ระบบควบคุมการเข้าออกที่ชาญฉลาด

SMART ACCESS CONTROL SYSTEM

แทนไท เอียการนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ระบบควบคุมการเข้าออกที่ชาญฉลาด

SMART ACCESS CONTROL SYSTEM

แทนไท เอียการนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2559

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมการเข้าออกที่ชาญฉลาด

SMART ACCESS CONTROL SYSTEM

ผู้จัดทำ

1. นายแทนไท เอียการนา รหัสนักศึกษา 56010492

_____อาจารย์ที่ปรึกษา

(ผศ. ธนา หงษ์สุวรรณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมการเข้าออกที่ชาญฉลาด

นายแทนไท เอียการนา 56010492
ผศ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2559

บทคัดย่อ

ในปัจจุบันนั้นหลายๆสถานที่มีการใช้ระบบควบคุมการเข้าออกเพื่อรักษาความปลอดภัย มากมายหลายประเภท แต่ละประเภทก็มีระบบการล็อคประตูที่แตกต่างกันไป แต่จะมีปัญหาหนึ่งที่พบคล้ายๆกันในระบบควบคุมการเข้าออกแต่ละประเภทก็คือ ไม่สามารถที่จะตรวจสอบสถานะของระบบควบคุมการเข้าออกได้ ว่ายังทำงานอยู่ปกติหรือไม่

ในระบบการรักษาความปลอดภัยนั้นควรที่จะตรวจสอบได้ว่าระบบทำงานปกติหรือไม่ เพื่อจะเป็นการแน่ใจได้ว่า จะไม่เกิดช่องโหว่ที่จะทำลายการรักษาความปลอดภัยลงได้ ผู้จัดทำเล็งเห็นถึงปัญหานี้จึงได้มีแนวคิดที่จะพัฒนาระบบที่จะใช้ตรวจสอบและควบคุม ระบบควบคุมการเข้าออกอีกชั้นหนึ่ง โดยใช้อุปกรณ์อิเล็กทรอนิกส์หลายๆตัวมาติดต่อสื่อสารกันในรูปแบบของ Mesh Network

โดยที่ในส่วนของการเก็บข้อมูลนั้นจะใช้การส่งผ่านข้อมูลจากอุปกรณ์แต่ละตัวส่งต่อกันมายังฐานข้อมูล โดยที่ไม่ใช้อุปกรณ์เสริมภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Smart Access Control System

Mr. Tanthai Earkanna 56010492

Asst. Prof. Thana Hongsuwan Advisor

Academic Year 2016

ABSTRACT

At the present time a various of Access Control System is commonly use to secure the place and building, although each system have different lock system but each system share the same problem. The problem is we can't know if the system is working properly or not

In the security system it should be able to check the status of system in order to ensure that the system is still secure. We consider that the problem must be solved, so we develop the system that monitor the Access Control System. In this system we use multiple electronic devices to communicate each other in a Mesh Network type.

For the storage part, we gather information by passed through the data from each electronic device to the database

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงาน ระบบควบคุมการเข้าออกที่ชาญฉลาด หรือ Smart Access Control System สามารถสำเร็จลุล่วงไปได้ ต้องขอขอบพระคุณ อาจารย์ ผศ. ธนา หงษ์สุวรรณ ผู้คอยให้คำปรึกษาทั้งด้าน ปัญหาและการตัดสินใจในแนวทางการทำโครงงาน

ขอขอบคุณคณาจารย์ทุกท่านที่สั่งสมความรู้ให้ผู้จัดทำมาตั้งแต่อดีตจนถึงปัจจุบัน ส่งผลให้ผู้จัดทำได้มีความรู้ ความสามารถที่จะกระทำสิ่งต่างเพื่อให้ปริญญานิพนธ์เล่มนี้สมบูรณ์ได้จนสำเร็จซึ่งผู้จัดทำได้ซาบซึ้งในพระคุณของคณาจารย์เป็นอย่างยิ่ง

ขอขอบคุณคณะวิศวกรรมศาสตร์ ภาควิชาวิศกรรมคอมพิวเตอร์ ที่เอื้อเฟื้อสถานที่และทรัพยากรต่าง ๆ ในการทำวิจัย

ขอขอบคุณห้องวิจัย ISAG ที่เอื้อเฟื้อสถานที่ในการทำงานวิจัย

สุดท้ายนี้ขอขอบพระคุณครอบครัว เพื่อนๆและผู้คนรอบข้างทุกๆท่าน ที่คอยเป็นทั้งกำลังใจและกำลังทรัพย์อันส่งผลให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

แทนไท เอียการนา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ..... II
สารบัญ.....IV
สารบัญภาพ.....VI

บทที่ 1 บทนำ..... 1
1.1 ความเป็นมาของปัญหา 1
1.2 วัตถุประสงค์ของการศึกษา 1
1.3 ขอบเขตของโครงงาน 1
1.4 วิธีการดำเนินการ 1
1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 2
1.6 ส่วนประกอบของปริญญานิพนธ์..... 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง
2.1 ระบบ Mesh Network 3
2.2 NoSQL Database 3
2.3 Arduino IDE.....4
2.4 HTML..... 5
2.5 Firebase..... 6
2.6 JSON (JavaScript Object Notation) 8
2.7 NodeMCU.....13

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและการพัฒนา

3.1 ความต้องการของระบบ.....15

3.2 ภาพรวมของระบบ..... 16

3.3 Use Case Diagram.....17

บทที่ 4 การทดลองระบบ

4.1 ทดลองการใช้ไมโครคอนโทรลเลอร์ควบคุมระบบควบคุมการเข้าออก.....18

4.2 ทดลองการส่งข้อมูลแบบ Mesh Network ด้วย NodeMCU..... 19

4.3 ทดลองการติดต่อระหว่างไมโครคอนโทรลเลอร์กับฐานข้อมูล.....20

บทที่ 5 บทสรุปและข้อเสนอแนะ

5.1 บทสรุป.....21

5.2 ปัญหาและอุปสรรคที่พบ.....21

5.3 แนวทางการแก้ไข.....21

5.4 แนวทางการพัฒนาต่อ.....21

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป		หน้า
2.1	ตัวอย่างการเชื่อมต่อแบบ Mesh Network	3
2.2	หน้าต่างโปรแกรม Arduino IDE	5
2.3	Logo Firebase	6
2.4	Logo JSON	8
2.5	โครงสร้าง Object ในภาษา JSON	9
2.6	โครงสร้าง Array ในภาษา JSON	9
2.7	โครงสร้าง Value ในภาษา JSON	10
2.8	โครงสร้าง String ในภาษา JSON	11
2.9	โครงสร้าง Number ในภาษา JSON	11
2.10	NodeMCU Devkit 1.0 (ESP-12E) Version 2	14
2.11	NodeMCU Devkit 1.0 (ESP-12E) Version 2 Pin Map	14
3.1	ภาพรวมของระบบ	16
3.2	Use Case Diagram	17
4.1	วงจรที่นำมาทดลอง	18
4.2	กราฟแสดงจำนวนครั้งการปลดล็อกประตู	20

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

เนื่องจากในปัจจุบันมีมิจฉาชีพมากมาย ทำให้แต่ละสถานที่นั้นต้องมีระบบควบคุมการเข้าออกเพื่อควบคุมให้ปลอดภัยจากมิจฉาชีพมากยิ่งขึ้น แต่การควบคุมการเข้าออกปกตินั้นไม่สามารถทราบสถานะในขณะนั้นได้ว่าระบบยังทำงานเป็นปกติอยู่หรือเปล่า ทำให้เกิดความลำบากในการตรวจสอบ อาจจะทำให้เกิดช่องโหว่ในการรักษาความปลอดภัยได้

