



การพัฒนาสถานีอากาศท้องถิ่นเพื่อการท่องเที่ยวเชิงอนุรักษ์
The Development of Local Weather Station for Ecotourism

ธนาธร เจ๊ะดาโห๊ะ

Thanathorn Jeadahoa

สัญญากรณ์ ปัทมรักษ์

Sanpakorn Pattamarak

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตร์บัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล
มหาวิทยาลัยวลัยลักษณ์

2565

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Information Technology and Digital Innovation

Walailak University

2022



การพัฒนาสถานีอากาศท้องถิ่นเพื่อการท่องเที่ยวเชิงอนุรักษ์
The Development of Local Weather Station for Ecotourism

ธนาธร เจ๊ะดาโห๊ะ
Thanathorn Jeadahoa

สัญญากรณ์ ปัทมรักษ์
Sanpakorn Pattamarak

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตร์บัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล
มหาวิทยาลัยวลัยลักษณ์
2565

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Information Technology and Digital Innovation
Walailak University

2022

ใบรับรองโครงการ

สำนักวิชาสารสนเทศศาสตร์ มหาวิทยาลัยวลัยลักษณ์

หัวข้อโครงการ การพัฒนาสถานีอากาศท้องถิ่นเพื่อการท่องเที่ยวเชิงอนุรักษ์

จัดทำโดย นายธนธร เจ๊ะดาโห๊ะ รหัสนักศึกษา 62103973

นายสัญญาภรณ์ ปัทมรักษ์ รหัสนักศึกษา 62109905

สาขาวิชา เทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล

อาจารย์ที่ปรึกษา ดร. ชนันกรณ จันทร์แดง

ปีการศึกษา 2565

บัณฑิตวิทยาลัย มหาวิทยาลัยวลัยลักษณ์ ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรวิทยาศาสตรมหาบัณฑิต

..... อาจารย์ที่ปรึกษาโครงการ

(..ดร. ชนันกรณ จันทร์แดง..)

พิจารณาเห็นชอบโดย

..... ประธานกรรมการ

..... กรรมการ

(.ผู้ช่วยศาสตราจารย์ ดร. ยรรยงค์ พันธุ์สวัสดิ์..)

(.อาจารย์ อธิษฐ์ สายชู..)

หัวข้อโครงการ	การพัฒนาสถานีอากาศท้องถิ่นเพื่อการท่องเที่ยวเชิงอนุรักษ์
ผู้เขียน	นายธนาธร เจ๊ะดาให้ะ นายสัญญากรณ์ ปัทมรักษ์
สาขาวิชา	เทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล
ปีการศึกษา	2565

บทคัดย่อ

งานวิจัยนี้ถูกจัดทำขึ้นเพื่อการพัฒนาสถานีอากาศสำหรับการท่องเที่ยวเชิงนิเวศที่สามารถตรวจสอบข้อมูลสภาพอากาศผ่านทางแพลตฟอร์มโปรแกรมไลน์โดยการพัฒนาสถานีอากาศเป็นการนำอุปกรณ์บอร์ด Micro Controller และอุปกรณ์เซ็นเซอร์ต่าง ๆ เช่น เซนเซอร์วัดอุณหภูมิ ความชื้น ปริมาณน้ำฝน ค่าฝุ่น PM2.5 ค่าความเร็วลม และภาพถ่ายบริเวณโดยรอบสถานที่ท่องเที่ยว ข้อมูลที่ได้จะส่งผ่านอุปกรณ์ Network module ไปเก็บยังฐานข้อมูล Thing speak ซึ่งข้อมูลที่ได้รับจะนำไปแสดงผลยังแพลตฟอร์มโปรแกรมไลน์ และผู้ใช้งานระบบสถานีอากาศ คือ นักท่องเที่ยวที่ต้องการทราบสภาพอากาศก่อนการเดินทาง โดยผลลัพธ์ที่ได้คือ สถานีอากาศที่ตรวจสอบข้อมูลสภาพอากาศได้อย่างมีประสิทธิภาพและแสดงผลข้อมูลสภาพอากาศได้อย่างแม่นยำ

คำสำคัญ : สถานีอากาศ, การท่องเที่ยวเชิงนิเวศ, ไลน์บอท, แพลตฟอร์ม, อินเทอร์เน็ตของสรรพสิ่ง

Title	The Development of Local Weather Station for Ecotourism
Author	Thanathorn Jeadahoa Sanpakorn Pattamarak
Major Program	Information Technology and Digital Innovation
Year	2022

Abstract

This research is to design and implement a weather station. The station can monitor and collect weather data using micro controller and other sensor device to measure temperature, humidity, rainfall data, pm2.5, wind speed and picture. The data will be sent through the network module to ThingSpeak database. The received data will be displayed to the chatbots on the LINE program. The users of the weather station system are tourists who want to know the weather conditions before traveling. The result is a weather station that effectively monitors weather information and displays accurate weather information.

Keyword : Weather Station, Ecotourism, Line Bots, Chatbots, Internet of Things

กิตติกรรมประกาศ

โครงการสำหรับการพัฒนาสถานีอากาศท้องถิ่นเพื่อการท่องเที่ยวเชิงอนุรักษ์ฉบับนี้สำเร็จลุล่วงได้ด้วย
ความช่วยเหลือและสนับสนุนจากผู้ทรงคุณวุฒิหลายท่านที่ช่วยให้คำแนะนำให้บรรลุตามขอบเขตและเป้าหมาย
งานที่วางไว้ ผู้จัดทำโครงการจึงใคร่ขอขอบพระคุณผู้ทรงคุณวุฒิทุกท่านไว้ ณ โอกาสนี้

อาจารย์ ดร. ชนันทกรณ จันแดง อาจารย์ที่ปรึกษาโครงการที่ได้ให้ความกรุณาเป็นอย่างสูงในการแนะนำ
ให้ความรู้ในการทำโครงการ การตรวจทาน การแก้ไขปัญหาและการแก้ไขของการเขียนโครงการจนสำเร็จลุล่วง
ด้วยดี ตลอดจนความห่วงใยและกำลังใจให้กับคณะผู้จัดทำเสมอมา

สุดท้ายนี้ คณะผู้จัดทำโครงการขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับ
การศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจกับคณะผู้จัดทำเสมอมา

นายธนาร ใจะดาโห๊ะ

นายสัญญากรณ์ ปัทมรักษ์

สารบัญ

เรื่อง	หน้า
บทคัดย่อ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ	จ
บทที่ 1	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของงาน	2
1.3.1 กลุ่มของผู้ใช้ระบบ.....	2
1.3.2 ฟังก์ชันงาน.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 ผลที่จะได้เมื่อเสร็จสิ้นโครงการ.....	3
1.6 แผนการดำเนินงาน.....	3
1.6.1 ศึกษาทฤษฎีพื้นฐานที่เกี่ยวข้อง	3
1.6.2 ทดสอบแนวคิดเบื้องต้น (proof of concept)	3
1.6.3 จัดทำรายงาน	4
1.7 เครื่องมือที่ใช้ในการพัฒนา	4
1.7.1 ฮาร์ดแวร์สำหรับการพัฒนาระบบและทดลอง.....	4
1.7.2 อุปกรณ์สำหรับสถานีอากาศ.....	4
1.7.3 ซอฟต์แวร์สำหรับการพัฒนาระบบ.....	5
บทที่ 2	6
2.1 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 การท่องเที่ยวเชิงนิเวศ.....	6
2.1.2 สถานีอากาศ.....	6
2.1.3 Internet of Things (IoT).....	8
2.1.4 Chatbot.....	8

สารบัญ (ต่อ)

เรื่อง	หน้า
2.1.5 Power Consumption	9
2.2 เทคโนโลยีที่ใช้.....	11
2.2.1 ซอฟต์แวร์.....	11
1. Arduino IDE	11
2. C Programming Language	13
3. PHP	14
2.2.2 อุปกรณ์สำหรับสถานีอากาศ	15
2.2.3 ระบบคลาวด์สาธารณะ	20
2.3 งานวิจัยหรือระบบที่ใกล้เคียง	22
2.4 ผลการศึกษาเบื้องต้น	23
2.4.1 การนำค่าจากเซนเซอร์ขึ้น ThingSpeak	23
2.4.3 Line Bot ตอบโต้ข้อความ	25
2.4.4 Line Bot ดึงค่าจาก ThingSpeak.....	26
2.4.5 Line Bot ดึงค่าจาก Firebase	26
บทที่ 3	28
3.1 องค์กรที่เกี่ยวข้อง	28
3.1.1 กลุ่มของผู้ใช้งาน.....	28
3.1.2 ข้อมูลเบื้องต้นขององค์กร หรือ หน่วยงานที่เกี่ยวข้อง.....	28
3.2 วิธีการรวบรวมข้อมูล	28
3.2.1 การค้นหาจากเอกสาร.....	28
3.2.2 การเก็บข้อมูลโดยแบบสอบถาม.....	30
3.3 การทำงานของระบบปัจจุบัน.....	37
3.3.1 กระบวนการทำงานของระบบปัจจุบัน	37
3.4 การวิเคราะห์ความต้องการของผู้ใช้	39
3.4.1 User requirement	39
3.4.2 Functional requirement.....	39
3.4.3 Non-Functional requirement.....	39

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4	40
4.1 การออกแบบสถาปัตยกรรมระบบ	40
4.1.1 NodeMCU ESP8266	40
4.1.2 Cloud Storage.....	41
4.1.3 Webhook	41
4.1.4 LineAPI	41
4.2 การออกแบบสถาปัตยกรรมซอฟต์แวร์	41
4.2.1 โมดูล Cloud storage	42
4.2.2 โมดูล Webhook.....	43
4.2.4 โมดูลแอปพลิเคชันไลน์	43
4.3 การออกแบบโครงสร้างข้อมูล	44
4.3.1 ตัวอย่างข้อมูลของเซ็นเซอร์รูปแบบ JSON ของ Thing speak.....	44
4.3.1 Data Dictionary.....	44
4.4 วางแผนการทดสอบ.....	45
4.4.1 Unit Test	45
4.5 การออกแบบส่วนการใช้งาน (INPUT DESIGN).....	47
4.6 การออกแบบส่วนแสดงผล (OUTPUT DESIGN).....	48
บทที่ 5	52
5.1 การพัฒนาระบบ	52
5.1.1 อธิบายไลบรารีที่ใช้ในการพัฒนาระบบ	52
5.2 การติดตั้งระบบ.....	54
5.2.1 โมดูลสถานีอากาศ.....	54
5.2.2 โมดูล Webhook	55
5.2.3 โมดูล Line Bot.....	57
5.2.4 โมดูลการเก็บข้อมูลจากเซ็นเซอร์	59
5.3 การทดสอบระบบ	62

สารบัญ (ต่อ)

เรื่อง	หน้า
5.3.1 Unit Test	62
บทที่ 6	65
6.1 สรุปผล	65
6.2 ปัญหาในการดำเนินงาน	65
6.3 ข้อเสนอแนะ	65
บรรณานุกรม	66
ประวัติผู้จัดทำ	68

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินการโครงการ.....	4
2.1 งานวิจัยหรือระบบที่เกี่ยวข้อง.....	22
4.1 DATA DICTIONARY.....	44
4.2 การทดสอบการทำงานของชุดคำสั่งข้อความ.....	45
4.3 การทดสอบการทำงานของ CLOUD STORAGE.....	45
4.4 การทดสอบการทำงานของ CLOUD IMAGE STORAGE.....	45
4.5 การทดสอบการใช้พลังงานสะสม.....	46
4.6 การออกแบบส่วนการใช้งาน.....	47
5.1 ไลบรารีที่ใช้ในการพัฒนา.....	52
5.2 การทดสอบการทำงานของชุดคำสั่งข้อความ.....	62
5.3 การทดสอบการทำงานของ CLOUD STORAGE.....	63
5.4 การทดสอบการทำงานของ CLOUD IMAGE STORAGE.....	63
5.5 การทดสอบการใช้พลังงานของระบบ.....	63

สารบัญภาพ

ภาพที่	หน้า
2.1 การทำงานของเซตบอท.....	9
2.2 ตัวอย่างการเชื่อมต่อ D0 และ RST	11
2.3 โปรแกรม BLINK.....	12
2.4 โปรแกรมการอ่านอุณหภูมิและค่าความชื้นของอากาศ.....	13
2.5 NODEMCU ESP8266	16
2.6 NODEMCU GPIO PINOUT	17
2.7 แอร์การ์ด	17
2.8 โมดูลกล้อง ESP32 CAM	18
2.9 เซนเซอร์ DHT22	19
2.10 เซนเซอร์วัดความเร็วลม ทิศทางลมและปริมาณน้ำฝน	19
2.11 PMS5003 SENSOR MODULE AIR PARTICLE DUST.....	20
2.12 หน้าจอ THINGSPEAK เมื่อดึงค่าจากเซนเซอร์ได้.....	25
2.13 หน้าข้อความตอบโต้ระหว่างผู้ใช้งานกับ LINE BOT	26
3.1 เพศ	30
3.2 อายุ	30
3.3 การเช็คสภาพอากาศก่อนออกเดินทาง	31
3.4 สภาพอากาศมีผลต่อการตัดสินใจในการไปหรือยกเลิกแผนหรือไม่	31

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.5 คุณตรวจสอบสภาพอากาศล่วงหน้ากี่วัน.....	32
3.6 ช่องทางตรวจสอบสภาพอากาศคือช่องทางใด เป็นหลัก.....	32
3.7 ข้อมูลที่ผู้ต้องการทราบก่อนเดินทาง.....	33
3.8 ผู้ใช้งานตรวจสอบสภาพอากาศผ่านช่องทาง	33
3.9 จำนวนของนักท่องเที่ยวในพื้นที่มีผลต่อการตัดสินใจไปกางเต็นท์หรือไม่	34
3.10 ภาพถ่ายของบรรยากาศที่นักท่องเที่ยวต้องการเห็น.....	34
3.11 หากผู้ใช้งานทราบสภาพอากาศล่วงหน้า จะส่งผลต่อการตัดสินใจไปกางเต็นท์หรือไม่.....	35
3.12 หากสามารถตรวจสอบสภาพอากาศได้ผ่านแชทบอทของโปรแกรมไลน์ ท่านอยากเห็นการนำเสนอข้อมูลในรูปแบบใด.....	35
3.13 สถานีอากาศที่จะพัฒนานี้ ควรแสดงข้อมูลใดๆ ได้บ้างในโปรแกรม LINE	36
3.14 ถ้ามีสถานีอากาศเฉพาะพื้นที่สำหรับพื้นที่กางเต็นท์ คุณคาดหวังสิ่งใดบ้าง	36
3.15 การทำงานของระบบปัจจุบัน.....	37
3.16 ปัญหาของระบบปัจจุบัน.....	38
4.1 สถาปัตยกรรมระบบ	40
4.2 สถาปัตยกรรมระบบ	41
4.3 โมดูล CLOUD STORAGE.....	42
4.4 โมดูล WEBHOOK.....	43
4.5 โมดูลแอปพลิเคชันไลน์	43

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.6 หน้าจอผลการทดสอบคำสั่งขอข้อมูลคำสั่ง.....	47
4.7 หน้าจอแสดงผลข้อมูลอุณหภูมิ	48
4.8 หน้าจอแสดงผลข้อมูลความชื้นสัมพัทธ์.....	48
4.9 หน้าจอแสดงผลข้อมูลคุณภาพอากาศ	49
4.10 หน้าจอแสดงผลข้อมูลฝน.....	49
4.11 หน้าจอแสดงผลภาพรวมของระบบ	50
5. 1 ภาพรวมสถานีอากาศ.....	54
5. 2 การเชื่อมต่อภายในสถานีอากาศ.....	54
5. 3 WIRING DIAGRAM	55
5. 4 สร้างแอปพลิเคชันใหม่.....	55
5. 5 การสร้าง GITHUB และสร้างไฟล์ PHP.....	56
5. 6 การเชื่อมต่อระหว่าง GITHUB และ HEROKU	57
5. 7 ไฟล์ PHP	57
5. 8 เลือกการสร้างบัญชีแบบ MESSAGING API	57
5. 9 กรอกรายละเอียดของ PROVIDER.....	58
5. 10 ตั้งค่า LINE DEVELOPER ให้เชื่อมต่อกับ WEBHOOK ของ HEROKU	58
5. 11 นำ ACCESS TOKEN ของ LINE DEVELOPER ที่ได้ไปไว้ในไฟล์ INDEX.PHP	58

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
5. 12 เว็บไซต์ THING SPEAK.....	59
5. 13 กรอกข้อมูลเพื่อสมัครสมาชิก.....	59
5. 14 กรอก USERNAME และ PASSWORD.....	60
5. 15 สร้าง CHANNELS เพื่อรองรับข้อมูล.....	60
5. 16 หน้าจอ NEW CHANNELS.....	61
5. 17 หน้าจอ CHANNELS	61
5. 18 หน้าจอ API KEYS	62
5. 19 ข้อมูลการเปรียบเทียบการใช้พลังงานสะสม 1 วัน.....	64

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันคนไทยหลายคนจัดอันดับการท่องเที่ยวไว้เป็นเป้าหมายที่มีความสำคัญอันดับต้น ๆ นอกเหนือจากการซื้อบ้านและรถยนต์ ความคิดที่จะท่องเที่ยวเปลี่ยนไปจากสมัยก่อนที่เที่ยว เมื่อพร้อมหรือเมื่อมีเงินเหลือเท่านั้น แต่ในปัจจุบันหลายคนเลือกที่จะเก็บเงินมาเพื่อเที่ยวเป็นเรื่องแรก ซึ่งการกางเต็นท์นับเป็นกิจกรรมหนึ่งในการท่องเที่ยวเชิงนิเวศเพราะเปรียบเสมือนการเปิดมุมมองโลกทัศน์ใหม่ ที่ช่วยในการสร้างแรงบันดาลใจและประสบการณ์ให้นักท่องเที่ยวแต่ละคน (กรัณฐรัตน์ เทียนทับทิม, 2561) โดยปี พ.ศ. 2550 อุทยานแห่งชาติเขาใหญ่มีนักท่องเที่ยว 871,268 คน ในจำนวนนี้มีนักท่องเที่ยวพักค้างคืน 130,758 คน และกิจกรรมหลักที่นักท่องเที่ยวให้ความสนใจ ได้แก่ การเดินป่าศึกษาธรรมชาติ เล่นน้ำตก ดูนก ส่องสัตว์และกางเต็นท์พักแรม (กรมอุทยานแห่งชาติสัตว์ป่าและพันธุ์พืช, 2551) ปัจจัยที่มีผลกระทบต่อการท่องเที่ยวเชิงนิเวศ คือ สภาพอากาศโดยปกติประเทศไทยจะมีฝนตกชุกในช่วงปลายปีและอาจมีพายุฤดูร้อนในช่วงต้นปี แม้ว่าทางกรมอุตุนิยมวิทยาสามารถทำนายสภาพอากาศล่วงหน้า (เบญจวรรณ เจ้ยทองศรี และ ธนวัฒน์ จารุพงษ์สกุล, 2557) งานวิจัยพบว่าสถานีดัดตั้งแบบจำเพาะในแต่ละพื้นที่พบได้น้อย และไม่สามารถแสดงผลข้อมูลที่เป็นปัจจุบันได้

นอกจากนี้สภาพอากาศยังมีตัวแปรอื่นที่ทำให้นักท่องเที่ยวต้องคำนึงถึงบรรยากาศโดยรวม ในปัจจุบันความชุกของนักท่องเที่ยวในพื้นที่และความปลอดภัยของสถานที่รวมถึงสิ่งอำนวยความสะดวกต่าง ๆ ทั้งในสภาพแวดล้อมและรายละเอียดยังส่งผลการตัดสินใจของนักท่องเที่ยวที่จะเข้ามาเยี่ยมชมสถานที่ ดังนั้นจึงจำเป็นต้องมีระบบที่สามารถช่วยสนับสนุนการตัดสินใจของนักท่องเที่ยวในการเข้าถึงข้อมูลของสถานที่ท่องเที่ยว นั้น ๆ โดยพิจารณาจากสภาพอากาศจริงและภาพถ่ายของพื้นที่โดยรอบของสถานที่ท่องเที่ยว ด้วยอุปกรณ์เคลื่อนที่ผ่านระบบเครือข่ายอินเทอร์เน็ต

ปัจจุบันมีสถานีอากาศที่ผลิตและจำหน่าย เช่น 7911 Anemometer for Weather monitor for wizard, AW006 Jedto , Vintage Pro2 Davis Weather station และ Datalogger Temlog 20 เป็นต้น โดยอุปกรณ์ข้างต้น สามารถตรวจจับและบันทึกสภาพอากาศต่าง ๆ ได้แก่ อุณหภูมิ ความชื้น แสงลม ปริมาณน้ำฝน เวลาดวงอาทิตย์ขึ้นและตก แต่อุปกรณ์เหล่านี้ผู้ใช้ทั่วไปอาจจะสามารถเข้าถึงได้ยาก นอกจากนี้ยังมีของบริษัทต่าง ๆ ที่ทำการสังเกตสภาพอากาศในพื้นที่บริเวณนั้น ๆ ดังนั้นข้อจำกัดของอุปกรณ์เหล่านี้คือ ผู้ใช้จะต้องมีสิทธิในการเข้าถึงระบบ และไม่สามารถเผยแพร่แบบสาธารณะได้

เนื่องด้วยงานวิจัยสถานีวิัดสภาพอากาศส่วนบุคคลสำหรับอุตสาหกรรมการท่องเที่ยวเป็นระบบการตรวจสอบสภาพอากาศที่มีปัญหาในส่วนของการให้บริการอินเทอร์เน็ต เช่น พื้นที่อุทยาน เป็นต้น (อินันท์และ

คณะ, 2560) โครงการนี้ผู้จัดทำจึงเสนอระบบที่สนับสนุนการท่องเที่ยวเชิงนิเวศโดยสามารถเข้าถึงในพื้นที่ ที่พลังงานไฟฟ้าไม่สามารถเข้าถึงได้ และผู้ใช้งานสามารถเข้าถึงข้อมูลได้ด้วยอุปกรณ์เคลื่อนที่ผ่านโปรแกรมไลน์

โครงการเล่มนี้จึงเสนอสถานีวิจัยอากาศสำหรับการท่องเที่ยวเชิงนิเวศ โดยเน้นไปยังสถานที่ ที่พลังงานไฟฟ้าไม่สามารถเข้าถึงได้ โดยฟังก์ชันการทำงานของระบบ คือ ผู้ใช้งานสามารถตรวจสอบ อุณหภูมิ ความชื้นสัมพัทธ์ คุณภาพอากาศ ฝน ลม และภาพถ่ายบริเวณโดยรอบผ่านโปรแกรมไลน์บนอุปกรณ์เคลื่อนที่

1.2 วัตถุประสงค์

1. เพื่อพัฒนาระบบสถานีวิจัยอากาศเพื่อสนับสนุนการท่องเที่ยวเชิงนิเวศ
2. เพื่อแสดงสภาพอากาศและภาพถ่ายในพื้นที่ที่สนใจผ่านโปรแกรมไลน์

1.3 ขอบเขตของงาน

1.3.1 กลุ่มของผู้ใช้ระบบ

1.3.1.1 นักท่องเที่ยว

นักท่องเที่ยวจะทราบถึงสภาพอากาศในพื้นที่ที่สนใจ โดยสามารถเข้าถึงระบบได้ผ่านโปรแกรมไลน์ที่รองรับทั้งคอมพิวเตอร์และอุปกรณ์เคลื่อนที่

1.3.1.2 เจ้าของสถานที่

เจ้าของสถานที่ในหน่วยงานรัฐและเอกชนที่สนใจและสามารถเข้าถึงระบบได้ผ่านโปรแกรมไลน์ที่รองรับทั้งคอมพิวเตอร์และอุปกรณ์เคลื่อนที่

1.3.2 ฟังก์ชันงาน

1.3.2.1 สถานีวิจัยอากาศ

1. เก็บข้อมูลสภาพอากาศ
2. แสดงข้อมูลจากสภาพแวดล้อมที่ติดตั้งสถานีวิจัยอากาศไว้ ได้แก่ อุณหภูมิ ความชื้นสัมพัทธ์ ฝน ลม และคุณภาพอากาศ
3. ถ่ายภาพของสถานที่ที่ติดตั้งสถานีวิจัยอากาศไปแสดงผ่านโปรแกรมไลน์
4. นำข้อมูลที่ได้จากเซนเซอร์ไปแสดงผ่านโปรแกรมไลน์

1.3.2.2 โปรแกรมไลน์สำหรับเข้าถึงข้อมูล

1. ผู้ใช้งานสามารถเข้าถึงข้อมูลสภาพอากาศหรือภาพบรรยากาศโดยรอบของสถานีวิจัยอากาศผ่านโปรแกรมไลน์
2. แสดงข้อมูลรูปแบบของห้องสนทนาผ่านโปรแกรมไลน์โดยผู้ใช้งานสามารถเข้าถึงผ่านโปรแกรมไลน์ที่รองรับทั้งคอมพิวเตอร์และอุปกรณ์เคลื่อนที่

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. นักท่องเที่ยวได้ทราบสภาพอากาศในพื้นที่ที่สนใจผ่านโปรแกรมไลน์
2. เจ้าของสถานที่ได้กลุ่มลูกค้าเพิ่มขึ้นอีกทั้งมีช่องทางการสื่อสารกับนักท่องเที่ยว

