สมาร์ทแอพพลิเคชั่นสำหรับผู้พิการทางสายตาเพื่อควบคุมเครื่องใช้ไฟฟ้า
SMART APPLICATION FOR VISUAL IMPAIRMENT PERSON TO CONTROL APPLIANCES

เบญจวรรณ วันดี
วันฉัตร ไชยวุฒิ
อรุณี บุญประมวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

สมาร์ทแอพพลิเคชั่นสำหรับผู้พิการทางสายตาเพื่อควบคุมเครื่องใช้ไฟฟ้า
SMART APPLICATION FOR VISUAL IMPAIRMENT PERSON TO CONTROL APPLIANCES

เบญจวรรณ วันดี
วันฉัตร ไชยวุฒิ
อรุณี บุญประมวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART APPLICATION FOR VISUAL IMPAIRMENT PERSON TO CONTROL APPLIANCES

BENJAWAN WANDEE
WANCHAT CHAIYAWUTH
ARUNEE BUNPRAMUAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2559
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ สมาร์ทแอพพลิเคชั่นสำหรับผู้พิการทางสายตาเพื่อควบคุมเครื่องใช้ไฟฟ้า
SMART APPLICATION FOR VISUAL IMPAIRMENT PERSON TO CONTROL APPLIANCES

นักศึกษาผู้จัดทำ นางสาวเบญจวรรณ วันดี **รหัสนักศึกษา** 56010706
นายวันฉัตร ไชยวุฒิ **รหัสนักศึกษา** 56011113
นางสาวอรุณี บุญประมวล **รหัสนักศึกษา** 56011433

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2559

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ทรงชัย วีระทวีมาศ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	สมาร์ทแอพพลิเคชั่นสำหรับผู้พิการทางสายตาเพื่อควบคุมเครื่องใช้ไฟฟ้า SMART APPLICATION FOR VISUAL IMPAIRMENT PERSON TO CONTROL APPLIANCES		
นักศึกษาผู้จัดทำ	นางสาวเบญจวรรณ	วันดี	รหัสนักศึกษา 56010706
	นายวันฉัตร	ไชยวุฒิ	รหัสนักศึกษา 56011113
	นางสาวอรุณี	บุญประมวล	รหัสนักศึกษา 56011433
อาจารย์ที่ปรึกษา	รองศาสตราจารย์.ทรงชัย	วีระทวีมาศ	
ปีการศึกษา	2559		

บทคัดย่อ

ในปัจจุบันมีเทคโนโลยีหลากหลายด้านที่ช่วยอำนวยความสะดวกให้แก่ผู้พิการทางสายตาแต่เทคโนโลยีเหล่านี้ก็ยังไม่ครอบคลุมในทุกๆด้าน ตัวอย่างเช่นเทคโนโลยีสำหรับช่วยเหลือด้านการควบคุมเครื่องใช้ไฟฟ้าซึ่งเป็นสิ่งจำเป็นที่ใช้ในการดำเนินชีวิตประจำวัน การใช้งานเครื่องใช้ไฟฟ้าสำหรับผู้บกพร่องทางสายตาค่อนข้างมีความลำบากและอันตราย ดังนั้นคณะผู้จัดทำจึงได้คิดค้นแอพพลิเคชั่นที่ช่วยเหลือผู้พิการทางสายตาในการควบคุมเครื่องใช้ไฟฟ้า โดยเป็นแอพพลิเคชั่นที่ใช้เสียงในการสั่งเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบแอนดรอยด์ที่เชื่อมต่อเครือข่ายอินเตอร์เน็ต ซึ่งเป็นการนำแนวคิดที่ทันสมัยอย่าง Internet of Things หรือ IoT ที่เป็นการนำสิ่งต่างๆเชื่อมโยงกับโลกอินเตอร์เน็ตมาเป็นแม่แบบในการจัดทำ ช่วยให้สามารถเชื่อมต่อข้อมูลและสั่งการควบคุมอุปกรณ์ต่างๆจากระยะไกลได้ง่ายยิ่งขึ้น นอกจากนี้แอพพลิเคชั่นนี้บุคคลทั่วไปสามารถนำไปใช้เพื่อให้เกิดความสะดวกสบายได้เช่นกัน เช่น สามารถสั่งปิดเครื่องใช้ไฟฟ้าในกรณีที่ลืมปิดเพื่อป้องกันอัคคีภัย สั่งเปิดเครื่องปรับอากาศหรือไฟก่อนกลับถึงบ้านเพื่อความสะดวกสบาย เป็นต้น แอพพลิเคชั่นที่คณะผู้จัดทำได้เขียนขึ้นนั้นจะใช้ Google Voice เป็นดาต้าเบสในการตรวจจับเสียงคำพูด สำหรับการสั่งการจะใช้ภาษาอังกฤษในควบคุมการเปิด-ปิดเครื่องใช้ไฟฟ้าในแต่ละแชนแนล ซึ่งจะมีสองโหมดการใช้งานคือ โหมดสำหรับบุคลทั่วไป เมื่อผู้ใช้งานสั่งการด้วยเสียงจะมีการแสดงประโยคที่ผู้ใช้งานพูดบนหน้าจอเพื่อใช้ในการเช็คความถูกต้องของคำสั่ง และโหมดสำหรับผู้พิการทางสายตา ซึ่งจะมีเสียงตอบกลับจากแอพพลิเคชั่นเพื่อเป็นการเช็คความถูกต้องของคำสั่งเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	SMART APPLICATION FOR VISUAL IMPAIRMENT PERSON TO CONTROL APPLIANCES	
Authors	Ms.Benjawan	Wandee
	Mr.Wanchat	Chaiyawuth
	Ms.Arunee	Bunpramuan
Thesis Advisor	Assoc.Prof.Songchai	Weerathaweemas
Year	2016	

ABSTRACT

At present, there are various technologies that facilitate the visually impaired person but these technologies are not covered on all sides such as technology for helping to control electrical appliances, which is essential for everyday life. The use of electric equipment for the visual impairment person is quite difficult and dangerous. Therefore, the team has developed applications that assist visually impaired person in controlling electric appliances. It is an application for controlling electric equipment by using voice command via the internet on Android mobile phone. This is bringing modern concepts like Internet of Things, or IoT, that is linking things to the internet world as a template for the development. It makes easier to connect and control devices remotely. In addition, these applications can be used by the general public for comfort. For example, it can be used to turn off the appliance in case of forgetting to close it to prevent fire or turn on the air conditioner or lamp before returning home for convenient etc. This application uses Google Voice as a database for speech-to-speech detection. For the direction will use English to control appliance on each channel. There are two modes of operation: General Mode, when user voice command the sentence spoken will display on the screen to check the accuracy of the command. And Mode for the visually impaired person, this will be the voice response from the application to check the accuracy of the command too.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุลวงไปได้ด้วยดีทางคณะผู้จัดทำขอขอบพระคุณทางหลักสูตรวิศวกรรมการวัดคุมที่ให้ความรู้ตลอดระยะเวลา 3 ปีและให้ความอนุเคราะห์ทางด้านสถานที่และอุปกรณ์ในการทำปริญญานิพนธ์ ขอขอบพระคุณอาจารย์ที่ปรึกษา รศ.ทรงชัย วีระทวีมาศ ที่คอยให้คำปรึกษาและความรู้ทางเทคนิคที่ใช้ในการทำปริญญานิพนธ์ตลอดระยะเวลาทั้งปีการศึกษา

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญรูป.....VI

บทที่ 1 บทนำ.....1
1.1 ความสำคัญของปริญญานิพนธ์.....1
1.2 วัตถุประสงค์ของการทำปริญญานิพนธ์.....1
1.3 ขอบเขตของปริญญานิพนธ์.....1
1.4 วิธีการดำเนินงาน.....2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์.....2

บทที่ 2 หลักการของโปรแกรมและอุปกรณ์ที่เกี่ยวข้อง.....3
2.1 MIT App Inventor.....3
2.1.1 การพัฒนา Apps ด้วย MIT App Inventor.....6
2.2 การสื่อสารผ่านระบบTCP/UDP.....7
2.2.1 TCP: (Transmission Control Protocol).....7
2.2.2 UDP: (User Datagram Protocol).....7
2.3 Arduino UNO.....9
2.4 ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino.....11
2.5 การเขียนโปรแกรมและการติดตั้ง.....13
2.6 การตั้งค่าบอร์ด ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino.....20
2.7 การใช้งาน Relay.....26

บทที่ 3 แผนการดำเนินงาน.....29
3.1 การดำเนินงานในส่วนที่เกี่ยวกับ Arduino.....29
3.1.1 โปรแกรมที่พัฒนาขึ้นในส่วนคำสั่งของ Arduino.....30
3.2 การดำเนินงานในส่วนของแอพพลิเคชั่น.....32
3.2.1 โปรแกรมที่พัฒนาขึ้นในส่วนคำสั่งของแอพพลิเคชั่น.....33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3 การดำเนินงานในส่วนของระบบวงจรไฟฟ้า....34
3.3.1 การพัฒนาในส่วนของระบบวงจรไฟฟ้า....34

บทที่ 4 โปรแกรมและ Application ที่ได้พัฒนาขึ้น....36
4.1 การทดสอบการทำงานในส่วนคำสั่งจากแอพพลิเคชั่น....37
4.2 การทดสอบการทำงานในส่วนคำสั่งจาก Arduino....38
4.2.1 แสดงการทำงานในส่วนของคำสั่งเปิดเครื่องใช้ไฟฟ้า (Turn on)....38
4.2.2 แสดงการทำงานในส่วนของคำสั่งปิดเครื่องใช้ไฟฟ้า (Turn off)....39

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ....42
5.1 สรุปผลการทดลอง....42
5.2 ปัญหาและวิธีการแก้ไข....42
5.2.1 ปัญหาที่พบ....42
5.2.2 วิธีแก้ไขปัญหา....42
5.3 ข้อเสนอแนะ....43

บรรณานุกรม....44

ภาคผนวก....45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 หน้าเว็บไซต์อย่างเป็นทางการของ MIT App Inventor....4
2.2 การใช้ Icon แทนรหัสคำสั่ง และผลจากการทำงาน....4
2.3 หน้าโปรแกรมเริ่มต้นการสร้างแอพพลิเคชั่นที่นำมาใช้งาน....5
2.4 การสร้าง Projects....5
2.5 หน้าแอพพลิเคชั่นจำลอง....6
2.6 รูปหน้าจอหลักที่ใช้พัฒนาแอพพลิเคชั่น....6
2.7 การสื่อสารแบบ TCP....8
2.8 บอร์ด Arduino ต่อกับ LED....9
2.9 Arduino UNO....10
2.10 ส่วนประกอบต่างๆเกี่ยวกับ Arduino UNO....11
2.11 ESP8266 ESP-13 Wi-Fi Web Sever Shield....12
2.12 ขาของ ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino....12
2.13 โปรแกรม Arduino....13
2.14 เลือกบอร์ดArduinoที่ต้องการUpload....13
2.15 เลือกหมายเลข Com Port ของบอร์ด....14
2.16 ตรวจสอบความถูกต้องโดยกดปุ่ม Verify และ Compile โค้ด....14
2.17 ข้อความแสดง “Done uploading”....15
2.18 หน้าแรกของโปรแกรม....15
2.19 การ Add Library....16
2.20 การเลือกชนิดของบอร์ด....16
2.21 การเลือก Port Wi-Fi....17
2.22 เลือก ESP8266 Wi-Fi....17
2.23 เลือก Target Board ESP8266....18
2.24 เลือก Serial Port....18
2.25 เลือก Programmer ไปที่ ESP8266....19
2.26 การ Compile โปรแกรม....19
2.27 ทดสอบการใช้งาน Webserver....20
2.28 หน้าต่างการกำหนดค่า....21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

2.29 โปรแกรม Arduino และรหัสตัวอย่าง....22
2.30 การเชื่อมต่อในรูปแบบ TCP กับ IP ที่อยู่....22
2.31 การส่งข้อมูลเมื่อเชื่อมต่อสำเร็จ....23
2.32 การรับที่อยู่ IP ชั่วคราว....23
2.33 การตั้งชื่อและ Password ของ Wi-Fi....24
2.34 การเชื่อต่อรูปแบบ TCP กับ IP ที่อยู่ในรูปแบบอัตโนมัติ....25
2.35 การส่งข้อมูลเมื่อเชื่อมต่อสำเร็จ....25
2.36 สัญลักษณ์ในวงจรไฟฟ้าของรีเลย์....26
2.37 การทำงานของรีเลย์ในสภาวะต่างๆ....26
2.38 จำนวนของ Pole และจำนวน Throw ของรีเลย์....27
2.39 หน้าสัมผัสแบบ A....27
2.40 หน้าสัมผัสแบบ B....28
2.41 หน้าสัมผัสแบบ C....28
2.42 Relay 8 Channel 5V relay Active High/Low Relay Module Board for Arduino....28
3.1 อุปกรณ์ที่เลือกใช้งาน....29
3.2 แผนผังแสดงการทำงานของคำสั่งการควบคุมรีเลย์....30
3.3 แผนผังแสดงการทำงานของคำสั่งการตั้งเวลา....31
3.4 MIT APP Inventor....32
3.5 ฟังก์ชั่นการทำงานใน MIT APP Inventor....32
3.6 แผนผังแสดงการทำงานของแอพพลิเคชั่น....33
3.7 วงจรไฟฟ้าที่ออกแบบ....34
3.8 แสดงการต่อวงจรไฟฟ้าภายในระบบที่ใช้งาน....35
4.1 การรับ-ส่งคำสั่งผ่านระบบ Wi-Fi ควบคุมเครื่องใช้ไฟฟ้า....36
4.2 แสดงคำสั่งเปิดการใช้งานเครื่องใช้ไฟฟ้าที่แสดงบนแอพพลิเคชั่น....37
4.3 แสดงคำสั่งปิดการใช้งานเครื่องใช้ไฟฟ้าที่แสดงบนแอพพลิเคชั่น....38
4.4 แสดงการทำงานในคำสั่งเปิด (Turn On) ของแต่ละช่องการควบคุม....39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.5 แสดงการทำงานในคำสั่งปิด (Turn Off) ของแต่ละช่องการควบคุม.....40
4.6 แสดงการทำงานในคำสั่งเปิดการควบคุมในโหมดการทำงานของผู้พิการทางสายตา.....40
4.7 สถานะแสดงการตอบสนองของเครื่องใช้ไฟฟ้าว่ามีการทำงานอยู่หรือไม่.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

Internet of Things (IoT) หรืออินเตอร์เน็ตในทุกสิ่ง เป็นแนวคิดใหม่ที่กำลังเป็นที่นิยมในปัจจุบัน เนื่องจากเป็นการนำอุปกรณ์ต่างๆเชื่อมโยงกับโลกอินเตอร์เน็ต ช่วยให้สามารถเชื่อมต่อข้อมูลได้อย่างกว้างขวาง อีกทั้งยังสามารถสั่งการควบคุมอุปกรณ์ต่างๆจากระยะไกลได้ ดังนั้นจึงนำแนวคิดนี้มาสร้างสรรค์เทคโนโลยีที่มีประโยชน์แก่ผู้อื่นโดยมุ่งเน้นไปที่ผู้พิการทางสายตา จากสถิติข้อมูลผู้พิการทางการมองเห็นที่มีบัตรประจำตัวทั่วประเทศไทยมีทั้งหมด 181,821 ราย (ข้อมูลวันที่ 2 พ.ย. 2558) โดยยังไม่รวมถึงจำนวนที่ยังไม่ได้ลงทะเบียนไว้ จะเห็นได้ว่าผู้พิการทางสายตามีจำนวนค่อนข้างสูงแต่เทคโนโลยีที่ช่วยเหลือกลับมีน้อย ดังนั้นจึงได้คิดค้นแอพพลิเคชั่นที่ช่วยเหลือผู้พิการทางสายตาในการควบคุมเครื่องใช้ไฟฟ้าขึ้น เนื่องจากเครื่องใช้ไฟฟ้าเป็นสิ่งจำเป็นที่ใช้ในการดำเนินชีวิตประจำวัน เป็นการอำนวยความสะดวกแก่ผู้พิการทางสายตาและช่วยป้องอันตรายจากการใช้งานเครื่องใช้ไฟฟ้าที่อาจเกิดขึ้น โดยเป็นแอพพลิเคชั่นที่ใช้เสียงในการสั่งเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือระบบแอนดรอยด์ซึ่งง่ายต่อการใช้งาน โดยเชื่อมต่อผ่านเครือข่ายอินเตอร์เน็ตทำให้สามารถสั่งงานได้จากระยะไกลได้ นอกจากนี้แอพพลิเคชั่นนี้บุคคลทั่วไปสามารถนำไปใช้ได้เช่นกันเพื่อให้เกิดความสะดวกสบาย เช่น สั่งเปิดเครื่องปรับอากาศก่อนกลับถึงบ้าน สั่งเปิดไฟขณะไม่อยู่บ้าน เป็นต้น อีกทั้งยังสามารถสั่งปิดเครื่องใช้ไฟฟ้าในกรณีที่ลืมปิดได้ด้วยเช่นกัน เพื่อป้องกันและลดความเสี่ยงการเกิดอัคคีภัยจากไฟฟ้าลัดวงจรเนื่องจากการทำงานเกินกำลัง

1.2 วัตถุประสงค์ของการทำปริญญานิพนธ์

1. เพื่อนำความรู้ที่ได้ศึกษามาสร้างประโยชน์ให้แก่ผู้พิการทางสายตาและบุคคลทั่วไป
2. เพื่อสร้างแอพพลิเคชั่นอำนวยความสะดวกแก่ผู้ใช้งานในการควบคุมเครื่องใช้ไฟฟ้าระยะไกล
3. เพื่อช่วยลดการเกิดอัคคีภัยจากเครื่องใช้ไฟฟ้าทำงานเกินกำลัง
4. เพื่อนำความรู้ทางด้านไมโครคอนโทรลเลอร์ วงจรไฟฟ้าอิเล็กทรอนิกส์ และเครือข่ายการสื่อสารข้อมูล มาประยุกต์ใช้งานจริง

1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถสร้างระบบควบคุมเครื่องใช้ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ตได้ โดยใช้การติดต่อสื่อสารแบบ TCP Protocol
2. เข้าใจและสามารถเขียนโปรแกรมการใช้งานบน Arduino Boards ได้
3. สามารถเขียนแอพพลิเคชั่นรับคำสั่งเสียงบนโทรศัพท์มือถือได้ โดยใช้ Google Voice เป็นดาต้าเบสเพื่อนำมาใช้งานบนเครือข่ายไร้สาย
4. สามารถนำความรู้ทางด้านไมโครคอนโทรลเลอร์ วงจรไฟฟ้าอิเล็กทรอนิกส์ และเครือข่ายการสื่อสารข้อมูล มาสร้างเป็นชิ้นงานที่สามารถใช้ได้จริ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีการดำเนินการ

1. ศึกษาการใช้งานและคุณสมบัติของ Arduino Boards
2. ศึกษาการเขียนแอพพลิเคชั่นโดยใช้โปรแกรม MIT APP INVENTOR
3. ศึกษาการสื่อสารผ่านระบบ TCP Protocol
4. สร้างรูปแบบแอพพลิเคชั่นและเขียนคำสั่งการทำงานบนแอพพลิเคชั่น
5. ต่อวงจรไฟฟ้าของอุปกรณ์ และทำการทดสอบการทำงานของระบบ
6. ทำการปรับปรุงและแก้ไขชุดคำสั่งบนแอพพลิเคชั่นและ Arduino Boards

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์

1. ได้รับความรู้ด้านการใช้ประโยชน์จาก Arduino Boards
2. สามารถนำ Arduino Boards มาประยุกต์ใช้งานจริงได้
3. ได้รับความรู้เกี่ยวกับเครือข่ายการสื่อสารแบบไร้สาย
4. เข้าใจและสามารถเขียนคำสั่งการทำงานของ Arduino Boards ได้
5. ได้รับความรู้ด้านการเขียนแอพพลิเคชั่นเพื่อใช้สั่งงานได้จริง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการของโปรแกรมและอุปกรณ์ที่เกี่ยวข้อง

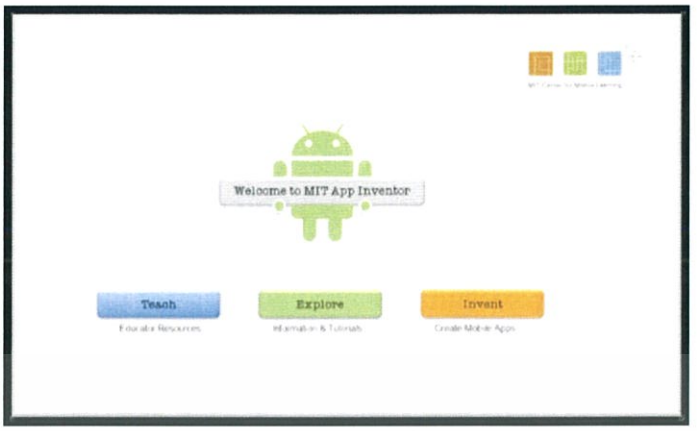
ในงานวิจัยนี้แบ่งเป็น 3 ส่วนสำคัญคือ ส่วนรับคำสั่ง ส่วนแปลงคำสั่ง และส่วนควบคุมการทำงาน โดยส่วนรับคำสั่งคือแอพพลิเคชั่นบนโทรศัพท์มือถือที่จะรับคำสั่งเสียงจากผู้ใช้งาน เพื่อแปลงเป็นข้อมูลส่งให้แก่บอร์ด Arduino แอพพลิเคชั่นนี้ถูกเขียนขึ้นด้วยใช้โปรแกรม MIT App Inventor และใช้ TCP Protocol ในการติดต่อสื่อสาร ส่วนที่สองคือส่วนแปลงคำสั่ง เป็นส่วนที่รับข้อมูลจากแอพพลิเคชั่นผ่านเครือข่ายอินเตอร์เน็ตแล้วแปลงข้อมูลที่ได้เป็นคำสั่งเพื่อส่งไปควบคุมการทำงานของรีเลย์ ส่วนแปลงคำสั่งมีอุปกรณ์ที่สำคัญคือ บอร์ด Arduino UNO และ ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino ซึ่งเป็นโมดูล Wi-Fi มาประกอบกันเป็นชุดไมโครคอนโทรลเลอร์ และส่วนสุดท้ายคือส่วนควบคุมการทำงาน ในที่นี้คือบอร์ดรีเลย์ซึ่งใช้ในการตัดต่อวงจรไฟฟ้าเพื่อควบคุมการทำงานของเครื่องใช้ไฟฟ้า

2.1 MIT App Inventor

MIT App Inventor [6] คือโปรแกรมสร้าง Application บนอุปกรณ์ที่ใช้ระบบปฏิบัติการ Android Apps โดยบริษัท Google และ MIT ได้ผลิตออกมา เพื่อให้ผู้ที่สนใจสามารถทำความเข้าใจและพัฒนา Apps ได้ง่ายเนื่องจากโปรแกรมนี้ง่ายต่อการทำความเข้าใจและใช้งานง่าย ต่อมา Google ได้ถอนตัวออกและปล่อยให้ MIT เป็นผู้พัฒนาต่อ โดยมุ่งเน้นในด้านการเรียนการสอนเป็นหลัก

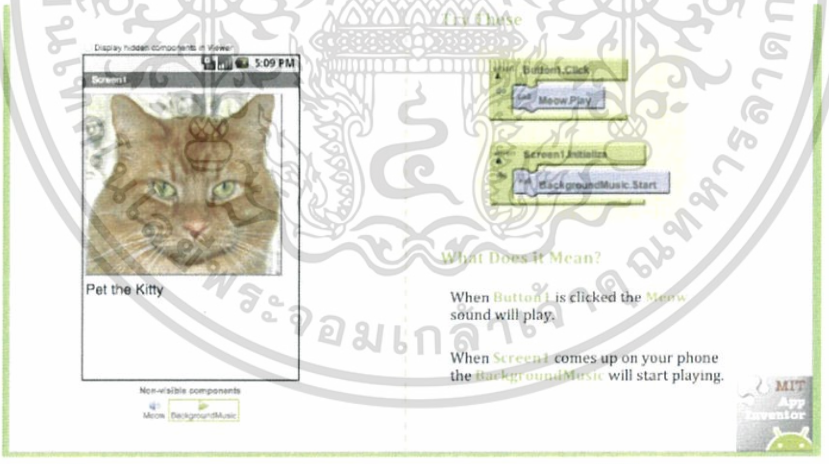
App Inventor เป็นเครื่องมือที่นิยมสำหรับการเรียนการสอน หรือการเริ่มต้นพัฒนา Apps บน Android เนื่องจาก App เป็นแบบ Visualization คือใช้ Block แทนรหัสคำสั่ง ผู้ใช้เพียงเลือก Block แล้วลากมาวางในพื้นที่ทำงาน กำหนดค่าให้กับ Block จากนั้น App จะทำการแปลงจาก Block เป็นรหัสคำสั่งเอง ผู้ใช้ App จึงไม่จำเป็นต้องจำรูปแบบรหัสคำสั่ง เพียงแค่ศึกษาคำสั่งการใช้งาน Block และการนำเข้า-ส่งออกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 หน้าเว็บไซต์อย่างเป็นทางการของ MIT App Inventor

