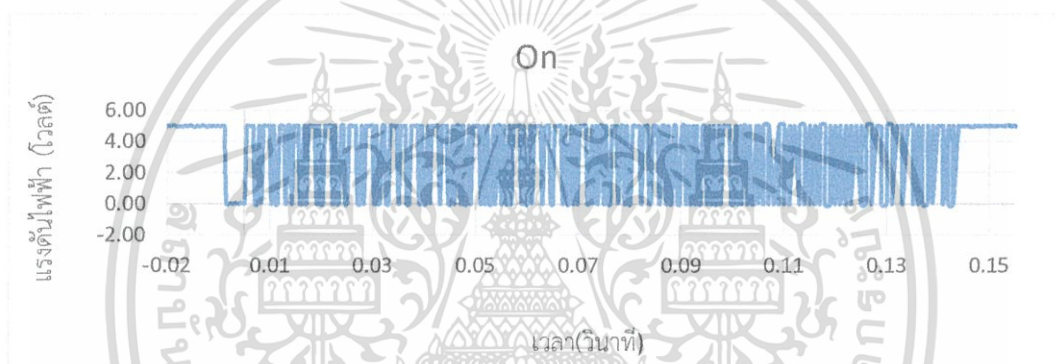
ระยะทาง	จำนวนครั้งที่สั่งการ	จำนวนครั้งที่เครื่องปรับอากาศทำงาน	เปอร์เซ็นต์ความผิดพลาด
200 เมตร	10	10	0
500 เมตร	10	10	0
1 กิโลเมตร	10	10	0
5 กิโลเมตร	10	10	0
10 กิโลเมตร	10	10	0

จากตารางที่ 4.6 สรุปผลได้ว่า ผู้ใช้งานสามารถควบคุมการเปิดของเครื่องปรับอากาศอัตโนมัติตามระยะทางโดยใช้ GPS สามารถทำงานได้โดยไม่มีข้อผิดพลาด แต่แอนดรอยด์แอปพลิเคชันจะต้องถูกเปิดใช้งานขณะนั้นด้วย นั่นถือเป็นข้อจำกัดของการควบคุมการเปิดเครื่องปรับอากาศอัตโนมัติตามระยะทางโดยใช้ GPS

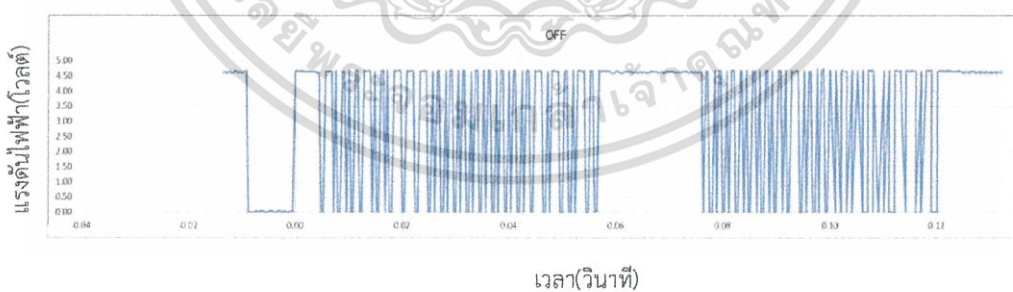
4.6 การเก็บบิตแพทเทิร์นจากรีโมทคอนโทรลรีโมท Sharp

4.6.1 เก็บค่าบิตแพทเทิร์นจากรีโมทอินฟราเรดด้วย Receiver TSOP4838 จากรีโมทเครื่องปรับอากาศ

ทำการเก็บบิตแพทเทิร์นจากรีโมทอินฟราเรดด้วย Receiver TSOP4838 จากรีโมทเครื่องปรับอากาศ เพื่อศึกษาส่วนประกอบ และคุณลักษณะสัญญาณอินฟราเรดของรีโมทเครื่องปรับอากาศที่ฝั่งส่ง แสดงดังรูปที่ 4.3 ถึง 4.6 และกำหนด interrupt routine ทุกๆ 50 ไมโครวินาที

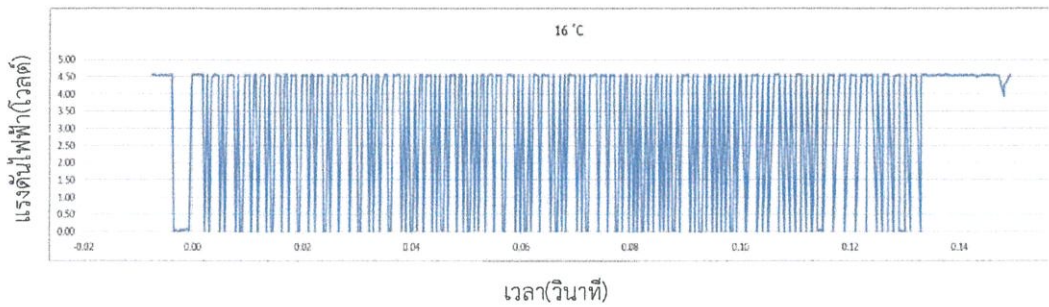


รูปที่ 4.3 กราฟลักษณะบิตสัญญาณที่โหมด On

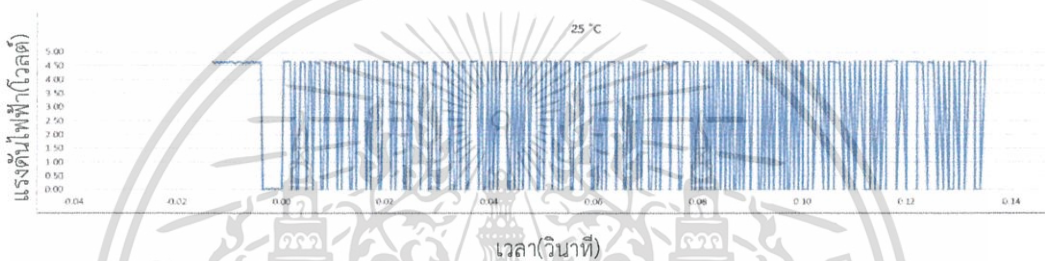


รูปที่ 4.4 กราฟลักษณะบิตสัญญาณที่โหมด OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