1.5 ผลที่จะได้เมื่อเสร็จสิ้นโครงการ

1. สถานีอากาศ
2. แชนบอทบนโปรแกรมไลน์

1.6 แผนการดำเนินงาน

โครงการชื่อ สถานีอากาศสำหรับการท่องเที่ยวเชิงนิเวศ มีขั้นตอนการดำเนินงาน ดังนี้

1.6.1 ศึกษาทฤษฎีพื้นฐานที่เกี่ยวข้อง

1.6.1.1 การท่องเที่ยวเชิงนิเวศ

เพื่อทำความเข้าใจและศึกษาความเป็นมาของการท่องเที่ยวเชิงนิเวศ ซึ่งจะเน้นเกี่ยวกับกิจกรรมทางเดินทัพักแรมโดยศึกษาว่ามีปัจจัยใดบ้างที่ส่งผลต่อการทำกิจกรรม และปัจจัยใดบ้างที่ทำให้ประชาชนสนใจเกี่ยวกับกิจกรรมนี้

1.6.1.2 เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งและเซนเซอร์

เพื่อนำองค์ความรู้และทฤษฎีพื้นฐานที่เกี่ยวข้องมาพัฒนาระบบสถานีอากาศ

1.6.1.3 ฐานข้อมูลและบริการบนคลาวด์

เพื่อศึกษาเกี่ยวกับระบบ Cloud Storage สำหรับพัฒนา Realtime Application และนำองค์ความรู้มาใช้เกี่ยวกับระบบสถานีอากาศ

1.6.1.4 การสร้างแชทบอทสำหรับโปรแกรมไลน์

เพื่อช่วยในการพูดคุยโต้ตอบ โดยสามารถออกแบบข้อความการโต้ตอบกับผู้ใช้ได้

1.6.2 ทดสอบแนวคิดเบื้องต้น (proof of concept)

1.6.2.1 เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งและเซนเซอร์

ทดสอบการเขียนโปรแกรมบน Arduino IDE เพื่อใช้ในการควบคุมการทำงานของเซนเซอร์และนำข้อมูลที่ได้จากเซนเซอร์ส่งผ่านอุปกรณ์เน็ตเวิร์คโมดูล

1.6.2.2 ฐานข้อมูลและบริการบนคลาวด์

ทดลองนำข้อมูลที่เก็บได้จากเซนเซอร์ส่งผ่านมายังเน็ตเวิร์คโมดูล เพื่อนำข้อมูลมาเก็บและแสดงผลผ่านแชทบอทบนโปรแกรมไลน์

1.6.2.3 การสร้างแชทบอทสำหรับโปรแกรมไลน์

แพลตฟอร์มโปรแกรมไลน์เป็นโปรแกรมที่สามารถตอบโต้กับผู้ใช้ได้อย่างอัตโนมัติ จึงมีความเหมาะสมที่จะนำมาใช้ประโยชน์สำหรับบริการด้านข้อมูลสภาพอากาศ

1.6.3 จัดทำรายงาน

ตารางที่ 1. 1 แผนการดำเนินการโครงการ

กิจกรรม	สัปดาห์ที่											
	1	2	3	4	5	6	7	8	9	10	11	12
1.1 การท่องเที่ยวเชิงนิเวศ												
1.2 เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งและเซนเซอร์												
1.3 ฐานข้อมูลและบริการบนคลาวด์												
1.4 การสร้างแพลตฟอร์มสำหรับโปรแกรมไลน์												
2.1 PoC เรื่อง เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งและเซนเซอร์												
2.2 PoC เรื่อง ฐานข้อมูลและบริการบนคลาวด์												
2.3 PoC เรื่อง การสร้างแพลตฟอร์มสำหรับโปรแกรมไลน์												
3.1 จัดทำรายงานบทที่ 1												
3.2 จัดทำรายงานบทที่ 2												

1.7 เครื่องมือที่ใช้ในการพัฒนา

1.7.1 ฮาร์ดแวร์สำหรับการพัฒนาระบบและทดลอง

1.7.1.1 เครื่องคอมพิวเตอร์

1. Processor : AMD Ryzen 3 3.0 GHz or better
2. RAM : 8 GB
3. Hard disk : SSD 128 GB

1.7.1.2 อุปกรณ์เคลื่อนที่ Huawei P30 Lite Android Version 10

1.7.2 อุปกรณ์สำหรับสถานีอากาศ

1.7.2.1 NodeMCU ESP8266

1.7.2.3 แอร์การ์ด

1.7.2.4 แหล่งพลังงาน

1.7.2.5 โมดูลกล้อง ESP32 Camera

1.7.2.6 เซนเซอร์วัดค่า

1. วัดค่าอุณหภูมิและความชื้น DHT22
2. ตรวจจับฝุ่นละออง รุ่น PM2.5 PMS5003 + Adapter Pin
3. เซนเซอร์ตรวจจับปริมาณน้ำฝน
4. เซนเซอร์ตรวจจับความเร็วลมและทิศทางลม

1.7.3 ซอฟต์แวร์สำหรับการพัฒนาระบบ

1.7.3.1 ฐานข้อมูล : ThingSpeak , Firebase

1.7.3.2 Library : Adafruit

1.7.3.3 Visual studio Code: ใช้สำหรับเขียนโปรแกรมเชื่อมต่อการทำงานของ Chat bot และ เซนเซอร์

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

การจัดทำโครงการพัฒนาสถานีอากาศท้องถิ่นเพื่อการท่องเที่ยวเชิงนิเวศโดยอาศัยหลักทฤษฎีและการนำเทคโนโลยีต่าง ๆ มาประกอบเข้าด้วยกันเพื่อให้ได้มาซึ่งสถานีอากาศที่สามารถใช้งานได้ตามวัตถุประสงค์ ในบทนี้จะกล่าวถึงเทคโนโลยีที่นำมาใช้ในการพัฒนารายละเอียดที่เกี่ยวข้องดังนี้

2.1.1 การท่องเที่ยวเชิงนิเวศ

การท่องเที่ยวเชิงนิเวศมีลักษณะที่สำคัญคือ เป็นการท่องเที่ยวที่ดำเนินการภายใต้ขีดจำกัดความสามารถของธรรมชาติ และต้องตระหนักถึงการมีส่วนร่วมของประชากร ชุมชน ประเพณีที่มีต่อการท่องเที่ยว อีกทั้งต้องยอมรับให้ประชาชนทุกส่วนได้รับผลประโยชน์ทางเศรษฐกิจที่เกิดจากการท่องเที่ยวอย่างเสมอภาคเท่าเทียมกัน และต้องเป็นความปรารถนาของประชาชนท้องถิ่นและชุมชนในพื้นที่ท่องเที่ยวนั้น ๆ ซึ่งควรจะเน้นการพัฒนาและการจัดการเฉพาะในแหล่งธรรมชาติซึ่งถือกันว่าเป็น รูปแบบการท่องเที่ยวเชิงนิเวศ เพื่อสนองตลาดการท่องเที่ยวเฉพาะ โดยเฉพาะอย่างยิ่งกลุ่มตลาด นักท่องเที่ยวที่แสวงหาความรู้และอยากเข้าไปมีส่วนร่วมในการอนุรักษ์ทรัพยากรธรรมชาติและระบบ นิเวศของแหล่งท่องเที่ยวควบคู่ไปกับการเพลิดเพลินและความสัมพันธ์กับคนท้องถิ่น รวมทั้งให้ ประโยชน์กลับคืนไปสู่แหล่งท่องเที่ยวและชุมชนท้องถิ่น (ภาณุภา สูดใจ, 2558)

2.1.2 สถานีอากาศ

ในปัจจุบันเครื่องวัดสภาพอากาศถูกนำมาใช้งานนอกเหนือจากการใช้เพื่อพยากรณ์อากาศ ยังนำมาใช้ในงานด้านอื่น ๆ เช่น โรงงานอุตสาหกรรม เกษตรกรรม จากข้อมูลอุตุนิยมวิทยาที่ตรวจวัดเพื่อพยากรณ์อากาศตัวแปรที่ส่งผลโดยตรงกับลักษณะอากาศที่เกิดขึ้นในแต่ละวัน สถานีอากาศของกรมอุตุนิยมวิทยาแต่ละแห่งจะต้องทำการติดตั้งอุปกรณ์ทำการตรวจวัด ดังนี้

1. Barometer เป็นเครื่องมือตรวจวัดความดันบรรยากาศ สำหรับวัดค่าความกดดันที่เกิดจากแรงดันของอากาศ โดยใช้ของเหลวหรือวัสดุที่สัมผัสโดยตรงกับ

อากาศ การเปลี่ยนแปลงของความกดดัน สามารถนำไปพยากรณ์การเปลี่ยนแปลงสภาพอากาศในช่วงเวลานั้นได้

2. Thermometer เป็นเครื่องมือสำหรับวัดอุณหภูมิอากาศ ที่ใช้กันในงานตรวจสอบสภาพอากาศ คือ เครื่องวัดเทอร์โมมิเตอร์ชนิดสูงสุดและต่ำสุด (Max-min Thermometer) เทอร์โมมิเตอร์สูงสุด (maximum thermometer) คือเทอร์โมมิเตอร์ชนิดปรอทบรรจุอยู่ในหลอดแก้ว เช่นเดียวกับเทอร์โมมิเตอร์ธรรมดา แต่แตกต่างกันตรงที่ว่าบริเวณลำเทอร์โมมิเตอร์เหนือกระเปาะบรรจุปรอทขึ้นมาเล็กน้อยจะเป็นคอคอดป้องกันปรอทที่ขยายตัวแล้วไหลกลับลงกระเปาะ ใช้วัดอุณหภูมิสูงสุด เทอร์โมมิเตอร์ต่ำสุด (minimum thermometer) คือเทอร์โมมิเตอร์ชนิดเอทิลแอลกอฮอล์บรรจุในหลอดแก้ว มีก้านชี้รูปดัมเบลล์ ยาวประมาณ 2 เซนติเมตรบรรจุอยู่ ใช้วัดอุณหภูมิต่ำสุด
3. Hygrometer เป็นเครื่องมือที่ใช้ในการวัดปริมาณไอน้ำที่อยู่ในอากาศ สามารถวัดความชื้นในช่วง 0 ถึง 100% หน่วยเป็นความชื้นสัมพัทธ์ (Relative Humidity หรือ RH)
4. Wind vane เป็นเครื่องมือที่ใช้ตรวจทิศทางของลมมีลักษณะที่มีหางเป็นแผ่นใหญ่กว่าหัวลูกศร เมื่อลมพัดทางลูกศรจะถูกแรงลมปะทะมากกว่าหัวลูกศร ทำให้หัวลูกศรชี้ไปในทิศทางที่ลมพัดมา จึงทราบได้ว่าทิศทางลมพัดมาจากทิศทางไหน
5. Anemometer เป็นเครื่องมือที่ใช้วัดความเร็วลม ซึ่งตรวจสอบปริมาณการไหลของอากาศ มีหน่วยวัดเป็นเมตรต่อวินาที (m/s) หรือ กิโลเมตรต่อชั่วโมง (km/hr)
6. Rain gauge เป็นเครื่องมือวัดปริมาณน้ำฝน ใช้เพื่อเก็บและวัดปริมาณน้ำฝนในช่วงระยะเวลาในช่วงระยะเวลานั้นๆ ในพื้นที่ที่ต้องการสำรวจวัดค่าปริมาณน้ำฝน
7. กล่องส่งข้อมูล ทำหน้าที่ประมวลผลข้อมูลสภาพอากาศและส่งไปยังเครื่องปลายทาง
8. แหล่งพลังงาน ทำหน้าที่เก็บพลังงานและส่งพลังงานเพื่อใช้เลี้ยงวงจรและเซนเซอร์วัดค่าต่างๆ

เครื่องมือที่ใช้ในระบบการพยากรณ์อากาศสามารถแบ่งออกได้เป็นประเภทใหญ่ๆ ได้ดังนี้

เครื่องมือตรวจอากาศผิวพื้น

สถานีอุตุนิยมวิทยาแต่ละแห่งจะติดตั้งอุปกรณ์ตรวจวัดอากาศผิวพื้นตามที่กล่าวมาแล้วข้างต้น โดยจะทำการตรวจวัดอากาศตามเวลาที่กำหนดไว้ในแต่ละวัน ซึ่งจะมีเวลาหลักของการตรวจวัด คือ 07.00 น. (00.00 UTC) และเวลา 19.00 น. (12.00 UTC) โดยในระหว่างเวลาหลักเหล่านี้ อาจมีการตรวจวัดเพิ่มเติมได้ตามที่กำหนดไว้เพื่อความเหมาะสม

เครื่องมือตรวจอากาศชั้นบน

เนื่องจากการเปลี่ยนแปลงของลักษณะอากาศบนพื้นผิว มีความเกี่ยวพันกับการเปลี่ยนแปลงของบรรยากาศชั้นบนของโทรโพสเฟียร์ โดยใช้บอลลูนติดตั้งอุปกรณ์ตรวจอากาศซึ่งได้กล่าวมาแล้วในเบื้องต้น และข้อมูลซึ่งรายงานโดยนักบิน ได้แก่ ทิศนวิสัย อุณหภูมิ ทิศทางและความเร็วลม ปริมาณและชนิดเมฆ

เครื่องมือตรวจอากาศพิเศษ

เป็นเครื่องมือที่ใช้สำหรับตรวจวัดปรากฏการณ์หรือลักษณะอากาศที่เกิดขึ้นเพื่อช่วยเสริมในการวิเคราะห์พยากรณ์อากาศ ได้แก่ เรดาร์ตรวจอากาศ และดาวเทียมอุตุนิยมวิทยา
(ศูนย์การเรียนรู้วิทยาศาสตร์โลกและดาราศาสตร์)

2.1.3 Internet of Things (IoT)

อินเทอร์เน็ตของสรรพสิ่ง หมายถึง การนำอุปกรณ์ต่าง ๆ มาทำให้สามารถเชื่อมต่อเข้ากับอินเทอร์เน็ตและส่งข้อมูลถึงกันได้ การเชื่อมต่อระหว่างอุปกรณ์ทำให้สามารถสั่งการควบคุมอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ผ่านทางเครือข่ายอินเทอร์เน็ตและสามารถเก็บข้อมูลที่บันทึกได้ลงบนฐานข้อมูลแบบกลุ่มเมฆ ระบบอินเทอร์เน็ตของสรรพสิ่งสามารถประยุกต์เข้ากับการใช้งานได้ทุกประเภท ตัวอย่างเช่น ระบบสมาร์ทโฮม สมาร์ทฟาร์ม เมืองอัจฉริยะ

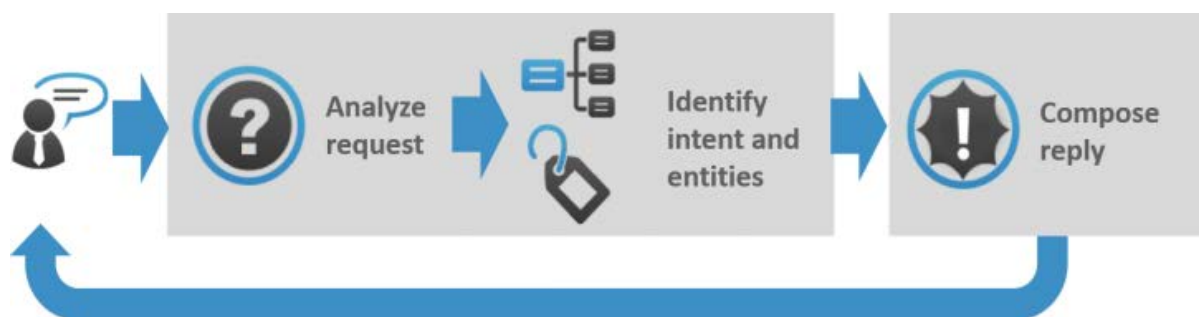
2.1.4 Chatbot

แชทบอทเป็นซอฟต์แวร์ที่พัฒนาขึ้นมาเพื่อช่วยตอบกลับการสนทนาผ่านข้อความหรือเสียงแบบอัตโนมัติและรวดเร็วซึ่งสามารถใช้งานได้ทั้งบน แอปพลิเคชันไลน์ แอปพลิเคชัน

ชั้นเฟซบุ๊ก และแอปพลิเคชันอินสตาแกรม และยังสามารถใช้งานผ่านเว็บไซต์ได้อีกด้วย (Kankann, 2563)

หลักการทำงานของแชทบอท

- วิเคราะห์คำถามของผู้ใช้งาน โดยทำการตรวจสอบหาคำหรือข้อความที่เหมือนหรือคล้ายกับคีย์เวิร์ดที่กำหนดไว้
- ตอบกลับผู้ใช้งาน เมื่อทำการตรวจสอบหาคำที่คล้ายกับคีย์เวิร์ด จากนั้นทำการตอบกลับผู้ใช้งานด้วยบทสนทนาที่เหมาะสมและรวดเร็ว



ภาพที่ 2. 1 การทำงานของแชทบอท

จากภาพที่ 2.1 อธิบายได้ว่าเมื่อผู้ใช้งานทำการส่งข้อความมายังแชทบอท ระบบนั้นจะทำการวิเคราะห์ความต้องการของผู้ใช้งาน หลังจากทีระบบวิเคราะห์ความต้องการเสร็จสิ้นระบบจะทำการส่งความต้องการไปตรวจสอบเพื่อหาคีย์เวิร์ดของคำตอบในการตอบกลับความต้องการไปยังผู้ใช้งาน จากนั้นทำการส่งคำตอบเพื่อตอบกลับไปยังผู้ใช้งานระบบ

2.1.5 Power Consumption

ในส่วนของหัวข้อนี้จะเป็นการคำนวณหาค่าการใช้พลังงานของสถานีอากาศที่ใช้ต่อหน่วยเวลา การใช้พลังงานมีความสำคัญโดยตรง เนื่องด้วยระบบสถานีอากาศไม่มีไฟฟ้าหลักที่สามารถจ่ายพลังงานให้แก่ระบบได้ตลอดเวลา จึงต้องพึ่งพาไฟฟ้าจากโมดูลแบตเตอรี่ที่ได้จากการชาร์จจากพลังงานแสงอาทิตย์ เวลาที่สถานีอากาศสามารถทำงานได้ขึ้นอยู่กับขนาดของแบตเตอรี่และกระแสไฟฟ้าพลังงานที่สถานีอากาศได้ใช้ (Current) ใช้สัญลักษณ์แทนกระแสไฟฟ้าคือ I มีหน่วยวัดคือ แอมแปร์ (Ampere)

กระแสไฟฟ้า มี 2 ชนิด

ไฟฟ้ากระแสตรง (Direct Current) ไฟฟ้าที่มีทิศทางการไหลของอิเล็กตรอนไปในทิศทางเดียวตลอด เช่น แบตเตอรี่

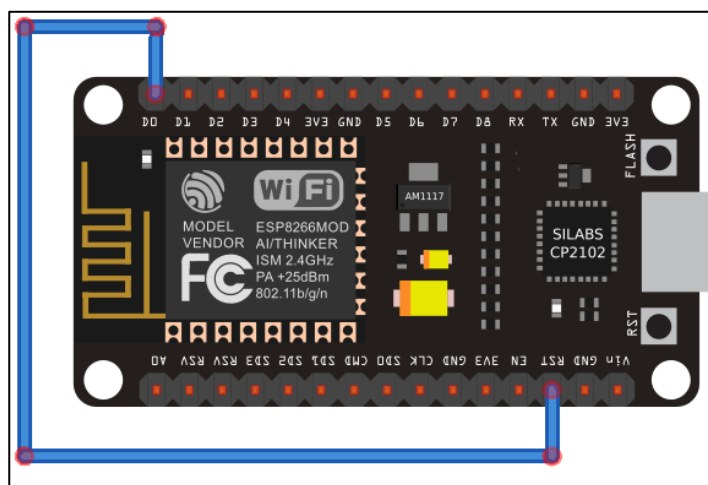
ไฟฟ้ากระแสสลับ (Alternation Current) ไฟฟ้าที่มีทิศทางการไหลของอิเล็กตรอนไปในทิศทางไปและกลับตลอดเวลา มีการสลับขั้วบวกและลบกับเป็นรูปคลื่นที่เรียกว่า Sine Wave

mAh (Milli ampere-hour) เป็นหน่วยวัดที่ใช้ในการอธิบายความจุของแบตเตอรี่ต่าง ๆ

ข้อจำกัดของระบบสถานีอากาศคือ พลังงานที่มีจำกัดและต้องหาวิธีลดการใช้พลังงานให้แก่วระบบสถานีอากาศ ซึ่ง NodeMCU ESP8266 มีฟังก์ชันการทำงานที่สามารถลดการใช้พลังงานเมื่อไม่มีการส่งข้อมูลหรืออ่านค่าจากเซ็นเซอร์ เรียกว่า sleep mode การหลับของ ESP8266 แบ่งระดับการหลับออกเป็น 3 ระดับ

- Modem Sleep โหมดนี้จะปิดเฉพาะการเชื่อมต่อไวไฟ แต่ในส่วนของ CPU, System clock, RTC ยังคงทำงานอยู่ โหมดนี้จะใช้กระแสประมาณ 15 mA ตัวอย่างการใช้งาน `WiFi.forceSleepBegin();` เพื่อเริ่มต้นการหลับ และ `WiFi.forceSleepWake();` เพื่อปลุกการทำงาน
- Light Sleep โหมดนี้จะปิดไวไฟ และ System clock และส่วนของ RTC ยังทำงานตามปกติ โหมดนี้จะใช้กระแสประมาณ 0.9 mA ตัวอย่างการใช้งาน โดยเพิ่มคำสั่ง `wifi_set_sleep_type(LIGHT_SLEEP_T)` ใน setup
- Deep Sleep โหมดนี้ ESP8266 จะสามารถประหยัดพลังงานได้มากที่สุด เพราะจะทำการปิด WiFi, System Clock และ CPU จะเปิด RTC ไว้เพื่อจับเวลาเพื่อเตือนให้ตัว ESP8266 ทำการตื่นมาทำงานอีกครั้ง โหมดนี้จะใช้กระแสประมาณ 20 μ A (0.02mA) ในโหมดนี้ ESP8266 ไม่สามารถเข้าสู่โหมด DeepSleep ได้อัตโนมัติ คำสั่งในเข้าสู่ `ESP.deepSleep(10e6)` ตัวเลขในวงเล็บคือ หน่วยไมโครวินาที (μ s) $10 \times 10^6 = 10000000$ ไมโครวินาที ใน

การใช้โหมดนี้จำเป็นต้องเชื่อมต่อระหว่าง D0 และ RST เข้าด้วยกัน เพื่อทำการปลุกให้ ESP8266 กลับมาทำงานอีกครั้ง



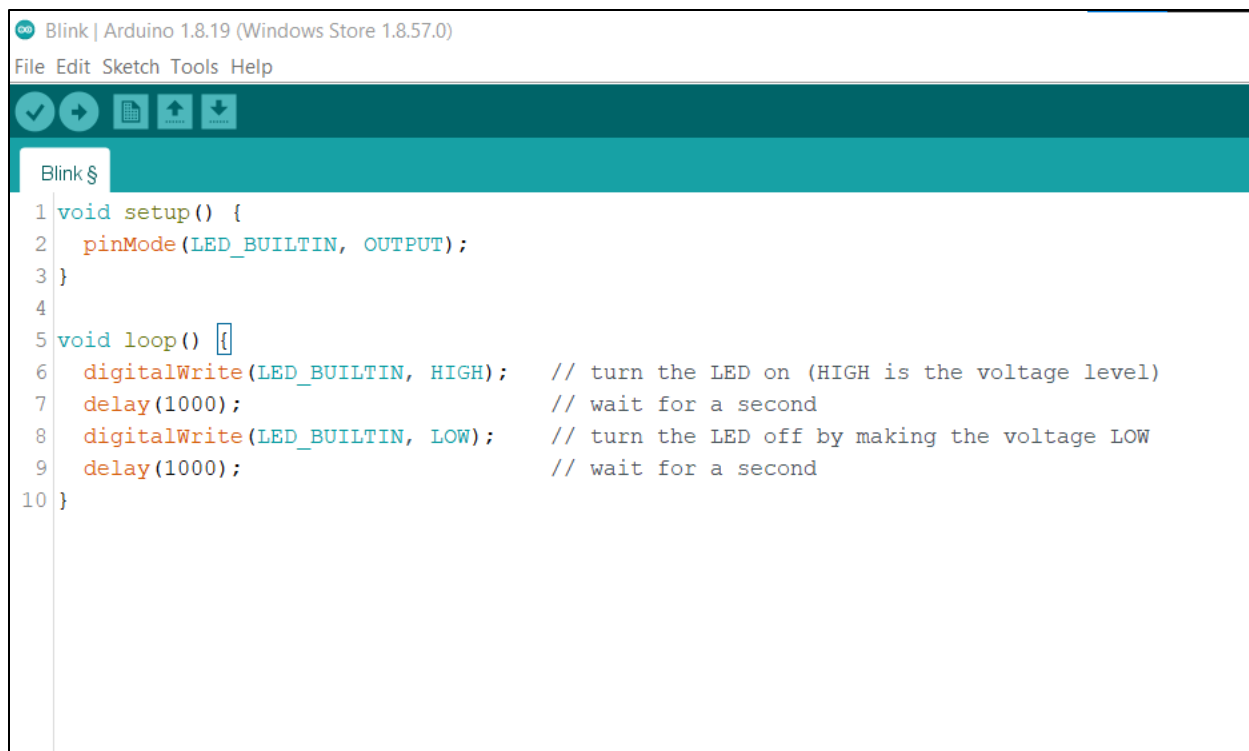
ภาพที่ 2. 2 ตัวอย่างการเชื่อมต่อ D0 และ RST

2.2 เทคโนโลยีที่ใช้

2.2.1 ซอฟต์แวร์

1. Arduino IDE

Arduino IDE เป็นซอฟต์แวร์ที่มีลักษณะเป็น Open source ที่มีหน้าที่เชื่อมต่อระหว่างคอมพิวเตอร์ที่รองรับระบบปฏิบัติการ Window, MacOS, Linux ซึ่งโปรแกรมถูกออกแบบมาเพื่อให้ง่ายต่อการเขียนโค้ดและอัปโหลดโปรแกรมเข้าสู่บอร์ด ส่วน IDE เป็นส่วนเสริมต่าง ๆ ที่คอยช่วยเหลือนักพัฒนาเพื่อให้เกิดความรวดเร็ว



