

ระบบอนุรักษ์พลังงานจากพัดลมไอน้ำเย็น
Energy Conservation System from Evaporative Fan



ธนากร จวงเจิม
พศวัต ไกรศรีสมบัติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยสยาม
ปีการศึกษา 2561

ระบบอนุรักษ์พลังงานจากพัดลมไอน้ำเย็น
Energy Conservation System from Evaporative Fan

ธนากร จวงเจิม
พศวัต ไกรศรีสมบัติ

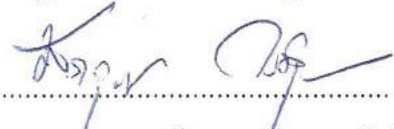
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยสยาม
ปีการศึกษา 2561


..... ประธานกรรมการสอบปริญญานิพนธ์
(อาจารย์ตะวัน ภู่วัต)


..... กรรมการ
(อาจารย์ขวัญชัย กังเจริญ)


..... กรรมการผู้ทรงคุณวุฒิ
(พล.อ.ต.ผศ.ดร.พารณ สงวนโกคัย)


..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ดร.กาญจนา ศิลาวราเวช)


..... คณบดีคณะวิศวกรรมศาสตร์
(ผู้ช่วยศาสตราจารย์สุราษฎร์ วรสุมนต์)

หัวข้อปริญญาานิพนธ์	ระบบอนุรักษ์พลังงานจากพัดลมไอน์	
หน่วยกิตของปริญญาานิพนธ์	2 หน่วยกิต	
โดย	นายธนากร จวงเจิม	5804000006
	นายพศวัต ไกรศรีสมบัติ	5804000007
อาจารย์ที่ปรึกษา	อาจารย์ตะวัน ภูริต	
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	2561	

บทคัดย่อ

การอนุรักษ์พลังงานมีผลแตกต่างกันไปขึ้นอยู่กับพฤติกรรมของพนักงานโดยเฉพาะอย่างยิ่งในเรื่องการควบคุมอุณหภูมิของสภาพแวดล้อม ความต้องการมีระบบอัตโนมัติเพื่อตัดทอนค่าไฟฟ้าในช่วงเวลาที่ไม่จำเป็นออกเช่นตอนพักเที่ยงจึงเป็นสิ่งจำเป็น ในบทความนี้เป็นการประยุกต์ใช้แนวคิดอินเทอร์เน็ตของสรรพสิ่งด้วยไอเอสพี 8266 โหนดเอ็มซียู, เซ็นเซอร์ดีเอชที22, ไวไฟ 2.4 จิกะเฮิร์ตซ์ และภูเกิลคลาวด์ ได้รับเพื่อสร้างระบบอนุรักษ์พลังงานจากพัดลมไอน์ การใช้ภูเกิลไฟร์เบสเป็นฐานข้อมูลทำให้ระบบนี้สามารถติดตามข้อมูลการใช้พลังงานเพื่อนำไปวิเคราะห์ได้ ผลการทดสอบยังแสดงให้เห็นว่าพนักงานสามารถตรวจสอบค่าอุณหภูมิและค่าความชื้น และสามารถกำหนดเวลาเปิดหรือปิดล่วงหน้าได้ทำให้ลดต้นทุนค่าไฟฟ้าในช่วงเวลาที่ไม่จำเป็นลงได้

คำสำคัญ: อินเทอร์เน็ตของสรรพสิ่ง / โหนดเอ็มซียู / จาวาสคริปต์ / ไฟร์เบส / พัดลมไอน์

Project Title Energy Conservation System From Evaporative Fan
Project Credit 2 Credits
By Mr. Thanakorn Joungjerm 5804000006
 Mr. Potsawat Krairisombat 5804000007
Project Advisor Mr. Tawan Phurat
Degree Bachelor of Engineering
Department Computer Engineering
Academic Year 2018

Abstract

Due to the fact that energy conservation varies on employee's behavior especially in the ambient temperature control, it's required an autonomous system to eliminated the unnecessary cost of electricity such as leaving work at noon. Therefore, the application of the Internet of Things, such as ESP8266 NodeMCU, DHT22 Ambient-Sensor, 2.4 GHz WiFi, and Google Cloud has been proposed in this paper in order to implement an energy conservation system from the evaporative fan. Through use of Google Firebase as a database can track energy usage information in order to make data analysis. Additionally, the testing results has been illustrated so employee can monitor temperature, humidity and can set a time to open or close in advance to reduce the cost of air-conditioning during unnecessary time.

Keywords: Internet of Things / NodeMCU / JavaScript / Firebase / Evaporative fan

Approved by

.....

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ตะวัน ภูริต อาจารย์ที่ปรึกษา ซึ่งกรุณาสละเวลา ให้ความรู้และคำแนะนำตลอดการทำโครงการ

ขอขอบพระคุณ คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสยาม ตลอดจนผู้เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามมา ณ ที่นี้

ท้ายที่สุด ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ผู้เป็นที่รัก ผู้ให้กำลังใจและให้โอกาส การศึกษาอันมีค่ายิ่ง



คณะผู้จัดทำโครงการ

สารบัญ

	หน้า
บทคัดย่อ (ภาษาไทย)	ข
บทคัดย่อ (ภาษาอังกฤษ)	ค
กิตติกรรมประกาศ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ
บทที่ 1 บทนำ	
1.1 ที่มาของโครงการ	1
1.2 คำสำคัญ	2
1.3 วัตถุประสงค์ในการทำโครงการ	2
1.4 ขอบเขตความสามารถของโครงการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 เครื่องมือ, เครื่องวัด และอุปกรณ์ที่ใช้	3
1.7 แผนการดำเนินงาน	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 Evaporative Fan (พัดลมไอน้ำเย็น)	5
2.2 IOT (Internet of things)	8
2.3 Arduino	12
2.4 ESP8266 NodeMCU	13
2.5 DHT22	13
2.6 Xampp	15
2.7 Visual Studio Code	15
2.8 Firebase	16
2.9 HTML	20
2.10 JavaScript	21
บทที่ 3 การวิเคราะห์และการออกแบบ	
3.1 ความต้องการของผู้ใช้งาน	22

สารบัญ (ต่อ)

	หน้า
3.2 การวิเคราะห์จากความต้องการของผู้ใช้งาน	22
3.3 การออกแบบระบบ	25
3.4 การทำงานของระบบ	25
3.5 แผนภาพแสดงสถานะ (State Diagram)	25
3.6 แผนภาพการไหลของข้อมูล (Context Diagram)	27
3.7 Data Flow Diagram (DFD)	28
3.8 Process Description	28
บทที่ 4 การทดลอง	
4.1 การกำหนดพอร์ตของ ESP8266 NodeMCU	31
4.2 การกำหนดไฟฟ้าภายใน	32
4.3 การทดลองเปิดพัดลมไอน์บนเว็บแอปพลิเคชัน	32
4.4 การทดลองปิดพัดลมไอน์บนเว็บแอปพลิเคชัน	33
4.5 การทดลองกำหนดเวลาเปิดพัดลมไอน์โดยอัตโนมัติ	34
4.6 การทดลองกำหนดเวลาปิดพัดลมไอน์โดยอัตโนมัติ	35
4.7 ทดลองการตรวจสอบค่าอุณหภูมิและค่าความชื้น	36
4.8 ทดลองเปิดและปิดปั๊มน้ำ	38
4.9 การทดลองเปิดระบบสายของพัดลมไอน์บนเว็บแอปพลิเคชัน	39
4.10 การทดลองปิดระบบสายของพัดลมไอน์บนเว็บแอปพลิเคชัน	40
4.11 การทดลองเลือกแรงลมของพัดลมไอน์บนเว็บแอปพลิเคชัน	41
บทที่ 5 สรุปผลการทดลอง	
5.1 สรุปผลการดำเนินโครงการ	43
5.2 ปัญหาการดำเนินโครงการ	44
5.3 ข้อเสนอแนะ	44
บรรณานุกรม	45
ภาคผนวก	46
ประวัติผู้จัดทำ	52

สารบัญตาราง

	หน้า
ตารางที่ 1.1 การวางแผนการดำเนินงาน	4
ตารางที่ 2.1 ตารางแสดงอุณหภูมิและความสามารถในการทำความเย็น	7
ตารางที่ 2.2 ตารางเปรียบเทียบความแตกต่างของอุณหภูมิ	8
ตารางที่ 2.3 การส่งข้อมูลของแต่ละ Access technologies	9
ตารางที่ 3.1 คำอธิบายการประมวลผลของกระบวนการที่ 1 : เปิด/ปิดเครื่อง	28
ตารางที่ 3.2 คำอธิบายการประมวลผลของกระบวนการที่ 2 : เปิด/ปิดระบบสาย	29
ตารางที่ 3.3 คำอธิบายการประมวลผลของกระบวนการที่ 3 : เปิด/ปิดปั๊มน้ำ	29
ตารางที่ 3.4 คำอธิบายการประมวลผลของกระบวนการที่ 4 : กำหนดเวลาเปิด/ปิดเครื่อง	29
ตารางที่ 3.5 คำอธิบายการประมวลผลของกระบวนการที่ 5 : ควบคุมแรงลม	30
ตารางที่ 3.6 คำอธิบายการประมวลผลของกระบวนการที่ 6 : แสดงค่าอุณหภูมิและความชื้น	30
ตารางที่ 4.1 การกำหนดพอร์ตการใช้งาน	31
ตารางที่ 4.2 ทดลองเปิดพัดลมไอน์บนบนเว็บแอปพลิเคชัน	33
ตารางที่ 4.3 ทดลองปิดพัดลมไอน์บนบนเว็บแอปพลิเคชัน	34
ตารางที่ 4.4 ทดลองกำหนดเวลาเปิดพัดลมไอน์โดยอัตโนมัติ	35
ตารางที่ 4.5 ทดลองกำหนดเวลาปิดพัดลมไอน์โดยอัตโนมัติ	36
ตารางที่ 4.6 ผลการทดลองการตรวจสอบค่าอุณหภูมิและค่าความชื้น	38
ตารางที่ 4.7 ผลการทดลองขณะเปิดปั๊มน้ำและปิดปั๊มน้ำ	39
ตารางที่ 4.8 ผลการทดลองขณะเปิดระบบสาย	40
ตารางที่ 4.9 ผลการทดลองขณะปิดระบบสาย	41
ตารางที่ 4.10 ทดลองเลือกแรงลมของพัดลมไอน์	42

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 พัดลมไอเย็น	5
รูปที่ 2.2 หลักการทำงานของพัดลมไอเย็น	5
รูปที่ 2.3 ภาพตัวอย่างการออกแบบการติดตั้ง	6
รูปที่ 2.4 ภาพอธิบาย Wireless Sensor Network	9
รูปที่ 2.5 ภาพแสดง WSN Nodes	10
รูปที่ 2.6 ภาพ Diagram อธิบายการเชื่อมต่อ Gateway หลายๆตัวเข้ากับ local network	10
รูปที่ 2.7 ภาพอธิบายแต่ละ Network Layers ของ Internet of Things โดย IBM	11
รูปที่ 2.8 บอร์ด ESP8266 NodeMCU	13
รูปที่ 2.9 DHT22	13
รูปที่ 2.10 DHT11 / DHT22 Protocol	14
รูปที่ 2.11 โปรแกรม Visual Studio Code	15
รูปที่ 2.12 Firebase	16
รูปที่ 2.13 ตัวอย่างการเก็บข้อมูลแบบ Document	18
รูปที่ 2.14 ตัวอย่างการเก็บข้อมูลแบบ Collection	18
รูปที่ 2.15 ตัวอย่างการเก็บข้อมูลแบบ Subcollection	19
รูปที่ 2.16 แสดงลักษณะการเพิ่มข้อมูล	20
รูปที่ 2.17 แสดงลักษณะการอัปเดตและลบข้อมูล	20
รูปที่ 3.1 ระบบฮาร์ดแวร์	23
รูปที่ 3.2 ระบบซอฟต์แวร์	24
รูปที่ 3.3 State Diagram	25
รูปที่ 3.4 Context Diagram	27
รูปที่ 3.5 Data Flow Diagram	28
รูปที่ 4.1 การกำหนดพอร์ตของ ESP8266 NodeMCU	32
รูปที่ 4.2 แสดงค่าอุณหภูมิและค่าความชื้น	37
รูปที่ ก.1 การเจาะฝาหลังของพัดลมไอเย็น	48

สารบัญ (ต่อ)

	หน้า
รูปที่ ก.2 การติดตั้งบอร์ด ESP8266 NodeMCU, Relay Module 4 Channel และ Switching Power Supply 5V 5A	48
รูปที่ ก.3 การติดตั้งเซ็นเซอร์ DHT22 บริเวณที่ลมผ่าน	49
รูปที่ ก.4 การติดตั้งชุดสายไฟ	49
รูปที่ ก.5 การติดตั้งชุดสายไฟที่ใช้ทริกลวงจรพัดลมไอน์	50
รูปที่ ก.6 ติดตั้งชุดระบบควบคุมเรียบร้อย	50
รูปที่ ก.7 พัดลมไอน์ที่ติดตั้งชุดระบบควบคุมแล้ว	51



บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

พลังงานในปัจจุบันมีราคาสูงไม่ได้มีราคาถูกเหมือนในอดีตที่ผ่านมา และเป็นค่าใช้จ่ายสำคัญอย่างหนึ่งของผู้บริหารสำนักงาน การอนุรักษ์พลังงานจึงเป็นสิ่งสำคัญที่ผู้บริหารสำนักงานต้องการให้เกิดขึ้น เพราะทำให้มีความคุ้มค่ามากที่สุดสำหรับค่าใช้จ่ายที่เสียไป และอาจจะช่วยลดค่าใช้จ่ายให้กับผู้บริหารสำนักงานได้ จากการสำรวจอุปกรณ์เครื่องใช้ไฟฟ้าภายในสำนักงานพบว่า พัดลมไอเย็นภายในสำนักงานนั้นมีการใช้พลังงานมากเป็นอันดับต้นๆ ดังนั้นการควบคุมการใช้งานพัดลมไอเย็นอย่างเต็มประสิทธิภาพจะช่วยอนุรักษ์พลังงานได้ แต่เนื่องจากสำนักงานในปัจจุบันยังไม่มีระบบควบคุมการใช้งานพัดลมไอเย็นเข้ามาช่วย ทำให้ไม่สามารถควบคุมการใช้งานพัดลมไอเย็นได้อย่างมีประสิทธิภาพ ซึ่งอาจเกิดปัญหาการแผ่รังสีการใช้พลังงานภายในสำนักงานคือ พนักงานลืมปิดพัดลมไอเย็น ส่งผลกระทบทำให้สูญเสียพลังงานอย่างไร้คุณค่า และค่าใช้จ่ายของผู้บริหารสำนักงานเพิ่มมากขึ้นอีกด้วย แต่ถ้ามีระบบควบคุมการใช้งานพัดลมไอเย็นจะสามารถควบคุมพัดลมไอเย็นจากระยะไกลได้ ลดขั้นตอนการใช้งานพัดลมไอเย็นของพนักงานลง และสามารถกำหนดเวลาเปิดหรือปิดพัดลมไอเย็นก่อนเวลาเริ่มงาน และเลิกงานได้ ทำให้มีความสะดวกสบายในการทำงานของพนักงาน แก้ไขปัญหาการลืมปิดพัดลมไอเย็นของพนักงาน และเกิดการอนุรักษ์พลังงานตามที่ผู้บริหารสำนักงานต้องการ

Internet of Thing (IoT) เป็นแนวคิดการนำอินเทอร์เน็ตไปเชื่อมต่อกับอุปกรณ์ ให้อุปกรณ์นั้นสามารถรับ-ส่งข้อมูล เพื่อให้เราสามารถควบคุมหรือนำข้อมูลจากอุปกรณ์นั้นมาใช้งานได้ จึงเป็นแนวคิดที่ดีที่จะนำมาพัฒนาใช้งานกับพัดลมไอเย็น

ดังนั้นทางคณะผู้จัดทำโครงการจึงมีแนวคิดที่จะสร้างระบบอนุรักษ์พลังงานจากพัดลมไอเย็นให้สามารถควบคุมการใช้งานพัดลมไอเย็นให้ได้อย่างเต็มประสิทธิภาพ โดยสามารถสั่งเปิดหรือปิด

และกำหนดเวลาเปิดหรือปิดพัดลมไอน้ำล่วงหน้าจากระยะไกลได้ โดยใช้งานผ่านเว็บแอปพลิเคชัน (Web Application) ทางอินเทอร์เน็ต (Internet)

1.2 คำสำคัญ (Keyword)

อินเทอร์เน็ตของสรรพสิ่ง / โหนดเอ็มซียู / จาวาสคริปต์ / ไฟร์เบส / พัดลมไอน้ำ

1.3 วัตถุประสงค์ในการทำโครงการ

1.3.1 เพื่อสร้างเครื่องควบคุมพัดลมไอน้ำให้เกิดการอนุรักษ์พลังงาน

1.3.2 เพื่อสร้างซอฟต์แวร์ใช้งานเครื่องควบคุมพัดลมไอน้ำ

1.4 ขอบเขตความสามารถของโครงการ

1.4.1 ควบคุมพัดลมไอน้ำได้ 1 เครื่อง

1.4.2 ระบบต้องมีการเชื่อมต่อ Internet

1.4.3 ใช้งานระบบควบคุมบน Web Application

1.4.4 Web Application มีฟังก์ชันการทำงานทั้งหมด 6 ฟังก์ชันดังนี้

1.4.4.1 แสดงค่าอุณหภูมิและค่าความชื้น

1.4.4.2 ใช้สั่งเปิดหรือปิดพัดลมไอน้ำ

1.4.4.3 ใช้สั่งเปิดหรือปิดระบบสายของพัดลมไอน้ำ

1.4.4.4 ใช้สั่งเปิดหรือปิดปั๊มน้ำของพัดลมไอน้ำ

1.4.4.5 ใช้กำหนดเวลาเปิดหรือปิดพัดลมไอน้ำ

1.4.4.6 ใช้เลือกแรงลมของพัดลมไอน้ำระดับ Low, Medium และ High

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 สามารถควบคุมการใช้งานพัดลมไอเย็นได้อย่างมีประสิทธิภาพ

1.5.2 มีความสะดวกสบายมากขึ้นในการควบคุมพัดลมไอเย็น

1.6 เครื่องมือ, เครื่องวัด และอุปกรณ์ที่ใช้

1.6.1 ซอฟต์แวร์

— Visual Studio Code	จำนวน 1 โปรแกรม
— Arduino IDE	จำนวน 1 โปรแกรม
— Xampp	จำนวน 1 โปรแกรม
— Firebase platform	จำนวน 1 โปรแกรม

1.6.2 ฮาร์ดแวร์

— NodeMCU Development Kit	จำนวน 1 ตัว
— DHT22 Digital Temperature and Humidity Sensor	จำนวน 1 ตัว
— Relay Module 4 Channel (5V)	จำนวน 2 ตัว
— Switching Power Supply 5V 5A	จำนวน 1 ตัว
— พัดลมไอเย็น	จำนวน 1 เครื่อง
— Computer Notebook	จำนวน 1 เครื่อง

1.7 แผนการดำเนินงาน

ตารางที่ 1.1 การวางแผนการดำเนินงาน

ลำดับที่	หัวข้องาน	%	2561						2562									
			ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.
1	ศึกษาข้อมูลเกี่ยวกับโครงงาน	5	■															
		10	■	■														
2	นำเสนองานและรูปแบบโครงงาน	5	■															
		5	■	■														
3	ศึกษาโครงสร้างและหลักการการทำงานของบอร์ด ESP8266	10			■	■												
		5			■													
4	ศึกษาโครงสร้างและหลักการทำงานของ Firebase	10					■											
		5				■												
5	ออกแบบการทำงาน	5						■										
		5						■										
6	เขียนคำสั่งการทำงานของบอร์ด ESP8266 บน Arduino IDE	20																
		35																
7	เขียนเว็บแอปพลิเคชันบน Visual Studio Code	20																
		35																
8	ทดสอบการทำงานของระบบ	5																
		10																
9	ปรับปรุงโครงงานและบันทึกผล	5																
		5																
10	จัดทำรายงานนิพนธ์	15																
		20																

ตามแผนงาน ตามงานจริง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะเป็นเนื้อหาที่ได้ทำการศึกษาค้นคว้าเพื่อนำมาใช้ในการพัฒนาซอฟต์แวร์และฮาร์ดแวร์โดยมีรายละเอียดดังนี้

2.1 Evaporative Fan (พัดลมไอเย็น)

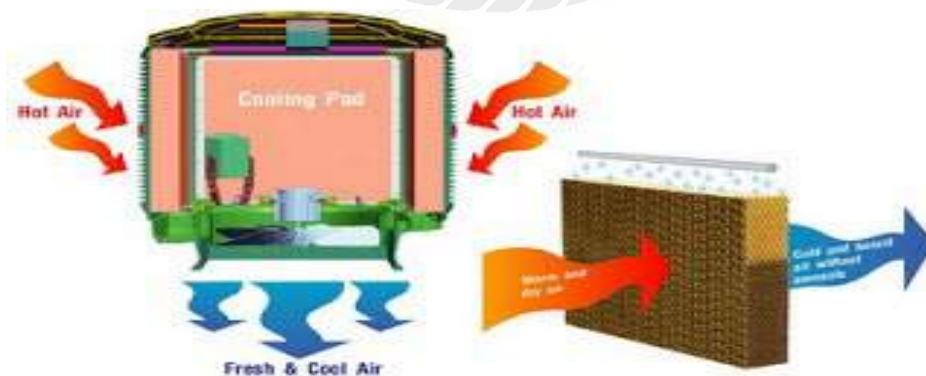


รูปที่ 2.1 พัดลมไอเย็น

เครื่องลดอุณหภูมิอากาศ ชนิดอีแวโพเรทีฟ (Evaporative Air Cooler) หรือ เรียกว่า พัดลมไอเย็น (Evaporative Fan) เย็นน้ำ ประหยัดไฟปลอดภัยต่อสุขภาพ แบ่งเป็น 2 แบบ คือ

- แบบเคลื่อนที่ (Moving Type)
- แบบต่อท่อลม (Duct Type)

Evaporative Fan (พัดลมไอเย็น)



คือ อาศัยการออกแบบโดยหลักการพื้นฐานของธรรมชาติ ใช้การระเหยของน้ำแลกเปลี่ยนความร้อนแฝงของอากาศ โดยการไหลของน้ำผ่าน Cooling Pad ก็จะถูกเพิ่มพื้นที่ผิว

และลดแรงตึงผิวระหว่างน้ำกับอากาศ ทำให้น้ำระเหยทันที โดยดึงเอาความร้อนจากอากาศรอบ ๆ Cooling Pad มาใช้ในการระเหย เมื่ออากาศถูกดึงความร้อนก็จะทำให้ อุณหภูมิลดลง

จะมาก-น้อยทั้งนี้ ขึ้นอยู่กับความชื้นสัมพัทธ์ในอากาศ อุณหภูมิอากาศในขณะนั้น และ ความเร็วลมที่ไหลผ่านผิวหน้าของ Cooling Pad

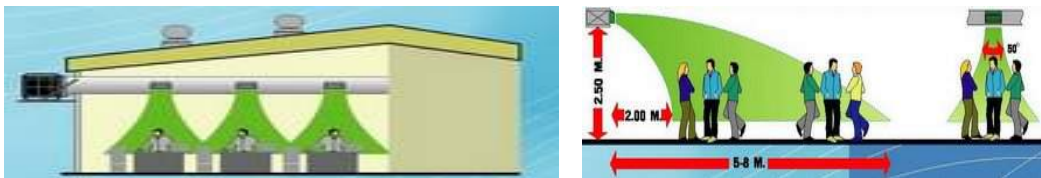
หลักการทำงาน

ใช้ระบบ Evaporative Cooling เมื่ออากาศผ่าน Cooling Pad (แผงทำความเย็น) ที่ถูกเลี้ยงด้วยน้ำ ในลักษณะเป็นม่านน้ำ ที่ทำหน้าที่ดูดซับความร้อนจากอากาศภายนอก เพื่อไปใช้ในการเปลี่ยนสถานะจากน้ำให้กลายเป็นไอ เรียกว่า " การระเหย " (Evaporation) เป็นผลทำให้ลมที่ผ่าน Cooling Pad (แผงทำความเย็น) มีอุณหภูมิลดลง ลมที่เป่าออกมาจึงเป็นลมเย็น ที่ปราศจากละอองน้ำ ทำให้บริเวณนั้น มีอุณหภูมิลดลง จากอุณหภูมิปกติ เฉลี่ย 5-10 องศาเซลเซียส ในขณะที่ระบบ " Anion " ที่ช่วยเพิ่มคุณภาพของอากาศ จึงส่งผลให้บุคคลที่อยู่ในบริเวณดังกล่าว รู้สึกเย็นสบาย และสดชื่นขึ้น

ประโยชน์ของการใช้งาน

เป็นเครื่องให้ลมเย็นเอนกประสงค์ เหมาะกับพื้นที่เปิด มีอากาศถ่ายเทสะดวก ช่วยลดอุณหภูมิลงในบริเวณที่ลมกระจายไปถึง ซึ่งเป็นลมที่มีความชุ่มชื้นพอเหมาะ มีหลายรุ่นหลายขนาดให้เลือกใช้ตามความต้องการ สามารถใช้ทั้งภายใน และภายนอกอาคาร ไม่ว่าจะเป็นห้องรับแขก, ห้องนั่งเล่น, ห้องนอน, สนามหน้าบ้าน, ร้านอาหาร, สวนอาหาร, งาน event, คลังสินค้า, พื้นที่ผลิตในโรงงานอุตสาหกรรม ฯลฯ

คุณสมบัติเด่น



รูปที่ 2.3 ภาพตัวอย่างการออกแบบการติดตั้ง

- การออกแบบ : สามารถออกแบบกำหนดจุดและทิศทาง ในการส่งลมเย็น
- ระยะแรงลม : ระยะส่งลมไกล 5-8 เมตร

- ไม่มีละอองน้ำ : เป็นเครื่องให้ลมเย็นที่ปราศจากละอองน้ำ จึงไม่เปียกชื้นทั้งตัวผู้ใช้และบริเวณนั้นๆ
- ประหยัดไฟฟ้า : ใช้ปริมาณไฟฟ้าน้อยมากในการทำงาน เมื่อเทียบกับเครื่องปรับอากาศชนิดอื่นๆ
- มีระบบฟอกอากาศ : มีระบบ ION ที่ทำหน้าที่ยิงประจุลบ เพื่อยับยั้งเชื้อรา, แบคทีเรีย และฝุ่นละอองในอากาศ
- ให้ความชุ่มชื้นพอเหมาะ : ใช้การระเหยของน้ำ โดยขึ้นอยู่กับความชื้นในอากาศ หากในพื้นที่ที่มีความชื้นมาก อัตราการระเหยจะลดลงโดยอัตโนมัติ และควรใช้ในพื้นที่ที่มีอากาศถ่ายเทสะดวก จึงไม่ก่อให้เกิดผลข้างเคียง

Cooling Pad คือแผงทำความเย็น ผลิตจากเยื่อไม้อย่างดีสำหรับ ระบบ Evaporative ซึ่งเป็นเทคโนโลยีจากสวีเดน ที่มีคุณภาพเป็นอันดับต้นๆของโลก มีความแข็งแรงทนทาน อายุการใช้งาน 5 ปี เคลือบด้วยสารป้องกันเชื้อราและแบคทีเรีย โดยหลักการนำน้ำผ่านออกมาให้สามารถสัมผัสอากาศได้มาก จึงมีอัตราการลดอุณหภูมิได้เป็นอย่างดี เฉลี่ย 5-10 องศาเซลเซียส มีขนาดที่ความหนา 10, 15, 20 cm. ความกว้าง 0.3-0.6 m.และความสูง 1.5, 1.8, 2.0 m.

