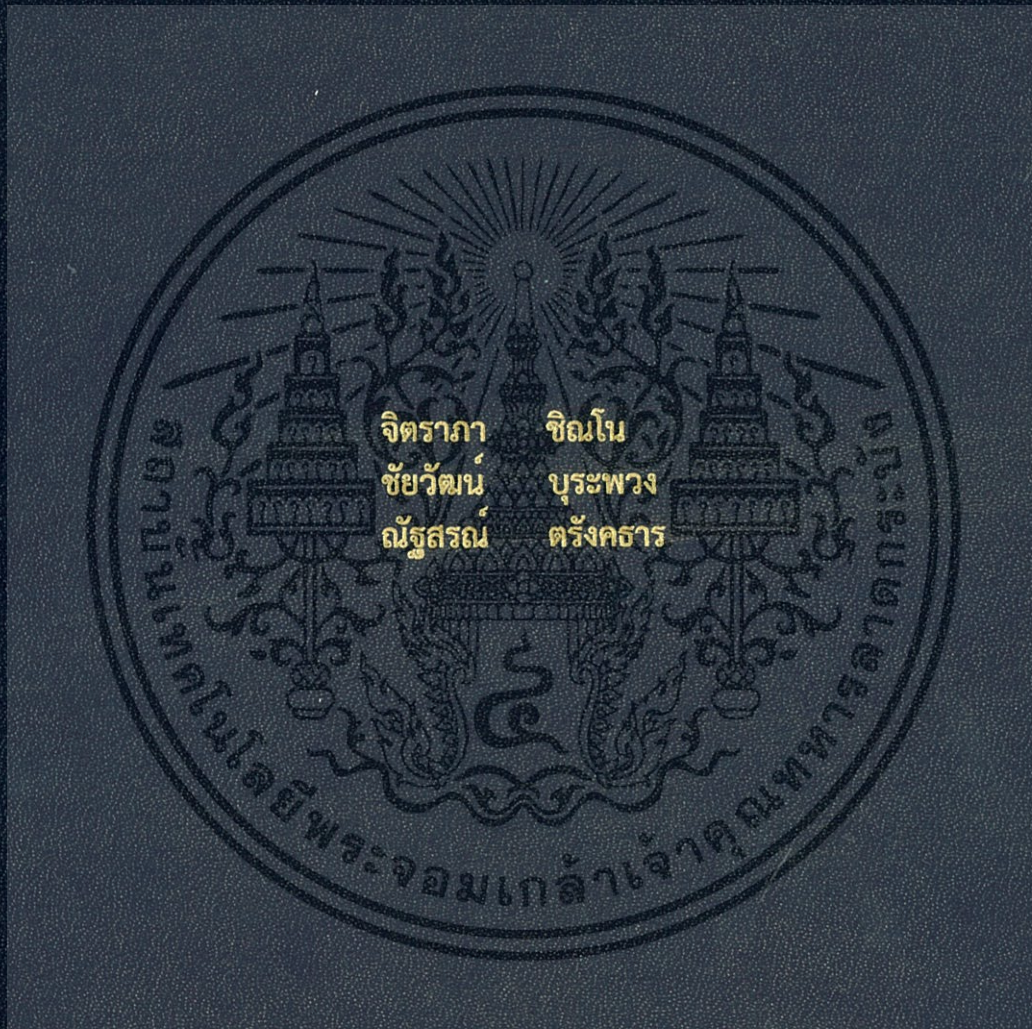


ระบบตั้งเวลาเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต
CONTROLLED TIMER PROGRAM AND ON-OFF ELECTRICAL
APPLIANCE THROUGH INTERNET



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ระบบตั้งเวลาเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต
CONTROLLED TIMER PROGRAM AND ON-OFF ELECTRICAL
APPLIANCE THROUGH INTERNET



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROLLED TIMER PROGRAM AND
ON-OFF ELECTRICAL APPLIANCE THROUGH INTERNET



A PROJECT 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

ปริญญาานิพนธ์ปีการศึกษา 2559
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ ระบบตั้งเวลาเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต
CONTROLLED TIMER PROGRAM AND ON-OFF ELECTRICAL
APPLIANCE THROUGH INTERNET

นักศึกษาผู้จัดทำ นางสาวจิตรภา ชินโน รหัสนักศึกษา 56010179
นายชัยวัฒน์ บุระพวง รหัสนักศึกษา 56010296
นายณัฐสรณ์ ตรังคธาร รหัสนักศึกษา 56010438

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2559

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ทรงชัย วีระทวิมาศ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบตั้งเวลาเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต		
	CONTROLLED TIMER PROGRAM AND ON-OFF ELECTRICAL APPLIANCE THROUGH INTERNET		
นักศึกษาผู้จัดทำ	นางสาวจิตรภา	ชิณโน	รหัสนักศึกษา 56010179
	นายชัยวัฒน์	บุระพวง	รหัสนักศึกษา 56010296
	นายณัฐสรณ์	ตรังคราร	รหัสนักศึกษา 56010438
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ทรงชัย วีระทวิมาศ		
ปีการศึกษา	2559		

บทคัดย่อ

ปริญญานิพนธ์นี้จัดทำเกี่ยวกับ การควบคุมการเปิดปิดเครื่องมือเครื่องใช้ไฟฟ้าโดยตั้งเวลาผ่านระบบ อินเทอร์เน็ต ซึ่งในปัจจุบันนี้ อินเทอร์เน็ตมีบทบาทกับชีวิตของเรามากขึ้น เราจึงสามารถใช้ระบบอินเทอร์เน็ตเข้ามาทำให้ชีวิตมีความสะดวกสบายมากยิ่งขึ้น จากการใช้งานเครื่องใช้ไฟฟ้าโดยสามารถตั้งเวลาเปิดปิดผ่านระบบอินเทอร์เน็ตได้ โดยการเข้าไปตั้งเวลาผ่านทางเว็บไซต์ และในขณะที่ระบบอินเทอร์เน็ตนั้นใช้งานไม่ได้ก็สามารถที่จะควบคุมผ่านบอร์ดปุมกดได้โดยใช้ Arduino ในการสั่งควบคุม ซึ่งระบบการทำงานหลักๆก็คือ การนำเอาบอร์ด Arduino และ Esp8266 มาใช้เป็นตัวควบคุมการทำงานของชุด Relay แล้วต่อสายจากเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิด-ปิดเข้ามายังชุด Relay หลังจากนั้น การเปิดปิดของเครื่องใช้ไฟฟ้าต่างๆ ก็จะขึ้นอยู่กับการทำงานของ Relay ตัวที่เครื่องใช้ไฟฟ้านั้นๆได้ต่ออยู่นั่นเอง ซึ่งหลังจากที่ได้จัดทำขึ้นงาน และได้ทดลองใช้งาน พบว่าสามารถตั้งเวลาเปิดปิดเครื่องใช้ไฟฟ้าผ่านระบบอินเทอร์เน็ตได้ตามวัตถุประสงค์

Thesis title	CONTROLLED TIMER PROGRAM AND ON-OFF ELECTRICAL APPLIANCE THROUGH INTERNET
Authors	Ms. Jitrapa Chinno Mr. Chaiwat Burapoung Mr. Nattasorn Trangkratarn
Thesis Advisor	Assoc.Prof. Songchai Weerathaweemas
Year	2016

ABSTRACT

This project designs a microcontroller based system that allows appliances switching via the Internet. Specifically, the design uses an Arduino as well as an Esp8266 to control the relay coil operation. The Arduino is connected to a web server as well as the Esp8266, in such the way that users can use timers to switch the appliances on and off when the Internet connection exists; and use the relay coil through the Arduino board when connection ceases. Results show that the designed system takes almost instantly to switch the appliances on and off. After the experimented have been created and first experimented, they are found to installed to turn on and turn off via the internet as expected.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณ ท่านรองศาสตราจารย์ทรงชัย วีระทวีมาศ อาจารย์ประจำภาควิชาวิศวกรรมการวัดคุม ซึ่งเป็นอาจารย์ผู้ควบคุมและคอยแนะนำให้คำปรึกษาเกี่ยวกับปริญญาานิพนธ์แก่คณะผู้จัดทำเป็นอย่างดี

ขอกราบขอบพระคุณคณาจารย์ประจำหลักสูตรวิศวกรรมการวัดคุม รวมถึงคณาจารย์ประจำคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆท่านที่ได้มอบความรู้ตั้งแต่ผู้จัดทำได้เข้าทำการศึกษา เพื่อนำความรู้ที่ได้สั่งสมมาประกอบในการทำปริญญาานิพนธ์

ขอขอบพระคุณพี่ๆเพื่อนๆทุกท่านที่คอยให้คำแนะนำ ช่วยเหลือเพื่อให้ปริญญาานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอกราบขอบพระคุณ บิดา มารดา ของคณะผู้จัดทำทุกท่าน ซึ่งเป็นผู้มอบชีวิต มอบการศึกษา และอนาคตที่ดี ตลอดจนให้ความช่วยเหลือ คำปรึกษา และเป็นกำลังใจในด้านต่างๆ ในการทำปริญญาานิพนธ์ฉบับนี้ให้สำเร็จลุล่วงไปได้ด้วยดี

คุณค่าและประโยชน์อันพึงมีจากปริญญาานิพนธ์ฉบับนี้ทั้งหมด ทางคณะผู้จัดทำขอมอบให้แก่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	1
1.4 ขั้นตอนการศึกษา.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	2
บทที่ 2 อุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง.....	3
2.1 Web Server.....	3
2.2 NETPIE (Network Platform for Internet of Everything).....	4
2.3 ไมโครคอนโทรลเลอร์ตระกูล AVR.....	5
2.3.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม.....	6
2.3.2 รูปแบบการเขียนโปรแกรมบน Arduino.....	6
2.4 Node MCU8266 Module Wifi.....	8
2.5 Relay 8 Channel.....	9
บทที่ 3 การใช้งาน Firebase การทำแอปพลิเคชัน และการดำเนินการด้านฮาร์ดแวร์.....	10
3.1 การใช้งาน Firebase.....	10
3.1.1 เตรียม Firebase ให้พร้อมใช้งาน.....	10
3.1.2 การกำหนดสิทธิ์การเข้าถึงข้อมูล.....	14
3.1.3 การใช้งานคอนโซลของ Firebase.....	15
3.1.4 ทำความเข้าใจกับ Realtime Database.....	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ IV อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.1.5 Firebase กับ ESP8266.....	23
3.1.6 ทำความรู้จักกับฟังก์ชันต่างๆ.....	25
3.1.7 การสตรีมมิ่งข้อมูล.....	26
3.1.8 ทดลองควบคุม LED แบบเรียลไทม์.....	27
3.2 การเชื่อม Esp8266 เข้ากับ Firebase.....	28
3.1.8 ทดลองควบคุม LED แบบเรียลไทม์.....	34
3.3 การทำแอปพลิเคชัน.....	36
3.3.1 ส่วนของ Designer.....	37
3.3.2 ส่วนของ Blocks.....	41
3.4 การดำเนินงานด้าน Hardware.....	43
บทที่ 4 การทดลองและผลการทดลอง.....	47
4.1 อุปกรณ์การทดลอง.....	47
4.1.1 อุปกรณ์ที่ใช้ส่งงานควบคุม.....	47
4.2.2 เรทเตอร์ตัวส่งสัญญาณอินเทอร์เน็ต.....	47
4.1.3 ชุดควบคุมต่างๆ.....	48
4.2 การทดลองควบคุมอุปกรณ์ไฟฟ้า.....	48
4.2.1 การติดตั้งแอปพลิเคชัน.....	48
4.2.2 การทดลองควบคุมเปิด-ปิด Firebase.....	52
4.2.2 การทดลองเปิด-ปิดผ่านแอปพลิเคชัน.....	53
4.2.4 การทดลองเปิด-ปิดร่วมกับเครื่องใช้ไฟฟ้า.....	55
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	58
5.1 สรุปผล.....	58
5.2 ข้อเสนอแนะ.....	58
5.3 แนวทางการพัฒนา.....	58
บรรณานุกรม.....	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และv้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางแสดงรายละเอียดและคำสั่ง set.....	25
3.2 ตารางแสดงรายละเอียดและคำสั่ง get.....	25
3.3 ตารางแสดงรายละเอียดและคำสั่ง push.....	25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา แลงหรืออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 เครือข่ายของ Internet of Things.....	3
2.2 ตัวอย่าง Arduino.....	5
2.3 การต่อ Arduino เข้ากับ LED	5
2.4 การที่ Arduino ต่อกับ XBee Shield	6
2.5 การเชื่อมต่อของ Arduino.....	6
2.6 เลือกบอร์ด Arduino ที่ต้องการ Upload.....	7
2.7 การเลือกหมายเลข Comport ของบอร์ด.....	7
2.8 การตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม.....	8
2.9 การ Upload โค้ดโปรแกรม.....	8
2.10 ตัวอย่างของ Node MCU8266.....	9
2.11 ตัวอย่างของ Relay 8 Channel.....	9
3.1 ขั้นตอนการลงชื่อเข้าใช้ Firebase.....	11
3.2 ขั้นตอนการลงชื่อเข้าใช้บัญชี Google.....	11
3.3 ขั้นตอนการสร้างโปรเจค.....	11
3.4 ขั้นตอนการระบุชื่อโปรเจค และ ชื่อประเทศ.....	12
3.5 ขั้นตอนการติดตั้งโปรเจคใหม่.....	12
3.6 ขั้นตอนการติดตั้งโปรเจคใหม่ (2).....	12
3.7 ขั้นตอนการติดตั้งโปรเจคใหม่ (3).....	13
3.8 ขั้นตอนการติดตั้งโปรเจคใหม่ (4).....	13
3.9 ขั้นตอนการคัดลอกลิงค์เพื่อใช้ในการคอนฟิก	13
3.10 ขั้นตอนการกำหนดสิทธิ์การเข้าถึงข้อมูล.....	14
3.11 ขั้นตอนการกำหนดสิทธิ์การเข้าถึงข้อมูล (2).....	14
3.12 ขั้นตอนการใช้งานคอนโซลของ Firebase.....	15
3.13 ขั้นตอนการเพิ่มข้อมูลลงในแอปเจค.....	15
3.14 เมื่อเพิ่มข้อมูลเสร็จสิ้นแล้ว.....	15
3.15 การกำหนดชื่อในแอปเจค.....	16
3.16 การกำหนดชื่อในแอปเจคย่อย.....	16
3.17 รูปแบบโครงสร้างข้อมูลแบบต้นไม้.....	16
3.18 การทดลองเปลี่ยนชื่อในส่วนของ Text.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา แลงVJ|องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.19 การทดลองเปลี่ยนชื่อในส่วนของ Text (2).....	17
3.20 การลบส่วนออปเจคย่อย subObj.....	17
3.21 การลบส่วนออปเจคย่อย subObj (2).....	18
3.22 การลบส่วนออปเจคย่อย subObj (3).....	18
3.23 การเพิ่มออปเจคที่มี Key เป็น LED และมีค่าเป็น ON ในคอมพิวเตอรืตัวที่ 1.....	19
3.24 การเปลี่ยนแปลงของ Firebase ในคอมพิวเตอรืตัวที่ 2.....	19
เมื่อเพิ่มออปเจคที่มี Key เป็น LED และมีค่าเป็น ON ในคอมพิวเตอรืตัวที่ 1	
3.25 การเพิ่มออปเจคที่มี Key เป็น LED และมีค่าเป็น Off ในคอมพิวเตอรืตัวที่ 1.....	20
3.26 การเปลี่ยนแปลงของ Firebase ในคอมพิวเตอรืตัวที่ 2.....	20
เมื่อเพิ่มออปเจคที่มี Key เป็น LED และมีค่าเป็น OFF ในคอมพิวเตอรืตัวที่ 1	
3.27 การลบออปเจค LED.....	21
3.28 การลบออปเจค LED ถูกลบไปแล้วบนคอมพิวเตอรืตัวที่ 1.....	22
3.29 การลบออปเจค LED ถูกลบไปแล้วบนคอมพิวเตอรืตัวที่ 2.....	22
3.30 รูปแบบของโปรแกรม ESP Blink.....	23
3.31 การเปิดไฟล์ในโปรแกรม ESP Blink.....	23
3.32 การอัปโหลดโค้ดโปรแกรม ESP Blink.....	24
3.33 การเปิดหน้าต่างคอนโซลร่วมกับหน้าต่างมอนิเตอร์ของโปรแกรม ESP Blink.....	24
3.34 รูปแบบการใช้งานฟังก์ชัน Firebase.stream().....	26
3.35 รูปแบบการใช้งานฟังก์ชัน Firebase.stream() (2).....	26
3.36 การต่อวงจรระหว่าง NodeMCU กับ บอร์ด.....	27
3.37 การทดลองควบคุม LED.....	27
3.38 การทดลองควบคุม LED เมื่อให้ค่า LED เป็น 0.....	28
3.39 การทดลองควบคุม LED เมื่อให้ค่า LED เป็น 1.....	28
3.40 การตั้งเวลาของจอ Serial Monitor.....	32
3.41 การตั้งค่าโครงการ.....	32
3.42 หน้าของบัญชีลับฐานข้อมูล.....	33
3.43 การคัดลอกคีย์.....	33
3.44 แถบเมนู Develop.....	34
3.45 ลิงค์ที่ใช้ในการคอนฟิก.....	34

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.46 หน้าจอ RULES.....	35
3.47 หน้าจอสำหรับกดปุ่ม PUBLISH.....	35
3.48 การตั้งชื่อแอปพลิเคชัน.....	36
3.49 ตัวเลือกคำสั่งต่างๆในส่วนของ Designer.....	37
3.50 การใช้งานคำสั่ง Horizontal และ Image.....	38
3.51 หน้า Screen1 หลังจากตกแต่งเสร็จแล้ว.....	39
3.52 หน้า Design ของ Screen2	40
3.53 การเชื่อมลิงค์ระหว่าง Firebase กับแอปพลิเคชัน	41
3.54 หน้าจอในส่วนของ Blocks	41
3.55 ตัวอย่างแถบคำสั่งต่างๆที่สามารถใช้ได้	42
3.56 ตัวอย่างคำสั่งเงื่อนไขในการทำงานของ Blocks	43
3.57 การเดินสาย Line ออกจากปลั๊ก Tower	44
3.58 หลังจากเดินสาย Line ออกมาจากปลั๊ก Towerแล้ว	44
3.59 การยึดอุปกรณ์เข้ากับแท่นอะคริลิก	45
3.60 การเชื่อมต่อสายไฟเข้ากับ Relay 8 Channel โดยผ่านเทอร์มินอล.....	45
3.61 Hardware ทั้งหมดหลังจากเชื่อมต่อสายเข้าด้วยกันแล้ว	46
3.62 วงจรไฟฟ้ารวมของทั้งระบบ	46
4.1 อุปกรณ์ที่ใช้สั่งงานควบคุม.....	47
4.2 ตัวอย่างเร้าเตอร์ตัวส่งสัญญาณ.....	47
4.3 Tower ปลั๊กพร้อมชุดควบคุม.....	48
4.4 แสดงแอปพลิเคชัน MIT AI2 Companion	48
4.5 ภายในแอปพลิเคชัน MIT AI2 Companion.....	49
4.6 การโหลดแอปพลิเคชันที่สร้างจากหน้าเว็บไซต์ลง Smartphone	49
4.7 การโหลดแอปพลิเคชันที่สร้างจากหน้าเว็บไซต์ลง Smartphone.....	50
4.8 หน้าจอ Install ของแอปพลิเคชัน.....	50
4.9 หน้าจอแอปพลิเคชัน Screen1.....	51
4.10 หน้าจอแอปพลิเคชัน Screen2 และ Screen3.....	51
4.11 การทดลองควบคุมการเปิดปิดผ่าน Firebase	52
4.12 แสดงการทดลองควบคุมการเปิดปิดผ่าน Firebase (2).....	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และXของอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.13 แสดงการทดลองควบคุมการเปิดปิดผ่านแอปพลิเคชัน.....	53
4.14 แสดงการทดลองควบคุมการเปิดปิดผ่านแอปพลิเคชัน (2).....	53
4.15 แสดงการทดลองใช้การตั้งเวลาผ่านแอปพลิเคชัน (3).....	54
4.16 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า.....	55
4.17 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า (2).....	55
4.18 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า (3).....	56
4.19 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า (4).....	56



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญของปัญญาประดิษฐ์

ปัจจุบันเทคโนโลยีเข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ความก้าวหน้าของเทคโนโลยีใหม่มากมาย เป็นส่วนทำให้เกิดอุปสรรคอำนวยความสะดวกต่างๆ ก่อให้เกิดความสะดวกสบายและสนองความต้องการของมนุษย์เป็นอย่างดี ซึ่งเทคโนโลยีที่เกี่ยวข้องอินเทอร์เน็ตนั้นมีความหมายหลายแบบ และเป็นเทคโนโลยีที่จำเป็นในปัจจุบันเลยก็ว่าได้จึงเป็นอีกหนึ่งเหตุผลที่อยากศึกษาและจัดทำโครงการนี้ขึ้นมา โดยได้ทำการศึกษาเกี่ยวกับการควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าโดยตั้งเวลาเปิดปิดผ่านทางอินเทอร์เน็ต

ปัญญาประดิษฐ์นี้จัดทำเกี่ยวกับ ระบบตั้งเวลาเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต ซึ่งในปัจจุบันนี้ อินเทอร์เน็ตมีบทบาทกับชีวิตประจำวันเรามากขึ้น เราจึงสามารถใช้ระบบอินเทอร์เน็ตเข้ามาทำให้ชีวิตมีความสะดวกสบายมากยิ่งขึ้นจากการใช้งานเครื่องใช้ไฟฟ้าโดยสามารถที่จะตั้งเวลาเปิดปิดผ่านระบบอินเทอร์เน็ตได้ รวมถึงสามารถที่จะใช้งานได้ในยามที่ไม่มีสัญญาณอินเทอร์เน็ตได้ด้วย ปัญญาประดิษฐ์นี้สามารถตอบโจทย์และอำนวยความสะดวกได้จริง โดยการที่เปิด-ปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ตนั้น สามารถเปิดปิดได้จากทางเว็บไซต์และสมาร์ตโฟน และรวมถึงตัวบอร์ดปุ่มกดที่สามารถใช้ได้ในเวลาที่ไม่สามารถใช้งานในส่วนของระบบอินเทอร์เน็ตได้ ซึ่งมีอุปกรณ์หลักคือ บอร์ดของ Arduino Esp8266 และชุด Relay ซึ่งตัวบอร์ดของ Arduino และ Esp8266 จะเป็น Controller ในการสั่งงานให้ Relay ทำงานตามโปรแกรมที่ได้อัปโหลดลงไปใ้ในอุปกรณ์ทั้งสองแล้วนำอุปกรณ์ไฟฟ้าต่างๆที่ต้องการควบคุมมาเชื่อมต่อกับ Relay จากนั้นก็จะสามารถควบคุมเครื่องใช้ไฟฟ้าจากสถานที่ต่างๆได้ทุกที่ ขอเพียงแค่มียินเทอร์เน็ตเป็นสื่อกลางในการเชื่อมต่อเท่านั้น

1.2 วัตถุประสงค์ของปัญญาประดิษฐ์

1. เพื่อศึกษาระบบการควบคุมเครื่องใช้ไฟฟ้าผ่านทางอุปกรณ์ต่างๆ
2. สามารถตั้งเวลาเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านระบบอินเทอร์เน็ตได้
3. สามารถควบคุมการเปิด-ปิดของเครื่องใช้ไฟฟ้าโดยใช้ Arduino และ Esp8266 ในการควบคุม
4. สามารถนำความรู้ที่ได้ไปใช้พัฒนาในงานต่อไปได้

1.3 ขอบเขตของปัญญาประดิษฐ์

1. สามารถควบคุมการเปิด-ปิดของอุปกรณ์เครื่องใช้ไฟฟ้าโดยตั้งเวลาผ่านระบบอินเทอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้กับการใช้งานเพื่อการศึกษานี้เท่านั้น เมื่อผู้ใช้เห็นใบใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. ศึกษาไมโครคอนโทรลเลอร์ที่ใช้ในการควบคุม
2. ศึกษาอุปกรณ์ที่ใช้เชื่อมต่อกับอินเทอร์เน็ตและต่อเข้ากับไมโครคอนโทรลเลอร์ได้
3. ศึกษาการควบคุม Relay โดยการส่งค่าจากไมโครคอนโทรลเลอร์เพื่อเปิด-ปิดอุปกรณ์เครื่องใช้ไฟฟ้า
4. ศึกษาการเขียนคำสั่งให้ไมโครคอนโทรลเลอร์
5. ศึกษาการใช้งานอุปกรณ์ร่วมกัน
6. เชื่อมต่ออุปกรณ์ต่างๆในส่วนของฮาร์ดแวร์เข้าด้วยกัน
7. เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์เพื่อใช้ประมวลผลคำสั่ง
8. เขียนโปรแกรมให้อุปกรณ์ต่างๆสามารถเชื่อมต่อร่วมกันได้
9. ทำการทดสอบการใช้งานของระบบ
10. สรุปผลการดำเนินงาน
11. จัดทำรูปเล่มและนำเสนอโครงการ

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถเปิดปิดเครื่องใช้ไฟฟ้าโดยตั้งเวลาผ่านระบบอินเทอร์เน็ตได้
2. สามารถเขียนโปรแกรมควบคุมอุปกรณ์คอนโทรลเลอร์ต่างๆได้
3. มีความรู้ความเข้าใจในระบบการทำงาน การนำอุปกรณ์อิเล็กทรอนิกส์มาใช้งานร่วมกัน และสามารถนำความรู้ที่ได้ไปใช้ต่อยอดได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ IoT จะเปลี่ยนรูปแบบและกระบวนการผลิตในภาคอุตสาหกรรมไปสู่ยุคใหม่ หรือที่เรียกว่า Industry 4.0 ที่จะอาศัยการเชื่อมต่อสื่อสารและทำงานร่วมกันระหว่างเครื่องจักร มนุษย์ และข้อมูล เพื่อเพิ่มอำนาจในการตัดสินใจที่รวดเร็วและมีความถูกต้องแม่นยำสูง โดยที่ข้อมูลทั้งหลายที่เก็บจากเซ็นเซอร์ที่ใช้ตรวจวัดตัวอุปกรณ์และสภาพแวดล้อมจะถูกนำมาวิเคราะห์ ให้ได้ผลลัพธ์เพื่อนำไปปรับปรุงกระบวนการผลิตได้อย่างทันที นอกจากการข้ามขีดจำกัดเรื่องเวลาแล้ว ระบบควบคุมหรือระบบวิเคราะห์ข้อมูล อาจไม่ได้ขึ้นอยู่กับเครื่องจักร แต่สามารถควบคุมสั่งการได้โดยไร้ขีดจำกัดเรื่องสถานที่