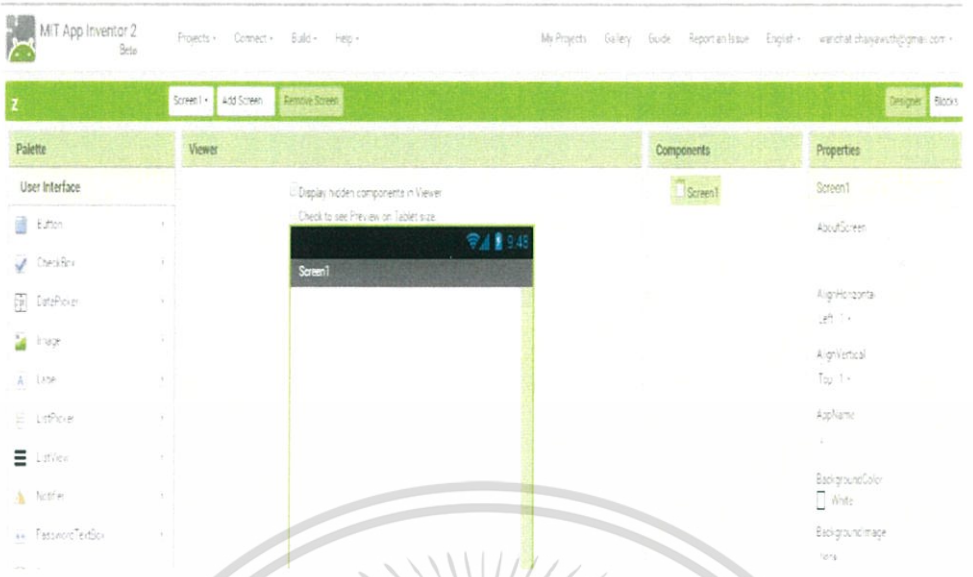
นอกจากนี้ MIT App Inventor สามารถใช้ประกอบการเรียนการสอน การเขียนโปรแกรม โดยใช้หลักการของ Component เพื่อให้นักเรียนหรือนักศึกษาเห็นภาพกว้างของการพัฒนาโปรแกรมเชิง Component หลักการรับและส่งค่าอาร์กิวเมนต์ ผลกระทบจาก Component หนึ่งๆต่อ Component อื่นๆ ใน App และช่วยให้นำความเข้าใจมาสอดแทรกรูปแบบของรหัสคำสั่ง ช่วยให้นักเรียนหรือนักศึกษาจดจำรูปแบบได้ดียิ่งขึ้น



รูปที่ 2.2 การใช้ Icon แทนรหัสคำสั่ง และผลจากการทำงาน

เมื่อเปิดโปรแกรม MIT App Inventor [6] ขึ้นมาจะมีหน้าแรกดังรูปที่ 2.3 โดยจะมีแถบเครื่องมืออยู่ด้านซ้ายมือ และบริเวณตรงกลางจะเป็นหน้าแอพพลิเคชั่นจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



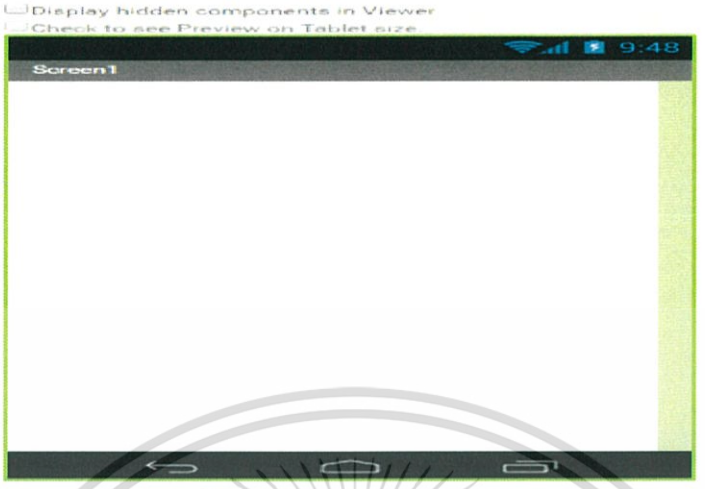
รูปที่ 2.3 หน้าโปรแกรมเริ่มต้นการสร้างแอพพลิเคชั่นที่ใช้งาน

ในการเริ่มต้นสร้างโปรเจค คลิกที่ Project > My projects

รูปที่ 2.4 การสร้าง Projects

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

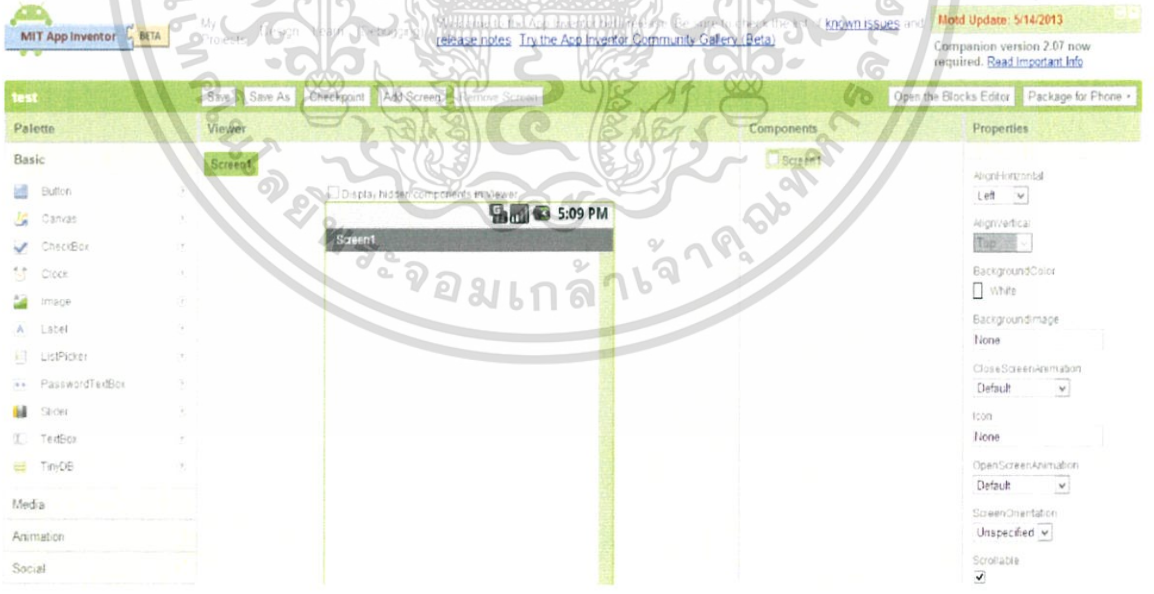
ขณะเริ่มต้นสร้างโปรเจค หน้าแอพพลิเคชั่นจำลองสำหรับออกแบบจะว่างเปล่าดังรูปที่ 2.5



รูปที่ 2.5 หน้าแอพพลิเคชั่นจำลอง

2.1.1 การพัฒนา Apps ด้วย MIT App Inventor

ผู้ใช้สามารถเข้าเว็บไซต์ของ MIT App Inventor แล้วเริ่มพัฒนา Apps บน Android ได้ทันที เนื่องจาก App Inventor จะทำการประมวลผลที่ฝั่ง Server ทำให้เครื่อง Client ไม่จำเป็นต้องลงโปรแกรมใดๆเพิ่มเติม



รูปที่ 2.6 รูปหน้าจอหลักที่ใช้พัฒนาแอพพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIT App Inventor ใช้หลักการของ Component-based Software Development ในรูปแบบของ Visual Programming และใช้หลักการการประมวลผลแบบ Client/Server ทำให้ไม่ต้องติดตั้งโปรแกรมเพิ่มเติมบนเครื่องที่ใช้สร้าง Apps โปรแกรมนี้มีความสะดวกในการใช้งาน เข้าใจง่าย เหมาะกับนักเรียนนักศึกษา หรือผู้ที่เริ่มต้นเขียนโปรแกรมบน Android โดยมีข้อจำกัดคือ Framework ไม่ครอบคลุมขนาดจอภาพของอุปกรณ์ และจะต้องเชื่อมต่อ Internet อยู่ตลอดเวลาที่ใช้งาน

2.2 การสื่อสารผ่านระบบ TCP/UDP

ชั้นสื่อสารนำส่งข้อมูล (Transport Layer) [7] แบ่งเป็นโพรโตคอล 2 ชนิด ชนิดแรกคือ Transmission Control Protocol (TCP) เป็นชนิดที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (Connection-Oriented) ส่งข้อมูลเป็นแบบ Byte Stream โดยไม่มีข้อผิดพลาด ข้อมูลที่มีขนาดใหญ่จะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า Message ซึ่งจะส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ด้านฝ่ายผู้รับจะทำการเรียง Message ต่อกันตามลำดับให้เป็นข้อมูลตัวเดิม TCP [7] สามารถควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทัน โปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) โปรโตคอลชนิดนี้มีการตรวจสอบความถูกต้องของข้อมูล แต่จะไม่มีการแจ้งกลับ จึงนับว่าไม่มีการตรวจสอบข้อมูล แต่วิธีการนี้มีข้อดีคือส่งข้อมูลรวดเร็ว นิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (Client/Server System) ซึ่งสื่อสารแบบถาม-ตอบ (Request/Reply) และยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการสงเสียง (Voice) ผ่านอินเทอร์เน็ต

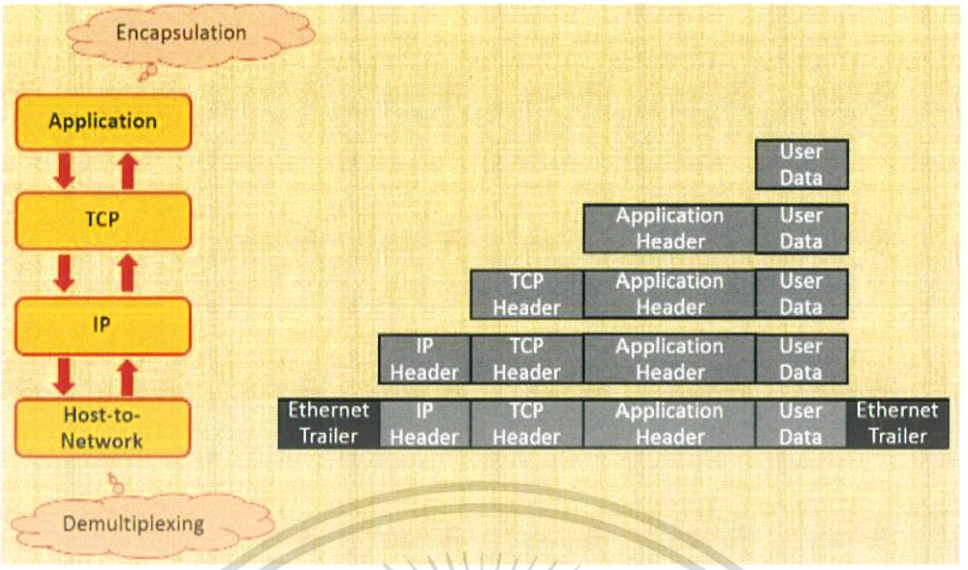
2.2.1 TCP: (Transmission Control Protocol)

เป็นจัดการและควบคุมการรับส่งข้อมูลที่มีความสามารถและรายละเอียดมากกว่า UDP โดยDatagram ของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน มีกลไกควบคุมการรับส่งข้อมูลให้ถูกต้อง (Reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (Connection-Oriented)

2.2.2 UDP: (User Datagram Protocol)

การส่งข้อมูลของ UDP เป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP Datagram โดยจะไม่มีความสัมพันธ์ระหว่าง Datagram และจะไม่มีการตรวจสอบการรับส่งข้อมูลกลไกการตรวจสอบ Checksum ของ UDP เป็นการป้องกันข้อมูลที่อาจถูกแก้ไขหรือมีความผิดพลาดระหว่างการส่งโดยปลายทางจะทราบหากเกิดข้อผิดพลาดขึ้น แต่จะตรวจสอบเพียงฝ่ายเดียวเท่านั้นโดยหากพบว่า Checksum Error จะให้ผู้รับปลายทางทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การสื่อสารแบบ TCP

ข้อแตกต่างระหว่าง TCP กับ UDP [7]

TCP:

มีความน่าเชื่อถือ: ข้อมูลที่ส่งจะถึงผู้รับอย่างแน่นอน TCP จะตรวจสอบว่าข้อมูลที่ส่งไปนั้นถึงผู้รับจริงๆหรือไม่ หากข้อมูลที่ส่งไปไม่ถึงผู้รับ TCP จะทำการส่งข้อมูลนั้นไปให้ใหม่อีกครั้ง

มุ่งเน้นการเชื่อมต่อ : มีการเชื่อมต่อช่องทางการรับส่งข้อมูลก่อนที่จะเริ่มส่ง เพื่อเตรียมความพร้อมระหว่างเครื่องผู้รับและผู้ส่ง เช่น หน่วยความจำและตัวแปรที่ใช้เก็บข้อมูลต่างๆ เพื่อให้การรับส่งข้อมูลระหว่างต้นทางและปลายทางเป็นไปอย่างราบรื่นและมีประสิทธิภาพ

การควบคุมการส่ง: เพื่อป้องกันไม่ให้มีการส่งข้อมูลมากเกินกว่าที่ Buffer ของผู้รับจะรับได้ จึงมีการควบคุมปริมาณข้อมูลที่รับส่งระหว่างต้นทางและปลายทาง

ควบคุมความแออัด : เป็นการควบคุมปริมาณการส่งข้อมูลเช่นเดียวกัน แต่เป็นการควบคุมเพื่อป้องกันไม่ให้ส่งข้อมูลเข้าไปในเครือข่ายที่มีความหนาแน่นของข้อมูลสูงมาก ณ เวลานั้น เพราะอาจเกิดความเสี่ยงที่ข้อมูลจะไปไม่ถึงผู้รับ

TCP เหมาะกับการรับส่งข้อมูลของ Protocol บน Application Layer ที่ต้องการความน่าเชื่อถือสูง เช่น SMTP, Telnet, HTTP, FTP เป็นต้น

UDP:

ไม่น่าเชื่อถือ : ข้อมูลที่ส่งอาจจะไม่ถึงผู้รับ

การเชื่อมต่อไม่ดี : ไม่มีการสร้างช่องทางการรับส่งข้อมูลก่อนเริ่มส่ง

ไม่มีการควบคุมการส่ง : ไม่มีการควบคุมปริมาณการรับส่งข้อมูลระหว่างต้นทางและปลายทาง

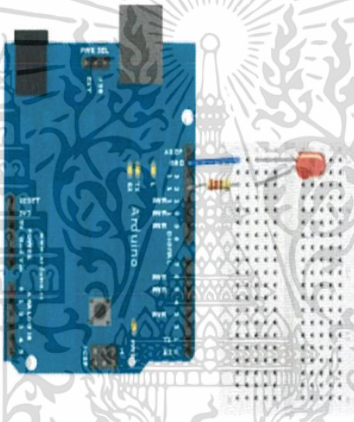
ไม่มีควบคุมความแออัด : ไม่มีการควบคุมปริมาณการส่งข้อมูลในช่วงที่เครือข่ายมีความหนาแน่นข้อมูลสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UDP เหมาะกับการรับส่งข้อมูลของ Protocol บน Application Layer ที่ยินยอมให้ข้อมูลบางส่วนสูญหายได้ โดยข้อมูลภาพรวมไม่ได้รับผลกระทบต่อคุณภาพ เช่น NFS, Streaming Multimedia Internet Telephony, SNMPRI, P DNS เป็นต้น

2.3 Arduino UNO

ในงานวิจัยนี้ใช้บอร์ด Arduino เป็นอุปกรณ์รับคำสั่งจากแอพพลิเคชั่นเพื่อควบคุมการทำงานรีเลย์ในวงจรไฟฟ้า เนื่องจากบอร์ด Arduino [1] เหมาะกับผู้เริ่มต้นศึกษา ใช้งานง่าย สามารถดัดแปลงและพัฒนาตัวบอร์ดหรือโปรแกรมได้ บอร์ด Arduino คือบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ซึ่งพัฒนาเป็นแบบ Open Source สามารถเปิดเผยข้อมูลได้ทั้ง Hardware และ Software บอร์ด Arduino สามารถต่ออุปกรณ์เสริมต่างๆหรือวงจรอิเล็กทรอนิคส์จากภายนอกเข้ามาที่ขา I/O หรือเชื่อมต่อกับบอร์ดเสริมประเภทต่างๆที่เรียกว่า Arduino Shield แล้วเขียนโปรแกรมเพิ่มเติม เช่น Arduino Relay Shield, Arduino Wireless Shield, Arduino XBee Shield เป็นต้น



รูปที่ 2.8 บอร์ด Arduino ต่อกับ LED

รูปที่2.8 แสดงการต่อใช้งานบอร์ด Arduino กับหลอดไฟ LED โดยใช้ตัวต้านทานต่อร่วมด้วยเพื่อลดทอนกระแสในวงจร จากรูปใช้เป็นการทดสอบความถูกต้องวงจรเบื้องต้น Arduino เป็นที่นิยมเนื่องจากง่ายต่อการใช้งาน ไม่ซับซ้อน เหมาะกับผู้เริ่มต้น มีรูปแบบคำสั่งพื้นฐาน Arduino เป็น Open Hardware ทำให้สามารถนำบอร์ดไปใช้งานได้หลายด้าน สามารถพัฒนาโปรแกรมได้บนหลาย OS และราคาไม่แพง โดยในงานวิจัยนี้เลือกใช้ Arduino UNO เพราะเป็น Arduino ที่ถูกออกแบบมาให้ใช้งานง่าย มี Shield ต่างๆที่รองรับหลายชนิด และราคาไม่แพง มีรายละเอียดดังนี้ [3]

ไมโครคอนโทรนเลอร์:	ATmega328
แหล่งจ่ายไฟ:	5V
ไฟเข้า(ที่แนะนำ):	7-12V
ไฟเข้า(ที่จำกัด):	6-20V
ขาดิจิตอลI/O:	14 ขา (6 รองรับเอาต์พุตแบบ PWM)
ขาอะนาล็อกอินพุต:	6 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

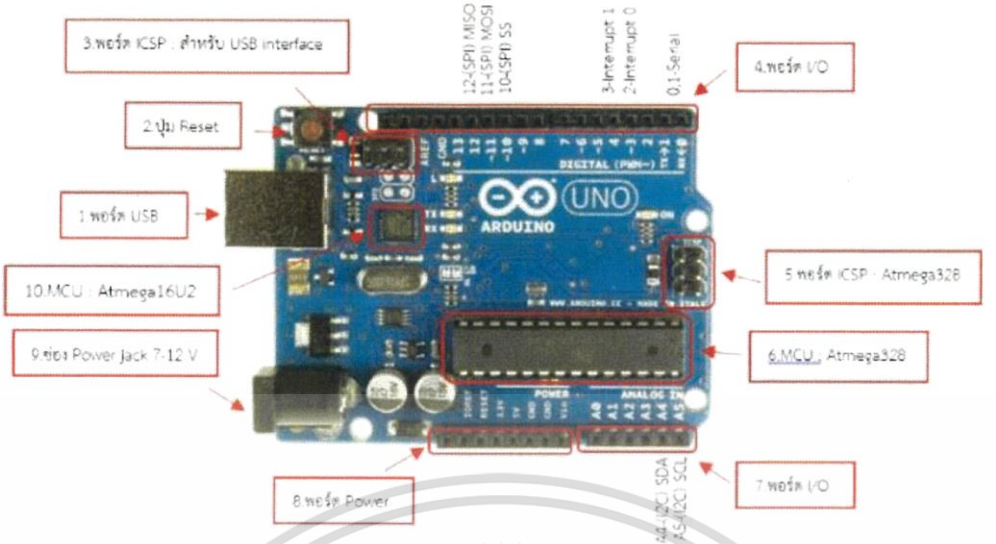
กระแสไฟฟ้าDCต่อขา I/O:	40 mA
กระแสไฟฟ้าออก DC สำหรับขา:	3.3V50 mA
หน่วยความจำแฟลช:	32 KB (ATmega328)
สแตติกแรม:	2 KB (ATmega328)
อิเล็คตริกคอลอีพรอม:	1 KB (ATmega328)
ความเร็วนาฬิกา:	16 MHz

รูปที่ 2.9 Arduino UNO

ส่วนประกอบของ Arduino UNO [3]

1) USB Port: ใช้เชื่อมต่อกับคอมพิวเตอร์ เพื่ออัพโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับตัวบอร์ด
2) Reset Button: คือปุ่ม Reset กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3) ICSP Port: เป็นพอร์ตที่ใช้โปรแกรม Visual Com Port บน Atmega16U2
4) I/O Port: ดิจิตอล I/O ตั้งแต่ขา D0 ถึง D13
5) ICSP Port: เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6) MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด
7) I/O Port: เป็น Digital I/O สามารถเปลี่ยนเป็นช่องรับสัญญาณอนาล็อกได้ ตั้งแต่ขา A0-A5
8) Power Port: เป็นไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก โดยมีขาไฟเลี้ยง +3.3 V, +5V, GND, V_{in}
9) Power Jack: เป็นขารับไฟจาก Adapter แรงดันอยู่ระหว่าง 7-12 V
10) MCU ของ Atmega16U2: เป็นไมโครคอนโทรลเลอร์อีกตัวหนึ่งที่ทำหน้าที่เป็น USB to Serial Converter โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



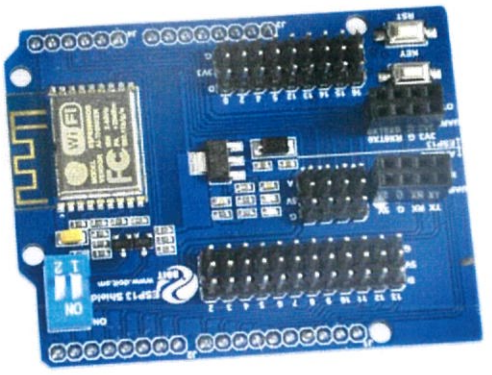
รูปที่ 2.10 ส่วนประกอบต่างๆเกี่ยวกับ Arduino UNO

2.4 ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino

คุณสมบัติของ ESP8266 ESP-13 Wi-Fi Web Sever Shield [2]

