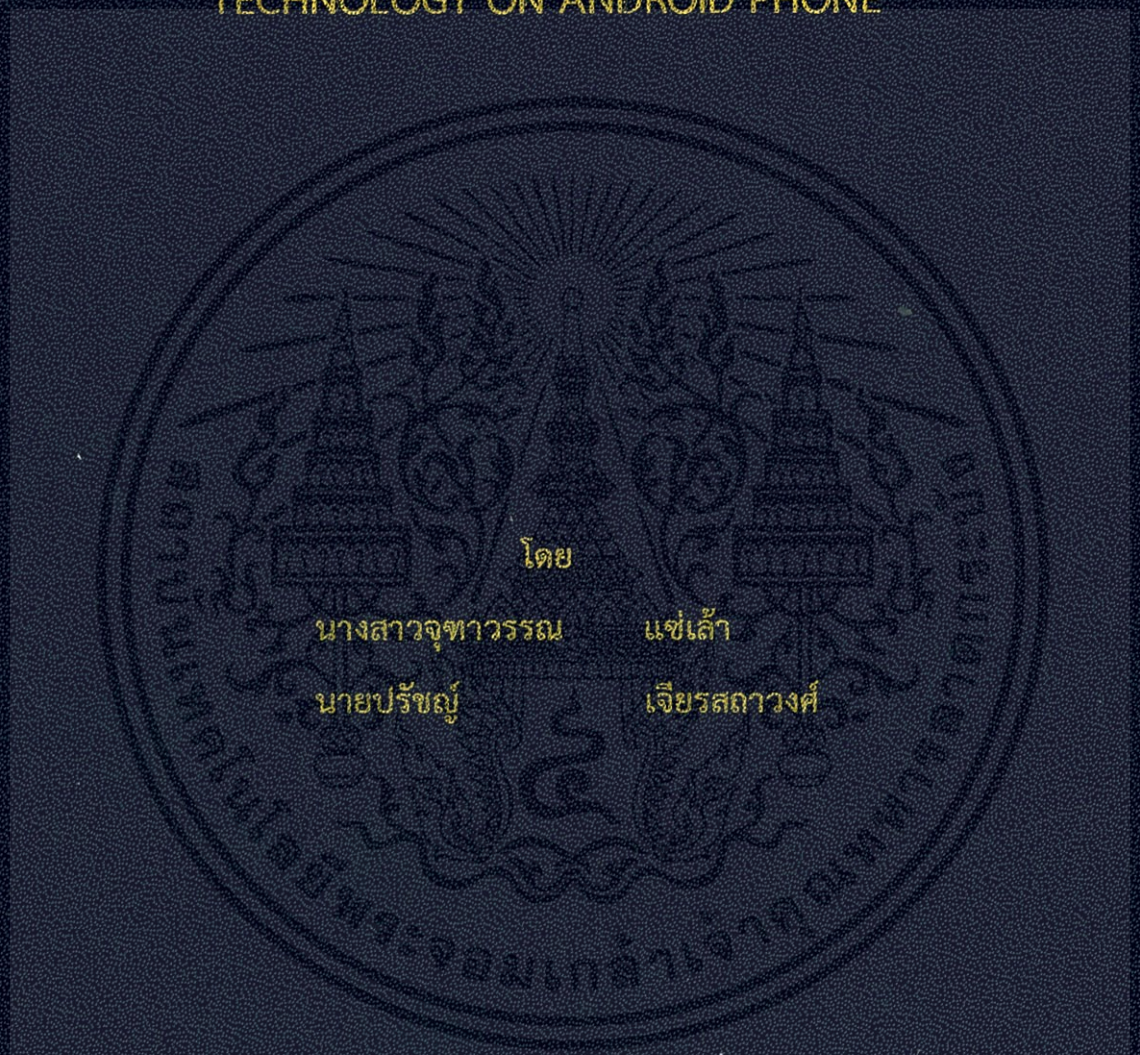
ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ด้วยเทคโนโลยี NFC และบลูทูธ

ELECTRIC EQUIPMENT CONTROL SYSTEM USING NFC AND BLUETOOTH TECHNOLOGY ON ANDROID PHONE



โดย

นางสาวจุฑาวรรณ แซ่เล้า

นายปรัชญ์ เจียรสถาวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ด้วยเทคโนโลยี NFC และบลูทูธ

ELECTRIC EQUIPMENT CONTROL SYSTEM USING NFC AND BLUETOOTH TECHNOLOGY ON ANDROID PHONE

โดย

นางสาวจุฑาวรรณ แซ่เล้า

นายปรัชญ์ เจียรสถาวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ด้วยเทคโนโลยี NFC และบลูทูธ

ELECTRIC EQUIPMENT CONTROL SYSTEM USING NFC AND BLUETOOTH TECHNOLOGY ON ANDROID PHONE

โดย

นางสาวจุฑาวรรณ	แซ่เล้า	53010245
นายปรัชญ์	เจียรสถาวงศ์	53010933

อาจารย์ที่ปรึกษา

ผศ. ดร. ธเนศ พัฒนธาดาพงษ์

ผศ. ดร. นภัทร สระเอี่ยม

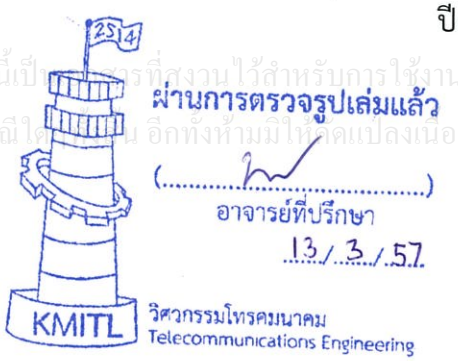
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ด้วยเทคโนโลยี NFC และบลูทูธ

ELECTRIC EQUIPMENT CONTROL SYSTEM USING NFC AND BLUETOOTH TECHNOLOGY ON ANDROID PHONE

ผู้จัดทำ

1. นางสาวจุฑาวรรณ แซ่เล้า 53010245
2. นายปรัชญ์ เจียรสถาวงศ์ 53010933

..... อาจารย์ที่ปรึกษา
(ผศ. ดร. ธเนศ พัฒนธาดาพงษ์)

..... อาจารย์ที่ปรึกษา
(ผศ. ดร. นภัทร สระเอี่ยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงด้วยดี เนื่องมาจากคณะผู้จัดทำได้รับความอนุเคราะห์อย่างสูง จากอาจารย์ที่ปรึกษา ผศ. ดร. ธเนศ พัฒนธาดาพงษ์ และอาจารย์ที่ปรึกษาร่วม ผศ. ดร. นภัทร สระเอี่ยม ที่ได้ให้ความช่วยเหลือและชี้แนะแนวทางในการแก้ปัญหา ตลอดจนให้ความรู้ความเข้าใจในด้านต่างๆ ขอขอบพระคุณบิดามารดาที่เลี้ยงดูและให้โอกาสในการศึกษา ขอขอบพระคุณคณาจารย์สาขาวิชาวิศวกรรมโทรคมนาคมทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ทางด้านโทรคมนาคมให้ทางคณะผู้จัดทำได้มีแนวทางในการทำปริญญานิพนธ์นี้ คณะผู้จัดทำจึงหวังเป็นอย่างยิ่งว่าปริญญานิพนธ์นี้จะสามารถนำไปเป็นแนวทางในการประยุกต์ใช้เพื่อให้เกิดประโยชน์ต่อๆไปได้

นางสาวจุฑาวรรณ แซ่เล้า
นายปรัชญ์ เจียรสถาวงศ์
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือ
ระบบปฏิบัติการแอนดรอยด์ด้วยเทคโนโลยี NFC และบลูทูธ
ELECTRIC EQUIPMENT CONTROL SYSTEM USING
NFC AND BLUETOOTH TECHNOLOGY ON ANDROID
PHONE

โดย นางสาวจุฑาวรรณ แซ่เล้า 53010245
นายปรัชญ์ เจียรสถาวงศ์ 53010933

อาจารย์ที่ปรึกษา ผศ. ดร. ธเนศ พัฒนธาดาพงษ์
ผศ. ดร. นภัทร สระเอี่ยม

บทคัดย่อ

ปัจจุบันสถานประกอบการเชิงที่พักอาศัย เช่น โรงแรม ได้มีการนำเทคโนโลยีต่างๆมาประยุกต์ใช้ เพื่อให้ความสะดวกสบายแก่ผู้ใช้บริการอย่างสูงสุด ซึ่งเป็นที่แน่นอนว่าผู้ใช้บริการจำเป็นต้องใช้งานเครื่องใช้ไฟฟ้าต่างๆภายในห้องพัก ดังนั้นหากผู้ใช้บริการสามารถควบคุมเครื่องใช้ไฟฟ้าภายในห้องพักได้จากระยะไกลโดยไม่ต้องไปสัมผัสสวิตช์ คงเพิ่มความสะดวกระหว่างการพักอาศัยได้ไม่น้อย

ปริญญานิพนธ์นี้จึงได้นำเสนอการประยุกต์ใช้แอพพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์, เทคโนโลยีบลูทูธและเทคโนโลยี NFC (NEAR FIELD COMMUNICATION) ในการควบคุมเครื่องใช้ไฟฟ้า โดยสั่งงานผ่านไมโครคอนโทรลเลอร์เพื่อควบคุมการเปิดและปิดของเครื่องใช้ไฟฟ้าผ่านรีเลย์ โดยระบบต้นแบบนี้อาจมีการนำไปปรับปรุงและประยุกต์ให้มีความหลากหลายในการใช้งานมากยิ่งขึ้นในอนาคต

ABSTRACT

Currently, the establishments like hotels applied technologies for provide convenient services to guest. It is exactly that guest will use electric equipment in their room. So if guest can remotely control electric equipment without touching switch, it will helpful during resident.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The objective of this project is to utilize the NFC (Near Field Communication) technology, Bluetooth technology and develop an electric equipment control system command by microcontroller and relay. This project can be improved for more features and applications in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		VII
สารบัญตาราง		IX
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 บลูทูธ	2
	2.1.1 การทำงานของบลูทูธ	2
	2.1.2 ประโยชน์ของบลูทูธ	3
	2.1.3 เป้าหมายของ BLUETOOTH	3
	2.2 ไมโครคอนโทรลเลอร์	4
	2.2.1 คุณสมบัติของอาร์ดูโน	4
	2.3 รีเลย์	6
	2.3.1 โครงสร้างและสัญลักษณ์ของรีเลย์	6
	2.3.2 หลักการทำงานของรีเลย์	7
	2.4 แอนดรอยด์ (ANDROID)	8
	2.5 จาวา (JAVA)	8
	2.6 เอ็นเอฟซี (NFC : NEAR FIELD COMMUNICATION)	8
	2.6.1 ประโยชน์ของ NFC และการนำไปใช้	13
	2.7 พัลส์วิดท์มอดูเลชั่น (PULSE WIDTH MODULATION)	13
	2.8 ซีอาร์ซี (CRC : CYCLIC REDUNDANCY CHECK)	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**14**
	3.1 การออกแบบ	14
	3.1.1 การเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์	15
	3.1.2 การส่งคำสั่งไปยังไมโครคอนโทรลเลอร์	17
	3.1.3 การรับคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้าของไมโครคอนโทรลเลอร์	17
	3.1.4 การส่งค่าสถานะของเครื่องใช้ไฟฟ้ากลับไปยังแอนดรอยด์	19
	3.1.5 การรับค่าสถานะจากไมโครคอนโทรลเลอร์	20
	3.2 เครื่องมือที่ใช้ในการทดลอง	22
	3.2.1 โทรศัพท์มือถือ	22
	3.2.2 ไมโครคอนโทรลเลอร์	22
	3.2.3 รีเลย์	23
	3.2.4 บลูทูธโมดูล	23
	3.2.5 โปรแกรม ECLIPSE	24
	3.2.6 NFC TAG	24
	3.2.7 เซ็นเซอร์วัดอุณหภูมิ LM35	25
	3.3 การจัดเก็บผลการทดลอง	25
	3.3.1 การเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์	25
	3.3.2 การส่งคำสั่งไปยังไมโครคอนโทรลเลอร์	26
	3.3.3 การรับและส่งคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้าของไมโครคอนโทรลเลอร์	27
	3.3.4 การส่งค่าสถานะของเครื่องใช้ไฟฟ้ากลับไปยังแอนดรอยด์	27
	3.3.5 การรับค่าสถานะจากไมโครคอนโทรลเลอร์	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 4	**ผลการทดลอง**	**29**
	4.1 ผลการทดสอบการเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์	29
	4.1.1 ผลการทดสอบการอ่านข้อมูลจาก NFC TAG	29
	4.1.2 ผลการทดสอบการเชื่อมต่อบลูทูธของแอนดรอยด์	32
	4.2 ผลการทดสอบการส่งคำสั่งไปยังไมโครคอนโทรลเลอร์	37
	4.3 ผลการทดสอบการรับและส่งคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้า	38
	4.3.1 ผลการทดสอบการทำงานของบอร์ดอาร์ดูโน่	38
	4.3.2 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ ATMEGA 168	39
	4.3.3 ผลการทดสอบการทำงานของวงจรรีเลย์	39
	4.3.4 ผลการทดสอบการสั่งการเครื่องใช้ไฟฟ้า	41
	4.4 การรับและส่งค่าสถานะของเครื่องใช้ไฟฟ้าจากไมโครคอนโทรลเลอร์ไปยังแอนดรอยด์	45
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**56**
	5.1 สรุปผล	56
	5.2 ข้อเสนอแนะ	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บรรณานุกรม		58
ภาคผนวก ก	โค้ดของโทรศัพท์แอนดรอยด์	61
ภาคผนวก ข	โค้ดของไมโครคอนโทรลเลอร์	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 ลักษณะของอาร์ดูโน ยูโน 5
2.2 สัญลักษณ์ของรีเลย์ 6
2.3 ตำแหน่งขาของรีเลย์แบบ 5 ขา 7
2.4 ขณะที่ไม่จ่ายกระแสไฟฟ้าแก่ขดลวดรีเลย์ 7
2.5 ขณะจ่ายกระแสไฟฟ้าแก่ขดลวดรีเลย์ 7
2.6 การใช้ประโยชน์ของเอ็นเอฟซี 9
2.7 การใช้งานบัตรวีซ่า เพย์เวฟ ที่ไม่ต้องรูด แค่สัมผัส 11
2.8 การใช้มือถือทำงานเป็นอาร์เอฟไอดีการ์ด 11
2.9 การสื่อสารในลักษณะเพียร์ทูเพียร์ 12
2.10 สัญญาณ PWM ที่มีคาบต่างกัน 13
3.1 บล็อกไดอะแกรมแสดงระบบของโครงงาน 14
3.2 แผนผังการทำงานของระบบ 15
3.3 โฟลว์ชาร์ตการเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านบลูทูธโมดูล 16
3.4 ส่วนติดต่อผู้ใช้งาน 17
3.5 การต่ออุปกรณ์เพื่อควบคุมการปรับระดับไฟของเครื่องใช้ไฟฟ้าเครื่องที่ 1 18
3.6 โฟลวชาร์ตการสั่งงานเครื่องใช้ไฟฟ้า 19
3.7 การทำงานของวงจร ring detector 20
3.8 โฟลว์ชาร์ตการรับค่าสถานะจากไมโครคอนโทรลเลอร์ 21
3.9 โทรศัพท์มือถือยี่ห้อซัมซุงแกแลคซี่ เอสสาม 22
3.10 ไมโครคอนโทรลเลอร์ : ATmega 168 22
3.11 รีเลย์ 5 ขา 23
3.12 บลูทูธโมดูล 23
3.13 สัญลักษณ์ของโปรแกรม Eclipse 24
3.14 บัตรไมแฟร์ คลาสสิก ความจุ 1 k 24
3.15 เซ็นเซอร์วัดอุณหภูมิ LM35 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.16	รายชื่ออุปกรณ์ที่จะทำการจับคู่	26
3.17	การใส่รหัส PIN เพื่อจับคู่อุปกรณ์	27
4.1	ข้อมูลทั้งหมดที่อ่านได้จาก NFC tag	29
4.2	ข้อมูลที่ได้จากการตัดข้อมูลครั้งแรก	30
4.3	ข้อมูลที่ได้จากการตัดข้อมูลครั้งที่สอง	30
4.4	ส่วนติดต่อผู้ใช้งานก่อนการแตะ NFC tag	30
4.5	ข้อมูลที่ได้จาก NFC tag	31
4.6	การร้องขอให้มีการเปิดใช้งานบลูทูธของแอนดรอยด์	31
4.7	ส่วนติดต่อผู้ใช้งานของแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้า	32
4.8	หน้าต่างแจ้งเตือนให้เปิดบลูทูธ	33
4.9	รายชื่อของอุปกรณ์ที่จับคู่แล้ว	34
4.10	ข้อความ Connect ! แสดงเมื่อมีการเชื่อมต่อ Socket	35
4.11	ข้อความ Can create stream แสดงเมื่อมีการสร้าง Stream	36
4.12	ข้อความ Device connected แสดงเมื่อมีการสร้าง Stream	36
4.13	ข้อความ Device cannot connect แสดงเมื่อมีการไม่มีสร้าง Stream	37
4.14	คำสั่งที่ไมโครคอนโทรลเลอร์รับได้เมื่อกดปุ่ม ON ของหลอดไฟดวงที่ 2	37
4.15	คำสั่งที่ไมโครคอนโทรลเลอร์รับได้เมื่อกดปุ่ม ON ของหลอดไฟดวงที่ 3	38
4.16	บอร์ดอาร์ดูโน่	38
4.17	ผลการทดสอบโปรแกรม blink	39
4.18	ผลการทดสอบการสั่งการรีเลย์พอร์ต 3	40
4.19	ผลการทดสอบการสั่งการรีเลย์พอร์ต 4	40
4.20	ผลการทดสอบการสั่งการรีเลย์พอร์ต 3 และ 4	41
4.21	เครื่องใช้ไฟฟ้าทั้ง 5 อุปกรณ์	41
4.22	หลอดไฟ Incandescent ติด (แสงไฟอ่อน)	42
4.23	หลอดไฟ Incandescent ติด (แสงไฟสว่าง)	42
4.24	หลอดไฟ Fluorescent ติด	43
4.25	โคมไฟติด	43
4.26	พัดลมดูดอากาศทำงาน	44
4.27	พัดลมทำงาน	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.28	สถานะของเครื่องใช้ไฟฟ้า	45
4.29	แผนภูมิแสดงความสัมพันธ์ระหว่างอุณหภูมิและแรงดันจากขา Vo ของ LM35	46
4.30	สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 1	47
4.31	ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 1	47
4.32	สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 2	48
4.33	ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 2	48
4.34	สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 3	49
4.35	ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 3	49
4.36	สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 4	50
4.37	ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 4	50
4.38	สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 5	51
4.39	ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 5	51
4.40	สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 6	52
4.41	ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 6	52
4.42	สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 7	53
4.43	ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 7	53
4.38	สถานะของหลอดไฟดวงที่ 2 ก่อนปิดแอพพลิเคชั่น	54
4.39	สถานะของหลอดไฟดวงที่ 2 หลังเปิดแอพพลิเคชั่นใหม่อีกครั้ง	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 แสดงลักษณะของอาร์ดูโน ยูโน (Arduino Uno) 5-6

2.2 รายละเอียดของเทคโนโลยี NFC ที่รองรับ RFID 10-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีของสมาร์ทโฟนมีการพัฒนาไปอย่างรวดเร็ว รวมถึงความต้องการในความสะดวกสบายที่มีเพิ่มขึ้นเรื่อยๆ ดังนั้นถ้ามีการนำเทคโนโลยีทางด้านสมาร์ทโฟนมาใช้งานในการควบคุมอุปกรณ์ไฟฟ้า จะทำให้เกิดความสะดวกรวดเร็วในชีวิตประจำวันมากยิ่งขึ้น โดยทำให้ผู้ใช้งานประหยัดเวลา ค่าใช้จ่าย มีความสะดวกรวดเร็ว รวมถึงเป็นการประหยัดพลังงานอีกด้วย จึงมีแนวคิดในการทำอุปกรณ์ควบคุมการเปิดและปิดของอุปกรณ์ไฟฟ้าด้วยโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ควบคู่กับไมโครคอนโทรลเลอร์

ปริญญานิพนธ์นี้ใช้สมาร์ทโฟนแทนรีโมตคอนโทรลแบบไร้สายในการควบคุมอุปกรณ์ไฟฟ้าโดยสามารถนำมาประยุกต์ใช้ในด้านการอำนวยความสะดวกให้สามารถเปิดและปิดอุปกรณ์ไฟฟ้าได้โดยที่ไม่ต้องเดินไปที่แผงสวิตช์ของอุปกรณ์ไฟฟ้านั้นๆ โดยผู้ใช้งานสามารถใช้คำสั่งจากสมาร์ทโฟนส่งผ่านไปยังบอร์ดไมโครคอนโทรลเลอร์ที่ควบคุมอุปกรณ์ไฟฟ้าให้ทำตามคำสั่ง ซึ่งคาดว่ามีจะประโยชน์เป็นอย่างมากในอนาคต

1.2 วัตถุประสงค์

1) ศึกษาหลักการทำงานของเทคโนโลยี NFC
2) ศึกษาการพัฒนาแอพพลิเคชั่นบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์
3) เพื่อเขียนคำสั่งการทำงานให้กับไมโครคอนโทรลเลอร์ (Microcontroller)
4) เพื่อสร้างระบบต้นแบบในการควบคุมเครื่องใช้ไฟฟ้าด้วยเทคโนโลยี NFC และ Bluetooth บนโทรศัพท์มือถือ ระบบปฏิบัติการแอนดรอยด์

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้นำเสนอระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบแอนดรอยด์ โดยทำการเขียนแอพพลิเคชั่นให้มีความสามารถในการเปิด/ปิดและปรับระดับเครื่องใช้ไฟฟ้า ซึ่งใช้ NFC tag เพื่อเปิดระบบ Bluetooth ของอุปกรณ์ และส่งคำสั่งในการควบคุมการทำงานผ่านระบบ Bluetooth ไปยังไมโครคอนโทรลเลอร์ โดยไมโครคอนโทรลเลอร์จะทำหน้าที่ควบคุมเครื่องใช้ไฟฟ้าที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 บลูทูธ

บลูทูธ (Bluetooth) เป็นเทคโนโลยีของอินเตอร์เฟสทางคลื่นวิทยุ ใช้ในการเชื่อมโยงการสื่อสารไร้สาย เป็นระบบสื่อสารของอุปกรณ์อิเล็คทรอนิคส์แบบสองทาง ด้วยคลื่นวิทยุระยะสั้น (Short-Range Radio Links) โดยปราศจากการใช้สายเคเบิ้ลหรือสายสัญญาณเชื่อมต่อ และไม่จำเป็นจะต้องใช้การเดินทางแบบเส้นตรงเหมือนกับอินฟราเรด ซึ่งถือว่าเพิ่มความสะดวกมากกว่าการเชื่อมต่อแบบอินฟราเรด ที่ใช้ในการเชื่อมต่อระหว่างโทรศัพท์มือถือกับอุปกรณ์ ในโทรศัพท์เคลื่อนที่รุ่นก่อนๆ และในการวิจัย ไม่ได้มุ่งเฉพาะการส่งข้อมูลเพียงอย่างเดียว แต่ยังศึกษาถึงการส่งข้อมูลที่เป็นเสียง เพื่อใช้สำหรับ Headset บนโทรศัพท์มือถือด้วย

