

การควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์มือถือ
THE CONTROL OF ELECTRICAL EQUIPMENT
BY EMPLOYING MOBILE PHONES



โดย

ถาวร วัฒนกุล
สุทินา ทรัพย์ยิ่ง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์มือถือ
THE CONTROL OF ELECTRICAL EQUIPMENT
BY EMPLOYING MOBILE PHONES

โดย

นายฉัตรวุฒิ เจนปรีดากุล 53010563

นางสาวสุทินา ทรัพย์ยิ่ง 53011744

อาจารย์ที่ปรึกษา

ผศ.ดร.สมเกียรติ ฤกษ์วีรณูญ

อาจารย์ที่ปรึกษาร่วม

รศ.ดร.พรชัย ทรัพย์นิธิ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

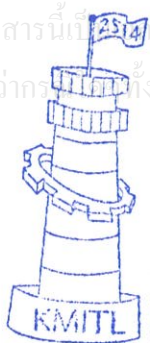
สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น เว้นแต่จะได้รับอนุญาตเป็นอย่างอื่น จากเอกสารทุกฉบับที่ปรากฏไปใช้

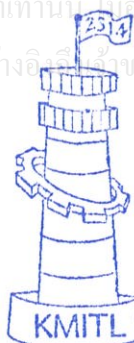


ผ่านการตรวจรูปเล่มแล้ว

(.....)
อาจารย์ที่ปรึกษา

๒๖/๓/๕๖

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

(.....)
กรรมการผู้ตรวจชิ้นงาน

15/5/57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาบัตรปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์มือถือ

THE CONTROL OF ELECTRICAL EQUIPMENT BY EMPLOYING MOBILE PHONES

ผู้จัดทำ

1. นายถิรวุฒิ เจนปรีดากุล 53010563
2. นางสาวสุทีนา ทรัพย์ยิ่ง 53011744



.....อาจารย์ที่ปรึกษา

(ผศ.ดร.สมเกียรติ ฤกษ์วรรณ)

.....อาจารย์ที่ปรึกษาร่วม

(รศ.ดร.พรชัย ทรัพย์นิธิ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ โดยความช่วยเหลือจากอาจารย์ที่ปรึกษา ผศ.ดร.สมเกียรติ ฤกษ์วีระบุญ และรศ.ดร.พรชัย ทรัพย์นิธิ ที่คอยให้ความช่วยเหลือ คำแนะนำ และแนวทางที่ดีในการทำโครงการที่มอบให้แก่คณะผู้จัดทำ ต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ที่ให้ความช่วยเหลือและคอยให้กำลังใจและแนวคิดดีๆ ทุกครั้งที่เกิดปัญหา และคอยร่วมทุกข์ร่วมสุขมาโดยตลอด

ขอบคุณครอบครัวอันเป็นที่รักยิ่งที่สนับสนุนในทุกๆ เรื่อง ไม่ว่าจะเป็นกำลังกายและกำลังใจตลอดเวลา คุณค่าและประโยชน์อันพึงได้รับจากปริญญาานิพนธ์นี้ ผู้จัดทำขอมอบผู้มีพระคุณทุกท่าน



อิรุติ เจนปรีดากุล
สุทินา ทรัพย์ยิ่ง
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์มือถือ
THE CONTROL OF ELECTRICAL EQUIPMENT
BY EMPLOYING MOBILE PHONES

โดย นายถิรวุฒิ เจนปรีดากุล 53010563
นางสาวสุทีนา ทรัพย์ยิ่ง 53011744

อาจารย์ที่ปรึกษา ผศ.ดร.สมเกียรติ ฤกษ์วีระบุญ
อาจารย์ที่ปรึกษาร่วม รศ.ดร.พรชัย ทรัพย์นินิ

บทคัดย่อ

ปริญญาานิพนธ์นี้ปริญญาานิพนธ์นี้มีวัตถุประสงค์เพื่อนำโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์ที่มีอยู่มาควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน โดยมีแอปพลิเคชันที่สั่งการสามารถเปิดและปิดอุปกรณ์ไฟฟ้าที่กำหนด อันได้แก่ พัดลม โทรทัศน์ เครื่องปรับอากาศ และหลอดไฟ นอกจากนี้ยังสามารถตรวจจับอุณหภูมิในบริเวณรอบๆได้และยังสามารถควบคุมความสว่างของหลอดไฟได้ โดยใช้บลูทูธ เป็นตัวกลางในการสื่อสารระหว่างโทรศัพท์มือถือและไมโครคอนโทรเลอร์ เพื่อควบคุมและสั่งการไปยังอุปกรณ์ไฟฟ้าได้

Abstract

This project, The Control of Electrical Equipment by Employing Mobile Phones, aims at controlling home electrical equipment by using mobile phones in which Android operating system and using the Bluetooth module for the medium in communication between the mobile phone and microcontroller. This application in mobile phone can control the states of opening and closing the specified device such as fan, television, air conditioner and light bulb that, in addition, will be adjusted the brightness. Moreover, the mobile phone can check the temperature in the surrounding area.

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ.....	I
บทคัดย่อ.....	II
สารบัญ.....	III
สารบัญรูป.....	VI
สารบัญตาราง.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตปริญญาานิพนธ์.....	2
1.4 ขั้นตอนการดำเนินการ.....	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	5
2.1 ระบบปฏิบัติการแอนดรอยด์ (Android OS).....	5
2.1.1 Eclipse.....	6
2.1.2 SDK (Software Development Kit).....	7
2.1.2.1 ADT (Android Development Tools).....	7
2.1.2.2 AVD (Android Virtual Device).....	8
2.2 บลูทูธ (Bluetooth).....	8
2.2.1 การทำงานของบลูทูธ.....	8
2.2.2 การเชื่อมโยงของบลูทูธ.....	9
2.3 ภาษาจาวา (Java).....	10
2.3.1 พื้นฐานในการเขียนโปรแกรมภาษาจาวา.....	10
2.3.1.1 คลาส (Class).....	10
2.3.1.2 วัตถุ (Object).....	10
2.4 ปุ่มสลับหรือปุ่มที่อกเกิด (Toggle Button).....	11
2.4.1 คำสั่งสำหรับสร้างปุ่มที่อกเกิด.....	12
2.4.2 การดัดแปลงการเรียกกลับเพื่อใช้กับปุ่มที่อกเกิด.....	12
2.4.3 การกำหนดสถานะ (State) ของปุ่มที่อกเกิด.....	12
2.5 การปรับความกว้างของสัญญาณพัลส์ (Pulse Width Modulation: PWM).....	13

สารบัญ (ต่อ)

	หน้า
2.5.1 ลูกคลื่นหรือพัลส์ (Pulse)	13
2.5.2 วงจรเปรียบเทียบสัญญาณ (Comparator)	15
2.5.2.1 Operating Amplifier (Op-Amp).....	15
2.6 รีเลย์ (Relay)	16
2.5.1 Operating Amplifier (Op-Amp).....	16
2.5.1 รีเลย์ควบคุม (Control Relay).....	17
2.5.1 คุณสมบัติและโครงสร้างของรีเลย์	17
2.5.1 การทำงานของรีเลย์.....	17
2.7 อาดูอิโน่ (Arduino).....	18
บทที่ 3 การออกแบบและการจัดทำปฏิญานิพนธ์	19
3.1 การออกแบบ	19
3.2 ขั้นตอนการออกแบบ.....	19
3.2.1 วิธีการออกแบบสำหรับไมโครคอนโทรลเลอร์.....	20
3.2.2 วิธีการออกแบบสำหรับแอปพลิเคชัน.....	20
3.3 วงจรที่ใช้ในการออกแบบ	20
3.3.1 วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์	20
3.3.2 วงจรตรวจจับอุณหภูมิ	21
3.3.3 วงจรหรีไฟโดยใช้สัญญาณการรับความกว้างของสัญญาณพัลส์.....	22
3.3.4 วงจรตรวจสอบสถานะการเปิดปิดของอุปกรณ์ไฟฟ้า	23
3.4 เครื่องมือที่ใช้ในการทดสอบ	24
3.5 Flow Chart แสดงการทำงานของระบบ	24
3.5.1 Flow Chart การทำงานของระบบในภาคการศึกษาที่ 1	25
3.5.2 Flow Chart ของระบบในส่วนของแอปพลิเคชันในภาคการศึกษาที่ 2	26
3.5.3 Flow Chart ของระบบในส่วนของฮาร์ดแวร์ในภาคการศึกษาที่ 2	28
บทที่ 4 ผลการออกแบบและการทดสอบ.....	29
4.1 บทนำ.....	29
4.2 ผลการทดสอบในส่วนของแอปพลิเคชัน	29
4.2.1 หน้าลงชื่อเข้าใช้ (Log-in Page)	29

เอกสารที่ 4.2 ผลการทดสอบในส่วนของแอปพลิเคชันและการสืบค้นงานนั้น ไปจนจบเล่มนี้ให้ไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใด 4.2.1 หน้าลงชื่อเข้าใช้ (Log-in Page)และต้องคำนึงถึงแล้วขอเอกสารทุกครั้งที่มีภาระน: 29 ใช้

สารบัญ (ต่อ)

	หน้า
4.2.2 หน้าหลักสำหรับควบคุม (Main Control Page)	30
4.3 ผลการทดสอบในส่วนของไมโครคอนโทรลเลอร์	31
4.3.1 การควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์.....	31
4.3.2 การปรับความสว่างของหลอดไฟ.....	33
4.3.3 การวัดอุณหภูมิ.....	42
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	50
5.1 สรุปผลการทดลองและภาพรวมของโครงการทั้งหมด	50
5.2 ข้อเสนอแนะ	50
บรรณานุกรม	51
ภาคผนวก ก. Source Code ที่ใช้ในการทำงานของแอปพลิเคชัน	52
ภาคผนวก ข. Source Code ที่ใช้ในการทำงานของอาตูดิโอ.....	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญรูป

รูปที่	หน้า
1.1 บล็อกไดอะแกรมของการควบคุมอุปกรณ์ไฟฟ้าในบ้านผ่านบลูทูธ.....	2
1.2 ขั้นตอนการทำโครงงาน	3
2.1 ตัวอย่างโปรแกรม Eclipse	7
2.2 ตัวอย่าง ADV ที่ถูกจำลองใช้บนคอมพิวเตอร์.....	8
2.3 อุปกรณ์ที่สามารถใช้กับบลูทูธได้ในการสื่อสารแบบไร้สาย	9
2.4 การเชื่อมโยงของบลูทูธ	10
2.5 เปรียบเทียบสัญญาณไฟฟ้ากระแสตรงกับสัญญาณสามเหลี่ยม.....	14
2.6 เปรียบเทียบความกว้างของพัลส์ที่ไม่คงที่กับสัญญาณไฟฟ้าที่เปลี่ยนแปลงตลอดเวลา.....	14
2.7 วงจรเปรียบเทียบเลขฐานสองขนาด 1 บิต.....	15
2.8 ทดลองวงจรการทำงานของวงจรเปรียบเทียบเลขฐานสองขนาด 1 บิต.....	15
2.9 สัญลักษณ์ออปแอมป์.....	16
2.10 รีเลย์กำลัง.....	16
2.11 รีเลย์ควบคุม.....	17
2.12 โครงสร้างทั่วไปของรีเลย์	17
2.13 รีเลย์ในสภาวะปกติ.....	18
2.14 รีเลย์ในสภาวะจ่ายไฟฟ้า.....	18
2.15 บอร์ด Arduino ต่อกับ LED.....	18
3.1 บล็อกไดอะแกรมของการควบคุมอุปกรณ์ไฟฟ้าในบ้านผ่านบลูทูธ.....	19
3.2 วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์	21
3.3 คุณสมบัติของ LM 35.....	22
3.4 การต่อ LM 35 กับบอร์ดอาดูอิโน่.....	22
3.5 วงจรหรีไฟโดยใช้เทคนิค Pulse Width Modulation.....	23
3.6 วงจรตรวจสอบสถานะการเปิดปิดของอุปกรณ์ไฟฟ้า.....	24
3.7 Flow Chart ของระบบในภาคการศึกษาที่ 1.....	25
3.8 Flow Chart แสดงการทำงานของระบบในส่วนของแอปพลิเคชันในภาคการศึกษาที่ 2....	26
3.9 Flow Chart แสดงการทำงานของระบบในส่วนของฮาร์ดแวร์ในภาคการศึกษาที่ 2.....	28
4.1 หน้าลือกอินของแอปพลิเคชัน	29
4.2 หน้าหลักของแอปพลิเคชัน	30

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 1 โดยการส่งสถานะ L.....	31
4.4 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 2 โดยการส่งสถานะ M.....	31
4.5 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 3 โดยการส่งสถานะ N	32
4.6 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 4 โดยการส่งสถานะ O	32
4.7 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 5 โดยการส่งสถานะ P.....	33
4.8 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 0 จากโทรศัพท์มือถือ	34
4.9 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 1 จากโทรศัพท์มือถือ	34
4.10 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 2 จากโทรศัพท์มือถือ	34
4.11 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 3 จากโทรศัพท์มือถือ	35
4.12 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 4 จากโทรศัพท์มือถือ	35
4.13 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 5 จากโทรศัพท์มือถือ	36
4.14 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 6 จากโทรศัพท์มือถือ	36
4.15 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 7 จากโทรศัพท์มือถือ	37
4.16 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 8 จากโทรศัพท์มือถือ	37
4.17 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 9 จากโทรศัพท์มือถือ	38
4.18 Pulse Width Modulation ที่มี Duty Cycle เป็น 10%	39
4.19 Pulse Width Modulation ที่มี Duty Cycle เป็น 25%	39
4.20 Pulse Width Modulation ที่มี Duty Cycle เป็น 50%	40
4.21 Pulse Width Modulation ที่มี Duty Cycle เป็น 75%	41
4.22 Pulse Width Modulation ที่มี Duty Cycle เป็น 80%	41
4.23 Pulse Width Modulation ที่มี Duty Cycle เป็น 100%	42
4.24 การพล็อตความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ.....	44
4.25 กราฟแสดงเส้นแนวโน้มของกราฟความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ	44
4.26 การสร้างสมการเส้นตรงของกราฟ สร้างโดยโปรแกรม Microsoft Excel.....	45
4.27 กราฟความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ และสมการเส้นตรง	45
4.28 กราฟความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ และสมการเส้นตรง	46

สารบัญรูป (ต่อ)

หน้า

รูปที่

- 4.29 การแสดงค่ากับอุณหภูมิบน Serial Monitor และหน้าแอปพลิเคชันที่อุณหภูมิ 20 °C... 48
- 4.30 การแสดงค่ากับอุณหภูมิบน Serial Monitor และหน้าแอปพลิเคชันที่อุณหภูมิ 25 °C... 49



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 รหัสชื่อเวอร์ชันเป็นชื่อขนมหวาน โดยจะเรียงตามลำดับตัวอักษร	5
2.2 ความแตกต่างระหว่างคลาสกับวัตถุ	11
4.1 ผลการทดลองค่าของแรงดันที่วัดได้เปรียบเทียบกับค่าอุณหภูมิจากสมการ	43
4.2 เปรียบเทียบค่าอุณหภูมิจากสมการและค่าความคลาดเคลื่อน	47



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันระบบปฏิบัติการแอนดรอยด์ (Android) ถือว่าทันสมัยและถูกใช้อย่างแพร่หลายเป็นอย่างมาก โดยมีอุปกรณ์ที่ใช้แอนดรอยด์มากมาย อาทิเช่น โทรศัพท์มือถือ (Smart Phone), แท็บเล็ต (Tablet), เน็ตบุ๊ก (Netbook), เครื่องอ่าน E-book, เครื่องเล่น MP4, โทรทัศน์ (Smart TV), เครื่องเล่นเกม, กล้องถ่ายรูป และนาฬิกาข้อมือ เป็นต้น

บลูทูธ (Bluetooth) เป็นระบบสื่อสารของอุปกรณ์อิเล็กทรอนิกส์แบบ 2 ทาง ด้วยคลื่นวิทยุระยะสั้น (Short-Range Radio Links) โดยปราศจากการใช้สายเคเบิล (Cable) หรือสายสัญญาณเชื่อมต่อ ช่วยให้อุปกรณ์อิเล็กทรอนิกส์สามารถเชื่อมต่อกันได้ ซึ่งทำให้การใช้ชีวิตของเราสะดวกขึ้น

ในปัจจุบันเทคโนโลยีสมาร์ทโฟนที่ใช้แอนดรอยด์เชื่อมต่ออุปกรณ์ต่างๆ ผ่านบลูทูธ ถูกนำมาประยุกต์ใช้งานในชีวิตประจำวันอย่างหลากหลาย เช่น ควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านผ่านบลูทูธ ควบคุมรถบังคับวิทยุผ่านบลูทูธ หรือการอ่านค่า DIA, SYS, Pluse จากเครื่องวัดความดันโลหิตผ่านบลูทูธ เพื่อส่งข้อมูลไปวิเคราะห์ยังเว็บเซิร์ฟเวอร์ หรือให้แพทย์ผู้เชี่ยวชาญได้วินิจฉัยโรค โดยในโครงการปริญญาโทฉบับนี้ จะเน้นถึงการนำหลักการควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้าน เพื่อพัฒนาและประยุกต์ใช้สำหรับ Wiser Home ในอนาคตอีกด้วย

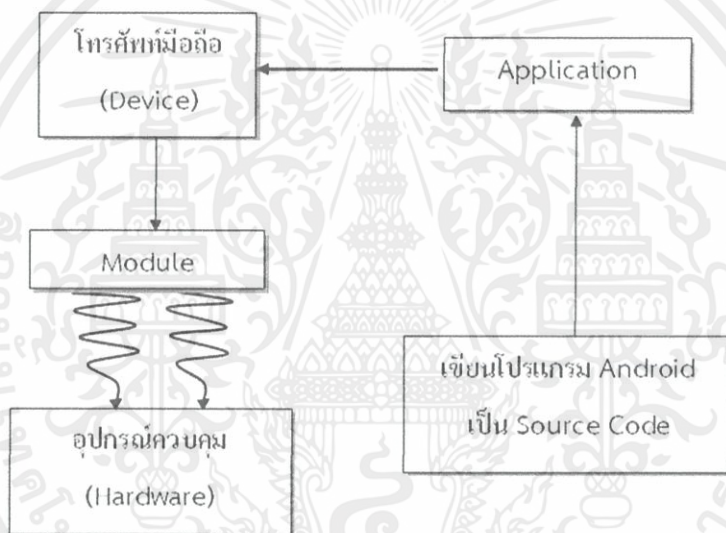
ดังนั้นผู้จัดทำโครงการนี้จึงได้นำเสนอโครงการการควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์มือถือ โดยเขียนโปรแกรมเป็นแอปพลิเคชัน (Application) บนแอนดรอยด์ เพื่อควบคุมและเชื่อมต่อกับบลูทูธ โดยมีโมดูลของบลูทูธ (Bluetooth Module) เป็นตัวส่งการไมโครคอนโทรลเลอร์ (Microcontroller) ในการควบคุมสวิทช์เปิดและปิดอุปกรณ์ไฟฟ้าต่างๆ ได้แก่ พัดลม เครื่องปรับอากาศ โทรทัศน์ และหลอดไฟ ในที่นี้ยังสามารถควบคุมความสว่างของหลอดไฟได้นอกจากนี้ยังควบคุมสัญญาณที่เป็นอนาล็อก (Analog) นั่นคือ การตรวจจับอุณหภูมิได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

การนำโทรศัพท์มือถือที่มีอยู่ มาควบคุมอุปกรณ์ไฟฟ้าภายในบ้านมีวัตถุประสงค์หลักๆดังนี้

- 1) ควบคุมสวิตช์เพื่อเปิดและปิดอุปกรณ์ไฟฟ้าที่ได้กำหนดขึ้น อันได้แก่ พัดลม เครื่องปรับอากาศ โทรทัศน์ และหลอดไฟ โดยสามารถสั่งการผ่านโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์ ซึ่งมีบลูทูธเป็นตัวกลางในการสั่งการ
- 2) ตรวจสอบและวัดอุณหภูมิของอากาศบริเวณรอบๆได้
- 3) สั่งการและควบคุมอุปกรณ์ไฟฟ้าทั้งดิจิทัล (Digital) และอนาล็อกโดยไม่ใช้สาย (Wireless) ในระยะทาง 5 – 10 เมตร



รูปที่ 1.1 บล็อกไดอะแกรมของการควบคุมอุปกรณ์ไฟฟ้าในบ้านผ่านบลูทูธ

1.3 ขอบเขตของปริญญาโท

ในปริญญาโทนี้ได้กำหนดขอบเขตการทำงานไว้ดังนี้

สร้างแอปพลิเคชันบนโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์เพื่อควบคุมอุปกรณ์ไฟฟ้าภายในบ้านโดยทำหน้าที่เป็นสวิตช์เปิดและปิดอุปกรณ์ 3 ตัว (3 ดิจิทัลเอาต์พุต (Digital Inputs)) ได้แก่ หลอดไฟ พัดลม และเครื่องปรับอากาศ โดยในหัวข้อนี้ได้ดำเนินการไปในภาคการศึกษาที่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) สร้างแอปพลิเคชันบนโทรศัพท์มือถือโดยเพิ่มสวิตช์ของการเปิดและปิดอุปกรณ์ไฟฟ้าเป็น 5 ตัว (5 ดิจิทัลเอาต์พุต) ได้แก่ พัดลม เครื่องปรับอากาศ โทรทัศน์ และหลอดไฟ และมีการปรับความสว่างของหลอดไฟได้อีก 1 ตัว (1 ดิจิทัลเอาต์พุต) และได้มีการเพิ่มการตรวจจับและวัดอุณหภูมิ 1 ตัว (1 อนุาล็อกอินพุต (Analog Input))
- 2) มีการลงชื่อเข้าใช้ (Log-in) ก่อนเข้าหน้าหลัก เพื่อความปลอดภัยของระบบ

1.4 ขั้นตอนการดำเนินการ

ในการทำโครงงานได้มีการศึกษา และการวางแผนดำเนินงานตามขั้นตอนดังรูปที่ 1.2



รูปที่ 1.2 ขั้นตอนการทำโครงงาน

การดำเนินงานการทำโครงงานได้ดำเนินงานตามรูปแบบดังรูป 1.2 ดังนี้

การศึกษาทฤษฎีจะกล่าวในบทที่ 2 โดยจะกล่าวถึงทฤษฎีต่างๆ และใช้ประกอบในการทำโครงงาน โดยเป็นความรู้พื้นฐานเกี่ยวกับเทคโนโลยีบลูทูธ ทฤษฎีการปรับความกว้างของสัญญาณพัลส์ และคุณลักษณะของอุปกรณ์ต่างๆของไมโครคอนโทรลเลอร์นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบ และสร้างสายอากาศจะอยู่ในบทที่ 3 โดยจะกล่าวถึงการออกแบบ และขั้นตอนในการสร้างแอฟลิเคชันและไมโครคอนโทรเลอร์

การทดสอบคุณลักษณะจะอยู่ในบทที่ 4 โดยจะกล่าวถึงการทดสอบคุณลักษณะต่างๆของแอฟลิเคชันและไมโครคอนโทรเลอร์

การวิเคราะห์ผลจะอยู่ในบทที่ 5 โดยทำการวิเคราะห์ผล และสรุปผลที่ได้จากการทดสอบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ระบบปฏิบัติการแอนดรอยด์ (Android OS)

คือระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือและคอมพิวเตอร์แบบแท็บเล็ต (Tablet Computer หรือ Tablet PC) เป็นต้น แอนดรอยด์นั้นพัฒนามาจากระบบปฏิบัติการลินุกซ์เคอร์เนล(Linux Kernel) เริ่มพัฒนาโดยบริษัทแอนดรอยด์ จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล(Google) และทางกูเกิลได้นำแอนดรอยด์ไปพัฒนาต่อ ส่วนด้านลิขสิทธิ์ของโค้ดแอนดรอยด์จะใช้ในลักษณะของซอฟต์แวร์เสรีหรือโอเพ่นซอร์ส (Open Source) ทำให้นักพัฒนาสามารถแก้ไข ดัดแปลงโค้ดแอนดรอยด์ได้อย่างอิสระ สำหรับระบบปฏิบัติการแอนดรอยด์เป็นที่รู้จักต่อสาธารณชนเมื่อวันที่ 5 พฤศจิกายน 2550 และแอนดรอยด์เวอร์ชัน 1.0 ถูกปล่อยออกมาใช้งานอย่างเป็นทางการครั้งแรกเมื่อวันที่ 28 กันยายน 2551 เวอร์ชัน (Version) ของระบบปฏิบัติการแอนดรอยด์ จะใช้รหัสชื่อเวอร์ชันเป็นชื่อขนมหวาน โดยจะเรียงตามลำดับตัวอักษร