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for opening, saving, and running the sketch. The code editor displays the following C++ code:

```

1 void setup() {
2   pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
7   delay(1000); // wait for a second
8   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
9   delay(1000); // wait for a second
10 }

```

ภาพที่ 2. 3 โปรแกรม Blink

บรรทัดที่ 1: ฟังก์ชัน setup เป็นฟังก์ชันพื้นฐานของโปรแกรม

บรรทัดที่ 2: ฟังก์ชัน pinMode ใช้กำหนดขาสัญญาณให้เป็นสัญญาณเข้าหรือสัญญาณออก

บรรทัดที่ 5: ฟังก์ชันวนลูป

บรรทัดที่ 6: ฟังก์ชันปล่อยสัญญาณดิจิทัล ไฟแอลอีดีที่อยู่บนบอร์ดให้ทำการเปิด

บรรทัดที่ 7: ดีเลย์ 1 วินาที

บรรทัดที่ 8: ฟังก์ชันปล่อยสัญญาณดิจิทัล ไฟแอลอีดีที่อยู่บนบอร์ดให้ทำการปิด

บรรทัดที่ 9: ดีเลย์ 1 วินาที

2. C Programming Language

การเขียนโปรแกรมบน Arduino IDE แบ่งได้ออกเป็น 2 ส่วนคือ การจัดรูปแบบโครงสร้างการเขียนโปรแกรมออกเป็นฟังก์ชันย่อยแต่ละฟังก์ชัน และเมื่อนำฟังก์ชันทั้งหมดมารวมกัน เรียกว่าโปรแกรม การเขียนโปรแกรมบน Arduino IDE จะประกอบด้วย 2 ฟังก์ชันหลักคือ setup() และ loop()

ตัวอย่างการใช้งานเบื้องต้นสำหรับการพัฒนา

DHT11 | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help



```

1 #include "DHT.h"
2 #define DHTPIN D4
3 #define DHTTYPE DHT11
4 DHT dht(DHTPIN, DHTTYPE);
5
6 void setup() {
7   Serial.begin(9600);
8   dht.begin();
9 }
10 void loop() {
11   float h = dht.readHumidity();
12   float t = dht.readTemperature();
13   Serial.print("Temp: ");
14   Serial.println(t);
15   Serial.print("Hum : ");
16   Serial.println(h);
17   delay(1000);
18 }

```

ภาพที่ 2. 4 โปรแกรมการอ่านอุณหภูมิและค่าความชื้นของอากาศ

บรรทัดที่ 1: เรียกใช้ Library DHT

บรรทัดที่ 2: กำหนดขาพินที่ทำการส่งข้อมูลระหว่างเซ็นเซอร์และบอร์ด

บรรทัดที่ 3: ระบุนชนิดของเซ็นเซอร์เป็น DHT11

บรรทัดที่ 4: ระบุน object ของ DHT หากมีมากกว่า 1 ตัวให้เพิ่มค่า DHT dht2 =

DHT(pin, type);

บรรทัดที่ 6: ฟังก์ชัน Setup

บรรทัดที่ 7: ตั้งค่าพอร์ต Serial ในการรับส่งข้อมูล Baudrate อยู่ที่ 9600 bits/s

บรรทัดที่ 8: เปิดใช้งานการส่งข้อมูลกับตัวเซ็นเซอร์

บรรทัดที่ 10: ฟังก์ชัน loop

บรรทัดที่ 11: ประกาศตัวแปร h เป็นค่าทศนิยมและเก็บค่าความชื้นที่อ่านได้จากเซ็นเซอร์

บรรทัดที่ 12: ประกาศตัวแปร t เป็นค่าทศนิยมและเก็บค่าอุณหภูมิที่อ่านได้จากเซ็นเซอร์

บรรทัดที่ 13: แสดงข้อความ Temp: ผ่านทาง Serial Monitor

บรรทัดที่ 14: แสดงค่าของอุณหภูมิและขึ้นบรรทัดใหม่

บรรทัดที่ 15: แสดงข้อความ Hum: ผ่านทาง Serial Monitor

บรรทัดที่ 16: แสดงค่าของความชื้นและขึ้นบรรทัดใหม่

บรรทัดที่ 17: ดีเลย์ 1 วินาที

โครงสร้างพื้นฐานภาษาซีที่เขียนบน Arduino IDE จะประกอบด้วย 3 ส่วน

- Header เป็นส่วนแรกสุดที่จะทำการเขียนเพื่อเรียกใช้ในส่วนของเราต่าง ๆ รวมไปถึงการประกาศตัวแปร และค่าคงที่ที่ใช้ในโปรแกรม
- setup() เป็นฟังก์ชันบังคับที่ต้องมีในทุกโปรแกรม ฟังก์ชันนี้จะใส่คำสั่งที่จะทำงานเพียงรอบเดียว เมื่อโปรแกรมเริ่มทำงาน คำสั่งในการที่ใช้ในฟังก์ชันนี้จะเกี่ยวกับการกำหนดค่าการทำงานต่าง ๆ เช่น ขา pinMode, พอร์ตที่ใช้ส่งข้อมูลเข้าหรือออก, การกำหนดค่า Baudrate
- loop() เป็นฟังก์ชันบังคับเช่นเดียวกับ setup() ที่จำเป็นต้องมีในทุกโปรแกรม ในฟังก์ชันนี้โปรแกรมจะทำงานวนซ้ำแบบไม่มีที่สิ้นสุด

3. PHP

PHP (ตัวย่อแบบเรียกซ้ำสำหรับPHP: Hypertext Preprocessor) เป็นภาษาสคริปต์แบบโอเพนซอร์สที่ใช้กันอย่างแพร่หลายเหมาะอย่างยิ่งสำหรับการพัฒนาเว็บและสามารถฝังลงใน HTML ได้ โดยมีรากฐานโครงสร้างคำสั่งมาจากหลายภาษา เช่น ภาษา C ภาษา JAVA และภาษา PERL เป็นต้น โดยเป้าหมายหลักของภาษานี้ คือ ให้นักพัฒนาเว็บไซต์สามารถเขียนเว็บเพจที่มีการตอบโต้ได้อย่างรวดเร็ว การแสดงผลของภาษา PHP จะปรากฏในลักษณะ HTML โดยจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่ภาษา PHP นั้นแตกต่างจากภาษาในลักษณะ

ไคลเอนต์-ไซด์ สคริปต์ เช่น JAVA สคริปต์ โดยความสามารถการประมวลผลหลักของภาษา PHP ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผลการอ่านข้อมูลจากฐานข้อมูล (www.php.net)

ตัวอย่างการใช้งานโค้ดเบื้องต้นสำหรับการพัฒนา

```
<?php

    /*Get Data From POST Http Request*/

    $datas = file_get_contents('php://input');

    /*Decode Json From LINE Data Body*/

    $deCode = json_decode($datas,true);

    file_put_contents('log.txt',
file_get_contents('php://input') . PHP_EOL, FILE_APPEND);

    $replyToken = $deCode['events'][0]['replyToken'];

    $recv_msg = $deCode['events'][0]['message']['text'];

    $messages = [];

    $messages['replyToken'] = $replyToken;

    $rep_msg = [];

?>
```

อธิบายโค้ด

เป็นโค้ด PHP ที่ใช้ในการเขียน Chat bot โดยมีฟังก์ชันต่าง ๆ เช่น การถอดรหัส Json การสร้างตัวแปรเพื่อรับข้อมูล และการดึงข้อมูลจากไฟล์ข้อมูลเพื่อมาแสดง

2.2.2 อุปกรณ์สำหรับสถานีอากาศ

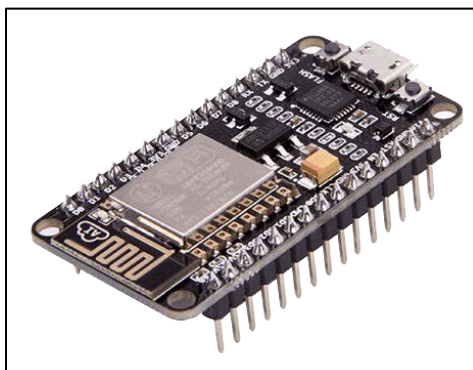
1. NodeMCU ESP8266

ESP8266 เป็นชื่อของโมดูลชิพขนาดเล็ก หรือจะเรียกว่าเป็นบอร์ดแบบฝังตัวชนิดที่สามารถเชื่อมต่อไวไฟได้ มาพร้อมชิพไวไฟ ESP-12 ขนาดเล็กใช้พลังงานต่ำ รองรับการทำงานทั้งแบบ Client, Access Point และ Client + Access Point รองรับการทํางาน

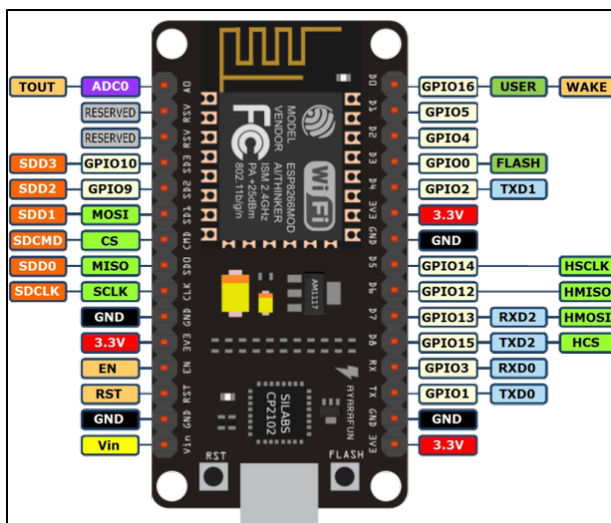
ที่แรงดันไฟฟ้า 3.0-3.6 โวลต์ ทำงานใช้กระแสโดยเฉลี่ยที่ 80 มิลลิแอมป์ ในโหมดประหยัดพลังงานจะใช้กระแสน้อยกว่า 10 ไมโครแอมป์และสามารถปลุกให้กลับมาทำงานภายในเวลาน้อยกว่า 2 มิลลิวินาที อุณหภูมิในการทำงานอยู่ในช่วง -40 ถึง 125 องศาเซลเซียส

โครงสร้างและพอร์ตเชื่อมต่อที่มีลักษณะการใช้งานที่เหมือนกันในหลาย ๆ รุ่น

- GPIO0 ใช้สำหรับเลือกโหมดการทำงาน เมื่อเชื่อมต่อเข้ากับขา GND
- CH_PD หรือ EN เป็นขาที่ต้องต่อไฟ VCC เพื่อ Pull up สัญญาณให้โมดูลทำงาน โมดูลบางรุ่นไม่มีขาเรซีต เมื่อต้องการเรซีตให้ต่อขา CH_PD และ GND
- VCC ขาสำหรับจ่ายไฟ 5 โวลต์
- GND ต่อกับขา GND หรือ – ระหว่างเซ็นเซอร์ หรือไฟ 0 โวลต์
- GPIO ขา Digital INPUT/OUTPUT ทำงานที่ไฟ 3.3 โวลต์
- ADC ขา Analog INPUT รับแรงดันสูงสุด 1 โวลต์ ความละเอียด 10 บิต หรือ 1024 ค่า



ภาพที่ 2. 5 NodeMCU ESP8266



ภาพที่ 2. 6 NodeMCU GPIO Pinout

2. แอร์การ์ด

อุปกรณ์โมเด็มที่ใช้เชื่อมต่ออินเทอร์เน็ตแบบไร้สาย ผ่านโครงข่ายสัญญาณโทรศัพท์มือถือ ซึ่งขณะที่เชื่อมต่ออินเทอร์เน็ตยังสามารถใช้โทรศัพท์มือถือโทรเข้า-ออกได้ ในเวลาเดียวกัน เพราะมีการแบ่งช่องสัญญาณแยกออกจากกัน แต่ใช้ Cellsite เดียวกัน ดังนั้นการใช้งานอินเทอร์เน็ตสามารถใช้งานได้เพียงแค่มีสัญญาณโทรศัพท์มือถือ ผ่านโครงข่ายเทคโนโลยี GPRS และ EDGE



ภาพที่ 2. 7 แอร์การ์ด

3. โมดูลกล้อง ESP32 CAM

โมดูลกล้องที่สามารถใช้งานร่วมกับแพลตฟอร์ม Arduino จะประกอบด้วยไมโครคอนโทรลเลอร์สามารถแยกการทำงานได้อย่างอิสระ มีโมดูลของ WiFi + Bluetooth และมีช่องเสียบ microSD สำหรับจัดเก็บข้อมูล



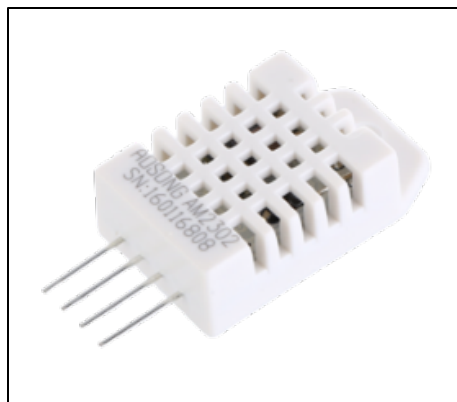
ภาพที่ 2. 8 โมดูลกล้อง ESP32 CAM

4. เซ็นเซอร์วัดค่าความชื้น

โมดูลหรือเซ็นเซอร์สำหรับวัดอุณหภูมิและความชื้นในอากาศ สามารถใช้งานร่วมกับ Single Board Computer หลายรุ่น โดยการรับส่งข้อมูลจาก DHT22 ผ่านสายสัญญาณเส้นเดียวและเป็นสัญญาณดิจิทัล

คุณสมบัติ

- ใช้แรงดันไฟฟ้า 3.3 – 6 โวลต์
- ใช้กระแสไฟฟ้าสูงสุด 2.5mA ขณะทำการหารวัดค่า
- เหมาะสำหรับความชื้น 20-80% ความผิดพลาดในการอ่านค่าไม่เกิน 5%
- เหมาะสำหรับอุณหภูมิ -40 ถึง 125 องศาเซลเซียส ความผิดพลาดในการอ่านค่าไม่เกิน 0.5 องศาเซลเซียส
- ความถี่ในการวัดค่า 2 Hz (1 ครั้งต่อวินาที)



ภาพที่ 2. 9 เซนเซอร์ DHT22

5. โมดูลวัดความเร็วลม ทิศทางลมและปริมาณน้ำฝน

ภายในโมดูลนี้จะประกอบด้วย โมดูลวัดความเร็วลม วัดทิศทางลมและปริมาณน้ำฝน เซ็นเซอร์ทุกตัวเชื่อมต่อผ่านพอร์ต RJ10 ผ่าน Pre processor เพื่อแปลงค่าที่อ่านได้จาก เซ็นเซอร์ส่งผ่าน Serial Port ไปยัง Processor หลัก



ภาพที่ 2. 10 เซนเซอร์วัดความเร็วลม ทิศทางลมและปริมาณน้ำฝน

6. เซนเซอร์วัดค่า PM2.5

PMS5003 Sensor Module Air Particle Dust

เป็นเซ็นเซอร์ตรวจวัดฝุ่นที่สามารถวัดค่า PM1.0 PM2.5 และ PM10 โดยเซ็นเซอร์วัดค่าอนุภาคของฝุ่นที่มีขนาดเล็กถึง 0.3 ไมโครเมตร ตัวเซ็นเซอร์จะใช้แสงเลเซอร์ในการตรวจวัด

ฝุ่นในอากาศทำให้มีความแม่นยำในการตรวจวัดค่าฝุ่น เซ็นเซอร์ทำงานที่แรงดัน 5V โดยจะมี Output ออกมาเป็น UART(TX/RX) ที่แรงดัน 3.3V



ภาพที่ 2. 11 PMS5003 Sensor Module Air Particle Dust

2.2.3 ระบบคลาวด์สาธารณะ

1. ThingSpeak

ThingSpeak เป็น cloud ที่ถูกออกแบบสำหรับเก็บข้อมูลของอุปกรณ์ IoT (internet of things) เป็นบริการฟรีที่มีข้อจำกัด เช่น การเก็บข้อมูลของ Arduino, Raspberry Pi เป็นต้น ไม่เป็นแค่เก็บข้อมูลเท่านั้น ยังสามารถแสดงผลข้อมูลที่เก็บเป็นกราฟ หรือ export ข้อมูลเพื่อทำไปใช้ต่อได้ จริงๆ มันไม่เพียงเก็บข้อมูลเฉพาะพวก IoT แต่ยังเก็บข้อมูลของ server ได้อีก

```
1 #include "ThingSpeak.h"
2 unsigned long myChannelNumber = ;
3 const char * myWriteAPIKey = "";
4   ThingSpeak.begin(client);
5   ThingSpeak.setField(1, temp);
6   ThingSpeak.writeFields(ChannelNumber,
WriteAPIKey);
```

บรรทัดที่ 1 เรียกใช้ไลบรารี ThingSpeak

บรรทัดที่ 2 กำหนดหมายเลข Channel ของ ThingSpeak

บรรทัดที่ 3 กำหนด API Keys

บรรทัดที่ 4 เริ่มเปิดใช้งาน ThingSpeak

บรรทัดที่ 5 กำหนด Field และตัวแปรที่ทำการบันทึกค่าลงบน Field

บรรทัดที่ 6 เขียนข้อมูลลงบน Field ที่กำหนดไว้

1. Firebase

Firebase คือผลิตภัณฑ์ของ Google เป็นบริการประเภท Platform as a Service ที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในฝั่งของฐานข้อมูลหรือ Backend ให้บริการในรูปแบบของ Realtime Database สำหรับการทำ Web Application และ Application อีกทั้งยังรองรับการให้บริการในส่วนของ Internet of Things (IoT) Firebase ทำหน้าที่เป็นตัวกลางที่จะคอยรับข้อมูลจากเซ็นเซอร์และผู้พัฒนาสามารถนำข้อมูลไปแสดงผลบนหน้าจอ พลิกแซตามที่ต้องการผ่านในส่วนของการเรียกใช้ API

2. GitHub

เป็นเว็บไซต์ที่ให้บริการพื้นที่จัดเก็บโปรเจกต์ที่เก็บเฉพาะโค้ดไฟล์ โดยสามารถร่วมพัฒนากับผู้อื่นได้โดยเป็นโอเพ่นซอร์ส และเป็นเว็บไซต์ที่ให้บริการ Git Server เป็นบริการฟรี ถ้าหากว่าโปรเจกต์นั้นเป็น Public คือ ผู้อื่นสามารถเข้ามาดูหรือดาวโหลดโค้ดได้ แต่ถ้าหากอยากให้ข้อมูลนั้นเป็น Private คือ เป็นส่วนตัวผู้อื่นไม่สามารถเข้าถึงได้ รูปแบบดังกล่าวจะต้องเสียค่าใช้จ่ายในการใช้งาน (Thanatcha Kromsang, 2560)

3. Heroku

เป็นบริการคลาวด์แบบ Platform as a Service (Paas) ที่ให้บริการสำหรับนักพัฒนาซอฟต์แวร์ สามารถรองรับภาษาต่างๆ เช่น Java , Python , PHP ,Ruby, Go และ Node.js เป็นต้น โดยนักพัฒนาซอฟต์แวร์สามารถใช้ Heroku ในการ deploy และจัดการแอปพลิเคชัน เพราะ Heroku มี Add-ons สำหรับเพิ่มเติมบริการอื่นๆ เช่น PostgreSQL, MongoDB, Redis เป็นต้น โดยสามารถเลือกใช้งานฟรีหรือเสียค่าบริการได้ (Onamon Ja, 2563)

4. Line Bot

LINE Messaging API เป็น Bot หรือช่วยในการพูดคุยโต้ตอบ สามารถออกแบบข้อความการโต้ตอบที่อยากคุยกับผู้ใช้ โดยกำหนดว่าจะดึงค่าอะไรมาโชว์เมื่อผู้ใช้งานร้องขอมาและรองรับเครื่องมือที่เอาไว้สำหรับพัฒนาโปรแกรมหรือแอปพลิเคชันบนระบบ Android, OS รองรับภาษา JAVA, Rudy, PHP และ Perl5 โดยหากต้องการใช้งาน Line Bot ให้ทำการเพิ่ม ID หรือ แสกน QR code เพื่อติดต่อกับ Line Bot (Nattapon Sirikamonnet, 2561)

5. Webhook

Webhook คือ event ต่าง ๆ ที่เกิดขึ้นกับ LINE Bot (Event trigger) โดยเมื่อ event เกิดขึ้นแล้วจะมีสัญญาณพร้อมกับข้อมูลในรูปแบบที่เป็น JSON วิ่งมาที่ Webhook API ที่เราผูกไว้ใน LINE Developers Console (Jirawatee, 2562)

2.3 งานวิจัยหรือระบบที่ใกล้เคียง

ตารางที่ 2.1 งานวิจัยหรือระบบที่เกี่ยวข้อง

	ปัญหา	วิธีแก้	ผล
The Weather-Proximity-Cognition (WPC) framework A camping, weather, and climate change case	ธุรกิจแคมป์ปี้งมีความอ่อนไหวในเรื่องของสภาพอากาศ	<ul style="list-style-type: none"> - ดึงข้อมูลภูมิอากาศ - สร้างการแสดงผลภาพกราฟิกของข้อมูลภูมิอากาศ - วิเคราะห์ความสัมพันธ์ระหว่างข้อมูลธุรกิจเกี่ยวกับการตั้งแคมป์และภูมิอากาศ 	การใช้กรอบของ WPC หรือกรอบความเข้าใจและการตัดสินใจเกี่ยวกับสภาพอากาศ โดยใช้ข้อมูลภูมิอากาศรายวันภายในฤดูของอุตุนิยมหาวิทยาลัย
Design and Implement of a Weather Monitoring Station using CoAP on NB-IoT Network	เพื่อตรวจสอบและแจ้งเตือนสภาพอากาศที่มีการเปลี่ยนแปลงได้ตลอดเวลา	สร้างสถานีอากาศโดยมุ่งเน้นไปที่ความหลากหลายของอุปกรณ์ IoT และถ่ายโอนข้อมูลผ่านเครือข่าย NB-IoT ข้อมูลจะถูกบันทึกไปยัง MySQL ข้อมูลที่ได้จะถูก	ระบบที่มีความสามารถในการแสดงข้อมูลสภาพอากาศ วัตถุประสงค์ ความขึ้น ความเร็วและทิศทางลม ก๊าซโอโซน ปริมาณน้ำฝน ข้อมูลที่กล่าวมาจะแสดงผล ณ เวลาจริงบนคอมพิวเตอร์ส่วนบุคคล

		วิเคราะห์และแสดงผลบน Grafana	
Line Bot Implementation for Automation Balinese Language Dictionary	การสื่อสารของนักท่องเที่ยวต่างชาติที่เข้ามาท่องเที่ยวมีการสื่อสารที่ไม่เข้าใจเนื่องจากข้อจำกัดทางด้านภาษา	<ul style="list-style-type: none"> - การสร้าง Chat bot ผ่านโปรแกรมออนไลน์ที่ช่วยในการสื่อสารกับนักท่องเที่ยวผ่านโปรแกรมไลน์ - การแปลภาษาจากภาษาที่นักท่องเที่ยวใช้งานเป็นภาษาบาลี 	โปรแกรม Chat bot ที่สามารถสื่อสารกับนักท่องเที่ยวผ่านโปรแกรมไลน์
A Development on Air Pollution Detection Sensors based on NB-IoT Network for Smart Cities	ปัญหามลพิษทางอากาศเป็นปัจจัยที่ส่งผลกระทบต่อการใช้ชีวิตของผู้คนในเมือง ผู้คนจึงไม่ทราบถึงปริมาณของมลพิษทางอากาศแต่ละพื้นที่ว่าคุณภาพของอากาศเป็นพิษหรือไม่	พัฒนาเซ็นเซอร์ตรวจจับมลพิษทางอากาศออกแบบระบบโดยใช้เซ็นเซอร์ 5 ตัวประมวลผลผ่าน Micro controller ส่งผ่าน NB-IoT และแสดงผลผ่านเว็บไซต์	เซ็นเซอร์สามารถวัดค่าคาร์บอนไดออกไซด์ โอโซน PM10 ไนโตรเจนไดออกไซด์ และซัลเฟอร์ไดออกไซด์ ค่าจะถูกประมวลผลและแสดงให้ทราบว่าพื้นที่นั้นมีคุณภาพอากาศเป็นอย่างไร

2.4 ผลการศึกษาเบื้องต้น

2.4.1 การนำค่าจากเซ็นเซอร์ขึ้น ThingSpeak

การทดสอบแนวคิดในส่วนของการนำค่าจากเซ็นเซอร์ที่ต่อกับบอร์ด NodeMCU ESP8266 และนำค่าที่อ่านได้มาแสดงผลบน ThingSpeak และใช้ ThingSpeak เป็น Cloud Database

```

1 #include <ESP8266WiFi.h>
2 #include "ThingSpeak.h"
3 #include "DHT.h"
4 #define DHTPIN D1
5 #define DHTTYPE DHT22
6 DHT dht(DHTPIN, DHTTYPE);

```

```

7  const char* ssid = "";
8  const char* password = "";
9  WiFiClient client;
10 unsigned long ChannelNumber =;
11 const char* WriteAPIKey = "";
12 int temperature;
13 void setup{
14     Serial.begin(9600);
15     dht.begin();
16     ThingSpeak.begin(client);
17     WiFi.begin(ssid, password);
18     while(WiFi.status() != WL_CONNECTED){
19         delay(1000);
20         Serial.print(".");
21     }
22 void loop{
23     temperature = dht.readTemperature();
24     Serial.print(F("Temp:")); Serial.println(t);
25     ThingSpeak.setField(1, temperature);
26     ThingSpeak.writeFields(ChannelNumber,
        WriteAPIKey);
27 }

