

การควบคุมอุปกรณ์รับสัญญาณอินฟราเรดผ่าน Wi-Fi

INFRARED DEVICES CONTROLLER VIA WI-FI

โดย

นางสาวเนตรลัดดา	เจริญพันธ์
นางสาวปณัฏตา	บุญญาศิริ
นางสาวปรารธนา	เดาปัญญา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การควบคุมอุปกรณ์รับสัญญาณอินฟราเรดผ่าน Wi-Fi
INFRARED DEVICES CONTROLLER VIA WI-FI

โดย

นางสาวเนตรลัดดา	เจริญพันธ์
นางสาวปณิตตา	บุญญาศิริ
นางสาวปรารณา	เถาปัญญา

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การควบคุมอุปกรณ์รับสัญญาณอินฟราเรดผ่าน Wi-Fi

INFRARED DEVICES CONTROLLER VIA WI-FI

โดย

นางสาวเนตรลัดดา	เจริญพันธ์	53010873
นางสาวปณิตตา	บุญญาศิริ	53010911
นางสาวปรารถนา	เถาปัญญา	53010942

อาจารย์ที่ปรึกษา

ผศ.ดร.นภัทร สระเอี่ยม

อาจารย์ที่ปรึกษาร่วม

ผศ.ดร.ธเนศ พัฒนธาดาพงษ์

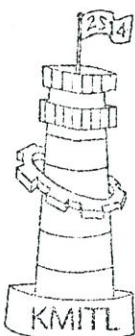
ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556



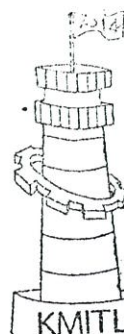
ผ่านการตรวจรูปเล่มแล้ว

()

อาจารย์ที่ปรึกษา

13 / 3 / 57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

()

กรรมการผู้ตรวจชิ้นงาน

19 / 8 / 57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมอุปกรณ์รับสัญญาณอินฟราเรดผ่าน Wi-Fi

INFRARED DEVICES CONTROLLER VIA WI-FI

ผู้จัดทำ

- | | |
|-------------------------------|----------|
| 1. นางสาวเนตรลัดดา เจริญพันธ์ | 53010873 |
| 2. นางสาวปณิตตา บุญญาศิริ | 53010911 |
| 3. นางสาวปรารถนา เถาปัญญา | 53010942 |



.....
(ผศ.ดร.นภัทร สระเอี่ยม)

อาจารย์ที่ปรึกษา



.....
(ผศ.ดร.ธเนศ พัฒนธาดาพงษ์)

อาจารย์ที่ปรึกษาร่วม

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ที่ปรึกษาโครงการ ผู้ช่วยศาสตราจารย์ ดร.นภัทร สระเอี่ยม และ ผู้ช่วยศาสตราจารย์ ดร.ธเนศ พัฒนธาดาทพงษ์ เป็นอย่างสูง ที่กรุณาให้ความรู้และคำแนะนำในการทำโครงการนี้ จนทำให้โครงการสามารถดำเนินไปตามแผนที่วางไว้ได้

ขอขอบคุณบิดามารดา ที่อุปการะเลี้ยงดูข้าพเจ้าตั้งแต่เล็กจนโต คอยอยู่เคียงข้างและเป็นกำลังใจที่สำคัญในยามที่ข้าพเจ้าพบกับปัญหา และให้การศึกษาทำให้ข้าพเจ้าได้มีโอกาสเข้ามาเรียนที่คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังแห่งนี้

ขอขอบคุณเพื่อนทุกคน ที่ให้คำปรึกษา แนะนำสิ่งดี ๆ ในการทำงาน ขอขอบคุณสำหรับ ความจริงใจที่มีต่อกันเสมอมา

นางสาวเนตรลัดดา เจริญพันธ์

นางสาวปณัตตา บุญญาศิริ

นางสาวปรารถนา เกาปัญญา

ผู้จัดทำ

การควบคุมอุปกรณ์รับสัญญาณอินฟราเรดผ่าน WI-FI
INFRARED DEVICES CONTROLLER VIA WI-FI

โดย นางสาวเนตรลัดดา เจริญพันธ์ 53010873
นางสาวปณิตตา บุญญาศิริ 53010911
นางสาวปรารถนา เถาปัญญา 53010942

อาจารย์ที่ปรึกษา ผศ.ดร.นภัทร สระเอี่ยม

อาจารย์ที่ปรึกษาร่วม ผศ.ดร.ธเนศ พัฒนธาดาพงษ์

บทคัดย่อ

โครงการนี้นำเสนอการควบคุมอุปกรณ์รับสัญญาณอินฟราเรดผ่าน WI-FI เพื่อเพิ่มความสะดวกสบายให้กับคนในบ้าน ซึ่งสามารถควบคุมเครื่องใช้ไฟฟ้าผ่านแอปพลิเคชันบน SMARTPHONE ระบบปฏิบัติการแอนดรอยด์ หลักการประกอบไปด้วย 2 ส่วนหลัก คือ ส่วนส่งการควบคุมจากโทรศัพท์ และส่วนประมวลผลเพื่อส่งสัญญาณอินฟราเรด เมื่อผู้ใช้งานเลือกอุปกรณ์ที่ต้องการควบคุมจากแอปพลิเคชันบนโทรศัพท์แล้ว คำสั่งจะถูกส่งผ่าน WI-FI ไปประมวลผลที่บอร์ด ARDUINO เพื่อแปลงคำสั่งเป็นสัญญาณอินฟราเรดส่งไปควบคุมเครื่องไฟฟ้านั้น

ABSTRACT

THIS PROJECT PRESENTS INFRARED DEVICES CONTROLLER VIA WI-FI TO CONTROL ELECTRICAL APPLIANCES IN YOUR HOUSEHOLD BY USING APPLICATION ON SMARTPHONE THAT HAS ANDROID OPERATING SYSTEM, WHICH WILL GIVE USER MORE COMFORT. IT CONSISTS OF 2 PARTS, WHICH ARE APPLICATION ON SMARTPHONE AND PROCESSOR TO TRANSMIT INFRARED SIGNAL. WHEN USER CHOOSE A DEVICE THEY WANT TO CONTROL FROM APPLICATION ON THE PHONE, IT WILL SEND COMMAND VIA WI-FI TO PROCESS AT ARDUINO BOARD, WHICH CONVERTED THE COMMAND TO INFRARED SIGNAL AND TRANSMITTED TO THE DEVICE.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	VII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 บอร์ด Arduino	3
2.1.1 รายละเอียดคุณลักษณะ	4
2.1.2 การติดตั้งก่อนใช้งาน	5
2.1.3 โปรแกรมภาษา C ของ Arduino	6
2.1.3.1 โครงสร้างโปรแกรม	6
2.1.3.2 คำสั่งในโปรแกรม	7
2.2 รีโมทคอนโทรลอินฟราเรด	13
2.2.1 แสงอินฟราเรด	13
2.2.2 การมอดูเลต (Modulation)	14
2.2.3 ตัวส่งสัญญาณ (Transmitter)	14
2.2.4 ตัวรับสัญญาณ (Receiver)	16
2.2.5 รูปแบบการส่งสัญญาณของรีโมทควบคุมโทรทัศน์	17
2.2.5.1 SAMSUNG Protocol	17

สารบัญ (ต่อ)

	หน้า
2.3 แอนดรอยด์ (Android)	19
2.3.1 สถาปัตยกรรมของแอนดรอยด์	19
2.3.1.1 Linux Kernel	19
2.3.1.2 Libraries	20
2.3.1.3 Android Runtime	21
2.3.1.4 Application Framework	22
2.3.1.5 Application	22
2.3.2 อีคลิพส์ (Eclipse)	23
2.4 โพรโตคอลที่ใช้ในการสื่อสารข้อมูล	23
2.4.1 TCP/IP (Transmission Control Protocol/Internet Protocol)	23
2.4.2 Socket	24
2.5 Wireless LAN	25
2.5.1 มาตรฐานการใช้งานเทคโนโลยี Wi-Fi	26
2.5.1.1 ช่วงความถี่ 900 MHz	26
2.5.1.2 ช่วงความถี่ 2.4 GHz	26
2.5.1.3 ช่วงความถี่ 5 GHz	26
2.5.2 ลักษณะการเชื่อมต่อของอุปกรณ์	27
2.5.2.1 โหมด Infrastructure	27
2.5.2.2 โหมด Ad-Hoc หรือ Peer-to-Peer	27
บทที่ 3 การออกแบบและการจัดทำโครงการ	28
3.1 การออกแบบ	28
3.1.1 การสร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์	28

สารบัญ (ต่อ)

	หน้า
3.1.2 การสื่อสารข้อมูลผ่าน Wi-Fi	30
3.1.3 การหาข้อมูลคำสั่งของรีโมท	32
3.1.3.1 ทารูปแบบข้อมูลของสัญญาณอินฟราเรดด้วยออสซิล- โลสโคป	32
3.1.3.2 โปรแกรมอ่านค่าและแสดงสัญญาณรีโมท	33
3.1.4 การส่งสัญญาณอินฟราเรด	35
3.1.4.1 วงจรส่งสัญญาณอินฟราเรด	35
3.1.4.2 โปรแกรมส่งสัญญาณอินฟราเรด	35
3.1.5 วงจรจ่ายไฟ	38
3.2 เครื่องมือที่ใช้ในการทดลอง	38
3.3 การจัดเก็บผลการทดลอง	42
3.3.1 การสร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการ แอนดรอยด์	42
3.3.2 การสื่อสารข้อมูลผ่าน WI-FI	42
3.3.3 การหาข้อมูลคำสั่งของรีโมท	43
3.3.4 การส่งสัญญาณอินฟราเรด	43
บทที่ 4 ผลการทดลอง	44
4.1 การสร้างแอปพลิเคชันรีโมทจำลองบนโทรศัพท์ระบบปฏิบัติการ แอนดรอยด์	44
4.1.1 ผลการทดลองโปรแกรมสร้างแอปพลิเคชันรีโมทจำลอง	44
4.1.2 ผลการทดลองการเชื่อมต่อ WI-FI ด้วย EMULATOR	45
4.2 การสื่อสารข้อมูลผ่าน WI-FI	47
4.2.1 ผลการทดลองส่งข้อมูลจากโทรศัพท์เคลื่อนที่ผ่าน WI-FI ไปยัง บอร์ด ARDUINO	47
4.3 การหาข้อมูลคำสั่งของรีโมท	51

สารบัญ (ต่อ)

	หน้า
4.3.1 ผลการทดลองวัดสัญญาณอินฟราเรดจากรีโมทด้วยออสซิล- โลสโคป	51
4.3.2 ผลการทดลองวัดความกว้างช่วงเวลาของบิตด้วยการ ประมวลผลผ่านโปรแกรม	53
4.4 การส่งสัญญาณอินฟราเรด	57
4.4.1 ผลการทดสอบโปรแกรมส่งสัญญาณอินฟราเรด	57
4.4.2 ทดสอบระบบรวมทั้งหมด	63
บทที่ 5 สรุปผลและข้อเสนอแนะ	67
5.1 สรุปผล	67
5.1.1 แอปพลิเคชันรีโมทจำลองบนโทรศัพท์ระบบปฏิบัติการ แอนดรอยด์	67
5.1.2 การสื่อสารข้อมูลผ่าน WI-FI	67
5.1.3 ข้อมูลคำสั่งของรีโมท	67
5.1.4 การส่งสัญญาณอินฟราเรด	68
5.2 ข้อเสนอแนะ	68
บรรณานุกรม	69

สารบัญรูป

รูปที่	หน้า	
2.1	บล็อกไดอะแกรม	3
2.2	Arduino UNO R3 Board	4
2.3	หน้าต่างโปรแกรม IDE	5
2.4	โครงสร้างโปรแกรมของ Arduino	6
2.5	การมอดูเลตสัญญาณที่ส่งออกและสัญญาณที่รับได้	14
2.6	วงจรส่งสัญญาณอินฟราเรด	15
2.7	บล็อกไดอะแกรมของตัวรับสัญญาณอินฟราเรด	16
2.8	ตัวรับสัญญาณอินฟราเรด	17
2.9	รูปค่าลอจิกของ SAMSUNG Protocol	17
2.10	สัญญาณอินฟราเรดของ SAMSUNG Protocol	18
2.11	สถาปัตยกรรมของแอนดรอยด์	20
2.12	Eclipse platform Architecture	23
2.13	ขั้นตอนการ Encapsulation และ Demultiplexing ในการรับส่งข้อมูล	24
3.1	บล็อกไดอะแกรมของปริญญานิพนธ์	28
3.2	แผนผังการทำงานของแอปพลิเคชัน	29
3.3	การต่อโมดูล Wi-Fi เข้ากับบอร์ด Arduino	30
3.4	การตั้งค่าโมดูล Wi-Fi	31
3.5	การวัดเอาต์พุตของตัวรับสัญญาณอินฟราเรดด้วยเครื่องออสซิลโลสโคป	32
3.6	แผนผังการทำงานของโปรแกรมอ่านสัญญาณอินฟราเรดจากกรีโมท	34
3.7	วงจรส่งสัญญาณอินฟราเรด	35
3.8	แผนผังการทำงานของโปรแกรมส่งสัญญาณอินฟราเรด	36
3.9	การต่อวงจรรวมในส่วนของบอร์ด Arduino เพื่อรับข้อมูลผ่าน Wi-Fi และส่งสัญญาณอินฟราเรดออก	37
3.10	วงจรจ่ายไฟ	38

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 Interface diagram ของโมดูล M03 – LVTTTL UART to Wi-Fi	40
3.12 รีโมทควบคุมโทรทัศน์ยี่ห้อ SAMSUNG รุ่น Slimfit	41
4.1 ลำดับหน้าจอของแอปพลิเคชัน	44
4.2 โปรแกรม NetBeans IDE เปิดพอร์ตหมายเลข 1234 รอรับการเชื่อมต่อ	45
4.3ก หน้าจออีมูเลเตอร์แสดงสถานะว่าเชื่อมต่อสำเร็จ	46
4.3ข หน้าจอโปรแกรมที่ใช้เปิดพอร์ตแสดงสถานะว่าพอร์ตหยุดการทำงาน	46
4.4ก หน้าจออีมูเลเตอร์แสดงสถานการณ์เชื่อมต่อไม่สำเร็จ	47
4.4ข หน้าจอโปรแกรมที่ใช้เปิดพอร์ตแสดงสถานะว่าพอร์ตยังเปิดอยู่	47
4.5 Serial Monitor แสดงการเริ่มต้นของโปรแกรมโดยไม่มีคำสั่งส่งข้อมูลมายังบอร์ด Arduino	48
4.6 Serial Monitor แสดงค่าที่รับได้เป็น 1 เมื่อมีการกดปุ่ม CH1	48
4.7 Serial Monitor แสดงค่าที่รับได้เป็น 1 เมื่อมีการกดปุ่ม CH1	49
4.8 Serial Monitor แสดงค่าที่รับได้เป็น a เมื่อมีการกดปุ่ม Vol+	49
4.9 Serial Monitor แสดงค่าที่รับได้เป็น b เมื่อมีการกดปุ่ม Vol-	50
4.10 Serial Monitor แสดงค่าที่รับได้เป็น c เมื่อมีการกดปุ่ม POWER	50
4.11 สัญญาณอินฟราเรดของปุ่มกดเลข 1 บนรีโมท SAMSUNG	51
4.12 สัญญาณ Start bit	52
4.13 สัญญาณบิต 1	53
4.14 สัญญาณบิต 0	53
4.15 สัญญาณ Stop bit มีความกว้างช่วง On เป็นเวลา 640us	53
4.16 เปรียบเทียบสัญญาณที่รับได้จากการกดปุ่มเลข 1 กับปุ่มเลข 2	54
4.17 การวิเคราะห์ผลช่วงเวลาเป็นบิตข้อมูล แล้วแปลงให้อยู่ในรูปแบบเลขฐานสิบหก	56
4.18 (ชาย) Serial Monitor แสดงการรับค่า 1 จากการกดแป้นพิมพ์และส่งออกค่า S1 (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรด	

สารบัญรูป(ต่อ)

รูปที่	หน้า
ของ CH1	58
4.19 (ชาย) Serial Monitor แสดงการรับค่า 2 จากการกดแป้นพิมพ์และส่งออกค่า S2 (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ CH2	58
4.20 (ชาย) Serial Monitor แสดงการรับค่า a จากการกดแป้นพิมพ์และส่งออกค่า VOLUP (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol+	58
4.21 (ชาย) Serial Monitor แสดงการรับค่า b จากการกดแป้นพิมพ์และส่งออกค่า VOLDW (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol-	59
4.22 (ชาย) Serial Monitor แสดงการรับค่า c จากการกดแป้นพิมพ์และส่งออกค่า PWR (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ POWER	59
4.23 ผลการวัดสัญญาณอินฟราเรดของปั๊ม CH1 ที่ระยะห่าง 50 cm	60
4.24 ผลการวัดสัญญาณอินฟราเรดของปั๊ม CH1 ที่ระยะห่าง 100 cm	60
4.25 ผลการวัดสัญญาณอินฟราเรดของปั๊ม CH1 ที่ระยะห่าง 150 cm	61
4.26 ผลการวัดสัญญาณอินฟราเรดของปั๊ม CH1 ที่ระยะห่าง 200 cm	61
4.27 ผลการวัดสัญญาณอินฟราเรดของปั๊ม CH1 ที่ระยะห่าง 300 cm	62
4.28 ผลการวัดสัญญาณอินฟราเรดของปั๊ม CH1 ที่ระยะห่าง 400 cm	62
4.29 ผลการวัดสัญญาณอินฟราเรดของปั๊ม CH1 ที่ระยะห่าง 500 cm	63
4.30 (ชาย) Serial Monitor แสดงการรับค่า 1 จากการกดปุ่มแอปพลิเคชันและส่งออกค่า S1 (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ CH1	64
4.31 (ชาย) Serial Monitor แสดงการรับค่า 2 จากการกดปุ่มแอปพลิเคชัน	

สารบัญรูป(ต่อ)

รูปที่	หน้า
และส่งออกคา S2 (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ CH2	64
4.32 (ซ้าย) Serial Monitor แสดงการรับคา a จากการกดปุ่มแอปพลิเคชัน และส่งออกคา VOLUP (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol+	64
4.33 (ซ้าย) Serial Monitor แสดงการรับคา b จากการกดปุ่มแอปพลิเคชัน และส่งออกคา VOLDW (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol-	65
4.34 (ซ้าย) Serial Monitor แสดงการรับคา c จากการกดปุ่มแอปพลิเคชัน และส่งออกคา PWR (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ POWER	65

สารบัญตาราง

ตารางที่		หน้า
2.1	คุณลักษณะของบอร์ด ARDUINO UNO R3	4
2.2	คำสั่งประเภทโครงสร้าง	7
2.3	คำสั่งกลุ่มตัวแปร	9
2.4	คำสั่งกลุ่มฟังก์ชัน	11
2.5	รูปแบบการส่งสัญญาณรีโมทควบคุมโทรศัพท์มือถือ SAMSUNG	18
3.1	รายละเอียดขา PIN ของโมดูล WI-FI	40
4.1	ความกว้างช่วงเวลาของบิตต่างๆ	54
4.2	ผลความกว้างช่วงเวลาของสัญญาณอินฟราเรดของปุ่มกดเลข 1	55
4.3	บิตข้อมูลการส่งสัญญาณของแต่ละปุ่มกดในรูปแบบเลขฐานสิบหก	56
4.4	สถิติความถูกต้องของการกดปุ่มแอปพลิเคชันรีโมทจำลอง	66

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันเทคโนโลยีการสื่อสารได้พัฒนาไปเร็วมาก และได้เข้ามาช่วยอำนวยความสะดวกให้แก่มนุษย์มากขึ้น เทคโนโลยีเหล่านี้พบได้ทั่วไป เช่น รีโมทคอนโทรล ซึ่งเป็นอุปกรณ์ที่ใช้ควบคุมอุปกรณ์ไฟฟ้าในระยะไกล เพิ่มความสะดวกให้ผู้ใช้งานสามารถเปิด-ปิดเครื่องใช้ไฟฟ้านั้นโดยไม่ต้องเดินไปถึงเครื่องใช้ไฟฟ้านั้นให้เสียเวลา แต่รีโมทคอนโทรลส่วนใหญ่จะมีใช้กับอุปกรณ์ไฟฟ้าที่ราคาค่อนข้างสูง เช่น โทรทัศน์ เครื่องปรับอากาศ เป็นต้น และในทุกวันนี้ Smartphone เป็นที่ได้รับความนิยมจากคนทุกเพศทุกวัย อาจเป็นเพราะ Smartphone มีแอปพลิเคชันที่สามารถตอบสนองความต้องการของคนส่วนใหญ่ในยุคนี้ได้เป็นอย่างดี ทางผู้จัดทำจึงเห็นว่า หากเราสามารถสร้างแอปพลิเคชันขึ้นมาใช้ควบคุมอุปกรณ์ไฟฟ้าบน Smartphone แทนการใช้รีโมทคอนโทรล ก็จะทำให้เราสะดวกมากยิ่งขึ้น ทำให้เกิดเป็นโครงการควบคุมอุปกรณ์รับสัญญาณอินฟราเรดด้วย Smartphone ผ่าน Wi-Fi นี้ขึ้นมา

1.2 วัตถุประสงค์

- 1.) เพื่อสั่งการและควบคุมอุปกรณ์รับสัญญาณอินฟราเรดผ่าน Wi-Fi
- 2.) เพื่อศึกษาการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ของ Smartphone
- 3.) เพื่อศึกษาการเชื่อมต่อผ่านระบบไร้สายแบบ Wi-Fi
- 4.) เพื่อศึกษาหลักการทำงานและการใช้บอร์ด Arduino

1.3 ขอบเขตของปริญญานิพนธ์

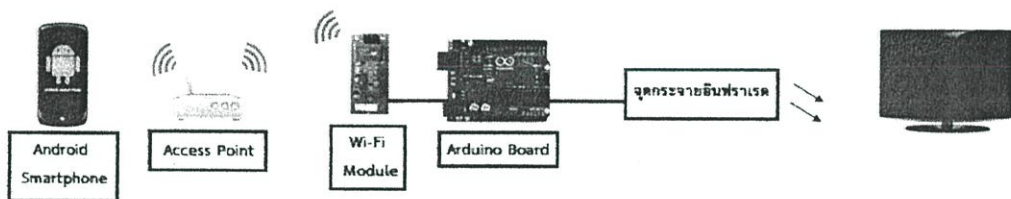
- 1.) สามารถควบคุมการทำงานของโทรทัศน์ 1 เครื่องภายในบ้านได้ด้วยแอปพลิเคชันรีโมทจำลอง

- 2.) แอปพลิเคชันริโมทจำลองบน Smartphone ประกอบด้วย ปุ่มกดรีโมทจำนวน 26 ปุ่มฟังก์ชันการทำงาน ให้เลือกใช้งานควบคุมโทรทัศน์
- 3.) โทรศัพท์เคลื่อนที่และเซิร์ฟเวอร์จะต้องเชื่อมต่อเครือข่ายไร้สายวงเดียวกัน

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

โครงการชิ้นนี้สามารถแสดงบล็อกไดอะแกรมการทำงานได้ดังรูปที่ 2.1 จึงมีทฤษฎีและหลักการที่เกี่ยวข้องด้วยกัน 5 หัวข้อหลักคือ บอร์ด Arduino, รีโมทคอนโทรลอินฟราเรด, แอนดรอยด์, โพรโตคอลที่ใช้ในการสื่อสารข้อมูล และ Wireless LAN



รูปที่ 2.1 บล็อกไดอะแกรม

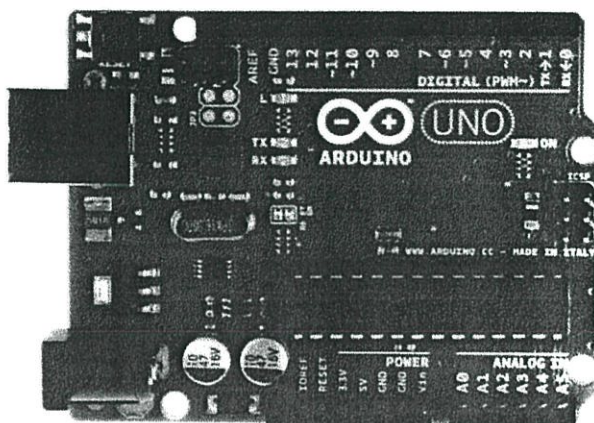
2.1 บอร์ด Arduino

ไมโครคอนโทรลเลอร์ (Microcontroller) คืออุปกรณ์ที่มีหน่วยประมวลผลและความจำขนาดเล็กภายในตัวเอง สามารถรับ-ส่งข้อมูลได้ทั้งแบบอนาล็อกและดิจิทัล ใช้พลังงานน้อยทำให้เป็นที่นิยม ซึ่ง Arduino นั้นก็เป็นไมโครคอนโทรลเลอร์ชนิดหนึ่งที่มีการพัฒนามาแล้วทำให้สามารถเขียนคำสั่งโปรแกรมได้เหมือนโปรแกรมภาษาชั้นสูงทั่วไป

Arduino มีการใช้งานไมโครคอนโทรลเลอร์ในตระกูล AVR และได้รับความนิยมอย่างสูงทั่วโลกเพราะว่าเป็น Open Source หรือก็คืออุปกรณ์ที่มีแบบส่วนประกอบเป็นมาตรฐานที่เปิดเผย ทำให้สามารถซื้อหาได้ง่าย ราคาถูกมีซอฟต์แวร์ (Software) ให้ใช้งานฟรีทำให้นำไปใช้ได้โดยไม่ต้องเสียค่าลิขสิทธิ์ และภาษาที่ใช้กับบอร์ดนี้จะเป็นลักษณะของ C++ ซึ่งถ้าผู้ใช้ไม่เคยศึกษามาก่อนก็สามารถเลือกใช้คำสั่งที่เป็นฟังก์ชันสำเร็จรูป (Built-in function) ของ Arduino ก็ได้

2.1.1 รายละเอียดคุณลักษณะ

บอร์ด Arduino ที่ได้เลือกศึกษาเพื่อนำมาใช้งานคือรุ่น Arduino UNO R3 ดังรูปที่ 2.2 ซึ่งเป็นรุ่นพื้นฐาน และมีรายละเอียดคุณลักษณะดังตารางที่ 2.1