เทคโนโลยีที่ทำให้ IoT เกิดขึ้นได้จริงและสร้างผลกระทบในวงกว้างได้ แบ่งออกเป็นสามกลุ่มได้แก่

1. เทคโนโลยีที่ช่วยให้สรรพสิ่งรับรู้ข้อมูลในบริบทที่เกี่ยวข้อง เช่น เซ็นเซอร์
2. เทคโนโลยีที่ช่วยให้สรรพสิ่งมีความสามารถในการสื่อสาร เช่น ระบบสมองกลฝังตัว รวมถึงการสื่อสารแบบไร้สายที่ใช้พลังงานต่ำ อาทิ Zigbee, 6LowPAN, Low-power Bluetooth
3. เทคโนโลยีที่ช่วยให้สรรพสิ่งประมวลผลข้อมูลในบริบทของตน เช่น เทคโนโลยีการประมวลผลแบบคลาวด์ และเทคโนโลยีการวิเคราะห์ข้อมูลขนาดใหญ่ หรือ Big Data Analytics

2.2 NETPIE (Network Platform for Internet of Everything)

NETPIE (Network Platform for Internet of Everything) คือ Cloud Platform ที่ถูกออกแบบและพัฒนาขึ้นเพื่ออำนวยความสะดวกให้เกิดการสื่อสารระหว่างอุปกรณ์หรือ Things ในเครือข่าย IoT โดยมีประโยชน์ต่อนักพัฒนาและอุตสาหกรรมไทย อาทิ NETPIE ช่วยให้อุปกรณ์สามารถคุยกันได้โดยผู้พัฒนาไม่ต้องกังวลว่า อุปกรณ์นั้นจะอยู่ที่ใด ทั้งในแง่ Physical และ Logical เพียงนำ NETPIE Library ไปติดตั้งในอุปกรณ์ NETPIE จะรับหน้าที่ดูแลการเชื่อมต่อให้ทั้งหมด ไม่ว่าจะอุปกรณ์นั้นจะอยู่ในเครือข่ายชนิดใด ลักษณะใด หรือแม้กระทั่งเคลื่อนย้ายไปอยู่ที่ใด ผู้พัฒนาสามารถตัดปัญหาความกังวลในการที่จะต้องมาออกแบบการเข้าถึงอุปกรณ์จากระยะไกล (Remote Access) ด้วยวิธีแบบเดิมๆ เช่น การใช้ Fixed Public IP หรือการตั้ง Port Forwarding ในเราท์เตอร์ หรือการต้องไปลงทะเบียนกับผู้ให้บริการ Dynamic DNS ซึ่งทั้งหมดล้วนมีความยุ่งยากและลดความยืดหยุ่นของระบบ ไม่นับว่า NETPIE ยังช่วยให้การเริ่มต้นใช้งานเป็นไปได้โดยง่ายโดยการออกแบบให้อุปกรณ์ถูกค้นพบและเข้าสู่บริการโดยอัตโนมัติ (Automatic Discovery, Plug and Play) NETPIE ถูกออกแบบให้มี Authorization/Access control ในระดับ Fine Grain กล่าวคือผู้พัฒนาสามารถออกแบบได้เองทั้งหมด เช่น สิ่งใดมีสิทธิคุยกับสิ่งใด สิ่งใดมีสิทธิหรือไม่-เพียงใดในการอ่านหรือเขียนข้อมูล และสิทธิเหล่านี้จะมีอายุเท่าใดหรือถูกเพิกถอนภายใต้เงื่อนไขใด เป็นต้น NETPIE มีสถาปัตยกรรมเป็น Cloud อย่างแท้จริงในทุกๆ ระดับของระบบ ทำให้เกิดความยืดหยุ่น และคล่องตัวสูงในการขยายตัว

นอกจากนี้ โมดูลต่างๆ ยังถูกออกแบบให้ทำงานแยกจากกันเพื่อให้เกิดสถานะ Loose Coupling และสื่อสารกันด้วยวิธีการ Asynchronous Messaging ช่วยให้แพลตฟอร์มมี Reliability สูง สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำไปใช้ซ้ำ และพัฒนาต่อเติมได้ง่าย ดังนั้นผู้พัฒนาไม่จำเป็นต้องกังวลกับการขยายตัวเพื่อรับโหลดที่เพิ่มขึ้นในระบบอีกต่อไป นอกจากนี้ทางเนคเทคจะเปิด NETPIE Library ในรูปแบบ Open-Source ให้นักพัฒนาสามารถนำไปปรับปรุงต่อให้ตรงกับความต้องการใช้งานโดยเปิดโอกาสให้นำไปใช้ในเชิงพาณิชย์ได้ โดยเนคเทคหวังที่จะให้เกิด Community ที่จะมาร่วมกันพัฒนาต่อยอดสร้างความเข้มแข็งให้กับวงการ IoT ของไทย

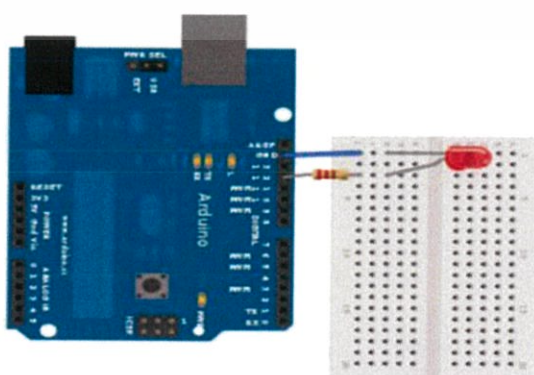
2.3 ไมโครคอนโทรลเลอร์ตระกูล AVR



รูปที่ 2.2 ตัวอย่าง Arduino [4]

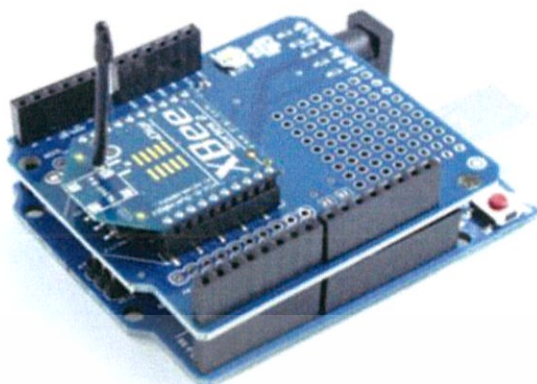
Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดหรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, 20Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 การต่อ Arduino เข้ากับ LED [5]

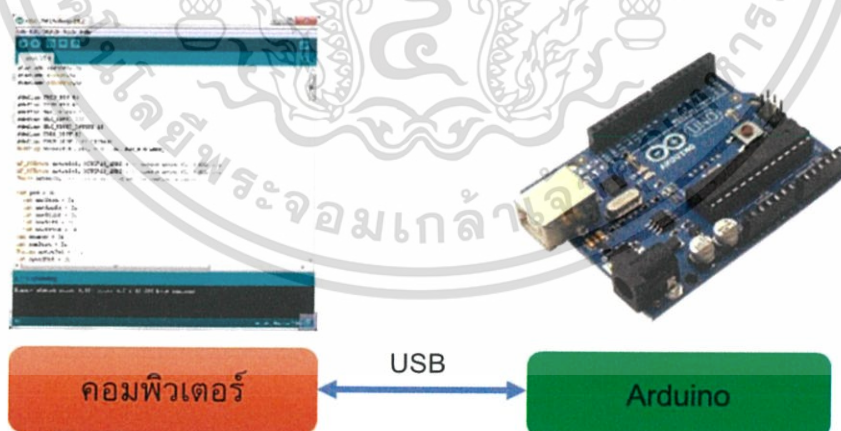


รูปที่ 2.4 การที่ Arduino ต่อกับ XBee Shield [6]

2.3.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

1. ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
2. มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
3. Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้านราคาไม่แพง
4. Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

2.3.2 รูปแบบการเขียนโปรแกรมบน Arduino

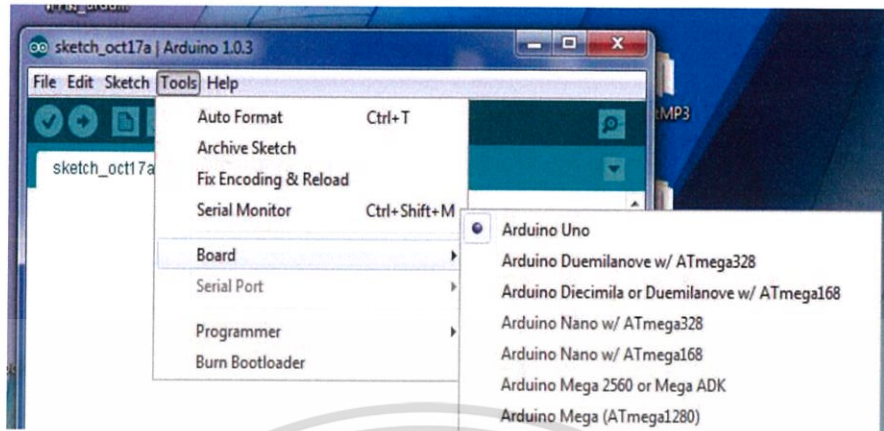


รูปที่ 2.5 การเชื่อมต่อของ Arduino [7]

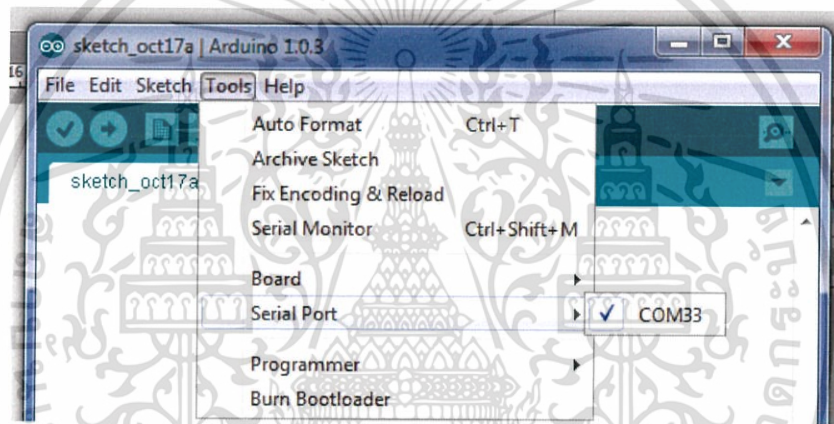
1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ Users เลือกบอร์ด Arduino ที่ใช้ และหมายเลข Com Port

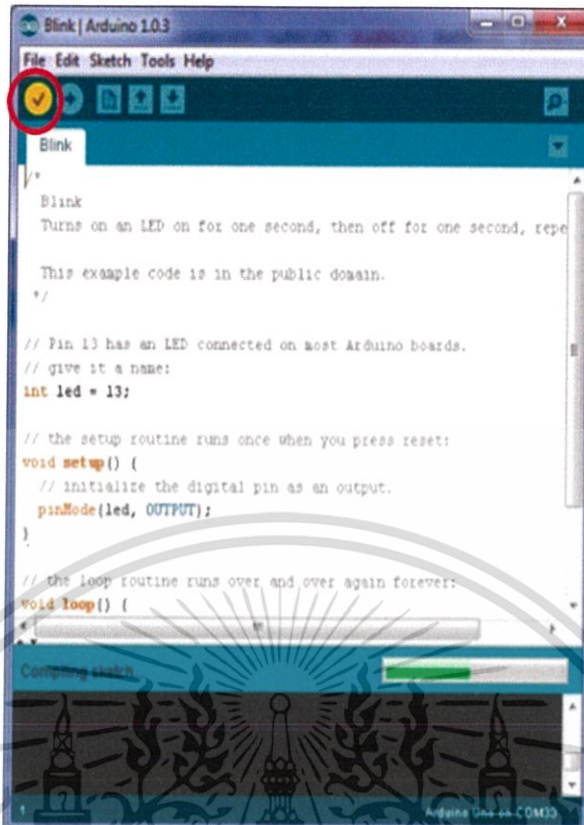


รูปที่ 2.6 เลือกบอร์ด Arduino ที่ต้องการ Upload

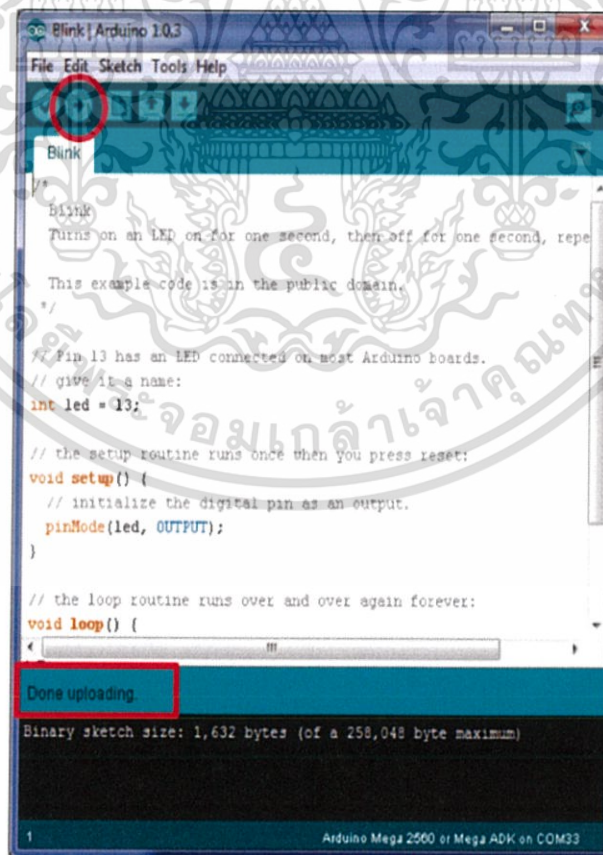


รูปที่ 2.7 การเลือกหมายเลข Com Port ของบอร์ด

3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done Uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



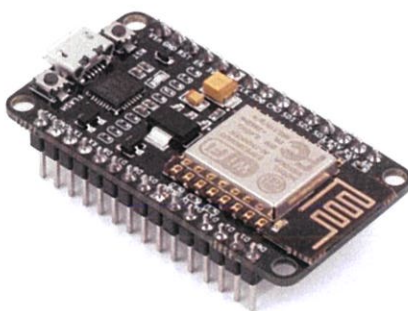
รูปที่ 2.8 การตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม



รูปที่ 2.9 การ Upload โค้ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

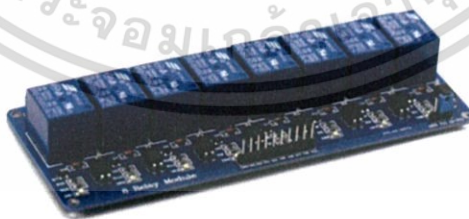
2.4 Node MCU8266 Module Wifi



รูปที่ 2.10 ตัวอย่างของ Node MCU8266 [8]

NodeMCU คือ แพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจกต์ Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (Software บนบอร์ด) ที่เป็น Open Source สามารถเขียนโปรแกรมด้วยภาษา Lua ได้ ทำให้ใช้งานได้ง่ายขึ้น มาพร้อมกับโมดูล WiFi (ESP8266) ซึ่งเป็นหัวใจสำคัญในการใช้เชื่อมต่อกับอินเทอร์เน็ตนั่นเอง ตัวโมดูล ESP8266 นั้นมีอยู่ด้วยกันหลายรุ่นมาก ตั้งแต่เวอร์ชันแรกที่เป็น ESP-01 ไล่ไปเรื่อยๆจนถึงปัจจุบันมีถึง ESP-12 NodeMCU นั้นมีลักษณะคล้ายกับ Arduino ตรงที่มีพอร์ต Input Output Built in มาในตัว สามารถเขียนโปรแกรมคอนโทรลอุปกรณ์ I/O ได้โดยไม่ต้องผ่านอุปกรณ์อื่นๆ และปัจจุบันได้พัฒนาสามารถทำให้ Arduino IDE ใช้งานร่วมกับ Node MCU ได้ จึงทำให้ใช้ภาษา C/C++ ในการเขียนโปรแกรมได้ ทำให้เราสามารถใช้งานได้หลากหลายมากยิ่งขึ้น NodeMCU ตัวนี้สามารถทำอะไรได้หลายอย่างมากโดยเฉพาะเรื่องที่เกี่ยวข้องกับ IoT ไม่ว่าจะเป็นการทำ Web Server ขนาดเล็ก การควบคุมการเปิดปิดไฟผ่าน WiFi และอื่นๆอีกมากมาย

2.5 Relay 8 Channel



รูปที่ 2.11 ตัวอย่างของ Relay 8 channel [9]

รีเลย์ 8 ตัว เพื่อใช้งานในการควบคุมอุปกรณ์ไฟฟ้า รับกระแสได้สูงถึง 10 A ใช้งานได้ทั้งไฟฟ้ากระแสตรง และ กระแสสลับ รับแรงดันระดับ 5 V ตรงจาก Arduino Board มี LED แสดงสถานะการทำงานของรีเลย์ ออกแบบให้ป้องกันวงจรด้านควบคุมออกจากด้านกำลังโดยการใช้การส่งผ่านด้วยแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน Firebase การทำแอปพลิเคชัน และการดำเนินการด้านฮาร์ดแวร์

ในบทนี้จะมีเนื้อหาเกี่ยวกับวิธีการนำ Firebase มาใช้งานอย่างละเอียด ขั้นตอนการสร้างแอปพลิเคชันบน Smartphone รวมถึงการดำเนินงานทางด้านฮาร์ดแวร์ที่จะต้องนำมาใช้งานร่วมกับ Firebase รวมถึงแอปพลิเคชันด้วย

3.1 การใช้งาน Firebase

Firebase มีบริการหลักเป็น Realtime Database เกิดขึ้นด้วยแนวคิดที่คนทำแอปพลิเคชันไม่จำเป็นต้องตั้งเซิร์ฟเวอร์เอง ซึ่งโดยปรกติจะทราบดีว่าการทำเว็บไซต์ ทำแอปพลิเคชัน จะต้องมีความรู้พื้นฐานข้อมูล และจะต้องมีการเก็บตารางของผู้ใช้งาน ระบบ Log ต่างๆ มีการติดต่อกับผู้ใช้ ซึ่งเป็นงานที่มีการทำซ้ำๆ ตลอด ดังนั้น Firebase จึงมาช่วยแก้ปัญหานี้ได้ โดยตัว Firebase ทำไว้ให้หมดแล้ว ในงานด้านแอปพลิเคชัน ตัว Firebase ถือเป็นบริการฐานข้อมูลออนไลน์ตัวหนึ่ง ซึ่งแอปพลิเคชันส่วนใหญ่ต้องใช้งานฐานข้อมูลตรงส่วนนี้

แต่หากมองในมุมมองของ IoT ตัว Firebase ถือว่าเป็นตัวกลางการเชื่อมต่อทุกอุปกรณ์เข้าด้วยกันได้ โดยมีจุดเด่นคือ เรียลไทม์ และสามารถบันทึกข้อมูลไว้ได้

Firebase คือฐานข้อมูลประเภท NoSQL (Not Only SQL) NoSQL คือ แนวทางหนึ่งสำหรับจัดการข้อมูลและการออกแบบฐานข้อมูลสำหรับข้อมูลขนาดใหญ่ ซึ่งอยู่อย่างกระจัดกระจายหลากหลายรูปแบบ

ฐานข้อมูลชนิด NoSQL จะไม่ใช่ภาษา SQL ในการจัดการข้อมูล และออกแบบให้มีความยืดหยุ่น และเน้นความเร็วในการใช้งานมากที่สุด ฐานข้อมูล NoSQL ที่นิยมใช้งานในปัจจุบันคือ MongoDB ซึ่งมีการเก็บข้อมูลเป็นชนิด JSON มีตารางเหมือนเดิม แต่ไม่มีคอลัมน์ข้อมูลที่ตายตัว ใน 1 แถว สามารถเก็บข้อมูลได้ทั้งข้อความ (String) ตัวเลข (Number) และอื่นๆ รวมไปถึงอาร์เรย์ และออบเจกต์

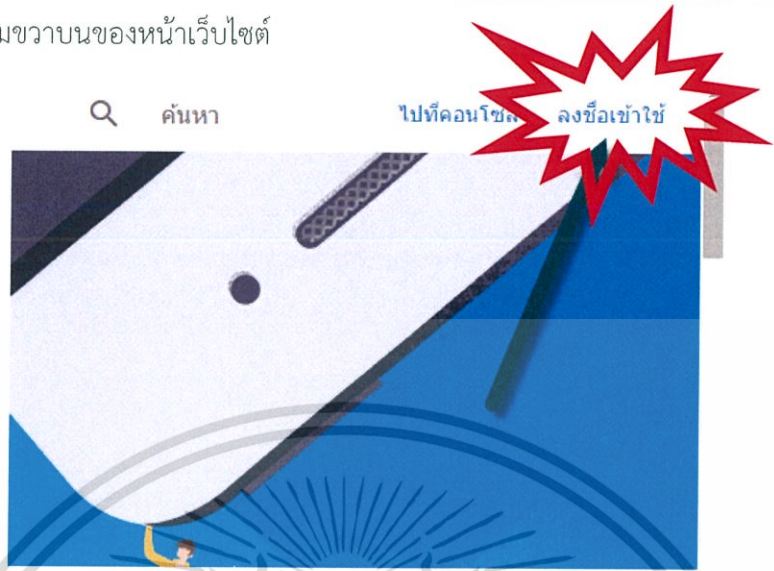
Firebase กับ API เพื่ออุปกรณ์ IoT

Firebase มี API ของหลายภาษาให้เลือกใช้งาน ทั้งภาษา Python (นิยมใช้ใน Embedded OS) JavaScript (ในบนหน้าเว็บไซต์) และรวมไปถึงใน ESP8266 ที่ใช้ Arduino IDE ด้วย

Google ได้จัดทำไลบรารี Firebase สำหรับ Arduino (ซึ่งใช้กับ ESP8266 ได้) ไว้บน GitHub

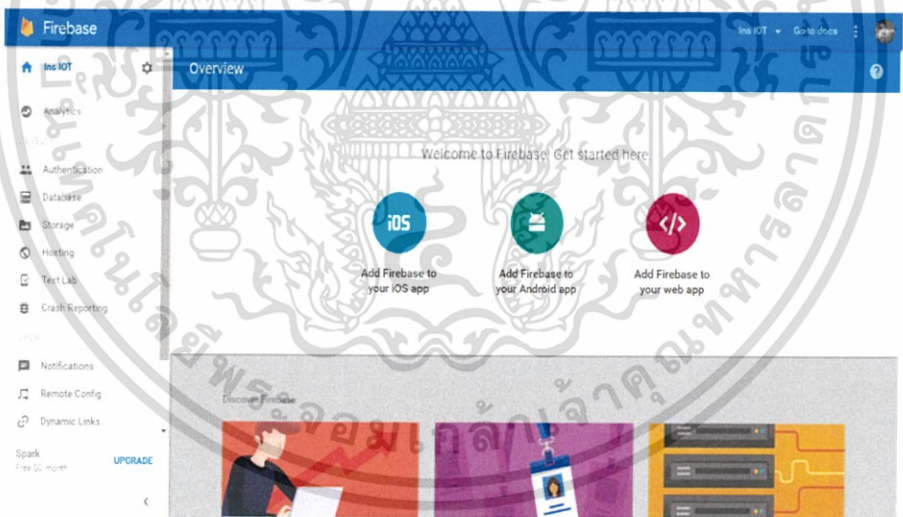
3.1.1 เตรียม Firebase ให้พร้อมใช้งาน

1. เข้าไปที่หน้าหลักเว็บไซต์ของ Firebase <https://firebase.google.com/> [10] กด ลงชื่อเข้าใช้ ที่ มุมขวาบนของหน้าเว็บไซต์



รูปที่ 3.1 ขั้นตอนการลงชื่อเข้าใช้ Firebase

2. เนื่องจาก Firebase ถูกพัฒนาโดย Google จึงต้องใช้บัญชี Google ในการเข้าสู่ระบบ กรอกรหัสและลงชื่อเข้าใช้ให้เรียบร้อย



รูปที่ 3.2 ขั้นตอนการลงชื่อเข้าโดยใช้บัญชีgoogle

3. จากนั้นจะดึงกลับมาที่หน้าหลัก ให้กดไปที่คอนโซล ในเมนูมุมขวามือ กด CREATE NEW PROJECT

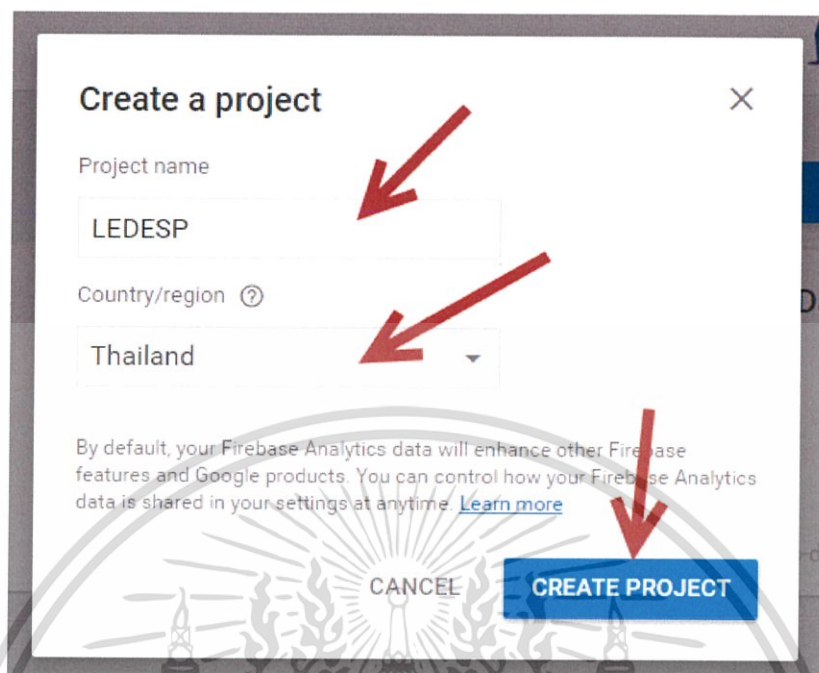


รูปที่ 3.3 ขั้นตอนการสร้างโปรเจค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในช่องแรกให้ตั้งชื่อโปรเจกใหม่ และในช่องที่ 2 เลือกเป็น Thailand แล้วกด