ตารางที่ 2.1 รหัสชื่อเวอร์ชันเป็นชื่อขนมหวาน โดยจะเรียงตามลำดับตัวอักษร

เวอร์ชัน	ชื่อเวอร์ชัน	เปิดตัว
1.0	Astro (แอสโตร)	28 กันยายน 2551
1.1	Bender (เบนเดอร์)	9 กุมภาพันธ์ 2552
1.5	Cupcake (คัพเค้ก)	30 เมษายน 2552
1.6	Donut (โดนัท)	15 สิงหาคม 2552
2.0/2.1	Eclair (เอแคลร์)	26 ตุลาคม 2552 (2.0)
		12 มกราคม 2553 (2.1)
2.2	Froyo (โฟรชเชนโยเกิร์ต)	20 พฤษภาคม 2555
2.3	Gingerbread (ขนมปังซิง)	6 ธันวาคม 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้สำหรับเรียนการสอนในชั้นเรียนเท่านั้น ไม่อนุญาตให้ทำซ้ำหรือเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.0/3.1	Honeycomb (รังผึ้ง)	22 กุมภาพันธ์ 2554
4.0	Ice Cream Sandwich (ไอศกรีมแซนด์วิช)	19 ตุลาคม 2554
4.1	Jelly Bean (เจลลี่빈)	28 มิถุนายน 2555

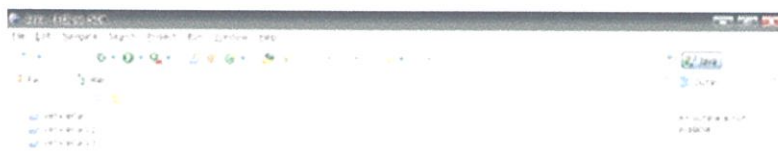
2.1.1 Eclipse

คือโปรแกรมที่ใช้สำหรับพัฒนาภาษา Java ซึ่งโปรแกรม Eclipse เป็นโปรแกรมที่ใช้ในการพัฒนา Application Server ได้อย่างมีประสิทธิภาพ และเนื่องจาก Eclipse เป็นซอฟต์แวร์เสรีที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆจากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมีองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug-ins ดังนั้นหากต้องการให้ Eclipse ทำงานใดเพิ่มเติม ก็เพียงแค่พัฒนาปลั๊กอินสำหรับงานนั้นขึ้นมา และนำปลั๊กอินนั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse Plug-in ที่มีมาพร้อมกับ Eclipse เมื่อดาวน์โหลดมาครั้งแรกก็คือองค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ Debug โปรแกรมภาษา Java

ข้อดีของโปรแกรม Eclipse คือ ติดตั้งง่าย สามารถใช้ได้กับ J2SDK ได้ทุกเวอร์ชัน (Version) รองรับภาษาต่างประเทศอีกหลายภาษา มี plug-in ที่ใช้เสริมประสิทธิภาพของโปรแกรม สามารถทำงานได้กับไฟล์หลายชนิด เช่น HTML, Java, C, JSP, EJB, XML และ GIF และที่สำคัญเป็นฟรีแวร์(Free ware)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ตัวอย่างโปรแกรม Eclipse

2.1.2 SDK (Software Development Kit)

คือเครื่องมือที่เอาไว้สำหรับพัฒนาโปรแกรมหรือแอปพลิเคชันบนระบบ Android OS ซึ่งทาง Google พัฒนาออกมาเพื่อแจกจ่ายให้นักพัฒนาแอปพลิเคชัน หรือผู้สนใจทั่วไปดาวน์โหลดไปใช้กันโดยไม่มีค่าใช้จ่าย และนี่ก็เป็นหนึ่งในปัจจัยที่ทำให้แอปพลิเคชันบนแอนดรอยด์นั้นเพิ่มขึ้นอย่างรวดเร็ว ซึ่งในชุด SDK นั้นจะมีโปรแกรมและไลบรารีต่างๆ ที่จำเป็นต่อการพัฒนาแอปพลิเคชันบนแอนดรอยด์ อย่างเช่น อิมูเลเตอร์ ซึ่งทำให้ผู้ใช้สามารถสร้างแอปพลิเคชันและนำมาทดลองรันบนตัวอิมูเลเตอร์ก่อน โดยมีสภาวะแวดล้อมเหมือนมือถือที่รันระบบปฏิบัติการแอนดรอยด์จริงๆ

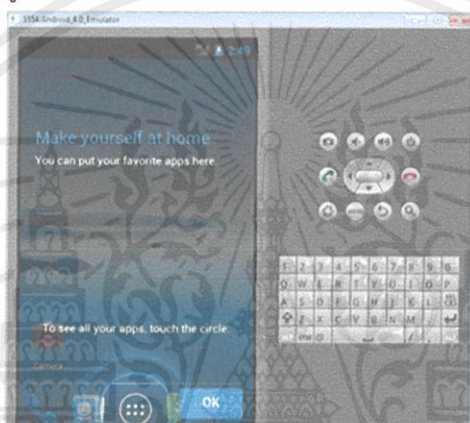
2.1.2.1 ADT (Android Development Tools)

คือ เครื่องมือที่ใช้พัฒนาแอนดรอยด์ ในการพัฒนาแอปพลิเคชันบนระบบ Android OS จะใช้ภาษา Java โดยต้องติดตั้งส่วนเสริม คือ ADT ซึ่งเป็นส่วนเสริมของ IDE(Integrated Development Environment) หรือที่หลายคนเรียกว่าเป็นปลั๊กอินของโปรแกรม Eclipse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 ADV (Android Virtual Device)

คือ การจำลอง หรือ อีมูเลเตอร์ เครื่องโทรศัพท์มือถือระบบปฏิบัติการ แอนดรอยด์บนเครื่องคอมพิวเตอร์ เพื่อทดสอบโปรแกรม หรือ โค้ดของโปรแกรมที่ได้เขียนขึ้น ข้อดีของ ADV คือ ADV นั้น เป็นอีมูเลเตอร์ที่เพิ่มความสะดวกสบายในการพัฒนาแอปพลิเคชัน สำหรับแอนดรอยด์โดยหลังจากที่ผู้พัฒนาเขียนแอปพลิเคชันเสร็จแล้ว สามารถส่งแอปพลิเคชันไป ทดลองรันบนคอมพิวเตอร์ ที่ได้ทำให้เป็นอีมูเลเตอร์ดูได้เลย แต่สำหรับผู้ใช้งานทั่วไปนั้น เราก็ สามารถนำแอปพลิเคชันตัวนี้ มาทดลองใช้เหมือนเป็น Android Device บนเครื่องหนึ่งบน คอมพิวเตอร์ที่ได้ทำให้เป็นอีมูเลเตอร์ได้



รูปที่ 2.2 ตัวอย่าง ADV ที่ถูกจำลองใช้บนคอมพิวเตอร์

2.2 บลูทูธ (Bluetooth)

เป็นเทคโนโลยีของอินเทอร์เน็ตเฟส (Interface) ทางคลื่นวิทยุ (คือการทำงานที่ใช้ไมโครชิพ ขนาด 9 มม. X 9 มม. ซึ่งทำงานเป็นตัวเชื่อมใช้สัญญาณวิทยุขนาดระยะสั้นและมีราคาถูก) ใช้ในการเชื่อมโยงสื่อสารไร้สายในแถบความถี่ 2.45GHz ทำให้อุปกรณ์อิเล็กทรอนิกส์ที่ถือเคลื่อนย้ายได้ สามารถติดต่อเชื่อมโยงสื่อสารแบบไร้สายระหว่างกันในช่วงระยะห่างสั้นๆ ได้

2.2.1 การทำงานของบลูทูธ

เป็นเทคโนโลยีการเชื่อมโยงหรือการสื่อสารแบบใหม่ที่ถูกคิดค้นขึ้นเป็นเทคโนโลยีของอินเทอร์เน็ตเฟสทางคลื่นวิทยุตั้งอยู่บนพื้นฐานของการสื่อสารระยะใกล้ที่ปลอดภัยผ่านช่องสัญญาณความถี่ 2.4 GHz โดยที่ถูกพัฒนาขึ้นเพื่อลดข้อจำกัดของการใช้สายเคเบิลในการเชื่อมโยงโดยมีความเร็วในการเชื่อมโยงสูงสุดที่ 1 mbp ระยะครอบคลุม 10 เมตรเทคโนโลยีการส่งคลื่นวิทยุของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บลูทูธจะใช้การกระโดดเปลี่ยนความถี่ (Frequency Hop) เพราะว่าเทคโนโลยีนี้เหมาะที่จะใช้กับการส่งคลื่นวิทยุที่มีกำลังส่งต่ำและราคาถูกโดยจะแบ่งออกเป็นหลายช่องความถี่ขนาดเล็กในระหว่างที่มีการเปลี่ยนช่องความถี่ที่ไม่แน่นอนทำให้สามารถหลีกเลี่ยงสัญญาณรบกวนที่เข้ามาแทรกแซงได้ซึ่งอุปกรณ์ที่จะได้รับการยอมรับว่าเป็นเทคโนโลยีบลูทูธต้องผ่านการทดสอบจาก Bluetooth SIG (Special Interest Group) เสียก่อนเพื่อยืนยันว่ามันสามารถที่จะทำงานร่วมกับอุปกรณ์บลูทูธตัวอื่นๆ และอินเทอร์เน็ตได้

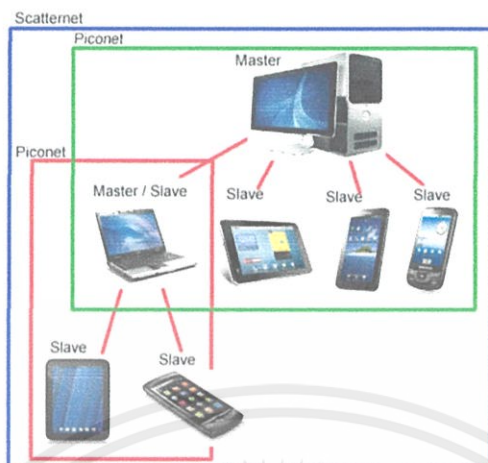


รูปที่ 2.3 อุปกรณ์ที่สามารถใช้กับบลูทูธได้ในการสื่อสารแบบไร้สาย

2.2.2 การเชื่อมโยงของบลูทูธ

Bluetooth เชื่อมต่อในรูปแบบ Oriented Service ซึ่งจะมี 2 ช่องการสื่อสารในตัวเองที่เรียกว่า Master และ Slave โดยอุปกรณ์ที่มีสถานะเป็น Master จะเป็นตัวส่งการเชื่อมต่อ การสื่อสาร ส่วนตัวลูกข่ายหรือ Slave จะเป็นตัวที่ถูกค้นหาเข้ามาเชื่อมต่อ โดยในหนึ่งอุปกรณ์ Bluetooth สามารถสลับสถานะ ทำหน้าที่สลับกันได้ (เป็นได้ทั้ง master และ slave ในเวลาเดียวกัน เรียกว่า Scatternet) สัญญาณได้จากภาพด้านบน notebook จะเป็น slave ของเครื่อง PC แต่ในเวลาเดียวกัน ตัว notebook จะเป็น Master เชื่อมต่ออีก 2 อุปกรณ์(ในวงรอบสีแดง) โดยปกติแล้ว Bluetooth นี้ สามารถเชื่อมต่ออุปกรณ์ได้ 7 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การเชื่อมโยงของบลูทูธ

2.3 ภาษาจาวา (Java)

จาวาเป็นภาษาโปรแกรมที่พัฒนาขึ้นโดยทีมวิจัยชื่อ Green ของบริษัท Sun Microsystems ซึ่งมีวัตถุประสงค์เพื่อพัฒนาภาษาที่ใช้โปรแกรมหรือระบบที่สนับสนุนการทำงานของแต่ละระบบงานย่อยๆ

2.3.1 พื้นฐานในการเขียนโปรแกรมภาษา Java

การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming: OOP) เป็นการแบ่งขอบเขตของงานออกเป็นส่วนย่อยๆ ที่เป็นอิสระต่อกันโดยมองส่วนต่างๆ เป็นวัตถุที่ไม่ขึ้นต่อกันแต่มีการทำงานร่วมกัน

2.3.1.1 คลาส (Class)

คือ การประกาศภาพรวมของกลุ่มวัตถุที่มีคุณลักษณะ (Attribute) ที่เหมือนกัน มีพฤติกรรมที่เหมือนกัน ในการประกาศคลาส สิ่งที่ต้องทำการประกาศ ประกอบด้วย ข้อมูล (Data) และ พฤติกรรม (Behavior) หรือการกระทำ (Operation) สำหรับคลาสนั้น

2.3.1.2 วัตถุ (Object)

คือ สิ่งที่เป็นตัวแทนของคลาส มีคุณลักษณะที่บ่งบอกถึงความเป็นตัวเอง

และสามารถแสดงพฤติกรรมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ความแตกต่างระหว่างคลาสกับวัตถุ

ชื่อ	คำอธิบาย	ตัวอย่าง
<i>Class</i>	เป็นพิมพ์เขียวหรือต้นแบบในการกำหนดคุณสมบัติและการทำงานไว้เพื่อนำไปใช้	<pre> Class Student { String id,name; double gpa; void show() { System.out.println("ID : "+id); System.out.println("Name : "+name); System.out.println("GPA : "+gpa); } } </pre>
ชื่อ	คำอธิบาย	ตัวอย่าง
<i>Object</i>	ออบเจกต์เกิดจากคลาส ประกอบด้วย คุณสมบัติและการทำงานตามที่คลาสดำหนด	<pre> Student s1 = new Student(); Student s2 = new Student(); </pre>

2.4 ปุ่มสลักหรือปุ่มที่อกเกิด (Toggle Button)

ลักษณะของปุ่มที่อกเกิด คือ เป็นปุ่มที่เสมือนตัวเก็บสถานะ มีลักษณะเพื่อควบคุมปุ่มตัวเลือก (Optional Button) และปุ่มคำสั่ง (Command Button) เมื่อมีการกด (Down) ปุ่มนี้จะมีการแสดงสถานะหนึ่ง และเมื่อกดปุ่มนี้ซ้ำ (Up) ไปอีกครั้งหนึ่งก็จะกลับสู่สภาพเดิม ดังนั้นปุ่มที่อกเกิดจึงเหมาะกับการแสดงสถานะ เช่น ปุ่มแสดงสถานะของปุ่ม Num Lock, Caps Lock, Scroll Lock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 คำสั่งสำหรับสร้างปุ่มที่อกเกิด ได้แก่

- 1) GtkWidget * gtk_toggle_button_new (void);
- 2) GtkWidget * gtk_toggle_button_new_with_label (gchar * label);

2.4.2 การดัดแปลงการเรียกกลับเพื่อใช้กับปุ่มที่อกเกิด

ในการดักจับสัญญาณนั้น ต้องตรวจสอบสถานะด้วยว่าปุ่มที่อกเกิดถูกกดค้างไว้หรืออยู่ในสภาพปกติ โดยตัวอย่างการดัดแปลงการเรียกกลับเพื่อใช้กับปุ่มที่อกเกิด คือ

```
void tgl_btn_callback (GtkWidget * widget, gpointer data)
{
    if (GTK_TOGGLE_BUTTON(widget) -> active)
    {
        /* กำหนดคำสั่งต่าง ๆ ที่รองรับกับสถานะ DOWN ของ toggle */
    } else {
        /* กำหนดคำสั่งรองรับกับสถานะ UP */
    }
}
```

2.4.3 การกำหนดสถานะ (State) ของปุ่มที่อกเกิด

คำสั่งในการกำหนดสถานะของปุ่มที่อกเกิด คือ

```
void gtk_toggle_button_set_state (GtkToggleButton *toggle_button,
gint state);
```

ซึ่งค่าของ gint state จะเป็นค่า TRUE หรือ FALSE ถ้าเป็น TRUE แสดงว่าให้ปุ่มที่อกเกิดเป็น DOWN และถ้าต้องการให้ค่าเป็น FALSE จะเป็นการกำหนดให้ปุ่มที่อกเกิดเป็น UP นอกจากนี้ยังมีคำสั่งอื่นอีกหนึ่งคำสั่งที่จะทำการเปลี่ยนสถานะของปุ่มที่อกเกิดให้ตรงกันข้ามกับในที่นี่ ได้แก่

```
void gtk_toggle_button_toggled (GTKToggleButton * toggle_button);
```

ซึ่งในการเปลี่ยนสถานะนั้น จะเป็นการส่งสัญญาณที่ชื่อว่า “Clicked” ไปที่ปุ่มที่อกเกิดไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การปรับความกว้างของสัญญาณพัลส์ (Pulse Width Modulation: PWM)

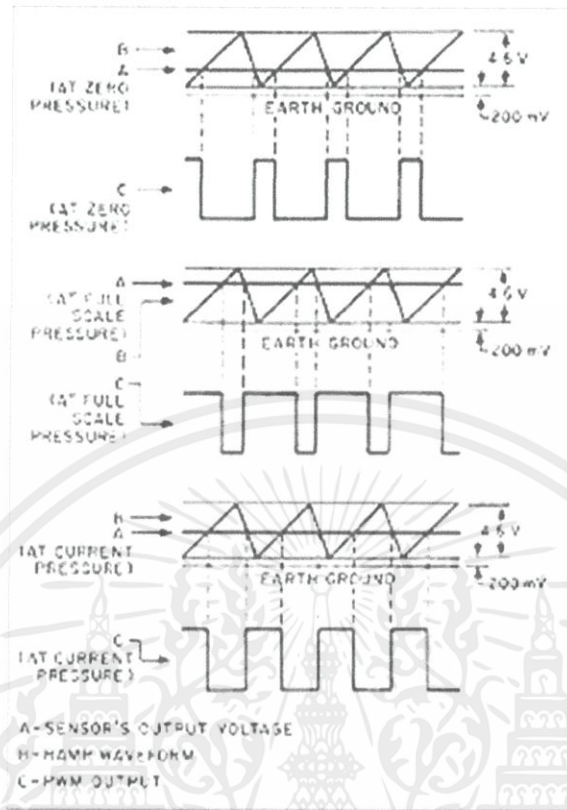
คือการผสมคลื่นความถี่ที่มีการตั้งค่าความถี่คงที่ โดยในการเปิด สัญญาณมีสถานะ High และการปิด สัญญาณมีสถานะเป็น Low เมื่อความกว้างของรูปคลื่นที่เป็น High และ Low มีความกว้างต่างกันโดยคิดเป็นเปอร์เซ็นต์ 0-100 % ค่าเฉลี่ยของแรงดันเอาต์พุตเมื่อผ่านการกรองก็จะมีค่ามากน้อยเป็นเปอร์เซ็นต์เหมือนกัน ซึ่งลักษณะของการทำงานจะเป็นการทำงานแบบสวิตช์ ซึ่งทำให้กระแสที่ไหลผ่านอุปกรณ์อิเล็กทรอนิกส์กำลังมีการสูญเสียพลังงานในรูปแบบของพลังงานความร้อนน้อยลง แต่การสูญเสียซึ่งเกิดการสวิตช์มีเพิ่มมากขึ้น การทำให้ค่าเอาต์พุตของแรงดันมีค่าการกระเพื่อมมากหรือน้อยขึ้นอยู่กับ การเปลี่ยนแปลงค่าความถี่ โดยส่วนมาก Pulse Width Modulation จะทำงานที่ความถี่สวิตช์ 10KHz ขึ้นไป (หรือที่ความถี่ที่มนุษย์ไม่ได้ยิน 20 KHz) อุปกรณ์อิเล็กทรอนิกส์กำลังที่ส่งผ่านพลังงานสูงๆ จะต้องสามารถรองรับสัญญาณอินพุตที่บังคับการปิดหรือเปิดสวิตช์ได้ และต้องออกแบบวิธีการสวิตช์ที่ลดการสูญเสียให้ได้มากที่สุด

2.5.1 ลูกคลื่นหรือพัลส์ (Pulse)

เป็นปรากฏการณ์ที่เกิดจากระดับของสัญญาณไฟฟ้าเปลี่ยนแปลงจากระดับหนึ่ง ไปสู่อีกระดับหนึ่งที่มีความแตกต่างกันระหว่างแรงดันต่ำสุดกับแรงดันสูงสุดอย่างรวดเร็ว อาจมีความต่อเนื่องหรือไม่ก็ได้ แบ่งเป็น 2 ชนิด คือ สัญญาณพัลส์บวก และสัญญาณพัลส์ลบ ซึ่งพัลส์จะมีลักษณะรูปคลื่นสูงหรือต่ำ (High-Low)

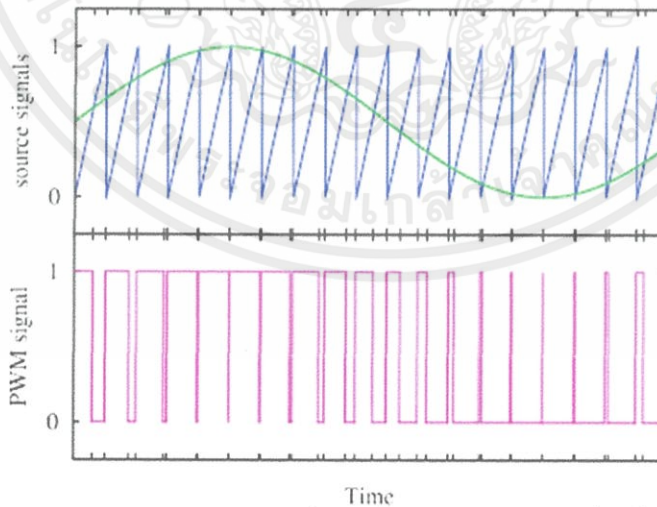
การปรับความกว้างของพัลส์โดยการนำเอาสองสัญญาณมาเปรียบเทียบกันและสองสัญญาณที่ว่านั้นก็คือสัญญาณสามเหลี่ยมกับสัญญาณที่ต้องการปรับความกว้างของพัลส์ เมื่อเปรียบเทียบสัญญาณไฟฟ้ากระแสตรงกับสัญญาณสามเหลี่ยม สัญญาณพัลส์จะมีความถี่คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 เปรียบเทียบสัญญาณไฟฟ้ากระแสตรงกับสัญญาณสามเหลี่ยม

หากไม่ใช่สัญญาณไฟฟ้ากระแสตรง แต่นำสัญญาณไฟฟ้าที่เปลี่ยนแปลงอยู่ตลอดเวลามาใช้ จะเห็นได้ว่าความกว้างของพัลส์ก็จะเปลี่ยนแปลงตามไปด้วย



รูปที่ 2.6 เปรียบเทียบความกว้างของพัลส์ที่ไม่คงที่กับสัญญาณไฟฟ้าที่เปลี่ยนแปลงตลอดเวลา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 วงจรเปรียบเทียบสัญญาณ (Comparator)

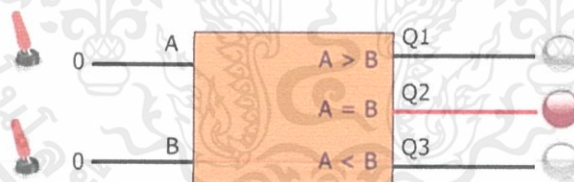
เป็นวงจรที่เปรียบเทียบระดับแรงดันอนาล็อกอินพุตสองค่า แล้วทำให้แรงดันดิจิทัลเอาต์พุตเป็น 1 หรือ 0 โดยขึ้นกับค่าแรงดันอินพุตทั้งสอง หรืออาจกล่าวได้ว่า วงจรเปรียบเทียบสัญญาณ คือ วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลแบบ 1 บิต ซึ่งวงจรเปรียบเทียบสัญญาณจะเป็นวงจรที่สร้างสัญญาณ Pulse Width Modulation