รูปที่ 2.2 Arduino UNO R3 Board

ตารางที่ 2.1 คุณลักษณะของบอร์ด Arduino UNO R3

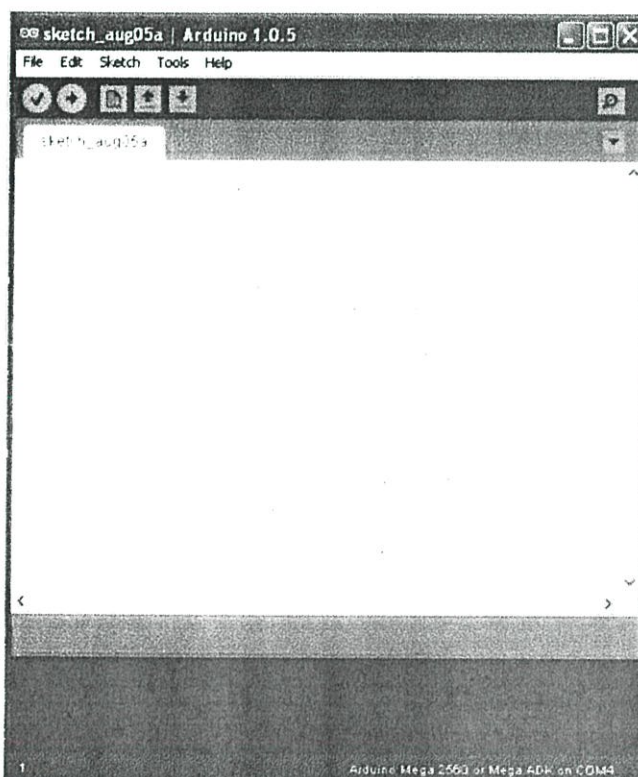
ไมโครคอนโทรลเลอร์	ATmega328
แรงดันที่ใช้งาน	5V
แรงดันอินพุต	7-12V(แรงดันสูงสุด 6-20V)
พินDigital I/O	14 พิน (6 พินให้เอาต์พุตแบบ PWM)
พินAnalog Input	6 พิน
แรงดันไฟตรงต่อพินI/O	40 mA
แรงดันไฟตรงที่พิน3.3V	50 mA
Flash Memory	32 KB (ATmega328) โดยโปรแกรม bootloader ใช้พื้นที่แล้ว 0.5 KB
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

2.1.2 การติดตั้งก่อนใช้งาน

สำหรับ Arduino UNO R3 นั้นจะมีช่อง USB port ซึ่งทำให้สามารถต่อกับคอมพิวเตอร์ผ่านสาย USB ได้ และไฟจากคอมพิวเตอร์ก็จะเป็นไฟเลี้ยง 5V ให้กับบอร์ดเอง แต่ก่อนที่จะสามารถนำโปรแกรมไปใส่ในบอร์ดได้ ก็ต้องมีการลงซอฟต์แวร์ที่จำเป็น 2 ส่วนคือ

- ไตรเวอร์บอร์ด Arduino
- โปรแกรมพัฒนา Arduino หรือ Arduino Environment (IDE)

ซึ่งซอฟต์แวร์ทั้งสองมีให้ดาวน์โหลดจากเว็บไซต์ <http://arduino.cc/> ซึ่งเป็นเว็บไซต์ของผู้ผลิตและจัดจำหน่ายบอร์ด Arduino นั้นเองเมื่อทำการติดตั้งเสร็จสิ้นแล้ว ก็จะได้หน้าต่างสำหรับการใช้งานเขียนโปรแกรมดังรูปที่ 2.3



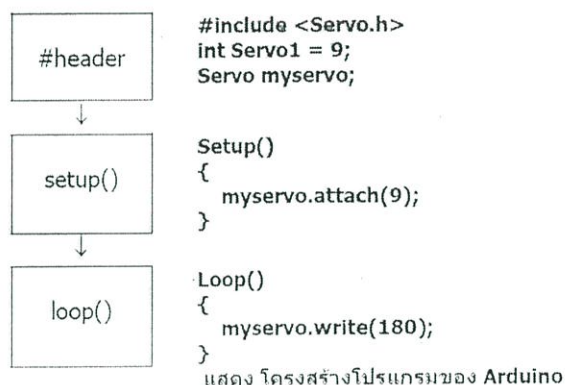
รูปที่ 2.3 หน้าต่างโปรแกรม IDE

2.1.3 โปรแกรมภาษา C ของ Arduino

2.1.3.1 โครงสร้างโปรแกรม

โปรแกรมที่ใช้บนบอร์ด Arduino จะใช้โครงสร้างของภาษาซี จะจัดรูปแบบโครงสร้างของการเขียนโปรแกรมออกเป็นส่วนย่อยๆ หลายส่วนโดยเรียกแต่ละส่วนว่า ฟังก์ชัน และพอมีฟังก์ชันหลายอันรวมกันก็จะเรียกเป็นโปรแกรม

ซึ่งโครงสร้างโปรแกรมของ Arduino ทุกอันจะต้องประกอบไปด้วยอย่างน้อย 2 ฟังก์ชันคือ ฟังก์ชัน `setup()` กับ `loop()` แสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 โครงสร้างโปรแกรมของ Arduino

จะเห็นได้ว่ามี 3 ส่วนใหญ่ๆ ด้วยกันคือ

1.) Header เป็นส่วนที่จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดไว้ในส่วนเริ่มต้นของโปรแกรม ได้แก่ส่วนที่เป็น Compiler Directive ต่างๆ รวมไปถึงการประกาศตัวแปรและค่าคงที่ต่างๆ ที่จะใช้ตลอดโปรแกรม

2.) `setup()` เป็นฟังก์ชันบังคับที่กำหนดให้ต้องมีทุกโปรแกรม ใช้บรรจุคำสั่งส่วนที่ต้องการให้ทำงานเพียงรอบเดียวตอนเริ่มต้นการทำงานของโปรแกรมครั้งแรกเท่านั้น เช่นการ set up ค่า baud rate, การกำหนดค่า pinMode หรือก็คือบอก pin ที่จะใช้เป็น I/P หรือ O/P

3.) `loop()` เป็นฟังก์ชันบังคับที่ต้องมีในทุกโปรแกรม โดยจะบรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานวนซ้ำเป็นรอบต่อไปไม่รู้จบ

2.1.3.2 คำสั่งในโปรแกรม

คำสั่งในโปรแกรม Arduino แบ่งออกเป็น 3 ประเภทหลักๆ คือ โครงสร้าง (Structure), ตัวแปร (Variable) และ ฟังก์ชัน (Function) มีรายละเอียดคร่าวๆ ของคำสั่งต่างๆ ดังที่แสดงในตารางที่ 2.2, 2.3 และ 2.4 ซึ่งส่วนสำคัญของการเขียนคำสั่งก็คือการประกาศตัวแปร ที่มีเงื่อนไขว่า ห้ามตั้งชื่อซ้ำกับคำสั่งหลัก ซึ่งสรุปได้ดังนี้

- 1.) ชื่อของตัวแปรต้องประกอบไปด้วยตัวอักษรตัวเลขและยอมให้ใช้เครื่องหมายพิเศษอีก 2 ตัวคือเครื่องหมาย Under Line (_) และ Dollar Sign (\$) มาใช้ในการตั้งชื่อได้โดยชื่อต้องเรียงติดทั้งหมดห้ามมีการเว้นวรรคและมีความยาวไม่เกิน 32 ตัวอักษร
- 2.) ชื่อของตัวแปรที่กำหนดต้องเริ่มต้นชื่อด้วยตัวอักษรเป็นลำดับแรกเสมอห้ามตั้งชื่อโดยเริ่มต้นด้วยตัวเลขหรือเครื่องหมาย
- 3.) ชื่อของตัวแปรต้องกำหนดด้วยตัวอักษรที่เป็นตัวอักษรพิมพ์เล็กเท่านั้นและในภาษาซีจะถือว่าตัวอักษรที่เป็นตัวอักษรพิมพ์ใหญ่และตัวอักษรพิมพ์เล็กมีความหมายต่างกันเช่น “LED” และ “led” และ “Led” ภาษาซีจะถือว่าเป็นคนละชื่อกัน
- 4.) ในการตั้งชื่อตัวแปรห้ามนำคำสงวน (Reserve Word) มาใช้ตั้งชื่อซึ่งคำสงวนได้แก่ชื่อคำสั่งและชื่อ Internal Function ต่างๆที่สร้างไว้แล้ว

ตารางที่ 2.2 คำสั่งประเภทโครงสร้าง

กลุ่ม	ชื่อคำสั่ง	การใช้งาน
เงื่อนไข	if	ตรวจสอบเงื่อนไขเพื่อให้โปรแกรมเลือกทำงาน
	if...else	เหมือน if แต่มีเงื่อนไขเพิ่มมากขึ้น
	For	สั่งโปรแกรมวนรอบทำงานตามค่าเริ่มต้นที่กำหนด มีเงื่อนไขการเริ่มต้นและสิ้นสุดที่แน่นอน
	switch case	ตรวจสอบเงื่อนไข แล้วโปรแกรมเลือกทำงานตามเงื่อนไขที่ต้องการเพียงเงื่อนไขเดียว

เงื่อนไข	While	สั่งให้โปรแกรมทำงานวนรอบซ้ำกันจนกว่าเงื่อนไขจะเป็นเท็จจึงหยุดทำงาน
	do...while	ทำตามคำสั่งหลัง do ก่อนแล้วจึงตรวจสอบเงื่อนไข
	Break	ใช้ร่วมกับคำสั่งวนรอบเพื่อจบการทำงานโดยไม่สนใจเงื่อนไข หรือข้ามการตรวจเงื่อนไขใน switch
	continue	ข้ามการทำงานของคำสั่งถัดไป 1 คำสั่ง
	Return	ใช้เมื่อจบการทำงานโปรแกรมย่อยและคืนค่าโปรแกรมย่อยไปยังโปรแกรมหลัก
	Goto	สั่งให้โปรแกรมกระโดดไปทำงานตำแหน่งต่างๆ ในโปรแกรมที่ระบุไว้
เครื่องหมาย	;	(Semicolon) บอกการสิ้นสุดประโยคคำสั่ง
	{ }	(curly brace) กำหนดขอบเขตประโยคคำสั่งหรือโปรแกรมย่อย
	//	จุดเริ่มต้นการอธิบายในบรรทัดคำสั่งเพียงบรรทัดเดียว ข้อความจะไม่ถูกนำไปประมวลผลด้วย
	/* */	อธิบายหลายบรรทัด
	#define	กำหนดค่าของตัวแปร เพื่อแทนที่ตัวแปรนั้นในโปรแกรมด้วยค่าที่กำหนด
	#include	เรียกใช้โปรแกรมภายนอก
ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)	=	ใช้ในการกำหนดค่าตัวแปร
	+	ใช้แทนการบวก
	-	ใช้แทนการลบ
	*	ใช้แทนการคูณ
	/	ใช้แทนการหาร
	%	ใช้แทนการ mod (หารแล้วเอาเศษมาใช้)
ตัวดำเนินการเปรียบเทียบ (Comparison Operator)	==	เท่ากับ
	!=	ไม่เท่ากับ
	<	น้อยกว่า
	>	มากกว่า

ตัวดำเนินการ เปรียบเทียบ	<=	น้อยกว่าหรือเท่ากับ
	>=	มากกว่าหรือเท่ากับ
ตัวดำเนินการ บูลีน (Boolean)	&&	ตรรกะ และ (AND)
		ตรรกะ หรือ (OR)
	!	ตรรกะ ไม่ (NOT)
ตัวดำเนินการบิต (Bit Operator)	&	การ AND ของบิต
		การ OR ของบิต
	^	การทำ XOR ของบิต
	~	การ NOT ของบิต
	<<	เลื่อนบิตไปทางซ้าย
	>>	เลื่อนบิตไปทางขวา
ตัวดำเนินการ ประกอบ (Compound Operator)	++	เพิ่มค่าขึ้นอีก 1
	--	ลดค่าลงอีก 1
	+=	ค่าทางซ้ายเท่ากับค่าทางซ้ายบวกค่าทางขวา
	-=	ค่าทางซ้ายเท่ากับค่าทางซ้ายลบค่าทางขวา
	*=	ค่าทางซ้ายเท่ากับค่าทางซ้ายคูณด้วยค่าทางขวา
	/=	ค่าทางซ้ายเท่ากับค่าทางซ้ายหารด้วยค่าทางขวา

ตารางที่ 2.3 คำสั่งกลุ่มตัวแปร

กลุ่ม	ชื่อคำสั่ง	การใช้งาน
ค่าคงที่	HIGH	แทนสถานะลอจิก 1
	LOW	แทนสถานะลอจิก 0
	INPUT, OUTPUT	กำหนดค่าสถานะให้กับฟังก์ชัน pinMode()
ค่าคงที่	TRUE	ค่าคงที่บูลีน เป็น 1 หรือ จริง หรือ TRUE
	FALSE	ค่าคงที่บูลีน เป็น 0 หรือ เท็จ หรือ FALSE
ชนิดข้อมูล	Void	ตัวแปรที่ไม่มีการเก็บค่า หรือก็คือไม่มีค่า
	boolean	เก็บค่าข้อมูล 2 จำนวนคือ TRUE และ FALSE

ชนิดข้อมูล	Char	เก็บค่าข้อมูลขนาด 8 บิต เป็นข้อมูลตัวอักษร หรือ ใช้เก็บเป็นเลขจำนวนเต็มได้ 256 ค่าตั้งแต่ -128 ถึง 127
	unsigned char	เหมือน char แต่เก็บค่าได้ตั้งแต่ 0 ถึง 255
	Word	เก็บค่าข้อมูลขนาด 16 บิตตั้งแต่ 0 ถึง 65535
	Byte	เหมือน unsigned char
	int	เก็บค่าข้อมูลขนาด 16 บิต ได้ 65536 ค่าคือ -32768 ถึง +32767
	unsigned int	เหมือน word
	Long	เก็บข้อมูลขนาด 32 บิต ตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
	unsigned long	เหมือน long แต่เก็บค่า 0 ถึง 4,294,967,295
	Short	เก็บเลขจำนวนเต็มในตัวแปรที่มีค่าน้อยกว่า int
	Float	เก็บค่าข้อมูลที่เป็นเลขทศนิยมขนาด 32 บิต เก็บได้ ตั้งแต่ 3.4E-38 ถึง 3.4E+308
	double	เก็บค่าข้อมูลที่เป็นเลขทศนิยม ละเอียดกว่า float 2 เท่าคือเก็บได้ถึง 1.7E+308
	String	เก็บข้อความหรืออักษรหลายๆ ตัว
	Array	เก็บข้อมูลหลายๆ ค่าไว้ในตัวแปรเพียงชื่อเดียว มีตัวเลขชี้ตำแหน่งเก็บข้อมูลที่ต่างกัน
pointer()	เก็บค่าตำแหน่งแอดเดรสของหน่วยความจำที่ใช้สร้างเป็นตัวแปรสำหรับเก็บข้อมูล	
Conversion	char()	เปลี่ยนค่าตัวแปรให้เป็นชนิด char
	byte()	เปลี่ยนค่าตัวแปรให้เป็นชนิด byte
	int()	เปลี่ยนค่าตัวแปรให้เป็นชนิด int
	word()	เปลี่ยนค่าตัวแปรให้เป็นชนิด word
	long()	เปลี่ยนค่าตัวแปรให้เป็นชนิด long
	float()	เปลี่ยนค่าตัวแปรให้เป็นชนิด float

Conversion	Static	มีขอบเขตการใช้งานเฉพาะบล็อกที่กำหนดไว้
	volatile	ประกาศว่าตัวแปรนั้นมีการเปลี่ยนแปลงตลอดเวลา
	Const	ประกาศว่าตัวแปรนั้นเป็นค่าคงที่
	sizeof()	คืนข้อมูลไบนารีนั้นในรูปแบบตัวแปรประเภทที่กำหนด
	PROGMEM	ตัวแปรที่ประกาศต้องการสร้างและเก็บไว้ใน MCU ใช้กับที่แปรที่ใช้อ่านข้อมูลอย่างเดียว

ตารางที่ 2.4 คำสั่งกลุ่มฟังก์ชัน

กลุ่ม	ชื่อคำสั่ง	การใช้งาน
ดิจิตอล I/O	pinMode()	กำหนดการใช้งานขา digital I/O ว่าขาใดจะใช้ งานเป็น Input หรือ Output
	digitalWrite()	กำหนดสถานะทาง O/P ให้กับขา pin digital I/O ว่ามีสถานะลอจิกเป็น HIGH หรือ LOW
	digitalRead()	อ่านค่าสถานะลอจิกอินพุตจากขาที่กำหนด
อะนาล็อก I/O	analogReference()	เลือกชนิดแรงดันอ้างอิงวงจร ใช้เป็น analog I/P
	analogRead()	อ่านค่าระดับแรงดันที่ขา pin analog อินพุต
	analogWrite()	กำหนดค่าระดับของสัญญาณ analog เอาท์พุต ให้กับขา pin digital เอาท์พุต
I/O ขั้นสูง	tone()	สร้างสัญญาณคลื่นรูปสี่เหลี่ยม ได้เป็นโทนเสียง
	noTone()	หยุดสัญญาณรูปสี่เหลี่ยมที่สร้างโดย tone()
	shiftOut()	ส่งข้อมูลแบบไบนารีออกไปทีละ 1 บิต
	shiftIn()	รับข้อมูลแบบไบนารีเข้ามาทีละ 1 บิต
	pulseIn()	อ่านค่าช่วงเวลาของพัลส์ที่กำหนด เลือกอ่านพัลส์ ที่เป็น HIGH หรือ LOW
เวลา	millis()	ส่งกลับค่าที่อ่านในหน่วย milli-second
	micros()	ส่งกลับค่าที่อ่านในหน่วย microsecond
	delay()	หน่วงเวลา มีหน่วยเป็น milli-second
	delayMicroseconds	หน่วงเวลา มีหน่วยเป็น micro-second

ตัวดำเนินการทางคณิตศาสตร์	min()	ตรวจสอบเลข 2 จำนวนแล้วคืนค่าตัวที่น้อยที่สุด
	max()	ตรวจสอบเลข 2 จำนวนแล้วคืนค่าตัวที่มากที่สุด
	abs()	คำนวณหาค่าสัมบูรณ์ของตัวแปร
	constrain()	ตรวจสอบและปรับค่าตัวเลขให้อยู่ในย่านที่กีดที่กำหนดไว้
	pow()	ยกกำลัง
	sqrt()	คำนวณรากที่สอง (Square-root)
ตรีโกณมิติ	sin()	หา sin ได้ผลเป็นแบบ double ตั้งแต่ -1 ถึง 1
	cos()	หา cos ได้ผลเป็นแบบ double ตั้งแต่ -1 ถึง 1
	tan()	หา tan ได้ผลเป็นตัวแบบ double ตั้งแต่ -1 ถึง 1
สุ่มตัวเลข	randomSeed()	สุ่มจำนวนครั้งในการสุ่มข้อมูลให้ random()
	random()	สุ่มค่าจำนวน
	bitRead()	อ่านค่าบิตจากจำนวนที่กำหนด
	bitWrite()	เขียนจำนวนบิต
	bitSet()	กำหนดจำนวนที่เลือกเป็นบิต 1
	bitClear()	ล้างค่าที่เลือกให้เป็นบิต 0
	bit()	แปลงค่าจากบิตเป็นเลขฐานสิบ
อินเตอร์รัปต์	interrupts()	อนุญาตให้งานบางประเภทดำเนินการได้
	noInterrupts()	ยกเลิกการอินเตอร์รัปต์
	attachInterrupt()	อินเตอร์รัปต์ฟังก์ชันจากภายนอก
	detachInterrupt()	ปิดการอินเตอร์รัปต์ที่มาจากอุปกรณ์ภายนอก
การสื่อสารอนุกรม	Serial.begin()	กำหนดค่าความเร็วของพอร์ตสื่อสารอนุกรม
	Serial.available()	ตรวจสอบจำนวนข้อมูลที่ได้รับจากพอร์ตสื่อสารอนุกรมและเก็บรอไว้ใน buffer
	Serial.read()	อ่านข้อมูลจากบัฟเฟอร์ของพอร์ตสื่อสารอนุกรม หรืออ่านข้อมูลจากหน้าจอ IDE
	Serial.print()	ส่งค่าข้อมูลออกไปยังพอร์ตสื่อสารอนุกรม โดยแสดงเป็นบรรทัดเดียวกันหมด

การสื่อสาร อนุกรม	Serial.println()	ส่งค่าข้อมูลออกไปยังพอร์ตสื่อสารอนุกรม โดย แสดง ผลแบบขึ้นบรรทัดใหม่
	Serial.flush()	ล้างข้อมูลในบัฟเฟอร์ของพอร์ตสื่อสารอนุกรม

2.2 รีโมทคอนโทรลอินฟราเรด

รีโมทคอนโทรลอินฟราเรด หมายถึง การควบคุมอุปกรณ์ระยะไกล ในระยะที่มองเห็น โดยใช้ลำแสงอินฟราเรด สาเหตุที่ใช้อินฟราเรดสำหรับทำรีโมทคอนโทรล คือ ไม่สามารถมองเห็นลำแสงขณะใช้งานและอินฟราเรดมีราคาถูกและทำได้ง่าย

ปัจจุบันรีโมทคอนโทรลอินฟราเรด ซึ่งทำหน้าที่เสมือนแขนขาของมนุษย์ จึงทำให้เป็นที่นิยมใช้กันอย่างแพร่หลาย และเริ่มเป็นสิ่งจำเป็นควบคู่กับเครื่องใช้ไฟฟ้าในปัจจุบัน จึงสมควรที่จะให้ความสนใจกันอย่างละเอียด

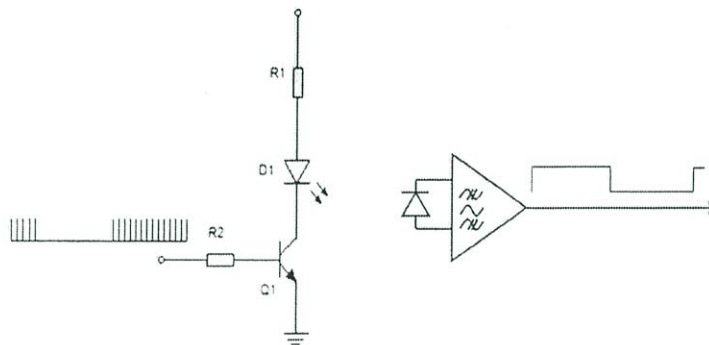
2.2.1 แสงอินฟราเรด

แสงอินฟราเรด หมายถึง แสงที่มีความยาวคลื่น 950 nm เป็นแสงปกติที่มีสีเฉพาะเจาะจงและสายตามนุษย์ไม่สามารถมองเห็นได้

แหล่งกำเนิดแสงอินฟราเรดมีมากมาย แสงอาทิตย์เป็นแหล่งกำเนิดที่สว่างที่สุด แต่ก็ยังมีแหล่งกำเนิดอีกหลายแห่ง เช่น แสงจากหลอดไฟที่ แสงเทียน หรือแม้แต่ร่างกายเราตามหลักความจริงแล้วทุกสิ่งทุกอย่างที่แผ่ความร้อนได้ สามารถแผ่อินฟราเรดได้เช่นกัน เราต้องหาวิธีการส่งข้อความไปยังตัวรับอย่างไม่มีข้อผิดพลาด

2.2.2 การมอดูเลต (Modulation)

การมอดูเลต คือ การนำเอาสัญญาณซีไปบนสัญญาณรบกวน โดยการมอดูเลตจะใช้ความถี่เฉพาะที่เกิดจากแสงของอินฟราเรด และตัวอินฟราเรดด้านรับ จะต้องปรับให้มีความถี่ตรงกัน



รูปที่ 2.5 การมอดูเลตสัญญาณที่ส่งออกและสัญญาณที่รับได้

การติดต่อสื่อสารของมนุษย์โดยทั่วๆ ไปมี 2 อย่างคือ การใช้คำพูด และการไม่ใช้คำพูด แต่ในทางการสื่อสารคมนาคม จะมีลักษณะเป็น “mark” และ “space” โดย “space” คือสัญญาณปกติ มีสถานการณ์ส่งปิด ซึ่งไม่มีการส่งสัญญาณแสงอินฟราเรดในสถานะนี้ และ “mark” เป็นสถานการณ์ส่งสัญญาณแสงอินฟราเรด โดยส่งเป็นพัลส์ “ON” และ “OFF” ตามความถี่เฉพาะนั้นๆ โดยทั่วๆ ไปของผู้ผลิตจะใช้ความถี่ 30 kHz และ 60 kHz

ทางด้านรับ “space” แสดงว่า เอาต์พุต เป็น High level และ “mark” แสดงว่า เอาต์พุตเป็น Low level แต่จำไว้ว่า “mark” และ “space” ไม่ใช่ 1 และ 0 ที่ต้องการส่ง แต่ความสัมพันธ์จริงๆ ระหว่าง 1 และ 0 ถูกกำหนดโดยโปรโตคอลที่เราใช้

2.2.3 ตัวส่งสัญญาณ (Transmitter)

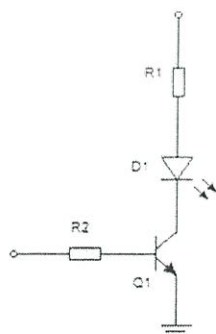
โดยปกติตัวส่งจะใช้พลังงานแบตเตอรี่ ซึ่งมีกำลังไม่สูงมากนัก แต่สัญญาณอินฟราเรดก็มีความเข้มเพียงพอที่จะส่งเป็นระยะทางที่สามารถควบคุมอุปกรณ์นั้นๆ โดยมีความทนทานต่อการสั่นสะเทือนได้เป็นอย่างดี

ชิป (Chips) หลายตัวได้ถูกออกแบบมาใช้เป็นตัวส่งสัญญาณอินฟราเรด ชิปรุ่นเก่าใช้โปรโตคอลเพียงตัวเดียวจากหลายโปรโตคอลที่มี ในปัจจุบันไมโครคอนโทรลเลอร์ มี กำลังงานต่ำมากๆ ได้ถูกนำมาใช้เป็นตัวส่งสัญญาณอินฟราเรด เพราะมีความยืดหยุ่นในการใช้งาน สูง ซึ่งเมื่อไม่มีการกดปุ่ม กำลังงานจะน้อยมากๆ แทบจะไม่มีกระแสเลยแต่เครื่องส่งจะทำงานเมื่อมีการกดปุ่มเกิดขึ้นเท่านั้น

ผลึก Crystals นานๆ ครั้งจึงถูกใช้ทำงาน ซึ่งจะมีความบอบบางมากๆ และหยุดการทำงานได้ง่ายเมื่อเครื่องหยุดใช้งาน ดังนั้นเซรามิกจึงมีความเหมาะสมมากกว่า เพราะโดยทางกายภาพแล้วจะทนทานต่อการสั่นสะเทือนสูง