CREATE PROJECT



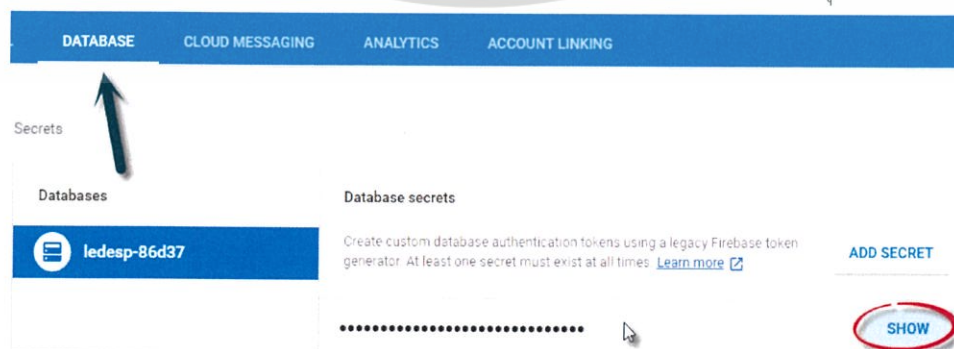
รูปที่ 3.4 ขั้นตอนการระบุชื่อโปรเจก และ ชื่อประเทศ

5. กดไปที่รูปเฟืองมุมซ้ายบน แล้วคลิกที่เมนู Project Settings



รูปที่ 3.5 ขั้นตอนการติดตั้งโปรเจกใหม่

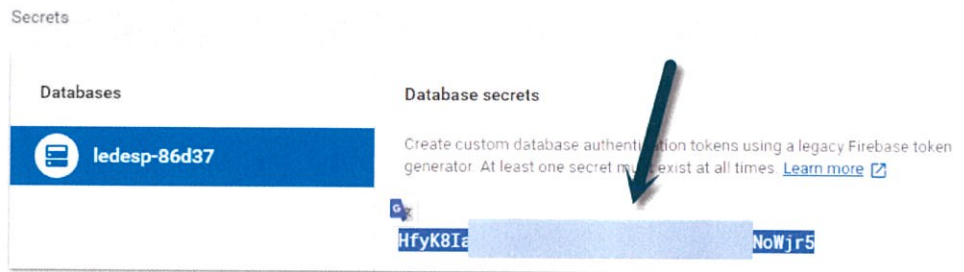
6. คลิกที่แท็บ DATABASE นำเมาส์เลื่อนไปตรงช่อง กดที่ปุ่ม SHOW



รูปที่ 3.6 ขั้นตอนการติดตั้งโปรเจกใหม่ (2)

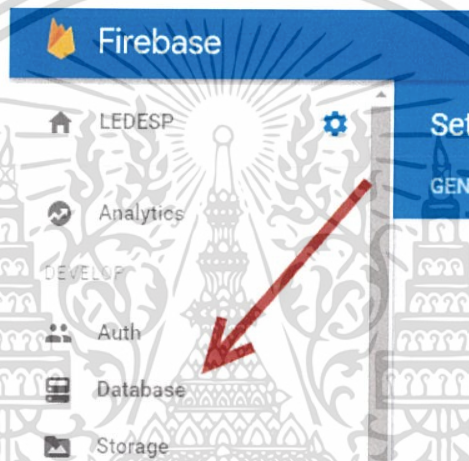
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. คัดลอก Key เก็บไว้ ตัว Key นี้จะใช้ในการคอนฟิกในโค้ด เพื่อให้สามารถเข้ามาแก้ไขข้อมูลในฐานข้อมูล Firebase ได้



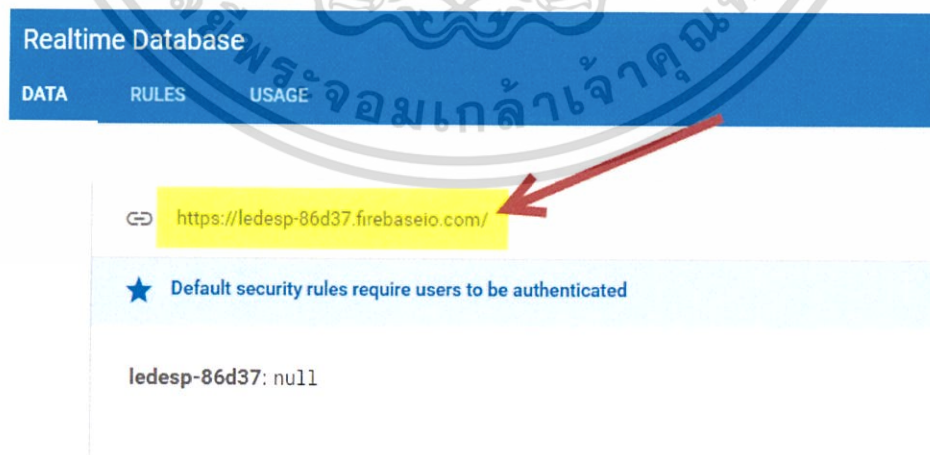
รูปที่ 3.7 ขั้นตอนการติดตั้งโปรเจกใหม่ (3)

8. กดไปที่เมนู Database ในเมนูด้านซ้าย



รูปที่ 3.8 ขั้นตอนการติดตั้งโปรเจกใหม่ (4)

9. คัดลอกลิงค์ตามตำแหน่งในรูปเก็บไว้ ลิงค์นี้จะเอาไว้ใช้คอนฟิกในโค้ดเช่นเดียวกัน



รูปที่ 3.9 ขั้นตอนการคัดลอกลิงค์เพื่อใช้ในการคอนฟิก

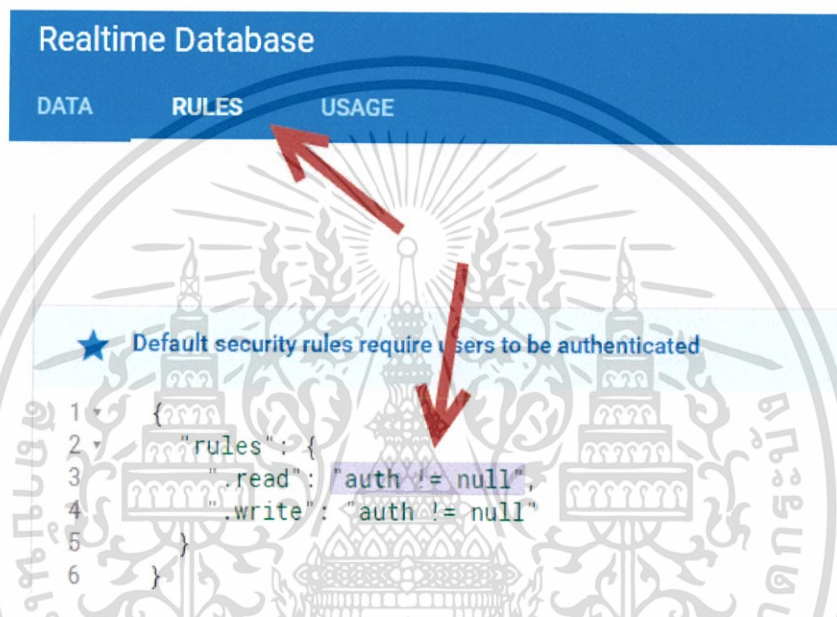
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การกำหนดสิทธิ์การเข้าถึงข้อมูล

ส่วนนี้เป็นส่วนเสริม และมีความสำคัญหากนำไปใช้งานจริง เนื่องจาก Firebase ได้กำหนดค่าเริ่มต้นให้ต้องมี Key เท่านั้น จึงจะสามารถอ่าน - เขียนฐานข้อมูลได้ ซึ่งจะมีปัญหาหากนำไปใช้งานบนหน้าเว็บไซต์ เพราะในหน้าเว็บจะไม่สามารถเปิดเผย Key ได้

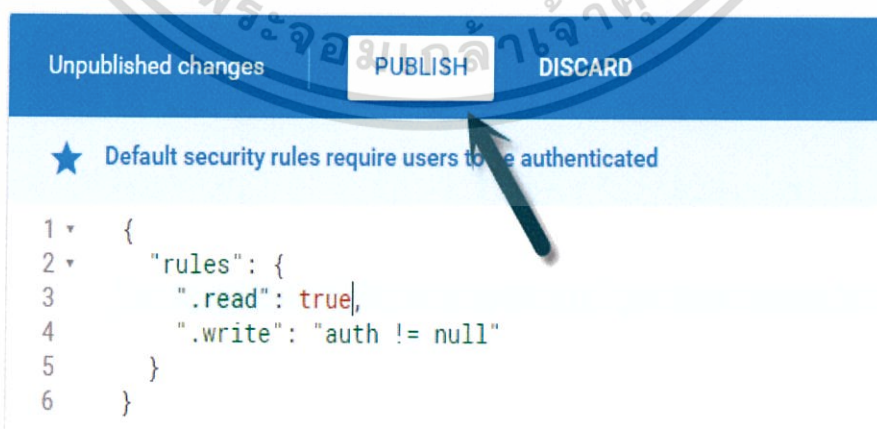
การเปิดเผย Key จะทำให้ Users สามารถเข้ามาแก้ไขข้อมูลในฐานข้อมูลได้ ดังนั้นจึงควรกำหนดให้ในการเขียนข้อมูลลงไปฐานข้อมูล สามารถทำได้เฉพาะมี Key และหากมี หรือไม่มี Key จะอ่านได้เท่านั้น ซึ่งสามารถกำหนดได้ดังนี้

1. กดไปที่แท็บ RULES ในส่วนที่คลุมแถบคำสั่งในรูป ให้ใส่ True ลงไปแทน



รูปที่ 3.10 ขั้นตอนการกำหนดสิทธิ์การเข้าถึงข้อมูล

2. เมื่อแก้ไขแล้ว ให้กดปุ่ม PUBLISH เพื่อเป็นการบันทึกค่า

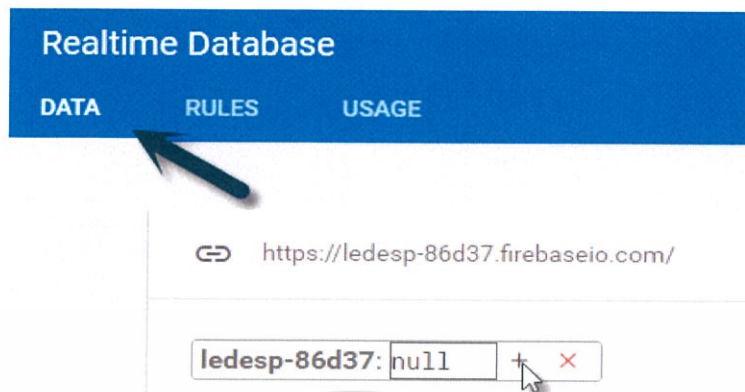


รูปที่ 3.11 ขั้นตอนการกำหนดสิทธิ์การเข้าถึงข้อมูล (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

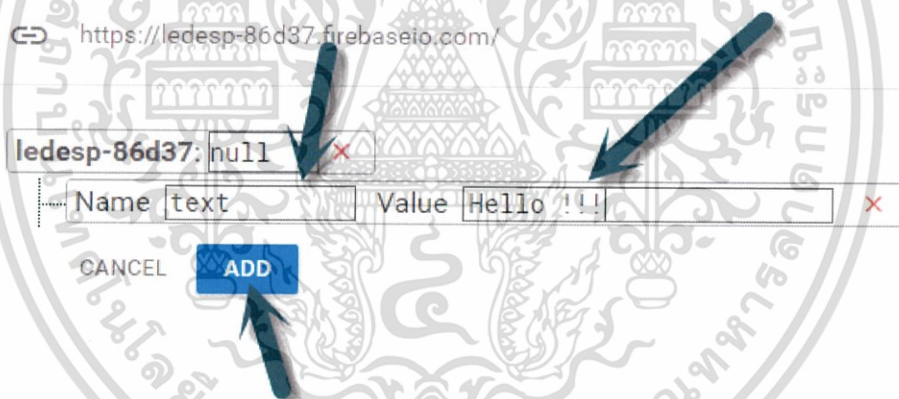
3.1.3 การใช้งานคอนโซลของ Firebase

กดกลับมาที่แท็บ DATA นำเมาส์เลื่อนไปที่ชื่อออปเจกแรก เครื่องหมาย + จะปรากฏขึ้นมา



รูปที่ 3.12 ขั้นตอนการใช้งานคอนโซลของ Firebase

1. กดที่เครื่องหมาย + จะมีชื่อขึ้นมา 2 ช่อง และเยื้องทางซ้ายเล็กน้อย การเยื้องทางซ้ายเพื่อให้ทราบได้ว่าเรากำลังจะเพิ่มข้อมูลลงในออปเจกใด ออปเจกที่อยู่ในระดับเดียวกันจะมีการเยื้องซ้ายเท่ากัน ในช่อง Name จะเป็นชื่อของออปเจก (Index, Key Name) ส่วน Value ทดลองใส่ Hello !!! แล้วกดปุ่ม ADD



รูปที่ 3.13 ขั้นตอนการเพิ่มข้อมูลลงในออปเจก

2. จากนั้นในระดับที่สูงกว่าทั้งหมดจะเปลี่ยนสีเป็นสีส้ม และออปเจกที่เพิ่มเข้ามาใหม่ (ในที่นี้คือ Text) จะเป็นสีเขียว เมื่อผ่านไปช่วงเวลาหนึ่งสีจะกลับมาเป็นสีดำดังเดิม

<https://ledesp-86d37.firebaseio.com/>

ledesp-86d37
text: "Hello !!!"

รูปที่ 3.14 เมื่อเพิ่มข้อมูลเสร็จสิ้นแล้ว

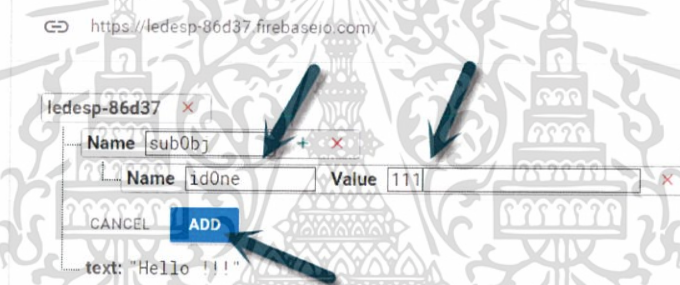
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ที่ออปเจกแรกสุด (อยู่สูงที่สุด) กด + อีกครั้ง พิมพ์ Name เป็น subObj แล้วกด เครื่องหมาย + ตามลูกศรชี้ในรูป



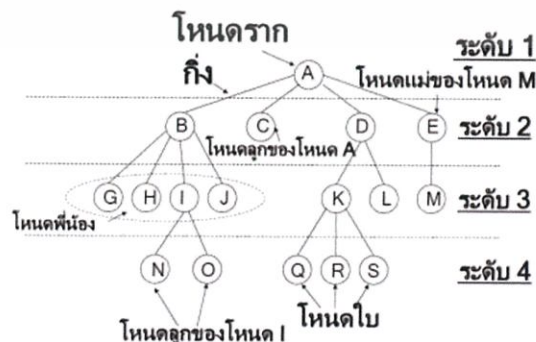
รูปที่ 3.15 การกำหนดชื่อในออปเจก

4. กรอก Name เป็น idOne แล้วกด Value เป็น 111 จากนั้นกด ADD จะเห็นได้ว่า ผลคือ ออปเจก ledesp-86d37 จะอยู่ในระดับสูงที่สุด มีออปเจก subObj เป็นออปเจกย่อย และมี idOne ซึ่งมีค่าเป็น 111 เป็นออปเจกย่อยของ subObj อีกที



รูปที่ 3.16 การกำหนดชื่อในออปเจกย่อย

5. หากนำไปเทียบกับเนื้อหาในวิชา โครงสร้างข้อมูล ซึ่งเป็นวิชาพื้นฐานที่ โปรแกรมเมอร์จะต้องเรียน จะพบว่าการทำงานแบบนี้จะคล้าย ๆ กับโครงสร้างข้อมูลแบบต้นไม้ ให้ มอง ledesp-86d37 เป็นโหนดราก มอง subObj เป็นโหนดลูกของโหนด edesp-86d37 และโหนด idOne เป็นโหนดลูกของโหนด subObj อีกที ซึ่งในโหนดใบแต่ละโหนด (โหนดที่ไม่มีโหนดลูก) สามารถมีค่าเป็นข้อความ ตัวอักษร หรือตัวเลขก็ได้



รูปที่ 3.17 รูปแบบโครงสร้างข้อมูลแบบต้นไม้ [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. นำเมาส์เลื่อนไปตรงออปเจกที่ชื่อ text แล้วคลุมแถบดำ ทดลองแก้เป็น 1212312121 แล้วกดปุ่ม Enter บนคีย์บอร์ด



รูปที่ 3.18 การทดลองเปลี่ยนชื่อในส่วนของ text

7. ผลที่ได้คือออปเจกที่ถูกแก้ไขจะเปลี่ยนเป็นสีส้ม แล้วออปเจกที่อยู่สูงกว่าทั้งหมด ก็เปลี่ยนเป็นสีส้มเช่นกัน



รูปที่ 3.19 การทดลองเปลี่ยนชื่อในส่วนของ text (2)

8. นำเมาส์ไปวางที่ออปเจก subObj จากนั้นกด X

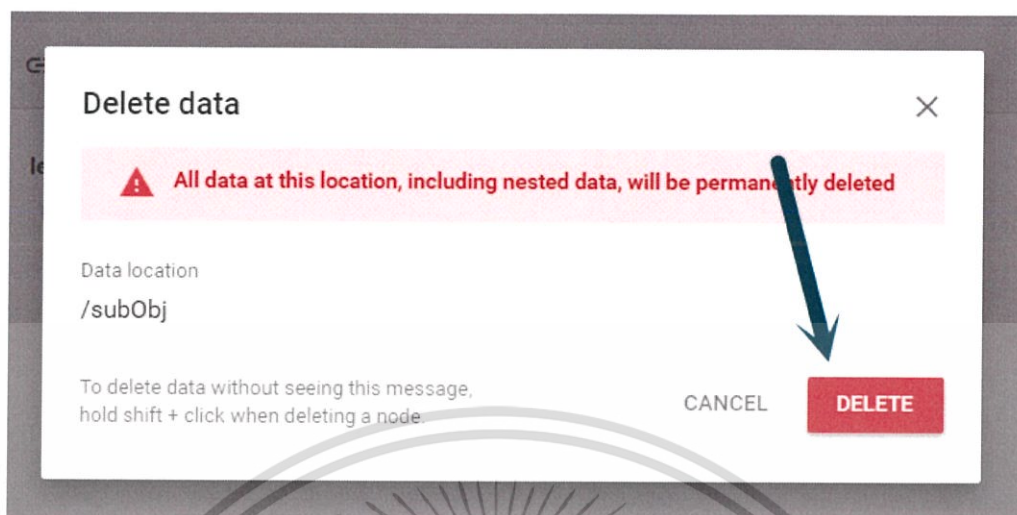


รูปที่ 3.20 การลบส่วนออปเจกย่อย subObj

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

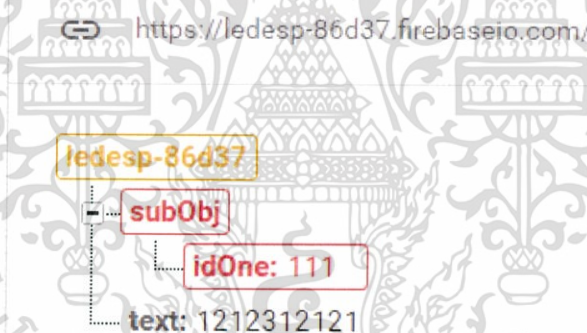
9. จะแจ้งว่าหากลบออปเจคนี้ไปแล้ว ออปเจที่อยู่ล่าง ๆ ก็จะถูกลบไปด้วย ให้กดปุ่ม

DELETE



รูปที่ 3.21 การลบส่วนออปเจย่อย subObj (2)

10. จากนั้นออปเจตั้งแต่ subObj ลงไปจะเปลี่ยนเป็นสีแดง หมายถึงออกเจคนั้นไม่มีอยู่แล้วและออปเจที่อยู่สูงกว่าจะเปลี่ยนเป็นสีส้ม หมายถึงมีการอัปเดตข้อมูล



รูปที่ 3.22 การลบส่วนออปเจย่อย subObj (3)

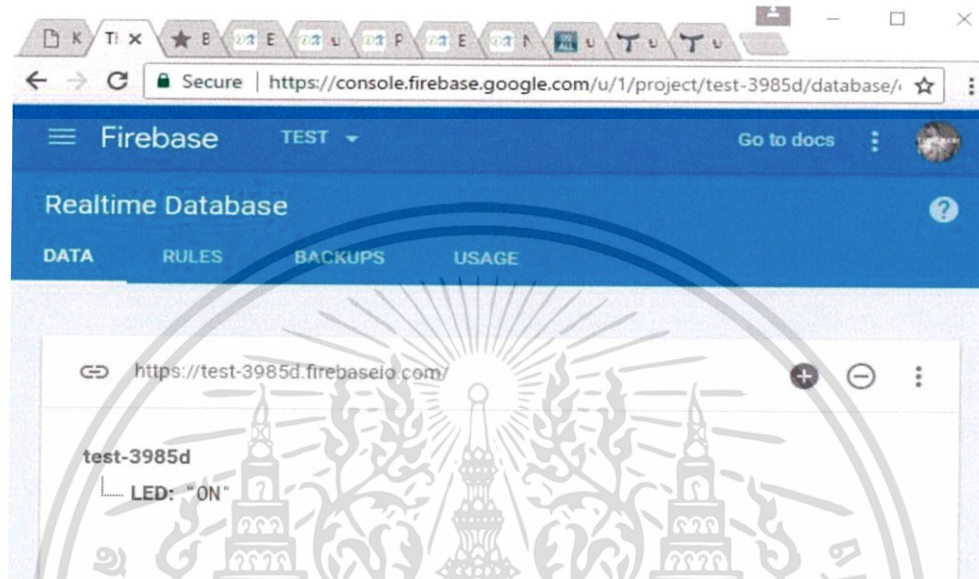
การใช้งานคอมโซลก็มีเพียงเท่านี้ ในตอนแก้ไขขอให้แก้ไขแล้วกด Enter ทันที เพราะหากไม่ทำการกด Enter ข้อมูลจะไม่ถูกแก้ไข และต้องเข้าใจไว้เสมอว่าหากลบออปเจที่อยู่ในระบบดับสูงๆ ก็จะทำให้ออปเจ ระดับต่ำลงมาถูกลบไปด้วย และไม่สามารถกู้คืนกลับมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

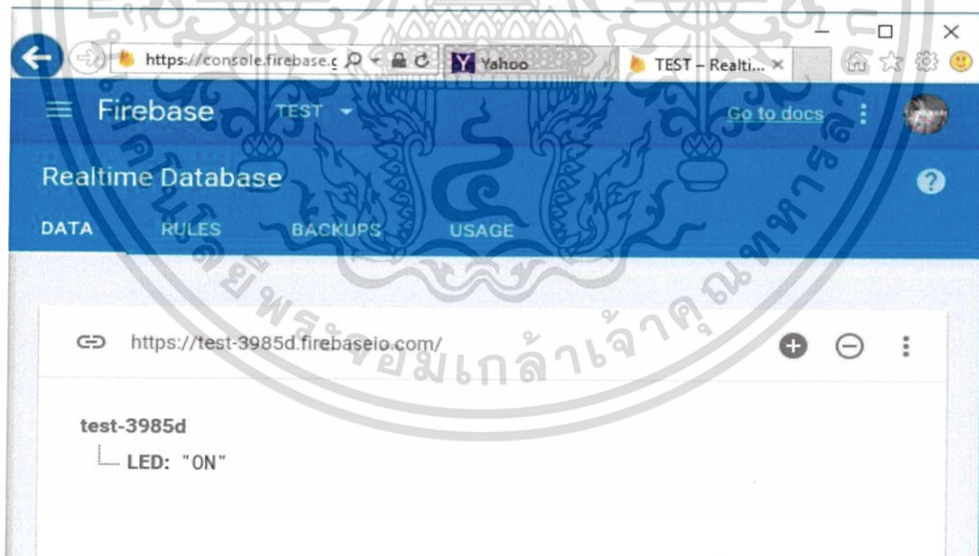
3.1.4 ทำความเข้าใจกับ Realtime Database

Realtime Database หมายถึงฐานข้อมูลที่มีเมื่อมีการอัปเดตข้อมูล จะมีการส่งข้อมูลที่อัปเดตนั้น ไปที่อุปกรณ์อื่นๆ ที่เชื่อมต่ออยู่ด้วย ดังตัวอย่างด้านล่างนี้

1. ในรูปด้านล่างนี้ รูปที่ 3.23 ได้เพิ่มแอปเจตที่มี Key เป็น LED และมีค่าเป็น ON เมื่อกดที่ปุ่ม ADD จะมีค่าของ LED : ON ในแต่ละ BROWSER พร้อมกัน



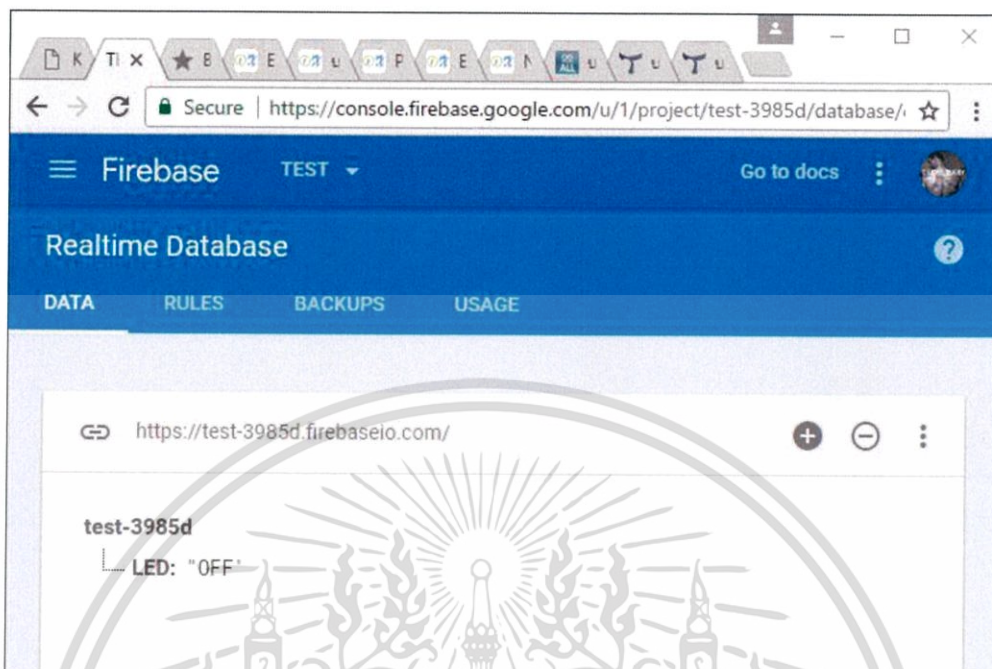
รูปที่ 3.23 การเพิ่มแอปเจตที่มี Key เป็น LED และมีค่าเป็น ON ในคอมพิวเตอร์ตัวที่ 1



