

รายงานโครงงานวิศวกรรมไฟฟ้า วิชา 2102499 ปีการศึกษา 2560

ระบบควบคุมไฟฟ้าแสงสว่างแบบไร้สาย Wireless Control Lighting System

นาย ณัฐพงษ์ พูลเกษม เลขประจำตัว 5630184721

อาจารย์ที่ปรึกษา: รศ. ดร.ธวัชชัย เตชสุนันต์

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อ

ระบบไฟฟ้าแสงสว่างในปัจจุบันไม่ได้มีไว้แค่เพื่อให้แสงสว่าง แต่ต้องสามารถควบคุมให้สอดคล้องกับกิจกรรมที่หลากหลายได้ โครงงานนี้จึงมีวัตถุประสงค์เพื่อสร้างอุปกรณ์ควบคุมระบบไฟฟ้าแสงสว่างต้นแบบ ที่สามารถควบคุมการเปิด-ปิด, หรี่แสงและเปลี่ยนสี แบบไร้สายด้วยโทรศัพท์เคลื่อนที่ระบบแอนดรอยด์ได้ ในระยะไม่น้อยกว่า 10 เมตร โดยอุปกรณ์ต้นแบบประกอบด้วย 2 ส่วน คือ ส่วนฮาร์ดแวร์ ได้แก่ วงจรรับข้อมูล ซึ่งใช้โมดูลบลูทูธในการสื่อสารแบบไร้สายกับโทรศัพท์เคลื่อนที่, ตัวควบคุมใช้เป็นบอร์ดอาดูอิน นาโน และวงจรขับสำหรับควบคุมหลอดแอลอีดี ส่วนของซอฟต์แวร์ ได้แก่ โปรแกรมสำหรับถอดรหัสและส่งสัญญาณควบคุมวงจรขับ และ โปรแกรมประยุกต์แอนดรอยด์สำหรับการสื่อสารของโทรศัพท์เคลื่อนที่กับโมดูลบลูทูธ โดยอุปกรณ์ต้นแบบที่สร้างเสร็จแล้วสามารถควบคุมหลอดแอลอีดีด้วยโปรแกรมประยุกต์แอนดรอยด์ในโทรศัพท์เคลื่อนที่ได้ตามที่คาดหวัง

คำสำคัญ : ระบบควบคุมไฟฟ้าแสงสว่างแบบไร้สาย, โปรแกรมประยุกต์แอนดรอยด์, อาดูอิน, การควบคุมด้วยบลูทูธ, วงจรควบคุมแอลอีดี

Abstract

Today, lighting system not only gives illuminance but also control to serve various user's activities. This project develops a wireless control lighting system device that can control using android smartphone in the range of 10 meter to turn on and turn off, dim and change the light color. This device contains hardware and software. Hardware is composed of Bluetooth receiver, microcontroller (Arduino Nano) and LED driver circuit. Software is composed of program for decode and generate PWM signal and android application to communicate with Bluetooth receiver. The completed prototype can control RGB LED using android application on smartphone as expected.

Keyword: Wireless lighting control, Android application, Arduino, Bluetooth control, LED controller circuit

สารบัญ	หน้า
1.บทนำ	2
2.ภาพรวมของโครงงาน	3
วัตถุประสงค์ของโครงงาน	3
ขอบเขตของโครงงาน	3
ผลลัพธ์ที่คาดหวังจากโครงงาน	3
3.ฮาร์ดแวร์และความรู้พื้นฐานที่เกี่ยวข้อง	3
บอร์ดอาดุยโนนาโน (Arduino Nano).....	3
บลูทูธโมดูล HM10	4
RGB แอลอีดี 9 วัตต์	4
Pulse Width Modulation (PWM)	4
โปรแกรมประยุกต์แอนดรอยด์ (Android Application).....	5
Receive Signal Strength Indication(RSSI).....	6
4.ผลลัพธ์ของโครงงาน	7
5.การวิเคราะห์และวิจารณ์	11
6.บทสรุป	13
7.เอกสารอ้างอิง.....	14
8.ภาคผนวก.....	15
i.) ซอร์สโค้ดอาดุยโนในส่วนควบคุม	15
ii.) ซอร์สโค้ดโปรแกรมประยุกต์แอนดรอยด์เฉพาะส่วน Main	16

1.บทนำ

การออกแบบระบบไฟฟ้าแสงสว่างในปัจจุบัน ผู้ออกแบบได้คำนึงถึงการออกแบบระบบแสงสว่างให้สอดคล้องกับกิจกรรมต่าง ๆ ที่เกิดขึ้นและออกแบบให้มีความสามารถที่มากกว่าการเปิด-ปิด เช่น สามารถปรับความสว่างได้, เปลี่ยนสีได้, ควบคุมจากระยะไกลได้ เป็นต้น ทั้งในสถานที่ขนาดใหญ่อย่างอาคารนิทรรศการ, ในห้องสตูดิโอไปจนถึงภายในบ้าน บริษัทอุปกรณ์แสงสว่างต่าง ๆ จึงพัฒนาผลิตภัณฑ์ออกมาเป็นจำนวนมากเพื่อตอบสนองการออกแบบระบบแสงสว่างในปัจจุบัน แต่บริษัทส่วนใหญ่จะแยกอุปกรณ์ออกเป็นสองส่วน คือ ส่วนหลอดไฟหรือโคมไฟที่ทำหน้าที่ส่องสว่างและส่วนของตัวควบคุมที่ทำหน้าที่ควบคุมความสว่างของหลอดไฟในส่วนแรก ซึ่งจะเห็นได้จากหน้าเว็บของบริษัทต่าง ๆ เช่น Philips [1], GE lighting [2] ฯลฯ ดังนั้นหากผู้ออกแบบหรือผู้ใช้ต้องการออกแบบหรือปรับปรุงระบบแสงสว่างให้สามารถตอบสนองกิจกรรมที่หลากหลายมากขึ้น จำเป็นต้องซื้ออุปกรณ์ทั้งสองส่วนมาประกอบเข้าด้วยกันทำให้สิ้นเปลืองงบประมาณเพราะอาจต้องรื้อระบบเก่าที่มีอยู่ออกทั้งหมด อีกทั้งมีความยืดหยุ่นน้อยเพราะส่วนโคมไฟและอุปกรณ์ควบคุมที่มาจากต่างบริษัทกันจะไม่สามารถประกอบเข้าด้วยกันได้หรือประกอบเข้าด้วยกันได้แต่ใช้งานได้ไม่สมบูรณ์ตามที่ต้องการ บางบริษัทก็ผลิตเป็นอุปกรณ์สำเร็จรูปที่รวมทั้งส่วนของโคมไฟและอุปกรณ์ควบคุมเข้าไว้ด้วยกันแต่ก็มีราคาสูงและมีให้เลือกไม่หลากหลาย เช่น Philips Hue [3] ดังแสดงในรูปที่ 1

