

# ตู้ใส่เครื่อง AED สำหรับ BeneHeart C2

AED Cabinet for BeneHeart C2

จิตตินาถ บุญหาทิพย์

Jittinat Boontathip

จิรภัทร ศักดิ์ภัทรกุล

Jirapat Sakpattarakul

ธนวัฒน์ บุญแก้ว

Tanapat Boonkaew

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ตู้ใส่เครื่อง AED สำหรับ BeneHeart C2

AED Cabinet for BeneHeart C2

โดย

จิตตินาถ บุญหาทิพย์

จิรภัทร ศักดิ์ภัทรกุล

ธนพัฒน์ บุญแก้ว

อาจารย์ที่ปรึกษา

อาจารย์ชินภัทร นันทจิวารชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2565

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตู้ใส่เครื่อง AED สำหรับ BeneHeart C2

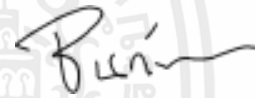
AED Cabinet for BeneHeart C2

ผู้จัดทำ นางสาวจิตติมาถ บุษยาทิพย์ รหัสประจำตัว 62010114

นายจิรภัทร ศักดิ์ภัทรกุล รหัสประจำตัว 62010118

นายธนวัฒน์ บุญแก้ว รหัสประจำตัว 62010357

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(อาจารย์ชินภัทร นันทจิวารชัย)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการงาน	ตู้ใส่เครื่อง AED สำหรับ BeneHeart C2
นักศึกษา	นางสาวจิตตินาถ บุญทาพิพย์ รหัสประจำตัว 62010114
	นายจิรภัทร ศักดิ์ภัทรกุล รหัสประจำตัว 62010118
	นายธนพัฒน์ บุญแก้ว รหัสประจำตัว 62010357
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษาโครงการงาน	อาจารย์ชินภัทร นันทจิวารักษ์

### บทคัดย่อ

โครงการตู้ใส่เครื่องเออีดีมีจุดประสงค์เพื่อนำระบบ IoT มาประยุกต์ใช้ในการตรวจสอบสถานะของเครื่อง AED เพื่อให้ลดเวลา และค่าใช้จ่ายในการติดตามสถานะของเครื่อง AED สามารถบันทึกภาพผู้ที่นำเครื่อง AED ออกไปจากตู้ล่าสุดเพื่อให้สามารถนำไปใช้เป็นหลักฐานได้หากเกิดการสูญหายขึ้น อีกทั้งยังสามารถแสดงสถานะของเครื่อง AED ให้ผู้ดูแลทราบได้ หากเครื่อง AED ไม่พร้อมใช้งานสามารถให้ผู้ดูแลดำเนินการแก้ไขได้อย่างทันที ตามสถานที่ที่ระบุไว้ เพื่อที่จะได้ช่วยเพิ่มโอกาสในการรอดชีวิตของผู้ป่วยฉุกเฉินจากภาวะหัวใจหยุดเต้นเฉียบพลันได้อีกด้วย หากมีผู้ที่มีความต้องการใช้เครื่อง AED จะต้องมีเครื่อง AED ที่พร้อมใช้งานรองรับอยู่เสมอ ในส่วนของแสดงสถานะ เป็นการนำเซนเซอร์มาใช้ตรวจจับค่าอุณหภูมิ ความชื้น และสถานะการเปิดปิดของตู้ใส่เครื่อง AED ส่งไปที่สมาร์ทโฟนผ่าน ESP32 และส่งค่าข้อมูลที่ได้จากเซนเซอร์ไปเก็บที่ database ในส่วนของเว็บไซต์จะเรียกข้อมูลสถานะล่าสุดของอุณหภูมิ ความชื้น และสถานะการเปิดปิดของตู้ ให้ไปแสดงบนเว็บไซต์ เพื่อให้ตรวจสอบสถานะปัจจุบันของตู้ใส่เครื่อง AED ได้ ในส่วนของการรักษาความปลอดภัยนั้น ทำการถ่ายภาพจาก ESP32-CAM ในทุก ๆ 1 วินาทีเมื่อตู้ใส่เครื่อง AED ถูกปิดอยู่ หลังจากนั้นภาพจะถูกบันทึกไปยังการ์ดความจำ และเมื่อตู้ใส่เครื่อง AED ถูกเปิดจะทำการส่งข้อมูลที่เก็บไว้ในการ์ดความจำพร้อมกับการถ่ายรูปปกติทุก 1 วินาที ส่งขึ้นไปยัง storage ไปจนกว่าตู้ใส่เครื่อง AED จะถูกปิด โดยสามารถเรียกดูข้อมูลย้อนหลังได้ หากเกิดการโจรกรรมหรือเกิดความเสียหายใด ๆ ขึ้นที่ตู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Project Title</b>	AED Cabinet for BeneHeart C2	
<b>Student</b>	Ms. Jittinat Boontathip	Student ID 62010114
	Mr. Jirapat Sakpattarakul	Student ID 62010118
	Mr. Tanapat Boonkaew	Student ID 62010357
<b>Degree</b>	Bachelor of Engineering	
<b>Program</b>	Electronics Engineering	
<b>Year</b>	2022	
<b>Project Advisor</b>	Chinnapat Nantajiwakornchai	

## ABSTRACT

AED machine cabinet project the purpose is to apply the IoT system to monitor AED status to reduce time and the cost of tracking the status of the AED. Footage of the last person who removed the AED from the cabinet can be recorded so that it can be used as evidence in case of loss. It can also show the status of the AED to the caregiver. If the AED is not available, the caregiver can take corrective action immediately. according to the specified location to help increase the chances of survival of emergency patients from acute cardiac arrest as well. If there are people who need an AED, there will always be an available AED. In terms of storage Use sensors to detect temperature, humidity, and on-off status of the AED enclosure, send it to the smartphone via ESP32, and send the data from the sensor to the Firebase in the Realtime Database, which is controlled by smartphone application and retrieve the latest status information of temperature, humidity and cabinet opening and closing status. to be displayed on the website to monitor the current status of the AED cabinet. Taking photograph from the ESP32-CAM every 1 second when the AED enclosure is turned off, after which images are saved to the memory card. When the AED cabinet is opened, the data stored on the memory card along with a normal snapshot every 1 second will be sent to storage until the AED cabinet is closed. If there is any theft or damage to the cabinet.

## กิตติกรรมประกาศ

การดำเนินงานโครงการเรื่อง “ตู้ใส่เครื่อง AED สำหรับ BeneHeart C2” ในครั้งนี้จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงการ คือ อาจารย์ชินภัทร นันทจิวากรชัย ที่ให้คำปรึกษาแนะนำ ชี้แนะแนวทางในการแก้ไขปัญหาต่าง ๆ รวมถึงเพื่อน ๆ ในภาควิชาอิเล็กทรอนิกส์ที่มีส่วนช่วยในการทำโครงการครั้งนี้สำเร็จลุล่วงออกมาได้ด้วยดี และมีความสมบูรณ์มากยิ่งขึ้น รวมถึงการได้รับความสนับสนุนในด้านของสถานที่ที่ใช้ในการจัดทำโครงการ เครื่องมือ และอุปกรณ์ต่าง ๆ ที่ใช้ระหว่างการจัดทำโครงการ คณะผู้จัดทำขอขอบคุณทุกคำแนะนำและคำชี้แนะ รวมถึงทุกการให้คำปรึกษาในเรื่องต่าง ๆ ไม่ว่าจะเป็นการแก้ปัญหาในขั้นตอนการทำ หรือคำแนะนำในการเลือกซื้ออุปกรณ์ ตลอดไปจนถึงขั้นตอนการทำรูปเล่มโครงการ และต้องขอขอบคุณทุกคนในครอบครัวที่คอยสนับสนุนค่าใช้จ่าย และอำนวยความสะดวกในการจัดซื้ออุปกรณ์ต่าง ๆ ทำให้สามารถจัดทำโครงการในครั้งนี้สำเร็จออกมาได้

สุดท้ายนี้ข้าพเจ้ารู้สึกขอบคุณในทุก ๆ ความช่วยเหลือของทุก ๆ ท่านที่มีส่วนร่วมในการจัดทำโครงการครั้งนี้ และข้าพเจ้าหวังว่าโครงการเล่มนี้จะมีประโยชน์ต่อผู้อ่านได้ไม่มากนักน้อย และหากมีข้อ ผิดพลาดประการใดต้องขออภัยมา ณ ที่นี้ด้วย

จิตตินาถ บุญทาทิพย์  
จิรภัทร ศักดิ์ภัทรกุล  
ธนพัฒน์ บุญแก้ว

# สารบัญ

หน้า

บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง .....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 ขอบเขตการวิจัย .....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง .....	4
2.1 เครื่องกระตุ้นหัวใจไฟฟ้าชนิดอัตโนมัติ (Automated External Defibrillator).....	4
2.1.1 Mindray BeneHeart C2.....	4
2.1.1.1 สภาพที่เหมาะสมต่อการใช้เครื่อง.....	4
2.1.1.2 คุณสมบัติของเครื่องที่สำคัญ.....	5
2.2 โซลาร์เซลล์ (Solar cell, Photovoltaic cell).....	5
2.2.1 หลักการทำงาน.....	5
2.2.2 ชนิดของโซลาร์เซลล์ .....	5
2.2.2.1 แผงโซลาร์เซลล์ชนิดโพลีคริสตัลไลน์ (Poly Crystalline).....	6
2.2.2.2 แผงโซลาร์เซลล์ชนิดโมโนคริสตัลไลน์ (Mono Crystalline).....	6
2.2.2.3 แผงโซลาร์เซลล์ชนิดฟิล์มบาง (Thin Film).....	7
2.3 แบตเตอรี่ลิเทียมไอออน (Lithium Ion Battery).....	7
2.3.1 การทำงาน .....	7
2.4 เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22 / AM2302.....	8
2.5 ESP32-CAM.....	9
2.6 ESP32.....	10
2.7 MIT app inventor.....	12
2.8 ฐานข้อมูลแบบคลาวด์ (Cloud Database) .....	12
2.9 ระบบฐานข้อมูล (Database System) .....	13
2.10 Firebase Realtime Database.....	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต่อ IV างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11	Firestore .....	14
2.11.1	Cloud Firestore.....	14
2.11.2	Hosting.....	14
2.11.3	Realtime Database – Realtime Database.....	14
2.12	Sealed Lead Acid Battery.....	14
2.13	เซนเซอร์ตรวจวัดสี RGB (APDS-9960) .....	15
บทที่ 3 วิธีการดำเนินงาน .....		17
3.1	ขั้นตอนการดำเนินงาน.....	17
3.2	ขั้นตอนการตั้งค่า .....	17
3.2.1	การตั้งค่า Firebase Authentication .....	17
3.2.2	การตั้งค่า Firebase Realtime database.....	18
3.2.3	การตั้งค่า Firebase Storage.....	18
3.2.4	การตั้งค่า Firebase Hosting.....	18
3.3	การคำนวณเลือกอุปกรณ์ในการสร้างชิ้นงาน.....	21
3.4	หลักการทำงาน.....	22
3.4.1	วงจรการทำงาน .....	22
3.4.1.1	แหล่งจ่ายไฟสำหรับตู้ AED.....	22
3.4.1.2	แหล่งจ่ายไฟสำหรับ ESP32-CAM โดยใช้ Solarcell.....	22
3.4.1.3	ส่วนควบคุมภายในตู้.....	23
3.4.2	การถ่ายภาพจาก ESP-32CAM ไปเก็บ Realtime Database .....	24
3.4.3	การใช้เซนเซอร์วัดค่าต่าง ๆ และนำข้อมูลไปเก็บที่ Realtime Database ..25	
3.4.4	การนำข้อมูลที่เก็บไว้ใน Realtime Database ไปแสดงที่ Web Hosting ..25	
3.5	แผนผังการทำงาน Flow charts .....	26
3.5.1	การทำงานของระบบโดยรวม .....	26
3.5.2	การทำงานของโทรศัพท์.....	27
3.5.3	การทำงานของ ESP32 .....	28
3.5.4	การทำงานของ ESP32-CAM .....	29
3.5.5	การทำงานของ Webapp.....	30
3.6	รูปภาพแสดงส่วนต่าง ๆ ในระบบ.....	30
3.6.1	ตู้ AED.....	30
3.6.2	บอร์ดคอนโทรลในตู้ AED .....	32
3.6.3	ตัวกล้อง ESP-CAM และระบบต่าง ๆ พร้อมโซล่าเซลล์ .....	33
3.6.4	ภายในตัวกล้องESP-CAM.....	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต่อ V ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 เครื่องมือที่ใช้ในการทำงาน.....	34
3.8 อุปกรณ์ที่ใช้ในการทำงาน.....	34
3.9 โปรแกรมที่ใช้ในการทำงาน.....	35
บทที่ 4 ผลการทดลอง.....	36
4.1 การถ่ายภาพจาก ESP-32CAM ไปเก็บ Realtime Database.....	36
4.1.1 การถ่ายภาพเมื่อตู้ AED มีสถานะปิดอยู่จะถ่ายภาพสภาพแวดล้อมด้วยความเร็วตามที่ตั้งไว้ 1 ภาพต่อวินาทีและบันทึกภาพไว้ในการ์ดความจำ จนกว่าจะมีการเปิดตู้.....	36
4.1.2 การถ่ายภาพเมื่อตู้ AED มีสถานะเปิดอยู่จะถ่ายภาพสภาพแวดล้อมด้วยความเร็วที่ 1 ภาพต่อวินาที จนกว่าจะมีการปิดตู้.....	36
4.2 การอ่านค่าเซนเซอร์ และนำข้อมูลไปเก็บที่ Realtime Database.....	37
4.2.1 หน้าตาแอปที่ควบคุมการส่งข้อมูลไปเก็บที่ Realtime Database.....	37
4.2.2 การส่งค่าสถานะต่าง ๆ วันที่และเวลาไปเก็บที่ Realtime Database.....	38
4.3 การนำข้อมูลที่เก็บไว้ใน Realtime Database ไปแสดงที่ Web Hosting.....	39
4.4 การทำงานต่าง ๆ ของระบบ.....	40
4.4.1 ตอนที่ตู้ถูกเปิด.....	40
4.4.2 การรายงานสถานะของเครื่อง AED.....	41
บทที่ 5 สรุปผลการทดลอง.....	42
5.1 สรุปผลการทดลอง.....	42
5.2 วิเคราะห์ผลการทดลอง.....	42
5.3 ข้อเสนอแนะ.....	42
อ้างอิง.....	44