การออกแบบวงจรโดยใช้ไอซีเบอร์ 7485 ที่ทำหน้าที่เปรียบเทียบข้อมูลขนาด 4 บิต การเปรียบเทียบเป็นฟังก์ชันทางลอจิกที่สำคัญอีกลอจิกหนึ่งซึ่งทำได้โดยใช้วงจร คอมพารเตเตอร์ ซึ่งปัจจุบันจะเป็นไอซีประเภท MSI IC คือ TTL เบอร์ 7485 ซึ่งเป็นคอมพารเตเตอร์ขนาด 4 บิต ซึ่งใช้สำหรับการเปรียบเทียบเลข 2 จำนวน คือ A และ B ซึ่งกำหนดออกมาเป็น 3 รูปแบบคือ $A < B$, $A > B$ และ $A = B$

วงจรเปรียบเทียบจะทำหน้าที่ในการเปรียบเทียบข้อมูลเลขฐาน 2 จำนวน เพื่อให้ทราบว่าข้อมูลจำนวนหนึ่งมีค่า มากกว่า น้อยกว่า หรือเท่ากับอีกจำนวนหนึ่ง โดยสามารถเปรียบเทียบค่าได้ตั้งแต่ 1 บิต ถึงหลายๆ บิตได้



รูปที่ 2.7 วงจรเปรียบเทียบเลขฐานสองขนาด 1 บิต



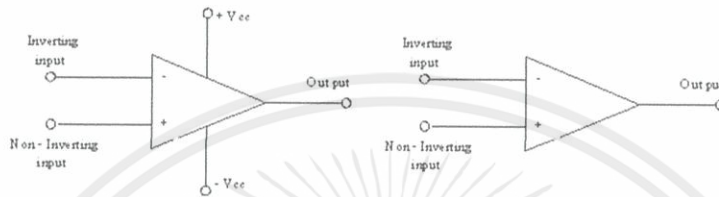
รูปที่ 2.8 ทดลองวงจรการทำงานของวงจรเปรียบเทียบเลขฐานสองขนาด 1 บิต

2.5.2.1 Operating Amplifier (Op-Amp)

เป็นวงจรขยายแบบต่อตรง (Direct coupled amplifier) ที่มีอัตราขยายสูงมากใช้การป้อนกลับแบบลบไปควบคุมลักษณะการทำงาน ทำให้ผลการทำงานของวงจรไม่ขึ้นกับพารามิเตอร์ภายในของออปแอมป์ วงจรภายในประกอบด้วยวงจรขยายที่ต่ออนุกรมกัน คือ วงจรขยายดิฟเฟอเรนเชียลด้านทางเข้า วงจรขยายดิฟเฟอเรนเชียลภาคที่สอง วงจรเลื่อนระดับและ วงจรขยายกำลังด้านทางออก สัญลักษณ์ที่ใช้แทนออปแอมป์จะเป็นรูปสามเหลี่ยม ไอซีออปแอมป์

เอกสารนี้เป็นลิขสิทธิ์ทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ผ่านการฉะพินิจจากทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไอซีที่แตกต่างไปจากลิเนียร์ไอซีต่างๆ ไปคือไอซีออปแอมป์มีขาอินพุท 2 ขา เรียกว่าขาเข้าไม่กลับเฟส (Non-Inverting Input) หรือ ขา + และขาเข้ากลับเฟส (Inverting Input) หรือขาส่วนทางด้านออกมีเพียงขาเดียว เมื่อสัญญาณป้อนเข้าขาไม่กลับเฟสสัญญาณทางด้านออกจะมีเฟสตรงกับทางด้านเข้า แต่ถ้าป้อนสัญญาณเข้าที่ขาเข้ากลับเฟส สัญญาณทางออกจะมีเฟสต่างไป 180 องศา จากสัญญาณทางด้านเข้า



รูปที่ 2.9 สัญลักษณ์ออปแอมป์

2.6 รีเลย์ (Relay)

คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับขดลวดแม่เหล็กไฟฟ้า (Solenoid) รีเลย์ใช้การควบคุมวงจรไฟฟ้าได้อย่างหลากหลายและยังเป็นอุปกรณ์แม่เหล็ก (Magnetics Device) ที่นิยมใช้อย่างแพร่หลายโดยไม่มีการแก้ไขสภาพโครงสร้างมากมาย รีเลย์แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภท

2.6.1 รีเลย์กำลัง (Power Relay)

ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา นิยมเรียกว่า คอนแท็คเตอร์ (Contactor)



รูปที่ 2.10 รีเลย์กำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 รีเลย์ควบคุม (Control Relay)

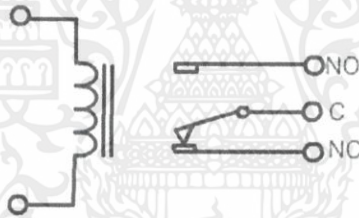
มีขนาดเล็กกว่า กำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทคเตอร์ขนาดใหญ่



รูปที่ 2.11 รีเลย์ควบคุม

2.6.3 คุณสมบัติและโครงสร้างของรีเลย์

ภายในโครงสร้างของรีเลย์ จะประกอบไปด้วยขดลวด (Coil) 1 ชุด และหน้าสัมผัส (Contactor) ซึ่งในหน้าสัมผัส 1 ชุด จะประกอบไปด้วย



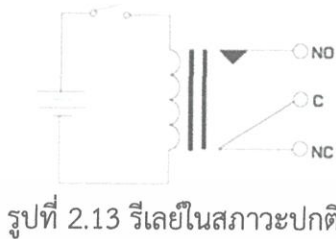
รูปที่ 2.12 โครงสร้างทั่วไปของรีเลย์

- 1) ขา C (Common) เป็นขาต่อระหว่าง NO และ NC เมื่อมีการทำงานของรีเลย์ โดยที่将有แรงดันที่ต้องการใช้งานไหลผ่านขานี้เสมอ
- 2) ขา NO (Normally Opened) หน้าสัมผัสแบบปกติเปิด โดยปกติขานี้จะเปิดเอาไว้ และจะทำงานเมื่อมีการจ่ายแรงดันไฟฟ้าให้รีเลย์
- 3) ขา NC (Normally Closed) หน้าสัมผัสแบบปกติปิด โดยปกติขานี้จะต่อกับขา C ในกรณีที่จ่ายแรงดันไฟฟ้าหน้าสัมผัสของ C และ NC จะต่อถึงกัน

2.6.4 การทำงานของรีเลย์

รีเลย์โดยทั่วไปจะทำงานก็ต่อเมื่อมีกระแสไฟฟ้าไหลผ่านขดลวด ก็จะทำให้ขดลวด
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

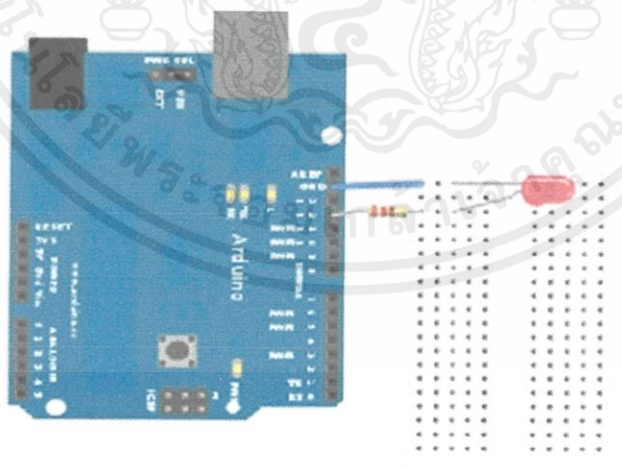
เกิดสนามแม่เหล็กไปดึงผ่านหน้าสัมผัส C ดึงลงมาแตะหน้าสัมผัส NO ทำให้มีกระแสไหลผ่านหน้าสัมผัสไปได้ ในสภาวะปกติขา C จะต่อกับ NC แต่ในสภาวะจ่ายไฟฟ้าขา C จะต่อกับ NO



2.7 อาดูอิโน้ (Arduino)

เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ ตัวบอร์ดถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้น ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ดอาดูอิโน้ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขาอินพุทของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Shield) ประเภทต่างๆ เช่น XBee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ดอาดูอิโน้ แล้วเขียนโปรแกรมพัฒนาต่อได้



รูปที่ 2.15 บอร์ด Arduino ต่อกับ LED

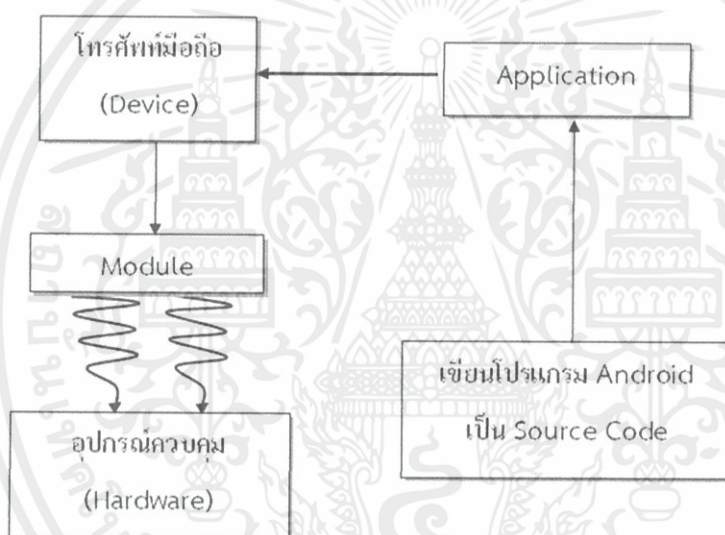
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

หลักการทำงานในภาพรวมคือ เขียนโปรแกรมเพื่อทำเป็นแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ที่สามารถทำงานบนโทรศัพท์มือถือที่รองรับระบบปฏิบัติการดังกล่าวนี้ จะนำมาเชื่อมต่อกับบลูทูธโดยมีตัวโมดูลของบลูทูธ ที่สามารถปฏิบัติการร่วมกับแอนดรอยด์ได้ มาส่งการบนไมโครคอนโทรลเลอร์เพื่อมาควบคุมสวิทช์ของอุปกรณ์ไฟฟ้าต่างๆที่ได้กำหนดขึ้น



รูปที่ 3.1 บล็อกไดอะแกรมของการควบคุมอุปกรณ์ไฟฟ้าในบ้านผ่านบลูทูธ

3.2 ขั้นตอนการออกแบบ

โครงการนี้จะแบ่งวิธีการออกแบบเป็น 2 ส่วน คือ วิธีการการออกแบบสำหรับไมโครคอนโทรลเลอร์และวิธีการการออกแบบสำหรับแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 วิธีการการออกแบบสำหรับไมโครคอนโทรเลอร์

- 1) ศึกษาวงจรที่เกี่ยวข้อง ได้แก่ วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์ วงจรตรวจจับอุณหภูมิ (Sensor) และวงจรปรับความกว้างของสัญญาณพัลส์ (Pulse Width Modulation: PWM)
- 2) ออกแบบวงจรด้วยโปรแกรม Proteus จากนั้นจึงนำอุปกรณ์มาประกอบเป็นวงจรอิเล็กทรอนิกส์ลงในโปรโตบอร์ด (Protoboard) และแผ่นบอร์ดพีซีบี (PCB board) ในภายหลัง
- 3) ทดสอบวงจรทั้งหมด เช่น ระบบไฟ เป็นต้น

3.2.2 วิธีการการออกแบบสำหรับแอปพลิเคชัน

- 1) ศึกษาออกแบบรูปร่างลักษณะและการทำงานของแอปพลิเคชัน
- 2) เขียนโปรแกรมเป็น Source Code สำหรับแอปพลิเคชันเพื่อส่งข้อมูล โดยให้โมดูลบลูทูธเป็นตัวกลางในการรับส่งสัญญาณระหว่างโทรศัพท์มือถือและไมโครคอนโทรเลอร์
- 3) ทดสอบการรับส่งข้อมูลระหว่างแอปพลิเคชันและไมโครคอนโทรเลอร์

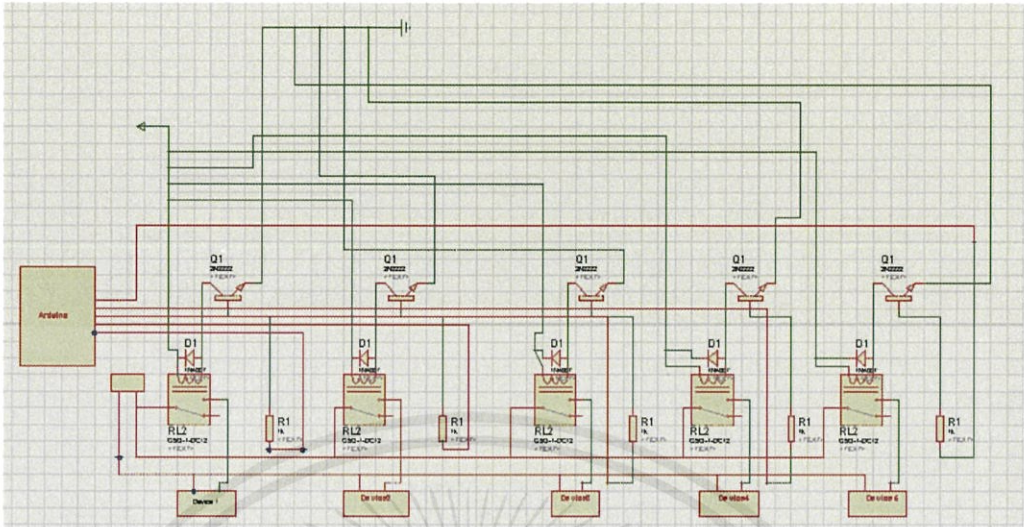
3.3 วงจรที่ใช้ในการออกแบบ

วงจรหลักที่ใช้สำหรับการออกแบบไมโครคอนโทรเลอร์นี้มีทั้งหมด 3 วงจร ได้แก่ วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์ วงจรตรวจจับอุณหภูมิ (Sensor) และวงจรปรับความกว้างของสัญญาณพัลส์ (Pulse Width Modulation: PWM)

3.3.1 วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์

เมื่อมีการสั่งการจากโทรศัพท์มือถือส่งสัญญาณไปที่อาคูโอโนเป็นตัวกลางเพื่อทำการสั่งการไปที่วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์ที่มีหน้าที่เปรียบเสมือนเป็นสวิตช์เปิดปิดไฟฟ้า ซึ่งเมื่ออาคูโอโนส่งสัญญาณไปยังวงจรรีเลย์นี้ ก็จะสามารถเปิดปิดอุปกรณ์ไฟฟ้าได้ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์

3.3.2 วงจรตรวจจับอุณหภูมิ

วงจรที่ใช้สำหรับการวัดและตรวจจับอุณหภูมิ ใช้อุปกรณ์ที่มีชื่อว่า LM 35 ซึ่งอุปกรณ์ตรวจจับอุณหภูมินี้ให้ Analog Output โดยต้องใช้ Analog-to-digital converter อ่านค่าเข้ามาแล้วคำนวณค่าจึงจะได้ค่าอุณหภูมิ ความแม่นยำในการตรวจจับอุณหภูมิจะขึ้นอยู่กับ Analog-to-digital converter ค่าที่ได้จะเป็นเชิงเส้น (Linear) แต่กราฟที่ได้จะไม่เป็นเส้นตรงเท่าใดนัก อาจจะมีเป็นเส้นโค้งเล็กน้อย โดยสมการที่คำนวณจากอาคูอิน์ คือ

คำนวณได้จาก ณ อุณหภูมิ 20 °C แรงดัน (V) = 14.891(ค่าอุณหภูมิ (°C)) – 20.601 mV

$$\text{แรงดัน (V)} = 14.891(20) - 20.601 \text{ mV}$$

$$\text{แรงดัน (V)} = 277.219 \text{ mV}$$

เมื่ออาคูอิน์ใช้ 10 บิต จะได้ค่า $2^{10} = 1,024$ Step Size และใช้แรงดันตั้งแต่ 0 - 5V

$$\text{จะได้ว่า } 5/1024 = 0.00488 \text{ V}$$

$$0.00488 \text{ V} = 1 \text{ Step Size}$$

$$277.219 \text{ mV} = 56.807 \text{ Step Size}$$

$$\text{ดังนั้น } 56.807 \text{ Step Size} = 20 \text{ }^{\circ}\text{C}$$

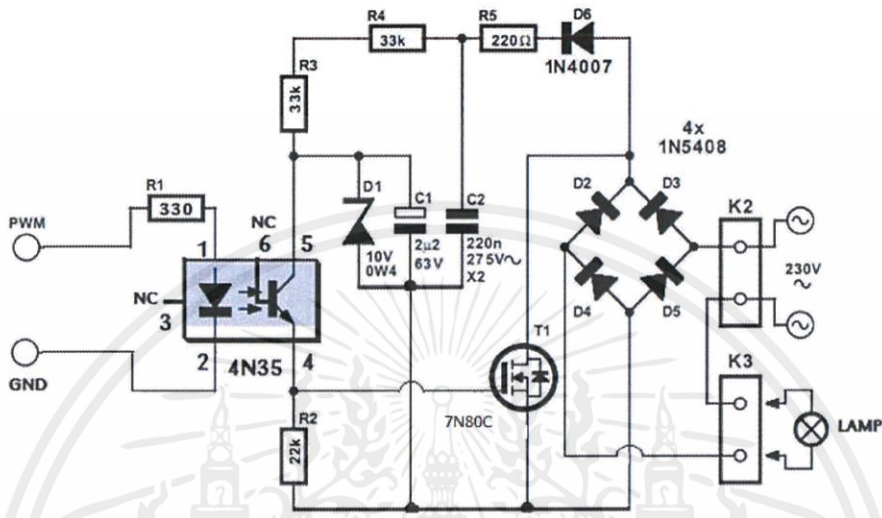
$$1 \text{ Step Size} = 0.35 \text{ }^{\circ}\text{C}$$

เพราะฉะนั้น จะได้สมการคำนวณค่าอุณหภูมิของอาคูอิน์ คือ

$$\text{temp} = \text{analogRead(tempPin)} * 0.35$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแรงดันน้อย ซึ่งจะทำให้ไฟสลัวลง แต่หากมีกระแส (Pulse) ไหลมาจากวงจร Pulse Width Modulation มาก ทำให้ Opto มีการเปล่งแสงมากขึ้น ซึ่งจะทำให้ไฟสว่างเพิ่มขึ้นตามแรงดันที่ส่งเข้ามาในวงจร

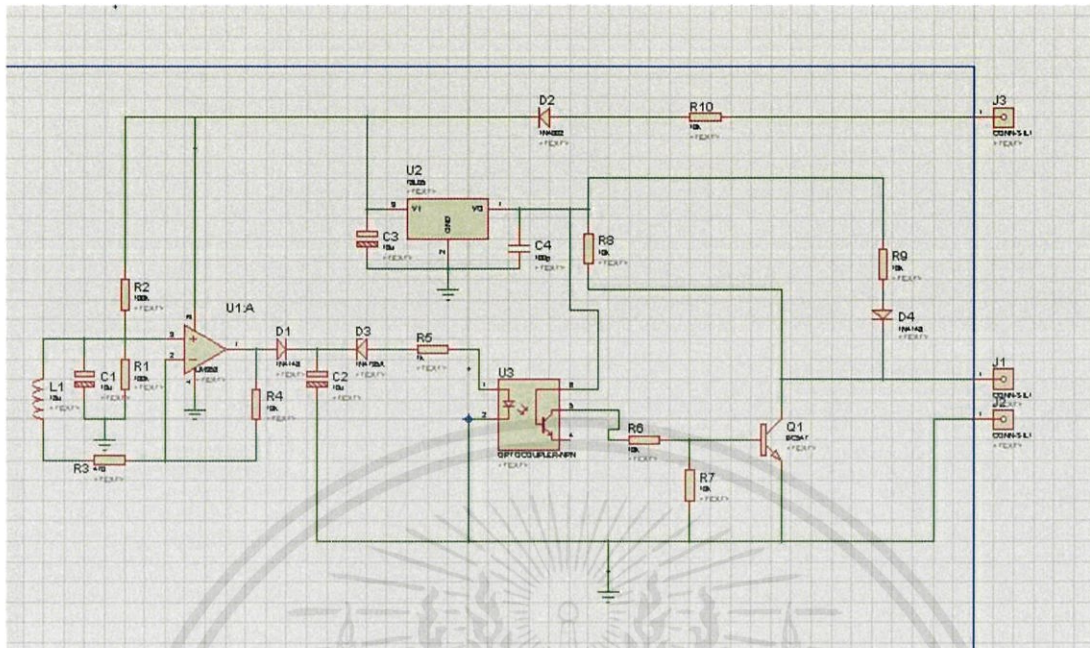


รูปที่ 3.5 วงจรหรี่ไฟโดยใช้เทคนิค Pulse Width Modulation

3.3.4 วงจรตรวจสอบสถานะการเปิดปิดของอุปกรณ์ไฟฟ้า

ในการตรวจสอบสถานะการเปิดปิดของอุปกรณ์ไฟฟ้าที่มีวัตถุประสงค์เพื่อต้องการทราบสถานะปัจจุบันที่แท้จริงของอุปกรณ์ไฟฟ้าที่ถูกสั่งการจากวงจรควบคุมว่าอยู่ในสถานะใด โดยการออกแบบจะใช้หลักการตรวจจับกระแสที่ไหลผ่านอุปกรณ์ไฟฟ้านั้นๆ จากนั้นจะส่งค่าของสถานะการเปิดปิดของอุปกรณ์ไฟฟ้าโดยกำหนดให้ค่า 0 คือ สถานะเปิดเมื่อแรงดันตกคร่อมมีค่าประมาณ 4 V และค่า 1 คือสถานะปิดเมื่อแรงดันตกคร่อมมีค่าเท่ากับ 0 V ไปยังอาตูดิวโนเพื่อส่งสถานะที่แท้จริงไปยังโทรศัพท์มือถือ โดยกำหนดให้มีพารามิเตอร์ (Parameter) ที่สำคัญได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรตรวจสอบสถานะการเปิดปิดของอุปกรณ์ไฟฟ้า

3.4 เครื่องมือที่ใช้ในการทดสอบ

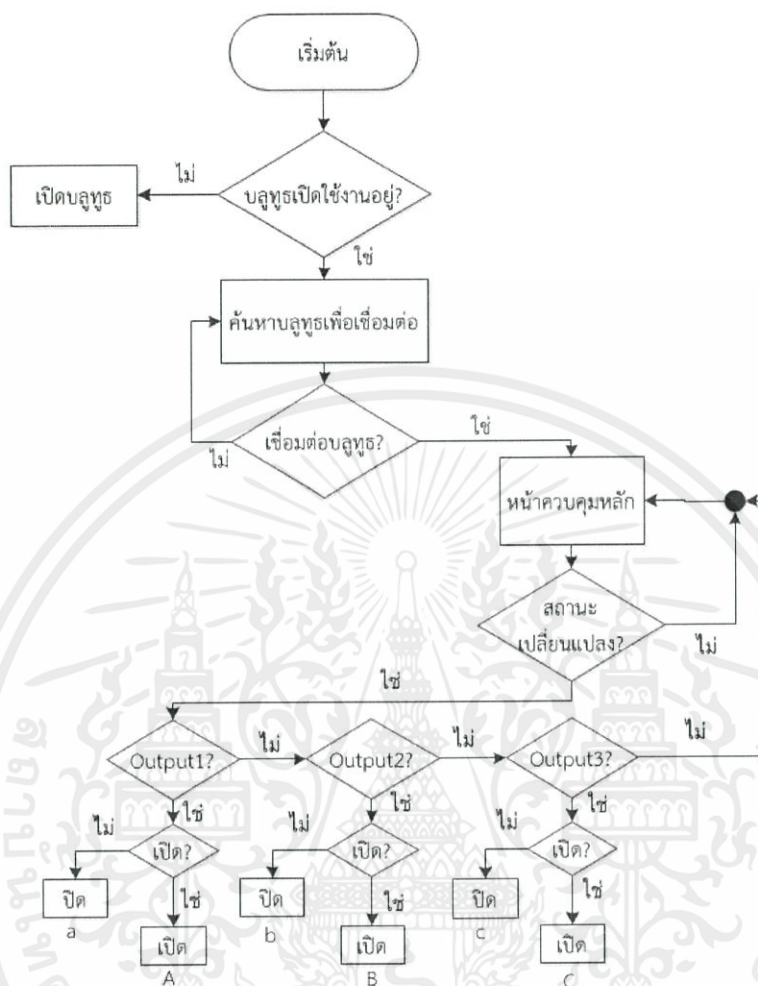
- 1) เครื่องออสซิลโลสโคป (Oscilloscope)
- 2) เครื่องคอมพิวเตอร์
- 3) โปรแกรมอาดูอินโน้ (Arduino Programming)
- 4) โมดูลของอาดูอินโน้ (Arduino Module)
- 5) โทรศัพท์มือถือที่รองรับปฏิบัติการแอนดรอยด์