```

บรรทัดที่ 1-3: ทำการ import library ที่ต้องใช้งาน

บรรทัดที่ 4-6: กำหนด Pin และประเภทของเซ็นเซอร์

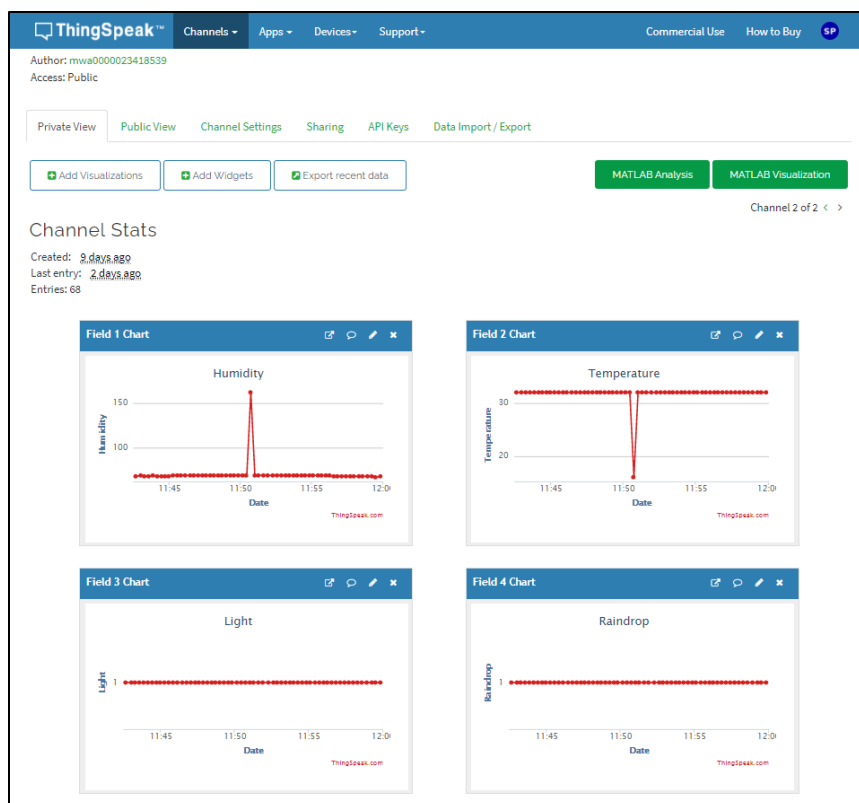
บรรทัดที่ 7-8: กำหนดค่า SSID และ Password

บรรทัดที่ 10-11: กำหนดเลขช่องและ API Key ของ Thingspeak

บรรทัดที่ 12: ประกาศตัวแปรเพื่อเก็บค่า

บรรทัดที่ 13-21: ฟังก์ชัน Setup กำหนดค่าความเร็วของพอร์ต Serial และเชื่อมต่อ WiFi

บรรทัดที่ 22-27: อ่านค่าจากเซ็นเซอร์และส่งค่า



ภาพที่ 2. 12 หน้าจอ ThingSpeak เมื่อดึงค่าจากเซนเซอร์ได้

2.4.3 Line Bot ตอบโต้ข้อความ

การทดลองแนวคิด (Proof of concept) ในส่วนของ Line bot ที่ทำการตอบโต้ข้อความกับผู้ใช้งาน เมื่อผู้ใช้งานนั้นทำการส่งข้อความเข้ามา Line Bot จะทำการตอบกลับข้อความที่ได้กำหนดไว้ไปยังผู้ใช้งาน

```
1 if($recv_msg == "สวัสดี") {
2     $rep_msg ['text'] = "สวัสดีครับ";
3     $rep_msg ['type'] = 'text';
```

บรรทัดที่ 1 เป็นการกำหนดคำหรือข้อความที่ผู้ใช้งานส่งเข้ามายังระบบ

บรรทัดที่ 2 เป็นการกำหนดคำหรือข้อความที่ระบบนั้นตอบกลับไปยังผู้ใช้งาน

บรรทัดที่ 3 เป็นการกำหนดประเภทของคำหรือข้อความที่ตอบกลับให้เป็นประเภทของข้อความ (text)



ภาพที่ 2. 13 หน้าข้อความตอบโต้ระหว่างผู้ใช้งานกับ Line Bot

2.4.4 Line Bot ดึงค่าจาก ThingSpeak

การทดลองแนวคิด (Proof of concept) ในส่วนของ Line Bot ดึงค่าจาก ThingSpeak เพื่อแสดงให้เห็น ในสิ่งที่ผู้ใช้งานนั้นต้องการ

```

1 if ($recv_msg == "อุณหภูมิ") {
2     $url =
    "https://api.thingspeak.com/channels/1483851/feeds.json?results=1";
3     $strRet = file_get_contents($url) ;
4     $strRet = json_decode($strRet) ;
5     $temp = $strRet->feeds [0]->field2 ;
6     $rep_msg['text'] = $temp;
7     $rep_msg['type']='text' ;

```

บรรทัดที่ 1 เป็นการกำหนดคำหรือข้อความที่ผู้ใช้งานส่งเข้ามายังระบบ

บรรทัดที่ 2-4 เป็นการใส่ url ไปในตัวแปร จากนั้นใช้ตัวแปรมารองรับเพื่อดึงค่าจาก url และทำการถอดรหัสตัวแปรดังกล่าว

บรรทัดที่ 5 เป็นการดึงค่า api จาก url เพื่อนำมาแสดง โดยจะรองรับด้วยตัวแปร temp

บรรทัดที่ 6 เป็นการกำหนดคำหรือข้อความที่ระบบนั้นตอบกลับไปยังผู้ใช้งาน โดยจะรับค่ามาจากตัวแปร temp

บรรทัดที่ 7 เป็นการกำหนดประเภทของคำหรือข้อความที่ตอบกลับให้เป็นประเภทของข้อความ (text)

2.4.5 Line Bot ดึงค่าจาก Firebase

การทดลองแนวคิด (Proof of concept) ในส่วนของ Line Bot ดึงค่าจาก Firebase เพื่อแสดงให้เห็น ในสิ่งที่ผู้ใช้งานนั้นต้องการ

```

1 if($recv_msg == "ภาพรวม") {
2 $rep_msg2['originalContentUrl'] =
    "https://firebasestorage.googleapis.com/v0/b/esp-

```

```

firebase-demo-
c8454.appspot.com/o/data%2Fphoto.jpg?alt=media&token=4
415c22a-a0ba-4813-a7c0-5691f71ed343";
3 $rep_msg2['previewImageUrl'] =
"https://firebasestorage.googleapis.com/v0/b/esp-
firebase-demo-
c8454.appspot.com/o/data%2Fphoto.jpg?alt=media&token=4
415c22a-a0ba-4813-a7c0-5691f71ed343";
4 $rep_msg2['type']='image';}

```

บรรทัดที่ 1 เป็นการกำหนดค่าหรือข้อความที่ผู้ใช้งานส่งเข้ามายังระบบ

บรรทัดที่ 2 เป็นการใส่ url ไปในตัวแปร เพื่อเรียกดูรูปภาพล่าสุดจาก url จากนั้นทำการส่งไปยังผู้ใช้ระบบ

บรรทัดที่ 6 เป็นการใส่ url ไปในตัวแปร เพื่อเรียกดูรูปภาพล่าสุดจาก url จากนั้นทำการส่งไปยังผู้ใช้ระบบเมื่อมีการกดขยายรูปภาพ

บรรทัดที่ 7 เป็นการกำหนดประเภทของรูปภาพหรือข้อความที่ตอบกลับให้เป็นประเภทของรูปภาพ (image)

บทที่ 3

การวิเคราะห์ระบบ

3.1 องค์กรที่เกี่ยวข้อง

3.1.1 กลุ่มของผู้ใช้งาน

กลุ่มของผู้ใช้งานแบ่งออกเป็น 2 ส่วน คือ

1. นักท่องเที่ยว คือ กลุ่มที่ท่องเที่ยวเชิงอนุรักษ์ที่หลงใหลในการเดินทางไปยังแหล่งธรรมชาติ และแหล่งวัฒนธรรมที่มีเอกลักษณ์ ซึ่งการทราบถึงสภาพอากาศล่วงหน้าก่อนออกเดินทางทำให้มีผลต่อการตัดสินใจในการไปท่องเที่ยวในสถานที่นั้น ๆ
2. เจ้าของสถานที่เป็นได้ทั้งหน่วยงานภาครัฐและเอกชนที่ต้องการนำสถานีอากาศไปติดตั้งในพื้นที่ ให้ผู้ใช้ทราบถึงข้อมูลสภาพอากาศก่อนที่จะเดินทางมายังสถานที่ที่ติดตั้งสถานีอากาศเพื่อเพิ่มความสะดวกสบายให้ผู้ใช้งาน

3.1.2 ข้อมูลเบื้องต้นขององค์กร หรือ หน่วยงานที่เกี่ยวข้อง

จากการศึกษางานวิจัยงานวิจัยที่เกี่ยวข้องของระบบการพัฒนาสถานีอากาศท้องถิ่นเพื่อการท่องเที่ยวเชิงอนุรักษ์พบว่าองค์กรหรือหน่วยงานที่เกี่ยวข้องกับระบบ ดังนี้

1. ที่พักเชิงอนุรักษ์ คือ ที่พักรักษ์โลก ที่มีแนวคิดในการอนุรักษ์สิ่งแวดล้อม เช่น การจัดการแบบอนุรักษ์พลังงาน ไม่รุกรานธรรมชาติ หรือสร้างห้องพักต่าง ๆ ด้วยวัสดุจากธรรมชาติ รวมไปถึงอาจมีกิจกรรมทางธรรมชาติสำหรับผู้เข้าพักด้วย (Travelaholic, 2561) เช่น แก่งกระจาน เจ็ดคต-โป่งก้อนเส้า ภูทับเบิก เป็นต้น
2. อุทยานแห่งชาติ คือ พื้นที่ที่สงวนไว้ เพื่อคุ้มครองรักษาทรัพยากรธรรมชาติ โดยเฉพาะป่าไม้และสัตว์ป่า ตลอดจนทิวทัศน์ ธรรมชาติที่สวยงาม และมหัศจรรย์อันเป็นที่ดึงดูดใจแก่ผู้พบเห็นให้คงอยู่ในสภาพธรรมชาติเดิมมิให้ถูกให้ ทำลายหรือเปลี่ยนแปลงไป และถูกใช้ไปในทางที่ผิด เพื่อรักษาสมบัติของธรรมชาติไว้ให้นุชนรุ่นหลัง ๆ ได้ชมได้ศึกษา ค้นคว้าธรรมชาตินั้น ๆ ต่อไป (อุทยานแห่งชาติ, 2556) เช่น อุทยานแห่งชาติห้วยน้ำดัง ดอยเสมอดาว อุทยานแห่งชาติศรีน่าน เป็นต้น

3.2 วิธีการรวบรวมข้อมูล

3.2.1 การค้นหาจากเอกสาร

1. การท่องเที่ยวเชิงนิเวศ (ecotourism) หมายถึงการท่องเที่ยวในแหล่งธรรมชาติที่มีเอกลักษณ์เฉพาะถิ่นและแหล่งวัฒนธรรมที่เกี่ยวข้องกับระบบนิเวศ โดยมีกระบวนการเรียนรู้ร่วมกันของผู้ที่เกี่ยวข้องภายใต้การจัดการสิ่งแวดล้อมและการท่องเที่ยวอย่างมีส่วนร่วมของ

ท้องถิ่นเพื่อมุ่งเน้นให้เกิดจิตสำนึกต่อการรักษาระบบนิเวศอย่างยั่งยืน โดยนักท่องเที่ยวที่มีการเดินทางโดยรถส่วนตัวหรือรถประจำทางเพื่อไปยังสถานที่นั้น ซึ่งรูปแบบการอยู่อาศัยนั้นคือ การกางเต็นท์หรือการแคมป์ปิ้ง เพื่อที่จะสัมผัสกับธรรมชาติและ การพักผ่อน

2. สถานีอากาศ (Weather Station) ถูกออกแบบให้สามารถทำงานได้ด้วยตนเองแบบอัตโนมัติ สามารถรับข้อมูลจากเซนเซอร์ และส่งออกเก็บในฐานข้อมูล พร้อมแสดงผลบนหน้าแอปพลิเคชันตามเวลาจริง สามารถติดตั้งได้หลากหลายสถานที่ โดยมีเงื่อนไขการใช้งานแค่แหล่งจ่ายไฟและอินเทอร์เน็ตแบบไร้สายที่ครอบคลุมอุปกรณ์ กระบวนการทำงานจะประกอบด้วย 5 ส่วน

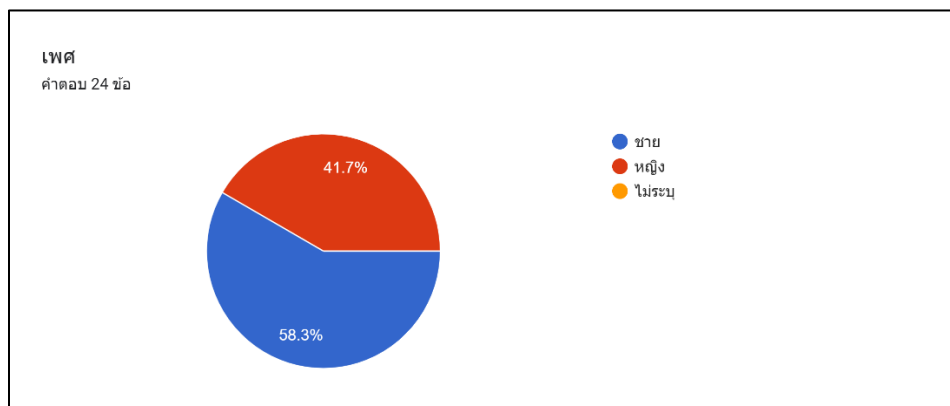
1. เซนเซอร์ เป็นส่วนที่ติดตั้งอุปกรณ์ตรวจค่าต่าง ๆ
2. ตัวประมวลผล มีความสามารถคล้ายคอมพิวเตอร์ ซอฟต์แวร์ประมวลผลจะถูกเขียนเข้าไปในตัวของหน่วยความจำ ตัวประมวลผลจะทำการอ่านข้อมูลจากเซนเซอร์ทุกตัวและแปลงค่าที่สามารถนำไปใช้ได้
3. เน็ตเวิร์ค เป็นส่วนเครือข่ายที่จะนำข้อมูลจากอุปกรณ์ประมวลผลไปยังฐานข้อมูลของระบบ และนำข้อมูลที่เก็บไว้ไปแสดงผลผ่านเว็บแอปพลิเคชันให้ผู้ใช้อีกครั้งหนึ่ง โดยระบบเครือข่ายที่ใช้ในระบบนี้จะใช้ในรูปแบบสัญญาณไร้สาย
4. ฐานข้อมูล ในส่วนนี้เป็นส่วนเก็บข้อมูลทั้งหมดของระบบที่ได้รับมาจากเซนเซอร์ซึ่งในขั้นตอนนี้จะใช้ฐานข้อมูล ซึ่งเป็นฐานข้อมูลแบบกลุ่มเมฆที่เหมาะสมกับการใช้งานในงานอินเทอร์เน็ตในสรรพสิ่ง หรือ Internet of Things ที่ระบบจะต้องมีการรับส่งข้อมูลตลอดเวลา และต้องการการแสดงผลแบบเวลาจริง (Real Time)
5. แสดงผลให้ผู้ใช้ ระบบสามารถแสดงผลผ่านเว็บแอปพลิเคชันได้จากอุปกรณ์ต่าง ๆ ไม่ว่าจะเป็นคอมพิวเตอร์ แท็บเล็ต หรือสมาร์ทโฟน

3.2.2 การเก็บข้อมูลโดยแบบสอบถาม

การทำแบบฟอร์มออนไลน์ผ่าน Google form เพื่อเก็บรวบรวมข้อมูลที่ใช้ต้องการ ซึ่งมีผู้ตอบแบบสอบถามจำนวน 24 คน ดังนี้

1. ข้อมูลส่วนบุคคล โดยจะมีการเก็บรวบรวมข้อมูลดังนี้

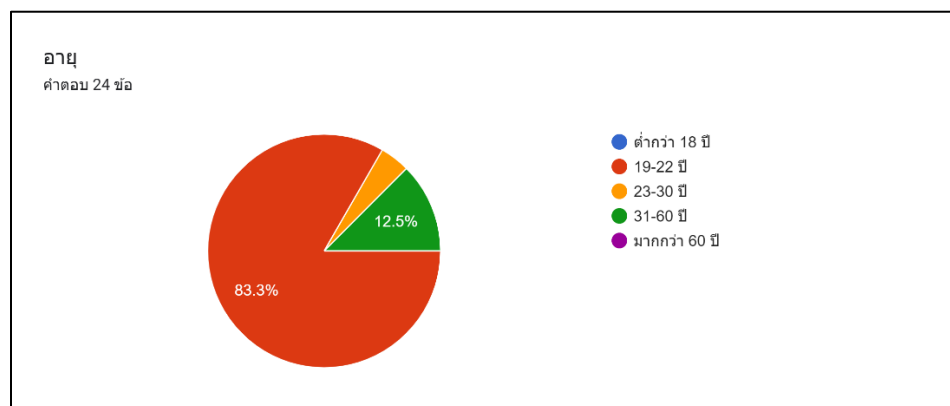
1.1. เพศ



ภาพที่ 3. 1 เพศ

จากภาพพบว่าผู้ตอบแบบสอบถามเป็นเพศชายจำนวน 14 คน คิดเป็นร้อยละ 58.3 เปอร์เซ็นต์ และเป็นเพศหญิงจำนวน 10 คน คิดเป็นร้อยละ 41.7 เปอร์เซ็นต์

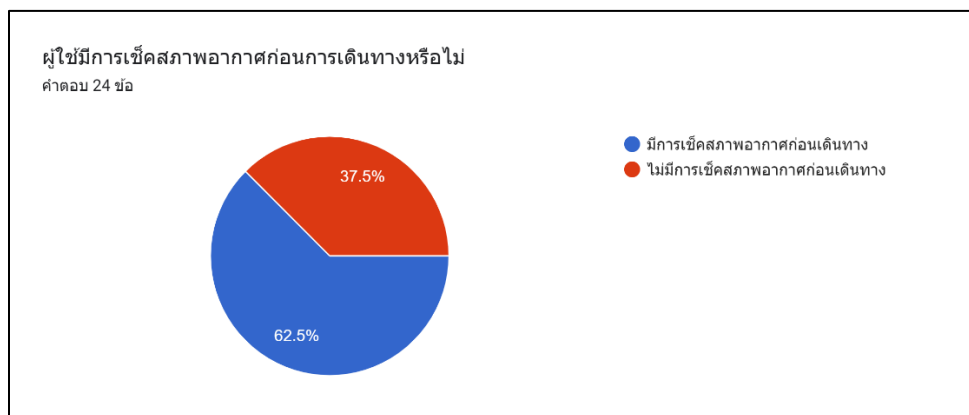
1.2. อายุ



ภาพที่ 3. 2 อายุ

จากภาพพบว่าผู้ตอบแบบสอบถามอยู่ในช่วงอายุระหว่าง 19-20 ปี จำนวน 20 คน คิดเป็นร้อยละ 83.3 เปอร์เซ็นต์ ช่วงอายุระหว่าง 23-30 ปี จำนวน 1 คน คิดเป็นร้อยละ 4.2 เปอร์เซ็นต์และช่วงอายุระหว่าง 31-60 ปี จำนวน 3 คน คิดเป็นร้อยละ 12.5 เปอร์เซ็นต์

1.3. การเช็คสภาพอากาศก่อนออกเดินทาง

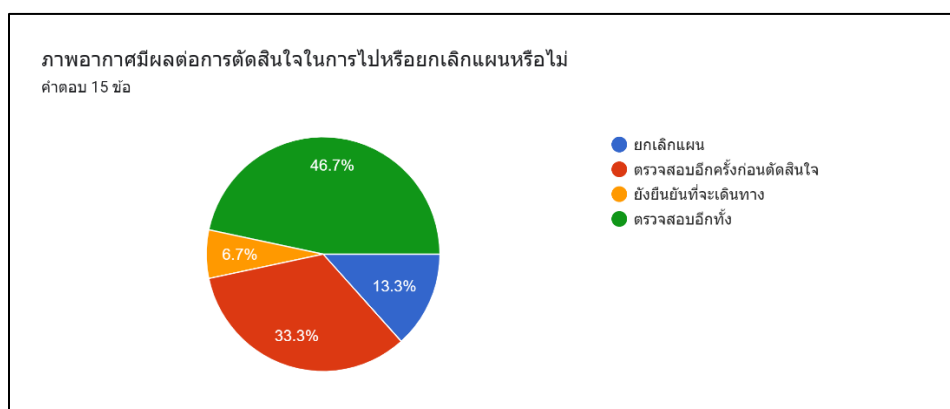


ภาพที่ 3. 3 การเช็คสภาพอากาศก่อนออกเดินทาง

จากภาพพบว่าผู้ตอบแบบสอบถามมีการเช็คสภาพอากาศก่อนออกเดินทาง จำนวน 15 คน คิดเป็นร้อยละ 62.5 เปอร์เซ็นต์และไม่มีการเช็คสภาพอากาศก่อนออกเดินทาง จำนวน 10 คน คิดเป็นร้อยละ 37.5 เปอร์เซ็นต์

2. สภาพอากาศและการตัดสินใจ โดยจะมีการเก็บรวบรวมข้อมูลดังนี้

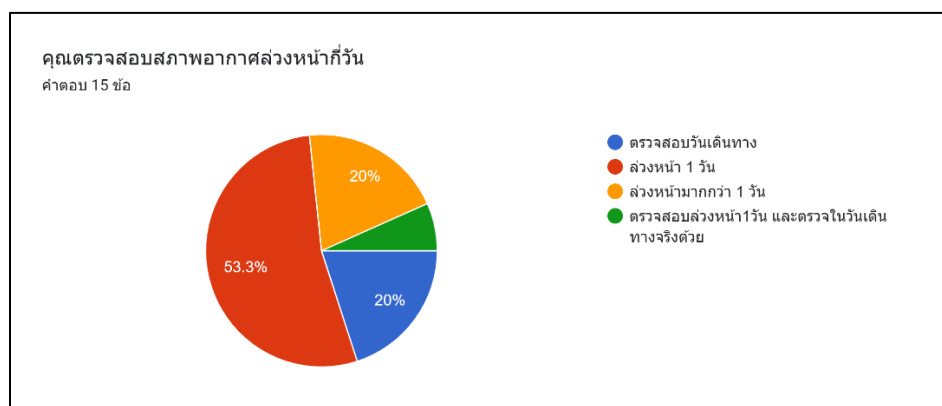
2.1. สภาพอากาศมีผลต่อการตัดสินใจในการไปหรือยกเลิกแผนหรือไม่



ภาพที่ 3. 4 สภาพอากาศมีผลต่อการตัดสินใจในการไปหรือยกเลิกแผนหรือไม่

จากภาพพบว่าสภาพอากาศนั้นมีผลต่อการตัดสินใจโดยมีการตรวจสอบสภาพอากาศอีกครั้งก่อนออกเดินทาง จำนวน 12 คน คิดเป็นร้อยละ 80 เปอร์เซ็นต์ ยกเลิกแผน จำนวน 2 คน คิดเป็นร้อยละ 13.3 เปอร์เซ็นต์และยืนยันที่จะเดินทาง จำนวน 1 คน คิดเป็นร้อยละ 6.7 เปอร์เซ็นต์

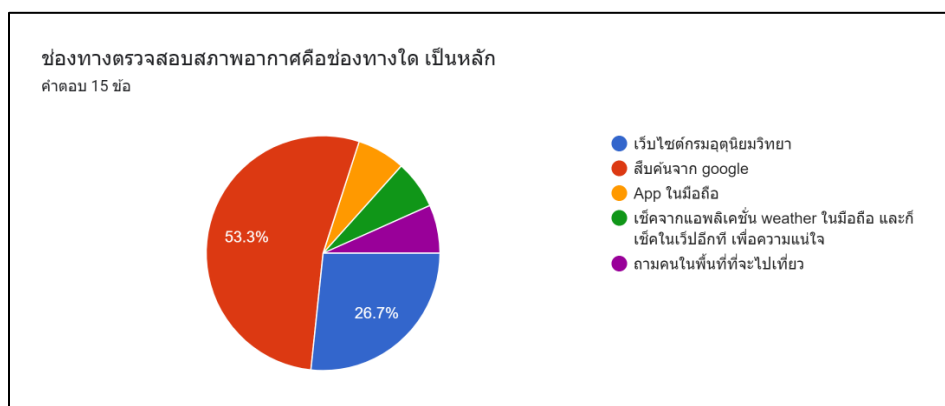
2.2. คุณตรวจสอบสภาพอากาศล่วงหน้ากี่วัน



ภาพที่ 3. 5 คุณตรวจสอบสภาพอากาศล่วงหน้ากี่วัน

จากภาพพบว่าก่อนออกเดินทางนั้นผู้ตอบแบบสอบถามนั้นมีการตรวจสอบสภาพอากาศล่วงหน้า 1 วัน จำนวน 8 คน คิดเป็นร้อยละ 53.3 เปอร์เซนต์ ล่วงหน้ามากกว่า 1 วัน จำนวน 3 คน คิดเป็นร้อยละ 20 เปอร์เซนต์ตรวจสอบวันเดินทาง จำนวน 3 คน คิดเป็นร้อยละ 20 เปอร์เซนต์และตรวจสอบล่วงหน้า 1 วัน โดยตรวจสอบในวันเดินทางจริงอีกครั้ง จำนวน 1 คน คิดเป็นร้อยละ 6.7 เปอร์เซนต์