ตารางที่ 2.1 ตารางแสดงอุณหภูมิและความสามารถในการทำความเย็น

ตารางแสดงอุณหภูมิและความสามารถในการทำความเย็น								
Family Air Engineering co.,ltd								
Outdoor Temperature %RH	40°C	38°C	36°C	34°C	32°C	30°C	28°C	
30%	26	25	23.8	21.8	20.5	19	17.5	
35%	27.5	26.3	24.5	23	21.5	20	18.5	
40%	29	27.5	26	24.2	22.8	21.2	19.5	
45%	30.5	28.8	27.2	25.5	23.8	22.2	20.5	
50%	32	30	28.5	26.8	25	23	21.2	
55%	33	31	29.2	28	25.8	24	22	
60%	34	32.5	30.5	28.6	26.8	25	23	
65%	35	33.5	31.5	29.5	28	26	24	
70%	36	34.5	32.5	30.5	28.8	27	25	
75%	37	36	33.2	31.5	29.5	27.8	26	
80%	36	37	34.2	32.2	30.5	28.5	26.5	

Remark: Wind-chill Integrote not include in the table.

ตารางที่ 2.2 ตารางเปรียบเทียบความแตกต่างของอุณหภูมิ

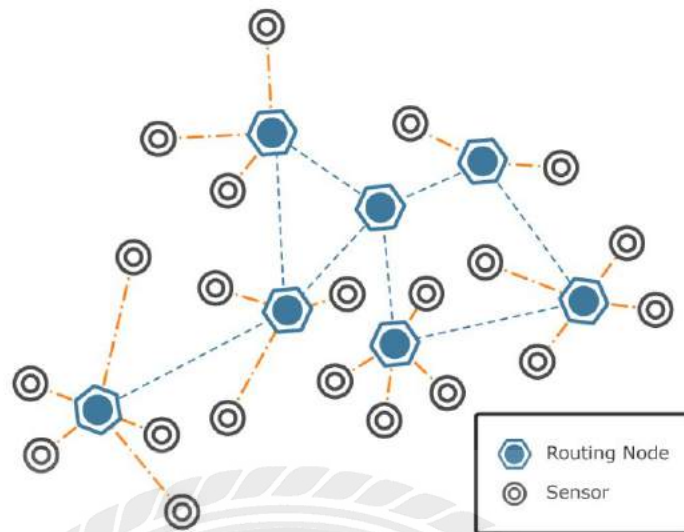
ตารางเปรียบเทียบความแตกต่างของอุณหภูมิเมื่อใช้ M Kool

สภาพอากาศ	แดดจัด	เมฆมาก	ฝนตก
อุณหภูมิภายนอกอาคาร (°C)	33	30	28
อุณหภูมิภายในอาคารที่ใช้ M Kool (°C)	26	26	26
ความแตกต่างของอุณหภูมิ (°C)	7	4	2
ความชื้นภายนอก (%)	60	72	80

2.2 IOT (Internet of Things)

แนวคิด Internet of Things ถูกคิดขึ้นโดย Kevin Ashton ในปี 1999 ซึ่งเขาเริ่มต้นโครงการ Auto-ID Center ที่มหาวิทยาลัย Massachusetts Institute of Technology หรือ MIT จากเทคโนโลยี RFID ที่จะทำให้เป็นมาตรฐานระดับโลกสำหรับ RFID Sensors ต่าง ๆ ที่จะเชื่อมต่อกันได้ ต่อมาในยุคหลังปี 2000 โลกมีอุปกรณ์อิเล็กทรอนิกส์ออกมาเป็นจำนวนมากและมีการใช้คำว่า Smart คือ smart device, smart grid, smart home, smart network, smart intelligent transportation เหล่านี้ล้วนมีโครงสร้างพื้นฐานที่สามารถเชื่อมต่อกับโลกอินเทอร์เน็ตได้ ซึ่งการเชื่อมต่อเหล่านั้นเองก็เลยมาเป็นแนวคิดที่ว่าอุปกรณ์เหล่านั้นก็ย่อมสามารถสื่อสารกันได้ด้วยเช่นกันโดยอาศัยตัว Sensor ในการสื่อสารถึงกัน นั้นแปลว่านอกจาก Smart devices ต่าง ๆ จะเชื่อมต่ออินเทอร์เน็ตได้แล้วมันยังสามารถเชื่อมต่อไปยังอุปกรณ์ตัวอื่นได้ด้วยโดย Kevin นิยามมันไว้ตอนนั้นว่าเป็น “internet-like” หรืออุปกรณ์อิเล็กทรอนิกส์สามารถสื่อสารพูดคุยกันเองได้ซึ่งอุปกรณ์อิเล็กทรอนิกส์แทนศัพท์คำว่า “Things”

A wireless sensor network (WSN) ตัวแปลสำคัญสำหรับ Internet of Things ที่ใช้ในการสื่อสารนั้นไม่เพียงแต่ Internet network เพียงเท่านั้นแต่ยังมีตัวแปลอื่นเข้ามาเกี่ยวข้องอีกนั่นคือ Sensor node ต่าง ๆ จำนวนมากที่ทำให้เกิด wireless sensor network (WSN) ให้กับอุปกรณ์ต่าง ๆ สามารถเชื่อมต่อเข้ามาได้ ซึ่ง WSNs สามารถตรวจจับปรากฏการณ์ต่าง ๆ (physical phenomena) ในเครือข่ายได้ยกตัวอย่าง เช่น แสง อุณหภูมิ ความดัน เป็นต้น เพื่อส่งค่าไปยังอุปกรณ์ในระบบให้ทำงานหรือสั่งงานอื่น ๆ ต่อไป



รูปที่ 2.4 ภาพอธิบาย Wireless Sensor Network

Access Technology

การพัฒนา Internet of Things นั้นนอกจากจะพัฒนาเทคโนโลยีในฝั่ง Hardware ได้แก่ processors, radios และ sensors ซึ่งจะถูกรวมเข้าด้วยกันเรียกว่า a single chip หรือ system on a chip (SoC) แล้วก็ยังพัฒนา WSN ไปพร้อม ๆ กันและการเชื่อมต่อปัจจุบันมีการพัฒนาเทคโนโลยีสำหรับการเชื่อมต่อสำหรับ Internet of Things หรือ Access technology

มี 3 ตัว ได้แก่

- Bluetooth 4.0
- IEEE 802.15.4e
- WLAN IEEE 802.11™ (Wi-Fi)

โดยในแต่ละ Access technologies นั้นมีการส่งข้อมูลที่แตกต่างกันดังนี้

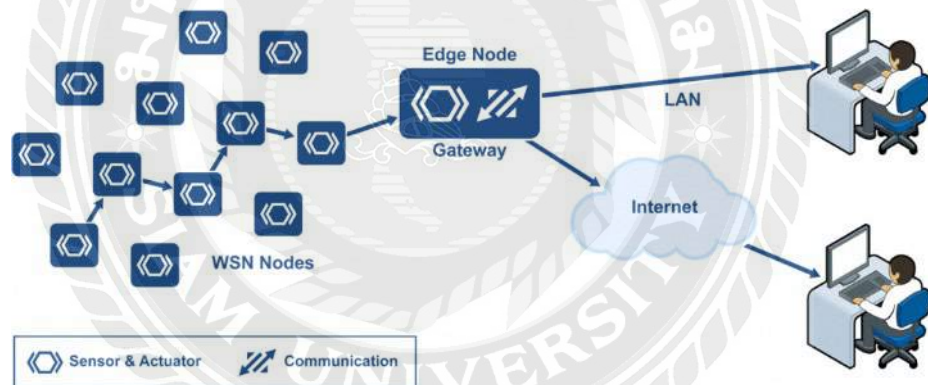
ตารางที่ 2.3 การส่งข้อมูลของแต่ละ Access technologies

	IEEE 802.15.4e	Bluetooth	WLAN IEEE 802.11
Frequency	868/915 MHz 2.4 GHz	2.4 GHz	2.4, 5.8 Ghz

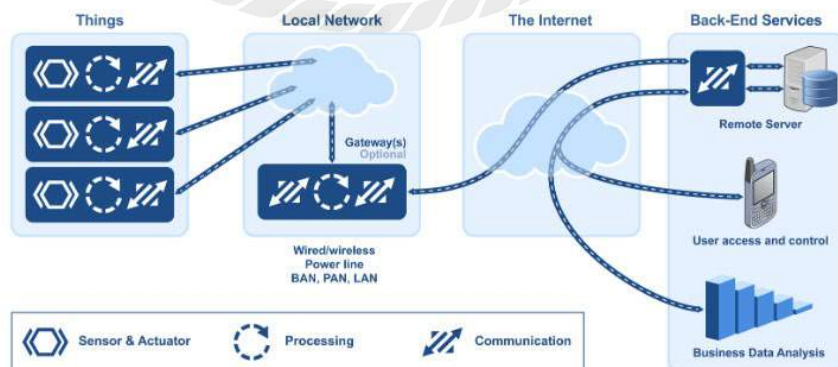
Data rate	250 Kbps	723 Kbps	11 – 105 Mbps
Power	Very low	Low	High

Gateway Sensor Nodes

เมื่อมีโครงข่าย Sensor nodes แล้วก็จำเป็นจะต้องมี Gateway Sensor Nodes เพื่อจะเชื่อมต่อไปยังโลกอินเทอร์เน็ตด้วย โดยตัว Gateway นี้จะทำหน้าที่เชื่อมต่อไปยังเครือข่าย Internet ให้อุปกรณ์ทั้งหมดในโครงข่าย Sensor nodes ทั้งหมดส่งข้อมูลเข้าสู่อินเทอร์เน็ตได้ และ Gateway นี้ก็จะอยู่ภายใต้ Local network ซึ่งจะมีการกำหนดกันต่อไปว่า Gateway ภายใต Local network นั้นจะให้เชื่อมต่อไปยัง Internet ได้หรือไม่ถ้าไม่ได้อุปกรณ์ที่เชื่อมเข้ามาใน Gateway ก็อาจจะสื่อสารกันได้เฉพาะภายใน Local network ได้เท่านั้น



รูปที่ 2.5 ภาพแสดง WSN Nodes

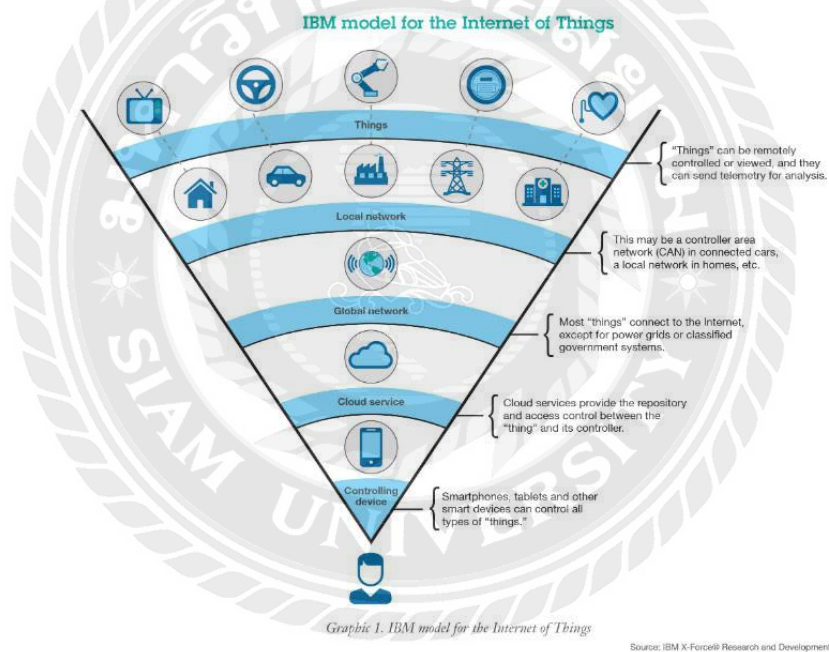


รูปที่ 2.6 ภาพ Diagram อธิบายการเชื่อมต่อ Gateway หลายๆตัวเข้ากับ local network

แบ่งกลุ่ม Internet of Things

ปัจจุบันมีการแบ่งกลุ่ม Internet of Things ออกตามตลาดการใช้งานเป็น 2 กลุ่มได้แก่

- Commercial IoT คือ แบ่งจาก local network ที่มีหลายเทคโนโลยีที่แตกต่างกัน ในโครงข่าย Sensor nodes โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะเชื่อมต่อแบบ IP network เพื่อเข้าสู่อินเทอร์เน็ต
- Industrial IoT คือ แบ่งจาก local communication ที่เป็น Bluetooth หรือ Ethernet (wired or wireless) โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะสื่อสารภายในกลุ่ม Sensor nodes เดียวกันเท่านั้นหรือเป็นแบบ local devices เพียงอย่างเดียวอาจไม่ได้เชื่อมสู่อินเทอร์เน็ต



รูปที่ 2.7 ภาพอธิบายแต่ละ Network Layers ของ Internet of Things โดย IBM

IPv6 คือส่วนสำคัญของ Internet of Things ตัวอุปกรณ์ IoT devices ต่าง ๆ นั้น จำเป็นต้องมีหมายเลขระบุเพื่อใช้ในการสื่อสารเปรียบเสมือนที่อยู่ของบ้าน และการที่จะทำให้ อุปกรณ์เหล่านั้นที่มีอยู่เป็นจำนวนมาก(รวมถึงอนาคตที่จะผลิตกันออกมา) จำเป็นจะต้องใช้ IP Address version 6 หรือ IPv6 มากำกับเพื่อให้ได้หมายเลขที่ไม่ซ้ำกันและต้องใช้ได้ทั้ง IoT network ที่เป็น LAN, PAN, และ BAN: Body Area Network หรือการสื่อสารของตัว Sensor กับ ร่างกายมนุษย์

Internet network (protocols) ที่เป็น IP, UDP, TCP, SSL, HTTP, HTTPS, และอื่น ๆ โดยทั้งหมดคือส่วนสำคัญของ Internet of Things ที่กำลังเกิดขึ้นและเป็นเทรนด์ที่กำลังมาแรงอยู่ในขณะนี้ หวังว่าบทความนี้จะช่วยให้หลายคนเข้าใจภาพของ IoT ได้ดีขึ้น สิ่งสำคัญคือศัพท์คำนี้จึงไม่ได้หมายถึง Smart device เช่น นาฬิกาอัจฉริยะและ Apple Watch หรือสายรัดข้อมือเพื่อสุขภาพเท่านั้น แต่มันยังครอบคลุมไปถึงอุปกรณ์ต่าง ๆ อีกหลากหลายด้านตัวกว้างไกลไปยังหลากหลายอุตสาหกรรมอีกด้วย โดยในอนาคตคุณจะได้เห็น ไม่ใคร่เวลาสื่อสารกับตู้เย็นให้สั่งอาหารมาเติม เครื่องซักผ้าสื่อสารกับทีวีบอกคุณว่าผ้าซักเสร็จแล้ว สายรัดข้อมือจะสื่อสารกับรถพยาบาลแจ้งให้ไปรับตัวผู้ป่วยที่กำลังหัวใจวาย เหล่านี้คืออนาคตของ Internet of Things ที่สิ่งต่าง ๆ กำลังจะสื่อสารกันได้