3.5 Flowchart แสดงการทำงานของระบบ

ในโครงงานนี้จะแบ่ง Flow Chart ออกเป็น 3 กลุ่ม ได้แก่ Flowchart แสดงการทำงานของระบบในภาคการศึกษาที่ 1, Flow Chart แสดงการทำงานของระบบในภาคการศึกษาที่ 2 จะเป็นออกเป็น Flow Chart แสดงการทำงานของแอปพลิเคชัน และ Flow Chart แสดงการทำงานของฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 Flow Chart แสดงการทำงานของระบบในภาคการศึกษาที่ 1



รูปที่ 3.7 Flow Chart ของระบบในภาคการศึกษาที่ 1

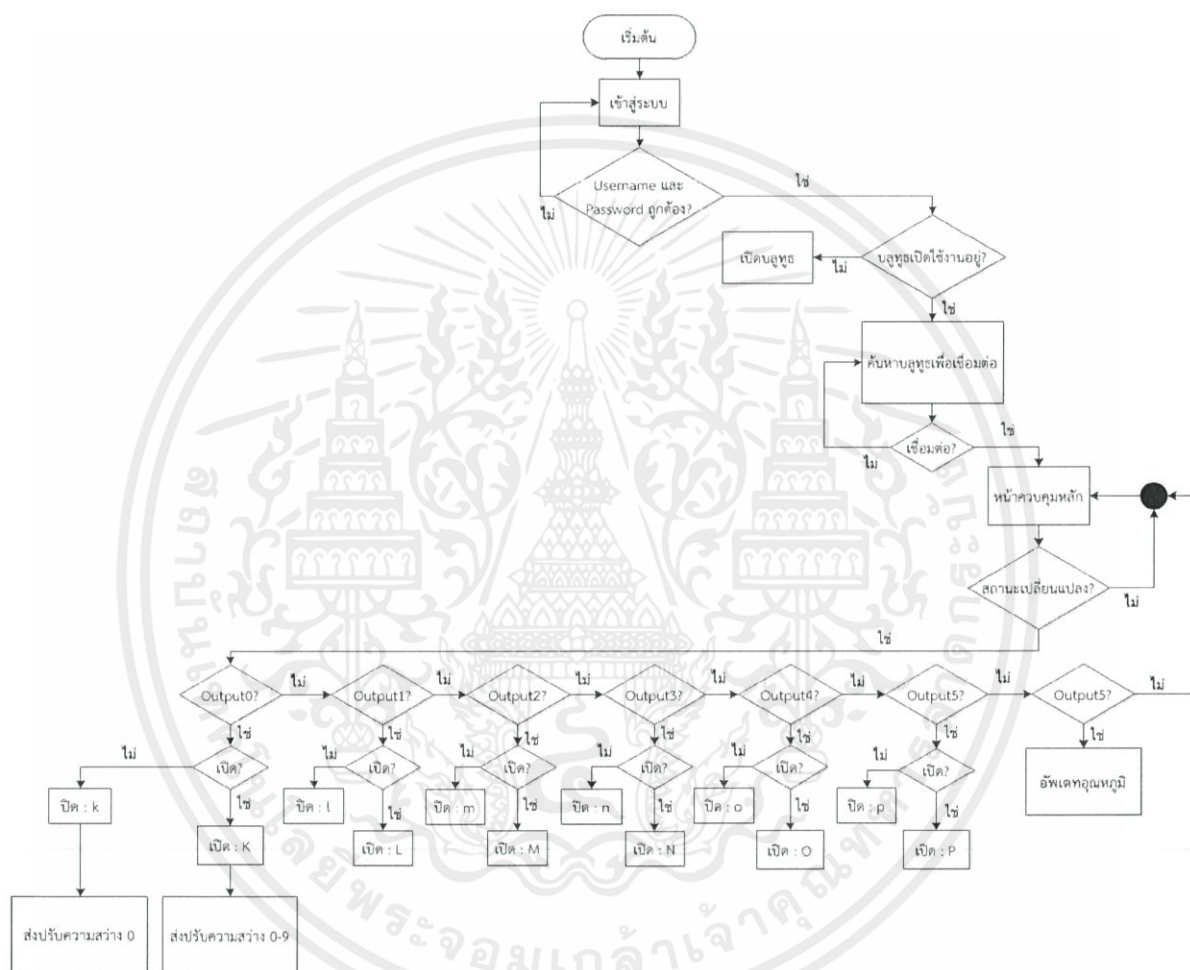
เมื่อเข้าสู่หน้าหลักของแอปพลิเคชัน จะมีการตรวจว่าได้ทำการเปิดบลูทูธในโทรศัพท์มือถือหรือไม่ หากเปิดแล้วจะสามารถเชื่อมต่อกับโมดูลของบลูทูธได้ จึงจะทำการเปิดปิดสถานะของเครื่องใช้ไฟฟ้าได้ โดยกำหนดให้เอาต์พุตตัวที่ 1, 2 และ 3 เป็นหลอดไฟ พัดลม และเครื่องปรับอากาศ ตามลำดับ โดยกำหนดให้มีพารามิเตอร์ (Parameter) ที่สำคัญได้แก่

- 1) A คือ สถานะเปิดของเอาต์พุตที่ 1
- 2) a คือ สถานะปิดของเอาต์พุตที่ 1
- 3) B คือ สถานะเปิดของเอาต์พุตที่ 2
- 4) b คือ สถานะปิดของเอาต์พุตที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) C คือ สถานะเปิดของเอาท์พุทที่ 3
- 6) c คือ สถานะปิดของเอาท์พุทที่ 3

3.5.2 Flow Chart แสดงการทำงานของระบบในส่วนของแอปพลิเคชันในภาคการศึกษาที่ 2



รูปที่ 3.8 Flow Chart แสดงการทำงานของระบบในส่วนของแอปพลิเคชันในภาคการศึกษาที่ 2

เมื่อจะเข้ามาในแอปพลิเคชันจะพบหน้าล็อกอิน โดยต้องใส่ชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) ให้ถูกต้องตามที่ได้กำหนดไว้ โดยผู้ใช้งาน ดังนั้น หากใส่ค่าทั้งสองไม่ตรงกับที่ตั้งค่าไว้ก็จะไม่สามารถ เข้าถึงหน้าหลักของแอปพลิเคชันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

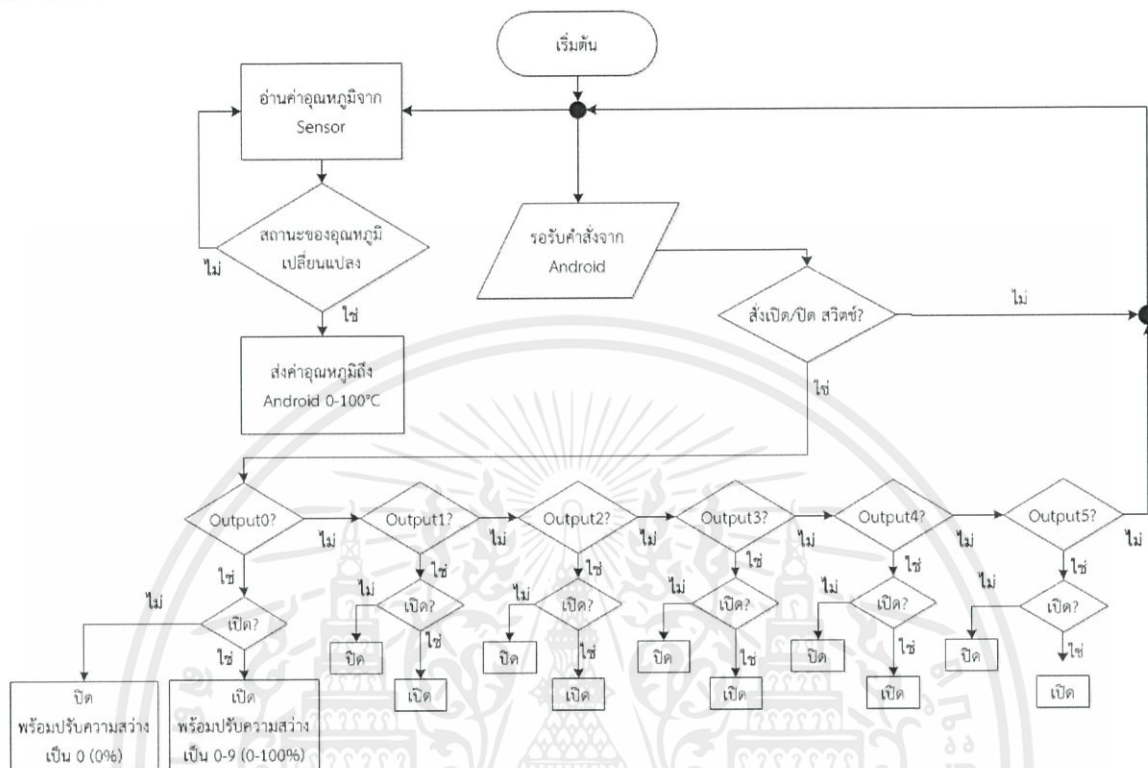
เมื่อเข้าสู่หน้าหลักของแอปพลิเคชัน จะมีการตรวจว่าได้ทำการเปิดบลูทูธในโทรศัพท์มือถือหรือไม่ หากเปิดแล้วจะสามารถเชื่อมต่อกับโมดูลของบลูทูธได้ จึงจะทำการเปิดปิดสถานะของเครื่องใช้ไฟฟ้าได้ โดยกำหนดให้เอาต์พุตตัว 0 เป็นหลอดไฟที่สามารถเปิดและปิดได้แล้วยังสามารถหรี่ไฟได้อีกด้วย โดยกำหนดค่าความสว่างต่ำสุดในการหรี่ไฟเป็นค่า 0 และความสว่างสูงสุดเป็นค่า 9 ส่วนเอาต์พุตตัวที่ 1, 2, 3, 4 และ 5 เป็นหลอดไฟ โทรทัศน์ เครื่องปรับอากาศ และพัดลมตามลำดับ โดยสามารถทำการเปิดและปิดได้ นอกจากนี้ยังมีส่วนที่แสดงค่าอุณหภูมิเป็นองศาเซลเซียสที่ตรวจจับได้ในบริเวณรอบๆอีกด้วย โดยมีพารามิเตอร์ที่สำคัญได้แก่

- 1) K คือ สถานะเปิดของเอาต์พุตที่ 0
- 2) k คือ สถานะปิดของเอาต์พุตที่ 0
- 3) L คือ สถานะเปิดของเอาต์พุตที่ 1
- 4) l คือ สถานะปิดของเอาต์พุตที่ 1
- 5) M คือ สถานะเปิดของเอาต์พุตที่ 2
- 6) m คือ สถานะปิดของเอาต์พุตที่ 2
- 7) N คือ สถานะเปิดของเอาต์พุตที่ 3
- 8) n คือ สถานะปิดของเอาต์พุตที่ 3
- 9) O คือ สถานะเปิดของเอาต์พุตที่ 4
- 10) o คือ สถานะปิดของเอาต์พุตที่ 4
- 11) P คือ สถานะเปิดของเอาต์พุตที่ 5
- 12) p คือ สถานะปิดของเอาต์พุตที่ 5
- 13) ค่า 0 คือ แรงดันต่ำสุดที่ถูกปรับจากโทรศัพท์มือถือ
- 14) ค่า 9 คือ แรงดันสูงสุดที่ถูกปรับจากโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 Flow Chart แสดงการทำงานของระบบในส่วนของฮาร์ดแวร์ในภาค

การศึกษาที่ 2



รูปที่ 3.9 Flow Chart แสดงการทำงานของระบบในส่วนของฮาร์ดแวร์ในภาคการศึกษาที่ 2

เมื่อไมโครคอนโทรลเลอร์ได้รับคำสั่งจากแอนดรอยด์ในการสั่งเปิดหรือปิดไฟ ก็ส่งสัญญาณไปยังอุปกรณ์ไฟฟ้าทั้งหมด หากเอาต์พุต 0 ถูกสั่งให้เปิด ไมโครคอนโทรลเลอร์จะสั่ง ON พร้อมปรับค่าความสว่างที่ส่งมาเป็น 0-9 หรือ 0-100% หากถูกสั่งให้ปิด ไมโครคอนโทรลเลอร์จะสั่ง OFF พร้อมปรับค่าความสว่างที่ส่งมาเป็น 0 ส่วนเอาต์พุตที่ 1, 2, 3, 4 และ 5 สามารถสั่งเปิดและปิดได้ ส่วนตัวตรวจจับอุณหภูมิมีการทำงานคือ อ่านค่าจากอุณหภูมิจาก เมื่อมีการเปลี่ยนแปลงขององศาเซลเซียสที่วัดได้ ก็จะส่งค่าของอุณหภูมิคือ 0-100 องศาเซลเซียส กลับไปยังโทรศัพท์มือถือในระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการออกแบบและการทดสอบ

4.1 บทนำ

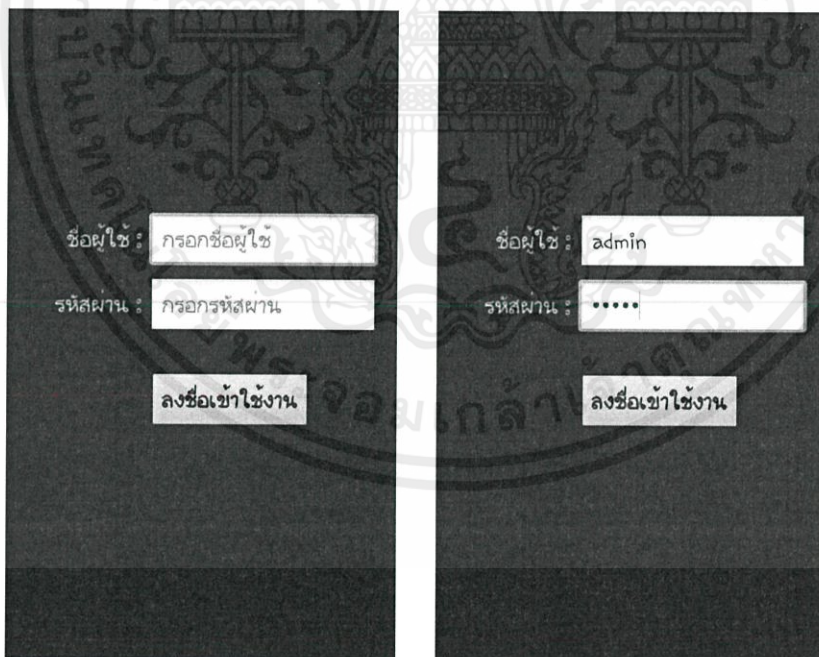
ในบทนี้จะกล่าวถึงผลการออกแบบและการทดสอบของแอปพลิเคชันและไมโครคอนโทรลเลอร์ที่ได้สร้างและออกแบบที่ได้กล่าวมาแล้วในบทที่ 3

4.2 ผลการทดสอบในส่วนของแอปพลิเคชัน

4.2.1 หน้าลงชื่อเข้าใช้ (Log-in Page)

เมื่อจะเข้ามาในแอปพลิเคชัน ต้องเจอหน้าล็อกอิน โดยต้องใส่ชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) ดังนั้นผู้ที่ไม่มีรหัสผ่านจะไม่สามารถเข้าถึงหน้าหลักของแอปพลิเคชันได้ เพื่อความปลอดภัยของระบบ

ในที่นี้จะใช้คำสำหรับ Username คือ admin และ Password คือ 15672 จึงจะผ่านเข้าไปยังหน้าหลักของแอปพลิเคชันได้



รูปที่ 4.1 หน้าล็อกอินของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 หน้าหลักสำหรับควบคุม (Main Control Page)

หน้าหลักของแอปพลิเคชันซึ่งจะมีหน้าที่ควบคุมอุปกรณ์ไฟฟ้าที่กำหนดไว้ คือ 6 ดิจิทัลเอาต์พุตซึ่งได้แก่ พัดลม เครื่องปรับอากาศ โทรทัศน์ และหลอดไฟ โดยมีอีก 1 ดิจิทัลเอาต์พุตที่สามารถปรับความสว่างของหลอดไฟได้ และเพิ่มการตรวจจับและวัดอุณหภูมิ 1 ตัว (1 อนุาล็อกเอาต์พุต)



รูปที่ 4.2 หน้าหลักของแอปพลิเคชัน

ในการเปิดและปิดอุปกรณ์ไฟฟ้าเป็น 5 ตัว (5 ดิจิทัลเอาต์พุต) ทำหน้าที่เป็นสวิตช์ในการเปิดและปิดไฟ โดยมีรีเลย์ในการควบคุมด้วย นอกจากนี้ หลอดไฟอีก 1 ดวงยังสามารถควบคุมการหรี่ไฟได้โดยใช้หลักการของการปรับความกว้างของสัญญาณพัลส์ (Pulse Width Modulation) เป็นตัวปรับความสว่างของหลอดไฟ โดยจากผลการทดลองสามารถเปิดและปิดอุปกรณ์ไฟฟ้าได้ทุกตัว และในการปรับความสว่างของหลอดไฟโดยกำหนดค่าในการปรับเป็นค่า 0 ถึง 9 (0-100%) โดยผลการทดสอบสามารถปรับความสว่างจากต่ำสุดคือแสงไฟดับไปและความสว่างมากที่สุดคือไฟสว่างที่สุดได้

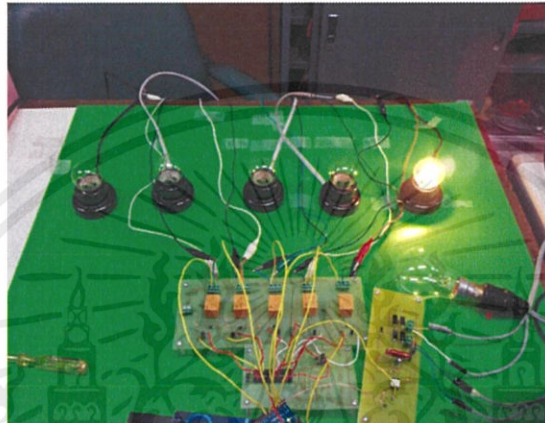
ในการวัดอุณหภูมิที่แสดงบนแอปพลิเคชัน จะแสดงอุณหภูมิตามที่วัดได้และจะมีการเปลี่ยนแปลงค่าของอุณหภูมิที่แสดงบนโทรศัพท์มือถือก็ต่อเมื่อมีการเปลี่ยนแปลงของอุณหภูมิภายนอกที่เซ็นเซอร์ตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบในส่วนของไมโครคอนโทรลเลอร์

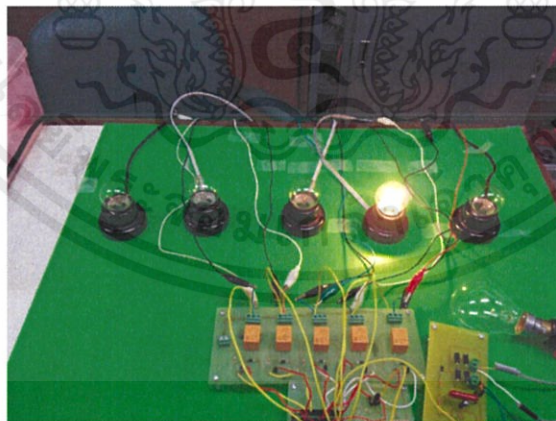
4.3.1 การควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์

ในการควบคุมอุปกรณ์ไฟฟ้าโดยการเปิดและปิดเสมือนการทำงานของสวิตช์ จะมีการควบคุมโดยรับคำสั่งจากโทรศัพท์มือถือ โดยมีผลการทดลองคือ เมื่อรับค่าสถานะเปิดจากเอาร์ทพุท 1 จะมีการส่งค่า L จากโทรศัพท์มือถือส่งค่าไปให้อาตูดิโนเพื่อควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 4.3 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 1 โดยการส่งสถานะ L

เมื่อรับค่าสถานะเปิดจากเอาร์ทพุท 2 จะมีการส่งค่า M จากโทรศัพท์มือถือส่งค่าไปให้อาตูดิโนเพื่อควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 4.4 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 2 โดยการส่งสถานะ M

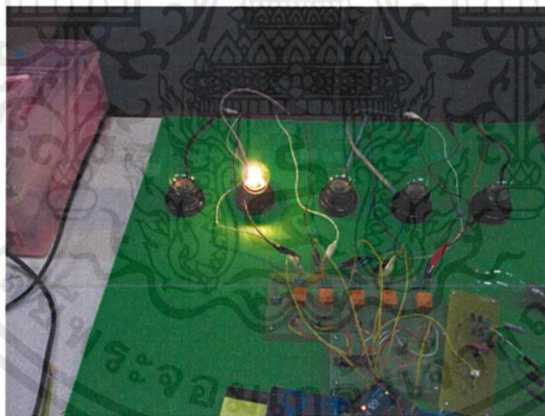
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรับค่าสถานะเปิดจากเอาต์พุต 3 จะมีการส่งค่า N จากโทรศัพท์มือถือส่งค่าไปให้อาตุน์โนเพื่อควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 4.5 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 3 โดยการส่งสถานะ N

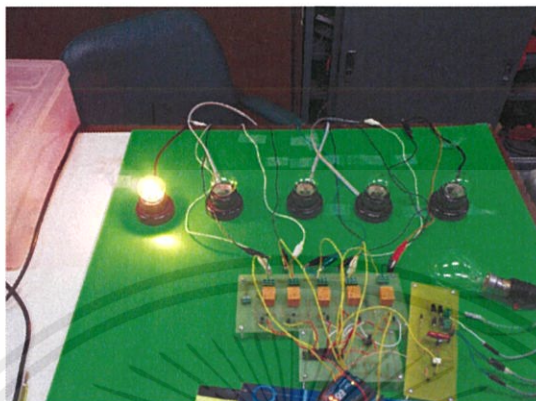
เมื่อรับค่าสถานะเปิดจากเอาต์พุต 4 จะมีการส่งค่า O จากโทรศัพท์มือถือส่งค่าไปให้อาตุน์โนเพื่อควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 4.6 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 4 โดยการส่งสถานะ O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรับค่าสถานะเปิดจากเอาต์พุต 5 จะมีการส่งค่า P จากโทรศัพท์มือถือส่งค่าไปให้อาอดูอินเพื่อควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 4.7 สถานะเปิดของอุปกรณ์ไฟฟ้าตัวที่ 5 โดยการส่งสถานะ P

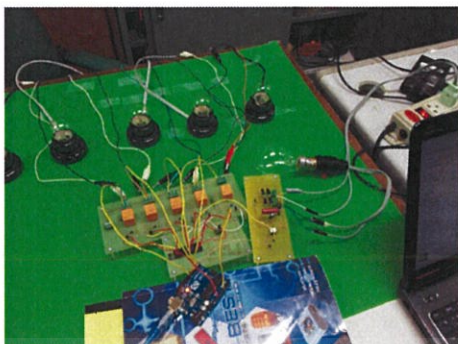
นอกจากนี้ เมื่อรับค่าสถานะเปิดจากเอาต์พุต 0 จะมีการส่งค่า K จากโทรศัพท์มือถือส่งค่าไปให้อาอดูอินเพื่อควบคุมอุปกรณ์ไฟฟ้า และยังมีการปรับค่าความสว่างของหลอดไฟโดยใช้หลักการ Pulse Width Modulation โดยกำหนดให้ค่า 0 คือ ความสว่างต่ำสุดหรือแรงดันต่ำสุดที่ถูกปรับจากโทรศัพท์มือถือ ไปจนถึงค่า 9 นั่นคือ ความสว่างสูงสุดหรือแรงดันสูงสุดที่ถูกปรับจากโทรศัพท์มือถือ ซึ่งจะกล่าวถึงในหัวข้อถัดไป

4.3.2 การปรับความสว่างของหลอดไฟ

เมื่อรวบรวมวงจรแล้ว จะสามารถปรับความสว่างของหลอดไฟได้ตามระดับที่กำหนดไว้ เมื่อมีการปรับระดับความสว่างที่ถูกส่งการมาจากโทรศัพท์มือถือไปยังไมโครคอนโทรลเลอร์ผ่านบลูทูธ ดังนี้