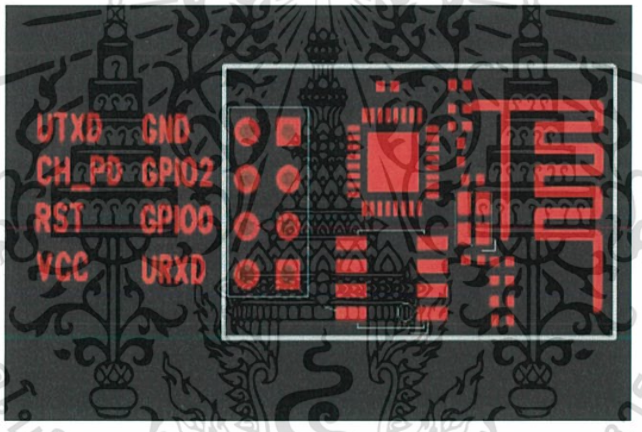
1) โมดูล Wi-Fi ชิปเกรดอุตสาหกรรม ESP8266 ซึ่งเป็น ESP-12E Shield โลหะที่สามารถป้องกันการรบกวนได้ดี
2) Shield สามารถใช้งานได้กับ Arduino Uno, Mega2560 และบอร์ดควบคุมอื่น ๆ
3) ชิปแปลงแรงดันไฟฟ้าที่ใช้คือ 3.3V (ESP8266) และ 5V (Arduino)
4) สวิทช์แบบ Dual DIP จะใช้สำหรับพอร์ตอนุกรม เพื่อให้ Module Shield สามารถใช้งานเดี่ยวๆได้
5) Serial ข้อมูลจะถูกส่งไปยังอุปกรณ์ Wi-Fi โดยโปรแกรม Arduino ไม่ต้องตั้งค่าใดๆ
6) เว็บเซิร์ฟเวอร์มีการพัฒนาเพื่อกำหนดค่าพารามิเตอร์ Wi-Fi และพารามิเตอร์พอร์ตอนุกรม
7) Module Shield สามารถนำมาใช้เป็นบอร์ดของ ESP8266 ได้อย่างอิสระ ตัวอย่างเช่น การดาวน์โหลดเฟิร์มแวร์คำสั่งของ Official AT หรือ เฟิร์มแวร์ Nod MCU Open Source จะสามารถใช้ได้
8) Module Shield สามารถใช้แบบสแตนด์อโลนได้ สำหรับ Arduino Uno

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ESP8266 ESP-13 Wi-Fi Web Sever Shield

โมดูลมีการจัดขาตามรูปด้านล่าง



รูปที่ 2.12 ขาของ ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino

การต่อวงจรจากบอร์ด Wi-Fi Module ESP8266 ไปบอร์ด Arduino

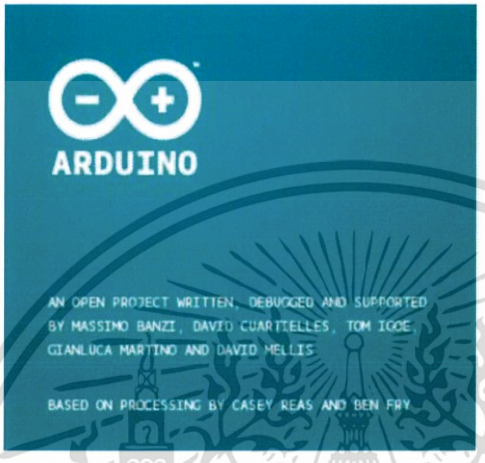
1) Vcc-3.3V
2) Gnd-Gnd
3) CH_PD-3.3V
4) TX-TX(ขา 1)
5) Rx-RX(ขา 0)

2.5 การเขียนโปรแกรมและการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการเขียนคำสั่งควบคุมการทำงานในการเปิด-ปิดเครื่องใช้ไฟฟ้านั้นคือโปรแกรม Arduino [1] โดยใช้การเขียนคำสั่งการทำงานให้กับบอร์ด Arduino Uno และ ESP-13 Wi-Fi Web Sever Shield

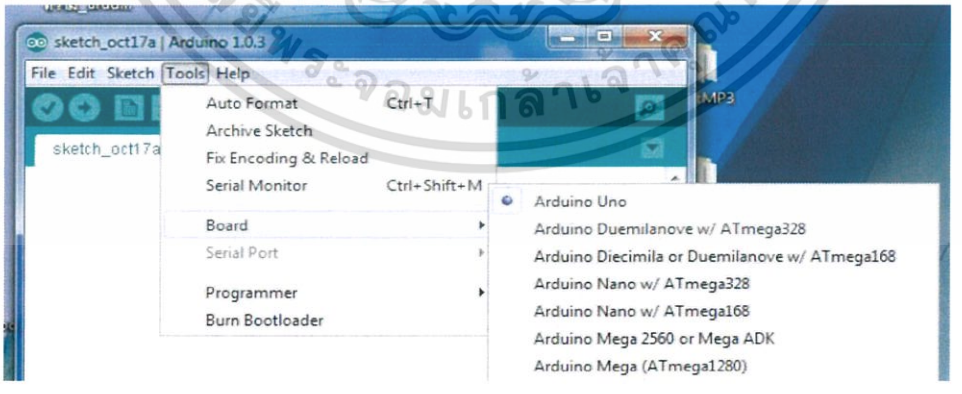
Arduino คือโปรแกรมเขียนคำสั่งเพื่อใช้งานกับ Arduino ทุกรุ่น โดยภายในจะมีเครื่องมือหรือคำสั่งที่จะใช้ติดต่อกับ Arduino โปรแกรมนี้เป็นโปรแกรมแบบเปิดที่สามารถนำไปใช้งานได้ฟรีๆ อีกทั้งและยังตัวอย่างให้ทดสอบกับเซ็นเซอร์ต่างๆอีกด้วย



รูปที่ 2.13 โปรแกรม Arduino

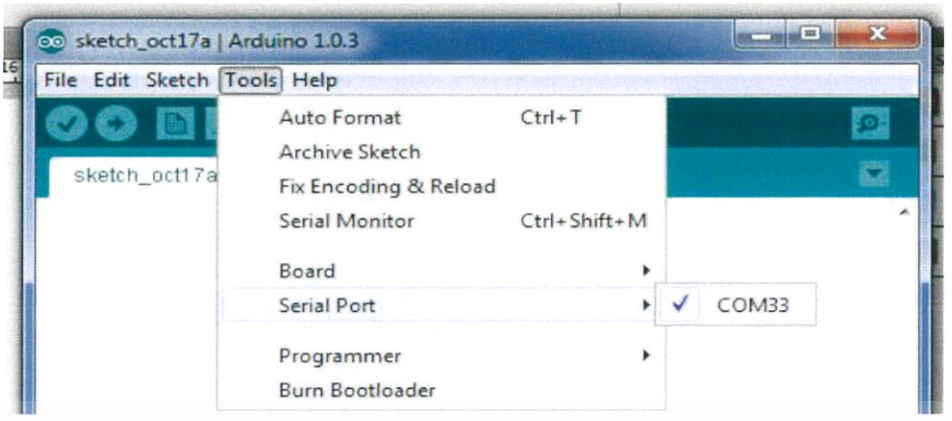
วิธีการใช้โปรแกรมเบื้องต้น

1) เขียนโปรแกรมโดยใช้ Arduino [1] ซึ่งสามารถดาวน์โหลดได้จากArduino.cc/en/main
2) เมื่อเขียนโค้ดโปรแกรมเสร็จเรียบร้อยให้เลือกรุ่นบอร์ด Arduinoที่ใช้และหมายเลข Com Port



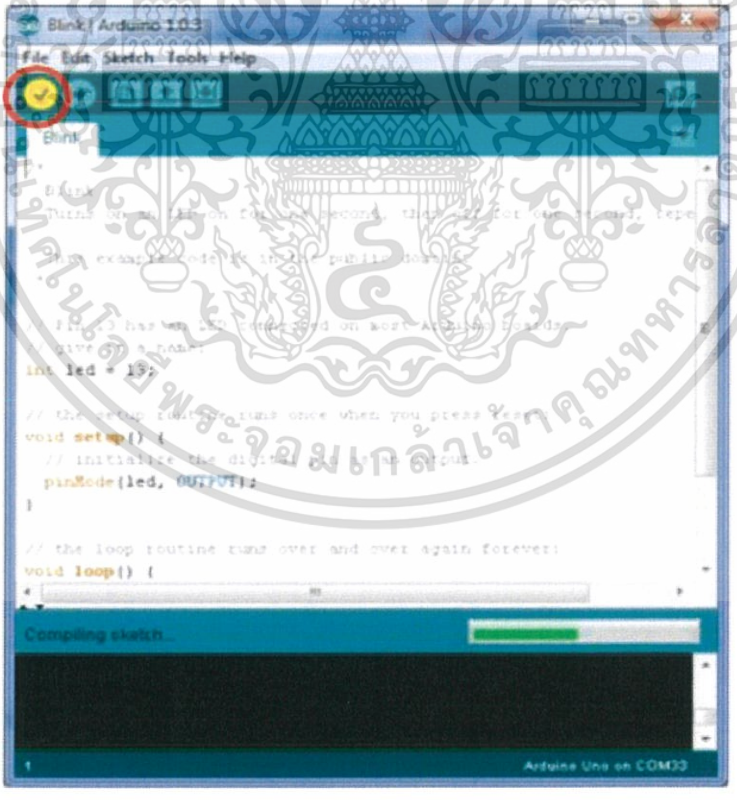
รูปที่ 2.14 เลือกรุ่นบอร์ด Arduino ที่ต้องการ Upload

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



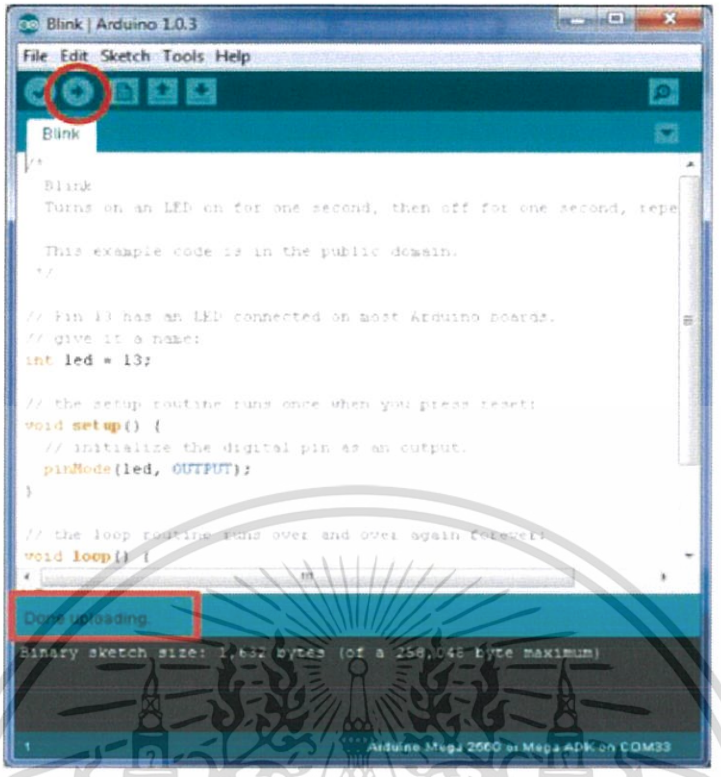
รูปที่ 2.15 เลือกหมายเลข Com Port ของบอร์ด

3) ตรวจสอบความถูกต้องโดยกดปุ่ม Verify และ Compile โค้ดจากนั้นให้กดปุ่ม Upload เพื่ออัพโหลดโค้ดไปยังบอร์ด Arduino โดยใช้สาย USB เมื่ออัพโหลดเสร็จเรียบร้อย จะมีข้อความแถบข้างล่างแสดงขึ้นว่า “Done uploading” และบอร์ดจะเริ่มทำงานตามโปรแกรมที่เขียนไว้ได้ทันที



รูปที่ 2.16 ตรวจสอบความถูกต้องโดยกดปุ่ม Verify และ Compile โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 ข้อความแสดง “Done uploading”

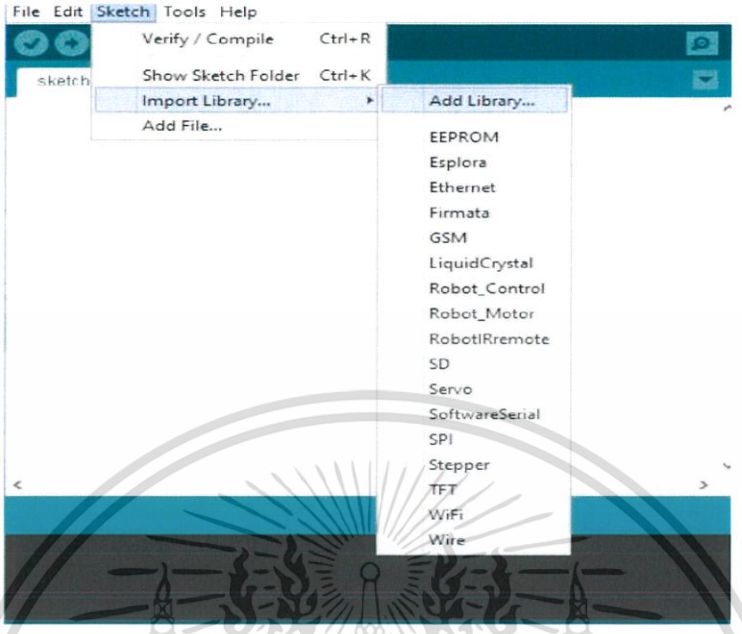
ตัวอย่างหน้าโปรแกรม

เมื่อเปิดโปรแกรม Arduino ขึ้นมา หน้าแรกของโปรแกรมจะว่างเปล่าดังรูปที่ 2.18

รูปที่ 2.18 หน้าแรกของโปรแกรม

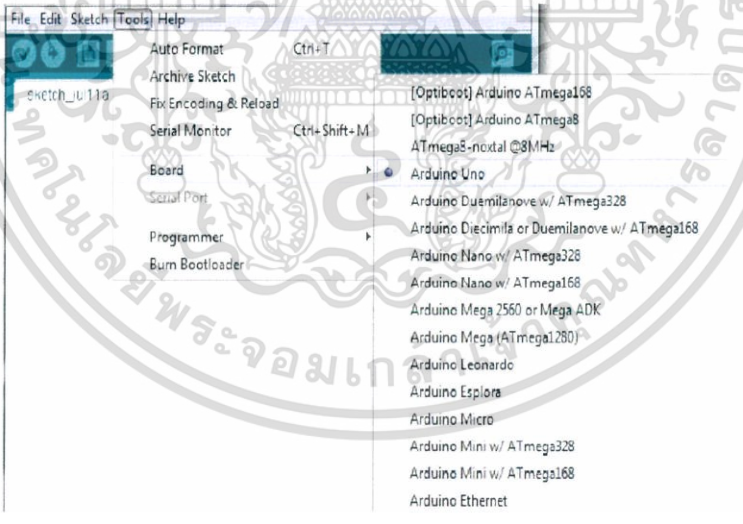
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการ Add Library โดยคลิกที่ Sketch > Import Library > Add Library ดังรูปที่ 2.19



รูปที่ 2.19 การ Add Library

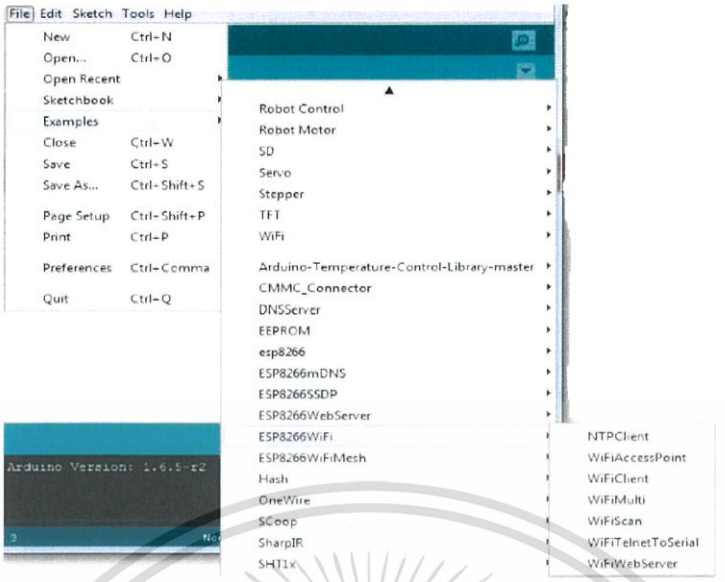
ทำการเลือกชนิดของบอร์ด โดยคลิกที่ Too > Board > Arduino Uno ดังรูปที่ 2.20



รูปที่ 2.20 การเลือกชนิดของบอร์ด

ทำการเลือก Port Wi-Fi โดยคลิกที่ File > Examples > ESP8266WiFi > NTPClient ดังรูปที่ 2.21

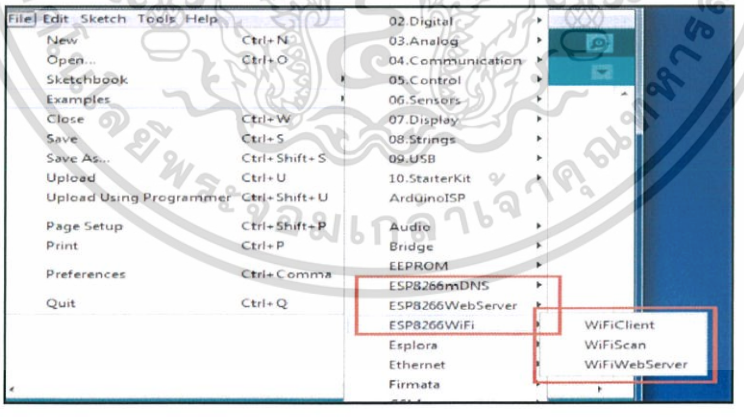
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 การเลือก Port Wi-Fi

การเริ่มต้นการใช้งาน ESP8266 [4] บนบอร์ด Arduino Uno

1) ดาวน์โหลดโค้ด Arduino Uno ที่ใช้งานกับ ESP8266 ได้จาก Link: https://github.com/esp8266/arduino
2) ทำการ Extract File ที่ดาวน์โหลด
3) เปิดโปรแกรมขึ้นมา
4) ที่ Menu File -> Examples แล้วจะเห็นตัวอย่าง การใช้งานที่มากับ Uno

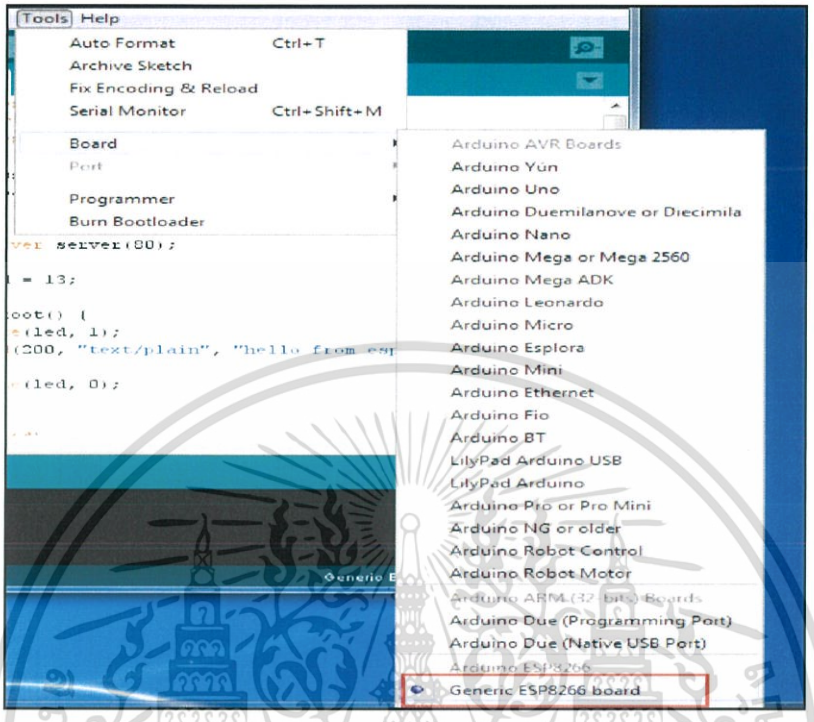


รูปที่ 2.22 เลือก ESP8266 Wi-Fi

5) คลิกที่ File -> Example -> ESP8266WebServer -> HelloServer เพื่อทดสอบ
6) ทำการตั้งค่า SSID (Service Set Identifier) ซึ่งเป็นตัวระบุเฉพาะที่ป้องกันการแทรกแซงเครือข่ายแบบไร้สาย และตั้งค่า Password

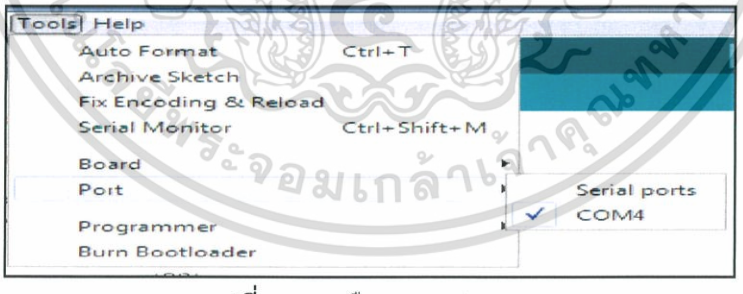
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) เลือก Target Board ESP8266 โดยคลิกที่ Menu Tools -> Boards -> Generic ESP8266 Board



รูปที่ 2.23 เลือก Target Board ESP8266

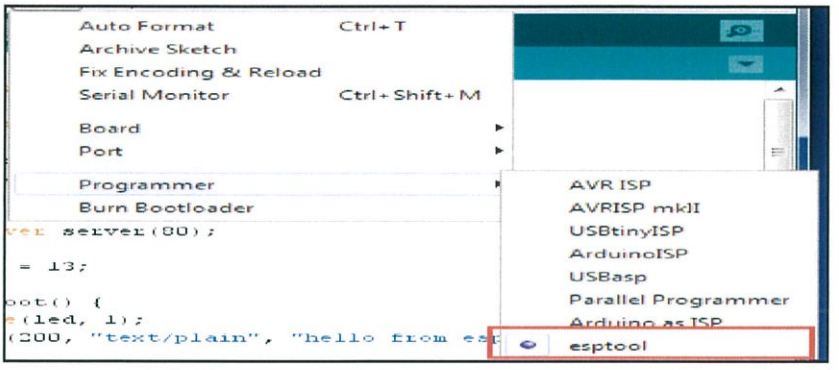
8) เลือก Serial Port ที่ต่อกับ ESP8266 โดยคลิกที่ Tools -> Port -> COM...