## สารบัญตาราง

หน้า

ตารางที่ 1.1 ระยะเวลาการทำวิจัย และแผนการดำเนินงานตลอดโครงการ.....	2
ตารางที่ 2.1 ข้อมูลจำเพาะด้านความปลอดภัยของ BeneHeart C2 .....	5
ตารางที่ 2.2 ข้อมูลจำเพาะด้านสภาพแวดล้อมของ BeneHeart C2 .....	5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต่อVIRลิงก์ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

	หน้า
รูปที่ 2.1 AED รุ่น BeneHeart C2.....	4
รูปที่ 2.2 แผงโซล่าเซลล์ชนิดโพลีคริสตัลไลน์ (Poly Crystalline) .....	6
รูปที่ 2.3 แผงโซล่าเซลล์ชนิดโมโนคริสตัลไลน์ (Mono Crystalline) .....	6
รูปที่ 2.4 แผงโซล่าเซลล์ชนิดฟิล์มบาง (Thin Film) .....	7
รูปที่ 2.5 แบตเตอรี่ลิเทียมไอออน .....	7
รูปที่ 2.6 การทำงานของแบตเตอรี่ลิเทียมไอออน .....	8
รูปที่ 2.7 เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22 .....	9
รูปที่ 2.8 ESP32-CAM .....	10
รูปที่ 2.9 ESP32-CAM Pinout.....	10
รูปที่ 2.10 ESP32 Pinout .....	12
รูปที่ 2.11 Sealed Lead Acid Battery .....	15
รูปที่ 2.12 เซนเซอร์ตรวจจับสี RGB .....	16
รูปที่ 3.1 การตั้งค่า Firebase Authentication.....	17
รูปที่ 3.2 การตั้งค่าหน้า Sign-in method .....	18
รูปที่ 3.3 การตั้งค่าหน้า User .....	18
รูปที่ 3.4 การตั้งค่าหน้า Rules.....	18
รูปที่ 3.5 การติดตั้ง firebase tool.....	19
รูปที่ 3.6 login firebase .....	19
รูปที่ 3.7 เริ่มต้น firebase.....	19
รูปที่ 3.8 ตัวเลือกพีเจอร์ท่าง ๆ.....	20
รูปที่ 3.9 เลือกโปรเจกต์ที่ต้องการใช้ .....	20
รูปที่ 3.10 Hosting setup .....	20
รูปที่ 3.11 SDK setup and configuration .....	21
รูปที่ 3.12 วงจรจ่ายไฟสำหรับตู้ AED .....	22
รูปที่ 3.13 วงจรจ่ายไฟสำหรับ ESP32-CAM.....	23
รูปที่ 3.14 วงจรส่วนควบคุมภายในตู้.....	24
รูปที่ 3.15 แผนภาพการทำงานของระบบโดยรวม .....	26
รูปที่ 3.16 แผนภาพแสดงการทำงานของโทรศัพท์ .....	27
รูปที่ 3.17 แผนภาพแสดงการทำงานของ ESP32.....	28
รูปที่ 3.18 แผนภาพแสดงการทำงานของ ESP32-CAM .....	29
รูปที่ 3.19 แผนภาพแสดงการทำงานของ Webapp .....	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต่อVIIIงอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.20 ตู้ AED ขณะปิดประตู.....	31
รูปที่ 3.21 ตู้ AED ขณะเปิดประตู.....	32
รูปที่ 3.22 บอร์ดคอนโทรลในตู้ AED .....	32
รูปที่ 3.23 ตัวกล้อง ESP-CAM และระบบต่าง ๆ พร้อมโซล่าเซลล์ .....	33
รูปที่ 3.24 ภายในตัวกล้องESP-CAM.....	33
รูปที่ 4.1 ภาพที่ถูkBันทึกไว้ในการ์ดความจำ .....	36
รูปที่ 4.2 รูปจากการ์ดความจำที่ถูกส่งไปเก็บยัง Storage .....	37
รูปที่ 4.3 รูปที่ถ่ายขณะที่ตู้เปิดถูกส่งไปเก็บยัง Storage .....	37
รูปที่ 4.4 หน้าแอปที่ควบคุมการรับส่งข้อมูลไปที่ Realtime database .....	38
รูปที่ 4.5 ตัวอย่างการเก็บข้อมูลที่ Realtime database .....	38
รูปที่ 4.6 ตัวอย่างการแสดงผลสถานะต่าง ๆ ของเครื่อง AED บนเว็บไซต์.....	39
รูปที่ 4.7 การแสดงตำแหน่งที่ตั้งของเครื่อง AED บนเว็บไซต์ .....	39
รูปที่ 4.8 ตัวอย่างการแสดงรูปภาพที่บันทึกขณะที่มีการเปิดปิดตู้บนเว็บไซต์ .....	40
รูปที่ 4.9 Realtime database และ webapp ตอนตู้ถูกเปิด.....	40
รูปที่ 4.10 Realtime database และ webapp แสดงสถานะของเครื่อง AED ว่า Flashing red ..	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต่อIX่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของปัญหา

ภาวะหัวใจหยุดเต้นเฉียบพลันเป็นภาวะที่สามารถเกิดขึ้นได้กับทุกคนในทุกสถานที่ และสามารถเกิดขึ้นได้ตลอดเวลาโดยที่ไม่สามารถคาดการณ์ล่วงหน้าได้ จากการสำรวจพบว่าในแต่ละปีคนไทยเสียชีวิตด้วยภาวะหัวใจหยุดเต้นเฉียบพลันกว่า 54,000 รายต่อปี หรือในทุก ๆ 1 ชั่วโมงจะมีคนเสียชีวิตด้วยภาวะหัวใจหยุดเต้นเฉียบพลัน 6 คน ซึ่งการเสียชีวิตด้วยภาวะหัวใจหยุดเต้นเฉียบพลันนี้เป็นสาเหตุการเสียชีวิตเป็นอันดับ 3 รองจากโรคมะเร็งและอุบัติเหตุ เนื่องด้วยเวลาในการช่วยเหลือผู้ที่หัวใจหยุดเต้นเฉียบพลันนั้นมีระยะเวลาการช่วยเหลือสั้น ๆ เนื่องจากทันทีที่หัวใจหยุดเต้นเลือดจะไม่ถูกสูบฉีดไปเลี้ยงสมองซึ่งจะทำให้สมองตายเนื่องจากขาดเลือดและออกซิเจน ในกลุ่มคนไข้ที่หมดสติแล้วมีภาวะหัวใจหยุดเต้น การทำ CPR อย่างเดียวอัตราการรอดชีวิตจะอยู่ที่ประมาณ 3-5 เปอร์เซ็นต์ แต่ถ้าใช้เครื่องกระตุกหัวใจไฟฟ้าชนิดอัตโนมัติ (Automated External Defibrillator, AED) ร่วมด้วยอัตราการรอดชีวิตเพิ่มขึ้นได้ประมาณ 10 เท่า ภาวะหัวใจหยุดเต้นเฉียบพลันนี้ถือเป็นภัยเงียบเพราะในบางครั้งก็ไม่อาจทราบสาเหตุที่แน่ชัดได้ เนื่องจากสาเหตุที่ทำให้เกิดความผิดปกติมีได้หลายสาเหตุ แม้ว่าจะดูแลสุขภาพอย่างดี ออกกำลังกายอย่างสม่ำเสมอก็สามารถเกิดภาวะนี้ได้เช่นกัน

ปัจจุบันประเทศไทยยังขาดแคลนเครื่อง AED ในพื้นที่สาธารณะเนื่องจากหลาย ๆ สาเหตุทั้งด้านการดูแลรักษา ระบบรักษาความปลอดภัย รวมถึงการต้องใช้บุคลากรในการมาตรวจสอบและติดตามสถานะการใช้งานของเครื่อง AED ให้พร้อมใช้งานอยู่เสมอ หากมีเครื่อง AED อยู่ในสถานที่ต่าง ๆ แต่ขาดการดูแลรักษาและการติดตามสถานะการใช้งานของเครื่องอาจทำให้เครื่องเกิดความเสียหาย และเมื่อเกิดเหตุการณ์ที่ต้องนำมาใช้อาจไม่สามารถช่วยชีวิตได้ทัน แต่การที่ต้องให้บุคลากรมาตรวจสอบเครื่อง AED ในแต่ละพื้นที่อยู่บ่อย ๆ นั้นก็เป็นการสิ้นเปลืองทั้งเวลาและงบประมาณในการเดินทาง

ผู้จัดทำได้เห็นความสำคัญในจุดนี้จึงได้จัดสนใจทำโครงการตู้ใส่เครื่อง AED ขึ้นมาเพื่อที่จะได้มีการนำระบบ IoT มาประยุกต์ใช้ทำให้มีระบบที่สามารถตรวจสอบสถานะของเครื่อง AED และการบันทึกภาพผู้ที่นำเครื่อง AED ออกไปจากตู้ล่าสุดเพื่อให้สามารถนำไปใช้เป็นหลักฐานได้หากเกิดการสูญหายขึ้น อีกทั้งยังมีการระบุตำแหน่งด้วย GPS ทำให้ไม่ต้องสิ้นเปลืองงบประมาณและบุคลากรที่ต้องมาตรวจสอบติดตามสถานะของเครื่อง AED ในพื้นที่ด้วยตัวเอง และผู้จัดทำก็ได้มีความสนใจที่จะเรียนรู้ด้านการประยุกต์ใช้ IoT และพัฒนาทักษะในด้านการเขียนโปรแกรมให้ดียิ่งขึ้นอีกด้วย

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- เพื่อเป็นการนำระบบ IoT มาประยุกต์ใช้ในการตรวจสอบสถานะของเครื่อง AED เพื่อให้ลดเวลา และค่าใช้จ่ายในการติดตามสถานะของเครื่อง AED
- เพื่อให้สามารถบันทึกภาพผู้ที่นำเครื่อง AED ออกไปจากตู้ล่าสุดเพื่อให้สามารถนำไปใช้เป็นหลักฐานได้หากเกิดการสูญหายขึ้น
- เพื่อเป็นระบบที่สามารถตรวจสอบความพร้อมในการใช้งานของเครื่อง AED ได้
- เพื่อช่วยฝึกฝนและพัฒนาทักษะในด้านการเขียนโปรแกรม

## 1.3 ขอบเขตการวิจัย

- สามารถบันทึกภาพและเวลาเมื่อมีการเปิด-ปิดตู้ใส่เครื่อง AED เมื่อมีการนำเครื่อง AED ออกไปใช้งาน และสามารถส่งข้อมูลไปยังผู้ที่ดูแลเพื่อให้ทราบสถานะการใช้งานของเครื่อง AED ได้
- สามารถส่งการแจ้งเตือนไปหาผู้ที่ดูแลได้เมื่อเมื่อเครื่อง AED ไม่พร้อมใช้งาน
- สามารถแสดงสถานะต่าง ๆ และตรวจสอบการนำเครื่อง AED ไปใช้งานได้
- ระยะเวลาการทำวิจัย และแผนการดำเนินงานตลอดโครงการ เริ่มต้นตั้งแต่ 1 กันยายน 2565 จนถึง 30 เมษายน 2566

ตารางที่ 1.1 ระยะเวลาการทำวิจัย และแผนการดำเนินงานตลอดโครงการ

ขั้นตอนการดำเนินงาน	ก.ย. 2565	ต.ค. 2565	พ.ย. 2565	ธ.ค. 2565	ม.ค. 2566	ก.พ. 2566	มี.ค. 2566	เม.ย. 2566
ทฤษฎี/ชื่ออุปกรณ์	<-----	----->						
จัดทำและประกอบชิ้นงาน		<-----	-----	-----	----->			
เก็บผลการทดลอง			<-----	-----	-----	----->		
สรุปผลการทดลอง						<-----	-----	----->
เขียนรายงานฉบับสมบูรณ์	<-----	-----	-----	-----	-----	-----	-----	----->