2.1.1 การทำงานของบลูทูธ

บลูทูธจะใช้สัญญาณวิทยุความถี่สูงในแถบความถี่ 2.4 GHz แต่จะแยกย่อยออกไป ตามแต่ละประเทศ อย่างในแถบยุโรปและอเมริกา จะใช้ช่วง 2.400 ถึง 2.4835 GHz แบ่งออกเป็น 79 ช่องสัญญาณ และจะใช้ช่องสัญญาณที่แบ่งนี้ เพื่อส่งข้อมูลสลับช่องไปมา 1,600 ครั้งต่อ 1 วินาที ส่วนที่ญี่ปุ่นจะใช้ความถี่ 2.402 ถึง 2.480 GHz แบ่งออกเป็น 23 ช่อง ระยะทำการของบลูทูธ จะอยู่ที่ 5-10 เมตร โดยมีระบบป้องกันโดยใช้การป้อนรหัสก่อนการเชื่อมต่อ และป้องกันการดักสัญญาณระหว่างสื่อสาร โดยระบบจะสลับช่องสัญญาณไปมา จะมีความสามารถในการเลือกเปลี่ยนความถี่ที่ใช้ในการติดต่อเองอัตโนมัติ โดยที่ไม่จำเป็นต้องเรียงตามหมายเลขช่อง ทำให้การดักฟังหรือลักลอบขโมยข้อมูลทำได้ยากขึ้น การทำงานของบลูทูธเป็นการส่งข้อมูลแบบ 2 ทางระหว่าง 2 อุปกรณ์ด้วยคลื่นวิทยุระยะสั้น ในสัญญาณความถี่ license-free ISM (ความถี่สำหรับอุตสาหกรรมและการทดลองทางวิทยาศาสตร์) ซึ่งเป็นความถี่ที่อยู่ในช่วง 2.400 และ 2.4835 GHz และเพื่อป้องกันการชนกันของสัญญาณ (มีช่วงความถี่ใกล้เคียงกับสัญญาณ Microwave) วิธีการส่งจะอาศัยเทคโนโลยีที่ชื่อว่า frequency hopping ซึ่งมีหลักการทำงานคือ จะแบ่งช่องสัญญาณออกเป็น 79 ช่องความถี่ (ช่องละ 1 MHz) และจะทำการเปลี่ยนแปลงระดับของความถี่ในขณะที่กำลังส่งสัญญาณ 1,600 ครั้งต่อ 1 วินาที บลูทูธเชื่อมต่อในรูปแบบ oriented service ซึ่งจะมีตัวหนึ่งเป็นตัวเริ่มสื่อสาร โดยเราจะเรียกตัวนี้ว่า Master ส่วนอีกตัวจะเป็น Slave สำหรับบลูทูธนั้น สามารถเชื่อมต่อได้ 7 ตัว (active slave) พร้อมๆกันนั่นคือมีการเชื่อมต่อจากจุดที่เป็น master ตัว บลูทูธนั้นสามารถทำหน้าที่เป็นได้ทั้ง master-slave (เป็นได้ทั้ง master และ slave ในเวลาเดียวกัน เรียกว่า Scatternet)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บลูทูธมีแบ่งการทำงานจาก class 1 ถึง class 3 พร้อมทั้งมีระยะในการทำงานที่ไกลขึ้น โดยบลูทูธตัวหนึ่งนั้นสามารถใช้ประโยชน์ได้มากขึ้นจากการใช้พลังงานที่ต่ำกว่าปกติ โดยมีรายละเอียดดังนี้

Class 1 : ใช้พลังงานในการส่ง 100 mW (ส่งได้ 100 m)
Class 2 : ใช้พลังงานในการส่ง 25 mW (ส่งได้ 20 m)
Class 3 : ใช้พลังงานในการส่ง 1 mW (ส่งได้ 10 m)

ความปลอดภัยในการส่งข้อมูลผ่านทางบลูทูธมีเป้าหมายดังนี้

- Confidentiality (การรักษาความลับ)
- (device) authentication (การพิสูจน์ตัวตน)
- (device) authorization (การกำหนดสิทธิ์)
- Integrity (ความถูกต้อง สมบูรณ์ของข้อมูล)

และได้มีการกำหนดความปลอดภัยดังนี้

- Security Mode 1 : ระดับที่ไม่มีระบบความปลอดภัย
- Security Mode 2 : Service level security (ความปลอดภัยในระดับการให้บริการ) เช่นพวก Application ต่างๆ ที่สามารถปรับปรุงให้มีการ Cryptographic (การเข้ารหัสรูปแบบต่าง ๆ)
- Security Mode 3 : Device level security (ความปลอดภัยในระดับอุปกรณ์) หมายถึง การเข้าใจการเข้ารหัสซึ่งเป็นการพัฒนาใน LMP (Link Manager Protocol) รวมถึง Application ต่างๆ ที่นอกเหนือจากนี้

2.1.2 ประโยชน์ของบลูทูธ

หากเราต้องเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ต่างๆ ไม่ว่าจะเป็น เครื่องพิมพ์ คีย์บอร์ด เมาส์ หรือลำโพง การเชื่อมต่อในปัจจุบัน ส่วนใหญ่จะใช้สายเคเบิ้ลเป็นตัวเชื่อมต่อทั้งหมด (Serial และ USB) ซึ่งอาจจะไม่สะดวกทั้งในด้านการใช้สอย การเคลื่อนย้าย และความเรียบร้อยต่างๆ แต่หากเครื่อง PC มีอุปกรณ์บลูทูธ ก็สามารถติดต่อเข้าหากันได้โดยใช้คลื่นแทนการใช้สายไฟเชื่อมต่ออุปกรณ์ต่างๆ ทั้งหมด ทั้งการส่งไฟล์ภาพ, เสียง, ข้อมูล อีกทั้งระบบเชื่อมต่อผ่าน CSD และ GPRS บนโทรศัพท์มือถือ ก็สามารถทำได้โดยไม่จำเป็นต้องใช้สาย ซึ่งจะช่วยลดความยุ่งยาก อีกทั้งยังเพิ่มความสะดวกสบายในการทำงานมากขึ้นด้วย

2.1.3 เป้าหมายของบลูทูธ

เทคโนโลยีบลูทูธพัฒนาขึ้นมา โดยมีเป้าหมายคือ

- Low cost implementation พัฒนาให้มีราคาต่ำ ที่สามารถให้คนทั่วไปใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Small implementation size ทำให้บลูทูธมีขนาดเล็กที่สุด เพื่อให้ใช้งานได้สะดวก

- Low power consumption ให้บลูทูธใช้พลังงานในการทำงานน้อย เพื่อให้สามารถติดต่อกันได้โดยไร้ข้อจำกัด

- Robust, high quality data & voice transfer พัฒนาให้บลูทูธมีความทนทานในการใช้งานและสามารถส่งทั้งข้อมูลและเสียงได้อย่างมีประสิทธิภาพ

- Open global standard เป็นมาตรฐานเปิด คือให้ผู้ที่สนใจสามารถนำไปพัฒนาต่อได้ ทำให้เทคโนโลยีพัฒนาได้อย่างรวดเร็ว

2.2 ไมโครคอนโทรลเลอร์

2.2.1 คุณสมบัติของอาร์ดูโน

อาร์ดูโนเป็นโครงการพัฒนาไมโครคอนโทรลเลอร์ของตระกูลเอวีอาร์ (AVR : Automatic Voltage Regulator) แบบโอเพ่นซอร์สโดยนิยมใช้เอวีอาร์ในตระกูล ATMEGA88, ATMEGA168, ATMEGA328 โดยมีตัวถังแบบดีไอพี (DIP) ทำให้ง่ายต่อการพัฒนา อาร์ดูโนคือเครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้เครื่องคอมพิวเตอร์ตั้งโต๊ะ ตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยว และมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน อาร์ดูโนสามารถประยุกต์ทำเครื่องใช้อัจฉริยะรับสัญญาณจากสวิทช์ เซนเซอร์ ควบคุมหลอดไฟ มอเตอร์หรืออุปกรณ์อื่น ๆ อาร์ดูโนเป็นได้ทั้งแบบทำงานอิสระ หรือทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ตั้งโต๊ะ บอร์ดสามารถประกอบขึ้นใช้เองหรือจะซื้อสำเร็จที่มีวางขาย ส่วนโปรแกรมพัฒนาอาร์ดูโนสามารถดาวน์โหลดได้ฟรี ในตลาดไมโครคอนโทรลเลอร์มีตัวเลือกมากมาย เช่น พาราแลกส์เบสิคสแตมป์ (Parallax Basic Stamp), เนทมีเดียบีเอกส์ 24 (Netmedia's BX-24), พิกเจตส์ (Pidgets), เอ็มไอทีแฮนดี้บอร์ด (MIT's Handyboard) และอีกหลายบริษัทที่มีคุณสมบัติใกล้เคียงกัน คือทำให้ใช้งานง่ายและเน้นการโปรแกรมไมโครคอนโทรลเลอร์เป็นหลัก ซึ่งอาร์ดูโนมีข้อแตกต่างที่เห็นได้ชัดเจน ดังนี้

- ราคาไม่แพงเมื่อเทียบกับบอร์ดอื่น บอร์ดอาร์ดูโนที่ราคาถูกที่สุดสามารถทำใช้เองได้หรือซื้อสำเร็จด้วยเงินไม่เกิน 30 ดอลล่าร์

- ทำงานได้หลากหลายแพลทฟอร์ม โปรแกรมพัฒนาอาร์ดูโนทำงานได้ทั้งบนวินโดวส์ แมคอินทอช (Macintosh OSX) และบนลีนุกซ์ในขณะที่บอร์ดอื่นทำงานได้เฉพาะบนวินโดวส์

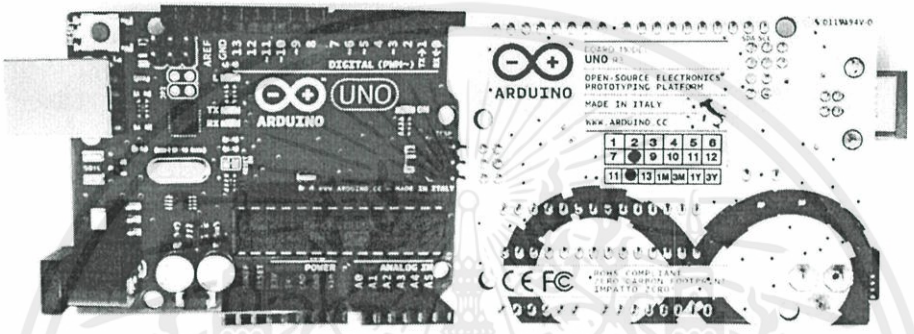
- ใช้งานง่าย มีโปรแกรมพัฒนาที่ไม่ซับซ้อน โปรแกรมพัฒนาอาร์ดูโนใช้งานง่ายสำหรับมือใหม่และมีความสามารถครบความต้องการของนักพัฒนามืออาชีพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เปิดเผยซอร์สโค้ดและนำไปพัฒนาต่อยอดได้และสามารถเพิ่มเติมความสามารถผ่านซีพลัสพลัสไลบรารี่ (C++ library)

- เปิดเผยวงจรและนำไปพัฒนาขยายฮาร์ดแวร์ได้ อาร์ดูโนใช้โมโครคอนโทรลเลอร์ของเอทเมลเบอร์ ATMEGA8 และ ATMEGA168 วงจรของบอร์ดตีพิมพ์แบบเปิดเผยวงจรภายใต้ Creative Commons License

โดยบอร์ดของอาร์ดูโน่แสดงดังรูปที่ 2.1



รูปที่ 2.1 ลักษณะของอาร์ดูโน ยูโน [5]

ตารางที่ 2.1 ลักษณะของอาร์ดูโน ยูโน (Arduino Uno)

ไมโครคอนโทรลเลอร์	ATmega328
แรงดันไฟฟ้า	5 โวลต์
แรงดันไฟฟ้าอินพุต (แนะนำ)	7-12 โวลต์
แรงดันไฟฟ้าอินพุต (จำกัด)	6-20 โวลต์
ขาอินพุต-เอาท์พุตดิจิทัล	14 (ซึ่งขา 6 ให้ผลลัพธ์เป็น การปรับความกว้างของพัลซ์ Pulse Width Modulation : PWM)
ขาอินพุตอนาล็อก	6
กระแสตรงต่อขาอินพุต-เอาท์พุต	40 มิลลิแอมแปร์
กระแสตรงสำหรับขา 3.3 โวลต์	50 มิลลิแอมแปร์
หน่วยความจำ	32 กิโลไบท์
หน่วยความจำชั่วคราว (static RAM : SRAM)	2 กิโลไบท์
หน่วยความจำรอม (Ectrically Erasable Programmable ROM : EEPROM)	1 กิโลไบท์
ความเร็วของสัญญาณนาฬิกา (Clock Speed)	16 เมกะเฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 รีเลย์

รีเลย์เป็นอุปกรณ์อิเล็กทรอนิกส์ ทำหน้าที่ตัดต่อวงจร โดยอาศัยหลักการของอำนาจแม่เหล็กไฟฟ้าเพื่อควบคุมการเปลี่ยนแปลงของหน้าสัมผัส รีเลย์มีความหมายในแบบของ นักอิเล็กทรอนิกส์ว่าตัวถ่ายทอดกำลังเพราะป้อนกำลังงานไฟฟ้าให้แก่รีเลย์เพียงเล็กน้อยก็สามารถควบคุมวงจรกำลังงานสูง ที่ต่ออยู่กับหน้าสัมผัส (นิยมเรียกว่าคอนแท็ก) ของรีเลย์ได้โดยเมื่อทำการจ่ายกระแสไฟฟ้าให้กับขดลวดรีเลย์ (Coil) จะทำให้เกิดสนามแม่เหล็กรอบขดลวดซึ่งอำนาจแม่เหล็กชั่วคราวที่เกิดขึ้นมีค่าเพียงพอที่จะชนะแรงสปริงทำให้เกิดการเปลี่ยนแปลงที่หน้าสัมผัส รีเลย์มีรูปร่างและขนาดที่แตกต่างกัน ในการเลือกใช้งาน จะต้องคำนึงถึงชนิดของรีเลย์ อัตรากำลังสูงสุดที่รีเลย์สามารถทนได้ ความถี่ที่ใช้งานและอื่น ๆ เพื่อให้สามารถใช้งานรีเลย์ได้อย่างถูกต้องและเหมาะสม

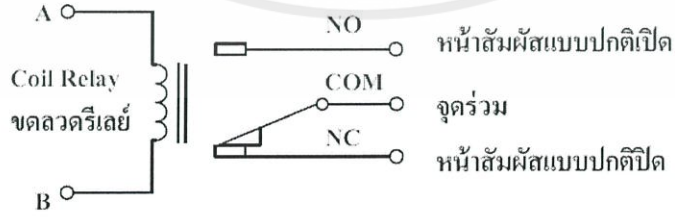
รีเลย์แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์ควบคุม (Control Relay) จะเป็นรีเลย์ที่มีขนาดเล็ก ใช้กำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าต่ำหรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุมนิยมเรียกกันว่า รีเลย์

2. รีเลย์กำลัง (Power relay) นิยมเรียกกันว่าคอนแทกเตอร์ (Contactor or Magnetic Contactor) ซึ่งเป็นรีเลย์ที่มีขนาดใหญ่กว่ารีเลย์ควบคุม นิยมใช้งานกับกำลังไฟฟ้าสูง ใช้ในการควบคุมไฟฟ้ากำลังที่มีขนาดใหญ่ เช่น การควบคุมมอเตอร์ไฟฟ้าสามเฟส เป็นต้น

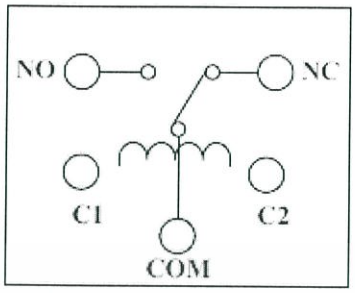
2.3.1 โครงสร้างและสัญลักษณ์ของรีเลย์

โครงสร้างภายในของรีเลย์จะประกอบไปด้วยขดลวด 1 ชุด และหน้าสัมผัสซึ่งในหน้าสัมผัส 1 ชุด จะประกอบด้วยหน้าสัมผัสแบบปกติปิด (Normally Close หรือ NC) และหน้าสัมผัสแบบปกติเปิด (Normally Open หรือ NO) ในสภาวะปกติขาเอ็นซีนี้จะต่ออยู่กับขาร่วม (Common หรือ COM) และเมื่อมีการจ่ายกระแสไฟฟ้าในปริมาณที่เพียงพอเข้าที่ขดลวดรีเลย์หรือมีแรงดันตกคร่อมขดลวดจะทาให้เกิดการเปลี่ยนแปลงที่หน้าสัมผัสทำให้จุดขาร่วมต่อกับหน้าสัมผัสแบบปกติปิดในรีเลย์ 1 ตัว อาจมีหน้าสัมผัสมากกว่า 1 ชุด ซึ่งขึ้นอยู่กับผู้ผลิต โดยสัญลักษณ์ของรีเลย์แสดงดังรูปที่ 2.2 และ 2.3



รูปที่ 2.2 สัญลักษณ์ของรีเลย์ [6]

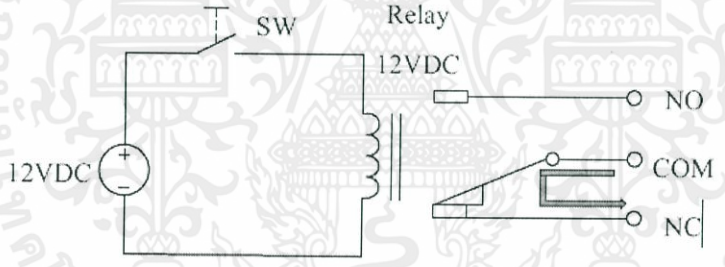
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



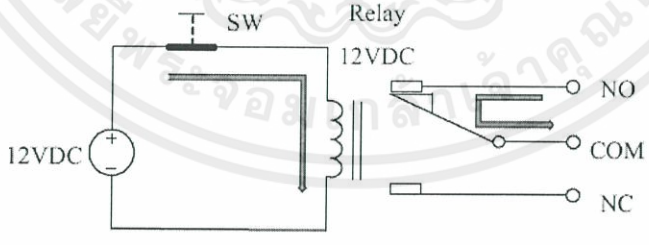
รูปที่ 2.3 ตำแหน่งขาของรีเลย์แบบ 5 ขา [7]

2.3.2 หลักการทำงานของรีเลย์

เมื่อไม่มีการจ่ายกระแสไฟฟ้าที่ขดลวดรีเลย์จะทำให้จุดร่วมที่ต่อกับหน้าสัมผัสแบบปกติปิด แต่เมื่อทำการจ่ายกระแสไฟฟ้าให้กับขดลวดรีเลย์จะทำให้เกิดสนามแม่เหล็กรอบขดลวดซึ่งอำนาจแม่เหล็กชั่วคราวที่เกิดขึ้นมีค่าเพียงพอที่จะชนะแรงสปริง เกิดการเปลี่ยนแปลงที่หน้าสัมผัสทำให้จุดร่วมต่ออยู่กับหน้าสัมผัสแบบปกติเปิดและเมื่อไม่ได้จ่ายกระแสไฟฟ้าสปริงจะดึงหน้าสัมผัสกลับมาต่อกับหน้าสัมผัสแบบปกติปิดอีกครั้งหนึ่ง ดังแสดงในรูปที่ 2.4 และ 2.5



รูปที่ 2.4 ขณะที่ไม่จ่ายกระแสไฟฟ้าแก่ขดลวดรีเลย์ [7]



รูปที่ 2.5 ขณะจ่ายกระแสไฟฟ้าแก่ขดลวดรีเลย์ [7]

จะเห็นว่ารีเลย์ทำงานโดยอาศัยการเหนี่ยวนำจากสนามแม่เหล็ก ดังนั้นการใช้งานรีเลย์จะเกิดผลกระทบจากอำนาจแม่เหล็กรีเลย์ ในขณะที่รีเลย์เป็นอุปกรณ์ที่มีความเร็วในการทำงานต่าง ๆ เช่น รีเลย์ชนิดแรงดันต่ำที่กระตุ้นขดลวดไม่เกิน 24 โวลต์ จะใช้เวลาในการทำงานประมาณ 10 – 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มิลลิวินาที (Millisec) และรีเลย์ขนาดใหญ่ที่ใช้ควบคุมมอเตอร์ในโรงงานอุตสาหกรรมนั้นอาจใช้เวลาในการทำงานมากกว่า 100 มิลลิวินาที ดังนั้นในการใช้งานควรแก้ไขปัญหาของรีเลย์ไม่ให้ไปรบกวนการทำงานของวงจรอื่น ๆ เช่น การแยกกราวด์ (Ground) ของวงจรหรือการแบ่งส่วนของวงจรรีเลย์กับวงจรควบคุมหรือทำการชีลด์ (Shield) วงจรเพื่อป้องกันการรบกวนของรีเลย์

2.4 แอนดรอยด์ (Android)

แอนดรอยด์ เป็นระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ต คอมพิวเตอร์ เน็ตบุ๊ก ทำงานบนลินุกซ์ เคอร์เนล เริ่มพัฒนาโดยบริษัทแอนดรอยด์ จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล และนำแอนดรอยด์ไปพัฒนาต่อ ภายหลังถูกพัฒนาในนามของโอเพ่นแฮนด์เซ็ท อัลลิแอนซ์ (Open Handset Alliance) ทางกูเกิลได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆ ด้วยภาษาจาวา และควบคุมอุปกรณ์ผ่านทางชุดจาวาไลบรารี่ (Java libraries) ที่กูเกิลพัฒนาขึ้น แอนดรอยด์ได้เป็นที่รู้จักต่อสาธารณชนเมื่อวันที่ 5 พฤศจิกายน พ.ศ. 2550 โดยทางกูเกิลได้ประกาศก่อตั้ง โอเพ่น แฮนด์เซ็ท อัลลิแอนซ์ กลุ่มบริษัทฮาร์ดแวร์, ซอฟต์แวร์ และการสื่อสาร 48 แห่ง ที่ร่วมมือกันเพื่อพัฒนา มาตรฐานเปิด สำหรับอุปกรณ์มือถือ ลิขสิทธิ์ของโค้ดแอนดรอยด์นี้จะใช้ในลักษณะของซอฟต์แวร์เสรี เวอร์ชั่นล่าสุด (วันที่ 9 ธันวาคม พ.ศ.2556) ของแอนดรอยด์คือ 4.4.2 ที่ชื่อคิทแคท (KitKat) ความสามารถใหม่ของ แอนดรอยด์ 4.4.2 ที่เพิ่มขึ้นมาคือการเปิดใช้งานกูเกิลด้วยคำสั่งเสียงและ Google Cloud Print ที่สามารถสั่งพิมพ์เอกสารได้จากระยะไกลได้

2.5 จาวา (JAVA)

ภาษาจาวา เป็นภาษาโปรแกรมเชิงวัตถุพัฒนาโดย เจมส์ กอสลิง และทีมงานวิศวกร ที่ซัน ไมโครซิสเต็มส์ ภาษาจาวาถูกพัฒนาขึ้นในปี พ.ศ. 2534 โดยเป็นส่วนหนึ่งของ โครงการกรีน และสำเร็จออกสู่สาธารณะในปี พ.ศ. 2538 ซึ่งภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส (C++) โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) และแม้ว่าจะมีชื่อคล้ายกัน แต่ภาษาจาวาไม่มีความเกี่ยวข้องใด ๆ กับภาษาจาวาสคริปต์ (JavaScript) ปัจจุบันมาตรฐานของภาษาจาวาดูแลโดยจาวาคอมมิวนิตี้โพรเซส (Java Community Process) ซึ่งเป็นกระบวนการอย่างเป็นทางการ ที่อนุญาตให้ผู้ที่สนใจเข้าร่วมกำหนดความสามารถในจาวาแพลตฟอร์มได้