2.3 Arduino

Arduino เป็นภาษาิตาเลียว่า อาดูอีโน้ หรือ อาดูยโน้ Arduino คือ Open-Source Platform สำหรับการสร้างต้นแบบทางอิเล็กทรอนิกส์ โดยมีจุดมุ่งหมายให้ Arduino Platform เป็น Platform ที่ง่ายต่อการใช้งาน โดย Arduino Platform มี 2 ประเภท

- ส่วนที่เป็นฮาร์ดแวร์ (Hardware)

บอร์ดอิเล็กทรอนิกส์ขนาดเล็ก ที่มีไมโครคอนโทรลเลอร์ (MCU) เป็นชิ้นส่วนหลัก ถูกนำมาประกอบร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ เพื่อให้ง่ายต่อการใช้งาน หรือที่เรียกกันว่าบอร์ด Arduino, โดยบอร์ด Arduino เองก็มีหลายรุ่นให้เลือกใช้ โดยมีความแตกต่างกันในเรื่องของขนาด หรือสเปค เช่น จำนวนของขารับส่งสัญญาณ, แรงดันไฟที่ใช้, และประสิทธิภาพของ MCU เป็นต้น

- ส่วนที่เป็นซอฟต์แวร์ (Software)

ซอฟต์แวร์ที่ใช้ในการพัฒนางาน สำหรับบอร์ด Arduino นั่นคือโปรแกรมที่เรียกว่า Arduino IDE ในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด โดย ขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชันอื่น ๆ ของ Arduino

2.4 ESP8266 NodeMCU



รูปที่ 2.6 บอร์ด ESP8266 NodeMCU

NodeMCU (โนนด เอ็มซียู) คือ บอร์ดคล้าย Arduino ที่สามารถเชื่อมต่อกับ WiFi ได้ สามารถเขียนโปรแกรมด้วย Arduino IDE ได้เช่นเดียวกับ Arduino และบอร์ดก็มีราคาถูกมาก ๆ เหมาะแก่ผู้ที่คิดจะเริ่มต้นศึกษา หรือทดลองใช้งานเกี่ยวกับ Arduino, IoT, อิเล็กทรอนิกส์ หรือแม้แต่การนำไปใช้จริงในโปรเจกต์ต่าง ๆ ก็ตาม เพราะราคาไม่แพง ภายในบอร์ดของ NodeMCU ประกอบไปด้วย ESP8266 (ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อ WiFi ได้) พร้อมอุปกรณ์อำนวยความสะดวก เช่น พอร์ต micro USB สำหรับจ่ายไฟและอัปโหลดโปรแกรม, ชิพสำหรับอัปโหลดโปรแกรมผ่านสาย USB, ชิพแปลงแรงดันไฟฟ้า และขาสำหรับเชื่อมต่ออุปกรณ์ภายนอก เป็นต้น

2.5 DHT22

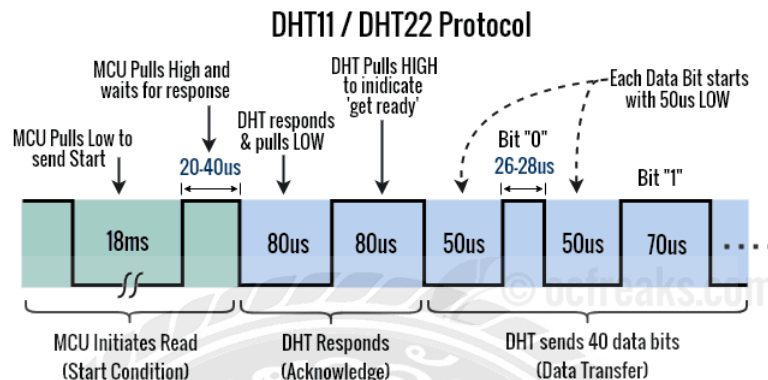
DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



รูปที่ 2.7 DHT22

DHT22 คือเซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์(Temperature & Relative Humidity Sensor) เป็นอุปกรณ์ที่สามารถนำมาประยุกต์ใช้งานทางด้านระบบสมองกลฝังตัวได้หลากหลาย เช่น การวัดและควบคุมอุณหภูมิและความชื้น ระบบบันทึกข้อมูลเกี่ยวกับอุณหภูมิและความชื้นในห้อง เป็นต้น อุปกรณ์ประเภทนี้แตกต่างกันตามผู้ผลิต ราคา ความแม่นยำ ความละเอียดในการวัด

ในการติดต่อสื่อสารกับ DHT22 จะใช้สายสัญญาณดิจิทัลเพียงเส้นเดียว (onewire bus) ในการเชื่อมต่อแบบบิตอนุกรมสองทิศทาง (serial data, bi-directional) โดยนำมาเชื่อมต่อกับขา Arduino เพื่ออ่านค่าจากเซนเซอร์



รูปที่ 2.10 DHT11 / DHT22 Protocol

คุณสมบัติของ DHT11

- ใช้แรงดันไฟฟ้า 3 ถึง 5V
- ใช้กระแสไฟฟ้าสูงสุด 2.5mA (ขณะทำการวัดค่า)
- เหมาะสำหรับวัดความชื้นระดับ 20-80% โดยมีความผิดพลาดในการวัดไม่เกิน 5%
- เหมาะสำหรับวัดอุณหภูมิ 0-50°C โดยมีความผิดพลาดในการวัดไม่เกิน $\pm 2^{\circ}\text{C}$
- ความถี่ในการวัด 1 Hz (อ่านค่าได้วินาทีละครั้ง)
- ขนาด 15.5mm x 12mm x 5.5mm
- 4 pins ใช้พื้นที่ในการวางขา 0.1"

คุณสมบัติของ DHT22

- ใช้แรงดันไฟเลี้ยงได้ในช่วง: 3.3V ถึง 5.5V DC (ดังนั้นจึงใช้ได้กับ 3.3V และ 5V)
- วัดอุณหภูมิได้ในช่วง: -40 to 80 °C ($\pm 0.5^{\circ}\text{C}$ accuracy)
- วัดความชื้นสัมพัทธ์ได้ในช่วง: 0 – 100 RH% (2 – 5% accuracy)
- อัตราการวัดสูงสุด: 0.5Hz
- คอนเนกเตอร์แบบ 4 ขา (0.1" / 2.54mm spacing)
- Pin 1 = VCC
- Pin 2 = SDA (Serial data, bidirectional)
- Pin 3 = N.C. (Not Connect)
- Pin 4 = GND

2.6 Xampp

Xampp คือโปรแกรมสำหรับจำลองเครื่องคอมพิวเตอร์ส่วนบุคคลของเรา ให้ทำงานในลักษณะของ WebServer นั่นคือเครื่องคอมพิวเตอร์ของเราจะเป็นทั้งเครื่องแม่ และเครื่องลูกในเครื่องเดียวกัน ทำให้ไม่ต้องเชื่อมต่อกับ Internet คุณก็สามารถทดสอบเว็บไซต์ที่คุณสร้างขึ้น ได้ทุกที่ทุกเวลา ปัจจุบันได้รับความนิยมจากผู้ใช้ CMS ในการสร้างเว็บไซต์

XAMPP ประกอบด้วย Apache, PHP, MySQL, PHP MyAdmin, Perl ซึ่งเป็นโปรแกรมพื้นฐานที่รองรับการทำงาน CMS ซึ่งเป็นชุดโปรแกรม สำหรับออกแบบเว็บไซต์ที่ได้รับความนิยมในปัจจุบัน ไฟล์สำหรับติดตั้ง xampp นั้นอาจมีขนาดใหญ่สักหน่อย เนื่องจาก มีชุดควบคุมการทำงานที่ช่วยให้การปรับแต่งส่วนต่าง ๆ ง่ายขึ้น XAMPP นั้นรองรับระบบปฏิบัติการหลายตัว เช่น Windows, Linux, Apple ทำงานได้ทั้งบนระบบปฏิบัติการแบบ 32 bit และ 64 bit สิ่งที่น่าสนใจกว่าโปรแกรมอื่นคือมีตัวช่วยติดตั้ง CMS ที่เรียกว่า BitNami ซึ่งช่วยให้คุณติดตั้ง CMS รุ่นใหม่ ๆ ที่ได้รับความนิยมอีกด้วย เยี่ยมชมเว็บไซต์อย่างเป็นทางการของ XAMPP

นอกจาก Xampp แล้วยังมีโปรแกรมในลักษณะนี้อีก เช่น Appserv, Wamp เป็นต้น สิ่งที่ต้องพิจารณาในการเลือกใช้งานคือเวอร์ชันของ Apache, PHP และ MySQL เนื่องจาก CMS แต่ละตัวนั้นมีความต้องการเวอร์ชันไม่เท่ากัน ก่อนใช้งานจึงต้องพิจารณาให้ดี ทั้งนี้เพื่อไม่ให้เกิดปัญหา หรือเกิดปัญหาในการใช้งานน้อยที่สุดนั่นเอง

2.7 Visual Studio Code



รูปที่ 2.11 โปรแกรม Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไข และปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ

ซึ่ง Visual Studio Code เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ดังนี้

- การเปิดใช้งานภาษาอื่น ๆ เช่น ภาษา C++, C#, Java, Python, PHP หรือ Go
- Themes
- Debugger
- Commands

ความแตกต่างระหว่าง VSCode และ Visual Studio คือ

VSCode ได้ทำการตัดในส่วนของ GUI designer ออกไป เหลือแต่เพียงตัว Editor เท่านั้น จึงทำให้ตัวโปรแกรมนั้นค่อนข้างเบากว่า Visual Studio เป็นอย่างมาก VSCode สามารถนำมาใช้งานได้ฟรี รองรับการทำงานข้ามแพลตฟอร์ม

2.8 Firebase



รูปที่ 2.12 Firebase

Firebase คือ Project ที่ถูกออกแบบมาให้เป็น API และ Cloud Storage สำหรับพัฒนา Realtime Application รองรับหลาย Platform ทั้ง iOS App, Android App, Web App บริการที่เลือกใช้

Database แบบ Realtime

ฐานข้อมูล Firebase Realtime เป็นฐานข้อมูลแบบ Cloud-hosted ข้อมูลจะถูกเก็บเป็น JSON และซิงโครไนส์ในแบบเรียลไทม์กับทุกไคลเอนต์ที่เชื่อมต่อ เมื่อผู้ใช้สร้างแอปพลิเคชันข้าม

แพลตฟอร์มด้วย SDK ของ iOS, Android และ JavaScript เก็บและซิงค์ข้อมูลกับฐานข้อมูล NoSQL cloud ของเรา ข้อมูลจะซิงค์กับผู้ใช้ทั้งหมดในแบบเรียลไทม์และยังสามารถใช้งานได้เมื่อแอปของผู้ใช้ออฟไลน์

- **Realtime** ฐานข้อมูล Firebase Realtime จะใช้การซิงค์ข้อมูลทุกครั้งที่มีการเปลี่ยนแปลงข้อมูลอุปกรณ์ใด ๆ ที่เชื่อมต่อ
- **Offline** Firebase ยังสามารถตอบสนองได้แม้ในขณะออฟไลน์เนื่องจากฐานข้อมูล Firebase Realtime Database SDK ยังคงมีข้อมูลของผู้ใช้ไปยังดีสก์
- **เข้าถึงได้จากอุปกรณ์ไคลเอนต์** ฐานข้อมูล Firebase Realtime สามารถเข้าถึงได้โดยตรงจากโทรศัพท์มือถือหรือเว็บเบราว์เซอร์ไม่จำเป็นต้องมีเซิร์ฟเวอร์แอพลิเคชัน ความปลอดภัยและการตรวจสอบความถูกต้องของข้อมูลสามารถได้จากกฎความปลอดภัยฐานข้อมูล
- **ปรับขนาดข้ามฐานข้อมูลหลายแห่ง** สามารถรองรับข้อมูลของแอปพลิเคชันของผู้ใช้ได้ตามความต้องการ โดยแยกข้อมูลของผู้ใช้ออกจากอินสแตนซ์ฐานข้อมูลหลายแห่งในโครงการ Firebase เดียวกัน

การเพิ่มข้อมูลฐานข้อมูล

Database

- พื้นฐานของการเก็บข้อมูล

ระบบฐานข้อมูลของ Cloud Firestore จะเป็น NoSQL แบบ Document ซึ่งจะไม่เหมือนระบบฐานแบบ SQL โดยจะไม่มีตาราง ไม่มีแถว ใด ๆ ทั้งสิ้น การเก็บข้อมูลภายใน Document จะเก็บแบบ Key-value โดยแต่ละ Document จะถูกเก็บไว้ใน Collection ซึ่งใน Document สามารถมี subcollection ได้ด้วย

หมายเหตุ : ให้นึกภาพว่ามีเอกสาร 1 แผ่น ข้างในนั้นก็จะมีย่อหาอยู่ภายในและเอกสารหลาย ๆ แผ่น ก็จะเก็บอยู่ในแฟ้มอีกทีหนึ่ง

โดยที่ไม่ต้องไปออกแบบตารางของฐานข้อมูลไว้ก่อนเหมือนอย่างระบบฐานแบบ SQL แต่สามารถสั่งเขียนข้อมูลลงไปได้เลย โดยระบุ Collection และ Document ถ้าหากไม่เคยมี Collection และ Document มาก่อนทาง Cloud Firestore จะสร้างให้เองโดยอัตโนมัติ ซึ่งทาง Cloud Firestore ก็ได้บอกไว้ว่า เหมาะสำหรับการเก็บข้อมูลโดยมี Collection จำนวนขนาดใหญ่ และ Document ขนาดเล็ก

ตัวอย่างของการเก็บข้อมูลแบบต่าง ๆ

- Document

```
User1
Username : "Thanongkiat"
Class : "highschool16"
born : 2049
```