ความสว่างของหลอดไฟเมื่อมีการส่งค่า 0 จากโทรศัพท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาอดูอินที่กำหนดไว้ คือ ค่าระดับที่ 0.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 0 จากทรอสต์พท์มือถือ

ความสว่างของหลอดไฟเมื่อมีการส่งค่า 1 จากทรอสต์พท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูโนที่กำหนดไว้ คือ ค่าระดับที่ 10.00



รูปที่ 4.9 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 1 จากทรอสต์พท์มือถือ

ความสว่างของหลอดไฟเมื่อมีการส่งค่า 2 จากทรอสต์พท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูโนที่กำหนดไว้ คือ ค่าระดับที่ 31.87



รูปที่ 4.10 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 2 จากทรอสต์พท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่สามารถนำออกจำหน่ายหรือใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสว่างของหลอดไฟเมื่อมีการส่งค่า 0 จากโทรศัพท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูโนที่กำหนดไว้ คือ ค่าระดับที่ 63.75



รูปที่ 4.11 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 3 จากโทรศัพท์มือถือ

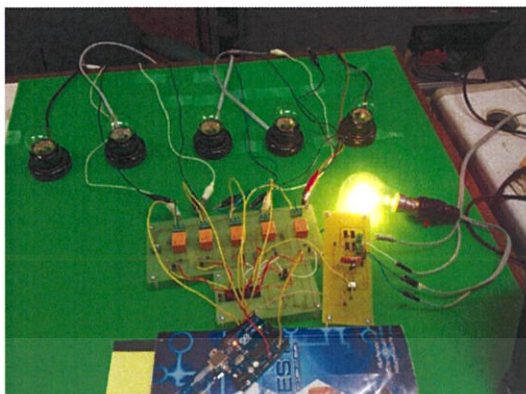
ความสว่างของหลอดไฟเมื่อมีการส่งค่า 4 จากโทรศัพท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูโนที่กำหนดไว้ คือ ค่าระดับที่ 95.62



รูปที่ 4.12 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 4 จากโทรศัพท์มือถือ

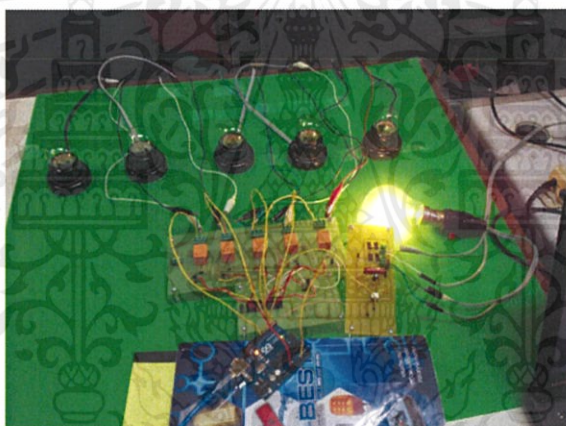
ความสว่างของหลอดไฟเมื่อมีการส่งค่า 5 จากโทรศัพท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูโนที่กำหนดไว้ คือ ค่าระดับที่ 127.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 5 จากทรานซิสเตอร์มือถือ

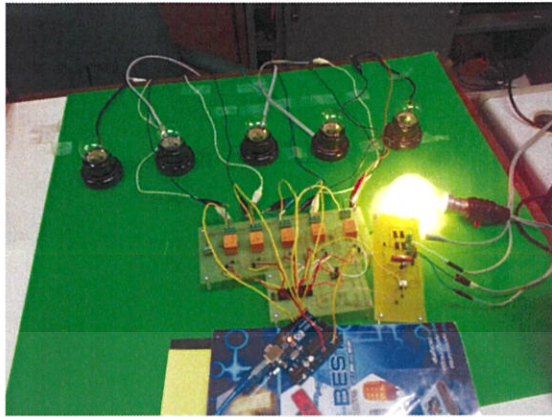
ความสว่างของหลอดไฟเมื่อมีการส่งค่า 6 จากทรานซิสเตอร์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูอิโนที่กำหนดไว้ คือ ค่าระดับที่ 159.37



รูปที่ 4.14 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 6 จากทรานซิสเตอร์มือถือ

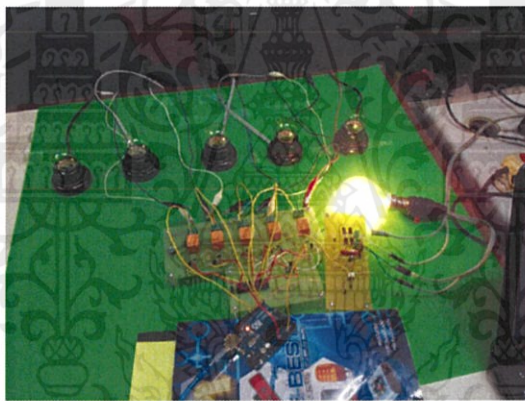
ความสว่างของหลอดไฟเมื่อมีการส่งค่า 7 จากทรานซิสเตอร์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูอิโนที่กำหนดไว้ คือ ค่าระดับที่ 191.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 7 จากโทรศัพท์มือถือ

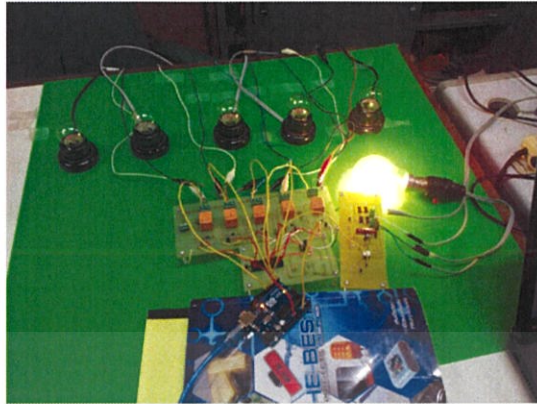
ความสว่างของหลอดไฟเมื่อมีการส่งค่า 8 จากโทรศัพท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูโนที่กำหนดไว้ คือ ค่าระดับที่ 223.12



รูปที่ 4.16 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 8 จากโทรศัพท์มือถือ

ความสว่างของหลอดไฟเมื่อมีการส่งค่า 9 จากโทรศัพท์มือถือ จะเทียบเท่าได้กับการส่งระดับความสว่างจากอาคูโนที่กำหนดไว้ คือ ค่าระดับที่ 255.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

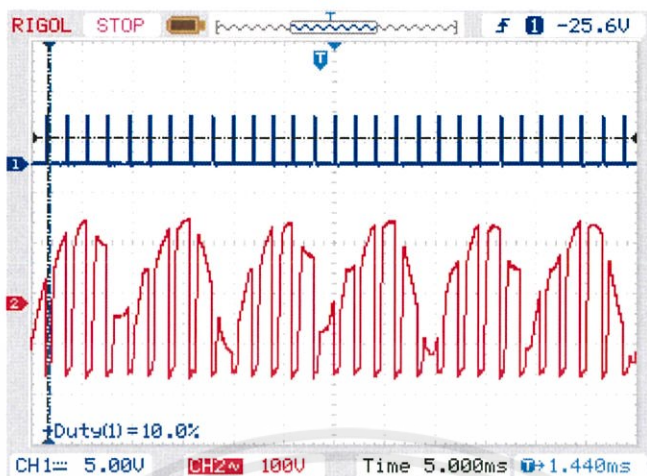


รูปที่ 4.17 ความสว่างของหลอดไฟเมื่อมีการปรับค่า 9 จากโทรศัพท์มือถือ

ในที่นี้ จะนำสัญญาณ Pulse Width Modulation จากไมโครคอนโทรลเลอร์ ซึ่งเป็นการกำหนด Duty Cycle ให้สัญญาณพัลส์ออกมาจากไมโครคอนโทรลเลอร์โดยตรง มาควบคุม มอสเฟตที่ไปขับเคลื่อนความสว่างของหลอดไฟ ซึ่งจะไม่มีการสร้างสัญญาณและวงจร Pulse Width Modulation ในการวัดความสว่างของหลอดไฟจะทดลองกับเครื่องออสซิลโลสโคปโดย กำหนดให้ Channel 1 สัญญาณสีน้ำเงินเป็นสัญญาณของ Pulse Width Modulation และ Channel 2 สัญญาณสีแดง เป็นสัญญาณ Full Wave Rectifier

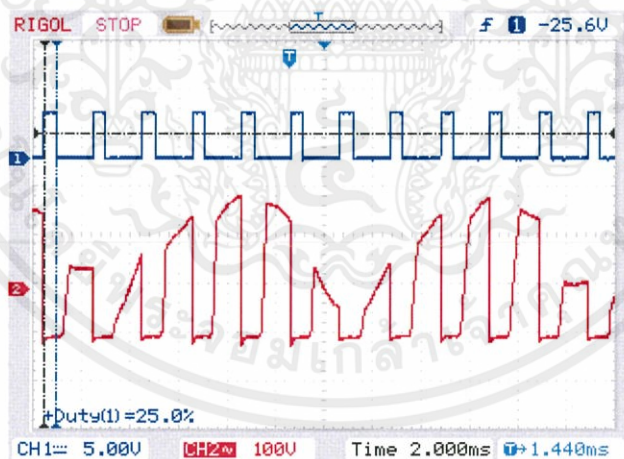
1) จากผลการทดลองที่ได้เป็นการวัดสัญญาณอินพุท นั่นคือ สัญญาณ Pulse Width Modulation ที่ป้อนจากไมโครคอนโทรลเลอร์เทียบสัญญาณ Full Wave Rectifier โดยช่วงสัญญาณขาขึ้นของ Pulse Width Modulation จะเป็นช่วงที่นำกระแสที่ขับเคลื่อนความ สว่างของหลอดไฟ ซึ่งในที่นี้จะมีช่วงที่ค่อนข้างแคบ จากกราฟจะเห็นว่า Duty Cycle ของ Pulse Width Modulation คือ 10% ซึ่งทำให้หลอดไฟหรี่จนใกล้จะดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 Pulse Width Modulation ที่มี Duty Cycle เป็น 10%

2) จากผลการทดลองที่ได้เป็นการวัดสัญญาณอินพุต นั่นคือ สัญญาณ Pulse Width Modulation ที่ป้อนจากไมโครคอนโทรลเลอร์เทียบกับสัญญาณ Full Wave Rectifier โดยช่วงสัญญาณขาขึ้นของ Pulse Width Modulation จะเป็นช่วงที่นำกระแสที่ขับเคลื่อนความสว่างของหลอดไฟ ซึ่งในที่นี้จะมีช่วงที่มีความกว้างเพิ่มขึ้น จากกราฟจะเห็นว่า Duty Cycle ของ Pulse Width Modulation คือ 25% ทำให้มีหลอดไฟเริ่มมีแสงสว่างขึ้นบ้างเล็กน้อย



รูปที่ 4.19 Pulse Width Modulation ที่มี Duty Cycle เป็น 25%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) จากผลการทดลองที่ได้เป็นการวัดสัญญาณอินพุต นั่นคือ สัญญาณ Pulse Width Modulation ที่ป้อนจากไมโครคอนโทรลเลอร์เทียบกับสัญญาณ Full Wave Rectifier โดยช่วงสัญญาณขาขึ้นของ Pulse Width Modulation จะเป็นช่วงที่นำกระแสที่ขับเคลื่อนความสว่างของหลอดไฟ ซึ่งในที่นี้จะมีช่วงที่มีขนาดเท่ากับช่วงที่แรงดันตกคร่อมเท่ากับ 0 จากกราฟจะเห็นว่า Duty Cycle ของ Pulse Width Modulation คือ 50% ทำให้มีหลอดไฟมีแสงสว่างกลางๆ



รูปที่ 4.20 Pulse Width Modulation ที่มี Duty Cycle เป็น 50%

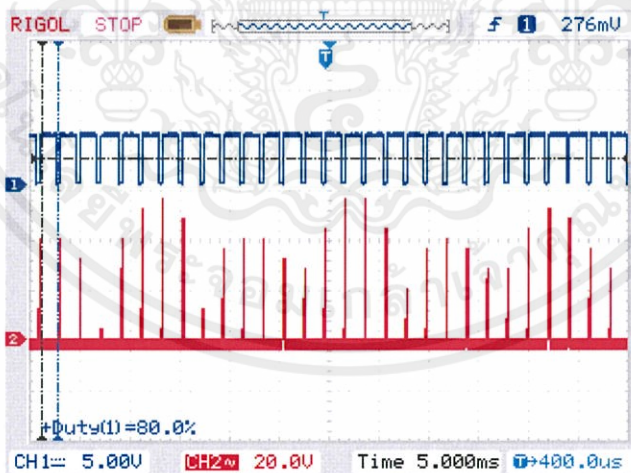
4) จากผลการทดลองที่ได้เป็นการวัดสัญญาณ input นั่นคือ สัญญาณ Pulse Width Modulation ที่ป้อนจากไมโครคอนโทรลเลอร์เทียบกับสัญญาณ Full Wave Rectifier โดยช่วงสัญญาณขาขึ้นของ Pulse Width Modulation จะเป็นช่วงที่นำกระแสที่ขับเคลื่อนความสว่างของหลอดไฟ ซึ่งในที่นี้จะมีช่วงที่ค่อนข้างกว้าง จากกราฟจะเห็นว่า Duty Cycle ของ Pulse Width Modulation คือ 75% ทำให้หลอดไฟสว่างเกือบสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 Pulse Width Modulation ที่มี Duty Cycle เป็น 75%

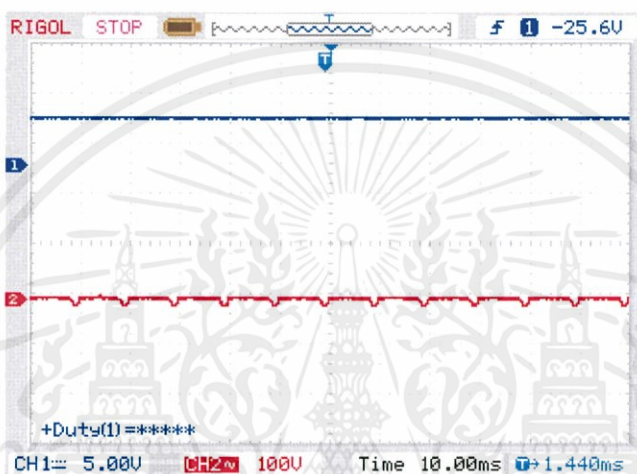
5) จากผลการทดลองที่ได้เป็นการวัดสัญญาณอินพุต นั่นคือ สัญญาณ Pulse Width Modulation ที่ป้อนจากไมโครคอนโทรลเลอร์เทียบกับสัญญาณ Full Wave Rectifier โดยช่วงสัญญาณขาขึ้นของ Pulse Width Modulation จะเป็นช่วงที่นำกระแสที่ขับเคลื่อนความสว่างของหลอดไฟ ซึ่งในที่นี้จะมีช่วงที่ค่อนข้างกว้าง เนื่องจากหลอดไฟสว่างเกือบเต็ม Cycle จากกราฟจะเห็นว่า Duty Cycle ของ Pulse Width Modulation คือ 80% จึงทำให้หลอดไฟสว่างเกือบสูงสุด



รูปที่ 4.22 Pulse Width Modulation ที่มี Duty Cycle เป็น 80%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) จากผลการทดลองที่ได้เป็นการวัดสัญญาณอินพุต นั่นคือ สัญญาณ Pulse Width Modulation ที่ป้อนจากไมโครคอนโทรลเลอร์เทียบกับสัญญาณ Full Wave Rectifier โดยช่วงสัญญาณขาขึ้นของ Pulse Width Modulation จะเป็นช่วงที่นำกระแสที่ขับเคลื่อนความสว่างของหลอดไฟ ซึ่งจะไม่มีกระแสไหล เพราะฉะนั้นจะมีแรงดันตกคร่อมสูงที่สุด จากกราฟจะเห็นว่า Duty Cycle ของ Pulse Width Modulation คือ 100% จึงทำให้หลอดไฟสว่างสูงสุด



รูปที่ 4.23 Pulse Width Modulation ที่มี Duty Cycle เป็น 100%

4.3.3 การวัดอุณหภูมิ

ผลการทดลองจากการรับค่าอุณหภูมิโดยมีเปรียบเทียบค่าระหว่างอุณหภูมิ ($^{\circ}\text{C}$) และแรงดัน (V) โดยใช้เครื่องออสซิลโลสโคปเป็นเครื่องมืออ่านค่า ในที่นี้จะเก็บค่าแรงดันที่อุณหภูมิ ตั้งแต่ 0 – 90 องศาเซลเซียส และเพิ่มค่าของอุณหภูมิทีละ 5 องศาเซลเซียส โดยจะเก็บค่าแรงดันทั้งหมด 5 ครั้ง แล้วจึงนำมาหาค่าแรงดันเฉลี่ย ณ อุณหภูมินั้นๆ

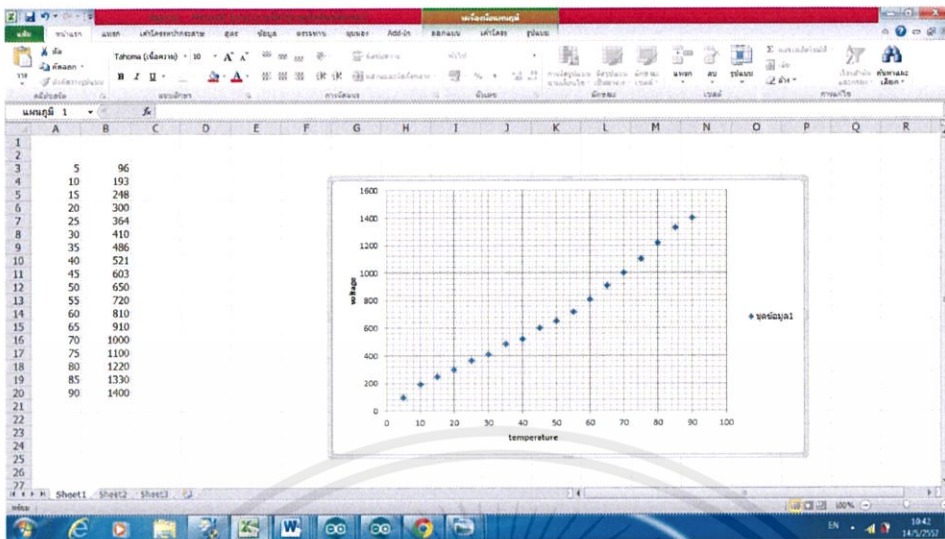
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทดลองค่าแรงดันที่วัดได้ทั้ง 3 ครั้งและแรงดันเฉลี่ยเปรียบเทียบกับค่าอุณหภูมิ

อุณหภูมิ (°C)	แรงดันที่วัดได้ ครั้งที่ 1 (mV)	แรงดันที่วัดได้ ครั้งที่ 2 (mV)	แรงดันที่วัดได้ ครั้งที่ 3 (mV)	แรงดันเฉลี่ย (mV)
0	0	0	0	0
5	99	93	96	96
10	196	193	190	193
15	242	260	242	248
20	300	302	298	300
25	344	388	360	364
30	406	414	410	410
35	480	490	488	486
40	510	524	529	521
45	600	605	604	603
50	650	670	630	650
55	700	740	720	720
60	780	830	820	810
65	903	906	921	910
70	1010	999	991	1000
75	1100	1120	1080	1100
80	1170	1230	1260	1220
85	1330	1260	1400	1330
90	1325	1425	1450	1400

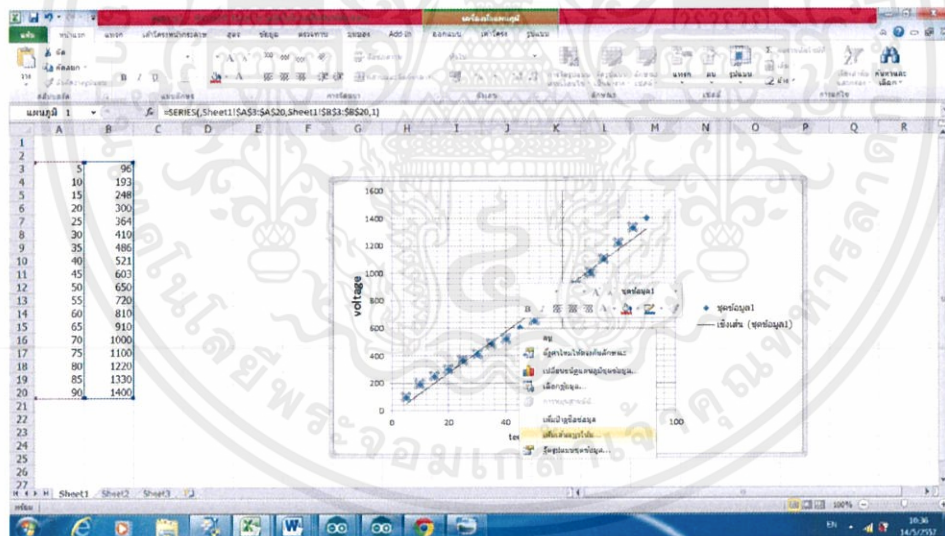
นำแรงดันเฉลี่ยที่วัดได้มาสร้างกราฟโดยโปรแกรม Microsoft Excel ให้แรงดันเฉลี่ยเป็นข้อมูลแกน y และอุณหภูมิเป็นแกน x โดยมีวิธีสร้างแนวโน้มความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 การพล็อตความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ

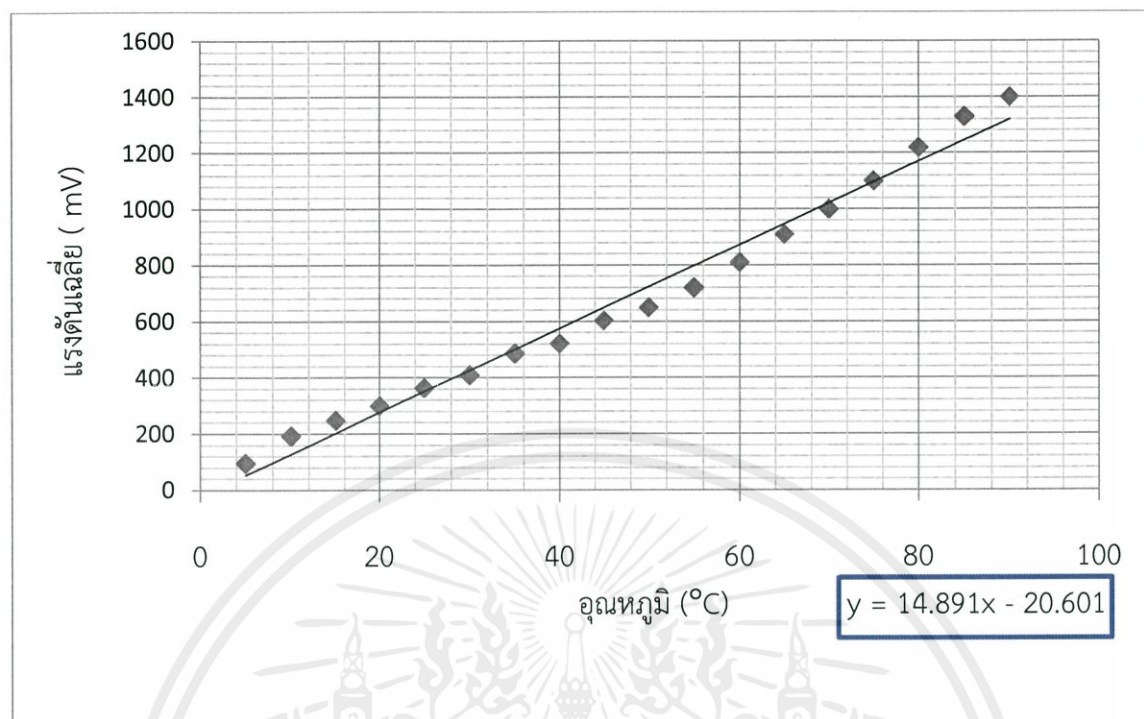
การหาเส้นตรงที่เป็นแนวโน้มของความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ โดยอันดับแรก คลิกขวาที่จุดที่ถูกพล็อตในกราฟ จากนั้นให้เลือกเพิ่มเส้นแนวโน้ม



รูปที่ 4.25 กราฟแสดงเส้นแนวโน้มของกราฟความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ

ในการหาสมการเส้นตรงของเส้นแนวโน้มของกราฟความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ ทำได้โดยเลือกชนิดของแนวโน้มให้เป็นเชิงเส้น และเลือกการแสดงผลการบนแผนภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 กราฟความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ และสมการเส้นตรง

การคำนวณหาอุณหภูมิที่คำนวณมาจากแรงดันเฉลี่ยที่วัดได้ จะได้สมการดังนี้

$$T(^{\circ}\text{C}) = \frac{V_{av}(\text{mV}) \times 0.35}{4.88}$$

ตารางที่ 4.2 เปรียบเทียบค่าอุณหภูมิจากสมการและค่าความคลาดเคลื่อน

อุณหภูมิ ($^{\circ}\text{C}$)	แรงดันเฉลี่ย (mV)	อุณหภูมิที่คำนวณได้จากสมการ ($^{\circ}\text{C}$)	ค่าความคลาดเคลื่อน ($^{\circ}\text{C}$)	ค่าความคลาดเคลื่อน (%)
0	0	0	0	0
5	96	6.88	1.88	37.60
10	193	13.80	3.80	38.00
15	248	17.78	2.78	18.53
20	300	21.51	2.78	7.55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์อื่น การค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 เปรียบเทียบค่าอุณหภูมิจากสมการและค่าความคลาดเคลื่อน (ต่อ)

อุณหภูมิ (°C)	แรงดันเฉลี่ย (mV)	อุณหภูมิที่คำนวณได้จากสมการ (°C)	ค่าความคลาดเคลื่อน (°C)	ค่าความคลาดเคลื่อน (%)
25	364	26.10	1.10	4.40
30	410	29.40	0.60	2.00
35	486	34.84	0.14	0.40
40	521	37.37	2.63	6.57
45	603	43.25	1.75	3.89
50	650	46.63	3.38	6.76
55	720	51.64	3.36	6.11
60	810	58.09	1.91	3.18
65	910	65.27	0.27	0.41
70	1000	71.71	1.71	2.44
75	1100	78.89	3.89	5.19
80	1220	87.50	7.50	9.37
85	1330	95.39	10.39	12.22
90	1400	100.41	10.41	11.57

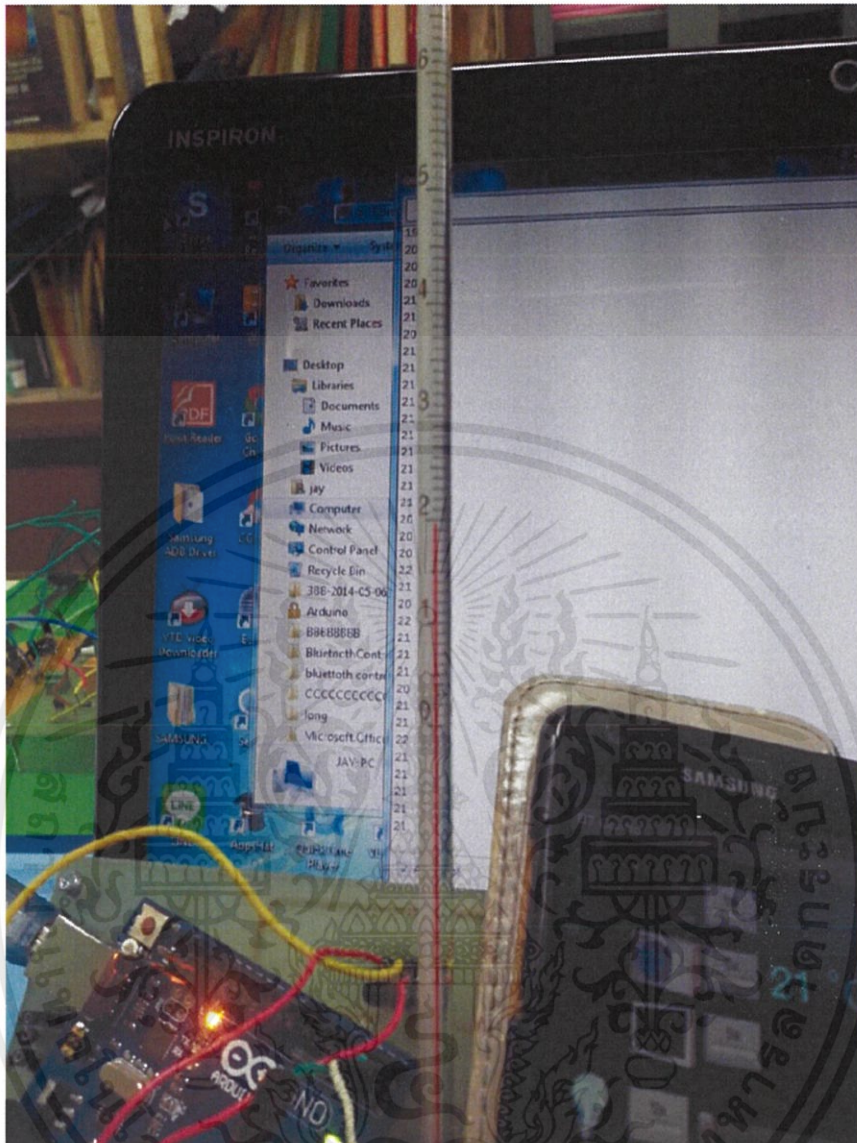
ระบบจะมีความคลาดเคลื่อนเฉลี่ย คือ 9.79%

เมื่อทดสอบการวัดอุณหภูมิที่ 20 °C บนเทอร์โมมิเตอร์ ออดิโอโนคำนวณค่าจากเซ็นเซอร์เพื่อหาความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ จากตารางที่ 4.2 ณ อุณหภูมิ 20 °C มีแรงดันเฉลี่ย 300 mV โดยมีการคำนวณดังต่อไปนี้

$$T(^{\circ}C) = \frac{300 \times 0.35}{4.88}$$

$$T(^{\circ}C) = 21.51^{\circ}C$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



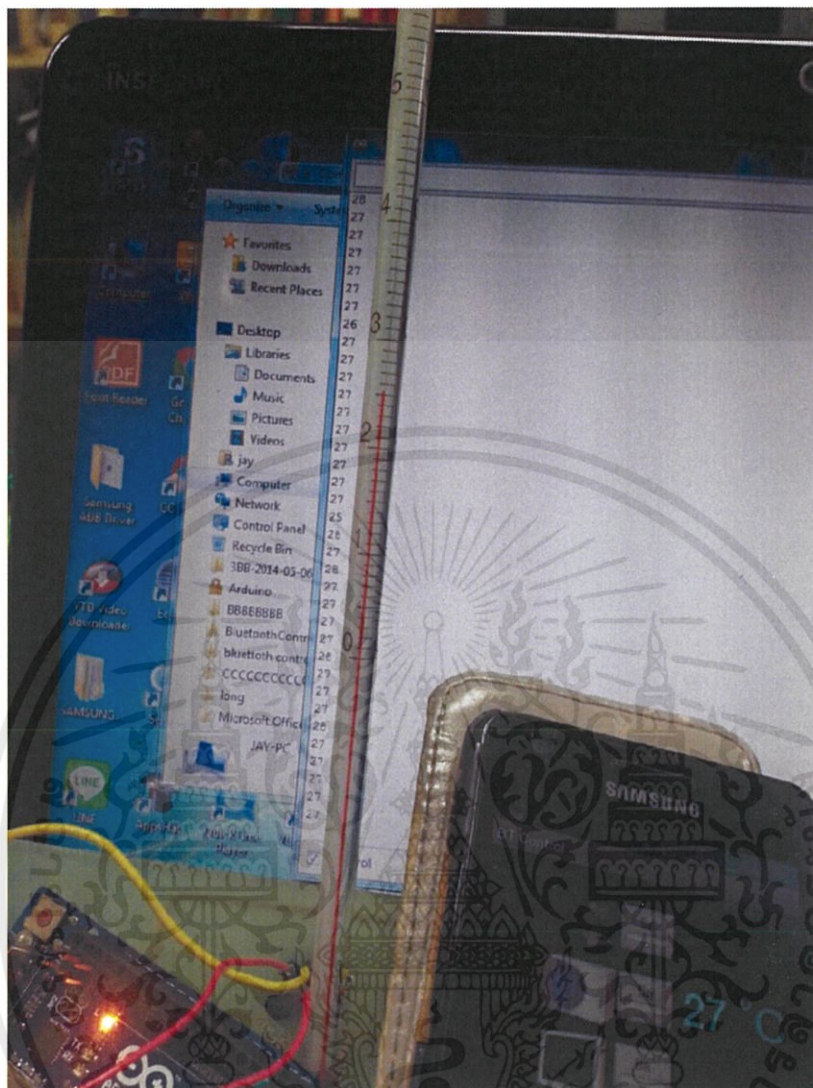
รูปที่ 4.29 การแสดงค่ากับอุณหภูมิบน Serial Monitor และหน้าแอปพลิเคชันที่อุณหภูมิ 20 °C

เมื่อทดสอบการวัดอุณหภูมิที่ 25 °C บนเทอร์โมมิเตอร์ อาดูอิโน้คำนวณค่าจากเซ็นเซอร์เพื่อหาความสัมพันธ์ระหว่างแรงดันเฉลี่ยและอุณหภูมิ จากตารางที่ 4.2 ณ อุณหภูมิ 25 °C มีแรงดันเฉลี่ย 364 mV โดยมีการคำนวณดังต่อไปนี้

$$T(^{\circ}\text{C}) = \frac{364 \times 0.35}{4.88}$$

$$T(^{\circ}\text{C}) = 26.10$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 การแสดงค่ากับอุณหภูมิบน Serial Monitor และหน้าแอปพลิเคชันที่อุณหภูมิ 25 °C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลองและภาพรวมของโครงการทั้งหมด

จากผลการทดลองนั้น สามารถสร้างสร้างแอปพลิเคชันบนโทรศัพท์มือถือโดยเพิ่มสวิตซ์ของการเปิดและปิดอุปกรณ์ไฟฟ้าเป็น 5 ตัว (5 ดิจิทัลอินพุท) ได้แก่ พัฒลม เครื่องปรับอากาศ โทรทัศน์ และหลอดไฟ และยังมี การปรับความสว่างของหลอดไฟได้อีก 1 ตัว (1 ดิจิทัลอินพุท) และเพิ่มการตรวจจับและวัดอุณหภูมิ 1 ตัว (1 อนุาล็อกอินพุท (Analog Input)) นอกจากนี้ยังสามารถมีการลงชื่อเข้าใช้ (Log-in) ก่อนเข้าหน้าหลัก เพื่อความปลอดภัยของระบบได้เป็นอย่างดี

การควบคุมอุปกรณ์ในไมโครคอนโทรลเลอร์สามารถสั่งการควบคุมรีเลย์ได้ทุกอุปกรณ์ ซึ่งมีหน้าที่เปิดปิดอุปกรณ์ไฟฟ้าเมื่อมีการสุมเปิดปิดได้

การปรับความสว่างของหลอดไฟ สามารถปรับความสว่างในการหรี่ไฟและเพิ่มไฟได้ จากความสว่างต่ำสุดและความสว่างสูงสุดไปมา

การวัดอุณหภูมิจากบรรยากาศภายนอกรอบๆในระยะของบลูทูธได้โดยจะมีการอัปเดตอุณหภูมิบนหน้าจอแอปพลิเคชันต่อเมื่ออุณหภูมิจากภายนอกเปลี่ยนแปลง

การตรวจสอบสถานะการเปิดปิดของอุปกรณ์ไฟฟ้าสามารถทราบสถานะปัจจุบันที่แท้จริงของอุปกรณ์ไฟฟ้าที่ถูกสั่งการจากวงจรควบคุมว่าอยู่ในสถานะใดโดยใช้วงจรตรวจสอบกระแส ดังนั้นสถานะที่แปลงแปลงจะถูกอัปเดตไปยังโทรศัพท์มือถือ

5.2 ข้อเสนอแนะ

ปัญหาที่พบคือการส่งค่าจากโทรศัพท์มือถือเข้าไมโครโทรลเลอร์ ค่าเกิดขึ้นมีความผิดพลาดทำให้ส่งผลถึงรีเลย์ที่ควบคุมการส่งกระแสไฟฟ้าบางตัว และปัญหาของอุณหภูมิที่แสดงบนหน้าจอแอปพลิเคชัน ที่เกิดจากความผิดพลาดของระบบการอ่านค่าของแอนดรอยด์เมื่อเปลี่ยนเวอร์ชันเก่าไปเป็นเวอร์ชันใหม่ จึงต้องมีการปรับเปลี่ยนโค้ด เช่น มีการเพิ่ม Header เข้าไปในการรับส่งข้อมูล เมื่อจะมีการเปลี่ยนเวอร์ชันของแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. กอบเกียรติ สระอุบล. (2556). พัฒนา App Android. กรุงเทพฯ: มีเดีย เนทเวิร์ค
2. จักรชัย โสอินทร์ และพงษ์ศธร จันทร์ยอย. (2554). Basic Android App Development. นนทบุรี: ไอดีซี พรีเมียร์
3. พร้อมเลิศ หล่อวิจิตร. (2556). คู่มือเขียนแอป Android ฉบับสมบูรณ์. กรุงเทพฯ: โปรวีชั่น
4. *Environment Sensors*. (2010). November 8,2013,from http://developer.android.com/topics/sensors/sensors_environment.html
5. *How relay works?*. (2011). November 10,2013,from <http://electronics.howstuffworks.com/relay.htm>
6. *Relay*. (2011). November 12,2013,from <http://kpp.ac.th/elearning/elearning3/book-09.html>
7. *Pulse Width Modulation*. (2012). November 28,2013,from http://en.wikipedia.org/wiki/Pulse-width_modulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

Source Code ที่ใช้ในการทำงานของแอปพลิเคชัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. BT Communicator

```

package com.control.bluetoothcontrol;

import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.UUID;

import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.os.AsyncTask;
import android.os.Handler;
import android.os.Looper;

public class BTcommunicator {

    // UUID for this application
    private static final UUID MY_UUID =
    UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

    private MainActivity main;
    private InService inTask;

    private BluetoothSocket Socket = null;
    private BluetoothDevice Device;

    boolean TASK_STATE = true;
    boolean conn = true;

    public BTcommunicator(MainActivity mainAct) {
        main = mainAct;
    }

    public void runTask(BluetoothDevice device) {

        Device = device;

        Connect();

        TASK_STATE = true;
        inTask = new InService();
        inTask.execute();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        /***** Read data from the bluetooth device
        *****/
        public class InService extends AsyncTask<Void, Void,
Void> {

            @Override
            protected void onPreExecute() {}

            @Override
            protected Void doInBackground(Void... params) {
                while(TASK_STATE) {
                    try {
                        BufferedReader in = new
BufferedReader(new
BufferedReader(new
InputStreamReader(Socket.getInputStream())));

                        final String incomingMsg =
in.readLine();

                        Handler refresh = new
Handler(Looper.getMainLooper());
                        refresh.post(new Runnable() {
                            public void run() {
                                main.updateDataFromBT(incomingMsg);
                            }
                        });

                        } catch (Exception e) {
                            // TODO Auto-generated catch block
                            e.printStackTrace();
                        }
                    }
                }
                return null;
            }

            protected void onPostExecute() {}
        }

        /***** Send data to the bluetooth device
        *****/
        public void sendMsg(final String msg) {
            Runnable runSend = new Runnable() {
                public void run() {
                    try {

                        BufferedOutputStream out = new
BufferedOutputStream(Socket.getOutputStream());

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        byte[] msgBuffer = msg.getBytes();

        out.write(msgBuffer);
        out.flush();

        conn = true;

    } catch (Exception e) {
        // TODO Auto-generated catch block
        conn = false;
    }
}

};
new Thread(runSend).start();
}

// Connect
public void Connect() {
    try {
        // Create a Socket connection: need the
server's UUID number of registered
        Socket =
Device.createRfcommSocketToServiceRecord(MY_UUID);
        Socket.connect();
        conn = true;
    } catch (IOException e) {
        conn = false;
        if (Socket != null) {
            try {
                Socket.close();
                return ;
            } catch (IOException e2) {}
        }
    }
}

// Checking the bluetooth socket
public BluetoothSocket statusSocket() {
    return Socket;
}

// Get state connection
public boolean getConnected() {
    return conn;
}

// Stop Tasking
public void killTask() {
    conn = false;
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

super.onCreate(savedInstanceState);
setContentView(R.layout.discovery_layout);

/* BT isEnabled */
if (!_bluetooth.isEnabled()) {
    Log.w(TAGAPP, ">>Bluetooth is disable!");
    finish();
    return;
}

/* Register Receiver*/
IntentFilter discoveryFilter = new
IntentFilter(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
registerReceiver(_discoveryReceiver,
discoveryFilter);
IntentFilter foundFilter = new
IntentFilter(BluetoothDevice.ACTION_FOUND);
registerReceiver(_foundReceiver, foundFilter);

/* show a dialog "Scanning..." */
SamplesUtils.indeterminate(DiscoveryActivity.this,
_handler, "Device Scanning...", _discoveryWorkder, new
OnDismissListener() {
    public void onDismiss(DialogInterface dialog)
{
    for (; _bluetooth.isDiscovering();) {
        _bluetooth.cancelDiscovery();
    }
    _discoveryFinished = true;
}
}, true);
}

private Runnable _discoveryWorkder = new Runnable() {
    public void run() {
        /* Start search device */
        _bluetooth.startDiscovery();
        Log.d(TAGAPP, ">>Starting Discovery");
        for (;;) {
            if (_discoveryFinished) {
                Log.d(TAGAPP, ">>Finished");
                break;
            }
            try {
                Thread.sleep(100);
            } catch (InterruptedException e){}
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/**
 * Receiver
 * When the discovery finished be called.
 */
private BroadcastReceiver _foundReceiver = new
BroadcastReceiver() {
    public void onReceive(Context context, Intent
intent) {
        /* get the search results */
        BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
        /* add to list */
        _devices.add(device);
        /* show the devices list */
        showDevices();
    }
};
private BroadcastReceiver _discoveryReceiver = new
BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent
intent) {
        /* unRegister Receiver */
        Log.d(TAGAPP, ">>unregisterReceiver");
        unregisterReceiver(_foundReceiver);
        unregisterReceiver(this);
        _discoveryFinished = true;
    }
};

/* Show devices list */
protected void showDevices() {
    List<String> list = new ArrayList<String>();
    for (int i = 0, size = _devices.size(); i < size;
++i) {
        StringBuilder b = new StringBuilder();
        BluetoothDevice d = _devices.get(i);
        b.append(d.getAddress());
        b.append('\n');
        b.append(d.getName());
        String s = b.toString();
        list.add(s);
    }

    Log.d(TAGAPP, ">>showDevices");

    final ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, list);

```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        _handler.post(new Runnable() {
            public void run() {
                setListAdapter(adapter);
            }
        });

        /* Prompted to select a server to connect */
        Toast.makeText(this, "Please select device to
connect", Toast.LENGTH_SHORT).show();
    }

    /* Select device */
    protected void onItemClick(AdapterView l, View v, int
position, long id) {
        Log.d(TAGAPP, ">>Click device");
        Intent result = new Intent();
        result.putExtra(BluetoothDevice.EXTRA_DEVICE,
_devices.get(position));
        setResult(RESULT_OK, result);
        finish();
    }
}

```

3. LoginActivity

```

package com.control.bluetoothcontrol;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import android.app.Activity;
import android.content.Intent;

public class LoginActivity extends Activity {

    public static final String USER = "admin";
    public static final String PASSWORD = "admin";

    private String strUser, strPass;
    private EditText etxtUser, etxtPass;

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

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        etxtUser = (EditText) findViewById(R.id.editUser);
        etxtPass = (EditText) findViewById(R.id.editPass);

    }

    /* Login to Control page */
    public void LoginButtonClicked(View view)
    {
        strUser = etxtUser.getText().toString();
        strPass = etxtPass.getText().toString();

        if( (strUser == null || strUser.equals("")) ||
            (strPass == null || strPass.equals("")) ) {
            Toast.makeText(this, "You did not enter User
            Name/Password", Toast.LENGTH_SHORT).show();
        }else{
            if(strUser.equals(USER)) {
                if(strPass.equals(PASSWORD)) {
                    Intent connect = new Intent(this,
                    MainActivity.class);
                    startActivity(connect);
                }else{
                    Toast.makeText(this, "User
                    Name/Password invalid", Toast.LENGTH_SHORT).show();
                }
            }else{
                Toast.makeText(this, "User Name/Password
                invalid", Toast.LENGTH_SHORT).show();
            }
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        return ;
    }
}

```

4. MainActivity

```
package com.control.bluetoothcontrol;
```

```
import java.io.IOException;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.view.View.OnClickListener;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.widget.SeekBar.OnSeekBarChangeListener;

public class MainActivity extends Activity {

    /* Get Default Adapter */
    private BluetoothAdapter _bluetooth =
BluetoothAdapter.getDefaultAdapter();
    private BTcommunicator communicator = null;
    private BluetoothSocket mSocket = null;

    /* request BT enable */
    private static final int REQUEST_DISCOVERABLE = 1;
    private static final int REQUEST_DISCOVERY = 2;

    private String strOutput="0";
    private ToggleButton toggleOut0, toggleOut1, toggleOut2,
toggleOut3, toggleOut4, toggleOut5;
    private ImageView imvOut0, imvOut1, imvOut2;
    private SeekBar barOut0;
    private int barValueOut0=0;

    private String out0On = "K";
    private String out1On = "L";
    private String out2On = "M";
    private String out3On = "N";
    private String out4On = "O";
    private String out5On = "P";

    private String out0Off = "k";
    private String out1Off = "l";
    private String out2Off = "m";
    private String out3Off = "n";
    private String out4Off = "o";
    private String out5Off = "p";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private boolean OUT0_ON = false;

private TextView txtTemp;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.main_layout);

    startService();

    /* Enable BT */
    EnableBT();

    toggleOut0 = (ToggleButton)
    findViewById(R.id.toggleOut0);

    toggleOut0.setOnClickListener(toggleButton0Listener);

    toggleOut1 = (ToggleButton)
    findViewById(R.id.toggleOut1);

    toggleOut1.setOnClickListener(toggleButton1Listener);

    toggleOut2 = (ToggleButton)
    findViewById(R.id.toggleOut2);

    toggleOut2.setOnClickListener(toggleButton2Listener);

    toggleOut3 = (ToggleButton)
    findViewById(R.id.toggleOut3);

    toggleOut3.setOnClickListener(toggleButton3Listener);

    toggleOut4 = (ToggleButton)
    findViewById(R.id.toggleOut4);

    toggleOut4.setOnClickListener(toggleButton4Listener);

    toggleOut5 = (ToggleButton)
    findViewById(R.id.toggleOut5);

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยศูนย์บริการและสนับสนุนด้านการศึกษา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        imvOut0 = (ImageView) findViewById(R.id.imvOut0);
        imvOut1 = (ImageView) findViewById(R.id.imvOut1);
        imvOut2 = (ImageView) findViewById(R.id.imvOut2);

        txtTemp = (TextView) findViewById(R.id.txtTemp);

        initSeekBar();

    }

    ///////////////////////////////////////////////////////////////////
    ///////////////////////////////////////////////////////////////////
    public void updateDataFromBT(String data) {

        String[] msgSplit = data.split(":");

        if (msgSplit[0].equals("$TE")) { // Temperature
header checking.
            updateTEMP(msgSplit[1]);
        }else if (msgSplit[0].equals("$ST")) { // Output
state checking.
            String[] msgState = msgSplit[1].split(",");
            updateState(msgState[0],msgState[1],msgState[2],msgState
[3],msgState[4]);
        }
    }

    public void startService() {
        communicator = new BTcommunicator(this);
    }

    public void updateTEMP(String data) {
        String temp = "";
        temp = data;
        txtTemp.setText(temp);
    }

