



Wireless Interactive Programmable LED (WIPLED)

หมายเลขอุปกรณ์ 11

รายชื่อสมาชิก

- นายนัทธพงศ์ เลิศอำนาจ โชค 55070501021 tuamanga@gmail.com
- นายศิระ สักกาياะกรรมคล 55070501047 sira.sk@gmail.com

ที่ปรึกษา

ดร.ปริยกร ปุสตีโร

ที่ปรึกษาร่วม

ดร.ชาตรุนต์ หาญสมบูรณ์

วันที่ 31 พฤษภาคม 2559

ข้าพเจ้าได้อ่านรายงานและตรวจเนื้อหาของรายงานเรียบร้อยแล้ว

Project Title	Wireless Interactive Programmable LED (WIPLED)
Project Credit	3 credits
Project Participant	Mr. Nuthapong Lertamnuaychok Mr. Sira Sakkayakornmongkol
Advisor	Dr.-Ing Priyakorn Pusawiro
Co-Advisor	Dr. Jaturon Harnsomburana
Degree of Study	Bachelor's Degree
Department	Computer Engineering
Academic Year	2015

Abstract

The most currently art installation pieces which being used for decorative beauty of shows or events emit patterns of light without interacting with visitors. The Art Installation may be more attractive if combined with interactive technology. Lately, event organizers interest in this technology that will be used to draw more people to events. But the interactive technology is still not widely used due to lack of developers and high cost.

From the problems mentioned above, we decide to design and implement interactive art installation piece which called “Wireless Interactive Programmable LED (WIPLED)”. WIPLED is design to encourage the interactivity between audiences and art piece. It can be applied for an event or any space that need more attraction. The visitors able to interact with WIPLED by moving within range of sensor. The WIPLED will display color changing effect according to pre-setting or the interaction.

หัวข้อโครงการ	Wireless Interactive Programmable LED (WIPLED)
หน่วยกิตของโครงการ	3 หน่วยกิต
จัดทำโดย	นายนพดล พิริยะกุล นายศิริ ลักษณ์กุล
อาจารย์ที่ปรึกษา	ดร.ปริยกร ปุสตีโร
อาจารย์ที่ปรึกษาร่วม	ดร.ชาตรุนทร์ หาญสมบูรณ์
ระดับการศึกษา	วิศวกรรมศาสตร์บัณฑิต
ภาควิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2558

บทคัดย่อ

เนื่องจากในปัจจุบัน Art Installation ที่ใช้ตกแต่งตามงานแสดงโชว์ หรืออีเว้นท์ต่างๆ ในเวลากลางคืนนั้น เป็นชิ้นงานที่แสดงเพียงแสง สี เท่านั้นผู้เข้าชมไม่สามารถมีส่วนร่วมหรือมีปฏิสัมพันธ์กับชิ้นงานนั้นได้จึงไม่ดึงดูดความสนใจมากนัก แต่ในภายหลังทางด้านผู้จัดงานอีเว้นท์ได้ให้ความสนใจงานประเภท Interactive มา ก็เพื่อเพิ่มความน่าสนใจของบรรยากาศในงาน แต่ในตลาดเมืองไทยยังมีผู้พัฒนางานประเภทนี้อยู่จำนวนน้อย

จากปัญหาที่ได้กล่าวมาข้างต้น ทางเราจึงได้คิดที่จะพัฒนาและจัดทำชิ้นงานประเภทนี้ให้สามารถมีปฏิสัมพันธ์กับชิ้นงานได้ ซึ่งจะช่วยให้ชิ้นงานเหล่านี้มีความดึงดูดน่าสนใจมากขึ้นและยังสร้างการมีส่วนร่วมของผู้ใช้งานกับชิ้นงานที่จัดแสดงอยู่ด้วยได้ ซึ่งชิ้นงานที่จะพัฒนานี้จะสามารถนำไปประยุกต์ใช้ได้กับงานอีเว้นท์ต่างๆ ได้

ชิ้นงานสามารถที่จะมีปฏิสัมพันธ์กับมนุษย์ได้ โดยเมื่อมีการสัมผัสกับชิ้นงานในระยะเวลาทำงานของตัวเซนเซอร์ จะทำให้เกิดการเปลี่ยนแปลงสีของชิ้นงานตามที่ได้ตั้งค่าเอาไว้ ซึ่งการเปลี่ยนแปลงที่เกิดขึ้นนั้นจะเป็นลักษณะคล้ายการกระจายของหยดน้ำ

กิจกรรมประจำ

โครงการ WIPLED (Wireless Interactive Programmable LED) นี้สำเร็จได้ด้วยความอนุเคราะห์จากบุคคลหลายท่าน ซึ่งทางเราไม่อาจจะนำกล่าวได้ทั้งหมด โดยผู้มีพระคุณท่านแรกที่ผู้จัดทำโครงการได้ขอกราบพระคุณคือ ท่านอาจารย์ ซึ่งเป็นอาจารย์ที่ให้คำแนะนำตรวจทาน การนำเสนอผลงาน ตลอดไปจนถึงการเขียนรายงาน และแก้ไขในจุดที่บกพร่อง ด้วยความเอาใจใส่ทุกขั้นตอน ท่านที่สองคือ อาจารย์ ที่ได้ให้ความรู้ คำแนะนำตรวจทาน ตลอดไปถึงการแก้ไขข้อบกพร่องต่างๆ ทางเทคนิค เพื่อให้โครงการนี้มีความสมบูรณ์มากที่สุด ผู้จัดทำโครงการได้ขอกราบขอบพระคุณเป็นอย่างสูง ไว้ ณ โอกาสนี้ นอกจากนี้ยังขอขอบคุณเพื่อนๆ นักศึกษาคณะวิศวกรรมศาสตร์สาขา วิศวกรรมคอมพิวเตอร์ในการแลกเปลี่ยนความรู้ ความคิดเห็นและให้กำลังใจการจัดทำโครงการนี้ตลอดมา

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ที่มาของปัญหาและแนวทางการแก้ปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของงานวิจัย	1
1.4 ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน	1
1.5 แผนการดำเนินงาน	2
บทที่ 2 ที่มา ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ทฤษฎีที่เกี่ยวข้อง	4
2.1.1 ไมโครคอนโทรลเลอร์และการเลือกใช้งาน	4
2.1.2 RGB color	5
2.1.3 Individually addressable LEDและการเลือกใช้งาน	6
2.1.4 Capacitive proximity sensor	7
2.1.5 การเลือกใช้มาตรฐานของการส่งข้อมูลผ่านเครือข่ายไร้สาย	8
2.2 งานวิจัยที่เกี่ยวข้อง	8
2.2.1 SIX-FORTY BY FOUR-EIGHTY	9
2.2.2 Mini Burble Paris	10
บทที่ 3 การออกแบบและระบบวิธีวิจัย	12
3.1 อุปกรณ์ที่ใช้ในการดำเนินงาน	12
3.1.1 ฮาร์ดแวร์	12
3.1.1.1 Arduino Nano	12
3.1.1.2 RGB LED Strip WS2812B	14
3.1.1.3 Capacitive Touch Sensor MPR121	16
3.1.1.4 Wi-Fi module ESP8266-12E	17
3.1.2 ซอฟต์แวร์	18
3.1.2.1 Arduino IDE Version 1.6.5	18
3.1.2.2 Eclipse Mars.1 Version 4.5.1	19
3.1.2.3 JavaFx Scene Builder 2.0	19
3.1.2.4 Fritzing Version 0.9.2	20
3.2 การออกแบบส่วนฮาร์ดแวร์	20
3.2.1 การคำนวณกระแสไฟฟ้าที่วงจรใช้งาน	21

3.2.2 วิจารณ์การเชื่อมต่ออุปกรณ์	21
3.2.3 การทำงานของระบบอาร์ดิแวร์	24
3.3 การออกแบบส่วนซอฟแวร์	24
3.4 การออกแบบข้อมูลสำหรับโปรแกรมควบคุมสั่งไปยัง WIPLED	27
บทที่ 4 ผลการวิจัยและอภิปรายผล	30
4.1 การทดลอง	30
4.1.1 การทดสอบการรับ DHCP ของ NodeMCU เมื่อเชื่อมต่อกับ Wireless Network	30
4.1.2 การทดสอบการจำลองตัว NodeMCU เป็น Client	31
4.1.3 การทดสอบการรับส่งข้อมูลผ่าน Wi-Fi โดยใช้ Arduino Nano กับ ESP8266-12E	31
4.1.4 ทดสอบการใช้ AT Command ในตั้งค่า Wi-Fi	33
4.2 การทดสอบการเชื่อมต่อของอุปกรณ์อิเล็กทรอนิกส์หรือเซนเซอร์ต่างๆ	34
4.2.1 การทดสอบใช้ Capacitive Touch Sensor MPR121	34
4.2.2 การทดสอบใช้ Capacitive Touch Sensor MPR121 ร่วมกับ NodeMCU	35
4.2.3 การทดสอบใช้ Capacitive Touch Sensor MPR121 เพื่อทำเป็น Capacitance Near Proximity Detection	36
4.2.4 การทดสอบใช้ RGB LED Strip WS2812B	37
4.3 การทดสอบ Server ของโปรแกรมควบคุม	38
4.3.1 การทดสอบการใช้ Socket Server/Client	38
4.3.2 การทดสอบการเชื่อมต่อ Client หลายตัวพร้อมกันบน Server	39
4.4 การทดสอบโปรแกรมควบคุมร่วมกับ WIPED	40
4.4.1 การทดสอบการทำงานของโปรแกรมควบคุม WIPLED ในโหมด Synchronize	40
4.4.2 การทดสอบการทำงานของโปรแกรมควบคุม WIPLED ในโหมด Interactive	46
4.5 ชิ้นงาน WIPLED ที่เสร็จสมบูรณ์	48
บทที่ 5 สรุปผล	52
5.1 ตารางแสดงสถานะความสำเร็จของงานในแต่ละส่วนของโครงการ	52
5.2 ปัญหาหรือประเด็นที่พบในการทำงานและแนวทางการแก้ไข	53
5.3 ข้อจำกัดของโครงการ	54
5.4 แผนการดำเนินงานในอนาคต	54

5.5 การนำโครงการไปใช้งาน	54
บรรณานุกรม	55
ภาคผนวก	57

รูปภาพประกอบ

รูปภาพ	หน้า
2.1 RGB color	6
2.2 LED RGB Strip	7
2.3 ส่วนประกอบของพร็อกซิมิตี้ชนิดเก็บประจุ	8
2.4 SIX-FORTY BY FOUR-EIGHTY	9
2.5 แสดงการ touch SIX-FORTY BY FOUR-EIGHTY	10
2.6 แสดงผลของ SIX-FORTY BY FOUR-EIGHTY	10
2.7 งานแสดงโชว์ Art installation	11
3.1 บอร์ดควบคุม Arduino Nano	12
3.2 RGB LED Strip WS2812B	14
3.3 แสดงข้อกำหนดสำหรับสัญญาณดิจิทัลที่ใช้กำหนดค่าบิตแต่ละบิต สำหรับ WS2812B ซึ่งเริ่มต้นด้วยช่วง HIGH และตามด้วยช่วง LOW	15
3.4 Capacitive Touch Sensor MPR121	16
3.5 Wi-Fi Module ESP8266-12E	17
3.6 Arduino IDE Version 1.6.5	18
3.7 Eclipse Mars.1 Version 4.5.1	18
3.8 JavaFx Scene Builder 2.0	19
3.9 Fritzing Version 0.9.2	19
3.10 ภาพรวมของอุปกรณ์ที่ใช้สำหรับชิ้นงาน	20
3.11 ภาพวงจรซีอัมต่อของอุปกรณ์	21
3.12 ภาพวงจรซีอัมต่ออุปกรณ์เพื่อทำ PCB	22
3.13 ภาพลายวงจร PCB	22
3.14 การทำงานของ WIPLED	23
3.15 Use case Diagram	24
3.16 การทำงานของโปรแกรม Server	25
3.17 การทำงานของโใหมด Synchronize	26
3.18 การทำงานของโใหมด Interactive	26
3.19 รูปแบบการส่งข้อมูลในส่วนของ Mode 1	27
3.20 รูปแบบการส่งข้อมูลในส่วนของ Mode 2	28
3.21 รูปแบบการส่งข้อมูลในโใหมด Interactive	29

4.1 แสดงการรับ DHCP ของ NodeMCU	30
4.2 แสดงข้อความเมื่อ NodeMCU ทำการเชื่อมต่อกับ Server สำเร็จ	31
4.3 ภาพแสดง Arduino กับ module Wi-Fi	32
4.4 รูปวงจรการเชื่อมต่อ Arduino Nano , Logic level Converter และ โมดูล Wi-Fi ESP8266	32
4.5 การทดสอบตั้งค่าโมดูล Wi-Fi ESP8266 ผ่าน AT Command	33
4.6 ตัวอย่างการต่อวงจรเพื่อทำการทดสอบ Capacitive Touch Sensor MPR121	34
4.7 แสดงผลการทำงานของ Capacitive Touch Sensor MPR121	35
4.8 ภาพตัวอย่างการใช้ Capacitive Touch Sensor MPR121 ร่วมกับ NodeMCU	36
4.9 การเชื่อมต่อแผ่นพอยล์กับเซนเซอร์	37
4.10 ภาพทดสอบการสัมผัสแบบ Capacitance Near Proximity Detection	37
4.11 แสดงผลของ RGB LED Strip WS2812B	37
4.12 ทดสอบการรับ-ส่งข้อมูลระหว่าง Client และ Server	38
4.13 แสดงการเชื่อมต่อของ Client หลายตัวมายัง Server พร้อมกัน	39
4.14 หน้าจอ User interface เมื่อทำการกดปุ่ม “Start Server”	40
4.15 หน้าจอ User interface เมื่อทำการกดปุ่ม “Waiting”	41
4.16 หน้าจอ User interface เมื่อทำการกดปุ่ม “Update Client”	41
4.17 แสดง Monitor ของ WIPLED เมื่อเชื่อมต่อ Server ของโปรแกรมได้สำเร็จ	42
4.18 หน้าจอ User interface เมื่อทำการกดปุ่ม “Set”	42
4.19 หน้าจอ User interface เมื่อทำการกดปุ่ม “Send”	43
4.20 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูล Pattern สีจากโปรแกรมควบคุม	43
4.21 หน้าจอ User interface เมื่อทำการกดปุ่ม “Play”	44
4.22 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูลให้เริ่มเล่น Pattern	44
4.23 หน้าจอ User interface เมื่อทำการกดปุ่ม “Delete”	45
4.24 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูลให้ลบข้อมูล Pattern ทั้งหมดใน WIPLED	45
4.25 หน้าจอ User interface เมื่อทำการกดปุ่ม “Set”	46
4.26 หน้าจอ User interface เมื่อทำการกดปุ่ม “Reset”	47
4.27 หน้าจอ User interface เมื่อทำการกดปุ่ม “Set color”	47
4.28 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูล การตั้งค่าสีในโหมด Interactive	48
4.29 แผ่น PCB ด้านลายทองแดงของ WIPLED	48
4.30 แผ่น PCB ด้านลงอุปกรณ์	49

4.31 WIPLED ทั้ง 10 ชิ้น	49
4.32 การทำงานของ WIPLED เมื่อโปรแกรมควบคุมส่งข้อมูลให้แสดงสี	50
4.33 การทำงานของ WIPLED ในโหมด Sychronize แบบ Pattern (mode 1)	50
4.34 การทำงานของ WIPLED ในโหมด Sychronize แบบกำหนดสีได้ (mode 2)	51
4.35 การทำงานของ WIPLED ในโหมด Interactive	51

บทที่ 1

บทนำ

โครงการนี้จะจัดสร้างอุปกรณ์ Art Installation ที่สามารถมีปฏิสัมพันธ์กับการสัมผัสของผู้ใช้งาน โดยจะแบ่งการใช้งานออกเป็นสองส่วน คือส่วนที่ผู้ดูแลเป็นผู้ควบคุมการแสดงออกของสีและส่วนที่ผู้ใช้งานสามารถสัมผัสให้เกิดการเปลี่ยนแปลงตอบสนองต่อการสัมผัสนั้น ซึ่งตัวชิ้นงานนั้นสามารถเปลี่ยนแปลงรูปแบบเพื่อให้เหมาะสมกับงานอีเว้นหรือพื้นที่ที่จะนำไปติดตั้ง เช่น วัตถุทรงกลมเพื่อนำไปติดตั้งตามทางเดิน วัตถุรูปดาวเพื่อนำไปแขวนบนต้นคริสต์มาส เป็นต้น

1.1 ที่มาของปัญหาและแนวทางการแก้ปัญหา

เนื่องด้วยในปัจจุบันในต่างประเทศ Digital Art Installation ได้รับความนิยมและแพร่หลายอย่างมากในการจัดงานแสดงorchestra และเทศกาลต่าง ๆ เพราะสามารถดึงดูดความสนใจจากผู้เข้าชมที่มาร่วมงานได้มาก แต่ส่วนใหญ่ในประเทศไทยงานประเภทนี้ยังไม่แพร่หลายมากและชิ้นงานที่มีอยู่ในปัจจุบันเป็นชิ้นงานที่ไม่สามารถมีปฏิสัมพันธ์กับมนุษย์ได้ ทางผู้จัดทำจึงเกิดความคิดที่จะพัฒนางานประเภทนี้โดยให้ผู้เล่นมีปฏิสัมพันธ์กับชิ้นงานได้ ซึ่งจะช่วยดึงดูดความสนใจและสร้างการมีส่วนร่วมของผู้เข้าชมกับชิ้นงานที่จัดแสดง

1.2 วัตถุประสงค์ของโครงการ

- เพื่อสร้างชิ้นงานตกแต่งที่สามารถมีปฏิสัมพันธ์กับผู้ใช้งาน
- เพื่อนำเครื่อข่ายไร้สายมาประยุกต์ใช้เข้ากับชิ้นงานตกแต่ง
- ใช้เทคโนโลยี RGB LED Strip แบบ Individual addressable
- เพื่อทำให้งานนิทรรศการมีความน่าตื่นตาตื่นใจ

1.3 ขอบเขตของงานวิจัย

- จำนวนชิ้นงานทั้งหมด 10 ตัว
- ระยะการส่งสัญญาณจาก Server ไปยัง WIPLED ไม่เกิน 50 เมตร
- แสดงผลลูกอมในรูปแบบของสีเท่านั้น
- ใช้ access point ในการรับ-ส่งสัญญาณ 1 ตัว

1.4 ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน

- มีทักษะและประสบการณ์ในการพัฒนาชิ้นงานที่สามารถมีปฏิสัมพันธ์กับผู้ใช้งาน
- นำความรู้ทางด้านเครือข่ายไร้สายมาประยุกต์ใช้กับฮาร์ดแวร์ให้เกิดประโยชน์ได้จริง
- สามารถเป็นชิ้นงานต้นแบบในการทำอุปกรณ์ตกแต่งที่สามารถมีปฏิสัมพันธ์กับผู้ใช้งาน
- มีทักษะการใช้อุปกรณ์ฮาร์ดแวร์ เครื่อข่ายไร้สายและโปรแกรมควบคุมชิ้นงานเพิ่มมากขึ้น

1.5 แผนการดำเนินงาน

Task/Date	deliverable	August				September				October				November				December				January				Febuary				March				April			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1.ปรึกษาเรื่องโปรเจค																																					
2.วางแผนหาข้อมูล																																					
3.ส่ง proposal	Project proposal																																				
4.เตรียมการ present proposal	Proposal presentation																																				
5.ศึกษาและสำรวจการทำงานของอุปกรณ์																																					
5.1 ศึกษาและสำรวจการใช้งาน protocol การรับ-ส่งข้อมูล																																					
5.2 ศึกษา Touch sensor																																					
5.3 ศึกษาการออกแบบ User Interfaces																																					
5.4 ศึกษาและสำรวจการออกแบบ protocol สำหรับ																																					
5.5 ศึกษาและสำรวจการออกแบบชิ้นงาน																																					
5.6 วิเคราะห์ข้อมูลที่ศึกษาและสำรวจ																																					
6.ส่ง report draft I	Report draft I																																				
7.ออกแบบ																																					
7.1 ออกแบบ User Interfaces																																					
7.2 ออกแบบ protocol สำหรับเครือข่ายไร้สาย																																					
7.3 ออกแบบชิ้นงาน																																					
8.พัฒนาซอฟแวร์และฮาร์ดแวร์																																					
8.1 พัฒนาซอฟแวร์เพื่อควบคุม module																																					
8.2 พัฒนาฮาร์ดแวร์ (module)																																					
9. Final presentation I	Final presentation I																																				

บทที่ 2

ที่มา ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

โครงการนี้ได้มีการรวบรวมความรู้ในด้านต่างๆ ที่มีประโยชน์ ทั้งที่เกี่ยวข้องในด้านวิศวกรรมคอมพิวเตอร์ และในด้านอื่นๆ เพื่อทำการพัฒนาโครงการวิจัยให้ประสบผลสำเร็จมากที่สุด โดยได้มีการทำการทำศึกษาและวิจัย โครงการที่มีลักษณะเกี่ยวข้อง หรือคล้ายคลึงกับโครงการที่นักวิจัยทำการพัฒนา การทบทวนวรรณกรรม อุปกรณ์ ที่ใช้ในการควบคุมและประมวลผล อุปกรณ์อิเล็กทรอนิกส์ต่างๆ และแสง สี จากหลอดไฟ

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ดังตารางที่ 2.1 คือ อุปกรณ์ควบคุมขนาดเล็กที่ประกอบไปด้วย หน่วยประมวลผลกลาง หน่วยความจำ พอร์ตอินพุต/เอาต์พุต ช่องทางเดินทางสัญญาณเพื่อรับ-ส่งข้อมูล และมีพอร์ตต่างๆ เพิ่มเติมขึ้นอยู่กับผู้ผลิต เช่น ADC , I2C , SPI , PWM , UART ซึ่งเพิ่มความสามารถให้ไมโครคอนโทรลเลอร์ทำงานได้หลากหลายมากขึ้น ซึ่งทั้งหมดรวมอยู่ภายในไอซีตัวเดียวกันและสามารถเขียนโปรแกรมเพื่อควบคุมการทำงานตามต้องการได้

ตารางที่ 2.1 แสดงการเปรียบเทียบคุณสมบัติไมโครคอนโทรลเลอร์

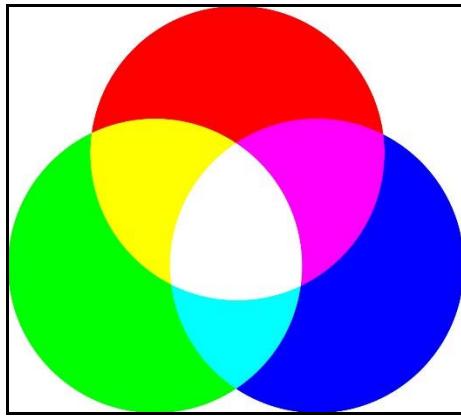
Wi-Fi module	Wi-Fi Shield V2 RN171	CC3000 Wi-Fi Shield	Spark Core with Chip Antenna	Node MCU
ภาพ				
MCU ที่รองรับการใช้งาน	Arduino UNO R3	Arduino UNO R3	มี ARM Cortex-M3 32-bit ในตัว	Tensilica L106 32-bit micro controller built in ใน ESP8266
Network	IEEE 802.11 b/g	IEEE 802.11 b/g	IEEE 802.11 b/g	IEEE 802.11 b/g/n

ภาษาที่ใช้พัฒนา	Arduino ภาษา C	Arduino ภาษา C	Wiring คล้ายกับ Arduino เป็นภาษา C	Arduino ภาษา C , LUA
PORT	14 digital input/output pins 6 analog inputs	14 digital input/output pins 6 analog inputs	8 digital I/O pins 4 PWM pins 8 analog I/O pins	10 GPIO, every GPIO can be PWM, I2C, 1-wire
Size	5.3 cm x 5.1 cm	5.3 cm x 5.1 cm	2.5 cm x 4.6 cm	2.54 cm x 4.83 cm
ข้อดี	สามารถใช้ร่วมกับเซนเซอร์ประเภทอื่นได้หลากหลาย มี library สำหรับ Arduino ให้ใช้	สามารถใช้ร่วมกับเซนเซอร์ประเภทอื่นได้หลากหลาย มี library สำหรับ Arduino ให้ใช้	ARM มีประสิทธิภาพการทำงานที่ดีกว่า ATmega Built in Wi-Fi ในตัว	สามารถใช้ได้กับ library Arduino ได้เกือบทั้งหมด MCU built in Wi-Fi
ข้อด้อย	การใช้งานไม่แพร์หลาย มีตัวอย่างน้อย Network ไม่รองรับ n	การใช้งานไม่แพร์หลาย มีตัวอย่างน้อย Network ไม่รองรับ n	บาง library ไม่สามารถใช้ร่วมกับ wiring ได้ Network ไม่รองรับ n	บาง library ต้องทำการแก้ไขคำสั่งก่อนถึงจะใช้งานได้
ราคา	Arduino UNO 790 บาท Wi-Fi Shield 2450 บาท	Arduino UNO 790 บาท Wi-Fi Shield 1650 บาท	Spark Core 2200 บาท	NodeMCU 490 บาท