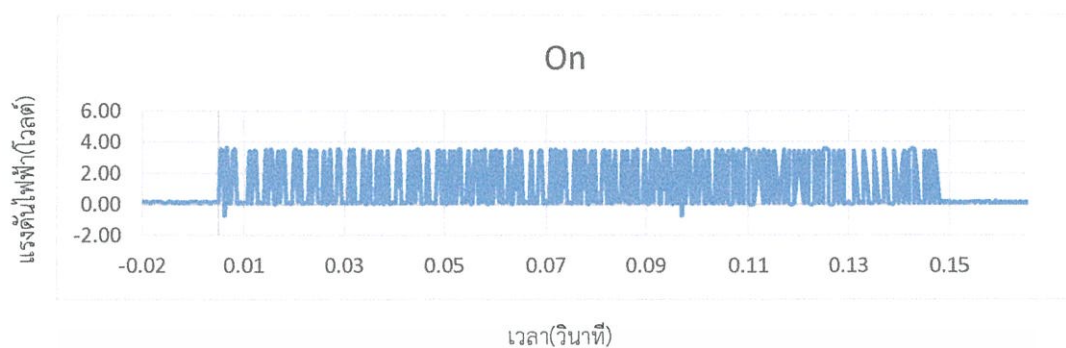
รูปที่ 4.5 กราฟลักษณะบิตสัญญาณที่อุณหภูมิ 16 °C



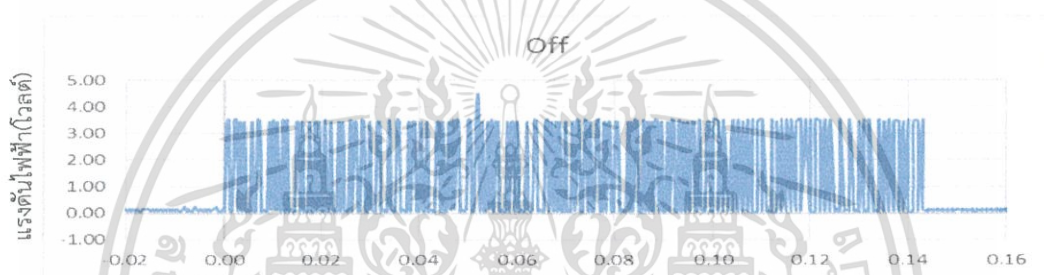
รูปที่ 4.6 กราฟลักษณะบิตสัญญาณที่อุณหภูมิ 25 °C

4.6.2 เก็บค่าบิตแพทเทิร์นจากรีโมทคอนโทรลด้วย Receiver TSOP4838 ทางฝั่งส่งของโหมดต่างๆ ดังนี้

ทำการเก็บบิตแพทเทิร์นจากตัวรับอินฟราเรดด้วย Receiver TSOP4838 จากสัญญาณที่ส่งออกจาก IR led ซึ่งสร้างโดย Raspberry Pi เพื่อศึกษาส่วนประกอบ และคุณลักษณะสัญญาณอินฟราเรด แสดงดังรูปที่ 4.7 ถึง 4.10 และกำหนด interrupt routine ทุกๆ 50 ไมโครวินาที



รูปที่ 4.7 กราฟลักษณะบิตสัญญาณที่โหมด on

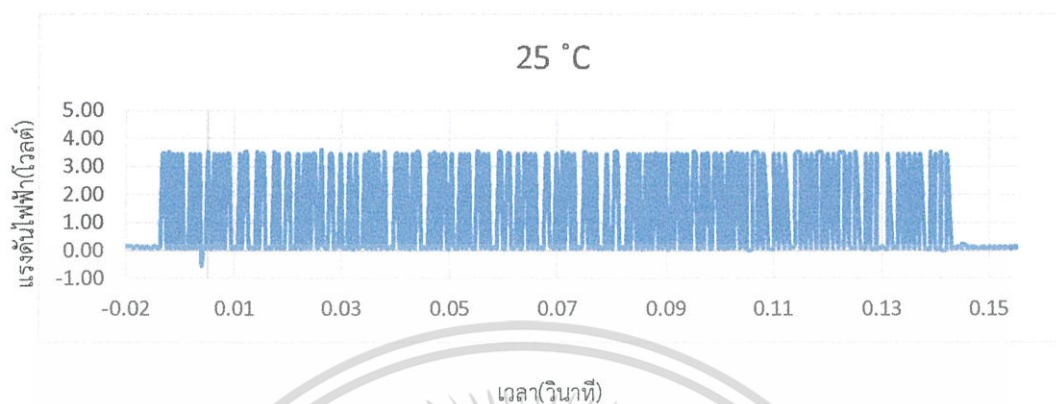


รูปที่ 4.8 กราฟลักษณะบิตสัญญาณที่โหมด off



รูปที่ 4.9 กราฟลักษณะบิตสัญญาณที่โหมด 16 °C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 กราฟลักษณะบิตสัญญาณที่โหมด 25 °C

ในเครื่องปรับอากาศยี่ห้ออื่นๆ ลักษณะบิตสัญญาณที่ทำการรับได้จะมีลักษณะที่แตกต่างกันโดยสิ้นเชิง โดยจะมีความยาวของสัญญาณที่แตกต่างกัน, รูปแบบบิต header และรูปแบบบิตแพทเทิร์นที่บอกคุณลักษณะของโหมดการทำงาน หรืออุณหภูมิ ตำแหน่ง CRC หรือ Cyclic Redundancy Check

ทำการแปลงสัญญาณที่ได้จากออสซิลโลสโคปให้อยู่ในรูปของบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 เพื่อง่ายต่อการสังเกตความแตกต่างของบิตแพทเทิร์นในแต่ละคำสั่งที่ใช้งาน ดังตารางที่ 4.7 ถึง 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 ตั้งแต่บิตที่ 1 ถึง 15

Temperature	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	ON	OFF
Header	3850	3850	3850	3850	3850	3850	3850	3850	3850	3850	3850	3850	3850	3850	3850	3900	3900
1	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