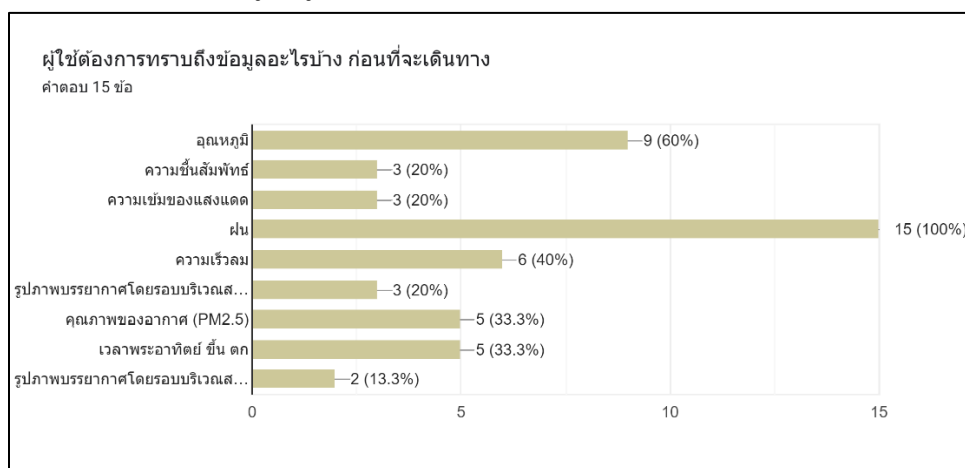
2.3. ช่องทางตรวจสอบสภาพอากาศคือช่องทางใด เป็นหลัก



ภาพที่ 3. 6 ช่องทางตรวจสอบสภาพอากาศคือช่องทางใด เป็นหลัก

จากภาพพบว่าผู้ตอบแบบสอบถามนั้นมีการตรวจสอบสภาพอากาศโดยสืบค้นจาก Google จำนวน 8 คน คิดเป็นร้อยละ 53.3 เปอร์เซนต์ เว็บไซต์กรมอุตุนิยมวิทยา จำนวน 4 คน คิดเป็นร้อยละ 26.7 เปอร์เซนต์ แอปพลิเคชันมือถือ จำนวน 1 คน คิดเป็นร้อยละ 6.7 เปอร์เซนต์ ถามคนในพื้นที่ที่จะเที่ยว จำนวน 1 คน คิดเป็นร้อยละ 6.7 เปอร์เซนต์และเช็คจากแอปพลิเคชัน Weather ในมือถือ และเช็คจากในเว็บอีกครั้งเพื่อความแน่ใจ จำนวน 1 คน คิดเป็นร้อยละ 6.7 เปอร์เซนต์

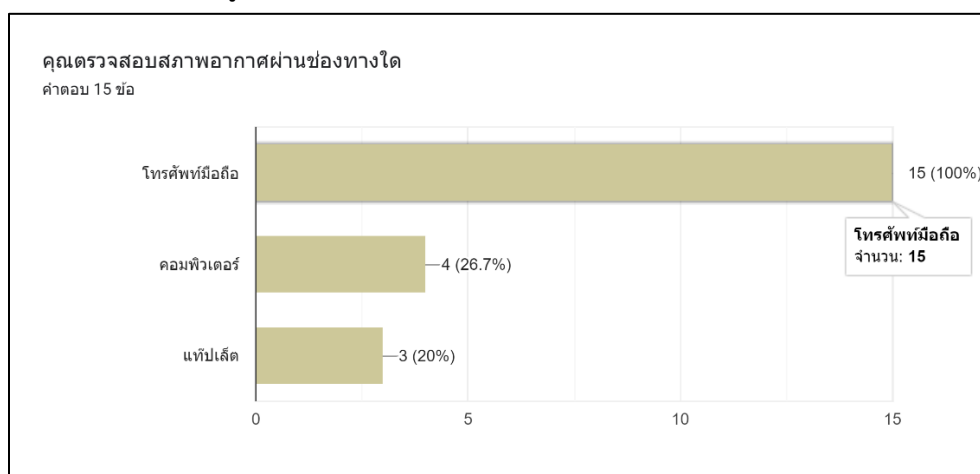
2.4. ข้อมูลที่ผู้ต้องการทราบก่อนเดินทาง



ภาพที่ 3. 7 ข้อมูลที่ผู้ต้องการทราบก่อนเดินทาง

จากภาพพบว่าผู้ตอบแบบสอบถามต้องการทราบข้อมูลสภาพอากาศก่อนออกเดินทาง 5 อันดับแรก ดังนี้ ฝน จำนวน 15 คน คิดเป็นร้อยละ 100 เปอร์เซนต์ อุณหภูมิ จำนวน 9 คน คิดเป็นร้อยละ 60 เปอร์เซนต์ ความเร็วลม จำนวน 6 คน คิดเป็นร้อยละ 40 เปอร์เซนต์ คุณภาพของอากาศ จำนวน 5 คน คิดเป็นร้อยละ 33.3 เปอร์เซนต์ และเวลาพระอาทิตย์ขึ้น - ตก จำนวน 5 คน คิดเป็นร้อยละ 33.3 เปอร์เซนต์

2.5. ผู้ใช้งานตรวจสอบสภาพอากาศผ่านช่องทาง



ภาพที่ 3. 8 ผู้ใช้งานตรวจสอบสภาพอากาศผ่านช่องทาง

จากภาพพบว่าผู้ตอบแบบสอบถามมีการตรวจสอบสภาพอากาศผ่านช่องทางของโทรศัพท์มือถือ จำนวน 15 คน คิดเป็นร้อยละ 100 เปอร์เซนต์ คอมพิวเตอร์ จำนวน 4 คน คิดเป็นร้อยละ 26.7 เปอร์เซนต์และแท็บเล็ต จำนวน 3 คน คิดเป็นร้อยละ 20 เปอร์เซนต์

3. ข้อมูลสนับสนุนการท่องเที่ยวเชิงนิเวศ โดยจะมีการเก็บรวบรวมข้อมูลดังนี้

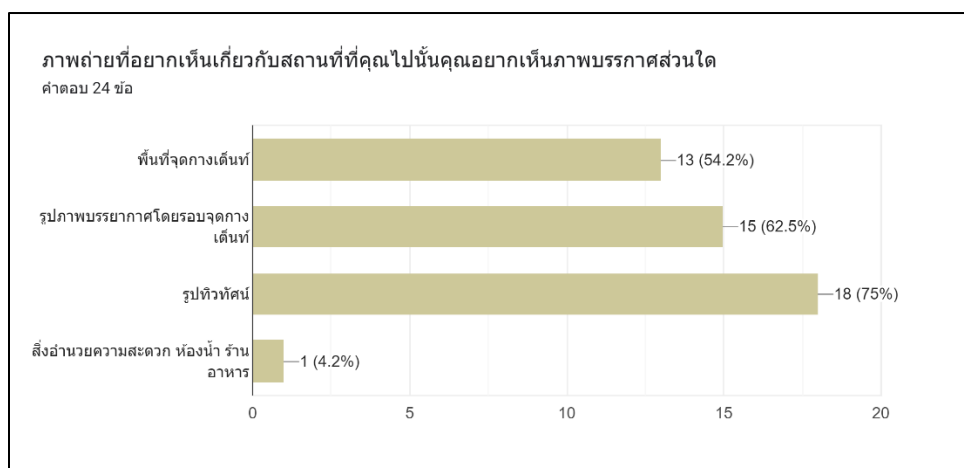
3.1. จำนวนของนักท่องเที่ยวในพื้นที่ที่มีผลต่อการตัดสินใจไปกางเต็นท์หรือไม่



ภาพที่ 3. 9 จำนวนของนักท่องเที่ยวในพื้นที่ที่มีผลต่อการตัดสินใจไปกางเต็นท์หรือไม่

จากภาพพบว่าจำนวนของนักท่องเที่ยวในพื้นที่นั้นมีผลต่อการตัดสินใจในการไปท่องเที่ยว
จำนวน 24 คน คิดเป็นร้อยละ 100 เปอร์เซ็นต์

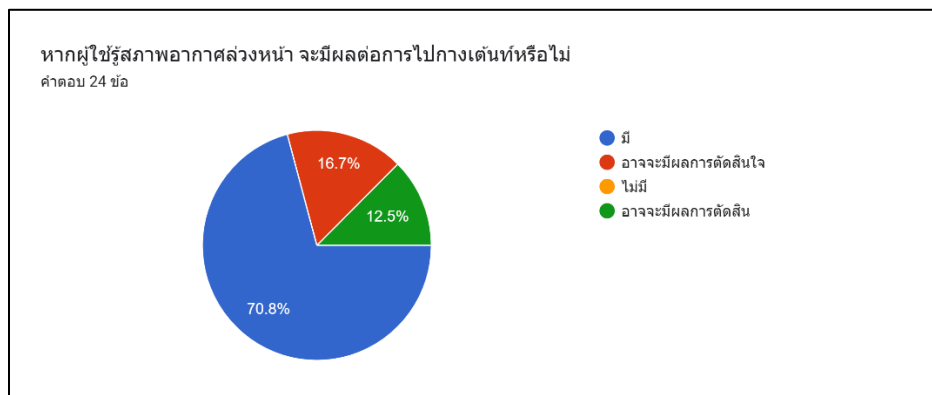
3.2. ภาพถ่ายของบรรยากาศที่นักท่องเที่ยวต้องการเห็น



ภาพที่ 3. 10 ภาพถ่ายของบรรยากาศที่นักท่องเที่ยวต้องการเห็น

จากภาพพบว่าผู้ตอบแบบสอบถามนั้นอยากเห็นภาพถ่ายของสถานที่ท่องเที่ยว ดังนี้ รูปทิวทัศน์
จำนวน 18 คน คิดเป็นร้อยละ 75 เปอร์เซ็นต์ รูปบรรยากาศรอบบริเวณจุดกางเต็นท์ จำนวน 15 คน คิด
เป็นร้อยละ 62.5 เปอร์เซ็นต์และพื้นที่กางเต็นท์ จำนวน 13 คน คิดเป็นร้อยละ 54.2 เปอร์เซ็นต์

3.3. หากผู้ใช้งานทราบสภาพอากาศล่วงหน้า จะส่งผลต่อการตัดสินใจไปกางเต็นท์หรือไม่

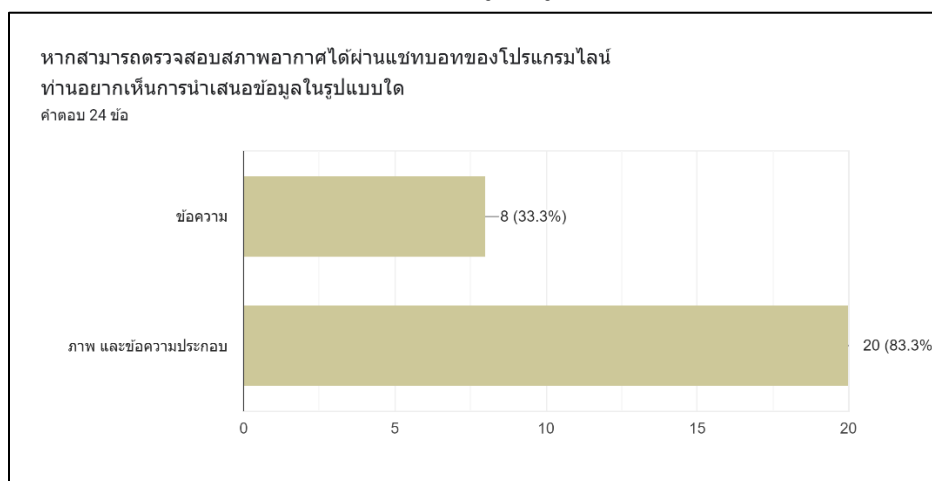


ภาพที่ 3. 11 หากผู้ใช้งานทราบสภาพอากาศล่วงหน้า จะส่งผลต่อการตัดสินใจไปกางเต็นท์หรือไม่

จากภาพพบว่าผู้ที่ตอบแบบสอบถามรู้สภาพอากาศล่วงหน้า นั้นมีผลต่อการไปกางเต็นท์จำนวน 17 คน คิดเป็นร้อยละ 70.8 เปอร์เซ็นต์และอาจจะมีการตัดสินใจการไปกางเต็นท์ จำนวน 7 คน คิดเป็นร้อยละ 29.2 เปอร์เซ็นต์

4. ข้อมูลเพิ่มเติมสำหรับการพัฒนาระบบ โดยจะมีการเก็บรวบรวมข้อมูลดังนี้

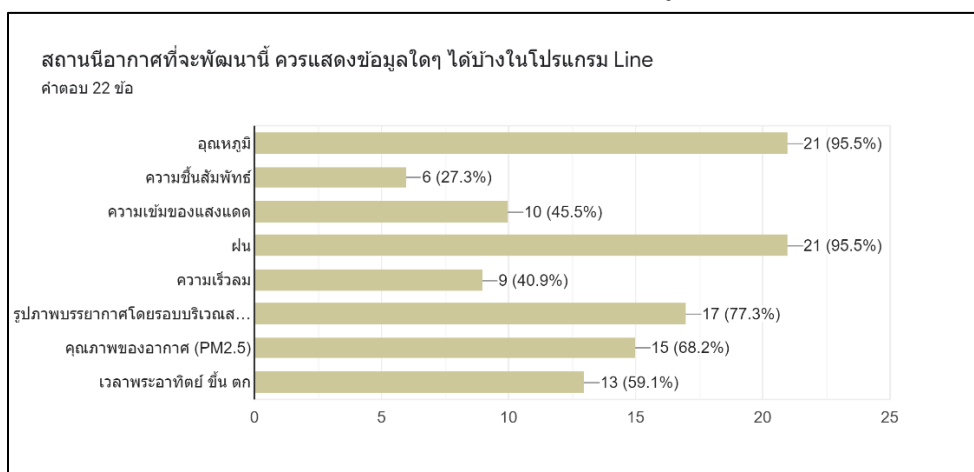
4.1. หากสามารถตรวจสอบสภาพอากาศได้ผ่านช่องทางของโปรแกรมไลน์ ท่านอยากเห็นการนำเสนอข้อมูลในรูปแบบใด



ภาพที่ 3. 12 หากสามารถตรวจสอบสภาพอากาศได้ผ่านช่องทางของโปรแกรมไลน์ ท่านอยากเห็นการนำเสนอข้อมูลในรูปแบบใด

จากภาพพบว่าผู้ที่ตอบแบบสอบถามอยากเห็นการนำเสนอในรูปแบบภาพและข้อความประกอบจำนวน 20 คน คิดเป็นร้อยละ 83.3 เปอร์เซ็นต์และข้อความ จำนวน 4 คน คิดเป็นร้อยละ 12.7 เปอร์เซ็นต์

4.2. สถานีอากาศที่จะพัฒนานี้ ควรแสดงข้อมูลใดๆ ได้บ้างในโปรแกรม Line



ภาพที่ 3. 13 สถานีอากาศที่จะพัฒนานี้ ควรแสดงข้อมูลใดๆ ได้บ้างในโปรแกรม Line

จากภาพพบว่าผู้ตอบแบบสอบถามต้องการทราบข้อมูลสภาพอากาศเพื่อแสดงบนโปรแกรมไลน์ ดังนี้ ฝน จำนวน 21 คน คิดเป็นร้อยละ 95.5 เปอร์เซ็นต์ อุณหภูมิ จำนวน 21 คน คิดเป็นร้อยละ 95.5 เปอร์เซ็นต์ รูปภาพบรรยากาศโดยรอบบริเวณสถานที่ท่องเที่ยว จำนวน 17 คน คิดเป็นร้อยละ 77.3 เปอร์เซ็นต์ คุณภาพของอากาศ จำนวน 15 คน คิดเป็นร้อยละ 68.2 เปอร์เซ็นต์ เวลาพระอาทิตย์ขึ้น - ตก จำนวน 13 คน คิดเป็นร้อยละ 59.1 เปอร์เซ็นต์ ความเข้มของแสงแดด จำนวน 10 คน คิดเป็นร้อยละ 45.5 เปอร์เซ็นต์ ความเร็วลม จำนวน 9 คน คิดเป็นร้อยละ 40.9 เปอร์เซ็นต์และความชื้นสัมพัทธ์ จำนวน 6 คน คิดเป็นร้อยละ 27.3 เปอร์เซ็นต์

4.3. ถ้ามีสถานีอากาศเฉพาะพื้นที่สำหรับพื้นที่ทางเดินเท้า คุณคาดหวังสิ่งใดบ้าง

สิ่งที่ผู้ใช้ต้องการ	จำนวน (คน)	ร้อยละ
ความแม่นยำของข้อมูล	7	29.4
การพยากรณ์สภาพอากาศ	8	33.6
การแจ้งเตือนสภาพอากาศที่มีประสิทธิภาพ	2	8.4
ข้อมูลสภาพอากาศ เช่น ภาพสถานที่ อุณหภูมิ ปริมาณน้ำฝน ความเข้มแสง เวลาพระอาทิตย์ขึ้น-ตก	1	4.2
ไม่แสดงความคิดเห็น	6	25.2

ภาพที่ 3. 14 ถ้ามีสถานีอากาศเฉพาะพื้นที่สำหรับพื้นที่ทางเดินเท้า คุณคาดหวังสิ่งใดบ้าง

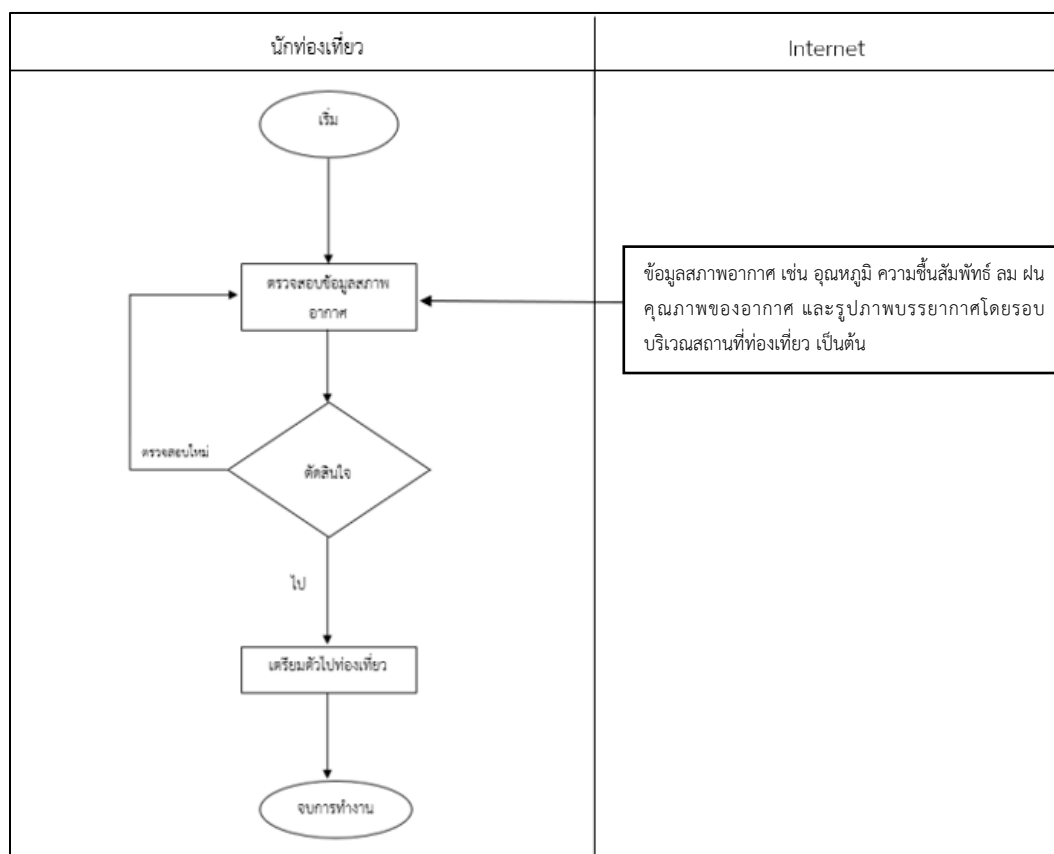
จากตารางพบว่าผู้ตอบแบบสอบถามมีความคาดหวังในสถานีอากาศดังนี้ ความแม่นยำของข้อมูล จำนวน 7 คน คิดเป็นร้อยละ 29.4 เปอร์เซ็นต์ การพยากรณ์สภาพอากาศ จำนวน 8 คน คิดเป็นร้อยละ 33.6 เปอร์เซ็นต์ การแจ้งเตือนสภาพอากาศที่มีประสิทธิภาพ จำนวน 2 คน คิดเป็นร้อยละ 8.4 เปอร์เซ็นต์

ข้อมูลสภาพอากาศ เช่น ภาพสถานที่ อุณหภูมิ ปริมาณน้ำฝน ความเข้มแสง เวลาพระอาทิตย์ขึ้น-ตก จำนวน 1 คน คิดเป็นร้อยละ 4.2 เปอร์เซ็นต์และไม่แสดงความคิดเห็น จำนวน 6 คน คิดเป็นร้อยละ 25.2 เปอร์เซ็นต์

จากการเก็บข้อมูลข้างต้นสรุปได้ว่าผู้ตอบแบบสอบถามนั้นส่วนใหญ่เป็นเพศชายที่อยู่ในช่วงอายุ 19-22 ปี ซึ่งมีการตรวจเช็คสภาพอากาศก่อนออกเดินทางทุกครั้งโดยการสภาพอากาศที่ตรวจสอบอีกครั้งก่อนการตัดสินใจอีกทั้งผู้ตอบแบบสอบถามนั้นมีการตรวจสอบก่อนการเดินทางล่วงหน้า 1 วัน และตรวจสอบสภาพอากาศด้วยมือถือผ่านการสืบค้นจาก Google โดยสภาพอากาศที่ผู้ตอบแบบสอบถามต้องการทราบมากที่สุดคือ ฝน และภาพถ่ายวิวทิวทัศน์ ซึ่งจำนวนของนักท่องเที่ยวและสภาพอากาศในพื้นที่นั้นยังส่งผลต่อการท่องเที่ยวอีกด้วย ซึ่งข้อมูลที่ตรวจสอบได้นั้นผู้ตอบแบบสอบถามต้องการให้มีการนำเสนอในรูปแบบของภาพและข้อความประกอบ โดยข้อมูลที่จะแสดงผ่านโปรแกรมไลน์ที่ผู้ตอบแบบสอบถามต้องการมากที่สุด คือ ฝนและอุณหภูมิ

3.3 การทำงานของระบบปัจจุบัน

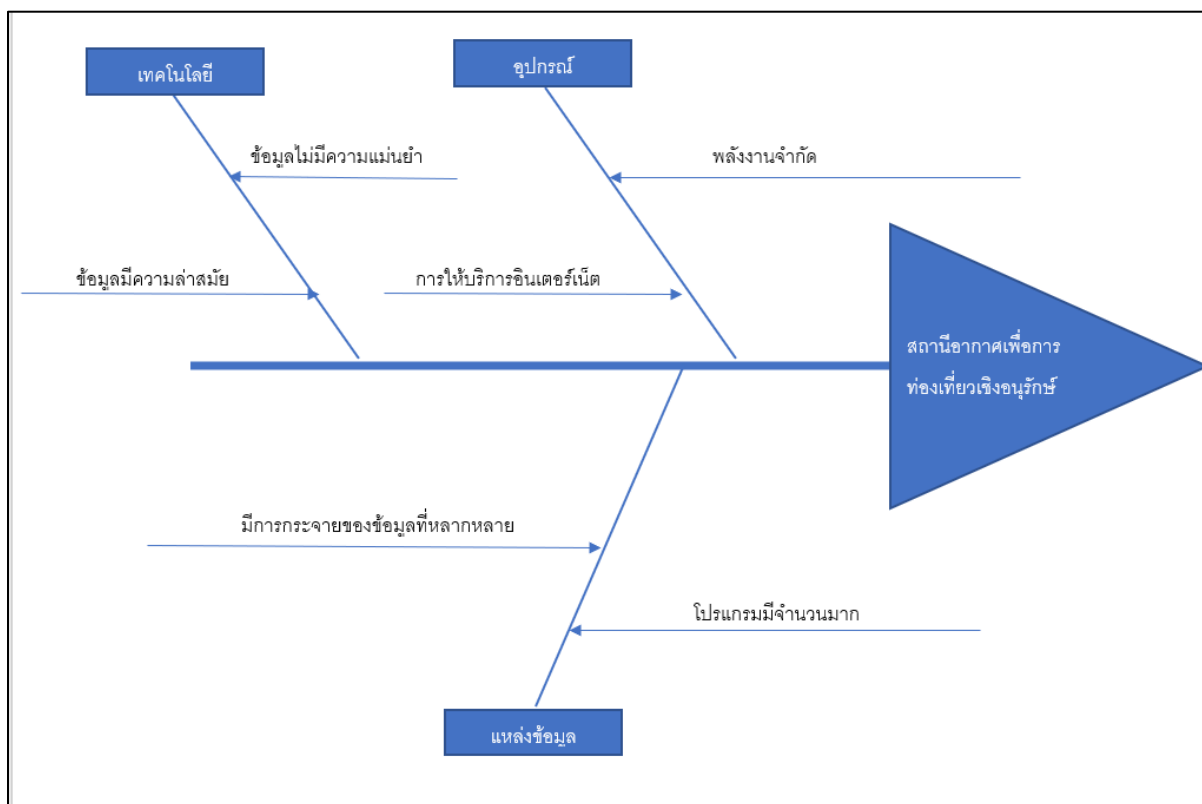
3.3.1 กระบวนการทำงานของระบบปัจจุบัน



ภาพที่ 3. 15 การทำงานของระบบปัจจุบัน

จากภาพที่ 3.15 เริ่มจากการที่นักท่องเที่ยวทำการตรวจสอบข้อมูลสภาพอากาศโดยข้อมูลสภาพอากาศ เช่น อุณหภูมิ ความชื้นสัมพัทธ์ คุณภาพของอากาศ ลม ฝน และรูปภาพบรรยากาศโดยรอบบริเวณสถานที่ท่องเที่ยว เป็นต้น หากสภาพอากาศเหมาะสมที่จะเดินทางก็สามารถเตรียมตัวเดินทางได้แต่หากว่าข้อมูลสภาพอากาศไม่เหมาะสมสำหรับการเดินทางก็ให้นักท่องเที่ยวกลับไปตรวจสอบสภาพอากาศใหม่อีกครั้งเพื่อความเหมาะสมในการเดินทาง