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อควบคุมดูแลสถานะของระบบควบคุมการเข้าออกผ่าน Web Server
2) เพื่อเป็นแนวทางในการทำ Mesh Network
3) เพื่อเป็นแนวทางในการพัฒนาการเขียนโปรแกรมบน NodeMCU ด้วย Arduino IDE

1.3 ขอบเขตของโครงงาน

1) พัฒนา Mesh Network โดยใช้ NodeMCU ให้ติดต่อสื่อสารกันได้
2) สามารถตรวจสอบสถานะของระบบล็อกประตูได้ผ่านWebserver

1.4 วิธีการดำเนินการ

1) ออกแบบโครงสร้างของระบบ
2) หาแนวทางการพัฒนาภาษาที่จะใช้เขียน ESP8266 เพื่อทำ Mesh Network
3) ศึกษาการพัฒนา NodeMCU ด้วย Arduino IDE
4) ศึกษาการเขียน Database โดยใช้ Framework Firebase
5) ศึกษาและทดลองเชื่อมต่อ NodeMCU แบบ Mesh Network
6) ศึกษาและทดลองเชื่อตต่อ NodeMCU กับ Database ของ Firebase
7) ทดลองติดตั้งระบบกับ ระบบควบคุมการเข้าออกเพื่อจำลองการใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้เรื่องการสร้าง Mesh Network ด้วย Node MCU

2) ได้รับความรู้เรื่อง NoSQL Database และการใช้ JSON

3) ได้รับความรู้เรื่องการเขียน web ด้วย HTML

1.6 ขอบเขตของโครงงาน

1) พัฒนา Mesh Network โดยใช้ Node MCU (ESP8266) ให้ติดต่อสื่อสารกันได้โดยไม่ต้องพึ่ง Router ภายนอก

2) สามารถตรวจสอบสถานะของระบบล็อคได้ผ่าน Firebase Webserver

1.7 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็นทั้งหมด5บท มีหัวข้อและรายละเอียดดังนี้

บทที่ 1 บทนำ จะนำเสนอเกี่ยวกับ ความเป็นมาของปัญหา วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง จะนำเสนอเกี่ยวกับทฤษฎีที่ถูกนำมาใช้ในการทำโครงงาน

บทที่ 3 การออกแบบและการพัฒนา จะนำเสนอเกี่ยวกับ รายละเอียดการออกแบบและการพัฒนาระบบ

บทที่ 4 การทดลองและผลการทดลอง จะนำเสนอเกี่ยวกับ การเตรียมการทดลองในส่วนต่างๆ กำหนดตามตัวแปรต้น ตัวแปรตามและตัวแปรควบคุมที่เหมาะสม เพื่อวัดผลการทดลองว่าเป็นไปตามเป้าหมายหรือไม่

บทที่ 5 บทสรุป จะนำเสนอเกี่ยวกับ บทสรุปของโครงงาน การวิจารณ์โครงงาน ข้อจำกัดของโครงงานและอุปสรรค์ต่างๆระหว่างการทำงาน ข้อเสนอแนะ และกิตติกรรมประกาศ

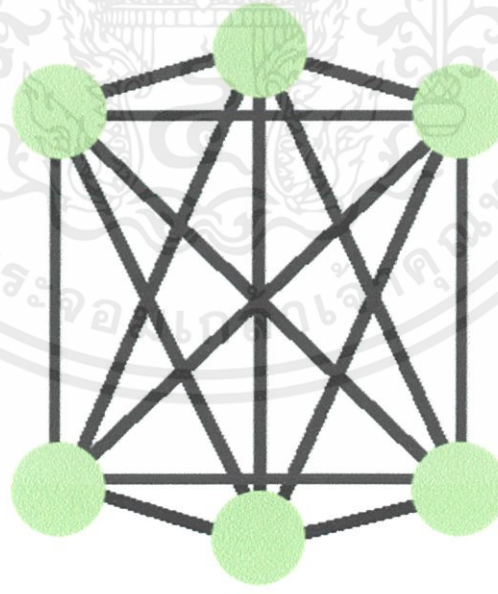
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎี,เทคโนโลยีและเครื่องมือที่เกี่ยวข้องในการสร้างระบบควบคุมการเข้าออกที่ชาญฉลาด

2.1 ระบบ Mesh Network

เป็น Topology รูปร่างเหมือนตาข่าย ปกติรูปแบบเครือข่ายนี้ ใช้ในระบบเครือข่ายที่มีบริเวณกว้างและต้องการความเสถียรสูง เพราะลักษณะการเชื่อมต่อจะมีการส่งข้อมูลจากโหนดไปยังโหนดอื่น ๆ ทุก ๆ ตัว ทำให้ข้อมูลสามารถเดินทางได้หลายทาง ทำให้ระบบมีความเสถียรมาก การล่มของโหนดใดโหนดหนึ่งจะไม่ทำให้ระบบล่ม ยังสามารถที่จะส่งข้อมูลผ่านไปทางโหนดอื่น ๆ ได้อยู่ ในอดีตไม่เป็นที่นิยมมากนักเพราะมีค่าใช้จ่ายในการเดินสายเชื่อมต่อสัญญาณค่อนข้างสูง แต่ในปัจจุบันนิยมนำมาใช้กับอุปกรณ์ Internet of Thing โดยใช้การเชื่อมต่อเป็นแบบไร้สายทำให้ประหยัดจากในอดีตไปได้มาก