กระแสที่ไหลผ่าน LED สามารถเปลี่ยนแปลงจาก 100mA ถึงมากกว่า 1 A เพื่อที่จะส่งสัญญาณไปควบคุมได้ในระยะทางไกลๆ กระแสของ LED ต้องสูง แต่ต้องแลกกันระหว่างตัว LED อายุการใช้งานของแบตเตอรี่ กับระยะทางสูงสุดในการควบคุม กระแสของ LED สามารถทำให้สูงได้เพราะตัวขับพัลส์ของ LED สั้นมากๆ กำลังงานเฉลี่ยของ LED ไม่ควรเกินค่าสูงสุดที่ผ่านได้ ซึ่งเห็นได้จากกระแสสูงสุดของ LED ไม่ได้สูงเกินไป โดยค่าต่างๆ เหล่านี้สามารถดูจากคุณสมบัติของ LED

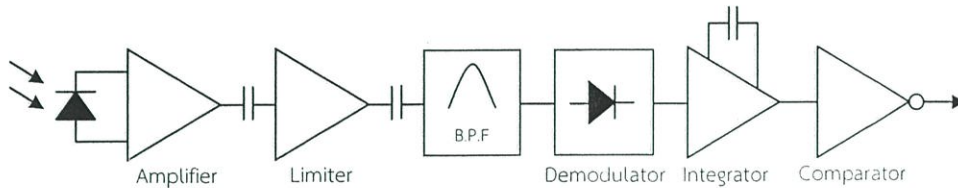
วงจรโดยทั่วไปของทรานซิสเตอร์ สามารถใช้เป็นตัวส่ง LED ได้ โดยมีวงจรการส่งสัญญาณอินฟราเรดพื้นฐานดังรูปที่ 2.6 จะใช้ทรานซิสเตอร์ซึ่งมีคุณสมบัติ HFE และความเร็วของสวิตช์จึงสามารถใช้ได้ตามจุดประสงค์นี้ โดยค่าความต้านทานหาได้โดยใช้กฎของโอห์มและค่าแรงดันตกคร่อมต่ำสุดที่อินฟราเรด LED ประมาณ 1.1 V



รูปที่ 2.6 วงจรส่งสัญญาณอินฟราเรด

2.2.4 ตัวรับสัญญาณ (Receiver)

วงจรเครื่องรับมีมากมาย แตกต่างกันไป แต่กฎเกณฑ์ในการเลือกที่สำคัญที่สุดคือความถี่ที่ใช้ในการมอดูเลตและความต้องการในการนำไปใช้ประโยชน์นั้นๆ

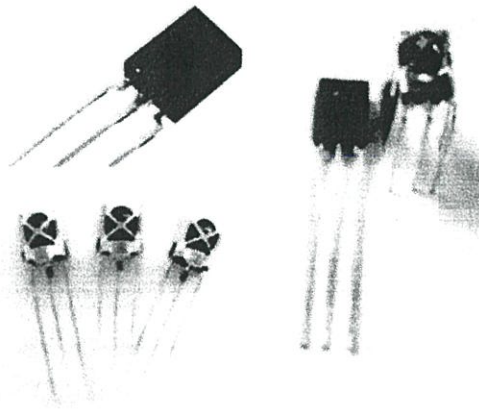


รูปที่ 2.7 บล็อกไดอะแกรมของตัวรับสัญญาณอินฟราเรด

จากรูปที่ 2.7 เมื่อรับสัญญาณอินฟราเรด โดยมีอินฟราเรดไดโอดเป็นตัวตรวจจับสัญญาณ (Detect) และสัญญาณนี้จะถูกขยายและจำกัดโดยใช้วงจร AGC เพื่อให้ระดับพัลส์คงที่โดยไม่คำนึงถึงระยะทาง

สัญญาณ AC จะถูกส่งผ่านวงจร Band Pass Filter และวงจรจะปรับความถี่ในการมอดูเลตโดยทั่วไปความถี่ที่ผู้ผลิตใช้กันคือ 30 kHz ถึง 60 kHz ขั้นตอนต่อมาคือ detector, integrator และ comparator จุดประสงค์ของ 3 ขั้นตอนนี้คือต้องการตรวจจับความถี่ของการมอดูเลต ถ้าความถี่มอดูเลตนี้ออกที่เอาต์พุตของ comparator จะมีความถี่ต่ำ

ดังที่กล่าวไปแล้ว ขั้นตอนเหล่านี้ถูกรวมอยู่ในอุปกรณ์ทางอิเล็กทรอนิกส์เพียงตัวเดียว มีผู้ผลิตหลายบริษัทที่ผลิตอุปกรณ์เหล่านี้ ออกสู่ท้องตลาด และอุปกรณ์ส่วนใหญ่ที่หาได้มีหลายแบบและได้ถูกปรับให้มีความถี่ของการมอดูเลตเฉพาะตัว ซึ่งตัวรับสัญญาณอินฟราเรดนี้ก็มีหน้าตาแตกต่างกันไปหลายแบบดังที่แสดงในรูปที่ 2.8

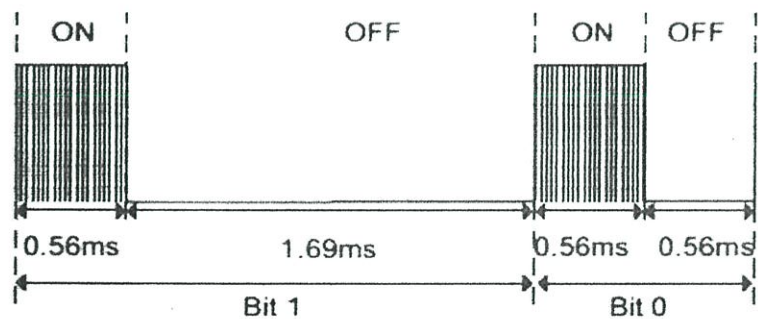


รูปที่ 2.8 ตัวรับสัญญาณอินฟราเรด

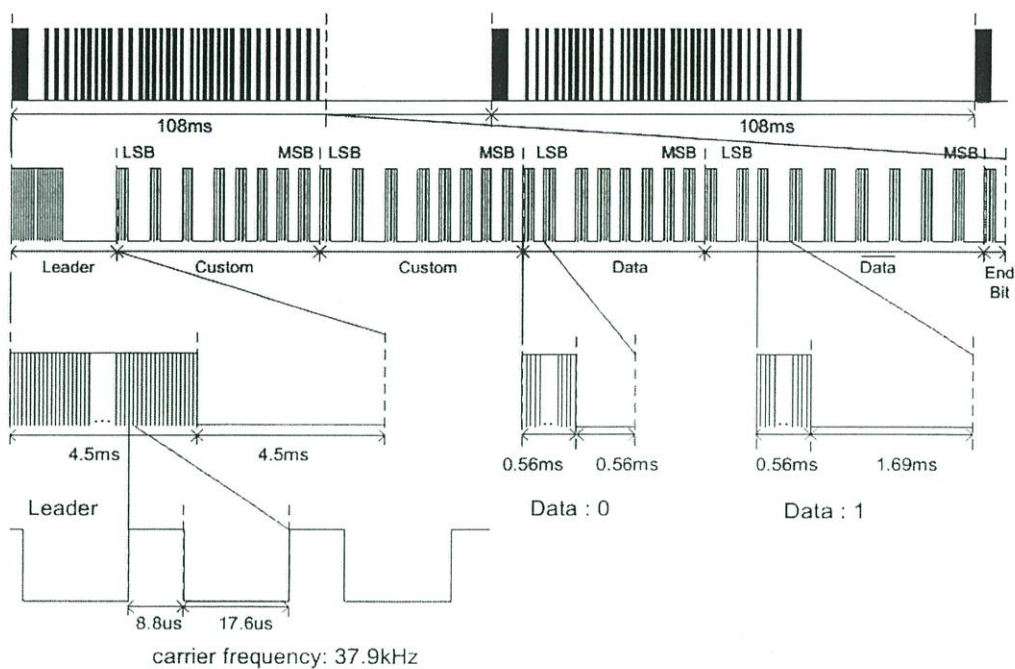
2.2.5 รูปแบบการส่งสัญญาณของรีโมทควบคุมโทรทัศน์

2.2.5.1 SAMSUNG Protocol

จากแหล่งอ้างอิงมีผู้ทำการทดสอบหารูปแบบการส่งสัญญาณของรีโมทควบคุมโทรทัศน์ยี่ห้อ SAMSUNG พบว่า คลื่นพาห์ที่ใช้ในการส่งข้อมูลของมาตรฐานของ SAMSUNG มีความถี่ 37.9 kHz โดยมีการเข้ารหัสแบบ Pulse distance และส่งข้อมูลบิตแรกออกไปเป็น LSB ซึ่งแสดงได้ดังรูปที่ 2.9 และ 2.10



รูปที่ 2.9 รูปค่าลอจิกของ SAMSUNG Protocol



รูปที่ 2.10 สัญญาณอินฟราเรดของ SAMSUNG Protocol

จากรูปที่ 2.9 และ 2.10 สามารถสรุปได้ดังตารางที่ 2.5

ตารางที่ 2.5 รูปแบบการส่งสัญญาณรีโมทควบคุมโทรทัศน์ยี่ห้อ SAMSUNG

	ช่วงเวลา (ms)
ขบวนสัญญาณพัลส์ 1 คาบ	108
บิตเริ่มต้น	On = 4.5 , Off = 4.5
บิต 1	On = 0.56 , Off = 1.69
บิต 0	On = 0.56 , Off = 0.56
Duty Cycle	1/3 เท่าของคลื่นนาฬิกา
ข้อมูลทั้งหมด	32 บิต
ข้อมูลเป็นแอดเดรส	16 บิต
ข้อมูลเป็นคำสั่ง	16 บิต

2.3 แอนดรอยด์ (Android)

ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ (เรียกอีกอย่างว่า สมาร์ทโฟน - Smartphone) และคอมพิวเตอร์แบบแท็บเล็ต (Tablet Computer หรือ Tablet PC) เป็นต้น แอนดรอยด์นั้นพัฒนามาจากระบบปฏิบัติการลินุกซ์ (Linux) อีกทีหนึ่ง

แรกเริ่มนั้นแอนดรอยด์ถูกพัฒนาโดยบริษัท Android Inc. ซึ่งก่อตั้งในปี ค.ศ.2003 ต่อมาในปี ค.ศ.2005 กูเกิลได้ซื้อกิจการของบริษัทนี้ จากนั้นได้ร่วมมือกับกลุ่มบริษัททางด้านฮาร์ดแวร์ ซอฟต์แวร์ และการสื่อสาร เช่น Intel, HTC, LG, Texas Instruments จัดตั้งองค์การความร่วมมือที่มีชื่อว่า Open Handset Alliance ขึ้นในปี ค.ศ.2007 โดยมีวัตถุประสงค์ในการสร้างแพลตฟอร์ม (Platform) สำหรับอุปกรณ์พกพาที่มีพื้นฐานอยู่บนมาตรฐานเปิด (Open Standard) นอกจากนี้การที่แอนดรอยด์เป็นระบบเปิดหรือโอเพ่นซอร์ส (Open Source) ก็เปิดโอกาสให้ผู้ผลิตสามารถปรับแต่งแอนดรอยด์ให้เหมาะกับฮาร์ดแวร์ของตนเองได้

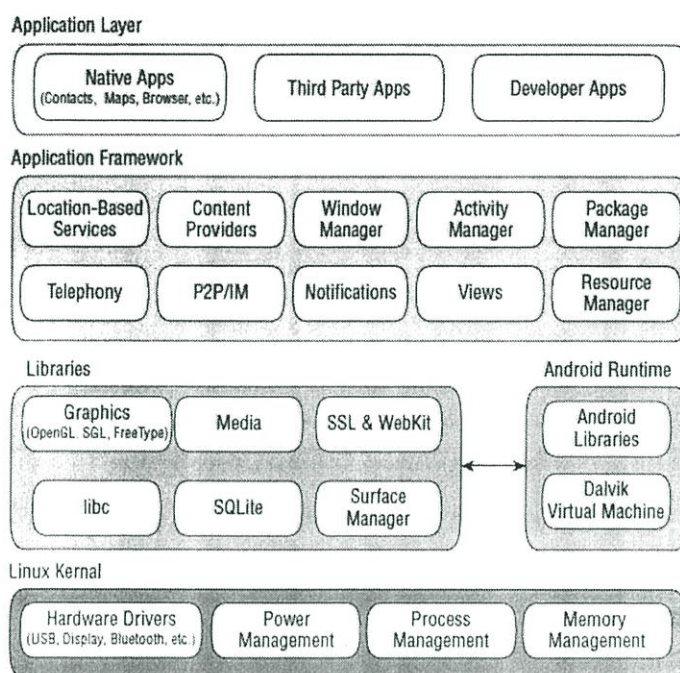
การพัฒนาแอปพลิเคชันที่รันบนแอนดรอยด์จะใช้ภาษาจาวา โดยการเข้าถึงความสามารถต่างๆ ของแอนดรอยด์จะสามารถกระทำผ่าน Java Library ที่กูเกิลได้จัดเตรียมไว้ให้ใน Android SDK หรือก็คือชุดพัฒนาซอฟต์แวร์สำหรับแอนดรอยด์นั่นเอง

2.3.1 สถาปัตยกรรมของแอนดรอยด์

ส่วนประกอบหลักๆ ของระบบปฏิบัติการแอนดรอยด์ถูกแสดงดังรูปที่ 2.11 โดยส่วนประกอบในแต่ละชั้นจะเรียกใช้บริการของชั้นที่อยู่ถัดลงไป

2.3.1.1 Linux Kernel

ส่วนที่เป็นแกนหลักหรือเคอร์เนล (Kernel) ของแอนดรอยด์นั้นความจริงก็คือเคอร์เนลของลินุกซ์ ซึ่งเป็นระบบปฏิบัติการที่สร้างโดย Linus Torvalds ในปี ค.ศ. 1991 จะทำหน้าที่เป็น Hardware Abstraction Layer กล่าวคือเป็นตัวกลางระหว่างฮาร์ดแวร์กับซอฟต์แวร์ที่อยู่ถัดขึ้นไป และทำหน้าที่บริหารจัดการทรัพยากรต่างๆ ของเครื่อง เช่น การจัดการหน่วยความจำ การจัดการโพเซส ฯลฯ ผู้ผลิตอุปกรณ์สามารถ “พอร์ต” (port) แอนดรอยด์ให้ไปรันบนฮาร์ดแวร์แบบต่างๆ ได้ โดยเปลี่ยนแปลงในส่วนนี้



รูปที่ 2.11 สถาปัตยกรรมของแอนดรอยด์

2.3.1.2 Libraries

ถัดขึ้นมาจาก Linux Kernel ก็คือส่วนที่เป็นไลบรารีของแอนดรอยด์ ซึ่งทั้งหมดเขียนด้วยภาษา C หรือ C++ และถูกคอมไพล์มาสำหรับฮาร์ดแวร์ของอุปกรณ์แต่ละรุ่น ไลบรารีที่น่าสนใจมีดังนี้

1.) Surface Manager คือ ไลบรารีจัดการส่วนแสดงผลที่มีความสามารถในการผสมกราฟฟิกทั้ง 2 มิติและ 3 มิติจากแอปพลิเคชันต่างๆ เข้าด้วยกัน ทำให้สามารถสร้างเอฟเฟ็คต์ เช่น วินโดว์ที่มองเห็นทะลุไปข้างหลังได้และ Transition ในรูปแบบต่างๆ

2.) Media Libraries คือไลบรารีที่จัดเตรียมบริการในการเล่นและบันทึกเสียง วิดีโอ และรูปภาพในฟอร์แมตต่างๆ เช่น MPEG4, H.264, MP3, AAC, AMR, JPG และ PNG

3.) SQLite คือ Database Engine ที่มีประสิทธิภาพและมีขนาดเล็ก เพื่อให้เราจัดเก็บข้อมูลของแอปพลิเคชันไว้ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

4.) WebKit คือไลบรารีที่ใช้แสดงเนื้อหาเว็บเพจ ซึ่งเป็นตัวเดียวกับที่ใช้ใน Google Chrome และ Apple Safari รวมถึงเว็บเบราว์เซอร์ในมือถือ iPhone และมือถือตระกูล S60 ของโนเกียด้วย

2.3.1.3 Android Runtime

1.) Dalvik VM (Virtual Machine) ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานในอุปกรณ์เคลื่อนที่ Dalvik VM จะแตกต่างจาก Java VM (Virtual Machine) คือ Dalvik VM จะรันไฟล์ .dex ที่คอมไพล์มาจากไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัดคลาส Java ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์เคลื่อนที่มากกว่า .class เพื่อต้องการใช้พลังงานจากแบตเตอรี่อย่างมีประสิทธิภาพสูงสุด

2.) Core Java Library ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็มี ความแตกต่างจากไลบรารีของ Java SE (Java Standard Edition) และ Java ME (Java Mobile Edition)

2.3.1.4 Application Framework

ในขั้นนี้จะอนุญาตให้นักพัฒนาสามารถเข้าเรียกใช้งาน โดยผ่าน API (Application Programming Interface) ซึ่ง Android ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งาน application component โดยในขั้นนี้ประกอบด้วยแอปพลิเคชันเฟรมเวิร์คดังนี้

1.) View System เป็นส่วนที่ใช้ในการควบคุมการทำงาน สำหรับการสร้างแอปพลิเคชัน เช่น lists, grids, text boxes, buttons และ embeddable web browser

2.) Location Manager เป็นส่วนที่จัดการเกี่ยวกับตำแหน่งของเครื่องอุปกรณ์พกพาเคลื่อนที่

- 3.) Content Provider เป็นส่วนที่ใช้ควบคุมการเข้าถึงข้อมูลที่มีการใช้งานร่วมกัน (Share data) ระหว่างแอปพลิเคชันที่แตกต่างกัน เช่น ข้อมูลผู้ติดต่อ (Contact)
- 4.) Resource Manager เป็นส่วนที่จัดการข้อมูลต่างๆ ที่ไม่ใช่ส่วนของโค้ดโปรแกรม เช่น รูปภาพ, localized strings, layout ซึ่งจะอยู่ในไดเรกทอรี res/
- 5.) Notification Manager เป็นส่วนที่ควบคุมอีเวนต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status bar) เช่น ในกรณีที่ได้รับข้อความหรือสายที่ไม่ได้รับ และการแจ้งเตือนอื่นๆ เป็นต้น
- 6.) Activity Manager เป็นส่วนควบคุม Life Cycle ของแอปพลิเคชัน

2.3.1.5 Application

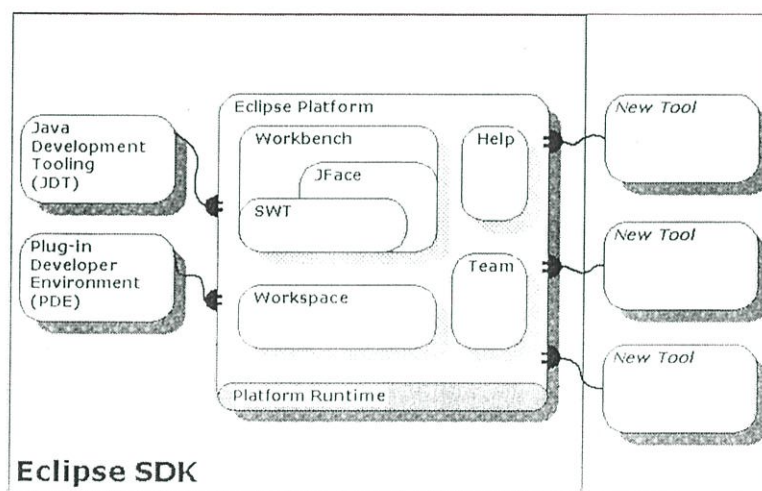
ส่วนบนสุดของสถาปัตยกรรมแอนดรอยด์ ก็คือแอปพลิเคชันต่างๆ ทั้งที่ติดตั้งมากับเครื่องอยู่แล้ว (Core Application) เช่น Phone dialer, Email, Contacts, Web browser และ Google Play (Android Market เดิม) เป็นต้น รวมถึงแอปพลิเคชันที่เราสร้างขึ้นด้วย ซึ่งแอปพลิเคชันทั้งหมดในส่วนนี้จะเขียนด้วยภาษาจาวา อยู่ในรูปแบบของไฟล์ .apk โดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app

2.3.2 อีคลิพส์ (Eclipse)

Eclipse เป็นชุดโปรแกรมสนับสนุนการพัฒนาระบบสำหรับใช้ในการพัฒนาซอฟต์แวร์ มีองค์ประกอบหลักที่เรียกว่า Eclipse platform ซึ่งรวบรวมเครื่องมือพัฒนาซอฟต์แวร์ต่างๆ จากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมีองค์ประกอบที่เรียกว่า Plug in Development Environment (PDE) ใช้เพิ่มความสามารถในการพัฒนาซอฟต์แวร์ เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug เมื่อดาวน์โหลดมาติดตั้งจะมี Plug-in ชื่อ Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ debug โปรแกรมภาษาจาวา

การทำงานของ Eclipse จะใช้ Workbench UI เป็นตัวหลักในการแสดงข้อมูลต่างๆ อีกทั้งยังรองรับการติดตั้ง plug-in โดยจะทำหน้าที่จัดการการแสดงผลหน้าจอต่างๆ ของ plug-in ซึ่งจะใช้ชุดคำสั่งของ Jface และ SWT ในการแสดงผลหน้าจอ

Eclipse platform ดังรูปที่ 2.12 มีโครงสร้างของ plug-in ประกอบไปด้วยข้อมูลรวมทั้งฟังก์ชันต่างๆ ที่เข้ากับระบบของ Eclipse ได้ โดย plug-in ที่ถูกพัฒนานั้นจะอยู่ในรูปของ code libraries (อยู่ในรูปของ file Java classes) ซึ่งการเพิ่มการทำงานของ plug-in ลงไปใน Eclipse นั้นผู้พัฒนา plug-in สามารถกำหนดตำแหน่งการแสดงผลในโปรแกรม Eclipse โดยอาศัย function ย่อยที่อยู่ใน Eclipse platform



รูปที่ 2.12 Eclipse platform Architecture

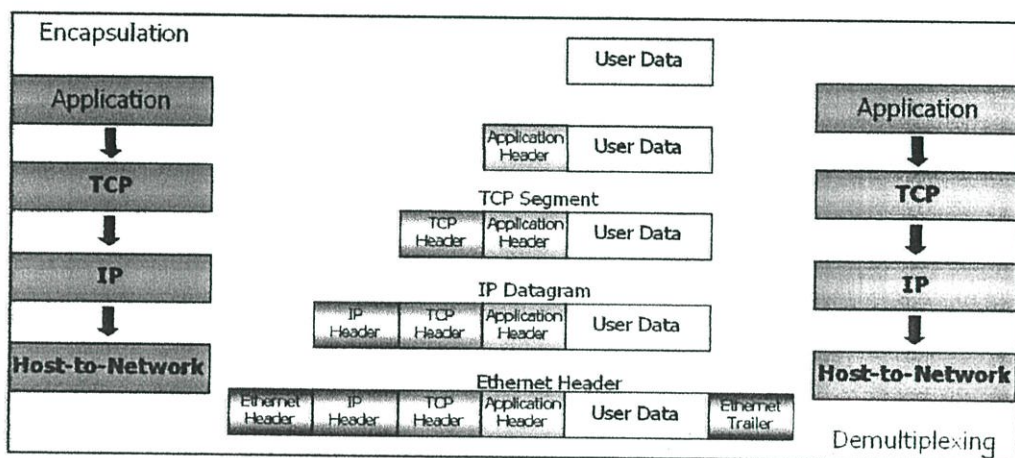
2.4 โพรโตคอลที่ใช้ในการสื่อสารข้อมูล

2.4.1 TCP/IP (Transmission Control Protocol/Internet Protocol)

อินเทอร์เน็ตเป็นเครือข่ายขนาดใหญ่ที่สุดในโลก ประกอบด้วยเครือข่ายคอมพิวเตอร์ และเครือข่ายของเครือข่าย เครือข่ายย่อยจำนวนมากต่อเชื่อมกันด้วย TCP/IP คอมพิวเตอร์ในอินเทอร์เน็ตทุกเครื่องจึงต้องใช้โปรโตคอล TCP/IP เพื่อสื่อสารระหว่างกัน จึงเป็นโปรโตคอลที่ได้รับการยอมรับว่ามีคนใช้งานมากที่สุดในปัจจุบัน

TCP เป็นโปรโตคอลที่ทำงานในระดับชั้นทรานสปอร์ต ในชั้นนี้ทำให้สถานีรับส่งแต่ละสถานีรับข้อมูลจากหลายงานวิ่งรวมเข้าไปด้วยกันได้

IP เป็นโปรโตคอลที่ทำงานในระดับชั้นอินเทอร์เน็ต ทำหน้าที่หาเส้นทางที่ดีที่สุดในการนำส่งข้อมูลในเครือข่าย และใช้ IP เพื่อนำเอาข้อมูลใส่เป็นแพ็กเก็ต แล้วส่งไปยังแอดเดรสปลายทางที่ต้องการดังที่แสดงในรูปที่ 2.13



รูปที่ 2.13 ขั้นตอนการ Encapsulation และ Demultiplexing ในการรับส่งข้อมูล

2.4.2 Socket

ซ็อกเก็ต คือ จุดปลายของการติดต่อสื่อสารในชั้นของทรานสปอร์ตเลเยอร์ (Transport layer) ในชั้นนี้เรามักจะกล่าวในระดับของระบบเครือข่ายคือติดต่อสื่อสารกันระหว่างเซิร์ฟเวอร์และไคลเอนต์ในลักษณะของจุดต่อจุด การสื่อสารโดยใช้ซ็อกเก็ตนี้จะมีส่วนสำคัญอยู่สองส่วนคือไคลเอนต์และเซิร์ฟเวอร์โครงสร้างข้อมูลของซ็อกเก็ตนั้นมีหลากหลายโครงสร้างจะขึ้นอยู่กับนำไปใช้ เช่น จุดปลาย (End-Point) ระหว่างโพรเซส (Process) ในเครื่องเดียวกัน จุดปลายระหว่างโพรเซสของเครือข่าย เป็นต้น