2.6 เอ็นเอฟซี (NFC : Near field communication)

เอ็นเอฟซีเป็นเทคโนโลยีสื่อสารไร้สายระยะสั้นระยะประมาณ 4 - 10 ซ.ม. ที่ใช้ได้ดีกับโครงสร้างพื้นฐานแบบไม่สัมผัส เพื่อการสื่อสารระหว่างเครื่องมืออิเล็กทรอนิกส์ในระยะใกล้ๆ เอ็นเอฟซี ถูกพัฒนาขึ้นโดยโซนี่ (Sony) และเอ็นเอ็กซ์พี (NXP) โดยใช้คลื่นความถี่ 13.56 เมกะเฮิรตซ์ บนพื้นฐานมาตรฐาน ISO 14443 ปัจจุบันบริษัททั้งสองได้ร่วมมือกับบริษัทผู้ผลิตและพัฒนา

เอกสารนี้เป็นลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามการนำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทรศัพท์เคลื่อนที่ จัดตั้งเป็น เอ็นเอฟซี ฟอรั่ม (NFC Forum) เพื่อให้เกิดการใช้งานในรูปแบบต่างๆ มากขึ้น ในระยะเริ่มแรกมีบริษัทโทรศัพท์มือถือชั้นนำของโลกประกาศนำเทคโนโลยีนี้มา ใช้กับโทรศัพท์มือถือแล้ว เช่น โนเกีย (Nokia), ซัมซุง (Samsung), โมโตโรล่า (Motorola) เป็นต้น เพื่อจะใช้เอ็นเอฟซีในมือถือนั้นจะต้องมีอยู่สองส่วนด้วยกันคือ อย่างแรกมือถือจะต้องเป็นเอ็นเอฟซีโฟน (NFC phone) คือจะต้องมีสายอากาศอยู่ภายใน (built-in antenna) และมีเอ็นเอฟซีชิป (NFC chip) และอย่างที่สองจะต้องมีซีเคียวชิป (secure chip) ที่ใช้เก็บข้อมูลที่ต้องการความปลอดภัย (security) สูงๆเช่น ข้อมูลบัตรเครดิต/เดบิต (credit/debit information), ธุรกรรมทางการเงิน (banking applications) ซึ่งในกรณีนี้มีอยู่สองแบบด้วยกัน แบบแรกคือมีชิปต่างหากเพื่อใช้เก็บข้อมูลเหล่านี้ กับอีกแบบหนึ่งคือเอาไปเก็บในซิมการ์ด (SIM card) ซึ่งแบบหลังเป็นสิ่งที่ตัวดำเนินการของโทรศัพท์มือถือ (mobile operators) ต่างๆต้องการ โดยประโยชน์ของเอ็นเอฟซีแสดงดังรูปที่ 2.6 เราจะเห็นว่าเอ็นเอฟซี สามารถใช้งานให้เกิดประโยชน์ได้อย่างหลากหลาย

รูปที่ 2.6 การใช้ประโยชน์ของเอ็นเอฟซี [8]

สำหรับเอ็นเอฟซีกับมือถือมีการปรากฏใช้ในหลากรูปแบบ เช่น นำไปฝังไว้หลังตัวเครื่อง เช่นในมือถือเน็กซัส เอส (Nexus S) หรือในมือถือหลายๆยี่ห้อที่ฝังชิปเอ็นเอฟซี ไว้ก่อนหน้านี้ ซึ่งมีโนเกียเป็นผู้นำนวัตกรรมเมื่อ 5 ปีที่แล้ว เช่น โนเกีย 6212, ซัมซุง SHW-A170K โดยปัจจุบันกูเกิลได้อัพเกรดให้เฟิร์มแวร์แอนดรอยด์ 2.3.3 รองรับชิปเอ็นเอฟซี ทั้งหมดแล้ว โดยในแรกเริ่มจะเน้นการใช้เอ็นเอฟซี บนมือถือเพื่อแทนกระเป๋าสตางค์เงินสด ในการชำระค่าโดยสาร ค่าอาหาร ซึ่งเป็นเงินจำนวนไม่มาก โดยรูปแบบการจ่ายเงินด้วยชิปเอ็นเอฟซี บนมือถือนี้มีใช้กันอย่างแพร่หลายในประเทศญี่ปุ่น เกาหลี และเกาะฮ่องกง ปัจจุบันรัฐบาลจีนก็เตรียมใช้ระบบเอ็นเอฟซี กับโครงสร้างคมนาคมของประเทศด้วยเช่นกัน ส่วนในอเมริกา และยุโรป ก็ได้ริเริ่มระบบการจ่ายเงินด้วยการแตะมือถือไปแล้วระยะหนึ่ง โดยได้สร้างและทดสอบระบบการจ่ายเงินบนมือถือ (Mobile Payment) ส่วนประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไทยได้เริ่มนำนวัตกรรมนี้เข้ามาตั้งแต่ 3 ปีก่อน โดยค่ายบัตรเครดิตวีซ่า (Vesa) ที่นำเอาระบบเพย์เวฟ (Paywave) มาใช้กับโนเกีย 6212 ซึ่งก็มีผู้ร่วมโครงการเป็นธนาคารกสิกรไทย ค่ายมือถือเอไอเอส ร่วมกับร้านค้ากว่า 1,500 ร้านร่วมทดลองระบบนี้ แต่ก็ไม่ต่อเนื่องเพราะมือถือรุ่นนี้ไม่ได้เป็นที่นิยม และต่อมาทางเครือข่ายโทรศัพท์ทรูก็ได้สร้าง ทัชซิม (Touch SIM) ที่ร่วมพัฒนาจากทีมพัฒนาในจีน ออกเป็นซิมที่ติดแผงวงจร RFID เข้าไป แต่ก็ไม่ประสบความสำเร็จอย่างที่คาดไว้ เพราะมีขนาดใหญ่ และใช้กับมือถือไม่ได้ทุกรุ่น ซึ่งเทคโนโลยีเอ็นเอฟซี จะรองรับเทคโนโลยีอาร์เอฟไอดี ตามรายละเอียดดังตารางที่ 2.2

ตารางที่ 2.2 รายละเอียดของเทคโนโลยี NFC ที่รองรับ RFID

NFC Forum Platform	RFID Compatible
NFC Forum Type 1 tag	Innovision Topaz
NFC Forum Type 2 tag	Mifare Ultralight Mifare Ultralight C
NFC Forum Type 3 tag	Sony Felica
NFC Forum Type 4 tag	DESfire SmartMX

เทคโนโลยีเอ็นเอฟซีนี้ช่วยให้การสื่อสารระหว่างอุปกรณ์สามารถทำได้ง่ายขึ้น ต่างจากเทคโนโลยีไร้สายประเภทอื่น ได้แก่ ไวไฟ (Wifi) หรือ บลูทูธ (Bluetooth) ที่ต้องมีตั้งค่าต่าง ๆ ก่อนการใช้งาน แต่เทคโนโลยีเอ็นเอฟซี เพียงแค่นำอุปกรณ์มือถือ เช่น โทรศัพท์ ไปใกล้กับเครื่องอ่านหรืออาร์เอฟไอดีการ์ด ก็สามารถที่จะทำการส่งข้อมูลระหว่างกันได้เลย โดยไม่จำเป็นต้องมีการตั้งค่าใดๆ ก่อนการใช้งาน การใช้งานเทคโนโลยีเอ็นเอฟซีมีได้ 3 ลักษณะ

1. ทำงานเป็นอาร์เอฟไอดีการ์ด (RFID Tag) ในอุปกรณ์มือถือ หมายความว่าอุปกรณ์มือถือตามมาตรฐานเอ็นเอฟซี จะทำตัวเป็นบัตรในรูปแบบใดก็ได้ตามมาตรฐาน ISO 14443 เพื่อใช้ในงานทางธุรกรรม โทรศัพท์มือถือที่มีเครื่องอ่านเอ็นเอฟซี ฝังอยู่สามารถทำงานเป็นการ์ดอาร์เอฟไอดีได้ ซึ่งต่างจากเครื่องอ่านอาร์เอฟไอดีในปัจจุบันที่ทำหน้าที่เป็นเครื่องอ่านเพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 การใช้งานบัตรวีซ่า เพย์เวฟ ที่ไม่ต้องรูด แค่แตะสัมผัส [9]

การทำงานในลักษณะนี้ส่วนใหญ่จะใช้ในแอพพลิเคชั่นในเรื่องการเงิน เช่น การจ่ายเงินชำระค่าผ่านทาง ต่าง ๆ เพียงแค่นำโทรศัพท์มือถือไปใกล้กับเครื่องอ่านอาร์เอฟไอดี ที่ติดตั้งไว้ที่จุดชำระเงิน ก็สามารถทำการชำระเงินได้ แทนการชำระเงินด้วยบัตรอาร์เอฟไอดีหรือเงินสด นอกจากนั้นเราสามารถขยายการใช้งานอุปกรณ์เอ็นเอฟซีชิ้นเดียวเป็นบัตรหลายใบได้ เช่น เป็นบัตรเครดิต บัตรโดยสารรถไฟฟ้า บัตรเงินสด บัตรสะสมแต้ม เป็นต้น ทำให้เมื่อเอ็นเอฟซีเป็นที่นิยมและผู้ให้บริการบัตรต่างๆทำแอพพลิเคชั่น สำหรับบริการของตนลงบนอุปกรณ์ เราจะได้ไม่ต้องพกบัตรหลายใบ โดยรูปที่ 2.7 แสดงตัวอย่างการใช้งานบัตรวีซ่า เพย์เวฟ

รูปที่ 2.8 การใช้มือถือทำงานเป็นอาร์เอฟไอดีการ์ด [10]

2. ทำงานเป็นเครื่องอ่านอาร์เอฟไอดี อุปกรณ์มือถือ เช่น โทรศัพท์มือถือที่มีเครื่องอ่านเอ็นเอฟซี ฝังอยู่ สามารถทำงานเป็นเครื่องอ่านอาร์เอฟไอดี ดังรูปที่ 2.8 เมื่อต้องการอ่านข้อมูลจากอาร์เอฟไอดีการ์ด ในโหมดนี้อุปกรณ์เอ็นเอฟซี สามารถทำตัวเสมือนเป็นเครื่องอ่านเขียนแท็ก โดยจะสามารถอ่านข้อมูลจากแท็ก ที่ติดอยู่ในจุดให้บริการข้อมูล โดยที่จุดให้บริการข้อมูลจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาร์เอฟไอดีสติกเกอร์ (RFID Sticker) เมื่อต้องการอ่านข้อมูลจากอาร์เอฟไอดีสติกเกอร์เพียงแค่นำโทรศัพท์มือถือที่มีเครื่องอ่านไปอ่านอาร์เอฟไอดีสติกเกอร์ที่จุดให้บริการ ข้อมูลข้างในก็จะปรากฏขึ้นมาบนโทรศัพท์มือถือ

3. การสื่อสารในลักษณะเพียร์ทูเพียร์ เครื่องอ่านเอ็นเอฟซี สองเครื่องสามารถที่จะติดต่อ สื่อสารกันโดยตรงได้ ดังรูปที่ 2.9 โหมดนี้จะทำการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์เอ็นเอฟซี ด้วยกัน คล้ายกับการที่มือถือมีบลูทูธ แล้วทำการแลกเปลี่ยนข้อมูลโดยการจับคู่ (Pair) เข้าด้วยกันแล้วแลกเปลี่ยนข้อมูล เช่น นามบัตร รูปถ่าย แฟ้มข้อมูลอื่นๆ แต่ว่าสำหรับเอ็นเอฟซีแล้ว ไม่ต้องมีการจับคู่เหมือนบลูทูธ เพียงแค่เลือกข้อมูลที่ต้องการแลกเปลี่ยนแล้วนำอุปกรณ์เอ็นเอฟซี ที่รองรับโหมดนี้มาแตะกัน ข้อมูลก็จะทำการโอนถ่ายกันระหว่างเครื่อง เพราะรัศมีทำการของเอ็นเอฟซี อยู่ในระดับน้อยกว่า 10 ซ.ม. ซึ่งต่างจากบลูทูธ ซึ่งออกแบบไว้ให้สื่อสารข้อมูลในระยะหลายเมตร การแลกเปลี่ยนข้อมูลทำได้ผ่านโปรโตคอลทีซีพี/ไอพี (TCP/IP) นอกจากแลกเปลี่ยนข้อมูลแล้วยังสามารถใช้ทำการเข้าจังหวะ ข้อมูลกับอุปกรณ์อื่นๆได้ด้วย ตัวอย่างเช่น โทรศัพท์มือถือสองเครื่องที่มีฟังก์ชันเอ็นเอฟซี สามารถที่จะส่งข้อมูลให้แก่กันได้โดยตรง เพียงนำโทรศัพท์ทั้งสองเครื่อง เข้ามาใกล้กันในระยะที่เครื่องอ่านที่อยู่ในโทรศัพท์ทั้งสองสามารถอ่านกันได้ ก็สามารถที่จะส่งข้อมูลถึงกันได้ โดยไม่ต้องผ่านเครือข่ายของโทรศัพท์มือถือ ไม่ว่าจะเป็นจีพีอาร์เอส (GPRS) หรือเอดจ์ (EDGE) เป็นต้น

รูปที่ 2.9 การสื่อสารในลักษณะเพียร์ทูเพียร์ [8]

ถึงแม้ว่าเทคโนโลยีเอ็นเอฟซี จะได้รับการกล่าวถึงเป็นอย่างมากในปัจจุบัน แต่อุปกรณ์มือถือที่รองรับเทคโนโลยีเอ็นเอฟซี ยังมีอยู่อย่างจำกัด ดังนั้นในช่วงการเปลี่ยนผ่านจึงมีการพัฒนาเทคโนโลยีต่าง ๆ ขึ้นมาเพื่อรองรับกับเทคโนโลยีเอ็นเอฟซี ที่กำลังจะเกิดขึ้น ตัวอย่างเช่น เทคโนโลยีซิมการ์ด + เสาอากาศ เป็นการต่อเสาอากาศเพิ่มจากซิมการ์ดของโทรศัพท์มือถือ ซึ่งเทคโนโลยีเหล่านี้ใช้อาร์เอฟไอดีคลื่นความถี่ 13.56 เมกะเฮิรตซ์ เหมือนกับเทคโนโลยีเอ็นเอฟซี เทคโนโลยีนี้มีการใช้อย่างแพร่หลายในประเทศจีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

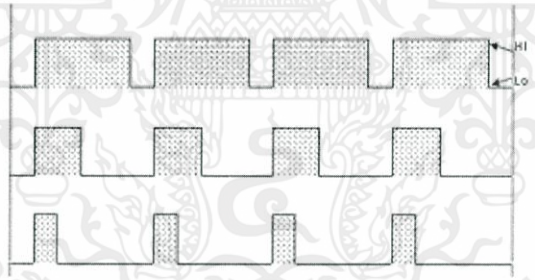
2.6.1 ประโยชน์ของ NFC และการนำไปใช้

- ใช้สำหรับการชำระเงิน
- ใช้ในการแลกเปลี่ยนข้อมูลระหว่างโทรศัพท์ หรืออุปกรณ์โมบาย เช่น แท็บเล็ต เป็นต้น
- การเชื่อมต่อกับอุปกรณ์อื่นๆ ระหว่างโทรศัพท์ กับอุปกรณ์ต่อพ่วง เช่น ลำโพง เป็นต้น
- การซื้อตั๋วหนัง
- กุญแจอิเล็กทรอนิคส์

การประยุกต์ใช้เทคโนโลยีเอ็นเอฟซีมาประยุกต์ใช้กับกิจกรรมอื่นๆยังมีอีกมากมาย เช่น นำมาใช้แทน บัตรเครดิต บัตรพนักงาน บัตรรถไฟฟ้า ซึ่งจะช่วยลดการใช้ทรัพยากรธรรมชาติได้มาก

2.7 พัลส์วิดท์มอดูเลชั่น (Pulse Width Modulation (PWM))

PWM จะใช้หลักการที่ส่งค่าแบบดิจิทัล (0,1) ด้วยความถี่ค่าหนึ่ง แต่จะส่งค่าดิจิทัลให้มีสัญญาณสูง (1) สลับกับสัญญาณต่ำ (0) โดยให้ระยะเวลาของแต่ละชนิดสัญญาณต่างกัน เพื่อให้ค่าเฉลี่ยของสัญญาณทั้งหมดออกมาเป็นค่าที่ต้องการ ดังรูปที่ 2.10



รูปที่ 2.10 สัญญาณ PWM ที่มีคาบต่างกัน [11]

2.8 ซีอาร์ซี (CRC : Cyclic Redundancy Check)

การตรวจสอบความผิดพลาดแบบ CRC เป็นวิธีหนึ่งที่ได้รับความนิยมในการใช้งานบนเครือข่ายและจัดได้ว่าเป็นวิธีที่มีประสิทธิภาพสูงกว่าการใช้บิตตรวจสอบ (Parity Bit) และการหาผลรวม (Check Sum) เนื่องจากสามารถตรวจจับบิตผิดพลาดได้ครอบคลุมและแน่นอนกว่า โดยเฉพาะ CRC-32 บิต นั้นมีอัตราความแน่นอนหรืออัตราความถูกต้องในการตรวจจับข้อผิดพลาดได้มากถึง 99.99999998%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

โครงงานนี้จัดทำขึ้นเพื่อเพิ่มความสะดวกและประหยัดเวลาในการควบคุมเครื่องใช้ไฟฟ้า โดยนำเสนอการใช้งานผ่านโทรศัพท์ที่มีเทคโนโลยีบลูทูธและเอ็นเอฟซี ซึ่งบล็อกไดอะแกรมของโครงงานแสดงดังรูปที่ 3.1

รูปที่ 3.1 บล็อกไดอะแกรมแสดงระบบของโครงงาน

3.1 การออกแบบ

การทำงานของระบบเริ่มจากนำโทรศัพท์มือถือระบบแอนดรอยด์ ซึ่งในที่นี้เป็นรุ่นซัมซุงแกแลคซี่ เอสสาม (Samsung Galaxy S3) ไปแตะกับแท็กเพื่อเปิดฟังก์ชั่นบลูทูธและเปิดแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้า โดยโทรศัพท์จะทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยอัตโนมัติ จากนั้นโทรศัพท์ก็จะส่งคำสั่งการควบคุมเครื่องใช้ไฟฟ้าตามที่เรากำหนดไปยังไมโครคอนโทรลเลอร์ โดยไมโครคอนโทรลเลอร์ก็จะทำการจ่ายไฟให้รีเลย์ ซึ่งจะทำให้เครื่องใช้ไฟฟ้าทำงาน จากนั้นไมโครคอนโทรลเลอร์จะส่งค่าสถานะของเครื่องใช้ไฟฟ้ากลับไปยังโทรศัพท์ นอกจากนี้ในแอพพลิเคชั่นยังมีปุ่มปรับระดับเครื่องใช้ไฟฟ้าและบอกอุณหภูมิ ณ ขณะนั้นอีกด้วย ซึ่งแผนผังการทำงานแสดงดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

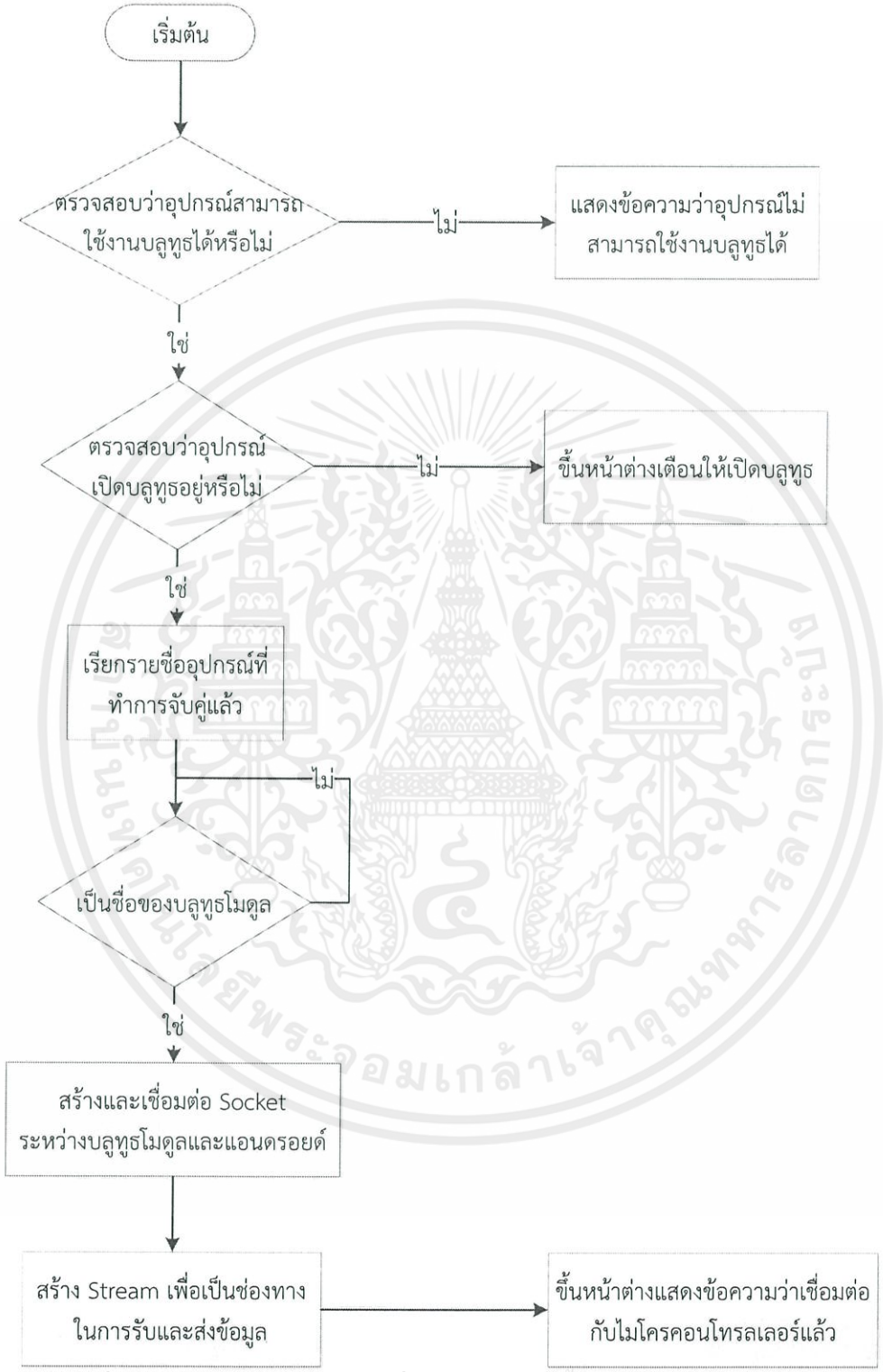
รูปที่ 3.2 แผนผังการทำงานของระบบ

โดยการทำงานของระบบสามารถอธิบายได้ ดังนี้

3.1.1 การเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์

การเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์นั้น เริ่มจากการนำโทรศัพท์ไปแตะกับแท็กเพื่อเปิดโหมดบลูทูธและเปิดแอพพลิเคชั่นในการควบคุมเครื่องใช้ไฟฟ้า ซึ่งแอพพลิเคชั่นนี้จะทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านบลูทูธโมดูล (ที่ได้จับคู่กันไว้แล้ว) โดยอัตโนมัติดังแสดงในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