รูปที่ 2.1 ตัวอย่างการเชื่อมต่อแบบ Mesh Network

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ฐานข้อมูลแบบ NoSQL

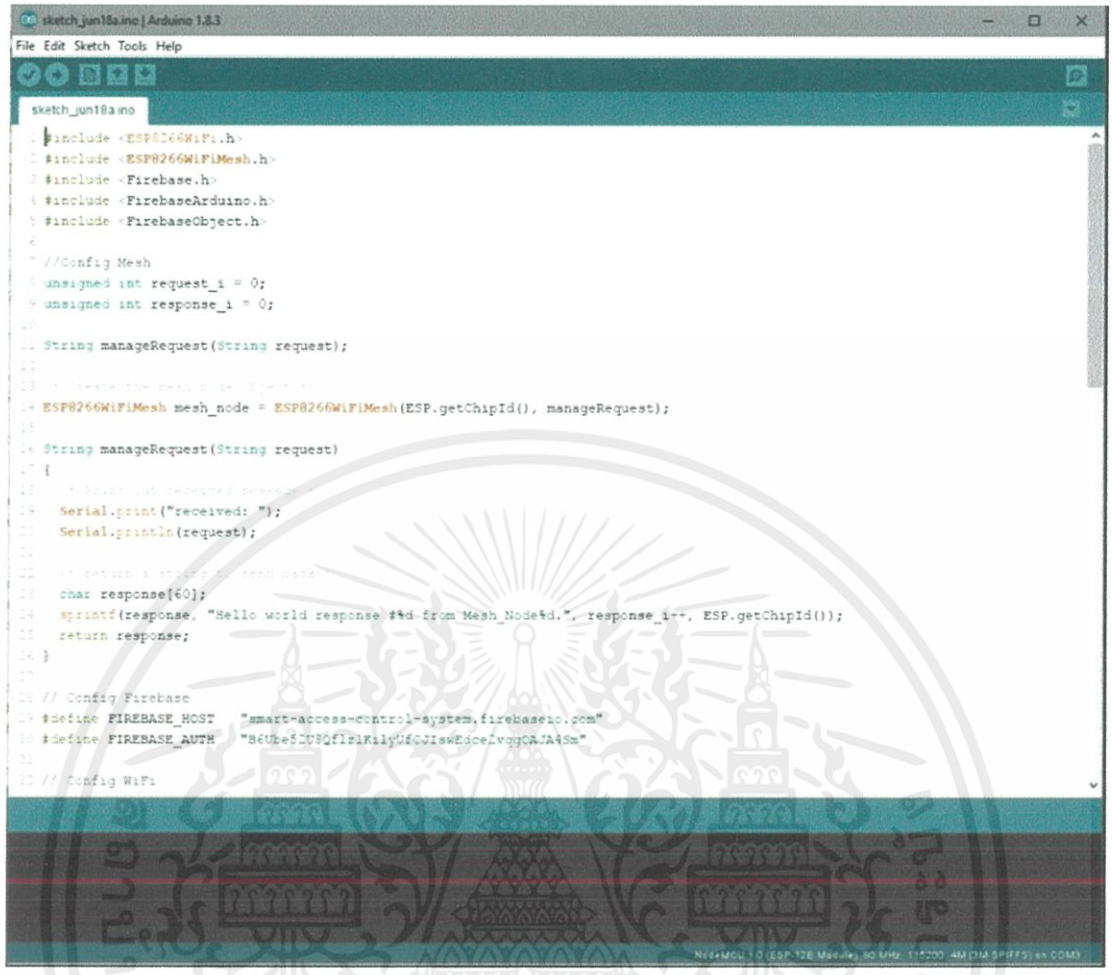
คือฐานข้อมูลชนิดหนึ่งที่เดิมแล้วหมายถึง “Non-relational” หรือฐานข้อมูลที่ไม่เป็นชนิดเชิงสัมพันธ์ โดยฐานข้อมูลนี้ให้กลไกในการเก็บและเรียกคืนข้อมูลซึ่งจะถูกจำลองในแบบที่ไม่ใช่ตารางเชิงสัมพันธ์ที่ถูกใช้ในฐานข้อมูลเชิงสัมพันธ์ ซึ่งปกติฐานข้อมูลเชิงสัมพันธ์นั้นมีตั้งแต่ยุคปลาย 1960 แต่ไม่ได้มี NoSQL อยู่ด้วย โดยตัวอย่างของฐานข้อมูล NoSQL ซึ่งมีหมวดหมู่และหมวดหมู่ย่อยแตกต่างกัน ซึ่งบางอย่างทับซ้อนกัน โดยการแยกแยะจะแยกโดยแบบจำลองของข้อมูลตัวอย่างดังต่อไปนี้

1) คอลัมน์ เช่น Accumulo, Cassandra, Druid HBASE, Vertica
2) เอกสาร (Document) เช่น Apache CouchDB, MongoDB, OrientDB
3) ค่าคีย์ (Key-value) เช่น Aerospike, Couchbase, Oracle NoSQL Database
4) กราฟ (Graph) เช่น Apache Giraph, OrientDB
5) หลายแบบจำลอง (Multi-model) เช่น Alchemy Database, ArangoDB, CortexDB, FoundationDB

2.3 Arduino IDE

ย่อมาจาก Arduino Integrated Development เป็นโปรแกรมฟรีที่ใช้พัฒนาซอฟท์แวร์สำหรับบอร์ดตระกูลอาดูโนและอื่น ๆ เพื่อที่จะเขียนโค๊ดโปรแกรมไปรันบนบอร์ดไมโครคอนโทรลเลอร์ต่างๆที่รองรับ มีเครื่องมือที่จะใช้ในการติดต่อสื่อสารกับ Arduino หรือบอร์ดที่มี Library รองรับ สามารถที่จะตรวจสอบรุ่นของอุปกรณ์ที่ใช้อยู่ได้ โดยมีพื้นฐานโปรแกรมจาก Java และ Open-Source Software อื่น ๆ โดยที่สามารถเขียน Code เป็นภาษา C++ ได้ ทำให้ง่ายต่อการเขียนโปรแกรมและมีความคุ้นชินในการเขียน และยังมี Source Code ตัวอย่าง ให้นำไปทดลองใช้หรือศึกษาเบื้องต้น ทั้งการใช้ Sensor ต่างๆหรือ การควบคุมอุปกรณ์พื้นฐานอย่าง LED เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.2 หน้าต่างโปรแกรม Arduino IDE

2.4 HTML

ย่อมาจาก Hypertext Markup Language คือภาษาที่ใช้ในการเขียนเว็บไซต์ สามารถเรียนรู้ใช้งานได้ง่าย โดยใช้ Tag ในการควบคุมการแสดงผลข้อความ รูป ภาพ เสียง และอื่นๆ เพื่อกำหนดโครงสร้างของเว็บ ให้เป็นไปตามที่เราต้องการ โดยสามารถเขียนบน Text Editor ต่าง ๆ เช่น Notepad หรือใช้โปรแกรมเครื่องมือช่วยสร้างเว็บเพจเช่น Dreamweaver แล้วนำไปรันบน Web browser เพื่อดูผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Firebase

Firebase นั้นเป็น Real-Time Cloud Database ที่ถูกออกแบบมาให้ใช้เป็น API และ Cloud Storage สำหรับการพัฒนา Application ในหลากหลาย Platform เช่น iOS, Android, Web app โดยที่ตัว Firebase นั้นเป็น NoSQL Database เก็บข้อมูลในรูปแบบของ JSON และยังสามารถที่จะทำงานได้แม้จะ Offline อยู่ โดยจะ Sync ข้อมูลใหม่เมื่อกลับมา Online อีกครั้ง และเรายังสามารถที่จะกำหนด Security Rules ได้เอง ทำให้เราสามารถที่จะออกแบบหรือกำหนดสิทธิการเข้าถึงในการอ่านและเขียนข้อมูลได้ตามที่ต้องการ

เรายังสามารถที่จะดูผลลัพธ์แบบ Real-Time ได้เลยที่ Firebase Console ของโปรเจคที่ทำอยู่ โดยจะมีสีที่แสดงบอกสถานะต่างๆใน Console ดังนี้

1) สีเขียว แสดงถึงการเพิ่มข้อมูล
2) สีเหลือง แสดงถึงการเปลี่ยนแปลงค่าข้อมูลเดิมที่มีอยู่
3) สีแดง แสดงถึงการลบข้อมูลนั้นๆ
4) สีน้ำเงิน แสดงถึงการเคลื่อนย้ายที่อยู่ข้อมูลนั้นๆ

รูป 2.3 Logo Firebase

2.5.1 ตัวอย่างคำสั่งการเขียนข้อมูลใน Firebase

การเขียนข้อมูล (Write) ใน Firebase นั้นรองรับค่าหลายประเภทเช่น String, Long, Double, Boolean, Map<String, Object> และ List<Object> โดยการเขียนข้อมูลนั้นจะมีด้วยกัน 4 รูปแบบดังนี้

1) setValue() เป็นการ Write หรือ Update ข้อมูลไปยัง Path ที่เราสามารถอ้างถึงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) push() เป็นการเพิ่มชุดของข้อมูล โดย Firebase นั้นจะสร้าง Unique Key ของข้อมูลชุดนั้นๆไว้ เพื่อที่จะไว้ใช้อ้างอิงต่อไป
3) updateChildren() เป็นการ Write หรือ Update ข้อมูลบางส่วนตามที่เราอ้างถึง โดยไม่ต้องทำการเปลี่ยนข้อมูลทั้งชุด โดยดูจากค่า Key โดยเราสามารถทำการแก้ไขพร้อมๆกันได้หลาย Object
4) runTransaction() เป็นการอัพเดทข้อมูล ที่มีการทำพร้อมกันมากๆ ซึ่งอาจจะทำให้เกิดการชนกันของข้อมูลแล้วเกิดข้อผิดพลาดขึ้นได้