ตารางที่ 4.8 ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 ตั้งแต่บิตที่ 16 ถึง 32

16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 ตั้งแต่บิตที่ 33 ถึง 50

33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
37	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1
38	1	1	1	1	0	0	0	0	1	1	1	1	0	1	0	0
39	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
42	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
43	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
44	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
48	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
49	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ตารางที่ 4.10 ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 ตั้งแต่บิตที่ 51 ถึง 68

51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
53	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
56	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
57	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
58	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 ตั้งแต่บิตที่ 69 ถึง 86

69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ตารางที่ 4.12 ค่าบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 ตั้งแต่บิตที่ 87 ถึง 106

87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
96	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
97	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
98	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
99	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
103	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	1
104	1	1	0	0	1	0	1	0	1	0	0	1	1	0	1	0
105	0	0	0	0	1	1	0	1	0	0	0	1	1	1	0	0
106	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำตารางที่ 4.7 ถึง 4.12 มาเขียนในรูปเฟรมบิตข้อมูลแสดงดังรูปที่ 4.11

	16-30 °C		เปิด-ปิด		CRC	
บิตที่	36	39	45	47	102	106

รูปที่ 4.11 เฟรมบิตข้อมูลสัญญาณเรโมทเครื่องปรับอากาศยี่ห้อ sharp

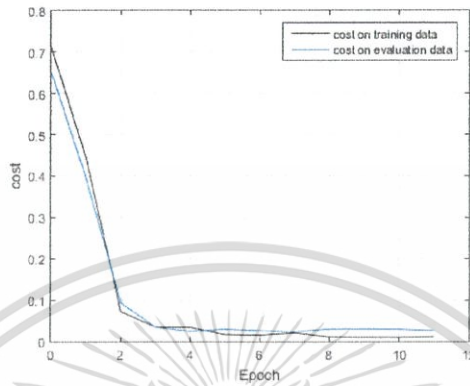
โดยจกตารางบิตแพทเทิร์นที่มีลักษณะเป็น 0, 1 จะเห็นได้ว่าในส่วนอุณหภูมิ 16 ถึง 30 องศาเซลเซียส ตำแหน่งบิตแพทเทิร์นที่ 36 ถึง 39 และ 102 ถึง 106 นั้นมีความแตกต่างกัน ซึ่งในตำแหน่งบิตแพทเทิร์นที่ 36 ถึง 39 แตกต่างกันเป็นตัวกำหนดคุณลักษณะของแต่ละอุณหภูมิ และตำแหน่งบิตแพทเทิร์นที่ 102 ถึง 106 นั้นเป็นตำแหน่งที่บอกถึง Cyclic Redundancy Check หรือ CRC

ในส่วนของโหมด On และ Off ตำแหน่งบิตแพทเทิร์นที่ 36 ถึง 39 จะมีค่าเหมือนกัน แต่ตำแหน่งบิตแพทเทิร์นที่ 45 ถึง 47 จะเป็นตำแหน่งที่บิตแพทเทิร์นแตกต่าง ซึ่งเป็นตัวกำหนดคุณลักษณะของแต่ละโหมดการทำงาน และตำแหน่งบิตแพทเทิร์นที่ 102 ถึง 106 นั้นเป็นตำแหน่งที่บอกถึง Cyclic Redundancy Check หรือ CRC

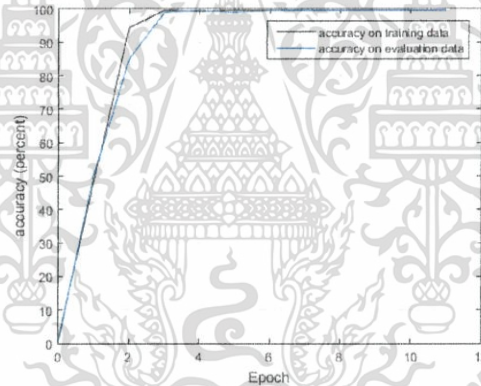
4.7 ทดสอบประสิทธิภาพโมเดลโครงข่ายประสาทเทียม

จากนั้นทำการหาค่า Cross-entropy cost และความแม่นยำ (accuracy) ในการวิเคราะห์สถานะเครื่องปรับอากาศที่เป็นอยู่โดยคิดในหน่วยเปอร์เซ็นต์ เพื่อทดสอบประสิทธิภาพหลังจากที่โครงข่ายประสาทเทียมได้เรียนรู้และอัปเดตค่า weight และ bias จากเวลา, อุณหภูมิ และความชื้น ทั้งกรณีที่ใช้งานเครื่องปรับอากาศ และไม่ได้ใช้งานเครื่องปรับอากาศโดยกำหนดให้

เรียนรู้และอัปเดตค่า weight และ bias จาก training data จำนวน 20,000 ข้อมูล และเรียนรู้ 12 รอบ ซึ่งค่า Cross-entropy cost function และความแม่นยำที่ได้แสดงดังรูปที่ 4.12 และ 4.13



รูปที่ 4.12 ค่า cost ที่ได้จากการเรียนรู้ weight และ bias



รูปที่ 4.13 ค่า accuracy ที่ได้จากการเรียนรู้ weight และ bias

จะเห็นได้ว่าในแต่ละรอบของจากการเรียนรู้ และอัปเดตค่า weight และ bias ส่งผลให้ค่า cross-entropy cost (ในรูปที่ 4.12 จะเรียก cross-entropy cost ว่า cost on training data) มีค่าลดลงจนเข้าใกล้ 0 มากขึ้น และมีความแม่นยำเพิ่มมากขึ้นโดยสถานะถูกต้องสูงสุดอยู่ที่ 99.5 เปอร์เซ็นต์ ส่งผลให้เอาต์พุตจากโครงข่ายประสาทเทียมมีค่าใกล้เคียงกับเอาต์พุตที่หวังไว้ และสามารถบอกได้ว่า ณ ปัจจุบันเครื่องปรับอากาศกำลังถูกใช้งานอยู่หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 ประมาณค่าเวลาที่เครื่องปรับอากาศใช้