ดังนั้นในโครงการนี้จะทำอุปกรณ์ควบคุมระบบไฟฟ้าแสงสว่างต้นแบบที่สามารถควบคุมระบบไฟฟ้าแสงสว่างได้จากระยะไกลผ่านโทรศัพท์เคลื่อนที่ โดยอุปกรณ์ต้นแบบนี้เหมาะสมกับระบบไฟฟ้าแสงสว่างขนาดเล็ก เช่น ในบ้านพักอาศัยหรือห้างร้านขนาดเล็ก เป็นต้น



รูปที่ 1: Philips Hue [3]

2.ภาพรวมของโครงการ

วัตถุประสงค์ของโครงการ

1. เพื่อสร้างอุปกรณ์ควบคุมระบบไฟฟ้าแสงสว่างต้นแบบ ที่สามารถควบคุมแบบไร้สายด้วยโทรศัพท์เคลื่อนที่ระบบแอนดรอยด์ให้เปิด-ปิดและหรี่แสงไฟได้
2. เขียนโปรแกรมประยุกต์แอนดรอยด์ (Android Application) ทั้งในส่วนของผัง (Layout) และโปรแกรม เพื่อสื่อสารแบบไร้สายกับตัวรับข้อมูล

ขอบเขตของโครงการ

พัฒนาอุปกรณ์ควบคุมระบบไฟฟ้าแสงสว่างต้นแบบที่สามารถเปิด-ปิด, หรี่แสงและเปลี่ยนสีได้(จากการผสมแสงสีในระบบ RGB ของหลอดแอลอีดีสีแดง, สีเขียวและสีน้ำเงิน) ที่มีขนาดกำลังของหลอดแอลอีดีรวม 9 วัตต์ ซึ่งสามารถควบคุมในระยะไกลแบบไร้สายผ่านทางโปรแกรมประยุกต์ในโทรศัพท์เคลื่อนที่ระบบแอนดรอยด์ได้ รวมทั้งเขียนโปรแกรมประยุกต์แอนดรอยด์ด้วยตนเอง

ผลลัพธ์ที่คาดหวังจากโครงการ

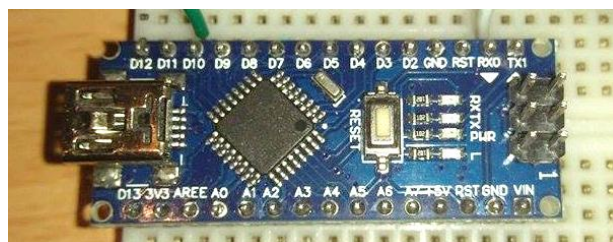
อุปกรณ์ต้นแบบที่สามารถควบคุมการเปิด-ปิด, หรี่แสงและเปลี่ยนสีได้ ภายในระยะ 10 เมตร แบบไร้สายผ่านทางโปรแกรมประยุกต์ที่เขียนขึ้นทางโทรศัพท์เคลื่อนที่ระบบแอนดรอยด์

3.ฮาร์ดแวร์และความรู้พื้นฐานที่เกี่ยวข้อง

บอร์ดอาดูยโนนาโน (Arduino Nano)

คือบอร์ดไมโครคอนโทรลเลอร์ขนาดเล็กของอาดูยโนจัดอยู่ในตระกูล AVR ใช้ชิป ATmega328 [4] เช่นเดียวกับอาดูยโนโนมีลักษณะดังรูปที่ 2 ในปัจจุบันมีการใช้งานอย่างกว้างขวาง มีข้อดีคือ

1. ราคาไม่แพง
2. มีขนาดเล็กติดตั้งบนแผงวงจรทดลองและวงจรจริงได้ง่าย
3. ซอฟต์แวร์ที่ใช้ในการเขียนโปรแกรมเป็นแบบ Open-Source
4. รองรับการเชื่อมต่อในรูปแบบดิจิทัลและแอนาล็อก
5. มีไลบรารีและมอดูลให้เลือกใช้จำนวนมาก
6. ในกรณีทำงานมีความซับซ้อน สามารถเขียนไลบรารีเพิ่มเติมได้ง่าย



รูปที่ 2:บอร์ดอาดูยโน นาโน

บลูทูธมอดูล HM10

คืออุปกรณ์บลูทูธที่สามารถใช้งานร่วมกับบอร์ดอาดุยโนได้โดยการสื่อสารแบบเรียงลำดับ (Serial) รองรับระบบการเชื่อมต่อบลูทูธ 4.0 หรือ Bluetooth Low Energy (BLE) ซึ่งเป็นมาตรฐานบลูทูธระบบใหม่ที่เน้นความเร็วและการประหยัดพลังงาน [5] มีความเสถียรในการรับส่งข้อมูล ใช้กันอย่างแพร่หลายในโทรศัพท์เคลื่อนที่ในปัจจุบัน สามารถทำงานได้ทั้งในโหมด Master และ Slave และมีระยะทำการที่กว้างมากกว่า 10 เมตร [5] เชื่อมต่อได้อย่างรวดเร็วและสามารถรับส่งข้อมูลได้โดยตรงไม่จำเป็นต้องมีการสร้างเซิร์ฟเวอร์ มีลักษณะดังรูปที่ 3



รูปที่ 3:บลูทูธมอดูล HM10

RGB แอลอีดี 9 วัดต์

คือหลอดแอลอีดีที่ภายในประกอบด้วยชิปแอลอีดีสีแดง สีเขียวและสีน้ำเงิน โดยเป็นชนิดกำลังสูงที่ให้แสงสว่างมาก โดยมีค่าความต่างศักย์ตกคร่อม, กระแสสูงสุดและความสว่างของแต่ละสีขณะทำงานตามที่แสดงในตารางที่ 1

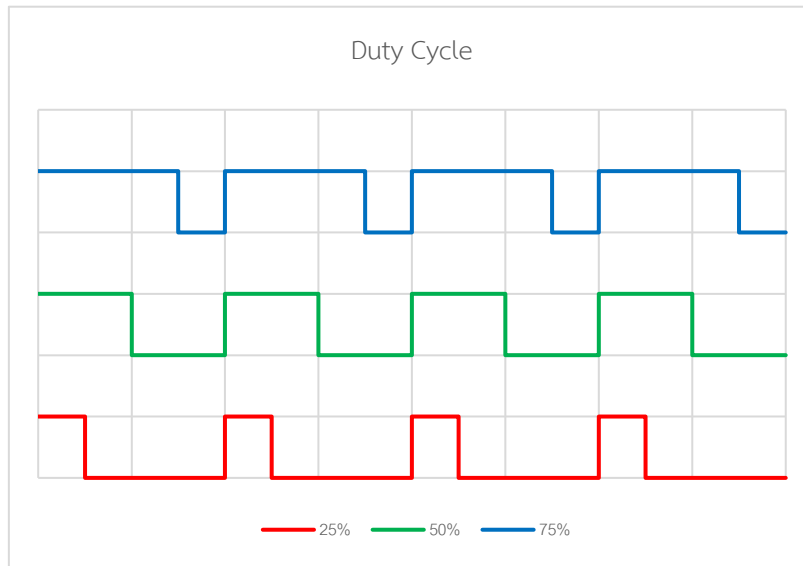
สีของชิป	ความต่างศักย์ตกคร่อม	กระแสสูงสุด	ความสว่าง
แดง	2.0-2.4 V	700 mA	100
เขียว	3.2-3.4 V	700 mA	140
น้ำเงิน	3.3-3.6 V	700 mA	80