2.5.2 ตัวอย่างคำสั่งการลบข้อมูลใน Firebase

การลบข้อมูล (Delete) ใน Firebase นั้นเราจะต้องทำการระบุ Path ที่เราต้องการจะจบ จากนั้นจึงเรียกใช้คำสั่ง removeValue() หรือจะใช้การส่งค่า null ไปด้วยการ setValue(null) หรือใช้ค่า null กับ updateChildren() เพื่อลบข้อมูลหลายๆ Object ได้อีกด้วย

2.5.3 ตัวอย่างคำสั่งการอ่านข้อมูลใน Firebase

การอ่านข้อมูล (Read) ใน Firebase นั้นเราจะต้องทำการประกาศตัวแปร DatabaseReference รับค่าตัวแปร และอ้างถึง Path ที่เราต้องการใน Database โดยการอ่านข้อมูลนั้นจะแบ่งออกได้เป็น 2 รูปแบบแยกประเภทตาม Listener ดังนี้

1) ValueEventListener จะอ่านข้อมูลตั้งแต่เริ่ม และ จะอ่านข้อมูลทุกครั้งที่มีการเปลี่ยนแปลงของข้อมูลทั้งหมดภายใต้ Path ที่เราอ้างถึง โดยจะใช้การเรียก Object ที่เราอ้างถึงมา addValueEventListener โดยจะมี Callback 2 รูปแบบ
    1) onDataChange จะถูกเรียกตอนเริ่ม และถูกเรียกทุกครั้งที่ข้อมูลภายใต้ Path ที่เราอ้างถึงมีการเปลี่ยนแปลงเกิดขึ้น
    2) onCancelled จะถูกเรียกเมื่อไม่สามารถอ่านข้อมูลจาก Database ได้
2) ChildEventListener จะอ่านข้อมูลจากการเพิ่ม, การเปลี่ยนแปลง, การลบ, การย้าย ของข้อมูลเฉพาะ Child ที่เราอ้างถึง โดยจะใช้การเรียก Object ที่เราอ้างถึงมา addChildEventListener โดยจะมี Callback 5 รูปแบบ
    1) onChildAdded() จะถูกเรียกเมื่อมีการเพิ่มชุดข้อมูลเข้ามาใน Child
    2) onChildChanged() จะถูกเรียกเมื่อข้อมูลใน Child มีการเปลี่ยนแปลง
    3) onChildRemoved() จะถูกเรียกเมื่อข้อมูลใน Child ถูกลบ
    4) onChildMoved() จะถูกเรียกเมื่อมีการเรียงลำดับของข้อมูลใน Child เกิดขึ้น
    5) onCancelled() จะถูกเรียกเมื่อเมื่อโหลดข้อมูลจาก Child ไม่สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 JSON (JavaScript Object Notation)

JSON เป็นรูปแบบการแลกเปลี่ยนข้อมูลน้ำหนักเบา ที่ง่ายต่อการอ่านและเขียนโดยมนุษย์ และยังง่ายต่อคอมพิวเตอร์ที่จะทำการสร้างและส่งต่อโดยสร้างขึ้นบนพื้นฐานของ JavaScript เป็นวิธีการที่ทำให้ Java Script แลกเปลี่ยนข้อมูลกับ Server ได้อย่างง่ายดาย โดย JSON นั้นเป็นภาษาที่ไม่ได้ขึ้นตรงกับภาษาใดเลย แต่มีความคล้ายคลึงกับภาษาตระกูล C เช่น C, C++, Java ทำให้เขียนได้ง่าย

โดย JSON นั้นมีโครงสร้างหลักๆในการจัดเก็บข้อมูลอยู่สองแบบคือ

1) ชุดข้อมูลของ คู่ของชื่อหรือค่าข้อมูล

2) ลำดับของค่าข้อมูลที่ถูกจัดเรียง โดยลำดับจะจัดอยู่ในรูปแบบ Array, Vector, List หรือ Sequence

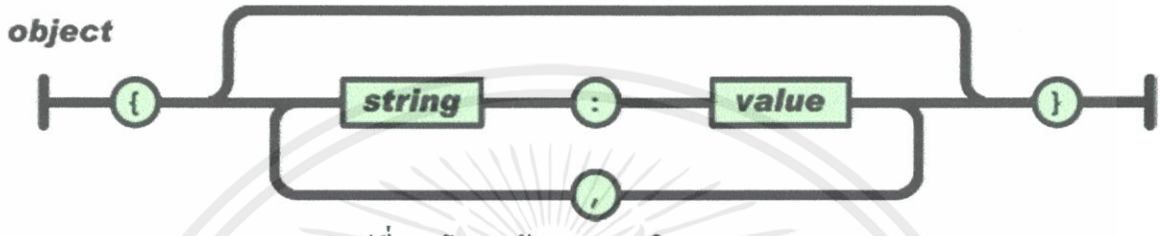
ในปัจจุบันมี Universal Data Structure ที่ภาษาเขียนโปรแกรมสมัยใหม่ล้วนให้การสนับสนุนในรูปแบบใดรูปแบบหนึ่งอยู่ และรูปแบบของข้อมูลที่ใช้แลกเปลี่ยนกับภาษาเขียนโปรแกรมอื่นๆทั้งหลาย เช่น JSON ก็สร้างอยู่บนพื้นฐานของโครงสร้างพวกนี้เพื่อที่จะได้ติดต่อสื่อสารกับภาษาอื่นๆได้อย่างราบรื่น

รูปที่ 2.4 Logo JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 Object

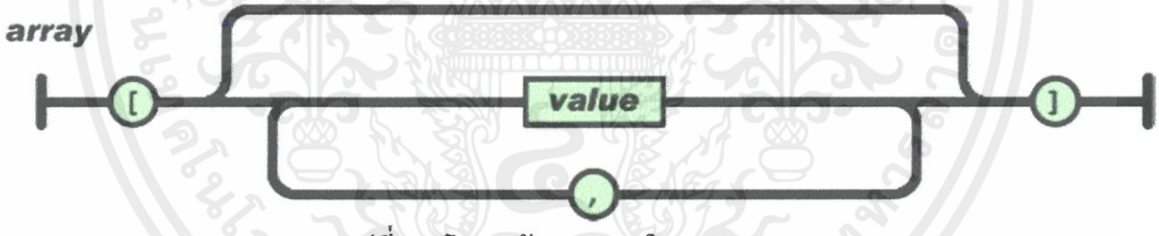
Object นั้นเป็นชุดของข้อมูลที่ไม่เรียงกัน โดยเป็นชื่อของข้อมูล และ ค่าของข้อมูลคู่กัน โดย Object นั้นจะเริ่มต้นด้วยเครื่องหมาย { (Left Bracket) และจะปิดท้ายข้อมูลด้วยเครื่องหมาย } (Right Bracket) ข้อมูลแต่ละค่าจะมีเครื่องหมาย : (Colon) กำกับไว้ระหว่างชื่อของข้อมูล กับ ค่าของข้อมูล และชุดข้อมูลแต่ละตัวจะถูกแยกจากกันโดยเครื่องหมาย , (Comma)



รูปที่ 2.6 โครงสร้าง Object ในภาษา JSON

2.6.2 Array

Array นั้นเป็นลำดับค่าของข้อมูล โดย Array นั้นจะเริ่มต้นด้วยเครื่องหมาย [(Left Bracket) และจะจบด้วยเครื่องหมาย] (Right Bracket) แต่ละค่าของข้อมูลจะถูกคั่นด้วยเครื่องหมาย , (Comma)