ทำการเปิดเครื่องปรับอากาศจำนวน 9 ชั่วโมง 40 นาที โดยเริ่มตั้งเวลา 21.20 น. ถึง 7.00 น. หากเทียบเวลาการใช้งานเครื่องปรับอากาศนี้กับโปรแกรมที่ได้จัดทำจะมีผลลัพธ์เป็นสัญลักษณ์สีเขียวเนื่องจากใช้งานเครื่องปรับอากาศต่ำกว่า 10 ชั่วโมง ต่อ 1 วัน หากวัดผลเทียบกับจำนวนเวลาที่ประมาณจากโปรแกรมจะได้ผลดังรูปที่ 4.14, 4.15 และ 4.16

<input type="checkbox"/>	 Edit	 Copy	 Delete	1705	On	2017-04-02 07:01:17
<input type="checkbox"/>	 Edit	 Copy	 Delete	1706	Off	2017-04-02 07:02:07

รูปที่ 4.14 เวลาเมื่อเริ่มตรวจจับสถานะเริ่มและสถานะหยุดการทำงานของเครื่องปรับอากาศ

```
pi@raspberrypi: ~/final_program
GNU nano 2.2.6
3939
```

รูปที่ 4.15 จำนวนชั่วโมงที่ใช้งานเครื่องปรับอากาศ

 Edit  Copy  Delete 94 green

รูปที่ 4.16 ประมาณอัตราการใช้งานเครื่องปรับอากาศ

เมื่อ Raspberry Pi เริ่มตรวจจับว่าเครื่องปรับอากาศเริ่มทำงานจึงจะเริ่มนับเวลาการใช้งานเครื่องปรับอากาศ โดยจะนับเวลาการทำงานไปอย่างต่อเนื่องในรูปชั่วโมง จนกว่าจะตรวจจับว่าเครื่องปรับอากาศหยุดการทำงานจึงจะหยุดนับเวลาการทำงาน และจะรีเซ็ตค่าเวลาทุกๆสิ้นเดือน ซึ่งในกรณีนี้ประมาณเวลาการทำงานของเครื่องปรับอากาศได้เป็น 9 ชั่วโมง 39 นาที ดังรูปที่ 4.16 และแสดงผลเป็นสัญลักษณ์สีเขียวดังรูปที่ 4.17 ซึ่งใกล้เคียงกับเวลาการใช้งานเครื่องปรับอากาศจริง และมีสัญลักษณ์ที่ตรงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการทดลองในทำนองเดียวกับข้างต้นโดยทดลองจำนวน 5 ครั้ง และทำการหาค่าความผิดพลาดในการสั่งการ ผลลัพธ์ที่ได้แสดงดังตารางที่ 4.13

ตารางที่ 4.13 ประมาณระยะเวลาที่เครื่องปรับอากาศใช้พลังงานในรูปสัญลักษณ์

สัญลักษณ์	จำนวนครั้งที่จำลอง สถานการณ์ตาม สัญลักษณ์	จำนวนครั้งที่แสดง สัญลักษณ์ถูกต้อง	เปอร์เซ็นต์ความ ผิดพลาด
แดง	5	5	0
เหลือง	5	5	0
เขียว	5	5	0

จากตารางที่ 4.13 สรุปได้ว่าการประมาณระยะเวลาที่เครื่องปรับอากาศใช้พลังงานโดยโปรแกรมไม่เกิดความผิดพลาด

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงการเรื่องการควบคุมเครื่องปรับอากาศผ่านระบบอินเทอร์เน็ตมีการจัดทำสองส่วน ได้แก่ อินเทอร์เน็ต และ core process ในส่วนอินเทอร์เน็ตได้จัดทำแอนดรอยด์แอปพลิเคชัน และ เว็บแอปพลิเคชัน ในส่วน core process ได้ใช้ Raspberry PI เป็น processor หลัก นอกจากนี้ยังมีการใช้โปรแกรม Appserv เพื่อจำลอง Raspberry PI เป็นเซิร์ฟเวอร์เพื่อเป็นสื่อกลางในการเก็บคำสั่งการสั่งการต่างๆ เช่น คำสั่งการสั่งงานเครื่องปรับอากาศ, คัดลอกสัญญาณรีโมทเครื่องปรับอากาศ หรือคำสั่งเรียกใช้ฟังก์ชันสมาร์ท ในการควบคุมเครื่องปรับอากาศ Raspberry PI จะส่งค่าบิตแพทเทิร์นให้อินฟราเรดจากนั้นอินฟราเรดจะส่งค่าบิตแพทเทิร์นไปยังเครื่องปรับอากาศ นอกจากนี้ยังมีเซนเซอร์วัดอุณหภูมิ DHT22 เพื่อตรวจสอบว่าอุณหภูมิโดยรอบมีการเปลี่ยนแปลงตามการสั่งการจากแอนดรอยด์แอปพลิเคชันหรือไม่ และค่าอุณหภูมิที่ได้จากเซนเซอร์วัดอุณหภูมิ DHT22 ยังถูกนำไปวิเคราะห์ผ่าน Neural Network เพื่อช่วยตรวจสอบสถานะการทำงานของเครื่องปรับอากาศให้แม่นยำยิ่งขึ้น และสถานะที่ได้จะถูกใช้ในฟังก์ชันประมาณการใช้พลังงานไฟฟ้าของเครื่องปรับอากาศต่อเดือนอีกด้วย นอกจากนี้ยังมีฟังก์ชันสมาร์ทที่เปิด หรือ ปิดเครื่องปรับอากาศแบบอัตโนมัติตามระยะเวลาของผู้ใช้งาน และแนะนำอุณหภูมิให้แก่ผู้ใช้งาน

โครงการการควบคุมเครื่องปรับอากาศผ่านระบบอินเทอร์เน็ตสามารถนำไปใช้งานได้จริง เนื่องจาก Raspberry PI เป็นอุปกรณ์ที่สามารถหาใช้งานได้ง่าย สามารถนำไปประยุกต์ใช้งานได้หลากหลาย

5.2 ข้อเสนอแนะ

1. ต้องทำ forward port ก่อนการใช้งานภายนอกวงแลน
2. ในส่วนนี้สามารถใช้งานร่วมกับเครื่องปรับอากาศ 3 ยี่ห้อเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Raspberry Pi [ออนไลน์].เข้าถึงได้จาก

<https://www.raspberrypi.org/weekly/connected/>(วันที่สืบค้น: 21 ตุลาคม 2559).