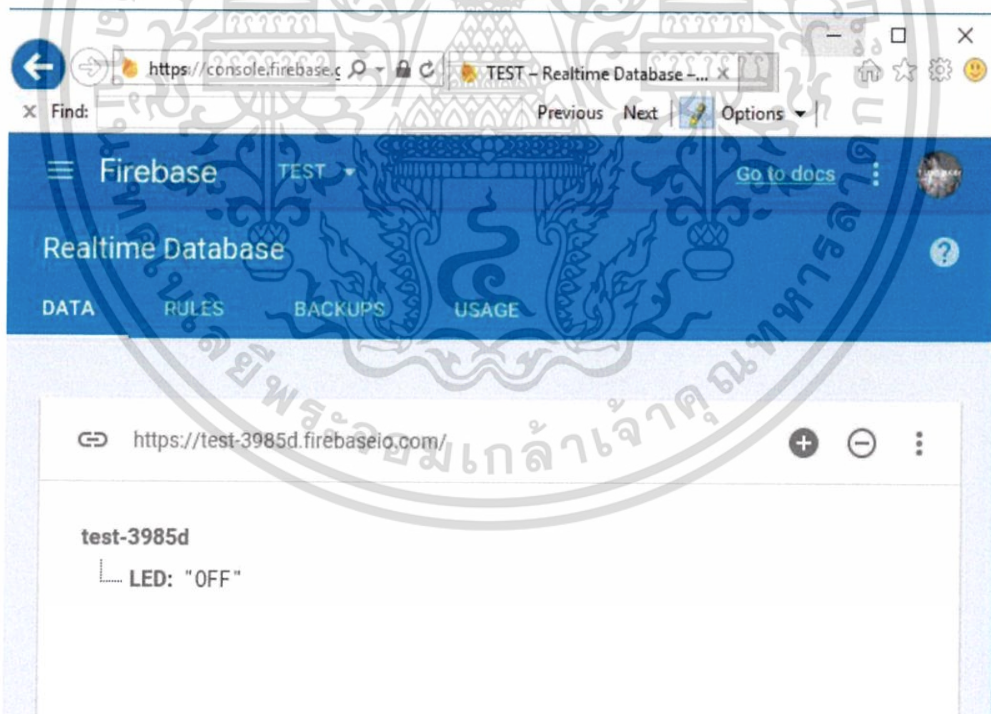
รูปที่ 3.24 การเปลี่ยนแปลงของ Firebase ในคอมพิวเตอร์ตัวที่ 2 เมื่อเพิ่มแอปเจตที่มี Key เป็น LED และมีค่าเป็น ON ในคอมพิวเตอร์ตัวที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ต่อมาแก้ไขข้อมูลของแอปเจค LED ในหน้าเว็บรูปที่ 3.25 เป็น OFF ผลที่ได้คือในหน้าต่างทั้ง 2 ด้านของ 2 BROWSER มีการเปลี่ยนแปลงทันที



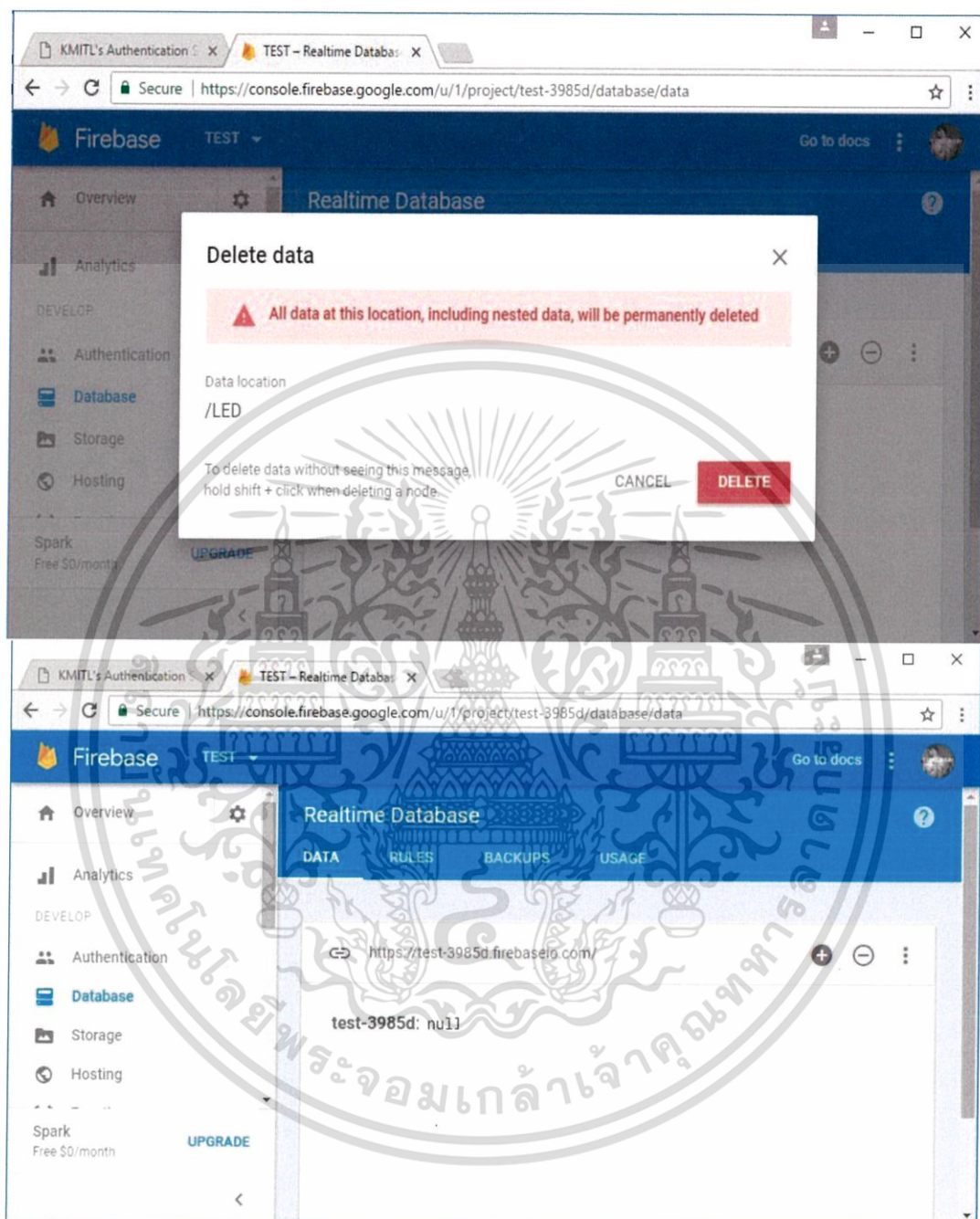
รูปที่ 3.25 การเพิ่มแอปเจคที่มี Key เป็น LED และมีค่าเป็น OFF ในคอมพิวเตอร์ตัวที่ 1



รูปที่ 3.26 การเปลี่ยนแปลงของ Firebase ในคอมพิวเตอร์ตัวที่ 2 เมื่อเพิ่มแอปเจคที่มี Key เป็น LED และมีค่าเป็น OFF ในคอมพิวเตอร์ตัวที่ 1

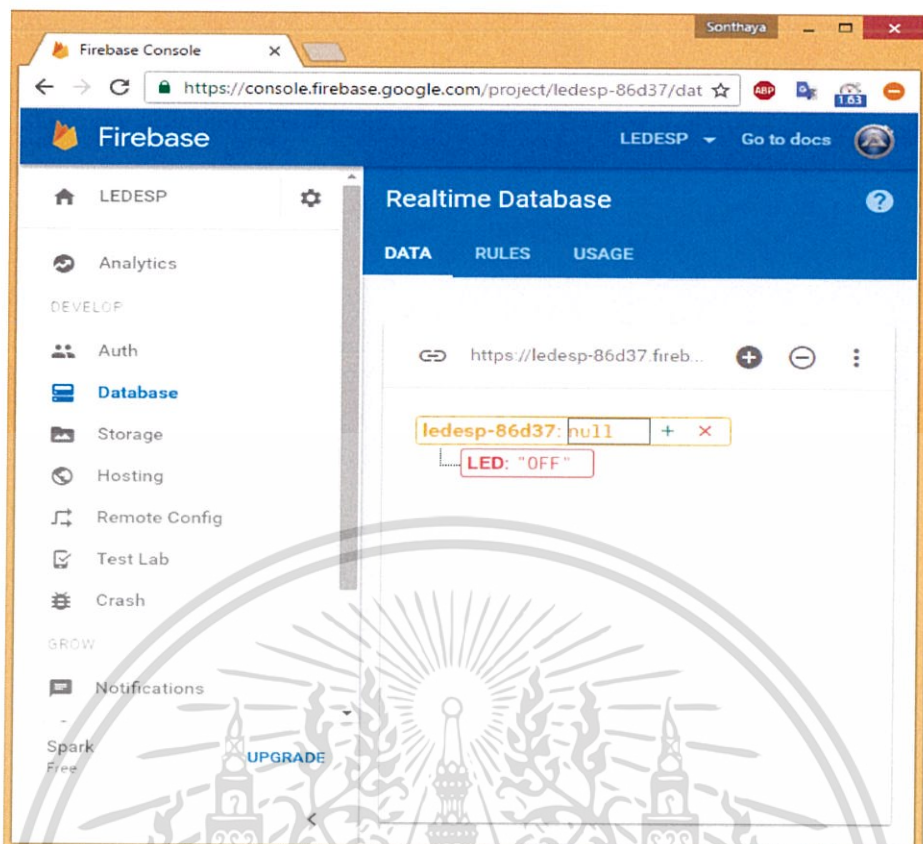
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สุดท้ายกดลบออกเจด LED ออกไป ผลคือออกเจด LED หายไปทันทีจากทั้ง 2 หน้า (สีแดงหมายถึงออกเจดถูกลบไปแล้ว)

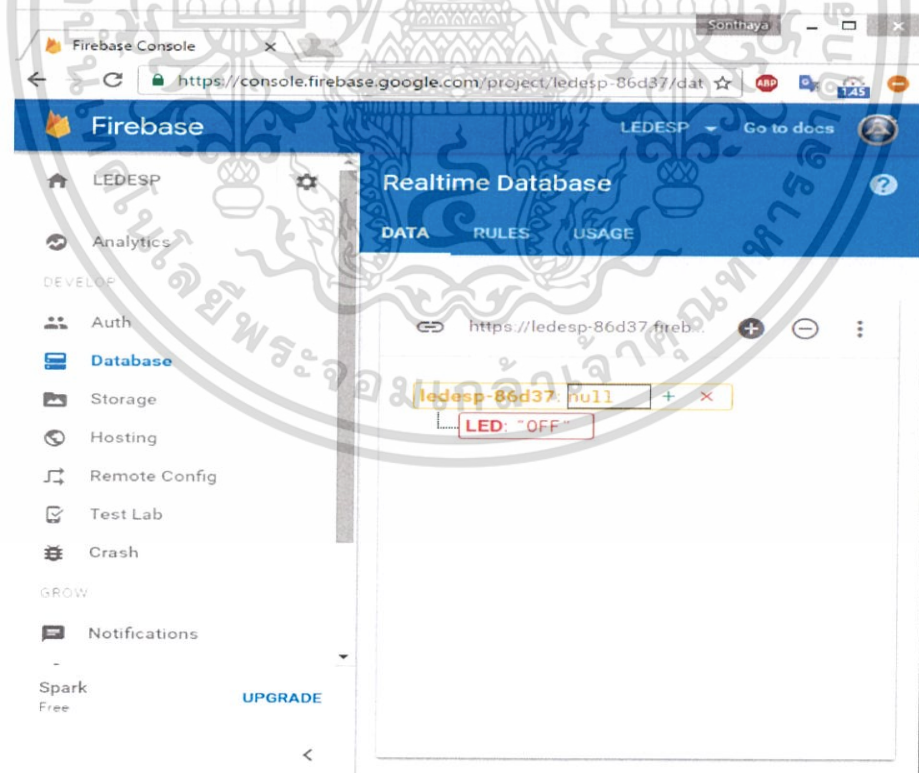


รูปที่ 3.27 การลบออกเจด LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 การลบบออปเจค LED ถูกลบไปแล้วบนคอมพิวเตอร์ตัวที่ 1



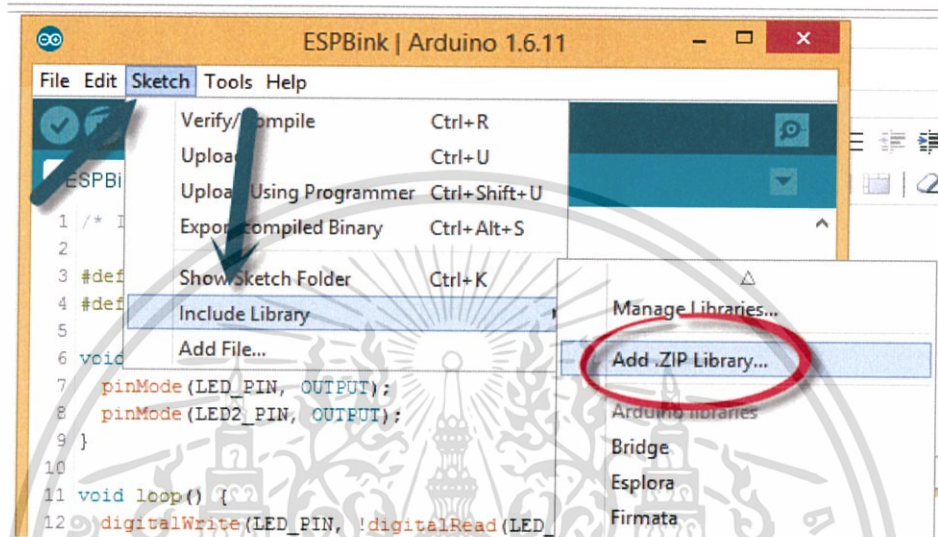
รูปที่ 3.29 การลบบออปเจค LED ถูกลบไปแล้วบนคอมพิวเตอร์ตัวที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 Firebase กับ ESP8266

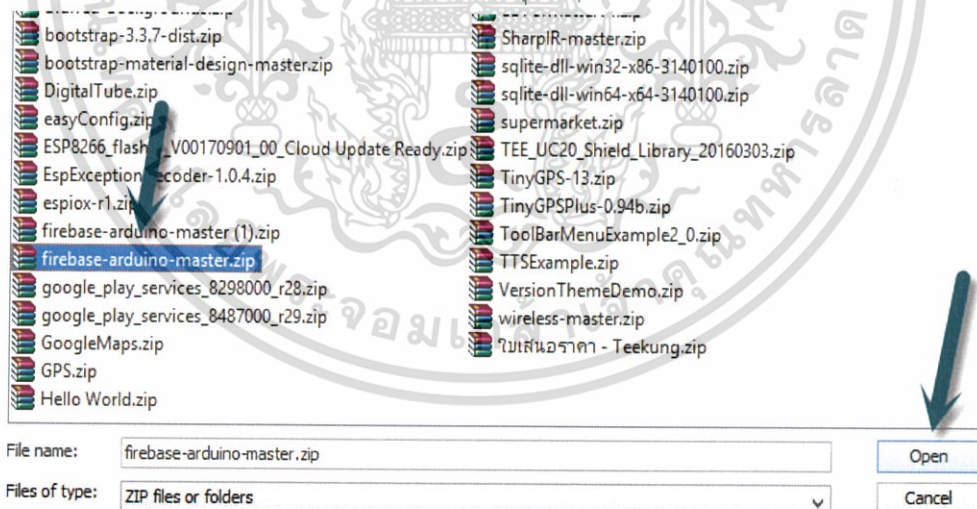
ทดลองให้ ESP8266 อัปเดตข้อมูลในฐานข้อมูลของ Firebase ก่อนอื่นให้ลงไลบรารีก่อน โดยดาวน์โหลดได้ที่ <https://github.com/googlesamples/firebase-arduino/archive/master.zip>

1. เปิดโปรแกรม Arduino ขึ้นมา กดไปที่ Sketch > Include Library > Add .ZIP Library...



รูปที่ 3.30 รูปแบบของโปรแกรม ESP Blink

2. เลือกไฟล์ที่ได้ดาวน์โหลดไว้ กดปุ่ม Open



รูปที่ 3.31 การเปิดไฟล์ในโปรแกรม ESP Blink

3. เมื่ออัปเดตโค้ดเข้าไปแล้ว และรอจนกว่าจะเชื่อมต่อ WiFi เสร็จ ESP8266 จะ Set ออปเจกต์ Number ให้มีค่าเพิ่มขึ้นเรื่อยๆ ทุกๆ ครั้งวินาที หากเปิดหน้าคอนโซลร่วมกับหน้ามอนิเตอร์จะเห็นผลที่ชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1 #include <ESP8266WiFi.h>
2 #include <FirebaseArduino.h>
3
4 // Config Firebase
5 #define FIREBASE_HOST "ledesp-86d37.firebaseio.com"
6 #define FIREBASE_AUTH "HfyK8Ia851Ufrkc7BWN0Wjr!"
7
8 // Config connect WiFi
9 #define WIFI_SSID "MaxHome3BB"
10 #define WIFI_PASSWORD "4321"
11
12 int i = 0;
13
14 void setup() {
15   Serial.begin(9600);
16
17   WiFi.mode(WIFI_STA);
18   // connect to wifi.

```

Done uploading ?

Invalid library found in C:\Users\Max\Documents\Arduino\libraries\
Invalid version found: 1.04
Invalid library found in C:\Users\Max\Documents\Arduino\libraries\

43 Generic ESP8285 Module, 80 MHz, 256000, 1M (512K SPIFFS) on COM3

รูปที่ 3.32 การอัปโหลดโค้ดโปรแกรม ESP Blink

Realtime Database

LEDESP

Analytics

DEVELOP

Auth

Database

Storage

Hosting

Remote Config

Test Lab

Crash

Spark Free

UPGRADE

https://console.firebase.google.com/project/ledesp-86d37/database/data

LEDESP

Go to

https://ledesp-86d37.firebaseio.com/

ledesp-86d37

number: 44

COM3

Send

set /number to 26
set /number to 27
set /number to 28
set /number to 29
set /number to 30
set /number to 31
set /number to 32
set /number to 33
set /number to 34
set /number to 35
set /number to 36
set /number to 37
set /number to 38
set /number to 39
set /number to 40
set /number to 41
set /number to 42
set /number to 43
set /number to 44

Autoscroll

Both NL & CR

9600 baud

รูปที่ 3.33 การเปิดหน้าคอนโซลร่วมกับหน้าจอมอนิเตอร์ของโปรแกรม ESP Blink

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 ทำความรู้จักกับฟังก์ชันต่างๆ

1. คำสั่งใช้ Set - เป็นคำสั่งที่ใช้เซตค่าในออปเจตต่างๆ เช่น เซตให้ออปเจต A มีค่าเป็น 10 (Int) การใช้งานอยู่ในรูปแบบ `Firestore.set???(const String &path, ??? value)`

ตารางที่ 3.1 แสดงรายละเอียดและคำสั่ง Set [12]

???	รายละเอียด	ตัวอย่างการใช้
Int	ให้นำตัวเลขจำนวนเต็มเข้าออปเจต	<code>Firestore.setInt("number", 10);</code>
Float	ให้นำตัวเลขที่มีจุดทศนิยมเข้าออปเจต	<code>Firestore.setFloat("number", 2.56);</code>
Bool	ให้นำค่า True หรือ False เข้าออปเจต	<code>Firestore.setBool("trueORfalse", true);</code>
String	ให้นำข้อความเข้าออปเจต	<code>Firestore.setString("title", "Hello !");</code>

2. คำสั่งใช้ Get - เป็นคำสั่งใช้อ่านค่าจากออปเจตต่างๆ ออกมาในรูปชนิดข้อมูลต่างๆ การใช้งานอยู่ในรูปแบบ `Firestore.get???(const String &path)`

ตารางที่ 3.2 แสดงรายละเอียดและคำสั่ง Get [13]

???	รายละเอียด	ตัวอย่างการใช้
Int	อ่านค่าตัวเลขจำนวนเต็มจากออปเจต	<code>Int number = Firestore.getInt("number");</code>
Float	อ่านค่าตัวเลขที่มีจุดทศนิยมจากออปเจต	<code>float number = Firestore.getFloat("number");</code>
Bool	อ่านค่าค่า True หรือ False จากออปเจต	<code>Bool trueORfalse = Firestore.getBool("trueORfalse");</code>
String	อ่านค่าข้อความจากออปเจต	<code>String title = Firestore.getString("title");</code>

3. คำสั่งใช้ Push - เป็นคำสั่งที่ใช้เพิ่มข้อมูลเข้าไปต่อท้ายในออปเจต โดยจะมี Key เป็นแบบ PRIMARY KEY คือชื่อ Key จะเป็นข้อความยาว ๆ และจะไม่มีซ้ำกันการใช้งานอยู่ในรูปแบบ `Firestore.push???(const String &path, ??? value)`

ตารางที่ 3.3 แสดงรายละเอียดและคำสั่ง Push [14]

???	รายละเอียด	ตัวอย่างการใช้
Int	ให้นำตัวเลขจำนวนเต็มเข้าไปต่อท้ายในออปเจต	<code>Firestore.pushInt("arrNumber", 10);</code>
Float	ให้นำตัวเลขจุดทศนิยมเข้าไปต่อท้ายในออปเจต	<code>Firestore.pushFloat("arrNumber", 2.56);</code>
Bool	ให้นำค่า True หรือ False เข้าไปต่อท้ายในออปเจต	<code>Firestore.pushBool(" arrTrueORfalse" , true);</code>
String	ให้นำข้อความเข้าไปต่อท้ายในออปเจต	<code>Firestore.pushString("arrTitle", "Hello !");</code>

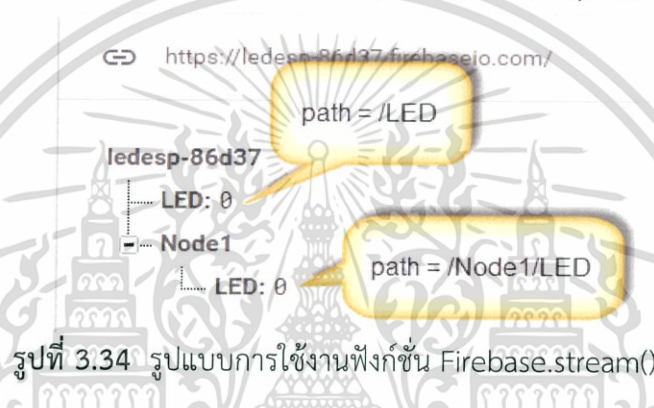
* ซึ่ง ??? คือ Int,Float,Bool,String ซึ่งเป็นตัวแปรของคำสั่ง Set,Get,Push

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.7 การสตรีมมิ่งข้อมูล

เนื่องจากการดาวน์โหลดไฟล์ทั้งไฟล์จะใช้เวลาค่อนข้างมากการเล่นไฟล์จากอินเทอร์เน็ตด้วยเทคนิค "สตรีมมิ่ง" จะทำให้สามารถแสดงผลข้อมูลได้ก่อนที่ไฟล์ทั้งหมดจะถูกส่งผ่านเข้ามายังเครื่องคอมพิวเตอร์ หลังจากที่ได้เรียนรู้การเพิ่มข้อมูลเข้าไปแล้ว ต่อมาเรามาลองเรียนรู้การรออัปเดตข้อมูลกันบ้าง ซึ่งสามารถนำไปประยุกต์ใช้การรับ - ส่งข้อมูลแบบเรียลไทม์ได้

1. ใน Void setup() จะต้องเรียกฟังก์ชัน Firebase.stream() ขึ้นมาก่อน โดยมีรูปแบบการใช้งานดังนี้ Firebase.Stream(Const String &Path) โดยที่ Path หมายถึงออปเจกที่เราต้องการตรวจจับความเปลี่ยนแปลง เช่น ถ้าหากออปเจก LED ซึ่งอยู่เป็นออปเจกนอกสุด ตัว Path ก็จะเป็น /LED , แต่หากออปเจก LED อยู่ในออปเจก Node1 และออปเจก Node1 เป็นออปเจกนอกสุด ตัว Path ก็จะเป็น /Node1/LED ตัวเครื่องหมาย / จะเป็นตัวคั่น Key หรือชื่อของออปเจก



รูปที่ 3.34 รูปแบบการใช้งานฟังก์ชัน Firebase.stream()

2. เลือกใช้ Path ที่ /Node1/LED ดังนั้นจึงต้องนำโค้ดด้านล่างนี้ไปไว้ใน void setup() >>>> [Firebase.stream("/Node1/LED");]