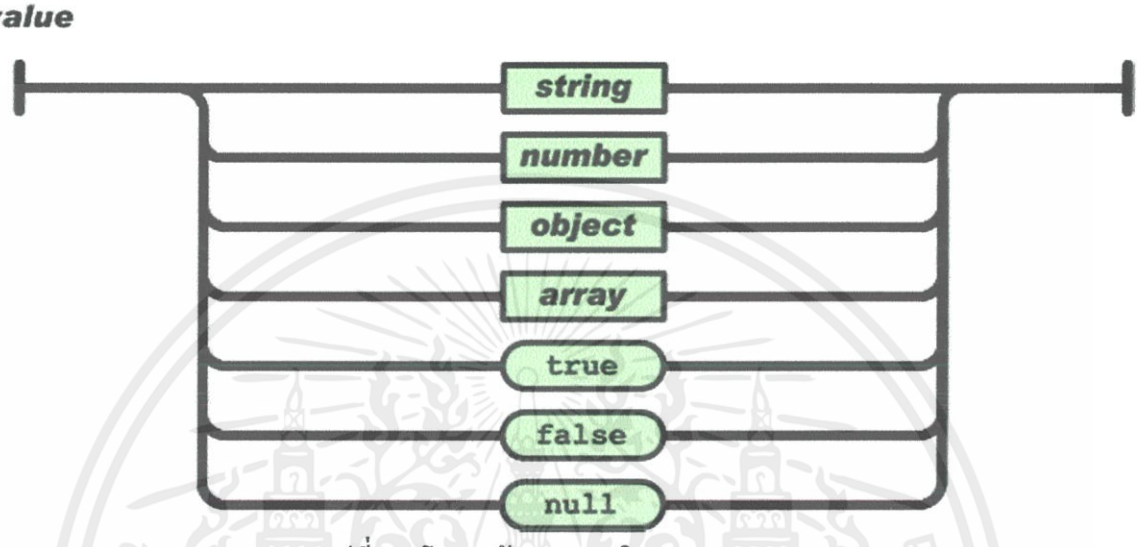


รูปที่ 2.7 โครงสร้าง Array ในภาษา JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 Value

Value นั้นสามารถที่จะเป็น String ในเครื่องหมาย ” “ (Double Quotes) หรือ Number, ค่าความจริง true/false/null, Object หรือ Array ก็ได้ โดยโครงสร้างนั้นสามารถที่จะทับซ้อนกันได้

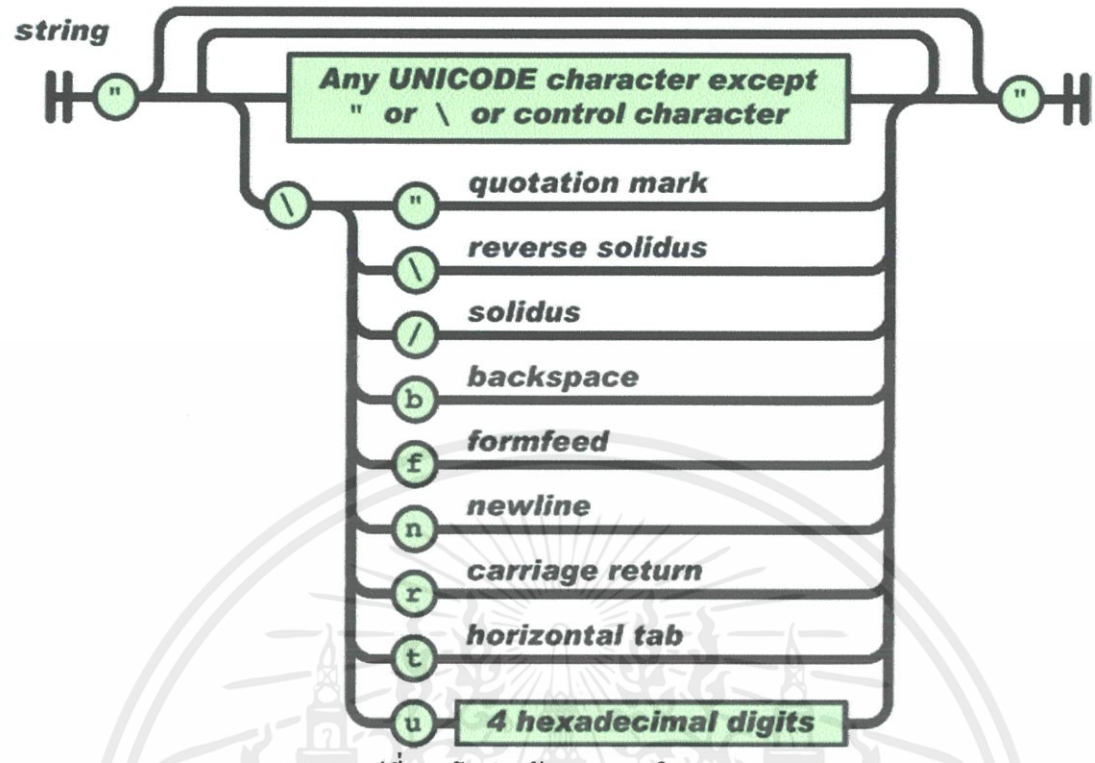


รูปที่ 2.8 โครงสร้าง Value ในภาษา JSON

2.6.4 String

String เป็นลำดับของตัวอักษร Unicode ตั้งแต่ 0 หรือมากกว่า ยกเว้นเครื่องหมาย ” ” (Double Quotes) และ / (Back Slash) ตัวอักษรเดี่ยวๆนั้นจะแสดงถึง String ซึ่งมีค่าตัวอักษรหนึ่งตัวบรรจุอยู่ ซึ่งมีลักษณะคล้ายกับ String ในภาษา C หรือ Java

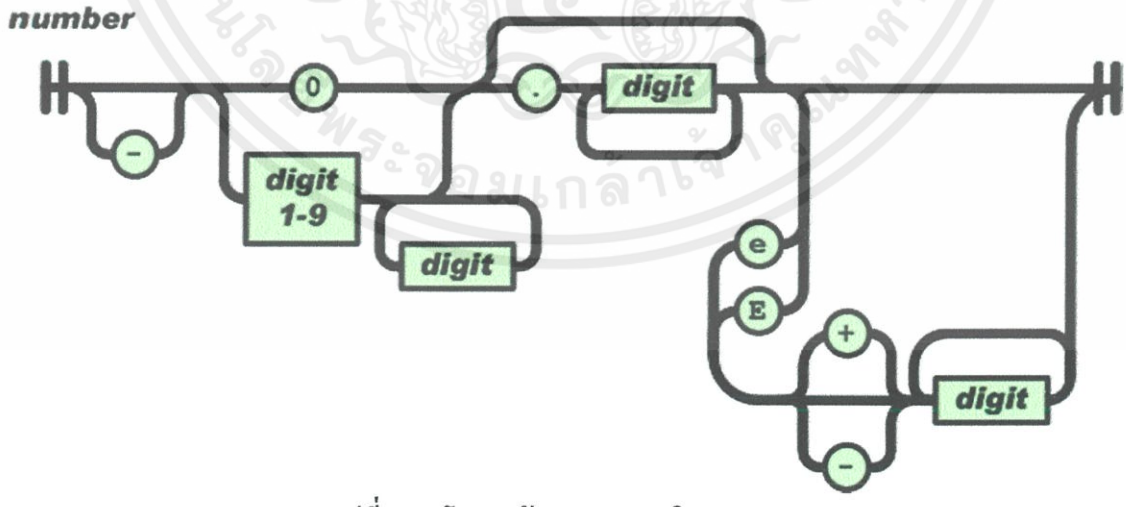
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 โครงสร้าง String ในภาษา JSON