ตารางที่ 1:ตารางแสดงคุณสมบัติเฉพาะของหลอดแอลอีดี [6]

Pulse Width Modulation (PWM)

คือการส่งสัญญาณที่มีรอบทำงาน (Duty Cycle) ซึ่งมีค่าตั้งแต่ 0-100% เพื่อควบคุมการทำงานของวงจร [7] โดยความแตกต่างของรอบทำงานต่าง ๆ แสดงดังรูปที่ 4 ซึ่งในวงจรควบคุมหลอดแอลอีดีความสว่างของหลอดแอลอีดีจะแปรผันโดยตรงกับรอบทำงาน คือเมื่อรอบทำงานมากขึ้นหลอดแอลอีดีจะสว่างมากขึ้น แต่เพื่อให้แสงสว่างที่ได้จากหลอดแอลอีดีมีความราบเรียบไม่เกิดแสงกระพริบ ความถี่ของสัญญาณที่ป้อนให้กับ

วงจรต้องมีค่าที่มากกว่าที่สายตามนุษย์แยกแยะได้ โดยในโครงงานนี้ใช้สัญญาณที่มีความถี่ 31 กิโลเฮิร์ตซ์ซึ่งเป็นความถี่ของขา PWM ขา 9,10,11 ของบอร์ดอาดุยโนนาโน [4]



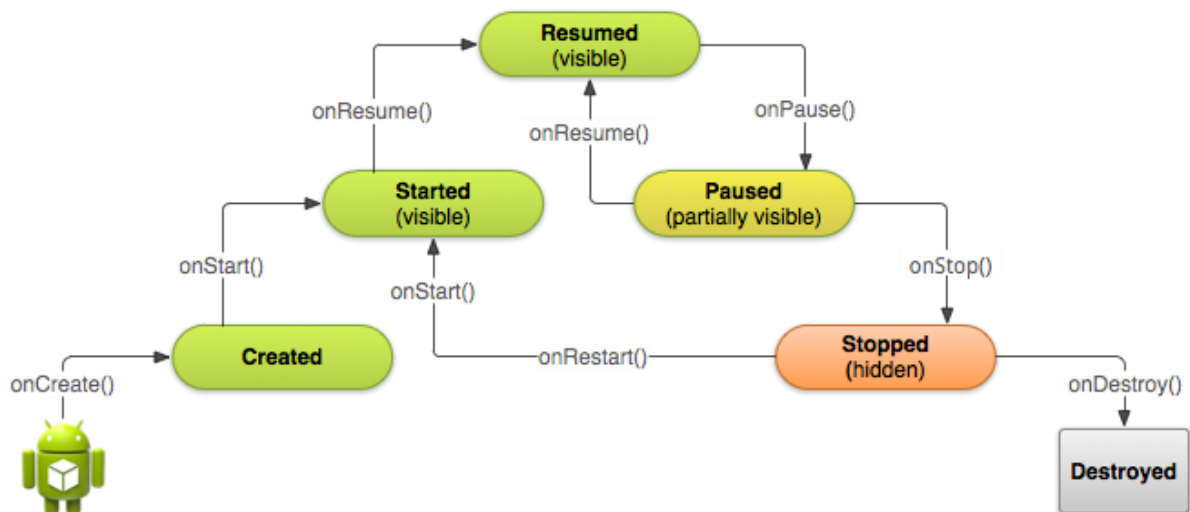
รูปที่ 4: กราฟแสดงรอบการทำงานต่าง ๆ

โปรแกรมประยุกต์แอนดรอยด์ (Android Application)

การพัฒนาโปรแกรมประยุกต์แอนดรอยด์ในโครงงานนี้จะใช้โปรแกรม Android Studio ซึ่งเป็นโปรแกรมฟรีที่พัฒนาโดยผู้เกี่ยวข้องกับบริษัทเจ็ทเบรนต์ ซึ่งการพัฒนาโปรแกรมประยุกต์แอนดรอยด์มีองค์ประกอบสำคัญ 2 ส่วน คือ

1. ผัง(Layout) คือ ส่วนที่รวมส่วนประกอบต่าง ๆ ของโปรแกรมเอาไว้ เช่น ปุ่ม, กล่องข้อความ เป็นต้น ซึ่งพัฒนาโดยใช้ภาษา xml
2. โปรแกรม คือ ส่วนสำหรับเขียนคำสั่งเมื่อกระทำใด ๆ บนส่วนประกอบในผัง เช่น กำหนดว่ากดปุ่มบนผังแล้วให้กล่องข้อความบนผังแสดงข้อความที่กำหนดไว้ ซึ่งพัฒนาโดยใช้ภาษา JAVA โดยส่วนของโปรแกรมนี้นี้ชื่อเฉพาะในทางเทคนิคว่า Activity ดังนั้นต่อไปจะขอใช้คำว่า Activity ในการอธิบายในส่วนของการเขียนโปรแกรมประยุกต์แอนดรอยด์

ลักษณะพิเศษของการเขียนโปรแกรมประยุกต์แอนดรอยด์ คือ Activity จะมีการเรียกคลาสที่แตกต่างกันในตัวเองเมื่อเกิดการกระทำต่าง ๆ โดยอัตโนมัติ เช่น เรียกคลาส onPause() อัตโนมัติเมื่อโปรแกรมถูกขัดจังหวะหรือเรียกคลาส onStop() อัตโนมัติเมื่อผู้ใช้ออกจากโปรแกรม ซึ่ง Activity มีวงจรการเรียกคลาสต่าง ๆ ดังแสดงในรูปที่ 5 ดังนั้นการเขียนโปรแกรมประยุกต์แอนดรอยด์จึงต้องวางแผนว่าจะให้เมทอดใดถูกเรียกขึ้นมาเมื่อใดอย่างเหมาะสม



รูปที่ 5:วงจรชีวิตของ Activity [8]