```

21 Serial.print("connecting");
22 while (WiFi.status() != WL_CONNECTED) {
23   Serial.print(".");
24   delay(500);
25 }
26 Serial.println();
27 Serial.print("connected: ");
28 Serial.println(WiFi.localIP());
29
30 Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
31 Firebase.stream("/Node1/LED");
32 }
33
34 |
35 void loop() {
36   if (Firebase.failed()) {
37     Serial.println("streaming error");
38     Serial.println(Firebase.error());

```

รูปที่ 3.35 รูปแบบการใช้งานฟังก์ชัน Firebase.stream() (2)

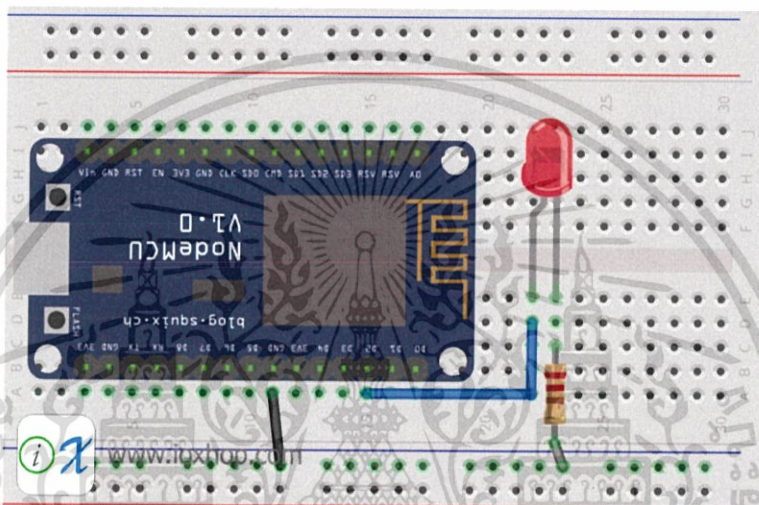
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ Void loop() สามารถ Copy ไปได้ทั้งหมดมีเพียงแต่การเรียกฟังก์ชัน event.get???("data"); ที่ต้องระวัง เพราะหากชนิดของข้อมูลผิด ก็จะทำให้ไม่สามารถรับค่าได้เลย

ตัวอย่างเช่น หากในหน้าคอนโซล ได้กรอกไว้เป็น 1 แต่รับค่าแบบ String ผลที่ได้จะเป็นค่าว่าง เพราะ 1 ถือเป็นตัวเลข คำสั่ง event.getString("data"); จะไม่สามารถรับค่าได้ แต่ถ้าหากใช้คำสั่ง event.getInt("data"); จะสามารถรับข้อมูลได้ปกติ

3.1.8 ทดลองควบคุม LED แบบเรียลไทม์

1. ต่อดิจตามรูปด้านล่าง และอัปโหลดโค้ด

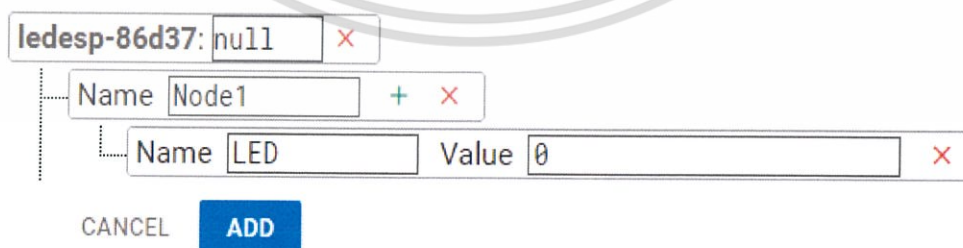


รูปที่ 3.36 การต่อวงจรระหว่าง NodeMCU กับ บอร์ด [15]

2. กลับไปที่คอนโซล ไปเพิ่มออปเจกต์ Node1 --> LED:0 ผลที่ได้คือหลอด LED ยัง

ดับอยู่

<https://ledesp-86d37.firebaseio.com/>



รูปที่ 3.37 การทดลองควบคุม LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.38 การทดลองควบคุม LED เมื่อให้ค่า LED เป็น 0

3. เมื่อแก้แอปเจค LED ให้เป็น 1 ผลคือหลอด LED ติดทันที



รูปที่ 3.39 การทดลองควบคุม LED เมื่อให้ค่า LED เป็น 1

การใช้ Firebase พื้นฐาน และการใช้ Firebase บน Arduino ESP8266 สามารถนำไปประยุกต์ใช้การสื่อสารกันระหว่างอุปกรณ์ กับ อุปกรณ์ได้แต่หากต้องการจะให้การควบคุมอุปกรณ์ผ่านหน้าเว็บไซต์ หรือผ่านแอปพลิเคชัน จำเป็นจะต้องเรียนรู้ API ในภาษาอื่น ๆ เพิ่มอีก แต่ยังคงมีฟังก์ชัน Set Get Push อยู่เหมือนเดิม

3.2 การเชื่อม Esp8266 เข้ากับ Firebase

ขั้นแรกให้ NodeMCU เชื่อมต่อกับ Access Point เพื่อให้ลิงค์กับอุปกรณ์ในเครือข่ายรวมทั้งเข้าถึงอินเทอร์เน็ตได้

1. เมื่อเชื่อมต่อสำเร็จ NodeMCU ก็จะได้ IP Address มาเพื่อให้อุปกรณ์อื่นในวงแลนสามารถติดต่อสื่อสารได้ โคลนี้ IP ที่ได้แต่ละครั้งจะไม่คงที่ ขึ้นอยู่กับ Access Point จะกำหนดให้ เรียกว่าแบบ DHCP ซึ่งไม่สะดวกในการติดต่อภายหลัง

1	void setup()
2	{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

3 Serial.begin(115200); //ตั้งค่าใช้งาน serial ที่ baudrate 115200
4 delay(10);
5 Serial.print("Connecting to "); //แสดงข้อความ "Connecting to"
6 Serial.println(ssid); //แสดงข้อความ ชื่อ SSID
7
8 WiFi.begin(ssid, password); // สั่งให้เชื่อมต่อกับ AP
9 //WiFi.begin(ssid); // กรณีเชื่อมต่อแบบไม่มี password
10
11 while (WiFi.status() != WL_CONNECTED) //รอกันกว่าจะเชื่อมต่อสำเร็จ
12 {
13     delay(500);
14     Serial.print("*");
15 }
16 Serial.println("");
17 Serial.println("WiFi connected"); //แสดงข้อความเชื่อมต่อสำเร็จ
18 Serial.println("IP address: ");
19 Serial.println(WiFi.localIP()); //แสดงหมายเลข IP NodeMCU ของเรา
20 }

```

2. สามารถฟิก IP ให้คงที่ หรือเรียกว่าแบบ Static IP ได้ดังนี้

```

1 void setup()
2 {
3     Serial.begin(115200);
4     delay(10);
5     Serial.println();
6     Serial.println();
7     Serial.print("Connecting to ");
8     Serial.println(ssid);
9     WiFi.begin(ssid, password);
10
11     while (WiFi.status() != WL_CONNECTED) //รอกันกว่าจะเชื่อมต่อสำเร็จ
12     {
13         delay(500);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

14 Serial.print("*");
15 }
16
17 IPAddress local_ip = {192, 168, 1, 45}; //พิกัด IP
18 IPAddress gateway = {192, 168, 1, 1}; //ตั้งค่า IP Gateway
19 IPAddress subnet = {255, 255, 255, 0}; //ตั้งค่า Subnet
20 WiFi.config(local_ip, gateway, subnet); //setค่าไปยังโมดูล
21
22 Serial.println("");
23 Serial.println("WiFi connected"); //แสดงข้อความเชื่อมต่อสำเร็จ
24 Serial.println("IP address: ");
25 Serial.println(WiFi.localIP()); //แสดงหมายเลข IP ของ NodeMCU
26 }

```

3. การดึงเวลาจาก Server มาใช้งาน

```

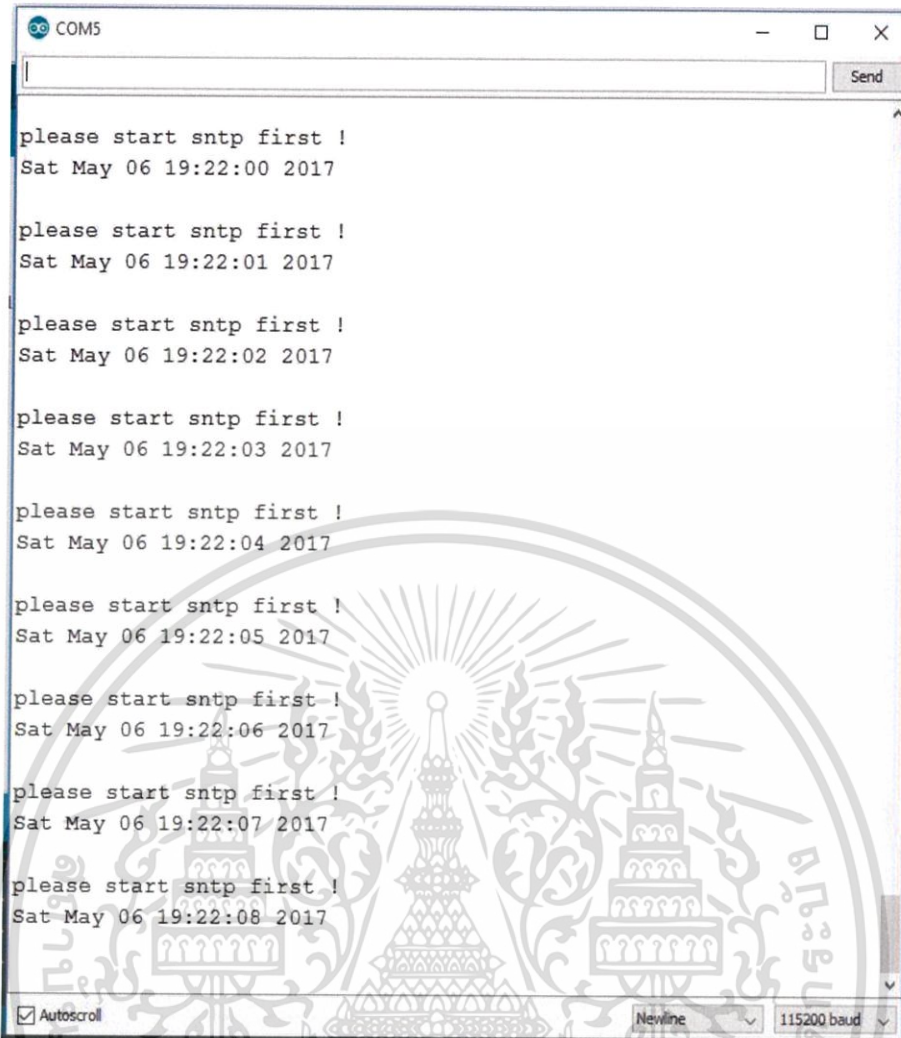
1 #include <ESP8266WiFi.h>
2 #include <time.h>
3 int timezone = 7 * 3600; //ตั้งค่า TimeZone ตามเวลาประเทศไทย
4 int dst = 0; //กำหนดค่า Date Swing Time
5
6 void setup()
7 {
8   Serial.begin(115200);
9   Serial.setDebugOutput(true);
10
11   WiFi.mode(WIFI_STA); //เชื่อมต่อ Wifi
12   WiFi.begin(ssid, password);
13   Serial.println("\nConnecting to WiFi");
14   while (WiFi.status() != WL_CONNECTED) {
15     Serial.print("*");
16     delay(1000);
17   }
18   configTime(timezone, dst, "pool.ntp.org", "time.nist.gov"); //ดึงเวลาจาก Server

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

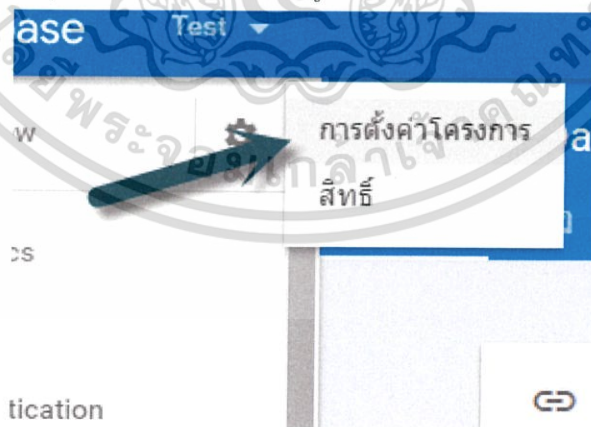
```
19 Serial.println("\nLoading time");
20 while (!time(nullptr)) {
21     Serial.print("*");
22     delay(1000);
23 }
24 Serial.println("");
25 }
26
27 void loop()
28 {
29     //configTime(timezone, dst, "pool.ntp.org", "time.nist.gov"); //แสดงเวลา
30     time_t now = time(nullptr);
31     struct tm* p_tm = localtime(&now);
32
33     Serial.print(p_tm->tm_hour);
34     Serial.print(":");
35     Serial.print(p_tm->tm_min);
36     Serial.print(":");
37     Serial.print(p_tm->tm_sec);
38     Serial.println("");
39     delay(1000);
40 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.40 การตั้งเวลาของจอ Serial Monitor

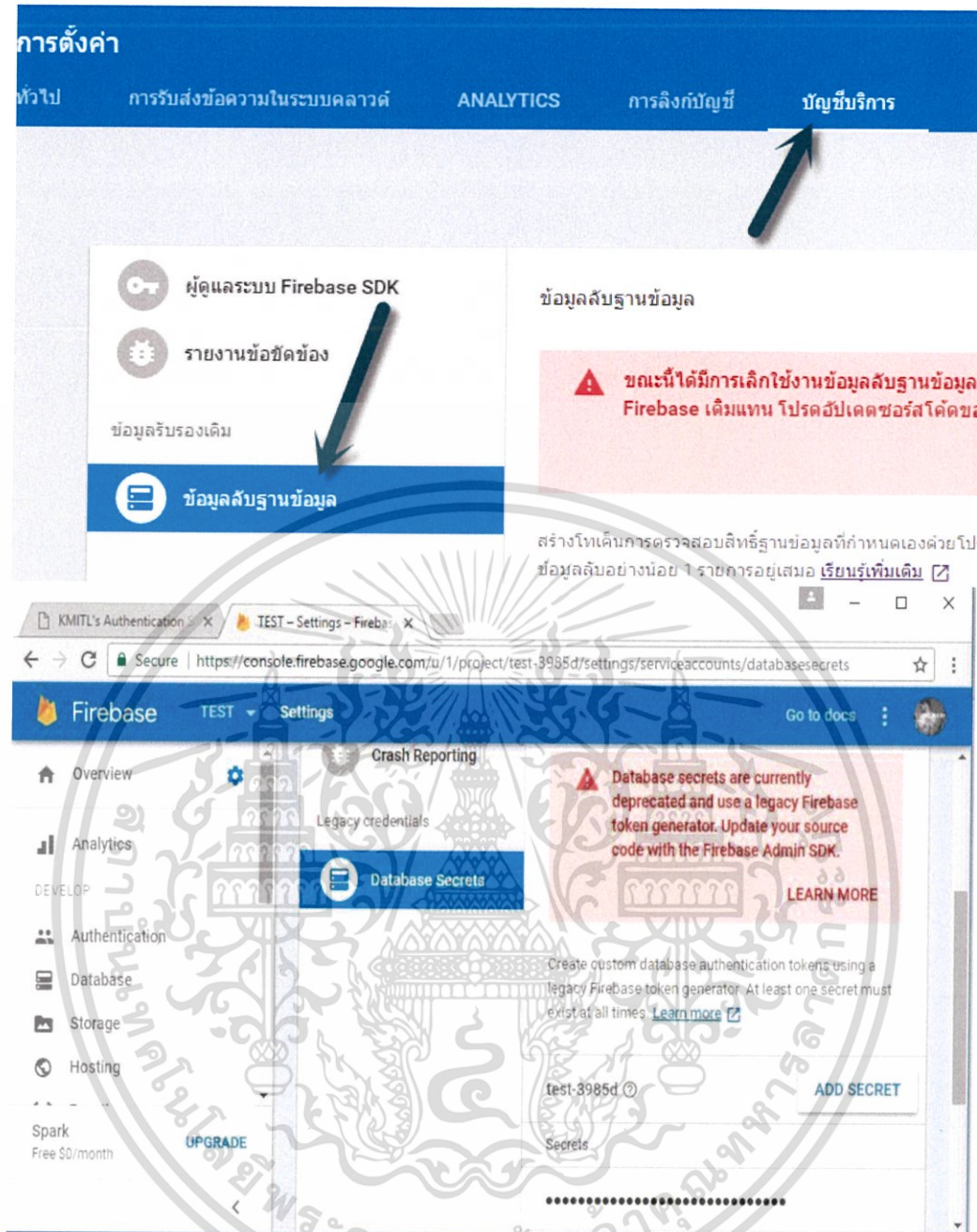
4. กดไปที่รูปเฟืองมุมซ้ายบน แล้วคลิกที่เมนู การตั้งค่าโครงการ



รูปที่ 3.41 การตั้งค่าโครงการ

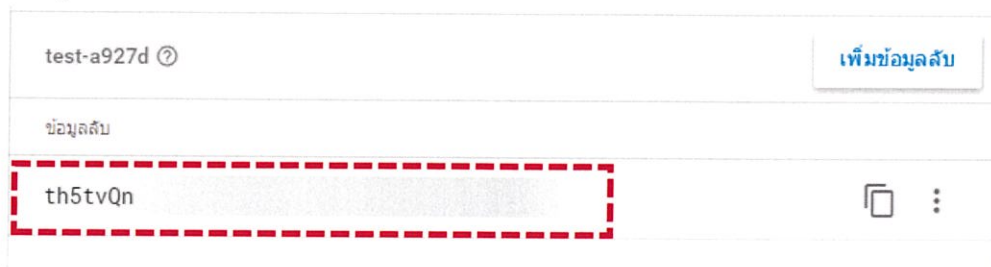
5. คลิกที่แท็บ บัญชีบริการ ในเมนูด้านซ้าย เลือก ข้อมูลพื้นฐานข้อมูล นำเมาส์เลื่อนไปตรงช่อง กดที่ปุ่ม แสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.42 หน้าของบัญชีลับฐานข้อมูล

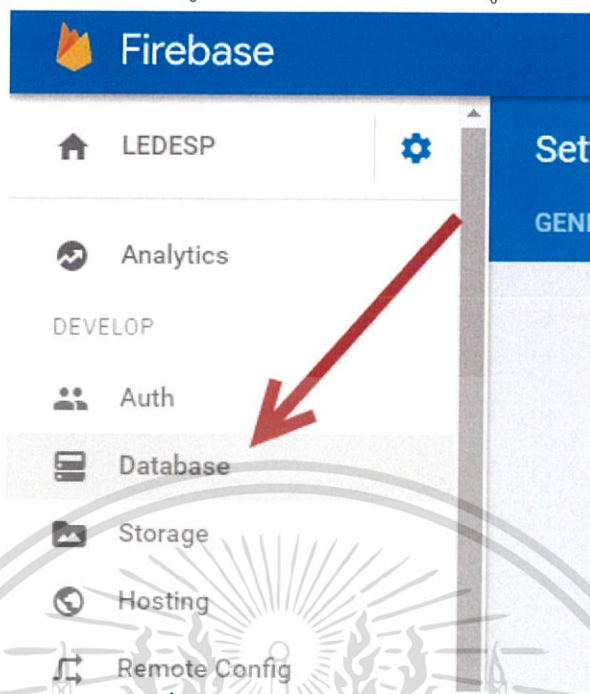
6. คัดลอก Key เก็บไว้ ตัว Key นี้จะใช้ในการคอนฟิกในโค้ด เพื่อให้สามารถเข้ามาแก้ไขข้อมูลในฐานข้อมูล Firebase ได้



รูปที่ 3.43 การคัดลอก Key

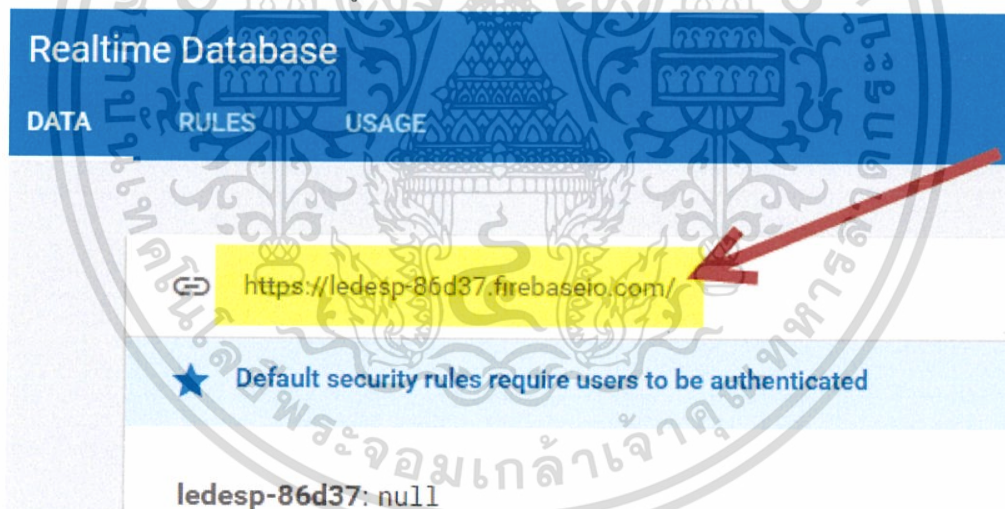
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. กดไปที่เมนู Database ในเมนูด้านซ้ายเพื่อเข้าถึงลิงค์ข้อมูล



รูปที่ 3.44 แถบเมนู Develop

8. ก๊อปลิงค์ตามตำแหน่งในรูปเก็บไว้ ลิงค์นี้จะเอาไว้ใช้คอนฟิกในโค้ดเช่นเดียวกัน



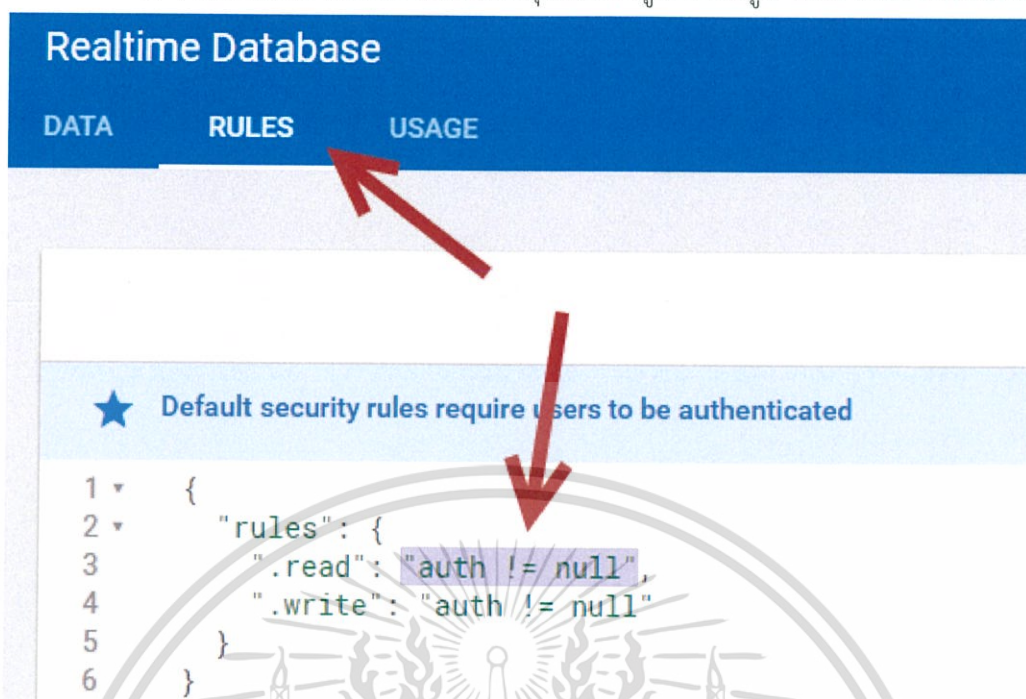
รูปที่ 3.45 ลิงค์ที่ใช้ในการคอนฟิก

3.2.1 การกำหนดสิทธิ์การเข้าถึงข้อมูล

เป็นส่วนที่สำคัญหากนำไปใช้งานจริง เนื่องจาก Firebase ได้กำหนดค่าเริ่มต้นให้ต้องมี Key เท่านั้น จึงจะสามารถอ่าน-เขียนฐานข้อมูลได้ ซึ่งจะมีปัญหาหากนำไปใช้งานบนหน้าเว็บไซต์ เพราะในหน้าเว็บจะไม่สามารถเปิดเผย Key ได้ การเปิดเผย Key จะทำให้ Users สามารถเข้ามาแก้ไขข้อมูลในฐานข้อมูลได้ ดังนั้นจึงควรกำหนดให้ในการเขียนข้อมูลลงไปในฐานข้อมูล สามารถทำได้เฉพาะมี Key และหากมี หรือไม่มี Key จะอ่านได้เท่านั้น ซึ่งสามารถกำหนดได้ดังนี้

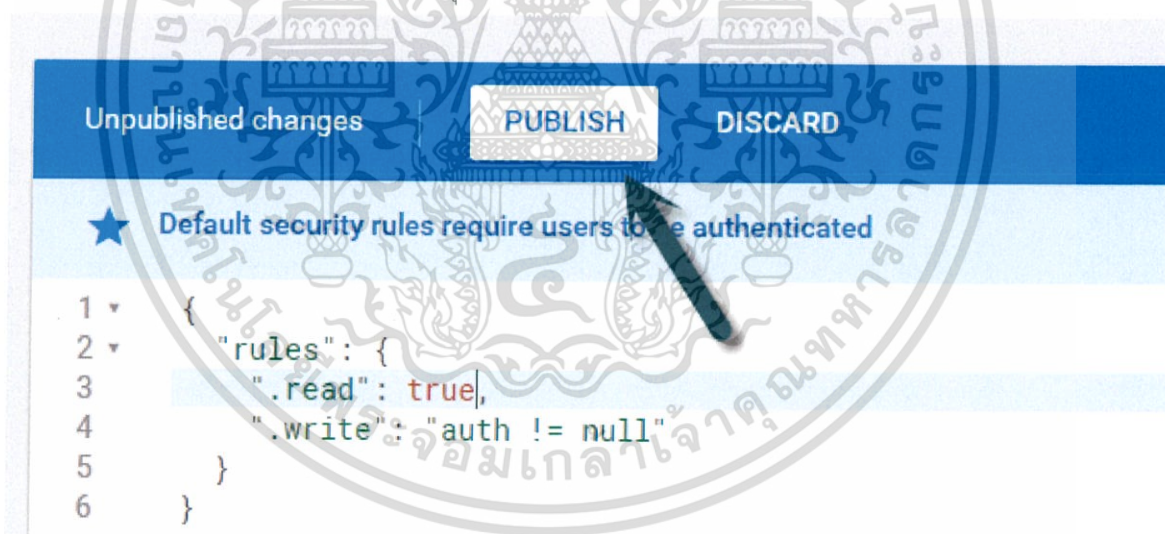
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กดไปที่แท็บ RULES ในส่วนที่คลุมแถบคำสั่งที่ในรูปแบบ ให้ใส่ True ลงไปแทน



รูปที่ 3.46 หน้าจอ RULES

2. เมื่อแก้ไขแล้ว ให้กดปุ่ม PUBLISH เพื่อเป็นการบันทึกค่า



รูปที่ 3.47 หน้าจอสำหรับกดปุ่ม PUBLISH

จากนี้ฐานข้อมูล Firebase ได้พร้อมใช้งาน...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การทำแอปพลิเคชัน