[2] Raspberry Pi Pinout [ออนไลน์].เข้าถึงได้จาก

<https://www.elektronik-kompodium.de/sites/raspberry-pi/1912111.htm>

(วันที่สืบค้น21 ตุลาคม 2559).

[3] Raspberry Pi 2 & 3 Pin Mappings [ออนไลน์].เข้าถึงได้จาก

<https://developer.microsoft.com/> (วันที่สืบค้น: 21 ตุลาคม 2559).

[4] Standard Wi-Fi [ออนไลน์].เข้าถึงได้จาก

https://th.wikipedia.org/wiki/IEEE_802.11 (วันที่สืบค้น: 3 พฤศจิกายน 2559).

[5] TSOP4838 [ออนไลน์].เข้าถึงได้จาก

[http://leksykon.forbot.pl/TSOP_\(np._TSOP1736\),137.htm](http://leksykon.forbot.pl/TSOP_(np._TSOP1736),137.htm) (วันที่สืบค้น: 5 พฤศจิกายน 2559).

[6] Modulation IR LED [ออนไลน์].เข้าถึงได้จาก

<http://www.sbprojects.com/knowledge/ir/index.php> (วันที่สืบค้น: 5 พฤศจิกายน 2559).

[7] Artificial neural network [ออนไลน์].เข้าถึงได้จาก

<https://th.wikipedia.org/wiki> (วันที่สืบค้น: 12 มกราคม 2560).

[8] Raspberry Pi 3 picture [ออนไลน์].เข้าถึงได้จาก

<https://www.peterdavehello.org/tag/linux/> (วันที่สืบค้น: 5 พฤศจิกายน 2559).

[9] DHT22 [ออนไลน์].เข้าถึงได้จาก

<http://www.electroschematics.com> (วันที่สืบค้น: 24 พฤศจิกายน 2559).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[10] Transistor PNP 2N2222 [ออนไลน์].เข้าถึงได้จาก

<http://www.amazon.com/eFuture-TM-Set-100Pcs-Transistor> (วันที่สืบค้น: 24 พฤศจิกายน 2559).

[11] IR LED [ออนไลน์].เข้าถึงได้จาก

<https://www.adafruit.com/product/387> (วันที่สืบค้น: 24 พฤศจิกายน 2559)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Measure temperature

```

import MySQLdb

import Adafruit_DHT as dht

import time

import datetime

conn = MySQLdb.connect(host="localhost",

                        user="root",

                        passwd="1231043h",

                        db="sitepoint")

x = conn.cursor()

while True:

    h,t = dht.read_retry(dht.DHT22, 4) #measure indoor temperature by DHT22 sensors.

    t_bias = t-1

    ts = time.time()

    timestamp = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S') #

get current timestamp

    print 'Temp={0:f}*C Humidity={1:f}%'.format(t_bias, h)

    x.execute("""INSERT INTO dht22 VALUES (%s,%s,%s,%s)""",(0,t_bias,h,timestamp)) #insert

the temperature.

    conn.commit()

    time.sleep(40)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Neural network code

"""

~~~~~

A module to implement the stochastic gradient descent learning algorithm for a feedforward neural network. Gradients are calculated using backpropagation. Note that I have focused on making the code simple, easily readable, and easily modifiable. It is not optimized, and omits many desirable features.

"""

#### Libraries

# Standard library

import json

import random

import sys

# Third-party libraries

import numpy as np

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class CrossEntropyCost(object):
```

```
    @staticmethod
```

```
    def fn(a, y):
```

```
        """Return the cost associated with an output ``a`` and desired output
        ``y``. Note that np.nan_to_num is used to ensure numerical
        stability. In particular, if both ``a`` and ``y`` have a 1.0
        in the same slot, then the expression (1-y)*np.log(1-a)
        returns nan. The np.nan_to_num ensures that that is converted
        to the correct value (0.0).
        """
```

```
        return np.sum(np.nan_to_num(-y*np.log(a)-(1-y)*np.log(1-a)))
```

```
    @staticmethod
```

```
    def delta(z, a, y):
```

```
        """Return the error delta from the output layer. Note that the
        parameter ``z`` is not used by the method. It is included in
        the method's parameters in order to make the interface
        consistent with the delta method for other cost classes.
        """
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return (a-y)

#### Main Network class

class Network(object):

    def __init__(self, sizes, cost=CrossEntropyCost):

        """The list ``sizes`` contains the number of neurons in the
        respective layers of the network.  For example, if the list
        was [2, 3, 1] then it would be a three-layer network, with the
        first layer containing 2 neurons, the second layer 3 neurons,
        and the third layer 1 neuron.  The biases and weights for the
        network are initialized randomly, using a Gaussian
        distribution with mean 0, and variance 1.  Note that the first
        layer is assumed to be an input layer, and by convention we
        won't set any biases for those neurons, since biases are only
        ever used in computing the outputs from later layers."""

        self.num_layers = len(sizes)

        print self.num_layers

        self.sizes = sizes

        self.cost=cost

        self.biases = [np.random.randn(y, 1) for y in self.sizes[1:]]

        self.weights = [np.random.randn(y, x)/np.sqrt(x)

            for x, y in zip(self.sizes[:-1], self.sizes[1:])]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def feedforward(self, a):

    """Return the output of the network if ``a`` is input."""

    for b, w in zip(self.biases, self.weights): #map weight and bias on each layer

        a = sigmoid(np.dot(w, a) + b)

        #print '\n'