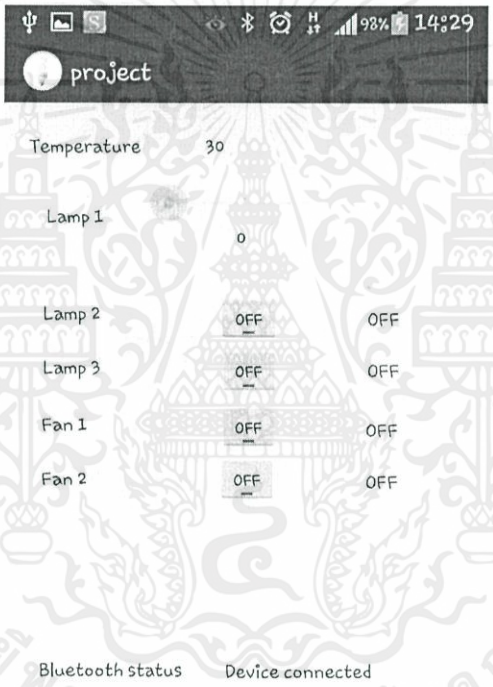


รูปที่ 3.3 โฟลว์ชาร์ตการเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านบลูทูธโมดูล

เอกสารนี้เป็นเอกสาร[illegible]ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การส่งคำสั่งไปยังไมโครคอนโทรลเลอร์

การส่งคำสั่งไปยังไมโครคอนโทรลเลอร์สามารถทำได้โดยการกดปุ่มที่ได้ออกแบบไว้ โดยมีแถบ seekbar และปุ่ม ON/OFF แบบ Toggle ซึ่งการกดปุ่มดังกล่าวจะเป็นการส่งเลขฐานสิบไปยังไมโครคอนโทรลเลอร์ ซึ่งข้อมูลที่ส่งมีการตรวจสอบความผิดพลาดแบบ CRC-32 โดยไมโครคอนโทรลเลอร์จะนำเลขฐานสิบนั้นไปคำนวณเพื่อแปลงเป็นคำสั่งในการควบคุมเครื่องใช้ไฟฟ้าต่อไป โดย Lamp 1 ควบคุมหลอด incandescent, Lamp 2 ควบคุมหลอด fluorescent, Lamp 3 ควบคุมโคมไฟ, Fan 1 ควบคุมพัดลมดูดอากาศ และ Fan 2 ควบคุมพัดลม สำหรับส่วนติดต่อผู้ใช้งานแสดงดังรูปที่ 3.4



รูปที่ 3.4 ส่วนติดต่อผู้ใช้งาน

3.1.3 การรับคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้าของไมโครคอนโทรลเลอร์

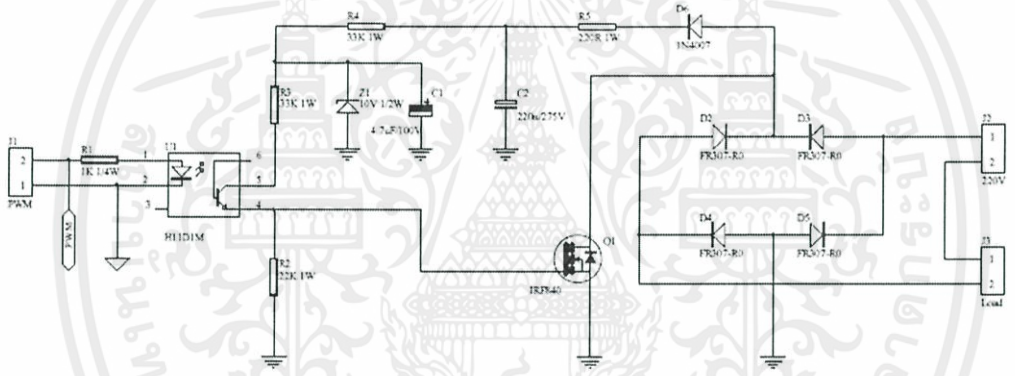
จะรับคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้าทั้งหมด 5 ชิ้นด้วยกัน โดยการรับคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้าในแต่ละชิ้นนั้น เริ่มต้นจากการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดพอร์ตในการรับ-ส่งข้อมูลของไมโครคอนโทรลเลอร์ ซึ่งในที่นี้กำหนดให้ดิจิทัลพอร์ต 6 เป็นพอร์ตรับข้อมูล (Rx) และดิจิทัลพอร์ต 7 เป็นพอร์ตส่งข้อมูล (Tx) โดยไมโครคอนโทรลเลอร์จะรับข้อมูลจากแอนดรอยด์ผ่านบลูทูธโมดูล การควบคุมเครื่องใช้ไฟฟ้าจะสั่งการผ่านดิจิทัลพอร์ตและอนาล็อกพอร์ตดังนี้

ดิจิทัลพอร์ต 11 ควบคุมอุปกรณ์ไฟฟ้าเครื่องที่ 1
ดิจิทัลพอร์ต 10 ควบคุมอุปกรณ์ไฟฟ้าเครื่องที่ 2
ดิจิทัลพอร์ต 12 ควบคุมอุปกรณ์ไฟฟ้าเครื่องที่ 3
อนาล็อกพอร์ต 1 ควบคุมอุปกรณ์ไฟฟ้าเครื่องที่ 4
อนาล็อกพอร์ต 2 ควบคุมอุปกรณ์ไฟฟ้าเครื่องที่ 5

เครื่องใช้ไฟฟ้าเครื่องที่ 1 จะสามารถปรับระดับไฟได้ โดยควบคุมผ่านดิจิทัลพอร์ต 11 โดยควบคุมระดับไฟด้วยการปรับระดับไฟแบบ PWM ดังรูปที่ 3.5



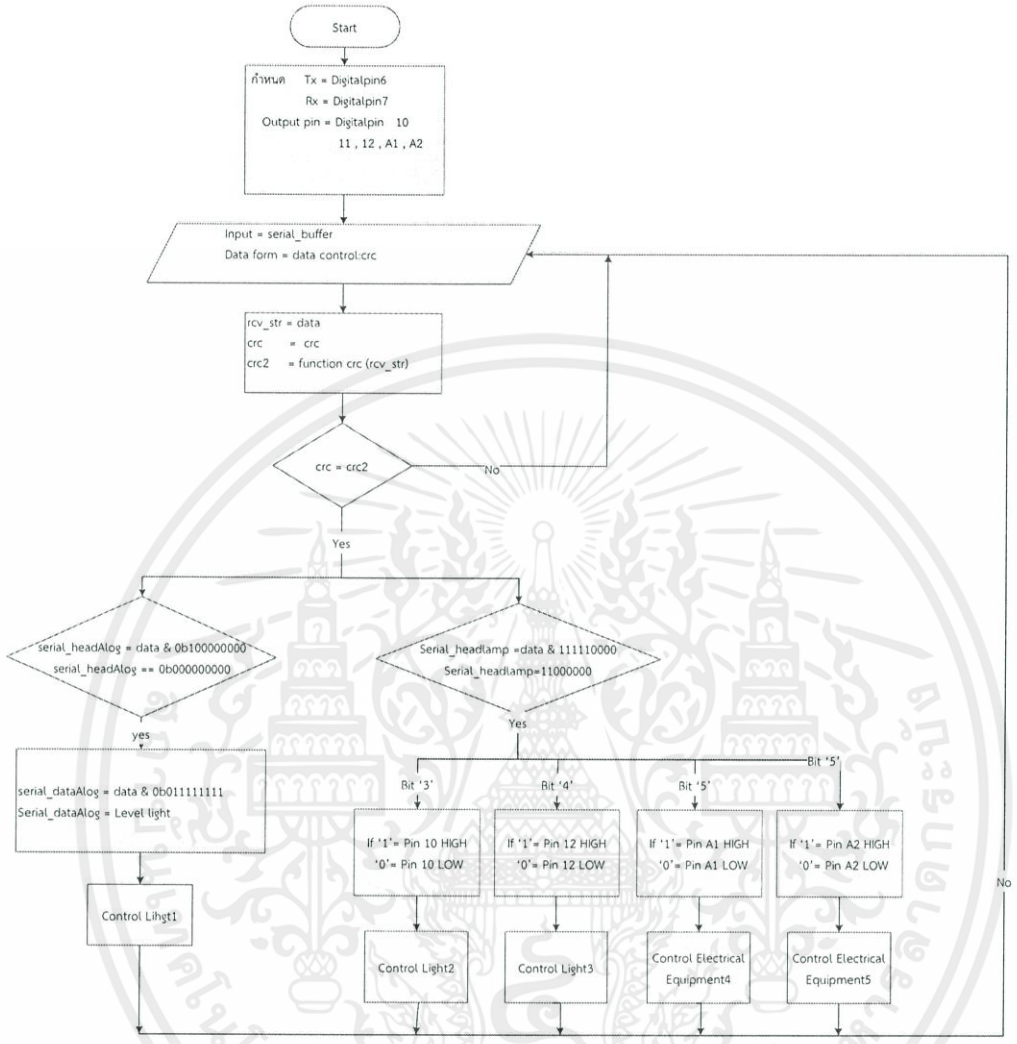
รูปที่ 3.5 วงจรปรับระดับไฟโดยใช้สัญญาณ PWM ของเครื่องใช้ไฟฟ้าเครื่องที่ 1 [19]

โดยข้อมูลที่รับได้จากบลูทูธโมดูลนั้นเป็นเลขฐานสิบ แต่ไมโครคอนโทรลเลอร์จะมองเป็นตัวเลขฐานสอง 9 บิต และเก็บไว้ในตัวแปร ‘serial_buffer’ จากนั้นนำมาประมวลผลดังนี้

ถ้าข้อมูลที่รับเข้ามาบิตแรกเป็น ‘0’ จะหมายถึงการควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าเครื่องที่ 1 โดย 8 บิตหลังจะเป็นบิตกำหนดระดับความสว่าง โดยมีระดับตั้งแต่ 0 ถึง 255

ถ้าข้อมูลที่รับเข้ามา 5 บิตแรกเป็น ‘10001’ จะหมายถึงการควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าที่เหลืออีกสี่เครื่อง โดย 4 บิตหลังจะเป็นบิตกำหนดสถานะเปิด/ปิดของเครื่องใช้ไฟฟ้า โดยถ้าได้รับบิต ‘1’ ไมโครคอนโทรลเลอร์จะจ่ายไฟให้พอร์ตซึ่งจะทำให้วงจรรีเลย์ทำงาน ส่งผลให้กระแสไฟไหลทำให้เครื่องใช้ไฟฟ้าเปิด และถ้าได้รับบิต ‘0’ ไมโครคอนโทรลเลอร์จะไม่จ่ายไฟให้พอร์ตซึ่งจะทำให้วงจรรีเลย์ไม่ทำงาน ส่งผลให้กระแสไฟไม่ไหลทำให้เครื่องใช้ไฟฟ้าปิด โดยโฟลว์ชาร์ตของการทำงานแสดงดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

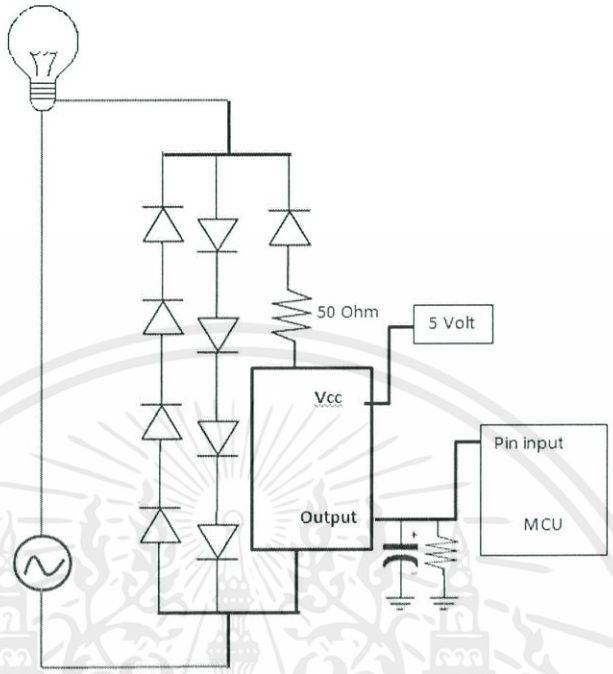


รูปที่ 3.6 โฟลวชาร์ตการสั่งงานเครื่องใช้ไฟฟ้า

3.1.4 การส่งค่าสถานะของเครื่องใช้ไฟฟ้ากลับไปยังแอนดรอยด์

การตรวจสอบสถานะการทำงานของเครื่องใช้ไฟฟ้า สามารถทำได้โดยการตรวจสอบจากกระแสไฟฟ้าที่ไหลผ่านเครื่องใช้ไฟฟ้า ซึ่งตรวจสอบโดยใช้วงจร ring detector โดยมีหลักการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



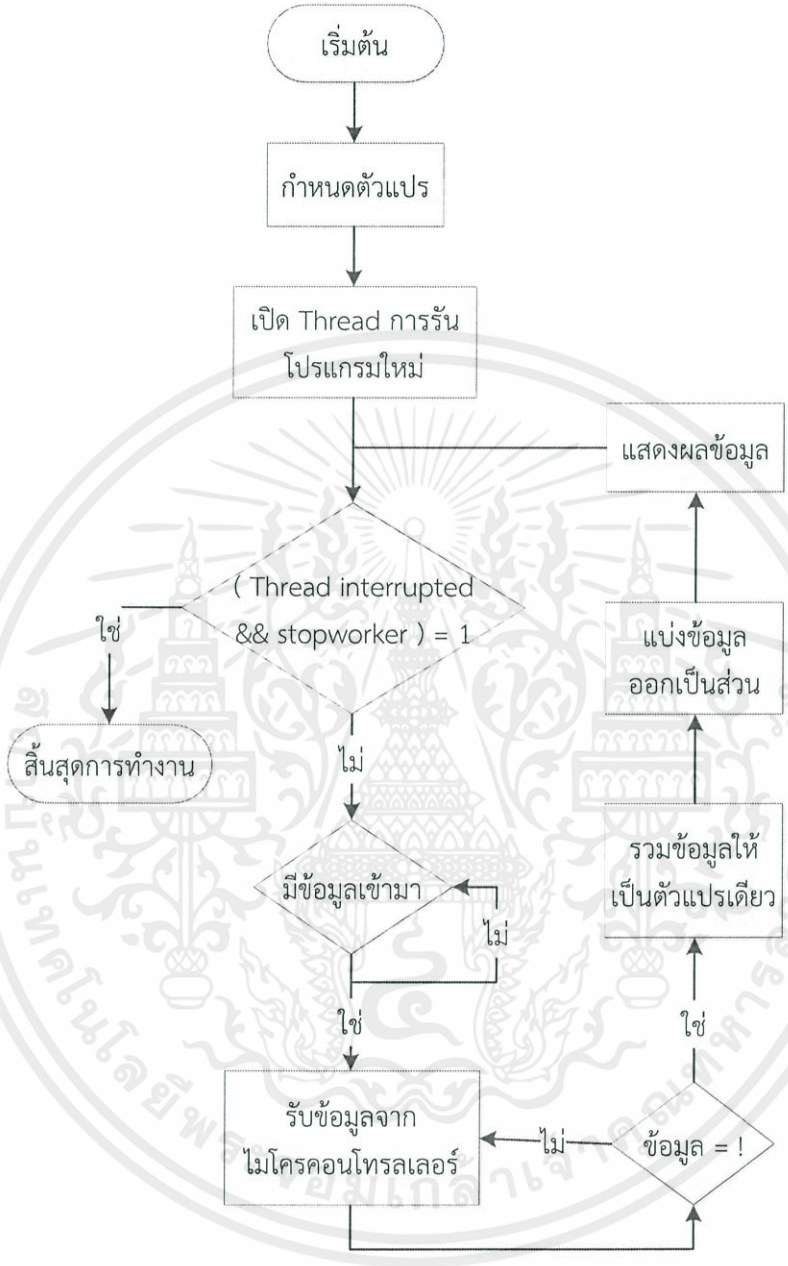
รูปที่ 3.7 การทำงานของวงจร ring detector

จะใช้การทำงานของ IC ชื่อ PC817 ซึ่งประกอบด้วยด้านส่งและด้านรับ โดยทางด้านส่งประกอบด้วยไดโอดเปล่งแสง ซึ่งจะเปล่งแสงเมื่อมีกระแสไฟฟ้าไหลผ่านวงจร และทางด้านรับประกอบด้วยโฟโต้ทรานซิสเตอร์ซึ่งได้จ่ายแรงดันไฟ 5 Volt ไว้ให้ เมื่อได้รับแสงจากไดโอดจะส่งผลให้แรงดันไฟสามารถไหลผ่านความต้านทานและตัวเก็บประจุ ซึ่งทำหน้าที่ในการควบคุมแรงดันให้คงที่ก่อนเข้าพอร์ตดิจิทัลอินพุตของไมโครคอนโทรลเลอร์เพื่อตรวจสอบสถานะของเครื่องใช้ไฟฟ้าดังรูปที่ 3.7

3.1.5 การรับค่าสถานะจากไมโครคอนโทรลเลอร์

การรับค่าสถานะจากไมโครคอนโทรลเลอร์ จะทำงานหลังจากที่เปิดแอพพลิเคชั่นแล้ว ซึ่งได้เขียนโปรแกรมให้เปิด Thread การรันโปรแกรมแยกออกมาต่างหาก โดยจะทำการรับข้อมูลจากไมโครคอนโทรลเลอร์ทีละไบต์ มาเก็บในตัวแปรแบบอาร์เรย์ จนกระทั่งเจอตัวแปร delimeter (ซึ่งในที่นี้กำหนดเป็น !) ก็จะทำการรวมตัวแปรอาร์เรย์ที่รับเข้ามาก่อนหน้านี้ ให้เป็นตัวแปรตัวเดียว จากนั้นจะทำการแยกข้อมูลออกเป็นส่วนๆ และนำมาแสดงผลออกทางส่วนติดต่อผู้ใช้งาน ถ้าหากเกิดข้อผิดพลาด Thread จะหยุดทำงาน โดยโฟลว์ชาร์ตของการทำงานแสดงดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 โฟลว์ชาร์ตการรับค่าสถานะจากไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 โทรศัพท์มือถือ

ต้องมีระบบเอ็นเอฟซี ในที่นี้ ใช้ยี่ห้อซัมซุงแกแลคซี่ เอสสาม ดังแสดงในรูปที่ 3.9

รูปที่ 3.9 โทรศัพท์มือถือยี่ห้อซัมซุงแกแลคซี่ เอสสาม [12]

3.2.2 ไมโครคอนโทรลเลอร์

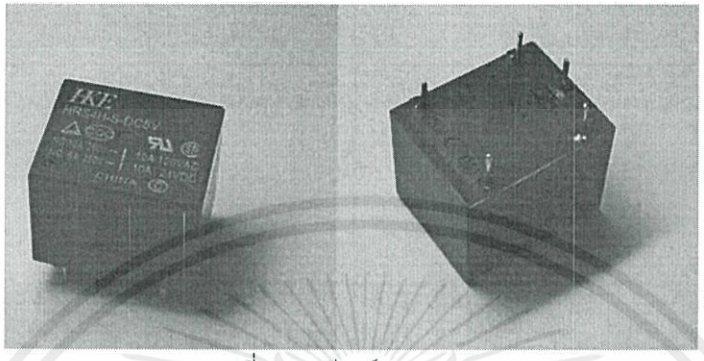
ATmega168 เป็นวงจรที่มีการสื่อสารแบบอนุกรม ทำงานที่แรงดันไฟ 1.8 - 5.5 โวลต์ โดยมีลักษณะดังรูปที่ 3.10

รูปที่ 3.10 ไมโครคอนโทรลเลอร์ : ATmega 168 [13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 รีเลย์

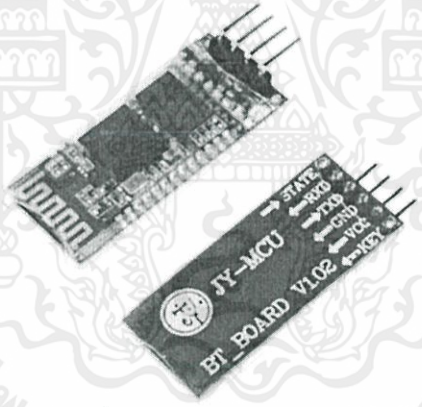
ใช้รีเลย์ 5 ขา ขนาด 5 V โดยมีลักษณะดังรูปที่ 3.11



รูปที่ 3.11 รีเลย์ 5 ขา [14]

3.2.4 บลูทูธโมดูล

เป็นอุปกรณ์รับ-ส่งสัญญาณข้อมูลด้วยเทคโนโลยีบลูทูธ ทำงานที่แรงดันไฟ 3.5-8 V โดยมีลักษณะดังรูปที่ 3.12



รูปที่ 3.12 บลูทูธโมดูล [15]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 โปรแกรม eclipse

เป็น platform เขียนโปรแกรมแอนดรอยด์ โดยมีสัญลักษณ์ดังรูปที่ 3.13

รูปที่ 3.13 สัญลักษณ์ของโปรแกรม Eclipse [16]

3.2.6 NFC tag

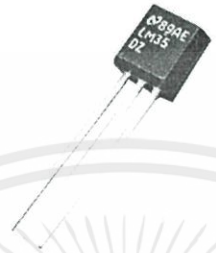
ใช้บัตรไมแฟร์ คลาสสิก (mifare classic) ความจุ 1 k ดังแสดงในรูปที่ 3.14

รูปที่ 3.14 บัตรไมแฟร์ คลาสสิก ความจุ 1 k [17]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 เซ็นเซอร์วัดอุณหภูมิ LM35

เป็นเซ็นเซอร์วัดอุณหภูมิแบบอนาล็อก แรงดันไฟฟ้าปฏิบัติการ: DC 4-30V ทำงานในช่วง -55 ถึง +150 องศาเซลเซียส ดังรูป 3.15



รูปที่ 3.15 เซ็นเซอร์วัดอุณหภูมิ LM35 [18]

3.3 การจัดเก็บผลการทดลอง

แบ่งเป็น 5 ขั้นตอนดังนี้

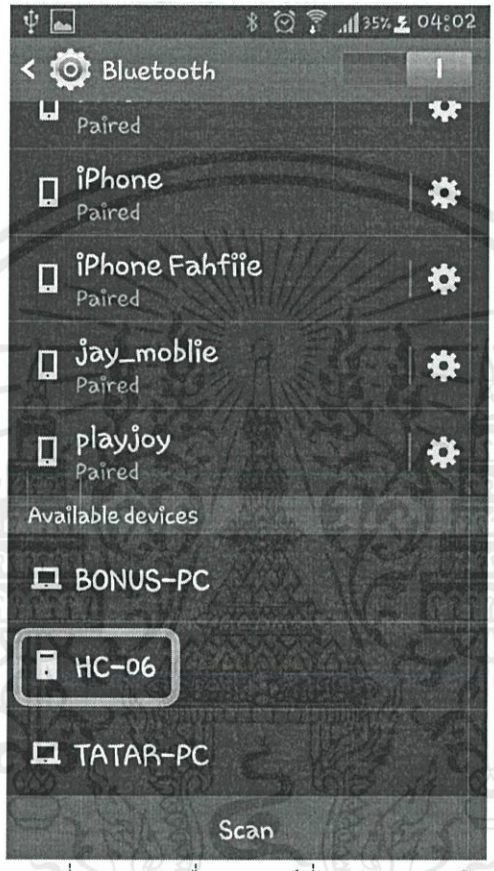
3.3.1 การเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์