โปรแกรม App Inventor ช่วยให้สามารถพัฒนาแอปพลิเคชันสำหรับ Smart Phone ระบบปฏิบัติการ Android ซึ่งทำผ่านการใช้เว็บเบราว์เซอร์และทดสอบบน Smart Phone ที่เชื่อมต่ออยู่กับคอมพิวเตอร์หรือทดสอบบนโทรศัพท์จำลองบนเครื่องคอมพิวเตอร์ โปรแกรมที่สร้างทั้งหมดจะถูกจัดเก็บไว้บนเซิร์ฟเวอร์ App Inventor ซึ่งช่วยให้สามารถพัฒนางานต่อที่เครื่องคอมพิวเตอร์เครื่องใดก็ได้ เพียงแค่ได้มีการเชื่อมต่อกับระบบอินเทอร์เน็ตไว้เท่านั้น

การสร้างแอปพลิเคชันจะแบ่งการทำงานออกเป็นสองส่วน คือ ส่วนออกแบบ (App Inventor Designer) ที่จะให้เราเลือกคอมโพเนนท์ที่ต้องการสำหรับที่จะให้สร้างแอปพลิเคชัน ส่วนที่สองเป็นส่วนการเขียนโค้ด (App Inventor Blocks Editor) ที่ให้เราเขียนโค้ดด้วยการต่อบล็อกต่างๆ เข้าด้วยกันเป็นคำสั่ง ซึ่งจะเป็นการกำหนดพฤติกรรมหรือเหตุการณ์ที่เกิดขึ้นกับคอมโพเนนท์ การเขียนโปรแกรมจะเหมือนการต่อชิ้นส่วนตัวต่อจิ๊กซอว์เข้าด้วยกัน ในแต่ละขั้นตอนการสร้างจะสามารถทำการทดสอบได้ และเมื่อสร้างเสร็จสมบูรณ์แล้วจะสามารถแพ็คเกจแอปพลิเคชันเพื่อนำไปใช้งานบนโทรศัพท์ระบบปฏิบัติการ Android เครื่องใดก็ได้

โดยก่อนอื่นเราก็จะต้องเข้าไปที่เว็บไซต์ <http://ai2.appinventor.mit.edu> [16] และล็อกอินด้วยบัญชีของ Google ก่อน จากนั้นเริ่มสร้างโปรเจกใหม่โดยคลิกไปที่ Start New Project แล้วทำการตั้งชื่อโปรเจกตามต้องการ ซึ่งชื่อนี้จะเป็นชื่อเดียวกับแอปพลิเคชันเมื่อติดตั้งบนโทรศัพท์



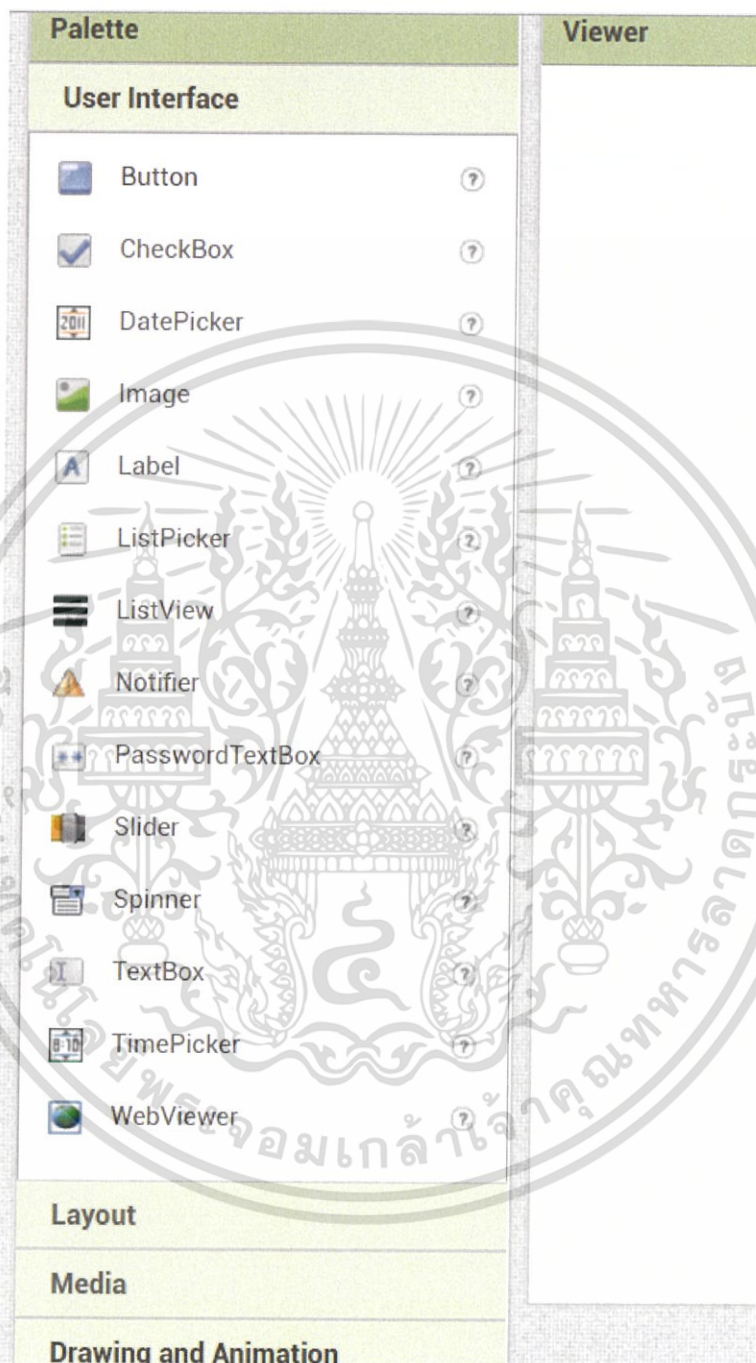
รูปที่ 3.48 การตั้งชื่อแอปพลิเคชัน

โดยการสร้างแอปพลิเคชันจะประกอบด้วย 2 ส่วนคือ ส่วนของ Designer ซึ่งจะเป็นการออกแบบหน้าจอแสดงผลของแอปพลิเคชัน และอีกส่วนคือส่วนของ Blocks ซึ่งจะเป็นส่วนของการจัดการด้านโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 ส่วนของDesigner

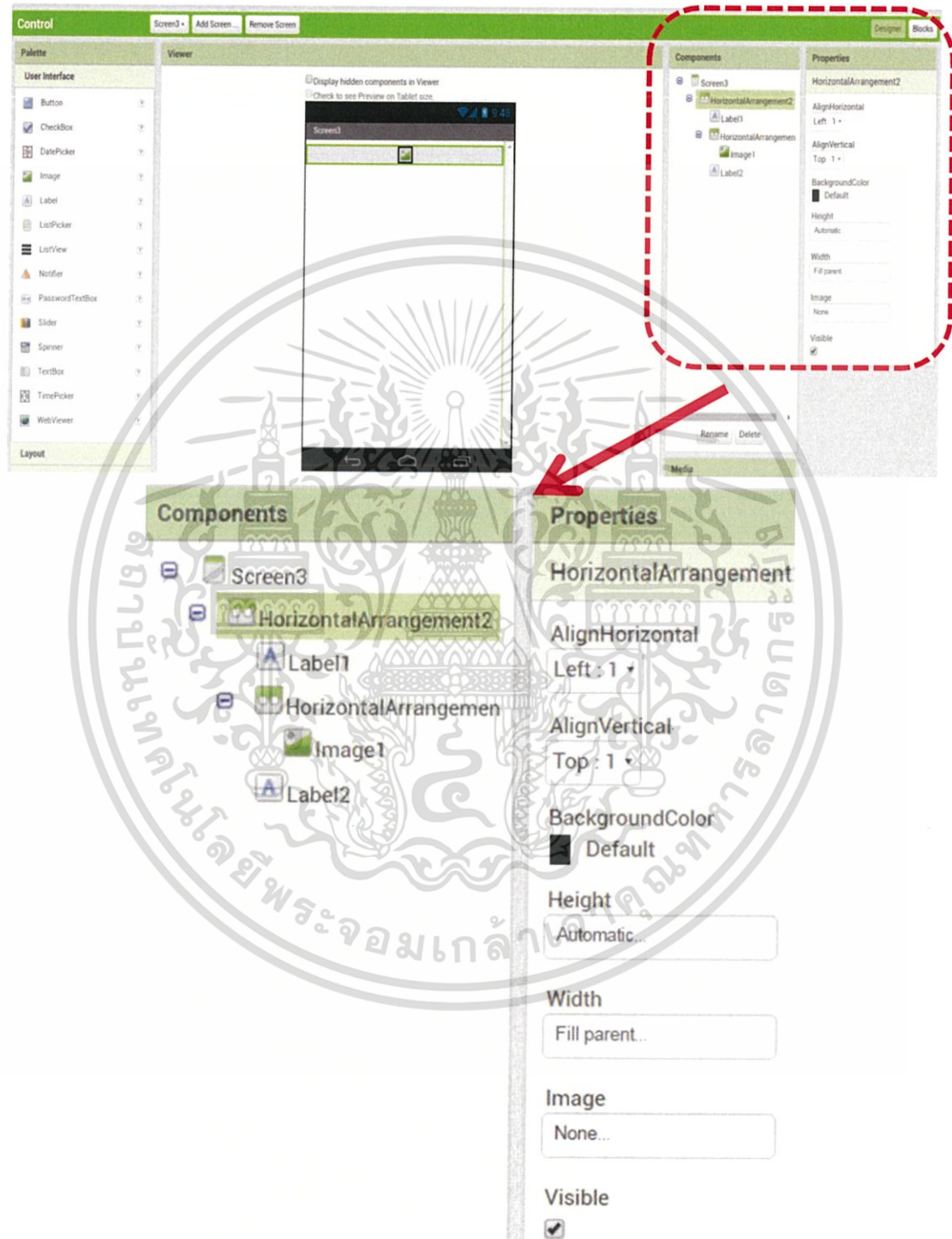
เราสามารถที่จะออกแบบในส่วนนี้เพื่อที่จะนำไปแสดงผลเป็นหน้าจอจริงในแอปพลิเคชัน หลังจากติดตั้งแล้ว ในส่วนนี้จะมีตัวเลือกต่างๆมากมายที่สามารถเลือกใช้อย่างอิสระ



รูปที่ 3.49 ตัวเลือกคำสั่งต่างๆในส่วนของ Designer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

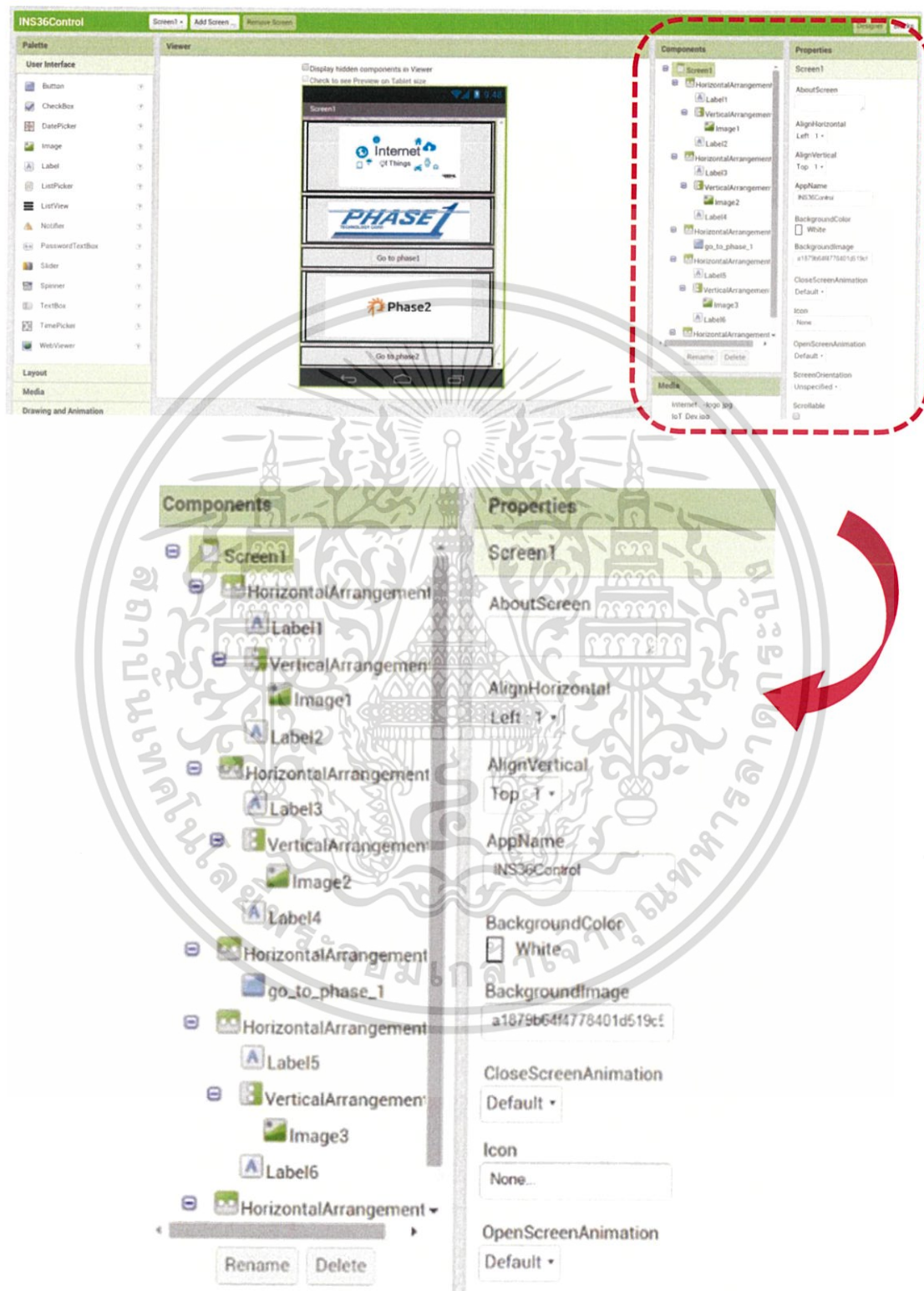
จากนั้นเลือกใช้คำสั่งตามต้องการ โดยในที่นี้ หน้า Screen1 จะเลือกใช้คำสั่ง Horizontal Arrangement เพื่อควบคุมขอบเขตการวางปุ่มต่างๆบนหน้าจอ โดยสามารถที่จะกำหนดความกว้างและความสูงของขอบเขตนี้ได้โดยการปรับที่ Height และ Width ฝั่งขวามือของหน้าจอ แล้วจึงเลือกใช้คำสั่ง Image เพื่อนำรูปภาพมาตกแต่งบนหน้าจอ



รูปที่ 3.50 การใช้งานคำสั่ง Horizontal และ Image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

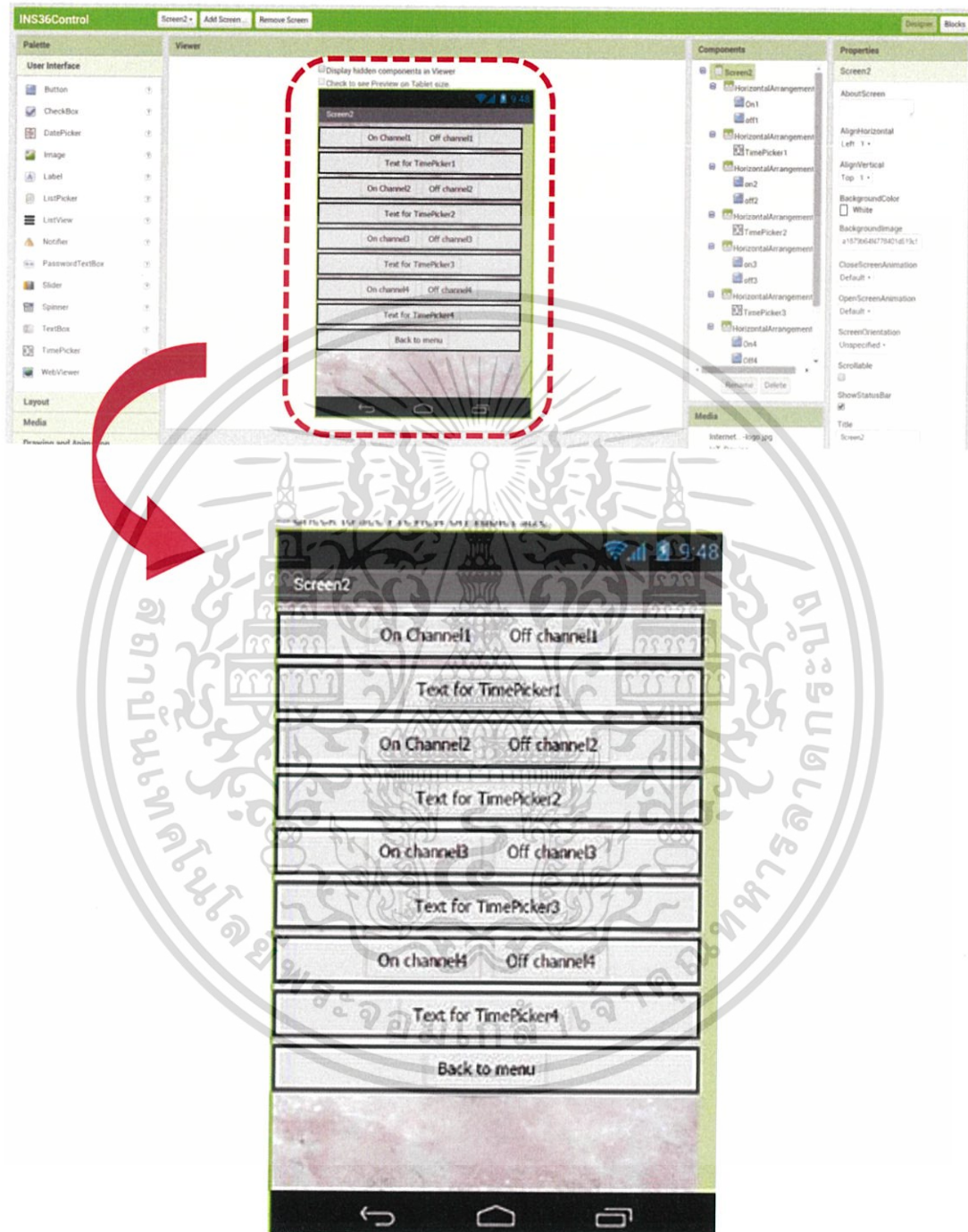
จากนั้นสามารถที่จะเลือกรูปที่จะใส่ได้ตามต้องการ โดยกดไปที่บล็อกของ Image และเลือกที่ Picture เพื่ออัปรูปที่ต้องการเข้ามาที่บล็อก Image ที่สร้างไว้ รวมถึงสามารถเพิ่มปุ่มกดเพื่อใช้งานโดยใช้คำสั่ง Button แล้วนำเข้ามาวางบนหน้าจอเหมือนเดิม



รูปที่ 3.51 หน้า Screen1 หลังจากตกแต่งเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

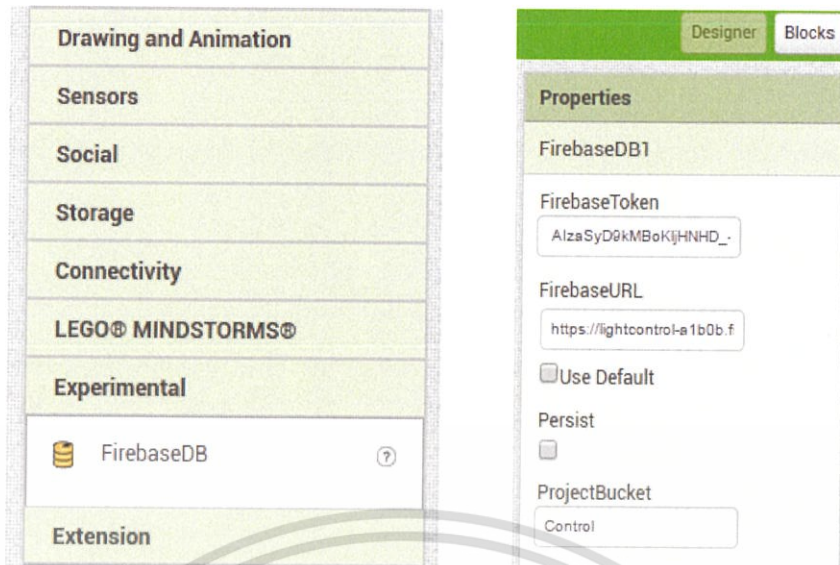
ที่ Screen2 และ Screen3 ก็เช่นกันโดย 2 หน้านี้จะ เป็นหน้าจอแสดงผลในส่วนของปุ่ม ควบคุมการทำงานและปุ่มของตัวตั้งเวลา ซึ่งก็สามารถใช้คำสั่ง Horizontal ร่วมกับ Button ได้ เหมือนเดิม เพียงเพิ่มปุ่มเข้าไปให้เพียงพอตามต้องการ



รูปที่ 3.52 หน้า Design ของ Screen2

รวมถึงการเพิ่มคำสั่ง Firebase เข้าไป เพื่อให้สามารถใช้งาน Firebase ร่วมกับแอปพลิเคชัน ได้ โดยคำสั่ง Firebase จะอยู่ที่ Experimental แล้วจึงทำการก๊อปปี้ลิงค์จาก Database ของ Firebase มาใส่ตรงช่อง FirebaseURL เพื่อเชื่อม Firebase กับแอปพลิเคชันให้ทำงานร่วมกัน

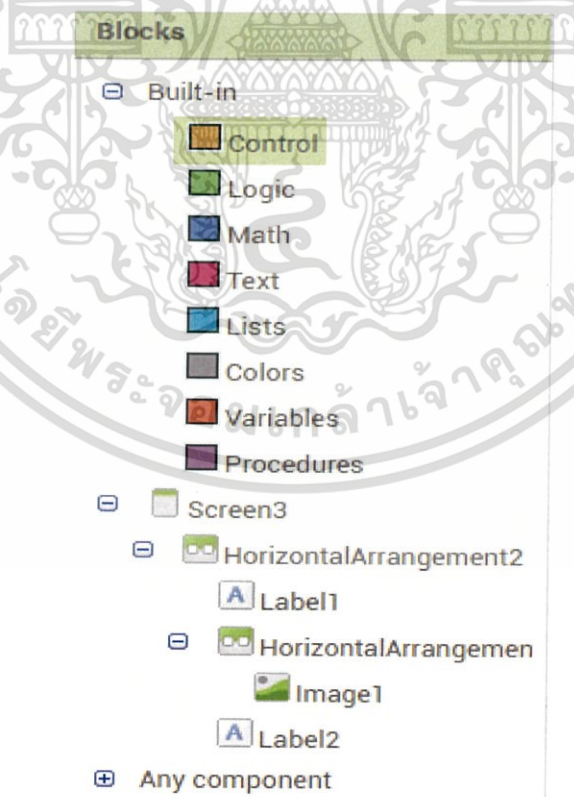
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.53 การเชื่อมลิงค์ระหว่าง Firebase กับแอปพลิเคชัน

3.3.2 ส่วนของ Blocks

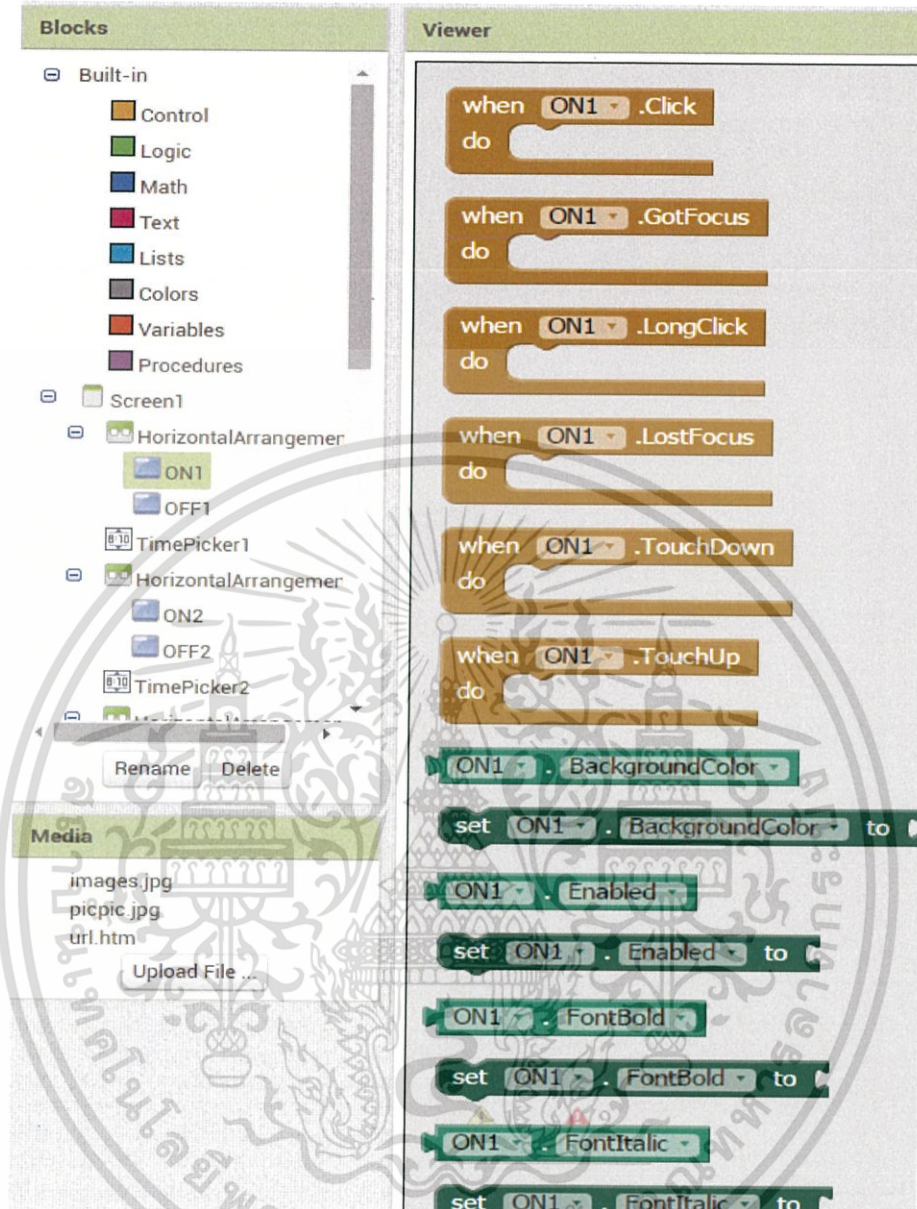
จะเป็นส่วนที่เป็นเหมือนโปรแกรมของแอปพลิเคชัน ซึ่งสามารถเขียนเงื่อนไขการทำงานได้อย่างอิสระ โดยนำปุ่มหรือคำสั่งต่างๆจากส่วนของ Designer มาใช้ในการกำหนดเงื่อนไขการทำงาน โดยสามารถเลือกใช้งานคำสั่งต่างๆได้จากเมนูฝั่งซ้ายมือของหน้าจอ