จากตารางที่ 2.1 การเปรียบเทียบไมโครคอนโทรลเลอร์กับโมดูลเครือข่ายไร้สาย พบร่วมกันว่า NodeMCU มีความเหมาะสมมากที่สุด เนื่องจาก ไมโครคอนโทรลเลอร์กับโมดูลเครือข่ายไร้สายรวมอยู่ใน ESP8266 ตัวเดียว ทำให้ง่ายต่อการใช้งานและลดขนาดของวงจรไฟฟ้าลงได้อย่างมาก

2.1.2 RGB color

RGB [1] ดังรูปที่ 2.1 ซึ่งเป็นระบบสีพื้นฐานที่ใช้ในการแสดงผล RGB ได้แก่ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) การผสมสีทั้งสามนี้ด้วยค่าต่างๆกัน จะก่อให้เกิดสีที่แตกต่างกัน โดยแต่ละสีจะเก็บข้อมูลขนาด 8 bit 0-255 ทำให้แต่ละสีมีค่าของสี 256 ระดับ และเมื่อนำมา 3 สีมาผสมกันจะได้สีทั้งหมด 16 ล้านสี



รูปที่ 2.1 RGB color

[ที่มา: <http://www.lcipaper.com/pantone-cmyk-rgb.html>]

2.1.3 Individually addressable LED

เป็น LED ที่มีตัวควบคุมขนาดเล็กอยู่ภายในหลอด LED หรือเชื่อมต่อกับ LED เมื่อนำมาต่อ กัน เป็นเส้น จะทำให้สามารถควบคุมความสว่างและสีได้ ซึ่งแต่ละดวงสามารถควบคุมได้อิสระต่อกันโดยใช้สายส่งข้อมูล เพียง 1-2 สายรับข้อมูลจากไมโครคอนโทรลเลอร์ การส่งข้อมูลเพื่อควบคุมสีแต่ละดวงนั้นจะขึ้นอยู่กับ จำนวนบิตของสี RGB ที่ LED รุ่นนั้นสามารถแสดงได้

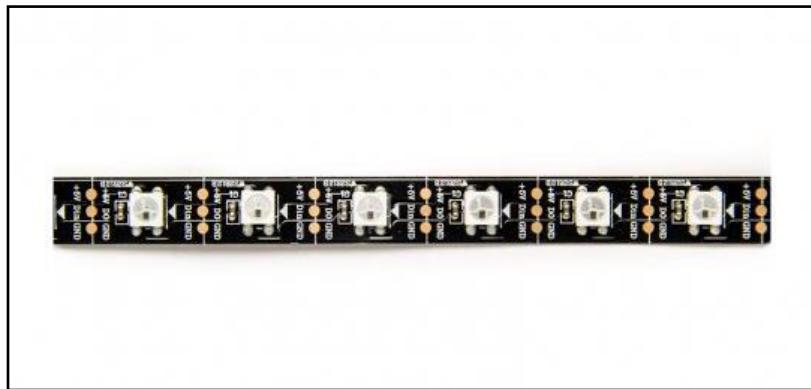
การเลือกใช้ RGB LED สำหรับการแสดงสีของชิ้นงาน

RGB LED ดังรูปที่ 2.2 จะเป็นสิ่งที่แสดงสีของชิ้นงานเมื่อเกิดเหตุการณ์ต่างๆ ซึ่ง RGB LED แต่ละ ชนิดนั้นจะมีความแตกต่างของจำนวนสีและระดับความสว่างดังตารางที่ 2.2

ตารางที่ 2.2 แสดงการเปรียบเทียบคุณสมบัติ individually addressable RGB LED

IC control LED	LPD6803	WS2811	WS2812
Wire	4 Wire (Data,CLK,Vcc,GND)	3 Wire (Data,Vcc,GND)	3 Wire (Data,Vcc,GND)
Power	12 Volt	5 Volt	5 Volt
Color	32,768 colors	16,777,216 colors	16,777,216 colors
Dimming level	128 Level	256 Level	256 Level
Pin	8 Pin	5 Pin	4 Pin
Chip Built in LED	No	No	Yes
Price (Strip 30pcs/m)	250	270	300

จากข้อมูลในตารางที่ 2.2 พบว่า RGB LED WS2812B มีความใกล้เคียงกับ WS2811 มากจึงได้ทำการศึกษาเพิ่มเติมพบว่า WS2812B เป็นการพัฒนามาจาก WS2811 ซึ่งจะมีข้อดีเพิ่มขึ้นคือ ให้ ความสว่างได้มากกว่าและเพิ่มวงจรที่ช่วยป้องกันให้ LED ลดอัตราการเสียหายจากแหล่งจ่ายไฟ WS2812B จึงเหมาะสมมากที่สุด

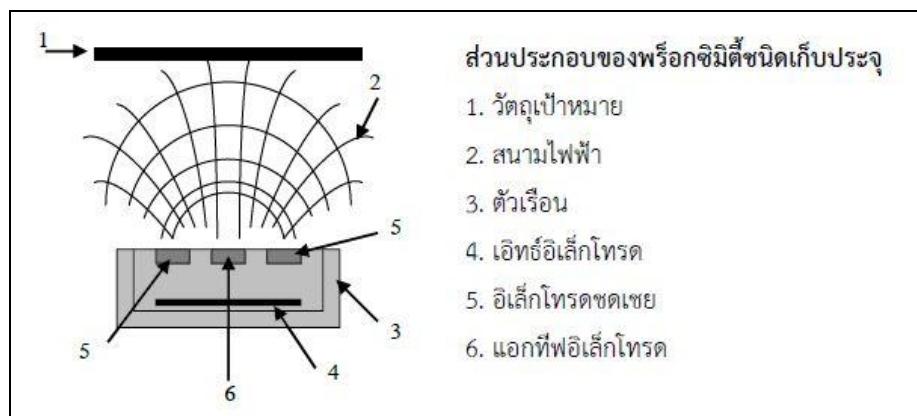


รูปที่ 2.2 LED RGB Strip

[ที่มา: https://store.arduino.cc/bmz_cache/0/0b9d1db06809416503fe9a41fcf29122.image.538x354.jpg]

2.1.4 Capacitive proximity sensor

Capacitive proximity sensor [2] เป็นเซนเซอร์สำหรับตรวจจับวัตถุที่เป็นโลหะและอโลหะ เพียงวัตถุเข้าใกล้หรือสัมผัสกับเซนเซอร์ พร้อมก็มิตี้เซนเซอร์ชนิดเก็บประจุทำงานโดยอาศัยหลักการเปลี่ยนแปลงค่าประจุไฟฟ้า เมื่อวัตถุเคลื่อนที่เข้ามาใกล้สนามไฟฟ้า ค่าความจุจะเกิดความเปลี่ยนแปลงจนถึงค่าที่เท่ากับค่าความต้านทานที่ปรับไว้ในตอนเริ่มต้น จะส่งผลให้เกิดสัญญาณออกซิลเลทขึ้น และส่งผลให้อาตโนมัติทำงาน รูปที่ 2.3 แสดงส่วนประกอบและการทำงานของพร็อกซิมิตี้เซนเซอร์ชนิดเก็บประจุ



รูปที่ 2.3 ส่วนประกอบของพร็อกซิมิตี้ชนิดเก็บประจุ

[ที่มา: <http://www.riko.in.th/knowledge-sensor/61-what-is-capacitive-sensor.html>]

2.1.5 การเลือกใช้มาตรฐานของการส่งข้อมูลผ่านเครือข่ายไร้สาย

ตารางที่ 2.3 แสดงมาตรฐานการสื่อสารไร้สายในปัจจุบัน

Features	Wi-Fi	Bluetooth	Zig bee
Frequency	2.4 GHz	2.4 GHz	2.4 GHz
Range	100 – 150 ft.	30 – 300 ft.	30 m – 1.6 km
Cost	High	Low	Low
Power Consumption	Low	High	Very Low
Data Rate	11 – 54 Mbps	1 Mbps	250 Kbps
Security	128 bits	64 or 128 bits	128 bit

จากตารางที่ 2.3 มาตรฐานการสื่อสารแบบไร้สาย เมื่อคำนึงถึงการส่งข้อมูลที่ต้องใช้ความเร็วและมีเดลai ใน การส่งน้อยที่สุด ขนาดพื้นที่การใช้งานไม่มากนัก ไวไฟ (Wi-Fi) จึงเป็นมาตรฐานการสื่อสารแบบไร้สายที่เหมาะสมที่สุดสำหรับการรับส่งข้อมูล

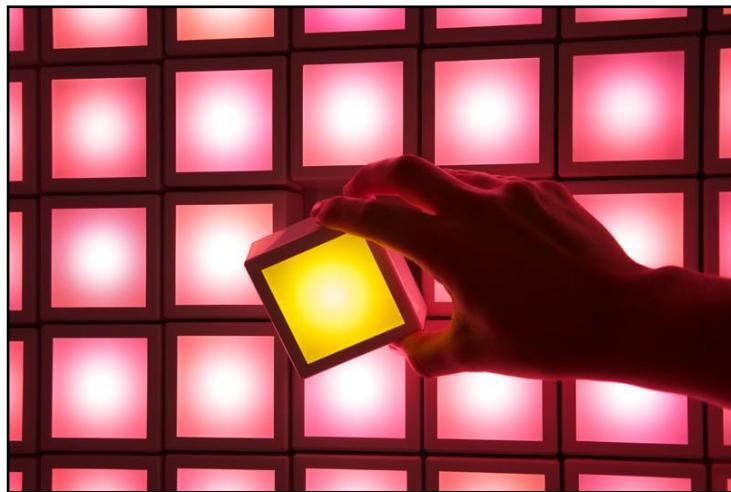
มาตรฐานการสื่อสารที่เลือกใช้ คือ Wi-Fi

Wi-Fi คือ องค์กรหนึ่งที่ทำการทดสอบผลิตภัณฑ์ Wireless LAN หรือระบบ Network แบบไร้สาย ด้วย เทคโนโลยีการสื่อสารภายใต้มาตรฐาน IEEE 802.11 ซึ่งอุปกรณ์ทุกตัวสามารถติดต่อสื่อสารกันได้โดยไม่มีปัญหา หากอุปกรณ์นั้นผ่านตามเกณฑ์มาตรฐานก็จะมีการประทับตรา Wi-Fi Certified ซึ่งหมายความว่า อุปกรณ์ตัวนี้ สามารถเชื่อมต่อแบบไร้สายกับ อุปกรณ์อื่นที่มีตรา Wi-Fi Certified ได้

2.2 งานวิจัยที่เกี่ยวข้อง

ในปัจจุบัน Art installation เป็นงานที่ได้รับความนิยมสูงมากในงานแสดงต่างๆ แต่การทบทวน วรรณกรรมจะศึกษางานที่มีลักษณะใกล้เคียงกับโครงงานที่ทำ ซึ่งเป็นชิ้นงานที่เคยนำเสนอมา แสดงในงานแสดง โชว์ต่างๆ โดยจะเน้นงานที่คนสามารถมีปฏิสัมพันธ์กับชิ้นงานได้

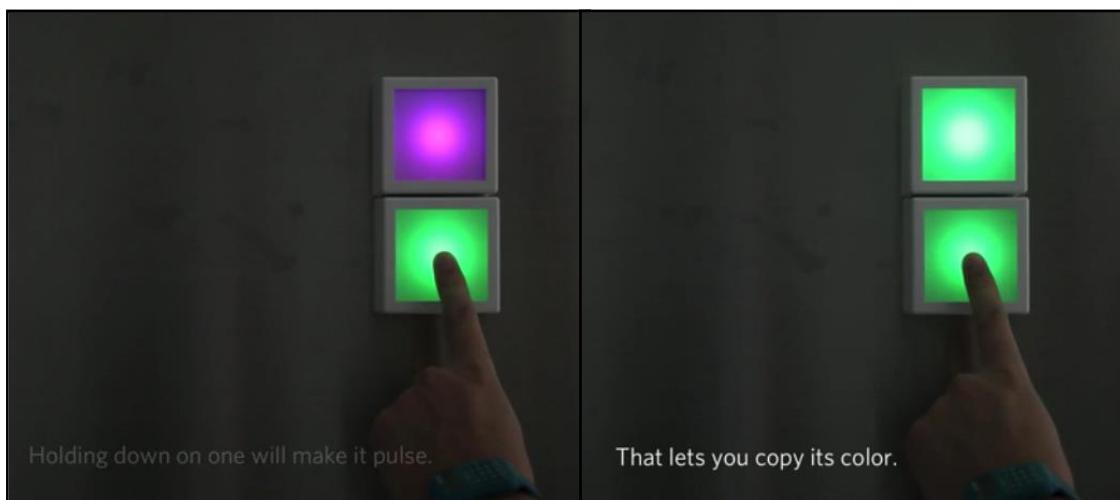
2.2.1 SIX-FORTY BY FOUR-EIGHTY



รูปที่ 2.4 SIX-FORTY BY FOUR-EIGHTY

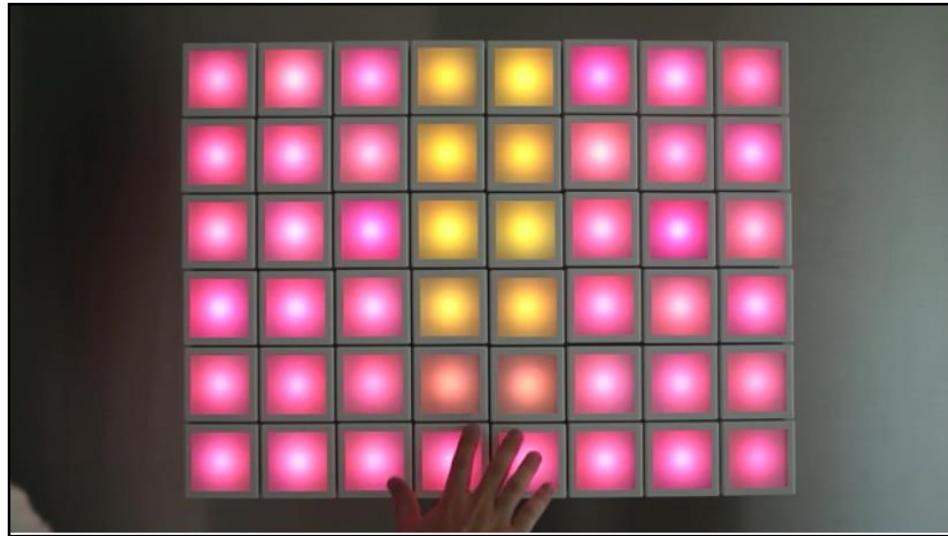
โครงงานนี้เป็นโครงงานประเภท Interactive Light Installation [10] มีลักษณะเป็นกล่องสี่เหลี่ยม(พิกเซล) ดังรูปที่ 2.4 พิกเซลแต่ละตัวเป็นอิสระต่อกัน สามารถเชื่อมต่อกันด้วยแม่เหล็กที่อยู่ด้านข้าง มีเซนเซอร์ Capacitive touch เป็นตัวรับค่าจากการสัมผัสโดยค่าจากการสัมผัสจะส่งผลให้เกิดการเปลี่ยนแปลงของสีแต่ละพิกเซลและส่งข้อมูลไปยังพิกเซลตัวข้างที่เชื่อมต่อกันผ่านแม่เหล็ก สามารถสร้างรูปแบบและภาพเคลื่อนไหวได้โดยใช้ออฟเวอร์ มีรูปแบบการทำงานดังนี้

- เมื่อผู้ใช้สัมผัสพิกเซลจะเปลี่ยนสีได้ตามที่ซอฟแวร์ได้ตั้งค่าไว้
- ผู้ใช้สามารถคัดลอกสีพิกเซลที่เชื่อมต่อกันได้โดย สัมผัสพิกเซลสีที่ต้องการคัดลอกค้างไว้จนเกิดการกระพริบ หลังจากนั้นสัมผัสพิกเซลที่ต้องการใช้สีที่ถูกคัดลอกจนสีเปลี่ยนเป็นสีเดียวกันดังรูปที่ 2.5



รูปที่ 2.5 แสดงการ touch SIX-FORTY BY FOUR-EIGHTY

3. ผู้ใช้สามารถใช้รูปโน๊ตคอนโทรลควบคุมสีของพิกเซลได้โดยจะเปลี่ยนรูปแบบตามที่ซอฟแวร์ได้กำหนดไว้
4. เมื่อนำพิกเซลมาเชื่อมต่อรวมกันจะเหมือนบอร์ด LED ที่สามารถสัมผัสแล้วเกิดการเปลี่ยนแปลงของสีได้ดังรูปที่ 2.6

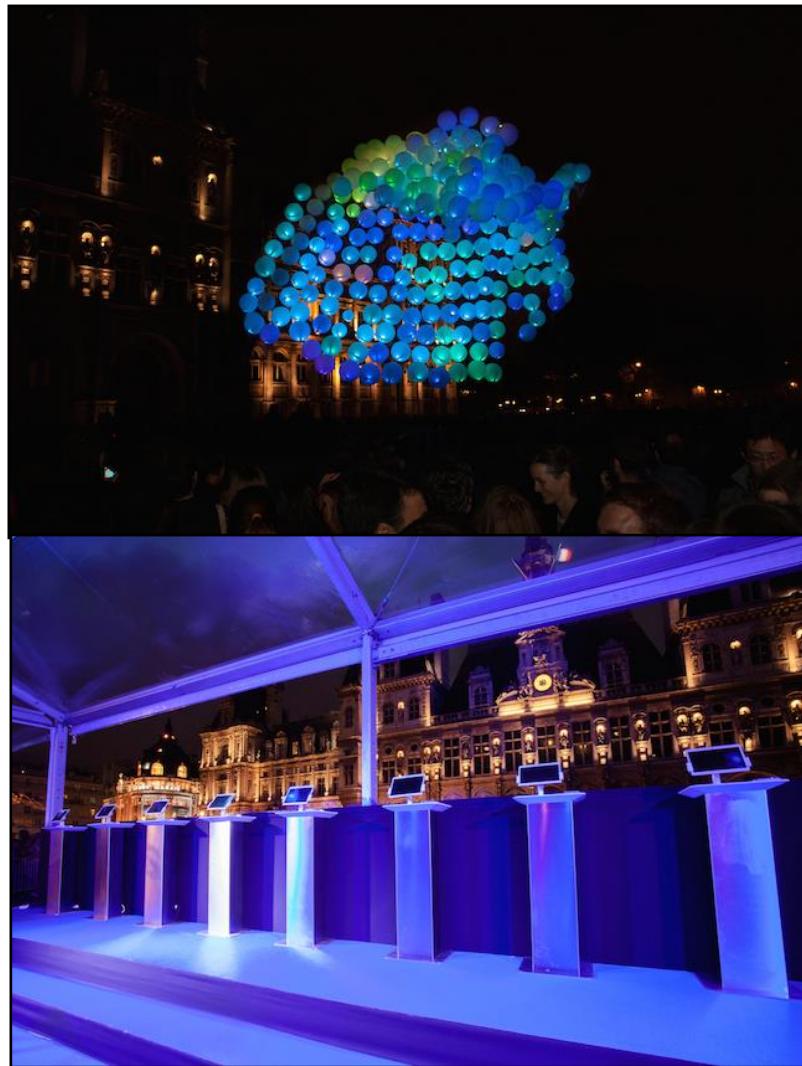


รูปที่ 2.6 แสดงผลของ SIX-FORTY BY FOUR-EIGHTY

[ที่มา: <http://fluid.media.mit.edu/projects/six-forty-four-eighty>]

2.2.2 Mini Bubble Paris

เป็นงานประเภท Interactive Light Installation [9] ได้รับการสนับสนุนจาก Samsung Electronics ดังรูปที่ 2.7 ให้จัดทำขึ้นเพื่อแสดงในงาน Nuit Blanche festival in Paris มีลักษณะเป็นบล็อกลูน 300 ลูก ภายในมีหลอด LED แสดงสี เรียงต่อกันเป็นแผง ซึ่งสามารถให้ผู้ร่วมงานใช้ Samsung Galaxy Tab S tablets ควบคุมสีของบล็อกลูนโดยการระบายน้ำ และรองรับผู้ใช้งานหลายคนพร้อมๆ กัน



รูปที่ 2.7 งานแสดงโชว์ Art installation

[ที่มา: <http://www.tetro.fr/en/tetro/tetro-realizations/samsung-electronics-mini-bubble-paris>]

จากการศึกษางานที่เกี่ยวข้องพบว่าในแต่ละงานนั้นจะออกแบบมาเฉพาะวัตถุประสงค์ของงานแสดงนั้นๆ ทำให้เกิดข้อจำกัดในการที่จะนำไปประยุกต์ใช้กับงานแสดงอื่นๆ ทางเรามีจุดที่จะพัฒนางานประเภท Interactive art installation ที่มีลักษณะเป็นโมดูลที่ติดต่อกันด้วยระบบเครือข่ายไร้สายสามารถวางแผนแยกกันได้อย่างอิสระ สามารถออกแบบผลิตภัณฑ์มาใส่โมดูลนี้เพื่อให้เข้ากับคอนเซ็ปต์ของงานได้อีกด้วย เช่น ออกแบบเป็นของเล่นต่างๆ ไปแขวนบนต้นคริスマสต์ เป็นกล่องวางกระจาดอยู่ในพื้นที่ต่างๆ โดยจะนำอุปกรณ์ต่างๆที่ได้สืบคันและเบรียบเทียบข้างตันมาพัฒนาให้เกิดเป็น WIPLED ในบทดังไป

บทที่ 3

การออกแบบและระบบวิธีวิจัย

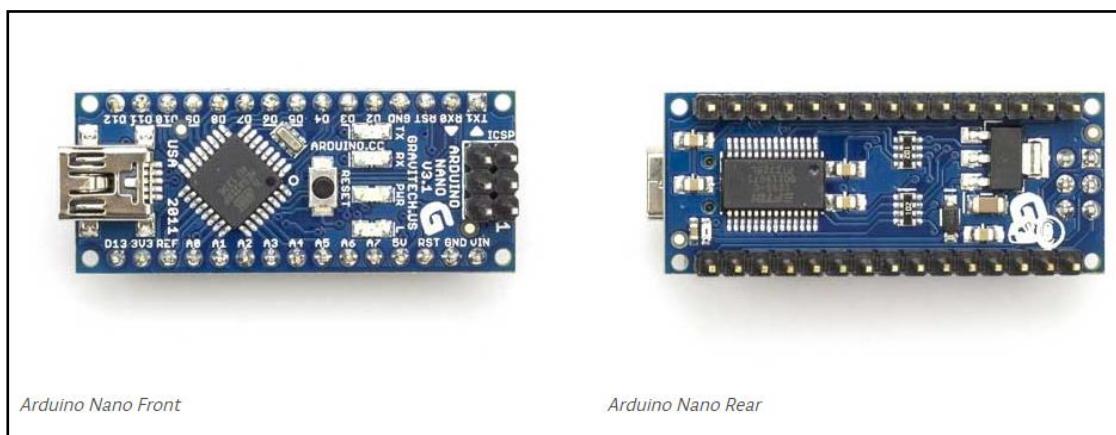
ในบทที่ 3 จะกล่าวถึงอุปกรณ์ที่ได้ทำการเลือกใช้ในการพัฒนาและการออกแบบ WIPLED ที่มีปฏิสัมพันธ์กับผู้ใช้งานและสามารถแสดงแสงได้ตามการควบคุมของโปรแกรม โดยจะมีทั้งหมด 4 ข้อดังนี้ ข้อที่หนึ่งอุปกรณ์และซอฟแวร์ที่ใช้ในการทำงาน จะกล่าวถึงคุณสมบัติของอุปกรณ์ทั้งหมดที่เลือกใช้ทั้งหมด ข้อที่สองการออกแบบจริงการทำงานที่ใช้สำหรับรับ-ส่งข้อมูลระหว่าง Server และส่วนที่มีปฏิสัมพันธ์กับผู้ใช้งาน ข้อที่สามการออกแบบโปรแกรมทดสอบสำหรับ WIPLED จะกล่าวถึง ส่วนประกอบของโปรแกรม หน้าจอการแสดงผล การทำงานของโปรแกรม และข้อที่สี่การออกแบบข้อมูลสำหรับโปรแกรมควบคุมส่งไปยัง WIPLED เพื่อสั่งการให้แสดงสี ซึ่งในโครงการได้เน้นการออกแบบส่วนฮาร์ดแวร์หรือ WIPLED เป็นหลัก