รูปที่ 2.24 เลือก Serial Port

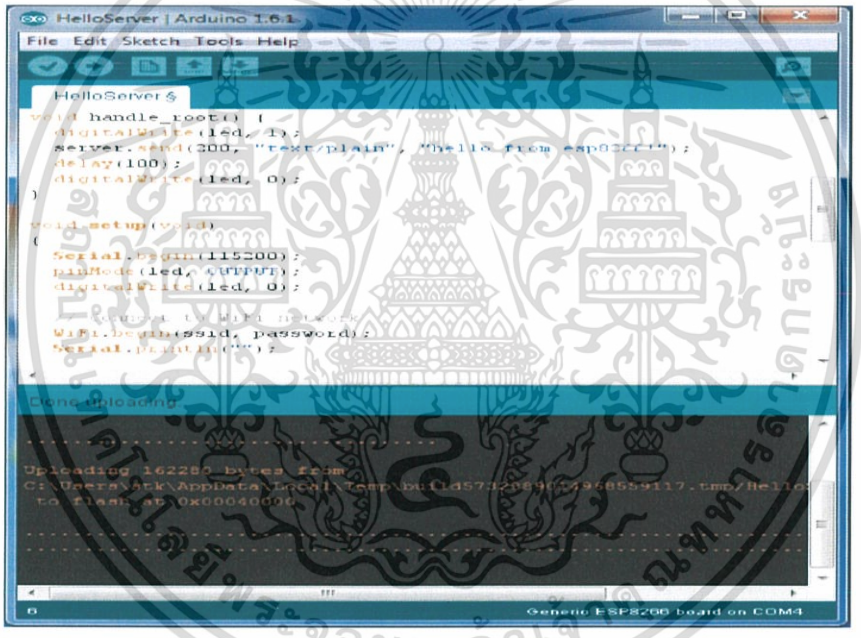
9) เลือก Programmer ไปที่ ESP8266 โดยคลิกที่ Tools -> Programmer -> esptool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น. อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 เลือก Programmer ไปที่ ESP8266

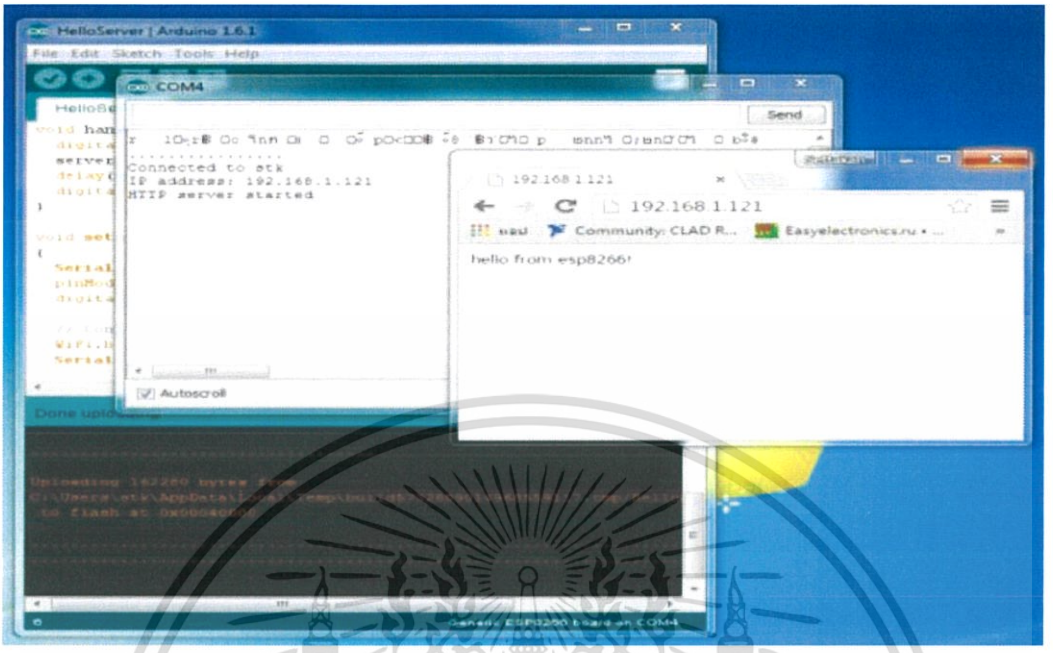
10) คลิก Compile และ Program ลงไปบน ESP8266



รูปที่ 2.26 การ Compile โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11) ทดสอบการใช้งาน Webserver



รูปที่ 2.27 ทดสอบการใช้งาน Webserver

2.6 การตั้งค่าบนบอร์ด ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino

การเตรียมงาน: การตั้งค่าโหมดการทำงานอินเตอร์เน็ตไร้สาย [4]

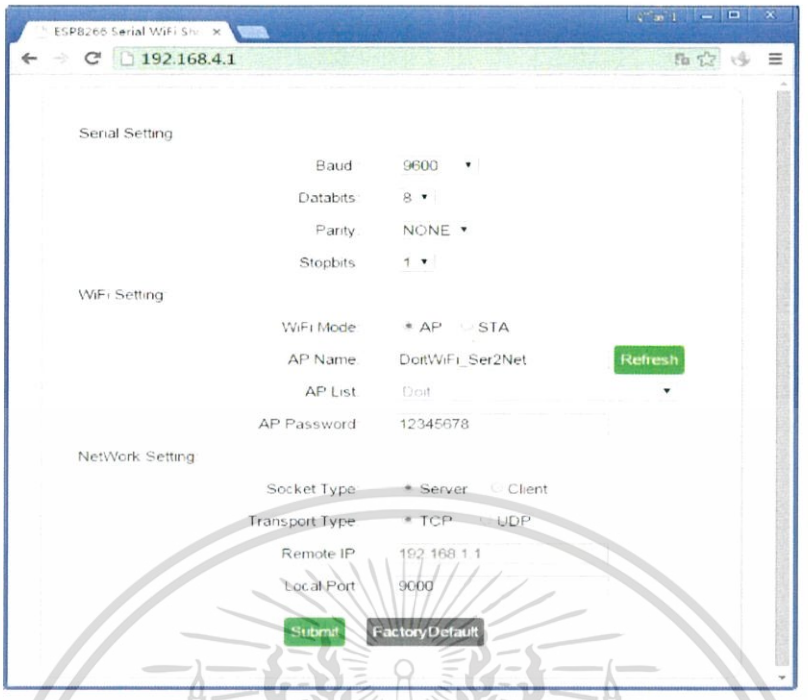
สำหรับ Serial WiFi Shield จะถูกตั้งค่าโหมดการทำงานเริ่มต้นจากโรงงาน การกำหนดค่าการสื่อสารจึงควรจะทำเป็นอันดับแรก

(หมายเหตุ : ถ้า Shiled ได้รับการกำหนดค่าให้ทำงานในโหมดการทำงานอื่นๆ Shiled โมดูลสามารถเข้าสู่โหมดการตั้งค่าผ่านวิธีต่อไปนี้ หนึ่งคือการกดที่ปุ่ม KEY นานกว่า 1 วินาที ส่วนอีกวิธีหนึ่งคือการส่งค่า "!@!" เพื่อให้ Serial WiFi Shield ตั้งค่าผ่านพอร์ตอนุกรม)

ขั้นตอนที่ 1: เปิดสวิทช์ Serial WiFi Shield และหาสัญญาณ WiFi จะมี SSID คือ "DoitWiFi_Config" และรหัสผ่านคือ "12345678" ซึ่งเป็นรหัสที่โรงงานกำหนด

ขั้นตอนที่ 2: เปิดเบราว์เซอร์และป้อน IP address WiFi: 192.168.4.1 ซึ่งเป็น IP ของ ESP8266 ESP-13 Wi-Fi Web Sever Shield จากโรงงาน จากนั้นหน้าต่างการกำหนดค่าจะแสดงดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 หน้าต่างการกำหนดค่า

ติดตั้งเว็บเซิร์ฟเวอร์ใน Serial WiFi Shield ที่สนับสนุนการกำหนดค่า Serial ของพารามิเตอร์ รวมถึงการกำหนดค่าพารามิเตอร์เครือข่าย และนอกจากนี้ยังสนับสนุนการสแกน AP แบบอัตโนมัติหรือ Manual

ขั้นตอนที่ 3: กดปุ่ม "Submit"หลังจากการส่ง Serial Wifi Shield จะรีบูตโดยอัตโนมัติในโหมด AP และกำหนดค่าเป็น: พารามิเตอร์ซีเรียลพอร์ต 9600, N, 8, 1
โหมดการทำงาน WiFi: AP;
SSID: "DoitWiFi_Ser2Net";
รหัสผ่าน: "12345678"
เซิร์ฟเวอร์ TCP และพอร์ตคือ 9000

ตัวอย่างที่ 1: Arduino กับการสื่อสาร WiFi

ขั้นตอนที่ 1: เสียบWiFi Serial เข้าไปในบอร์ด Arduino Uno สวิทช์ DIPแบบ Dual-Ports จะเปลี่ยนไปยังตำแหน่ง "OFF" ซึ่งจะยกเลิกการเชื่อมต่อซีเรียลพอร์ตภายใน ESP8266 กับ Arduino (หมายเหตุ: เมื่อมีการดาวน์โหลดโปรแกรม Arduino กับ IDE ซีเรียลพอร์ตของ Arduino ไม่ควรเปิดใช้งานเพราะ Arduino Uno มีเพียงซีเรียลพอร์ตเดียวสำหรับการดาวน์โหลดโปรแกรม จึงต้องเปลี่ยนสวิทช์แบบ Dual-Ports ไปอยู่ในตำแหน่ง "OFF" เพื่อตัดการเชื่อมต่อของ ESP8266) [4]

ขั้นที่ 2: โปรแกรม Arduino Uno และรหัสตัวอย่างจะแสดงให้เห็นตามภาพด้านล่าง:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

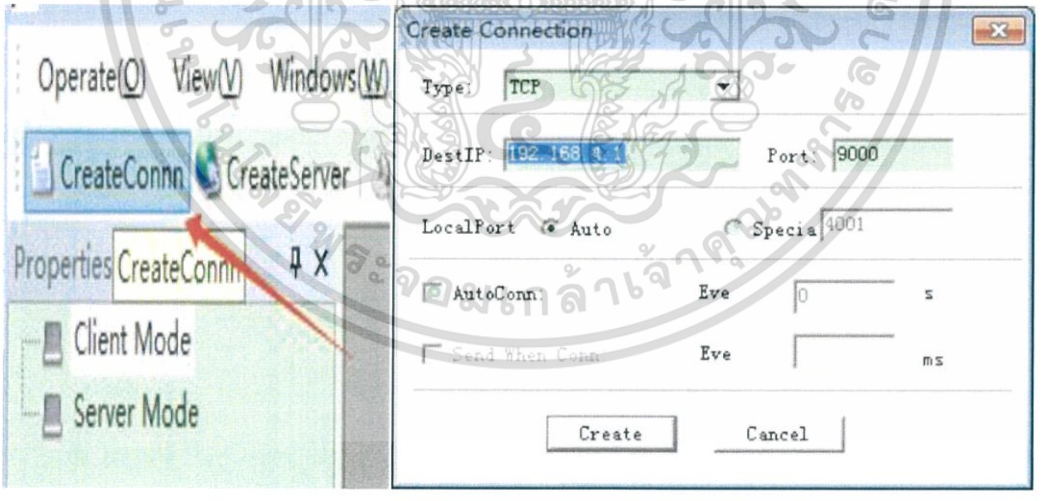
```
1.  void setup()
2.  {
3.    Serial.begin(9600);
4.  }
5.  void loop()
6.  {
7.     delay(1000);
8.     Serial.println("hello ESP8266 WiFi"); //output the serial data
9.  }
```

รูปที่ 2.29 โปรแกรม Arduino และรหัสตัวอย่าง

ขั้นที่ 3: เปลี่ยนสวิทช์ Dual-Port ไปที่ตำแหน่ง"ON" ตอนนี้ ESP8266 จะเชื่อมต่อกับ Arduino Uno

ขั้นที่ 4: หาสัญญาณ WiFi "DoitWiFi_Ser2Net" กับเครื่องคอมพิวเตอร์หรือโทรศัพท์มือถือ รหัสผ่าน WiFi คือ "12345678"

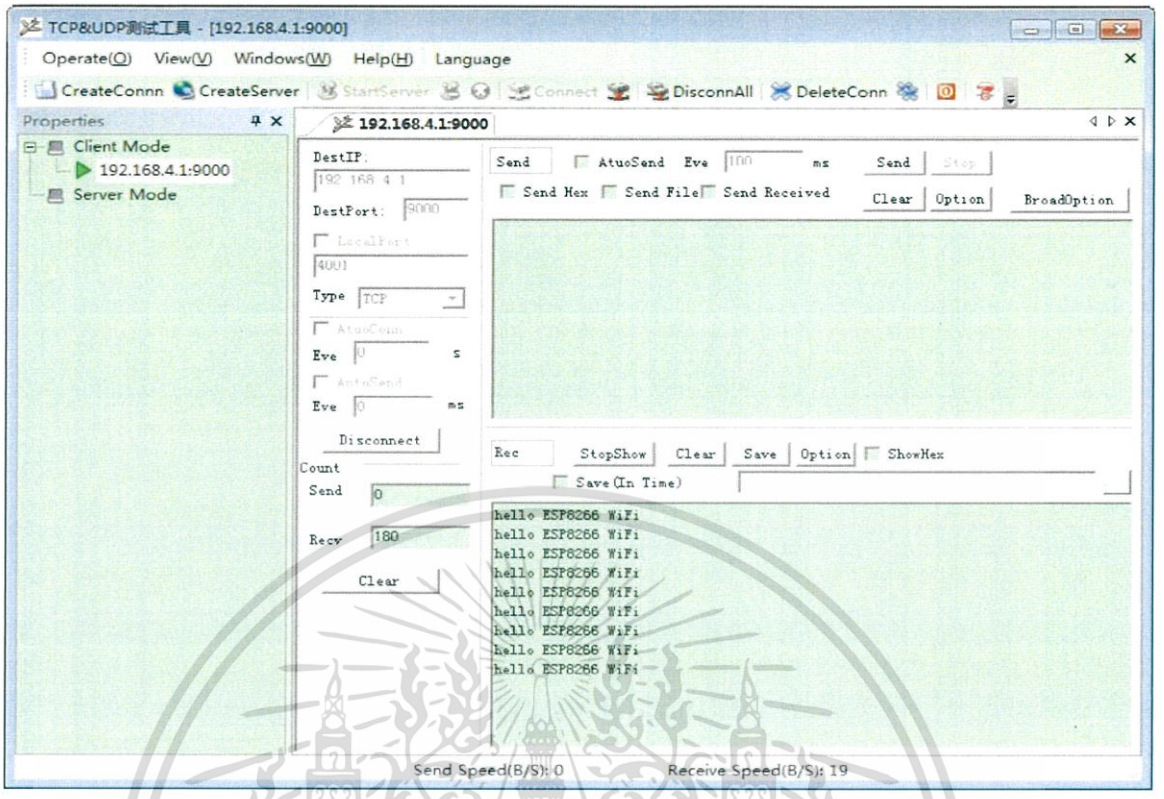
ขั้นที่5: รันซอฟท์แวร์TCP/UDP Debugging (ลิงค์ดาวน์โหลดซอฟท์แวร์: http://bbs.doit.am/forum.php?mod=viewthread&tid=174&page=1&extra=#pid206) สร้างการเชื่อมต่อ TCP ของ Client ที่อยู่ IP ปลายทางคือ 192.168.4.1 หมายเลขพอร์ตคือ 9000.Local พอร์ตมีการกำหนดให้ "อัตโนมัติ"



รูปที่ 2.30 การเชื่อมต่อในรูปแบบ TCP กับ IP ที่อยู่

เมื่อการเชื่อมต่อสำเร็จ ซอฟแวร์จะแสดงข้อมูลที่มาจาก Serial WiFi Shield ทุกหนึ่งวินาที ดังนั้น Arduino สามารถส่งข้อมูลแบบอนุกรมในเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 การส่งข้อมูลเมื่อเชื่อมต่อสำเร็จ

ตัวอย่างที่ 2: การสื่อสารกับเซิร์ฟเวอร์ระยะไกล
ในที่นี้ Doit WiFi สาธารณะซึ่งเป็นเครื่องมือเครือข่าย TCP ฟรีจะถูกนำมาใช้กับตัวอย่างนี้
ขั้นตอนที่ 1: รับที่อยู่ IP ชั่วคราวและพอร์ตจากเซิร์ฟเวอร์ของ TCP ที่อยู่ IP ที่ได้ในตัวอย่างคือ "115.29.109.104" และหมายเลขพอร์ตคือ "6533"

รูปที่ 2.32 การรับที่อยู่ IP ชั่วคราว

(หมายเหตุ: ที่อยู่ IP และหมายเลขพอร์ตสามารถหาได้จาก http://tcp.doit.am)
ขั้นตอนที่ 2: เข้าสู่โหมดการตั้งค่า หลังจากเปิด Serial WiFi Shield กดปุ่ม KEY นานกว่าหนึ่งวินาที ทำการเชื่อมต่อ SSID ที่ชื่อ "DoitWiFi_Config" ด้วย Passward "12345678" จากนั้นใส่ "192.168.4.1" ในเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3: กำหนดค่า Serial WiFi Shield ไปยังโหนด (โหมด STA) เพื่อเชื่อมต่อกับเราเตอร์ไร้สาย "WiFi Mode": STA;
เราเตอร์ไร้สายที่ใช้ในตัวอย่างมีชื่อว่า "MIFI_A6_cd1c" รหัสผ่านคือ "mifi66666666" เว็บเซิร์ฟเวอร์จะ Refresh รายชื่อ AP ของ ESP8266 โดยอัตโนมัติเมื่อมีการโหลดหน้าเว็บ หากเลือก "STA" ปุ่ม "Refresh" จะถูกเปิดใช้งานเพื่อฟื้นฟูผลการสแกน WiFi

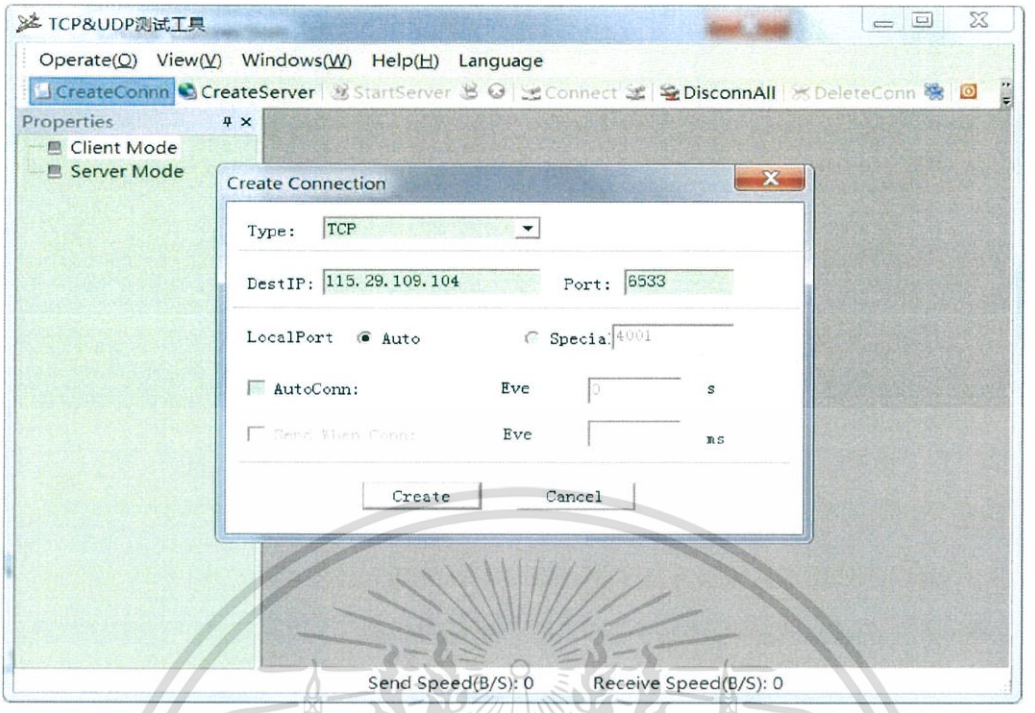
รูปที่ 2.33 การตั้งชื่อและ Password ของ Wi-Fi

ขั้นตอนที่ 4: กำหนดเครือข่ายพารามิเตอร์
ประเภทซ็อกเก็ต: " Client"
ประเภทของการขนส่ง: "TCP"
IP ระยะไกล: "115.28.109.104"
พอร์ตระยะไกล: "6533"

ขั้นตอนที่ 5: กดปุ่ม "ส่ง"

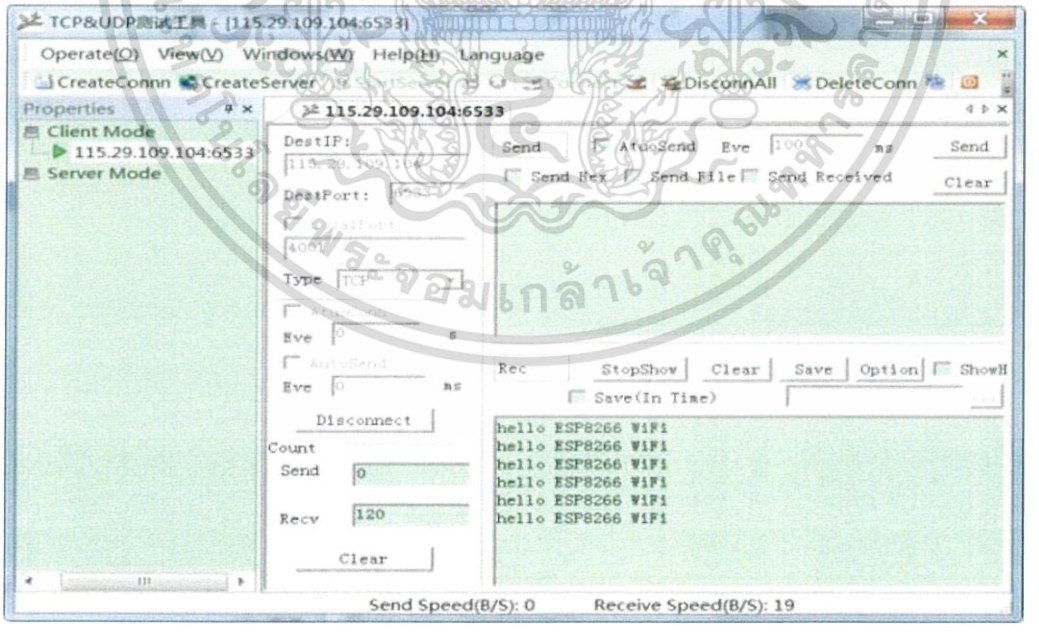
ขั้นตอนที่ 6: การเรียกใช้ซอฟแวร์การแก้ปัญหา TCP/UDP สร้างการเชื่อมต่อ TCP Client ที่อยู่ของ IP ปลายทางคือ 115.29.109.104 หมายเลขพอร์ตคือ 6533. Local พอร์ตมีการกำหนดให้ "อัตโนมัติ"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 การเชื่อมต่อในรูปแบบ TCP กับ IP ที่อยู่ในรูปแบบอัตโนมัติ

ขั้นตอนที่ 7: ซอฟแวร์จะแสดงข้อมูลการขนส่งจาก Serial WiFi shield ผ่านเราเตอร์ไร้สาย ดังนั้น Arduino สามารถส่งข้อมูลแบบอนุกรมไปยังเครือข่ายในโหมด STA

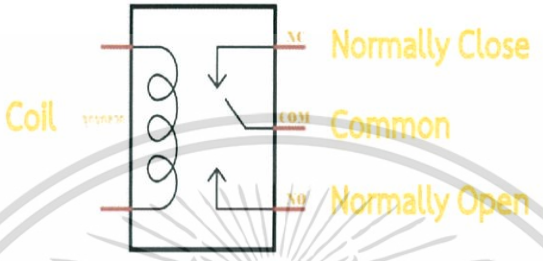


รูปที่ 2.35 การส่งข้อมูลเมื่อเชื่อมต่อสำเร็จ

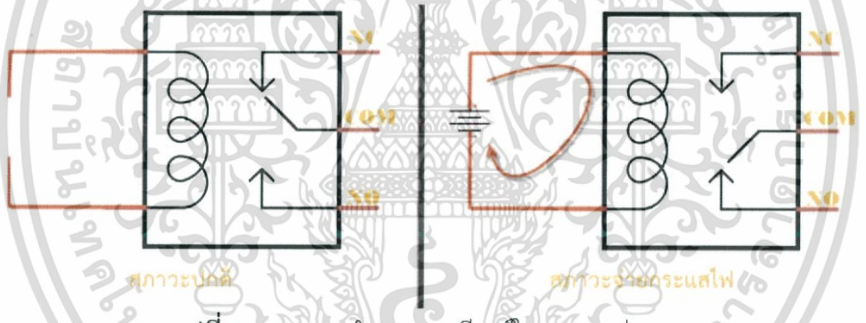
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Relay 8 ช่อง Module Board 5V for Arduino

รีเลย์ [5] เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็กชนิดหนึ่ง มีทำหน้าที่ตัดต่อวงจรซึ่งคล้ายสวิตช์อิเล็กทรอนิกส์ หลักการทำงานคือการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัส รีเลย์สามารถนำไปประยุกต์ใช้ในวงจรต่างๆได้มากมาย รีเลย์มีหลายประเภทหลายขนาด แต่ละขนาดก็จะมีการใช้งานที่แตกต่างกันออกไป เช่น รีเลย์ขนาดเล็กจะใช้กับงานจำพวกอิเล็กทรอนิกส์ทั่วไป แต่รีเลย์ขนาดใหญ่จะใช้ในงานจำพวกไฟฟ้าแรงสูง เป็นต้น



รูปที่ 2.36 สัญลักษณ์ในวงจรไฟฟ้าของรีเลย์



รูปที่ 2.37 การทำงานของรีเลย์ในสภาวะต่างๆ

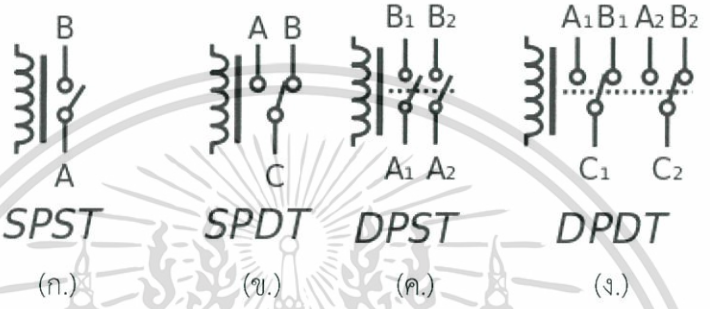
ภายในรีเลย์จะประกอบด้วยส่วนสำคัญ 2 ส่วน คือ ส่วนขดลวด (Coil) **และส่วนหน้าสัมผัส** (Contact)