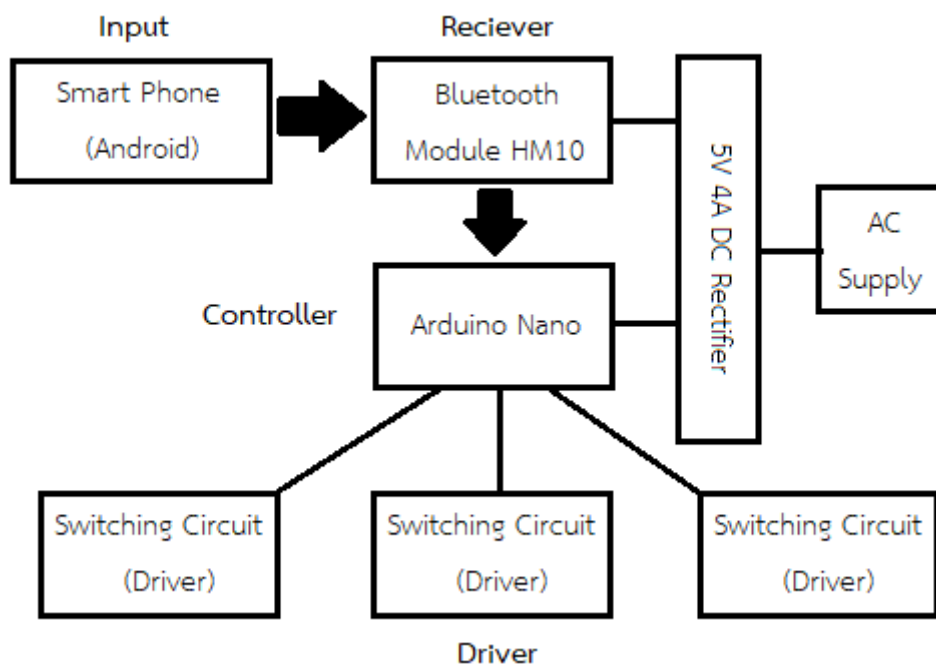
Receive Signal Strength Indication(RSSI)

คือค่าที่บอกถึงความแรงของสัญญาณมีหน่วยเป็นเดซิเบล เป็นค่าที่มอดูลส่งกลับมาเพื่อระบุว่าได้รับสัญญาณจากจุดเข้าถึง (Access Point) มากแค่ไหน โดยค่า RSSI ที่มอดูลส่งกลับมาเป็นค่าเปรียบเทียบซึ่งกำหนดโดยโรงงานที่ผลิต ไม่ใช่ค่าที่มีมาตรฐานกำหนดชัดเจน เช่น Cisco กำหนดไว้ที่ -100 เดซิเบลถึง 0 เดซิเบล, Atheros กำหนดไว้ที่ -60 เดซิเบลถึง 0 เดซิเบล เป็นต้น โดยหากค่า RSSI มีค่ามาก (ใกล้เคียง 0) หมายถึงคุณภาพสัญญาณที่ดีขึ้น [9]

ค่า RSSI สามารถนำมาประยุกต์เพื่อหาระยะห่างระหว่างมอดูลกับจุดเข้าถึงและหาระยะห่างสูงสุดที่มอดูลกับจุดเข้าถึงยังสามารถสื่อสารหากันได้

4.ผลลัพธ์ของโครงการ

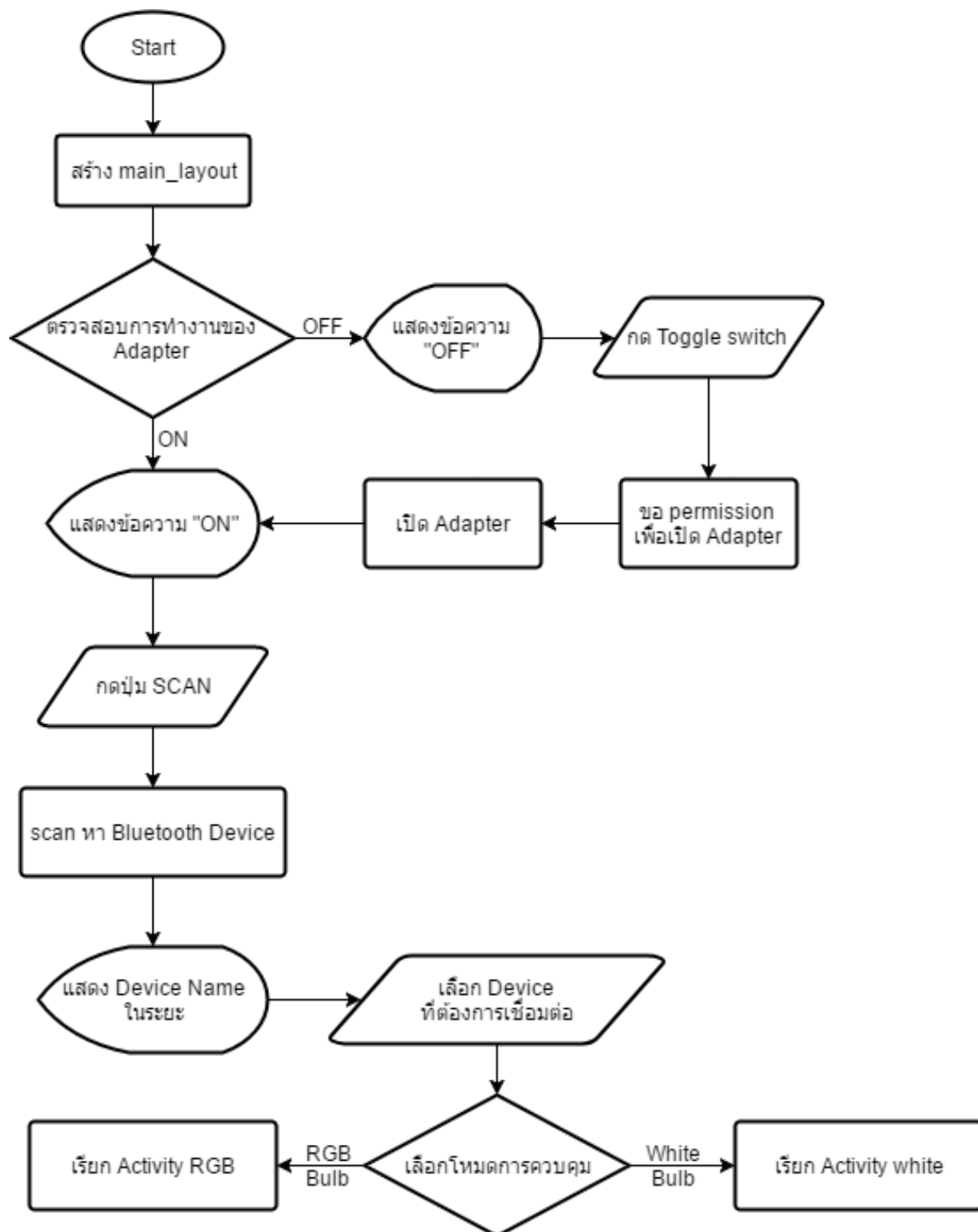
อุปกรณ์ต้นแบบสำหรับควบคุมระบบไฟฟ้าแสงสว่างแบบไร้สายผ่านโทรศัพท์เคลื่อนที่ระบบแอนดรอยด์มีบล็อกไดอะแกรมของวงจรดังรูปที่ 6