        #print a

    return a

def feedforward_x(self, a):

    net = load('filename')

    print net.weights

    for b, w in zip(net.biases, net.weights): #map weight and bias on each layer

        a = sigmoid(np.dot(w, a) + b)

        if (a <= 0.4).any(): #compare the value in the numpy matrix

            u = np.zeros((1))

        elif (a >= 0.6).any():

            u = np.array([[1]])

        else:

            u = 'distortion'

    print u

    return u

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
def SGD(self, training_data, evaluation_data, epochs, mini_batch_size, eta,
```

```
    lambda = 0.0,
```

```
    monitor_training_cost=False,
```

```
    monitor_training_accuracy=False,
```

```
    monitor_evaluation_accuracy=False,
```

```
    monitor_evaluation_cost=False):
```

```
    """Train the neural network using mini-batch stochastic
    gradient descent. The ``training_data`` is a list of tuples
    ``(x, y)`` representing the training inputs and the desired
    outputs. The other non-optional parameters are
    self-explanatory. If ``test_data`` is provided then the
    network will be evaluated against the test data after each
    epoch, and partial progress printed out. This is useful for
    tracking progress, but slows things down substantially."""
```

```
    #if evaluation_data: n_data = len(evaluation_data)
```

```
    n_data = len(evaluation_data)
```

```
    n = len(training_data)
```

```
    training_cost, training_accuracy = [], []
```

```
    evaluation_accuracy, evaluation_cost = [], []
```

```
    for j in xrange(epochs): # xrange = increase each time by 1
```

```
        random.shuffle(training_data)
```

```
        mini_batches = [
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

training_data[k:k+mini_batch_size] #word[0:2] characters from position 0
(included) to 2 (excluded)

for k in xrange(0, n, mini_batch_size)]

for mini_batch in mini_batches:

    #print mini_batch

    self.update_mini_batch(

        mini_batch, eta, lambda, len(training_data)) #get the update weights and bias
of every layer

print "Epoch %s training complete" % j

if monitor_training_cost:
    cost = self.total_cost(training_data, lambda)
    training_cost.append(cost)
    print "Cost on training data: {}".format(cost)
if monitor_training_accuracy:
    accuracy = self.accuracy(training_data)
    training_accuracy.append(accuracy)
    print "Accuracy on training data: {} / {}".format(
        accuracy, n)

if monitor_evaluation_cost:
    cost = self.total_cost(evaluation_data, lambda)
    evaluation_cost.append(cost)
    print "Cost on evaluation data: {}".format(cost)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if monitor_evaluation_accuracy:
    accuracy = self.accuracy(evaluation_data)
    evaluation_accuracy.append(accuracy)
    print "Accuracy on evaluation data: {} / {}".format(
        self.accuracy(evaluation_data), n_data)

```

```

return training_cost,evaluation_accuracy,training_accuracy

```

```

def update_mini_batch(self, mini_batch, eta, lmbda, n):
    """Update the network's weights and biases by applying
    gradient descent using backpropagation to a single mini batch.
    The ``mini_batch`` is a list of tuples ``(x, y)``, and ``eta``
    is the learning rate."""
    #print self.biases
    nabla_b = [np.zeros(b.shape) for b in self.biases]
    nabla_w = [np.zeros(w.shape) for w in self.weights]
    for x, y in mini_batch:
        delta_nabla_b, delta_nabla_w = self.backprop(x, y) #receive partial derivative from
        backprop
        nabla_b = [nb+dnb for nb, dnb in zip(nabla_b, delta_nabla_b)]
        nabla_w = [nw+dnw for nw, dnw in zip(nabla_w, delta_nabla_w)]
    self.weights = [(1-eta*(lmbda/n))*w-(eta/len(mini_batch))*nw

        for w, nw in zip(self.weights, nabla_w)]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

self.biases = [b-(eta/len(mini_batch))*nb
                for b, nb in zip(self.biases, nabla_b)]

def backprop(self, x, y):
    """Return a tuple ``(nabla_b, nabla_w)`` representing the
    gradient for the cost function C_x. ``nabla_b`` and
    ``nabla_w`` are layer-by-layer lists of numpy arrays, similar
    to ``self.biases`` and ``self.weights``."""
    nabla_b = [np.zeros(b.shape) for b in self.biases]
    nabla_w = [np.zeros(w.shape) for w in self.weights]
    # feedforward
    activation = x
    activations = [x] # list to store all the activations, layer by layer
    zs = [] # list to store all the z vectors, layer by layer
    activation = np.reshape(activation, (3, 1))
    for b, w in zip(self.biases, self.weights):
        z = np.dot(w, activation)+b
        zs.append(z)
        activation = sigmoid(z)
        activations.append(activation)
    # backward pass
    delta = (self.cost).delta(zs[-1], activations[-1], y) #cross entropy
    nabla_b[-1] = delta # thrid equation of backprob

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nabla_w[-1] = np.dot(delta, activations[-2].transpose()) # forth equation of backprob

# Note that the variable l in the loop below is used a little
# differently to the notation in Chapter 2 of the book. Here,
# l = 1 means the last layer of neurons, l = 2 is the
# second-last layer, and so on. It's a renumbering of the
# scheme in the book, used here to take advantage of the fact
# that Python can use negative indices in lists.
```

```
for l in xrange(2, self.num_layers): #start second equation of backprop
```

```
    z = zs[-l]
```

```
    sp = sigmoid_prime(z)
```

```
    delta = np.dot(self.weights[-l+1].transpose(), delta) * sp
```

```
    nabla_b[-l] = delta
```

```
    nabla_w[-l] = np.dot(delta, activations[-l-1].transpose())
```

```
return (nabla_b, nabla_w)
```

```
def cost_derivative(self, output_activations, y):
```

```
    """Return the vector of partial derivatives  $\partial C_x /$ 
```

```
     $\partial a$  for the output activations."""
```

```
    return (output_activations-y)
```

```
def total_cost(self, data, lmbda, convert=False):
```

```
    """Return the total cost for the data set ``data``. The flag
```

```
    ``convert`` should be set to False if the data set is the
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

training data (the usual case), and to True if the data set is the validation or test data. See comments on the similar (but reversed) convention for the ``accuracy`` method, above.

```
"""
```

```
cost = 0.0
```

```
for x, y in data:
```

```
    a = self.feedforward(x)
```

```
    #print a
```

```
    if convert: y = vectorized_result(y)
```

```
    cost += self.cost.fn(a, y)/len(data)
```

```
    cost += 0.5*(lmbda/len(data))*sum(
        np.linalg.norm(w)**2 for w in self.weights)
```

```
return cost
```

```
def accuracy(self, data):
```

```
    """Return the number of inputs in ``data`` for which the neural
    network outputs the correct result. The neural network's
    output is assumed to be the index of whichever neuron in the
    final layer has the highest activation.
```