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถนำระบบ IoT มาประยุกต์ใช้ในการตรวจสอบสถานะของเครื่อง AED เพื่อให้ลดเวลา และค่าใช้จ่ายในการติดตามสถานะของเครื่อง AED ได้
- สามารถบันทึกภาพผู้ที่นำเครื่อง AED ออกไปจากตู้ล่าสุดเพื่อให้สามารถนำไปใช้เป็นหลักฐานได้หากเกิดการสูญหายขึ้น
- สามารถแสดงสถานะของเครื่อง AED ให้ผู้ดูแลทราบได้ หากเครื่อง AED ไม่พร้อมใช้งาน สามารถแสดงให้ผู้ดูแลทราบได้ และสามารถมาดำเนินการแก้ไขได้อย่างทันที ตามสถานที่ที่ระบุไว้ เพื่อที่จะได้ช่วยเพิ่มโอกาสในการรอดชีวิตของผู้ป่วยฉุกเฉินจากภาวะหัวใจหยุดเต้นเฉียบพลันได้
- สามารถสร้างระบบที่สามารถบันทึกภาพผู้ที่นำเครื่อง AED ไปใช้งาน และสามารถบันทึกเวลาเมื่อมีการเปิด-ปิดตู้ใส่เครื่อง AED ให้ผู้ดูแลทราบได้
- ได้ฝึกฝนและพัฒนาทักษะในด้านการเขียนโปรแกรมให้ดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# หลักการและทฤษฎีที่เกี่ยวข้อง

### 2.1 เครื่องกระตุ้นหัวใจไฟฟ้าชนิดอัตโนมัติ (Automated External Defibrillator)

เป็นอุปกรณ์อิเล็กทรอนิกส์แบบพกพาชนิดหนึ่งซึ่งสามารถวินิจฉัยสถานะหัวใจเต้นผิดจังหวะที่เป็นอันตรายถึงชีวิตได้โดยอัตโนมัติ และสามารถให้การรักษาดูแลด้วยการช็อกไฟฟ้ากระตุ้นหัวใจได้โดยใช้กระแสไฟฟ้าหยุดการเต้นของหัวใจที่ผิดจังหวะ เพื่อเปิดโอกาสให้หัวใจกลับมาเต้นใหม่ในจังหวะที่ถูกต้อง



รูปที่ 2.1 AED รุ่น BeneHeart C2

แหล่งที่มา: <https://www.mindray.com/en/products/aed/beneheart-c-series>

#### 2.1.1 Mindray BeneHeart C2

##### 2.1.1.1 สภาพที่เหมาะสมต่อการใช้เครื่อง

สภาพแวดล้อมที่เหมาะสมต่อการทำงานของเครื่องจะต้องเป็นไปตามคุณสมบัติที่ได้ระบุไว้ โดยจะต้องปราศจากผลของสัญญาณรบกวน การสั่น ผุน การกัดกร่อน เปลวไฟ และสารที่ส่งผลให้เกิดการระเบิด

การดูแลและรักษา จำเป็นต้องทำให้อุปกรณ์หรือส่วนเสริมต่างๆ ให้ปลอดภัยจากฝุ่นและสิ่งสกปรก เพื่อหลีกเลี่ยงความเสียหายของอุปกรณ์ต่างๆ จำเป็นต้องปฏิบัติ ดังนี้

- เจือจางตามคำแนะนำของผู้ผลิตหรือใช้ความเข้มข้นต่ำที่สุดเท่าที่จะเป็นไปได้
- ห้ามจุ่มอุปกรณ์บางส่วนลงในของเหลว
- อย่าเทของเหลวลงบนอุปกรณ์หรืออุปกรณ์เสริม
- อย่าให้ของเหลวเข้าไปในตัวเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ห้ามใช้วัสดุที่มีฤทธิ์กัดกร่อน (เช่น ฝอยเหล็กหรือน้ำยาขัดเงาเงิน)

ควรทำความสะอาดอุปกรณ์เป็นประจำ หากมีมลพิษมากหรือมีฝุ่นและทรายจำนวนมากในพื้นที่ที่ทำการติดตั้ง ควรทำความสะอาดอุปกรณ์ให้บ่อยขึ้น ก่อนทำความสะอาดอุปกรณ์ ให้ศึกษาคู่มือข้อบังคับของสถานที่ก่อนจะทำความสะอาดอุปกรณ์

### 2.1.1.2 คุณสมบัติของเครื่องที่สำคัญ

ตารางที่ 2.1 ข้อมูลจำเพาะด้านความปลอดภัยของ BeneHeart C2

ระดับการป้องกันอันตรายจากการเข้าของของแข็ง	IP5X
ระดับการป้องกันน้ำเข้าที่เป็นอันตราย	IPX5

ตารางที่ 2.2 ข้อมูลจำเพาะด้านสภาพแวดล้อมของ BeneHeart C2

รายการ	อุณหภูมิ	ความชื้นสัมพัทธ์	บรรยากาศ
เงื่อนไขการจัดเก็บ แบบระยะสั้น	-30°C to 70°C	5% to 95% (ไม่ควบแน่น)	57.0 ถึง 106.2 kPa
เงื่อนไขการจัดเก็บ แบบระยะยาว	15°C to 35°C	5% to 95% (ไม่ควบแน่น)	57.0 ถึง 106.2 kPa

## 2.2 โซลาร์เซลล์ (Solar cell, Photovoltaic cell)

เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่ง ที่มีคุณสมบัติที่สามารถเปลี่ยนแปลงพลังงานแสงอาทิตย์เป็นพลังงานไฟฟ้า โดยกระแสไฟฟ้าที่ผลิตได้จากโซลาร์เซลล์นั้น จะเป็นไฟฟ้ากระแสตรงซึ่งเราสามารถนำมาใช้ประโยชน์ได้ทันทีหรือสามารถนำมาเก็บไว้ในแบตเตอรี่เพื่อใช้งานภายหลังได้ พลังงานที่ผลิตได้จากแผงโซลาร์เซลล์นี้ จัดว่าเป็นพลังงานสะอาด ไม่สร้างมลภาวะแก่สิ่งแวดล้อมและไม่ปล่อยก๊าซเรือนกระจก

### 2.2.1 หลักการทำงาน

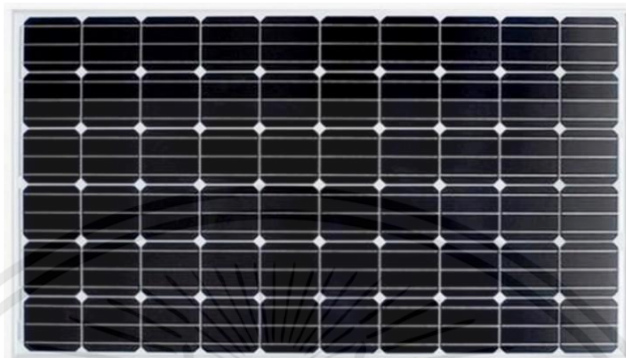
โดยเมื่อแสงซึ่งเป็นคลื่นแม่เหล็กไฟฟ้าและมีพลังงานไปกระทบกับสารกึ่งตัวนำ จะเกิดการถ่ายเทพลังงานระหว่างกัน พลังงานจากแสงจะทำให้เกิดการเคลื่อนที่ของกระแสไฟฟ้า(อิเล็กตรอน) ขึ้นในสารกึ่งตัวนำ จึงสามารถต่อกระแสไฟฟ้าง่ายๆไปใช้งานได้

### 2.2.2 ชนิดของโซลาร์เซลล์

แผงโซลาร์เซลล์แบ่งออกเป็น 3 ชนิด

### 2.2.2.1 แผงโซลาร์เซลล์ชนิดโพลีคริสตัลไลน์ (Poly Crystalline)

เป็นแผงโซลาร์เซลล์ชนิดแรก ที่ทำมาจากผลึกซิลิคอน บางครั้งเรียกว่า มัลติ-คริสตัลไลน์ (Multi-Crystalline) โดยกระบวนการผลิต จะนำเอาซิลิคอนเหลว มาเทใส่โมลด์ที่เป็นสี่เหลี่ยม ก่อนจะนำมาตัดเป็นแผ่นบางอีกที จึงทำให้แต่ละเซลล์เป็นรูปสี่เหลี่ยมจัตุรัส สีของแผงจะออกสีน้ำเงิน

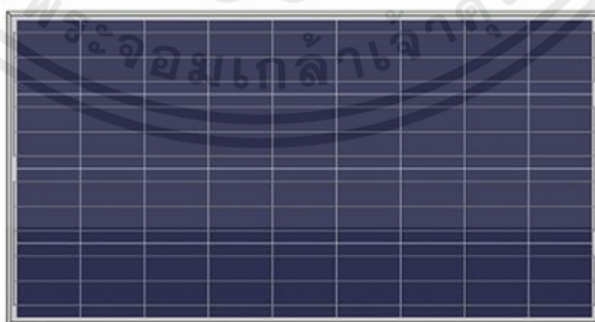


รูปที่ 2.2 แผงโซลาร์เซลล์ชนิดโพลีคริสตัลไลน์ (Poly Crystalline)

แหล่งที่มา: <https://www.gump.in.th/article/535>

### 2.2.2.2 แผงโซลาร์เซลล์ชนิดโมโนคริสตัลไลน์ (Mono Crystalline)

เป็นแผงโซลาร์เซลล์ที่ทำมาจาก ผลึกซิลิคอนเชิงเดี่ยว (Mono- Silicon) บางครั้งเรียกว่า Single Crystalline ลักษณะแต่ละเซลล์เป็นสี่เหลี่ยมตัดมุมทั้งสี่มุม และมีสี่เหลี่ยมมุมฉาก ความบริสุทธิ์สูง กวมน้ำผลึกเกาะกันที่แกนกลาง ทำให้เกิดแหงทรงกระบอก จากนั้นนำมาตัดให้เป็นสี่เหลี่ยมและลบมุมทั้งสี่ออก ทำให้ได้ประสิทธิภาพสูงสุด และลดการใช้วัตถุดิบ Mono- Silicon ลง ก่อนที่จะนำมาตัดเป็นแผ่นอีกที



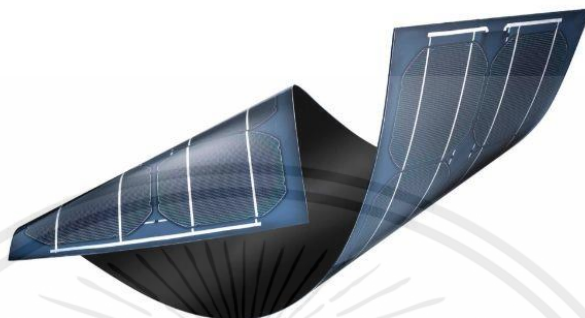
รูปที่ 2.3 แผงโซลาร์เซลล์ชนิดโมโนคริสตัลไลน์ (Mono Crystalline)

แหล่งที่มา: <https://www.gump.in.th/article/535>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.2.3 แผงโซลาร์เซลล์ชนิดฟิล์มบาง (Thin Film)

เป็นแผงโซลาร์เซลล์ที่ทำมาจาก การนำสารที่แปลงพลังงานแสงเป็นพลังงานไฟฟ้า มาฉาบเป็นชั้นบางๆ ซ้อนกันหลายๆชั้น จึงเรียกโซลาร์เซลล์ชนิดนี้ว่า ฟิล์มบาง (thin film) แผงโซลาร์เซลล์ ชนิดฟิล์มบาง มีประสิทธิภาพเฉลี่ยอยู่ที่ 7-13 % ทั้งนี้ขึ้นอยู่กับชนิดของวัสดุที่นำมาทำเป็นฟิล์มฉาบ



รูปที่ 2.4 แผงโซลาร์เซลล์ชนิดฟิล์มบาง (Thin Film)

แหล่งที่มา: <https://www.gump.in.th/article/535>

## 2.3 แบตเตอรี่ลิเทียมไอออน (Lithium Ion Battery)

แบตเตอรี่ลิเทียมไอออน มีจุดเด่นที่ความจุพลังงานและกำลังไฟฟ้าที่สูงกว่าแบตเตอรี่ตระกูลนิกเกิลและกรดตะกั่ว นอกจากนี้ ยังมีค่าศักย์ไฟฟ้าสูง มีอัตราการสูญเสียประจุระหว่างไม่ใช้งาน (self-discharge rate) ที่ต่ำ ไม่มีปรากฏการณ์ความจำและมีความปลอดภัยสูงกว่าแบตเตอรี่ที่ใช้โลหะลิเทียมเป็นขั้ว



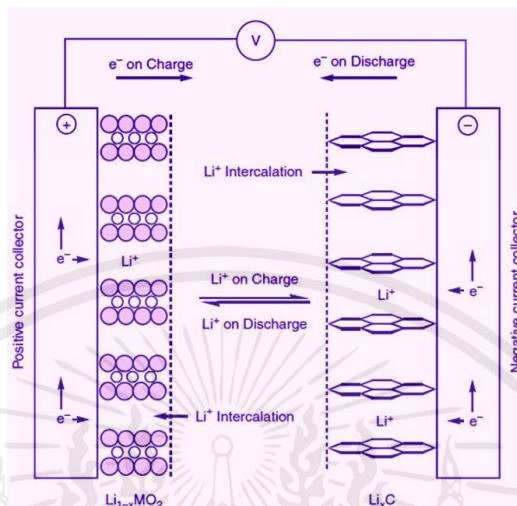
รูปที่ 2.5 แบตเตอรี่ลิเทียมไอออน

แหล่งที่มา: <https://www.kanbkam.com/ae/en>

### 2.3.1 การทำงาน

เมื่อมีการอัดประจุ (charge) ไอออนของลิเทียมจะเคลื่อนออกจากโครงสร้างของขั้วบวก ผ่านเยื่อเลือกผ่าน เข้าสู่ขั้วลบ เกิดเป็นสารประกอบของลิเทียมคาร์บอน และขณะเดียวกันอิเล็กตรอนจะแยกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนจากขั้วบวกสู่ขั้วลบผ่านวงจรภายนอก และขณะเกิดการคายประจุ(discharge) ปฏิกริยาจะเกิดในทางตรงกันข้าม กระบวนการที่ไอออนของลิเทียมสอดแทรกเข้าไปอยู่ในโครงสร้างของวัสดุขั้วบวกหรือขั้วลบ เรียกว่า lithium intercalation หรือ lithium insertion