3.3.2 ปัญหาของระบบปัจจุบัน



ภาพที่ 3. 16 ปัญหาของระบบปัจจุบัน

จากภาพทำให้พบปัญหาของงานวิจัย 3 ข้อ คือ

1. ปัญหาจากอุปกรณ์ อุปกรณ์ไม่มีแหล่งเก็บพลังงานสำรอง หากไม่ไฟฟ้าจากแหล่งจ่ายไฟหลัก ตัวอุปกรณ์ไม่สามารถทำงานได้ และหากจะนำไปใช้จำเป็นต้องมีแหล่งพลังงานสำรองหากไม่มีกระแสไฟฟ้าจากแหล่งจ่ายไฟหลัก ปัญหาในส่วนของการให้บริการอินเทอร์เน็ตยังต้องพึ่งพาอินเทอร์เน็ตไร้สายความเร็วสูง (WIFI) ทำให้เมื่อติดตั้งในสถานที่ที่ไม่มี WIFI อุปกรณ์ก็ไม่สามารถส่งข้อมูลได้
2. ปัญหาจากเทคโนโลยี การให้ข้อมูลสภาพอากาศที่ไม่มีความแม่นยำในระดับเจาะจงเฉพาะพื้นที่ เช่น เมื่อนักท่องเที่ยวต้องการทราบสภาพอากาศในสถานที่ทางเดินเท้า สวนพฤกษศาสตร์ มหาวิทยาลัยวลัย

ลักษณะ อำเภอท่าศาลา จังหวัดนครศรีธรรมราช เมื่อนักท่องเที่ยวตรวจสอบสภาพอากาศจะพบว่าระบบจะแสดงเฉพาะสภาพอากาศของอำเภอท่าศาลา จังหวัดนครศรีธรรมราช และข้อมูลที่แสดงจะเป็นข้อมูลย้อนหลังประมาณ 45 นาที

3. ปัญหาจากแหล่งข้อมูล การกระจายของข้อมูลที่หลากหลายแหล่งที่มาทั้งกรมอุตุนิยมวิทยา สถานีอากาศส่วนบุคคล ข้อมูลที่มาจากหลายแหล่งทำให้ได้ข้อมูลที่ไม่ตรงกัน และปัจจุบันมีโปรแกรมที่รายงานสภาพอากาศจำนวนมากทำให้ข้อมูลกระจาย

3.4 การวิเคราะห์ความต้องการของผู้ใช้

3.4.1 User requirement

1. นักท่องเที่ยว
 - นักท่องเที่ยวสามารถพิมพ์เพื่อขอข้อมูลสภาพอากาศและรูปภาพของสถานที่ ณ ปัจจุบันผ่านแชทบอทบนโปรแกรมไลน์
 - สามารถดูข้อมูลสภาพอากาศและรูปภาพผ่านแชทบอทบนโปรแกรมไลน์ซึ่งรองรับทั้งคอมพิวเตอร์และอุปกรณ์เคลื่อนที่
2. ความสามารถของอุปกรณ์
 - ส่งข้อมูลสภาพอากาศและรูปภาพให้แสดงผ่านแชทบอทบนโปรแกรมไลน์

3.4.2 Functional requirement

- แชทบอทบนโปรแกรมไลน์ที่สามารถตอบโต้กับผู้ใช้
- การให้ข้อมูลของสภาพอากาศ เช่น ฝน อุณหภูมิ ความชื้นสัมพัทธ์ ฝน ลม คุณภาพของอากาศ และ รูปภาพบรรยากาศโดยรอบบริเวณสถานที่ท่องเที่ยว เป็นต้น

3.4.3 Non-Functional requirement

- พลังงานที่ใช้สำหรับเลี้ยงสถานีอากาศ
- Internet ที่สามารถเข้าถึงได้ตลอด

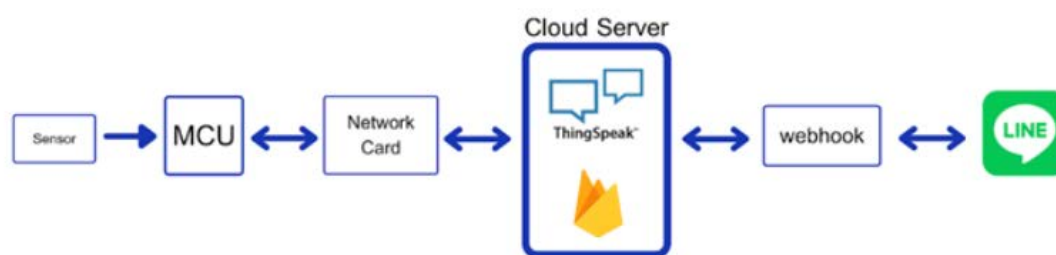
บทที่ 4

การออกแบบระบบ

จากการวิเคราะห์ระบบงานในปัจจุบันและความต้องการของผู้ใช้งาน ทำให้ได้ขอบเขตของความต้องการของระบบสถานีอากาศเพื่อการท่องเที่ยวเชิงอนุรักษ์ซึ่งนำมาใช้ในการออกแบบเพื่อตอบสนองความต้องการของผู้ใช้งาน ผู้พัฒนาได้ออกแบบโดยยึดหลักการทำงานของอุปกรณ์ที่สามารถทำงานได้ต่อเนื่อง และทำการออกแบบระบบให้เหมาะสมกับผู้ใช้งานผ่านสมาร์ทโฟนให้สะดวกมากยิ่งขึ้น การออกแบบระบบประกอบไปด้วยการออกแบบสถาปัตยกรรมระบบทั้งด้านฮาร์ดแวร์และซอฟต์แวร์โดยมีรายละเอียดดังนี้

4.1 การออกแบบสถาปัตยกรรมระบบ

สถาปัตยกรรมระบบในด้านของอุปกรณ์จะประกอบไปด้วย 4 ส่วน



ภาพที่ 4. 1 สถาปัตยกรรมระบบ

4.1.1 NodeMCU ESP8266

มีหน้าที่ควบคุมการทำงานของอุปกรณ์ต่าง ๆ ที่นำมาเชื่อมต่อเข้ากับตัวประมวลผล ซึ่งใน NodeMCU มีอุปกรณ์ต่าง ๆ ดังนี้

1. Sensor เซ็นเซอร์ มีหน้าที่ในการวัดค่าอุณหภูมิ ความชื้น ค่าฝุ่น PM2.5 ค่าความเข้มแสง ในบริเวณที่ทำการติดตั้งสถานีอากาศเพื่อนำค่าหรือข้อมูลที่ได้ส่งให้กับผู้ใช้งาน
2. Network Card มีหน้าที่ทำการเชื่อมต่ออินเทอร์เน็ตเพื่อทำการส่งข้อมูลไปยัง Sensor Cloud storage และ Image Cloud storage ภายในเน็ตเวิร์คการ์ดจะมีชิพการดอยู่ภายใน และมีการจำกัดปริมาณการส่งข้อมูลตามแพ็คเกจที่ได้ลงทะเบียนกับผู้ให้บริการอินเทอร์เน็ต
3. Camera Module มีหน้าที่ถ่ายภาพบรรยากาศรอบบริเวณที่ได้ติดตั้งสถานีอากาศไว้เพื่อเป็นข้อมูลแสดงผลให้ผู้ใช้งาน

4.1.2 Cloud Storage

ทำหน้าที่เก็บข้อมูลที่ NodeMCU ส่งข้อมูลมาเก็บไว้ ซึ่งทางผู้พัฒนาได้เลือกใช้ผู้ให้บริการชื่อ Thing speak เป็นผู้ให้บริการคลาวด์ที่อนุญาตให้เราสามารถส่งค่าต่าง ๆ ขึ้นไปเก็บบนพื้นที่ และยังเปิดให้สามารถเข้าถึงข้อมูลผ่านเว็บเบราว์เซอร์ทั่วไป และ Firebase เป็นผู้ให้บริการคลาวด์ที่ใช้สำหรับเก็บข้อมูลภาพบริการคลาวด์ที่กล่าวมาข้างต้นเป็นผู้ให้บริการที่ไม่มีค่าใช้จ่ายในส่วนของการใช้งานแบบเริ่มต้น และยังให้บริการ API แก่ผู้พัฒนาทั่วไป

4.1.3 Webhook

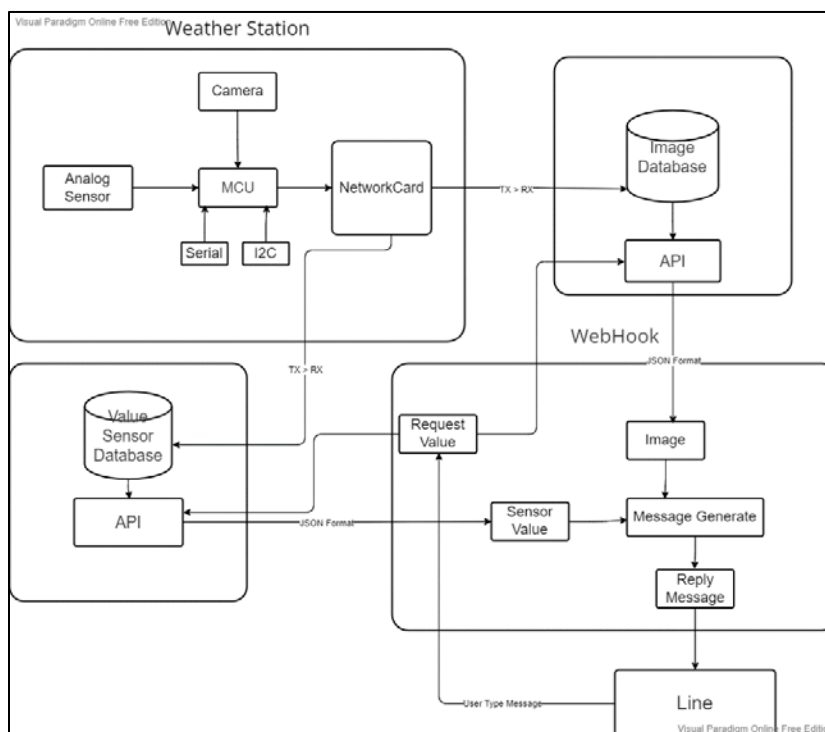
ทำหน้าที่ให้บริการเป็นเครื่องแม่ข่ายที่ให้บริการเชื่อมต่อระหว่าง LineAPI และ Cloud Storage ในการแลกเปลี่ยนข้อมูล

4.1.4 LineAPI

ทำหน้าที่เชื่อมต่อระหว่างผู้ใช้แอปพลิเคชันไลน์ที่ส่งรีควีสข้อมูลจากเครื่องแม่ข่าย

4.2 การออกแบบสถาปัตยกรรมซอฟต์แวร์

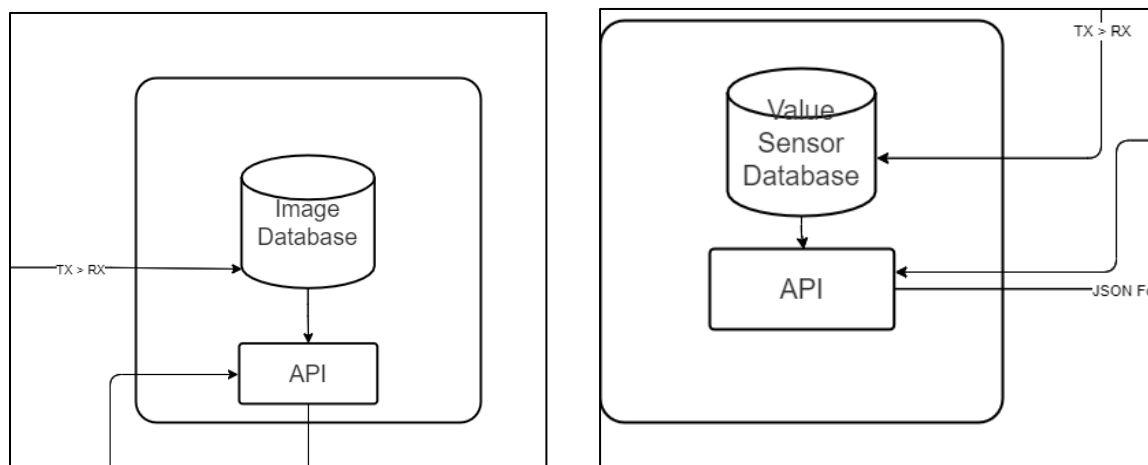
การออกแบบสถาปัตยกรรมซอฟต์แวร์ เป็นการอธิบายภาพรวมของระบบสถาปัตยกรรมเพื่อการท่องเที่ยวเชิงอนุรักษ์ เพื่อให้มองเห็นถึงโครงสร้างส่วนประกอบ ด้านฟังก์ชันและคุณภาพ ซึ่งแสดงให้เห็นถึงผลกระทบ คุณสมบัติ ของการเชื่อมต่อระหว่างองค์ประกอบเหล่านั้นของระบบสถาปัตยกรรมเพื่อการท่องเที่ยวเชิงนิเวศ



ภาพที่ 4. 2 สถาปัตยกรรมระบบ

การทำงานของระบบสถานีอากาศเพื่อการท่องเที่ยวเชิงนิเวศ มีโมดูลหลักในการทำงานของระบบ 5 โมดูล ประกอบไปด้วย โมดูลเซ็นเซอร์, โมดูล Image cloud storage, โมดูล Sensor cloud storage, โมดูล Webhook และโมดูลแอปพลิเคชัน Line โดยในแต่ละโมดูลจะมีหน้าที่การทำงานที่แตกต่างกันไป เริ่มต้นที่โมดูลสถานีอากาศจะแบ่งการทำงานออกเป็น 3 ส่วน คือ เซ็นเซอร์ หน่วยประมวลผล NodeMCU และเน็ตเวิร์คการ์ด ส่วนเซ็นเซอร์ทำหน้าที่ตรวจจับการเปลี่ยนแปลงของสภาพอากาศ ข้อมูลจากเซ็นเซอร์จะอยู่ในรูปแบบของ Analog, I2C และ Serial ข้อมูลเหล่านี้ถูกประมวลผลโดย NodeMCU เพื่อส่งค่าไปยัง Sensor Cloud storage ซึ่งในขณะเดียวกัน ข้อมูลบางส่วนจะไปประมวลผลเข้ากับรูปภาพ โมดูลกล้องทำหน้าที่ถ่ายภาพและนำข้อมูลบางส่วนจาก Sensor cloud storage มาประกอบเข้ากับรูปภาพบน Image cloud storage โมดูล Webhook ทำหน้าที่ข้อมูลต่าง ๆ ที่เกิดขึ้นกับ Line chat bot ที่ผู้ใช้ได้ร้องขอเข้ามา โมดูล Webhook จะประมวลผลข้อมูลที่ได้รับมา หากข้อมูลตรงกับข้อมูลที่โมดูล Webhook จะนำข้อมูลจาก Cloud storage โดยใช้ API ในการขอข้อมูล Cloud storage จะแปลงข้อมูลให้อยู่ในรูปแบบของ JSON format และตอบกลับไปยัง โมดูล Line

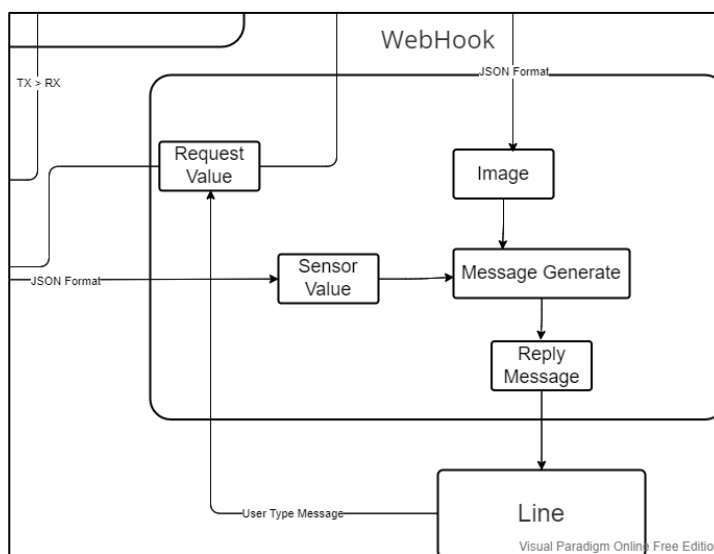
4.2.1 โมดูล Cloud storage



ภาพที่ 4. 3 โมดูล Cloud storage

ในการอัปโหลดรูปจากสถานีอากาศไปยัง Image cloud storage ในส่วนของ Image cloud storage จะรับข้อมูลภาพและทำการเก็บบันทึก ข้อมูลที่เก็บจะอยู่ในรูปแบบของ URL เพื่อให้การเข้าถึงรูปภาพทำได้ง่าย

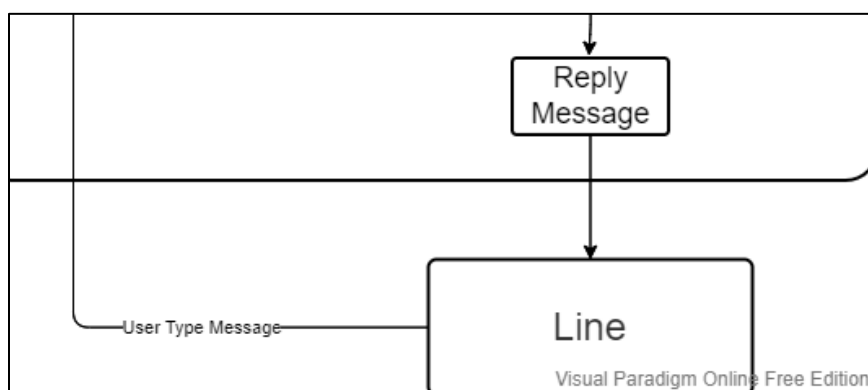
4.2.2 โมดูล Webhook



ภาพที่ 4. 4 โมดูล Webhook

โมดูล Webhook ทำหน้าที่รับประมวลผล รับและส่งข้อมูลให้กับโมดูลที่ได้ทำการเชื่อมต่อไว้ ได้แก่ โมดูล Cloud storage ทั้งโมดูลเซ็นเซอร์ โมดูลรูปภาพ และโมดูลแอปพลิเคชันไลน์ เมื่อ Webhook ได้รับคำร้องขอข้อมูลจากโมดูลไลน์ จะทำการตรวจสอบข้อมูลที่คำร้องขอนั้นถูกต้องตามเงื่อนไขที่กำหนดไว้ หากข้อมูลที่ร้องขอมาถูกต้องตามเงื่อนไข โมดูล Webhook จะทำการตอบกลับข้อมูลตามที่ได้กำหนดเอาไว้ หากไม่ตรงตามเงื่อนไขก็จะตอบกลับข้อมูลในอีกรูปแบบตามที่ได้กำหนด เช่น หากผู้ใช้เลือกข้อมูลอุณหภูมิบน Rich menu โมดูล Webhook จะร้องขอข้อมูลจากโมดูลเซ็นเซอร์ผ่านทาง API ของผู้ให้บริการคลาวด์ ข้อมูลจะส่งมาในรูปแบบของ JSON หลังจากนั้น โมดูล Webhook จะแปลงข้อมูลในรูปแบบที่ผู้ใช้เข้าใจง่ายและแสดงผลให้ผู้ใช้ผ่านโมดูลไลน์

4.2.4 โมดูลแอปพลิเคชันไลน์



ภาพที่ 4. 5 โมดูลแอปพลิเคชันไลน์

ทำหน้าที่เชื่อมต่อระหว่างผู้ใช้แอปพลิเคชันและ Webhook เพื่อทำการแสดงผลข้อมูลให้กับผู้ใช้ผ่านทาง API

4.3 การออกแบบโครงสร้างข้อมูล

4.3.1 ตัวอย่างข้อมูลของเซ็นเซอร์รูปแบบ JSON ของ Thing speak

```
{
  "channel": {
    "id": 1461270,
    "name": "BME280",
    "latitude": "0.0",
    "longitude": "0.0",
    "field1": "temperatureC",
    "field2": "humidity",
    "field3": "pressure",
    "field4": "LDR",
    "created_at": "2021-08-01T16:03:01Z",
    "updated_at": "2021-12-20T09:33:13Z",
    "last_entry_id": 635,
    "feeds": [
      {
        "created_at": "2021-12-21T06:09:06Z",
        "entry_id": 634,
        "field1": "27.73000"
      },
      {
        "created_at": "2021-12-23T04:10:21Z",
        "entry_id": 635,
        "field1": "0"
      }
    ]
  }
}
```

4.3.1 Data Dictionary

ตารางที่ 4.1 Data Dictionary

ชื่อ Attribute	คำอธิบาย	ข้อมูล	ขนาด	Null	Key	ตัวอย่าง
ID	id ของช่องสัญญาณ	VARCHAR	7	N	PK	1461270
name	ชื่อของช่องสัญญาณ	VARCHAR	81	N		BME280
field	พื้นที่ที่ใช้แสดงค่า	VARCHAR	81	N		Pressure

4.4 วางแผนการทดสอบ

4.4.1 Unit Test

Unit Test คือ การทดสอบส่วนย่อยของ Method ที่เขียนว่าสามารถใช้งานได้ตามวัตถุประสงค์ของระบบที่ระบุไว้หรือไม่ โดยใช้ Test case มาทดสอบในแต่ละขั้นตอนเพื่อความพร้อมก่อนจะนำระบบจริงไปใช้งานจริง

ตารางที่ 4. 2 การทดสอบการทำงานของชุดคำสั่งข้อความ

ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	กดเมนูหรือพิมพ์ อุณหภูมิ	แสดงข้อมูลอุณหภูมิ		
2	กดเมนูหรือพิมพ์ ความชื้นสัมพัทธ์	แสดงข้อมูลความชื้นสัมพัทธ์		
3	กดเมนูหรือพิมพ์ คุณภาพอากาศ	แสดงข้อมูลคุณภาพอากาศ		
4	กดเมนูหรือพิมพ์ ฝน	แสดงข้อมูลฝน		
5	กดเมนูหรือพิมพ์ ลม	แสดงข้อมูลลม		
6	กดเมนูหรือพิมพ์ ภาพรวม	แสดงภาพรวมทั้งหมดของระบบ		

ตารางที่ 4. 3 การทดสอบการทำงานของ Cloud Storage

ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	ข้อมูลมีการเปลี่ยนแปลง	ข้อมูลอัปเดต		
2	ข้อมูลผิดปกติ	ข้อมูลอัปเดต		

ตารางที่ 4. 4 การทดสอบการทำงานของ Cloud Image Storage

ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	การอัปโหลดข้อมูล	ข้อมูลอัปโหลด		
2	การตอบกลับการอัปโหลด	ข้อมูลมีการตอบกลับ		

ตารางที่ 4. 5 การทดสอบการใช้พลังงานสะสม

ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	ทดสอบการเปิดระบบทิ้งไว้โดยไม่มีการรันโปรแกรม เพื่อทดสอบการใช้พลังงานของระบบ และใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 21 มิลลิแอมป์		
2	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมแสดงผลข้อมูลทุก ๆ 1 วินาทีผ่าน Serial Monitor ใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 116 มิลลิแอมป์		
3	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมส่งข้อมูลทุก ๆ 15 วินาที ใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 390 มิลลิแอมป์		
4	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมส่งข้อมูลทุก ๆ 1 นาที โดยจะเพิ่มฟังก์ชัน Deep Sleep ใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 184 มิลลิแอมป์		
5	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมส่งข้อมูลทุก ๆ 5 นาที โดยจะเพิ่มฟังก์ชัน Deep Sleep ใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 152 มิลลิแอมป์		

4.5 การออกแบบส่วนการใช้งาน (Input Design)

ตารางที่ 4. 6 การออกแบบส่วนการใช้งาน

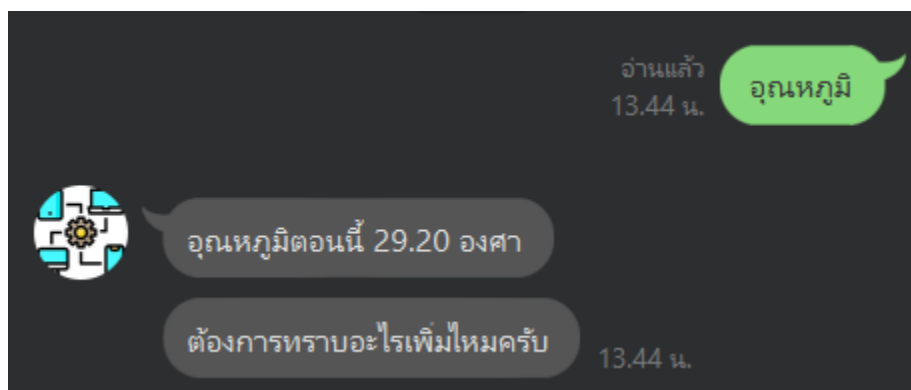
ลำดับ	ชุดคำสั่ง	ผลลัพธ์
1	อุณหภูมิ	แสดงข้อมูลอุณหภูมิ
2	ความชื้นสัมพัทธ์	แสดงข้อมูลความชื้นสัมพัทธ์
3	คุณภาพอากาศ	แสดงข้อมูลคุณภาพอากาศ
4	ฝน	แสดงข้อมูลฝน
5	ลม	แสดงข้อมูลลม
6	ภาพรวม	แสดงภาพรวมทั้งหมดของระบบ