    ///////////////////////////////////////////////////////////////////
    ///////////////////////////////////////////////////////////////////
    public void updateState(String out1, String out2, String
out3, String out4, String out5) {
        if (out1.equals("0")) {
            // The toggle is off
            updateOutput(out1Off);
            toggleOut1.setChecked(false);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ใ้ใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}else if (out1.equals("1")) {
    // The toggle is on
    updateOutput(out1On);
    toggleOut1.setChecked(true);
    imvOut1.setImageResource(R.drawable.lamp_on);
}

if (out2.equals("0")) {
    // The toggle is off
    updateOutput(out2Off);
    toggleOut2.setChecked(false);

imvOut2.setImageResource(R.drawable.lamp_off);
}else if (out2.equals("1")) {
    // The toggle is on
    updateOutput(out2On);
    toggleOut2.setChecked(true);
    imvOut2.setImageResource(R.drawable.lamp_on);
}

if (out3.equals("0")) {
    // The toggle is off
    updateOutput(out3Off);
    toggleOut3.setChecked(false);
}else if (out3.equals("1")) {
    // The toggle is on
    updateOutput(out3On);
    toggleOut3.setChecked(true);
}

if (out4.equals("0")) {
    // The toggle is off
    updateOutput(out4Off);
    toggleOut4.setChecked(false);
}else if (out4.equals("1")) {
    // The toggle is on
    updateOutput(out4On);
    toggleOut4.setChecked(true);
}

if (out5.equals("0")) {
    // The toggle is off
    updateOutput(out5Off);
    toggleOut5.setChecked(false);
}else if (out5.equals("1")) {
    // The toggle is on
    updateOutput(out5On);
    toggleOut5.setChecked(true);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/* Connect to bluetooth */
public void ConnectButtonClicked(View view)
{
    /* Select device for list */
    Intent connect = new Intent(this,
DiscoveryActivity.class);
    startActivityForResult(connect,
REQUEST_DISCOVERY);
}

/* Disconnect to bluetooth */
public void DisconnectButtonClicked(View view)
{
    /* Disconnect with bluetooth */
    mSocket = communicator.statusSocket();
    if (mSocket != null) {
        try {
            communicator.killTask();
            mSocket.close();
            finish();
        } catch (IOException e) {
            Toast.makeText(this, "Destroy to close socket
failed", Toast.LENGTH_SHORT).show();
        }
    }
}

/* after select, connect to device */
@Override
protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode,
data);
    if (requestCode == REQUEST_DISCOVERY) {
        if (resultCode == RESULT_OK) {
            try {

// Set up a pointer to the remote
node
                final BluetoothDevice mDevice =
data.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);

// Connect to Bluetooth device
communicator.runTask(mDevice);

                Thread.sleep(100);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        mSocket =
communicator.statusSocket();
        if ( (mSocket != null) &&
(communicator.isConnected() == true) ) {
            sendStart();
        }
    } catch (Exception e) {}
    }
} else {
    return;
}
}

private void initSeekBar() {
    barOut0 = (SeekBar)
findViewById(R.id.seekBarOut0);
    barOut0.setOnSeekBarChangeListener(new
OnSeekBarChangeListener() {
        @Override
        public void onStopTrackingTouch(SeekBar arg0)
{
            // TODO Auto-generated method stub
            if(getOut0On()) {
                updateOutput(out0On);
                sendPacketSeekBar();
            }
        }
        @Override
        public void onStartTrackingTouch(SeekBar
arg0) {
            // TODO Auto-generated method stub
        }
        @Override
        public void onProgressChanged(SeekBar arg0,
int arg1, boolean arg2) {
            // TODO Auto-generated method stub
            barValueOut0 = arg1;
        }
    });
}