รูปที่ 2.6 การทำงานของแบตเตอรี่ลิเทียมไอออน

แหล่งที่มา: [https://www2.mtec.or.th/th/e-magazine/admin/upload/298\\_63.pdf](https://www2.mtec.or.th/th/e-magazine/admin/upload/298_63.pdf)

## 2.4 เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22 / AM2302

เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22 / AM2302 หรือ Temperature and Humidity Sensor Module DHT22 / AM2302 เป็นโมดูลเซ็นเซอร์ที่สามารถวัดความชื้นและอุณหภูมิได้ในตัวเดียว มีความแม่นยำสูง ความละเอียดในการวัดอุณหภูมิ 0.1 องศาเซลเซียส ความละเอียดในการวัดค่าความชื้น 1% RH สื่อสารผ่านบัส 1-wire ใช้สายสัญญาณเพียง 1 เส้น มีตัวต้านทาน Pull up สามารถต่อขาใช้งานสะดวกมากยิ่งขึ้น สามารถใช้งานร่วมกับ Arduino ได้ง่าย มีไลบรารีพร้อมใช้งาน ซึ่งเซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22 มีคุณสมบัติดังต่อไปนี้

- แรงดันไฟเลี้ยง 3.3-6 V<sub>DC</sub>
- ใช้กระแสสูงสุด 1.5mA
- วัดค่าความชื้นได้ตั้งแต่ 0-100%
- วัดอุณหภูมิได้ตั้งแต่ -40 ถึง 80 องศาเซลเซียส
- ค่าความผิดพลาดในการวัดความชื้น 2%
- ค่าความผิดพลาดในการวัดอุณหภูมิ 0.5 องศาเซลเซียส
- ตัวโมดูลต่อตัวต้านทาน Pull-up ค่า 47k ไว้ให้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22

แหล่งที่มา: <https://www.spmicrotech.com/>

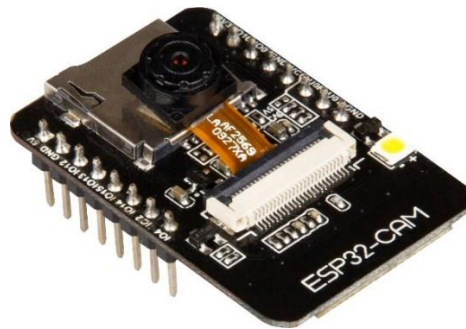
## 2.5 ESP32-CAM

ESP32-CAM เป็นโมดูลที่คุณสามารถใช้กับโครงการต่างๆมากมายและกับ Arduino เป็นโมดูลที่สมบูรณ์พร้อมด้วยไมโครคอนโทรลเลอร์ในตัวซึ่งสามารถทำให้ทำงานได้อย่างอิสระ นอกเหนือจากการเชื่อมต่อ Wi-Fi + Bluetooth แล้วโมดูลนี้ยังมีกล้องวิดีโอในตัวและช่องเสียบ microSD สำหรับจัดเก็บข้อมูล สามารถนำมาประยุกต์ใช้กับ IoT ธรรมดาไปจนถึงขั้นสูงอื่น ๆ สำหรับการตรวจสอบและจดจำภาพโดยใช้ AI และแม้กระทั่งเป็นระบบเฝ้าระวังเพื่อตรวจสอบสิ่งที่เกิดขึ้นในสถานที่ต่าง ๆ จากระยะไกลโดยสามารถดูภาพแบบ real-time ผ่านโทรศัพท์มือถือหรือทางหน้าเว็บเบราว์เซอร์ได้สะดวก โดย ESP32-CAM มีคุณสมบัติดังต่อไปนี้

- Wi-Fi 802.11b / g / n + Bluetooth 4.2 พร้อม BLE รองรับการอัปเดตภาพผ่าน WiFi
- UART, เอสพีไอ, I2C และ PWM มีหมด GPIO 9 ตัว
- ความถี่สัญญาณนาฬิกาสูงสุด 160 Mhz.
- กำลังประมวลผลไมโครคอนโทรลเลอร์ สูงสุด 600 DMIPS
- หน่วยความจำ 520KB ของ SRAM + 4MB ของช่องเสียบการ์ด PSRAM + SD
- บริการเสริม มีโหมดสลิปหลายโหมดอัปเดตเฟิร์มแวร์ได้โดย OTA และ LED สำหรับใช้

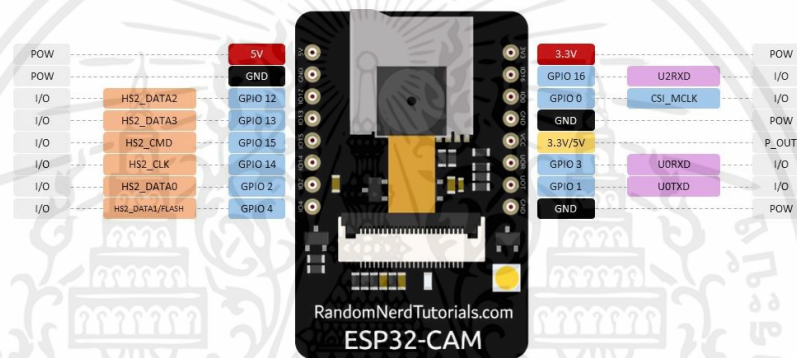
หน่วยความจำแฟลชในตัว

- กล้อง 2 MP บนเซ็นเซอร์ของคุณ, ขนาดอาร์เรย์ UXGA 1622 × 1200 พิกเซล, รูปแบบเอาต์พุต YUV422, YUV420, RGB565, RGB555 และการบีบอัดข้อมูล 8 บิต สามารถถ่ายโอนภาพระหว่าง 15 ถึง 60 FPS



รูปที่ 2.8 ESP32-CAM

แหล่งที่มา: <https://www.conrad.com/p/joy-it-sbc-esp32-cam-camera-module-1-pcs-2332111>



รูปที่ 2.9 ESP32-CAM Pinout

แหล่งที่มา: <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>

## 2.6 ESP32

ESP32 เป็นชิปไมโครคอนโทรลเลอร์ที่มาพร้อม WiFi มาตรฐาน 802.11 b/g/n และบลูทูธ เวอร์ชัน 4.2 เป็นรุ่นต่อยอดความสำเร็จของ ESP8266 โดยในรุ่นนี้ได้ออกมาแก้ไขข้อเสียของ ESP8266 ทั้งหมด โดย CPU ใช้สถาปัตยกรรม Tensilica LX6 จำนวน 2 คอร์ สัญญาณนาฬิกา 240MHz สามารถแยกการทำงานระหว่างโปรแกรมจัดการ WiFi และแอปพลิเคชันออกจากกันได้ ทำให้มีเสถียรภาพเพิ่มมากขึ้น มีแรม 520KB มาในตัว นอกจากนี้ยังมี GPIO เพิ่มขึ้นมาก และมีช่อง ADC เพิ่มขึ้นเป็น 12 ช่อง จากเดิม ESP8266 มีเพียงช่องเดียว ใช้แรงดันไฟฟ้า 3.3V ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA ผลิตโดยบริษัท Espressif จากประเทศจีน โดยตัวไอซี ESP32 มีสเปคโดยละเอียด ดังนี้

- ชิพใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz
- มีแรมในตัว 512KB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB
- มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi direct

- มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE
- ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V
- ทำงานได้ที่อุณหภูมิ -40°C ถึง 125°C
- วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ
- เซ็นเซอร์แม่เหล็ก
- เซ็นเซอร์สัมผัส (Capacitive touch) รองรับ 10 ช่อง
- รองรับการเชื่อมต่อคลิสตอล 32.768kHz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะ

การใช้งานต่าง ๆ ของ ESP32 รองรับการเชื่อมต่อบัสดังต่อไปนี้

- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I2S จำนวน 2 ช่อง
- รองรับ PWM / Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card

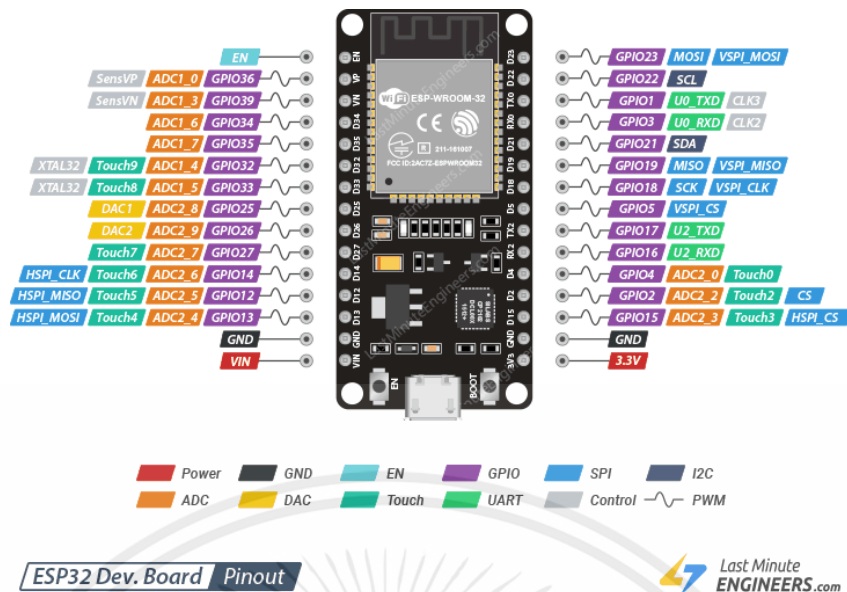
นอกจากนี้ ESP32 ยังรองรับฟังก์ชันเกี่ยวกับความปลอดภัยต่าง ๆ ดังนี้

- รองรับการเข้ารหัส WiFi แบบ WEP และ WPA/WPA2 PSK/Enterprise
- มีวงจรเข้ารหัส AES / SHA2 / Elliptical Curve Cryptography / RSA-4096 ในตัว

ในด้านประสิทธิภาพการใช้งาน ตัว ESP32 สามารถทำงานได้ดี โดย

- รับ - ส่ง ข้อมูลด้วยความเร็วสูงสุดที่ 150Mbps เมื่อเชื่อมต่อแบบ 11n HT40 ได้ความเร็วสูงสุด 72Mbps เมื่อเชื่อมต่อแบบ 11n HT20 ได้ความเร็วสูงสุดที่ 54Mbps เมื่อเชื่อมต่อแบบ 11g และได้ความเร็วสูงสุดที่ 11Mbps เมื่อเชื่อมต่อแบบ 11b
- เมื่อใช้การเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับ - ส่งข้อมูลได้ด้วยความเร็ว 135Mbps
- ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ESP32 Pinout

แหล่งที่มา: <https://lastminuteengineers.b-cdn.net/wp-content/uploads/iot/ESP32-Pinout.png>

## 2.7 MIT app inventor

เป็นตัวสร้างแอปสำหรับสมาร์ทโฟนและแท็บเล็ต ถูกพัฒนาโดยสถาบันเทคโนโลยีแมสซาชูเซตส์ การเขียนแอปจะเป็นลักษณะการเขียนโปรแกรมแบบบล็อก (Block-based coding) ซึ่งเครื่องมือแบบบล็อกช่วยอำนวยความสะดวกในการสร้างแอปอย่างมาก

## 2.8 ฐานข้อมูลแบบคลาวด์ (Cloud Database)

ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่บังคับว่าข้อมูลทั้งหมดจะต้องถูกเก็บไว้ในที่เดียวกันหรือแยกเก็บหลายๆแฟ้มข้อมูล ซึ่งข้อมูลจะถูกจัดเก็บอย่างเป็นระเบียบโดยจะมีซอฟต์แวร์เข้ามาควบคุมกระบวนการใช้งาน การทำงาน หรือการประมวลผล ทำให้ผู้ใช้บริการสามารถเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ

ฐานข้อมูลแบบคลาวด์ คือ ระบบของฐานข้อมูลรูปแบบใหม่ ที่สามารถทำงานได้แบบออนไลน์ สามารถที่จะทำงานเชื่อมโยงกับ Web Application ได้อย่างสะดวกและลงตัวมากขึ้น หรืออาจจะกล่าวได้อีกอย่างว่า cloud database คือ ระบบฐานข้อมูลที่ทำงานอยู่บนเทคโนโลยี Cloud ทำให้ระบบฐานข้อมูลมีความยืดหยุ่น สามารถรับส่งข้อมูลได้อย่าง Realtime อัปเดตข้อมูลได้แบบทันที และยังสามารถปรับแต่ง เพิ่มขนาดหรือขยายฐานข้อมูลและทำการแก้ไขได้อย่างอิสระอีกด้วย นั่นเท่ากับว่าระบบฐานข้อมูลแบบนี้สามารถตอบสนองการทำงานในยุคปัจจุบันได้ดีกว่า ระบบฐานข้อมูลแบบเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่างๆ มากกว่า 2 ฐานข้อมูล เป็นต้นไป ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบ มีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ในระบบ ฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมี ซอฟต์แวร์ ที่เปรียบเสมือนสื่อกลางระหว่าง ผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล (data base management system) มีหน้าที่ช่วยให้ผู้ใช้ เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ

## 2.10 Firebase Realtime Database

Firebase Realtime Database เป็นฐานข้อมูลประเภท SQL cloud database ที่เก็บ ข้อมูลในรูปแบบของ JSON และมีการซิงค์ ข้อมูลแบบ เรียลไทม์ กับทุกอุปกรณ์ รองรับการ ทำงาน เมื่อออฟไลน์รวมถึงมี Security Rules ให้เราสามารถออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการ อ่าน และ เขียนข้อมูล ทั้งในเว็บและโทรศัพท์

โดยปัจจุบันเป็นฐานข้อมูลที่มีความนิยมสำหรับการพัฒนาโมบายแอป , เว็บ, เกม และ IoT เนื่องจากมีจุดเด่นคือสามารถซิงค์ข้อมูลของให้เครื่องผู้ใช้ทั้งหมดได้แบบเรียลไทม์

ความสามารถที่สำคัญ

- Realtime ใช้วิธีการซิงโครไนซ์ข้อมูลเพื่อให้มีการเปลี่ยนแปลงข้อมูลระหว่างอุปกรณ์ที่เชื่อมต่อ เพื่อให้มีความรวดเร็วในการเปลี่ยนแปลงข้อมูล แทนการใช้วิธี HTTP
- Offline แอป Firebase ยังคงตอบสนองได้ขณะออฟไลน์ เนื่องจาก Firebase Realtime Database SDK เก็บข้อมูลไว้ในดิสก์ และเมื่อสร้างการเชื่อมต่อใหม่ อุปกรณ์จะได้รับการเปลี่ยนแปลงของข้อมูลที่เปลี่ยนแปลงไป โดยจะซิงโครไนซ์กับสถานะเซิร์ฟเวอร์ปัจจุบัน
- Accessible from Client Devices ฐานข้อมูลเรียลไทม์ของ Firebase สามารถเข้าถึงได้โดยตรงจากอุปกรณ์พกพาหรือเว็บเบราว์เซอร์ ไม่จำเป็นต้องใช้เซิร์ฟเวอร์แอปพลิเคชัน การรักษาความปลอดภัยและการตรวจสอบความถูกต้องของข้อมูลสามารถใช้งานผ่าน Firebase Realtime Database Security Rules ซึ่งเป็นกฎที่จำเป็นเมื่อมีการอ่านหรือเขียนข้อมูล
- Scale across multiple databases

คุณสามารถสนับสนุนความต้องการข้อมูลของแอปได้โดยแยกข้อมูลออกเป็นหลายกรณีของ ฐานข้อมูลใน Firebase เดียวกัน สามารถเพิ่มประสิทธิภาพการตรวจสอบสิทธิ์ด้วย

Firebase Authentication และตรวจสอบสิทธิ์ผู้ใช้ในกรณีอื่นที่ได้สร้างไว้ ฐานข้อมูลของคุณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมการเข้าถึงข้อมูลในแต่ละฐานข้อมูลด้วยกฎความปลอดภัยของฐานข้อมูลเรียลไทม์ของ Firebase ที่กำหนดเองได้

## 2.11 Firebase

Firebase คือ Platform ที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วนของ Backend หรือ Server side ซึ่งทำให้สามารถ Build Mobile Application ได้อย่างมีประสิทธิภาพ และยังลดเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูลให้อีกด้วย โดยมีทั้งเครื่องมือที่ฟรีและเครื่องมือที่มีค่าใช้จ่าย (สำหรับการ Scale) โดยบริการของ Firebase ที่เราเลือกใช้งานมีดังนี้

### 2.11.1 Cloud Firestore

Cloud Firestore จัดเก็บและซิงค์ข้อมูลระหว่างผู้ใช้และอุปกรณ์ในระดับโลกโดยใช้ฐานข้อมูล NoSQL ที่โฮสต์บนคลาวด์ Cloud Firestore ให้การซิงโครไนซ์แบบสดและการสนับสนุนออฟไลน์พร้อมกับการสืบค้นข้อมูลที่มีประสิทธิภาพ การผสมผสานร่วมกับผลิตภัณฑ์ Firebase อื่น ๆ ช่วยให้คุณสร้างแอปแบบไร้เซิร์ฟเวอร์ได้อย่างแท้จริง

### 2.11.2 Hosting

Hosting ลดความซับซ้อนของเว็บโฮสติ้งของคุณด้วยเครื่องมือที่สร้างขึ้นเฉพาะสำหรับเว็บแอปสมัยใหม่ เมื่อคุณอัปเดตเนื้อหาเว็บของคุณเราจะส่งเนื้อหาเหล่านั้นไปยัง CDN ทั่วโลกของเราโดยอัตโนมัติและมอบใบรับรอง SSL ฟรีเพื่อให้ผู้ใช้ของคุณได้รับประสบการณ์ที่ปลอดภัยเชื่อถือได้และมีเวลาแฝงต่ำไม่ว่าจะอยู่ที่ใดก็ตาม

### 2.11.3 Realtime Database – Realtime Database

Realtime Database – Realtime Database คือฐานข้อมูลดั้งเดิมของ Firebase เป็นโซลูชันที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับแอปบนอุปกรณ์เคลื่อนที่ที่ต้องการสถานะการซิงค์ระหว่างไคลเอนต์แบบเรียลไทม์ เราขอแนะนำ Cloud Firestore แทน Realtime Database สำหรับนักพัฒนาส่วนใหญ่ที่เริ่มโปรเจกต์ใหม่

## 2.12 Sealed Lead Acid Battery

แบตเตอรี่ หรือแบตเตอรี่ชนิดตะกั่วกรดชนิดซีล (Sealed Lead Acid Battery หรือ Valve Regulated Lead Acid Battery นิยมใช้งานกันมากมาย ทั้งในงานขนาดเล็ก ขนาดกลาง และงานที่มีขนาดใหญ่ มาก ๆ โดยแบตเตอรี่ชนิดตะกั่วกรดชนิดซีลมีลักษณะเด่น ดังต่อไปนี้

- มีราคาต่อหน่วยต่ำ
- ต้องการการดูแลรักษาต่ำ
- มีความปลอดภัยสูง
- มีอายุการใช้งานยาวนานสูงสุดถึง 20 ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้งานง่าย และในการใช้งานไม่จำเป็นต้องติดตั้ง BMS (Battery Management System)
- มีความจุให้เลือกใช้งานได้หลากหลายตามขนาด

สำหรับแบตเตอรี่ประเภทนี้ สำหรับงานขนาดเล็กมักถูกใช้งานอยู่ในเครื่องสำรองไฟฉุกเฉินต่างๆ เช่น ไฟฉุกเฉินสำหรับส่องทางเดิน, บ้ายทางออก, บ้ายหนีไฟ, เครื่องสำรองไฟฟ้า (Uninterruptible Power Supply: UPS) อย่างระบบสำรองไฟในอาคาร หรือสำนักงานต่าง ๆ

ในส่วนองงานที่มีขนาดใหญ่ขึ้น แบตเตอรี่สำรองไฟถูกนำมาใช้เป็นแหล่งพลังงานสำรองของระบบสำรองไฟทางการแพทย์ ทั้งในส่วนของเครื่องมือแพทย์ และการสำรองไฟห้องฉุกเฉิน ระบบสำรองไฟสำหรับอาคารโรงพยาบาล โรงไฟฟ้า เขื่อน กังหันน้ำ กังหันลม

นอกจากนี้ ยังมีส่วนสำหรับแผงโซลาร์เซลล์สำหรับผลิตกระแสไฟฟ้า หรือใช้ในระบบจ่ายไฟพลังงานแสงอาทิตย์ ตามสถานที่ที่ไฟฟ้าเข้าไม่ถึง เช่น ไฟส่องสว่างถนน หมู่บ้าน ตามป่าเขา บ้ายจราจรต่างๆ รวมถึงสามารถนำไปใช้งานในพื้นที่โล่งที่ลำบากต่อการเดินไฟเข้าไป เช่น ในท้องไร่ท้องนา สำหรับติดตั้งเครื่องสูบน้ำ หรือเครื่องมือทางการเกษตรอื่นๆ นอกจากนี้ยังถูกใช้ในระบบสำรองไฟของเสาส่งสัญญาณมือถือ วิทยุชุมชน และอุปกรณ์ที่มีความสำคัญต่างๆ สำหรับข้อมูลทางเทคนิคของแบตเตอรี่ประเภทนี้จะแบ่งแยกตามประเภทการใช้งาน



รูปที่ 2.11 Sealed Lead Acid Battery

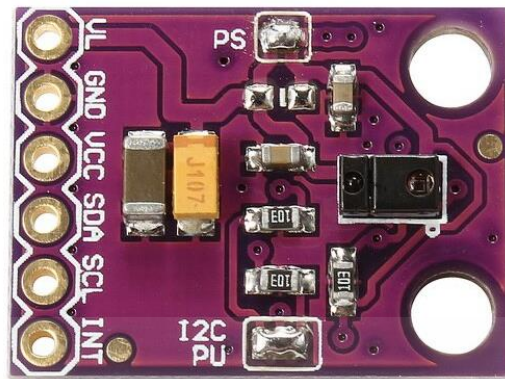
แหล่งที่มา: <https://www.ubuy.co.th/>

### 2.13 เซนเซอร์ตรวจจับสี RGB (APDS-9960)

APDS-9960 มีคุณสมบัติการตรวจจับท่าทาง การตรวจจับความใกล้เคียง ตรวจจับแสงสภาพแวดล้อมแบบดิจิทัล (ALS) และการรับรู้สี (RGBC) รวม IR LED และไดร์เวอร์ LED เซนเซอร์นี้จะใช้การสื่อสารแบบ I2C

ช่อง R, G, B, C แต่ละช่องนั้นจะมีตัวกรองปิดกัน UV และ IR และแปลงข้อมูลออกมาเป็น 16 บิตซึ่งทำให้สามารถรับค่าแสงโดยรอบและรับรู้แสงได้อย่างแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 เซนเซอร์ตรวจวัดสี RGB

แหล่งที่มา : <https://th.aliexpress.com/item/32855567259.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# วิธีการดำเนินงาน

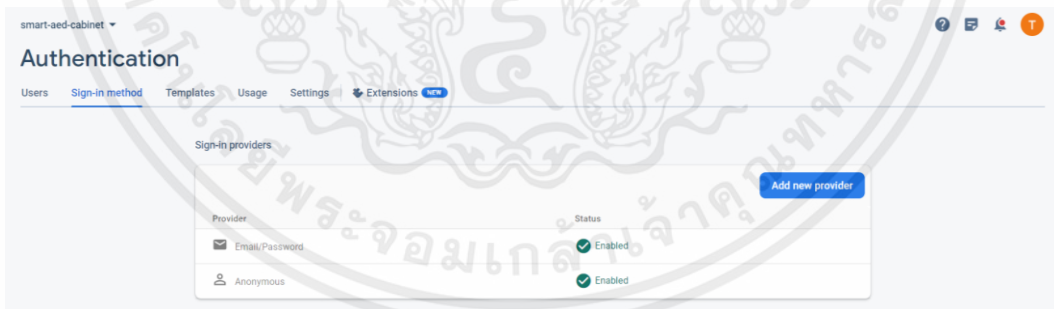
### 3.1 ขั้นตอนการดำเนินงาน

- เลือกหัวข้อโครงการที่สนใจและนำไปปรึกษากับอาจารย์ที่ปรึกษาโครงการงาน
- ศึกษาข้อมูลที่เกี่ยวข้องกับโครงการ และรายละเอียดของอุปกรณ์ที่ต้องใช้
- ออกแบบการจัดวางตำแหน่งของอุปกรณ์ และเขียนโปรแกรมเพื่อควบคุมการทำงาน
- ประกอบชิ้นงาน
- ทดสอบการทำงานของอุปกรณ์และระบบ
- เก็บข้อมูล
- สรุปผลการทดลอง
- จัดทำรูปเล่มรายงานฉบับสมบูรณ์

### 3.2 ขั้นตอนการตั้งค่า

#### 3.2.1 การตั้งค่า Firebase Authentication

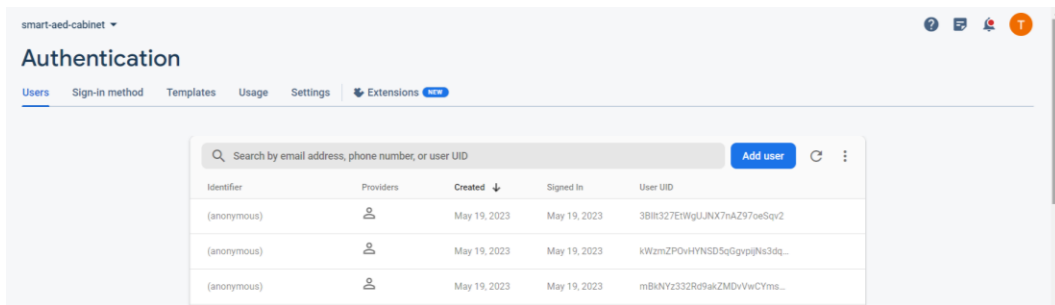
- หน้า Sign-in method ให้เลือก Add new provider เพื่อทำการเลือกวิธีสำหรับการยืนยันตัวตน สำหรับการเข้าใช้งาน