2.6.5 Number

Number นั้นมีความคล้ายคลึงกับ Number ในภาษา C หรือ Java เป็นอย่างมาก แต่ไม่สามารถใช้เลขฐาน 8 กับเลขฐาน 16 ได้



รูปที่ 2.10 โครงสร้าง Number ในภาษา JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.6 ตัวอย่างโครงสร้างของ JSON

JSON นั้นใช้ลักษณะภาษาของ Javascript แต่ไม่ถูกมองเป็นภาษาเขียนโปรแกรม ถูกมองว่าเป็นภาษาในการแลกเปลี่ยนข้อมูล ในปัจจุบันมีไลบรารีของภาษาเขียนโปรแกรมอื่น ๆ ที่ใช้ประมวลผลโดยข้อมูลในรูปแบบ JSON มากมาย

ตัวอย่างโครงสร้าง JSON

```
[
 {"firstname":"name","lastname":"name"},
 {"firstname":"name1", "lastname":"name2"}
]
```

โดยJSON นั้นสามารถที่จะเก็บข้อมูลที่เป็นลักษณะของ Master – Detail ได้อีกด้วย

ตัวอย่างการจัดเก็บข้อมูลรูปแบบ Master - Detail

```
  [
    {"firstname" : "name",
     "lastname": "name",
    "address" : [
                {
                 "address1" : "adress",
                 "province" : "bangkok",
                 "country" : "Thailand"
                }
               ]
    }
  ]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 NodeMCU

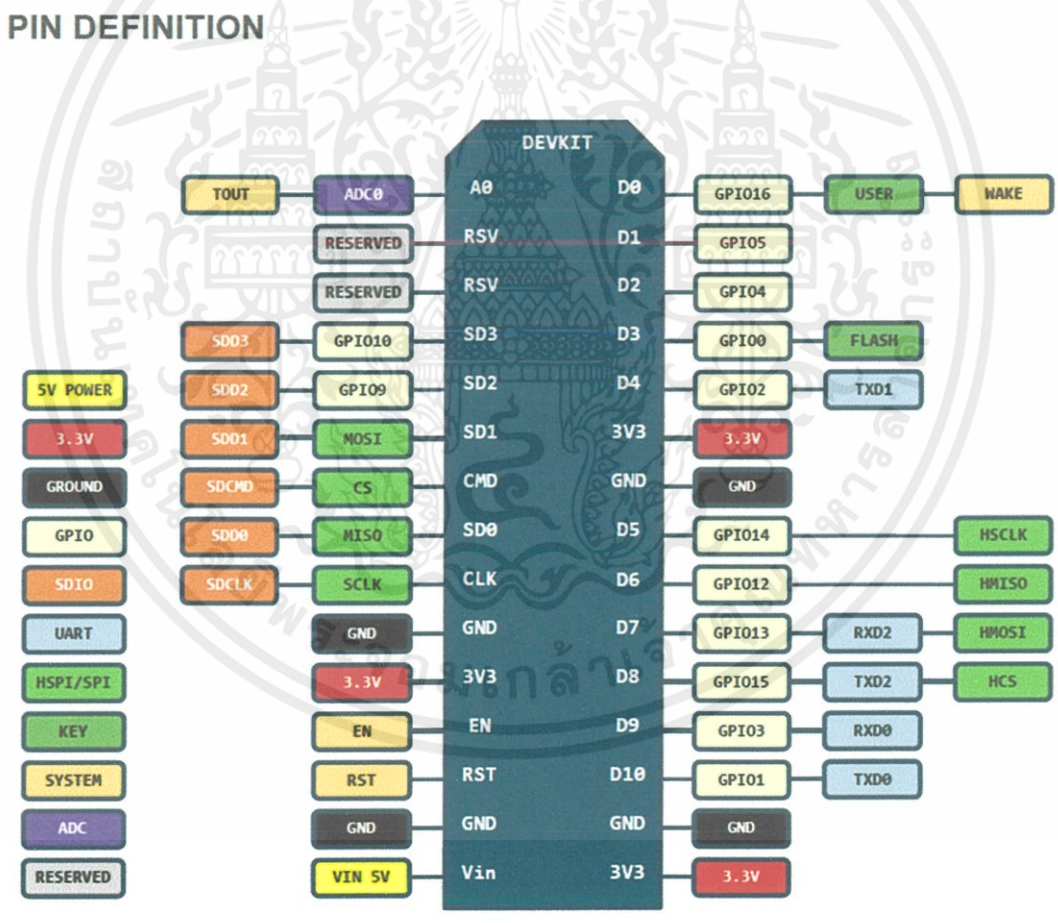
NodeMCU นั้นเป็นแพลตฟอร์มหนึ่งที่นิยมใช้ในการสร้างโปรเจคเกี่ยวกับ Internet of Thing (IoT) โดยตัว NodeMCU นั้นจะประกอบไปด้วยสองส่วนคือ ตัว Development Kit หรือตัวบอร์ดที่เป็นฮาร์ดแวร์ กับ Firmware ที่เป็นซอฟท์แวร์ที่ใช้บนบอร์ดในช่วงแรกๆนั้น NodeMCU จะใช้ภาษา Lua ในการเขียนโปรแกรมสั่งการ แต่ในภายหลังกลุ่มนักพัฒนาของ ESP8266 ก็ได้ปรับปรุงให้ NodeMCU สามารถเขียนด้วย Arduino IDE ได้ จึงทำให้เขียนโปรแกรมด้วยภาษา C/C++ แทนได้ ซึ่งง่ายและสะดวกกว่าการเขียนโปรแกรมด้วยภาษา Lua และยังทำให้สั่งการได้หลากหลายมากยิ่งขึ้นอีกด้วย ในปัจจุบัน นิยมนำ NodeMCU ไปใช้ในเรื่องที่เกี่ยวกับ Internet of Thing เช่นการใช้เป็น Web Server ขนาดเล็ก, ใช้ควบคุมอุปกรณ์วัด อุณหภูมิ และความชื้น, ใช้เปิดปิดไฟ ผ่าน Wi-Fi

2.7.1 ข้อดีของบอร์ด NodeMCU

1) ตัวบอร์ดนั้นจะมาพร้อมกับ Module Wi-Fi ESP8266 ซึ่งทำให้การเชื่อมต่ออินเทอร์เน็ตเป็นเรื่องง่ายขึ้นมาก โดยโมดูล ESP8266 นั้นมีหลายรุ่นมาก แต่รุ่นที่ใช้กับ NodeMCU รุ่นล่าสุด คือ ESP-12E และด้วยเหตุนี้ ทำให้ NodeMCU มีต้นทุนในการเริ่มใช้งานที่ต่ำมาก
2) NodeMCU นั้นมีลักษณะคล้ายกับ Arduino คือ มีพอร์ต Input และ Output มาให้ในตัวด้วยเลย และถึงแม้จะมีจำนวนขาไม่มากเท่า Arduino แต่สามารถเขียนโปรแกรมลงบนขา GPIO ได้ทุกขาและยังใช้ไฟแรงดัน +3.3 V ทำให้สามารถเขียนควบคุมอุปกรณ์ I/O ต่างๆได้ หลากหลาย โดยไม่ต้องใช้ อุปกรณ์เพิ่มเติม อีกทั้งตัวบอร์ดเองก็มีขนาดพอดีกับขนาดของบอร์ดทดลอง ทำให้สามารถ ใช้บอร์ดทดลองอันเดียวกันกับอุปกรณ์ I/O ต่างๆได้
3) ชิปภายใน ESP8266 นั้นมี CPU ขนาด 32 Bit ซึ่งใหญ่กว่า CPU ขนาด 8 Bit ของ Arduino ทำให้ทำงานได้รวดเร็วกว่า มีพลังในการประมวณผลมากกว่า
4) สามารถ Upload โปรแกรมที่เขียนลงบอร์ดได้ง่าย ผ่านการเชื่อมต่อพอร์ต USB
5) ชิป Wi-Fi ESP-12E มีจำนวนพินที่มากและพื้นที่แฟลชที่มากถึง 32 Mbits
6) มีปุ่ม RST ไว้ใช้สำหรับรีเซตการทำงานได้เวลาเกิดปัญหา หรือจะเริ่มทดลองใหม่ และปุ่ม Flash ไว้ใช้สำหรับการ Flash Firmware ใมห่
7) มีขา A0 ที่ใช้รับ Input แรงดัน Analog สำหรับวงจร ADC ขนาด 10 บิต อ่านค่าตั้งแต่ 0V ถึง 1V

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 NodeMCU Devkit 1.0 (ESP-12E) Version 2