3.1 อุปกรณ์ที่ใช้ในการทำโครงการ

3.1.1 ฮาร์ดแวร์

ในหัวข้อนี้จะกล่าวถึงคุณสมบัติของอุปกรณ์ทั้งหมดที่เลือกใช้ ซึ่งประกอบไปด้วย บอร์ดควบคุม ตลอดไฟ LED แบบเส้น เช่นเซอร์ชnidเก็บประจุ และ โมดูล Wi-Fi

3.1.1.1 บอร์ดควบคุม Arduino Nano



รูปที่ 3.1 บอร์ดควบคุม Arduino Nano

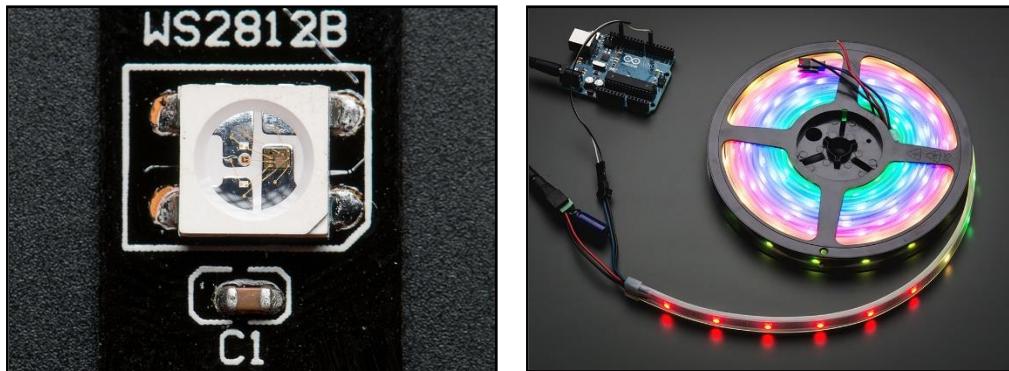
ตารางที่ 3.1 แสดงรายละเอียดของบอร์ดคอนโทรลเลอร์

Microcontroller	ATmel ATmega168
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz

Arduino Nano ดังรูปที่ 3.1 เป็นบอร์ดควบคุมขนาดเล็กที่ใช้ในการรับค่าและประมวลผล รวมไปถึงควบคุมเข็มเซอร์ต่างๆ จากนั้นทำการส่งค่าข้อมูลที่ได้ไปยังคอมพิวเตอร์หรือหน่วยประมวลผลเพื่อนำไปใช้งานกับโปรแกรมทดลองการทำงาน ในการแสดงผล และการรับ-ส่งข้อมูล ซึ่ง Arduino Nano นั้นมีลักษณะที่เหมาะสมกับการทำงานของโครงการ เนื่องจากตัวชิ้นงานต้องการขนาดเล็ก มีพอร์ตสำหรับรับ-ส่งค่า เพียงพอต่อใช้งาน และยังมีพอร์ตกระแสไฟฟ้าทั้ง 3.3 โวลต์และ 5 โวลต์ ซึ่งจำเป็นต้องใช้กับเซนเซอร์ตัวอื่น รายละเอียดแสดงดังตารางที่ 3.1

ในช่วงแรกที่ทำการทดลองได้ใช้งาน NodeMCU ตามที่ได้ทำการเปรียบเทียบจากบทที่ 2 พบร้า NodeMCU สามารถเชื่อมต่อกับ Capacitive Touch MPR121 ได้ในบางครั้ง ซึ่งส่งผลต่อการทำงานของระบบ จึงได้ทำการเปลี่ยนบอร์ดควบคุมเป็น Arduino Nano

3.1.1.2 RGB LED Strip WS2812B



รูปที่ 3.2 RGB LED Strip WS2812B

[ที่มา: <https://www.adafruit.com/products/1376>]

ตารางที่ 3.2 แสดงรายละเอียดของ RGB LED WS2812B

ผลิตภัณฑ์ (Product)	RGB LED WS2812B
Type	Individually addressable LEDs
Operating Voltage	3.5~5.3V
Working Current	20 mA , 50 mA(max) ต่อดวง
RGB color	24 bit , 16,777,216 color
Brightness display	256 level
Send data speed	800 Kbps
Communication	Single NZR communication (1 wire)

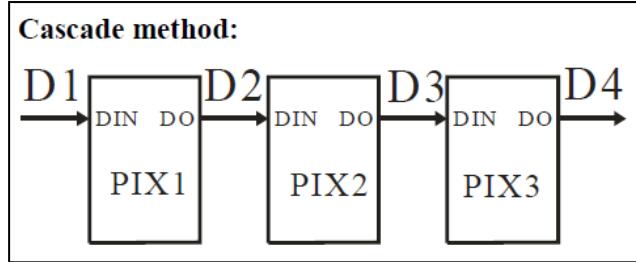
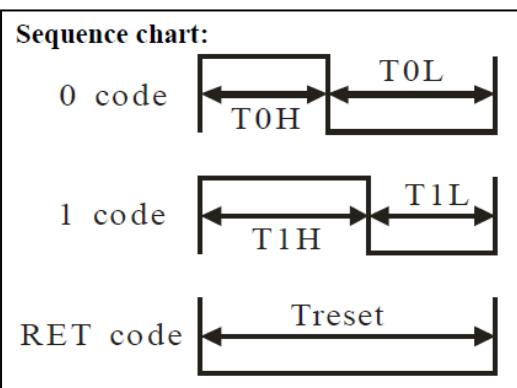
[ที่มา: <https://www.adafruit.com/datasheets/WS2812B.pdf>]

ใช้ RGB LED [3] ดังรูปที่ 3.2 สำหรับการแสดงค่าสีต่างๆ โดย Arduino Nano เป็นบอร์ดควบคุมสีและรูปแบบต่างๆ รายละเอียดแสดงดังตารางที่ 3.2

การควบคุมสีของ RGB LED ทำได้โดยการส่งข้อมูลควบคุมจาก Arduino ซึ่งจะต้องมี 24 บิต ต่อ RGB LED หนึ่งดวง ถ้ามีทั้งหมด N ดวง ที่ต้องเข้าด้วยกันแบบ Cascade จะต้องส่งข้อมูลทั้งหมด $24 \times N$ บิต เมื่อส่งข้อมูลครบตาม จำนวนดวงต้องส่ง RET code เพื่อเป็นการสิ้นสุดการส่งข้อมูลและการส่งข้อมูลจะไม่มีสัญญาณนาฬิกา แต่จะมีเพียงข้อมูลเท่านั้น และเพื่อกำหนดว่าบิตใดมีค่าเป็น 0 หรือเป็น 1 จะดูจากความกว้างของช่วงสัญญาณดังรูปที่ 3.3

Data transfer time(TH+TL=1.25μs±600ns)

T0H	0 code ,high voltage time	0.4us	±150ns
T1H	1 code ,high voltage time	0.8us	±150ns
T0L	0 code , low voltage time	0.85us	±150ns
T1L	1 code ,low voltage time	0.45us	±150ns
RES	low voltage time		Above 50μs



รูปที่ 3.3 แสดงข้อกำหนดสำหรับสัญญาณดิจิทัลที่ใช้กำหนดค่าบิตแต่ละบิต
สำหรับ WS2812B ซึ่งเริ่มต้นด้วยช่วง HIGH และตามด้วยช่วง LOW [4 , 7]
[ที่มา: <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>]

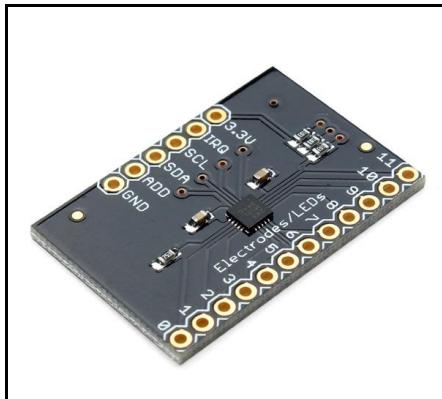
การส่ง Data ไปยัง LED 1ดวงต้องใช้ทั้งหมด 24 Bit โดยเริ่มจาก Green ,Red และ Blue ตามลำดับ

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Follow the order of GRB to sent data and the high bit sent at first.

3.1.1.3 Capacitive Touch Sensor MPR121



รูปที่ 3.4 Capacitive Touch Sensor MPR121

[ที่มา: <http://img.banggood.com/images/2014/xiemeijuan/03/SKU207358/SKU207358-4.jpg>]

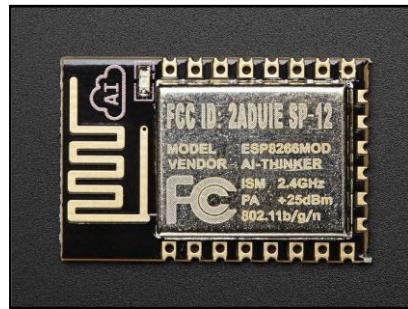
ตารางที่ 3.3 แสดงรายละเอียดของเซนเซอร์ Capacitive Touch MPR121

ผลิตภัณฑ์ (Product)	Proximity Capacitive Touch Sensor Controller
รุ่น	MPR121
Operating Voltage	1.71~3.6V
Working Current	29 μ A
Input (Touch pad)	12 electrodes/capacitance
Communication	I2C
I2C address	0x5A – 0x5D

[ที่มา: <https://www.sparkfun.com/datasheets/Components/MPR121.pdf>]

Capacitive Touch Sensor [5] ดังรูปที่ 3.4 เป็นอุปกรณ์สำหรับตรวจจับวัตถุ โดยตัวเซนเซอร์จะเป็น เสมือนสนามไฟฟ้า เมื่อมีวัตถุเข้ามาเหนี่ยวนำส่งผลให้ค่าความจุของสนามไฟฟ้าเกิดการเปลี่ยนแปลง เมื่อเกิดการเปลี่ยนแปลงถึงค่าหนึ่งเอาร์พุตจะทำงานโดยส่งข้อมูลผ่านการสื่อสารแบบ I2C ไปยังบอร์ดควบคุม ซึ่งเซนเซอร์นี้ สามารถปรับค่าความไวต่อการสัมผัสได้ (Near Proximity) เพียงแค่นำวัตถุที่เหนี่ยวนำมาใกล้แต่ยังไม่ถึงกับสัมผัส เชนเซอร์ก็สามารถทำงานได้ รายละเอียดแสดงดังตารางที่ 3.3

3.1.1.4 Wi-Fi Module ESP8266-12E



รูปที่ 3.5 Wi-Fi Module ESP8266-12E

[ที่มา : <https://www.adafruit.com/products/2491>]

ตารางที่ 3.4 แสดงรายละเอียดของโมดูล Wi-Fi ESP8266 12-E

Wi-Fi Protocols	802.11 b/g/n
Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control/GPIO/PWM
Operating Voltage	3.0~3.6V
Operating Current	Average value: 100mA
Wi-Fi mode	station/softAP/SoftAP+station
Security	WPA/WPA2
Network Protocols	IPv4, TCP/UDP/HTTP/FTP
User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App

ESP8266 ดังรูปที่ 3.5 คือ โมดูล Wi-Fi ที่มีขนาดเล็ก ใช้พลังงานน้อย และ รองรับการใช้งาน ได้ หลากหลายรูปแบบทั้ง Client, Access Point และ Client+AP ESP8266 ใช้การเชื่อมต่อด้วย Serial (UART 3.3V) จึงทำให้ง่ายต่อการนำไปใช้งานร่วมกับไมโครคอนโทรลเลอร์ และมีรูปแบบคำสั่งแบบ AT Command ใน การตั้งค่า ซึ่งทางกลุ่มได้นำมาใช้ในส่วนของกล่องคอลโทรล เป็น Client เพื่อรับ-ส่งข้อมูลจาก Server เนื่องจากได้เปลี่ยนจาก NodeMCU มาเป็น Arduino Nano ซึ่งไม่มีโมดูล Wi-Fi ทำให้ไม่สามารถสื่อสารโดยใช้ เครือข่ายไร้สายได้ จึงต้องเพิ่ม Wi-Fi module เข้าไป ทางกลุ่มจึงเลือก ESP8266-12E ซึ่งเป็นตัวเดียวกับที่ NodeMCU ใช้งาน แต่การใช้งานจะแตกต่างกันตรงที่ NodeMCU จะมี Library ใน Arduino ให้ใช้สำหรับรูป สำหรับการตั้งค่า Wi-Fi รายละเอียดของโมดูลแสดงดังตารางที่ 3.4

3.1.2 ซอฟต์แวร์

3.1.2.1 Arduino IDE Version 1.6.5



รูปที่ 3.6 Arduino IDE Version 1.6.5

เป็นซอฟแวร์สำหรับเขียนโปรแกรมเพื่อควบคุมการทำงานของบอร์ด Arduino เพื่อใช้รับและส่งข้อมูลไปยังอุปกรณ์อื่นที่ต่อเข้ามกับบอร์ด ซึ่งซอฟแวร์นี้เป็น open source และมี Library สำหรับใช้ร่วมกับเซนเซอร์ต่างๆได้อย่างหลากหลายดังรูปที่ 3.6

3.1.2.2 Eclipse Mars.1 Version 4.5.1 (JAVA Language)



รูปที่ 3.7 Eclipse Mars.1 Version 4.5.1

เป็นซอฟต์แวร์สำหรับใช้พัฒนาซอฟแวร์ภาษา Java โดยทางกลุ่มใช้พัฒนา Server สำหรับรับ-ส่งข้อมูลการตั้งค่าและควบคุมการทำงานของ Arduino Nano ผ่าน Socket เครือข่ายไร้สาย และพัฒนา User Interfaces เพื่อให้ผู้ใช้งานสามารถควบคุมในส่วนต่างๆได้ดังรูปที่ 3.7

3.1.2.3 JavaFx Scene Builder 2.0



รูป 3.8 JavaFx Scene Builder 2.0

JavaFx เป็นซอฟแวร์สำหรับการออกแบบ User interfaces ในส่วนของซอฟต์แวร์ Server ทำให้ผู้ใช้งานสามารถควบคุม WIPLED ได้จาก Server อย่างสะดวกดังรูปที่ 3.8

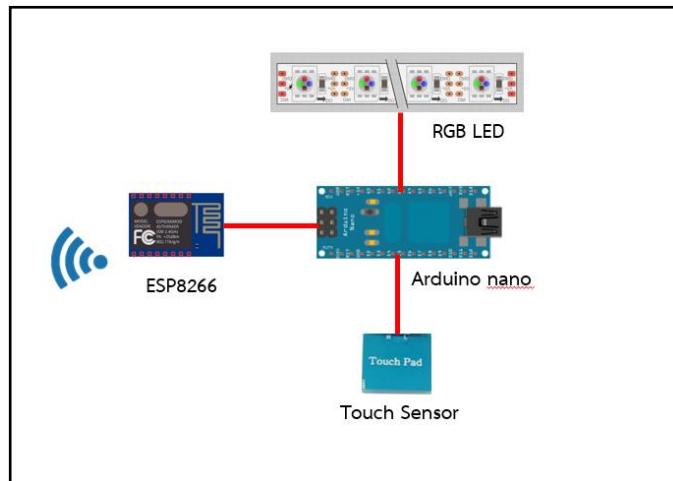
3.1.2.4 Fritzing Version 0.9.2



รูปที่ 3.9 Fritzing Version 0.9.2

Fritzing ซอฟแวร์สำหรับออกแบบลายวงจร สามารถทำ Simulation ในโปรแกรมได้ดังรูปที่ 3.9

3.2. การออกแบบส่วนฮาร์ดแวร์



รูปที่ 3.10 ภาพรวมของอุปกรณ์ที่ใช้สำหรับขั้นงาน

ฮาร์ดแวร์ใน WIPLED จะแบ่งเป็น 4 โมดูล โมดูลแรกคือพร็อกซิมิตี้เซนเซอร์ชนิดเก็บประจุ (Capacitive Touch) โดยใช้ Capacitive Touch Sensor MPR121 ใช้สำหรับรับค่าจากการสัมผัส โมดูลที่สองคือโมดูล Wi-Fi โดยใช้ ESP8266 12-E ใช้สำหรับรับ-ส่งค่าระหว่าง WIPLED กับ Server โมดูลที่สามคือ RGB LED โดยใช้ RGB LED WS2812B Strip ใช้สำหรับการแสดงผลในรูปแบบของแสง สี และโมดูลสุดท้ายคือ Arduino Nano ซึ่งจะทำหน้าที่ในการประมวลผลจากแต่ละเซนเซอร์และส่งข้อมูลไปยัง โมดูล Wi-Fi เพื่อให้ส่งต่อไปยัง Server ดังรูปที่ 3.10

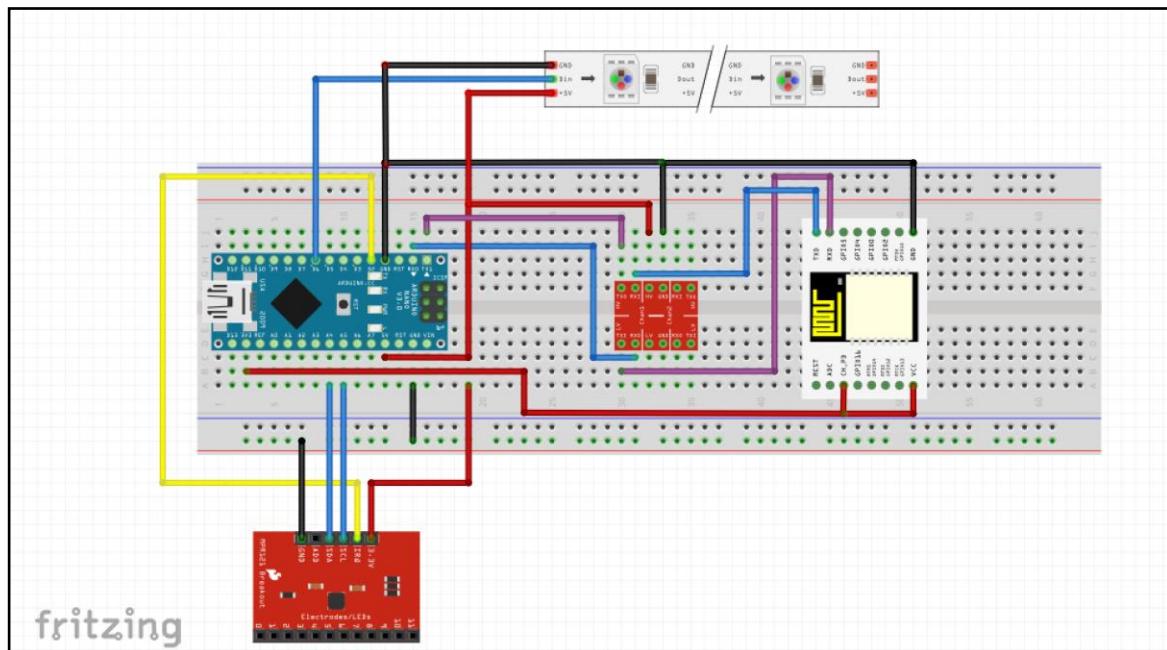
3.2.1 การคำนวณกระแสไฟฟ้าของ WIPLED

ตารางที่ 3.5 แสดงกระแสไฟฟ้าที่แต่ละอุปกรณ์ใช้งาน

อุปกรณ์	กระแสไฟฟ้าที่ใช้งาน	โวลต์ที่ใช้งาน
Arduino Nano	100 mA	5 Volt
ESP8266-12E	100 mA	3.3 Volt
RGB LED WS2812B (10 ดวง)	600 mA	5 Volt
Touch sensor MPR121	30 uA	3.3 Volt

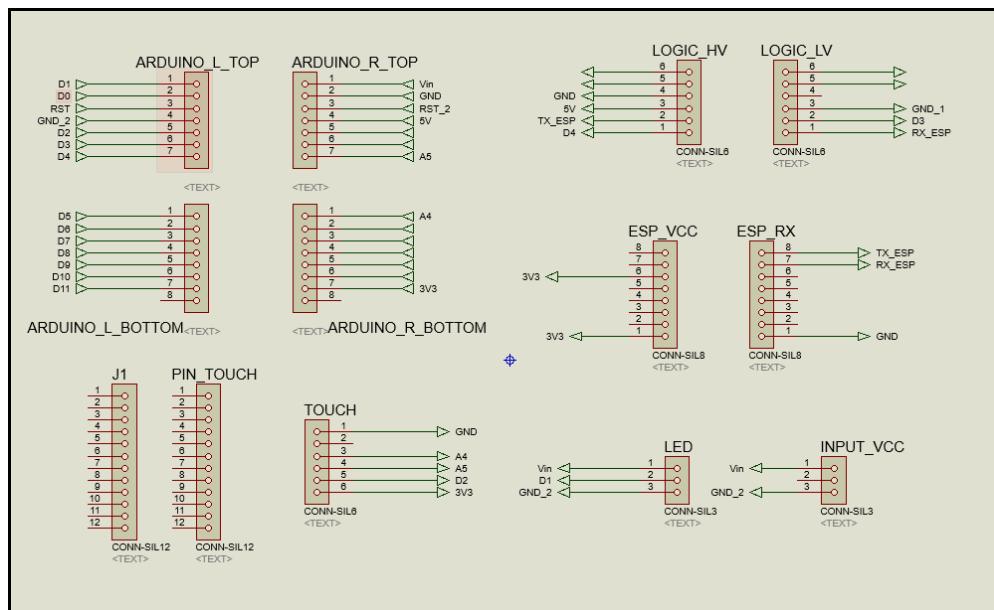
จากตารางที่ 3.5 และ Datasheet ของ Arduino Nano ซ่องไฟ 5 โวลต์ และ 3.3 โวลต์ จะสามารถจ่ายกระแสไฟฟ้าได้ช่องละ 150 มิลลิแอมป์ ซึ่งในช่องไฟฟ้า 3.3 โวลต์ สามารถจ่ายกระแสไฟฟ้าเลี้ยงอุปกรณ์ได้เพียงพอทั้ง 2 ชิ้น ส่วน ช่องไฟฟ้า 5 โวลต์ มีการจ่ายไฟฟ้าเพียง 150 มิลลิแอมป์ แต่ RGB LED WS2812B ต้องการใช้ประมาณ 600 มิลลิแอมป์ ที่ไฟ 5 โวลต์ ซึ่ง Arduino Nano ไม่สามารถจ่ายกระแสไฟฟ้าได้เพียงพอ จึงจำเป็นที่จะต้องจ่ายแหล่งไฟฟ้าจากภายนอก รวมกระแสไฟฟ้าที่ใช้ทั้งหมด 1 แอมป์ ที่ไฟ 5 โวลต์

3.2.2 วงจรการเชื่อมต่ออุปกรณ์



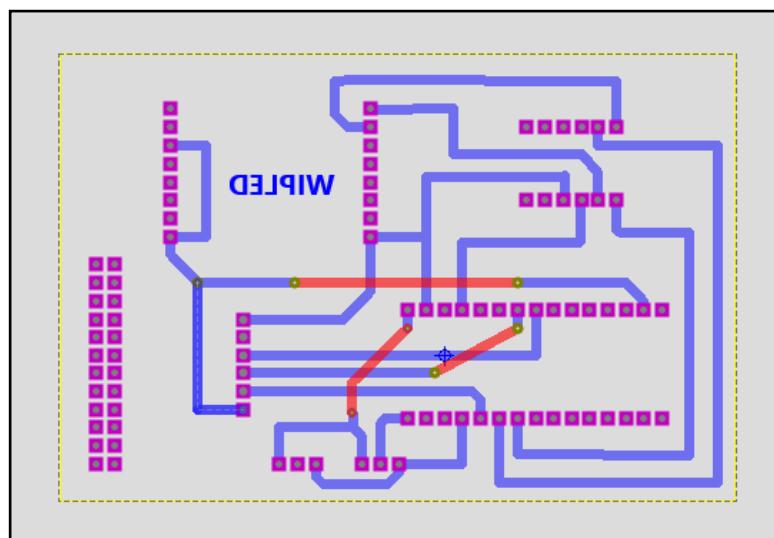
รูปที่ 3.11 ภาพวงจรเชื่อมต่อของอุปกรณ์

จากที่ได้ออกแบบวงจรอุปกรณ์ในโปรแกรม Fritzing แล้วดังรูปที่ 3.11 ต้องนำวงจรที่ได้มาออกแบบ Print Circuit Board (PCB) ดังรูปที่ 3.12 เพื่อนำไปต่อเข้าอุปกรณ์