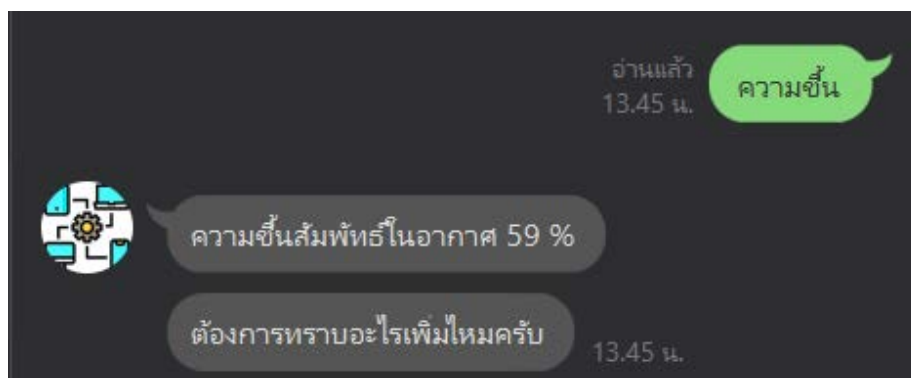
ภาพที่ 4. 6 หน้าจอผลการทดสอบคำสั่งขอข้อมูลคำสั่ง

- อุณหภูมิ เป็นค่าที่จะแสดงข้อมูลของอุณหภูมิเมื่อมีการกดหรือพิมพ์
- ความชื้นสัมพัทธ์ เป็นค่าที่จะแสดงข้อมูลของความชื้นสัมพัทธ์ในอากาศเมื่อมีการกดหรือพิมพ์
- คุณภาพอากาศ เป็นค่าที่จะแสดงข้อมูลของคุณภาพอากาศเมื่อมีการกดหรือพิมพ์
- ฝน เป็นค่าที่จะแสดงข้อมูลของฝนเมื่อมีการกดหรือพิมพ์
- ลม เป็นค่าที่จะแสดงข้อมูลของลมเมื่อมีการกดหรือพิมพ์
- ภาพรวม เป็นค่าที่จะแสดงข้อมูลของภาพรวมระบบซึ่งจะประกอบไปด้วยข้อมูล เช่น รูปภาพสถานที่ อุณหภูมิ ความชื้นสัมพัทธ์ คุณภาพอากาศ ฝน และลมเมื่อมีการกดหรือพิมพ์

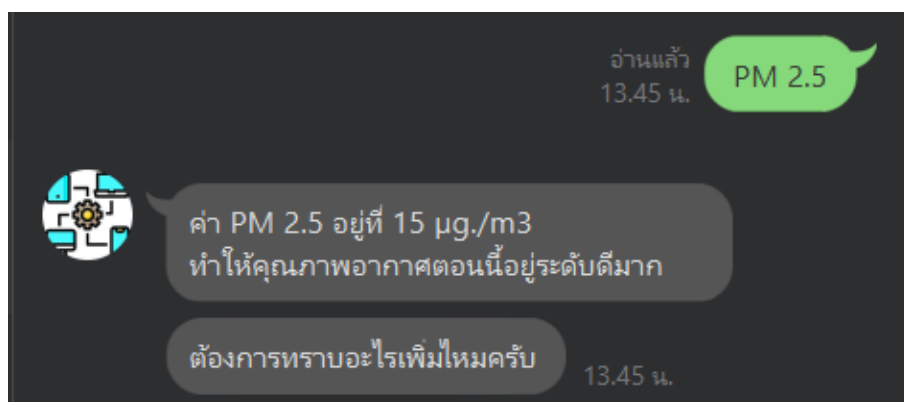
4.6 การออกแบบส่วนแสดงผล (Output Design)



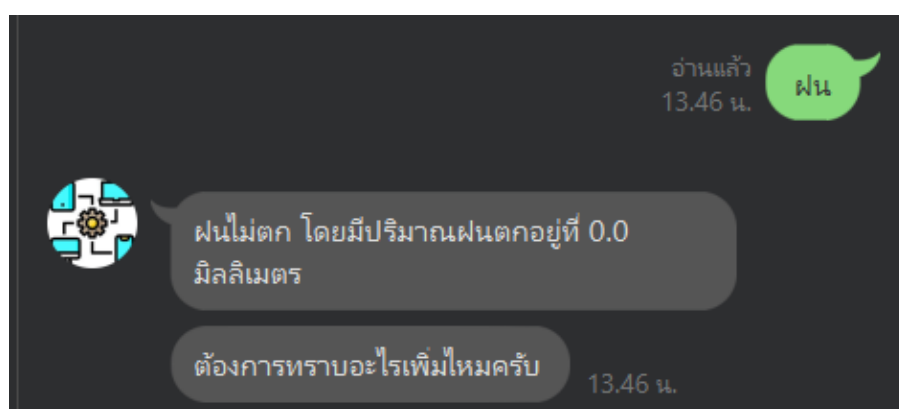
ภาพที่ 4. 7 หน้าจอแสดงผลข้อมูลอุณหภูมิ



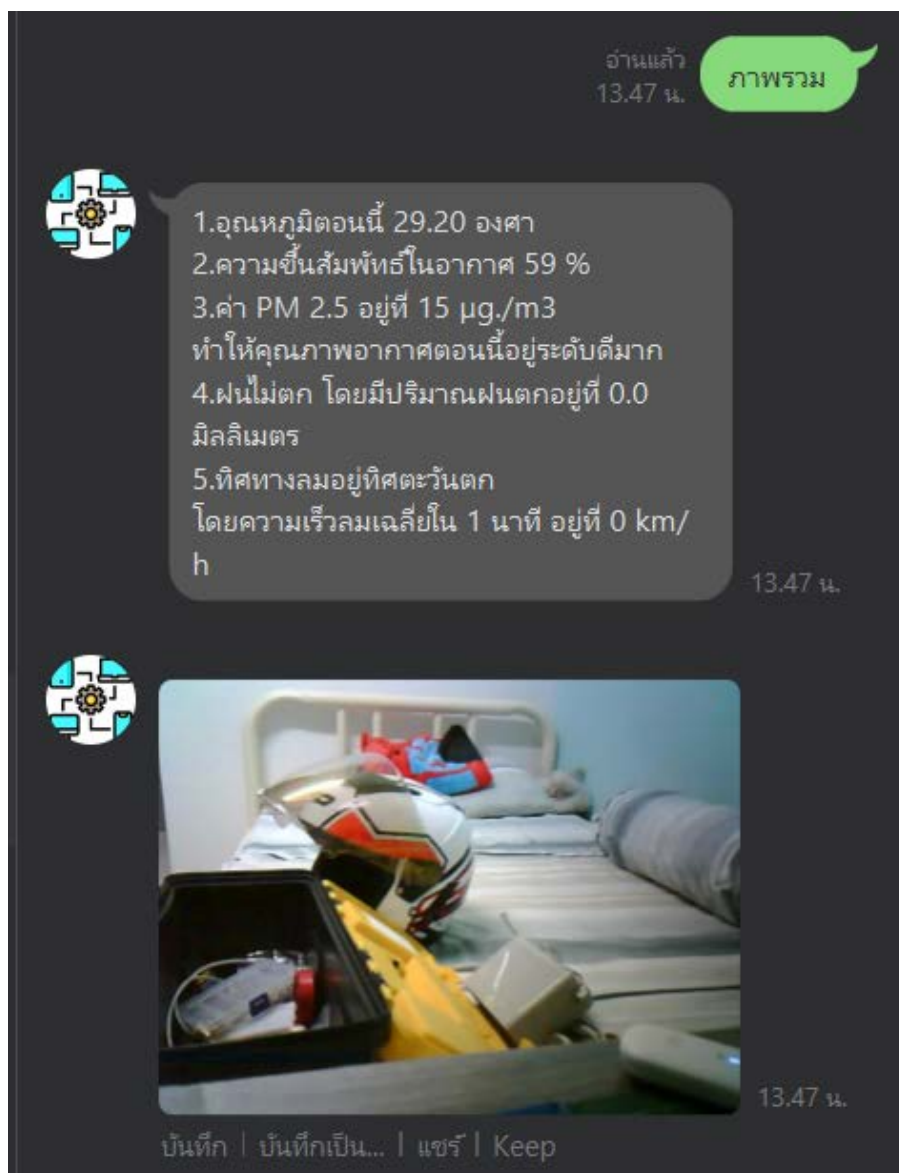
ภาพที่ 4. 8 หน้าจอแสดงผลข้อมูลความชื้นสัมพัทธ์



ภาพที่ 4. 9 หน้าจอแสดงผลข้อมูลคุณภาพอากาศ



ภาพที่ 4. 10 หน้าจอแสดงผลข้อมูลฝุ่น



ภาพที่ 4. 11 หน้าจอแสดงผลภาพรวมของระบบ

จากภาพที่ 4.12 เป็นภาพหน้าจอแสดงผลภาพรวมของระบบ โดยหน้าจอแสดงผลจะประกอบไปด้วยรูปภาพสถานที่และข้อมูลสภาพอากาศแวดล้อมบริเวณที่ตั้งสถานีอากาศ โดยข้อมูลสภาพอากาศจะประกอบไปด้วย ดังนี้

1. แสดงข้อมูลอุณหภูมิ
2. แสดงข้อมูลความชื้นสัมพัทธ์
3. แสดงข้อมูลคุณภาพอากาศ
4. แสดงข้อมูลฝน
5. แสดงข้อมูลลม
6. แสดงรูปภาพสถานที่ที่ติดตั้งระบบ

บทที่ 5

การพัฒนาและทดสอบระบบ

การพัฒนาติดตั้งและทดสอบระบบรวมไปถึงการเขียนโปรแกรมควบคุมการทำงาน การติดตั้งโปรแกรม การทดสอบโปรแกรม การจัดทำเอกสารประกอบการทำระบบ และเอกสารประกอบขั้นตอนการทดสอบระบบ เพื่อตรวจสอบหาข้อผิดพลาดของโปรแกรมก่อนจะนำระบบไปติดตั้งและใช้งานจริงเพื่อป้องกันปัญหาที่จะเกิดขึ้นหรือลดปัญหาให้เกิดขึ้นน้อยที่สุด เพื่อให้ระบบมีความสมบูรณ์มากที่สุด

5.1 การพัฒนาระบบ

5.1.1 อธิบายไลบรารีที่ใช้ในการพัฒนาระบบ

ตารางที่ 5. 1 ไลบรารีที่ใช้ในการพัฒนา

ชื่อไลบรารี	คำอธิบาย
ESP8266WiFi.h	การเรียกใช้โมดูล ESP8266WiFi บนบอร์ด NodeMCU เพื่อทำการเชื่อมต่อ Wi-Fi ตัวอย่างการเรียกใช้งานไลบรารี <code>#include <ESP8266WiFi.h></code> ฟังก์ชันอื่น ๆ เพิ่มเติม <code>WiFi.begin("network-name", "password");</code> เริ่มเชื่อมต่อกับเน็ตเวิร์คที่ได้ประกาศค่าตัวแปร <code>while(WiFi.status() != WL_CONNECTED)</code> จนกว่าจะสามารถเชื่อมต่อได้
PMS	การเรียกใช้ไลบรารีของตัวเซ็นเซอร์ตรวจจับฝุ่นละออง PM2.5 PM10 รองรับเซ็นเซอร์ซีรีส์ PMSx003 sensors (1003, 3003, 5003, 6003, 7003) ตัวอย่างการเรียกใช้งานไลบรารี <code>#include <PMserial.h></code> หลังจากนั้นต้องประกาศพอร์ตที่จะเชื่อมต่อ <code>SerialPM pms(PMSx003, Serial);</code> ในกรณีนี้จะเชื่อมต่อผ่าน TX และ RX บนบอร์ด
SoftwareSerial	ไลบรารี SoftwareSerial ได้รับการพัฒนาเพื่อให้บอร์ด Arduino สามารถสื่อสารแบบอนุกรม(TX, RX) ต่อกับพินดิจิตอลอื่น ๆ บนบอร์ด โดยใช้คำสั่งในการเรียกใช้งาน <code>#include <SoftwareSerial.h></code> หลังจากนั้นประกาศตัวแปรสำหรับพินที่ทำการเชื่อมต่อ ตัวอย่างเช่น <pre>const byte rxPin = 2; const byte txPin = 3; SoftwareSerial mySerial (rxPin, txPin);</pre>

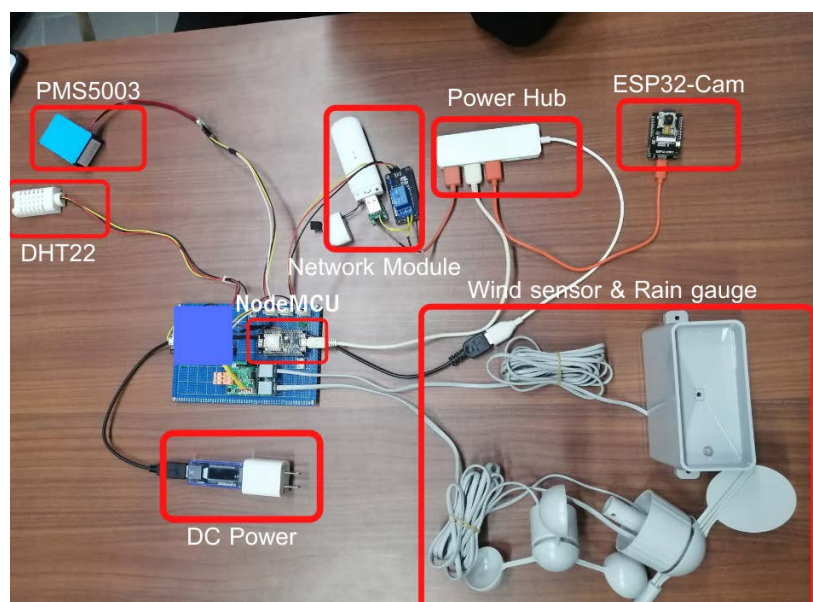
<p>ThingSpeak</p>	<p>ThingSpeak คือผู้ให้บริการระบบคลาวด์เก็บข้อมูลและสามารถเข้าถึงข้อมูลได้ผ่านระบบอินเทอร์เน็ต ได้รับการพัฒนาบนบอร์ด Arduino เพื่อให้สามารถเขียนหรืออ่านข้อมูลไปยัง ThingSpeak ด้วยการวิเคราะห์และการแสดงภาพ MATLAB ตัวอย่างการเรียกใช้งานไลบรารี</p> <pre>#include "ThingSpeak.h"</pre> <p>ThingSpeak จำเป็นต้องมี Channel Number และ API สำหรับการเขียนข้อมูลลงบน fields</p> <pre>unsigned long myChannelNumber = ;</pre> <pre>const char * myWriteAPIKey = ;</pre> <p>เริ่มต้นการใช้งาน ThingSpeak</p> <pre>ThingSpeak.begin(client);</pre> <p>อ่านค่าและทำการเซต Field ที่ทำการเขียน</p> <pre>ThingSpeak.setField(fields, value);</pre> <p>เขียนข้อมูลลงบน Field</p> <pre>ThingSpeak.writeFields(ChannelNumber, APIkeys);</pre>
<p>DHT</p>	<p>ไลบรารีสำหรับเซ็นเซอร์ตรวจจับค่าอุณหภูมิและความชื้นที่อยู่ในอากาศ รองรับการใช้งานหลากหลายตัว เช่น DHT11, DHT22, DHT21</p> <p>ตัวอย่างการใช้งาน</p> <p>เรียกใช้งานไลบรารี</p> <pre>#include "DHT.h"</pre> <pre>#define DHTPIN 2 // เลือกพินดิจิทัลที่เชื่อมต่อบนบอร์ด</pre> <pre>#define DHTTYPE DHT22 // เลือกประเภทเซ็นเซอร์</pre> <pre>DHT dht(DHTPIN, DHTTYPE); // เริ่มการใช้งาน</pre> <pre>dht.begin();</pre> <pre>dht.readTemperature(); // อ่านค่าอุณหภูมิจากเซ็นเซอร์</pre> <p>และสามารถเขียนการดักจับข้อผิดพลาดหากไม่สามารถอ่านข้อมูลจากเซ็นเซอร์ได้</p> <pre>If (isnan(value)){</pre> <pre>Serial.print("Failed to read");</pre> <pre>}</pre>

5.2 การติดตั้งระบบ

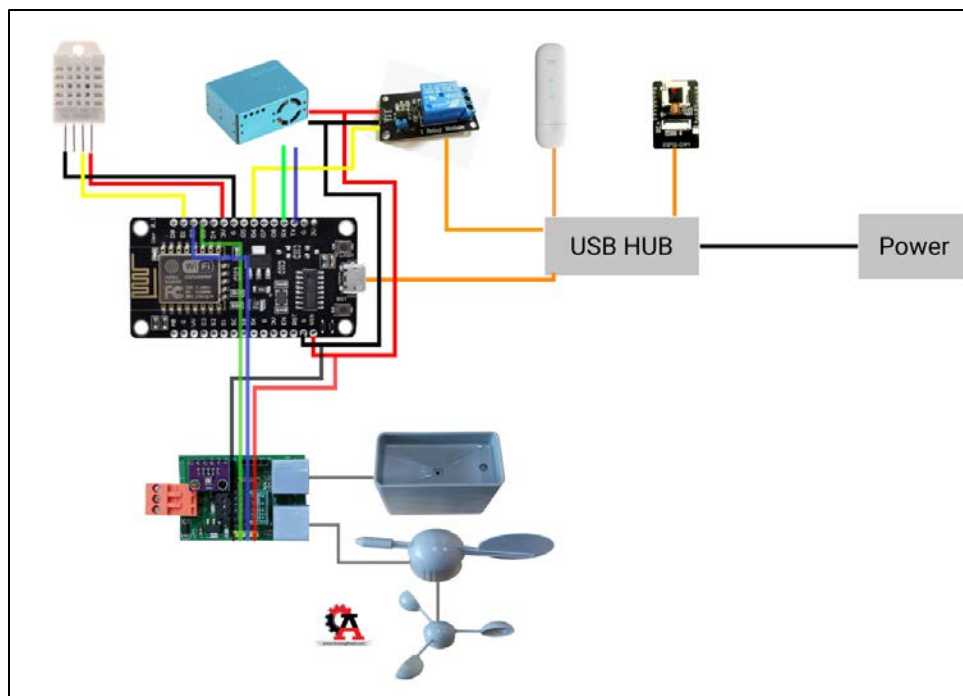
5.2.1 โมดูลสถานีอากาศ



ภาพที่ 5. 1 ภาพรวมสถานีอากาศ



ภาพที่ 5. 2 การเชื่อมต่อภายในสถานีอากาศ



ภาพที่ 5. 3 Wiring Diagram

5.2.2 โมดูล Webhook

ส่วนของโมดูล Webhook จะใช้การให้บริการของ Heroku เนื่องจากเป็นการให้บริการเว็บเซิร์ฟเวอร์ที่ไม่มีค่าบริการ

ก. ติดตั้ง Heroku

า. เริ่มโดยการสร้างแอปพลิเคชันใหม่

The screenshot shows the 'Create New App' form on the Heroku website. The form includes the following fields and buttons:

- App name:** A text input field with the value 'weather-' and a dropdown arrow on the right.
- Choose a region:** A dropdown menu showing 'United States' with a flag icon on the left and a dropdown arrow on the right.
- Add to pipeline...** A button below the region dropdown.
- Create app** A purple button at the bottom of the form.

ภาพที่ 5. 4 สร้างแอปพลิเคชันใหม่

- ตั้งชื่อ App ที่ใช้งาน หรือให้ระบบสุ่มตั้งชื่อให้
- เลือกที่ตั้งของเซิร์ฟเวอร์ให้เป็น United States

ข. สร้าง GitHub และสร้างไฟล์ PHP

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * ProjectWeater **Repository name *** Chatbot-IoT

Great repository names are short and memorable. Need inspiration? How about [didactic-eureka?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository

ProjectWeater / Chat-IoT Public

< Code Issues Pull requests Actions Projects Wiki Security Insights Settings

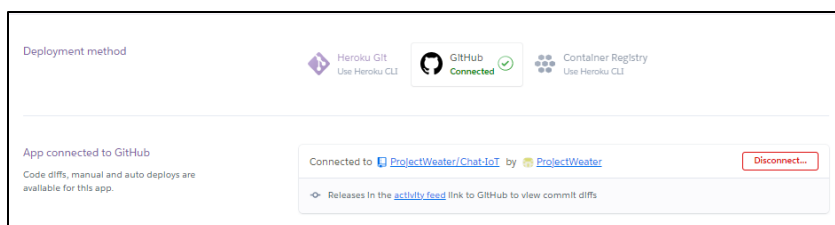
Chat-IoT / index.php in main

Cancel changes

```
< Edit file Preview changes
1 <?php
2 /*Get Data from POST http Request*/
3 $datas = file_get_contents('php://input');
4 /*Decode json from LINE Data Body*/
5 $decode = json_decode($datas,true);
6
7
8 file_put_contents('log.txt', file_get_contents('php://input') . PHP_EOL, FILE_APPEND);
9
10 $replyToken = $decode['events'][0]['replyToken'];
11 $recv_msg = $decode['events'][0]['message']['text'];
12
```

ภาพที่ 5. 5 การสร้าง GitHub และสร้างไฟล์ PHP

- ตรวจสอบการเชื่อมต่อระหว่าง GitHub และ Heroku โดยใช้ชื่อ Repository ที่ตรงกัน



ภาพที่ 5. 6 การเชื่อมต่อระหว่าง GitHub และ Heroku

b. การนำ LINE API ไปใส่ยังไฟล์ index.php เพื่อร้องขอใช้บริการ

```
$LINEData['url'] = "https://api.line.me/v2/bot/message/reply";
$LINEData['token'] = "VwMkO5S39Ih00MeaM5SASHLT8V7G3MC1aFeMy3HI19FP28GZz8v/K2LpDhqmjMhuhZzUPLMe4sJG0cjLZAm2ofYv8/dtH8ILQPGaUeQgQMTdw35+o8Zb07yDg1qu7AYw";
$results = sendMessage($encodeJson,$LINEData);

/*Return HTTP Request 200*/
http_response_code(200);
```

ภาพที่ 5. 7 ไฟล์ PHP

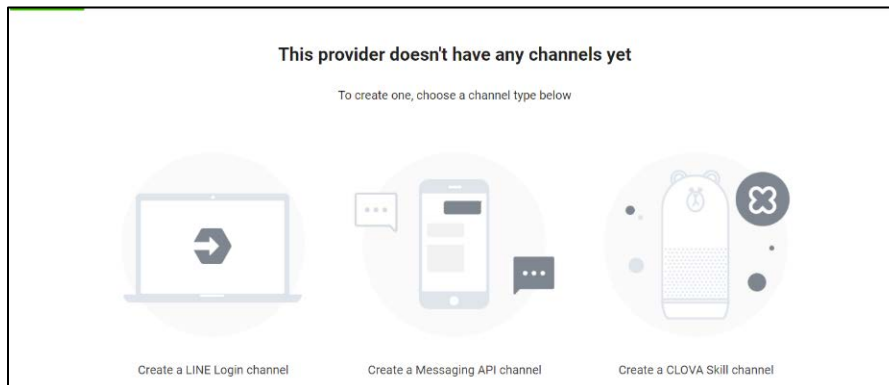
5.2.3 โมดูล Line Bot

โปรแกรมคอมพิวเตอร์ที่มีไว้สื่อสารโดยการสนทนากับมนุษย์แบบอัตโนมัติ

ก. สร้างบัญชี Line Developer

a. การสร้างบัญชี Line Developer

1. เลือกการสร้างบัญชีแบบ Messaging API



ภาพที่ 5. 8 เลือกการสร้างบัญชีแบบ Messaging API

2.กรอกรายละเอียดของ Provider

Create a new channel

Channel type: Messaging API (Don't leave this empty)

Provider: TESTtttt (Don't leave this empty)

Company or owner's country or region: Not set (Don't leave this empty)

ภาพที่ 5. 9 กรอกรายละเอียดของ Provider

3.ตั้งค่า Line Developer ให้เชื่อมต่อกับ Webhook ของ Heroku โดยการนำ Link มาใส่ในช่อง Webhook

Webhook settings

Webhook URL: <https://chat-bot-iot.herokuapp.com/>

Buttons: Verify, Edit

ภาพที่ 5. 10 ตั้งค่า Line Developer ให้เชื่อมต่อกับ Webhook ของ Heroku

4.นำ Access Token ของ Line Developer ที่ได้ไปไว้ในไฟล์ index.php

Channel access token

Channel access token (long-lived)

Issue

```
$LINEData["url"] = "https://api.line.me/v2/bot/message/reply";
$LINEData["token"] = "VwK05539110NEvH5SASHL7B7G3HC1aFeMky3H119FP28GZz8v/K2Lp0HqgeJdhuwZzUHLMe4s3G0cJLZAm2ofyyv8/dtH8ELQ6u6uqg0HTdu35+o8Zb07y0g1ou7AYu5rKb9HOj2vuu/sg8";
$results = sendMessage($sendCode,$son,$LINEData);

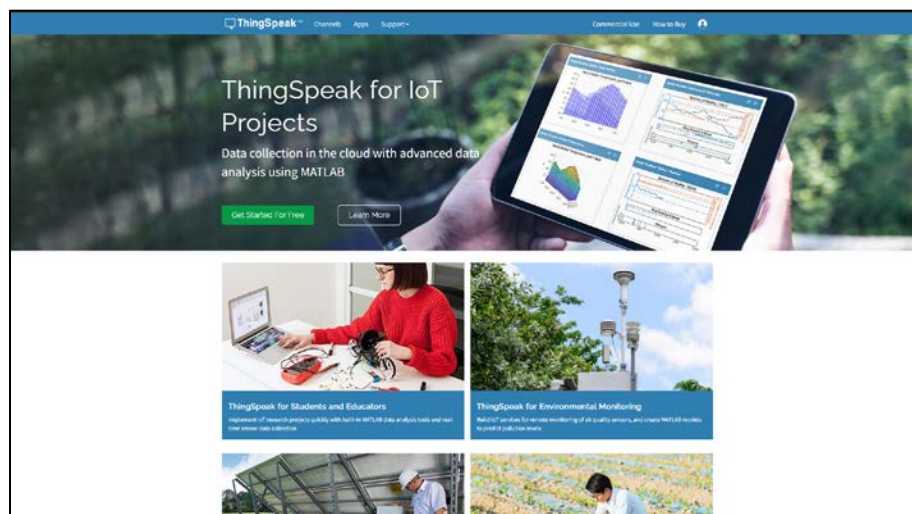
/*Return HTTP Request 200*/
http_response_code(200);
```