รูปที่ 2.6 NodeMCU Devkit 1.0 (ESP-12E) Version 2 Pin Map

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและพัฒนา

เนื้อหาในบทนี้นั้นจะกล่าวถึงการออกแบบและพัฒนาระบบ โดยมีเนื้อหาเกี่ยวกับภาพรวมความต้องการของระบบ และการออกแบบการทำงานต่างๆ โดยแบ่งเป็น 3 ส่วน คือ ความต้องการของระบบ ภาพรวมของระบบ และ Use Case Diagram

โดยในโครงงานนี้จะเป็นระบบที่มีไว้เพื่อควบคุมดูแลระบบควบคุมการเข้าออก ตรวจสอบสถานะ เก็บข้อมูลแล้ว นำข้อมูลส่งผ่านอุปกรณ์ NodeMCU แต่ละ Node เพื่อที่จะส่งออกไปยัง Gateway เพื่อเชื่อมต่อไปยังอินเทอร์เน็ตอีกทีหนึ่ง โดยข้อมูลจะถูกส่งจาก Gateway ไปรวบรวมยังฐานข้อมูล Firebase แล้วจึงนำข้อมูลในฐานข้อมูลนั้นไปประมวลผลมาพล็อตเป็นกราฟ แสดงไปยังหน้าเว็บเพจอีกทีหนึ่ง โดยผู้ใช้จะสามารถเลือกช่วงเวลา ความถี่ที่ต้องการจะดูข้อมูลได้ โดยข้อมูลจะเป็นกราฟแยกแต่ละ Node ไป

3.1 ความต้องการของระบบ

3.1.1 Input

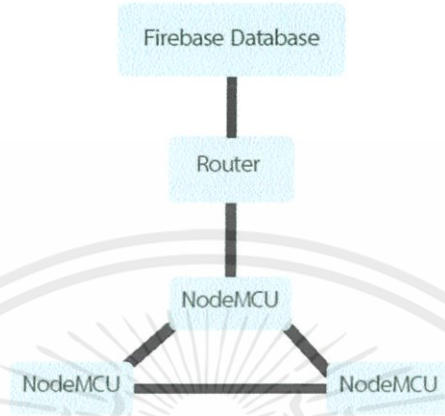
1) ข้อมูลการปลดล็อกประตู
2) ข้อมูลเวลาที่ทำการปลดล็อก

3.1.2 Output

1) กราฟแสดงผลจำนวนครั้งการปลดล็อกในหนึ่งช่วงเวลา
2) หน่วยความละเอียดของกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ภาพรวมของระบบ



รูปที่ 3.1 ภาพรวมของระบบ

ระบบประกอบไปด้วย

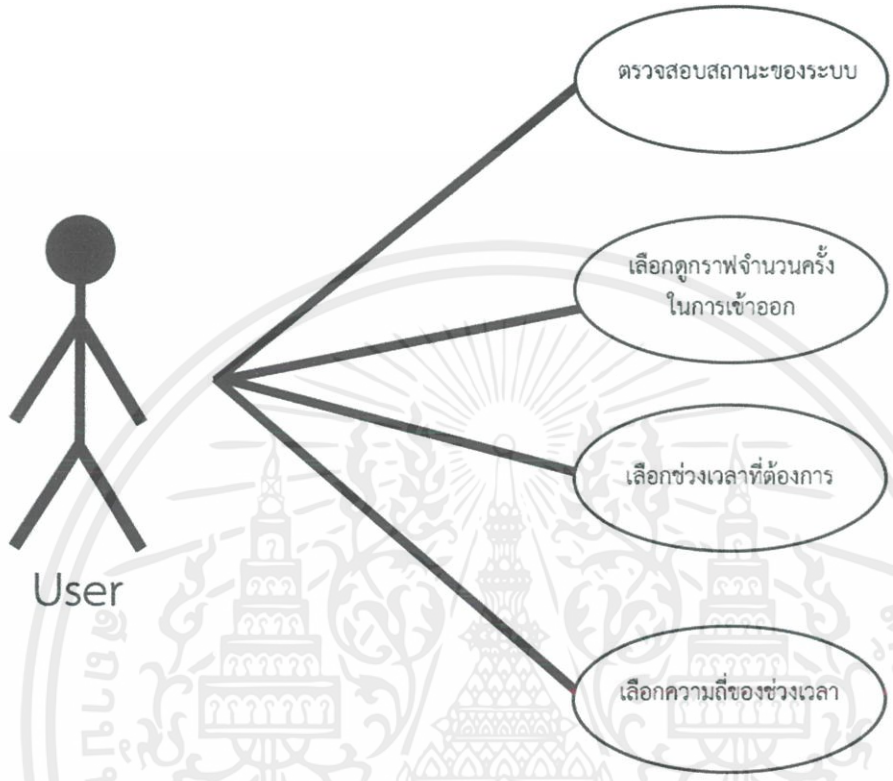
1) อุปกรณ์ Node MCU
2) ฐานข้อมูล Firebase
3) อุปกรณ์ Router สำหรับเชื่อมต่ออินเทอร์เน็ต

ตัวระบบนั้นจะติดต่อกันผ่าน Wi-Fi ด้วยตัวอุปกรณ์ NodeMCU เองโดยอุปกรณ์ทุกตัวจะเชื่อมต่อกับอุปกรณ์ในระยะ เป็น Mesh Network และจะมีอุปกรณ์อยู่ 1 ตัวที่จะทำการเชื่อมต่อกับอุปกรณ์Gateway Router รับ IP Address และเชื่อมต่อไปยัง Internet เพื่อที่จะใช้ติดต่อไปยังฐานข้อมูลและแลกเปลี่ยนข้อมูลกับฐานข้อมูลต่อไป

ในฐานข้อมูลของเรานั้นจะใช้เป็นแบบ NoSQL เพื่อที่จะทำให้ใช้งานได้ง่ายและเก็บข้อมูลได้มาก โดยใช้เครื่อง Server ที่ไม่ต้องใหญ่ก็สามารถทำงานได้ โดยที่จะใช้ ฐานข้อมูลบน Web server ของ Framework Firebase เพราะว่าเป็น Real Time Database ซึ่งจะมีการแก้ไขเปลี่ยนแปลงข้อมูลได้รวดเร็ว มีการแก้ไขข้อมูลที่แสดงผลทันทีที่มีการเปลี่ยนแปลงข้อมูลนั้น และยังสามารถที่จะทำงานต่อได้แม้ไม่มีการเชื่อมต่ออินเทอร์เน็ต โดยจะใช้ข้อมูลที่เก็บไว้ที่ตัวอุปกรณ์เองจนกว่าจะกลับมาเชื่อมต่ออินเทอร์เน็ตอีกครั้ง จึงจะทำการตรวจสอบการเปลี่ยนแปลงของข้อมูลกับฐานข้อมูลอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Use Case Diagram



รูปที่ 3.2 Use Case Diagram

ในระบบผู้ใช้สามารถเช็คสถานะของระบบได้ตลอดเวลา พร้อมกับเลือกช่วงเวลา ความถี่ของเวลาที่ต้องการจะตรวจสอบได้ เลือก Node ที่ต้องการจะดูได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

โครงงานนี้เป็นการสร้างระบบที่คอยเก็บข้อมูลการเข้าออกผ่านทางไมโครคอนโทรลเลอร์และส่งข้อมูลกับแบบ mesh network เพื่อที่จะส่งข้อมูลไปยัง node สุดท้ายที่เชื่อมต่ออินเทอร์เน็ต แล้วนำไปเก็บยังฐานข้อมูลออนไลน์ มีการทดลองสามประเภทคือ

4.1 ทดลองการใช้ไมโครคอนโทรลเลอร์ควบคุมระบบควบคุมการเข้าออก

4.1.1 จุดประสงค์การทดลอง

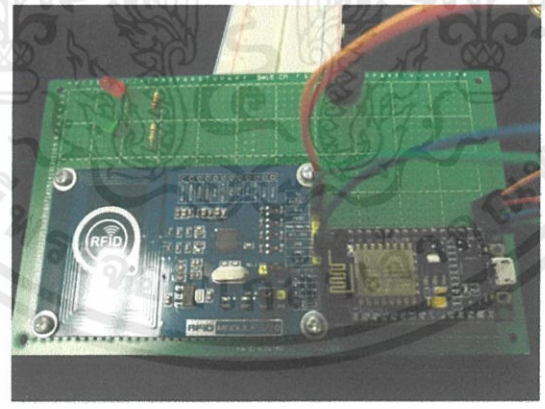
การทดลองนี้จะทดลองการใช้ไมโครคอนโทรลเลอร์ควบคุมวงจรระบบควบคุมการเข้าออก ว่าสามารถสั่งการได้อย่างถูกต้องหรือไม่

4.1.2 วิธีการทดลอง

ใช้ NodeMCU สั่งการวงจรที่นำมาทดลอง

4.1.3 ผลการทดลอง

สามารถใช้ NodeMCU สั่งการวงจรระบบได้ถูกต้อง