รูปที่ 2.11 ตัวอย่างการเก็บข้อมูลแบบ Document

Document เป็นการเรียนชื่อแทนหน่วยการเก็บของข้อมูลใน Cloud Firestore ภายในจะประกอบไปด้วย ชื่อของ Document , ชื่อของ key และ ค่าข้อมูล (value) โดยชื่อของ Document ห้ามซ้ำกัน ซึ่งใน Cloud Firestore สามารถให้เราระบุ Type ของข้อมูลได้คือ boolean, number, string, geo point, timestamp, array, object, reference และ null

เช่นดังรูป

ชื่อของ Document คือ User1

ชื่อของ key คือ Username , Class , born

ค่าข้อมูล คือ "Thanongkiat" (string) , "highschool6" (string) , 2049 (number)

- Collection

```
users
  user1
    Username : "Thanongkiat"
    Class : "highschool6"
    born : 2049
  user2
    Username : "Somchai"
    Class : "highschool14"
    born : 1999
```

รูปที่ 2.12 ตัวอย่างการเก็บข้อมูลแบบ Collection

Collection เป็นการเรียกชื่อแทนของการเก็บหลายๆเอกสารไว้ด้วยกัน เช่น เราจะเก็บข้อมูลของ user หลายๆ Document ไว้ด้วยกัน จึงตั้งชื่อ Collection ว่า Users ซึ่งใน Collection เดียวกันเราสามารถใส่ข้อมูลที่แตกต่างกันในแต่ละ Key แต่ละ Document ก็ได้ โดยในแต่ละ Key และ Document จะมีอิสระในการใส่ข้อมูลต่าง ๆ ลงไป แต่เราควรใส่ข้อมูลในแต่ละ Key ของ Document เป็นประเภทเดียวกันเพราะจะทำให้การค้นหาและการจัดเรียงลำดับของข้อมูลนั้นง่ายขึ้น

หมายเหตุ : ข้อที่เราควรรู้ ของการเก็บข้อมูลแบบ Collection คือ ชื่อของ Collection ห้ามซ้ำกัน, ห้ามมีข้อมูลดิบหรือ Collection อื่น ๆ อยู่ใน Collection ภายในจะเก็บได้เพียง Document เท่านั้น ไม่ต้องสร้างหรือลบ Collection ถ้าหากมีการเพิ่มข้อมูลลงไปโดยระบุชื่อ Collection และ Document จะทำให้ Collection นั้นถูกสร้างขึ้นโดยอัตโนมัติและถ้าหาก Collection นั้น ไม่มี Document ใด ๆ เหลืออยู่เลย Collection จะถูกลบโดยอัตโนมัติเช่นกัน

- Subcollection

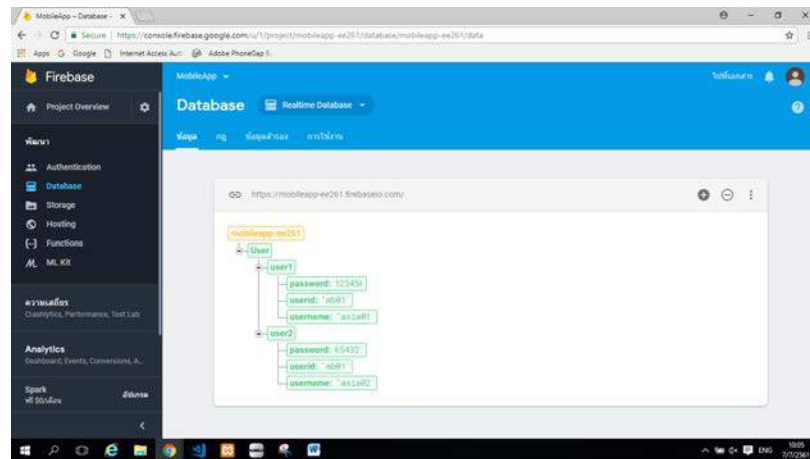


รูปที่ 2.13 ตัวอย่างการเก็บข้อมูลแบบ Subcollection

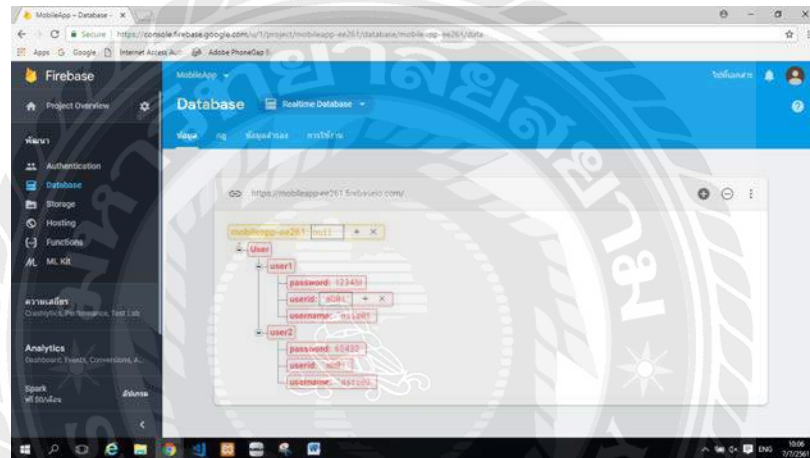
หากว่าเราอยากทำลำดับชั้นของข้อมูลทาง Cloud Firestore ก็ตอบสนองของสิ่งนั้น โดยเราสามารถสร้าง Subcollection ไว้ข้างใน Document เท่านั้น ซึ่งใน Subcollection ก็สามารมี Document ได้อีก เราซึ่งสามารถสร้าง Subcollection ของ Subcollection ได้อีก โดยสามารถซ้อนกันได้เลยอะสุดคือ 100 ลำดับชั้น

ตัวอย่างการเพิ่มข้อมูล

- ลักษณะการจัดการฐานข้อมูล



รูปที่ 2.14 แสดงลักษณะการเพิ่มข้อมูล



รูปที่ 2.15 แสดงลักษณะการอัปเดตและลบข้อมูล

สีที่แสดงใน console จะบอกสถานะแตกต่างกันดังนี้

- สีเหลือง: แสดงการอัปเดต
- สีเขียว: แสดงการเพิ่ม
- สีแดง: แสดงการลบ
- สีน้ำเงิน: แสดงการเคลื่อนย้าย

2.9 HTML

HTML คือ ภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล HTML ย่อมาจากคำว่า Hypertext Markup Language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิงค์ (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่ง

ต่าง ๆ ที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink

2.10 JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ ที่เรียกกันว่า สคริปต์ (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ ใช้ร่วมกับ HTML เพื่อให้เว็บไซต์ได้มีการเคลื่อนไหวสามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ แปลความและดำเนินงานไปทีละคำสั่ง (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)

บทที่ 3

การวิเคราะห์และการออกแบบ

ในบทนี้จะอธิบายการวิเคราะห์และการออกแบบทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ของระบบพร้อมทั้งอธิบายถึงภาพรวมในการทำงานของระบบเพื่อให้เห็นลักษณะการทำงานทั้งหมดของระบบ

3.1 ความต้องการของผู้ใช้งาน

จากการสำรวจความต้องการของผู้บริหารและพนักงานในการควบคุมพัดลมไอน้ำมีดังนี้

- 3.1.1 ผู้บริหารต้องการให้ไม่มีการใช้พลังงานในช่วงเวลาที่ไม่จำเป็น
- 3.1.2 พนักงานต้องการให้เปิดพัดลมไอน้ำก่อนเวลาทำงาน 10 นาที
- 3.1.3 พนักงานต้องการให้ปิดพัดลมไอน้ำก่อนเวลาเลิกงาน 10 นาที
- 3.1.4 พนักงานต้องการให้สามารถควบคุมพัดลมไอน้ำจากระยะไกลได้

3.2 การวิเคราะห์จากความต้องการของผู้ใช้งาน

จากความต้องการของผู้บริหารและพนักงานสามารถวิเคราะห์ได้ดังนี้

- 3.2.1 ระบบต้องสามารถกำหนดเวลาเปิดพัดลมไอน้ำล่วงหน้าได้
- 3.2.2 ระบบต้องสามารถกำหนดเวลาปิดพัดลมไอน้ำล่วงหน้าได้
- 3.2.3 ระบบต้องสามารถใช้บนเว็บแอปพลิเคชันได้

3.3 การออกแบบระบบ

ในโครงการนี้จะทำการสร้างระบบที่ใช้ควบคุมพัดลมไอน้ำให้สามารถควบคุมการใช้งานได้อย่างเต็มประสิทธิภาพ โดยสามารถสั่งเปิดหรือปิด, กำหนดเวลาเปิดหรือปิดล่วงหน้า และตรวจสอบค่าอุณหภูมิและค่าความชื้นบริเวณพัดลมไอน้ำ เพื่อใช้ในสำนักงานที่มีพัดลมไอน้ำจำนวนมาก โดยนำแนวคิดอินเทอร์เน็ตของสรรพสิ่งมาประยุกต์ใช้งาน สามารถใช้ระบบควบคุมบนเว็บแอปพลิเคชันได้ทำให้ใช้งานได้ในสถานที่และนอกสถานที่จากระยะไกลได้ ซึ่งระบบมีอยู่ด้วยกัน 2 ส่วนคือฮาร์ดแวร์และซอฟต์แวร์

3.3.1 การออกแบบระบบฮาร์ดแวร์

ใช้บอร์ด NodeMCU ที่มี Module ESP8266 เป็นตัวหลักและเรียกใช้ Library ESP8266WiFi เชื่อมต่อกับ 2.4 GHz WiFi ให้สามารถเข้าถึง Internet ได้เพื่อติดต่อกับ Google Firebase โดยใช้ Library FirebaseArduino เพื่อส่งข้อมูลค่าอุณหภูมิและค่าความชื้นที่อ่านได้จากเซ็นเซอร์ DHT22 และรับข้อมูลจาก Google Firebase เพื่อนำมาเป็นเงื่อนไขในการควบคุม Relay Module จำนวน 2 ตัว เพื่อทริกวงจรของพัดลมไอเย็นจำนวน 6 Channel ดังนี้

3.3.1.1 Channel ที่ 1 ทริก Power

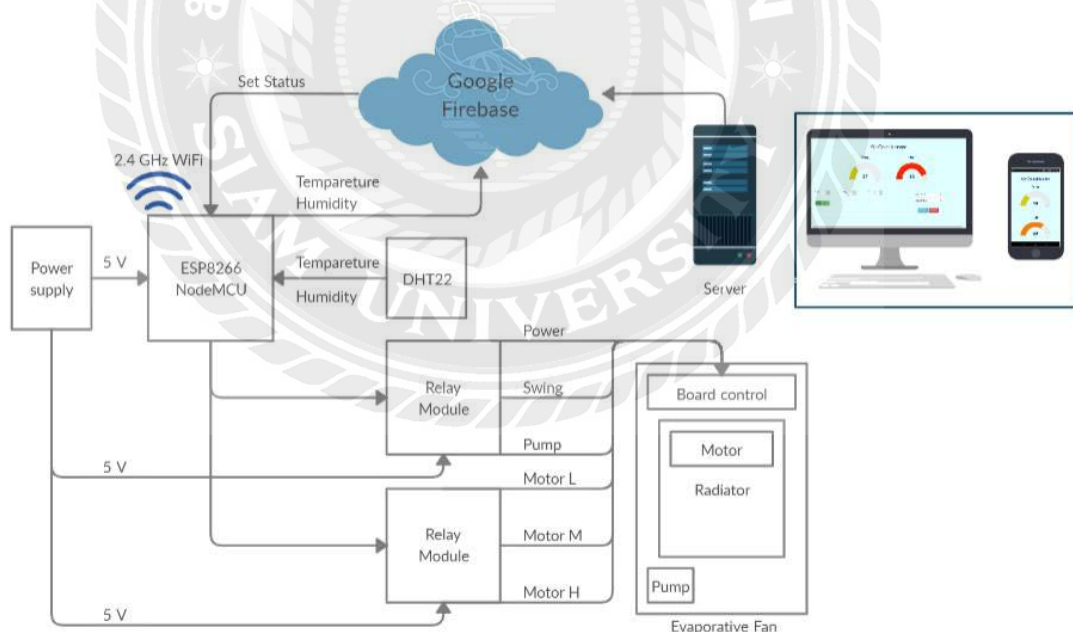
3.3.1.2 Channel ที่ 2 ทริก Swing

3.3.1.3 Channel ที่ 3 ทริก Pump

3.3.1.4 Channel ที่ 4 ทริก Motor Low

3.3.1.5 Channel ที่ 5 ทริก Motor Medium

3.3.1.6 Channel ที่ 6 ทริก Motor High



รูปที่ 3.1 ระบบฮาร์ดแวร์

3.2.2 การออกแบบระบบซอฟต์แวร์

เขียนเว็บแอปพลิเคชันเพื่อควบคุมพัดลมไอเย็นด้วยการออกแบบหน้าตา Interface โดยใช้ Front-end Framework Bootstrap และใช้ Library Firebase ในการติดต่อกับ Google

Firebase เพื่อนำค่าอุณหภูมิและค่าความชื้นมาแสดงโดยใช้ Library justGage ในการแสดงค่า และส่งข้อมูลเข้า Google Firebase โดยใช้ภาษา JavaScript ในการกำหนดการส่งข้อมูล มีทั้งหมด 5 สถานะดังนี้