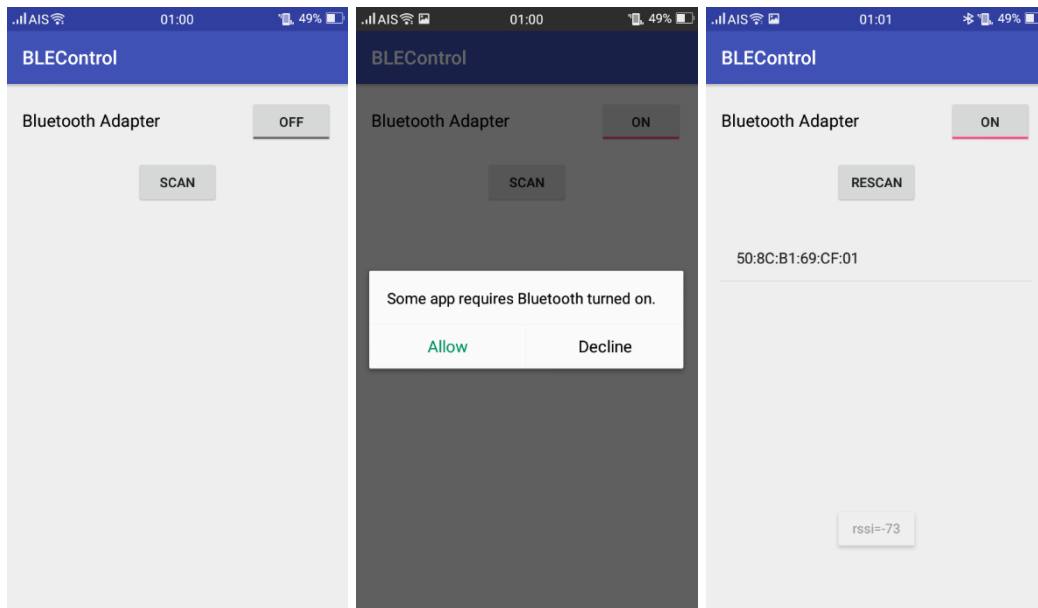
รูปที่ 6:บล็อกไดอะแกรมของวงจร

จากบล็อกไดอะแกรมของวงจรจะเห็นว่าวงจรประกอบด้วยองค์ประกอบหลักทั้งหมด 4 ส่วน คือ

1). ส่วนรับข้อมูลจากผู้ใช้ (Input) คือ โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบแอนดรอยด์ โดยโปรแกรมประยุกต์ที่เขียนขึ้น ประกอบด้วย Activity ทั้งหมด 3 หน้า คือ หน้าหลัก (Main), หน้าควบคุมแอลอีดีระบบ RGB และ หน้าควบคุมแอลอีดีสีเดียว มีผังงาน (flow chart) ดังแสดงในรูปที่ 7 และตัวอย่างการทำงานของ Activity ในหน้าหลักดังแสดงในรูปที่ 8



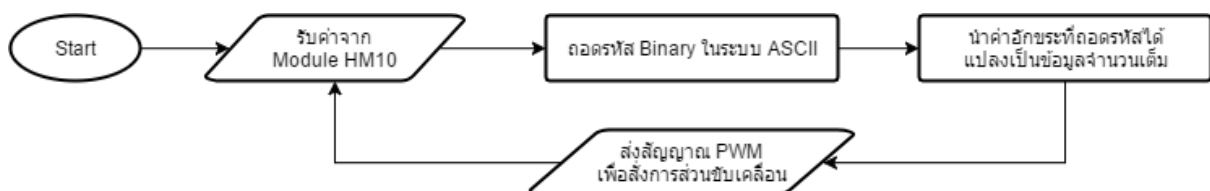
รูปที่ 7:ผังการทำงานของโปรแกรมประยุกต์แอนดรอยด์



รูปที่ 8:ตัวอย่างการทำงานของ Main Activity

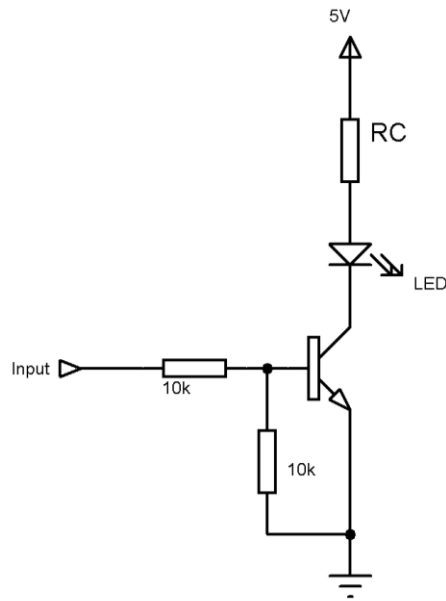
2). ส่วนรับข้อมูล (Receiver) คือ ส่วนของมอดูลบลูทูธ HM10 ซึ่งต่ออยู่กับบอร์ดอาดุยโน นาโน ทำหน้าที่รับข้อมูลทวิภาค (Binary) ที่ส่งมาจากโปรแกรมประยุกต์แอนดรอยด์ในส่วนที่ 1

3). ส่วนควบคุม (Controller) คือ ส่วนของบอร์ดอาดุยโน นาโนซึ่งเชื่อมต่อกับส่วนรับข้อมูลและส่วนขับเคลื่อน ทำหน้าที่นำข้อมูลทวิภาคจากส่วนรับข้อมูลมาถอดรหัสและสั่งการให้ส่วนขับเคลื่อนทำงาน โดยในโครงงานนำไลบรารี SerialSoftware มาใช้ในการสื่อสารกับมอดูล HM10 และส่งรับค่าจากมอดูลตลอดเวลา โดยโปรแกรมของส่วนควบคุมมีผังงานดังแสดงในรูปที่ 9