SOCK_STREAM ช็อกเก็ตประเภทนี้จะใช้สำหรับส่วนที่เรียกว่า Connection-Oriented คือจะใช้ ทีซีพี โพรโตคอลในการส่งจะต้องมีการสร้างการติดต่อกันระหว่าง เซสชัน (Session) ทั้งสองด้านเสียก่อนแล้วจึงส่งข้อมูล โดยลักษณะการส่งข้อมูลนั้นจะทำการส่งแบบ Stream Oriented Protocol หมายความว่า การรับส่งข้อมูลนั้นจะไม่คำนึงถึงปริมาณของข้อมูลที่ส่งไป แต่จะแบ่งข้อมูลออกเป็นส่วนย่อย ๆ แล้วบรรจุลงในกลุ่มข้อมูลแต่ละกลุ่มข้อมูลจะมี ส่วนประกอบสองส่วน ส่วนแรกคือ ส่วนหัวเป็นข้อมูลเกี่ยวกับที่อยู่และพอร์ตของผู้รับและผู้ส่ง รวมทั้งข้อมูลเท่าที่จำเป็นในการนำกลุ่มข้อมูลมาประกอบกันเป็นข้อมูลดั้งเดิม อีกส่วนเรียกว่า เพย์-โหลด (Payload) คือ ข้อมูลย่อยที่จะถูกส่งไปนั้นเองจากนั้นจึงจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล หากในกรณีที่ส่วนใดส่วนหนึ่งสูญหายไปจะทำการส่งใหม่ ซึ่งปลายทางจะจัดการเรียงข้อมูลเอง ซึ่งการใช้งานทีซีพี จะใช้หน่วยความจำและขนาดของช่องสัญญาณ (Bandwidth) มากกว่าการส่งแบบยูดีพี

ในการเขียนโปรแกรมติดต่อกับช็อกเก็ต จะมีการอ้างอิงคู่ลำดับของ หมายเลขไอพีกับเบอร์พอร์ต ซึ่งมีเบอร์พอร์ตแตกต่างกันไปตามการใช้งานดังนี้

- 1) 0 ไม่ใช้งาน
- 2) 1-1023 จองไว้สำหรับบริการที่เป็นที่รู้จักกันดี เช่น TELNET 23
- 3) 1024-65535 อนุญาตให้ผู้ใช้ใช้งานได้

2.5 Wireless LAN

Wireless เป็นเทคโนโลยีที่ใช้ในการส่งภาพ เสียง และข้อมูล จากอีกที่หนึ่งไปอีกที่หนึ่ง โดยไม่มีใช้สายไฟ แต่ใช้คลื่นวิทยุหรืออาจใช้คลื่น Infrared เป็นตัวกลางในการสื่อสาร ส่วน LAN มาจากคำว่า Local Area Network คือระบบเครือข่ายในระยะทางใกล้ๆ เช่น ภายในห้องทำงาน การเชื่อมต่อระหว่างห้องทำงาน ระหว่างชั้น หรือระหว่างตึก

ดังนั้น Wireless LAN หมายถึง การติดต่อสื่อสารในระยะทางใกล้ๆ โดยไม่มี ใช้ สายไฟ และอุปกรณ์สื่อสารคือ Mobile Device หรือ Notebook ซึ่งระบบเครือข่ายแบบไร้สาย หรือ Wireless LAN ที่นิยมกันมากที่สุดจะเป็นแบบไร้สายตามมาตรฐานของ IEEE 802.11 หรือที่เรียก

กันว่า Wi-Fi จะต้องประกอบไปด้วยอุปกรณ์อย่างน้อย 2 สิ่ง คือ อุปกรณ์ส่งสัญญาณ (Access Point) และอุปกรณ์รับสัญญาณ (Wireless Card)

2.5.1 มาตรฐานการใช้งานเทคโนโลยี Wi-Fi

Wi-Fi (Wireless Fidelity) คือองค์การหนึ่ง ที่ทำหน้าที่ทดสอบผลิตภัณฑ์ Wireless LAN หรือระบบ Network แบบไร้สายภายใต้เทคโนโลยีการสื่อสาร ภายใต้มาตรฐาน IEEE 802.11 ว่าอุปกรณ์ทุกตัวซึ่งต่างยี่ห้อกันนั้นมันสามารถติดต่อสื่อสารกันได้โดยไม่มีปัญหา

Wi-Fi ใช้มาตรฐาน IEEE 802.11 ซึ่งสามารถส่งข้อมูลโดยใช้ความถี่ในลักษณะคล้ายกับการส่งสัญญาณวิทยุ โดยมีช่วงความถี่ที่ใช้ในการส่งข้อมูลทั้งหมด 3 ช่วง ดังนี้

2.5.1.1 ช่วงความถี่ 900 MHz

สามารถรับ/ส่งข้อมูลได้ที่ความเร็วเพียง 1-2 Mbps

2.5.1.2 ช่วงความถี่ 2.4 GHz

สามารถรับ/ส่งข้อมูลได้ที่ความเร็วตั้งแต่ 1 Mbps ถึง 300 Mbps ครอบคลุมพื้นที่ในการรับ/ส่งข้อมูลได้ในระยะทางที่น้อยกว่าช่วงความถี่ 900 MHz ในช่วงความถี่ 2.4 GHz ยังแบ่งย่อยออกเป็นมาตรฐานต่างๆ อีก เช่น

1.) มาตรฐาน IEEE 802.11b รับ/ส่งข้อมูลได้ความเร็วสูงสุดที่ 11 Mbps

2.) มาตรฐาน IEEE 802.11g ีความเร็วสูงสุดที่ 54 Mbps

3.) มาตรฐาน IEEE 802.11n ที่มีความเร็วสูงกว่า 300 Mbps ขึ้นอยู่กับระยะห่างระหว่าง Access Point กับ Wireless Card

2.5.1.3 ช่วงความถี่ 5 GHz

เป็นช่วงความถี่ที่สามารถรับ/ส่งข้อมูลได้ในระยะทางที่สั้นที่สุดในทั้ง 3 ช่วงความถี่ ซึ่งมีความเร็วที่ 54 Mbps

2.5.2 ลักษณะการเชื่อมต่อของอุปกรณ์

Wi-Fi ได้กำหนดลักษณะการเชื่อมต่อของอุปกรณ์ภายในเครือข่าย WLAN ไว้ 2 ลักษณะคือโหมด Infrastructure และโหมด Ad-Hoc หรือ Peer-to-Peer

2.5.2.1 โหมด Infrastructure

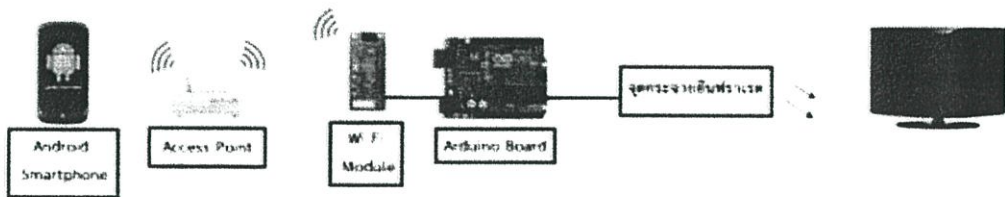
โดยทั่วไปแล้วอุปกรณ์ในเครือข่าย Wi-Fi จะเชื่อมต่อกันในลักษณะของโหมด Infrastructure ซึ่งเป็นโหมดที่อนุญาตให้อุปกรณ์ภายใน WLAN สามารถเชื่อมต่อกับเครือข่ายอื่นได้ ในโหมด Infrastructure นี้จะประกอบไปด้วยอุปกรณ์ 2 ประเภทได้แก่ สถานีผู้ใช้ (Client Station) ซึ่งก็คืออุปกรณ์คอมพิวเตอร์ (Desktop, Laptop หรือ PDA ต่างๆ) ที่มีอุปกรณ์ Client Adapter เพื่อใช้รับส่งข้อมูลผ่าน Wi-Fi และสถานีแม่ข่าย (Access Point) ซึ่งทำหน้าที่ต่อเชื่อมสถานีผู้ใช้เข้ากับเครือข่ายอื่น (ซึ่งโดยปกติจะเป็นเครือข่าย IEEE 802.3 Ethernet LAN) การทำงานในโหมด Infrastructure มีพื้นฐานมาจากระบบเครือข่ายโทรศัพท์มือถือ กล่าวคือสถานีผู้ใช้จะสามารถรับส่งข้อมูลโดยตรงกับสถานีแม่ข่ายที่ให้บริการแก่สถานีผู้ใช้นั้นอยู่เท่านั้น ส่วนสถานีแม่ข่ายจะทำหน้าที่ส่งต่อ (forward) ข้อมูลที่ได้รับจากสถานีผู้ใช้ไปยังจุดหมายปลายทางหรือส่งต่อข้อมูลที่ได้ รับจากเครือข่ายอื่นมายังสถานีผู้ใช้

2.5.2.2 Ad-Hoc หรือ Peer-to-Peer

เครือข่าย Wi-Fi ในโหมด Ad-Hoc หรือ Peer-to-Peer เป็นเครือข่ายที่ปิดคือไม่มีสถานีแม่ข่ายและไม่มีการเชื่อมต่อกับเครือข่ายอื่น บริเวณของเครือข่าย Wi-Fi ในโหมด Ad-Hoc จะถูกเรียกว่า Independent Basic Service Set (IBSS) ซึ่งสถานีผู้ใช้หนึ่ง สามารถติดต่อสื่อสารข้อมูลกับสถานีผู้ใช้อื่นๆในเขต IBSS เดียวกันได้โดยตรงโดยไม่ต้องผ่านสถานีแม่ข่าย แต่สถานีผู้ใช้จะไม่สามารถรับส่งข้อมูลกับเครือข่ายอื่นๆได้

บทที่ 3

การออกแบบและการจัดทำปฏิญาณนิพนธ์



รูปที่ 3.1 บล็อกไดอะแกรมของปฏิญาณนิพนธ์

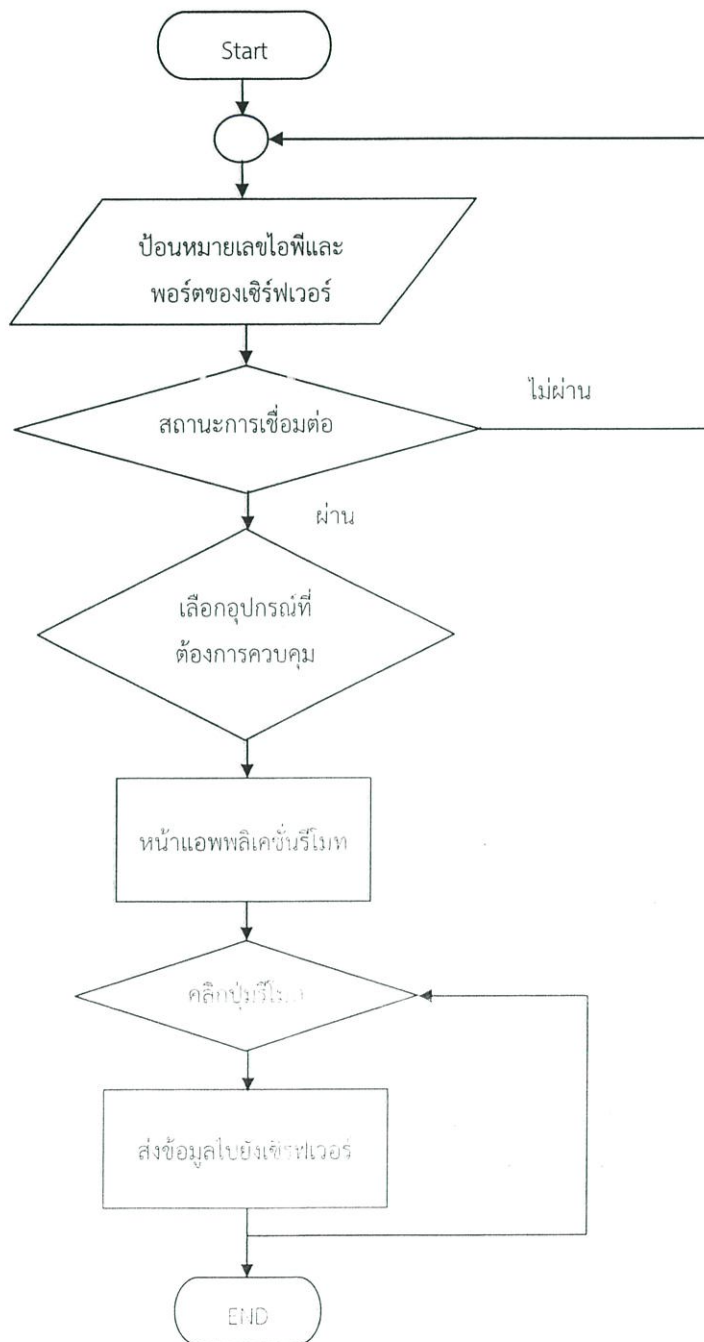
จากบล็อกไดอะแกรมดังรูปที่ 3.1 การทำงานของระบบเริ่มต้นจากการกดปุ่มบนหน้าจอแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ที่ทำการเชื่อมต่อกับอินเทอร์เน็ตไว้แล้ว ซึ่งในส่วนโทรศัพท์เคลื่อนที่นั้นจะเป็นส่วนของไคลเอ็นท์ โดยค่าที่ป้อนจะถูกส่งไปยังส่วนของเซิร์ฟเวอร์คือบอร์ด Arduino เพื่อทำการควบคุมบอร์ด Arduino ให้ส่งการไปยังจุดกระจายสัญญาณอินฟราเรดซึ่งเป็นวงจรส่งสัญญาณอินฟราเรดทั่วไปให้ส่งสัญญาณไปควบคุมอุปกรณ์รับสัญญาณอินฟราเรดอีกทีหนึ่ง สำหรับโครงการนี้เราจะใช้โทรศัพท์เป็นอุปกรณ์รับสัญญาณอินฟราเรด และนำการเชื่อมต่อแบบ Wi-Fi มาใช้รองรับการทำงานระหว่างไคลเอ็นท์และเซิร์ฟเวอร์

จากระบบข้างต้น จะสามารถแบ่งงานออกเป็น 4 ส่วนหลักๆ คือ การสร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์, การสื่อสารข้อมูลผ่าน Wi-Fi, การหาข้อมูลคำสั่งของรีโมท และการส่งสัญญาณอินฟราเรด ซึ่งได้อธิบายไว้ในหัวข้อถัดไปนี้

3.1 การออกแบบ

3.1.1 การสร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

เขียนแผนผังแสดงขั้นตอนการทำงานของแอปพลิเคชันดังรูปที่ 3.2



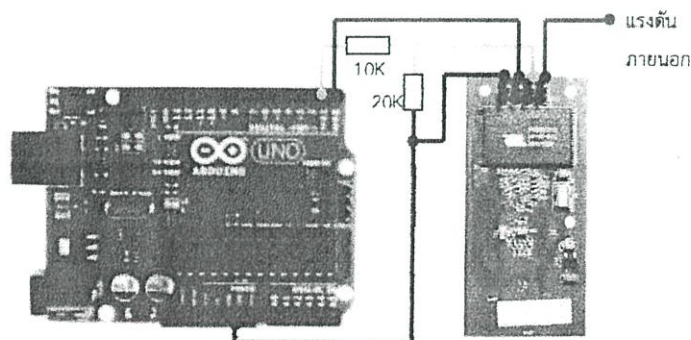
รูปที่ 3.2 แผนผังการทำงานของแอปพลิเคชัน

ส่วนนี้จะเป็นส่วนของโคลเอนท์ การทำงานจะเริ่มต้นจากการที่ผู้ใช้ใส่ค่า IP Address ของเซิร์ฟเวอร์ (บอร์ด Arduino) จากนั้นจึงทำการเชื่อมต่อ หากเชื่อมต่อสำเร็จ ผู้ใช้สามารถเลือกได้ว่าจะอุปกรณ์ขึ้นใด โดยเมื่อเลือกอุปกรณ์เสร็จแล้ว จะปรากฏหน้าต่างที่ประกอบไปด้วยปุ่มรีโมทที่ใช้ในการควบคุม เมื่อกดปุ่มก็จะมีคำสั่งไปยังเซิร์ฟเวอร์ และสามารถทำงานไปได้เรื่อยๆ จนกว่าผู้ใช้จะปิดแอปพลิเคชัน

3.1.2 การสื่อสารข้อมูลผ่าน Wi-Fi

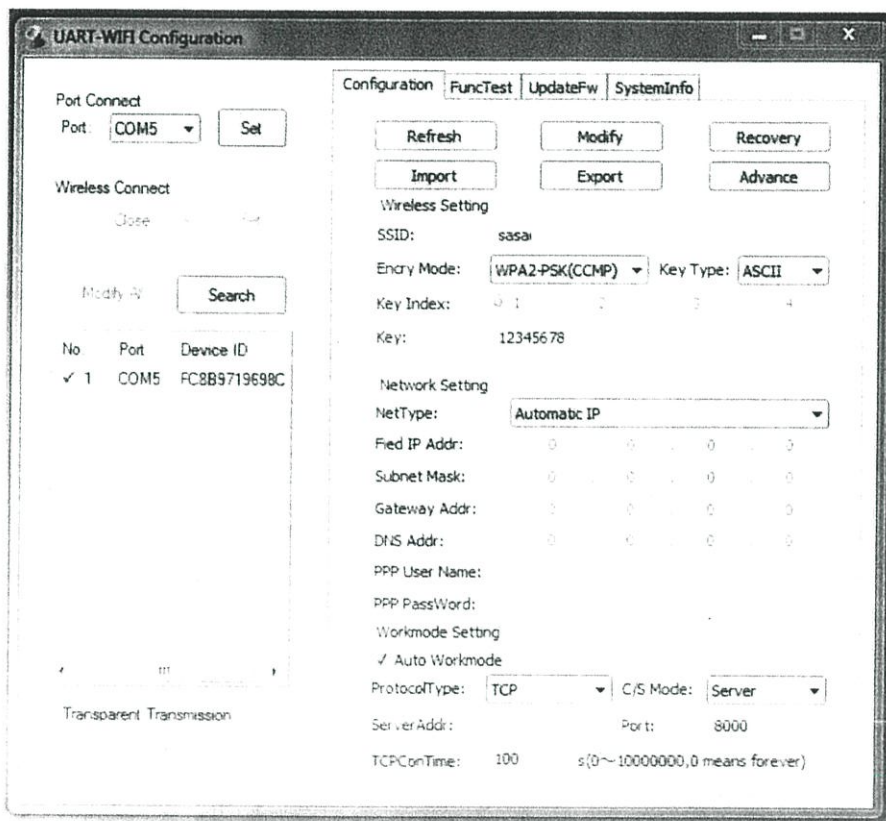
ในส่วนนี้จะมีการเชื่อมต่อบอร์ด Arduino กับโมดูล Wi-Fi เพื่อให้บอร์ด Arduino สามารถสื่อสารรับข้อมูลผ่าน Wi-Fi ได้ โดยจะให้โมดูล Wi-Fi ทำงานเป็น TCP Server โดยใช้ IP Address รับข้อมูลผ่านพอร์ตที่กำหนด ซึ่งกำหนดให้โมดูลทำงานในโหมด Auto Work ทำการเชื่อมต่อกับ Access Point และรับข้อมูลในรูปแบบของ Wi-Fi มาตรฐาน IEEE 802.11b/g ที่ส่งมาจากโทรศัพท์เคลื่อนที่

ข้อมูลเหล่านี้จะถูกแปลงให้อยู่ในรูปแบบ UART และส่งเข้า Pin 0 ซึ่งเป็นขารับสัญญาณจากภายนอกของบอร์ด Arduino แต่เนื่องจากโมดูล Wi-Fi ใช้ระดับสัญญาณเป็น LVTTTL คือ 0 - 3.3 โวลต์ ซึ่งต่างจากบอร์ด Arduino ที่ใช้ระดับสัญญาณในระดับ TTL คือ 0 - 5 โวลต์ จึงจำเป็นต้องมีการลดทอนแรงดันที่บอร์ด Arduino จะจ่ายแรงดันไปยังโมดูล Wi-Fi คือขาที่ Tx ของบอร์ด Arduino ส่งไปยังขา Rx ของโมดูล Wi-Fi โดยใช้วงจร R Divider ซึ่งมีการเชื่อมต่อดังรูปที่ 3.3



รูปที่ 3.3 การต่อโมดูล Wi-Fi เข้ากับบอร์ด Arduino

การตั้งค่าโมดูล Wi-Fi ให้เชื่อมต่อกับ Access point จะต้องเลือก Port ที่เชื่อมต่อและตั้งค่าอัตราการรับส่งข้อมูลให้โมดูล (Baud Rate) เท่ากับ 115200 bps Data Bit = 8 , Parity Bit = None , Stop Bit = 1 จากนั้นให้ออกจากการทำงานแบบ Auto Work mode ก่อนเพื่อให้เราสามารถติดต่อกับโมดูลได้ และจะสามารถค้นหา Access point เพื่อทำการตั้งค่าได้ ดังรูปที่ 3.4



รูปที่ 3.4 การตั้งค่าโมดูล Wi-Fi

จากรูปที่ 3.4 เมื่อสามารถติดต่อกับโมดูลได้ จะพบ MAC ของโมดูลคือ FC8B9719698C เชื่อมต่อผ่าน COM5 และมีการตั้งค่าต่างๆ ดังนี้

1) Wireless Setting

SSID : sasai (ชื่อของเครือข่าย)

Encry Mode : WPA2-PSK (การเข้ารหัส)

Key Type : ASCII

Key : 12345678

2) Workmode Setting

เลือก Auto Work Mode

ใช้ TCP Protocol

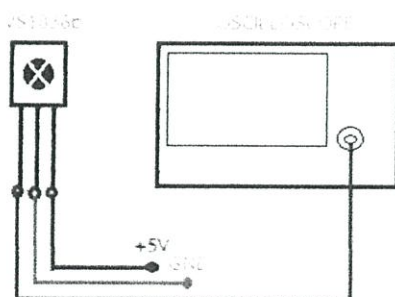
ทำงานเป็น Server ผ่าน Port 8000

3.1.3 การหาข้อมูลคำสั่งของรีโมท

ก่อนจะหาตัวคำสั่งข้อมูลคำสั่งของปุ่มกดบนรีโมทได้ เราต้องทราบลักษณะและวิธีการส่งสัญญาณของสัญญาณเสียก่อน จึงต้องแบ่งการทดลองออกเป็นสองส่วนคือ การหารูปแบบข้อมูลของสัญญาณอินฟราเรดที่ส่งจากรีโมทด้วยออสซิลโลสโคป กับการหาสัญญาณคำสั่งของปุ่มกดด้วยโปรแกรม

3.1.3.1 หารูปแบบข้อมูลของสัญญาณอินฟราเรดด้วยออสซิลโลสโคป

ใช้วงจรรับสัญญาณอินฟราเรดรับสัญญาณจากรีโมทแล้วแสดงผลสัญญาณด้วยเครื่องออสซิลโลสโคปดังรูปที่ 3.5 ซึ่งวงจรรับสัญญาณอินฟราเรดจะใช้ตัวรับอินฟราเรดเบอร์ VS1838B ซึ่งตัวรับนี้จะใช้ไฟเลี้ยง 5 โวลต์



รูปที่ 3.5 การวัดแอมพลิจูดของตัวรับสัญญาณอินฟราเรดด้วยเครื่องออสซิลโลสโคป

3.1.3.2 โปรแกรมอ่านค่าและแสดงผลสัญญาณจากรีโมท

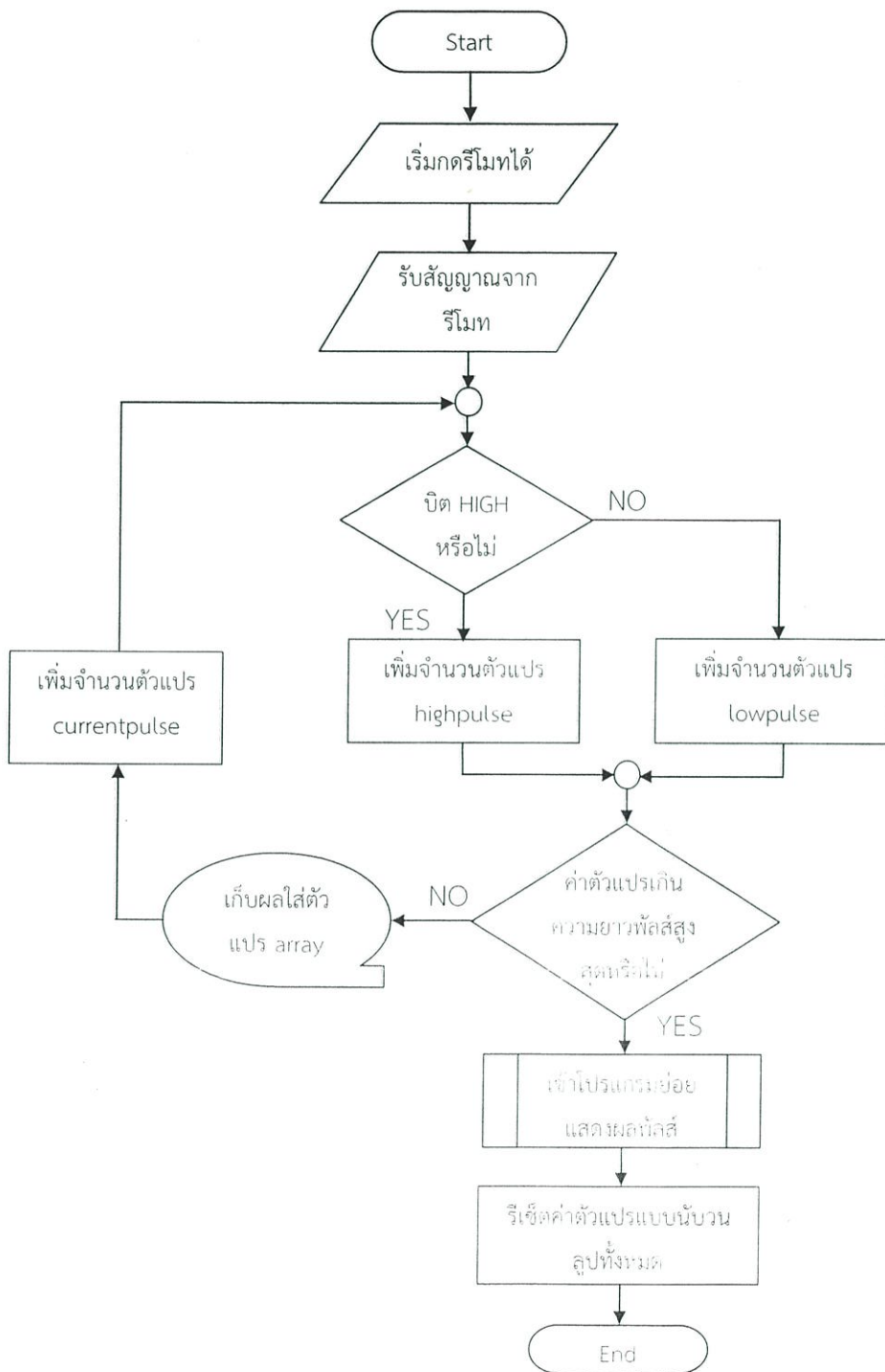
ขั้นตอนนี้ต้องมีทั้งการอ่านค่าและแสดงผลสัญญาณโดยใช้บอร์ด Arduino เป็นตัวประมวลผล ทำให้สามารถเขียนโปรแกรมได้ด้วยโปรแกรมเขียนคำสั่งของบอร์ด Arduino ที่ชื่อว่า Arduino IDE สรุปขั้นตอนได้ดังนี้