1. ส่วนขดลวด (Coil) มีหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แกนโลหะ หลักการทำงานคือเมื่อขดลวดได้รับแรงดันไฟฟ้า จะเกิดสนามแม่เหล็กไฟฟ้า ทำให้แกนโลหะด้านในไปผลักให้ส่วนหน้าสัมผัสต่อกัน
2. ส่วนของหน้าสัมผัส (Contact) มีหน้าที่คล้ายกับสวิตช์จ่ายกระแสไฟให้กับอุปกรณ์
    a. **หน้าสัมผัส NC (Normally Close)** ความหมายคือ ปกติปิด ในสภาวะปกติหน้าสัมผัสนี้จะต่อเข้ากับขา COM แต่เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดหน้าสัมผัสจะลอยหรือไม่สัมผัสกัน นิยมต่อเข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา
    b. **หน้าสัมผัส NO (Normally Open)** ความหมายคือ ปกติเปิด ในสภาวะปกติหน้าสัมผัสนี้จะลอยอยู่ แต่เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดหน้าสัมผัสจะเชื่อมต่อกัน นิยมต่อเข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

c. **ขา COM (Common)** คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ ใช้งานร่วมระหว่าง NC และ NO ขึ้นอยู่กับกระแสไฟฟ้าที่ไหลผ่านขดลวด

มีการแยกประเภทสวิตช์ตามจำนวนของ Pole และจำนวนของ Throw โดยจำนวน Pole คือจำนวนของขา COM หรือจำนวนวงจรที่ทำการควบคุมการเปิด-ปิด ส่วนจำนวน Throw คือจำนวนของตัวเลือกของ Pole เช่น SPST (Single Pole Single Throw) ซึ่งจะสามารถเลือกได้เพียงอย่างเดียวโดยจะเป็น NO หรือ NC แต่ถ้าเป็น SPDT (Single Pole Double Throw) จะมีหนึ่งคู่เป็น NO และอีกหนึ่งคู่เป็น NC เสมอ ดังภาพด้านล่าง



รูปที่ 2.38 จำนวนของ Pole และจำนวน Throw ของรีเลย์
(ก.) SPST คือ Single Pole Single Throw
(ข.) DPST คือ Double Pole Single Throw
(ค.) SPDT คือ Single Pole Double Throw
(ง.) DPDT คือ Double Pole Double Throw

หน้าสัมผัสแบบ A (Form A) คือหน้าสัมผัสของรีเลย์แบบปกติเปิด ถ้าเป็นแบบ SPST จะเขียนสัญลักษณ์ได้ดังนี้

รูปที่ 2.39 หน้าสัมผัสแบบ A

หน้าสัมผัสแบบ B (Form B) คือหน้าสัมผัสของรีเลย์แบบปกติปิด ถ้าเป็นแบบ SPST จะเขียนสัญลักษณ์ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

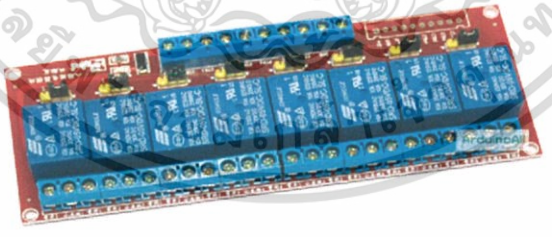
รูปที่ 2.40 หน้าสัมผัสแบบ B

หน้าสัมผัสแบบ C (Form C) หรือเรียกว่า "Break, Make, Transfer" เป็นหน้าสัมผัสแบบ SPDT จะเขียนสัญลักษณ์ได้ดังนี้

รูปที่ 2.41 หน้าสัมผัสแบบ C

หน้าสัมผัสแบบ C จะมี 3 ขา ถ้ารีเลย์ยังไม่ทำงาน หน้าสัมผัส 1 และ 2 จะต่อกันอยู่ แต่ถ้ารีเลย์มีการทำงาน หน้าสัมผัส 1 จะมาต่อกับหน้าสัมผัส 3 แทน

รีเลย์ที่นำมาใช้งานคือ Relay 8 Channel 5V relay Active High/Low Relay Module Board for Arduino [5] เป็นบอร์ดรีเลย์แบบแยกอิสระ 8 ช่อง มีไฟ LED แสดงผลการทำงาน ใช้แรงดันไฟฟ้าขนาด 5 โวลต์ โมดูลนี้สามารถทำงานได้ทั้งแบบ Active High โดยเมื่อป้อนไฟ 3-5V หรือสัญญาณ 1 ไป บอร์ดรีเลย์จะทำงาน หรือแบบ Active Low คือเมื่อป้อนไฟ 0V หรือสัญญาณ 0 ไป บอร์ดจะทำงาน โดยเซตได้ที่จัมเปอร์ของรีเลย์แต่ละช่อง โดยช่องแต่ละ terminal สามารถต่อสายไฟ



รูปที่ 2.42 Relay 8 Channel 5V relay Active High/Low Relay Module Board for Arduino

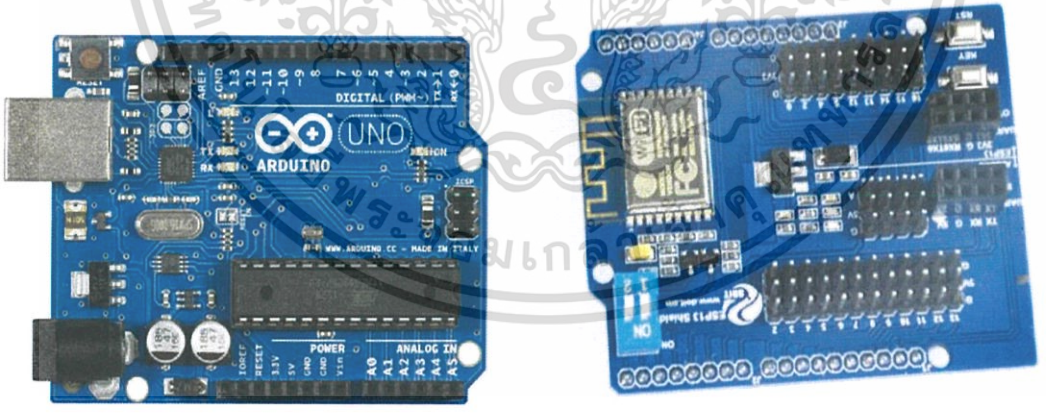
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การดำเนินงานและโปรแกรมที่พัฒนา

สำหรับการดำเนินงานในงานวิจัยนี้มีส่วนสำคัญ 3 ส่วนคือ ส่วนที่หนึ่งซึ่งเป็นส่วนของแอพพลิเคชั่น มีการดำเนินงานเกี่ยวกับการเขียนแอพพลิเคชั่นจากโปรแกรม MIT APP Inventor และการรับคำสั่งเสียงจากผู้ใช้งานโดยใช้ Google Voice เป็นดาต้าเบสตรวจสอบความถูกต้องของคำสั่งก่อนส่งข้อมูลผ่านเครือข่ายอินเตอร์เน็ตให้กับ Arduino ส่วนที่สองคือส่วนของไมโครคอนโทรลเลอร์ จะมีการดำเนินงานด้านการเลือกใช้ Arduino และ Wi-Fi Web Sever Shield ให้เหมาะสมกับการใช้งานและการเขียนโปรแกรมคำสั่งการทำงานบน Arduino เพื่อควบคุมการทำงานของรีเลย์ และส่วนสุดท้ายคือส่วนของระบบวงจรไฟฟ้า เป็นการดำเนินงานด้านวงจรไฟฟ้าขั้นพื้นฐานและอุปกรณ์ไฟฟ้าต่างๆที่นำมาใช้

3.1 การดำเนินงานและการพัฒนาในส่วนที่เกี่ยวกับ Arduino

การดำเนินงานในส่วนนี้เราจะเลือกใช้ Arduino UNO และ ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino เพราะทั้ง Arduino UNO และ ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino ใช้งานง่ายเหมาะสำหรับในการนำมาใช้งานในการพัฒนาการทำงานรูปแบบใหม่และรองรับการทำงานระบบสื่อสารไร้สายที่ปัจจุบันกำลังนิยม สามารถแสดงข้อดี-ข้อเสียได้ดังต่อไปนี้



(ก) (ข)

รูปที่ 3.1 อุปกรณ์ที่เลือกใช้งาน

(ก) Arduino UNO

(ข) ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของ Arduino UNO : ใช้งานง่าย มีความหลายหลายในการใช้งาน มีอุปกรณ์เสริมต่างๆที่รองรับการทำงานเยอะ และเป็นที่นิยมในการใช้งานมากที่สุด

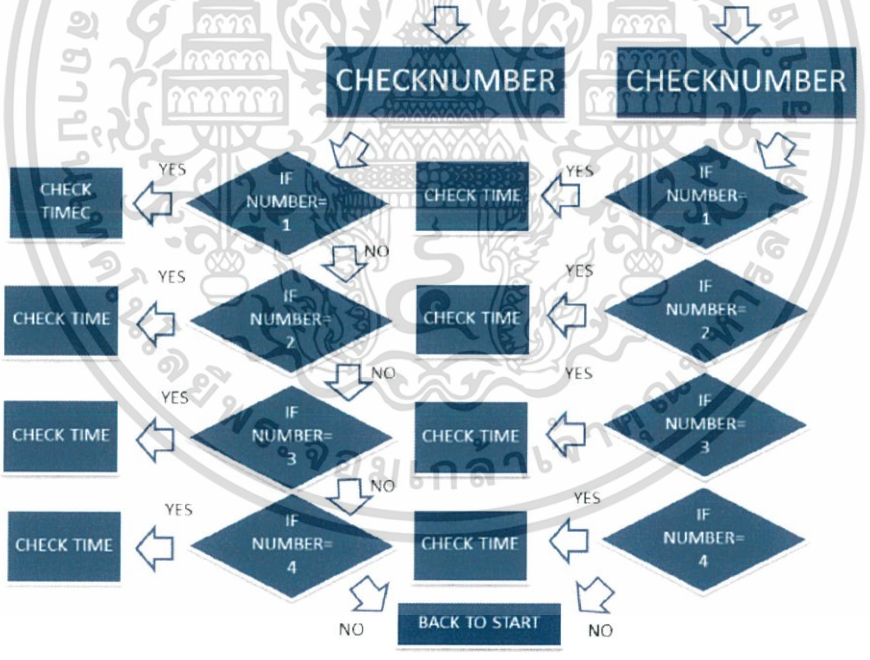
ข้อเสีย : ในกรณีที่ไมโครคอนโทรเลอร์ที่ติดตั้งมาเสียตัว Arduino UNO สามารถเปลี่ยนไมโครคอนโทรเลอร์ได้แต่ต้องระวังในการเปลี่ยนอาจทำให้ตัวบอร์ดหรืออุปกรณ์ที่นำมาต่อการใช้งานบางตัวเกิดความเสียหายได้จึงควรระมัดระวังในส่วนนี้

ข้อดีของ ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino : ใช้งานง่ายและสะดวก สามารถนำไปเชื่อมต่อกับ Arduino Uno ได้เลยสามารถตอบสนองความต้องการของงานที่ทำขึ้นได้

ข้อเสีย : มีจำนวนช่องในการเชื่อมต่อใช้งานที่จำกัดหากมีการนำไปใช้งานในลักษณะหลายๆ ช่องสัญญาณอาจเกิดปัญหาในส่วนนี้ได้

3.1.1 โปรแกรมที่พัฒนาขึ้นในส่วนคำสั่งของ Arduino

การทำงานในส่วนนี้จะเป็นการควบคุมคำสั่งเพื่อไปกำหนดค่าให้รีเลย์ทำงาน โดยรับคำสั่งมาจาก แอพพลิเคชั่นแสดงการทำงานได้ตามแผนผังด้านล่างดังรูปที่ 3.2 และรูปที่ 3.3 โปรแกรมการทำงานรับส่งข้อมูลผ่านระบบ Wi-Fi สามารถดูได้จากภาคผนวก ก.

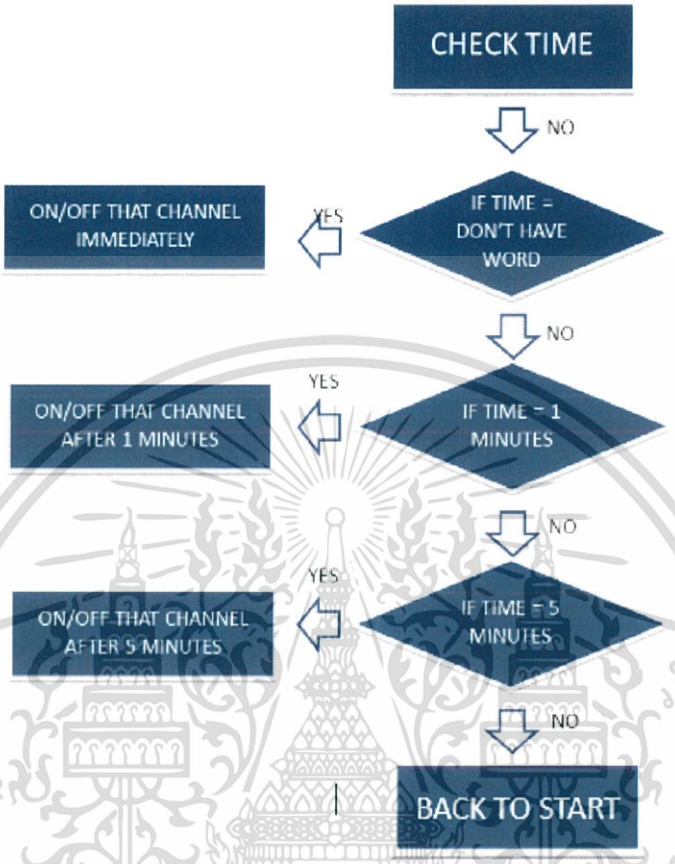


รูปที่ 3.2 แผนผังแสดงการทำงานของคำสั่งการควบคุมรีเลย์

ในส่วนรหัสของช่องสัญญาณการควบคุมที่รับมาจากคำพูดเป็นตัวเลขโดยมีตั้งแต่เลขที่ 1 ถึงเลขที่ 4 เป็นการแสดงถึงรหัสที่ต้องการเปิดปิดนั้นๆเมื่อตัวโปรแกรมทราบรหัสที่ต้องการเปิดตัวโปรแกรมจะที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบท้ายประโยคเพื่อดูว่ามีคำต่อจากตัวเลขหรือไม่เพื่อตรวจสอบว่ามีคำสั่งที่เกี่ยวข้องกับเวลาหรือไม่ถ้าหากไม่มีคำสั่งที่ไม่เกี่ยวข้องกับเวลาก็จะส่งคำสั่งค่าหนึ่งไปที่ Arduino เพื่อทำงานทันที



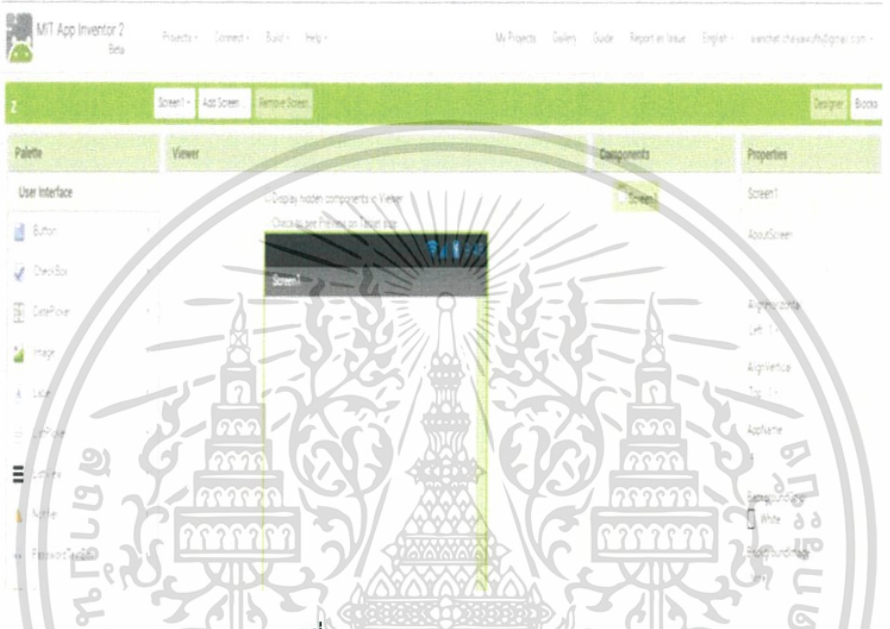
รูปที่ 3.3 แผนผังแสดงการทำงานของคำสั่งการตั้งเวลา

คำสั่งที่เกี่ยวข้องกับเวลาก็จะส่งคำสั่งอีกคำสั่งไปที่อาดูโน่โดยคำสั่งที่ได้รับจะเป็นการเช็คจำนวนเวลาที่ได้รับมาหลังจากการเลือกช่องสัญญาณการควบคุมโดยจะมีเวลาการตั้งเวลาเช่น 5 นาที 10 นาที 20 นาที 30 นาที 60 นาที หรือ 1 ชั่วโมง จากนั้นตัวโปรแกรมจะส่งข้อมูลไปที่ตัว Arduino จะรับค่ามาโดยหากมีการตั้งเวลา เช่น 5 นาที อาดูโน่จะนำเวลาที่คำสั่งเข้ามามาเทียบค่ากับเวลาที่เปิดตัวอาดูโน่ว่าตรงกันหรือยังหากตรงกัน จากนั้นจะทำงานตามเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การดำเนินงานในส่วนของแอพพลิเคชั่น

การดำเนินงานในส่วนนี้เราจะต้องเริ่มจากการศึกษาการเขียนแอพพลิเคชั่นจาก “MIT APP Inventor” ซึ่งทำได้ง่ายและสะดวกสามารถใช้ประโยชน์จากแอพพลิเคชั่นนี้ได้หลากหลายเพราะมีฟังก์ชั่นการใช้งานให้เลือกมากมาย สามารถหาวิธีการเขียนได้จากสื่อในอินเทอร์เน็ตทำให้สะดวกต่อการนำมาใช้งานสำหรับมือใหม่และผู้ที่ต้องการพัฒนาแอพพลิเคชั่น



รูปที่ 3.4 MIT APP Inventor

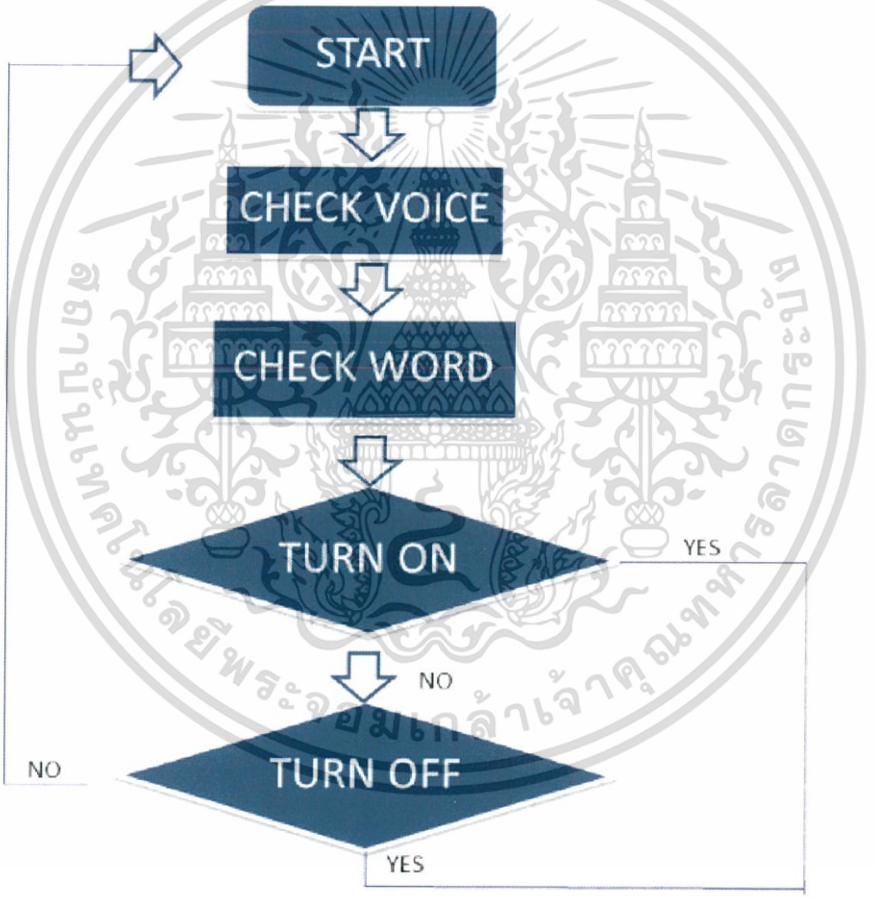
รูปที่ 3.5 ฟังก์ชั่นการทำงานใน MIT APP Inventor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.5 และรูปที่ 3.6 เป็นลักษณะโปรแกรมภายใน MIT APP Inventor ที่แสดงถึงฟังก์ชั่นและอุปกรณ์ที่ใช้งานในการสร้างตัวแอพพลิเคชั่นเพื่อนำมาใช้งานโดยแสดงหน้าแอพพลิเคชั่นที่ได้สร้างขึ้นในรูปที่ 3.6 โดยแถบเมนูด้านขวาจะแสดงถึงอุปกรณ์ในการใช้งานเพื่อสร้างฟังก์ชั่นการทำงานต่างๆ

3.2.1 โปรแกรมที่พัฒนาขึ้นในส่วนคำสั่งของแอพพลิเคชั่น

การทำงานในส่วนนี้จะเป็นการรับคำสั่งจากผู้ใช้โดยใช้เสียงที่ Google Voice ที่เป็ดาต้าเบสของโปรแกรมจับได้ส่งผ่านระบบ Wi-Fi และส่งค่าให้กับ Arduino โดยแสดงการทำงานของแอพพลิเคชั่นตามแผนผังแสดงการทำงานข้างล่างดังแสดงในรูปที่ 3.7 โดยฟังก์ชั่นคำสั่งการทำงานที่ได้ออกแบบสามารถดูได้จากภาคผนวก ข.