รูปที่ 3.54 หน้าจอในส่วนของ Blocks

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

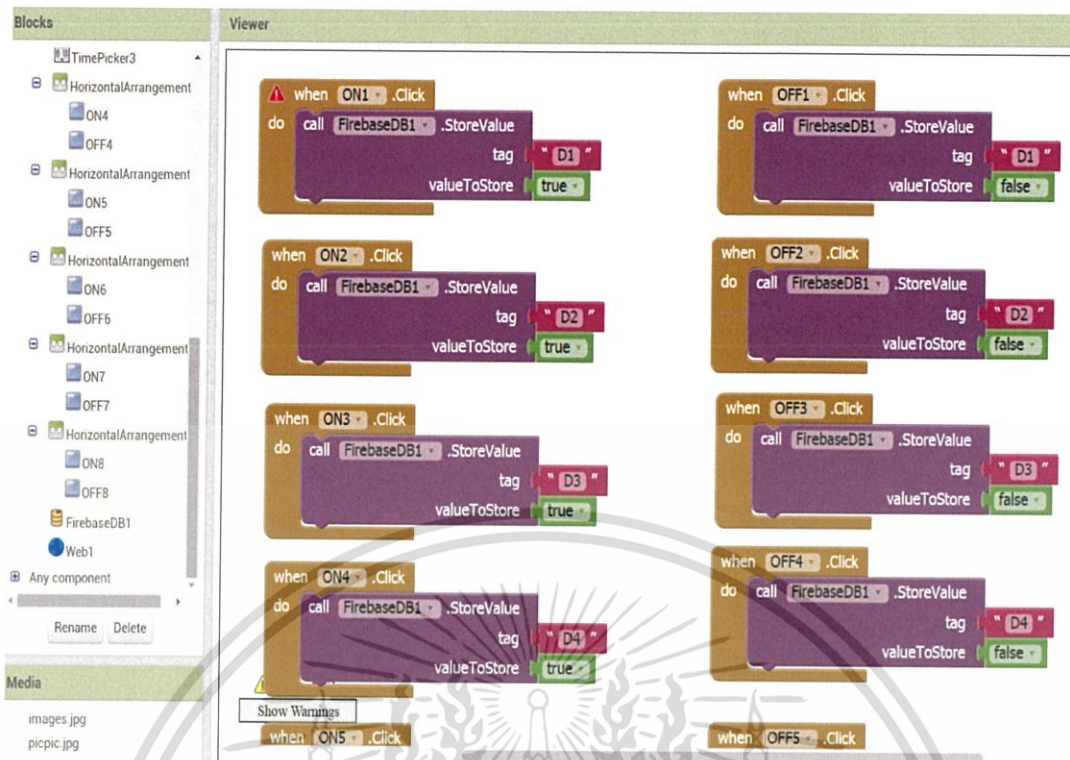
จากนั้นลากคำสั่งการใช้งานจากฝั่งซ้ายมือ มาเชื่อมต่อกันที่พื้นที่ว่างของหน้าจอ



รูปที่ 3.55 ตัวอย่างแถบคำสั่งต่างๆที่สามารถใช้ได้

โดยแต่ละปุ่มที่เราสร้างในหน้า Designer ก็จะมีปรากฏอยู่ในแถบด้านซ้ายมือของหน้า Blocks ด้วย โดยเราสามารถลากคำสั่งต่างๆมาเชื่อมกับตามเงื่อนไขที่ต้องการ ซึ่งในที่นี้จะใช้ในการเปิด-ปิดการทำงานของรีเลย์ โดยเชื่อมสถานะการเปิด-ปิด เข้ากับ Firebase จึงใช้คำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.56 ตัวอย่างคำสั่งเงื่อนไขในการทำงานของ Blocks

โดยคำสั่งที่ใช้ในหน้า Blocks คำสั่งแรกก็คือ เมื่อปุ่ม ON1 ถูกกด ให้เรียกใช้งานไปยัง Firebase และสั่งให้หัวข้อ D1 ใน Firebase นั้นเปลี่ยนสถานะเป็น True ส่วนคำสั่งที่สองก็คือ เมื่อปุ่ม OFF1 ถูกกด ให้เรียกใช้งานไปยัง Firebase และสั่งให้หัวข้อ D1 ใน Firebase นั้นเปลี่ยนสถานะเป็น False และคำสั่งอื่นๆก็เช่นกัน โดยรวมแล้วก็คือใช้เปลี่ยนแปลงค่าสถานะ True หรือ False ที่ตัว Firebase นั้นเอง

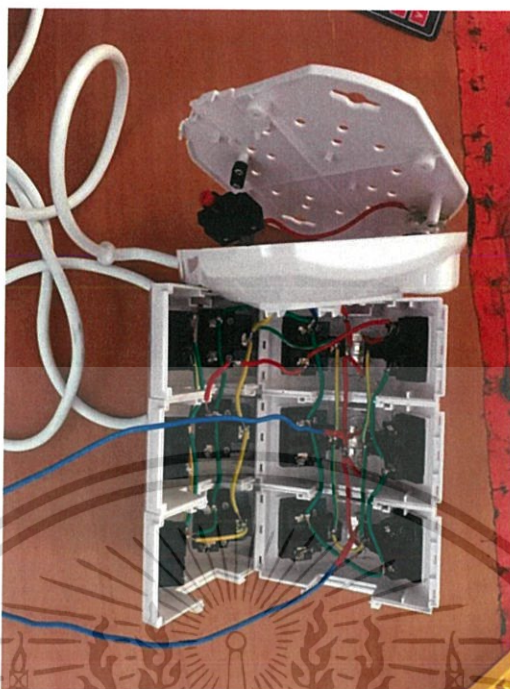
3.4 การดำเนินงานด้าน Hardware

ในที่นี้จะทำการนำ Relay 8 Channel ไปควบคุมการทำงานของปลั๊กแบบ Tower อุปกรณ์ทั้งหมดประกอบด้วย

1. Relay 8 Channel Module
2. Esp8266 Module
3. ปลั๊ก Tower แบบ 3 ชั้น
4. สายไฟ
5. เทอร์มินอลสำหรับช่วยในการเดินสายไฟ

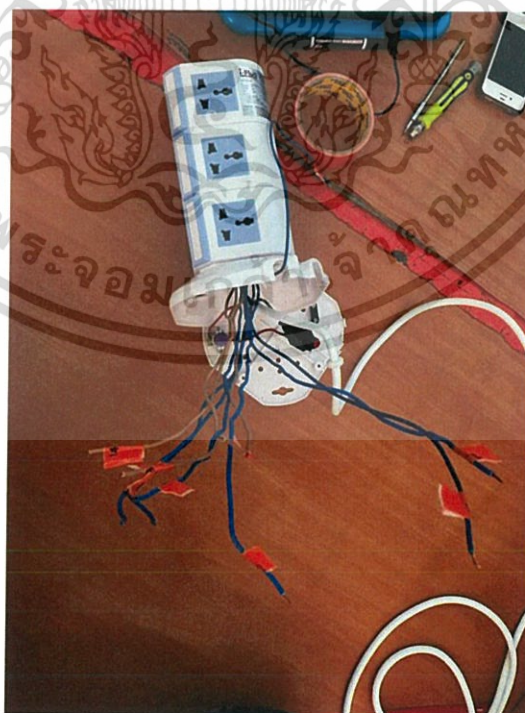
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนแรก ทำการเดินสาย Line ออกมาจาก Tower ปลั๊ก



รูปที่ 3.57 การเดินสาย Line ออกมาจากปลั๊ก Tower

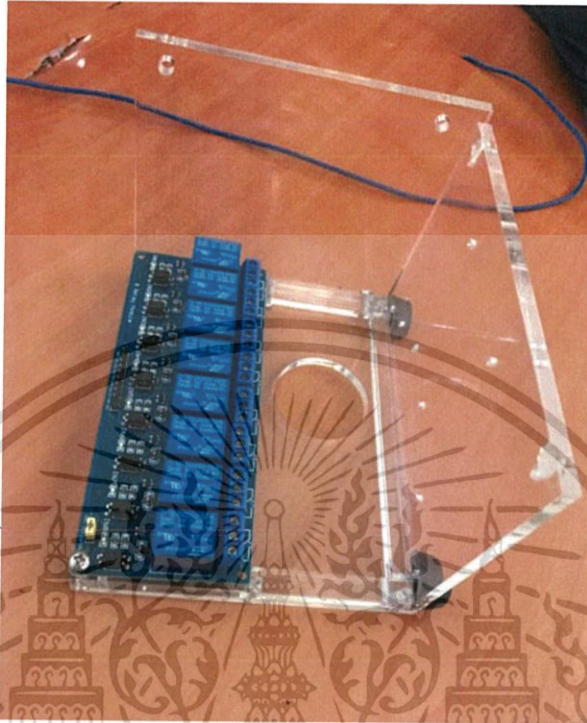
เนื่องจากว่าปลั๊ก Tower นั้นมีทั้งหมด 3 ชั้น แต่ละชั้นมี 4 ช่องเสียบ แต่ Relay ที่ใช้นั้นมีเพียง 8 ช่องเสียบ จึงทำการเดินสาย Line ออกมาจากปลั๊กเพียง 8 สาย ซึ่งมาจาก 2 ชั้นบน เพื่อให้เพียงพอต่อการควบคุมของ Relay ส่วนชั้นล่างที่เหลือก็สามารถที่จะนำไปจ่ายไฟให้กับ Esp8266 และ Relay ได้ โดยผ่านการแปลงไฟจาก Adapter ก่อน เพื่อให้ได้ไฟเลี้ยงที่เหมาะสม



รูปที่ 3.58 หลังจากเดินสาย Line ออกมาจากปลั๊ก Tower แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเดินสายไฟออกมาจากปลั๊ก Tower แล้ว ก็ทำการประกอบอุปกรณ์อื่นๆเข้าด้วยกัน โดยทำการยึด Relay และ Esp8266 เข้ากับแท่นอะคริลิก เพื่อให้ง่ายต่อการขนย้าย และประหยัดพื้นที่การใช้งาน



รูปที่ 3.59 การยึดอุปกรณ์เข้ากับแท่นอะคริลิก

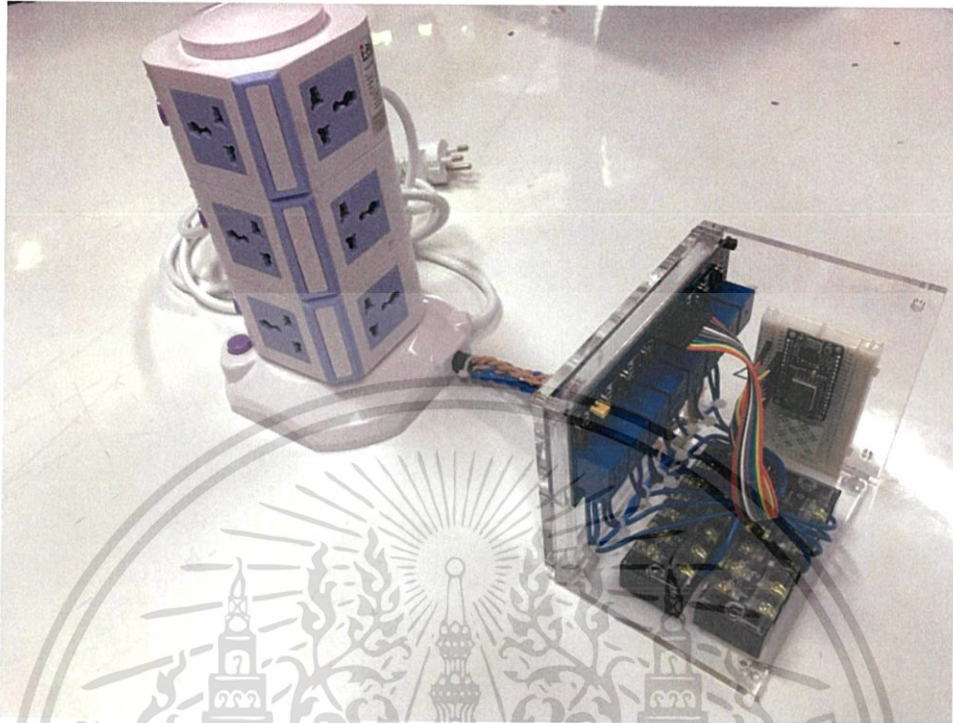
จากนั้นทำการเชื่อมสายไฟจาก Tower ปลั๊กเข้าไปที่ Relay โดยเชื่อมผ่านเทอร์มินอล เพื่อประหยัดจำนวนสายไฟที่ใช้



รูปที่ 3.60 การเชื่อมสายไฟเข้ากับ Relay 8 Channel โดยผ่านเทอร์มินอล

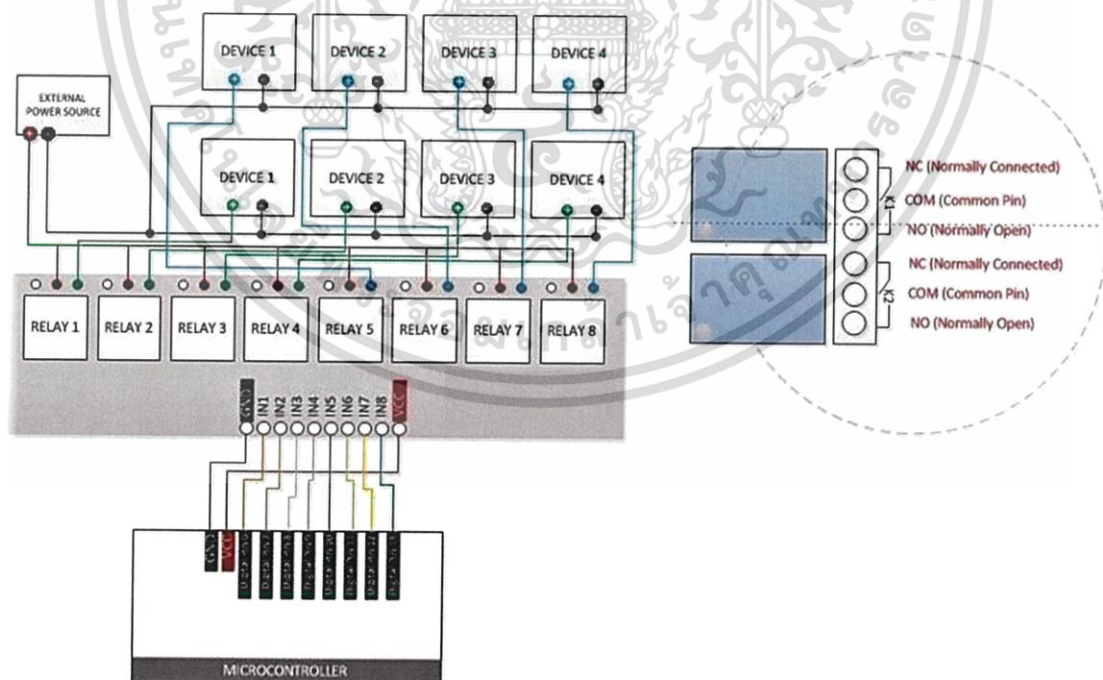
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการยึด Esp8266 เข้ากับบอร์ดแล้วโยงสายไฟต่างๆเข้าไปยังบอร์ด Relay ก็เป็นอันเสร็จสิ้นด้าน Hardware



รูปที่ 3.61 Hardware ทั้งหมดหลังจากเชื่อมต่อสายเข้าด้วยกันแล้ว

วงจรไฟฟ้ารวมของทั้งระบบ



รูปที่ 3.62 วงจรไฟฟ้ารวมของทั้งระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 อุปกรณ์การทดลอง

4.1.1 อุปกรณ์ที่ใช้ส่งงานควบคุม

ได้แก่ Smartphone ระบบ Android และ Computer ทำหน้าที่ในการสั่งเปิดปิด Relay โดยส่งงานผ่าน Esp8266 Module



รูปที่ 4.1 อุปกรณ์ที่ใช้ส่งงานควบคุม [17]

4.2.2 เราท์เตอร์ตัวส่งสัญญาณอินเทอร์เน็ต

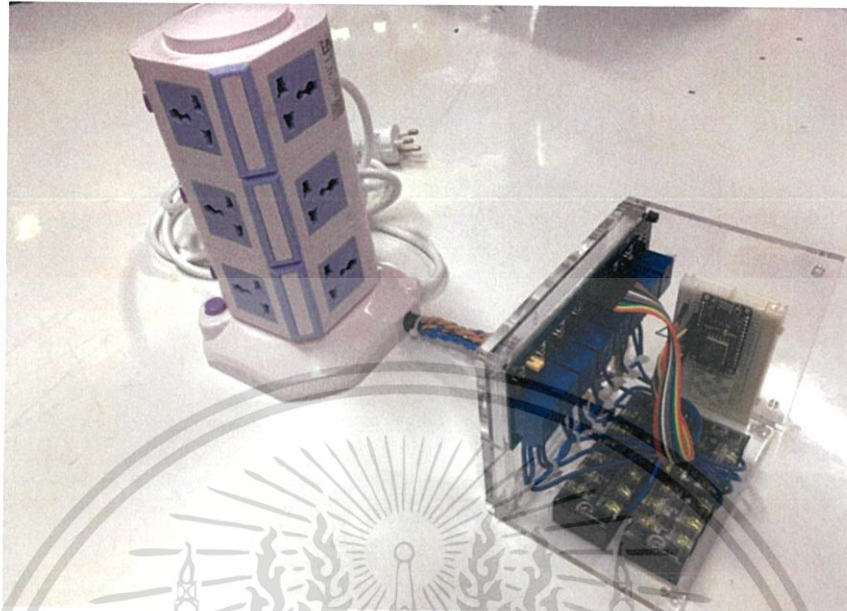


รูปที่ 4.2 ตัวอย่างเราท์เตอร์ตัวส่งสัญญาณ [18]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ชุดควบคุมต่างๆ

อุปกรณ์ควบคุมต่างๆที่ได้ประกอบไว้ สำหรับต่ออุปกรณ์ไฟฟ้าเข้ามายัง Tower ปลั๊ก



รูปที่ 4.3 Tower ปลั๊กพร้อมชุดควบคุม

4.2 การทดสอบควบคุมอุปกรณ์ไฟฟ้า

4.2.1 การติดตั้งแอปพลิเคชัน

1. ดาวน์โหลดแอปพลิเคชัน MIT AI2 Companion จาก Playstore มาไว้ใน

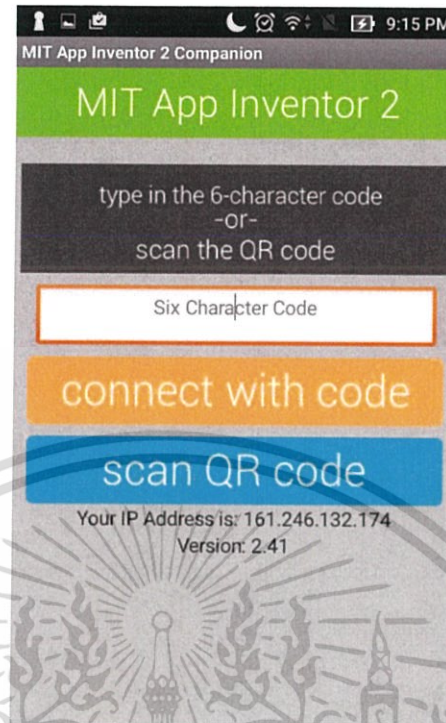
Smartphone ก่อน



รูปที่ 4.4 แสดงแอปพลิเคชัน MIT AI2 Companion

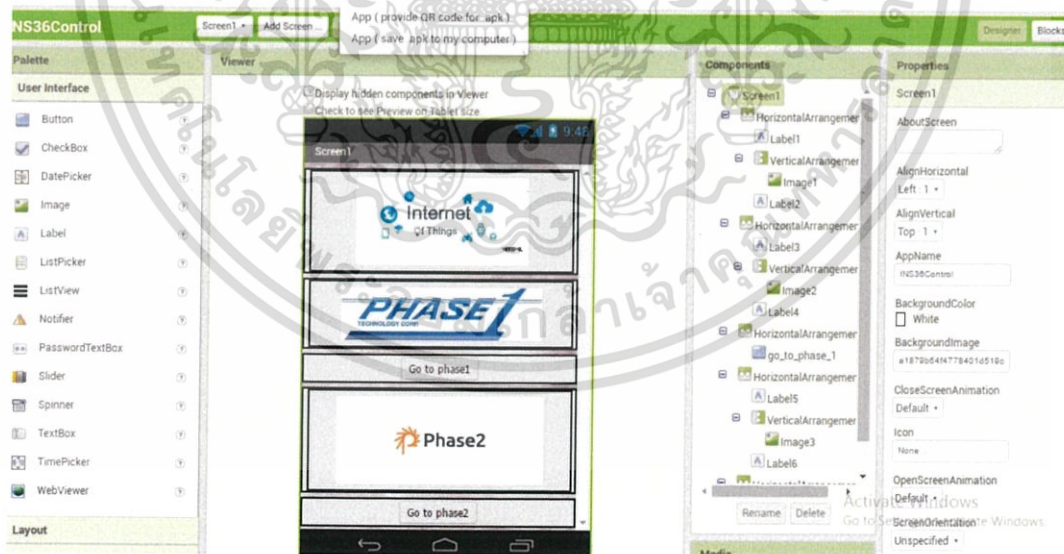
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ติดตั้งแอปพลิเคชันและเปิดแอปพลิเคชันบน Smartphone



รูปที่ 4.5 ภายในแอปพลิเคชัน MIT AI2 Companion

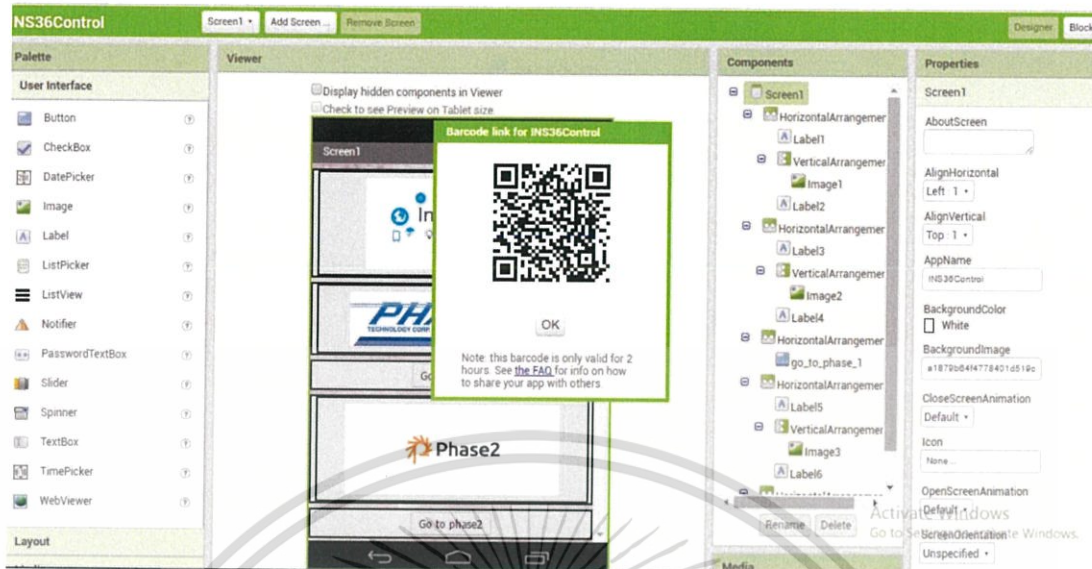
3. จากนั้นจะต้องทำการดาวน์โหลดแอปพลิเคชันที่เราสร้างจากเว็บไซต์ของ MIT APP Inventor2



รูปที่ 4.6 การโหลดแอปพลิเคชันที่สร้างจากหน้าเว็บไซต์ลง Smartphone

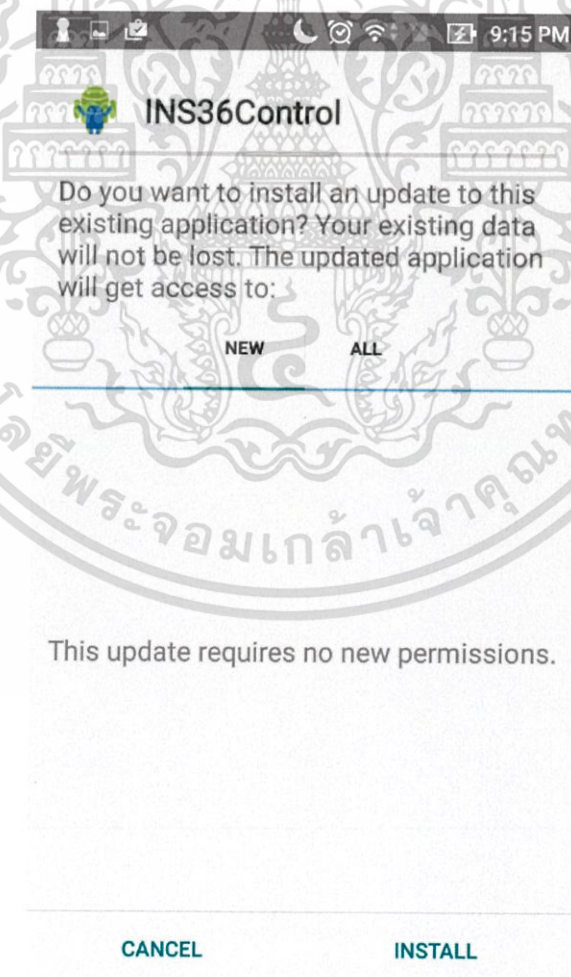
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. จากนั้นเลือกคำสั่ง Build และเลือกใช้ Provide QR Code for.apk