ใช้โปรแกรม eclipse เขียนแอพพลิเคชั่นในการเปิดบลูทูธและเปิดแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้าเมื่อแตะ tag ซึ่งในแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้าได้ทำการเขียนโปรแกรมในการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ด้วยเทคโนโลยีบลูทูธ โดยตรวจสอบผลการทำงานของโปรแกรมผ่าน Logcat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การส่งคำสั่งไปยังไมโครคอนโทรลเลอร์

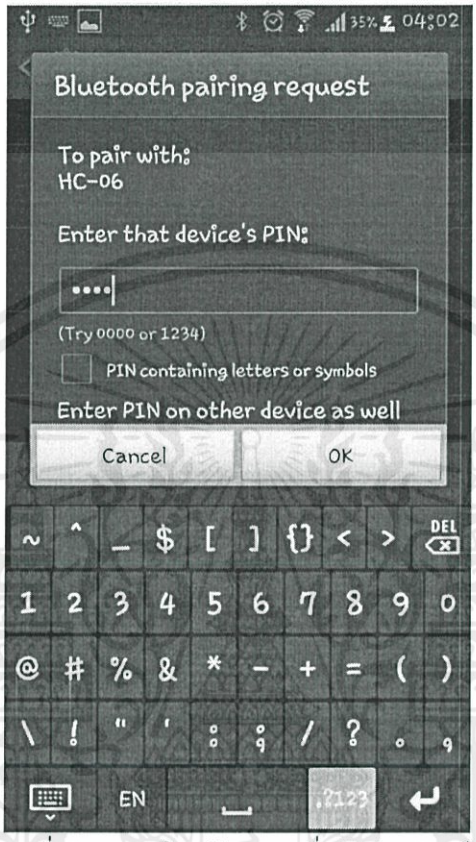
ต้องทำการจับคู่โทรศัพท์มือถือกับบลูทูธโมดูลก่อนโดยชื่อของบลูทูธโมดูลคือ HC-06 ดังรูปที่ 3.16



รูปที่ 3.16 รายชื่ออุปกรณ์ที่จะทำการจับคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นต้องทำการใส่รหัส PIN เพื่อจับคู่กับบลูทูธซึ่งในที่นี้คือ 1234 ดังรูปที่ 3.17



รูปที่ 3.17 การใส่รหัส PIN เพื่อจับคู่อุปกรณ์

โดยการส่งคำสั่งไปยังไมโครคอนโทรลเลอร์ทำได้โดยกดปุ่มที่ได้ออกแบบไว้ เพื่อส่งข้อมูลไปยังไมโครคอนโทรลเลอร์และตรวจสอบการส่งข้อมูลผ่าน Serial monitor

3.3.3 การรับคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้าของไมโครคอนโทรลเลอร์

เขียนโปรแกรมสั่งไมโครคอนโทรลเลอร์ ให้สามารถรับข้อมูลผ่านบลูทูธโมดูลกัดพริ้นท์บอร์ดอาร์ดูโน่ ทดสอบบอร์ดอาร์ดูโน่ ทดสอบไมโครคอนโทรลเลอร์ ทดสอบการรับข้อมูลและการสั่งงานเครื่องใช้ไฟฟ้า

3.3.4 การส่งค่าสถานะของเครื่องใช้ไฟฟ้ากลับไปยังแอนดรอยด์

เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ส่งค่าสถานะของเครื่องใช้ไฟฟ้ากลับไปยังแอนดรอยด์ในรูปแบบดังนี้ (อุณหภูมิ,เครื่องใช้ไฟฟ้า2,เครื่องใช้ไฟฟ้า3,เครื่องใช้ไฟฟ้า4,เครื่องใช้ไฟฟ้า5) เช่น (28,OFF,ON,ON,OFF)

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 การรับค่าสถานะจากไมโครคอนโทรลเลอร์

ใช้โปรแกรม eclipse เขียนแอพพลิเคชั่นในการรับข้อมูลจากไมโครคอนโทรลเลอร์ด้วยเทคโนโลยีบลูทูธ โดยตรวจสอบผลการทำงานของโปรแกรมทางส่วนติดต่อผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

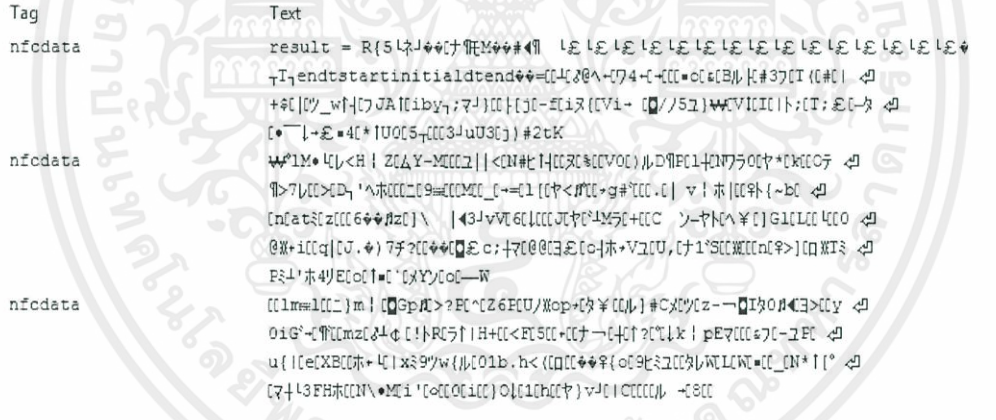
ผลการทดลอง

ในส่วนของการเก็บผลการทดลองนั้น จะทำการทดสอบการเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์ (การอ่านข้อมูลจาก tag และการเชื่อมต่อบลูทูธของแอนดรอยด์), การส่งคำสั่งไปยังไมโครคอนโทรลเลอร์, การสั่งการเครื่องใช้ไฟฟ้าของไมโครคอนโทรลเลอร์ (การทดสอบไมโครคอนโทรลเลอร์และอุปกรณ์ไฟฟ้าต่างๆ) และการรับและส่งค่าสถานะของเครื่องใช้ไฟฟ้าจากไมโครคอนโทรลเลอร์ไปยังแอนดรอยด์

4.1 ผลการทดสอบการเข้าสู่สถานะพร้อมใช้งานของแอนดรอยด์

4.1.1 ผลการทดสอบการอ่านข้อมูลจาก NFC tag

การทดสอบเริ่มจากการเขียนโปรแกรมให้แอนดรอยด์มีความสามารถในการอ่านข้อมูลจาก NFC tag จากนั้นทำการรันโปรแกรมและนำโทรศัพท์ไปแตะกับ NFC tag โดยข้อมูลที่อ่านได้แสดงผลออกทาง Logcat ดังรูปที่ 4.1



รูปที่ 4.1 ข้อมูลทั้งหมดที่อ่านได้จาก NFC tag

จากนั้นจึงนำข้อมูลที่ได้มาตัดเอาเฉพาะข้อมูลที่ต้องการ โดยแสดงผลการตัดข้อมูลครั้งแรกและครั้งที่สองออกทาง Logcat ดังรูปที่ 4.2 และ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

nfcdata first split = initialdtend

nfcdata

nfcdata

รูปที่ 4.2 ข้อมูลที่ได้จากการตัดข้อมูลครั้งแรก

nfcdata second split = initial

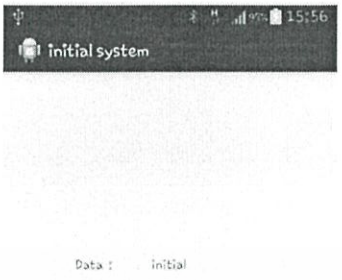
รูปที่ 4.3 ข้อมูลที่ได้จากการตัดข้อมูลครั้งที่สอง

จากนั้นจึงเก็บผลการทดลองจากส่วนติดต่อผู้ใช้งาน โดยเมื่อทำการรันโปรแกรม ส่วนติดต่อผู้ใช้งานจะแสดงดังรูปที่ 4.4

รูปที่ 4.4 ส่วนติดต่อผู้ใช้งานก่อนการแตะ NFC tag

จากนั้นจึงนำโทรศัพท์ไปแตะกับ NFC tag และให้แสดงข้อมูลที่ได้ออกทางส่วนติดต่อผู้ใช้งาน ดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

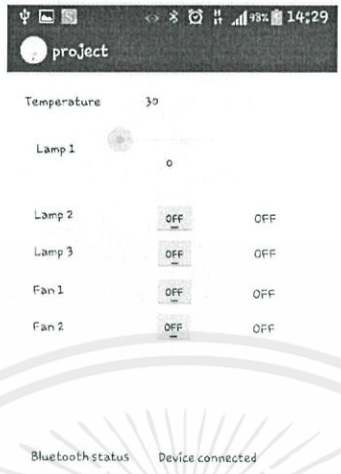


รูปที่ 4.5 ข้อมูลที่ได้จาก NFC tag

เมื่อได้รับข้อมูลที่เป็นคำว่า initial โปรแกรมจะทำการร้องขอให้มีการเปิดการใช้งานบลูทูธ และทำการเปิดแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้า ดังรูปที่ 4.6 และ 4.7

รูปที่ 4.6 การร้องขอให้มีการเปิดใช้งานบลูทูธของแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ส่วนติดต่อผู้ใช้งานของแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้า

4.1.2 ผลการทดสอบการเชื่อมต่อบลูทูธของแอนดรอยด์

การทดสอบเริ่มจากเขียนโปรแกรมเช็คว่าอุปกรณ์ที่นำมารันแอพพลิเคชั่นนั้นสามารถใช้งานบลูทูธได้หรือไม่ โดยใช้คำสั่ง

```
BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if (mBluetoothAdapter == null) { // Device does not support Bluetooth}
```

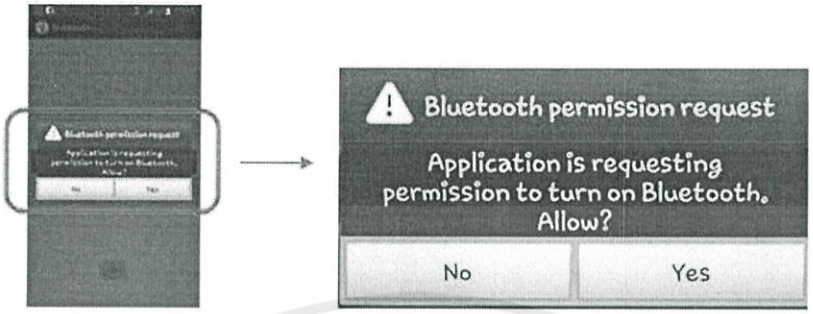
โดยถ้าอุปกรณ์ไม่สามารถใช้งานบลูทูธได้ ตัวแปร mBluetoothAdapter จะเท่ากับ null (ค่าว่าง)

จากนั้นเขียนโปรแกรมตรวจสอบว่าอุปกรณ์ทำการเปิดบลูทูธอยู่หรือไม่ โดยใช้คำสั่ง

```
if (!mBluetoothAdapter.isEnabled()) {
Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(enableBtIntent, 1);}
```

ถ้าหากไม่จะขึ้นหน้าต่างเตือนให้เปิดบลูทูธ ดังรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าต่างแจ้งเตือนให้เปิดบลูทูธ

ก่อนที่จะทำการเชื่อมต่อกับอุปกรณ์ที่ต้องการ ต้องทำการเรียกรายชื่อของชุดอุปกรณ์ที่จับคู่แล้ว โดยใช้คำสั่ง getBondedDevices() ดังนี้

```
Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
// If there are paired devices
if (pairedDevices.size() > 0) {
   // Loop through paired devices
   for (BluetoothDevice device : pairedDevices) {
       // Add the name and address to an array adapter to show in a ListView
      // mArrayAdapter.add(device.getName() + "\n" + device.getAddress());
        String name = device.getName();
        Log.v("LogTxt", name);
```

ซึ่งจะได้รายชื่อของอุปกรณ์ที่จับคู่แล้วทั้งหมด โดยจะเก็บรายชื่อไว้ในตัวแปรชื่อ name โดยตรวจสอบรายชื่อที่ได้จากการแสดงรายชื่อออกทาง Logcat ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tag	Text
LogTxt	iPhone
LogTxt	SeeedBTSlave
LogTxt	THENENN-PC
LogTxt	GT-P7300B
LogTxt	jay_moblie
LogTxt	KKAN's iPhone
LogTxt	Galaxy S4
LogTxt	playjoy
LogTxt	GT-N7000
LogTxt	S. Pokawanwit's MBPro
LogTxt	buus
LogTxt	iPhone Fahfiie
LogTxt	GT-N7100
LogTxt	BONUZiii;)
LogTxt	iPhone
LogTxt	HC-06

รูปที่ 4.9 รายชื่อของอุปกรณ์ที่จับคู่แล้ว

จากนั้นเขียนคำสั่ง if(name.equals("HC-06")) ซึ่งหมายความว่าจะทำคำสั่งนี้เมื่อตัวแปร name มีค่าเท่ากับ HC-06 (ชื่อบลูทูธโมดูล) และทำการเขียนคำสั่งเชื่อมต่อระหว่าง Android และบลูทูธโมดูล โดยการสร้างและเชื่อมต่อ Socket ระหว่างกัน ดังนี้

```
if(name.equals("HC-06"))
{
    //Connect
    final BluetoothSocket mmSocket;
    final BluetoothDevice mmDevice;
    final UUID MY_UUID = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");
    BluetoothSocket tmp = null;
    mmDevice = device;
    try {
       // MY_UUID is the app's UUID string, also used by the server code
       tmp = device.createRfcommSocketToServiceRecord(MY_UUID);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
} catch (IOException e) { }
mmSocket = tmp;
 try {
 // Connect the device through the socket. This will block
// until it succeeds or throws an exception
mmSocket.connect();
Log.v("LogTxt", "Connect!");
//#### Connect!!!!!!!!!!!!!!!!
//Start manage socket
```

โดยตรวจสอบการเชื่อมต่อด้วยการแสดงข้อความ “Connect !” ออกทาง Logcat ดังรูปที่ 4.10

Tag	Text
LogTxt	KKAN's iPhone
LogTxt	Galaxy S4
LogTxt	playjoy
LogTxt	GT-N7000
LogTxt	S. Pokawanwit's MBPro
LogTxt	buus
LogTxt	iPhone Fahfiie
LogTxt	GT-N7100
LogTxt	BONUZiii;)
LogTxt	iPhone
LogTxt	HC-06
LogTxt	Connect!
LogTxt	Can create stream

รูปที่ 4.10 ข้อความ Connect ! แสดงเมื่อมีการเชื่อมต่อ Socket

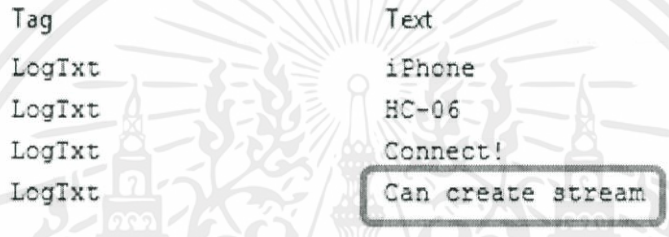
จากนั้นเขียนคำสั่งสร้าง Stream เพื่อเป็นช่องทางการรับและส่งข้อมูล หากทำการสร้าง Stream สำเร็จดังนี้

```
try
{
        tmpIn = mmSocket.getInputStream();
        tmpOut = mmSocket.getOutputStream();
```