รูปที่ 3.6 แผนผังแสดงการทำงานของแอพพลิเคชั่น

เริ่มจากตัวแอปพลิเคชั่นจะเริ่มเช็คเสียงที่เข้ามายังโทรศัพท์จากนั้นจะทำการตรวจสอบคำในแต่ละคำจะเริ่มตรวจสอบคำว่า Turn On และ Turn Off โดยตัวแอปพลิเคชั่นจะทำการตรวจสอบ 2 คำ จากนั้นตัวโปรแกรมจะเข้าสู่การตรวจสอบในส่วนรหัสของช่องสัญญาณการควบคุม

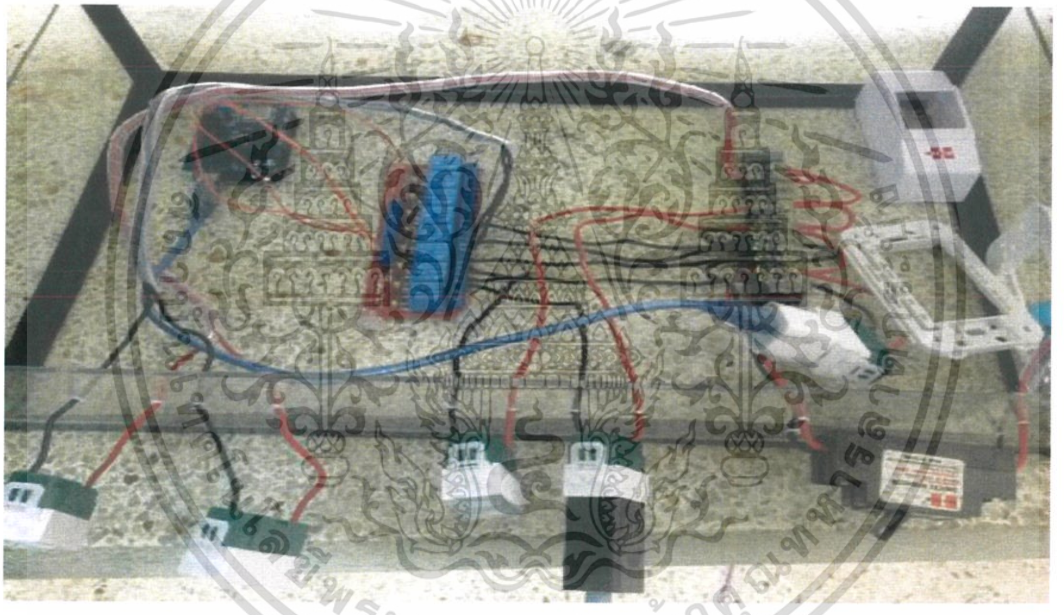
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การดำเนินงานในส่วนของระบบวงจรไฟฟ้า

การดำเนินงานในส่วนนี้เราจะต้องเริ่มจากการศึกษาวงจรไฟฟ้าขั้นพื้นฐานและอุปกรณ์ต่างๆที่นำมาใช้งาน โดยวงจรไฟฟ้าที่ใช้ต้องสามารถรองรับการทำงานของอุปกรณ์ไฟฟ้าต่างๆตามที่พักอาศัยได้จริง โดยวงจรไฟฟ้าที่ออกแบบนั้นจะจำลองการทำงานของระบบไฟฟ้าภายในบ้านออกมา

3.3.1 การพัฒนาในส่วนของระบบวงจรไฟฟ้า

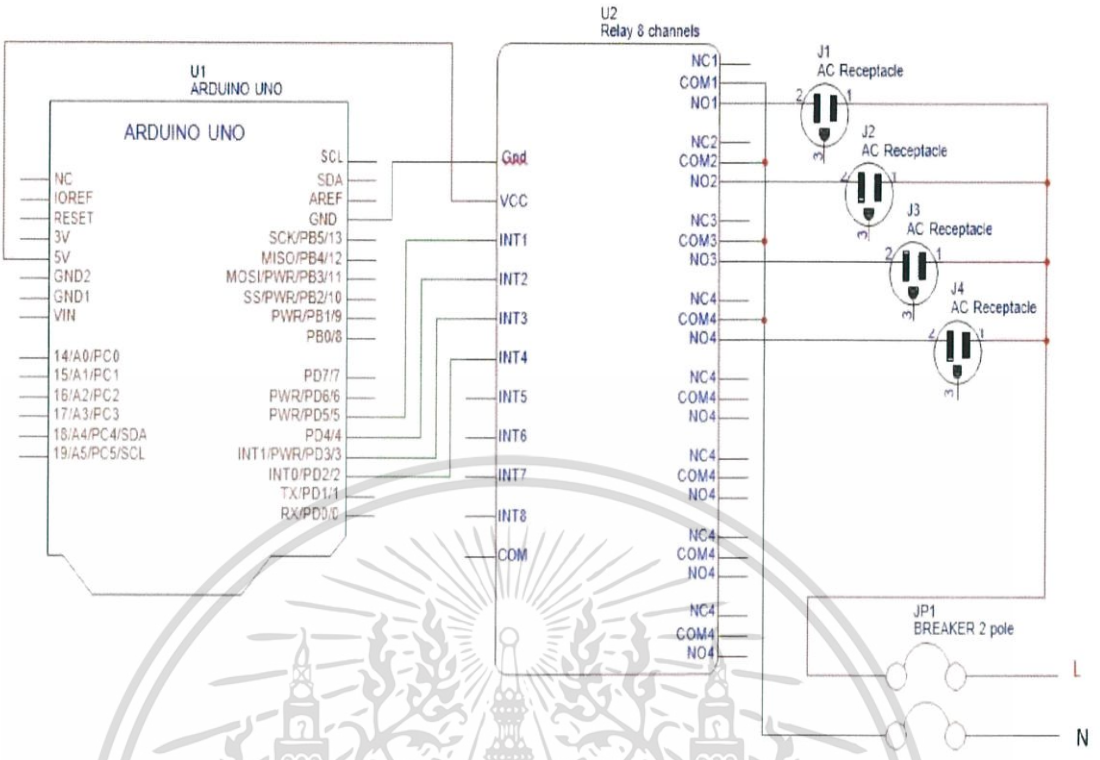
การทำงานในส่วนนี้จะเป็นการตัดต่อรีเลย์เพื่อควบคุมเครื่องใช้ไฟฟ้า โดยมีอินพุตที่ใช้ในการสั่งงานคือชุดคำสั่งที่รับมาจากโปรแกรม Arduino คำสั่งนี้จะไปควบคุมรีเลย์เพื่อใช้ตัดต่อวงจรไฟฟ้าในการทำงาน



รูปที่ 3.7 วงจรไฟฟ้าที่ออกแบบ

เมื่อไฟฟ้าเข้ามาที่เต้าเสียบจะผ่านเบรคเกอร์ แล้วเข้าไปที่เทอร์มินอลซึ่งหากเปรียบเป็นไฟบ้านจะเปรียบได้กับระบบวงจรย่อยของไฟฟ้าในอาคารโดนที่ตัววงจรย่อยจะมีสาย LINE ไปต่อค้างอยู่ จากนั้นเทอร์มินอลจะส่งไฟฟ้าไปยัง Arduino และตัวรีเลย์ ที่เทอนินอลจะมีสาย LINE เช่นกัน สำหรับตัวปลั๊กจะทำการต่อสายนิวตรอนเชื่อมต่อกัน เมื่อรีเลย์ต่อกระแสไฟเข้ามาก็จะครบวงจรพอดีสามารถใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การต่อวงจรไฟฟ้าภายในระบบที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดสอบส่วนการทำงานต่างๆ

ในการทดสอบการทำงานของชิ้นงานจะมีทั้งหมด 2 ขั้นตอน คือ การทดสอบการทำงานของแอพพลิเคชั่น ซึ่งเป็นการทดสอบเพื่อดูการทำงานของแอพพลิเคชั่นในการรับคำสั่งเสียงที่เป็นภาษาอังกฤษและตรวจสอบความแม่นยำในการตรวจจับคำสั่งเสียงด้วย Google Voice และขั้นตอนที่สองคือการทดสอบการทำงานของบอร์ด Arduino โดยเป็นการทดสอบการทำงานหลังรับคำสั่งจากแอพพลิเคชั่นแล้วทำการสั่งการควบคุมการทำงานของรีเลย์ โดยในการทดสอบเบื้องต้นจะใช้หลอด LED แทนอุปกรณ์ไฟฟ้าที่นำมาใช้งานจริง

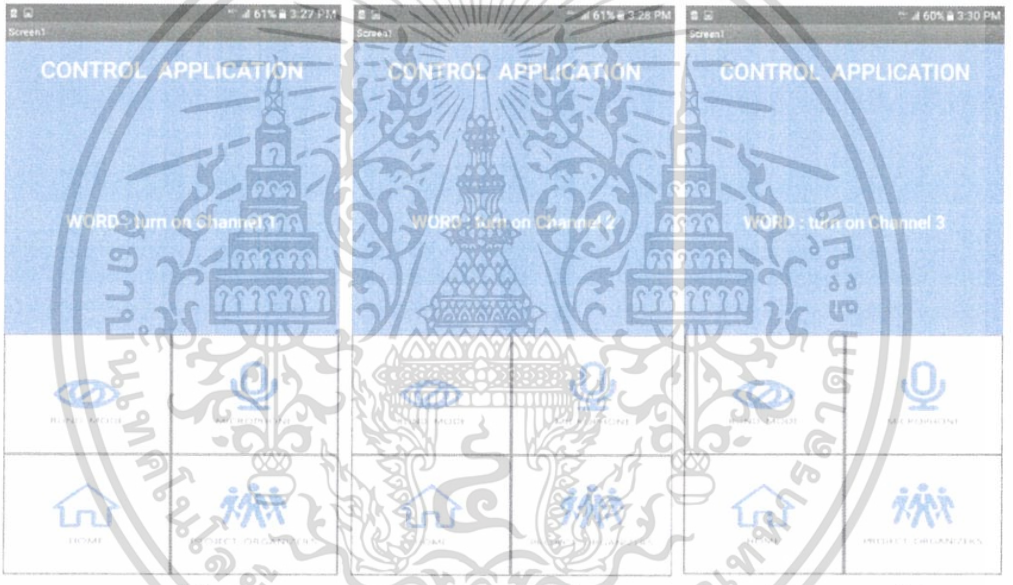
รูปที่ 4.1 การรับ-ส่งคำสั่งผ่านระบบ Wi-Fi ควบคุมเครื่องใช้ไฟฟ้า

ภาพรวมของระบบการทำงานจะเริ่มจากผู้ใช้สั่งการด้วยเสียงผ่านแอพพลิเคชั่นบนสมาร์ทโฟนหลังจากได้รับคำสั่งแล้วข้อมูลจะส่งผ่านไปยัง Wi-Fi เพื่อส่งข้อมูลที่ได้รับไปยัง Arduino เพื่อประมวลผลข้อมูลออกมาเป็นค่าที่ใช้สั่งการรีเลย์เพื่อไปควบคุมการทำงานของเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การทดสอบการทำงานในส่วนคำสั่งจากแอพพลิเคชั่น

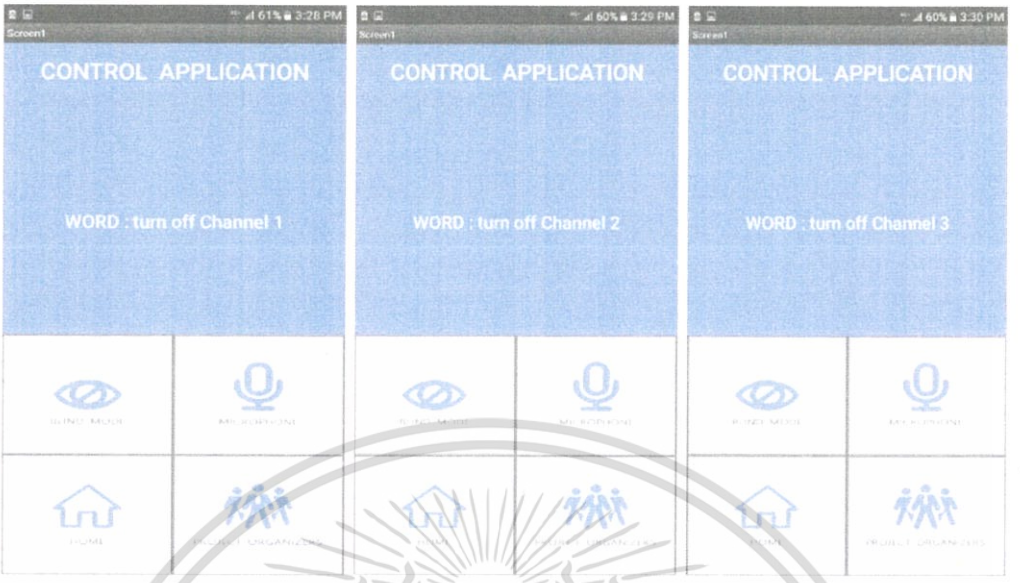
เพื่อทดสอบระบบฟังก์ชั่นการทำงานของแอพพลิเคชั่นบนโทรศัพท์มือถือหรือสมาร์ทโฟนว่าสามารถรับคำสั่งด้วยเสียงที่เป็นภาษาอังกฤษได้หรือไม่ โดยแอพพลิเคชั่นจะรับคำสั่งจากผู้ใช้งานแล้วใช้ดาต้าเบสจาก Google Voice ในการตรวจจับคำสั่ง สำหรับคำสั่งที่เขียนในโปรแกรมจะเป็นภาษาอังกฤษเพื่อให้แอพพลิเคชั่นมีความเป็นสากลมากขึ้น แต่แอพพลิเคชั่นนี้สามารถเปลี่ยนการรับคำสั่งเป็นภาษาไทยได้หากต้องการ เนื่องจาก Google Voice มีดาต้าเบสที่เป็นภาษาไทยเช่นกัน โดยจะต้องทำการเซ็ตโปรแกรมใหม่ให้รับคำสั่งจากภาษาอังกฤษเป็นภาษาไทย แล้วทำการแก้ข้อความที่ใช้ในการตรวจสอบคำสั่งภายในโปรแกรมเป็นข้อความภาษาไทยแทน คำสั่งที่ได้รับจะไปแสดงบนหน้าจอของแอพพลิเคชั่น ถ้าคำสั่งได้รับการตรวจสอบว่าถูกต้องจะส่งค่านั้นไปยัง Arduino บอร์ดเพื่อทำงานต่อไป แต่ในกรณีที่ตรวจสอบแล้วว่าคำสั่งไม่ถูกต้องจะไม่ส่งค่านั้นไปทำงาน



(ก) (ข) (ค)

รูปที่ 4.2 แสดงคำสั่งเปิดการใช้งานเครื่องใช้ไฟฟ้าที่แสดงบนแอพพลิเคชั่น
(ก) คำสั่งเปิดเครื่องใช้ไฟฟ้าช่องการควบคุมที่ 1
(ข) คำสั่งเปิดเครื่องใช้ไฟฟ้าช่องการควบคุมที่ 2
(ค) คำสั่งเปิดเครื่องใช้ไฟฟ้าช่องการควบคุมที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข) (ค)

รูปที่ 4.3 แสดงคำสั่งปิดการใช้งานเครื่องใช้ไฟฟ้าที่แสดงบนแอพพลิเคชั่น
(ก) คำสั่งปิดเครื่องใช้ไฟฟ้าช่องการควบคุมที่ 1
(ข) คำสั่งปิดเครื่องใช้ไฟฟ้าช่องการควบคุมที่ 2
(ค) คำสั่งปิดเครื่องใช้ไฟฟ้าช่องการควบคุมที่ 3

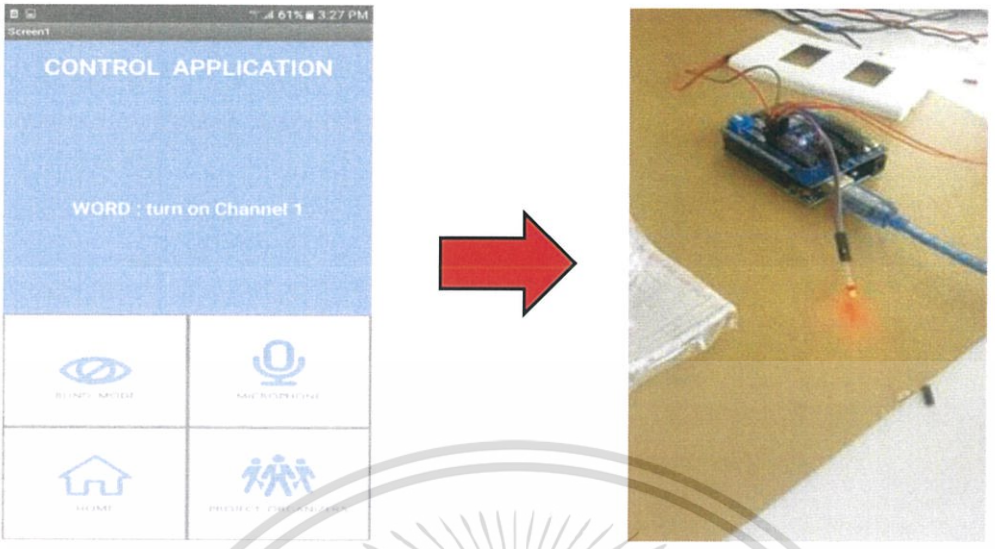
4.2 การทดสอบการทำงานในส่วนคำสั่งจาก Arduino บอร์ด

เพื่อทดสอบฟังก์ชั่นการทำงานเมื่อรับคำสั่งมาจากแอพพลิเคชั่นแล้ว Arduino บอร์ดสามารถสั่งการให้รีเลย์มีการตัดไฟหรือจ่ายไฟให้แก่วงจรไฟฟ้าได้หรือไม่ ในการทดสอบนี้จะเป็นการจำลองการใช้งานจริงโดยใช้หลอด LED แทนอุปกรณ์ไฟฟ้าที่นำมาใช้งานจริงมาต่อเข้ากับ Arduino บอร์ดโดยตรงเพื่อแสดงให้เห็นการทำงานของคำสั่งอย่างชัดเจน

4.2.1 แสดงการทำงานในส่วนของคำสั่งเปิดเครื่องใช้ไฟฟ้า (Turn On)

ในส่วนนี้จะทำการทดสอบในคำสั่งเปิด (Turn On) โดยที่ผู้ใช้งานจะสั่งการผ่านแอพพลิเคชั่นด้วยคำพูด “Turn On Channel 1” ถ้าคำสั่งถูกต้องจะแสดงค่าบนหน้าจอของแอพพลิเคชั่นดังรูปที่ 4.4 (ก) หลังจากนั้นค่าที่ถูกต้องจะส่งไปยัง Arduino บอร์ดเพื่อทำการควบคุมหลอดไฟ LED ตามที่ได้เขียนโปรแกรมการควบคุมตามรูปที่ 4.4 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข)

รูปที่ 4.4 แสดงการทำงานในคำสั่งเปิด (Turn On) ของแต่ละช่องการควบคุม
(ก) คำสั่งเปิด (Turn On) ใช้งานที่ถูกต้องแสดงบนแอพพลิเคชั่น
(ข) หลอดไฟ LED ที่ได้รับคำสั่งควบคุมการใช้งานมาจากแอพพลิเคชั่น

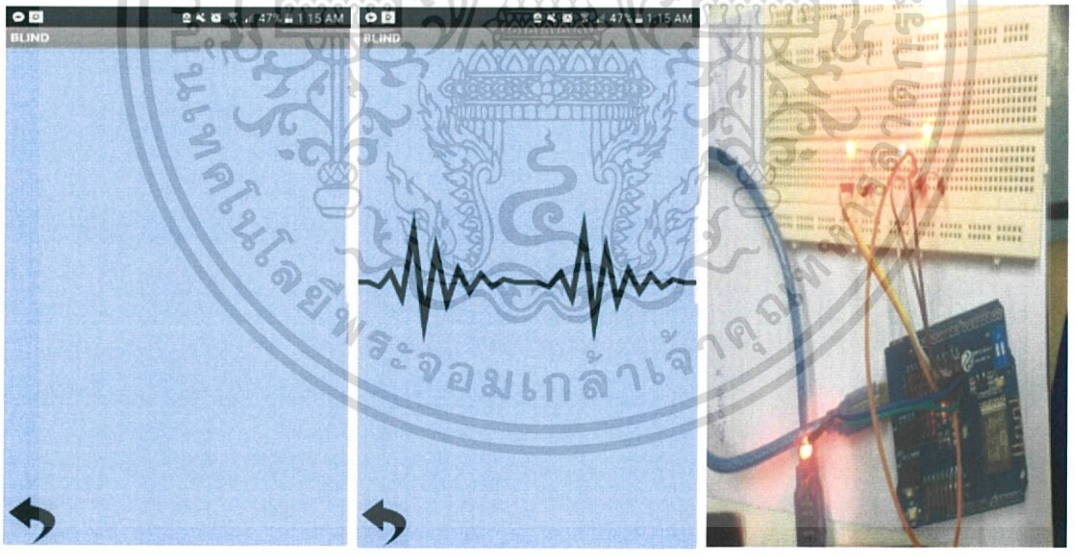
4.2.2 แสดงการทำงานในส่วนของคำสั่งปิดเครื่องใช้ไฟฟ้า (Turn Off)

ในส่วนนี้จะทำการทดสอบในคำสั่งปิด (Turn Off) โดยที่ผู้ใช้งานจะสั่งการผ่านแอพพลิเคชั่นด้วยคำพูด “Turn Off Channel 1” ถ้าคำสั่งถูกต้องจะแสดงค่าบนหน้าจอของแอพพลิเคชันดังรูปที่ 4.5 (ก) หลังจากนั้นค่าที่ถูกต้องจะส่งไปยัง Arduino บอร์ดเพื่อทำการควบคุมหลอดไฟ LED ตามที่ได้เขียนโปรแกรมการควบคุมตามรูปที่ 4.5 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) (ข)

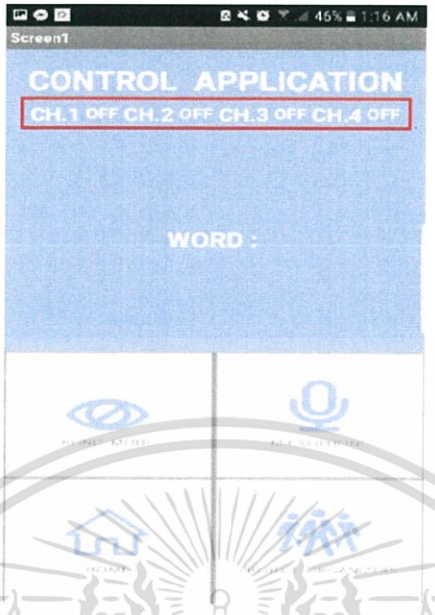
รูปที่ 4.5 แสดงการทำงานในคำสั่งปิด (Turn Off) ของแต่ละช่องการควบคุม
(ก) คำสั่งปิด (Turn Off) ใช้งานที่ถูกต้องแสดงบนแอพพลิเคชั่น
(ข) หลอดไฟ LED ที่ได้รับคำสั่งควบคุมการใช้งานมาจากแอพพลิเคชั่น



(ก) (ข) (ค)

รูปที่ 4.6 แสดงการทำงานคำสั่งเปิดการควบคุมในโหมดการทำงานของผู้พิการทางสายตา
(ก) หน้าจอแอพพลิเคชั่นในโหมดผู้พิการทางสายตาสามารถสัมผัสทุกบริเวณบนหน้าจอเพื่อใช้งาน
(ข) หน้าจอแอพพลิเคชั่นในโหมดผู้พิการทางสายตาจะตอบสนองคำสั่งของผู้ใช้งานด้วยเสียง เพื่อตรวจสอบความถูกต้องของคำสั่ง
(ค) หลอดไฟ LED ที่ได้รับคำสั่งควบคุมการใช้งานที่มาจากแอพพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 สถานะแสดงการตอบสนองของเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการดำเนินงานจัดทำระบบสั่งการเปิด-ปิดเครื่องใช้ไฟฟ้าด้วยเสียงผ่านอินเตอร์เน็ต ระบบสามารถควบคุมการทำงานเครื่องใช้ไฟฟ้าจากระยะไกลได้ โดยใช้เสียงสั่งการผ่านแอพพลิเคชั่น แล้วส่งข้อมูลไปที่ Arduino เพื่อสั่งตัดต่อรีเลย์ในการควบคุมการเปิด-ปิดเครื่องใช้ไฟฟ้า โดยมีกล่องบรรจุวงจรควบคุม ในที่นี้ได้จัดทำการควบคุมเครื่องใช้ไฟฟ้าจำนวนสามช่องการควบคุมเพื่อทดสอบการทำงานของระบบให้เป็นไปตามคำสั่งที่ได้โปรแกรมไว้ ซึ่งแต่ละช่องการควบคุมสามารถเปิด-ปิดได้ตามคำสั่ง แต่อาจมีความล่าช้าในการตอบสนองบ้างเล็กน้อยเนื่องจากมีขั้นตอนในการตรวจสอบที่ค่อนข้างมาก

5.2 ปัญหาและวิธีการแก้ไข

5.2.1 ปัญหาที่พบ

1. ขณะพูดสั่งการผ่านแอพพลิเคชั่น หากมีเสียงดังรบกวนจากภายนอก จะทำให้คำสั่งที่โปรแกรมตรวจจับได้เกิดความผิดพลาด

2. การส่งข้อมูลจากแอพพลิเคชั่นไปยัง Arduino มีความล่าช้า เนื่องจากกระบวนการในการตรวจเช็คตามคำสั่งที่ได้โปรแกรมไว้มีหลายขั้นตอนตั้งแต่การตรวจสอบคำพูด การตรวจสอบช่องการควบคุม การตรวจสอบเวลาในการทำการควบคุม จึงเป็นสาเหตุให้เกิดความล่าช้า

3. การตรวจสอบผลตอบสนองว่าเครื่องใช้ไฟฟ้าแต่ละชนิดนั้นทำงานอยู่หรือไม่ผ่านแอพพลิเคชั่นโดยการแสดงผลนั้นไม่สามารถทำได้ เนื่องจากการส่งข้อมูลจากแอพพลิเคชั่นไปยัง Arduino และส่งค่ากลับมามีการสื่อสารการทำงานกันคนละแบบจึงทำให้ไม่สามารถแสดงค่าได้