รูปที่ 4.1 วงจรที่นำมาทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ทดลองการส่งข้อมูลกันแบบ Mesh Network ด้วย NodeMCU

4.2.1 จุดประสงค์การทดลอง

การทดลองนี้จะทดลองการใช้ NodeMCU ซึ่งเป็น Module Wi-Fi ติดต่อกันเองในรูปแบบ Mesh network ไม่พึ่งอุปกรณ์ภายนอกว่าสามารถส่งข้อมูลได้หรือไม่

4.2.2 วิธีการทดลอง

ติดตั้ง NodeMCU ที่เขียนโค๊ดเรียบร้อยแล้วให้ห่างกันในระยะพอสมควรแล้ว ตรวจดูการส่งข้อมูลทาง Serial monitor

4.2.3 ผลการทดลอง

อุปกรณ์ NodeMCU สามารถติดต่อกับแบบ Mesh network ได้อย่างไม่มีปัญหา ในระยะสั้นๆ ที่ไม่มีสิ่งกีดขวาง หรือสัญญาณรบกวนมากนัก แต่เมื่อผ่านกำแพงหลายชั้น หรือระยะทางเริ่มห่างค่อนข้างมาก สัญญาณจะค่อย ๆ อ่อนลงจนขาดการติดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดลองการติดต่อระหว่างไมโครคอนโทรลเลอร์กับฐานข้อมูล

4.3.1 จุดประสงค์การทดลอง

การทดลองนี้จะทดลองการใช้ NodeMCU ติดต่อกับฐานข้อมูลออนไลน์ Firebase ว่าสามารถแลกเปลี่ยนข้อมูลได้ถูกต้องหรือไม่

4.3.2 วิธีการทดลอง

นำวงจรระบบทั้งหมดมาเชื่อมต่อกันแล้วจำลองการเข้าออกเพื่อให้มีการบันทึกข้อมูลส่งไปยังฐานข้อมูล แล้วไปเช็คที่หน้าเว็บว่าข้อมูลแสดงผลถูกต้องหรือไม่

4.3.3 ผลการทดลอง

ระบบสามารถเก็บบันทึกข้อมูลแล้วส่งไปยังฐานข้อมูลเพื่อนำมาแสดงผลในรูปแบบกราฟได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ระบบควบคุมการเข้าออกที่ชาญฉลาดที่ผู้จัดทำได้พัฒนาขึ้น เป็นระบบที่สามารถตรวจสอบสถานะของระบบผ่านทางเว็บไซต์ได้ โดยใช้การนำข้อมูลบันทึกการเข้าออกจากฐานข้อมูล Firebase ที่จัดเก็บข้อมูลไว้ในรูปแบบ JSON มาแสดงผลบนหน้า Website

5.2 ปัญหาและอุปสรรคที่พบ

1) ถ้าเชื่อมต่ออินเทอร์เน็ตไม่ได้เป็นเวลานานๆ จะทำให้ไม่สามารถทราบสถานะของระบบได้
2) วงจรระบบควบคุมการเข้าออกมีการกินกระแสไฟมากเกินกว่าที่บอร์ด NodeMCU จะสามารถจ่ายไฟได้ไหว
3) การเชื่อมต่อ NodeMCU กับ Database ของ Firebase แลกเปลี่ยนข้อมูลเป็นเวลานานๆ จะทำให้เมมโมรี่เต็ม ส่งผลกระทบให้ NodeMCU หยุดทำงานไปชั่วขณะ

5.3 แนวทางการแก้ไข

1) สำรองไฟให้ระบบและระบบเครือข่าย ปรับปรุงให้ระบบเครือข่ายมีความเสถียรมากขึ้น
2) ต้องหาแหล่งจ่ายไฟภายนอกเข้ามาช่วยแล้วแปลงความต่างศักย์มาใช้งาน
3) กำหนดเวลา Sleepให้ระบบทำการรีเซตตัวเองหลังครบกำหนดเวลาที่ตั้งไว้

5.4 แนวทางการพัฒนาต่อ

1) พัฒนาให้ระบบสามารถควบคุมเปิดปิดระบบล็อกผ่านเว็บเซิฟเวอร์ได้
2) พัฒนาให้ระบบล็อกสามารถระบุจำนวนคนเข้าออกได้แม่นยำ เที่ยงตรงมากยิ่งขึ้น
3) พัฒนา Mobile Application เพื่อเพิ่มความสะดวกในการใช้งานให้มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Ian F. Akyildiz , Xudong Wang z , Weilin Wang. 2004. **Wireless mesh networks: a survey.** Georgia Institute of Technology.

ESP8266 GitHub : **ESP8266**

[Online]. Available : https://github.com/esp8266/Arduino

MongoDB : **NoSQL Explain**

[Online]. Available : https://www.mongodb.com/nosql-explained

Ascend Developer : **Firebase Realtime Database**

[Online]. Available : https://developers.ascendcorp.com/รู้จัก-firebase-realtime-database-ตั้งแต่-zero-จนเป็น-hero-5d09210e6fd6#.r37haj74m

JSON.org : **Introducing JSON**

[Online]. Available : http://www.json.org/

W3.org : HTML & CSS

[Online]. Available https://www.w3.org/standards/webdesign/htmlcss

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้