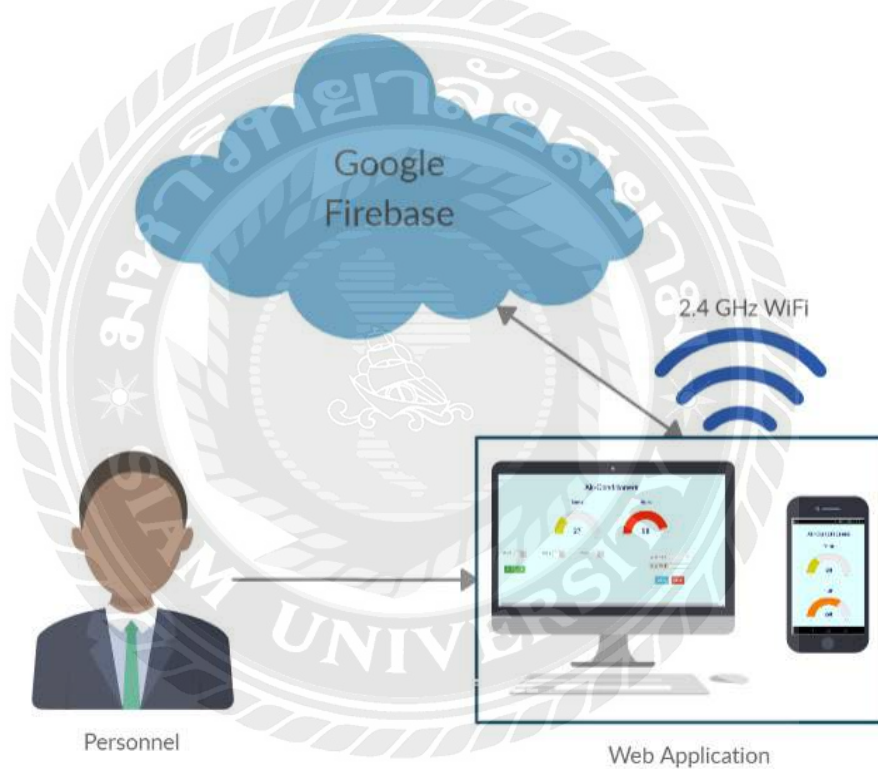
3.2.2.1 สั่งเปิดหรือปิด

3.2.2.2 สั่งเปิดหรือปิดระบบสาย

3.2.2.3 สั่งเปิดหรือปิดปั๊มน้ำ

3.2.2.4 ตั้งเวลาเปิดหรือปิด

3.2.2.5 เลือกแรงลมระดับ Low, Medium และ High



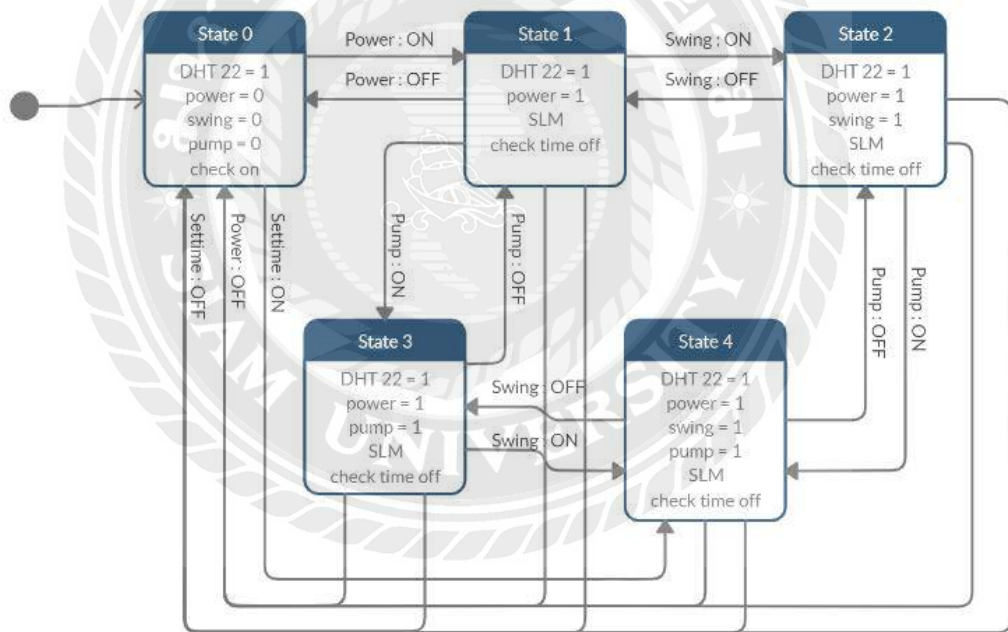
รูปที่ 3.2 ระบบซอฟต์แวร์

3.4 การทำงานของระบบ

ในการทำงานของระบบนั้น ทางด้านฮาร์ดแวร์จะทำการเชื่อมต่อ 2.4 GHz WiFi ให้สามารถเข้าถึง Google Firebase และทำการเก็บข้อมูลค่าอุณหภูมิและค่าความชื้นที่ได้จากเซ็นเซอร์ DHT22 และดึงข้อมูลการกำหนดสถานะพัดลมไอน้ำเข้ามาควบคุม Relay Module ไปทริก

วงจรพัดลมไอน้ำคือ สถานะเปิดพัดลมไอน้ำ Relay 1 ทำงาน, สถานะปิดพัดลมไอน้ำ Relay 1 ไม่ทำงาน, สถานะเปิดระบบสาย Relay 2 ทำงาน, สถานะปิดระบบสาย Relay 2 ไม่ทำงาน, สถานะเปิดปั้มน้ำ Relay 3 ทำงาน, สถานะปิดปั้มน้ำ Relay 3 ไม่ทำงาน, สถานะแรงลมต่ำ Relay 4 ทำงาน, สถานะแรงลมกลาง Relay 5 ทำงาน และสถานะแรงลมสูง Relay 6 ทำงาน ส่วนทางด้านซอฟต์แวร์เมื่อเปิดใช้งานเว็บแอปพลิเคชันจะติดต่อกับ Google Firebase และดึงข้อมูลค่าอุณหภูมิและค่าความชื้นมาแสดงบน Gage และสามารถส่งค่ากำหนดสถานะของพัดลมไอน้ำได้คือ เปิดหรือปิดพัดลมไอน้ำ, เปิดหรือปิดระบบสาย, เปิดหรือปิดปั้มน้ำ, กำหนดเวลาเปิดหรือปิด และเลือกระดับแรงลมต่ำ, กลาง และสูง

3.5 แผนภาพแสดงสถานะ (State Diagram)



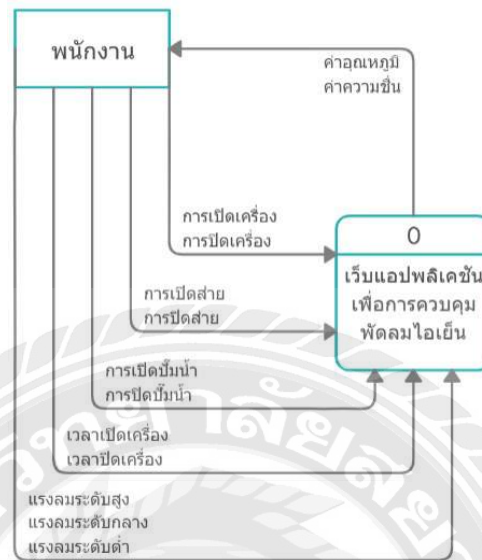
รูปที่ 3.3 State Diagram

จากรูปที่ 3.3 เป็นการแสดงสถานะการทำงานของระบบ มีทั้งหมด 5 สถานะ

- State 0 คือสถานะปิดพัดลมไอน้ำ State นี้จะทำการเก็บค่าอุณหภูมิและค่าความชื้น, เช็คเวลาเปิดอัตโนมัติ เมื่อถึงเวลาที่กำหนดจะเปลี่ยนสถานะไปที่ State 4 และถ้าค่า Power = ON จะเปลี่ยนสถานะไปที่ State 1

- State 1 คือสถานะเปิดพัดลมไอเย็น State นี้จะทำการเก็บค่าอุณหภูมิและค่าความชื้น, Power ทำงาน, สามารถเลือกกระดပ်แรงลม, เช็คเวลาปิดอัตโนมัติ เมื่อถึงเวลาที่กำหนดจะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Power = OFF จะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Swing = ON จะเปลี่ยนสถานะไปที่ State 2 และถ้าค่า Pump = ON จะเปลี่ยนสถานะไปที่ State 3
- State 2 คือสถานะเปิดพัดลมไอเย็นและระบบส่าย State นี้จะทำการเก็บค่าอุณหภูมิและค่าความชื้น, Power ทำงาน, Swing ทำงาน, สามารถเลือกกระดပ်แรงลม, เช็คเวลาปิดอัตโนมัติ เมื่อถึงเวลาที่กำหนดจะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Power = OFF จะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Swing = OFF จะเปลี่ยนสถานะไปที่ State 1 และถ้าค่า Pump = ON จะเปลี่ยนสถานะไปที่ State 4
- State 3 คือสถานะเปิดพัดลมไอเย็นและปั้มน้ำ State นี้จะทำการเก็บค่าอุณหภูมิและค่าความชื้น, Power ทำงาน, Pump ทำงาน, สามารถเลือกกระดပ်แรงลม, เช็คเวลาปิดอัตโนมัติ เมื่อถึงเวลาที่กำหนดจะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Power = OFF จะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Swing = ON จะเปลี่ยนสถานะไปที่ State 4 และถ้าค่า Pump = OFF จะเปลี่ยนสถานะไปที่ State 1
- State 4 คือสถานะเปิดพัดลมไอเย็นระบบส่ายและปั้มน้ำ State นี้จะทำการเก็บค่าอุณหภูมิและค่าความชื้น, Power ทำงาน, Swing ทำงาน, Pump ทำงาน, สามารถเลือกกระดပ်แรงลม, เช็คเวลาปิดอัตโนมัติ เมื่อถึงเวลาที่กำหนดจะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Power = OFF จะเปลี่ยนสถานะไปที่ State 0, ถ้าค่า Swing = OFF จะเปลี่ยนสถานะไปที่ State 3 และถ้าค่า Pump = OFF จะเปลี่ยนสถานะไปที่ State 2

3.6 แผนภาพการไหลของข้อมูล (Context Diagram)

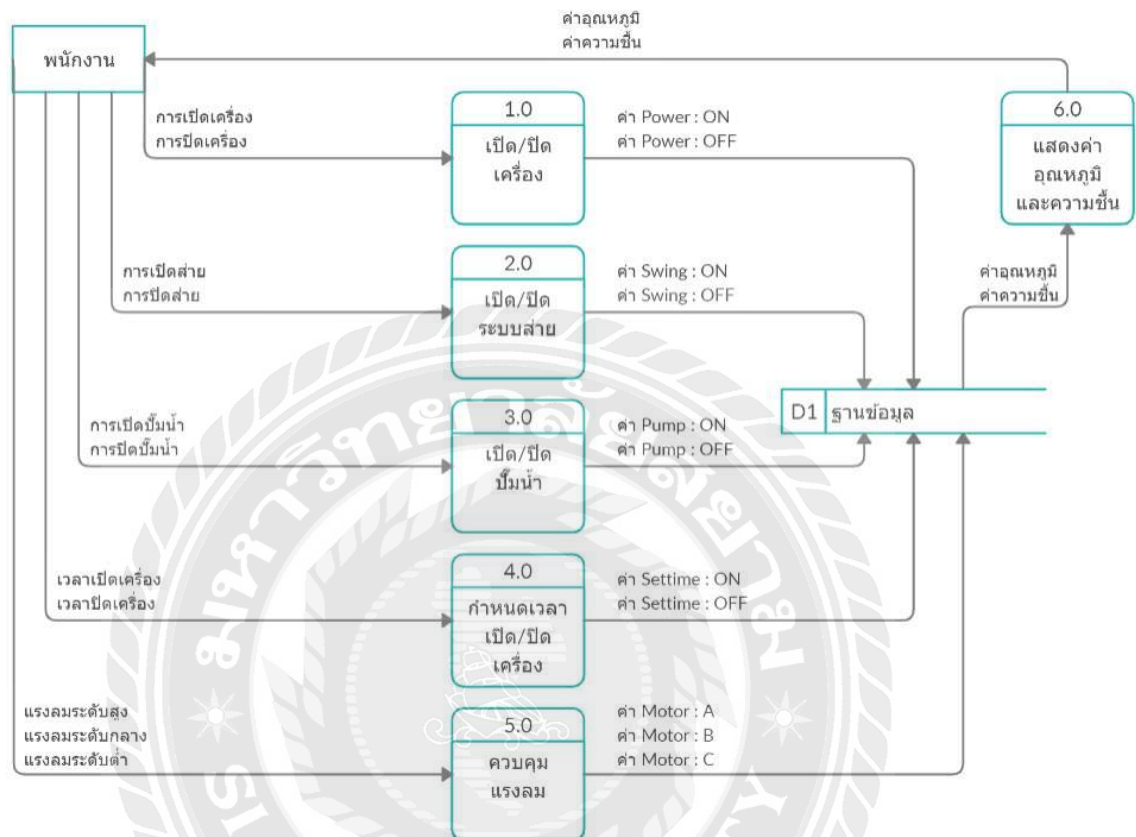


รูปที่ 3.4 Context Diagram

จากรูปที่ 3.4 แสดงผู้ที่เกี่ยวข้องกับข้อมูลและทิศทางการไหลของข้อมูลภายในระบบ คือ

- พนักงานเพิ่มข้อมูลเข้าสู่ระบบคือ การเปิดเครื่อง, การปิดเครื่อง, การเปิดสาย, การปิดสาย, การเปิดปั้มน้ำ, การปิดปั้มน้ำ, เวลาเปิดเครื่อง, เวลาปิดเครื่อง, แรงลมระดับต่ำ, แรงลมระดับกลาง และแรงลมระดับสูง ส่วนข้อมูลที่พนักงานได้จากระบบคือ แสดงค่าอุณหภูมิ และแสดงค่าความชื้น

3.7 Data Flow Diagram (DFD)



รูปที่ 3.5 Data Flow Diagram

3.8 Process Description

ตารางที่ 3.1 คำอธิบายการประมวลผลของกระบวนการที่ 1 : เปิด/ปิดเครื่อง

Process Description	
System	การควบคุมพัดลมไอน้ำ
DFD Number	1
Process Name	เปิด/ปิดเครื่อง
Input data flow	- การเปิดเครื่อง, การปิดเครื่อง
Output data flow	- ค่า Power : ON, ค่า Power : OFF
Description	- พนักงานกดปุ่มเปิดหรือปิดเครื่อง

ตารางที่ 3.2 คำอธิบายการประมวลผลของกระบวนการที่ 2 : เปิด/ปิดระบบสาย

Process Description	
System	การควบคุมพัดลมไอน้ำ
DFD Number	2
Process Name	เปิด/ปิดระบบสาย
Input data flow	- การเปิดสาย, การปิดสาย
Output data flow	- ค่า Swing : ON, ค่า Swing : OFF
Description	- พนักงานกดปุ่มเปิดหรือปิดสาย