The flag ``convert`` should be set to False if the data set is validation or test data (the usual case), and to True if the

data set is the training data. The need for this flag arises

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

due to differences in the way the results ``y`` are represented in the different data sets. In particular, it flags whether we need to convert between the different representations. It may seem strange to use different representations for the different data sets. Why not use the same representation for all three data sets? It's done for efficiency reasons – the program usually evaluates the cost on the training data and the accuracy on other data sets.

These are different types of computations, and using different representations speeds things up. More details on the representations can be found in `mnist_loader.load_data_wrapper`.

"""

```
#results = [(np.argmax(self.feedforward(x)), y)
```

```
results = [(self.estimate(x), y) for(x, y) in data]
```

```
return sum(int(x == y) for (x, y) in results)
```

```
def estimate(self, x):
```

```
    """estimate the output of network into binary bits"""
```

```
    a = self.feedforward(x) #let the input 'x' through the NN.
```

```
    if (a <= 0.2).any(): #compare the value in the numpy matrix
```

```
        x = np.zeros((1)) # create array with zero value.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

elif (a >= 0.8).any():

    x = np.array([[1]]) #create array with one value.

else:

    x = 'miscarry'

return x

```

```

def save(self, filename):

```

```

    """Save the neural network to the file `filename`."""

```

```

    data = {"sizes": self.sizes,

```

```

           "weights": [w.tolist() for w in self.weights],

```

```

           "biases": [b.tolist() for b in self.biases],

```

```

           "cost": str(self.cost.__name__)}

```

```

    f = open(filename + ".w")

```

```

    json.dump(data, f)

```

```

    f.close()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
##### Loading a Network
```

```
def load(filename):
```

```
    """Load a neural network from the file ``filename``. Returns an
    instance of Network.
```

```
    """
```

```
    f = open('filename', 'r')
```

```
    data = json.load(f)
```

```
    f.close()
```

```
    cost = getattr(sys.modules[__name__], data["cost"])
```

```
    net = Network(data["sizes"], cost=cost)
```

```
    net.weights = [np.array(w) for w in data["weights"]]
```

```
    net.biases = [np.array(b) for b in data["biases"]]
```

```
    return net
```

```
##### Miscellaneous functions
```

```
def sigmoid(z):
```

```
    """The sigmoid function."""
```

```
    return 1.0/(1.0+np.exp(-z))
```

```
def sigmoid_prime(z):
```

```
    """Derivative of the sigmoid function."""
```

```
    return sigmoid(z)*(1-sigmoid(z))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Suggest Temp

```

import MySQLdb

import Adafruit_DHT as dht

import pyowm

conn = MySQLdb.connect(host= "localhost",
                        user="root",
                        passwd="1231043h",
                        db="sitepoint")

x = conn.cursor()

def temp_api(): # receive outdoor temperature api
    owm = pyowm.OWM('5126ba6c540ffc58a9c314c8000a031f')
    observation = owm.weather_at_place('Bangkok, TH')
    w = observation.get_weather()

    temp_out = w.get_temperature('celsius')['temp']

    print temp_out

    return temp_out

```

while True:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

try:

```
h,t = dht.read_retry(dht.DHT22, 4)
```

print 'Temp={0:f}\*C Humidity={1:f}%' .format(t, h) #measure indoor temperature by DHT22 sensors.

```
temp_out = temp_api()
```

```
if (t-temp_out) >= 1.5 and t < 27:
```

```
    temp_s = round((t+temp_out)/2) #find the mean of two temps.
```

```
    print temp_s
```

```
    x.execute("""INSERT INTO suggest VALUES (%s,%s)""",(0,temp_s+4))
```

```
        print 'store %d celcius'%(temp_s+4)
```

```
    conn.commit()
```

```
    time.sleep(240)
```

```
    x.execute("""INSERT INTO suggest VALUES (%s,%s)""",(0,temp_s+3))
```

```
        print 'store %d celcius'%(temp_s+3)
```

```
    conn.commit()
```

```
    time.sleep(240)
```

```
    x.execute("""INSERT INTO suggest VALUES (%s,%s)""",(0,temp_s+2))
```

```
        print 'store %d celcius'%(temp_s+2)
```

```
    conn.commit()
```

```
    time.sleep(30)
```

```
    x.execute("""INSERT INTO suggest VALUES (%s,%s)""",(0,'0'))
```

```
    conn.commit()
```

else:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
x.execute("""INSERT INTO suggest VALUES (%s,%s)""",(0,'0'))
conn.commit()

print 'no suggestion'
```

except:

```
print('nope');
```

```
time.sleep(25)
```

```
conn.close()
```

ได้ประมาณค่าไฟต่อเดือน

```
import MySQLdb
```

```
import datetime
```

```
import time
```

```
import electricity_consump
```

```
from decimal import *
```

```
#diff_total = '0'
```

```
#print diff_total
```

```
diff_total = float('0.0')
```

```
conn = MySQLdb.connect(host= "localhost",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

user="root",

passwd="1231043h",

db="sitepoint")