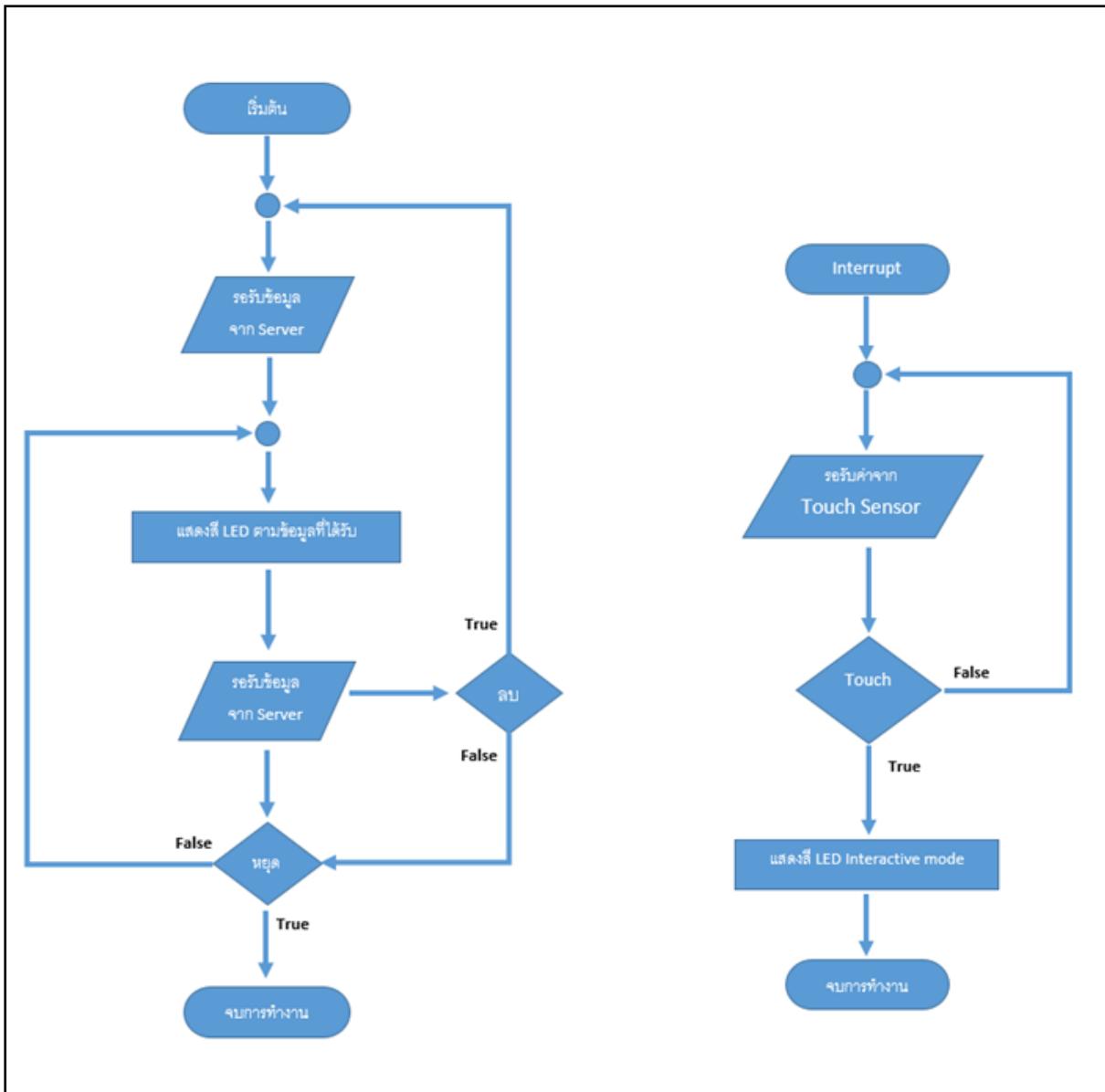
รูปที่ 3.12 ภาพวงจรเข็มต่ออุปกรณ์เพื่อทำ PCB

หลังจากนั้นต้องทำการจัดตำแหน่งของอุปกรณ์และสายไฟบน PCB ดังรูปที่ 3.13



รูปที่ 3.13 ภาพลายวงจร PCB

วงจรที่ได้จะนำไปประกอบกับอุปกรณ์เป็น WIPLED ที่สามารถรับคำสั่งจาก Server เพื่อให้แสดงสีตามความต้องการได้

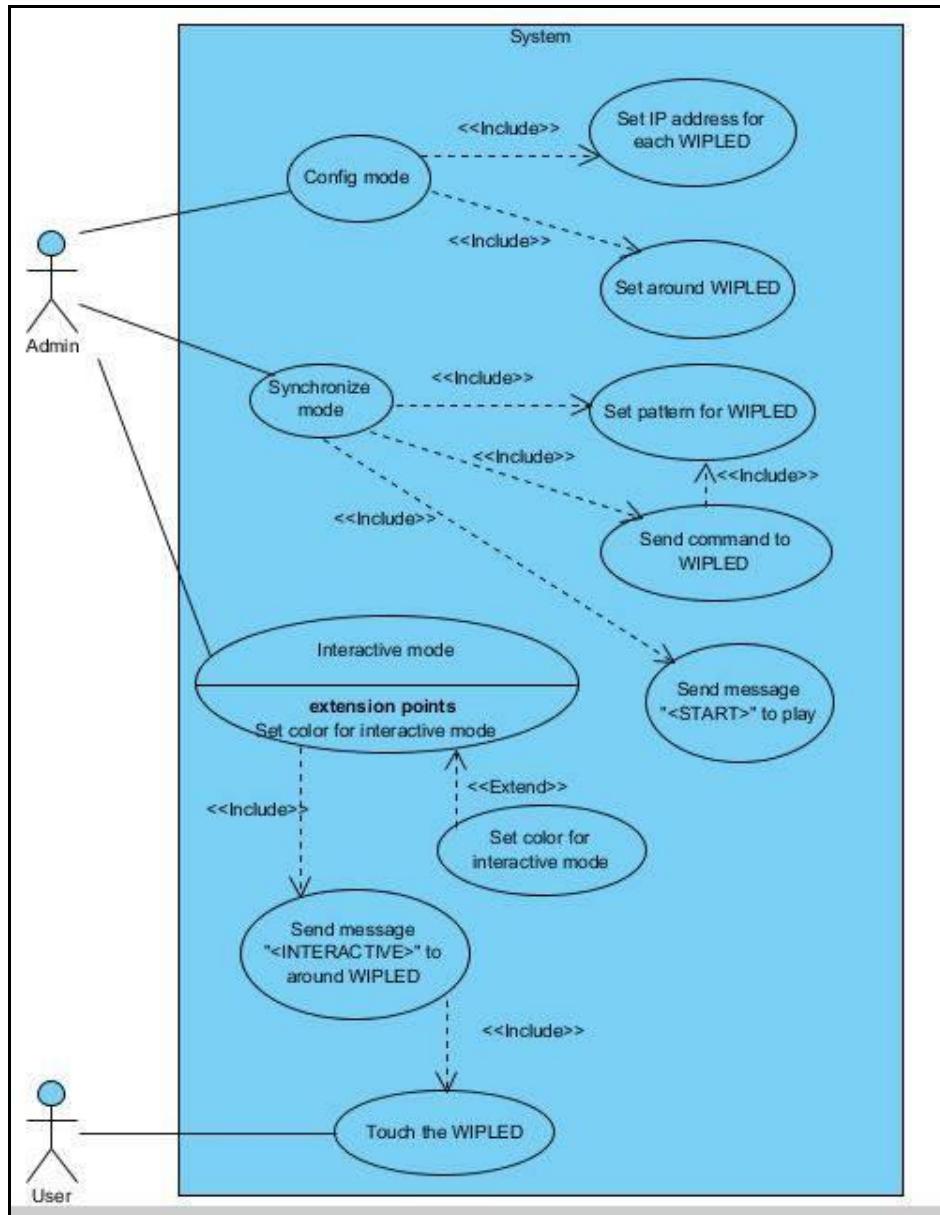


รูปที่ 3.14 การทำงานของ WIPLED

3.2.3 การทำงานของระบบฮาร์ดแวร์

เริ่มต้น WIPLED จะเชื่อมต่อกับ Server เมื่อเชื่อมต่อได้แล้ว WIPLED จะต้องรอรับข้อมูลจากเพื่อสั่งให้ LED แสดงสีตามที่ข้อมูลจาก Server ส่งมา หลังจากเล่นเสร็จ WIPLED จะรอรับข้อมูลจาก Server ให้ครบข้อมูลเพื่อที่จะส่งข้อมูลมาใหม่หรือให้หยุดเล่น ซึ่งในระหว่างการทำงานนั้นจะมีการทำงานของ Interrupt เมื่อมีการรับค่าจาก Capacitive touch โปรแกรมจะออกมายกเว้น main loop และมาเล่น Interactive mode และ WIPLED ชิ้นที่มีการสัมผัสจะส่งข้อมูลกลับไปยัง Server เพื่อให้ Server สั่งที่อยู่รอบข้างดังรูปที่ 3.14

3.3 การออกแบบส่วนซอฟแวร์



รูปที่ 3.15 Use case Diagram

จากการ Use case diagram ดังรูปที่ 3.15 แสดงถึงการทำงานโดยรวมทั้งหมดแบ่งการทำงานเป็น 3 ส่วนดังนี้

1. Use case การทำงานของโปรแกรมควบคุม

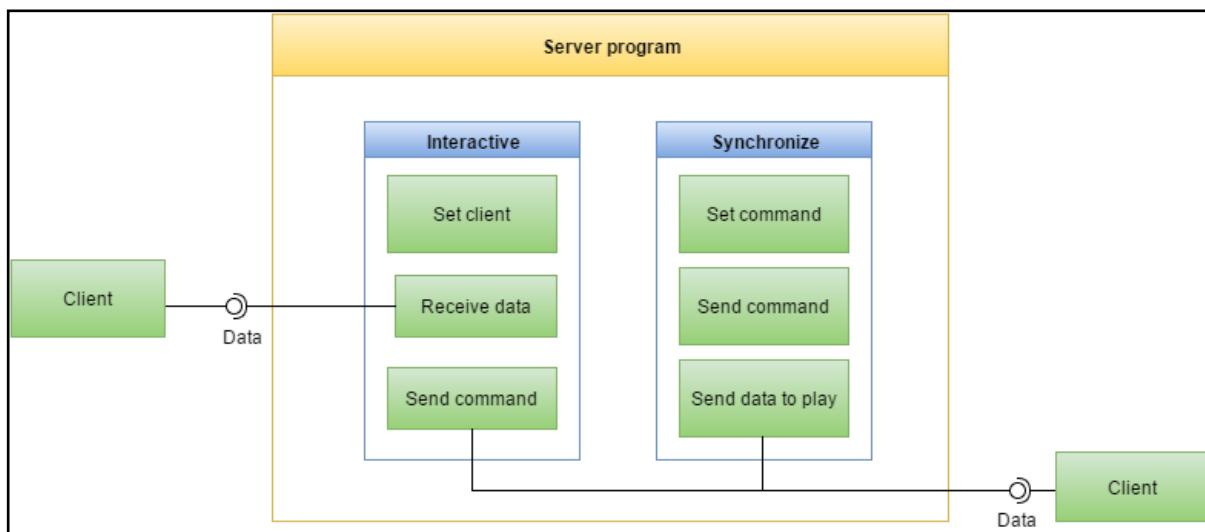
Admin สามารถกำหนด IP address ของ WIPLED แต่ละชิ้นโดยโปรแกรมควบคุม ระบุตำแหน่งที่อยู่ใกล้กัน เพื่อใช้ในการส่งค่าไปยัง WIPLED ที่อยู่รอบข้างในโหมด Interactive

2. Use case การทำงานโหมด Synchronize

Admin สามารถกำหนด Pattern ของสีที่โปรแกรมมีให้อยู่แล้ว โดยมีให้เลือกทั้งหมด 3 โหมด ซึ่งสามารถเปลี่ยนแปลงการตั้งค่าได้เมื่อต้องการ หลังจากเลือก Pattern เสร็จแล้ว สามารถส่งข้อมูลไปยัง WIPLED และส่งข้อมูลไปอีกรังเพื่อให้ WIPLED เริ่มเล่น Pattern ที่ส่งไปก่อนหน้า

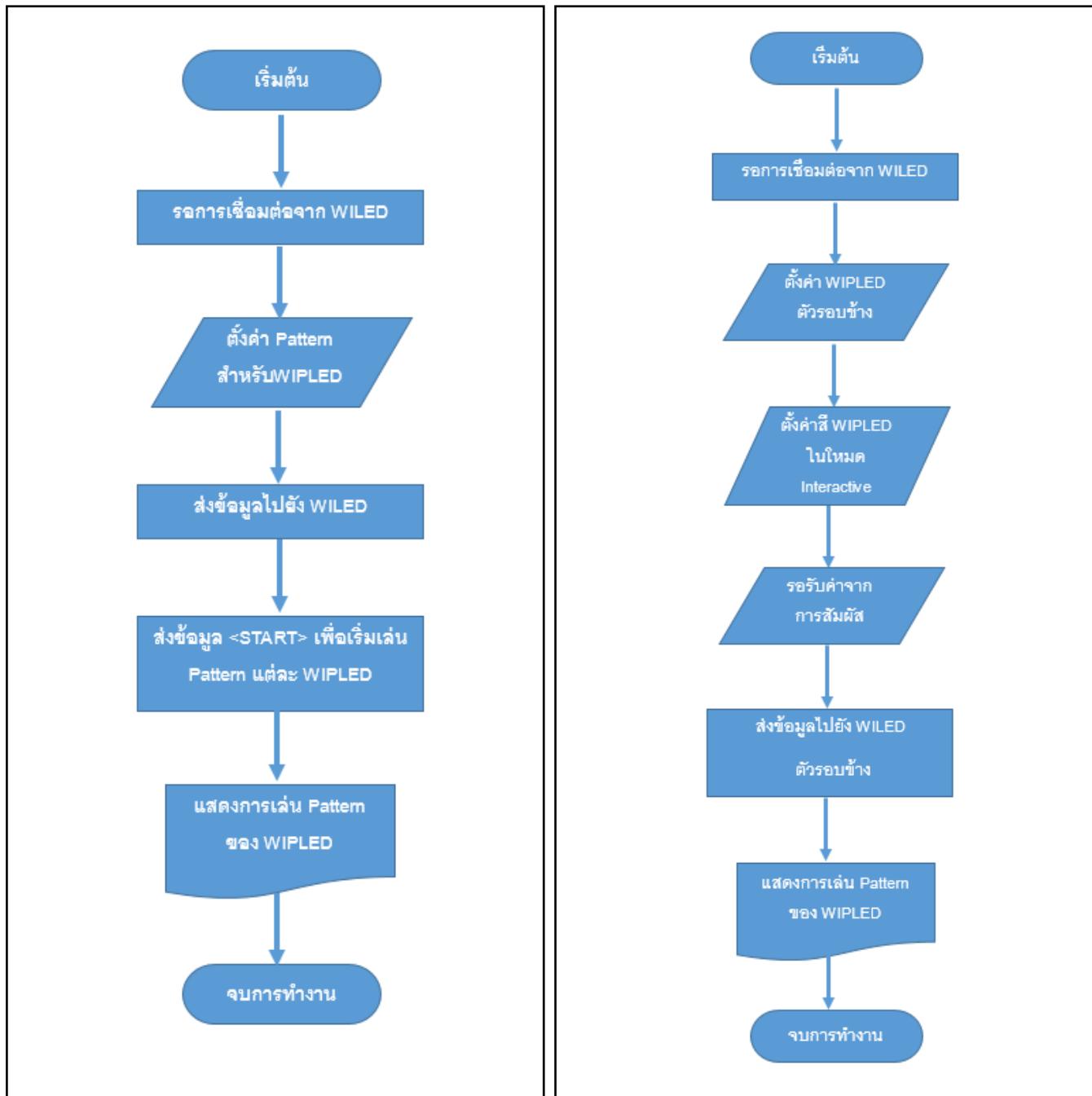
3. Use case การทำงานโหมด Interactive

สามารถเลือกสีได เมื่อมีการสัมผัส WIPLED ซึ่ง Pattern การกระจายของสี สามารถปรับเปลี่ยนได้ในภายหลัง เช่น กระจายเข้าและออก เมื่อ WIPLED ถูกสัมผัสด้วยผู้ใช้งาน WIPLED จะส่งข้อมูลไปยัง Server ของโปรแกรมควบคุม หลังจากนั้น Server ของโปรแกรมควบคุมจะส่งข้อมูลไปยัง WIPLED ขึ้นรอบข้างส่งผลให้เกิดการเปลี่ยนแปลงไปตาม Pattern ที่ได้กำหนดเอาไว้



รูปที่ 3.16 การทำงานของโปรแกรม server

โปรแกรมควบคุมใช้สำหรับการสั่งการผ่านทาง internet จะพัฒนาด้วยภาษา Java และใช้ Scene builder เพื่อแสดงหน้าจอป้อนคำสั่งส่งไปยัง Client (WIPLED) ให้สะดวกต่อการใช้งานจริง ซึ่งทางกลุ่มนี้พื้นฐานในการพัฒนาโปรแกรมด้วยภาษา Java อยู่แล้ว อีกทั้งยังเป็นภาษาที่เหมาะสมแก่การทำ Server โปรแกรมควบคุมแบ่งออกเป็น 2 ส่วนหลัก โดยส่วนแรกจะเป็นโหมด Synchronize ซึ่งจะเป็นโหมดที่สามารถสั่งการกับ WIPLED ทุกตัวให้แสดงผลในรูปแบบของสีเป็นไปตามที่ ผู้ใช้งานต้องการ ส่วนที่สองคือโหมด Interactive ซึ่งจะเป็นโหมดที่รอรับค่าจาก WIPLED เมื่อมีการสัมผัสถูกขึ้น เมื่อรับค่าที่ส่งมาจาก WIPLED แล้ว Server ของโปรแกรมควบคุมจะทำการส่งคำสั่งไปยัง WIPLED ตัวรอบข้างของตัวที่ถูกสัมผัสถูกสัมผัส ให้แสดงผลตามที่ได้ตั้งค่าเอาไว้ดังรูปที่ 3.17 และ 3.18



รูปที่ 3.17 การทำงานของโหมด Synchronize

รูปที่ 3.18 การทำงานของโหมด Interactive

3.4 การออกแบบข้อมูลสำหรับโปรแกรมควบคุมส่งไปยัง WIPLED

รูปแบบการส่งข้อมูลของโหมด Synchronize จะแบ่งเป็น 3 ส่วนดังนี้

- Mode 0 เป็นการส่งข้อมูลไปยัง WIPLED เพื่อสั่งให้ LED ทุกดวงของ WIPLED ดับ

รูปแบบการส่งข้อมูล : <M0:TIME>

- TIME คือ เวลาที่ต้องการให้ LED ของ WIPLED ตัวนั้นดับ หน่วยเป็นมิลลิวินาที

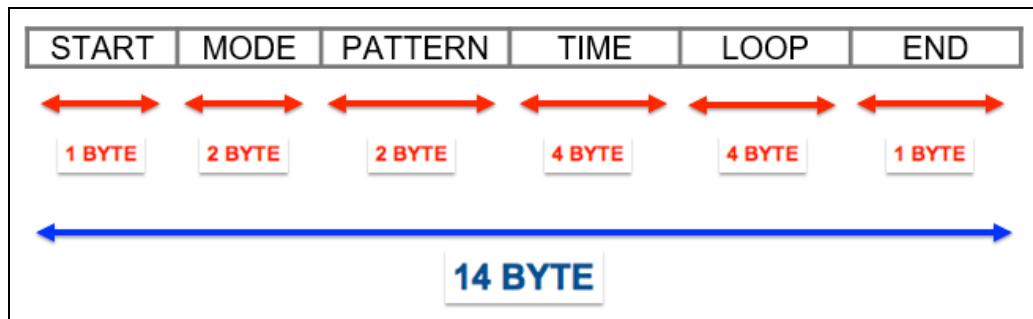
- Mode 1 เป็นการส่งข้อมูลไปยัง WIPLED โดยให้ WIPLED เล่น Pattern สีที่บอร์ดคอนโทรลเลอร์มีอยู่แล้วรูปแบบการส่งข้อมูล : <M1:PATTERN:TIME:LOOP> ข้อมูลที่ใช้ทั้งหมด 14 byte ดังรูปที่ 3.19

- PATTERN คือ รูปแบบการเล่นสีโดยมี 3 Pattern คือ P0 (Rainbow Pattern) , P1

(rainbowCycle Pattern) และ P2 (theaterChaseRainbow Pattern)

- TIME คือ เวลาที่ต้องการให้สีของ LED ใน WIPLED ตัวนั้นเปลี่ยนແຕลสีใน Pattern หน่วยเป็นมิลลิวินาที

- LOOP คือ จำนวนรอบในการเล่น Pattern



รูปที่ 3.19 รูปแบบการส่งข้อมูลในส่วนของ Mode 1

- Mode 2 เป็นการส่งข้อมูลไปยัง WIPLED โดยสามารถเลือกสี การ Fade ของสี ได้ตามความต้องการ รูปแบบการส่งข้อมูล : <M2:COLOR:FADE:TIME:LOOP> ข้อมูลที่ใช้ทั้งหมด 19 byte ดังรูปที่ 3.20

- COLOR คือ สีที่ต้องการให้ WIPLED ตัวนั้นแสดงเป็นตัวเลขฐาน 16 0xFFFFFFF

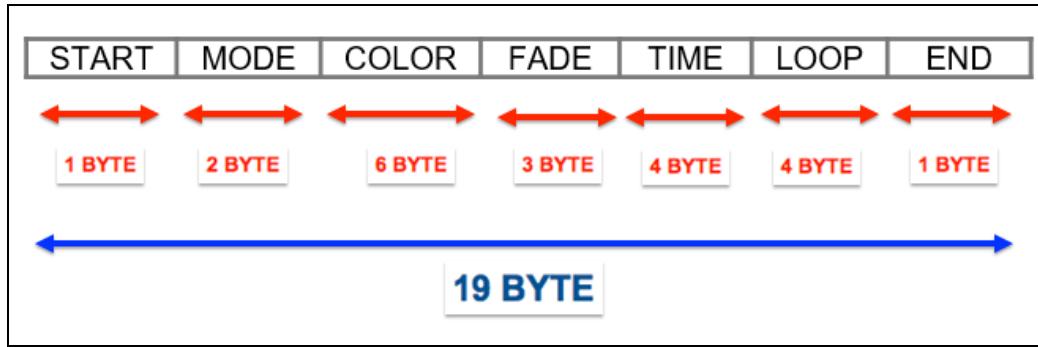
- FADE คือ การกำหนดการหรีแสง มี 2 แบบ คือ

1. IN คือการหรีแสงให้สว่างมากขึ้น

2. OUT คือการหรีแสงให้มืดลง

- TIME คือ เวลาที่ต้องการให้สีของ LED ใน WIPLED ตัวนั้นเปลี่ยนความสว่าง หน่วยเป็นมิลลิวินาที

- LOOP คือ จำนวนรอบในการเล่น



รูปที่ 3.20 รูปแบบการส่งข้อมูลในส่วนของ Mode 2

เมื่อทำการส่งข้อมูลจาก Server ของโปรแกรมควบคุมรูปแบบการส่งข้อมูลจะเป็นดังนี้

<SEND><จำนวน Pattern ที่ต้องการส่ง><ข้อมูลของแต่ละ Pattern>!

- <SEND> คือ เป็นข้อมูลที่บอกกับ WIPLED ให้รู้ว่ามีข้อมูลจากโหมด Synchronize เข้ามา

- “!” คือ สัญลักษณ์ที่บอกจุดสิ้นสุดการส่งข้อมูลจาก Server ของโปรแกรมควบคุม

ตัวอย่างเช่น

<SEND><2><M1:P1:20:5><M2:00FF1A:IN:50:5>!