ตารางที่ 3.3 คำอธิบายการประมวลผลของกระบวนการที่ 3 : เปิด/ปิดปั๊มน้ำ

Process Description	
System	การควบคุมพัดลมไอน้ำ
DFD Number	3
Process Name	เปิด/ปิดปั๊มน้ำ
Input data flow	- การเปิดปั๊มน้ำ, การปิดปั๊มน้ำ
Output data flow	- ค่า Pump : ON, ค่า Pump : OFF
Description	- พนักงานกดปุ่มเปิดหรือปิดปั๊มน้ำ

ตารางที่ 3.4 คำอธิบายการประมวลผลของกระบวนการที่ 4 : กำหนดเวลาเปิด/ปิดเครื่อง

Process Description	
System	การควบคุมพัดลมไอน้ำ
DFD Number	4
Process Name	กำหนดเวลาเปิด/ปิดเครื่อง
Input data flow	- เวลาเปิดเครื่อง, เวลาปิดเครื่อง
Output data flow	- ค่า Settime : ON, ค่า Settime : OFF
Description	- พนักงานเลือกเวลาเป็นชั่วโมงและนาที - กดปุ่ม SetOn/SetOff

ตารางที่ 3.5 คำอธิบายการประมวลผลของกระบวนการที่ 5 : ควบคุมแรงลม

Process Description	
System	การควบคุมพัดลมไอน้ำเย็น
DFD Number	5
Process Name	ควบคุมแรงลม
Input data flow	- แรงลมระดับสูง, แรงลมระดับกลาง, แรงลมระดับต่ำ
Output data flow	- ค่า Motor : A, ค่า Motor : B, ค่า Motor : C
Description	- พนักงานกดปุ่มเลือกกระดืบ 1, 2, 3

ตารางที่ 3.6 คำอธิบายการประมวลผลของกระบวนการที่ 6 : แสดงค่าอุณหภูมิและความชื้น

Process Description	
System	การควบคุมพัดลมไอน้ำเย็น
DFD Number	6
Process Name	แสดงค่าอุณหภูมิและความชื้น
Input data flow	- ค่าอุณหภูมิ, ค่าความชื้น
Output data flow	- ค่าอุณหภูมิ, ค่าความชื้น
Description	- ระบบแสดงค่าอุณหภูมิและความชื้น

บทที่ 4

การทดลอง

เพื่อให้ทราบถึงผลการทำงานของระบบจึงทำการทดลองการทำงานของระบบ ว่าถูกต้องตามที่ออกแบบหรือเกิดข้อผิดพลาด ซึ่งถ้าเกิดข้อผิดพลาดจะได้นำมาทำการแก้ไข ปรับปรุงระบบให้สามารถทำงานได้อย่างมีประสิทธิภาพ

4.1 การกำหนดพอร์ตของ ESP8266 NodeMCU

- 4.1.1 พอร์ตที่ห้า Relay 1
- 4.1.2 พอร์ตที่สี่ Relay 2
- 4.1.3 พอร์ตที่ศูนย์ Relay 3
- 4.1.4 พอร์ตที่สอง Relay 4
- 4.1.5 พอร์ตที่สิบสี่ Relay 5
- 4.1.6 พอร์ตที่สิบสอง Relay 6
- 4.1.7 พอร์ตที่สิบสาม DHT22

ตารางที่ 4.1 การกำหนดพอร์ตการใช้งาน

Port	นำมาใช้	ใช้งาน
Digital port 1	Relay 1	Power
Digital port 2	Relay 2	Swing
Digital port 3	Relay 3	Pump
Digital port 4	Relay 4	MotorLow
Digital port 5	Relay 5	MotorMedium
Digital port 6	Relay 6	MotorHigh
Digital port 7	DHT22	



รูปที่ 4.1 การกำหนดพอร์ตของ ESP8266 NodeMCU

4.2 การกำหนดไฟฟ้าภายใน

- 4.2.1 ใช้ไฟฟ้ากระแสตรง 5 V
- 4.2.2 ใช้ไฟฟ้ากระแสสลับ 220 V
- 4.2.3 การใช้ไฟฟ้าในอุปกรณ์ต่างๆ
- 4.2.4 ไฟเลี้ยง Switching Power Supply 220 VAC
- 4.2.5 ไฟเลี้ยงบอร์ด ESP8266 NodeMCU 5 VDC
- 4.2.6 ไฟเลี้ยง Relay Module 4 Channel 2 ตัว 5 VDC

4.3 การทดลองเปิดพัดลมไอน์บนเว็บแอปพลิเคชัน

วัตถุประสงค์

- 4.3.1 เพื่อเปิดพัดลมไอน์จากระยะไกลได้

อุปกรณ์ที่ใช้ในการทดลอง

4.3.2	พัดลมไอนเย็น	จำนวน 1 เครื่อง
4.3.3	คอมพิวเตอร์โน้ตบุ๊ก	จำนวน 1 เครื่อง
4.3.4	สมาร์ทโฟน	จำนวน 1 เครื่อง
4.3.5	ชุดระบบควบคุมพัดลมไอนเย็น	จำนวน 1 ชุด

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟน กดปุ่ม Power ให้เป็นสถานะเปิด

ตารางผลการทดลองที่ได้

— ทดสอบโดยพนักงาน 5 คน จำนวน 10 ครั้ง

ตารางที่ 4.2 ทดลองเปิดพัดลมไอนเย็นบนเว็บแอปพลิเคชัน

คนที่	ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10
1	เปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	เปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	เปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	เปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	เปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4.4 การทดลองปิดพัดลมไอนเย็นบนเว็บแอปพลิเคชัน

วัตถุประสงค์

4.4.1 เพื่อปิดพัดลมไอนเย็นจากระยะไกลได้

อุปกรณ์ที่ใช้ในการทดลอง

4.4.2	พัดลมไอนเย็น	จำนวน 1 เครื่อง
4.4.3	คอมพิวเตอร์โน้ตบุ๊ก	จำนวน 1 เครื่อง
4.4.4	สมาร์ทโฟน	จำนวน 1 เครื่อง
4.4.5	ชุดระบบควบคุมพัดลมไอนเย็น	จำนวน 1 ชุด

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟน กดปุ่ม Power ให้เป็นสถานะปิด

ตารางผลการทดลองที่ได้

— ทดสอบโดยพนักงาน 5 คน จำนวน 10 ครั้ง

ตารางที่ 4.3 ทดลองปิดพัดลมไอนเย็นบนเว็บแอปพลิเคชัน

คนที่	ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10
1	ปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	ปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	ปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	ปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	ปิด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4.5 การทดลองกำหนดเวลาเปิดพัดลมไอนเย็นโดยอัตโนมัติ

วัตถุประสงค์

4.5.1 เพื่อกำหนดเวลาเปิดพัดลมไอนเย็นล่วงหน้า

อุปกรณ์ที่ใช้ในการทดลอง

4.5.2	พัดลมไอนเย็น	จำนวน 1 เครื่อง
4.5.3	คอมพิวเตอร์โน้ตบุ๊ก	จำนวน 1 เครื่อง
4.5.4	สมาร์ทโฟน	จำนวน 1 เครื่อง
4.5.5	ชุดระบบควบคุมพัดลมไอนเย็น	จำนวน 1 ชุด

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟน ทำการกำหนดเวลาที่แถบ Select Option และกดปุ่ม SetOn

ตารางผลการทดลองที่ได้

- ทดสอบโดยกำหนดเวลาเปิดพัดลมไอนเย็น 8:00, 8:30, 9:00, 9:30 และ 10:00 น. จำนวน 3 วัน

ตารางที่ 4.4 ทดลองกำหนดเวลาเปิดพัดลมไอนเย็นโดยอัตโนมัติ

วันที่	ทดสอบ	08:00	08:30	09:00	09:30	10:00
1	กำหนดเวลาเปิด	✓	✓	✓	✓	✓
2	กำหนดเวลาเปิด	✓	✓	✓	✓	✓
3	กำหนดเวลาเปิด	✓	✓	✓	✓	✓

4.6 การทดลองกำหนดเวลาปิดพัดลมไอนเย็นโดยอัตโนมัติ

วัตถุประสงค์

- 4.6.1 เพื่อกำหนดเวลาปิดพัดลมไอนเย็นล่วงหน้า

อุปกรณ์ที่ใช้ในการทดลอง

4.6.2	พัดลมไอนเย็น	จำนวน 1 เครื่อง
4.6.3	คอมพิวเตอร์โน้ตบุ๊ก	จำนวน 1 เครื่อง
4.6.4	สมาร์ทโฟน	จำนวน 1 เครื่อง
4.6.5	ชุดระบบควบคุมพัดลมไอนเย็น	จำนวน 1 ชุด

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟน ทำการกำหนดเวลาที่แถบ Select Option และกดปุ่ม SetOff

ตารางผลการทดลองที่ได้

- ทดสอบโดยกำหนดเวลาเปิดตอน 8:15, 8:45, 9:15, 9:45 และ 10:15 น. จำนวน 3 วัน

ตารางที่ 4.5 ทดลองกำหนดเวลาเปิดพัดลมไอนเย็นโดยอัตโนมัติ

วันที่	ทดสอบ	08:15	08:45	08:15	08:45	08:15
1	กำหนดเวลาเปิด	✓	✓	✓	✓	✓
2	กำหนดเวลาเปิด	✓	✓	✓	✓	✓
3	กำหนดเวลาเปิด	✓	✓	✓	✓	✓

4.7 การทดลองการตรวจสอบค่าอุณหภูมิและค่าความชื้น

วัตถุประสงค์

- 4.7.1 เพื่อตรวจสอบค่าอุณหภูมิและค่าความชื้นบนเว็บแอปพลิเคชัน

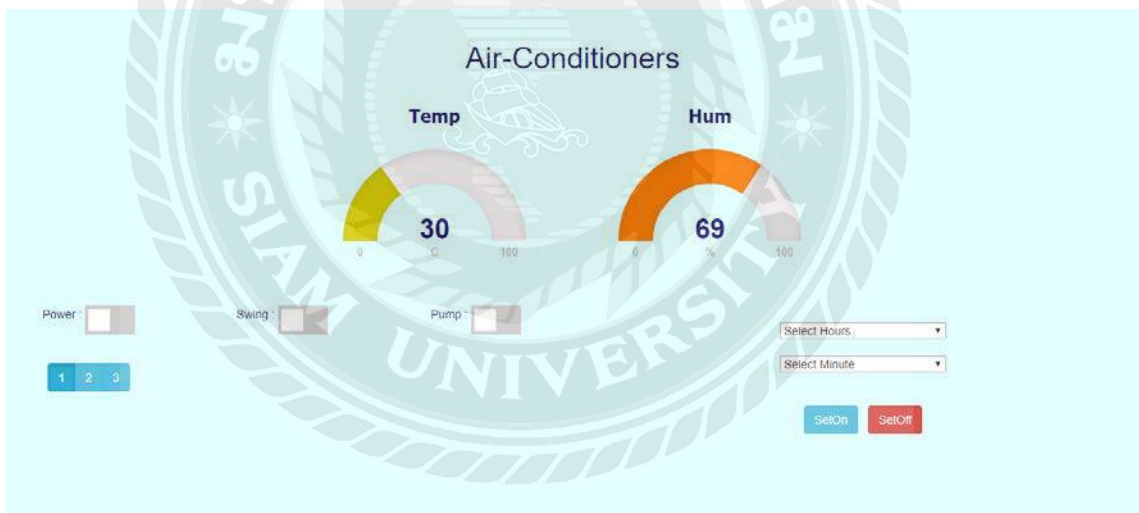
อุปกรณ์ที่ใช้ในการทดลอง

4.7.2	พัดลมไอเย็น	จำนวน 1 เครื่อง
4.7.3	คอมพิวเตอร์โน้ตบุ๊ก	จำนวน 1 เครื่อง
4.7.4	สมาร์ทโฟน	จำนวน 1 เครื่อง
4.7.5	ชุดระบบควบคุมพัดลมไอเย็น	จำนวน 1 ชุด

ขั้นตอนการทดลอง

เมื่อเสียบพัดลมไอเย็นระบบควบคุมจะทำงานและเมื่อทำการเปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟนสามารถตรวจสอบค่าอุณหภูมิและค่าความชื้นได้บนหน้าจอเว็บแอปพลิเคชัน

ดังรูปที่ 4.3



รูปที่ 4.3 แสดงค่าอุณหภูมิและค่าความชื้น

ตารางผลการทดลองที่ได้

- ตรวจสอบค่าอุณหภูมิและค่าความชื้นทุกๆ 5 นาที ภายในเวลา 30 นาที

ตารางที่ 4.6 ผลการทดลองการตรวจสอบค่าอุณหภูมิและค่าความชื้น

นาทีที่	ค่าอุณหภูมิที่ได้	ค่าความชื้นที่ได้
5	30 °C	68 %
10	31 °C	68 %
15	30 °C	69 %
20	29 °C	68 %
25	30 °C	69 %
30	30 °C	69 %

4.8 การทดลองเปิดและปิดปั๊มน้ำ

วัตถุประสงค์

- 4.8.1 เพื่อทดลองเปิดและปิดปั๊มน้ำบนเว็บแอปพลิเคชัน

อุปกรณ์ที่ใช้ในการทดลอง

- | | | |
|-------|--------------------------|-----------------|
| 4.8.2 | พัดลมไอเย็น | จำนวน 1 เครื่อง |
| 4.8.3 | คอมพิวเตอร์โน้ตบุ๊ก | จำนวน 1 เครื่อง |
| 4.8.4 | สมาร์ทโฟน | จำนวน 1 เครื่อง |
| 4.8.5 | ชุดระบบควบคุมพัดลมไอเย็น | จำนวน 1 ชุด |

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและทำการกดปุ่ม Power ให้เป็นสถานะเปิด เพื่อให้พัดลมไอเย็นเริ่มทำงานก่อนหลังจากนั้นทำการกดปุ่ม Pump ให้เป็นสถานะเปิด ปั๊มน้ำของ

พัดลมไอน์เริ่มทำงาน และเมื่อทำการกดปุ่ม Pump ให้เป็นสถานะปิด ปุ่มน้ำของพัดลมไอน์หยุดทำงาน

ตารางผลการทดลองที่ได้

- ค่าอุณหภูมิและค่าความชื้นขณะเปิดปั้มน้ำและปิดปั้มน้ำทุกๆ 5 นาที ภายในเวลา 30 นาที