รูปที่ 3.1 การตั้งค่า Firebase Authentication

- หน้า User จะแสดงผู้ที่เข้าใช้งาน และสามารถเพิ่มผู้ใช้งานได้โดย กดที่ Add Users ตามที่เราต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การตั้งค่าหน้า Sign-in method

### 3.2.2 การตั้งค่า Firebase Realtime database

- หน้า Rules ให้ทำการตั้งค่าดังนี้



รูปที่ 3.3 การตั้งค่าหน้า User

### 3.2.3 การตั้งค่า Firebase Storage

- หน้า Rules ให้ทำการตั้งค่าดังนี้



รูปที่ 3.4 การตั้งค่าหน้า Rules

### 3.2.4 การตั้งค่า Firebase Hosting

- ทำการติดตั้งโปรแกรม Visual Studio Code และ Node.js (เวอร์ชัน LTS ) หลังจากนั้นติดตั้งส่วนขยาย node.js extension pack ลง VS code
- ติดตั้ง Firebase tool โดยใช้คำสั่ง `npm -g install firebase-tools`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PS C:\webapp> cd \test
PS C:\test> npm -g install firebase-tools
npm WARN deprecated @npmcli/move-file@2.0.1: This functionality has been moved to @npmcli/fs
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

added 79 packages, removed 115 packages, changed 573 packages, and audited 653 packages in 37s

51 packages are looking for funding
  run `npm fund` for details

1 moderate severity vulnerability

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
PS C:\test>

```

รูปที่ 3.5 การติดตั้ง firebase tool

- หลังจากนั้น ใช้คำสั่ง firebase login

```

PS C:\test> firebase login
Already logged in as ford14.ff@gmail.com
PS C:\test>

```

รูปที่ 3.6 login firebase

- หลังจากนั้น ใช้คำสั่ง firebase init

```

PS C:\test> firebase init

#####
##          ##          ##          ##          ##          ##          ##          ##
##          ##          ##          ##          ##          ##          ##          ##
#####
##          ##          ##          ##          ##          ##          ##          ##
##          ##          ##          ##          ##          ##          ##          ##
##          ##          ##          ##          ##          ##          ##          ##
#####

```

รูปที่ 3.7 เริ่มต้น firebase

- หลังจากนั้นทำการตั้งค่าดังนี้
- Are you ready to proceed? เลือก Y
- Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. เลือก Hosting:Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
  ( ) Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance
  ( ) Firestore: Configure security rules and indexes files for Firestore
  ( ) Functions: Configure a Cloud Functions directory and its files
> (*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
  ( ) Hosting: Set up GitHub Action deploys
  ( ) Storage: Configure a security rules file for Cloud Storage
  ( ) Emulators: Set up local emulators for Firebase products
(Move up and down to reveal more choices)

```

รูปที่ 3.8 ตัวเลือกฟีเจอร์ต่าง ๆ

- เลือกโปรเจกต์ที่เราต้องการจะใช้

```

? Please select an option: (Use arrow keys)
> Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project

```

รูปที่ 3.9 เลือกโปรเจกต์ที่ต้องการใช้

ทำการตั้งค่า Hosting ดังนี้

1. What do you want to use as your public directory? Public
2. Configure as a single-page app (rewrite all urls to /index.html)? Yes
3. Set up automatic builds and deploys with GitHub? No

เมื่อเสร็จแล้ว จะได้ไฟล์ index.html มาอยู่ในตำแหน่งที่เราได้กำหนดไว้

```

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
+ Wrote public/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
i Writing gitignore file to .gitignore...

+ Firebase initialization complete!

```

รูปที่ 3.10 Hosting setup

- หลังจากนั้น เพิ่มส่วน config ของ index.html ลงไปโดยที่ Project setting เลือก CDN  
จะได้ config ที่เป็น script ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### SDK setup and configuration

npm
  CDN
  Config

Load Firebase JavaScript SDK libraries from the CDN (content delivery network). [Learn more](#).

Copy and paste these scripts into the bottom of your `<body>` tag, but before you use any Firebase services:

```

<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/
  import { getAnalytics } from "https://www.gstatic.com/firebasejs/9
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  const firebaseConfig = {
    apiKey: "AIzaSyCgWOCsOD1AqLBBAEXBWWt71nt_002iVVI",
    authDomain: "smart-aed-cabinet.firebaseio.com",
    databaseURL: "https://smart-aed-cabinet-default-rtdb.firebaseio.com",
    projectId: "smart-aed-cabinet",
    storageBucket: "smart-aed-cabinet.appspot.com",
    messagingSenderId: "275573352069",
    appId: "1:275573352069:web:344d04c5a0b7c1ae42abec",
    measurementId: "G-010Q2XKSBN"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const analytics = getAnalytics(app);
</script>

```

รูปที่ 3.11 SDK setup and configuration

### 3.3 การคำนวณเลือกอุปกรณ์ในการสร้างชิ้นงาน

แหล่งจ่ายพลังงาน ESP32-CAM จากการการวัดค่ากำลังงานที่ใช้ ใช้พลังงาน 1.5 W ใช้กระแสสูงสุด 250 mA กำหนดให้สำรองพลังงาน 24 ชั่วโมง สำหรับการทำงานต่อเนื่อง ดังนั้นจะใช้พลังงาน

$$1.5(W) \times 24(h) = 36(Wh)$$

- แบตเตอรี่แบบลิเทียมไอออน มีแรงดันไฟฟ้า 3.7 โวลต์ มีค่า DoD ประมาณ 80 %

ดังนั้นเลือกใช้แบตเตอรี่ดังนี้

$$\frac{\text{พลังงานที่ใช้}(Wh) \times 1.25}{\text{แรงดันแบตเตอรี่}(V)} = \text{ความจุแบตเตอรี่}(Ah)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{36(Wh) \times 1.25}{3.7(V)} = 12.2(Ah)$$

จึงเลือกใช้ แบตเตอรี่ลิเทียม 3.7 V, 3350 mAh ต่อขนานกัน 4 ก้อน

- แผงโซลาร์เซลล์ จากใช้พลังงาน 36 Wh มีค่า Insolation = 5h/day

$$\frac{\text{พลังงานที่ใช้}(Wh)}{\text{Insolation}(h/day)} = \text{กำลังงานที่แผงจะผลิตได้}(W)$$

$$\frac{36(Wh)}{5(h)} = 7.2(W)$$

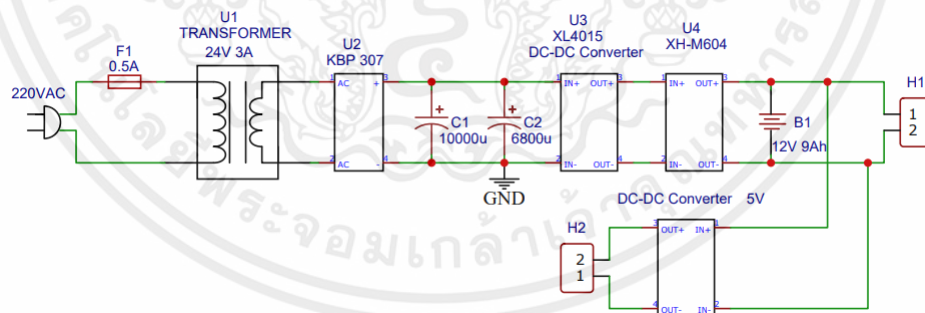
แผงจะต้องให้กำลังงาน 7.2 W เนื่องจาก Control charger รับแรงดันได้ 6 V ดังนั้นจึงเลือกใช้แผง 12W 6V

### 3.4 หลักการทำงาน

#### 3.4.1 วงจรการทำงาน

##### 3.4.1.1 แหล่งจ่ายไฟสำหรับตู้ AED

เริ่มต้นจากใช้หม้อแปลงแปลงแรงดัน 220 V ให้เป็นแรงดัน 24 V และใช้ Bridge Diode แบบ Full wave ในการ Rectifier และใช้ตัวเก็บประจุ Filter แรงดันให้เรียบขึ้น และต่อเข้ากับ Module XL4015 DC-DC Buck Converter เพื่อลิมิตกระแสและแรงดัน จากนั้นต่อเข้ากับ Module XH-M604 ซึ่งเป็น Module ควบคุมการชาร์จแบตเตอรี่ จากนั้นต่อเข้ากับแบตเตอรี่ และต่อเอาต์พุตเพื่อไปใช้เป็นแหล่งจ่ายไฟให้กับอุปกรณ์ที่อยู่ในตู้ใส่เครื่อง AED

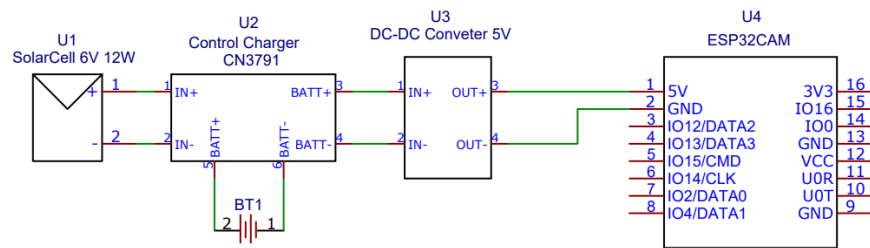


รูปที่ 3.12 วงจรจ่ายไฟสำหรับตู้ AED

##### 3.4.1.2 แหล่งจ่ายไฟสำหรับ ESP32-CAM โดยใช้ Solarcell

เมื่อแผงโซลาร์เซลล์จ่ายพลังงานไปให้แก่ solar control charger ตัว solar control charger จะทำหน้าที่จ่ายพลังงานเพื่อไปชาร์จแก่แบตเตอรี่และทำหน้าที่จัดการจ่ายพลังงาน จากแบตเตอรี่เพื่อไปจ่ายให้แก่ ESP32 Cam โดยผ่านโมดูลเรกูเรเตอร์ เพื่อแปลงแรงดันไปเป็น 5 V แก่ ESP32- CAM เนื่องจาก solar control charge จ่ายแรงดันออกมา 6V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

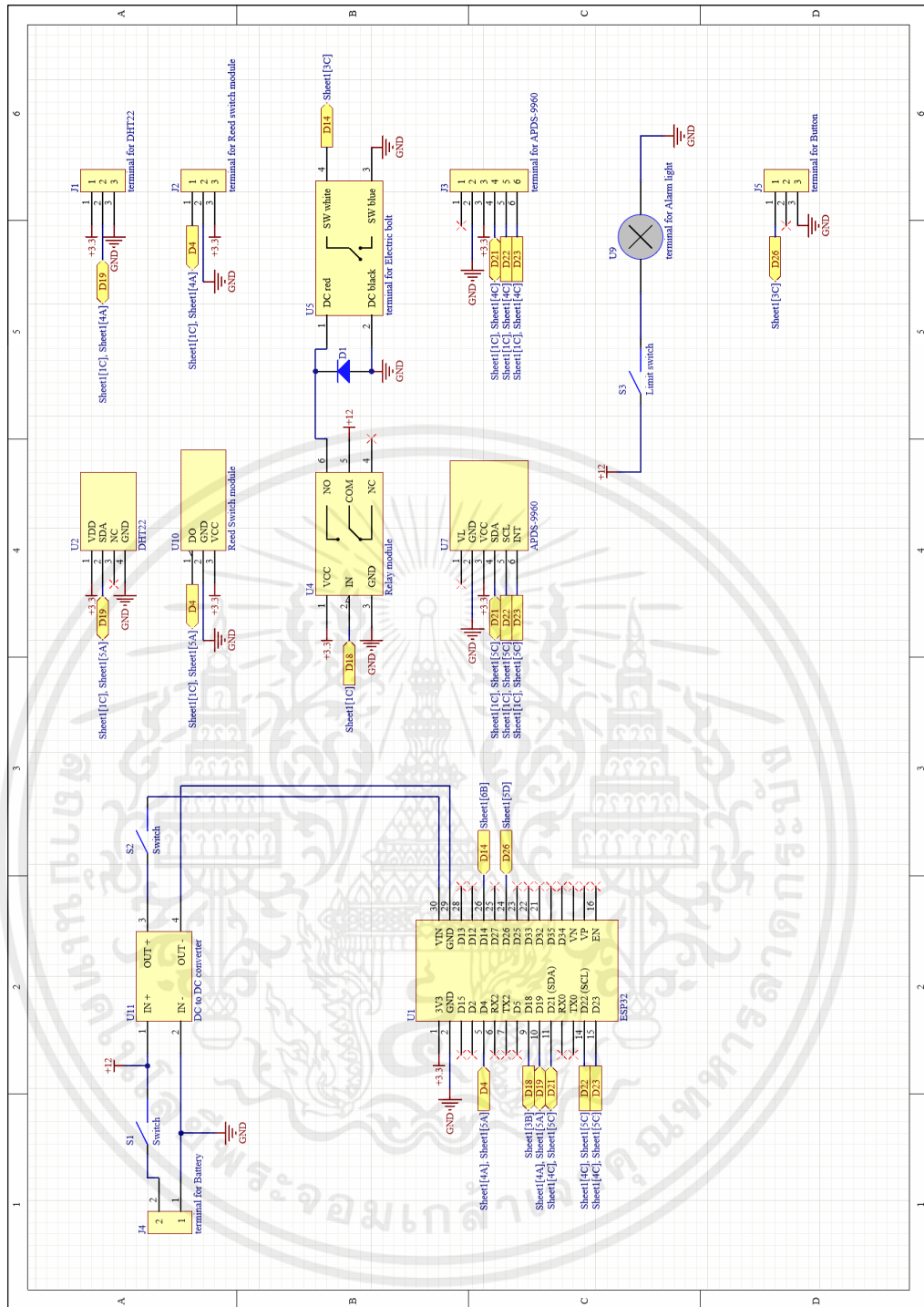


รูปที่ 3.13 วงจรจ่ายไฟสำหรับ ESP32-CAM

### 3.4.1.3 ส่วนควบคุมภายในตู้

ประกอบด้วย ESP32 ทำหน้าที่อ่านค่าเซนเซอร์ อ่านค่าสวิตช์ ควบคุมกลอนไฟฟ้า รับส่งข้อมูลกับมือถือผ่านบลูทอธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 วงจรส่วนควบคุมภายในตู้