นับรอบลูปการทำงาน

- 1.) กำหนดตัวแปรหลักที่ใช้ค่าเดิมทั้งโปรแกรม
- 2.) กำหนดขา pin ที่เป็นอินพุต และความเร็ว baud rate
- 3.) กำหนดตัวแปรที่จะมีการเปลี่ยนแปลงค่า หรือตัวแปรที่ใช้
- 4.) กำหนดตัวแปรแบบ array เพื่อใช้เก็บข้อมูล
- 5.) รับสัญญาณจากรีโมท
- 6.) อ่านค่าว่าเป็นบิต 0 หรือบิต 1
- 7.) แสดงผล
- 8.) ล้างค่าตัวแปรทั้งหมด
- 9.) จบการทำงาน

ซึ่งขั้นตอนประมวลผลทั้งหมดนี้ สามารถแสดงเป็นผังงาน

(Flowchart) ได้ดังรูปที่ 3.6

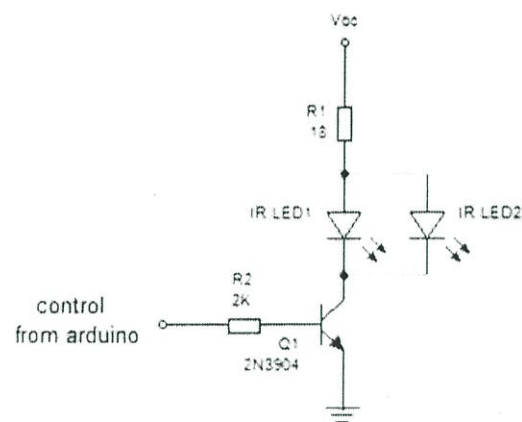


รูปที่ 3.6 แผนผังการทำงานของโปรแกรมอ่านสัญญาณอินฟราเรด

3.1.4 การส่งสัญญาณอินฟราเรด

3.1.4.1 วงจรส่งสัญญาณอินฟราเรด

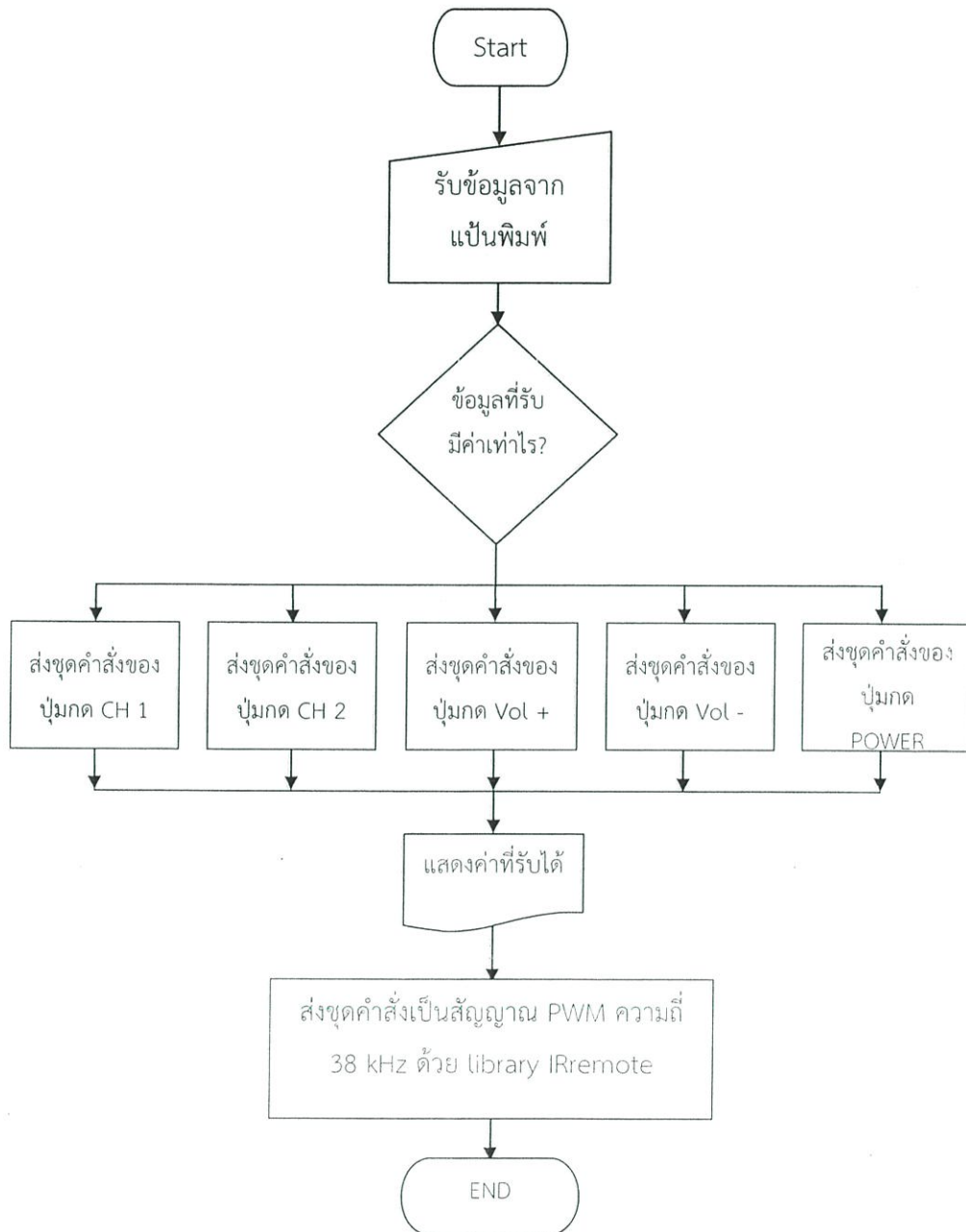
ออกแบบโดยใช้วงจรถานซิสเตอร์ทั่วไปเป็นตัวส่ง LED ซึ่งจะต้องใช้ทรานซิสเตอร์ที่เหมาะสมกับการสวิตซ์ซึ่ง และสามารถไต่กระแสสูงๆได้ เพื่อควบคุมการเปิด-ปิดกระแส และมีขาควบคุมซึ่งจะต้องนำไปต่อกับขาสัญญาณเอาต์พุตของบอร์ด Arduino (Digital Pin 3) และด้วยเหตุผลที่ว่าต้องการให้สัญญาณที่ส่งออกแรงขึ้น สามารถส่งได้ในระยะที่ไกลขึ้น จึงได้ออกแบบให้วงจรถานซิสเตอร์อินฟราเรดมี LED สองตัวต่อแบบขนานกันเป็นดังรูปที่ 3.7



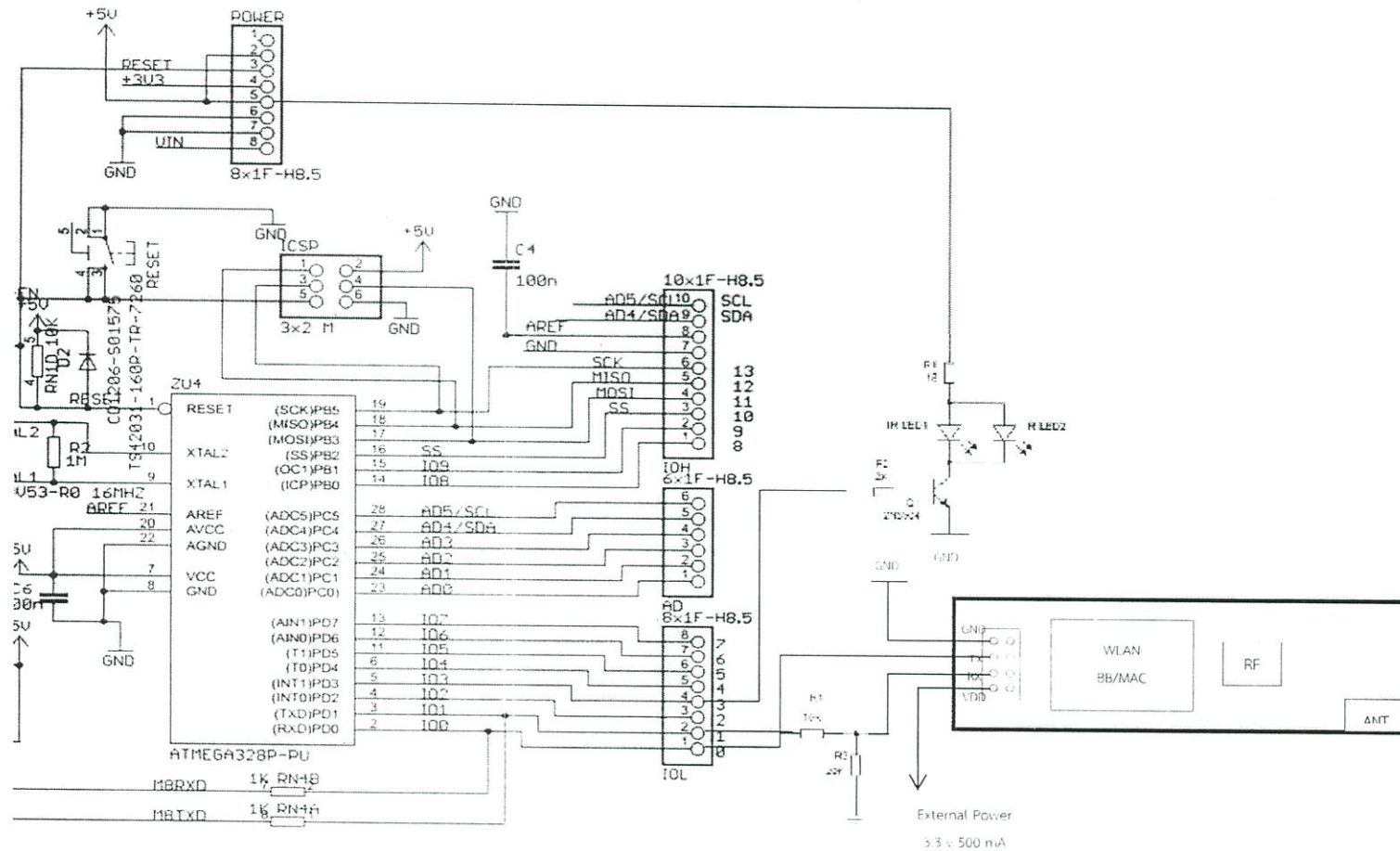
รูปที่ 3.7 วงจรส่งสัญญาณอินฟราเรดที่ใช้ทดลอง

3.1.4.2 โปรแกรมส่งสัญญาณอินฟราเรด

โปรแกรมที่ใช้ส่งสัญญาณจะเริ่มทำงานโดยการรับคำสั่งจากทางคีย์บอร์ด เมื่อได้คำสั่งมาแล้วก็จะทำการประมวลผล แล้วส่งสัญญาณออกไปด้วย library ที่มีอยู่คือ IRremote โดยกระบวนการทำงานของโปรแกรมสามารถแสดงได้ดังรูปที่ 3.8 โดยในการต่อวงจรจริงนั้น จะต่อกับบอร์ด Arduino ดังที่แสดงในรูปที่ 3.9



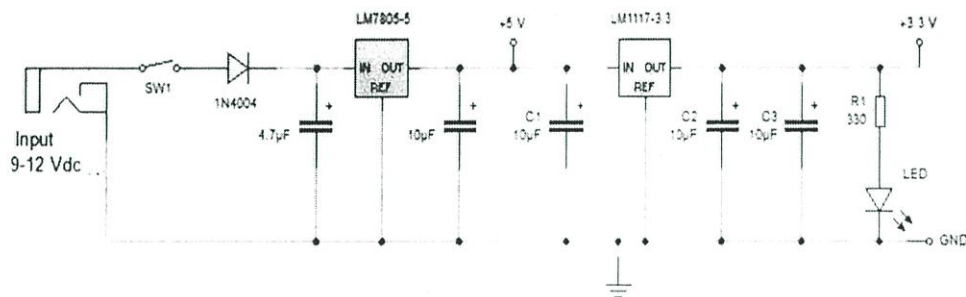
รูปที่ 3.8 แผนผังการทำงานของโปรแกรมส่งสัญญาณอินฟราเรด



รูปที่ 3.9 การต่อวงจรรวมในส่วนของบอร์ด Arduino เพื่อรับข้อมูลผ่าน Wi-Fi และส่งสัญญาณอินฟราเรดออก

3.1.5 วงจรจ่ายไฟ

เนื่องจากอุปกรณ์ที่ใช้ในการทดลองแต่ละชนิดใช้แรงดันอินพุทไม่เท่ากัน โดยบอร์ด Arduino จะใช้แรงดันอินพุทที่ 5 โวลต์ และโมดูล Wi-Fi ใช้แรงดันอินพุทที่ 3.3 โวลต์ จึงต้องมีการออกแบบวงจรจ่ายไฟรวมให้อุปกรณ์ทั้งหมดดังรูปที่ 3.10



รูปที่ 3.10 วงจรจ่ายไฟ

3.2 เครื่องมือที่ใช้ในการทดลอง

1. โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ (Android Smartphone)

ผู้ใช้สามารถเปิดใช้งานแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อควบคุมอุปกรณ์รับสัญญาณอินฟราเรด ได้แก่ โทรศัพท์ โดยแอปพลิเคชันนี้จะเป็นวิธีที่ง่ายที่สุดให้ผู้ใช้งานเลือกสั่งการตามต้องการ เช่น เปิด-ปิด และเปลี่ยนช่องสัญญาณ

2. อีคลิปส์ (Eclipse)

เป็นชุดโปรแกรมสนับสนุนการพัฒนาสำหรับใช้ในการพัฒนาซอฟต์แวร์ มีองค์ประกอบหลักที่เรียกว่า Eclipse platform ซึ่งรวบรวมเครื่องมือพัฒนาซอฟต์แวร์ต่างๆ จากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน ซึ่งจะมี Emulator เพื่อใช้ทดสอบแอปพลิเคชันที่เขียนขึ้นมาโดยไม่จำเป็นต้องมีโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ และโปรแกรม NetBeans IDE ใช้สำหรับเขียนโปรแกรมเซิร์ฟเวอร์ให้เปิดพอร์ตคอมพิวเตอร์เพื่อรับการเชื่อมต่อจาก Emulator

3. โมดูล Wi-Fi (M03 – LVTTTL UART to Wi-Fi)

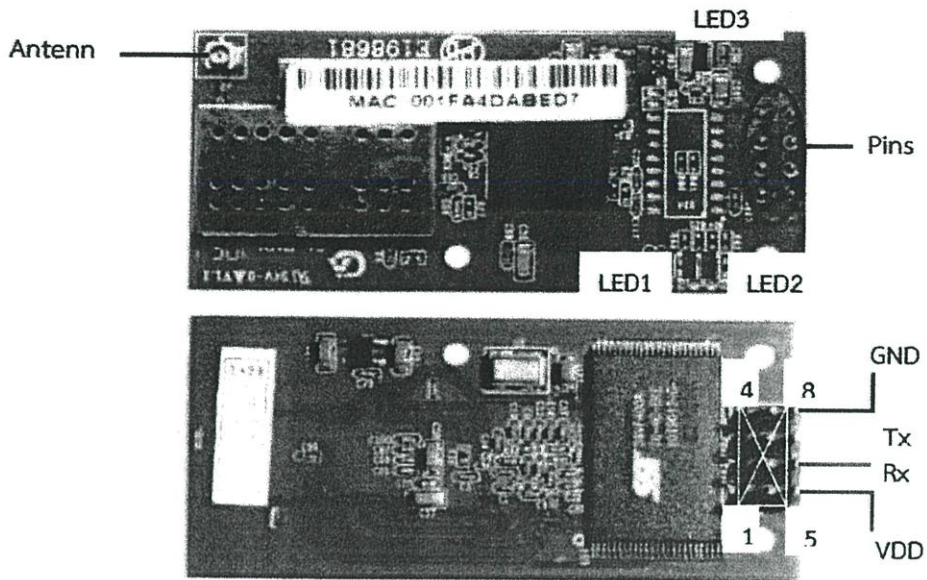
โมดูลนี้มีหน้าที่แปลงการรับส่งข้อมูลในรูปแบบ UART เป็นการรับส่งข้อมูลในรูปแบบของ Wireless LAN หรือ Wi-Fi (IEEE 802.11 b/g) ซึ่งภายในโมดูลมี Software TCP/IP Stack อยู่ทำให้ใช้งานได้ง่าย สะดวกและรวดเร็ว เหมาะสำหรับนำมาใช้กับระบบประมวลผลหรือไมโครคอนโทรลเลอร์ที่มีความเร็วในการประมวลผลต่ำ ใช้หน่วยความจำน้อยและลดเวลาในการพัฒนาลง โดยโมดูลมีการจัดสรร Software ที่เป็นส่วนของ TCP/IP Stack และการควบคุม Hardware ที่ใช้รับส่งข้อมูลผ่านทาง Wireless ไว้แล้ว เพียงตั้งค่าโมดูลให้สามารถเชื่อมต่อกับระบบ Network ผ่านทาง Software ก็สามารถสื่อสารข้อมูลได้แล้ว นอกจากนี้ยังสามารถสั่งงานหรือเปลี่ยนแปลงค่าต่างๆ ผ่านทาง AT Command ด้วยการเขียนโปรแกรมจากไมโครคอนโทรลเลอร์ได้ด้วย

คุณสมบัติของ M03 – LVTTTL UART to Wi-Fi

- สามารถรับส่งข้อมูลผ่านทาง UART ที่ความเร็ว 1200 – 115200 bps
- ใช้กระแสไฟฟ้าตรง 3.3 โวลต์
- รองรับการรับส่งข้อมูลผ่าน Wireless มาตรฐาน IEEE 802.11 b/g
- ใช้ช่วงความถี่ 2.412 -2.484 GHz
- รองรับการใช้งานแบบ Ad hoc และ Infrastructure
- รองรับมาตรฐานความปลอดภัย WEP64/WEP128/TKIP/CCMP(AES) /WEP/WPA-PSK/WPA2-PSK
- รองรับ Network Protocol แบบTCP/UDP/ICMP/DHCP/DNS/HTTP

การทำงานของโมดูลจะให้ทำงานในโหมด Auto work มีรายละเอียดดังนี้

- โมดูลทำตัวเป็น Transparent transmission mode คือไมโครคอนโทรลเลอร์สามารถสื่อสารโดยตรงกับเซิร์ฟเวอร์ได้โดยไม่ต้องใช้คำสั่งใดๆ ไปยังตัวโมดูล (Plug and Play serial)
- ในโหมดนี้ผู้ใช้ควรตั้งค่าพารามิเตอร์เท่าที่จำเป็น
- หลังจากจ่ายไฟให้โมดูล โมดูลจะเชื่อมต่อเครือข่ายไร้สายและเซิร์ฟเวอร์โดยอัตโนมัติ



รูปที่ 3.11 Interface diagram ของโมดูล M03 – LVTTTL UART to Wi-Fi

Pin ของโมดูลมีทั้งหมด 8-pin โดยทั่วไปมีการใช้งานแค่ pin 5-8 เท่านั้น ดังรูปที่ 3.11 และ pin 1-4 เป็น Optional feature pin ซึ่งแต่ละขามีรายละเอียดดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดขา Pin ของโมดูล Wi-Fi

No	Function	Direction	Description
1	nRTS/nREADY	O	เชื่อมต่อกับ LED1 เพื่อแสดงสถานะ LED ดับ : ต่อกับเครือข่ายสำเร็จ/ready LED ติด : ไม่ได้ต่อกับเครือข่าย
2	nCTS/MODE/ GPIO	I/O	เชื่อมต่อกับ LED2 โหมดทำงาน : ขา CTS โหมดตั้งค่า : หลังรีเซ็ตภายใน 300 ms ถ้า ปุ่ม High จะเข้าสู่โหมดตั้งค่า ถ้าปุ่ม Low จะเป็นโหมดปกติ
3	Link	O	ใช้ในการติดตั้งการส่งแบบไร้สายและ เชื่อมต่อกับ LED3 จะกระพริบแสดง auto work mode หรือ scan หาเครือข่าย

No	Function	Direction	Description
4	nRESET	I	ป้อนสัญญาณ low เพื่อรีเซ็ต
5	VDD	I	กระแสไฟตรงบวก 3.3 โวลต์
6	RXD	I	รับข้อมูลแบบอนุกรม
7	TXD	O	ส่งข้อมูลแบบอนุกรม
8	GND	I	กราวนด์

4. บอร์ด Arduino UNO R3

Arduino เป็นไมโครคอนโทรลเลอร์ชนิดหนึ่งที่มีการพัฒนามาแล้ว ทำให้สามารถเขียนคำสั่งโปรแกรมได้เหมือนโปรแกรมภาษาชั้นสูงทั่วไป มีการใช้งานไมโครคอนโทรลเลอร์ในตระกูล AVR สำหรับ รุ่น UNO R3 นี้จะใช้ ATmega328 และใช้เป็นตัวประมวลผลสัญญาณอินฟราเรดที่รับได้จากรีโมทควบคุมโทรทัศน์และใช้ควบคุมการส่งสัญญาณอินฟราเรด

5. Arduino IDE

Arduino IDE (IDE : Integrated Development Environment) เป็นส่วนซอฟต์แวร์ จะรวบรวมชุดไลบรารีสำหรับการทำงานร่วมกับ Arduino hardware ซึ่งใช้งานได้กับแทบทุกระบบปฏิบัติการ และภาษาที่ใช้ในการเขียนโปรแกรมจะเป็นลักษณะของ C++

6. รีโมทควบคุมโทรทัศน์

รีโมทควบคุมโทรทัศน์ยี่ห้อ SAMSUNG รุ่น Slimfit ขนาด 32 นิ้ว มีปุ่มกดควบคุมจำนวน 45 ปุ่ม ผู้จัดทำนำมาใช้ทดลองส่งสัญญาณอินฟราเรดเพื่อหารูปแบบการส่งสัญญาณของรีโมทนำมาสร้างฐานข้อมูลของรีโมทนี้



รูปที่ 3.12 รีโมทควบคุมโทรทัศน์ยี่ห้อ SAMSUNG รุ่น Slimfit

7. ออสซิลโลสโคป (Oscilloscope)

เป็นเครื่องมือวัดทางอิเล็กทรอนิกส์ที่สำคัญที่ใช้วัดเพื่อแสดงรูปคลื่นสัญญาณต่างๆ ออกมาเป็นภาพ เช่น การวัดสัญญาณกระแสไฟฟ้าหรือแรงดันไฟฟ้า (ที่เป็นไฟ AC หรือ DC) การวัดความถี่ของสัญญาณ การวัดเฟสของสัญญาณ และรวมถึงการวัดสัญญาณพัลส์ การอ่านค่าแอมพลิจูดของสัญญาณ สำหรับการทดลองจะใช้วัดเอาต์พุตของตัวรับสัญญาณอินฟราเรด

3.3 การจัดเก็บผลการทดลอง

3.3.1 การสร้างแอปพลิเคชันรีโมทจำลองบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์

1) โปรแกรมสร้างแอปพลิเคชันรีโมทจำลอง

รันโปรแกรมที่ออกแบบไว้ด้วย Emulator เพื่อแสดงลำดับหน้าจอการทำงานของแอปพลิเคชันจากหน้าแรกจนถึงหน้าสุดท้าย

2) การเชื่อมต่อ Wi-Fi ด้วย Emulator

ทดสอบการเชื่อมต่อ Wi-Fi ด้วย Emulator ใน Emulator จะมองเห็นการเชื่อมต่ออินเทอร์เน็ตของเครื่องคอมพิวเตอร์ที่ Emulator ทำงานอยู่เป็นการเชื่อมต่อ (Mobile Internet) ทำการป้อน IP Address และ Port ของเซิร์ฟเวอร์ ซึ่งต้องเปิดเซิร์ฟเวอร์ที่เขียนด้วยโปรแกรม NetBeans IDE รอรับการเชื่อมต่อไว้ก่อน เก็บผลโดยป้อน IP Address และ Port ที่ถูกต้องและไม่ถูกต้อง สังเกตการเชื่อมต่อระหว่างโทรศัพท์เคลื่อนที่กับเซิร์ฟเวอร์

3.3.2 การสื่อสารข้อมูลผ่าน Wi-Fi

1) การส่งข้อมูลจากโทรศัพท์เคลื่อนที่ผ่าน Wi-Fi ไปยังบอร์ด Arduino

เขียนโปรแกรมให้บอร์ด Arduino รอรับค่าที่ Serial Pin แล้วทำการกดปุ่มจากแอปพลิเคชันรีโมทจำลอง เพื่อส่งข้อมูลไปยังบอร์ด Arduino โดย Arduino จะแสดงผลของค่าที่ได้รับมาด้วย Serial Monitor ซึ่งแต่ละปุ่มจะถูกกำหนดให้ส่งค่าที่แตกต่างกันออกไป คือ CH1 ส่งค่าเป็น 1, CH2 ส่งค่าเป็น 2, Vol+ ส่งค่าเป็น a, Vol- ส่งค่าเป็น b

3.3.3 การหาข้อมูลคำสั่งของรีโมท

1) การวัดสัญญาณอินฟราเรดจากรีโมทด้วยออสซิลโลสโคป

กดปุ่มเลข 1 บนรีโมท แล้วเก็บภาพสัญญาณอินฟราเรดที่วัดได้จากออสซิลโลสโคป โดยผลที่เก็บมาต้องมีภาพของสัญญาณทั้งหมด รวมไปถึงวัดความกว้างช่วงเวลาของบิตแต่ละชนิด เพื่อนำไปวิเคราะห์หารูปแบบของสัญญาณ และช่วงเวลาที่ใช้แสดงแทนบิตต่างๆ