```

```
x = conn.cursor()
```

```
def load_data():
```

```
    # Open database connection
```

```
    db = MySQLdb.connect(host= "localhost",
```

```
        user="root",
```

```
        passwd="1231043h",
```

```
        db="sitepoint")
```

```
    # prepare a cursor object using cursor() method
```

```
    cursor = db.cursor()
```

```
    cursor2 = db.cursor()
```

```
    cursor3 = db.cursor()
```

```
    # Prepare SQL query to INSERT a record into the database.
```

```
    cursor.execute("SELECT @smart_id2 := MAX(ID) FROM smart")
```

```
    cursor.execute("SELECT * FROM smart WHERE id = @smart_id2 ")
```

```
    cursor2.execute("SELECT @smart_id := MAX(ID)-1 FROM smart")
```

```
    cursor2.execute("SELECT * FROM smart WHERE ID = @smart_id ")
```

```
    cursor3.execute("SELECT @smart_id := MAX(ID) FROM btu")
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cursor3.execute("SELECT * FROM btu WHERE ID = @smart_id ")

data = cursor.fetchall() #MaxID

data2 = cursor2.fetchall() #MaxID -1

data3 = cursor3.fetchall() #BTU

print data,data2,data3

db.close()

return data,data2,data3

while True:

    data,data2,data3 = electricity_consump.load_data() #get data from the database

    now = datetime.datetime.now() # current time

    month_n = Decimal(now.month)

    day_n = Decimal(now.day)

    year_n = now.year

    for row in data:

        status_Maxid = row[1] #the latest status

        end = row[2] #the latest time

        insert_time = end.strftime('%H:%M:%S')

        print status_Maxid,end

        print day_n

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for row in data2:

    status_Maxid_1 = row[1] #the almost lastest status

    start = row[2] #the almost lastest time

    month_d = Decimal(end.strftime('%m'))

    print status_Maxid_1,start

```

```

for row in data3:

    BTU = row[1] #BTU

    EER = 11.6

    print BTU,EER

if status_Maxid == 'On' and status_Maxid == status_Maxid_1:

    diff = end - start

    th = float(diff.total_seconds()/60/60) #duration that user turn on A/C

    diff_total += th #total time in hour

    print diff_total

##### BTU to watt #####

power = BTU / EER

unit_per_hour = power*0.85/1000 #total power

print power,unit_per_hour

```

#####

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

diff_total_s = '%s'%(diff_total)#change type from float into string
a = "statistic_%s_%s.txt" %(month_n,year_n)

if month_n == month_d:

    file = open(a,"w") #write diff_total in the notepad file.

    file.write(diff_total_s)

    file.close()

    total_unit = unit_per_hour*diff_total

    x.execute("""INSERT INTO electrical VALUES (%s,%s)""",(0,total_unit))

    print total_unit

    conn.commit()

elif month_n != month_d: #when start new month

    diff_total = float('0.0') #reset the unit.

else:

    print 'unexpect happened'

else:

    print 'not calculate'

time.sleep(50) #run every 50 sec

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โค้ดสั่งการแอร์ nodeJS

```
#!/usr/bin/env node

require('shelljs/global');

var cronJob = require('cron').CronJob;

//run every 5 sec

var prepend = require('prepend');

var b;

var test = require('./sleep');

var myJob = new cronJob('*/*/* * * * *', function() {

var test = require('./sleep');

var exec = require('child_process').exec;

var mysql = require("mysql");

var fs = require('fs');

// First you need to create a connection to the db

var con = mysql.createConnection({

  host: 'localhost',

  user: 'root',

  password: '1231043h',

  database: 'sitepoint',
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

multipleStatements: true
});
con.connect(function(err){
  if(err){
    console.log('Error connecting to Db');//show error
    return;
  }
  console.log('Connection established');
});
con.query('CALL sp_get_employee2;',function (err,rows){
  if (err) throw err;
  for (var i = 0; i < rows[1].length; i++) {
    var mode = rows[1][i].Mode;
    var brand = rows[1][i].Brand;
    var value = rows[1][i].Value;
  };
  console.log(mode);
  console.log(brand);
  console.log(value);
  console.log(b);
  var check = `${brand} ${value}`;

```

```

if(check != b){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//check that ls data the same.

if(mode == '0'){
echo("sudo /etc/init.d/lirc start");
exec("sudo /etc/init.d/lirc start");//make module ready before send any infrared

var cmd = `irsend SEND_ONCE ${brand} ${value}`;
console.log(cmd);
exec(cmd, function(error, stdout, stderr) {
console.log('stdout: ' + stdout);// command output is in stdout
console.log('stderr: ' + stderr);//command error
});//exec and get respond
}
else if(mode == '1'){
echo("sudo /etc/init.d/lirc stop");
exec("sudo /etc/init.d/lirc stop");
cd('/home/pi/testty');
exec('mkdir -p '+brand);//create Brand directory
echo('mkdir -p '+brand);
cd('${brand}/');
console.log(`cd ${brand}/`);
cp('-f', '/home/pi/testty/CentralAir/trailer', '/home/pi/testty/${brand}');
cp('-f', '/home/pi/testty/CentralAir/header', '/home/pi/testty/${brand}');
exec(`timeout 3 sudo mode2 -d /dev/lirc0 -m > ${value}`);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//exec(cre); //start copying

test.delay(2500); //delay

console.log('Success Copying!!');

var del = `ex -sc '1d|x' ${value}`;

exec(del); //delete the first line

console.log('deleted');
```

```
//start change files

cat('header').to(`header_${brand}`); //back up file

sed('-i', 'Daikin', `${brand}`, `header_${brand}`); //change brand

echo('cp1');

echo('cp2');

var data = fs.readFileSync(`${value}`).toString().split("\n");

data.splice('lineNumber', 0, `name ${value}`);

var text = data.join("\n");

fs.writeFileSync(`${value}`, text);

cat(`header_${brand}`, `${value}`, 'trailer').toEnd(`lircd_${brand}.conf`);

cp('-f', `lircd_${brand}.conf`, '/etc/lirc/');

cd('~');

cd('/etc/lirc/');
```

```
prepend('lircd.conf', `include "lircd_${brand}.conf"`, function(error) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (error)
    console.error(error.message);
}); //add new remote

exec('sudo /etc/init.d/lirc restart');
    }

    else{console.log('no support')}
    }

    else{console.log('SleepZzz')}
    b = `${brand} ${value}`;
    console.log(b);
    }
    );

con.end(function(err) {
    // The connection is terminated gracefully
    // Ensures all previously enqueued queries are still
    // before sending a COM_QUIT packet to the MySQL server.
});

});

myJob.start();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้