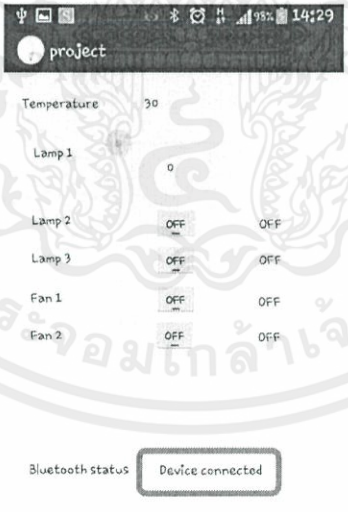
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Log.v("LogTxt", "Can create stream");
Toast.makeText(getApplicationContext(), "Device connected",
Toast.LENGTH_LONG).show();
```

และกำหนดให้แสดงข้อความ “Can create stream” ออกทาง Logcat ดังรูปที่ 4.11 ซึ่งส่วนติดต่อผู้ใช้จะแสดงข้อความ “Device connected” ดังรูปที่ 4.12 แต่ถ้าหากไม่สามารถสร้าง Stream ได้จะแสดงข้อความ “Device cannot connect” ดังรูปที่ 4.13

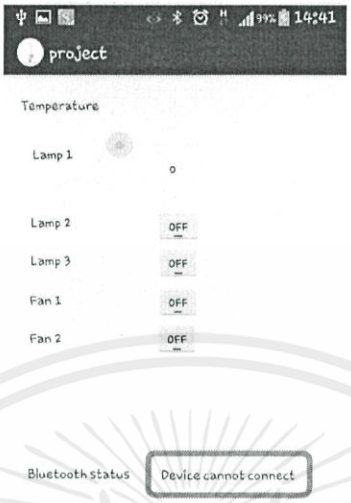


รูปที่ 4.11 ข้อความ Can create stream แสดงเมื่อมีการสร้าง Stream



รูปที่ 4.12 ข้อความ Device connected แสดงเมื่อมีการสร้าง Stream

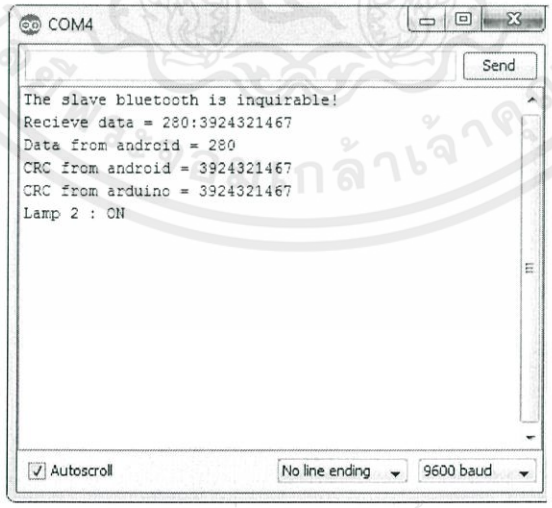
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ข้อความ Device cannot connect แสดงเมื่อมีการไม่มีสร้าง Stream

4.2 ผลการทดสอบการส่งคำสั่งไปยังไมโครคอนโทรลเลอร์

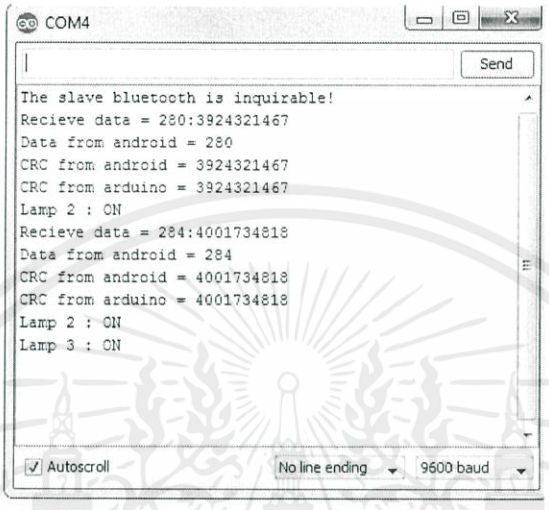
ผลการทดสอบสามารถตรวจสอบได้จาก Serial monitor โดยการส่งคำสั่งเริ่มจากการกดปุ่ม ON ของหลอดไฟดวงที่ 2 ซึ่งจะเป็นการส่งเลขฐานสิบ 280 (100011000 ในเลขฐานสอง) ซึ่งมีการตรวจสอบความผิดพลาดแบบ CRC-32 (ในที่นี้คือ 3924321467 ในเลขฐานสิบ) ไปยังไมโครคอนโทรลเลอร์ ดังรูปที่ 4.14



รูปที่ 4.14 คำสั่งที่ไมโครคอนโทรลเลอร์รับได้เมื่อกดปุ่ม ON ของหลอดไฟดวงที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการกดปุ่ม ON ของหลอดไฟดวงที่ 3 ซึ่งจะเป็นการส่งเลขฐานสิบ 284 (100011100 ในเลขฐานสอง) ซึ่งมีการตรวจสอบความผิดพลาดแบบ CRC-32 (ในที่นี้คือ 4001734818 ในเลขฐานสิบ) ไปยังไมโครคอนโทรลเลอร์ ดังรูปที่ 4.15

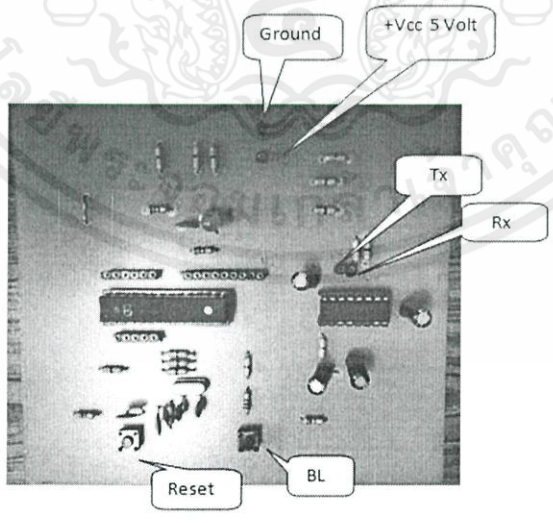


รูปที่ 4.15 คำสั่งที่ไมโครคอนโทรลเลอร์รับได้เมื่อกดปุ่ม ON ของหลอดไฟดวงที่ 3

4.3 ผลการทดสอบการรับคำสั่งเพื่อควบคุมเครื่องใช้ไฟฟ้า

4.3.1 ผลการทดสอบการทำงานของบอร์ดอาร์ดูโน่

ปริญญานิพนธ์นี้ได้ทำการกัดพริ้นท์บอร์ดอาร์ดูโน่ขึ้นมา ดังรูปที่ 4.16



รูปที่ 4.16 บอร์ดอาร์ดูโน่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการทดสอบบอร์ดอาร์ดูโน่ทำได้ดังนี้

1.กดสวิตช์ BL ค้าง

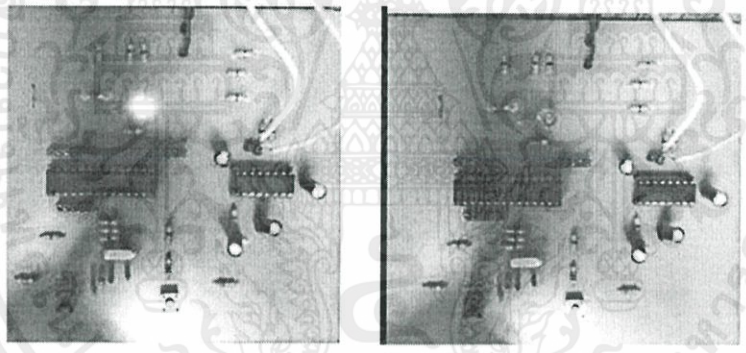
2.กดสวิตช์ reset แล้วปล่อย

3. ปล่อยสวิตช์ BL

จะเห็น LED กระพริบสามครั้งจากนั้นจะติดค้างแสดงถึงบอร์ดสามารถใช้งานได้

4.3.2 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ ATmega 168

ทำได้โดยการเบิร์นโปรแกรม Blink ซึ่งเป็นโปรแกรมที่ทำให้ output Digital pin 13 เปิดและปิดสลับกัน ส่งผลให้หลอด LED ซึ่งเชื่อม กับ output Digital pin 13 ติดและดับสลับกัน ดังรูปที่ 4.17 จากการทดลองปรากฏว่า LED ติดและดับสลับกัน แสดงให้เห็นว่าไมโครคอนโทรลเลอร์สามารถใช้งานได้



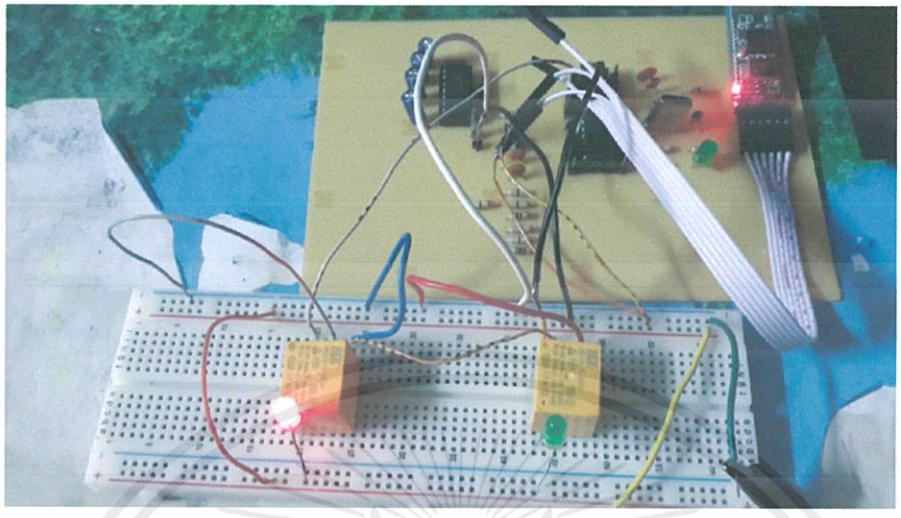
รูปที่ 4.17 ผลการทดสอบโปรแกรม blink

4.3.3 ผลการทดสอบการทำงานของวงจรรีเลย์

ทำการต่อไมโครคอลโทรลเลอร์เพื่อสั่งงานวงจรรีเลย์สองตัวผ่าน output Digital pin 3 และ output Digital pin 4 โดยทำการต่อ LED ให้กับรีเลย์แต่ละตัว ซึ่ง LED สีแดงต่อกับรีเลย์พอร์ต 3 และ LED สีเขียวต่อกับรีเลย์พอร์ต 4 ผลการทดสอบเป็นดังนี้

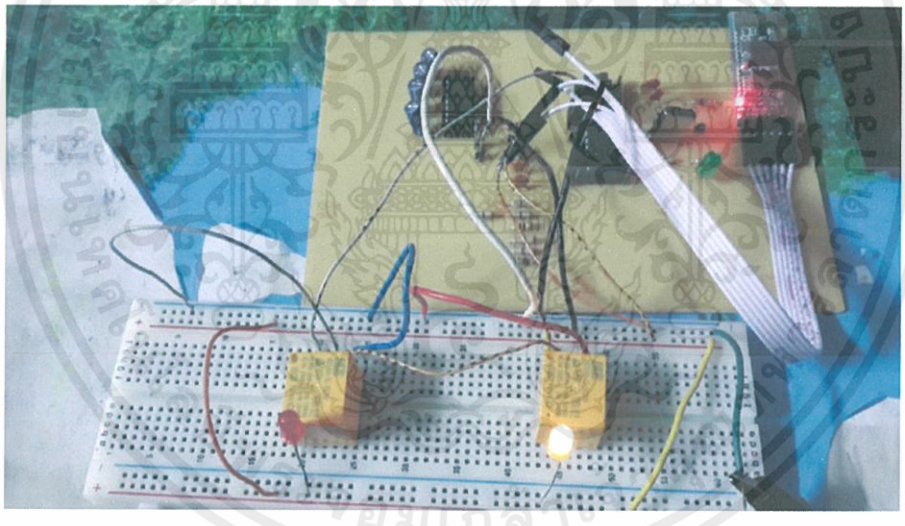
เมื่อทำการส่งเลข 1 ไมโครคอลโทรลเลอร์จะจ่ายไฟให้พอร์ต 3 ซึ่งทำให้ LED สีแดงสว่าง ดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 ผลการทดสอบการสั่งการรีเลย์พอร์ต 3

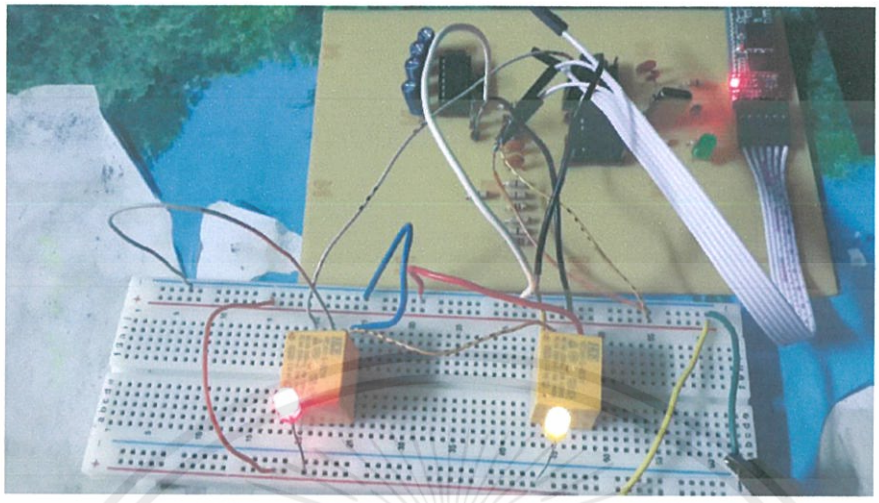
เมื่อทำการส่งเลข 2 ไมโครคอลโทรลเลอร์จะจ่ายไฟให้พอร์ต 4 ทำให้ LED สีเขียวสว่าง ดังรูปที่ 4.19



รูปที่ 4.19 ผลการทดสอบการสั่งการรีเลย์พอร์ต 4

เมื่อทำการส่งเลข 3 ไมโครคอลโทรลเลอร์จะจ่ายไฟให้พอร์ต 3 และ 4 ทำให้ LED ทั้งสีแดงและสีเขียวสว่าง ดังรูปที่ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 ผลการทดสอบการสั่งการรีเลย์พอร์ต 3 และ 4

เมื่อทำการส่งเลข 0 ไมโครคอลโทรลเลอร์จะไม่จ่ายไฟให้พอร์ต 3 และ 4 ทำให้ LED ทั้งสีแดงและสีเขียวดับ

จากผลการทดลองทำให้ทราบว่ารีเลย์สามารถใช้งานได้และไมโครคอนโทรลเลอร์สามารถรับคำสั่งและสั่งการได้

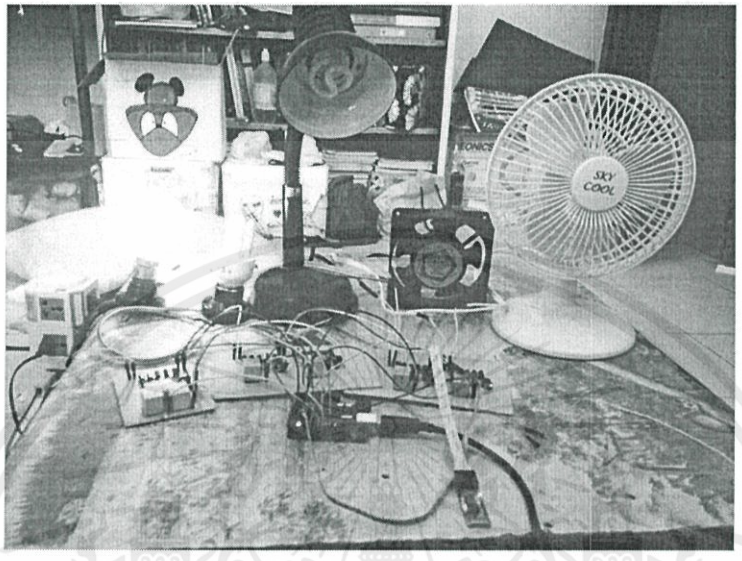
4.3.4 ผลการทดสอบการสั่งการเครื่องใช้ไฟฟ้า

การทดสอบเริ่มจากการกดปุ่มสั่งการเครื่องใช้ไฟฟ้าจากโทรศัพท์แอนดรอยด์ โดยเครื่องใช้ไฟฟ้ามีทั้งหมด 5 อุปกรณ์ดังรูปที่ 4.21

รูปที่ 4.21 เครื่องใช้ไฟฟ้าทั้ง 5 อุปกรณ์

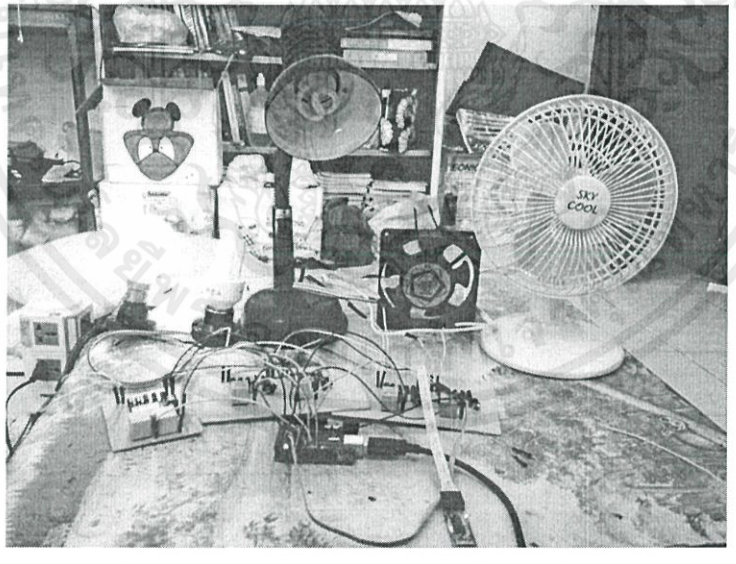
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลื่อน seekbar ของหลอดไฟดวงที่ 1 (หลอด incandescent) ไปที่ 50% จะทำให้หลอดไฟติด (แสงไฟอ่อน) ดังรูปที่ 4.22



รูปที่ 4.22 หลอดไฟ incandescent ติด (แสงไฟอ่อน)

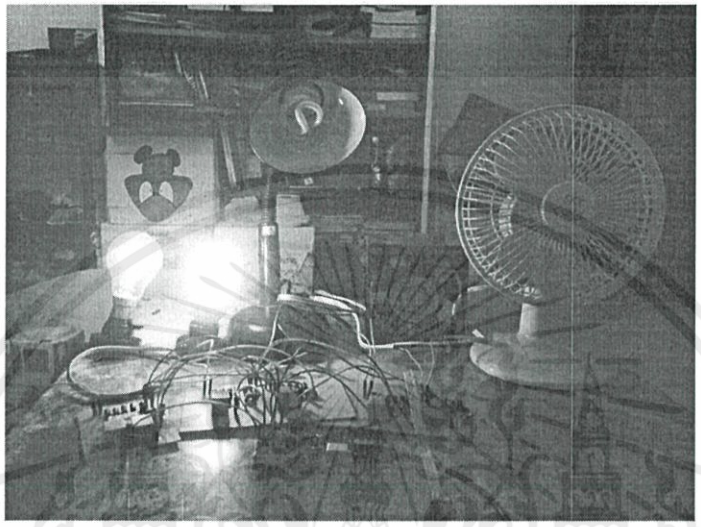
เมื่อเลื่อน seekbar ของหลอดไฟดวงที่ 1 (หลอด incandescent) ไปที่ 100% จะทำให้หลอดไฟติด (แสงไฟสว่าง) ดังรูปที่ 4.23



รูปที่ 4.23 หลอดไฟ incandescent ติด (แสงไฟสว่าง)

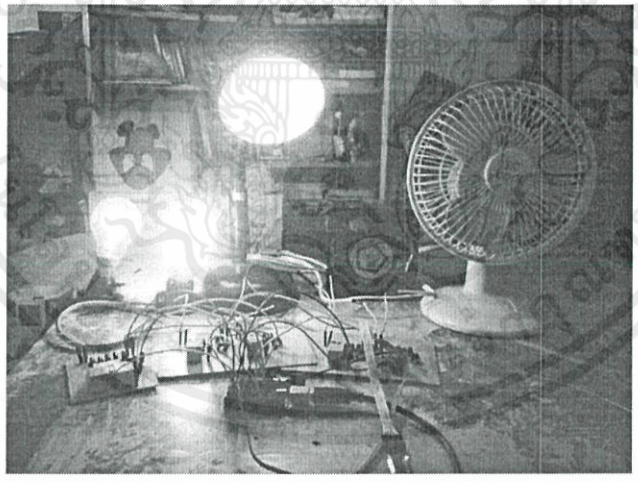
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกดปุ่ม “ON” ของหลอดไฟดวงที่ 2 (หลอด fluorescent) จะทำให้หลอดไฟติด ดังรูปที่ 4.24



รูปที่ 4.24 หลอดไฟ fluorescent ติด

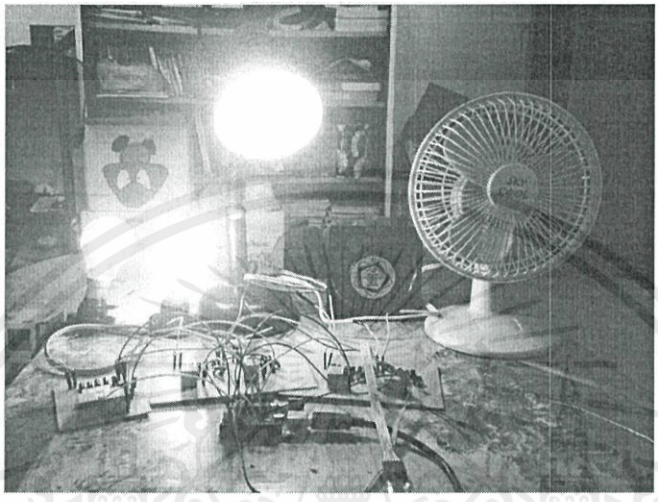
เมื่อกดปุ่ม “ON” ของหลอดไฟดวงที่ 3 (โคมไฟ) จะทำให้หลอดไฟติด ดังรูปที่ 4.25



รูปที่ 4.25 โคมไฟติด

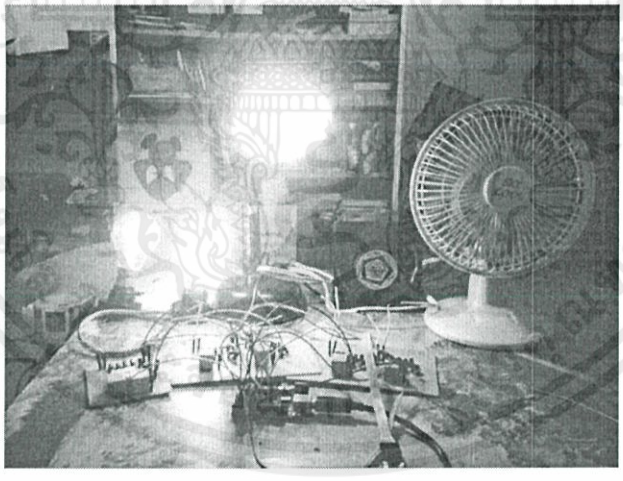
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกดปุ่ม “ON” ของพัดลมตัวที่ 1 (พัดลมดูดอากาศ) จะทำให้พัดลมทำงาน ดังรูปที่ 4.26



รูปที่ 4.26 พัดลมดูดอากาศทำงาน

เมื่อกดปุ่ม “ON” ของพัดลมตัวที่ 2 จะทำให้พัดลมทำงาน ดังรูปที่ 4.27

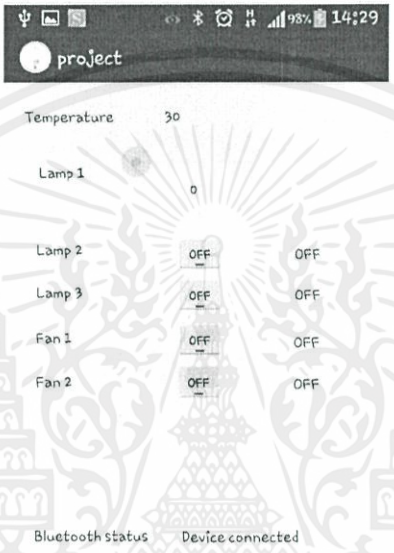


รูปที่ 4.27 พัดลมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การรับและส่งค่าสถานะของเครื่องใช้ไฟฟ้าจากไมโครคอนโทรลเลอร์ไปยังแอนดรอยด์

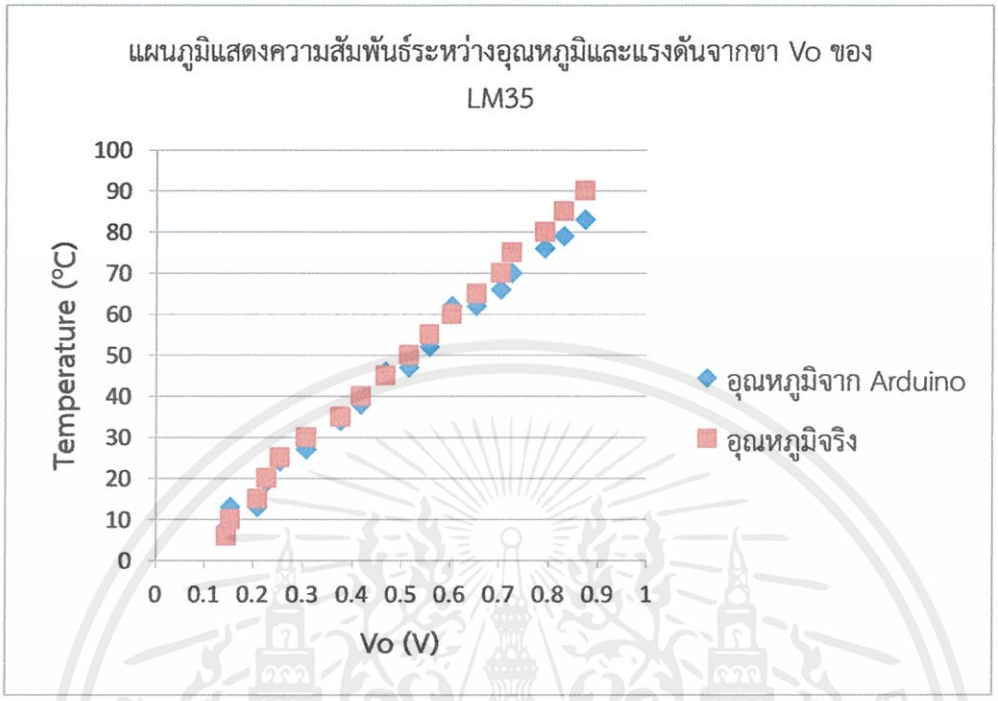
การทดสอบเริ่มจากการเปิดแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้า โดยเริ่มจากการไม่เปิดเครื่องใช้ไฟฟ้าชิ้นใดเลย ซึ่งเครื่องใช้ไฟฟ้าจะมีสถานะเป็น OFF ทุกชิ้น ดังรูปที่ 4.28



รูปที่ 4.28 สถานะของเครื่องใช้ไฟฟ้า

โดยส่วนติดต่อผู้ใช้งานจะแสดงอุณหภูมิที่ได้จากไมโครคอนโทรลเลอร์ ซึ่งผลการทดสอบความสัมพันธ์ของอุณหภูมิที่ได้จากไมโครคอนโทรลเลอร์และอุณหภูมิจริงเทียบกับแรงดันที่ได้จาก LM35 แสดงดังรูปที่ 4.29

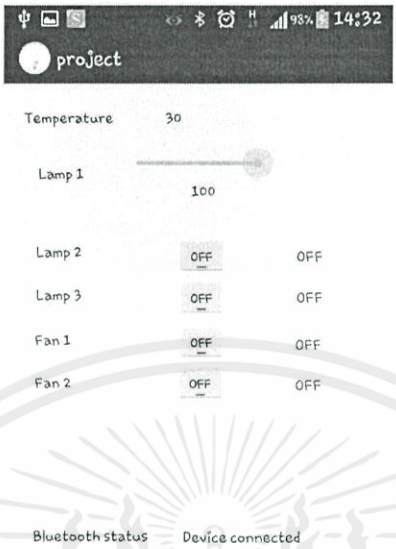
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



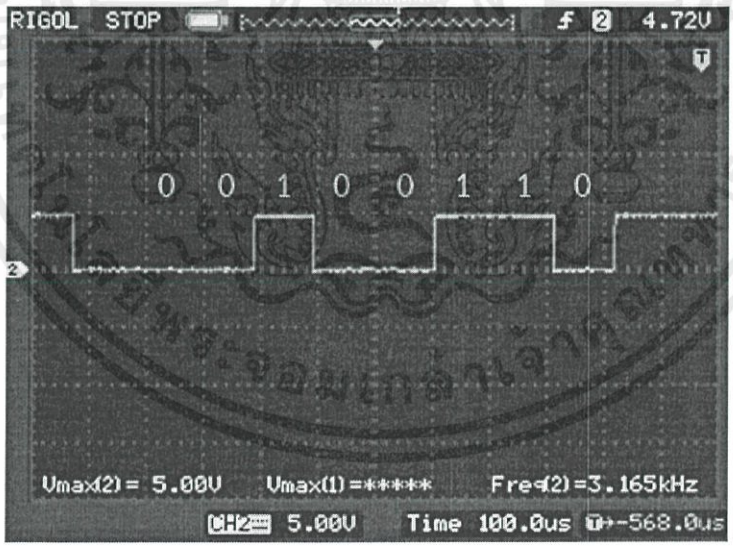
รูปที่ 4.29 แผนภูมิแสดงความสัมพันธ์ระหว่างอุณหภูมิและแรงดันจากขา Vo ของ LM35

จากนั้นจึงทำการเปิดเครื่องใช้ไฟฟ้าทีละเครื่องจนครบทุกเครื่อง โดยเมื่อกดปุ่มคำสั่งโทรศัพท์แอนดรอยด์จะทำการส่งข้อมูลคำสั่งไปยังไมโครคอนโทรลเลอร์ ตัวอย่างการควบคุมเครื่องใช้ไฟฟ้าแสดงดังรูปที่ 4.30, 4.32, 4.34, 4.36, 4.38, 4.40 โดยสถานะของเครื่องใช้ไฟฟ้าจะเปลี่ยนเป็น ON เมื่อทำการเปิดเครื่องใช้ไฟฟ้าชิ้นนั้นๆ ดังรูปที่ 4.31, 4.33, 4.35, 4.37, 4.39, 4.41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

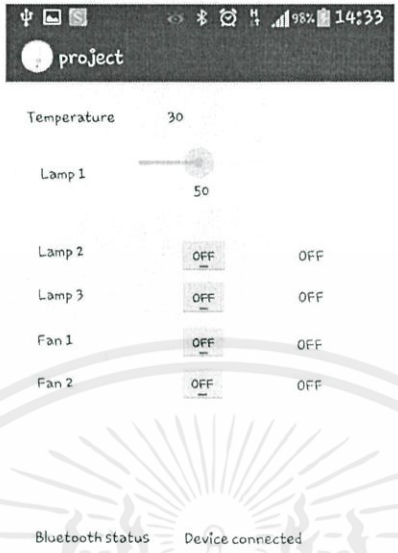


รูปที่ 4.30 สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 1

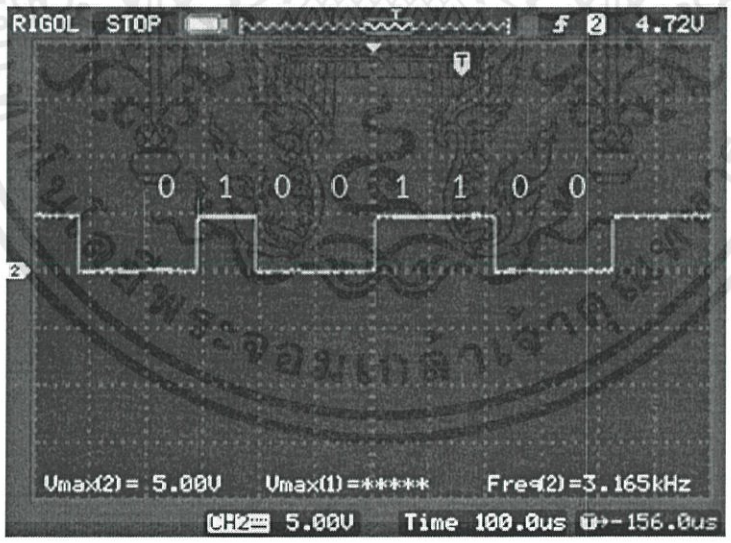


รูปที่ 4.31 ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

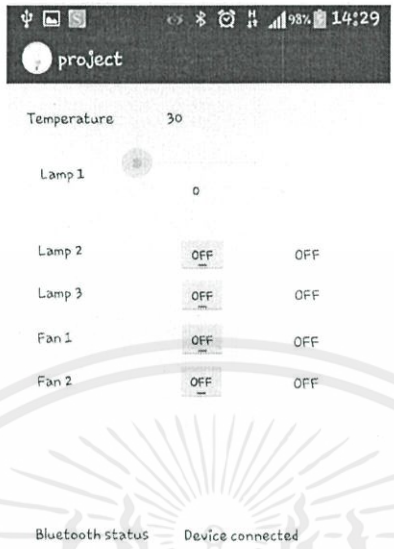


รูปที่ 4.32 สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 2

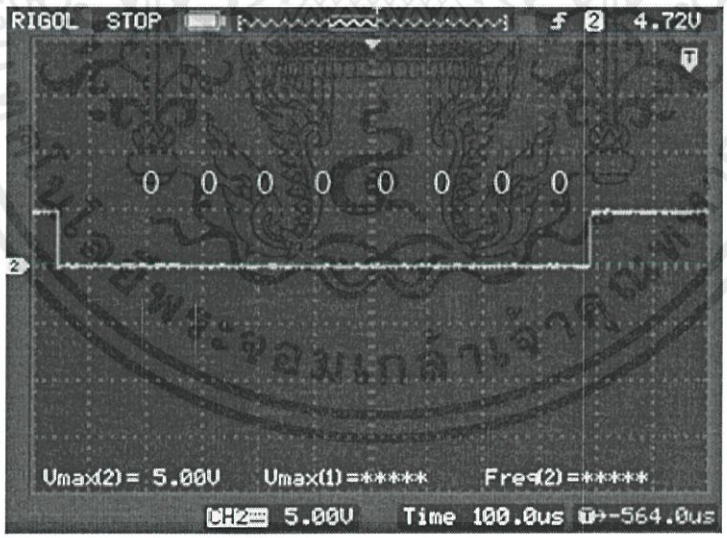


รูปที่ 4.33 ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

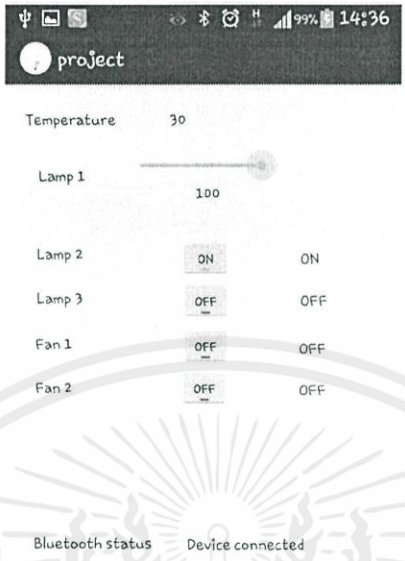


รูปที่ 4.34 สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 3

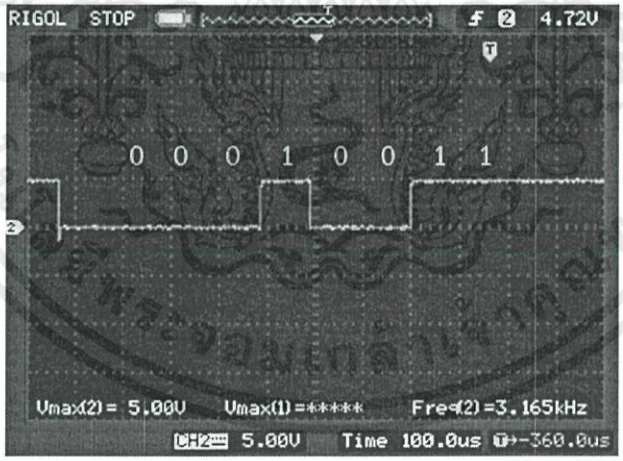


รูปที่ 4.35 ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

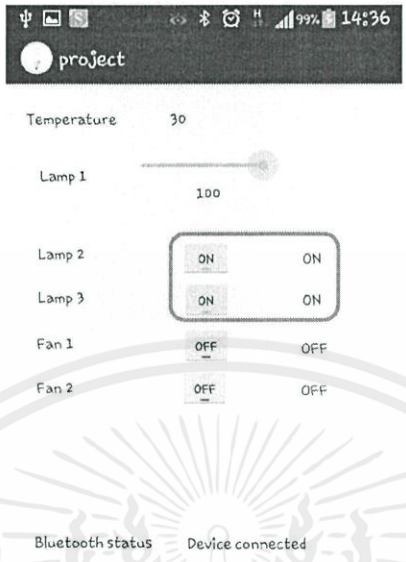


รูปที่ 4.36 สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 4

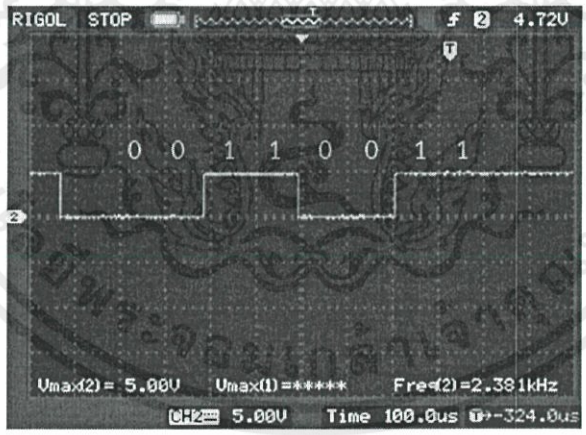


รูปที่ 4.37 ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

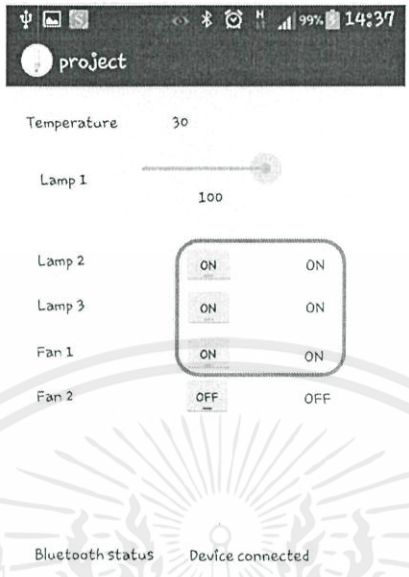


รูปที่ 4.38 สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 5

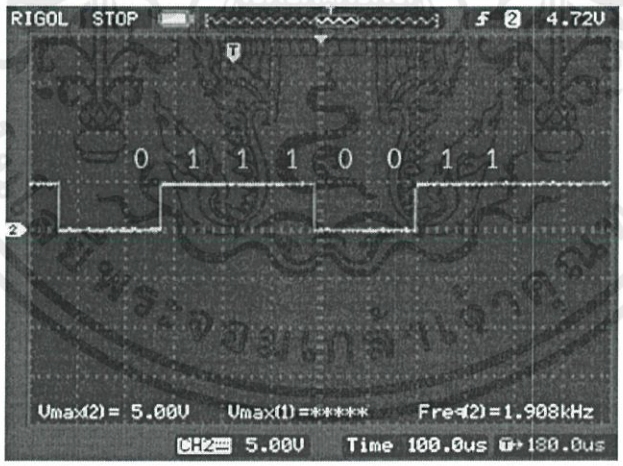


รูปที่ 4.39 ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

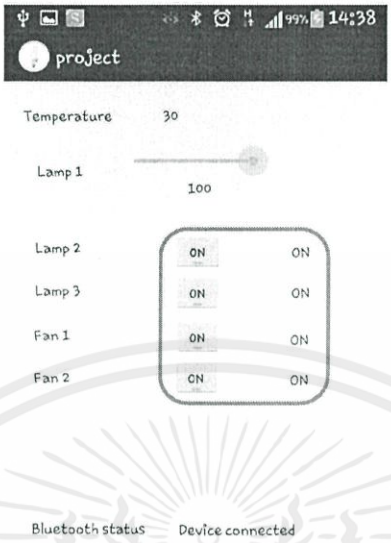


รูปที่ 4.40 สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 6

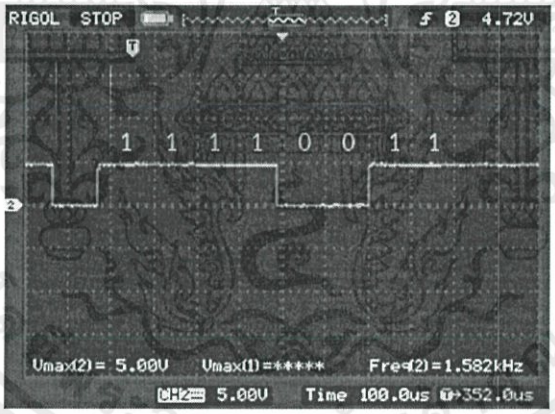


รูปที่ 4.41 ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



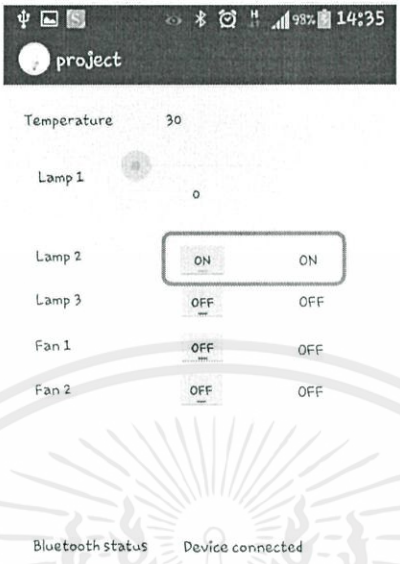
รูปที่ 4.42 สถานะของเครื่องใช้ไฟฟ้าตัวอย่างที่ 7



รูปที่ 4.43 ข้อมูลคำสั่งที่โทรศัพท์แอนดรอยด์ส่งไปยังไมโครคอนโทรลเลอร์ของตัวอย่างที่ 7

จากนั้นจึงทำการทดสอบสถานะของเครื่องใช้ไฟฟ้าที่มีการใช้งานก่อนการเปิดแอพพลิเคชั่น โดยการทดสอบเริ่มจากการเปิดเครื่องใช้ไฟฟ้าทิ้งไว้แล้วปิดแอพพลิเคชั่น ซึ่งในที่นี้เป็นหลอดไฟดวงที่ 2 โดยก่อนปิดแอพพลิเคชั่นหลอดไฟดวงที่ 2 มีสถานะดังรูปที่ 4.44

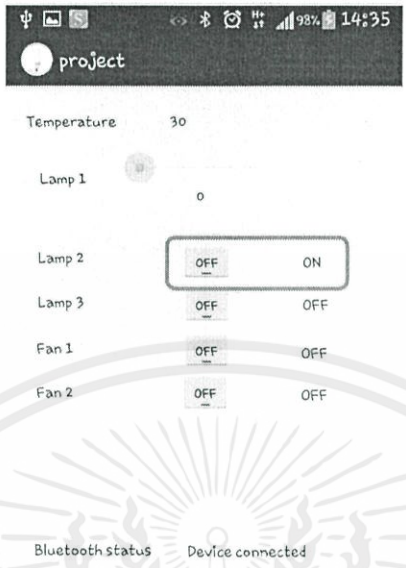
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.44 สถานะของหลอดไฟดวงที่ 2 ก่อนปิดแอพพลิเคชั่น

จากนั้นจึงทำการเปิดแอพพลิเคชั่นใหม่อีกครั้ง จะพบว่าแม้ยังไม่ได้สั่งการเปิดหลอดไฟดวงที่ 2 แต่สถานะของหลอดไฟดวงที่ 2 ยังคงเป็น ON ดังรูปที่ 4.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.45 สถานะของหลอดไฟดวงที่ 2 หลังเปิดแอพพลิเคชั่นใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์เรื่องระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ด้วยเทคโนโลยี NFC และบลูทูธ นำเสนอระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบแอนดรอยด์ โดยทำการเขียนแอพพลิเคชั่นให้มีความสามารถในการเปิด/ปิดและปรับระดับเครื่องใช้ไฟฟ้า, มีความสามารถในการเชื่อมต่อระบบบลูทูธ และมีความสามารถในการส่งคำสั่งควบคุมการทำงานผ่านระบบบลูทูธไปยังไมโครคอนโทรลเลอร์ซึ่งได้ทำการตรวจสอบความผิดพลาดแบบ CRC ด้วย โดยไมโครคอนโทรลเลอร์จะทำหน้าที่ควบคุมเครื่องใช้ไฟฟ้าที่ต้องการ ซึ่งผลการทดลองของฝั่งส่งข้อมูล (โทรศัพท์แอนดรอยด์) สามารถทำงานได้ตามเป้าหมายตั้งแต่การอ่านข้อมูลจาก NFC tag, การเชื่อมต่อกับบลูทูธโมดูล, การรับและส่งคำสั่งไปยังไมโครคอนโทรลเลอร์ ส่วนฝั่งรับข้อมูล (ไมโครคอนโทรลเลอร์) และอุปกรณ์ไฟฟ้าต่างๆ ก็สามารถทำงานได้ตามความต้องการของผู้ใช้ ตั้งแต่การรับและส่งคำสั่งไปยังแอนดรอยด์, การทดสอบการทำงานของอุปกรณ์ต่างๆ และการทดสอบการสั่งการเครื่องใช้ไฟฟ้า โดยแอพพลิเคชั่นในการควบคุมเครื่องใช้ไฟฟ้ามีความสามารถดังนี้

1) แสดงสถานะการเชื่อมต่อกับระบบบลูทูธ
2) ควบคุมหลอดไฟให้สามารถปรับระดับความสว่างได้
3) ควบคุมเครื่องใช้ไฟฟ้าได้ 5 อุปกรณ์
4) แสดงสถานะการทำงานของเครื่องใช้ไฟฟ้า
5) แสดงอุณหภูมิในหน่วยองศาเซลเซียส
6) เปิดระบบการทำงานด้วย NFC tag

แต่การเปิดระบบการทำงานด้วย NFC tag นั้น ยังทำงานไม่สมบูรณ์ โดยแรกเริ่มได้ออกแบบให้โทรศัพท์แอนดรอยด์ที่ไม่ต้องเปิดแอพพลิเคชั่นใดๆ เมื่อนำไปแตะกับ NFC tag ก็จะสามารถเปิดระบบบลูทูธและเปิดแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้าโดยอัตโนมัติ แต่เนื่องจากฟังก์ชั่นของ NFC เช่น การรับค่า intents ของ tag ยังไม่สามารถทำงานเป็น Service ได้ จึงทำให้ต้องเปิดแอพพลิเคชั่นการเปิดระบบการทำงานก่อนการแตะ NFC tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

1) สามารถนำโครงงานนี้ไปประยุกต์ใช้งานกับเครื่องใช้ไฟฟ้าชนิดอื่นๆได้

2) สามารถนำโครงงานไปประยุกต์ให้มีคุณภาพและประสิทธิภาพยิ่งๆขึ้นไป เช่น สามารถใช้กับเครื่องใช้ไฟฟ้าที่มีกำลังไฟฟ้าสูง เช่น เครื่องปรับอากาศ ตู้เย็น ได้

3) สามารถนำอุปกรณ์เพิ่มความปลอดภัยทางไฟฟ้ามาประยุกต์ใช้กับวงจรได้ เช่น ฟิวส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] เจน สงสมพันธุ์. *เทคโนโลยีอิเล็กทรอนิกส์ 1.* พิมพ์ครั้งที่ 17. กรุงเทพฯ : สถาบันอิเล็กทรอนิกส์กรุงเทพรังสิต, 2541.

[2] อภิเชษฐ์ การัยภูมิ. *อิเล็กทรอนิกส์อย่างง่าย.* กรุงเทพฯ : นานมีบุ๊คส์, 2544.

[3] กิตติ ภักดีวัฒนะกุล. *คัมภีร์ Java เล่ม 1.* พิมพ์ครั้งที่ 3. กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2546.

[4] ไพบูลย์ สวัสดิ์ปัญญาโชติ. *Android app : the Android developer's cookbook.* พิมพ์ครั้งที่ 2. กรุงเทพฯ : ทรู ดิจิตอล คอนเท้นท์ แอนด์ มีเดีย, 2554.

[5] "Arduino Uno"
http://arduino.cc/en/Main/arduinoBoardUno
(สืบค้นวันที่ 29 กรกฎาคม 2556)

[6] "อุปกรณ์ ตอน รีเลย์"
http://www.semi-shop.com/knowledge/knowledge_detail.php?sk_id=28
(สืบค้นวันที่ 29 กรกฎาคม 2556)

[7] "รู้จักกับรีเลย์ (Relay)"
http://www.smartlearningweb.com/knowledge/relay/relay.htm
(สืบค้นวันที่ 29 กรกฎาคม 2556)

[8] "การใช้ประโยชน์ของ NFC"
http://www.kmitl.ac.th/~ktnarin/nfc.pdf
(สืบค้นวันที่ 10 มกราคม 2557)

[9] "รู้จักกับเทคโนโลยี Contactless Payment"
http://www.siamintelligence.com/nfc-mobile-payment/
(สืบค้นวันที่ 10 มกราคม 2557)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[10] “เทคโนโลยีการสื่อสารข้อมูลยุคใหม่”
http://drivers-area.blogspot.com/2012_08_01_archive.html
(สืบค้นวันที่ 10 มกราคม 2557)

[11] “หลักการทำงานแบบ PWM (Pulse width modulation)”
http://www.thaicpf.com/webboard/index.php?topic=68.0
(สืบค้นวันที่ 12 กุมภาพันธ์ 2557)

[12] “Samsung Galaxy”
http://www.techinfo2.com/samsung-galaxy-grand-duos-5-inch-dual-sim-phone.html
(สืบค้นวันที่ 29 กรกฎาคม 2556)

[13] ““Ren” AVR”
http://beta.knowitlabs.no/elektronikk-for-programmorerprogrammoser-del-1/
(สืบค้นวันที่ 29 กรกฎาคม 2556)

[14] “รีเลย์ 5V 5ขา HKE”
http://www.oocities.org/toyga55/de_relay_5V.htm
(สืบค้นวันที่ 29 กรกฎาคม 2556)

[15] “Bluetooth Serial Module (HC-06 Slave mode)”
http://www.arduino.in.th/product/6/bluetooth-serial-module-hc-06-slave-mode
(สืบค้นวันที่ 10 สิงหาคม 2556)

[16] “Android development tips - Eclipse crash on deploy”
http://kevinpelgrims.com/blog/2012/05/24/android-development-tips-eclipse-crash-on-deploy
(สืบค้นวันที่ 29 กรกฎาคม 2556)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[17] "Smart Card Mifare Classic 1K White PVC Cards"
http://www.eyenetwatch.com/Products/216-smart-card-mifare-classic-1k-white-pvc-cards.aspx
(สืบค้นวันที่ 22 มกราคม 2557)

[18] "LM35 Temperature Sensor"
http://www.rapidsignalph.com/shop/lm35-temperature-sensor/
(สืบค้นวันที่ 22 มกราคม 2557)

[19] "Home Control กับระบบหรี่ไฟ"
http://www.openfog.net/homectrl_doc_p6.html
(สืบค้นวันที่ 12 กุมภาพันธ์ 2557)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โค้ดของโทรศัพท์แอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้า