ตารางที่ 4.7 ผลการทดลองขณะเปิดปั้มน้ำและปิดปั้มน้ำ

นาทีที่	อุณหภูมิตอนปั้มน้ำเปิด	ความชื้นตอนปั้มน้ำเปิด	อุณหภูมิตอนปั้มน้ำปิด	ความชื้นตอนปั้มน้ำปิด
5	24 °C	89 %	26 °C	78 %
10	24 °C	91 %	26 °C	77 %
15	23 °C	92 %	27 °C	78 %
20	23 °C	91 %	26 °C	79 %
25	24 °C	93 %	27 °C	78 %
30	23 °C	90 %	26 °C	79 %

4.9 การทดลองเปิดระบบสายของพัดลมไอน์บนเว็บแอปพลิเคชัน

วัตถุประสงค์

4.9.1 เพื่อเปิดระบบสายของพัดลมไอน์บนเว็บแอปพลิเคชัน

อุปกรณ์ที่ใช้ในการทดลอง

- | | | |
|-------|------------------------|-----------------|
| 4.9.2 | พัดลมไอน์ | จำนวน 1 เครื่อง |
| 4.9.3 | คอมพิวเตอร์โน้ตบุ๊ก | จำนวน 1 เครื่อง |
| 4.9.4 | สมาร์ทโฟน | จำนวน 1 เครื่อง |
| 4.9.5 | ชุดระบบควบคุมพัดลมไอน์ | จำนวน 1 ชุด |

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟนและทำการกดปุ่ม Power ให้เป็นสถานะเปิดเพื่อให้พัดลมไอเย็นเริ่มทำงานก่อนหลังจากนั้นทำการกดปุ่ม Swing ให้เป็นสถานะเปิด ระบบสายของพัดลมไอเย็นเริ่มทำงาน

ตารางผลการทดลองที่ได้

— ทดสอบโดยพนักงาน 5 คน จำนวน 10 ครั้ง

ตารางที่ 4.8 ผลการทดลองขณะเปิดระบบสาย

คนที่	ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10
1	เปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	เปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	เปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	เปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	เปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4.10 การทดลองปิดระบบสายของพัดลมไอเย็นบนเว็บแอปพลิเคชัน

วัตถุประสงค์

4.10.1 เพื่อปิดระบบสายของพัดลมไอเย็นบนเว็บแอปพลิเคชัน

อุปกรณ์ที่ใช้ในการทดลอง

- | | |
|----------------------------|-----------------|
| 4.10.2 พัดลมไอเย็น | จำนวน 1 เครื่อง |
| 4.10.3 คอมพิวเตอร์โน้ตบุ๊ก | จำนวน 1 เครื่อง |
| 4.10.4 สมาร์ทโฟน | จำนวน 1 เครื่อง |

4.10.5 ชุดระบบควบคุมพัดลมไอน้เย็น

จำนวน 1 ชุด

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟน กดปุ่ม Swing ให้เป็นสถานะปิด

ตารางผลการทดลองที่ได้

— ทดสอบโดยพนักงาน 5 คน จำนวน 10 ครั้ง

ตารางที่ 4.9 ผลการทดลองขณะปิดระบบสาย

คนที่	ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10
1	ปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	ปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	ปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	ปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	ปิดสาย	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4.11 การทดลองเลือกแรงลมของพัดลมไอน้เย็นบนเว็บแอปพลิเคชัน

วัตถุประสงค์

4.11.1 เพื่อเลือกแรงลมของพัดลมไอน้เย็นระดับต่ำ, กลาง และสูงบนเว็บแอปพลิเคชัน

อุปกรณ์ที่ใช้ในการทดลอง

4.11.2 พัดลมไอน้เย็น

จำนวน 1 เครื่อง

4.11.3 คอมพิวเตอร์โน้ตบุ๊ก

จำนวน 1 เครื่อง

4.11.4 โทรศัพท์

จำนวน 1 เครื่อง

4.11.5 ชุดระบบควบคุมพัดลมไอน์

จำนวน 1 ชุด

ขั้นตอนการทดลอง

เปิดเว็บแอปพลิเคชันบนคอมพิวเตอร์โน้ตบุ๊กและสมาร์ทโฟนและทำการกดปุ่ม Power ให้เป็นสถานะเปิดเพื่อให้พัดลมไอน์เริ่มทำงานก่อนหลังจากนั้นทำการกดปุ่ม 1 เป็นการเลือก ระดับต่ำ กดปุ่ม 2 เป็นการเลือกระดับกลาง และกดปุ่ม 3 เป็นการเลือกระดับสูง

ตารางผลการทดลองที่ได้

— ทดสอบโดยพนักงาน 5 คน

ตารางที่ 4.10 ทดลองเลือกแรงลมของพัดลมไอน์

คนที่	ทดสอบ	ต่ำ	กลาง	สูง
1	เลือกแรงลม	✓	✓	✓
2	เลือกแรงลม	✓	✓	✓
3	เลือกแรงลม	✓	✓	✓
4	เลือกแรงลม	✓	✓	✓
5	เลือกแรงลม	✓	✓	✓

บทที่ 5

สรุปผลและข้อเสนอแนะ

การจัดทำโครงการระบบอนุรักษ์พลังงานจากพัดลมไอน้ำได้สรุปผลการดำเนินงาน ปัญหา และข้อเสนอแนะในการทำงานดังต่อไปนี้

5.1 สรุปผลการดำเนินโครงการ

การดำเนินงานในระบบอนุรักษ์พลังงานจากพัดลมไอน้ำ โดยทำการทดลอง สรุปผลการทดสอบการใช้งานได้ดังนี้

5.1.1 การเปิดพัดลมไอน้ำบนเว็บแอปพลิเคชันทำการทดสอบ เปิด พบว่าสามารถใช้งานระบบได้จริง

5.1.2 การปิดพัดลมไอน้ำบนเว็บแอปพลิเคชันทำการทดสอบ ปิด พบว่าสามารถใช้งานระบบได้จริง

5.1.3 การกำหนดเวลาเปิดพัดลมไอน้ำโดยอัตโนมัติทำการทดสอบ กำหนดเวลาเปิด พบว่าสามารถใช้งานระบบได้จริง

5.1.4 การกำหนดเวลาปิดพัดลมไอน้ำโดยอัตโนมัติทำการทดสอบ กำหนดเวลาปิด พบว่าสามารถใช้งานระบบได้จริง

5.1.5 การตรวจสอบค่าอุณหภูมิและค่าความชื้นทำการทดสอบ ตรวจสอบ พบว่าสามารถใช้งานระบบได้จริง

5.1.6 การเปิดและปิดปั๊มน้ำทำการทดสอบ เปิดและปิด พบว่าสามารถใช้งานระบบได้จริง

5.1.7 การเปิดระบบสายของพัดลมไอน้ำบนเว็บแอปพลิเคชันทำการทดสอบ เปิด พบว่าสามารถใช้งานระบบได้จริง

5.1.8 การปิดระบบสายของพัดลมไอน้ำบนเว็บแอปพลิเคชันทำการทดสอบ ปิด พบว่าสามารถใช้งานระบบได้จริง

5.1.9 การเลือกแรงลมของพัดลมไอน้ำบนเว็บแอปพลิเคชันทำการทดสอบ เลือกระดับต่ำ, กลาง และสูง พบว่าสามารถใช้งานระบบได้จริง

5.2 ปัญหาการดำเนินโครงการ

5.1.1 บอร์ด อีเอสพี 8266 โหนดเอ็มซียูต้องมีการเข้าถึงกุเกิลไฟร์เบส และ ไวไฟ 2.4 จิกะเฮิร์ตซ์ทำให้เกิดความล่าช้าในการทำงานของระบบ

5.1.2 ไลบรารีไฟร์เบสของอาดูยโนมีปัญหาที่ไม่ได้รับการแก้ไขจากผู้พัฒนา

5.3 ข้อเสนอแนะ

5.1.1 ควรมีเซ็นเซอร์ตรวจวัดกระแสไฟฟ้า

5.1.2 สามารถเพิ่มการแจ้งเตือนสถานะของพัดลมไอเย็นทาง Line Notification



บรรณานุกรม

- เท็นเนอียี่ อินโนเวชัน. (2562). การอ่านค่าอุณหภูมิและความชื้นตัวเซนเซอร์ DHT22 ผ่าน *onewire bus*. เข้าถึงได้จาก https://www.tenergyinnovation.co.th/arduino_learning_kit/arduino-learning-kit-ep7-การอ่านค่าอุณหภูมิ
- แฟมิลีแอร์ เอ็นจิเนียริง. (ม.ป.ป.). *พัดลมไอน้ำเย็น (Evaporative Air Cooler)*. เข้าถึงได้จาก <https://www.familyairengineering.com/14489908/พัดลมไอน้ำเย็น-evaporative-air-cooler>
- วิกิพีเดีย. (ม.ป.ป.). HTML. วันที่สืบค้น 13 กันยายน 2562, จาก <https://en.wikipedia.org/wiki/HTML>
- Firebase. (n.d.). *Firebase Realtime Database*. Retrieved from <https://firebase.google.com/docs/database/>
- mindphp.com. (2557). *คู่มือ php การใช้งาน function class ฟังก์ชัน Mysql ฐานข้อมูล CSS และ HTML การตัวแปรแบบ static ภายในฟังก์ชัน*. เข้าถึงได้จาก <http://mindphp.com/คู่มือ/73-คืออะไร2187-java-javascript-คืออะไร.html>
- mindphp.com. (2560). *รู้จักกับ Visual Studio Code (วิซวล สตูดิโอ โค้ด) โปรแกรมฟรีจากค่ายไมโครซอฟท์*. เข้าถึงได้จาก <http://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>
- mindphp.com. (2560). *Xampp คืออะไร*. เข้าถึงได้จาก <http://mindphp.com/คู่มือ/73-คืออะไร/2637-xampp-คืออะไร.html>
- Poundxi.com. (2559). *Arduino คืออะไร*. เข้าถึงได้จาก <https://poundxi.com/arduino>
- Poundxi.com. (2561). *NodeMCU คืออะไร*. เข้าถึงได้จาก <https://poundxi.com/nodemcu>
- VEEDVIL.COM. (2558). *ทำความเข้าใจเรื่อง Internet of Things (IoT) เทรนด์ที่หลายคนกำลังพูดถึง*. เข้าถึงได้จาก <http://www.veedvil.com/news/internet-of-things-iot/>



ภาคผนวก



ภาคผนวก ก
การติดตั้งอุปกรณ์ฮาร์ดแวร์

การติดตั้งชุดระบบควบคุมเข้ากับพัดลมไอเย็น



รูปที่ ก.1 การเจาะฝาหลังของพัดลมไอเย็น



รูปที่ ก.2 การติดตั้งบอร์ด ESP8266 NodeMCU, Relay Module 4 Channel และ Switching

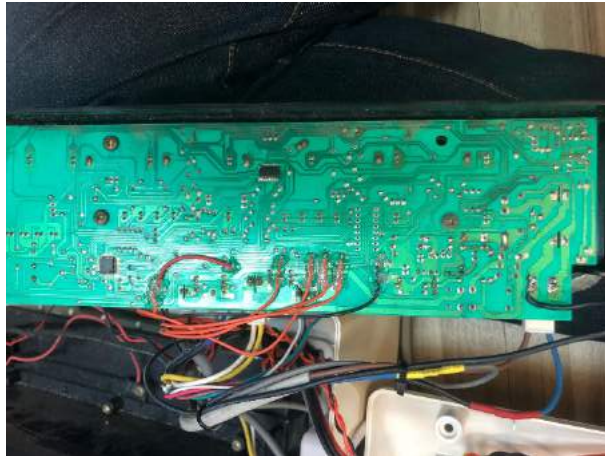
Power Supply 5V 5A



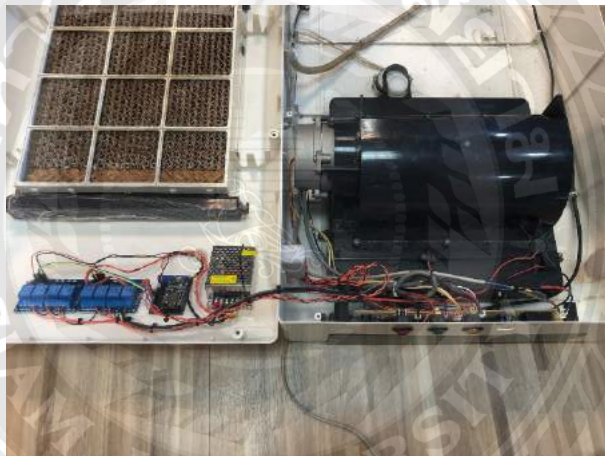
รูปที่ ก.3 การติดตั้งเซ็นเซอร์ DHT22 บริเวณที่ลมผ่าน



รูปที่ ก.4 การติดตั้งชุดสายไฟ



รูปที่ ก.5 การติดตั้งชุดสายไฟที่ใช้ทริกวงจรพัดลมไอน์



รูปที่ ก.6 ติดตั้งชุดระบบควบคุมเรียบร้อย



รูปที่ ก.7 พัดลมไอเย็นที่ติดตั้งชุดระบบควบคุมแล้ว



ประวัติผู้จัดทำ

นายธนกร จวงเจิม

วันเดือนปีเกิด 23 พฤษภาคม พ.ศ. 2540

รหัสนักศึกษา 5804000006

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์



ที่อยู่ : 10/236 หมู่.7 ตำบลโคกขาม อำเภอเมือง จังหวัดสมุทรสาคร 74000

ประวัติการศึกษา

- ปีพุทธศักราช 2555 ประกาศนียบัตรวิชาชีพ แผนกอิเล็กทรอนิกส์ วิทยาลัยเทคโนโลยีฐานเทคโนโลยี

นายพศวัต ไกรศรีสมบัติ

วันเดือนปีเกิด 22 เมษายน พ.ศ. 2539

รหัสนักศึกษา 5804000007

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์



ที่อยู่ : 60 หมู่.5 ตำบลทุ่งกระพังโหม อำเภอกำแพงแสน จังหวัดนครปฐม 73140

ประวัติการศึกษา

- ปีพุทธศักราช 2555 ประกาศนียบัตรวิชาชีพ แผนกอิเล็กทรอนิกส์ วิทยาลัยเทคนิคนครปฐม