### 3.4.2 การถ่ายภาพจาก ESP-32CAM ไปเก็บ Realtime Database

ในส่วนของการถ่ายภาพจาก ESP32-CAM รับค่าจากสถานะจาก Realtime database ว่า ตู้ปิดอยู่จะให้ทำการถ่ายภาพแล้วส่งไปเก็บไว้ยังการ์ดความจำ โดยกำหนดให้ถ่ายภาพในทุก ๆ 1 วินาทีโดยมีการวนบันทึกรูปภาพซ้ำตามจำนวนที่ได้กำหนดไว้เรื่อย ๆ และเมื่อพบว่าค่าสถานะที่รับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่าตู้เปิดอยู่ จะทำการถ่ายภาพทุก 1 วินาที แล้วส่งรูปภาพที่ถ่ายขณะนั้น พร้อมกับการส่งรูปจากการ์ดความจำไปยัง Storage และพบเมื่อพบว่าสถานะตู้ปิด ระบบก็จะกลับมาถ่ายภาพแล้วส่งไปเก็บไว้ยังการ์ดความจำ

### 3.4.3 การใช้เซนเซอร์วัดค่าต่าง ๆ และนำข้อมูลไปเก็บที่ Realtime Database

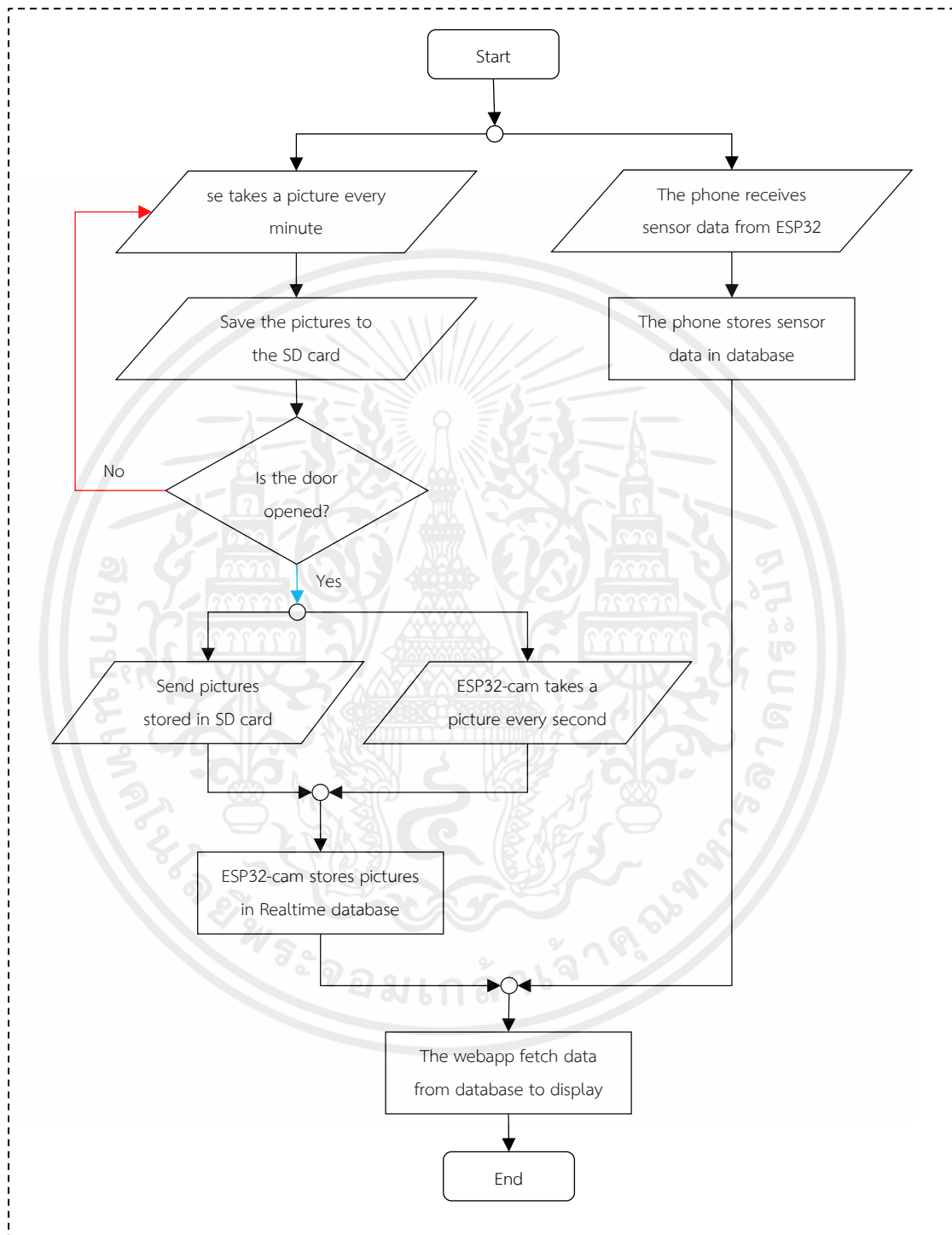
การใช้เซนเซอร์ตรวจวัดอุณหภูมิ ความชื้น เซนเซอร์วัดสี และreed switch ส่งค่าที่ตรวจวัดได้ผ่าน ESP-32 ไปยังโทรศัพท์มือถือในทุก ๆ 2 วินาที โดยควบคุมด้วยแอปบนโทรศัพท์จากนั้นนำข้อมูลที่ได้ไปเก็บที่ Firebase ในส่วนของ Realtime Database

### 3.4.4 การนำข้อมูลที่เก็บไว้ใน Realtime Database ไปแสดงที่ Web Hosting

เป็นการนำข้อมูลที่เก็บไว้ใน Firebase ในส่วนของ Realtime Database ข้อมูลล่าสุดมาแสดงผลบน Web Hosting โดยข้อมูลที่นำมาแสดงบน Web Hosting จะมี 4 ส่วนได้แก่ ค่าอุณหภูมิและความชื้น สถานะการเปิดปิดของประตูตู้ที่ใส่เครื่อง AED และเวลาที่แสดงข้อมูลปัจจุบัน โดยข้อมูลจะมีการอัปเดตตามข้อมูลล่าสุดในแต่ละส่วนที่ถูกนำมาเก็บใน Realtime Database

### 3.5 แผนผังการทำงาน Flow charts

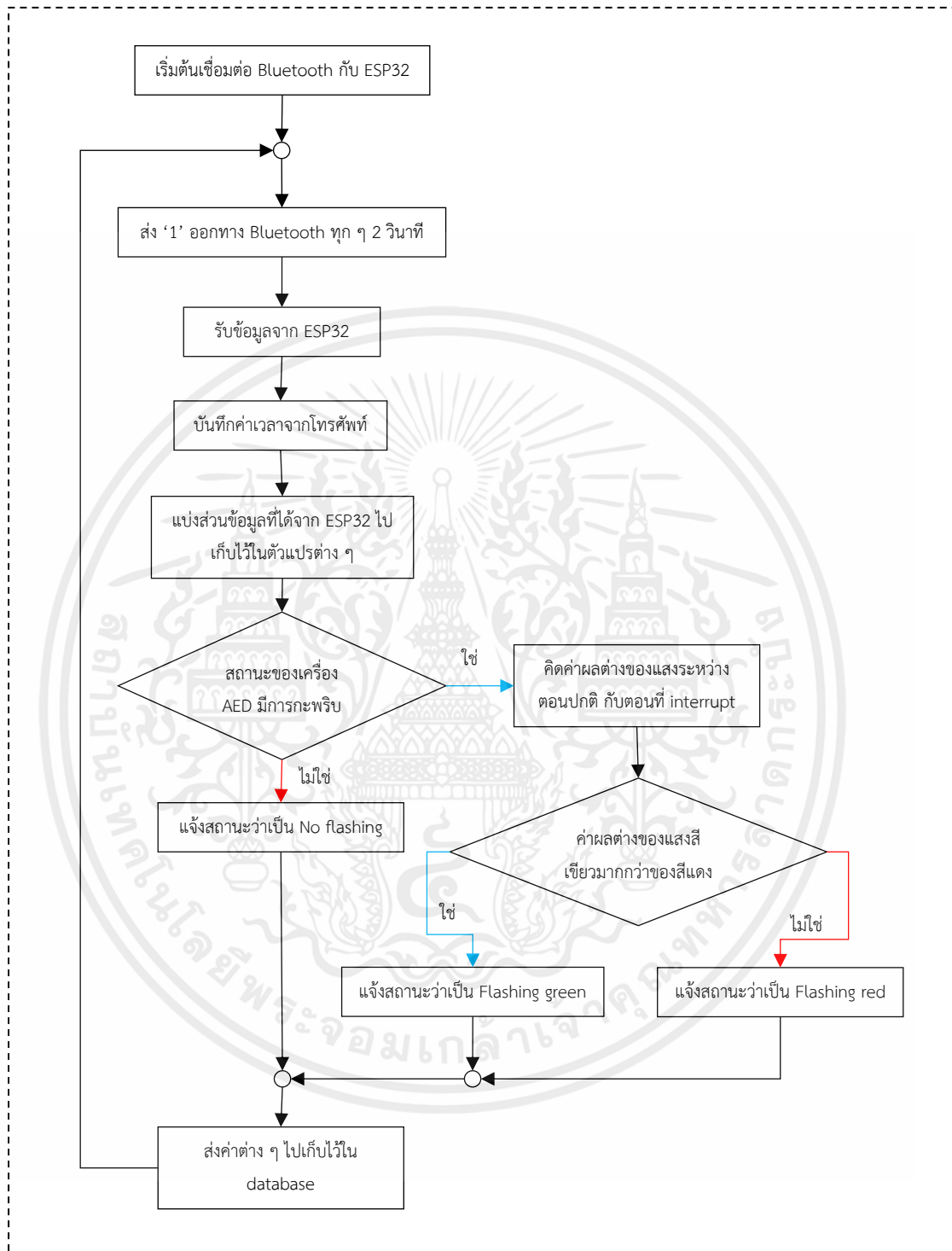
#### 3.5.1 การทำงานของระบบโดยรวม



รูปที่ 3.15 แผนภาพการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

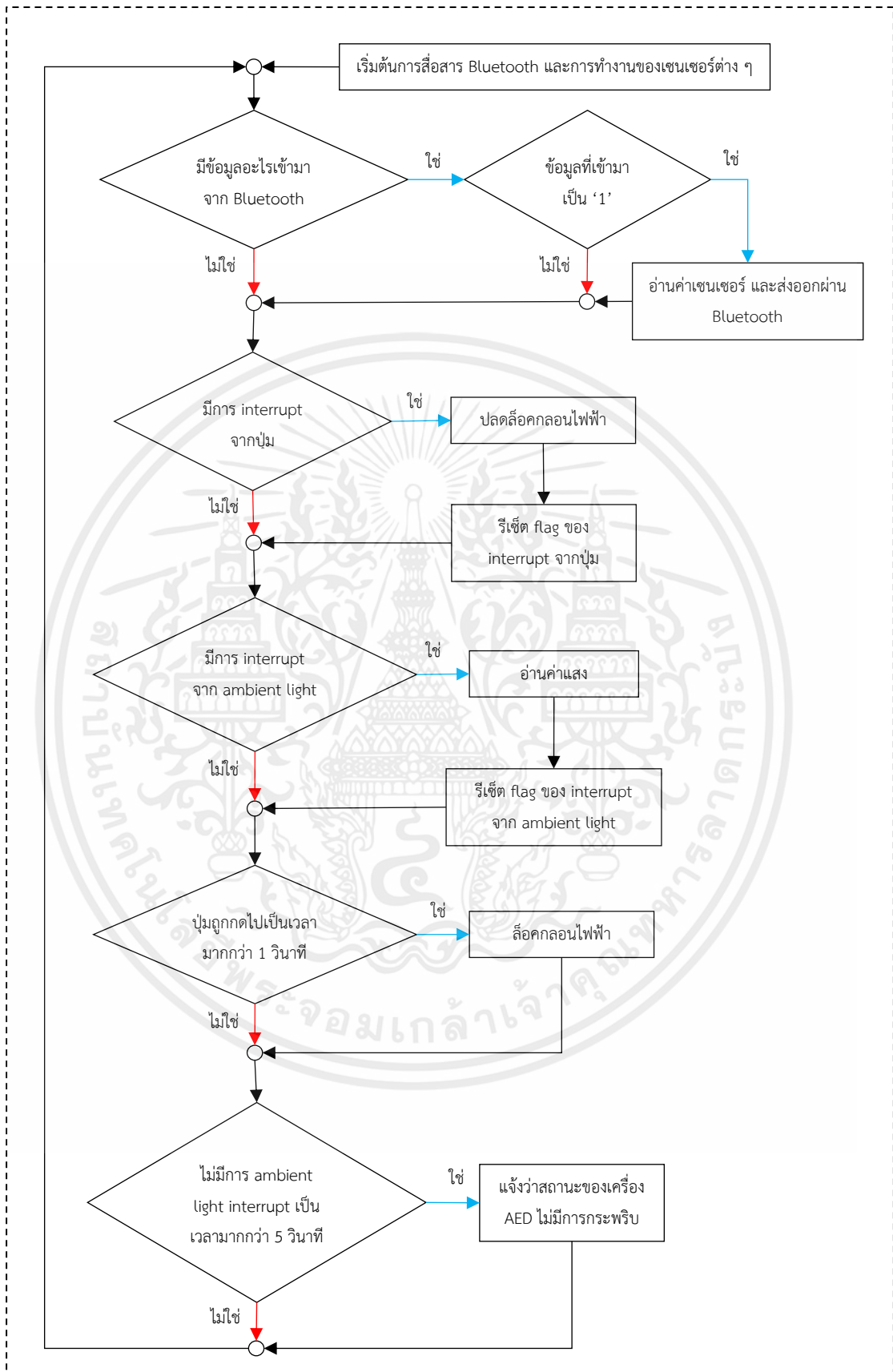
## 3.5.2 การทำงานของโทรศัพท์



รูปที่ 3.16 แผนภาพแสดงการทำงานของโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