```
package com.example.project;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.charset.Charset;
import java.util.Set;
import java.util.UUID;
import java.util.zip.CRC32;
import java.util.zip.Checksum;

import android.R.string;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;
import android.widget.SeekBar.OnSeekBarChangeListener;
public class MainActivity extends Activity {

	volatile boolean stopWorker;
	int readBufferPosition;
	byte[] readBuffer;
	Thread workerThread;

	SeekBar seekbar;
	TextView value;

	String temp;
	String lamp1;
	String lamp2;
	String lamp3;
	String lamp4;
	String lamp5;

	int a=0,b=0,c=0,d=0,e=0,f=0,g=0,h=0,i=0,result;
	int pass0,pass1,pass2,pass3;
	char[] dvpw = {' '};
	BluetoothSocket mmSocket;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BluetoothDevice mmDevice;

       InputStream tmpIn = null;
    OutputStream tmpOut = null;

    public String encrypt(String data){
       String str = data;
       byte bytes[] = str.getBytes();
        Checksum checksum = new CRC32();
        checksum.update(bytes,0,bytes.length);
        long lngChecksum = checksum.getValue();
        String crcvalue = Long.toString(lngChecksum);
        return crcvalue;
    }



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

              BluetoothAdapter mBluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
              if (mBluetoothAdapter == null) {
                 // Device does not support Bluetooth
              }

              if (!mBluetoothAdapter.isEnabled()) {
                 Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                 startActivityForResult(enableBtIntent, 1);
              }

              Set<BluetoothDevice> pairedDevices =
mBluetoothAdapter.getBondedDevices();
              // If there are paired devices
              if (pairedDevices.size() > 0) {
                 // Loop through paired devices
                 for (BluetoothDevice device : pairedDevices) {
                    // Add the name and address to an array adapter to
show in a ListView
                    String name = device.getName();
                    Log.v("LogTxt", name);

                    if(name.equals("HC-06"))
                    {
                           //Connect
                        final UUID MY_UUID = UUID.fromString("00001101-0000-
1000-8000-00805f9b34fb");

                        BluetoothSocket tmp=null;
                        mmDevice = device;

                        try {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        // MY_UUID is the app's UUID string, also used
by the server code
                        tmp =
device.createRfcommSocketToServiceRecord(MY_UUID);
                    } catch (IOException e) { }
                mmSocket = tmp;

                try {
                        // Connect the device through the socket. This
will block
                        // until it succeeds or throws an exception
                        mmSocket.connect();
                        Log.v("LogTxt", "Connect!");

                        //#### Connect!!!!!!!!!!!!!!!
                //Start manage socket
try
                        {

                            tmpIn = mmSocket.getInputStream();
                            tmpOut = mmSocket.getOutputStream();
                            Log.v("LogTxt", "Can create stream");