2) การวัดผลด้วยการประมวลผลผ่านโปรแกรม

สำหรับการประมวลผลสัญญาณที่ได้รับจากรีโมทใน Arduino IDE จะมีรูปแบบเป็นตัวเลขของคาบเวลาในหน่วยไมโครวินาที ซึ่งจะแทนการส่งสัญญาณอินฟราเรดเป็นบิตสตริม บิตสตริมนี้อาจประกอบไปด้วย บิตเริ่มต้น (Start Bit), บิตข้อมูลที่เป็น '0' หรือ '1' (Bit 0, Bit 1) และบิตหยุด (Stop Bit) โดยจะต้องนำผลไปวิเคราะห์ออกมาเป็นเลขฐานสิบหก

3.3.4 การส่งสัญญาณอินฟราเรด

1) โปรแกรมส่งสัญญาณอินฟราเรด

รันโปรแกรมที่ได้ออกแบบไว้ดังรูปที่ 3.8 เพื่อส่งสัญญาณอินฟราเรดออกไปยังตัวรับสัญญาณอินฟราเรดโดยจะวัดสัญญาณที่รับได้ด้วยออสซิลโลสโคป และทำการวัดรูปสัญญาณที่ระยะทางต่างๆ เพื่อให้ทราบว่าสามารถส่งสัญญาณได้ไกลที่สุดเป็นระยะเท่าไรโดยที่ยังได้รับสัญญาณที่ถูกต้อง การทดสอบโปรแกรมนี้จะป้อนอินพุตจากการกดแป้นพิมพ์เป็นค่าของแต่ละกรณี

2) การทดสอบระบบรวม

การทดสอบนี้เป็นทดสอบการทำงานของระบบทั้งหมด โดยจะให้โปรแกรมส่งสัญญาณอินฟราเรดรอรับข้อมูลอินพุตจากโทรศัพท์เคลื่อนที่ผ่าน Wi-Fi แทนการกดจากแป้นพิมพ์ สังเกตค่าที่รับได้ที่ Serial Monitor ของบอร์ด และวัดสัญญาณที่ตัวรับด้วยออสซิลโลสโคป ทำการเปรียบเทียบการกดปุ่มกับค่าที่บอร์ดได้รับและรูปสัญญาณอินฟราเรดว่าตรงกันหรือไม่ และทำการเก็บผลสถิติความถูกต้องของการกดปุ่มแต่ละปุ่ม จำนวนการกดปุ่ม 10 ครั้ง วัดผลด้วยโทรศัพท์ยี่ห้อ SAMSUNG รุ่น Slimfit ขนาด 32 นิ้ว การต่อวงจรเพื่อทดสอบเป็นดังรูปที่ 3.9

บทที่ 4

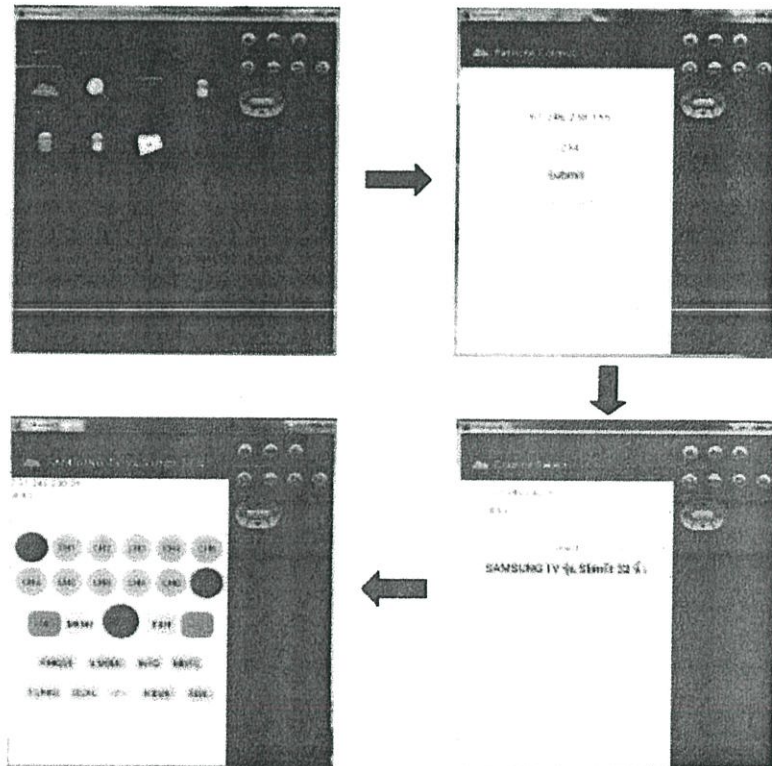
ผลการทดลอง

จากบทที่ 3 การออกแบบและจัดทำโครงการได้แบ่งการทดลองออกเป็น 4 ส่วนด้วยกันคือ การสร้างแอปพลิเคชันรีโมทจำลองบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์, การสื่อสารผ่าน Wi-Fi, การหาข้อมูลคำสั่งของรีโมท และการส่งสัญญาณอินฟราเรด ซึ่งผลของการทดลองแต่ละส่วนจะแสดงได้ดังต่อไปนี้

4.1 การสร้างแอปพลิเคชันรีโมทจำลองบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์

4.1.1 ผลการทดลองโปรแกรมสร้างแอปพลิเคชันรีโมทจำลอง

จากการออกแบบเพื่อสร้างแอปพลิเคชันรีโมทจำลองด้วยโปรแกรม eclipse จะมีการลำดับหน้าจอการทำงานของแอปพลิเคชัน ดังรูปที่ 4.1



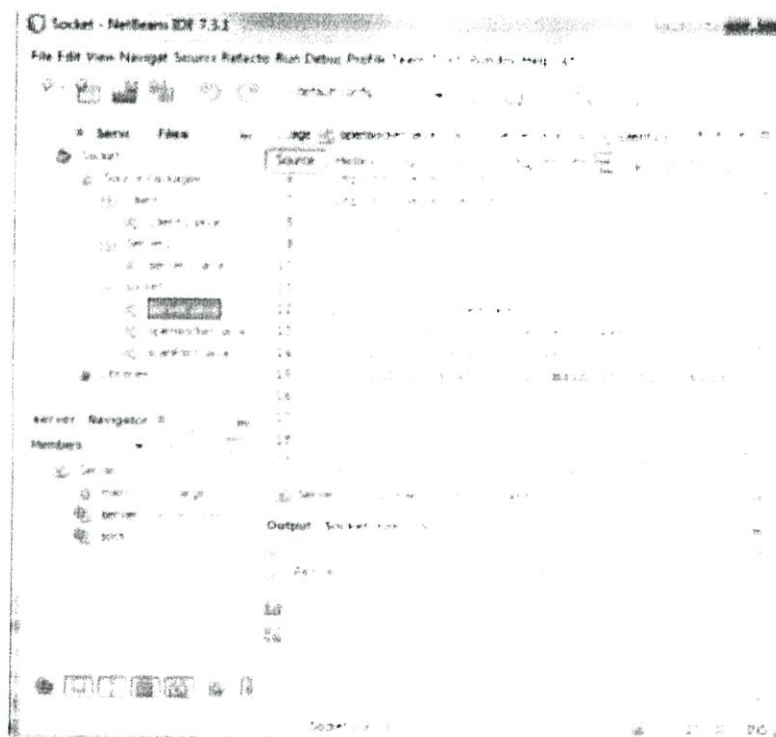
รูปที่ 4.1 ลำดับหน้าจอของแอปพลิเคชัน

การทำงานของแอปพลิเคชันจะเริ่มจากการเปิดแอปพลิเคชันด้วยไอคอนหน้าหลัก จะพบหน้าแรกของแอปพลิเคชันที่มีช่องให้ป้อน IP Address และ Port ของเซิร์ฟเวอร์กดปุ่ม Submit เพื่อเริ่มการเชื่อมต่อ จากนั้นจะมีสถานะของการเชื่อมต่อปรากฏเพื่อบอกให้ผู้ใช้ทราบหน้าที่สองเป็นการเลือกควบคุมอุปกรณ์ที่เราต้องการ และหน้าสุดท้ายเป็นการจำลองปุ่มกดของรีโมทที่จำเป็นต่อการใช้ควบคุมอุปกรณ์ ซึ่งเราได้ออกแบบให้มีปุ่มกดทั้งหมดจำนวน 26 ปุ่มฟังก์ชันการทำงาน ได้แก่ POWER, CH1, CH2, CH3, CH4, CH5, CH6, CH7, CH8, CH9, CH0, TV, VOL+, MENU, ENTER, EXIT, VOL-, P.MODE, S.MODE, INFO, MUTE, TURBO, DUAL, -/--, P.SIZE และ SIZE

4.1.2 ผลการทดลองการเชื่อมต่อ Wi-Fi ด้วย Emulator

- 1) เปิดคอมพิวเตอร์เชื่อมต่อ Wi-Fi
- 2) เปิดพอร์ตรับการเชื่อมต่อจากแอนดรอยด์อิมูเลเตอร์ ด้วยโปรแกรม

NetBeans IDE ดังรูปที่ 4.2



รูปที่ 4.2 โปรแกรม NetBeans IDE เปิดพอร์ตหมายเลข 1234 รอรับการเชื่อมต่อ

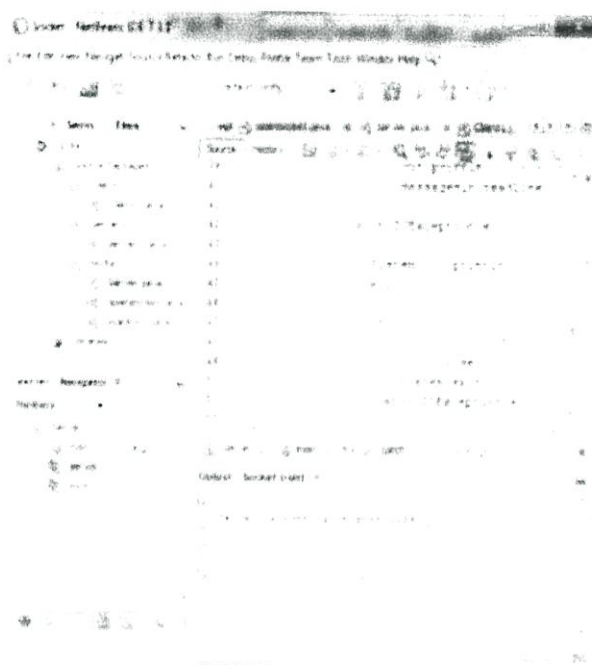
3) เปิดแอปพลิเคชันด้วยแอนดรอยด์อิมูเลเตอร์ ป้อนไอพีแอดเดรสและพอร์ต แล้วกดปุ่ม Submit เพื่อเชื่อมต่อ ถ้าหมายเลขไอพีแอดเดรสและพอร์ตถูกต้อง Connection Status ก็จะเปลี่ยนเป็น Connection Complete ดังรูปที่ 4.3 (ก) แสดงว่าการเชื่อมต่อสำเร็จ ก็จะไม่ปรากฏหน้าต่างแอปพลิเคชันลำดับถัดไป ส่วนของโปรแกรมที่ใช้เปิดพอร์ตก็จะหยุดการทำงาน นั่นคือพอร์ตถูกปิด ดังรูปที่ 4.3 (ข)



161 246 238 155

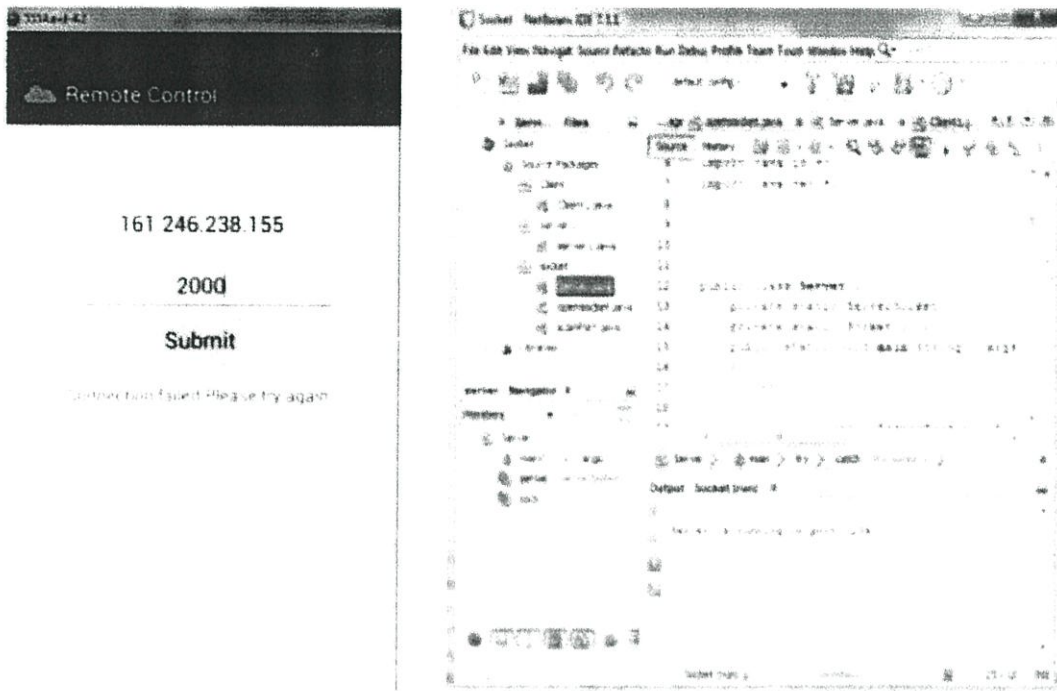
1234

Submit



รูปที่ 4.3 (ก) หน้าจออิมูเลเตอร์แสดงสถานะว่าเชื่อมต่อสำเร็จ
 (ข) หน้าจอโปรแกรมที่ใช้เปิดพอร์ตแสดงสถานะว่าพอร์ตหยุดการทำงาน

4) แต่ถ้าหากหมายเลขไอพีแอดเดรสและพอร์ตที่ป้อนไม่ถูกต้องก็จะขึ้นสถานะ Connection failed. Please try again ดังรูปที่ 4.4 (ก) แสดงว่าการเชื่อมต่อไม่สำเร็จ ก็จะไม่ปรากฏหน้าต่างแอปพลิเคชันต่อไป ส่วนของโปรแกรมที่ใช้เปิดพอร์ตก็ยังคงสถานะพอร์ตเปิดใช้งานอยู่ ดังรูปที่ 4.4 (ข)

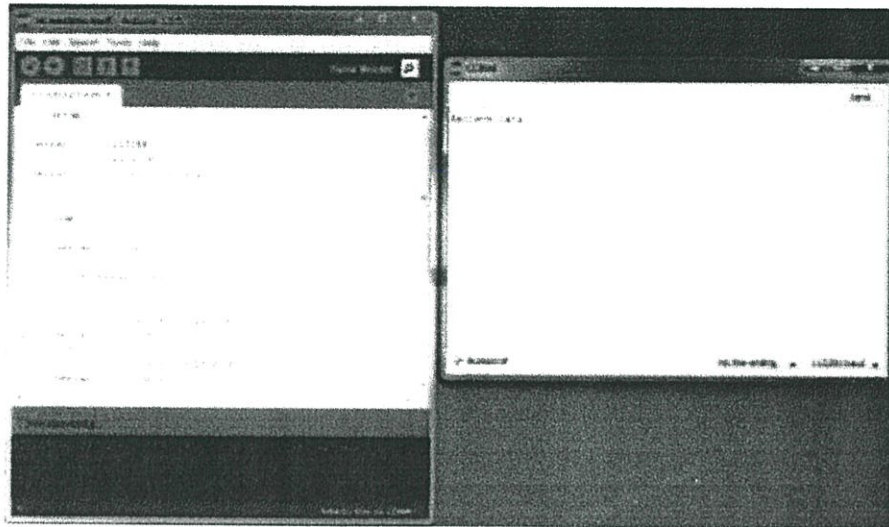


รูปที่ 4.4 (ก) หน้าจออิมูเลเตอร์แสดงสถานการณ์เชื่อมต่อไม่สำเร็จ
 (ข) หน้าจอโปรแกรมที่ใช้เปิดพอร์ตแสดงสถานะว่าพอร์ตยังเปิดอยู่

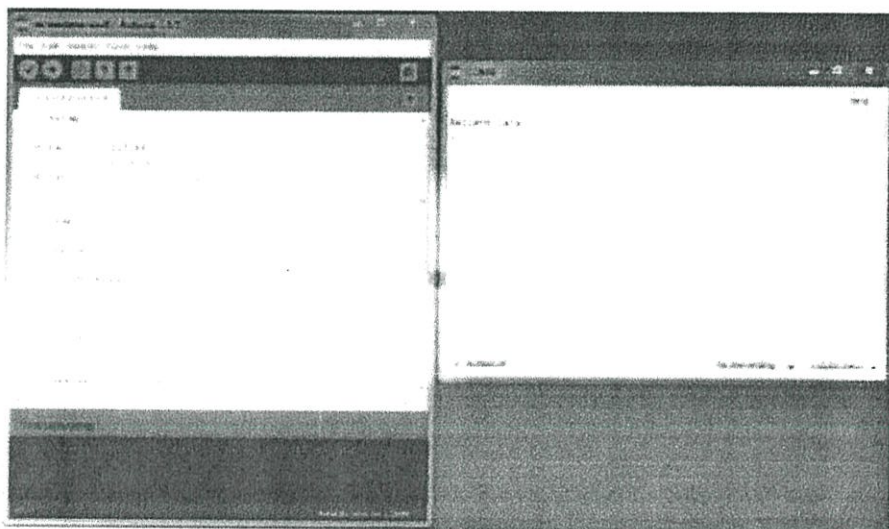
4.2 การสื่อสารข้อมูลผ่าน Wi-Fi

4.2.1 ผลการทดลองส่งข้อมูลจากโทรศัพท์เคลื่อนที่ผ่าน Wi-Fi ไปยังบอร์ด Arduino

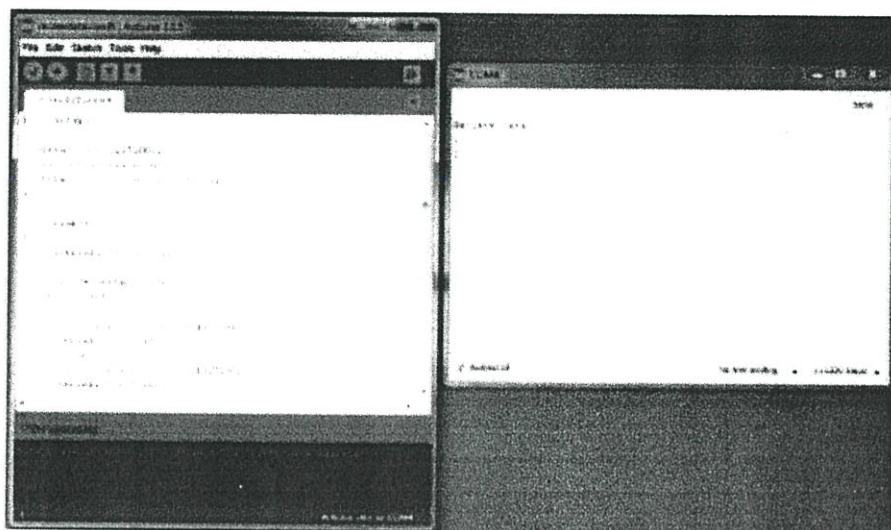
ทำการกดปุ่มจากแอปพลิเคชันรีโมทจำลอง เพื่อส่งข้อมูลไปยังบอร์ด Arduino โดย Arduino จะแสดงผลของค่าที่ได้รับมาด้วย Serial Monitor ซึ่งแต่ละปุ่มจะถูกกำหนดให้ส่งค่าที่แตกต่างกันออกไป คือ CH1 ส่งค่าเป็น 1, CH2 ส่งค่าเป็น 2, Vol+ ส่งค่าเป็น a, Vol- ส่งค่าเป็น b และ POWER ส่งค่าเป็น c ผลการทดลองในขั้นตอนนี้เป็นไปตามรูปที่ 4.5 - 4.10



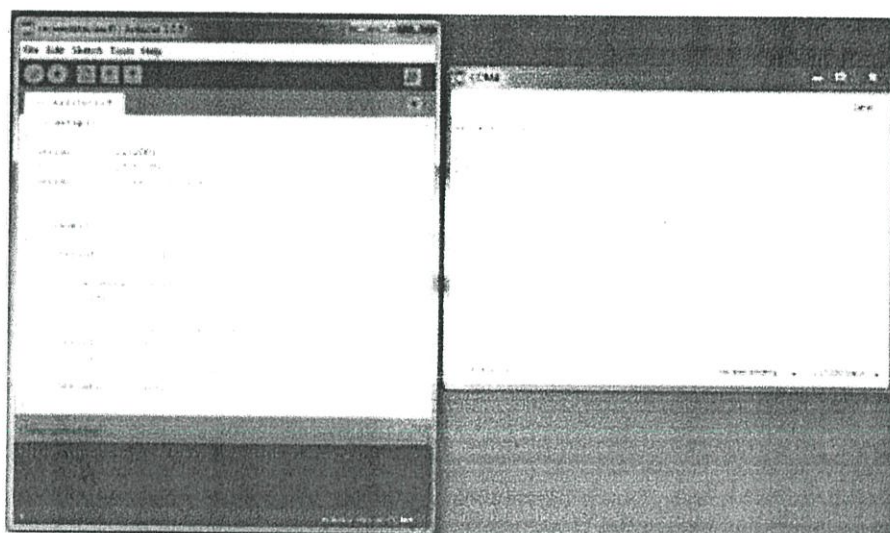
รูปที่ 4.5 Serial Monitor แสดงการเริ่มต้นของโปรแกรมโดยไม่มีการส่งข้อมูลมายังบอร์ด Arduino



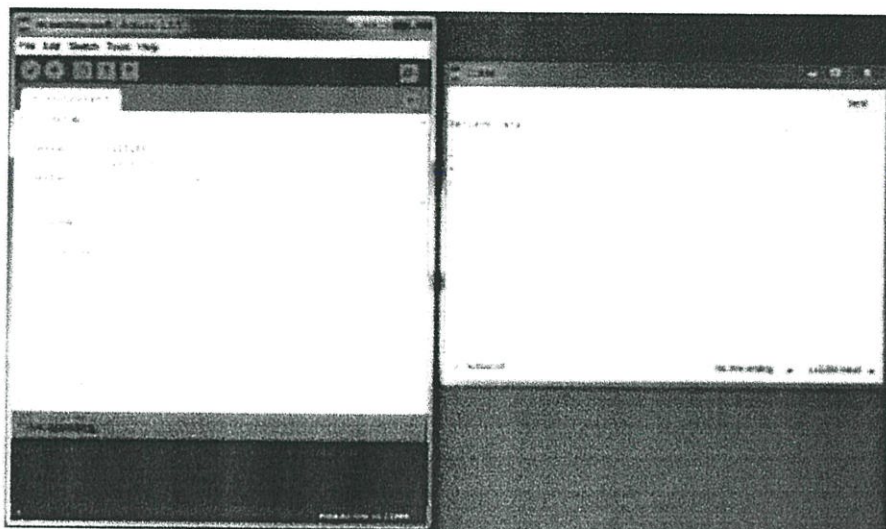
รูปที่ 4.6 Serial Monitor แสดงค่าที่รับได้เป็น 1 เมื่อมีการกดปุ่ม CH1



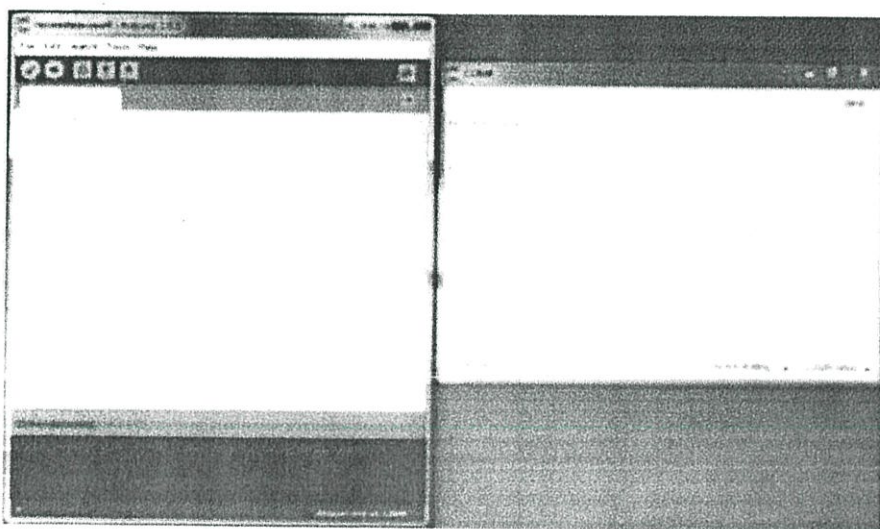
รูปที่ 4.7 Serial Monitor แสดงค่าที่รับได้เป็น 2 เมื่อมีการกดปุ่ม CH2