ภาพที่ 5. 11 นำ Access Token ของ Line Developer ที่ได้ไปไว้ในไฟล์ index.php

5.2.4 โมดูลการเก็บข้อมูลจากเซ็นเซอร์

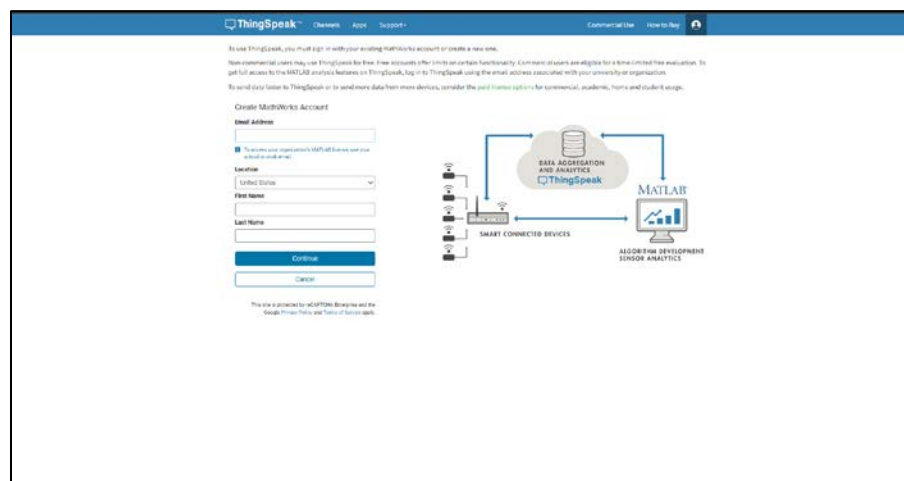
ก. ขั้นตอนการใช้งาน Thing Speak

1. สมัครสมาชิกหรือในกรณีที่สมัครสมาชิกแล้วให้กดที่ Sign In



ภาพที่ 5. 12 เว็บไซต์ Thing Speak

2. กรอกข้อมูลเพื่อสมัครสมาชิก



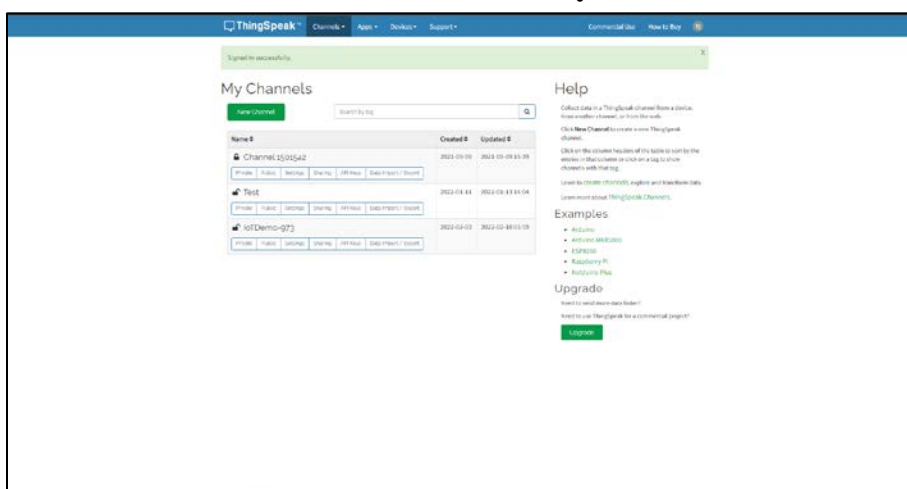
ภาพที่ 5. 13 กรอกข้อมูลเพื่อสมัครสมาชิก

3. กรอก Username และ Password



ภาพที่ 5. 14 กรอก Username และ Password

4. สร้าง Channels เพื่อรองรับข้อมูล



ภาพที่ 5. 15 สร้าง Channels เพื่อรองรับข้อมูล

5. กรอกข้อมูลรายละเอียดต่าง ๆ และกำหนดชื่อ Field แต่ละ Field เพื่อระบุให้ตรงกับข้อมูล

ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy

New Channel

Name:

Description:

Field 1: ☐

Field 2: ☐

Field 3: ☐

Field 4: ☐

Field 5: ☐

Field 6: ☐

Field 7: ☐

Field 8: ☐

Metadata:

Tags:

Link to External Site:

Link to GitHub:

Location:

Latitude:

Longitude:

Show Video: ☐

Help

Channels store all the data that a ThingSpeak app sends to them. Each channel includes eight fields that can hold any type of data, plus they hold the upload date and size for each data item. Since you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, and tags to complete your channel.
- Channel Name:** Give a name to your channel for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that describe the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
 - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is 0.1278.
 - Location:** Specify the location location name. For example, the location of London is 51.5072.
- Website:** If you have a "http://" or "https://" address that displays your channel information, specify the full path of the address (URL).
- Link to GitHub:** If you have your ThingSpeak code on GitHub, specify the GitHub repository URL.

Using the Channel

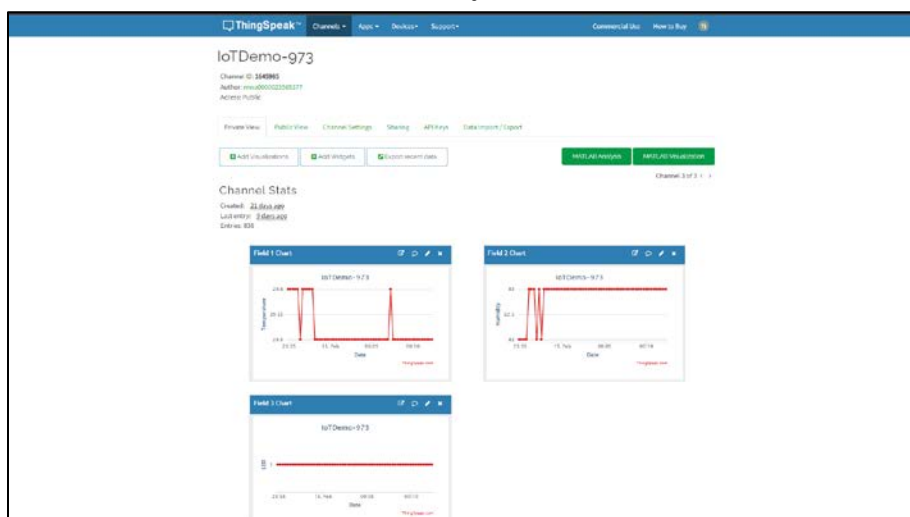
You can get data from a channel from a website, mobile, or another ThingSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

See Get Data with ThingSpeak for an example of measuring temperature in a weather station that displays data from an Arduino sensor.

[Learn More](#)

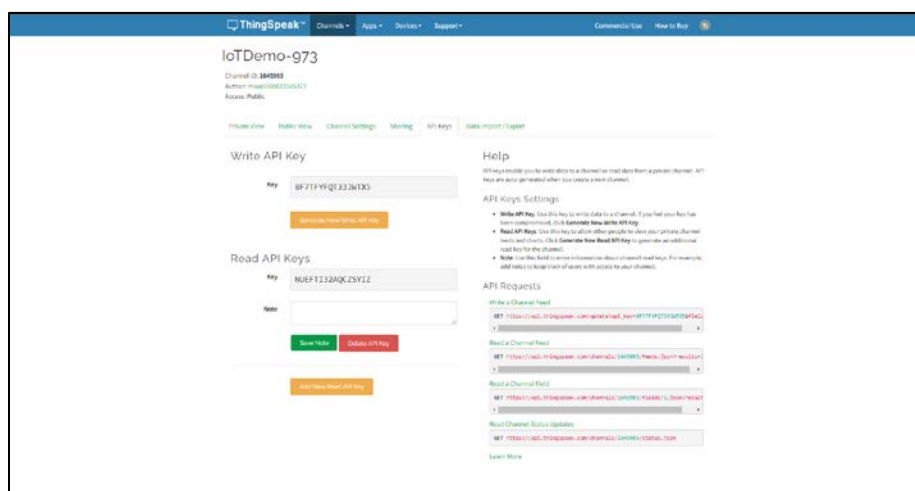
ภาพที่ 5. 16 หน้าจอ New Channels

6. Channels การรับข้อมูลต่าง ๆ



ภาพที่ 5. 17 หน้าจอ Channels

7. API Keys ที่ใช้ในการติดต่อกับ Thing Speak



ภาพที่ 5. 18 หน้าจอ API Keys

5.3 การทดสอบระบบ

5.3.1 Unit Test

Unit Test คือ การทดสอบส่วนย่อยของ Method ที่เขียนว่าสามารถใช้งานได้ตามวัตถุประสงค์ของระบบที่ระบุไว้หรือไม่ โดยใช้ Test case มาทดสอบในแต่ละขั้นตอนเพื่อความพร้อมก่อนจะนำระบบจริงไปใช้งานจริง

ตารางที่ 5. 2 การทดสอบการทำงานของชุดคำสั่งข้อความ

ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	กดเมนูหรือพิมพ์ อุณหภูมิ	แสดงข้อมูลอุณหภูมิ	✓	
2	กดเมนูหรือพิมพ์ ความชื้นสัมพัทธ์	แสดงข้อมูลความชื้นสัมพัทธ์	✓	
3	กดเมนูหรือพิมพ์ คุณภาพอากาศ	แสดงข้อมูลคุณภาพอากาศ	✓	
4	กดเมนูหรือพิมพ์ ฝน	แสดงข้อมูลฝน	✓	
5	กดเมนูหรือพิมพ์ ลม	แสดงข้อมูลลม	✓	
6	กดเมนูหรือพิมพ์ ภาพรวม	แสดงภาพรวมทั้งหมดของระบบ	✓	

ตารางที่ 5. 3 การทดสอบการทำงานของ Cloud Storage

ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	ข้อมูลมีการเปลี่ยนแปลง	ข้อมูลอัปเดต	✓	
2	ข้อมูลผิดปกติ	ข้อมูลอัปเดต	✓	

ตารางที่ 5. 4 การทดสอบการทำงานของ Cloud Image Storage

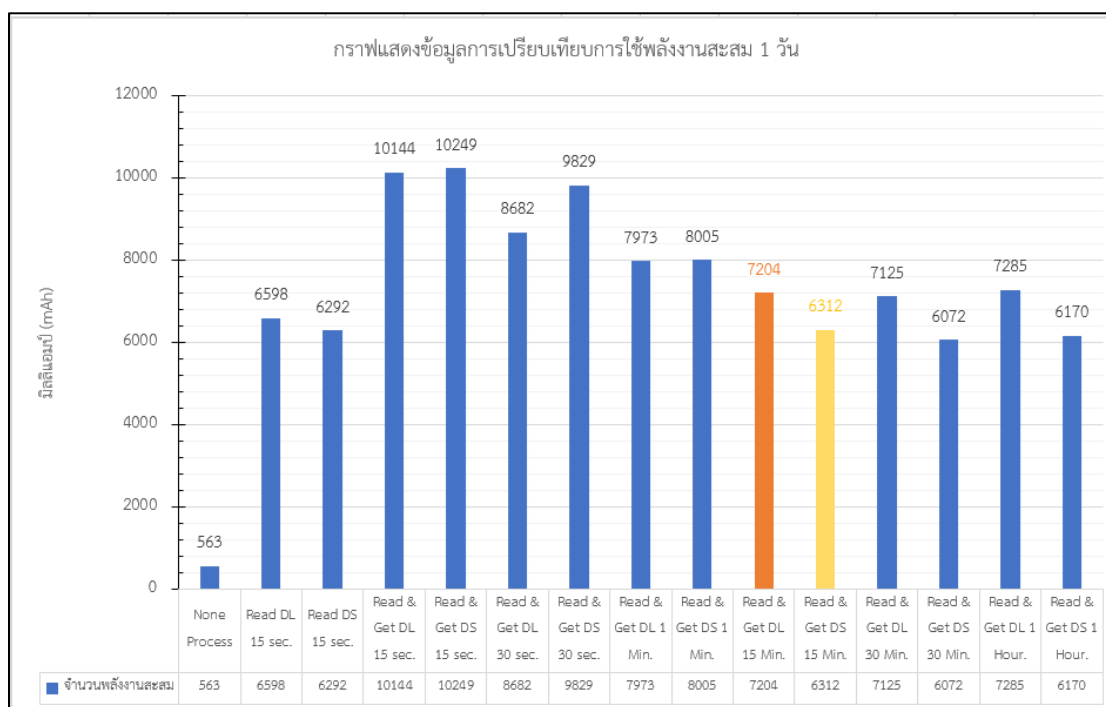
ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	การอัปโหลดรูปภาพ	ข้อมูลอัปโหลด	✓	
2	การตอบกลับการอัปโหลดรูปภาพ	ข้อมูลมีการตอบกลับ	✓	

ตารางที่ 5. 5 การทดสอบการใช้พลังงานของระบบ

ลำดับ	วิธีการทดสอบ	ผลลัพธ์	ผลการทดสอบ	
			ผ่าน	ไม่ผ่าน
1	ทดสอบการเปิดระบบทิ้งไว้โดยไม่มีการรันโปรแกรม เพื่อทดสอบการใช้พลังงานของระบบ และใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 21 มิลลิแอมป์	✓	
2	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมแสดงผลข้อมูลทุก ๆ 1 วินาทีผ่าน Serial Monitor ใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 116 มิลลิแอมป์	✓	
3	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมส่งข้อมูลทุก ๆ 15 วินาที ใช้เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 390 มิลลิแอมป์	✓	

4	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมส่งข้อมูลทุก ๆ 1 นาที โดยจะเพิ่มฟังก์ชัน Deep Sleep ใช้ เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 184 มิลลิแอมป์	✓	
5	ทดสอบการต่อเซ็นเซอร์เข้ากับระบบ และรันโปรแกรมส่งข้อมูลทุก ๆ 5 นาที โดยจะเพิ่มฟังก์ชัน Deep Sleep ใช้ เวลาในการวัด 1 ชั่วโมง	ใช้พลังงานสะสม 1 ชั่วโมง รวม 152 มิลลิแอมป์	✓	

ผลการศึกษาเรื่องพลังงานพบว่า การใช้พลังงานสะสมใน 1 วันที่มีระยะเวลาในการทำงาน การอ่านข้อมูล การส่งข้อมูลที่เหมือนกัน โดยแตกต่างกันที่ฟังก์ชันในการทำงาน ซึ่งฟังก์ชันในการทำงานที่นำมาเปรียบเทียบกับ เพื่อหาความต่างต่างนั้น คือ ฟังก์ชัน Delay และ ฟังก์ชัน Deep Sleep โดยระยะเวลาในการทำงาน การอ่านข้อมูล การส่งข้อมูลนั้นจะเริ่มตั้งแต่ 15 วินาที 30 วินาที 1 นาที 15 นาที 30 นาที และ 1 ชั่วโมง ตามลำดับ โดยในภาพที่ 5.19 นั้น จะเห็นได้ว่าตั้งแต่ระยะเวลา 15 นาทีขึ้นไปนั้น การใช้พลังงานสะสมของฟังก์ชัน Deep Sleep มีการใช้พลังงานที่ลดลงอย่างเห็นได้ชัด ซึ่งทำให้เวลาในการอ่านข้อมูลและส่งข้อมูลอยู่ในช่วงเวลาที่ข้อมูลนั้นไม่เร็ว จนทำให้พลังงานที่ใช้นั้นสูงเกินไป และไม่ได้เข้าจนทำให้ข้อมูลมีการอัปเดตที่ล่าช้าเกินไป



ภาพที่ 5. 19 ข้อมูลการเปรียบเทียบการใช้พลังงานสะสม 1 วัน

บทที่ 6

สรุปผลการดำเนินงาน

6.1 สรุปผล

การพัฒนาสถานีอากาศสำหรับการท่องเที่ยวเชิงนิเวศเป็นระบบที่สามารถตรวจสอบข้อมูลสภาพอากาศผ่านทางเซตบอทบนโปรแกรมไลน์โดยการพัฒนาสถานีอากาศเป็นการนำอุปกรณ์บอร์ด Micro Controller และอุปกรณ์เซ็นเซอร์ต่าง ๆ เช่น เซนเซอร์วัดอุณหภูมิ ความชื้น ปริมาณน้ำฝน ค่าฝุ่น PM2.5 ค่าความเร็วลม และภาพถ่ายบริเวณโดยรอบสถานที่ท่องเที่ยว โดยข้อมูลต่าง ๆ นั้นจะทำการอัปเดตทุก ๆ 15 นาทีซึ่งอยู่ภายใต้เงื่อนไขของปริมาณพลังงานที่มีความจำกัด โดยผู้พัฒนาระบบได้จัดทำระบบนี้ขึ้นเพื่อใช้ในการท่องเที่ยวเชิงนิเวศโดยเน้นไปยังสถานที่ท่องเที่ยวที่ไฟฟ้าและผู้ให้บริการอินเทอร์เน็ตไม่สามารถเข้าถึงได้ ซึ่งระบบนั้นจะทำงานอยู่ได้ประมาณ 3-4 วันและผู้ใช้งานระบบสามารถเข้าถึงข้อมูลทั้งหมดได้ผ่านโปรแกรมไลน์

6.2 ปัญหาในการดำเนินงาน

1. ปัญหาการเลือกใช้อุปกรณ์ เช่น เซ็นเซอร์ เนื่องจากเซ็นเซอร์มีให้เลือกใช้หลายชนิด แต่ละชนิดมีความสามารถในการทำงานแตกต่างกันไม่ว่าจะเป็น ไฟเลี้ยง การเชื่อมต่อ โปรโตคอลในการสื่อสาร
2. สายไฟที่ใช้ในเวอร์ชันพัฒนามีความแข็งแรงไม่เพียงพอเกิดความเสียหายได้ง่าย
3. การเชื่อมต่อสายไฟต้องทำอย่างระมัดระวังและรอบคอบเป็นอย่างมาก หากต่อผิดจะส่งผลให้อุปกรณ์ตัวอื่นเสียหายตามไปด้วย
4. ความไม่พร้อมของอุปกรณ์ สถานีอากาศจำเป็นต้องใช้อุปกรณ์หรือชิ้นส่วนที่ไม่ได้วางไว้ในรายการและจำเป็นต้องสั่งจากแพลตฟอร์มซื้อขายออนไลน์
5. การทดสอบการใช้พลังงานของสถานีอากาศต้องใช้เวลาในการทดสอบ
6. อุปกรณ์ที่ใช้ในการวัดการใช้พลังงานไม่มีการบันทึกค่าแบบอัตโนมัติ

6.3 ข้อเสนอแนะ

ในการพัฒนาระบบควรคำนึงถึงการวางแผนการทำงานให้ถูกต้องและมีความรอบคอบมากยิ่งขึ้น อาจด้วยปัจจัยหลาย ๆ อย่างที่ส่งผลกระทบต่องานทั้งที่สามารถควบคุมได้และไม่สามารถควบคุมได้ โดยในส่วนของอุปกรณ์ที่ใช้นั้น ต้องมีการใช้อุปกรณ์ที่สามารถเปรียบเทียบค่าของเซ็นเซอร์ที่มีประสิทธิภาพและได้มาตรฐาน อีกทั้งยังต้องมีความรู้พื้นฐานเกี่ยวกับภาษาอังกฤษและภาษาโปรแกรม เนื่องจากข้อมูลที่มีนั้นส่วนใหญ่เป็นเนื้อหาภาษาอังกฤษ ส่วนของระยะเวลาการทำงานจะขึ้นอยู่กับขนาดของแบตเตอรี่ หากต้องการระยะเวลาการใช้งานที่นานกว่าระบบปัจจุบันก็ให้ทำการเพิ่มความจุของแบตเตอรี่

บรรณานุกรม

- Arduino.cc. (2018). Two Port Receive สืบค้น 19 มีนาคม 2565 จาก <https://docs.arduino.cc/tutorials/communication/TwoPortReceive>
- Duangsuwan, S., Takarn, A., & Jamjareegulgarn, Punyaw. (2561). A Development on Air Pollution Detection Sensors based on NB-IoT Network for smart Cities. 313-317.
- fitrox.lnwshop.com. (2020). [ESP8266 ตอน 8] Sleep Mode. สืบค้น 24 มีนาคม 2565 จาก <http://fitrox.lnwshop.com/article/35/esp8266-ตอน-8-sleep-mode>
- Jirawatee. (2562). 15 สัญญาณจาก Webhook Events ที่จะปลูกให้ LINE Bot ของคุณตื่นจากภวังค์. สืบค้น 10 กันยายน 2564. จาก <https://1th.me/PsjWm>
- Kankann. (2563). Chatbot คืออะไร ? Chatbot มีกี่ประเภท และมีประโยชน์อย่างไร ?. สืบค้น 10 กันยายน 2564. จาก <https://tips.thaiware.com/1323.html>
- Nattapon Sirikamonnet. (2561). สร้าง LINE BOT กันเถอะ (เริ่มต้น + reply message). สืบค้น 10 กันยายน 2564. จาก <https://1th.me/AgjEG>
- Onamon Ja. (2563). Heroku คืออะไร?. สืบค้น 10 กันยายน 2564, จาก <https://1th.me/7LKUYU>
- php.net. What is PHP?. Retrieved from <https://www.php.net/manual/en/intro-what-is.php>
- Ponpe. (2022). Training ส่งสินค้า ดูแลลูกค้านอกสถานที่ AMT-138E เครื่องวัดและบันทึกค่าอุณหภูมิ ความชื้น SMS Alarm Temperature and Humidity Data logger. สืบค้น 17 มีนาคม 2565, จาก <https://shorturl.asia/YDljf>
- Rui Santos. (2017). ESP8266 Deep Sleep with Arduino IDE (NodeMCU). สืบค้น 17 มกราคม 2565, จาก <https://randomnerdtutorials.com/esp8266-deep-sleep-with-arduino-ide/>
- Rui Santos. (2019). ESP32-CAM Take Photo and Display in Web Server. สืบค้น 18 มกราคม 2565, จาก <https://randomnerdtutorials.com/esp32-cam-take-photo-display-web-server/>
- Rui Santos. (2022). ESP32-CAM Save Picture in Firebase Storage. สืบค้น 14 มีนาคม 2565, จาก <https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/>
- Sudiatmika, I., Dewi, K., Putra, I., & Aryawan, I. (2562). Line Bot Implementation for Automation Balinese Language Dictionary. 227-232.
- Taifur. (2018). Sevel Pro Tips for ESP8266. สืบค้น 17 มีนาคม 2565, จาก shorturl.at/jowHP
- blog.creations.de. (2017). The sleep states for the EPS8266. สืบค้น 17 มีนาคม 2565, จาก The sleep states of the ESP8266 – Toller Text (creations.de)

Thanatcha Kromsang. (2560). เรียนรู้ Git และ Github ฉบับได้กมลาลัย. สืบค้น 10 กันยายน 2564, จาก <https://1th.me/rbV35>

กรัณท์รัตน์ เทียนทับทิม. (2561). กลยุทธ์การบริหารจัดการกระบวนการทำงานภายในองค์กรของธุรกิจรีสอร์ท เพื่อเพิ่มประสิทธิภาพในการทำงาน. กรุงเทพฯ: มหาวิทยาลัยมหิดล, 1-2.

กรมอุทยานแห่งชาติสัตว์ป่าและพันธุ์พืช, (2551). อุทยานแห่งชาติเขาใหญ่. สืบค้นจาก <https://www.dnp.go.th/nprd/project/khaoyai.php>

เบญจวรรณ เจียทองศรี. (2557). การเปลี่ยนแปลงมรสุมและพายุหมุนเขตร้อนในช่วงปี 2524-2556 ในพื้นที่ อ่าวไทยตอนบน. วารสารวิทยาศาสตร์และเทคโนโลยี, 324-327.

สารานุกรมเสรี. (2021). บาร์อมิเตอร์. สืบค้น 20 มีนาคม 2565, จาก <https://shorturl.asia/P793w>

สารานุกรมเสรี. (2021). แอร์การ์ด. สืบค้น 20 มีนาคม 2565, จาก <https://shorturl.asia/41Ni0>

อินันท์ รักหอม และคณะ. (2560). สถานีวัดสภาพอากาศส่วนบุคคลสำหรับอุตสาหกรรมการท่องเที่ยว. มหาวิทยาลัยวลัยลักษณ์, 1-4.

ประวัติผู้จัดทำ

ชื่อผู้จัดทำ	นายธนธร เจ๊ะดาโห๊ะ
รหัสประจำตัว	62103973
สำนักวิชา	สารสนเทศศาสตร์ สาขา เทคโนโลยีสารสนเทศ
วันเดือนปีเกิด	14/04/2544
ที่อยู่	57/2 ม.5 ต.ท่าศาลา อ.ท่าศาลา จ.นครศรีธรรมราช 80160



ชื่อผู้จัดทำ	นายสัญญาภรณ์ ปัทมรักษ์
รหัสประจำตัว	62109905
สำนักวิชา	สารสนเทศศาสตร์ สาขา เทคโนโลยีสารสนเทศ
วันเดือนปีเกิด	13/01/2543
ที่อยู่	191 ม.2 ต.ท่าเรือ อ.เมือง จ.นครศรีธรรมราช 80290