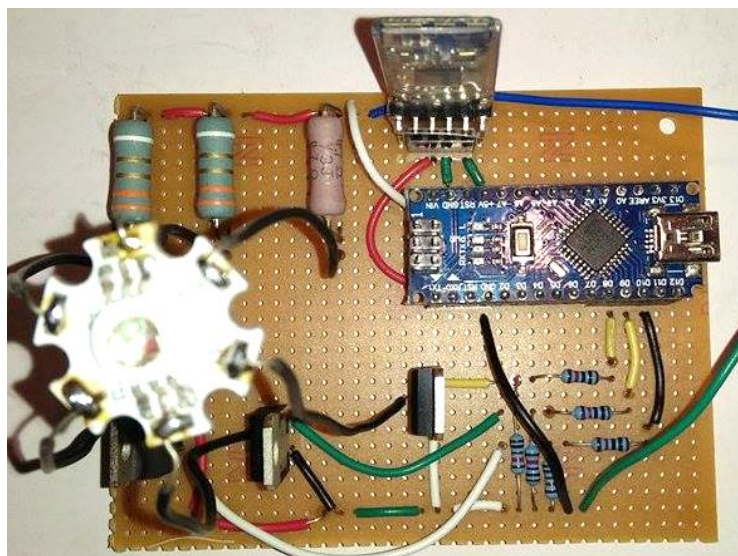
รูปที่ 9:ผังงานแสดงการทำงานของส่วนควบคุม

4). ส่วนขับเคลื่อน (Driver) คือ ส่วนที่ควบคุมการทำงานของหลอดแอลอีดี โดยในโครงงานใช้วงจรอิมิตเตอร์ร่วมมาประยุกต์ใช้เป็นวงจรสลับสาย (Switching Circuit) [10] โดยมีแผนภาพวงจรดังแสดงในรูปที่ 10 โดยการใช่วงจรอิมิตเตอร์ร่วมเป็นวงจรสวิตช์มีลักษณะสำคัญคือการจ่ายกระแสเบสให้วงจรอิมิตตลอดเวลา ทำให้ Vce มีค่าน้อยมาก (ประมาณ 0 โวลต์) จึงสามารถละลายได้ ทำให้คำนวณหาค่า Rc ซึ่งเป็นตัวต้านทานสำหรับจำกัดกระแสได้โดยง่ายจากกฎของเคอร์ชอฟฟ์

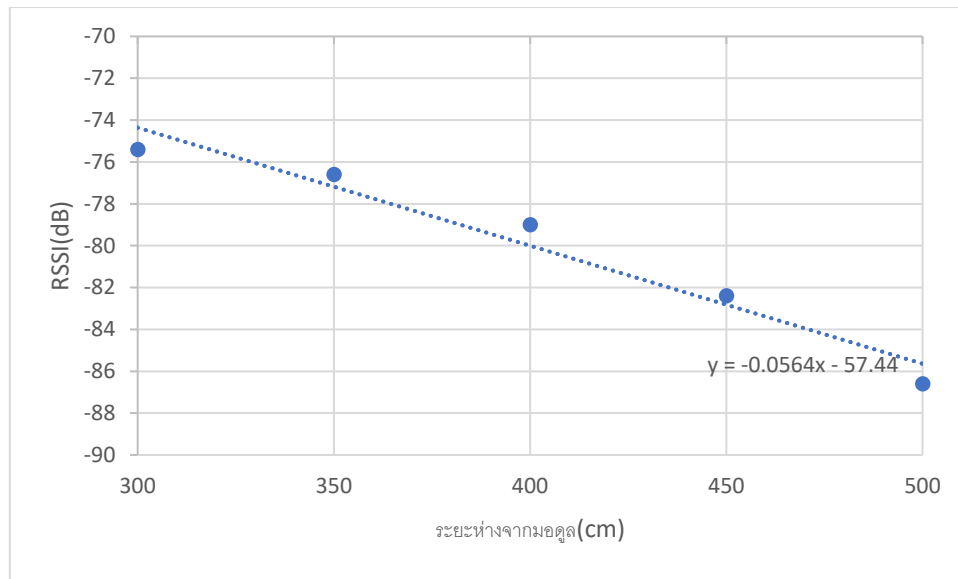


รูปที่ 10: วงจรสลับสายจากวงจรอิมิตเตอร์ร่วม

ในการทดลองเบื้องต้นพบว่าอุปกรณ์ต้นแบบสามารถทำงานได้เป็นอย่างดีตามที่ได้ออกแบบไว้ โดยสามารถควบคุมให้เปิด-ปิด, หรี่แสงและเปลี่ยนสีได้ ด้วยโทรศัพท์เคลื่อนที่แบบไร้สาย และพบว่าอุปกรณ์ต้นแบบสามารถสื่อสารได้ที่ระยะมากกว่า 10 เมตร จึงออกแบบการทดลองเพื่อหาระยะสูงสุดที่สามารถควบคุมอุปกรณ์ได้โดยวัดค่า RSSI ที่ระยะต่าง ๆ โดยอุปกรณ์ต้นแบบที่ใช้ในการทดลองมีลักษณะดังแสดงในรูปที่ 11 และนำค่าที่ได้มาเขียนกราฟได้ดังรูปที่ 12



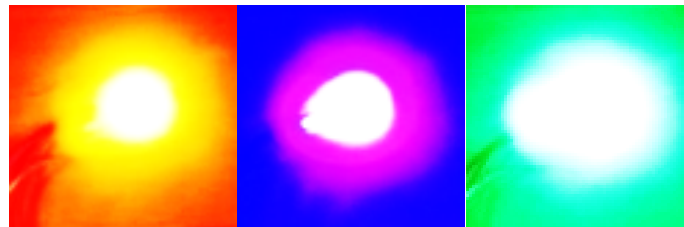
รูปที่ 11: อุปกรณ์ต้นแบบที่ใช้ทดลอง



รูปที่ 12:กราฟแสดงค่า RSSI ที่ระยะห่างต่าง ๆ

โดยในแต่ละระยะทำการเก็บค่า 5 ครั้งแล้วนำมาหาค่าเฉลี่ยเพื่อเขียนกราฟ จากนั้นนำกราฟที่ได้มา คำนวณหาระยะสูงสุด กำหนดให้ที่ระยะสูงสุดค่า RSSI=-100 เดซิเบล [9] ทำให้ได้ระยะสูงสุดประมาณ 75.5 เมตร

เมื่อทำการทดลองสั่งงานเพื่อควบคุมวงจรพบว่าวงจรสามารถทำงานได้ตามผลลัพธ์ที่คาดหวังคือ เปลี่ยนสีได้ดังแสดงในรูปที่ 13 และทดลองหรีแสงได้ดังแสดงในรูปที่ 14