รูปที่ 4.8 Serial Monitor แสดงค่าที่รับได้เป็น a เมื่อมีการกดปุ่ม Vol+



รูปที่ 4.9 Serial Monitor แสดงค่าที่รับได้เป็น b เมื่อมีการกดปุ่ม Vol-



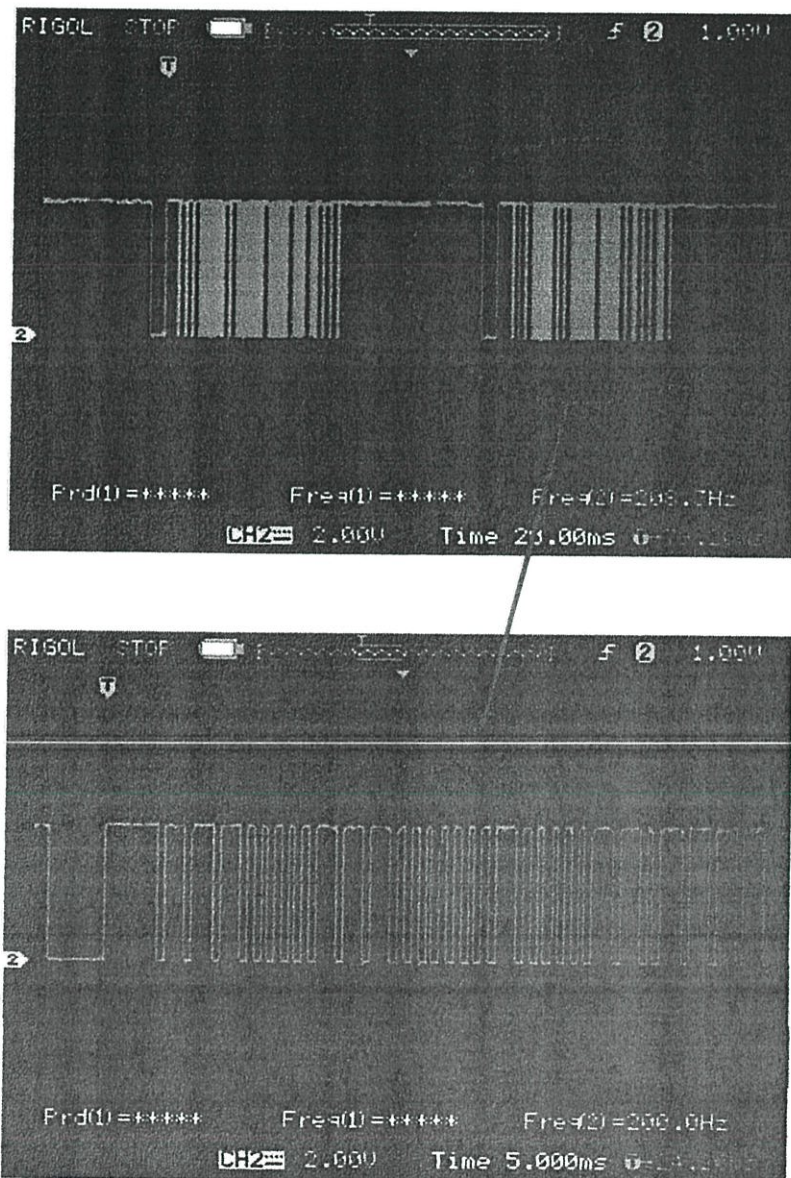
รูปที่ 4.10 Serial Monitor แสดงค่าที่รับได้เป็น c เมื่อมีการกดปุ่ม POWER

จากผลการทดลองรูปที่ 4.5 -4.10 แสดงให้เห็นว่าสามารถส่งข้อมูลด้วยการกดปุ่มจากแอปพลิเคชันรีโมทจำลองไปยังบอร์ด Arduino โดยผ่าน Wi-Fi ได้ ซึ่งบอร์ดก็สามารถรับค่าได้อย่างถูกต้อง

4.3 การหาข้อมูลคำสั่งของรีโมท

4.3.1 ผลการทดลองการวัดสัญญาณอินฟราเรดจากรีโมทด้วยออสซิลโลสโคป

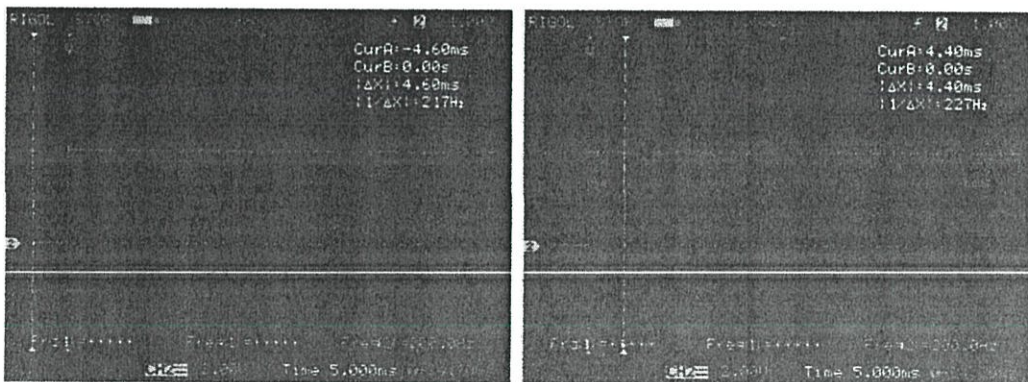
จากการทดลองวัดสัญญาณอินฟราเรดที่ได้จากการกดปุ่มเลข 1 บนรีโมท ด้วยเครื่องออสซิลโลสโคป จะสามารถแสดงสัญญาณออกมาดังรูปที่ 4.11



รูปที่ 4.11 สัญญาณอินฟราเรดของปุ่มเลข 1 บนรีโมท SAMSUNG

จากรูปที่ 4.11 จะเห็นว่าสัญญาณแบ่งออกเป็น 2 ชุด ซึ่งเมื่อทำการสังเกตสัญญาณดังกล่าวจะพบว่า สัญญาณทั้ง 2 ชุดนั้นเป็นสัญญาณเดียวกัน จึงได้ทำการขยายรูปสัญญาณออกให้เหลือเพียงข้อมูลชุดเดียว

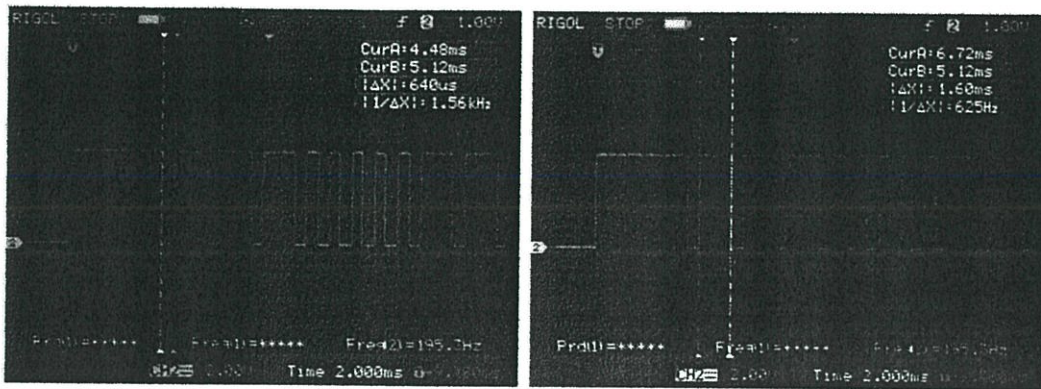
จากบทที่ 2 จะทราบว่าข้อมูลชุดหนึ่งจะประกอบไปด้วยสัญญาณที่ off และ on โดยการ on และ off 1 ครั้งก็นับเป็นข้อมูล 1 ชุด ซึ่งเมื่อนับจำนวนการ on และ off ของสัญญาณในรูปที่ 4.11 ก็จะสามารถนับข้อมูลได้ทั้งหมด 34 ชุด พอเอาไปเปรียบเทียบกับ SAMSUNG Remote protocol ก็พบว่า ข้อมูลชุดแรกที่วัดได้ ก็คือ Start bit ที่วัดความกว้างช่วงเวลา on และ off จากรูปที่ 4.12 ได้ว่า สัญญาณ on เป็นเวลา 4600 us และ off เป็นเวลา 4400 us ทำซ้ำกับข้อมูลชุดอื่นไปเรื่อยๆ ก็พบว่า ข้อมูลอีก 32 ชุดถัดไป เป็นข้อมูลของ bit 1 และ bit 0 สลับกันไปตามคำสั่งแต่ละปุ่มของรีโมท จากทั้งหมดนี้รวมกันมีข้อมูลแล้ว 33 ชุด ทำให้เหลือข้อมูลชุดสุดท้าย ชุดที่ 34 คือ stop bit ที่มี on เป็นเวลา 640 us และ off เป็นเวลา 18 ms ความกว้างของช่วงเวลาต่างๆ นั้น สามารถใช้ oscilloscope วัดได้ดังรูปที่ 4.12 ถึง 4.15



(ก) ช่วงเวลาการ On เป็น 4.60ms

(ข) ช่วงเวลา Off เป็น 4.40ms

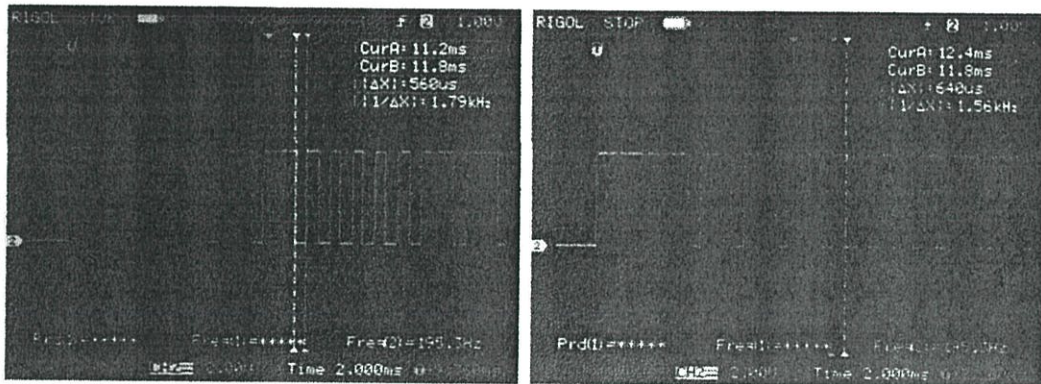
รูปที่ 4.12 สัญญาณ Start Bit



(ก) ช่วงเวลาการ On เป็น 640us

(ข) ช่วงเวลาการ Off เป็น 1.60ms

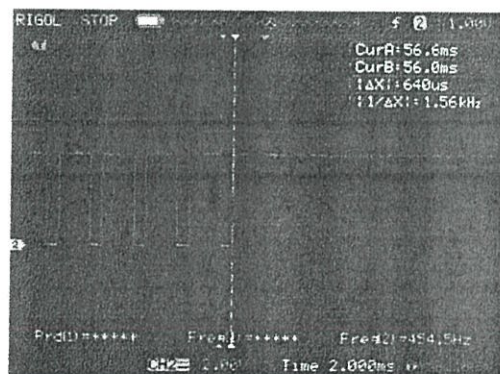
รูปที่ 4.13 สัญญาณบิต 1



(ก) ช่วงเวลาการ On เป็น 560us

(ข) ช่วงเวลาการ Off เป็น 640us

รูปที่ 4.14 สัญญาณบิต 0



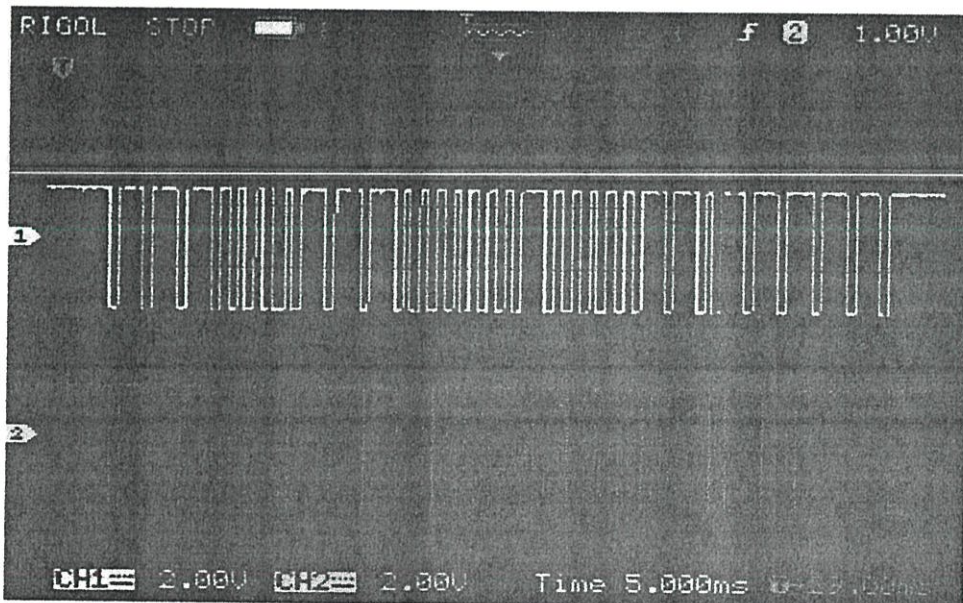
รูปที่ 4.15 สัญญาณ Stop Bit มีความกว้างช่วง On เป็นเวลา 640us

จากข้อมูลข้างต้น สามารถนำมาสรุปรูปแบบของบิตข้อมูลจากความกว้างช่วงเวลาต่างๆ ได้ดังตารางที่ 4.1

ตารางที่ 4.1 ความกว้างช่วงเวลาของบิตต่างๆ

ช่วงเวลา		บิต
On (us)	Off (us)	
4600	4400	Start Bit
640	1600	Bit 1
540	640	Bit 0
640	0	Stop Bit

เมื่อหาความสัมพันธ์ของช่วงเวลาที่เราวัดได้เข้ากับบิตข้อมูลพื้นฐานของคำสั่งจากรีโมทแล้ว ก็จะทำให้การเปรียบเทียบสัญญาณของแต่ละปุ่มกดว่ามีคำสั่งที่ต่างกันจริงหรือไม่ ในที่นี้เราจะทำการเปรียบเทียบคำสั่งที่ได้จากรีโมทเมื่อกดปุ่มเลข 1 กับ เลข 2 ซึ่งแสดงผลข้อมูลของสัญญาณผ่าน oscilloscope ได้ดังรูปที่ 4.16



รูปที่ 4.16 เปรียบเทียบสัญญาณที่รับได้จากการกดปุ่มเลข 1 กับปุ่มเลข 2

จากรูปข้างต้นเป็นการแสดงสัญญาณที่วัดได้เมื่อกดปุ่มเลข 1 และเลข 2 บนรีโมท โดย Channel 1 ของเครื่องวัดแสดงสัญญาณที่ได้จากการกดปุ่มเลข 1 ส่วนสัญญาณที่แสดงใน Channel 2 คือสัญญาณที่วัดได้จากการกดปุ่มเลข 2 ผลที่ได้คือ สัญญาณทั้งสองมีขนาดเท่ากันที่ประมาณ 5 V แต่ในส่วนของข้อมูลสัญญาณ 32 ชุดนั้น จะมีบางส่วนที่ช่วงเวลาการ on/off ต่างกัน ทำให้สรุปได้ว่า ข้อมูลสัญญาณอินฟราเรดที่ส่งจากรีโมทแต่ละปุ่มนั้นต่างกัน

4.3.2 ผลการทดลองวัดผลด้วยการประมวลผลผ่านโปรแกรม

จากโปรแกรมอ่านค่าสัญญาณอินฟราเรดที่รับได้จากรีโมท จะแสดงผลสัญญาณในรูปความกว้างช่วงเวลาต่างๆ กัน ซึ่งเก็บผลมาได้ทั้งหมด 34 ช่วงเวลาดังตารางที่ 4.2

ตารางที่ 4.2 ผลช่วงเวลาของสัญญาณอินฟราเรดของปุ่มกดเลข 1

ค่าที่	On (us)	Off (us)	ค่าที่	On (us)	Off (us)
1	4520	4400	18	600	520
2	600	1640	19	600	500
3	580	1620	20	600	1620
4	580	1640	21	600	500
5	580	500	22	600	500
6	600	500	23	600	500
7	600	500	24	600	500
8	600	500	25	600	500
9	600	520	26	600	1620
10	580	1640	27	580	1640
11	580	1640	28	580	500
12	580	1620	29	600	1620
13	600	480	30	580	1640
14	620	500	31	580	1640
15	600	500	32	580	1620
16	600	500	33	580	1640
17	600	500	34	580	0

เมื่อนำผลสรุปจากตารางที่ 4.1 มาวิเคราะห์ช่วงเวลาที่ได้ในตารางที่ 4.2 จะสามารถหาบิตข้อมูลการส่งสัญญาณของปุ่มกดหมายเลข 1 ได้เป็นเลขฐานสองบรรทัดแรกในรูปที่ 4.16 แต่จากทฤษฎีการส่งข้อมูลของรีโมทในบทที่ 2 ทำให้ทราบว่า ข้อมูลของสัญญาณอินฟราเรดที่ส่งจากรีโมทมานั้นจะเป็นการส่งจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ไปยังบิตที่มีนัยสำคัญสูงสุด (MSB) ทำให้ก่อนจะแปลงเลขฐานสองเป็นฐานสิบหก ต้องมีการสลับตำแหน่งบิตให้เรียงจาก MSB ไปยัง LSB ได้ดังเลขฐานสองบรรทัดที่ 2 ของรูปที่ 4.16 จึงจะสามารถแปลงออกมาเป็นเลขฐานสิบหกได้

	LSB	→	MSB
เลขฐานสอง	1110 0000 1110 0000 0010 0000 1101 1111		
	MSB	→	LSB
เลขฐานสอง	1111 1011 0000 0100 0000 0111 0000 0111		
เลขฐานสิบหก	F B 0 4 0 7 0 7		

รูปที่ 4.17 การวิเคราะห์ผลช่วงเวลาเป็นบิตข้อมูลในรูปแบบเลขฐานสิบหก

จากทั้งหมดข้างต้น จะสามารถสรุปผลบิตข้อมูลที่ใช้ในการส่งสัญญาณอินฟราเรดของปุ่มทั้ง 45 ปุ่มของรีโมทควบคุมโทรทัศน์ยี่ห้อ SAMSUNG รุ่น Slimfit ได้ดังตารางที่ 4.3

ตารางที่ 4.3 บิตข้อมูลการส่งสัญญาณของแต่ละปุ่มกดในรูปแบบเลขฐานสิบหก

ปุ่มกด	บิตข้อมูล	ปุ่มกด	บิตข้อมูล
เลข 1	FB 04 07 07	MENU	E5 1A 07 07
เลข 2	FA 05 07 07	EXIT	D2 2D 07 07
เลข 3	F9 06 07 07	UP	9F 60 07 07
เลข 4	F7 08 07 07	DOWN	9E 61 07 07
เลข 5	F6 09 07 07	LEFT	9A 65 07 07
เลข 6	F5 0A 07 07	RIGHT	1D 62 07 07
เลข 7	F3 0C 07 07	ENTER	97 68 07 07

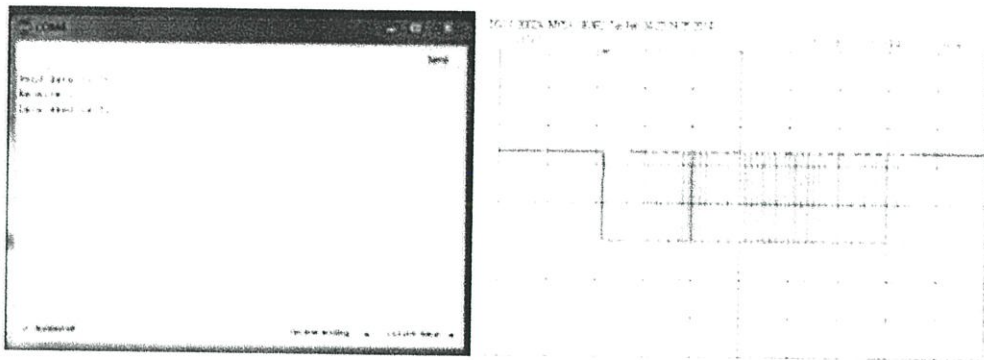
ปุ่มกด	บิตข้อมูล
เลข 8	F2 0D 07 07
เลข 9	F1 0E 07 07
เลข 0	EE 11 07 07
POWER	FD 02 07 07
SOURCE	FE 01 07 07
-/--	DC 23 07 07
PRE-CH	EC 13 07 07
TV	E4 1B 07 07
CH MGR	94 6B 07 07
SLEEP	FC 03 07 07
SRS	91 6E 07 07
MUTE	F0 0F 07 07
Volume +	F8 07 07 07
Volume -	F4 0B 07 07
Program ^	ED 12 07 07
Program v	EF 10 07 07

ปุ่มกด	บิตข้อมูล
PIP	DF 20 07 07
INFO	E0 1F 07 07
TTX/MIX	D3 2C 07 07
P.MODE	93 6C 07 07
S.MODE	EB 14 07 07
STILL	EA 15 07 07
P.SIZE	E9 16 07 07
SOURCE	DB 24 07 07
SWAP	DE 21 07 07
SIZE	D0 2F 07 07
DUAL	FF 00 07 07
TURBO	9B 64 07 07
POSITION	DD 22 07 07
P ^	CD 32 07 07
P v	CC 33 07 07

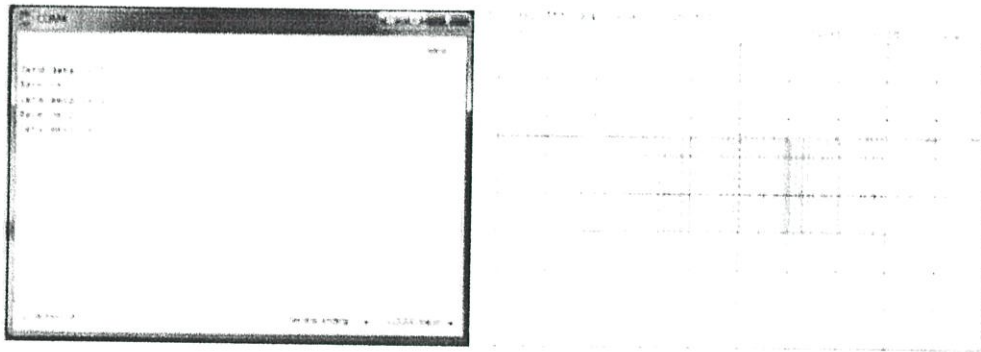
4.4 การส่งสัญญาณอินฟราเรด

4.4.1 ทดสอบโปรแกรมส่งสัญญาณอินฟราเรด

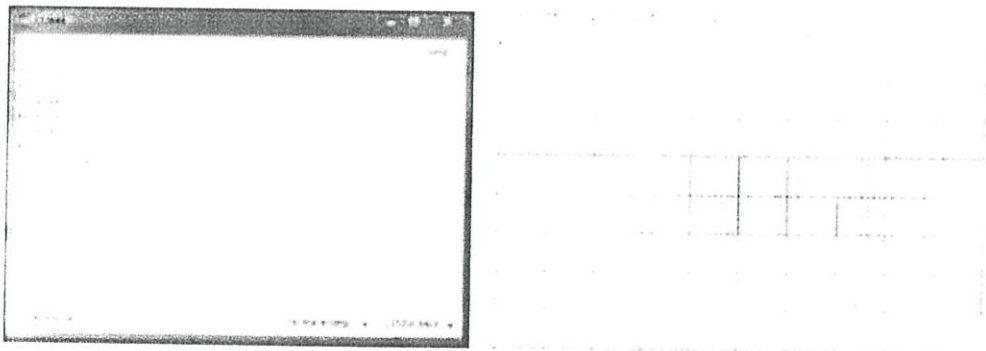
1) กดแป้นพิมพ์เพื่อป้อนอินพุตให้โปรแกรม คือ 1,2,a,b และ c สังเกตค่าที่รับได้ที่ Serial Monitor ของ Arduino IDE และรูปสัญญาณที่วัดได้ที่ตัวรับสัญญาณอินฟราเรด จะได้ผลดังรูปที่ 4.18 – 4.22



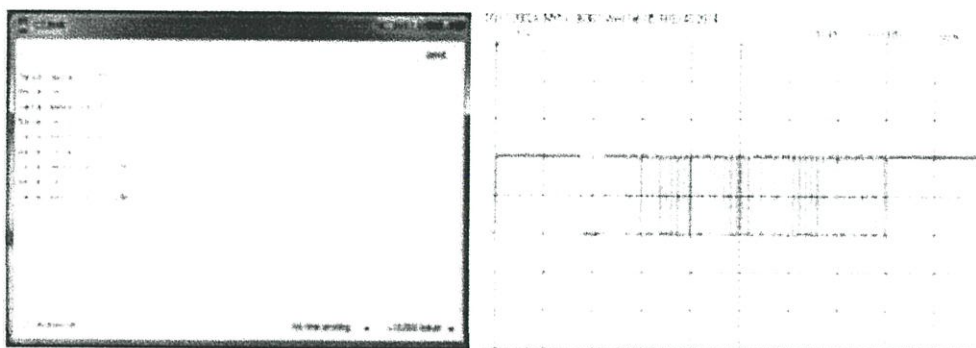
รูปที่ 4.18 (ซ้าย) Serial Monitor แสดงการรับค่า 1 จากการกดแป้นพิมพ์และส่งออกค่า S1
(ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ CH1



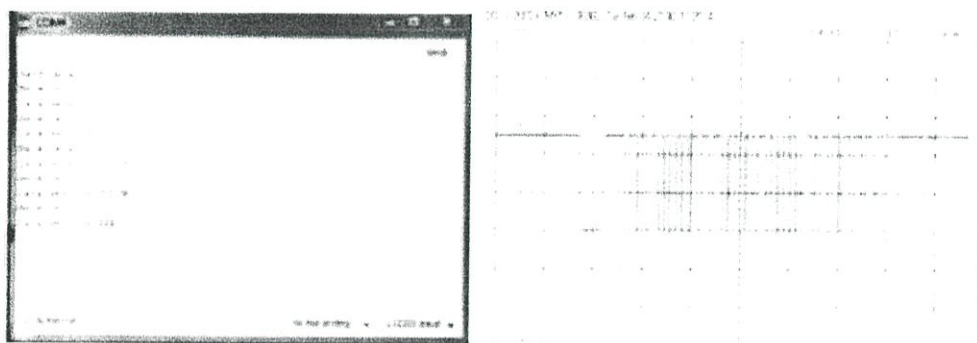
รูปที่ 4.19 (ซ้าย) Serial Monitor แสดงการรับค่า 2 จากการกดแป้นพิมพ์และส่งออกค่า S2
(ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ CH2



รูปที่ 4.20 (ซ้าย) Serial Monitor แสดงการรับค่า a จากการกดแป้นพิมพ์และส่งออกค่า VOLUP
(ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol+

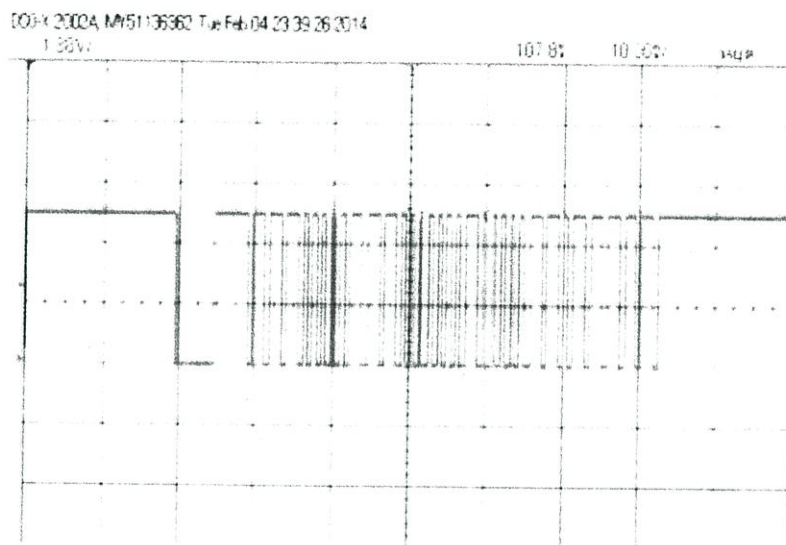


รูปที่ 4.21 (ซ้าย) Serial Monitor แสดงการรับค่า b จากการกดแป้นพิมพ์และส่งออกค่า VOLDW
(ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol-