รูปที่ 4.7 การโหลดแอปพลิเคชันที่สร้างจากหน้าเว็บไซต์ลง Smartphone

5. ทำการสแกน QR Code ลงในแอปพลิเคชัน MIT AI2 บน Smartphone

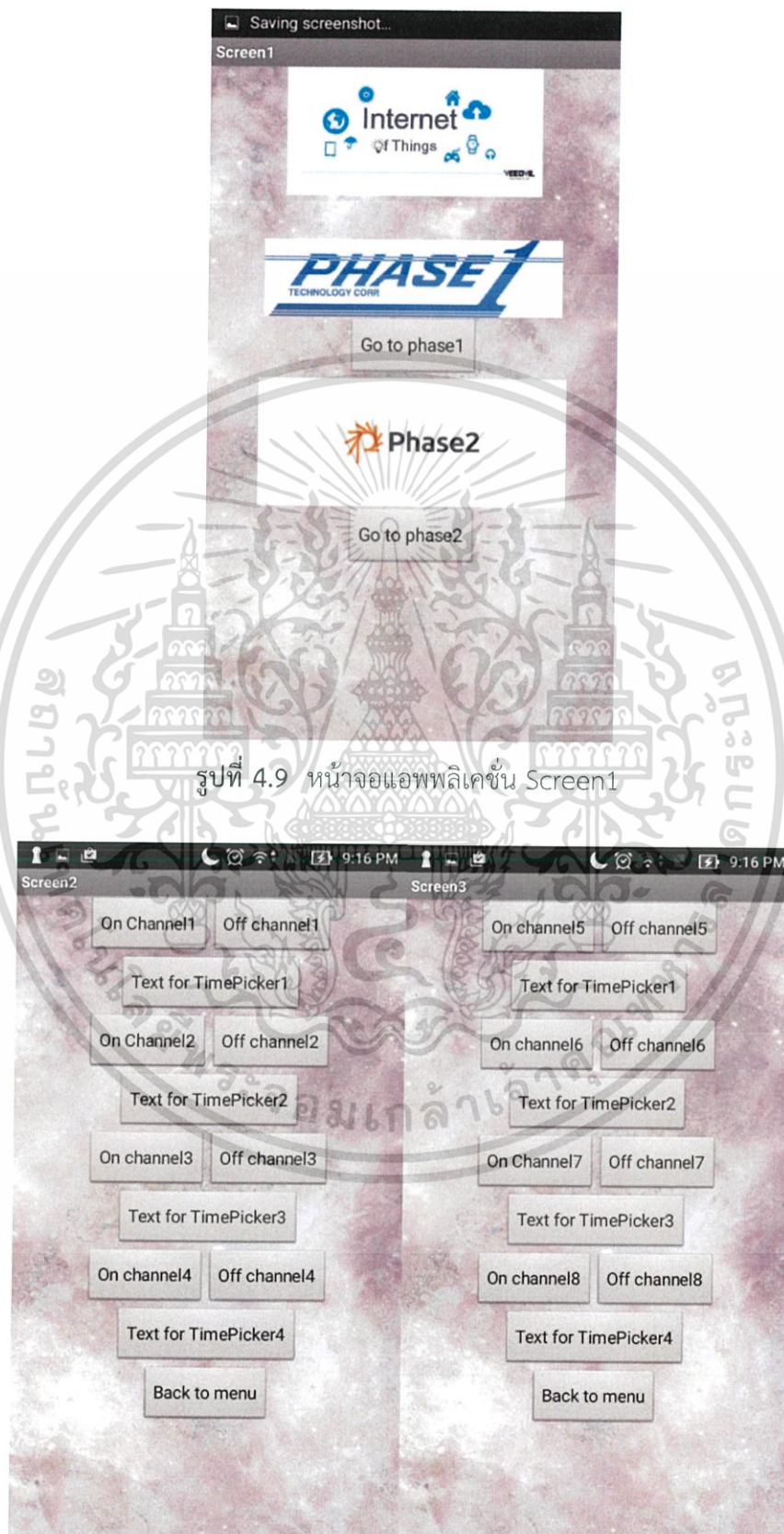


รูปที่ 4.8 หน้าจอ Install ของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสร้างขึ้นมา

6. กด Install แอปพลิเคชันลงบน Smartphone จากนั้นจะได้แอปพลิเคชันที่

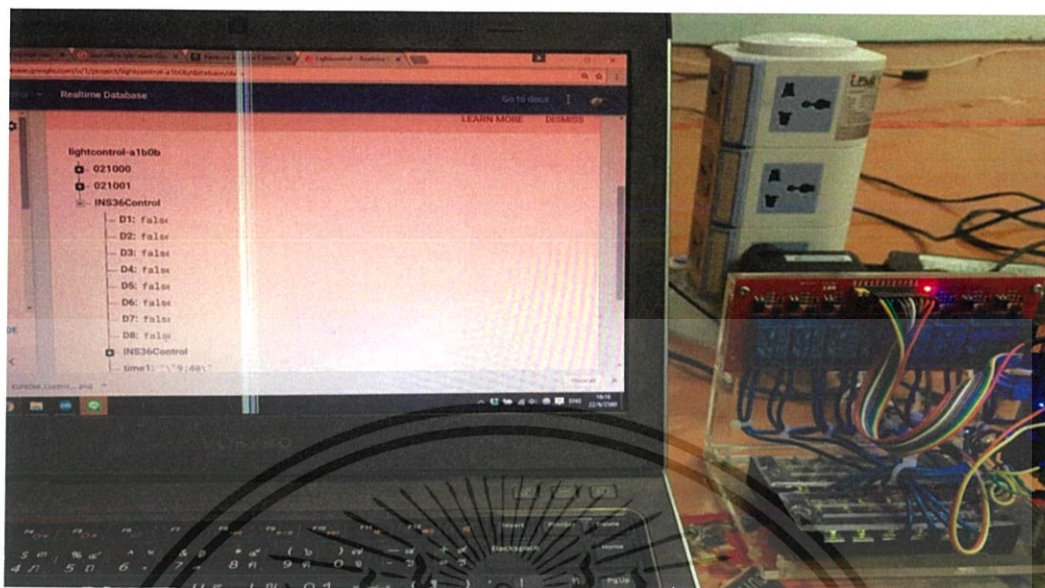


รูปที่ 4.9 หน้าจอแอปพลิเคชัน Screen1

รูปที่ 4.10 หน้าจอแอปพลิเคชัน Screen2 และ Screen3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทดลองเปิด-ปิดผ่าน Firebase



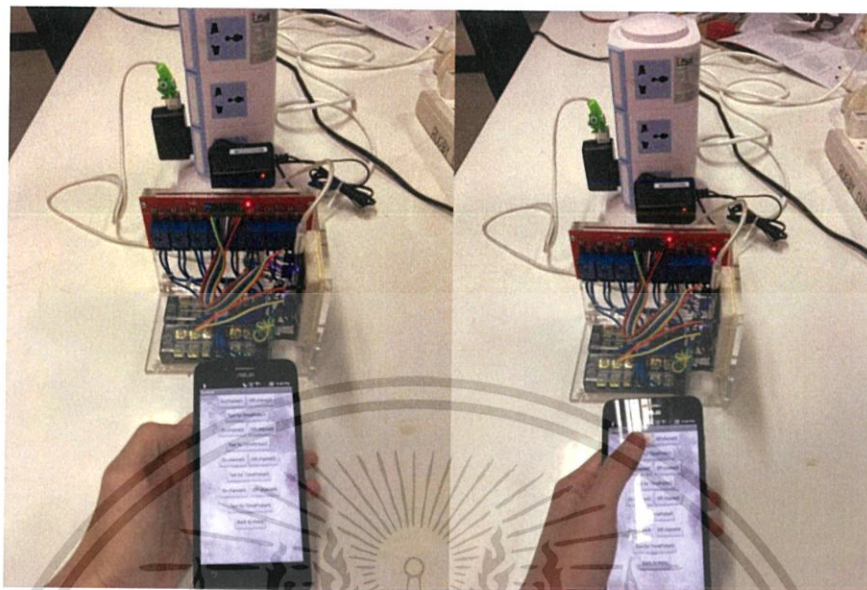
รูปที่ 4.11 การทดลองควบคุมการเปิดปิดผ่าน Firebase



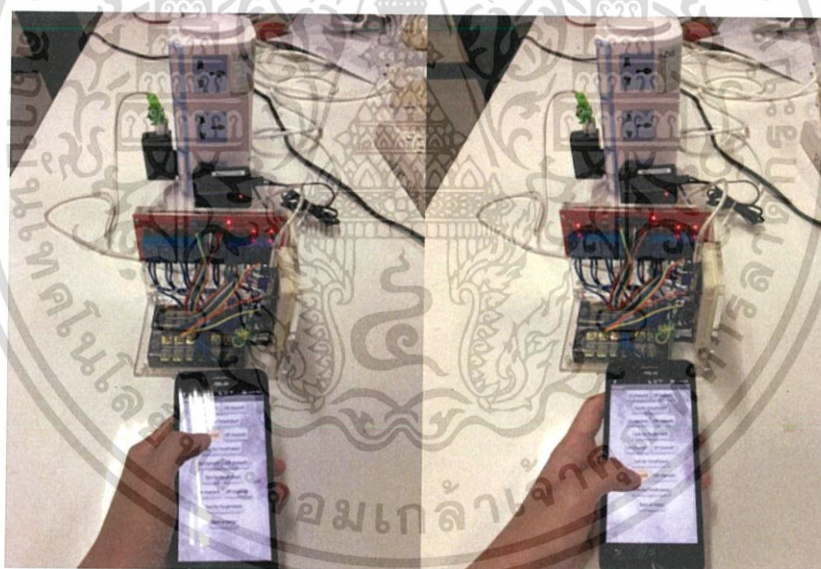
รูปที่ 4.12 แสดงการทดลองควบคุมการเปิดปิดผ่าน Firebase (2)

จากการทดลองควบคุมการเปิดปิดผ่าน Firebase จากรูปที่ 4.11 จะเห็นได้ว่าที่หน้า Firebase นั้น สถานการณ์ทำงานของแต่ละตัวมีค่าเป็น False ทั้งหมด และ Relay แต่ละตัวก็ยังมีสถานะเป็น Off อยู่ แต่ในรูปที่ 4.12 นั้น จะทำการเปลี่ยนสถานะของ D1, D2, D7 และ D8 ที่หน้า Firebase ให้มีสถานะเป็น True และจะสังเกตได้ว่า สถานะการทำงานของ Relay ตัวที่ 1, 2, 7 และ 8 ก็จะมีการเปลี่ยนสถานะเป็น ON ตามสถานการณ์ทำงานที่หน้า Firebase ด้วยเช่นกัน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดลองเปิด-ปิดผ่านแอปพลิเคชัน



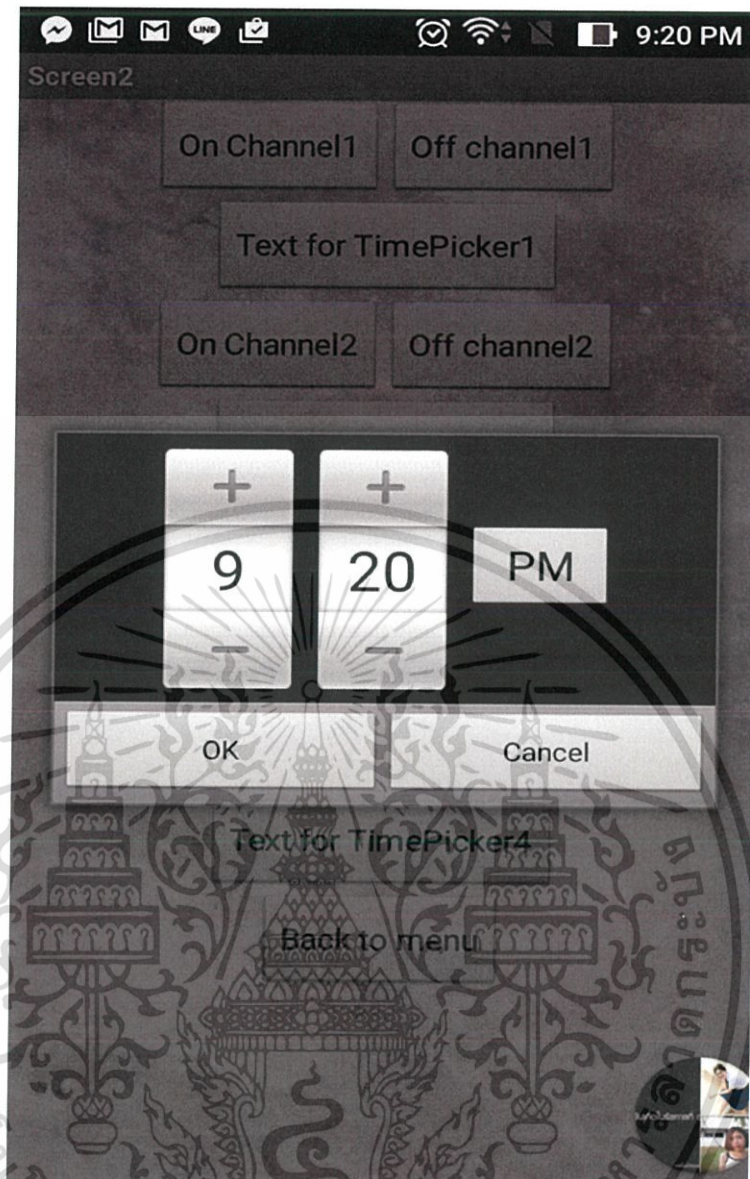
รูปที่ 4.13 แสดงการทดลองควบคุมการเปิดปิดผ่านแอปพลิเคชัน



รูปที่ 4.14 แสดงการทดลองควบคุมการเปิดปิดผ่านแอปพลิเคชัน (2)

จากการทดลอง จะเห็นได้ว่าในตอนแรก Relay แต่ละช่องยังมีสถานะเป็น Off ทั้งหมด แต่เมื่อกดปุ่ม On Channel1 ผ่านโทรศัพท์มือถือ Relay ตัวที่ 1 ก็ทำงาน เมื่อกดปุ่ม On Channel2 ผ่านโทรศัพท์มือถือ Relay ตัวที่ 2 ก็ทำงาน รวมถึงตัวอื่นๆด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงการทดลองใช้การตั้งเวลาผ่านแอปพลิเคชัน(3)

จากการทดลองควบคุมการเปิดปิดผ่านแอปพลิเคชันพบว่าสามารถควบคุมการเปิดปิด Relay ได้ทั้ง 8 Channel ตามปุ่มที่ได้ตั้งไว้บนแอปพลิเคชัน โดยจะเห็นได้ว่าเมื่อทำการกดปุ่ม ON Channel1 Relay ช่องที่ 1 ก็จะทำงาน และสาย Line ที่ดึงออกมาจาก Tower ปลั๊กก็จะมีไฟวิ่งกลับไปยังช่องที่ 1 ของปลั๊ก Tower ทำให้มีไฟที่ช่องที่ 1 ของปลั๊ก และช่องอื่นๆก็เช่นกัน เมื่อทำการกดปุ่มเปิดช่องใด Relay ช่องนั้นๆก็จะทำงาน และปลั๊ก Tower ตามช่องต่างๆที่เชื่อมกับ Relay อยู่ นั้น ก็จะมีไฟด้วยเช่นกัน รวมถึงการตั้งเวลาก็สามารถทำได้โดยการใช้ปุ่ม Timer for Channel1 จนถึง Timer for Channel8 ด้วยเช่นกัน โดยหลังจากกดปุ่มจะมีเวลาให้เลือก ก็สามารถที่จะเลือกเวลาที่ต้องการให้ Channel นั้นๆ เริ่มเปิดได้ตามต้องการ

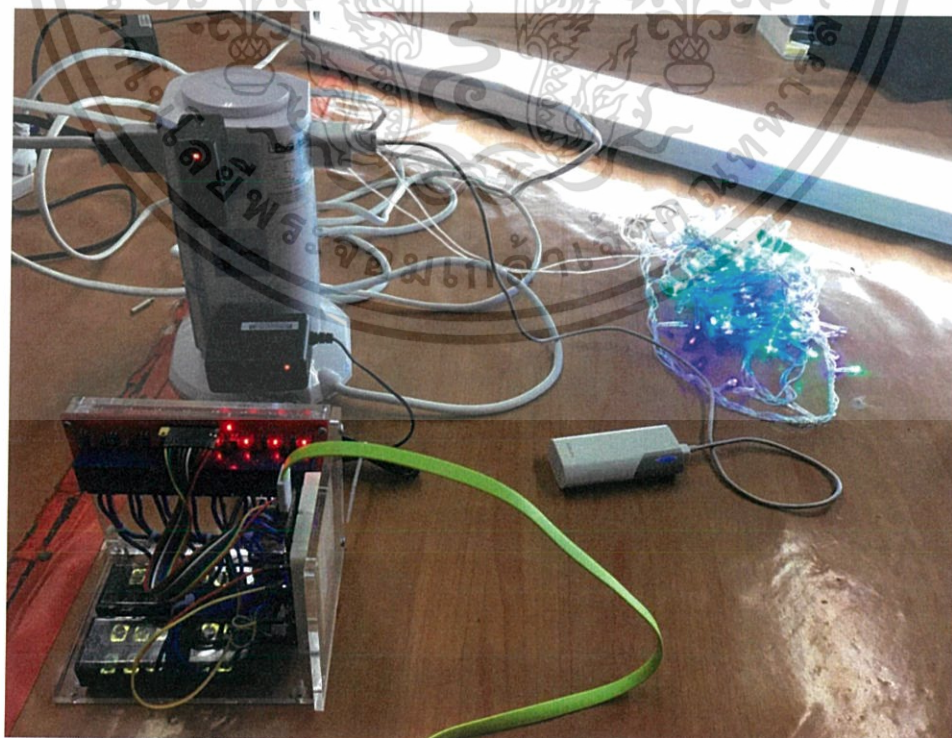
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การทดลองเปิด-ปิดร่วมกับเครื่องใช้ไฟฟ้า

การทดลองนี้ จะนำเครื่องใช้ไฟฟ้าต่างมารวมทดลองเพื่อให้เห็นผลลัพธ์ที่ชัดเจนขึ้น เช่น หลอดไฟกระพริบ หลอดไฟลูออเรสเซนต์ แบตสำรอง และพัดลม เป็นต้น



รูปที่ 4.16 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า

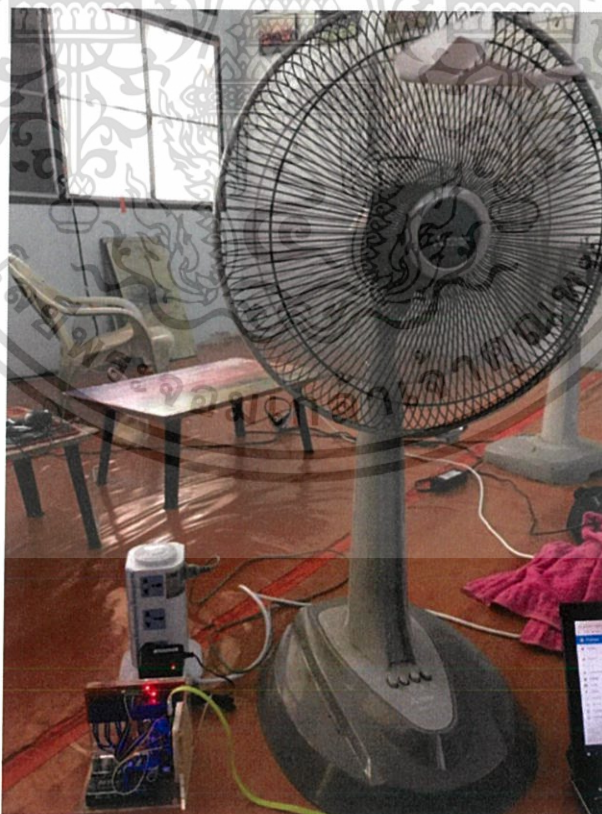


รูปที่ 4.17 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า (3)



รูปที่ 4.17 การทดลองควบคุมการเปิดปิดโดยต่อเข้ากับเครื่องใช้ไฟฟ้า (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.16 จะเห็นได้ว่า อุปกรณ์ไฟฟ้าที่นำมาต่อทั้งหมดยังไม่มีการทำงานใดๆ แต่หลังจากการทดลองการใช้งานโดยสั่งเปิดการทำงานของ Relay ในช่องที่อุปกรณ์นั้นๆต่ออยู่ คือ Relay ในช่องที่ 1-4 ซึ่งจะเชื่อมกับปลั๊ก Tower แถวบนทั้ง 4 ช่องอยู่ ก็จะพบว่าเมื่อ Relay ทั้ง 4 ช่องมีสถานะเป็น On ทั้งหมดแล้ว อุปกรณ์ไฟฟ้าก็จะทำงานตามสถานะการทำงานของ Relay ด้วย ดังรูปที่ 4.17 ส่วนรูปที่ 4.18 จะเห็นได้ว่าพัลลัมยังไม่มีการทำงาน แต่เมื่อได้เปิดการทำงานของ Relay ในช่องที่พัลลัมต่ออยู่ พัลลัมก็จะทำงานดังรูปที่ 4.19 นั้นเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผล

โครงการระบบตั้งเวลาเปิดปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต สามารถทำงานได้โดยการทำงานร่วมกันระหว่าง Firebase Esp8266 Module และ Relay 8 Channel Module โดยการนำ Firebase ไปสั่งงานให้ Esp8266 ทำงาน จากนั้น Esp8266 จะรับคำสั่ง แล้วสั่งงานไปยัง Relay ให้ทำการเปิดปิดการจ่ายไฟเข้าไปยังช่องต่างๆของปลั๊ก Tower เมื่อนำเครื่องใช้ไฟฟ้ามาเสียบเข้ากับปลั๊ก Tower ก็จะสามารถควบคุมการเปิดปิดได้จาก Smartphone หรือ Computer ได้ทันที โดยสามารถควบคุมจากที่ใดก็ได้ เพียงแค่มีอินเทอร์เน็ตเท่านั้น อย่างไรก็ตามโครงการระบบตั้งเวลาเปิดปิดเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต ยังมีข้อบกพร่องบางประการในการทำงาน ทางคณะผู้จัดทำจึงได้รวบรวมปัญหาที่เกิดขึ้น เพื่อที่จะเป็นแนวทางในการพัฒนาในอนาคตต่อไป

5.2 ข้อเสนอแนะ

เนื่องจากการทำงานของโครงการชุดนี้ มีข้อบกพร่องด้านการออกแบบและควบคุมอยู่บ้าง ซึ่งสามารถแยกแยะปัญหาต่างๆที่เกิดขึ้นได้ดังนี้

1. ปัญหา การติดต่อด้านฟังก์ชันของ Firebase และ Esp8266 ไม่สามารถติดต่อกันได้
แนวทางแก้ไข ต้องเขียนฟังก์ชันเสริมเพิ่มใน Library ของ Esp8266 เพิ่มจากตัวโค้ดหลัก
2. ปัญหา การติดต่อสื่อสารของ Esp8266 ไม่สามารถเชื่อม Internet ผ่านเครือข่ายภายในสถาบันได้เนื่องจากติด KMITL's Authentication System
แนวทางแก้ไข การคอนฟิก IP ของ Esp8266 ไปฝากไว้ที่ Router เพื่อให้ IP เป็นวงเดียวกัน หลังจากนั้นจึงทำการ Authentication ผ่านตัว Router

5.3 แนวทางการพัฒนา

1. สามารถพัฒนาให้เป็นระบบ Smart Home หรือ Smart Farm ได้
2. สามารถพัฒนาให้อุปกรณ์ทั้งหมด ใส่รวมเข้าไปในปลั๊ก เพื่อง่ายต่อการใช้งาน
3. พัฒนาให้ใช้งานร่วมกับอุปกรณ์ไฟฟ้าที่ต้องมีการควบคุมหลายทาง เช่น การปรับอุณหภูมิในเครื่องปรับอากาศ หรือการควบคุมการเปลี่ยนช่องของโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Internet of Things : โอกาสจากการเชื่อมต่อ.2015. [Online]. Available : <http://www.thaieasyelec.com/article-wiki/embedded-electronics-application/getting-started-with-esp8266-nodemcu-ch5.html>
- [2] บทความการใช้งานเริ่มต้น ESP8266 NodeMCU และการใช้งาน Application ต่างๆ. [Online]. Available : <http://www.thaieasyelec.com/article-wiki/embedded-electronics-application/getting-started-with-esp8266-nodemcu-ch5.html>
- [3] Internet of Things (IoT) คืออะไร. 2016. [Online] Available : <http://www.Mcuthailand.com/articles/iot/IOT.html>
- [4] Arduino. 2017. [Online]. Available : <https://en.wikipedia.org/wiki/Arduino>
- [5] จุดเด่นของบอร์ด Arduino. 2016. [Online]. Available : <https://embeddedbblog.wordpress.com/page/2/>
- [6] จุดเด่นของบอร์ด Arduino. 2016. [Online]. Available : <https://embeddedbblog.wordpress.com/page/2/>
- [7] รูปแบบการเขียนโปรแกรมบน Arduino. 2016. [Online]. Available : <http://www.Thaeasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html>
- [8] NodeMCU - Lua based ESP8266 development kit. 2016. [Online]. Available : <https://shop.pimoroni.com/products/nodemcu-v2-lua-based-esp8266-development-kit>
- [9] 8 Channel Relay 5V 10A (Optocoupler) รีเลย์ 8 ช่อง. 2017. [Online]. Available : <https://www.arduitronics.com/article/applications-of-relay-with-arduino>
- [10] Firebase helps you build better mobile apps and grow your business. อัปเดตล่าสุดเมื่อ มิถุนายน 16, 2017. [Online]. Available : <https://firebase.google.com/>
- [11] โครงสร้างข้อมูล. 2009. [Online]. Available : <http://piyapan-aod.blogspot.com/2009/03/tree.html>
- [12] ESP8266 ESP8285 กับการใช้งาน Firebase ระบบฐานข้อมูลเรียลไทม์จาก Google. 2016 [Online]. Available : <https://www.ioxhop.com/article/45>
- [13] ESP8266 ESP8285 กับการใช้งาน Firebase ระบบฐานข้อมูลเรียลไทม์จาก Google. 2016 [Online]. Available : <https://www.ioxhop.com/article/45>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] ESP8266 ESP8285 กับการใช้งาน Firebase ระบบฐานข้อมูลเรียลไทม์จาก Google. 2016 [Online]. Available : <https://www.ioxhop.com/article/45>
- [15] ESP8266 ESP8285 กับการใช้งาน Firebase ระบบฐานข้อมูลเรียลไทม์จาก Google. 2016 [Online]. Available : <https://www.ioxhop.com>
- [16] MIT App Inventor 2. Effective Date: April 20, 2015. [Online]. Available : <http://ai2.appinventor.mit.edu/>
- [17] 5 of Most Problematic Android Phones. April 15, 2016. [Online]. Available : <http://techiesprout.com/2016/04/15/5-problematic-android-phones/>
- [18] TOP TEN TIPS FOR SETTING UP YOUR NETWORK. 2017. [Online]. Available : <http://www.linksys.com/pt/tips-for-setting-up-your-router/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้