รูปที่ 13:รูปแสดงการควบคุมเปลี่ยนสีหลอดแอลอีดี



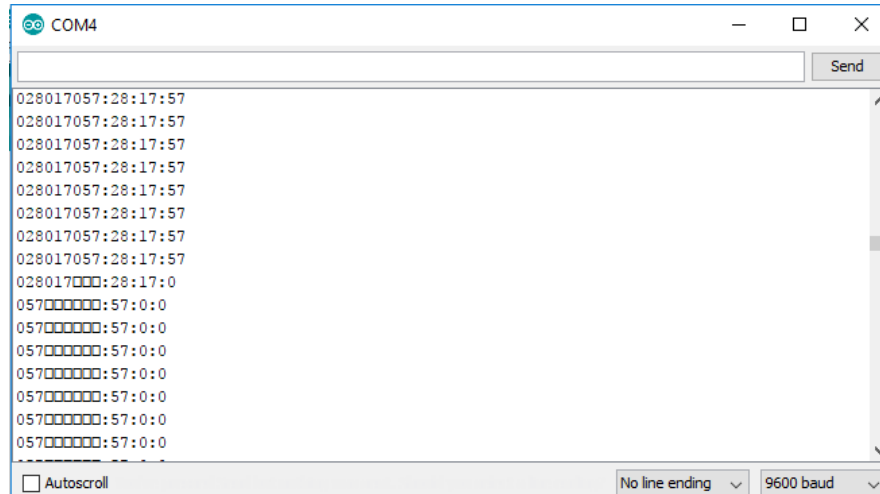
รูปที่ 14:รูปแสดงการควบคุมความสว่างของหลอดแอลอีดี (หรีแสง)

5.การวิเคราะห์และวิจารณ์

1. โปรแกรมประยุกต์แอนดรอยด์ ส่วนเชื่อมต่อกับผู้ใช้ (User Interface) ยังใช้งานได้ยาก โดยผู้ใช้งานจำเป็นต้องค้นหาอุปกรณ์ใหม่ทุกครั้งหากปิดโปรแกรมประแกรมประยุกต์หรือต้องการเชื่อมต่อกับ

อุปกรณ์อื่น มีความยืดหยุ่นในการควบคุมน้อยทำให้ยังไม่เหมาะกับกิจกรรมที่ต้องการให้มีการเปลี่ยนสีหรือความสว่างบ่อยครั้ง

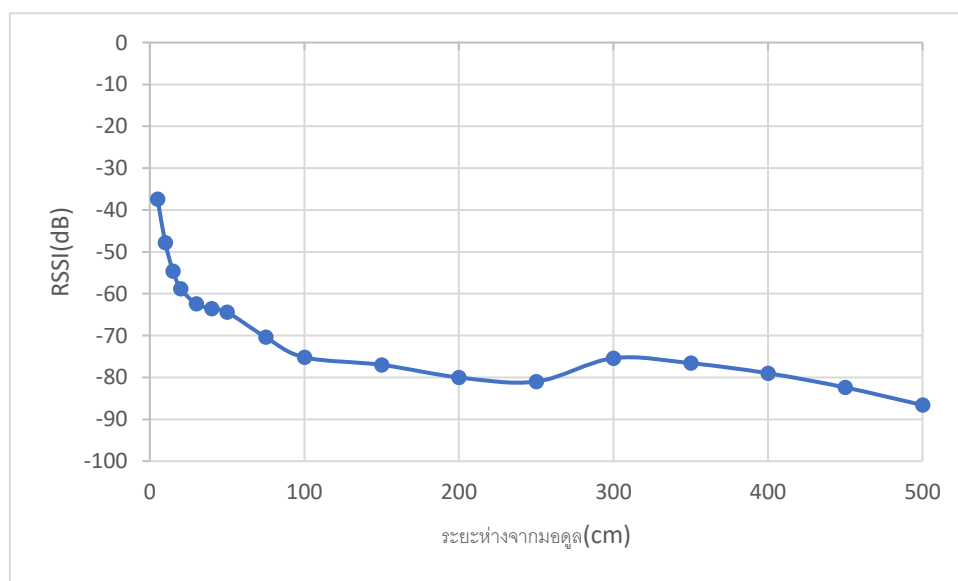
2. ส่วนของโปรแกรมสั่งการ จะมีการถอดรหัสที่ผิดพลาดหากข้อมูลที่ได้รับมามีการเปลี่ยนแปลงที่เร็วมากเกินไปดังแสดงในรูปที่ 15



รูปที่ 15:หน้าต่างแสดงข้อมูลที่ถอดรหัสแล้ว

3. ส่วนของวงจรขับเคลื่อน ในตอนเริ่มทำโครงงานนั้นได้ใช้ชิปไอซีสำหรับขับแอลอีดีแบบกระแสคงที่มาใช้ แต่พบว่าวงจรมีขนาดใหญ่มาก ทำให้ไม่สะดวกต่อการนำมาใช้กับ RGB แอลอีดีจึงเปลี่ยนมาใช้วงจรสลับสายในการขับเคลื่อนแทนเพราะมีองค์ประกอบน้อยทำให้ขนาดเล็กกว่ามาก แต่ก็มีข้อเสียในเรื่องของกำลังสูญเสียบนตัวต้านทานเนื่องจากกระแสมีค่าน้อยจึงทำให้กำลังสูญเสียมีค่าน้อยลงตาม

4. ระยะสูงสุดที่สามารถสื่อสารกับมอดูลได้นั้นเป็นเพียงค่าประมาณที่นำค่า RSSI ในช่วงที่เป็นเชิงเส้นมาคำนวณเท่านั้นเพราะในช่วงแรกนั้นค่า RSSI จะลดลงเร็วมากดังแสดงในรูปที่ 16



รูปที่ 16:กราฟแสดงความสัมพันธ์ของค่า RSSI กับระยะห่าง

6.บทสรุป

ผลลัพธ์ที่คาดหวังของโครงการ คือ อุปกรณ์ต้นแบบที่สามารถควบคุมการเปิด-ปิด, หรี่แสงและเปลี่ยนสีได้ ภายในระยะ 10 เมตร แบบไร้สายผ่านทางโปรแกรมประยุกต์ที่เขียนขึ้นทางโทรศัพท์เคลื่อนที่ระบบแอนดรอยด์ ซึ่งโครงการสามารถทำออกมาได้บรรลุตามผลลัพธ์ที่คาดหวังไว้ แต่ยังมีปัญหาในหลายเรื่อง เช่น โปรแกรมประยุกต์ที่ยังมีส่วนเชื่อมต่อกับผู้ใช้ (User Interface) ที่ยังไม่สวยงามใช้งานได้ยาก, มีปัญหาในการถอดรหัสข้อมูลเมื่อข้อมูลขาเข้าส่งมาเร็วมากเกินไปทำให้การควบคุมเกิดความผิดพลาดขึ้น ในอนาคตจะพัฒนาอุปกรณ์นี้ให้มีขนาดที่เล็กลงจนเท่ากับขนาดหลอดไฟมาตรฐานที่ใช้ในปัจจุบันเพื่อแก้ปัญหาในการติดตั้งของผู้ใช้ รวมทั้งพัฒนาซอฟต์แวร์ให้มีความเสถียรมากขึ้น, เพิ่มระบบจดจำอุปกรณ์ ทำให้ผู้ใช้ไม่จำเป็นต้องค้นหาอุปกรณ์ใหม่ทุกครั้งก่อนใช้งาน และ เพิ่มคุณสมบัติต่าง ๆ ให้มากขึ้นเพื่อให้ผู้ใช้สามารถใช้ได้ในทุก ๆ กิจกรรม เช่น ทำให้สามารถเพิ่ม-ลดแสงเองให้เหมาะสมกับความสว่างหน้าจอโทรศัพท์เพื่อลดปัญหาทางสายตาของผู้ใช้งาน