นอกจากการส่งข้อมูลสำหรับการเล่นสีแล้วยังมีคำสั่งอื่นที่ใช้สำหรับ WIPLED ดังนี้

- <DELETE> คือ ลบข้อมูลการเล่นสีทั้งหมดใน WIPLED

- <START> คือ คำสั่งให้ WIPLED ที่ได้รับข้อมูลการเล่นสีแล้วเริ่มแสดงสี

รูปแบบการส่งข้อมูลของโหมด Interactive มีดังนี้

1. การตั้งค่าสีในโหมด Interactive ให้กับ WIPLED ทุกชิ้น

รูปแบบการส่งข้อมูล : <SET><COLOR>

- <SET> คือ ข้อมูลเริ่มต้นให้ WIPLED ของการตั้งค่าสีในโหมด Interactive

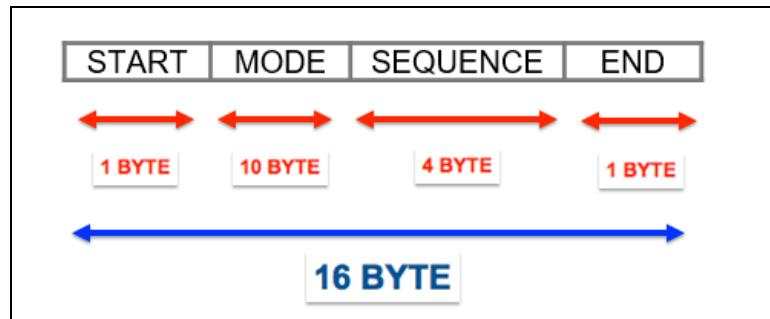
- COLOR คือ สีที่ต้องการให้ WIPLED ตัวนั้นแสดงเป็นตัวเลขฐาน 16 0xFFFFFFF

2. ข้อมูลที่ใช้ในการส่งให้ WIPLED ที่อยู่รอบข้างของ WIPLED ชิ้นที่ถูกสัมผัส ข้อมูลที่ใช้ทั้งหมด 16 byte ดังรูปที่ 3.21

รูปแบบการส่งข้อมูล : <INTERACTIVE><ORDER>

- <INTERACTIVE> คือ ข้อมูลเริ่มต้นให้ WIPLED ชิ้นที่อยู่รอบข้างชิ้นที่ถูกสัมผัส

- <ORDER> คือ ลำดับชิ้นของ WIPLED ที่ถูกสัมผัส



รูปที่ 3.21 รูปแบบการส่งข้อมูลในโหมด Interactive

จากการออกแบบ硬件ได้นำอุปกรณ์ต่างๆที่ได้เลือกไว้มาประกอบเข้าด้วยกันโดยออกแบบลายวงจรและจะทดสอบการทำงานของอุปกรณ์แต่ละชิ้นและระบบรวมทั้งหมดในบทถัดไป ในส่วนของการออกแบบซอฟต์แวร์ได้ออกแบบ Use case และลำดับขั้นตอนการทำงานในแต่ละโหมดเพื่อใช้ในการเขียนโปรแกรมควบคุมในบทถัดไป และส่วนสุดท้าย การออกแบบข้อมูลในการรับ-ส่งระหว่าง Server และ WIPLED ได้ออกแบบข้อมูลสำหรับใช้ในการตั้งค่าการทำงานของ WIPLED และข้อมูลในการแสดงสีแต่ละโหมดซึ่งในบทต่อไปจะนำเสนอที่ออกแบบทั้งหมดรวมเป็นระบบเดียวกันเพื่อให้สามารถควบคุม WIPLED ผ่านเครือข่ายไร้สายได้

บทที่ 4

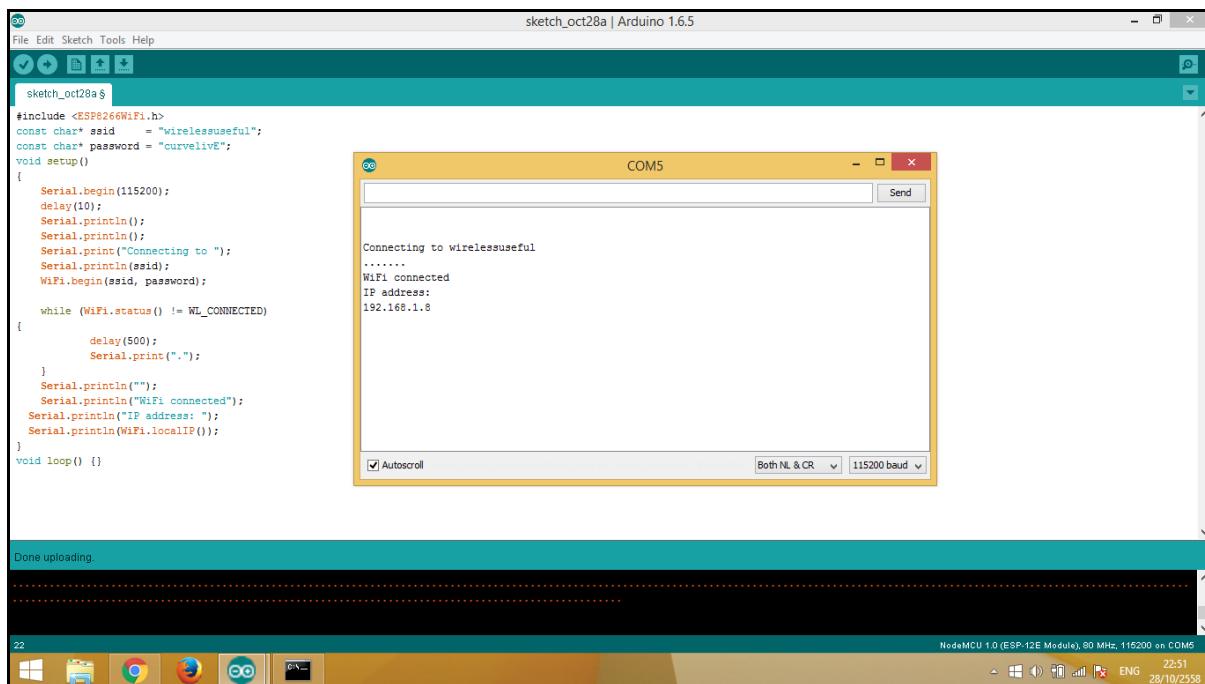
ผลการวิจัยและอภิปรายผล

การศึกษาและพัฒนา WIPLED และโปรแกรมควบคุม วางแผนการทดลองออกแบบเป็นสี่ส่วน ส่วนแรกคือ การทดลองการรับ-ส่งข้อมูลผ่านเครือข่ายไร้สายส่วนของกล่องคอนโทรล (WIPLED) ส่วนที่สองคือการเชื่อมต่อ ของอุปกรณ์อิเล็กทรอนิกส์หรือเซนเซอร์ต่างๆที่จะมาเป็นส่วนรับอินพุตและส่วนแสดงผลของ WIPLED ส่วนที่ สามคือส่วนของการทดสอบ Server ของโปรแกรมควบคุม การเชื่อมและการรับ-ส่งข้อมูลกับ Client ส่วนที่สี่คือ การทดสอบโปรแกรมสำหรับควบคุม WIPLED ให้แสดงผลตามคำสั่งที่โปรแกรมได้ส่ง และสุดท้ายภาพ WIPLED ที่เสร็จสมบูรณ์

4.1 การทดลองการรับ-ส่งข้อมูลผ่านเครือข่ายไร้สาย

4.1.1 การทดสอบการรับ DHCP ของ NodeMCU เมื่อเชื่อมต่อกับ Wireless Network

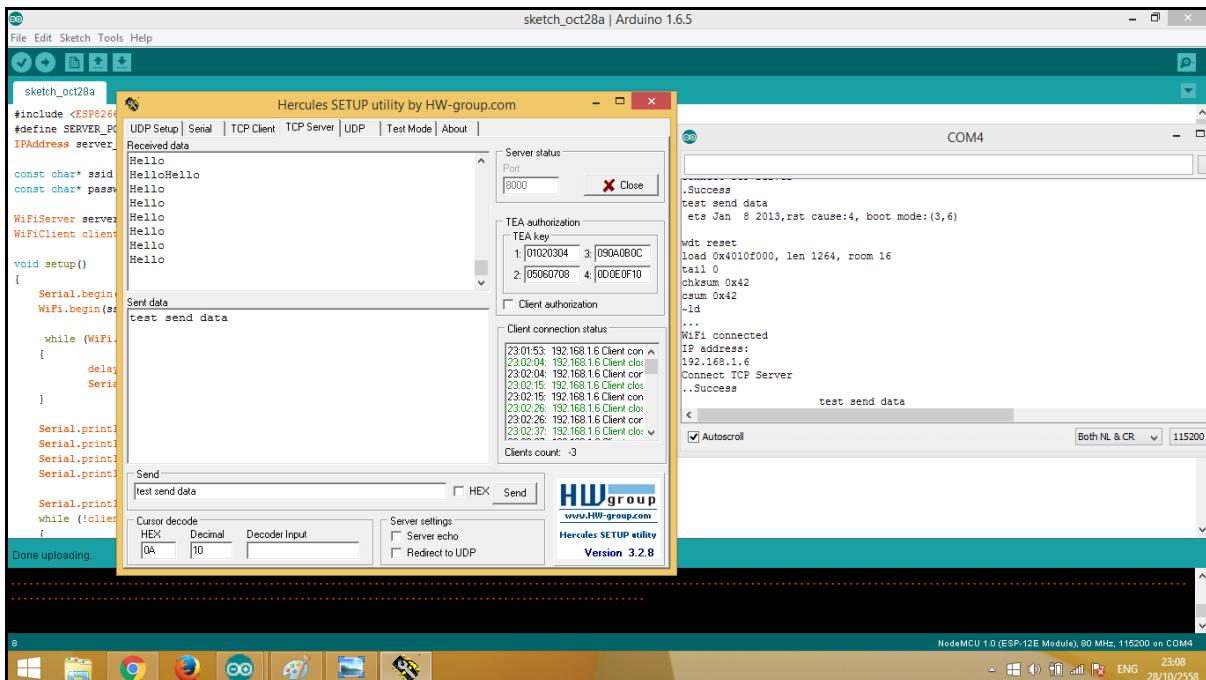
เมื่อทำการ Upload โปรแกรมไปยังตัวของ NodeMCU โดยจะให้เชื่อมต่อกับเร้าเตอร์ เพื่อ ทดสอบการรับ DHCP และทำการแสดง IP Address ที่ได้รับมาพบว่าตัว Router นั้นมีการแจกว า IP Address มา เป็น 192.168.1.8 ดังรูปที่ 4.1



รูปที่ 4.1 แสดงการรับ DHCP ของ NodeMCU

4.1.2 การทดสอบการจำลอง NodeMCU เป็น Client

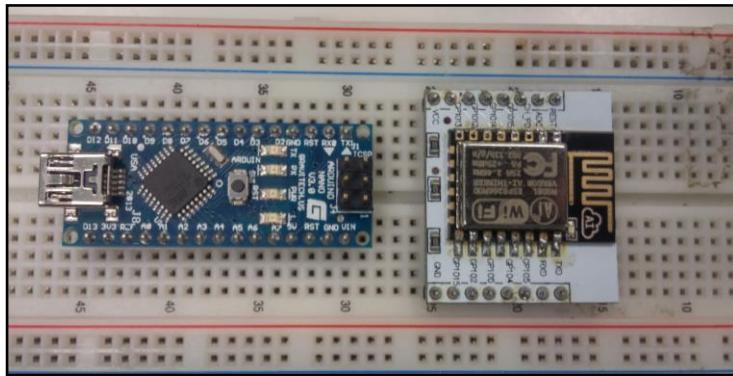
ทำการทดสอบโดยให้เครื่องคอมพิวเตอร์นั้นเป็น server และตัว NodeMCU เป็น client ซึ่งจะใช้โปรแกรม Hercules เป็นโปรแกรมทดสอบโดยจะกำหนดพอร์ตของตัว Server และทำการตั้งค่าพอร์ตของ Client ให้ตรงกัน จากนั้นจึงทำการเชื่อมต่อเมื่อมีการเชื่อมต่อสำเร็จ Server จะได้รับข้อความ “Hello” เพื่อเป็นการยืนยันการเชื่อมต่อระหว่าง Client กับ Server ดังรูปที่ 4.2



รูปที่ 4.2 แสดงข้อความเมื่อ NodeMCU ทำการเชื่อมตอกับ Server สำเร็จ

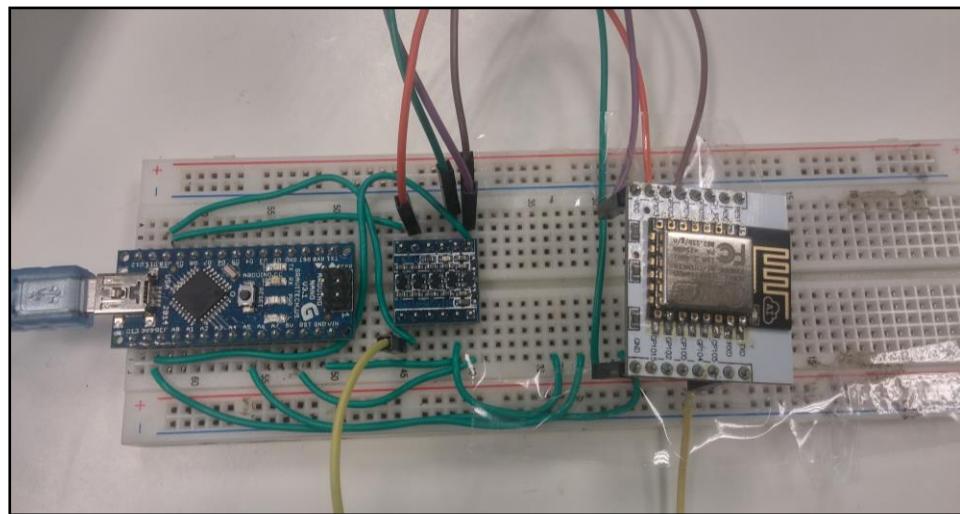
4.1.3 การทดสอบการรับส่งข้อมูลผ่าน Wi-Fi โดยใช้ Arduino Nano กับ ESP8266-12E

เนื่องจาก Arduino Nano มี Logic level 5 โวลต์ แต่ ESP8266-12E ใช้ Logic level 3.3 โวลต์ ในการทำงาน ทำให้การเชื่อมต่อระหว่าง Arduino Nano กับโมดูล Wi-Fi ESP8266 ไม่สามารถเชื่อมตอกันได้โดยตรง ส่งผลให้ โมดูล Wi-Fi ESP8266 ไม่ทำงาน ภาพวงจรแสดงดังรูปที่ 4.3



รูปที่ 4.3 ภาพแสดง Arduino กับ module Wi-Fi

จึงต้องทำการศึกษาเพิ่มเติมโดยใช้ Logic Level Converter Bi-Directional [6] ในการแปลง Logic level 5 โวลต์ เป็น 3.3 โวลต์ ซึ่งการส่งข้อมูลจาก Arduino Nano ไปยังโมดูล WiFi ESP8266 ต้องแปลง Logic level จาก 5 โวลต์ เป็น 3.3 โวลต์ ในทางกลับกัน การส่งข้อมูลจากโมดูล WiFi ESP8266 ไปยัง Arduino Nano จะต้องแปลง logic จาก 3.3 โวลต์ เป็น 5 โวลต์ ดังรูปที่ 4.4

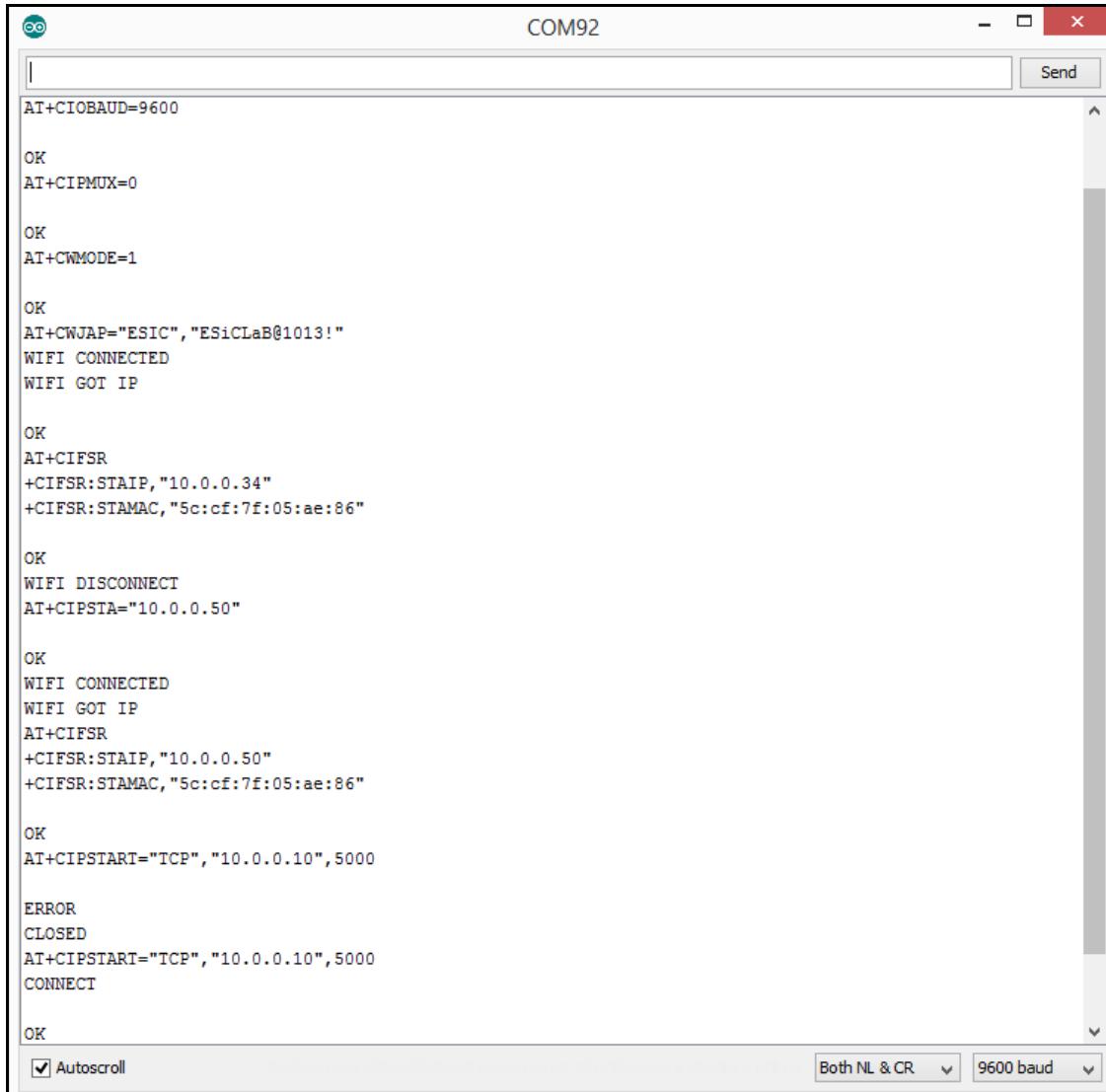


รูปที่ 4.4 รูปวงจรการเชื่อมต่อ Arduino Nano , Logic level Converter และ โมดูล WiFi ESP8266

หลังจากที่ใช้ Logic Level Converter Bi-Directional ทำให้ Arduino Nano สามารถรับ-ส่งข้อมูลกับ 모ดูล WiFi ESP8266 ได้

4.1.4 ทดสอบการใช้ AT Command ในตั้งค่า Wi-Fi

ทำการทดสอบการตั้งค่า เมดูล Wi-Fi ESP8266 โดยใช้ AT Command ผลทดสอบปรากฏว่า ทุกคำสั่งสามารถใช้งานได้ ซึ่งเมื่อส่งแต่ละคำสั่งไปที่เมดูล Wi-Fi ESP8266 และสามารถตั้งค่าได้สำเร็จ เมดูล Wi-Fi ESP8266 จะส่งข้อความตอบกลับ “OK” ดังรูปที่ 4.5



```
AT+CIOBAUD=9600
OK
AT+CIPMUX=0
OK
AT+CWMODE=1
OK
AT+CWJAP="ESIC","ESiCLaB@1013!"
WIFI CONNECTED
WIFI GOT IP

OK
AT+CIFSR
+CIFSR:STAIP,"10.0.0.34"
+CIFSR:STAMAC,"5c:cf:7f:05:ae:86"

OK
WIFI DISCONNECT
AT+CIPSTA="10.0.0.50"

OK
WIFI CONNECTED
WIFI GOT IP
AT+CIFSR
+CIFSR:STAIP,"10.0.0.50"
+CIFSR:STAMAC,"5c:cf:7f:05:ae:86"

OK
AT+CIPSTART="TCP","10.0.0.10",5000
ERROR
CLOSED
AT+CIPSTART="TCP","10.0.0.10",5000
CONNECT

OK
```

The terminal window shows the following sequence of AT commands and their responses:

- AT+CIOBAUD=9600
- OK
- AT+CIPMUX=0
- OK
- AT+CWMODE=1
- OK
- AT+CWJAP="ESIC","ESiCLaB@1013!"
- WIFI CONNECTED
- WIFI GOT IP
- OK
- AT+CIFSR
- +CIFSR:STAIP,"10.0.0.34"
- +CIFSR:STAMAC,"5c:cf:7f:05:ae:86"
- OK
- WIFI DISCONNECT
- AT+CIPSTA="10.0.0.50"
- OK
- WIFI CONNECTED
- WIFI GOT IP
- AT+CIFSR
- +CIFSR:STAIP,"10.0.0.50"
- +CIFSR:STAMAC,"5c:cf:7f:05:ae:86"
- OK
- AT+CIPSTART="TCP","10.0.0.10",5000
- ERROR
- CLOSED
- AT+CIPSTART="TCP","10.0.0.10",5000
- CONNECT
- OK

At the bottom of the terminal window, there is an "Autoscroll" checkbox checked and a baud rate selection dropdown set to "9600 baud".

รูปที่ 4.5 การทดสอบตั้งค่าเมดูล Wi-Fi ESP8266 ผ่าน AT Command

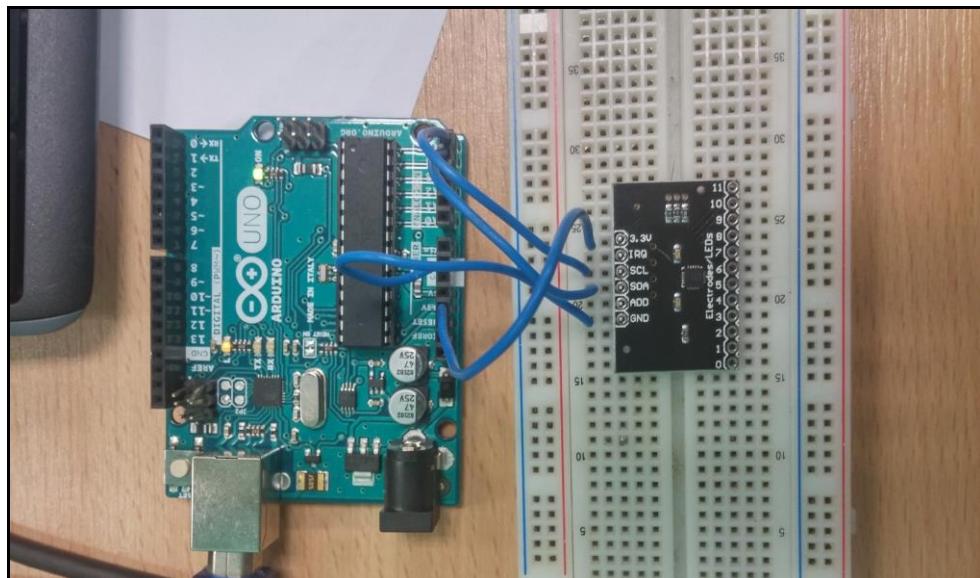
AT Command [8] ที่ใช้ในการตั้งค่ามีความหมายดังนี้

1. AT+CIOBAUD=9600 : ตั้งค่า baudrate ของโมดูล Wi-Fi ESP8266
2. AT+CIPMUX=0 : ตั้งค่าการเชื่อมต่อ สำหรับการเชื่อมต่อจุดแบบจุดเดียว (Single Connection)
3. AT+CWJAP="ESIC","ESiCLaB@1013!" : เชื่อมต่อ Access Point ชื่อ ESIC รหัส: ESiCLaB@1013!"
4. AT+CIFSR : ตรวจสอบ IP Address ของโมดูล Wi-Fi ESP8266
5. AT+CIPSTA="10.0.0.50" : ตั้งค่า IP Address ของโมดูล Wi-Fi ESP8266
6. AT+CIPSTART="TCP","10.0.0.10",5000 : เชื่อมต่อกับ Server IP 10.0.0.10 Port 5000 โดยใช้ TCP Connection

4.2 การทดสอบการเชื่อมต่อของอุปกรณ์อิเล็กทรอนิกส์หรือเซนเซอร์ต่างๆ

4.2.1 การทดสอบใช้ Capacitive Touch Sensor MPR121

ทำการทดสอบโดยใช้ Library ของ Adafruit ซึ่ง MPR121 นี้สามารถรับอินพุตได้ถึง 12 ช่อง จากรูป ด้านล่างเลข 0-11 จากการทดสอบดังรูปที่ 4.7 สามารถใช้งานได้ แต่ต้องพัฒนาเพิ่มเติมโดยการเพิ่มการตั้งค่าเพื่อทำ Near Proximity ให้เซนเซอร์สามารถอ่านค่าได้เพียงแค่นำมือไปใกล้ๆ (ยังไม่สมผัส) ภาพแสดงว่างรูปที่ 4.6



รูปที่ 4.6 ตัวอย่างการต่อวงจรเพื่อทำการทดสอบ Capacitive Touch Sensor MPR121

The screenshot shows the Arduino IDE interface. The left window displays the code for the MPR121test sketch, which includes setup and loop functions for initializing the MPR121 sensor and printing touch status to the serial port. The right window shows the Serial Monitor with the following output:

```

11 released
Adafruit MPR121 Capacitive Touch sensor test
MPR121 found!
3 touched
3 released
4 touched
4 released
8 touched
9 touched
10 touched
11 touched
8 released
9 released
10 released
11 released

```

Autoscroll

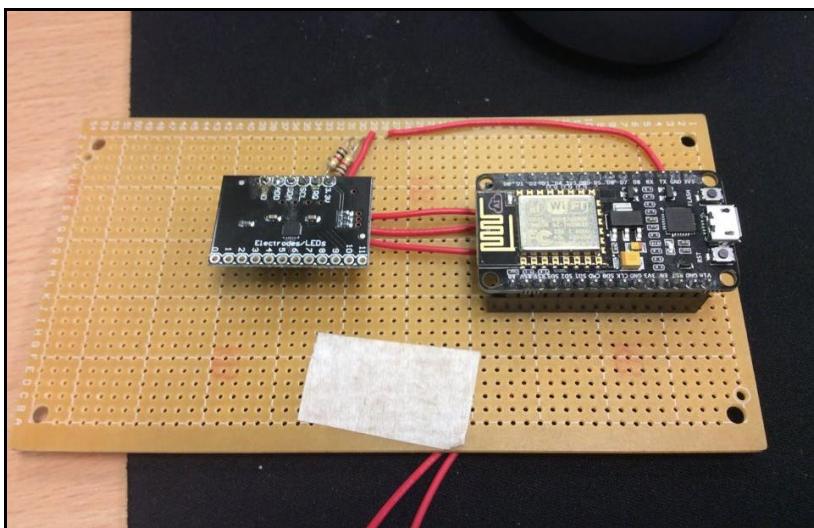
Done uploading.