5.2.2 วิธีแก้ไขปัญหา

1. ขณะพูดสั่งการผ่านแอพพลิเคชั่นควรอยู่ในที่ๆไม่มีเสียงรบกวน หรือพูดคำสั่งด้วยเสียงที่ชัดเจนขึ้น ถ้าต้องการให้คำสั่งที่ใช้ในการทำงานเป็นภาษาไทยสามารถทำได้โดยการแก้ไขตรงฟังชั่นการทำงานของแอพพลิเคชั่นแต่ที่เลือกใช้ภาษาอังกฤษเพื่อให้การทำงานเป็นสากลมากขึ้น

2. ทำการทดสอบการใช้งานหลังจากเปลี่ยนโปรแกรมใหม่และวิเคราะห์หาข้อผิดพลาดตามจุดที่เกิดปัญหาเพื่อให้งานออกมาสมบูรณ์

3. เนื่องจากเราไม่สามารถตรวจสอบผลตอบสนองการทำงานของเครื่องใช้ไฟฟ้าโดยการส่งข้อมูลผ่านแอพพลิเคชั่นโดยตรงได้จึงใช้การตรวจสอบจากการทำงานของรีเลย์ว่ามีการทำงานอยู่หรือไม่แทนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

การเขียนโปรแกรมในการติดต่อสื่อสารระหว่างแอพพลิเคชั่นและ Arduino จำเป็นต้องมีความรู้เกี่ยวกับโครงสร้างทางด้านอินเตอร์เน็ตเพื่อให้กระบวนการทำงานมีความรวดเร็วมากขึ้นและสามารถรับคำสั่งอื่นๆหรือตรวจสอบผลตอบสนองได้ดีเยี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Arduino คืออะไร; แหล่งที่มา: https://www.arduinoall.com/product/16/arduin-uno-r3
สืบค้นเมื่อวันที่ 3 เดือนตุลาคม ปี พ.ศ.2559

[2] ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino; แหล่งที่มา: https://www.arduitronics.com/product/1027/ep8266-esp-13-wifi-web-sever-shield-for-arduino
สืบค้นเมื่อวันที่ 3 เดือนตุลาคม ปี พ.ศ.2559

[3] Arduino UNO; แหล่งที่มา: http://www.thaieasyelec.com/article-wiki/basic-electronics/arduino.html
สืบค้นเมื่อวันที่ 12 เดือนตุลาคม ปี พ.ศ.2559

[4] การตั้งค่า ESP8266 ESP-13 Wi-Fi Web Sever Shield for Arduino; แหล่งที่มา: https://fineshang.gitbooks.io/esp8266-based-serial-wifi-shield-for-arduino-user/content/chapter4.html
สืบค้นเมื่อวันที่ 25 เดือนพฤศจิกายน ปี พ.ศ.2559

[5] การต่อใช้งาน Relay Module; แหล่งที่มา: http://thaieasyelec.com/article-wiki/review-product-article.html
สืบค้นเมื่อวันที่ 24 เดือนมกราคม ปี พ.ศ.2560

[6] MIT App Invertor คืออะไร; แหล่งที่มา: https://programmingappinventor.wordpress.com/app-inventor/app-inventor
สืบค้นเมื่อวันที่ 25 เดือนมกราคม ปี พ.ศ.2560

[7] การสื่อสารข้อมูลแบบ TCP/IP; แหล่งที่มา: http://www.tnetsecurity.com/content_basic/tcp_ip_knowledge.php
สืบค้นเมื่อวันที่ 8 เดือนกุมภาพันธ์ ปี พ.ศ.2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โปรแกรมส่วน Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการทำงานรับส่งข้อมูลผ่านระบบ Wi-Fi

(เป็นตัวแปรที่ถูกกำหนดขึ้นมาเพื่อไว้ในการเปรียบเทียบค่ากับเวลาที่เข้ามา)

```
char x;

int  y=0;

unsigned long Time;

unsigned long Time1;

unsigned long Time2;

unsigned long Time3;

unsigned long Time4;

unsigned long Time5;

unsigned long Time6;

unsigned long Time7;

unsigned long Time8;

unsigned long Time9;

unsigned long Time10;

unsigned long Time11;

unsigned long Time12;

unsigned long Time13;

unsigned long Time14;

unsigned long Time15;

unsigned long Time16;

unsigned long Time17;

unsigned long Time18;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(เป็นตัวแปรที่ถูกกำหนดขึ้นมาเพื่อไว้ในการเปรียบเทียบค่ากับเวลาที่เข้ามา)

```
unsigned long Time19;

unsigned long Time20;

unsigned long Time21;

unsigned long Time22;

unsigned long Time23;

unsigned long Time24;

unsigned long Time25;

unsigned long Time26;

unsigned long Time27;

unsigned long Time28;

unsigned long Time29;

unsigned long Time30;

unsigned long Time31;

unsigned long Time32;

unsigned long Time33;

unsigned long Time34;

unsigned long Time35;

unsigned long Time36;

unsigned long Time37;

unsigned long Time38;

unsigned long Time39;

unsigned long Time40;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned long Time41;
unsigned long Time42;
unsigned long Time43;
unsigned long Time44;
unsigned long Time45;
unsigned long Time46;
unsigned long Time47;
unsigned long Time48;
(ช่องสัญญาณการควบคุมที่ต้องการให้เอาต์พุตออกมา)
boolean buttonState;
boolean lastState;
boolean state = HIGH;
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(10, INPUT);
  Serial.begin(9600);
}
void loop() {
   int reading = digitalRead(10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if ( reading == LOW && lastState == HIGH ) {
  delay (10);
  if (digitalRead(10) == LOW) state = !state;
 }
 digitalWrite(2,state);
 lastState = reading;
 //delay(1000);
 Time=millis();
 Serial.println(Time);
 Serial.println(Time1);
 if (Serial.available())
{
```

(เป็นส่วนที่ตรวจสอบคำสั่งที่เข้ามายังระบบโดยคำสั่งทั้งหมดเป็นการตรวจสอบแบบไม่มีดีเลย์ โดยจะทำการเปิดปิดตามคำสั่งที่ได้รับเช่นถ้าหากได้รับตัวเอเข้ามาระแบบจะทำการเปิดช่องสัญญาณการควบคุมที่2แบบทันที)

```
  x=Serial.read();
 Serial.println(x);
 if((x=='?'&& y==0) || y==1)
  {
   if ((x=='A'&& y==1)) {digitalWrite(2, HIGH);Serial.println("\nLED1 ON");y=0;}
//No deley//
  else if ((x=='c'&& y==1)) {digitalWrite(2, LOW);Serial.println("\nLED1
OFF");y=0;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   else if ((x=='H'&& y==1)) {digitalWrite(3, HIGH);Serial.println("\nLED2
ON");y=0;}

   else if ((x=='j'&& y==1)) {digitalWrite(3, LOW);Serial.println("\nLED2
OFF");y=0;}

   else if ((x=='O'&& y==1)) {digitalWrite(4, HIGH);Serial.println("\nLED3
ON");y=0;}

    else if ((x=='q'&& y==1)) {digitalWrite(4, LOW);Serial.println("\nLED3
OFF");y=0;}

    else if ((x=='V'&& y==1)) {digitalWrite(5, HIGH);Serial.println("\nLED4
ON");y=0;}

    else if ((x=='x'&& y==1)) {digitalWrite(5, LOW);Serial.println("\nLED4
OFF");y=0;}
```

(ทั้งหมดเป็นส่วนของตัวสัญญาณที่ต้องมีการดีเลย์ในระแบบก่อนปิดและเปิดตัวปลั๊กไฟโดยจะมีการกำหนดค่าขึ้นมาทั้งหมด48ค่าตามจำนวนคำสั่งเวลาที่ต้องได้รับทั้งหมด)

```
    //delay 1 minutes//
    else if ((x=='B'&& y==1)) {Time1=millis();}
    else if ((x=='d'&& y==1)) {Time2=millis();}
    else if ((x=='l'&& y==1))  {Time3=millis();}
    else if ((x=='k'&& y==1)) {Time4=millis();}
    else if ((x=='P'&& y==1)) {Time5=millis();}
    else if ((x=='r'&& y==1)) {Time6=millis();}
    else if ((x=='W'&& y==1)) {Time7=millis();}
    else if ((x=='y'&& y==1)) {Time8=millis();}
    else if ((x=='C'&& y==1)) {Time9=millis();} //delay 10 minutes//
    else if ((x=='e'&& y==1)) {Time10=millis();}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 else if ((x=='J'&& y==1)) {Time11=millis();}

 else if ((x=='l'&& y==1)) {Time12=millis();}

 else if ((x=='Q'&& y==1)) {Time13=millis();}

 else if ((x=='s'&& y==1)) {Time14=millis();}

 else if ((x=='X'&& y==1)) {Time15=millis();}

 else if ((x=='z'&& y==1)) {Time16=millis();}

 else if ((x=='D'&& y==1)) {Time17=millis();} //delay 15 minutes//

else if ((x=='f'&& y==1)) {Time18=millis();}

else if ((x=='K'&& y==1)) {Time19=millis();}

else if ((x=='n'&& y==1)) {Time20=millis();}

else if ((x=='R'&& y==1)) {Time21=millis();}

 else if ((x=='t'&& y==1)) {Time22=millis();}

 else if ((x=='Y'&& y==1)) {Time23=millis();}

 else if ((x=='1'&& y==1)) {Time24=millis();}

 else if ((x=='E'&& y==1)) {Time25=millis();} //delay 30 minutes//

 else if ((x=='g'&& y==1)) {Time26=millis();}

 else if ((x=='I'&& y==1)) {Time27=millis();}

 else if ((x=='m'&& y==1)) {Time28=millis();}

 else if ((x=='S'&& y==1)) {Time29=millis();}

 else if ((x=='u'&& y==1)) {Time30=millis();}

  else if ((x=='Z'&& y==1)) {Time31=millis();}

  else if ((x=='2'&& y==1)) {Time32=millis();}

  else if ((x=='F'&& y==1)) {Time33=millis();} //delay 60 minutes//
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if ((x=='h'&& y==1)) {Time34=millis();}

else if ((x=='N'&& y==1)) {Time35=millis();}

else if ((x=='o'&& y==1)) {Time36=millis();}

else if ((x=='T'&& y==1)) {Time37=millis();}

else if ((x=='v'&& y==1)) {Time38=millis();}

else if ((x=='a'&& y==1)) {Time39=millis();}

else if ((x=='3'&& y==1)) {Time40=millis();}

else if ((x=='G'&& y==1)) {Time41=millis();} //delay 300 minutes//

else if ((x=='i'&& y==1)) {Time42=millis();}

else if ((x=='M'&& y==1)) {Time43=millis();}

else if ((x=='p'&& y==1)) {Time44=millis();}

else if ((x=='U'&& y==1)) {Time45=millis();}

else if ((x=='w'&& y==1)) {Time46=millis();}

else if ((x=='b'&& y==1)) {Time47=millis();}

else if ((x=='4'&& y==1)) {Time48=millis();}

else {y=1;}

}
```

(ในส่วนนี้เป็นส่วนของ1นาทีโดยโปรแกรมจะตัวค่าที่ได้เป็นเซตเท่ากับ TIME จากนั้นจะทำการเปรียบเทียบเวลาโดย1นาทีจะเท่ากับ 60000 มิลลิวินาทีเมื่อตัวโปรแกรมเปรียบเทียบเท่าแล้วก็จะทำการสั่งเปิดหรือปิด)

```
}

if( Time>=Time1+60000 && Time<=Time1+61000 )

{

  digitalWrite(2, HIGH);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.println("\nLED1 ON1");}
else if (Time>=Time2+60000 && Time<=Time2+61000)
{digitalWrite(2, LOW);Serial.println("\nLED1 OFF1");}
else if (Time>=Time3+60000 && Time<=Time3+61000)
{digitalWrite(3, HIGH);Serial.println("\nLED2 ON1");}
else if (Time>=Time4+60000 && Time<=Time4+61000)
{digitalWrite(3, LOW);Serial.println("\nLED2 OFF1");}
else if (Time>=Time5+60000 && Time<=Time5+61000)
{digitalWrite(4, HIGH);Serial.println("\nLED3 ON1");}
else if (Time>=Time6+60000 && Time<=Time6+61000)
{digitalWrite(4, LOW);Serial.println("\nLED3 OFF1");}
else if (Time>=Time7+60000 && Time<=Time7+61000)
{digitalWrite(5, HIGH);Serial.println("\nLED4 ON1");}
else if (Time>=Time8+60000 && Time<=Time8+61000)
{digitalWrite(5, LOW);Serial.println("\nLED4 OFF1");}
```

(ในส่วนนี้เป็นส่วนของ10นาทีโดยโปรแกรมจะตัวค่าที่ได้เป็นเซตเท่ากับ TIME จากนั้นจะทำการเปรียบเทียบเวลาโดย1นาทีจะเท่ากับ 600000 มิลลิวินาทีเมื่อตัวโปรแกรมเปรียบเทียบเท่าแล้วก็จะทำการสั่งเปิดหรือปิดเช่นเดียวกัน)

```
//10 minutes
else if (Time>=Time9+600000 && Time<=Time9+601000)
{digitalWrite(2, HIGH);Serial.println("\nLED1 ON5");}
else if (Time>=Time10+600000 && Time<=Time10+601000)
{digitalWrite(2, LOW);Serial.println("\nLED1 OFF5");}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if (Time>=Time11+600000 && Time<=Time11+601000)
{digitalWrite(3, HIGH);Serial.println("\nLED2 ON5");}
else if (Time>=Time12+600000 && Time<=Time12+601000)
{digitalWrite(3, LOW);Serial.println("\nLED2 OFF5");}
else if (Time>=Time13+600000 && Time<=Time13+601000)
{digitalWrite(4, HIGH);Serial.println("\nLED3 ON5");}
else if (Time>=Time14+600000 && Time<=Time14+601000)
{digitalWrite(4, LOW);Serial.println("\nLED3 OFF5");}
else if (Time>=Time15+600000 && Time<=Time15+601000)
{digitalWrite(5, HIGH);Serial.println("\nLED4 ON5");}
else if (Time>=Time16+600000 && Time<=Time16+601000)
{digitalWrite(5, LOW);Serial.println("\nLED4 OFF5");}
```

(ในส่วนนี้เป็นส่วนของ15นาทีโดยโปรแกรมจะตัวค่าที่ได้เป็นเซตเท่ากับ TIME จากนั้นจะทำการเปรียบเทียบเวลาโดย1นาทีจะเท่ากับ 900000 มิลลิวินาทีเมื่อตัวโปรแกรมเปรียบเทียบเท่าแล้วก็จะทำการสั่งเปิดหรือปิดโดยโปรแกรมจะทำการเขียนคล้ายๆกันแตกต่างกันที่เวลาที่เราต้องการเปรียบเทียบ)

```
//15 minutes
  else if (Time>=Time17+900000 && Time<=Time17+901000)
{digitalWrite(2, HIGH);Serial.println("\nLED1 ON5");}
else if (Time>=Time18+900000 && Time<=Time18+901000)
{digitalWrite(2, LOW);Serial.println("\nLED1 OFF5");}
else if (Time>=Time19+900000 && Time<=Time19+901000)
{digitalWrite(3, HIGH);Serial.println("\nLED2 ON5");}
else if (Time>=Time20+900000 && Time<=Time20+901000)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{digitalWrite(3, LOW);Serial.println("\nLED2 OFF5");}
else if (Time>=Time21+900000 && Time<=Time21+901000)
{digitalWrite(4, HIGH);Serial.println("\nLED3 ON5");}
else if (Time>=Time22+900000 && Time<=Time22+901000)
{digitalWrite(4, LOW);Serial.println("\nLED3 OFF5");}
else if (Time>=Time23+900000 && Time<=Time23+901000)
{digitalWrite(5, HIGH);Serial.println("\nLED4 ON5");}
else if (Time>=Time24+900000 && Time<=Time24+901000)
{digitalWrite(5, LOW);Serial.println("\nLED4 OFF5");}
```

(ในส่วนนี้เป็นส่วนของ30นาทีโดยโปรแกรมจะตัวค่าที่ได้เป็นเซตเท่ากับ TIME จากนั้นจะทำการเปรียบเทียบเวลาโดย1นาทีจะเท่ากับ 180000 มิลลิวินาทีเมื่อตัวโปรแกรมเปรียบเทียบเท่าแล้วก็จะทำการสั่งเปิดหรือปิดเช่นเดียวกันกับด้านบน)

```
//30 minutes
else if (Time>=Time25+1800000 && Time<=Time25+1801000)
{digitalWrite(2, HIGH);Serial.println("\nLED1 ON5");}
else if (Time>=Time26+1800000 && Time<=Time26+1801000)
{digitalWrite(2, LOW);Serial.println("\nLED1 OFF5");}
else if (Time>=Time27+1800000 && Time<=Time27+1801000)
{digitalWrite(3, HIGH);Serial.println("\nLED2 ON5");}
else if (Time>=Time28+1800000 && Time<=Time28+1801000)
{digitalWrite(3, LOW);Serial.println("\nLED2 OFF5");}
else if (Time>=Time29+1800000 && Time<=Time29+1801000)
{digitalWrite(4, HIGH);Serial.println("\nLED3 ON5");}
else if (Time>=Time30+1800000 && Time<=Time30+1801000)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{digitalWrite(4, LOW);Serial.println("\nLED3 OFF5");}
else if (Time>=Time31+1800000 && Time<=Time31+1801000)
{digitalWrite(5, HIGH);Serial.println("\nLED4 ON5");}
else if (Time>=Time32+1800000 && Time<=Time32+1801000)
{digitalWrite(5, LOW);Serial.println("\nLED4 OFF5");}
```

(ในส่วนนี้เป็นส่วนของ1ชั่วโมงโดยโปรแกรมจะตัวค่าที่ได้เป็นเซตเท่ากับ TIME จากนั้นจะทำการเปรียบเทียบเวลาโดย1นาทีจะเท่ากับ 36000000 มิลลิวินาทีเมื่อตัวโปรแกรมเปรียบเทียบเท่าแล้วก็จะทำการสั่งเปิดหรือปิด)

```
//60 minutes
else if (Time>=Time33+3600000 && Time<=Time33+3601000)
{digitalWrite(2, HIGH);Serial.println("\nLED1 ON5");}
else if (Time>=Time34+3600000 && Time<=Time34+3601000)
{digitalWrite(2, LOW);Serial.println("\nLED1 OFF5");}
else if (Time>=Time35+3600000 && Time<=Time35+3601000)
{digitalWrite(3, HIGH);Serial.println("\nLED2 ON5");}
else if (Time>=Time36+3600000 && Time<=Time36+3601000)
{digitalWrite(3, LOW);Serial.println("\nLED2 OFF5");}
else if (Time>=Time37+3600000 && Time<=Time37+3601000)
{digitalWrite(4, HIGH);Serial.println("\nLED3 ON5");}
else if (Time>=Time38+3600000 && Time<=Time38+3601000)
{digitalWrite(4, LOW);Serial.println("\nLED3 OFF5");}
else if (Time>=Time39+3600000 && Time<=Time39+3601000)
{digitalWrite(5, HIGH);Serial.println("\nLED4 ON5");}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if (Time>=Time40+3600000 && Time<=Time40+3601000)

{digitalWrite(5, LOW);Serial.println("\nLED4 OFF5");}
```

(ในส่วนสุดท้ายนี้เป็นส่วนของ5ชั่วโมงโดยโปรแกรมจะตัวค่าที่ได้เป็นเซตเท่ากับ TIME จากนั้นจะทำการเปรียบเทียบเวลาโดย1นาทีจะเท่ากับ 18001000 มิลลิวินาทีเมื่อตัวโปรแกรมเปรียบเทียบเท่าแล้วก็จะทำการสั่งเปิดหรือปิดตามลำดับ)

```
//300minutes

else if (Time>=Time41+18000000 && Time<=Time41+18001000)

 {digitalWrite(2, HIGH);Serial.println("\nLED1 ON5");}

else if (Time>=Time42+18000000 && Time<=Time42+18001000)

{digitalWrite(2, LOW);Serial.println("\nLED1 OFF5");}

else if (Time>=Time43+18000000 && Time<=Time43+18001000)

{digitalWrite(3, HIGH);Serial.println("\nLED2 ON5");}

else if (Time>=Time44+18000000 && Time<=Time44+18001000)

{digitalWrite(3, LOW);Serial.println("\nLED2 OFF5");}

else if (Time>=Time45+18000000 && Time<=Time45+18001000)