private OnClickListener toggleButton0Listener = new
OnClickListener() {

```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของบริษัทฯ ใช้เพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void onClick(View view) {
    if (toggleOut0.isChecked()) {
        // The toggle is on
        OUT0_ON = true;
        updateOutput(out0On);

imvOut0.setImageResource(R.drawable.lamp_on);
    } else {
        // The toggle is off
        OUT0_ON = false;
        updateOutput(out0Off);

imvOut0.setImageResource(R.drawable.lamp_off);
    }
    sendPacketSockBar();
}

};

private OnClickListener toggleButton1Listener = new
OnClickListener() {
    public void onClick(View view) {
        if (toggleOut1.isChecked()) {
            // The toggle is on
            updateOutput(out1On);

imvOut1.setImageResource(R.drawable.lamp_on);
        } else {
            // The toggle is off
            updateOutput(out1Off);

imvOut1.setImageResource(R.drawable.lamp_off);
        }
        sendPacket();
    }
};

private OnClickListener toggleButton2Listener = new
OnClickListener() {
    public void onClick(View view) {
        if (toggleOut2.isChecked()) {
            // The toggle is on
            updateOutput(out2On);

imvOut2.setImageResource(R.drawable.lamp_on);
        } else {
            // The toggle is off
            updateOutput(out2Off);

imvOut2.setImageResource(R.drawable.lamp_off);

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        }
        sendPacket();
    }
};

private OnClickListener toggleButton3Listener = new
OnClickListener() {
    public void onClick(View view) {
        if (toggleOut3.isChecked()) {
            // The toggle is on
            updateOutput(out3On);
        } else {
            // The toggle is off
            updateOutput(out3Off);
        }
        sendPacket();
    }
};

private OnClickListener toggleButton4Listener = new
OnClickListener() {
    public void onClick(View view) {
        if (toggleOut4.isChecked()) {
            // The toggle is on
            updateOutput(out4On);
        } else {
            // The toggle is off
            updateOutput(out4Off);
        }
        sendPacket();
    }
};

private OnClickListener toggleButton5Listener = new
OnClickListener() {
    public void onClick(View view) {
        if (toggleOut5.isChecked()) {
            // The toggle is on
            updateOutput(out5On);
        } else {
            // The toggle is off
            updateOutput(out5Off);
        }
        sendPacket();
    }
};

/*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* GETTERS & SETTERS
*/
private void updateOutput(String strOut) {
    strOutput = strOut;
}
private String getOutput() {
    return strOutput;
}
private String getLevelOut0() {
    if(getOut0On()) {
        return Integer.toString(barValueOut0);
    }else{
        return "0";
    }
}

private boolean getOut0On() {
    return OUT0_ON;
}

/* Send Seek Bar packet */
public void sendPacketSeekBar() {
    if(communicator != null) {
        if ( (mSocket != null) &&
(communicator.isConnected() == true) ) {
            /*
             * $BT,output,levelPWM
            */
            communicator.sendMessage("$BT"+","+getOutput()+","+getLevelOut0()+"\r\n");
        } else {
            Toast.makeText(this, "Device does not connect to Bluetooth", Toast.LENGTH_LONG).show();
        }
    }else{
        Toast.makeText(this, "Device does not create service with bluetooth", Toast.LENGTH_LONG).show();
    }
}

}

/* Send output packet */
public void sendPacket() {
    if(communicator != null) {
        if ( (mSocket != null) &&
(communicator.isConnected() == true) ) {
            /*
             * $BT,output

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในห้องเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        */

        communicator.sendMessage("$BT"+","+getOutput()+"\r\n");
    } else {
        Toast.makeText(this, "Device does not
connect to Bluetooth", Toast.LENGTH_LONG).show();
    }
} else {
    Toast.makeText(this, "Device does not create
service with bluetooth", Toast.LENGTH_LONG).show();
}

}

////////////////////////////////////
////////////////////////////////////
/* Send start packet */
public void sendStart() {
    if(communicator != null) {
        if ( (mSocket != null) &&
(communicator.isConnected() == true) ) {
            /*
             * $START
             */
            communicator.sendMessage("$START\r\n");
        } else {
            Toast.makeText(this, "Device does not
connect to Bluetooth", Toast.LENGTH_LONG).show();
        }
    } else {
        Toast.makeText(this, "Device does not create
service with bluetooth", Toast.LENGTH_LONG).show();
    }

}

private void EnableBT() {
    // Check for Bluetooth support and then check to
make sure it is turned on
    // Emulator doesn't support Bluetooth and will
return null
    if (_bluetooth == null) {
        Toast.makeText(this, "Device does not support
Bluetooth", Toast.LENGTH_LONG).show();
        finish();
    } else {
        if (!_bluetooth.isEnabled()) {
            //Prompt user to turn on Bluetooth
            _bluetooth.enable();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้เฉพาะในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        //Prompt user to turn on Bluetooth
        Intent enabler = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
        startActivityForResult(enabler,
REQUEST_DISCOVERABLE);
    }
}

@Override
public void onStop() {
    super.onStop();
}

@Override
public void onDestroy() {
    super.onDestroy();
    /* Disconnect with bluetooth */
    mSocket = communicator.statusSocket();
    if (mSocket != null) {
        try {
            mSocket.close();
        } catch (IOException e) {
            Toast.makeText(this, "Destroy to close socket
failed", Toast.LENGTH_SHORT).show();
        }
    }
    communicator.killTask();

    /* Close bluetooth */
    if (_bluetooth.isEnabled()) {
        _bluetooth.disable();
        System.exit(0);
        return ;
    }
    System.exit(0);
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

Source Code ที่ใช้ในการทำงานของอาตูดิวโนเพื่อควบคุมอุปกรณ์ไฟฟ้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define LED_PIN1 4
#define LED_PIN2 5
#define LED_PIN3 6
#define LED_PIN4 7

#define LED_PIN5 8
#define LED_PIN0 3

float temp;

int pushButton1 = 2;

int pushButton2 = 9;

int pushButton3 = 10;

int pushButton4 = 11;

int pushButton5 = 12;
int g;
int brightness;

int tempPin = 0;

boolean status_unlock;
boolean status_bluetooth;

int firstSensor = 0;
int secondSensor = 0;
int thirdSensor = 0;
int Byte = 0;
long interval = 1000;
long previousMillis = 0;
int minute, sec;

void setup()
{
    Serial.begin(9600);

    pinMode(pushButton1, INPUT);
    pinMode(pushButton2, INPUT);
    pinMode(pushButton3, INPUT);
    pinMode(pushButton4, INPUT);
    pinMode(pushButton5, INPUT);
    pinMode(LED_PIN1, OUTPUT);
    pinMode(LED_PIN2, OUTPUT);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pinMode(LED_PIN3, OUTPUT);
pinMode(LED_PIN4, OUTPUT);

pinMode(LED_PIN5, OUTPUT);

pinMode(tempPin, INPUT);
pinMode(LED_PIN0, OUTPUT);
status_bluetooth = true;
status_unlock = false;
sec = 0;
//temp = analogRead(tempPin);
}

void loop()

{
int buttonState1 = digitalRead(pushButton1);
Serial.print(buttonState1);
Serial.print (": ");

int buttonState2 = digitalRead(pushButton2);
Serial.print(buttonState2);
Serial.print (": ");

int buttonState3 = digitalRead(pushButton3);
Serial.print(buttonState3);
Serial.print (": ");

int buttonState4 = digitalRead(pushButton4);
Serial.print(buttonState4);
Serial.print (": ");

int buttonState5 = digitalRead(pushButton5);
Serial.print(buttonState5);
Serial.print (" , ");

// g = Serial.read();
temp = analogRead(tempPin)* 0.48828125;
delay(10);
// temp= temp * 0.48828125;
// Serial.println((byte)temp);
g = Serial.read();
switch(g )
{
case 'L':

digitalWrite(LED_PIN1, HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    //Serial.print('L');
    break;

    case 'l':
        digitalWrite(LED_PIN1, LOW);
        // Serial.print('l');
        break;

    case 'M':
        digitalWrite(LED_PIN2, HIGH);
        // Serial.print('M');
        break;

    case 'm':
        digitalWrite(LED_PIN2, LOW);
        //Serial.print('m');
        break;

    case 'N':
        digitalWrite(LED_PIN3, HIGH);
        //Serial.print('N');
        break;

    case 'n':
        digitalWrite(LED_PIN3, LOW);
        //Serial.print('n');
        break;

    case 'O':
        digitalWrite(LED_PIN4, HIGH);
        //Serial.print('O');
        break;

    case 'o':
        digitalWrite(LED_PIN4, LOW);
        // Serial.print('o');
        break;

    case 'P':
        digitalWrite(LED_PIN5, HIGH);
        //Serial.print('P');
        break;

    case 'p':

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    digitalWrite(LED_PIN5, LOW);
    //Serial.print('p');
    break;

    case'0':
    analogWrite(LED_PIN0, 0);
    // Serial.print('0');
    break;

    case'1':
    analogWrite(LED_PIN0, 10);
    // Serial.print('1');
    break;

    case'2':
    analogWrite(LED_PIN0, 31.87);
    // Serial.print('2');
    break;

    case'3':
    analogWrite(LED_PIN0, 63.75);
    // Serial.print('3');
    break;

    case'4':
    analogWrite(LED_PIN0, 95.62);
    // Serial.print('4');
    break;

    case'5':
    analogWrite(LED_PIN0, 127.5);
    // Serial.print('5');
    break;

    case'6':
    analogWrite(LED_PIN0, 159.37);
    // Serial.print('6');
    break;

    case'7':
    analogWrite(LED_PIN0, 191.25);
    //Serial.print('7');
    break;

    case'8':
    analogWrite(LED_PIN0, 223.12);
    // Serial.print('8');
    break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case '9':  
  analogWrite(LED_PIN0, 255);  
  // Serial.print('9');  
  break;  
  
}
```

```
Serial.println((byte)temp);
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM35 Precision Centigrade Temperature Sensors

FEATURES

- **Calibrated Directly in ° Celsius (Centigrade)**
- **Linear + 10 mV/°C Scale Factor**
- **0.5°C Ensured Accuracy (at +25°C)**
- **Rated for Full -55°C to +150°C Range**
- **Suitable for Remote Applications**
- **Low Cost Due to Wafer-Level Trimming**
- **Operates from 4 to 30 V**
- **Less than 60-µA Current Drain**
- **Low Self-Heating, 0.08°C in Still Air**
- **Nonlinearity Only ±¼°C Typical**
- **Low Impedance Output, 0.1 Ω for 1 mA Load**

DESCRIPTION

The LM35 series are precision integrated-circuit temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. Thus the LM35 has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full -55°C to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The low output impedance, linear output, and precise inherent calibration of the LM35 make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 is rated to operate over a -55°C to +150°C temperature range, while the LM35C is rated for a -40°C to +110°C range (-10° with improved accuracy). The LM35 series is available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

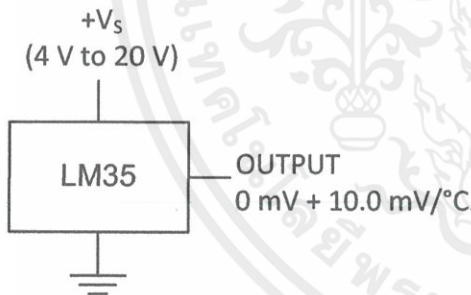
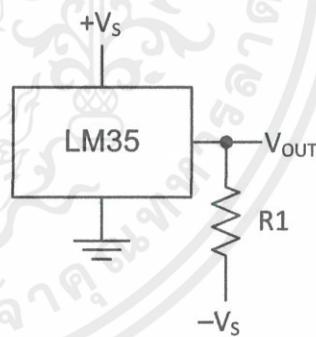


Figure 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S / 50 \mu\text{A}$
 $V_{\text{OUT}} = 1500 \text{ mV at } 150^\circ\text{C}$
 $V_{\text{OUT}} = 250 \text{ mV at } 25^\circ\text{C}$
 $V_{\text{OUT}} = -550 \text{ mV at } -55^\circ\text{C}$

Figure 2. Full-Range Centigrade Temperature Sensor



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

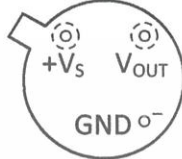
All trademarks are the property of their respective owners.



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

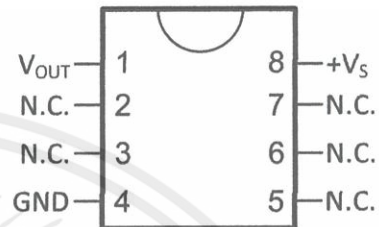
CONNECTION DIAGRAMS

METAL CAN PACKAGE TO (NDV)



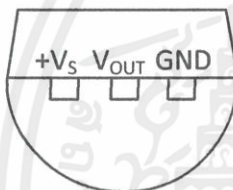
Case is connected to negative pin (GND)

SMALL-OUTLINE MOLDED PACKAGE SOIC-8 (D) TOP VIEW

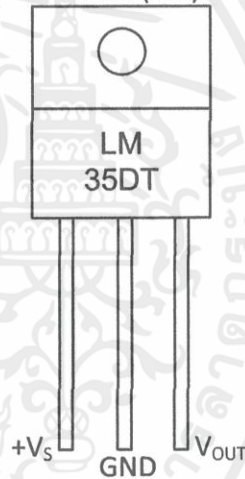


N.C. = No connection

PLASTIC PACKAGE TO-92 (LP) BOTTOM VIEW



PLASTIC PACKAGE TO-220 (NEB)



Tab is connected to the negative pin (GND).

NOTE: The LM35DT pinout is different than the discontinued LM35DP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS⁽¹⁾⁽²⁾

		MIN	MAX	UNIT
Supply voltage		-0.2	35	V
Output voltage		-1	6	V
Output current			10	mA
Electrostatic discharge (ESD) susceptibility ⁽³⁾			2500	V
Storage temperature	TO Package	-60	180	°C
	TO-92 Package	-60	150	
	TO-220 Package	-65	150	
	SOIC-8 Package	-65	150	
Lead temperature	TO Package (soldering, 10 seconds)		300	°C
	TO-92 and TO-220 Package (soldering, 10 seconds)		260	
	SOIC Package	Infrared (15 seconds)	220	
		Vapor phase (80 seconds)	215	
Specified operating temperature range: T_{MIN} to T_{MAX} ⁽⁴⁾	LM35, LM35A	-55	150	°C
	LM35C, LM35CA	-40	110	
	LM35D	0	100	

- (1) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
- (2) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See [Note 1](#).
- (3) Human body model, 100 pF discharged through a 1.5-k Ω resistor.
- (4) Thermal resistance of the TO-46 package is 400°C/W, junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-220 package is 90°C/W junction to ambient. For additional thermal resistance information see table in the [APPLICATIONS](#) section.

ELECTRICAL CHARACTERISTICS⁽¹⁾⁽²⁾

PARAMETER	TEST CONDITIONS	LM35A			LM35CA			UNITS (MAX.)
		TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	
Accuracy ⁽⁵⁾	$T_A = 25^\circ\text{C}$	± 0.2	± 0.5		± 0.2	± 0.5		°C
	$T_A = -10^\circ\text{C}$	± 0.3			± 0.3		± 1	
	$T_A = T_{MAX}$	± 0.4	± 1		± 0.4	± 1		
	$T_A = T_{MIN}$	± 0.4	± 1		± 0.4		± 1.5	
Nonlinearity ⁽⁶⁾	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.18		± 0.35	± 0.15		± 0.3	°C
Sensor gain (average slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	+10	+9.9, +10.1		+10		+9.9, +10.1	mV/°C
Load regulation ⁽⁷⁾ $0 \leq I_L \leq 1 \text{ mA}$	$T_A = 25^\circ\text{C}$	± 0.4	± 1		± 0.4	± 1		mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.5		± 3	± 0.5		± 3	
Line regulation ⁽⁷⁾	$T_A = 25^\circ\text{C}$	± 0.01	± 0.05		± 0.01	± 0.05		mV/V
	$4 \text{ V} \leq V_S \leq 30 \text{ V}$	± 0.02		± 0.1	± 0.02		± 0.1	

- (1) Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq 150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq 110^\circ\text{C}$ for the LM35C and LM35CA; and $0^\circ\text{C} \leq T_J \leq 100^\circ\text{C}$ for the LM35D. $V_S = 5 \text{ Vdc}$ and $I_{LOAD} = 50 \mu\text{A}$, in the circuit of [Figure 2](#). These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of [Figure 1](#). Specifications in boldface apply over the full rated temperature range.
- (2) Specifications in boldface apply over the full rated temperature range.
- (3) Tested Limits are ensured and 100% tested in production.
- (4) Design Limits are ensured (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.
- (5) Accuracy is defined as the error between the output voltage and 10 mV/°C times the case temperature of the device, at specified conditions of voltage, current, and temperature (expressed in °C).
- (6) Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the rated temperature range of the device.
- (7) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

ELECTRICAL CHARACTERISTICS⁽¹⁾⁽²⁾ (continued)

PARAMETER	TEST CONDITIONS	LM35A			LM35CA			UNITS (MAX.)
		TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	
Quiescent current ⁽⁸⁾	$V_S = 5\text{ V}, 25^\circ\text{C}$	56	67		56	67		μA
	$V_S = 5\text{ V}$	105		131	91		114	
	$V_S = 30\text{ V}, 25^\circ\text{C}$	56.2	68		56.2	68		
	$V_S = 30\text{ V}$	105.5		133	91.5		116	
Change of quiescent current ⁽⁷⁾	$4\text{ V} \leq V_S \leq 30\text{ V}, 25^\circ\text{C}$	0.2	1		0.2	1		μA
	$4\text{ V} \leq V_S \leq 30\text{ V}$	0.5		2	0.5		2	
Temperature coefficient of quiescent current		+0.39		+0.5	+0.39		+0.5	$\mu\text{A}/^\circ\text{C}$
Minimum temperature for rate accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2	+1.5		+2	$^\circ\text{C}$
Long term stability	$T_J = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

(8) Quiescent current is defined in the circuit of Figure 1.

ELECTRICAL CHARACTERISTICS⁽¹⁾⁽²⁾

PARAMETER	TEST CONDITIONS	LM35			LM35C, LM35D			UNITS (MAX.)
		TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	
Accuracy, LM35, LM35C ⁽⁵⁾	$T_A = 25^\circ\text{C}$	± 0.4	± 1		± 0.4	± 1		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.5			± 0.5		± 1.5	
	$T_A = T_{\text{MAX}}$	± 0.8	± 1.5		± 0.8		± 1.5	
	$T_A = T_{\text{MIN}}$	± 0.8		± 1.5	± 0.8		± 2	
Accuracy, LM35D ⁽⁵⁾	$T_A = 25^\circ\text{C}$				± 0.6	± 1.5		$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$				± 0.9		± 2	
	$T_A = T_{\text{MIN}}$				± 0.9		± 2	
Nonlinearity ⁽⁶⁾	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.3		± 0.5	± 0.2		± 0.5	$^\circ\text{C}$
Sensor gain (average slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+10	+9.8, +10.2		+10		+9.8, +10.2	$\text{mV}/^\circ\text{C}$
Load regulation ⁽⁷⁾ $0 \leq I_L \leq 1\text{ mA}$	$T_A = 25^\circ\text{C}$	± 0.4	± 2		± 0.4	± 2		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.5		± 5	± 0.5		± 5	
Line regulation ⁽⁷⁾	$T_A = 25^\circ\text{C}$	± 0.01	± 0.1		± 0.01	± 0.1		mV/V
	$4\text{ V} \leq V_S \leq 30\text{ V}$	± 0.02		± 0.2	± 0.02		± 0.2	

(1) Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq 150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq 110^\circ\text{C}$ for the LM35C and LM35CA; and $0^\circ\text{C} \leq T_J \leq 100^\circ\text{C}$ for the LM35D. $V_S = 5\text{ Vdc}$ and $I_{\text{LOAD}} = 50\text{ }\mu\text{A}$, in the circuit of Figure 2. These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of Figure 1. Specifications in boldface apply over the full rated temperature range.

(2) Specifications in boldface apply over the full rated temperature range.

(3) Tested Limits are ensured and 100% tested in production.

(4) Design Limits are ensured (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

(5) Accuracy is defined as the error between the output voltage and $10\text{ mV}/^\circ\text{C}$ times the case temperature of the device, at specified conditions of voltage, current, and temperature (expressed in $^\circ\text{C}$).

(6) Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the rated temperature range of the device.

(7) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

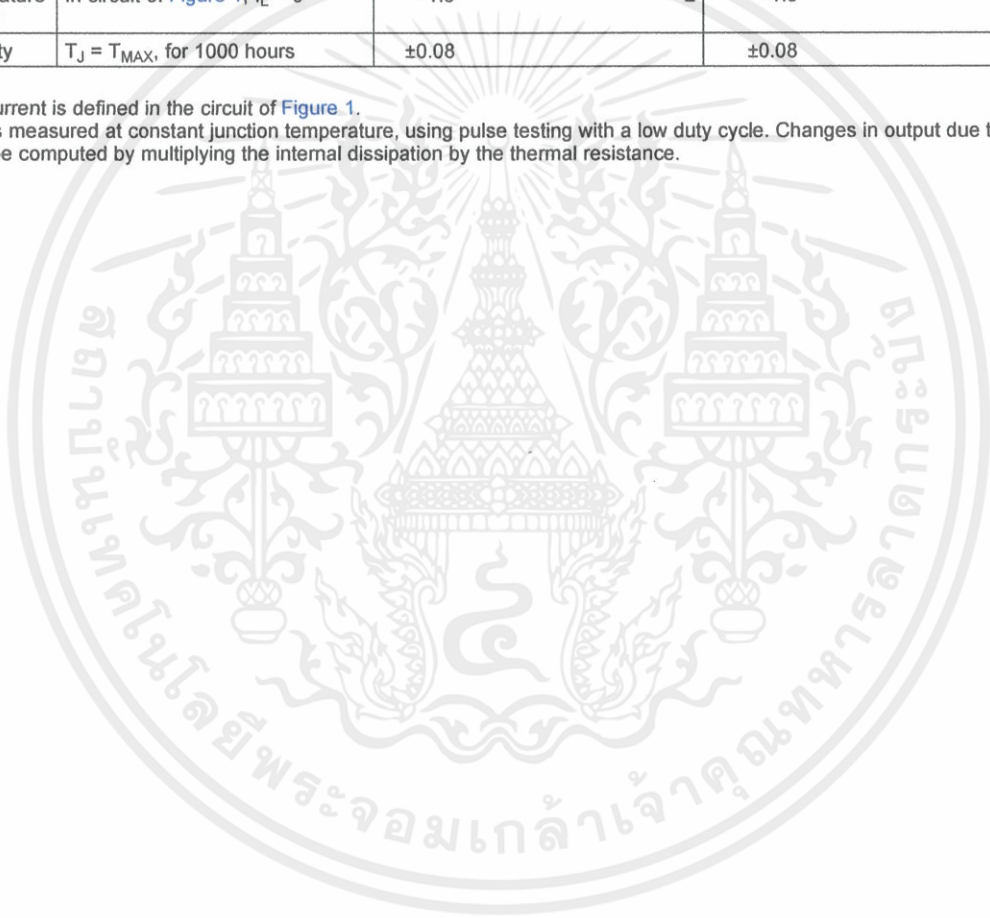
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS⁽¹⁾⁽²⁾ (continued)

PARAMETER	TEST CONDITIONS	LM35			LM35C, LM35D			UNITS (MAX.)
		TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	TYP	TESTED LIMIT ⁽³⁾	DESIGN LIMIT ⁽⁴⁾	
Quiescent current ⁽⁸⁾	$V_S = 5\text{ V}, 25^\circ\text{C}$	56	80		56	80		μA
	$V_S = 5\text{ V}$	105		158	91		138	
	$V_S = 30\text{ V}, 25^\circ\text{C}$	56.2	82		56.2	82		
	$V_S = 30\text{ V}$	105.5		161	91.5		141	
Change of quiescent current ⁽⁹⁾	$4\text{ V} \leq V_S \leq 30\text{ V}, 25^\circ\text{C}$	0.2	2		0.2	2		μA
	$4\text{ V} \leq V_S \leq 30\text{ V}$	0.5		3	0.5		3	
Temperature coefficient of quiescent current		+0.39		+0.7	+0.39		+0.7	$\mu\text{A}/^\circ\text{C}$
Minimum temperature for rate accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2	+1.5		+2	$^\circ\text{C}$
Long term stability	$T_J = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

(8) Quiescent current is defined in the circuit of Figure 1.

(9) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL PERFORMANCE CHARACTERISTICS

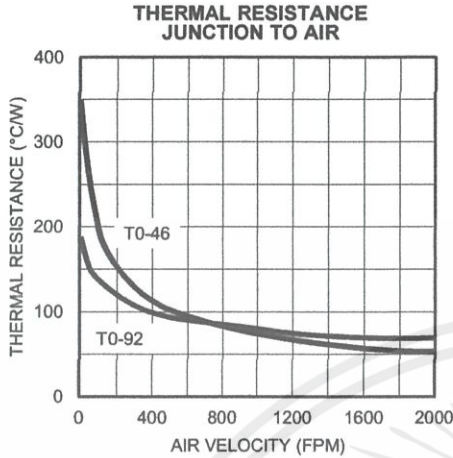


Figure 3.

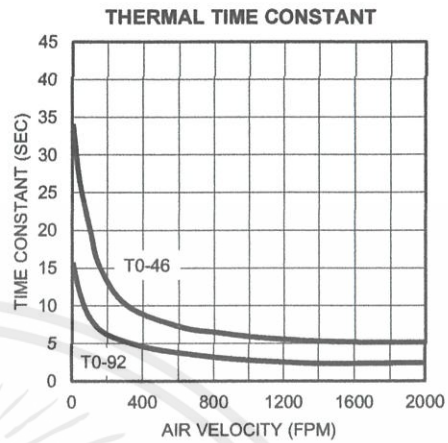


Figure 4.

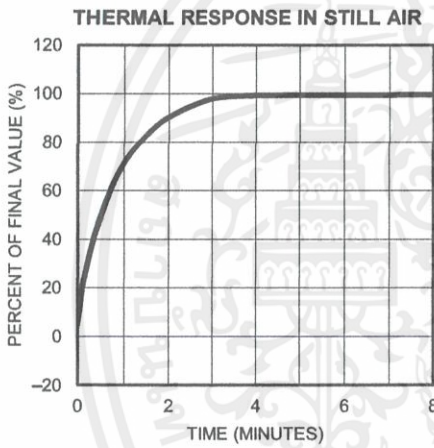


Figure 5.

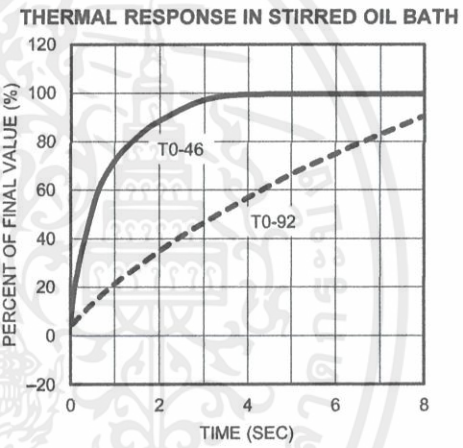


Figure 6.

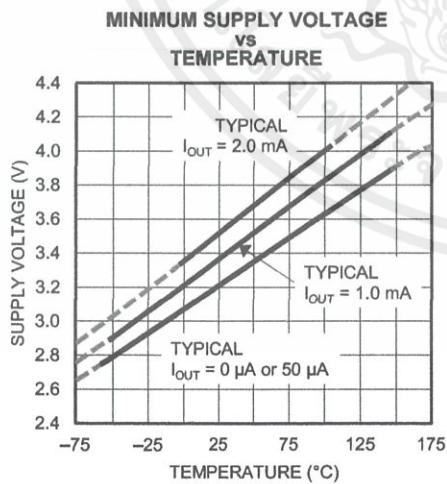


Figure 7.

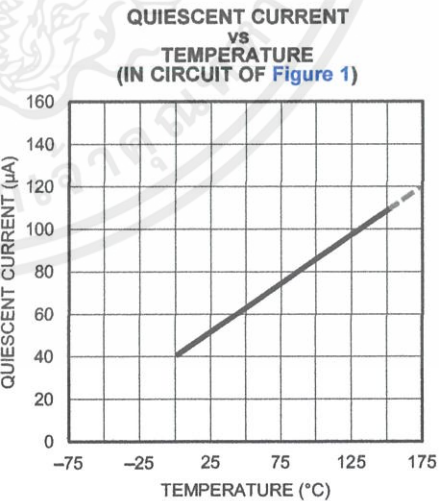


Figure 8.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL PERFORMANCE CHARACTERISTICS (continued)
QUIESCENT CURRENT

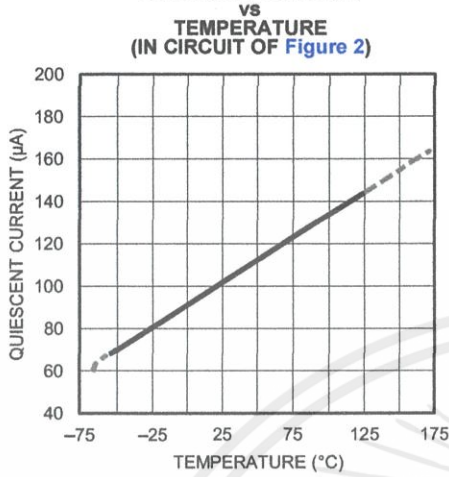


Figure 9.

ACCURACY vs TEMPERATURE (ENSURED)

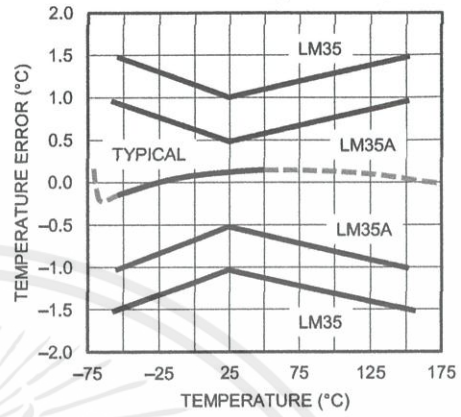


Figure 10.

ACCURACY vs TEMPERATURE (ENSURED)

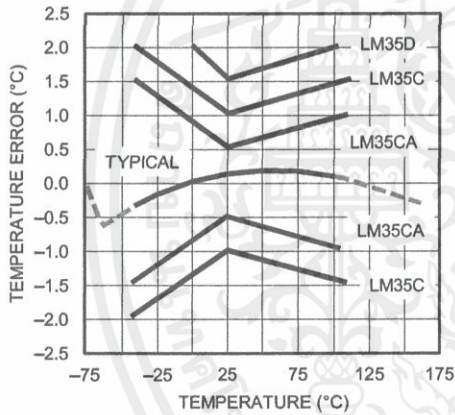


Figure 11.

NOISE VOLTAGE

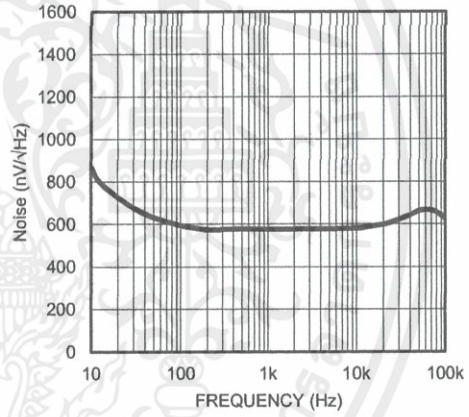


Figure 12.

START-UP RESPONSE

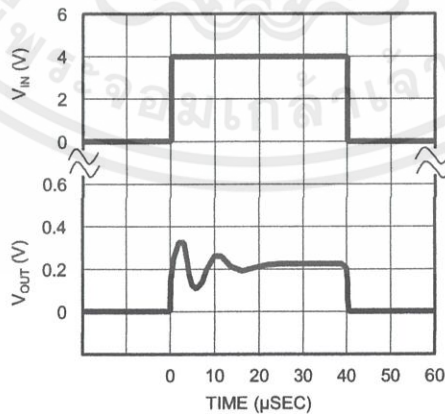


Figure 13.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATIONS

The LM35 is applied easily in the same way as other integrated-circuit temperature sensors. Glue or cement the device to a surface and the temperature should be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature. If the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature, which is especially true for the TO-92 plastic package where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, ensure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the temperature of the LM35 die is not affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V- terminal of the circuit will be grounded to that metal. Alternatively, mount the LM35 inside a sealed-end metal tube, and then dip into a bath or screw into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

Table 1. Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, θ_{JA})

	TO, no heat sink	TO ⁽¹⁾ , small heat fin	TO-92, no heat sink	TO-92 ⁽²⁾ , small heat fin	SOIC-8, no heat sink	SOIC-8 ⁽²⁾ , small heat fin	TO-220, no heat sink
Still air	400°C/W	100°C/W	180°C/W	140°C/W	220°C/W	110°C/W	90°C/W
Moving air	100°C/W	40°C/W	90°C/W	70°C/W	105°C/W	90°C/W	26°C/W
Still oil	100°C/W	40°C/W	90°C/W	70°C/W			
Stirred oil	50°C/W	30°C/W	45°C/W	40°C/W			
(Clamped to metal, Infinite heat sink)	(24°C/W)				(55°C/W)		

(1) Wakefield type 201, or 1-in disc of 0.02-in sheet brass, soldered to case, or similar.

(2) TO-92 and SOIC-8 packages glued and leads soldered to 1-in square of 1/16-in printed circuit board with 2-oz foil or similar.

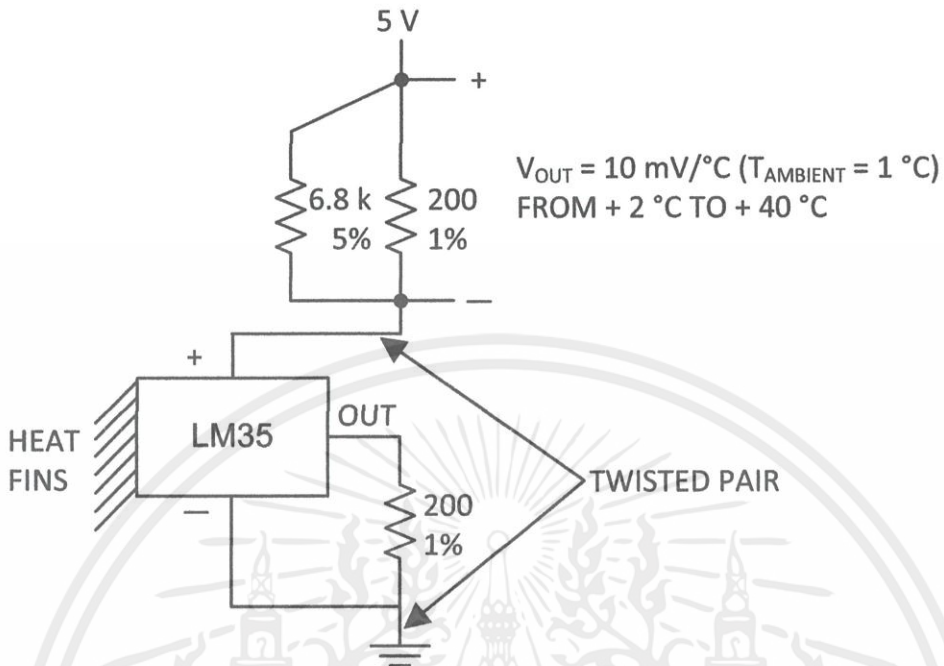


Figure 16. Two-Wire Remote Temperature Sensor (Grounded Sensor)

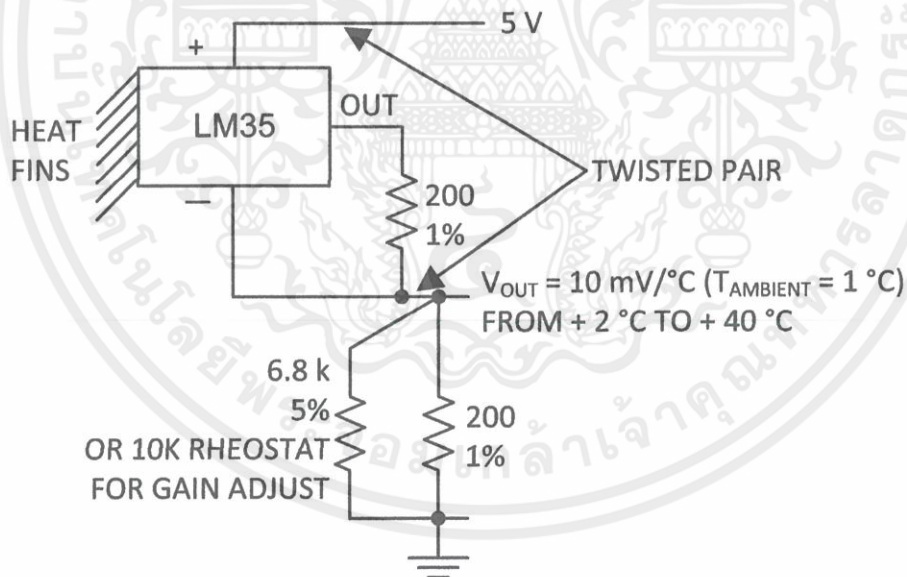


Figure 17. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

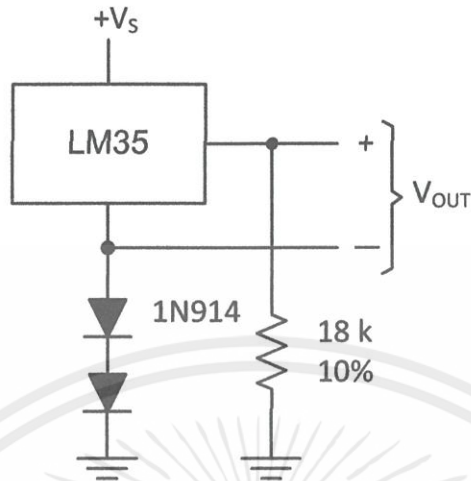


Figure 18. Temperature Sensor, Single Supply
(-55° to +150°C)

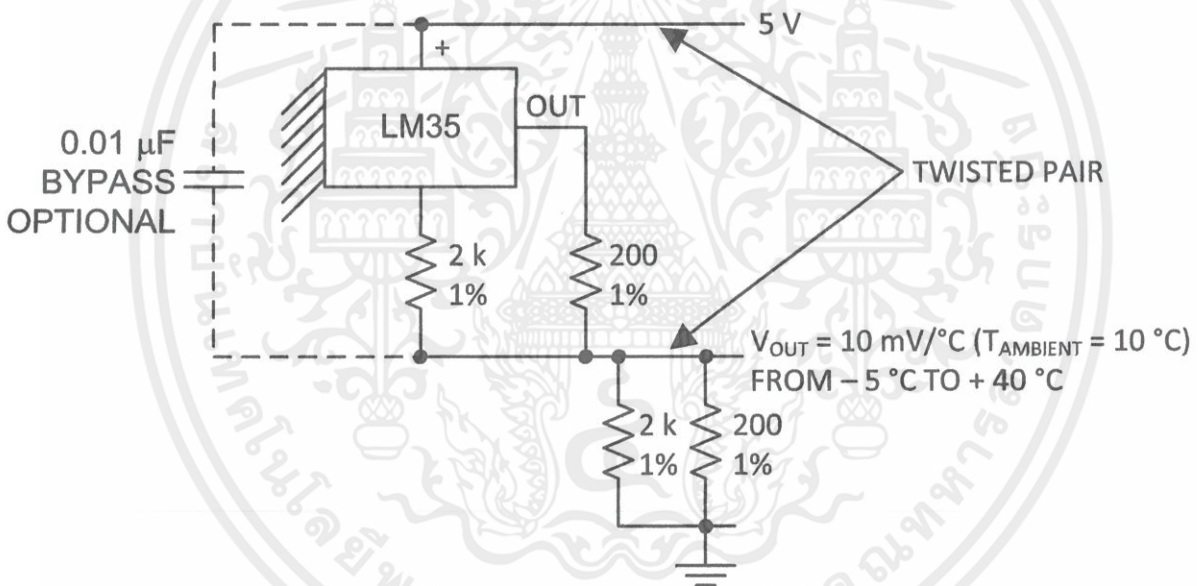


Figure 19. Two-Wire Remote Temperature Sensor
(Output Referred to Ground)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

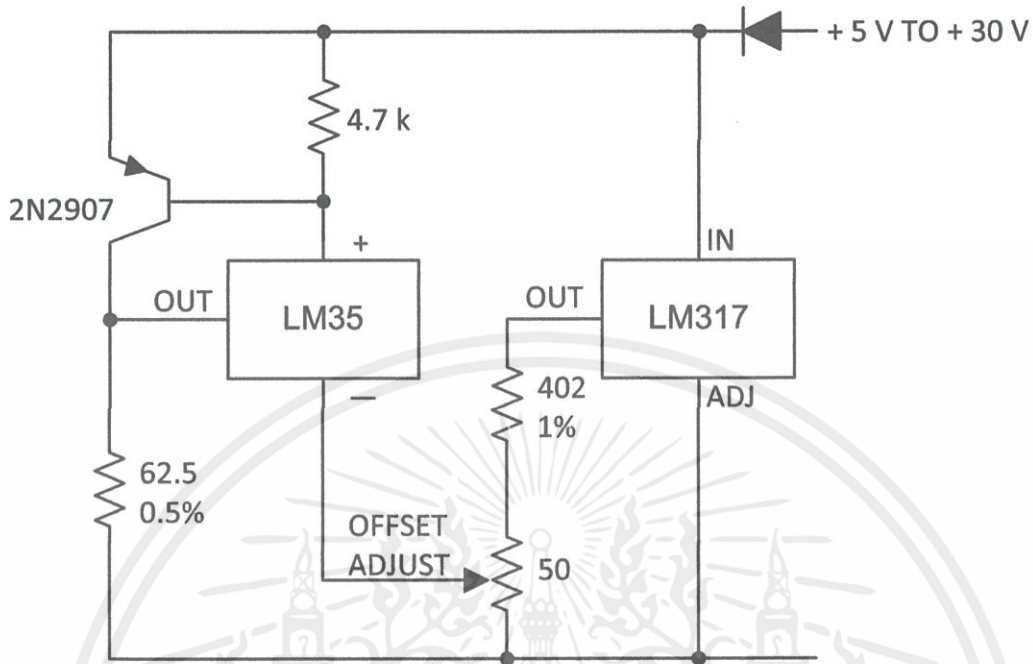


Figure 20. 4-To-20 mA Current Source (0°C to 100°C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

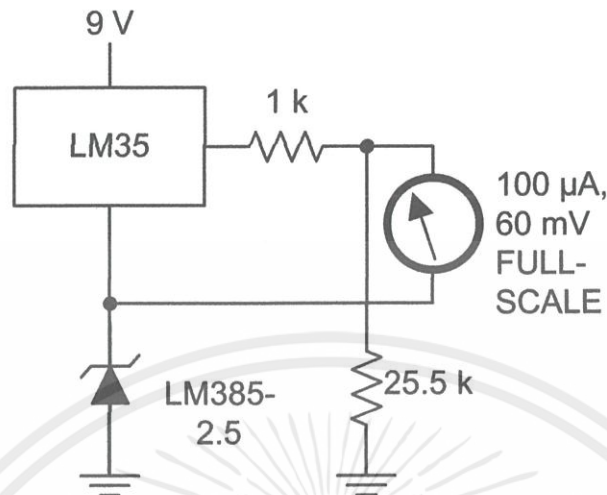


Figure 23. Fahrenheit Thermometer, Expanded Scale Thermometer (50°F to 80°F, for Example Shown)

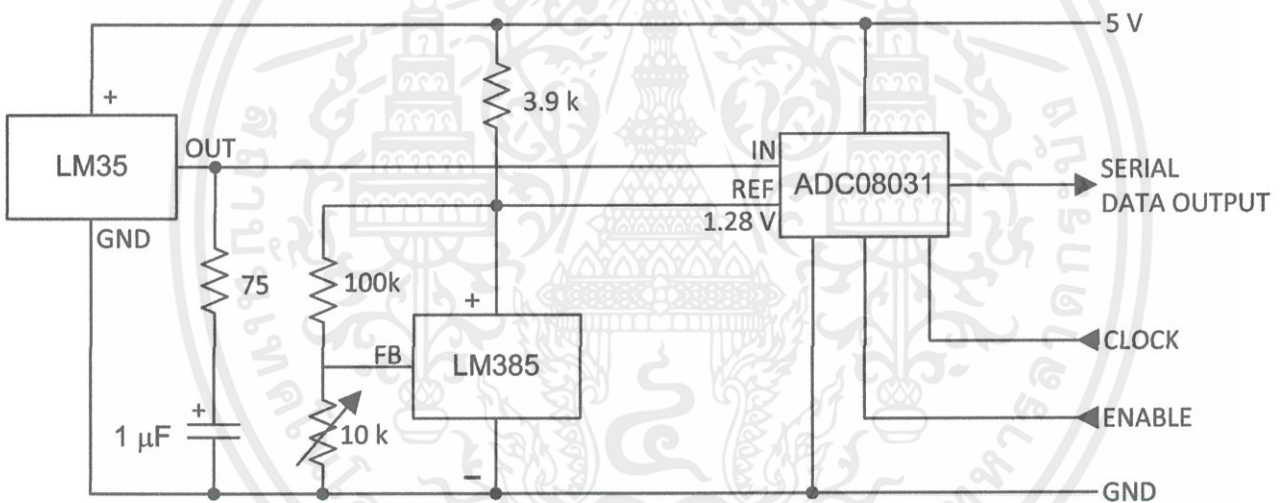


Figure 24. Temperature To Digital Converter (Serial Output) (128°C Full Scale)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