รูปที่ 4.7 แสดงผลการทำงานของ Capacitive Touch Sensor MPR121

4.2.2 การทดสอบใช้ Capacitive Touch Sensor MPR121 ร่วมกับ NodeMCU

ทำการทดสอบเพิ่มเติมจากที่ได้เคยทดสอบ MPR121 กับ Arduino โดยเปลี่ยน

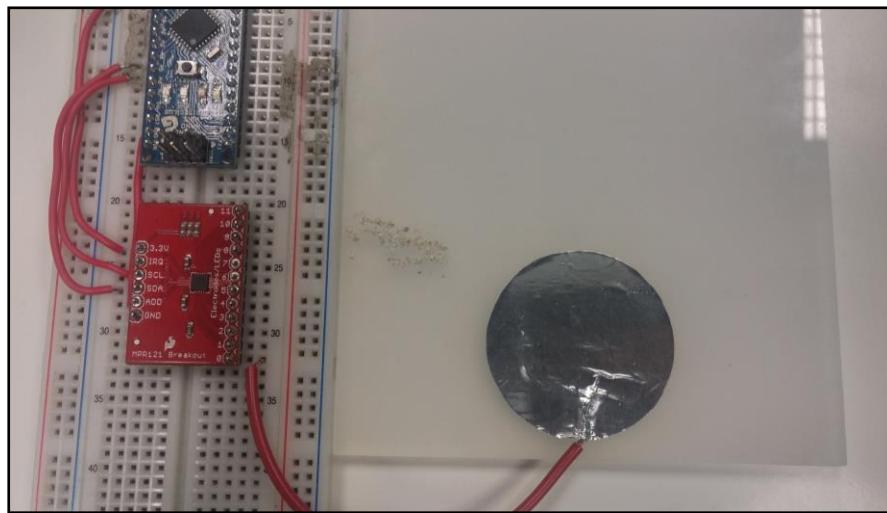
ไมโครคอนโทรลเลอร์ เป็น NodeMCU เพื่อทดสอบการทำงานร่วมกันได้ปกติหรือไม่ ซึ่งผลการทดสอบปรากฏว่า NodeMCU ทำงานร่วมกับ Capacitive Touch Sensor MPR121 ได้ไม่ดีนัก บางครั้ง NodeMCU ไม่สามารถติดต่อกับ I2C Address ของเซนเซอร์ได้หรือบางครั้งเมื่อเซนเซอร์รับอินพุตเป็นเวลาประมาณ 10 นาทีขึ้นไปจะทำให้ NodeMCU มีอาการค้าง ไม่สามารถทำงานต่อไปได้ จึงเลือกที่จะทดสอบกับไมโครคอนโทรลเลอร์รุ่นอื่นซึ่งได้เลือกทดสอบ Arduino Nano ร่วมกับ Capacitive Touch Sensor MPR121 ดังรูปที่ 4.8



รูปที่ 4.8 ภาพตัวอย่างการใช้ Capacitive Touch Sensor MPR121 ร่วมกับ NodeMCU

4.2.3 การทดสอบใช้ Capacitive Touch Sensor MPR121 เพื่อทำเป็น Capacitance Near Proximity Detection

เดิมจากที่เคยได้ทดสอบ Capacitive Touch Sensor MPR121 ไม่สามารถทำ Near Proximity Detection ได้ จึงทำการศึกษาข้อมูลจาก DataSheet เพิ่มขึ้น พบว่าสามารถทำได้โดยตั้งค่าที่ Baseline Value Register , Touch / Release Threshold เพื่อเพิ่มให้ MPR121 เพิ่มความไวต่อการสัมผัส ทดสอบโดยต่อสายออกจาก Sensor และเชื่อมเข้ากับแผ่นฟอยล์ซึ่งนำไปไฟฟ้าเพื่อให้มีหน้าสัมผスマากขึ้น หลังจากการทดสอบ MPR121 สามารถรับการสัมผัสได้โดยไม่ต้องนำมือสัมผัสกับ Sensor โดยตรงดังรูปที่ 4.9 และจะเห็นได้ว่ารูปที่ 4.10 มืออยู่ไม่สัมผัสกับแผ่นฟอยล์ชนเชอร์กีสามารถตรวจจับได้



รูปที่ 4.9 การเชื่อมต่อแผ่นฟอยล์กับเซนเซอร์



รูปที่ 4.10 ภาพทดสอบการสัมผัสแบบ Capacitance Near Proximity Detection

4.2.4 การทดสอบใช้ RGB LED Strip WS2812B

ทำการทดสอบโดยใช้ Library Adafruit NeoPixel ซึ่งตัวอย่างนี้เป็นการทดสอบการควบคุมสีของ LED แต่ละดวง รวมถึงการควบคุมเป็นรูปแบบเช่น ไฟวิ่ง ไฟสีรุ้ง ดังรูปที่ 4.11

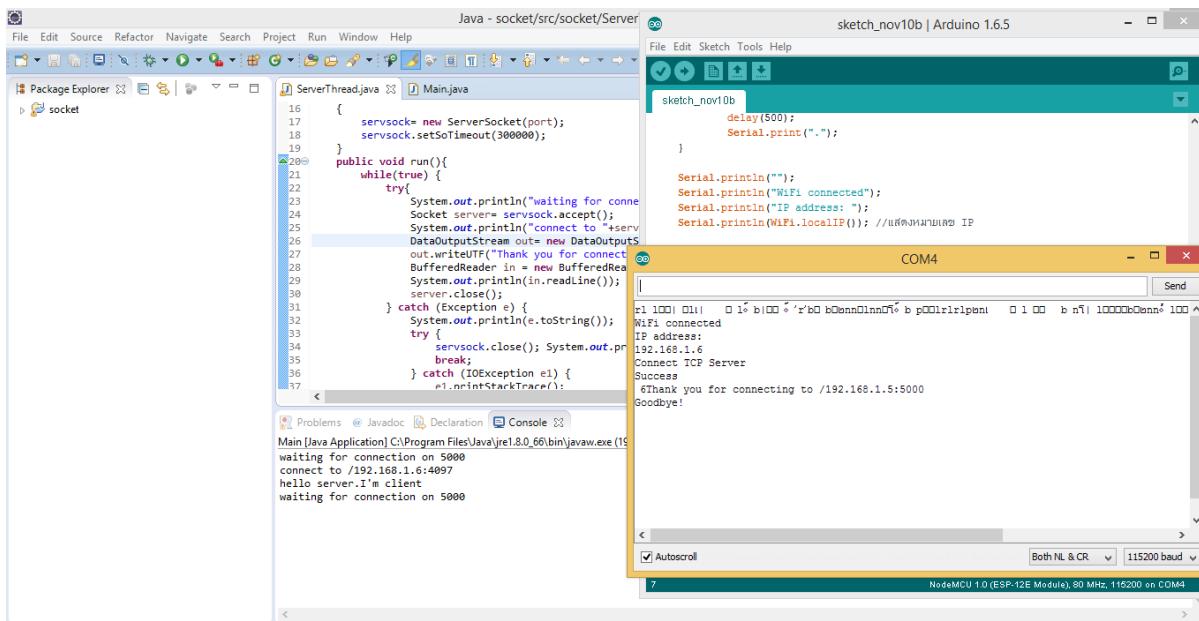


รูปที่ 4.11 แสดงผลของ RGB LED Strip WS2812B

4.3 การทดสอบ Server ของโปรแกรมควบคุม

4.3.1 การทดสอบการใช้ Socket Server/Client

ทำการทดสอบโดยให้ NodeMCU เป็น Socket Client และใช้ภาษา Java บนเครื่องคอมพิวเตอร์เป็น Server โดย Server จะทำการเปิดพอร์ตเพื่อรับการเชื่อมต่อจาก Client และทดสอบรับ-ส่งข้อมูลระหว่าง Client กับ Server ดังรูปที่ 4.12



รูปที่ 4.12 ทดสอบการรับ-ส่งข้อมูลระหว่าง Client และ Server

4.3.2 การทดสอบการเชื่อมต่อ Client หลายตัวพร้อมกันบน Server

ทำการทดสอบโดยให้เครื่อง Server เป็นตัวรับ Client ทุกตัว โดยการแบ่ง Thread การทำงานของแต่ละตัวเมื่อ Client เชื่อมต่อเข้ามาแล้วให้แยก Thread ออกไปทำงานของแต่ละตัว ซึ่งเมื่อได้ทำการทดสอบแล้วพบว่า Client นั้นสามารถเชื่อมต่อเข้ามายัง Server ได้พร้อมๆ กันโดยไม่ต้องรอให้ตัวใดตัวหนึ่งทำงาน เสร็จก่อน ดังรูปที่ 4.13

The screenshot shows the Eclipse IDE interface with the following details:

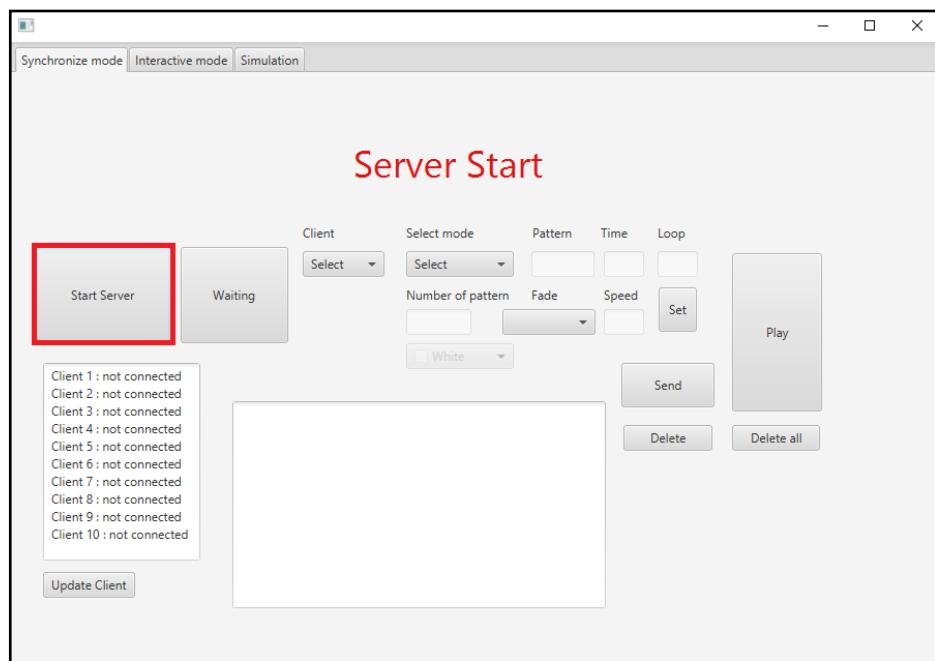
- Left Panel (Code Editor):** Displays the `ServerThread.java` file. The code implements a `Runnable` interface for a server thread. It creates a `ServerSocket` on port 5000, accepts connections, and handles them in a loop. The `run()` method contains a try-catch block for handling exceptions.
- Right Panel (Console):** Shows the output of the application named `ServerThread`. It prints "Welcome" when a connection is established. It also shows two separate connection logs, each indicating a connection from a client at IP 192.168.1.6 to the server at port 5000. The logs are timestamped at 23:47:39.

รูปที่ 4.13 แสดงการเชื่อมต่อของ Client หลายตัวมายัง Server พร้อมกัน

4.4 การทดสอบโปรแกรมควบคุมร่วมกับ WIPLED

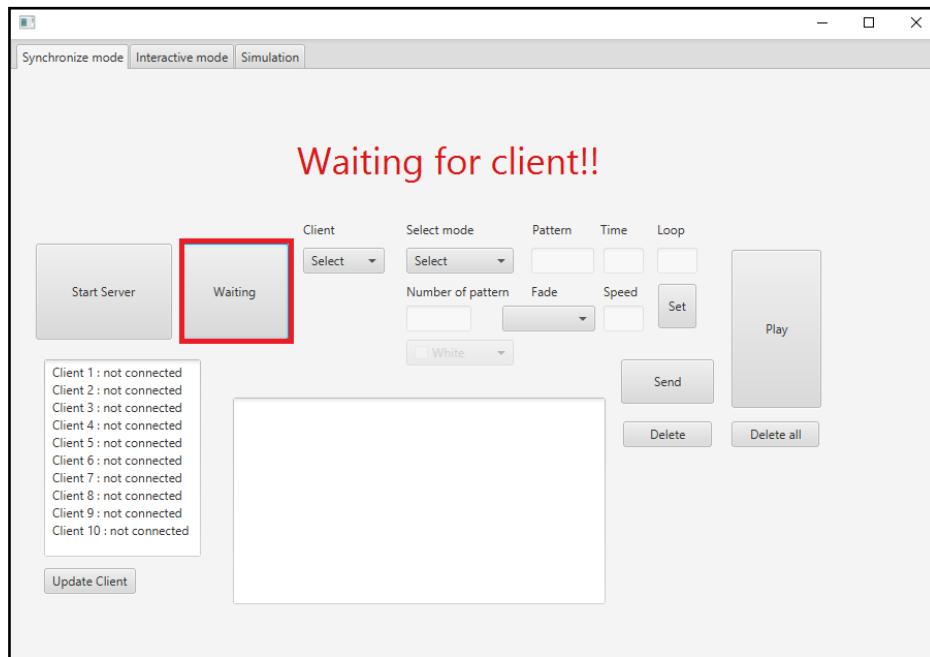
การทดสอบการทำงานของโปรแกรมควบคุมจะแบ่งเป็นสองส่วน ส่วนแรกคือการทดสอบโปรแกรมในโหมด Synchronize ซึ่งเป็นโหมดที่ให้ผู้ใช้งานสามารถสั่งให้ WIPLED แสดงสีในรูปแบบตามที่ต้องการได้ ส่วนที่สองคือโหมด Interactive ซึ่งเป็นโหมดที่เมื่อผู้ใช้งานสัมผัสกับเซนเซอร์จะออกจากโหมด Synchronize และมาเล่นรูปแบบที่กำหนดไว้

4.4.1 การทดสอบการทำงานของโปรแกรมควบคุม WIPLED ในโหมด Synchronize



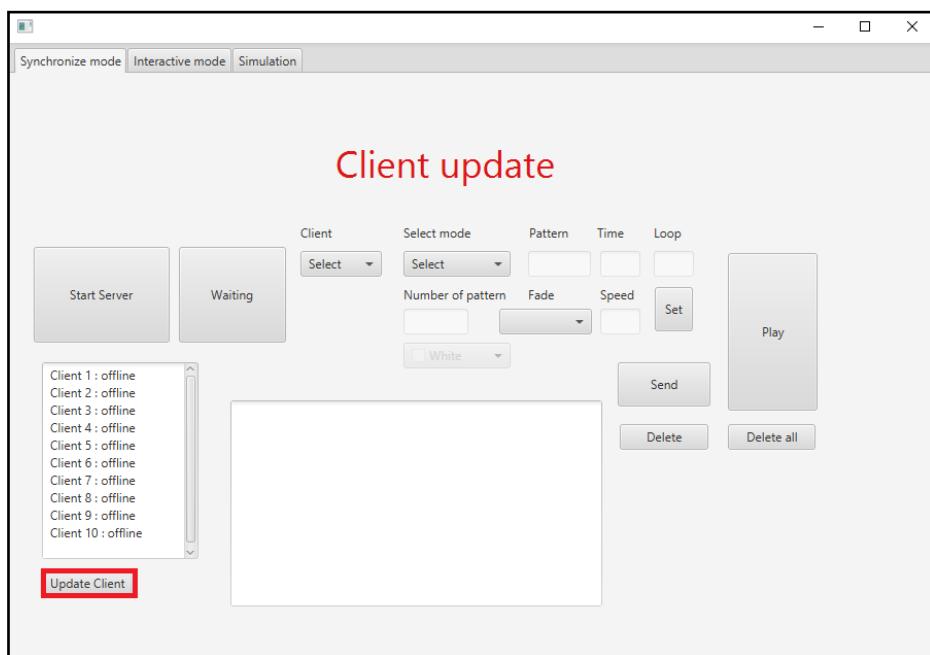
รูปที่ 4.14 หน้าจอ User interface เมื่อทำการกดปุ่ม “Start Server”

เมื่อทำการกดปุ่ม “Start Server” จะเป็นการสั่งให้โปรแกรมควบคุมทำการเปิดพอร์ตให้ WIPLED สามารถเชื่อมต่อเข้ามายัง Server ของโปรแกรมได้ดังรูปที่ 4.14



รูปที่ 4.15 หน้าจอ User interface เมื่อทำการกดปุ่ม “Waiting”

เมื่อทำการกดปุ่ม “Waiting” จะเป็นการสั่งให้โปรแกรมควบคุมให้รอรับ WIPLED ตัวอื่นเข้ามายัง Server ของโปรแกรมได้ดังรูปที่ 4.15



รูปที่ 4.16 หน้าจอ User interface เมื่อทำการกดปุ่ม “Update Client”

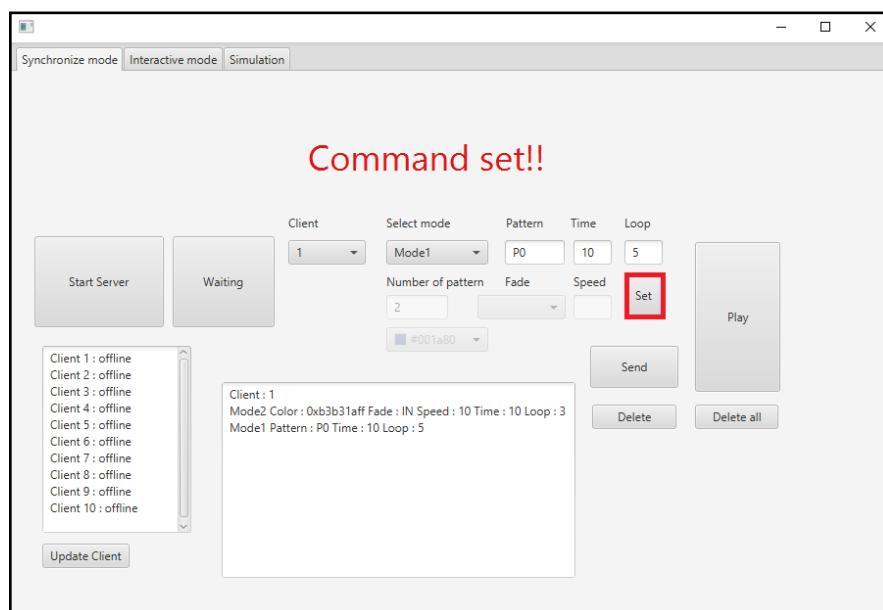
เมื่อทำการกดปุ่ม “Update” จะเป็นการ Update สถานะของ WIPLED แต่ละตัว Online คือ WIPLED สามารถเชื่อมต่อเข้ากับ Server ของโปรแกรมควบคุมได้ หรือ Offline คือ WIPLED ยังไม่มีการเชื่อมต่อเข้ามาใน Server ของโปรแกรมควบคุมดังรูปที่ 4.16

```
AT+CIPSTA="10.0.0.52"
OK
AT+CWJAP="ESIC","ESiCLaB@1013!"
WIFI CONNECTED
WIFI GOT IP
AT+CIPCLOSE
busy p...
AT+CIPSTART="TCP","10.0.0.10",5000
busy p...

OK
MPR121 found!
AT+CIPSTART="TCP","10.0.0.10",5000
CONNECT
```

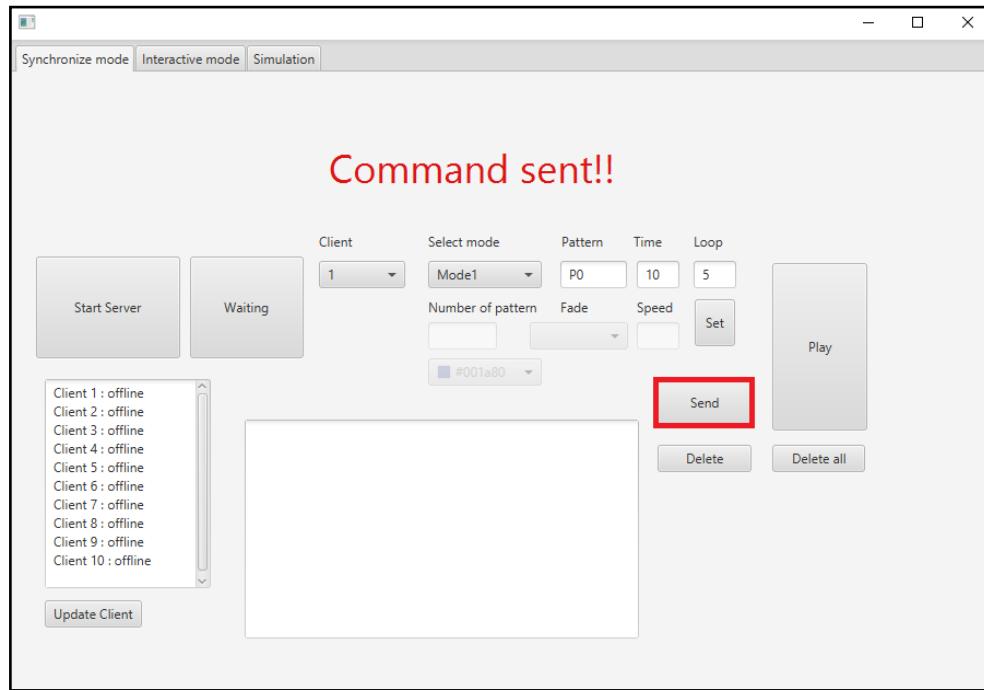
รูปที่ 4.17 แสดง Monitor ของ WIPLED เมื่อเชื่อมต่อ Server ของโปรแกรมได้สำเร็จ

จากภาพด้านบน WIPLED ทำการเชื่อมต่อเข้ากับ Server ของโปรแกรมควบคุมจะเห็นว่าส่วนของโปรแกรมควบคุม Client1 แสดง IP Address ของ WIPLED ตัวที่เชื่อมต่อเข้ามาซึ่งคือ IP Address ของภาพด้านขวาดังรูปที่ 4.17