                            TextView btstatusbox =
(TextView)findViewById(R.id.btstatus);
                            btstatusbox.setText("Device connected");

                            //Try to send data by string
                            //sendString("3221:"+encrypt(3221));

                            //Try to receive data

                            beginListenForData();

                            //Try to send data

                            value = (TextView)
findViewById(R.id.valueofsb);
                                    seekbar = (SeekBar)
findViewById(R.id.seekBar1);

                                    seekbar.setOnSeekBarChangeListener(
new OnSeekBarChangeListener()
                                    {
                                            public void
onProgressChanged(SeekBar seekBar, int progress,boolean fromUser)
                                            {
                                                    // TODO Auto-generated
method stub
                                                    String sbvalue =
Integer.toString(progress);
                                                    value.setText(sbvalue);
                                                    int seekValue =
seekBar.getProgress();
                                                    int adj_seekValue =
(seekValue*255)/100 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

```
                                                    String str =
Integer.toString(adj_seekValue);
                                                    str +=
":"+encrypt(str)+"$";
                                                    byte[] buffer =
str.getBytes();
                                                    try {
                                                    tmpOut.write(buffer);
                                                    Log.v("hey", "Send
string \""+str+"\" via BT socket.");
                                                    } catch (IOException e)
{
                                                    // TODO Auto-generated
catch block
                                                    e.printStackTrace();
                                                    }
                                                }
                                                    public void
onStartTrackingTouch(SeekBar seekBar)
                                                    {
                                                    // TODO Auto-generated
method stub
                                                    }
                                                    public void
onStopTrackingTouch(SeekBar seekBar)
                                                    {
                                                    // TODO Auto-generated
method stub
                                                    }
                                            });
                                    ToggleButton toggle2 = (ToggleButton)
findViewById(R.id.button2);
                                    toggle2.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
                                    {
                                        public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked)
                                        {
                                            if (isChecked)
                                            {
                                                seta(1);
                                                setb(0);
                                                setc(0);
                                                setd(0);
                                                sete(1);
                                                setf(1);
                                                senddata();
                                            }
                                            else
                                            {
                                                seta(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    setb(0);
                                    setc(0);
                                    setd(0);
                                    sete(1);
                                    setf(0);
                                    senddata();
                                }
                            }
                        });

                        ToggleButton toggle3 = (ToggleButton)
findViewById(R.id.button3);
                        toggle3.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
                        {
                            public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked)
                            {
                                if (isChecked)
                                {
                                    seta(1);
                                    setb(0);
                                    setc(0);
                                    setd(0);
                                    sete(1);
                                    setg(1);
                                    senddata();
                                } else
                                {
                                    seta(1);
                                    setb(0);
                                    setc(0);
                                    setd(0);
                                    sete(1);
                                    setg(0);
                                    senddata();
                                }
                            }
                        });

                        ToggleButton toggle4 = (ToggleButton)
findViewById(R.id.button4);
                        toggle4.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
                        {
                            public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked)
                            {
                                if (isChecked)
                                {
                                    seta(1);
                                    setb(0);
                                    setc(0);
                                    setd(0);
                                    sete(1);
                                    seth(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        senddata();
                                    } else
                                    {
                                        seta(1);
                                        setb(0);
                                        setc(0);
                                        setd(0);
                                        sete(1);
                                        seth(0);
                                        senddata();
                                    }
                                }
                                });

                                ToggleButton toggle5 = (ToggleButton)
findViewById(R.id.button5);
                                toggle5.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
                                {
                                    public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked)
                                {
                                    if (isChecked)
                                    {
                                        seta(1);
                                        setb(0);
                                        setc(0);
                                        setd(0);
                                        sete(1);
                                        seti(1);
                                        senddata();
                                    } else
                                    {
                                        seta(1);
                                        setb(0);
                                        setc(0);
                                        setd(0);
                                        sete(1);
                                        seti(0);
                                        senddata();
                                    }
                                }
                            });
} catch (IOException e) { }


                        } catch (IOException connectException) {
                            // Unable to connect; close the socket and get
out
                            TextView btstatusbox =
(TextView)findViewById(R.id.btstatus);
                            btstatusbox.setText("Device cannot connect");
                            Log.v("LogTxt", "Cannot connect");
                            try {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                mmSocket.close();
                            } catch (IOException closeException) { }
                            return;
                        }

                    }
                }
            }

        }

        void beginListenForData()
        {

            final Handler handler = new Handler();
            final byte delimiter = 33;

            stopWorker = false;
            readBufferPosition = 0;
            readBuffer = new byte[1024];


            workerThread = new Thread(new Runnable()
            {
                public void run()
                {
                    while(!Thread.currentThread().isInterrupted() &&
!stopWorker)
                    {
                        try
                        {
                            int bytesAvailable = tmpIn.available();

                            if(bytesAvailable > 0)
                            {

                                byte[] packetBytes = new
byte[bytesAvailable];

                                tmpIn.read(packetBytes);
                                for(int i=0;i<bytesAvailable;i++)
                                {
                                    byte b = packetBytes[i];

                                    if(b == delimiter)
                                    {
                                        byte[] encodedBytes = new
byte[readBufferPosition];

                                        System.arraycopy(readBuffer, 0,
encodedBytes, 0, encodedBytes.length);

                                        final String data = new
String(encodedBytes, "US-ASCII");

                                        readBufferPosition = 0;

                                        String[] splitdata =
data.split(",");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    temp = splitdata[0];
                                lamp1 = onoff(splitdata[1]);
                                lamp2 = onoff(splitdata[2]);
                                lamp3 = onoff(splitdata[3]);
                                lamp4 = onoff(splitdata[4]);


                                    handler.post(new Runnable()
                                    {
                                        public void run()
                                        {
                                                        TextView textbox
= (TextView)findViewById(R.id.temp);

      textbox.setText(temp+" C");
                                                        TextView textbox1
= (TextView)findViewById(R.id.status2);

      textbox1.setText(lamp1);
                                                        TextView textbox2
= (TextView)findViewById(R.id.status3);

      textbox2.setText(lamp2);
                                                        TextView textbox3
= (TextView)findViewById(R.id.status4);

      textbox3.setText(lamp3);
                                                        TextView textbox4
= (TextView)findViewById(R.id.status5);

      textbox4.setText(lamp4);

                                        }
                                    });
                                }
                                else
                                {
                                    readBuffer[readBufferPosition++] = b;
                                }
                            }
                        }
                        catch (IOException ex)
                        {
                            stopWorker = true;
                        }
                    }
                }
            });

            workerThread.start();
        }
        String onoff(String splitdata)
        {
            String _splitdata="";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (splitdata.equals("1"))
        {
                _splitdata = "ON";
        }
        else if (splitdata.equals("0"))
        {
                _splitdata = "OFF";
        }
        return _splitdata;
}


void seta(int data)
{
        a = data ;
}

void setb(int data)
{
        b = data ;
}

void setc(int data)
{
        c = data ;
}

void setd(int data)
{
        d = data ;
}

void sete(int data)
{
        e = data ;
}

void setf(int data)
{
        f = data ;
}

void setg(int data)
{
        g = data ;
}

void seth(int data)
{
        h = data ;
}

void seti(int data)
{
        i = data ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	void senddata()
	{
		result =
(a*100000000)+(b*10000000)+(c*1000000)+(d*100000)+(e*10000)+(f*1000)+(g*100)
+(h*10)+i;
		int resultx = Integer.parseInt(String.valueOf(result), 2);
		String str = Integer.toString(resultx);
		str += ":"+encrypt(str)+"$";
		byte[] buffer = str.getBytes();
		try {
			tmpOut.write(buffer);
			Log.v("hey", "Send string \""+str+"\" via BT socket.");
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}

	void sendString(String str){
		str += "$";
		byte[] buffer = str.getBytes();
		try {
			tmpOut.write(buffer);
			Log.v("hey", "Send string \""+str+"\" via BT socket.");
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}

	void setpw0 (int password0)
	{
		seta(1);
		setb(0);
		setc(0);
		setd(1);
		sete(0);
		transet(password0);
		senddata();
	}
	void setpw1 (int password1)
	{
		seta(1);
		setb(0);
		setc(0);
		setd(1);
		sete(1);
		transet(password1);
		senddata();
	}
	void setpw2 (int password2)
	{
		seta(1);
		setb(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setc(1);
        setd(0);
        sete(0);
        transet(password2);
        senddata();
}
void setpw3 (int password3)
{
        seta(1);
        setb(0);
        setc(1);
        setd(0);
        sete(1);
        transet(password3);
        senddata();
}

void transet (int Rxpw)
{
        if (Rxpw == 1)
        {
                setf(0);
                setg(0);
                seth(0);
                seti(1);
        }
        else if (Rxpw == 2)
        {
                setf(0);
                setg(0);
                seth(1);
                seti(0);
        }
        else if (Rxpw == 3)
        {
                setf(0);
                setg(0);
                seth(1);
                seti(1);
        }
        else if (Rxpw == 4)
        {
                setf(0);
                setg(1);
                seth(0);
                seti(0);
        }
        else if (Rxpw == 5)
        {
                setf(0);
                setg(1);
                seth(0);
                seti(1);
        }
        else if (Rxpw == 6)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                setf(0);
                setg(1);
                seth(1);
                seti(0);
        }
        else if (Rxpw == 7)
        {
                setf(0);
                setg(1);
                seth(1);
                seti(1);
        }
        else if (Rxpw == 8)
        {
                setf(1);
                setg(0);
                seth(0);
                seti(0);
        }
        else if (Rxpw == 9)
        {
                setf(1);
                setg(0);
                seth(0);
                seti(1);
        }
        else if (Rxpw == 0)
        {
                setf(0);
                setg(0);
                seth(0);
                seti(0);
        }

}

void confirm(int cfpass)
{
        try {
                tmpOut.write(cfpass);
        } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
}

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดอ่าน NFC tag เพื่อเปิดระบบบลูทูธและแอพพลิเคชั่นควบคุมเครื่องใช้ไฟฟ้า

```
package com.example.initialsystem;

import java.io.IOException;

import android.nfc.NfcAdapter;
import android.nfc.Tag;
import android.nfc.tech.MifareClassic;
import android.os.Bundle;
import android.app.Activity;
import android.app.PendingIntent;
import android.bluetooth.BluetoothAdapter;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.IntentFilter.MalformedMimeTypeException;
import android.util.Log;
import android.view.Menu;
import android.widget.TextView;

public class MainActivity extends Activity {
NfcAdapter NFC;
PendingIntent pIntent;
IntentFilter filter;
IntentFilter[] filters;
String[][] namelist;
BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

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

        try{
                NFC = NfcAdapter.getDefaultAdapter(this);
                pIntent = PendingIntent.getActivity(this, 0, new Intent(this,
getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP), 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                filter = new IntentFilter(NfcAdapter.ACTION_TECH_DISCOVERED);

                try{      filter.addDataType("*//*");       }
                catch (MalformedMimeTypeException e) {
                        throw new RuntimeException("fail", e);
                }

                filters = new IntentFilter[] {
                                filter,
                };

                namelist = new String[][] { new String[] { MifareClassic.class.getName() }
};
        }catch(Exception e){ Log.v("msg", "Err : "+e); }
}

@Override
public void onResume()
        {
        super.onResume();
        try{
                NFC.enableForegroundDispatch(this, pIntent, filters, namelist);
        }catch(Exception e){ Log.v("msg", "Err : "+e); }
}

@Override
public void onNewIntent(Intent intent){
        readNfcData(intent);

}

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return true;
}
public void readNfcData(Intent intent) {
        String action = intent.getAction();
        Log.v("msg", "Doing...");
        if (NfcAdapter.ACTION_TECH_DISCOVERED.equals(action)) {
                Log.v("msg", "Search!");
            Tag tagFromIntent = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
            MifareClassic mytag = MifareClassic.get(tagFromIntent);
            byte[] data;
            String result = "";
            try {
                mytag.connect();
                boolean auth = false;
                int totalSec = mytag.getSectorCount();
                Log.v("msg", "Sector = "+totalSec);
                int totalBlock = 0;
                int currBlock = 0;

                for(int j = 0; j < totalSec; j++){
                    auth = mytag.authenticateSectorWithKeyB(j,
MifareClassic.KEY_DEFAULT);
                    if(auth){
                        Log.v("msg", "Authen OK : j="+j);
                        totalBlock = mytag.getBlockCountInSector(j);
                        Log.v("msg", "totalBlock = "+totalBlock);
                        int i;
                        currBlock = mytag.sectorToBlock(j);
                        for(i = 0; i < 3 ; i++){

                            data = mytag.readBlock(currBlock);
                            Log.v("msg", "data = "+data);
                            result += getReadableData(data);
                            Log.v("msg", "i = "+i);
                            currBlock++;
                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }else{
                        Log.v("msg", "Section "+j+" : Authen Fail.");
                }
                Log.v("msg", "j = "+j);
        }

        Log.v("nfcdata", "result = "+result);

        String[] read_data = result.split("dtstart");
        if(read_data.length == 2){
                String[] read_data2 = read_data[1].split("dtend");
                if(read_data2.length == 2){
                        TextView tv =
(TextView)findViewById(R.id.textView1);
                        String text2 = read_data2[0];
                        tv.setText(text2);

                        Log.v("nfcdata", "result = "+result);
                        Log.v("nfcdata", "first split = "+read_data[1]);
                        Log.v("nfcdata", "second split = "+read_data2[0]);

                        if(text2.equals("initial")){


                                if (mBluetoothAdapter == null) {
                                    // Device does not support Bluetooth
                                }

                                if (!mBluetoothAdapter.isEnabled()) {
                                    Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                                    startActivityForResult(enableBtIntent,
1);

                                }
                                try {
                                        Thread.sleep(8000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        } catch (InterruptedException e) {
                                                // TODO Auto-generated catch
block
                                                e.printStackTrace();
                                        }

                                        Intent LaunchIntent =
getPackageManager().getLaunchIntentForPackage("com.example.project");
                                        startActivity(LaunchIntent);

                                }

                                Log.v("msg", "Final Result = "+text2);

                        }
                    }
                }catch (IOException e){
                        Log.e("TAG", e.getLocalizedMessage());
                        Log.v("msg", "Tag Lost");
                }

        }
}

public String getReadableData(byte[] _bytes)
{
        String file_string = "";
        for(int i = 0; i < _bytes.length; i++)
        {
                file_string += (char)_bytes[i];
        }
        return file_string;
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โค้ดของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <SoftwareSerial.h>   //Software Serial Port
#include <avr/pgmspace.h>
#include <stdio.h>
#include <stdlib.h>
#include <String.h>
#define RxD 6
#define TxD 7
#define DEBUG_ENABLED  1
   static int digit[4];
   static int lamp[6];
   static int serial_buffer;
   static int serial_headBt;
   static int serial_dataBt;
   static int serial_headlamp;
   static int serial_datalamp;
   static int data_i;
   static int serial_headAlog;
   static int serial_dataAlog;
SoftwareSerial blueToothSerial(RxD,TxD);

static PROGMEM prog_uint32_t crc_table[16] = {
    0x00000000, 0x1db71064, 0x3b6e20c8, 0x26d930ac,
    0x76dc4190, 0x6b6b51f4, 0x4db26158, 0x5005713c,
    0xedb88320, 0xf00f9344, 0xd6d6a3e8, 0xcb61b38c,
    0x9b64c2b0, 0x86d3d2d4, 0xa00ae278, 0xbdbdf21c
};

unsigned long crc_update(unsigned long crc, byte data)
{
    byte tbl_idx;
    tbl_idx = crc ^ (data >> (0 * 4));
    crc = pgm_read_dword_near(crc_table + (tbl_idx & 0x0f)) ^ (crc >> 4);
    tbl_idx = crc ^ (data >> (1 * 4));
    crc = pgm_read_dword_near(crc_table + (tbl_idx & 0x0f)) ^ (crc >> 4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return crc;
}

unsigned long crc_string(char *s)
{
  unsigned long crc = ~0L;
  while (*s)
    crc = crc_update(crc, *s++);
  crc = ~crc;
  return crc;
}

char rcv_str[20];
int str_pos = 0;
void setup() {
  Serial.begin(9600);
  pinMode(RxD, INPUT);
  pinMode(TxD, OUTPUT);
  setupBlueToothConnection();
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);



}

void loop() {

   static int now,last;
   now = millis();
   ReadNSetValue();
   //Stat will be sent every 200 ms
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if(now - last > 500)
  {
    ReadStat2BT();
    last = now;
  }
        }

void ReadNSetValue(void)
{

  if(blueToothSerial.available())
  {
   Serial.print("Seen2");

    serial_buffer = blueToothSerial.read();
    if(serial_buffer != '$')
    {
     rcv_str[str_pos] = serial_buffer;
     str_pos++;
    }else{//Stop reading
     rcv_str[str_pos] = '\0';
     str_pos = 0;
     Serial.print("Recieve data = ");
     Serial.print(rcv_str);
     Serial.print("\n");

    //Split
    char data[20]="", crc[20]="";


    int mode=0, data_pos=0, crc_pos=0;
    for(int i=0; i<20; i++){
     if(rcv_str[i] == ':'){
       mode = 1;
       continue;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }

  if(mode == 0){
    data[data_pos] = rcv_str[i];
    data_pos++;
  }else{
    crc[crc_pos] = rcv_str[i];
    crc_pos++;
  }
}

//Dump var
 Serial.print("Data from android = ");
 Serial.print(data);
 Serial.print("\n");


 Serial.print("CRC from android = ");
 Serial.print(crc);
 Serial.print("\n");

//convert to int
int data_i, crc_i;
data_i = strtoint(data);
crc_i = strtoint(crc);
//Serial.print("Data int = ");
//Serial.print(data_i);
//Serial.print("\n");
//Serial.print(" CRC int = ");
//Serial.print(crc_i);
//Serial.print("\n");

//Encrypt
unsigned long crc_ul = strtoul(crc,NULL,10);
//Serial.print(" CRC from arduino = ");
//Serial.print(crc_ul);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Serial.print("\n");
unsigned long crc2 = crc_string(data);
//int enc_code = encrypt(data_i);
//Serial.print("Encrypted data = ");
//Serial.print(enc_code);
Serial.print("CRC from arduino = ");
Serial.print(crc2);
Serial.print("\n");
if(crc2==crc_ul)
{
serial_headBt   = data_i & 0b111110000;
serial_dataBt   = data_i & 0b000001111;
serial_headlamp = data_i & 0b111110000;
serial_datalamp = data_i & 0b000001111;
serial_headAlog = data_i & 0b100000000;
serial_dataAlog = data_i & 0b011111111;



if(serial_headAlog == 0b000000000)
{
        int c = serial_dataAlog;
        Serial.print(c);
        analogWrite(11, c);
}

if(serial_headlamp == 0b100010000)
{

  delay(1000);

   lamp[0] = 0b000001000 & serial_datalamp;
   lamp[1] = 0b000000100 & serial_datalamp;
   lamp[2] = 0b000000010 & serial_datalamp;
   lamp[3] = 0b000000001 & serial_datalamp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Lamp1
if(lamp[0] != 0)
{
   digitalWrite(10, HIGH);
 Serial.print("Lamp 2 : ON");
 Serial.print("\n");
}
else
{
   digitalWrite(10, LOW);
}


//Lamp2
if(lamp[1] != 0)
{
   digitalWrite(12, HIGH);
   Serial.print("Lamp 3 : ON");
   Serial.print("\n");
}
else
{
   digitalWrite(12, LOW);
}

//Lamp3
if(lamp[2] != 0)
{
   digitalWrite(A1, HIGH);
}
else
{
   digitalWrite(A1, LOW);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    //Lamp4
    if(lamp[3] != 0)
    {
      digitalWrite(A2, HIGH);
    }
    else
    {
      digitalWrite(A2, LOW);
    }



}

//Pass digit 0
if(serial_headBt == 0b100100000)
{
  digit[0] = 0b000001111 & serial_dataBt;
//delay(1000);
  //Serial.print(digit[0]);
}

//Pass digit 1
if(serial_headBt == 0b100110000)
{
  digit[1] = 0b000001111 & serial_dataBt;
  //Serial.print(digit[1]);
}

//Pass digit 2
if(serial_headBt == 0b101000000)
{
  digit[2] = 0b000001111 & serial_dataBt;
  //Serial.print(digit[2]);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Pass digit 3
if(serial_headBt == 0b101010000)
{
   digit[3] = 0b000001111 & serial_dataBt;
   //Serial.print(digit[3]);
}

//Pass Confirm
if(serial_headBt == 0b101100000)
{
   delay(10000);
   /*Serial.print("AT+PIN");
   Serial.print(digit[0]);
   Serial.print(digit[1]);
   Serial.print(digit[2]);
   Serial.print(digit[3]);*/
   blueToothSerial.print("AT+PIN");
   blueToothSerial.print(digit[0]);
   blueToothSerial.print(digit[1]);
   blueToothSerial.print(digit[2]);
   blueToothSerial.print(digit[3]);
   delay(1000);
   Serial.print("YESH");
}
}
else
{
  delay(500);
}


}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void ReadStat2BT(void)
{
  pinMode(3, INPUT);
  pinMode(4, INPUT);
  pinMode(5, INPUT);
  pinMode(8, INPUT);

  int tempC;
  int lamp_stat[5];
  int temPin=0;
  tempC = analogRead(temPin);
  tempC = (5.0 * tempC * 100.0)/1024.0;

  lamp_stat[0] = digitalRead(3);
  lamp_stat[1] = digitalRead(4);
  lamp_stat[2] = digitalRead(5);
  lamp_stat[3] = digitalRead(8);


  String tempCs = String(tempC);
  String lamp0 = String(lamp_stat[0]);
  String lamp1 = String(lamp_stat[1]);
  String lamp2 = String(lamp_stat[2]);
  String lamp3 = String(lamp_stat[3]);



  String status = tempCs+","+lamp0+","+lamp1+","+lamp2+","+lamp3+"!";
  blueToothSerial.print(status);

}
void setupBlueToothConnection()
{
  blueToothSerial.begin(9600); //Set BluetoothBee BaudRate to default baud rate
38400
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  blueToothSerial.print("\r\n+STWMOD=0\r\n"); //set the bluetooth work in slave
mode
  blueToothSerial.print("\r\n+STNA=SeeedBTSlave\r\n"); //set the bluetooth name as
"SeeedBTSlave"
  blueToothSerial.print("\r\n+STOAUT=1\r\n"); // Permit Paired device to connect me
  blueToothSerial.print("\r\n+STAUTO=0\r\n"); // Auto-connection should be forbidden
here
  delay(2000); // This delay is required.
  blueToothSerial.print("\r\n+INQ=1\r\n"); //make the slave bluetooth inquirable
  Serial.println("The slave bluetooth is inquirable!");
  delay(2000); // This delay is required.
  blueToothSerial.flush();
}

int encrypt(int data){
  int result = (data + 2500) / 100;
   return result;
}



int strtoint_n(char* str, int n)
{
    int sign = 1;
    int place = 1;
    int ret = 0;

    int i;
    for (i = n-1; i >= 0; i--, place *= 10)
    {
        int c = str[i];
        switch (c)
        {
            case 45:
                if (i == 0) sign = -1;
                else return -1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
            default:
                if (c >= 48 && c <= 57) ret += (c - 48) * place;
                else return -1;
        }
    }

    return sign * ret;
}

int strtoint(char* str)
{
    char* temp = str;
    int n = 0;
    while (*temp != '\0')
    {
        n++;
        temp++;
    }
    return strtoint_n(str, n);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้