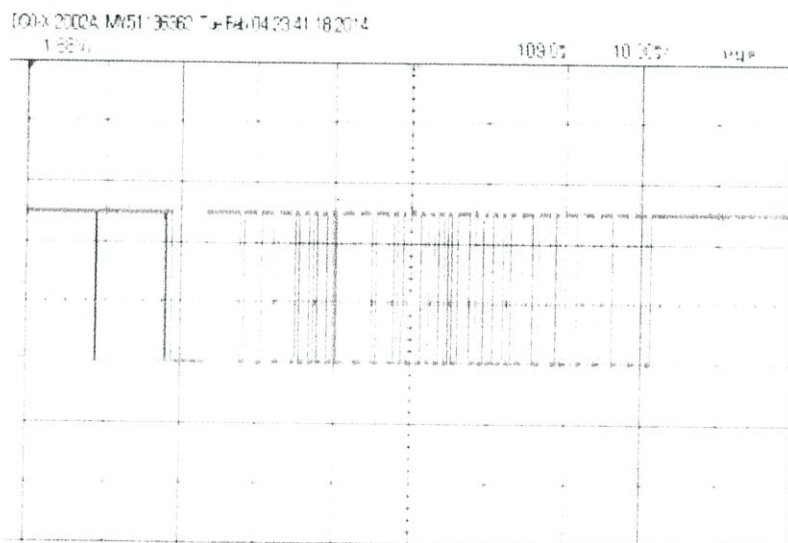


รูปที่ 4.22 (ซ้าย) Serial Monitor แสดงการรับค่า c จากการกดแป้นพิมพ์และส่งออกค่า PWR
(ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ POWER

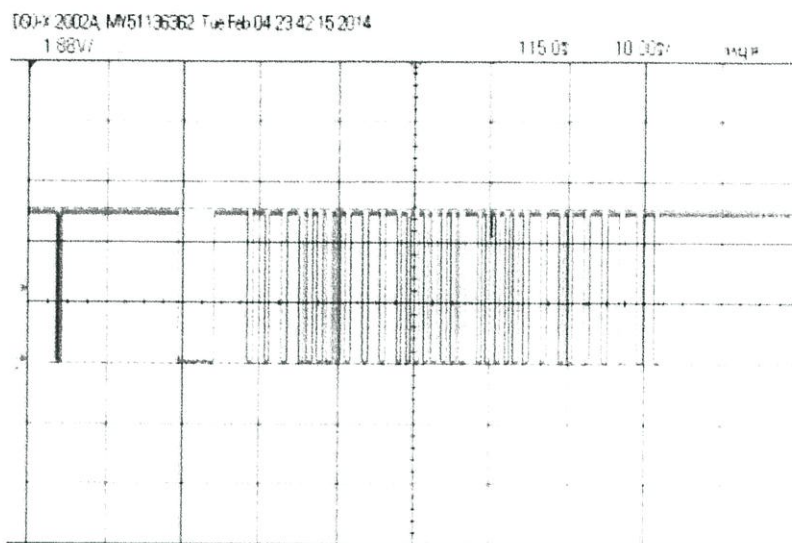
2) รูปสัญญาณอินฟราเรดที่ส่งออกที่ระยะต่างๆ วัดจากตัวรับสัญญาณอินฟราเรด ทดสอบด้วยการกดแป้นพิมพ์เลข 1 จะให้ผลสัญญาณอินฟราเรดของปั๊ม CH1 ได้ผลดังรูปที่ 4.23 - 4.29



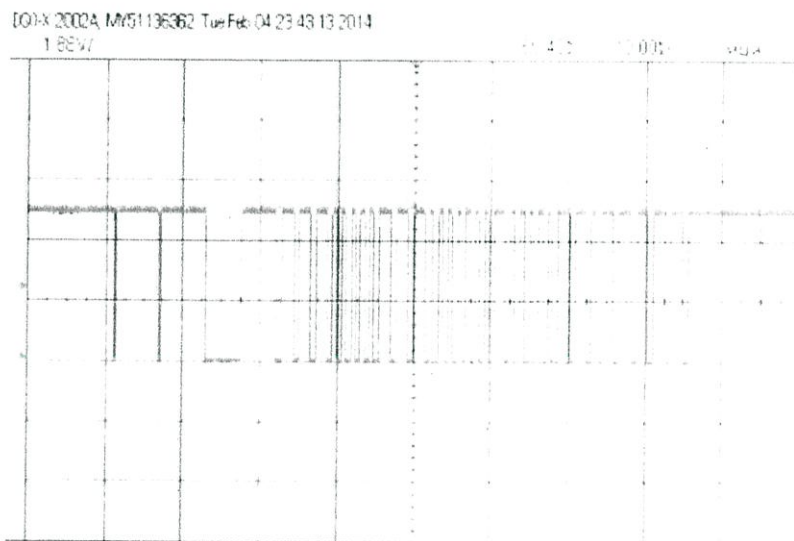
รูปที่ 4.23 สัญญาณอินฟราเรดของปั๊ม CH1 วัดที่ระยะห่าง 50 cm



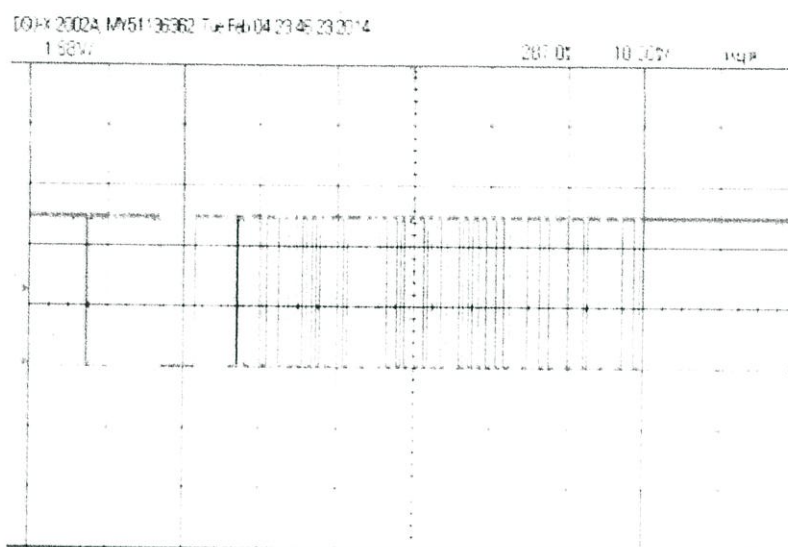
รูปที่ 4.24 สัญญาณอินฟราเรดของปั๊ม CH1 วัดที่ระยะห่าง 100 cm



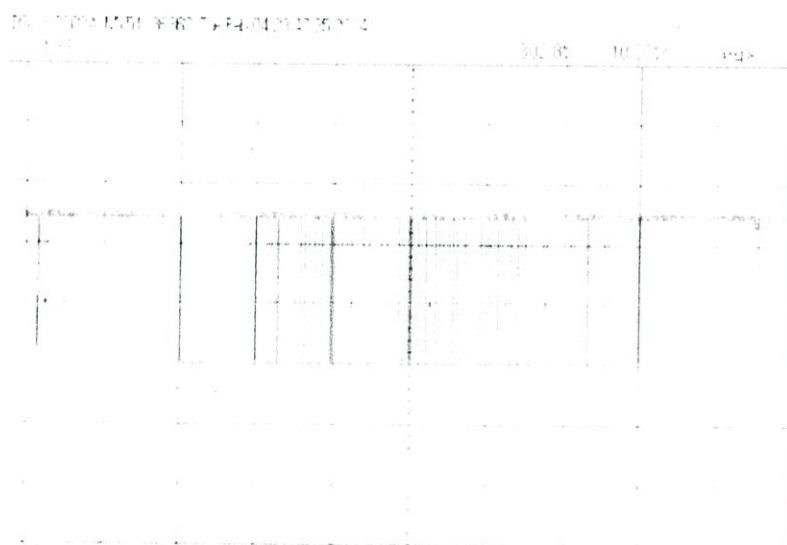
รูปที่ 4.25 สัญญาณอินฟราเรดของปั๊ม CH1 วัดที่ระยะห่าง 150 cm



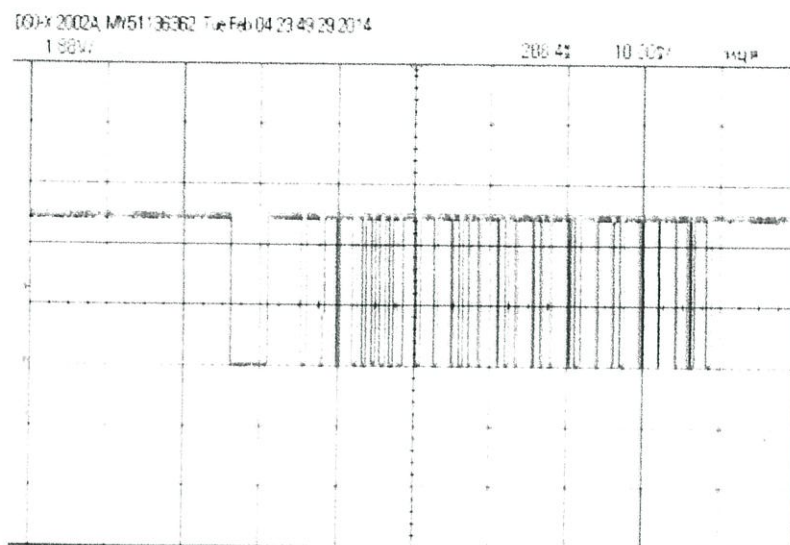
รูปที่ 4.26 สัญญาณอินฟราเรดของปั๊ม CH1 วัดที่ระยะห่าง 200 cm



รูปที่ 4.27 สัญญาณอินฟราเรดของปั๊ม CH1 วัดที่ระยะห่าง 300 cm



รูปที่ 4.28 สัญญาณอินฟราเรดของปั๊ม CH1 วัดที่ระยะห่าง 400 cm

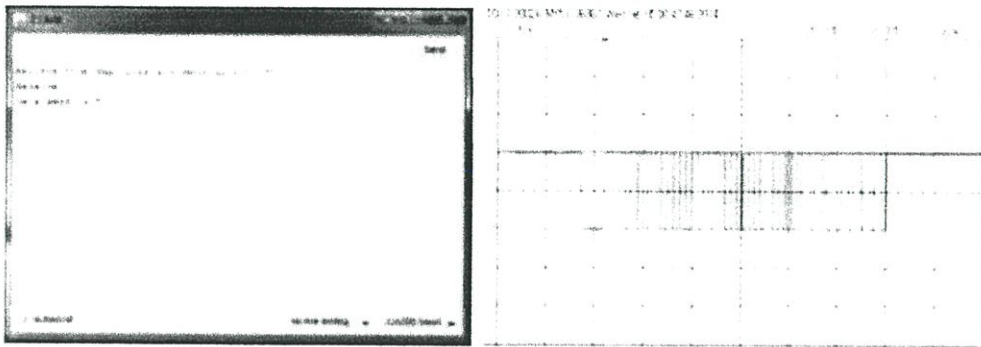


รูปที่ 4.29 สัญญาณอินฟราเรดของปั๊ม CH1 วัดที่ระยะห่าง 500 cm

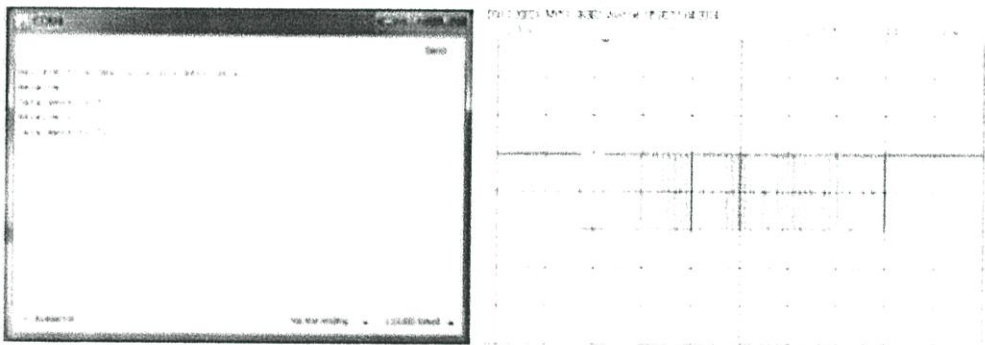
จากผลการทดสอบดังรูปที่ 4.23 - 4.29 พบว่าที่ระยะห่าง 100 cm เริ่มมีสัญญาณรบกวนเข้ามาและเพิ่มขึ้นเรื่อยๆ เมื่อระยะห่างสูงขึ้น ทำให้สัญญาณไม่คมชัด แต่ก็ยังให้ผลของสัญญาณอินฟราเรดที่ถูกต้อง จนมาถึงที่ระยะห่าง 400 cm พบว่าสัญญาณเพี้ยนมากขึ้นจึงทำให้สัญญาณอินฟราเรดที่รับได้มีค่าผิดพลาด ดังนั้นตั้งแต่ระยะห่าง 400 cm เป็นต้นไปตัวรับสัญญาณอินฟราเรดจะตรวจจับสัญญาณได้ผิดพลาด

4.4.2 ทดสอบระบบรวม

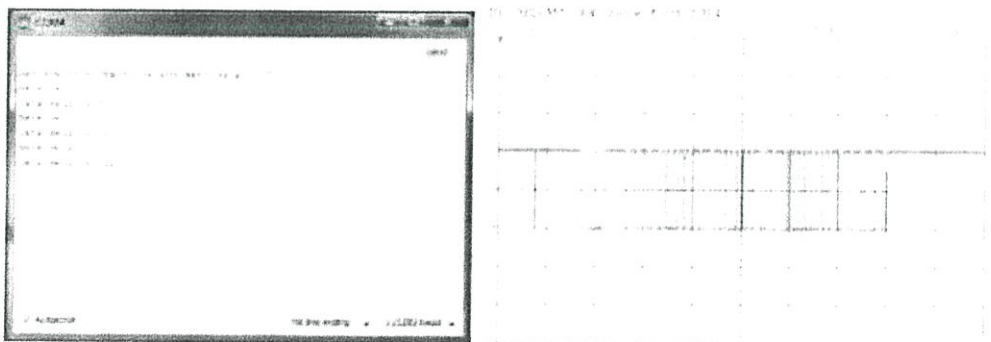
1) กัดปั๊มจากแอปพลิเคชันรีโมทจำลองบนโทรศัพท์เคลื่อนที่เพื่อส่งข้อมูลไปยังบอร์ด Arduino ผ่าน Wi-Fi ส่งค่าที่รับได้ที่ Serial Monitor ของ Arduino IDE และรูปสัญญาณที่วัดได้ที่ตัวรับสัญญาณอินฟราเรดด้วยออสซิลโลสโคป โดยระยะห่างระหว่างตัวรับกับวงจรส่งสัญญาณอินฟราเรด เท่ากับ 100 cm จะได้ผลดังรูปที่ 4.30 - 4.34



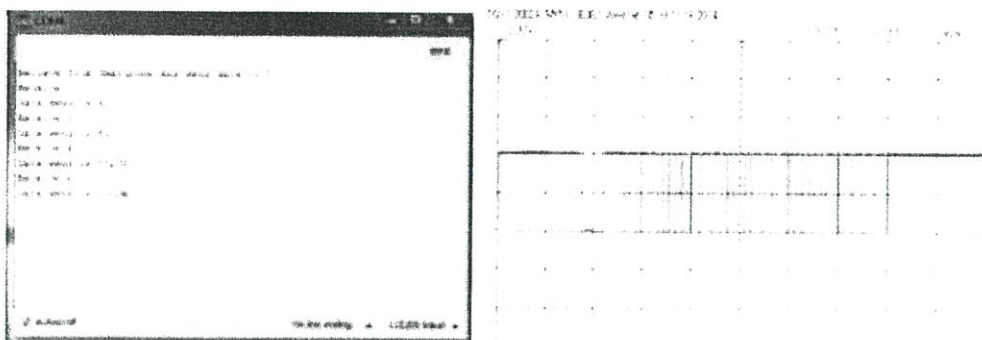
รูปที่ 4.30 (ซ้าย) Serial Monitor แสดงการรับค่า 1 จากการกดปุ่มแอปพลิเคชันและส่งออกค่า S1
(ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ CH1



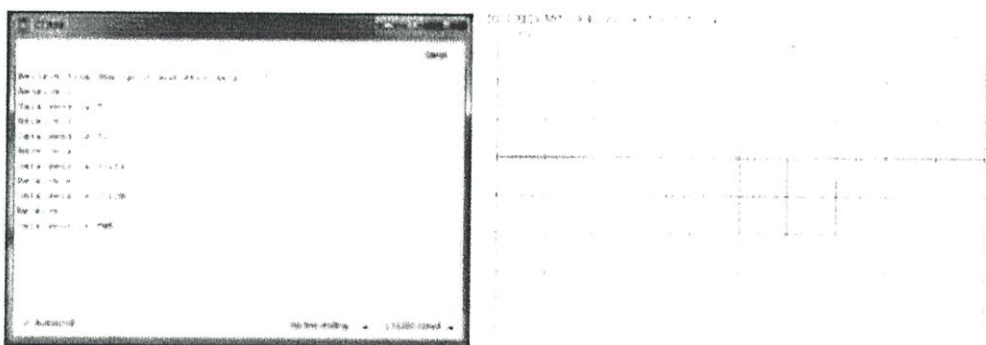
รูปที่ 4.31 (ซ้าย) Serial Monitor แสดงการรับค่า 2 จากการกดปุ่มแอปพลิเคชันและส่งออกค่า S2
(ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ CH2



รูปที่ 4.32 (ซ้าย) Serial Monitor แสดงการรับค่า a จากการกดปุ่มแอปพลิเคชันและส่งออกค่า
VOLUP (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol+



รูปที่ 4.33 (ซ้าย) Serial Monitor แสดงการรับค่า b จากการกดปุ่มแอปพลิเคชันและส่งออกค่า VOLDW (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ Vol-



รูปที่ 4.34 (ซ้าย) Serial Monitor แสดงการรับค่า c จากการกดปุ่มแอปพลิเคชันและส่งออกค่า PWR (ขวา) สัญญาณอินฟราเรดที่วัดได้ตรงกับสัญญาณอินฟราเรดของ POWER

2) วัดผลสถิติความถูกต้องของการกดปุ่มแต่ละปุ่ม จำนวนการกดปุ่ม 10 ครั้ง โดยระยะห่างระหว่างโทรศัพท์กับวงจรส่งสัญญาณอินฟราเรด เท่ากับ 100 cm วัดผลของการเปลี่ยนช่องสัญญาณของโทรศัพท์เมื่อมีการกดปุ่ม สังเกตว่าตรงตามที่กดหรือไม่ จะแสดงผลดังตารางที่ 4.4

ตารางที่ 4.4 สถิติความถูกต้องของการกดปุ่มแอปพลิเคชันรีโมทจำลอง

ปุ่มกด	จำนวนการกดปุ่ม (ครั้ง)	จำนวนครั้งการกดที่ ได้ผลถูกต้อง (ครั้ง)	จำนวนครั้งการกดที่ ได้ผลไม่ถูกต้อง (ครั้ง)	เปอร์เซ็นต์ความ ผิดพลาดของ ผลการกดปุ่ม
CH1	10	10	0	0%
CH2	10	10	0	0%
Vol+	10	10	0	0%
Vol-	10	10	0	0%
POWER	10	10	0	0%

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

5.1.1 แอปพลิเคชันรีโมทจำลองบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์

สามารถสร้างแอปพลิเคชันที่จำลองปุ่มกดของรีโมทควบคุมโทรทัศน์ได้จำนวน 26 ปุ่มฟังก์ชันการทำงาน มีลำดับหน้าจอแอปพลิเคชันดังรูปที่ 4.1 และแอปพลิเคชันสามารถเชื่อมต่อกับเซิร์ฟเวอร์ด้วยหมายเลข IP Address และ Port ผ่านเครือข่าย Wi-Fi ได้

5.1.2 การสื่อสารข้อมูลผ่าน Wi-Fi

บอร์ดประมวลผล Arduino รับข้อมูลจากโทรศัพท์เคลื่อนที่ผ่าน Wi-Fi ได้ผลเป็นข้อมูลแบบตัวอักษร ทำให้สามารถนำไปใช้ประมวลผลต่อไปได้

5.1.3 ข้อมูลคำสั่งของรีโมท

รูปแบบการส่งสัญญาณของรีโมทควบคุมโทรทัศน์ยี่ห้อ Samsung รุ่น Slimfit คือสัญญาณ PWM ที่ส่งบนความถี่ 38 kHz ประกอบไปด้วยบิต 3 รูปแบบคือ

- 1) บิตเริ่มต้น 1 บิต มีช่วงเวลา ON เป็นเวลา 4600 ไมโครวินาที และ OFF เป็นเวลา 4400 ไมโครวินาที
- 2) บิตข้อมูล 32 บิต แบ่งเป็นบิต 1 ซึ่ง ON เป็นเวลา 640 ไมโครวินาที และ OFF เป็นเวลา 1600 ไมโครวินาที กับบิต 0 ซึ่ง ON เป็นเวลา 580 ไมโครวินาที และ OFF เป็นเวลา 640 ไมโครวินาที
- 3) บิตหยุด 1 บิตมีช่วงเวลา ON เป็นเวลา 580 ไมโครวินาที และ OFF เป็นเวลา 18000 ไมโครวินาที

ซึ่งเมื่อนำมาเปรียบเทียบกับข้อมูล Samsung Protocol ในบทที่ 2 พบว่า ค่าช่วงเวลาที่ได้ของแต่ละบิตมีค่าใกล้เคียงกันและจากรูปแบบการส่งสัญญาณของรีโมทที่ได้ ทำให้สามารถวิเคราะห์บิตข้อมูลของทุกปุ่มกดบนรีโมท โดยจะมี 16 บิตแรกที่แตกต่างกัน หรือที่เรียกว่า บิตคำสั่ง แต่จะมี 16 บิตหลังที่เหมือนกัน คือ $(0707)_{16}$ เรียกว่าบิตแอดเดรส

5.1.4 การส่งสัญญาณอินฟราเรด

โปรแกรมส่งสัญญาณอินฟราเรดรอรับค่าจากการกดแป้นพิมพ์และการกดปุ่มแอปพลิเคชันรีโมทจำลองบนโทรศัพท์เคลื่อนที่ผ่าน Wi-Fi สามารถนำค่าที่ได้รับมาเปรียบเทียบกับกรณีที่กำหนดได้อย่างถูกต้อง จึงทำให้วงจรส่งสามารถส่งสัญญาณอินฟราเรดออกไปได้ถูกต้อง ตรงตามการกดปุ่มในแต่ละครั้งได้ระยะทางไกลที่สุดไม่เกิน 4 เมตร

จากการทำงานทั้งหมดของระบบนี้ทำให้ผู้ใช้งานสามารถเลือกสั่งการควบคุมโทรทัศน์ยี่ห้อ Samsung รุ่น Slimfit ผ่านแอปพลิเคชันรีโมทจำลองบนโทรศัพท์เคลื่อนที่ ระบบปฏิบัติการแอนดรอยด์ได้ตามที่ต้องการ โดยไม่มีความผิดพลาดเกิดขึ้น

5.2 ข้อเสนอแนะ

- 1.) พัฒนาระบบให้สามารถควบคุมเครื่องใช้ไฟฟ้าที่สั่งการด้วยรีโมทได้มากกว่าหนึ่งชนิดเช่น เครื่องปรับอากาศ เครื่องเล่นวีซีดี เป็นต้น
- 2.) สร้างชุดคำสั่งข้อมูลของรีโมทของเครื่องใช้ชนิดต่างๆ ให้ผู้ใช้งานไปดาวน์โหลดมาติดตั้งในแอปพลิเคชันได้ด้วยตัวเอง
- 3.) พัฒนารฐานข้อมูลให้ผู้ที่ต้องการพัฒนาโปรแกรมเฉพาะส่วนคำสั่งของรีโมท ให้สามารถอัปโหลดข้อมูลชุดคำสั่งของรีโมทไปเป็นส่วนหนึ่งของฐานข้อมูล

บรรณานุกรม

- [1] เอกชัย มะการ. *เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino*. กรุงเทพฯ : อีทีที, 2552.
- [2] นภัทร สระเอี่ยม, และธเนศ พัฒนาดาพงษ์. “การพัฒนาระบบควบคุมจากระยะไกลโดยใช้แอนดรอยด์แพลตฟอร์ม.” รายงานโครงการวิจัยโดยเงินรายได้, คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2554.
- [3] เขาวลิต ชันคำ. *Java Network Programming การเขียนโปรแกรมโครงข่ายด้วยภาษาจาวา*. กรุงเทพฯ : ม.ป.ท., 2554.
- [4] พร้อมเลิศ หล่อวิจิตร. *คู่มือเขียนแอป Android ฉบับสมบูรณ์*. กรุงเทพฯ : โปริวิชั่น, 2556.
- [5] Ken Shirriff. “A Multi-Protocol Infrared Remote Library for the Arduino.”
<http://www.righto.com/2009/08/multi-protocol-infrared-library.html?m=1>
- [6] “SAMSUNG Protocol.” <http://rusticengineering.com/2011/02/09/infrared-room-control-with-samsung-ir-protocol/>.
- [7] “SAMSUNG Protocol.” <http://www.techdesign.be/projects/011/011.htm>.
- [8] “Arduino.” <https://sites.google.com/site/123microcontroller/Basic-Microcontroller-Programming/Comparison-Adriano-Platform-with-other-MCU>.
- [9] “Arduino UNO.” <http://arduino.cc/en/Main/arduinoBoardUno>.
- [10] “Bug in Raw IR Decoder for Arduino Code?” <http://forums.adafruit.com/viewtopic.php?f=19&t=28448&hilit=necremote&sid=c6fb7d0b382ce-43164e8b609280b3cfe&start=15>.
- [11] “การส่งข้อมูลผ่าน WLAN ด้วย TCP Socket.” <http://www.axexorcist.com/2012/09/android-code-wlan-tcp-socket.html>