รูปที่ 4.18 หน้าจอ User interface เมื่อทำการกดปุ่ม “Set”

เมื่อทำการกดปุ่ม “Set” จะเป็นการ Set คำสั่งที่มีการเลือกไว้ว่าเป็นของ Client ไหน Mode รูปแบบ การเล่นของ WIPLED แต่ละตัว สามารถเพิ่มจำนวน Pattern ได้โดยการเลือกข้อมูลในรูปแบบที่ต้องการอีกรึเปล่า กดปุ่ม “Set” จะเป็นการเพิ่มเป็น Pattern ถัดไปโดย WIPLED จะแสดง Pattern นี้ก็ต่อเมื่อเล่น Pattern แรกเสร็จแล้วดังรูปที่ 4.18

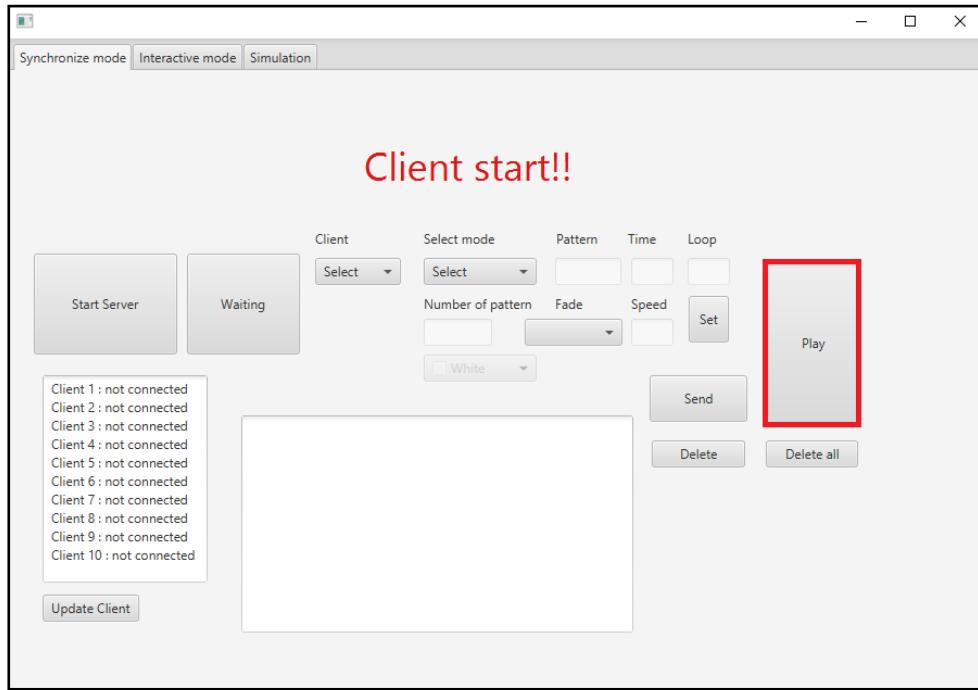


รูปที่ 4.19 หน้าจอ User interface เมื่อทำการกดปุ่ม “Send”

เมื่อทำการกดปุ่ม “Send” จะเป็นการส่งข้อมูลไปยัง WIPLED ที่เลือกไว้ ซึ่งข้อมูลที่ส่งไปยัง WIPLED ดังรูปที่ 4.19 จะมีรูปแบบข้อมูลเป็น <SEND><จำนวน Pattern><ข้อมูลแต่ละ Pattern>! ตัวอย่างเช่น <SEND><2><M1:P0:20:4><M2:FF00FF:IN:20:5>!

```
+IPD,31:<SEND><1><M2:fffff4d:IN:10:10:5&
+IPD,2:>!DATA Correct
Number of Pattern :1
MODE2
```

รูปที่ 4.20 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูล Pattern สืบจากโปรแกรมควบคุม WIPLED ได้รับข้อมูลจากโปรแกรมควบคุมถูกต้องตามที่โปรแกรมส่งดังรูปที่ 4.20



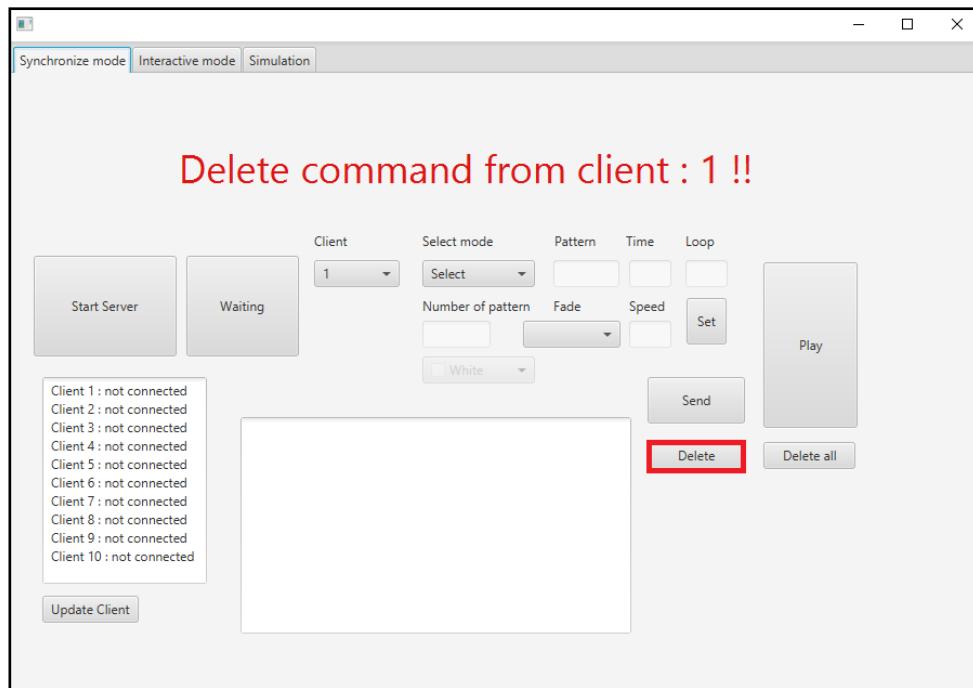
รูปที่ 4.21 หน้าจอ User interface เมื่อทำการกดปุ่ม “Play”

เมื่อทำการกดปุ่ม “Play” จะเป็นการส่งข้อมูลไปยัง WIPLED ทุกตัวให้แสดงเล่น Pattern ที่ได้ทำการส่งข้อมูลไปก่อนหน้านี้ดังรูปที่ 4.21

```
+IPD,8:<START>!PLAY_LED
```

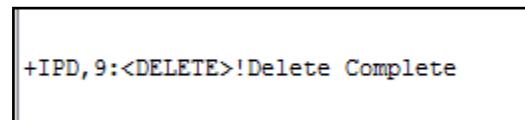
รูปที่ 4.22 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูลให้เริ่มเล่น Pattern

เมื่อมีการกดปุ่ม “Play” โปรแกรมควบคุมจะส่งข้อมูล <START> ไปยัง WIPLED ให้เริ่มเล่น Pattern ดังรูปที่ 4.22



รูปที่ 4.23 หน้าจอ User interface เมื่อทำการกดปุ่ม “Delete”

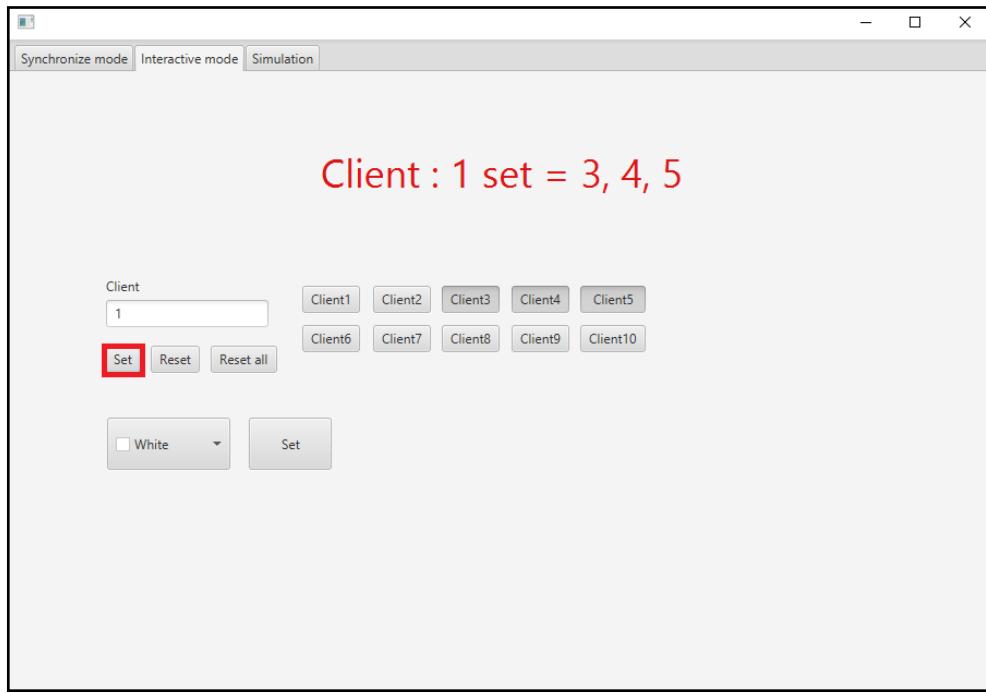
เมื่อทำการกดปุ่ม “Delete” จะเป็นการลบข้อมูลของ Pattern ที่ได้ส่งไปยัง WIPLED ของตัวที่ถูกเลือก และ ปุ่ม “Delete all” จะเป็นการลบข้อมูลของ WIPLED ทุกตัวดังรูปที่ 4.23



รูปที่ 4.24 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูลให้ลบข้อมูล Pattern ทั้งหมดใน WIPLED

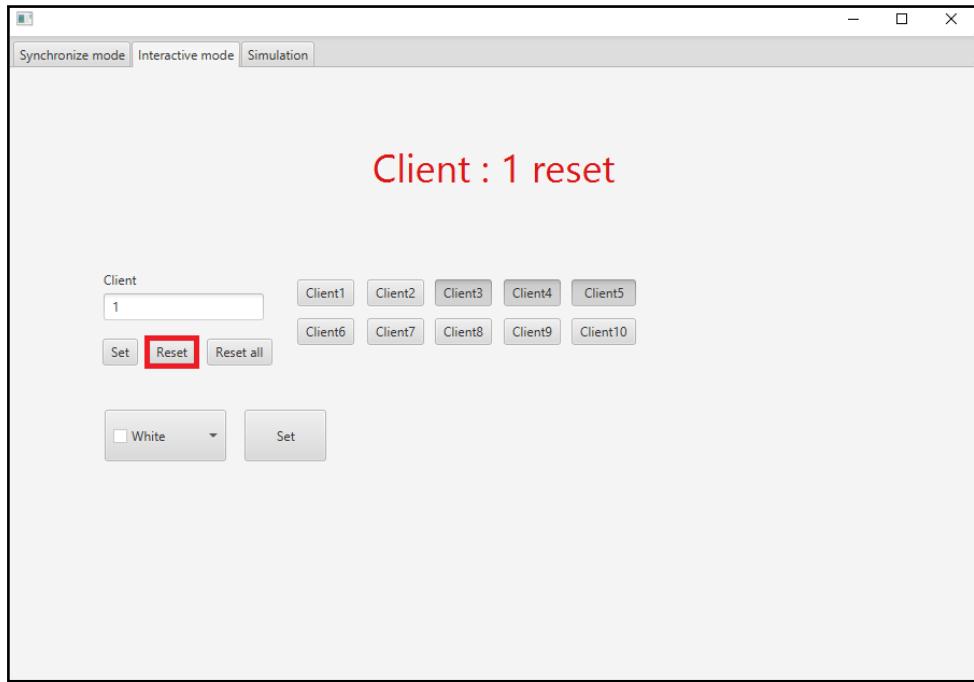
โปรแกรมควบคุมจะส่งข้อมูล <DELETE> มา�ัง WIPLED เพื่อลบข้อมูลทั้งหมดดังรูปที่ 4.24

4.4.2 การทดสอบการทำงานของโปรแกรมควบคุม WIPLED ในโหมด Interactive



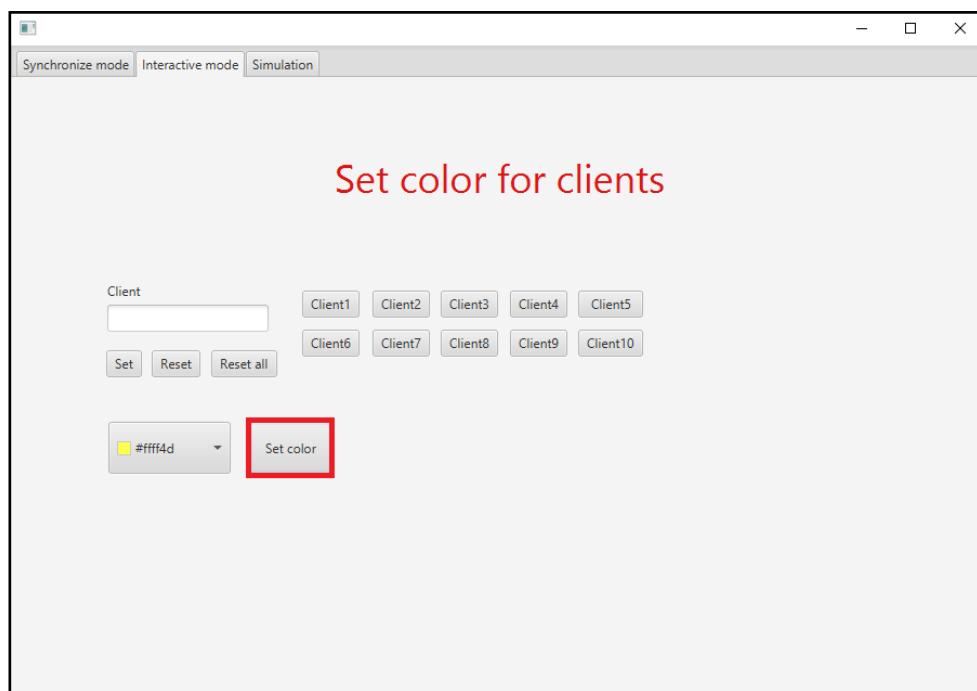
รูปที่ 4.25 หน้าจอ User interface เมื่อทำการกดปุ่ม “Set”

เมื่อทำการกดปุ่ม “Set” จะเป็นการตั้งค่า WIPLED ตัวที่อยู่โดยรอบของ WIPLED ตัวที่เลือกไว้ในช่อง Client ดังรูปที่ 4.25 ซึ่งการตั้งค่านี้จะมีผลก็ต่อเมื่อ WIPLED ถูกสัมผัส หลังจากนั้น จะเล่น Pattern การกระจายไฟยัง WIPLED ตัวรอบข้างของตัวที่ถูกสัมผัส



รูปที่ 4.26 หน้าจอ User interface เมื่อทำการกดปุ่ม “Reset”

เมื่อทำการกดปุ่ม “Reset” จะเป็นการลบข้อมูลการตั้งค่าตัวที่อยู่โดยรอบของ WIPLED ที่เลือกไว้ และปุ่ม “Reset all” จะลบข้อมูลการตั้งค่าทั้งหมดดังรูปที่ 4.26



รูปที่ 4.27 หน้าจอ User interface เมื่อทำการกดปุ่ม “Set color”

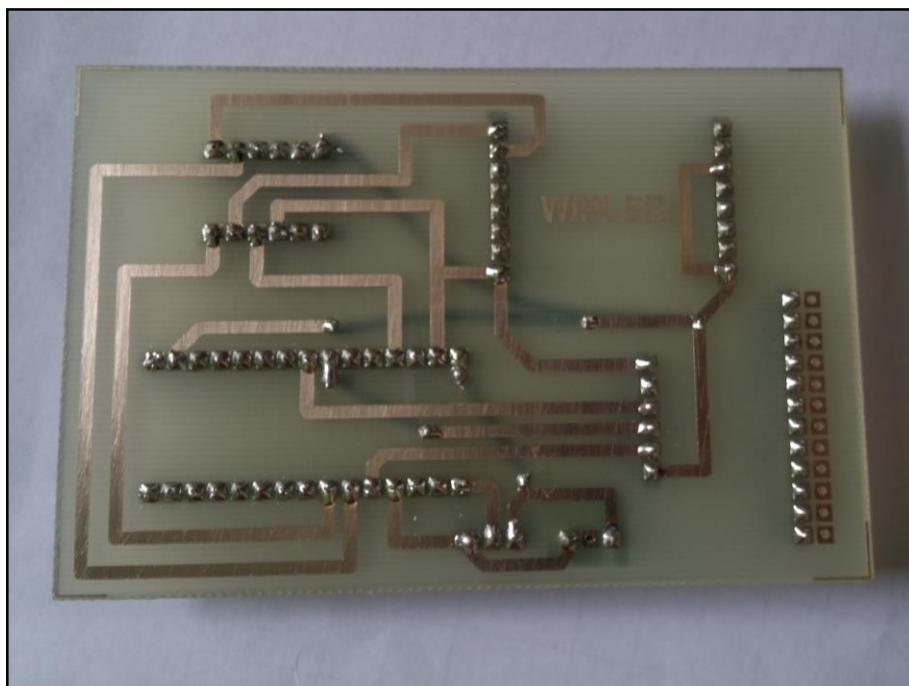
เมื่อทำการกดปุ่ม “Set color” ดังรูปที่ 4.27 จะเป็นการตั้งค่าสีของ WIPLED ทุกตัว เมื่อถูกสัมผัส ซึ่งในโหมด Interactive ถ้าไม่มีการตั้งค่าสี ค่าเริ่มต้นจะเป็นสีขาว

+IPD, 14:<SET><001a80>!SET COMPLETE

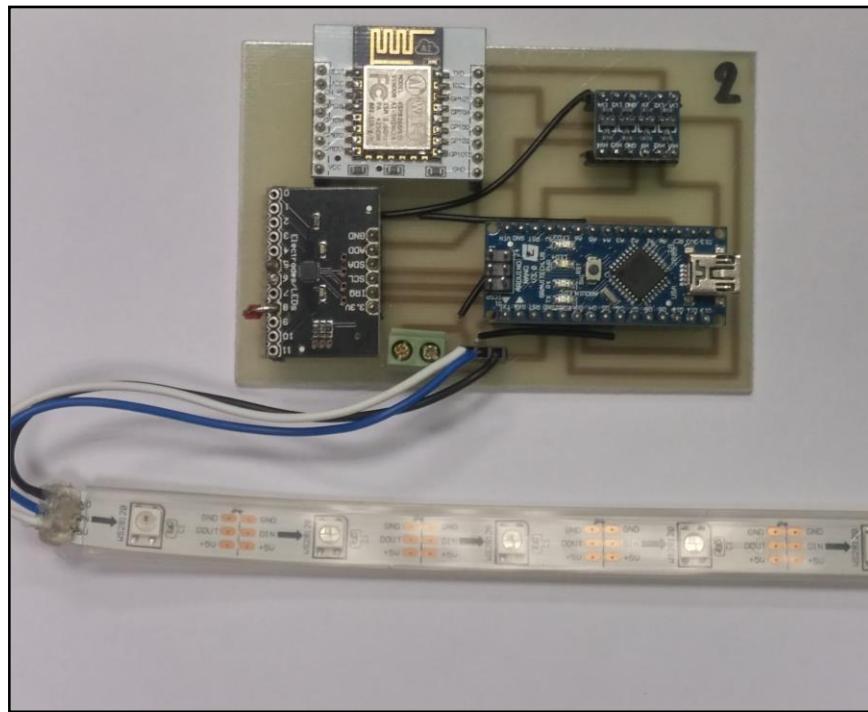
รูปที่ 4.28 แสดงผล Monitor ของ WIPLED เมื่อได้รับข้อมูลการตั้งค่าสีในโหมด Interactive

โปรแกรมควบคุมจะส่งข้อมูล <SET><COLOR> นายัง WIPLED เพื่อตั้งค่าในโหมด Interactive ดังรูปที่ 4.28

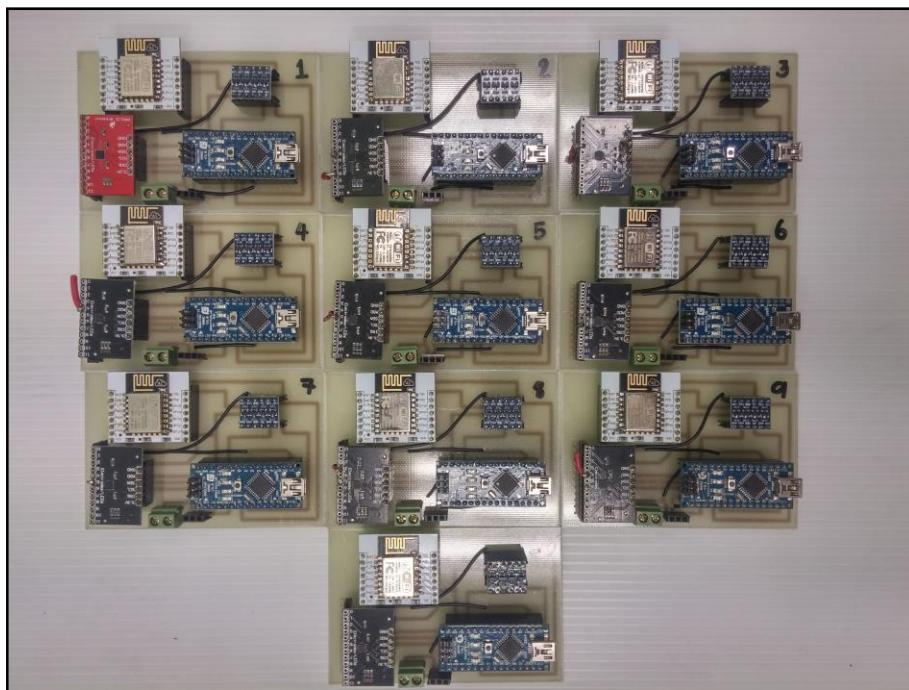
4.5 ชิ้นงาน WIPLED ที่เสร็จสมบูรณ์



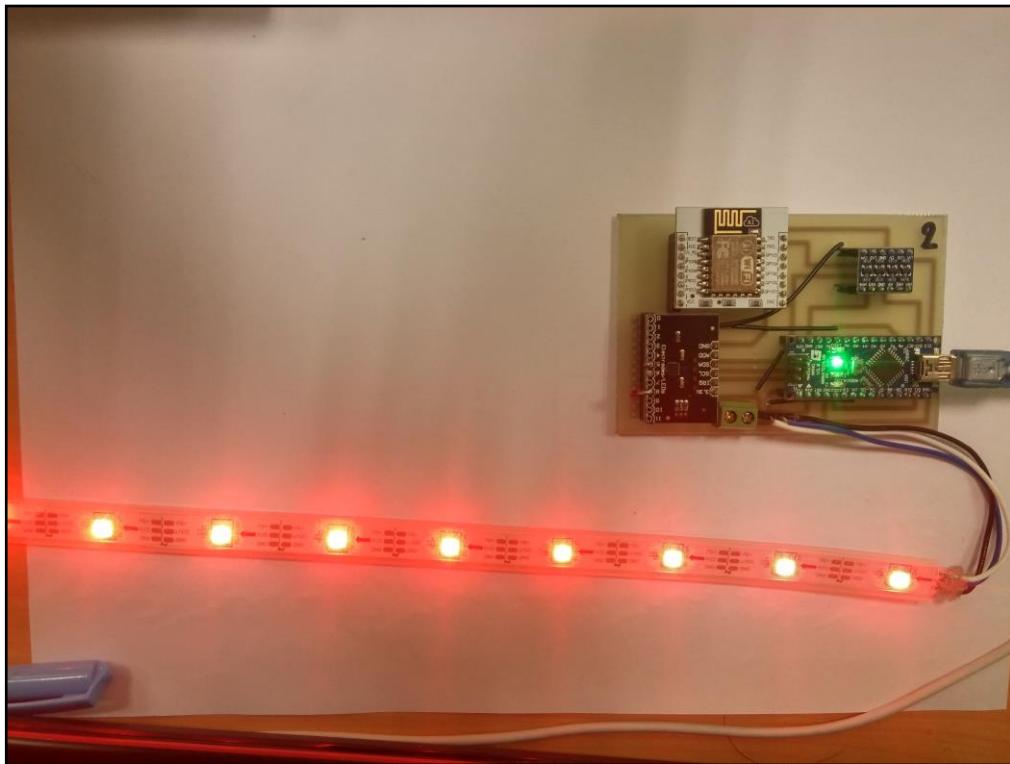
รูปที่ 4.29 แผ่น PCB ด้านลายทองแดงของ WIPLED