7.เอกสารอ้างอิง

- [1] Philips, "Lighting Control Systems," 2017. [Online]. Available:
<http://www.lighting.philips.com/main/products/lighting-controls>.
- [2] G. E. Company, "Control systems," 2017. [Online]. Available:
<http://www.gelighting.com/LightingWeb/na/solutions/control-systems/>.
- [3] Philips, "hue personal wireless lighting," 2017. [Online]. Available:
<http://www2.meethue.com/en-us>.
- [4] Arduino, "Arduino Nano," 2017. [Online]. Available:
<https://www.arduino.cc/en/Main/ArduinoBoardNano>.
- [5] Bluetooth, "Bluetooth Low Energy," 2017. [Online]. Available:
<https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy>.
- [6] B. Led, "LED 9W," LnwShop.com, 2017. [Online]. Available:
<http://www.bkkled.com/product/51/led-9w-สีละ-3w-rgb-6-ขา>.
- [7] JORDANDEE, "Pulse-width Modulation," Sparkfun, [Online]. Available:
<https://learn.sparkfun.com/tutorials/pulse-width-modulation>.
- [8] Ethan, "[Dev] Activity Lifecycle พื้นฐาน Android ที่ Developer ควรรู้," Ethan's Blog, 4 August 2014. [Online]. Available: <http://www.artit-k.com/android-activity-lifecycle/>.
- [9] Metageek, "Wifi Lesson," 2017. [Online]. Available:
<http://www.metageek.com/training/resources/understanding-rssi.html>.
- [10] JIMBO, "Transistor," Sparkfun, [Online]. Available:
<https://learn.sparkfun.com/tutorials/transistors/applications-i-switches>.

8.ภาคผนวก

i.) ขอร์สโค้ดอาตุนโนในส่วนควบคุม

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(0, 1); // RX, TX

String s="0000000000";

int r=0;

int g=0;

int b=0;

void setup() {

    // Open serial communications and wait for port to open:
    pinMode(9,OUTPUT);
    pinMode(10,OUTPUT);
    pinMode(11,OUTPUT);
    Serial.begin(9600);
    Serial.println("Bluetooth Ready");
    // set the data rate for the SoftwareSerial port
    mySerial.begin(9600);
}

void loop() { // run over and over
    if (mySerial.available()) {
        s.setCharAt(0,(char)mySerial.read());
        s.setCharAt(1,(char)mySerial.read());
        s.setCharAt(2,(char)mySerial.read());
        s.setCharAt(3,(char)mySerial.read());
        s.setCharAt(4,(char)mySerial.read());
        s.setCharAt(5,(char)mySerial.read());
        s.setCharAt(6,(char)mySerial.read());
        s.setCharAt(7,(char)mySerial.read());
        s.setCharAt(8,(char)mySerial.read());
        r=s.substring(0,3).toInt();
        g=s.substring(3,6).toInt();
```



```

        b=s.substring(6,9).toInt();
    }
    Serial.print(s);
    Serial.print(":");
    Serial.print(r);
    Serial.print(":");
    Serial.print(g);
    Serial.print(":");
    Serial.println(b);
    analogWrite(9,r);
    analogWrite(10,g);
    analogWrite(11,b);
    delay(50);
}

```

ii.) ซอร์สโค้ดโปรแกรมประยุกต์แอนดรอยด์เฉพาะส่วน Main

ซอร์สโค้ดในส่วนของโปรแกรม

```

package com.example.blecontrol;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothManager;
import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.ListView;
import android.widget.Toast;

```

```

import android.widget.ToggleButton;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private BluetoothAdapter btAdapter;
    private ToggleButton on_offButton;
    private Handler btHandler;
    private List<String> list_Device;
    private ArrayAdapter<String> listAdapter;
    private ListView listView;
    private String old_device="";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //initial
        on_offButton = (ToggleButton) findViewById(R.id.on_off_Button);
        btHandler = new Handler();
        //initial btAdapter
        BluetoothManager btManager = (BluetoothManager)
getSystemService(BLUETOOTH_SERVICE);
        btAdapter = btManager.getAdapter();
        //check btAdapter state
        if (btAdapter == null || !btAdapter.isEnabled()) {
            on_offButton.setChecked(false);
        } else on_offButton.setChecked(true);
    }

    @Override

```

```

protected void onResume() {
    super.onResume();
    //set on-off
    on_offButton.                setOnCheckedChangeListener(                new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            Intent                enableBtIntent                =                new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivity(enableBtIntent);
        }
    }
});
//find BLE device
final Button scan_Button = (Button) findViewById(R.id.scan_Button);
scan_Button.setText(R.string.scan);
listView = (ListView) findViewById(R.id.device_List);
list_Device = new ArrayList<>();
listAdapter=new ArrayAdapter<>(this,android.R.layout.simple_list_item_1,list_Device);
scan_Button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        scan_Button.setText(R.string.rescan);
        old_device="";
        list_Device.clear();
        scanBleDevice(true);
    }
});
}

```

```

private BluetoothAdapter.LeScanCallback scanCallback = new
BluetoothAdapter.LeScanCallback() {

```

```

@Override
public void onLeScan(final BluetoothDevice device, final int rssi, byte[] scanRecord) {
    runOnUiThread(new Runnable() {

```

```

@Override
public void run() {
    if (!device.toString().equalsIgnoreCase(old_device)){
        list_Device.add(device.toString());
        listView.setAdapter(listAdapter);

```

```

Toast.makeText(getApplicationContext().getBaseContext(),"rssi="+rssi,Toast.LENGTH_SHORT).show();

```

```

        old_device=device.toString();

```

```

    }

```

```

}

```

```

});

```

```

}

```

```

};

```

```

private void scanBleDevice(final boolean enable) {

```

```

    if (enable) {

```

```

        btHandler.postDelayed(new Runnable() {

```

```

            @Override

```

```

            public void run() {

```

```

                btAdapter.stopLeScan(scanCallback);

```

```

                Toast. makeText( getApplication( ) . getBaseContext( ) ," Stop
Scan",Toast.LENGTH_SHORT).show();

```

```

            }

```

```

        }, 5000);

```

```

        btAdapter.startLeScan(scanCallback);
        Toast.makeText(getApplicationContext(),
            "Start Scan", Toast.LENGTH_SHORT).show();
    } else {
        btAdapter.stopLeScan(scanCallback);
        Toast.makeText(getApplicationContext(),
            "Stop Scan", Toast.LENGTH_SHORT).show();
    }
}
}

```