{digitalWrite(4, HIGH);Serial.println("\nLE
```

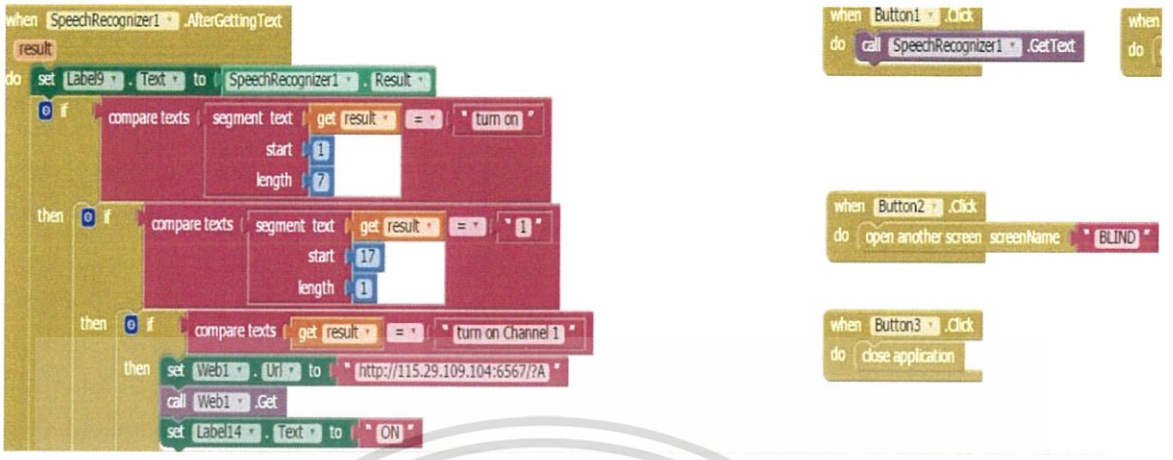
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ฟังก์ชั่นการทำงานในส่วนของแอพพลิเคชั่น

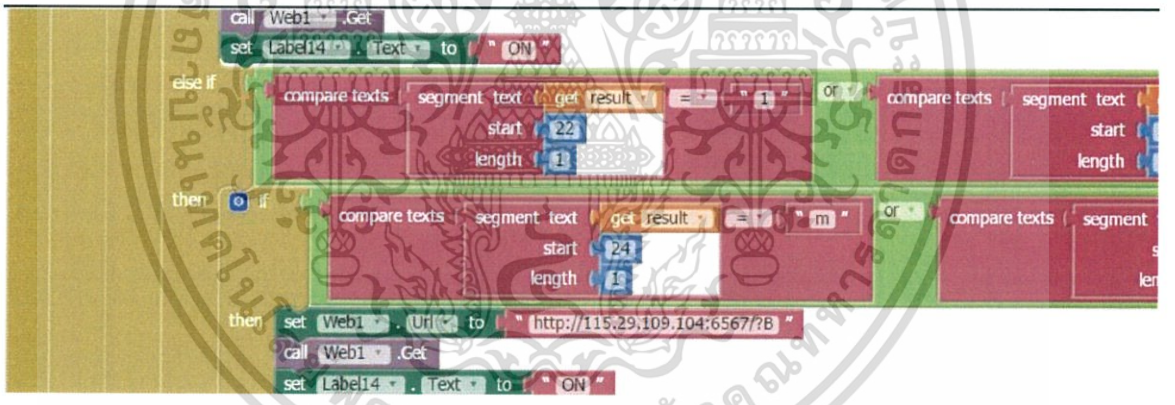
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่นคำสั่งการทำงานใน Application



รูปที่ ข.1 ฟังก์ชั่นคำสั่งการทำงานใน Application 1

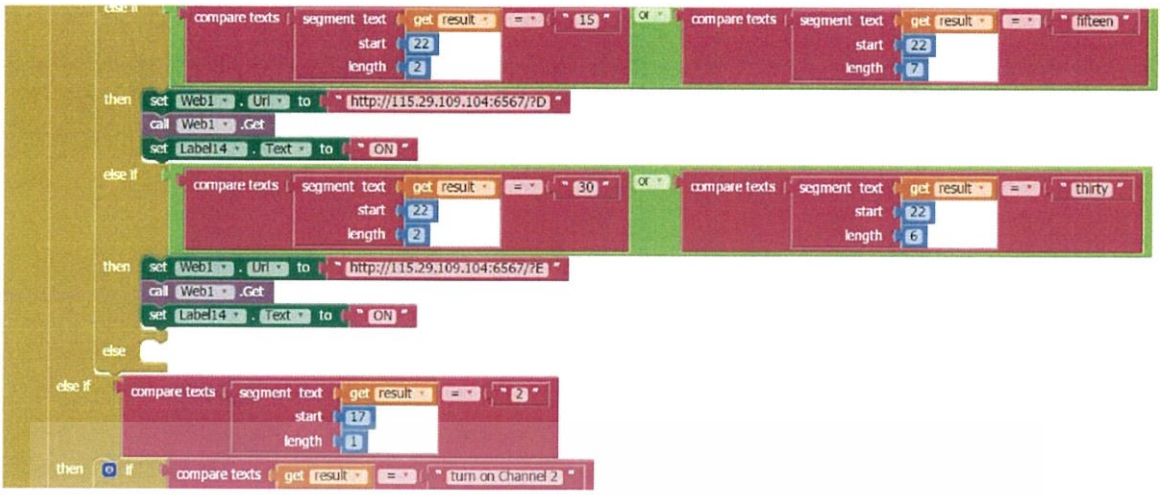
ด้านซ้ายเป็นส่วนจองการตรวจสอบเสียงที่เข้ามาแล้วตรวจสอบคำพูดว่าเป็นแชลแนลที่ 1 จากนั้นจะส่งคำสั่งไปยังIP 115.29.109.104 ที่พอต 6567และด้านซ้ายบนสุดเป็นปุ่มเรียกตัวรับเสียง บนที่ 2 เป็นปุ่มเข้าสู่หน้าผู้พิการทางสายตายและปุ่มที่ 3 เป็นปุ่มปิดแอปพลิเคชั่น



รูปที่ ข.2 ฟังก์ชั่นคำสั่งการทำงานใน Application 2

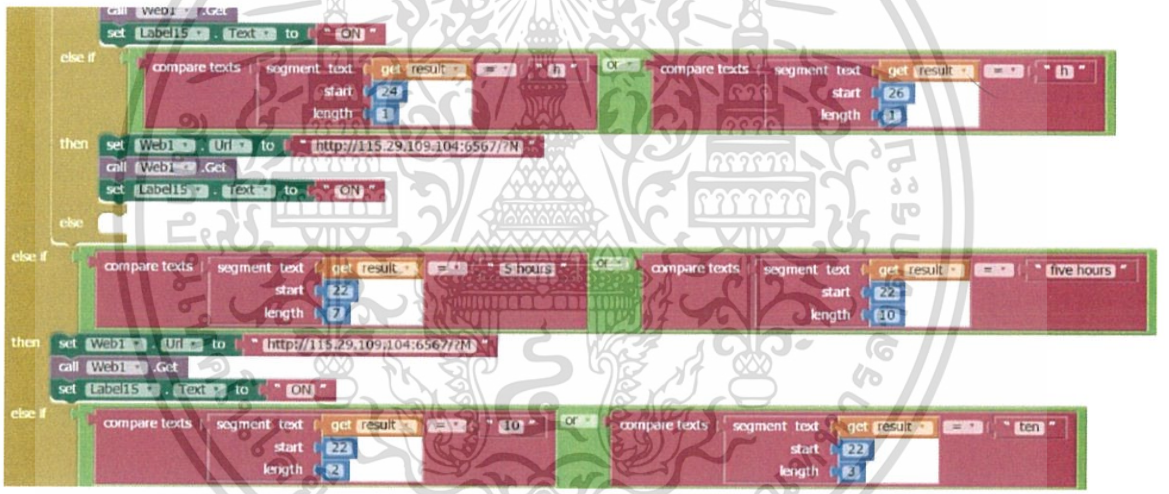
ในส่วนนี้เป็นส่วนของการเช็คจำนวนเวลาที่เข้ามาว่าใช้ 1 นาทีหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.3 ฟังก์ชั่นคำสั่งการทำงานใน Application 3

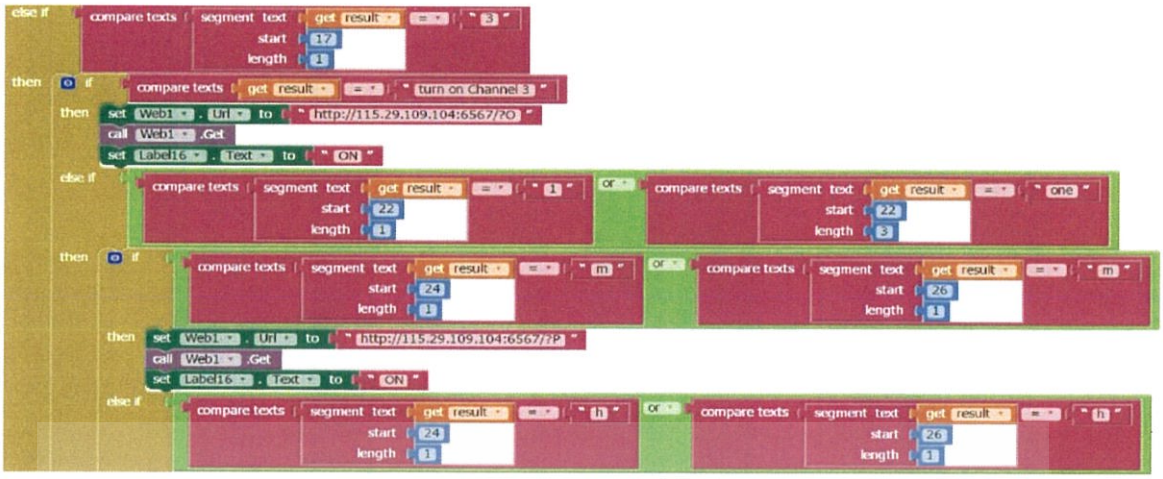
ในส่วนของหน้านี้เป็นส่วนที่ไว้เช็คว่าเป็น 15 นาทีหรือ 30 นาทีถ้าหากไม่ใช่ทั้งคู่โปรแกรมจะทำการเช็คแชแนลใหม่ว่าใช่แชแนล 2 หรือไม่



รูปที่ ข.4 ฟังก์ชั่นคำสั่งการทำงานใน Application 4

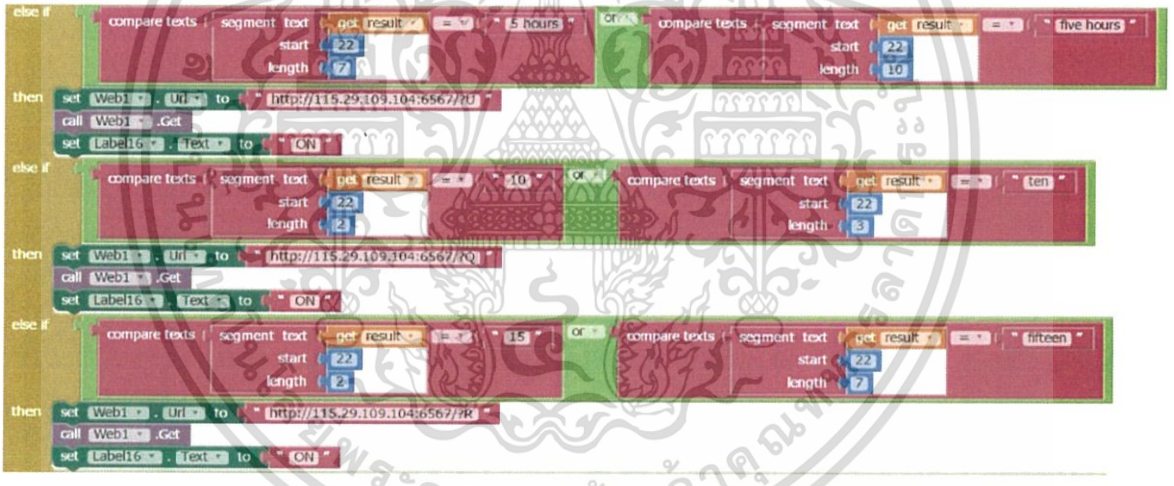
ในส่วนนี้เป็นส่วนที่ตรวจสอบว่าข้อมูลที่เข้ามาเป็น 1 ชั่วโมงหรือ 5 ชั่วโมงหรือ 10 นาทีจากนั้นจะทำการส่งข้อมูลไปที่ IP 115.29.109.104 ที่พอต 6567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.5 ฟังก์ชั่นคำสั่งการทำงานใน Application 5

ในส่วนนี้เป็นส่วนของการตรวจสอบการเปิดของแชลแนลที่ 3 และตรวจสอบจำนวนเวลาที่ได้รับเข้ามาเพื่อดีเลย์ในการเปิดว่าเป็น 1 นาทีหรือ 1 ชั่วโมงจากนั้นจะส่งข้อมูลไปที่ IP 115.29.109.104 ที่พอต 6567



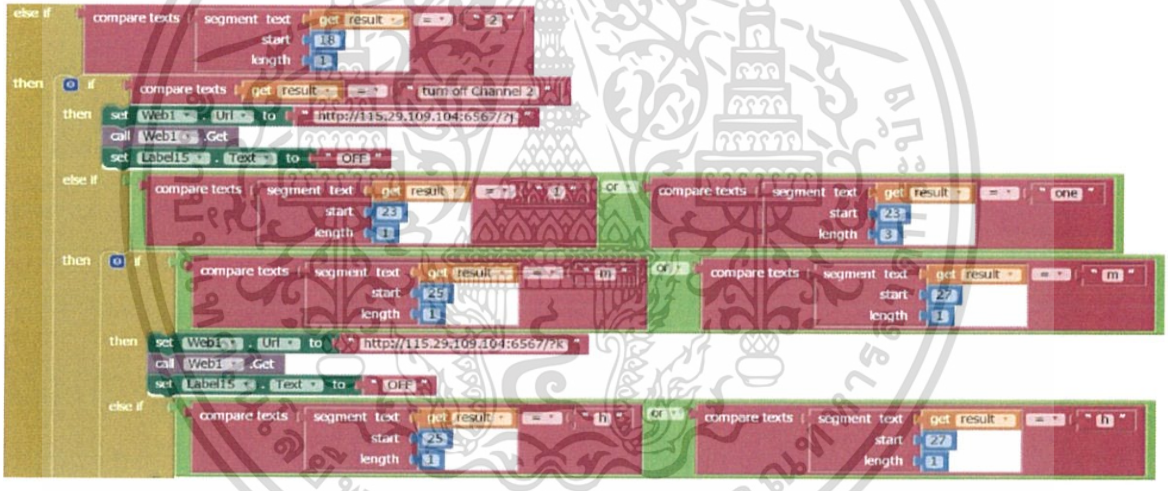
รูปที่ ข.6 ฟังก์ชั่นคำสั่งการทำงานใน Application 6

เป็นส่วนต่อเนื่องจากด้านบนโดยจะตรวจสอบว่าคำสั่งที่ได้เป็น 10 นาที 15 นาที หรือ 5 ชั่วโมง จากนั้นจะส่งคำสั่งไปที่ IP 115.29.109.104 ที่พอต 6567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.7 ฟังก์ชั่นคำสั่งการทำงานใน Application 7

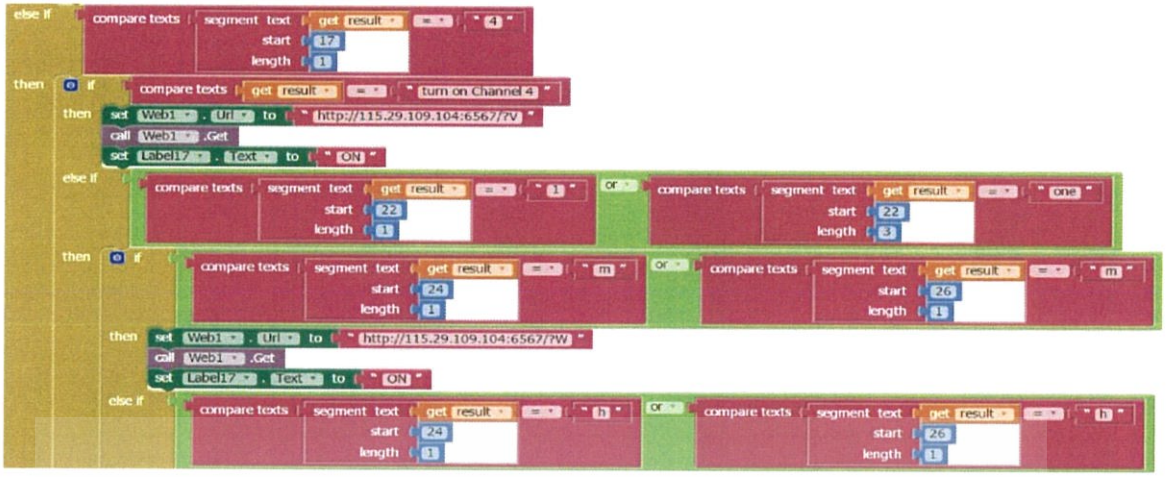
ในส่วนนี้จะเป็นการตรวจสอบการปิดของแชลแนลที่ 1 และตรวจสอบคำสั่งว่าให้ดีเลย์หรือไม่ถ้าหากดีเลย์จะให้ดีเลย์กี่นาที



รูปที่ ข.8 ฟังก์ชั่นคำสั่งการทำงานใน Application 8

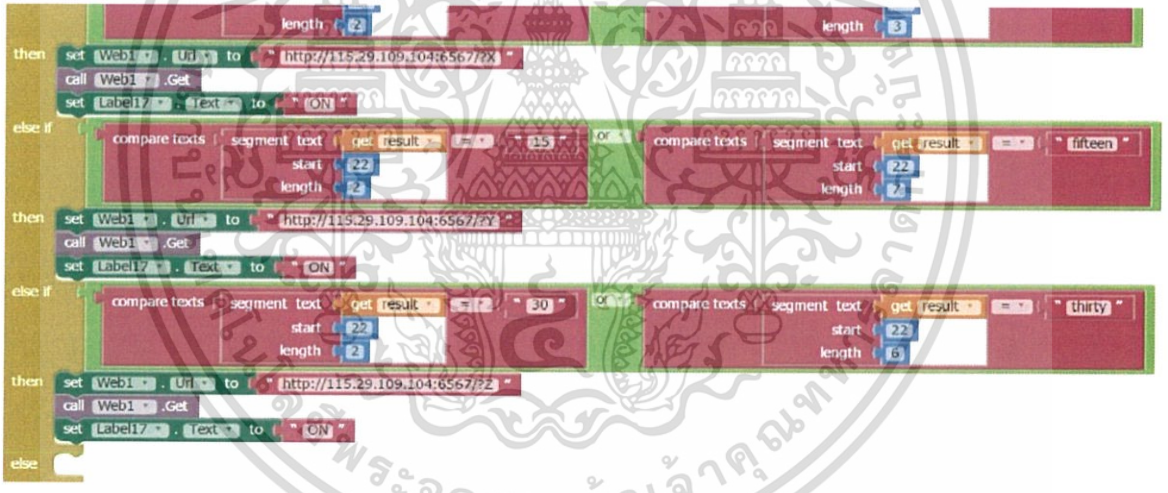
เป็นการตรวจสอบของคำสั่งปิดของแชแนลที่ 2 และตรวจสอบคำสั่งว่าให้ดีเลย์หรือไม่ถ้าหากดีเลย์จะให้ดีเลย์กี่นาทีเหมือนกับแชลแนลที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.9 ฟังก์ชั่นคำสั่งการทำงานใน Application 9

เป็นโปรแกรมตรวจสอบการสั่งปิดของแชแนลที่ 3 โดยมนส่วนของโปรแกรมด้านบนจะประกอบด้วยการสั่งปิดแบบไม่มีการดีเลย์และการสั่งปิดแบบดีเลย์ 1 นาทีและ 1 ชั่วโมงจากนั้นจะส่งคำสั่งไปที่ IP 115.29.109.104 ที่พอต 6567



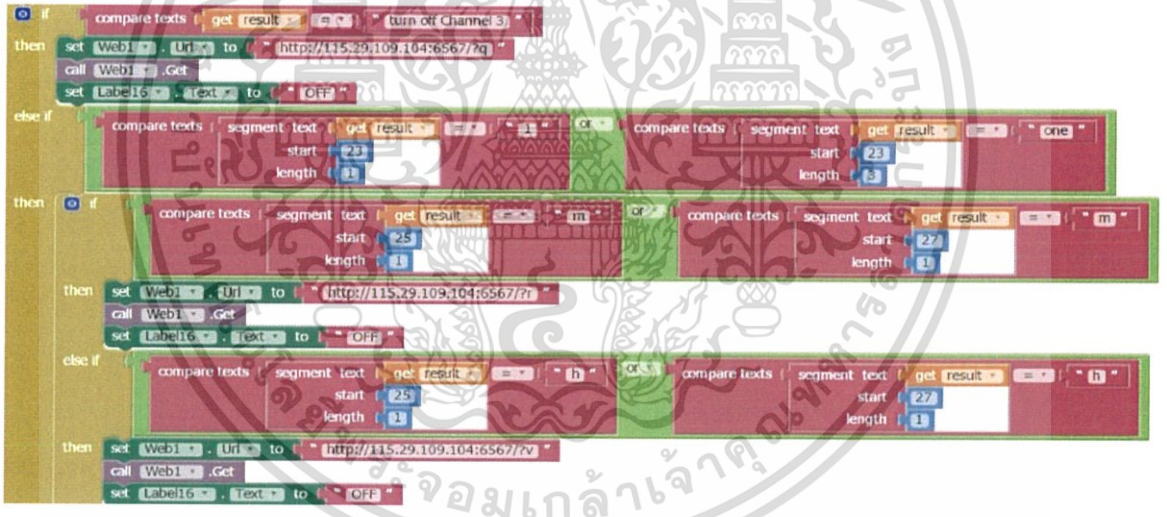
รูปที่ ข.10 ฟังก์ชั่นคำสั่งการทำงานใน Application 10

รูปนี้ก็เช่นเดียวกับด้านบนโดยเป็นโปรแกรมตรวจสอบการสั่งปิดของแชแนลที่ 3 โดยส่วนของโปรแกรมจะประกอบด้วยการสั่งปิดแบบดีเลย์ 15 นาทีและการสั่งปิดแบบดีเลย์ 30 นาทีจากนั้นจะส่งคำสั่งไปที่ IP 115.29.109.104 ที่พอต 6567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.11 ฟังก์ชั่นคำสั่งการทำงานใน Application 11

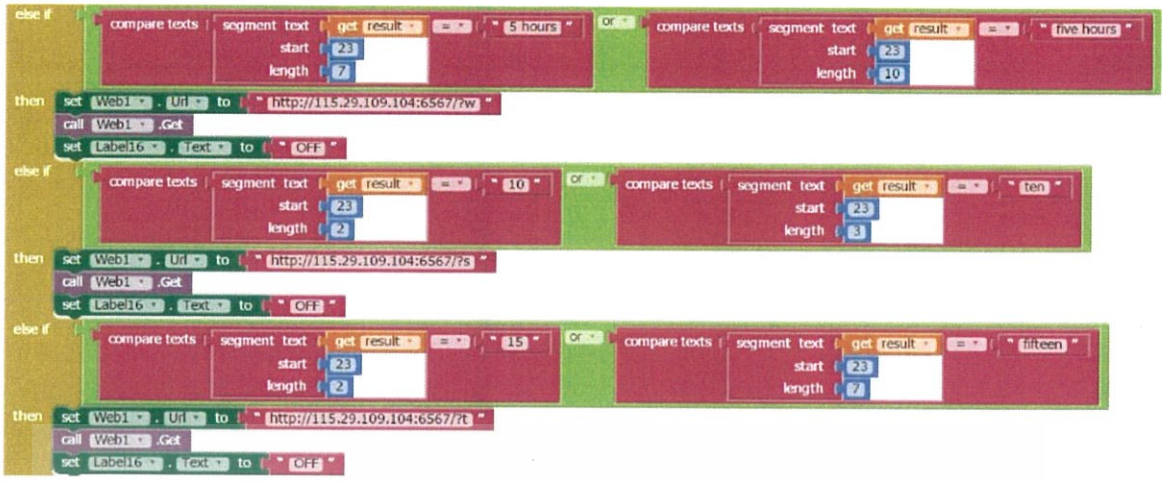
เป็นโปรแกรมตรวจสอบการสั่งปิดของแชแนลที่3 โดยมนส่วนของโปรแกรมด้านบนจะประกอบด้วยการสั่งปิดแบบไม่มีการดีเลย์และการสั่งเปิดแบบดีเลย์ 10 นาทีและ 15 นาทีจากนั้นจะส่งคำสั่งไปที่ IP 115.29.109.104 ที่พอต 6567



รูปที่ ข.12 ฟังก์ชั่นคำสั่งการทำงานใน Application 12

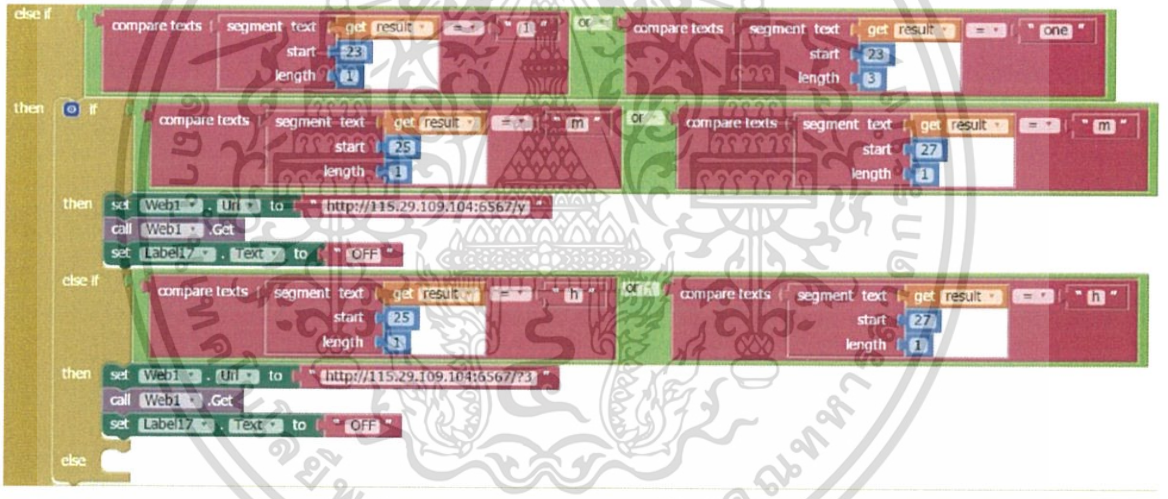
ในส่วนสุดท้ายเป็นส่วนของการสั่งปิดโปรแกรมของแชลแนลสุดท้ายคือแชแนลที่4 ซึ่งจะมีปิดแบบทันทีกับปิดแบบต้องการดีเลย์โดยในรูปนี้จะประกอบด้วยการปิดแบบดีเลย์ 1 นาทีและ 1 ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.13 ฟังก์ชั่นคำสั่งการทำงานใน Application 13

ในส่วนนี้เป็นการตรวจชอบว่าคำสั่งที่ได้รับเข้ามานั้นเป็นคำสั่งที่ให้ดีเลย์การปิดกี่นาทีโดยในรูปนี้จะทำการตรวจสอบที่ 5 ชั่วโมง 10 นาทีและ 15 นาที



รูปที่ ข.14 ฟังก์ชั่นคำสั่งการทำงานใน Application 14

ในส่วนนี้เป็นส่วนสุดท้ายคือเช็คเวลาที่เหลือของการสั่งปิดคือการสั่งการตรวจสอบที่ 30นาที และส่งคำสั่งไปที่ IP 115.29.109.104 ที่พอต 6567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้