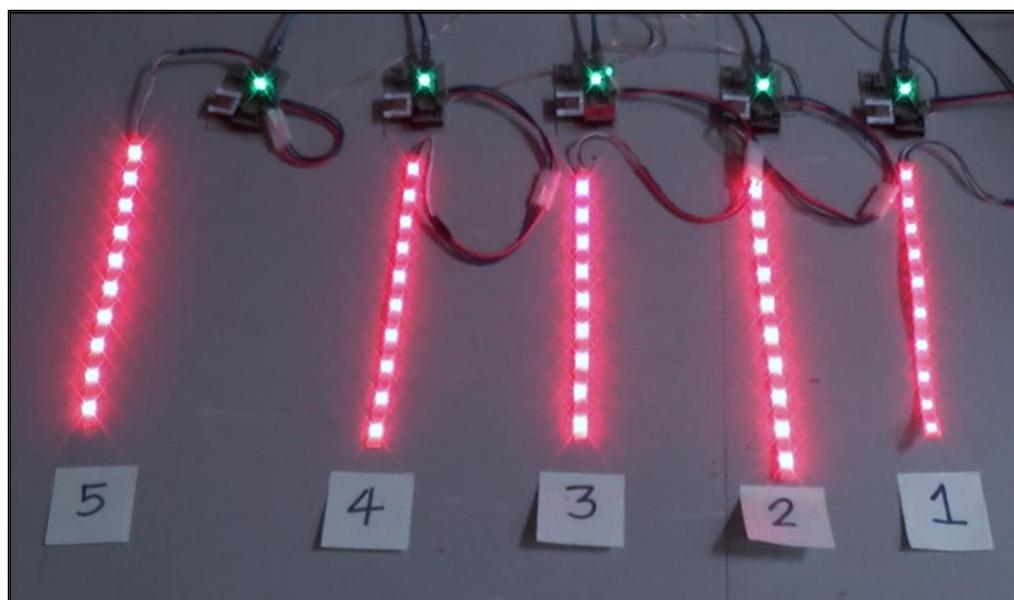
รูปที่ 4.30 แผ่น PCB ด้านลงอุปกรณ์



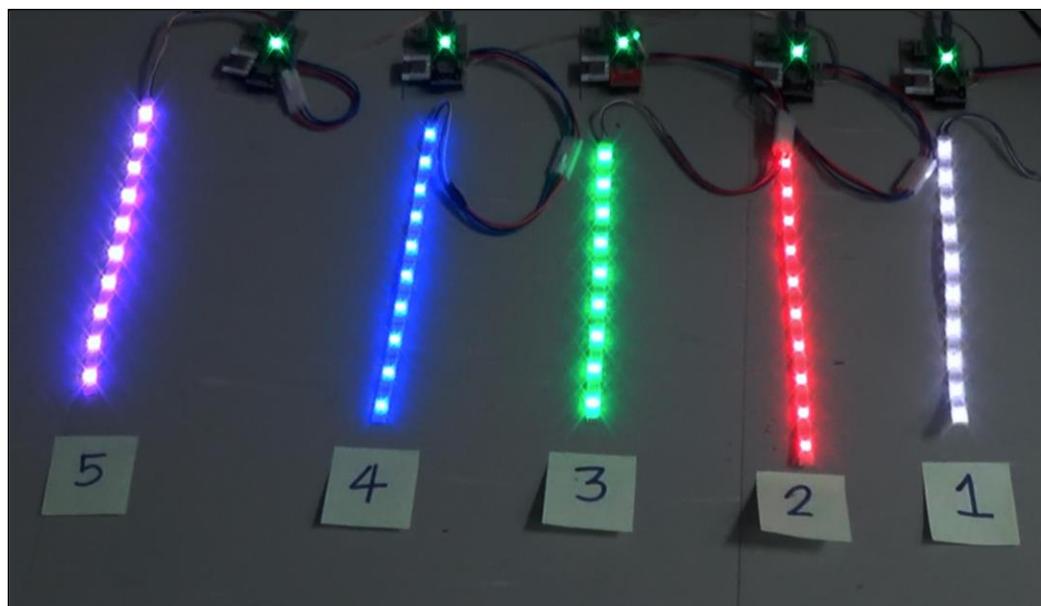
รูปที่ 4.31 WIPLED ทั้ง 10 ชิ้น



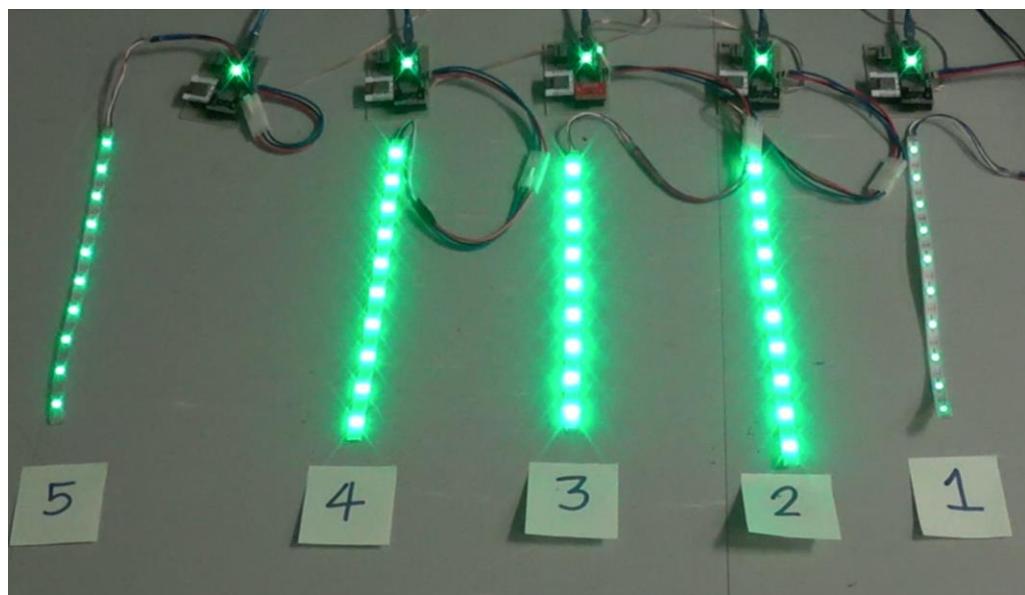
รูปที่ 4.32 การทำงานของ WIPLED เมื่อโปรแกรมควบคุมส่งข้อมูลให้แสดงสี



รูปที่ 4.33 การทำงานของ WIPLED ในโหมด Synchronize แบบ Pattern (mode 1)



รูปที่ 4.34 การทำงานของ WIPLED ในโหมด Sychronize แบบกำหนดสีได้ (mode 2)



รูปที่ 4.35 การทำงานของ WIPLED ในโหมด Interactive

จากที่ได้ทำการทดลองและสร้างชิ้นงานขึ้นรวมทั้งได้ทดสอบและแก้ไขปัญหาต่างๆที่เกิดขึ้นทำให้ สุดท้ายได้ WIPLED เสร็จสมบูรณ์สามารถทำงานในโหมด Synchronize โดยรับข้อมูลจากโปรแกรมควบคุมและ WIPLED สามารถแสดงสีได้ถูกต้องตามที่โปรแกรมได้ส่งมาดังรูปที่ 4.32 รูปที่ 4.33 และรูปที่ 4.34 ส่วนโหมด Interactive สามารถทำงานได้เช่นกันโดยการทดสอบตั้งค่า WIPLED ชิ้นที่อยู่ข้างกัน เมื่อสัมผัส WIPLED ที่รอบข้างไฟจาก LED จะติดตามที่ได้กำหนดไว้ดังรูปที่ 4.35 การทำงานระหว่าง 2 โหมดสามารถทำงานสลับกันได้อย่างไม่มีปัญหา

บทที่ 5

สรุปผล

ในบทที่ 5 จะกล่าวถึงสถานะในการทำงานแต่ละส่วนก่อนที่โครงงานจะสำเร็จ ปัญหาที่พะรุงห่วงการทดลองรวมถึงวิธีที่แก้ปัญหานั้นเพื่อให้โครงงานสามารถดำเนินต่อไปได้ การวางแผนในอนาคตเพื่อที่พัฒนางานต่อให้สามารถนำไปเป็นสินค้าเพื่อให้ผู้ที่สนใจนำไปใช้งานได้

5.1 ตารางแสดงสถานะความสำเร็จของงานในแต่ละส่วนของโครงการ

งานในแต่ละส่วนของโครงงาน	สถานะการทำงาน
การเชื่อมต่อผ่าน Router	เสร็จสิ้นสมบูรณ์
การรับ-ส่งข้อมูลระหว่างโมดูล กับ Server	เสร็จสิ้นสมบูรณ์
การทำงานระบบ Hardware	เสร็จสิ้นสมบูรณ์
การรับ-ส่งข้อมูลระหว่างโมดูล กับ โมดูล	เสร็จสิ้นสมบูรณ์
การเชื่อมต่อ Client หลายๆตัวพร้อมกันบน Server	เสร็จสิ้นสมบูรณ์
User Interfaces	เสร็จสิ้นสมบูรณ์
การออกแบบ WIPLIED	เสร็จสิ้นสมบูรณ์
การประกอบชิ้นงาน	เสร็จสิ้นสมบูรณ์
การทดสอบรับ-ส่งข้อมูลระหว่างโปรแกรมควบคุมกับ WIPLIED	เสร็จสิ้นสมบูรณ์

5.2 ปัญหาหรือประเด็นที่พบในการทำงานและแนวทางการแก้ไข

ปัญหา : เนื่องจากการส่งข้อมูลจากซอฟแวร์ควบคุมไปยัง client พร้อมกันต้องใช้การ broadcast ไปยัง ทุกตัวแต่ยังไม่สามารถทำได้ ทำให้เกิด delay ใน การส่งข้อมูลที่มีนุյย์สามารถรู้สึกได้

แนวทางการแก้ไข : เราจึงแก้ปัญหาโดยการเปลี่ยนซอฟแวร์ควบคุมโดยการแบ่ง thread ในการส่งข้อมูล ของแต่ละตัวซึ่งเสมือนกับว่าเป็นการ broadcast ไปยังทุกตัวพร้อมกัน

ปัญหา : NodeMCU สามารถทำงานร่วมกับ Capacitive Touch Sensor MPR121 ไม่เสียมากนัก มีอาการ Sensor ขาดการเชื่อมต่อบ้างในบางครั้ง และ เมื่อเปิด NodeMCU ทำงานไฟสักระยะหนึ่ง ประมาณ 10 นาที จะเกิดอาการห้าง Sensor ไม่สามารถรับการสัมผัสได้

แนวทางการแก้ไข : เปลี่ยนจากใช้ NodeMCU มาใช้ Arduino Nano ซึ่งทำให้ Capacitive Touch Sensor MPR121 ใช้งานได้ปกติ และเพิ่ม Wi-Fi Module ESP8266 เข้าไปเพื่อใช้ในการสื่อสารไร้สาย ทำให้การทำงาน ทั้งหมดเหมือนกับ NodeMCU ตามจุดประสงค์ที่ได้เลือกใช้ตั้งแต่แรก

ปัญหา : การ Update client บนหน้า UI ไม่สามารถ update แบบเรียลไทม์ได้

แนวทางการแก้ไข : ใช้วิธีการสร้างปุ่ม “Update client” ขึ้นมาแล้วจะ Update ทุกครั้งเมื่อมีการกดปุ่ม

ปัญหา : การส่งข้อมูลจาก Server ไปยัง client นั้น ตัวclientไม่สามารถอ่านค่าเป็น string ได้

แนวทางการแก้ไข : ส่งข้อมูลไปยังclientทีละ byte ให้ครบถ้วนตัว

ปัญหา : การส่งข้อมูลไปยัง Client ผ่าน Wi-Fi เมื่อส่งเป็นข้อมูลที่มีขนาดมากกว่า 60 byte จะทำให้ ข้อมูลสูญหายระหว่างการส่งจึงทำให้ Client ได้รับข้อมูลไม่ครบถ้วนและ WIPLED ไม่สามารถทำงานได้

แนวทางการแก้ไข : แบ่งการส่งข้อมูลออกเป็นชุด โดยความยาวของข้อมูลแต่ละชุดไม่เกิน 30 byte

ปัญหา : Digital Pin 0 และ 1 ของ Arduino Nano ไม่สามารถใช้งานร่วมกับ RGB LED ได้ตามที่ได้ออกแบบแผ่น PCB ไว้

แนวทางการแก้ไข : เนื่องจากการใช้ Arduino Nano มีการใช้ Serial Monitor ซึ่งใช้สำหรับการรับค่า และการแสดงผลในคอมพิวเตอร์ ซึ่งการรับ-ส่งข้อมูลนี้ใช้ Digital Pin 0 และ 1 ในการสื่อสารแบบ UART จึงแก้ไขด้วยการแก้วงจรในแผ่น PCB เปลี่ยนไปใช้ Digital Pin 6 แทน

5.3 ข้อจำกัดของโครงงาน

- การแสดงผลในโหมด Interactive ของ WIPLED สามารถแสดงผลในรูปแบบของการແຜ່ກະຈາຍໄປຢັງ WIPLED รอบข้างเท่านั้น และจะມີແສດງຜລ້າກັບ WIPLED ທີ່ເກີດການເປົ້າຍືນແປລັງແລ້ວ
- Capacitive touch sensor MPR121 สามารถตรวจจับຮຍະການສັນພັບໄດ້ມີເກີນ 1 ເຊັນຕີເມຕຣ
- ການຮັບ-ສ່າງຂໍ້ມູນຮະຫວ່າງ Server ກັບ WIPLED ມີຮຍະມີເກີນ 50 ເມຕຣ
- Arduino Nano ມີໜ່ວຍຄວາມຈຳ 32 Kbyte ປຶ້ງທຳໃຫ້ເມື່ອ Server ສ່າງຂໍ້ມູນໄປຢັງ WIPLED ນາກກວ່າ 10 Pattern ຈະທຳໃຫ້ໜ່ວຍຄວາມຈຳເຕີມ

5.4 ແຜນກາຣດຳເນີນຈານໃນอนาคต

- ເພີ່ມຈຳນວນ WIPLED ໃຫ້ມາກເຂົ້ນ ປຶ້ງຈຳນວນນາກທີ່ສຸດທີ່ສາມາດເຂື່ອມຕ່ອງ Server ໄດ້ເກົ່າຂົ້ນອູ່ກັບ Class Network address ປຶ້ງໃນໂຄຮງຈານນີ້ໃໝ່ Class C ສາມາດເຂື່ອມຕ່ອງ WIPLED ໄດ້ທັງໝາດ 253 ຕ້າ
- ເພີ່ມ Features ໃຫ້ກັບ WIPLED ເຊັ່ນ ຜູ້ໃຊ້ງານສາມາດຄວບຄຸມການແສດງສື່ຂອງ LED ໄດ້ມາກເຂົ້ນ ອີ່ວ
- ເພີ່ມກາຣເລີ່ມເສີຍງວ່າມກັບແສງໄຟ
- ພັດນາ User Interfaces ໃຫ້ຜູ້ໃຊ້ງານຄວບຄຸມໄດ້ຈ່າຍເຂົ້ນ
- ພັດນາ Application ບນ Smart Phone ໃນກາຣຄວບຄຸມ WIPLED

5.5 ການນຳໂຄຮງຈານໄປໃຊ້ງານ

ເມື່ອກາຣທຳໂຄຮງຈານເສົ່າງສົມບູຮົນ ໃນສ່ວນຂອງ WIPLED ຂອງເຮັນສາມາດນຳໄປໃຊ້ໃນການແສດງ ທີ່ອງການອືເວນທ່າງໆໄດ້ ໂດຍ WIPLED ນັ້ນສາມາດທີ່ຈະອອກແບບກລ່ອງໄສ່ວ່າຈະໃໝ່ຮູ່ປ່າງຕາມຄວາມຕ້ອງການຂອງ ຈານນັ້ນໄດ້ ແລະສາມາດເຄີ່ອນຍ້າຍຕໍ່ແນ່ງຂອງWIPLED ໃຫ້ເໜາະສົມກັບຈານທ່າງໆ ໄດ້ໂດຍຈ່າຍ ປຶ້ງຜູ້ໃຊ້ຈະສາມາດ ກຳທັນຮຽບແບບກາຮງຂອງ WIPLEDໄດ້ດ້ວຍຕົນເອງແລະປັບປຸງເປົ້າຍືນໄດ້ຍ່າງຈ່າຍໄໝ່ຢູ່ຍາກ ມີໜ້າໂປຣແກຣມທີ່ສະດວກ ຕ່ອກາຣໃຊ້ງານແລະເຂົ້າໃຈ່ຈ່າຍຈຶ່ງທຳໃຫ້ງານທີ່ຜູ້ນຳພົງງານຂອງເຮົາໄປປະຍຸກຕິໃຫ້ນັ້ນດູໂດດເດັ່ນ ແລະນ່າສັນໃຈຢູ່ເຂົ້ນ ຖ້ວອ່າງງານທີ່ໂຄຮງຈານຂອງເຮົາສາມາດນຳໄປປະຍຸກຕິໃຫ້ໄດ້ ເຊັ່ນ ຈາກໄຟທຽນລານໜ້າເຊົ່າເວົ້າໃນຫ່ວງເທິກາລຄ ຮິສາມາສົດ ຈານແສດງໄຟ່ຈ່າຍເທິກາລສົງທ້າຍປີເກ່າຕ້ອນຮັບປີໃໝ່ທີ່ລານຄນເມື່ອງ ເປັນຕົ້ນ

ໂຄຮງຈານ WIPLED ໃນສ່ວນຂອງໂໂມດ Interactive ສາມາດນຳປັບປຸງເປົ້າຍືນຮຽບແບບກາຮງເລີ່ມໄດ້ ເຊັ່ນ ເມື່ອມີກາຣສັນພັບທີ່ WIPLED ຂົ້ນທີ່ໄຟຈະສຸມໄປຕິດທີ່ WIPLED ຂົ້ນອື່ນ ເມື່ອມີກາຣສັນພັບທີ່ຂົ້ນທີ່ໄຟຕິດ ໄຟຈະດັບລົງ ແລະສຸມໄປຕິດທີ່ຕ້ວອື່ນ ເປັນເສີມອິນເກມທີ່ໃຫ້ໜ້າສາມາດເລັນມີປົງສັນພັນຮີໄດ້

บรรณานุกรม

1. LCI Paper Co., Inc. , “The RGB Color Model” , lcipaper [ออนไลน์] , เข้าถึงได้จาก :
<http://www.lcipaper.com/pantone-cmyk-rgb.html> [18 พฤศจิกายน 2558].
2. ผศ.ดร.พิมพ์เพ็ญ พรเฉลิมพงศ์, ผศ.ดร.นวภัทรานุนาค , “Capacitive proximity sensor” , riko [ออนไลน์] , เข้าถึงได้จาก : <http://www.riko.in.th/knowledge-sensor/61-what-is-capacitive-sensor.html> [18 พฤศจิกายน 2558].
3. ENGINEERED IN NYC Adafruit , “ADAFRUIT NEOPIXEL DIGITAL RGB LED STRIP” , adafruit [ออนไลน์], เข้าถึงได้จาก : <https://www.adafruit.com/products/1376> [18 พฤศจิกายน 2558].
4. Worldsemi , “WS2812B Intelligent control LED integrated light source” , adafruit [ออนไลน์] , เข้าถึงได้จาก : <https://www.adafruit.com/datasheets/WS2812B.pdf> [19 พฤศจิกายน 2558].
5. Freescale Semiconductor, Inc. , “Freescale Semiconductor TechnicalData” , sparkfun [ออนไลน์], เข้าถึงได้จาก :
<https://www.sparkfun.com/datasheets/Components/MPR121.pdf> [19 พฤศจิกายน 2558].
6. ETT, “คู่มือการใช้งาน ET-MINI LOGIC LEVEL SHIFTER” , ett [ออนไลน์], เข้าถึงได้จาก :
http://www.ett.co.th/product/InterfaceBoard/P-ET-A-00297/ET_MINI_LOGIC_LEVEL.pdf [7 กุมภาพันธ์ 2559]

7. ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, “คู่มือการใช้งาน RGB LED WS2812” , Embedded Space [ออนไลน์], เข้าถึงได้จาก :
<http://cpre.kmutnb.ac.th/esl/learning/index.php?article=single-ws2812-rgb-fpga> [10 กุมภาพันธ์ 2559]
8. Espressif Systems, “Datasheet AT Command สำหรับโมดูล Wi-Fi ESP8266” , Espressif Systems IOT Team [ออนไลน์], เข้าถึงได้จาก :
https://cdn.sparkfun.com/assets/learn_tutorials/4/0/3/4A-ESP8266_AT_Instruction_Set_EN_v0.30.pdf [21 กุมภาพันธ์ 2559]
9. TETRO, “mini burble” , TETRO [ออนไลน์], เข้าถึงได้จาก :
<http://www.tetro.fr/en/tetro/tetro-realizations/samsung-electronics-mini-burble-paris> [3 มีนาคม 2559]
10. Marcelo Coelho, Jamie Zigelbaum and Joshua Kopin, “six-forty-four-eighty” , MIT Media Lab [ออนไลน์], เข้าถึงได้จาก : <http://fluid.media.mit.edu/projects/six-forty-four-eighty> [15 มีนาคม 2559]

ภาคผนวก

1. AT Command สำหรับใช้ตั้งค่าโมดูล Wi-Fi ESP8266

1.1 Basic AT Command Set

The ESP8266 wireless WiFi modules can be driven via the serial interface using the standard AT commands. Here is a list of some basic AT commands that can be used.

Basic	
Command	Description
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT commands echo or not
AT+RESTORE	Factory Reset
AT+UART	UART configuration, [@deprecated]
AT+UART_CUR	UART current configuration
AT+UART_DEF	UART default configuration, save to flash
AT+SLEEP	Sleep mode
AT+RFPOWER	Set maximum value of RF TX Power
AT+RFVDD	Set RF TX Power according to VDD33

1.2 Wi-Fi Functions

WiFi Connectivity Functions Invoked by AT commands	
Command	Description
AT+CWMODE	WIFI mode (sta/AP/sta+AP) , [@deprecated]
AT+CWMODE_CUR	WIFI mode (sta/AP/sta+AP) Won't save to Flash
AT+CWMODE_DEF	WIFI default mode (sta/AP/sta+AP) Save to Flash
AT+CWJAP	Connect to AP, [@deprecated]
AT+CWJAP_CUR	Connect to AP, won't save to Flash
AT+CWJAP_DEF	Connect to AP, save to Flash
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP
AT+CWSAP	Set configuration of ESP8266 softAP [@deprecated]
AT+CWSAP_CUR	Set configuration of ESP8266 softAP Won't save to Flash.
AT+CWSAP_DEF	Set configuration of ESP8266 softAP Save to Flash.
AT+CWLIF	Get station's IP which is connected to ESP8266 softAP
AT+CWDHCP	Enable/Disable DHCP, [@deprecated]
AT+CWDHCP_CUR	Enable/Disable DHCP, won't save to Flash
AT+CWDHCP_DEF	Enable/Disable DHCP, save to Flash
AT+CWAUTOCONN	Connect to AP automatically when power on
AT+CIPSTAMAC	Set mac address of ESP8266 station [@deprecated]
AT+CIPSTAMAC_CUR	Set mac address of ESP8266 station Won't save to Flash.

1.3 TCP/IP Related AT Commands

TCP/IP	
Command	Description
AT+ CIPSTATUS	Get connection status
AT+CIPSTART	Establish TCP connection or register UDP port
AT+CIPSEND	Send data
AT+CIPSENDEX	Send data, if <length> or "\0" is met, data will be sent
AT+CIPSENDBUF	Write data into TCP-send-buffer
AT+CIPBUFORESET	Reset segment ID count
AT+CIPBUFSTATUS	Check status of TCP-send-buffer
AT+CIPCHECKSEQ	Check if a specific segment is sent or not
AT+CIPCLOSE	Close TCP/UDP connection
AT+CIFSR	Get local IP address
AT+CIPMUX	Set multiple connections mode
AT+CIPSERVER	Configure as server
AT+CIPMODE	Set transmission mode
AT+SAVETRANSLINK	Save transparent transmission link to Flash
AT+CIPSTO	Set timeout when ESP8266 runs as TCP server
AT+CIUPDATE	Upgrade firmware through network
AT+PING	Function PING
AT+CIPDINFO	Show remote IP and remote port with "+IPD"