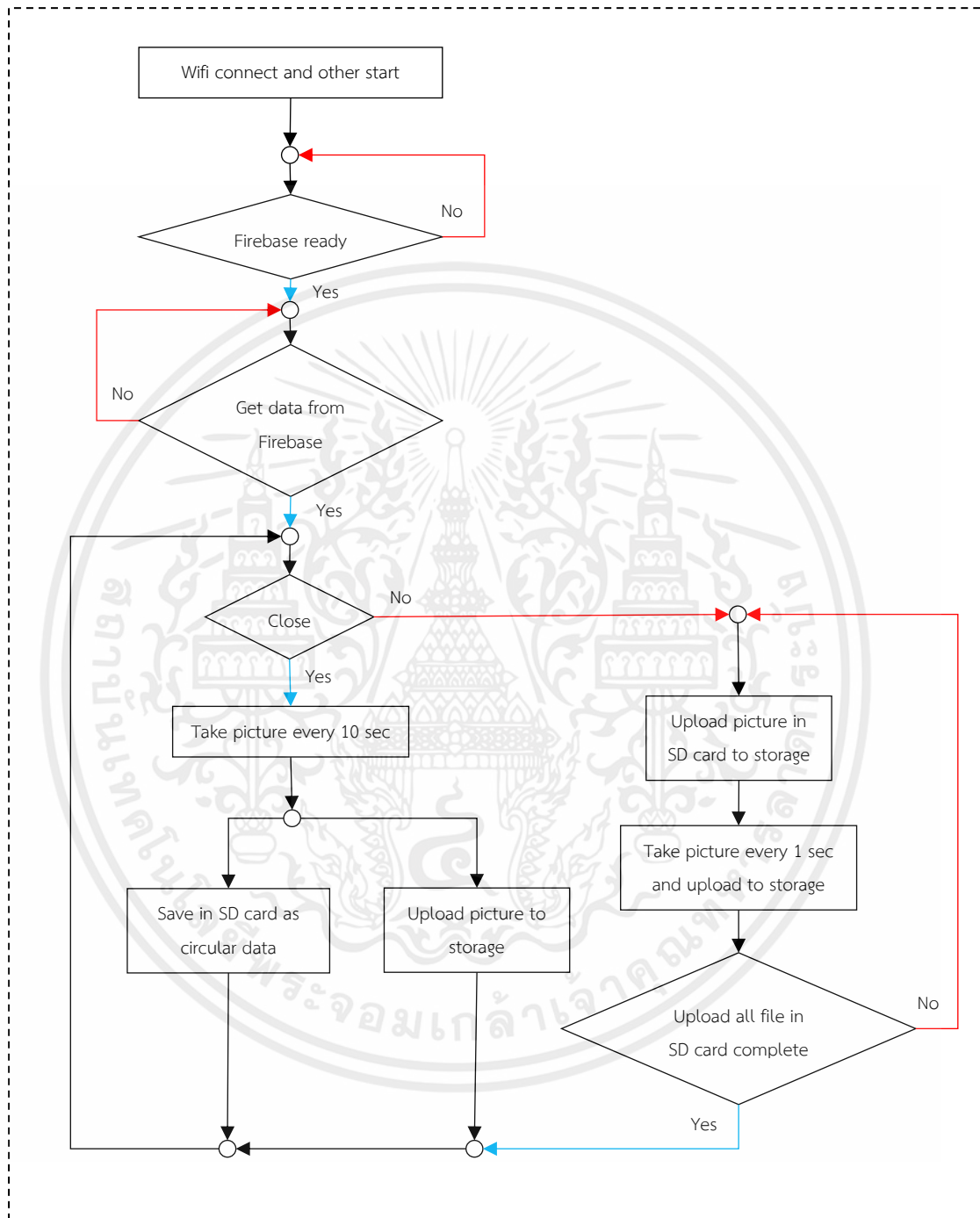
3.5.3 การทำงานของ ESP32



รูปที่ 3.17 แผนภาพแสดงการทำงานของ ESP32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

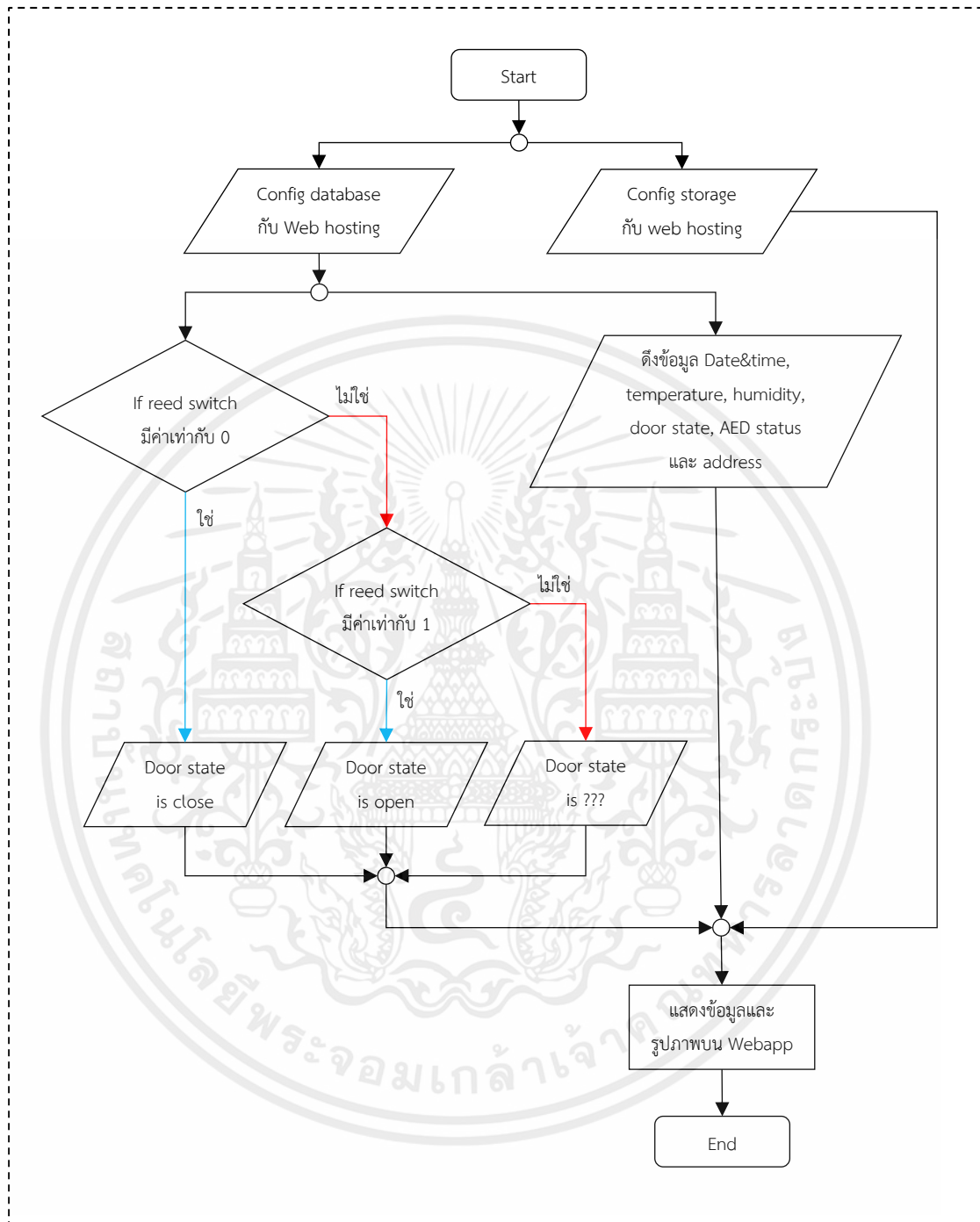
## 3.5.4 การทำงานของ ESP32-CAM



รูปที่ 3.18 แผนภาพแสดงการทำงานของ ESP32-CAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.5 การทำงานของ Webapp



รูปที่ 3.19 แผนภาพแสดงการทำงานของ Webapp

## 3.6 รูปภาพแสดงส่วนต่าง ๆ ในระบบ

### 3.6.1 ตู้ AED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



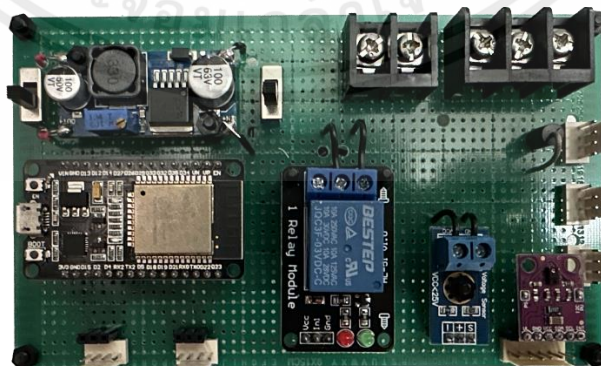
รูปที่ 3.20 ตู้ AED ขณะปิดประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 ตู้ AED ขณะเปิดประตู

### 3.6.2 บอร์ดคอนโทรลในตู้ AED



รูปที่ 3.22 บอร์ดคอนโทรลในตู้ AED

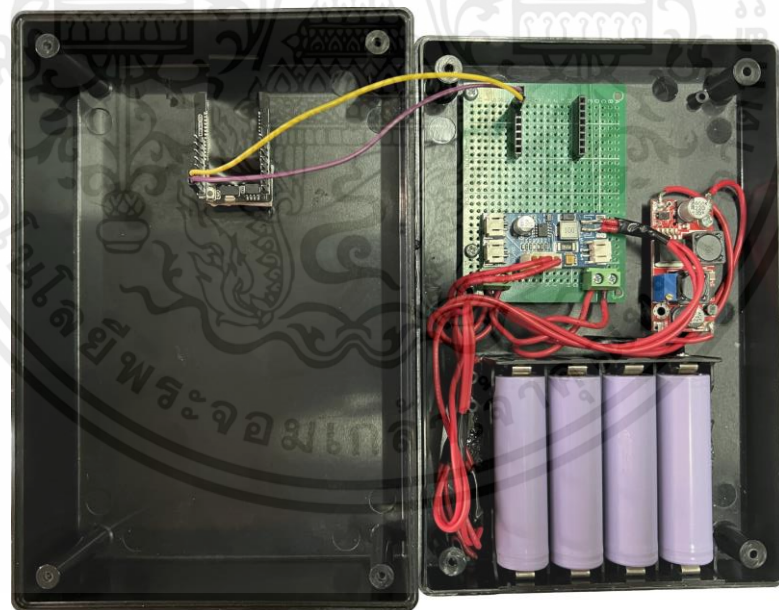
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6.3 ตัวกล่อง ESP-CAM และระบบต่าง ๆ พร้อมโซล่าเซลล์



รูปที่ 3.23 ตัวกล่อง ESP-CAM และระบบต่าง ๆ พร้อมโซล่าเซลล์

### 3.6.4 ภายในตัวกล่องESP-CAM



รูปที่ 3.24 ภายในตัวกล่องESP-CAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.7 เครื่องมือที่ใช้ในการทำงาน

- Digital multimeter
- หัวแร้ง
- คัดเตอร์
- ไชควง
- สว่าน
- คีมตัดลวด
- ที่ดูดตะกั่ว
- ปากกามาร์เกอร์

### 3.8 อุปกรณ์ที่ใช้ในการทำงาน

บอร์ดไขปลา .....	1 แผ่น
Bridge diode KBP307 .....	1 ตัว
Capacitor 10,000 $\mu$ F .....	1 ตัว
Capacitor 6,800 $\mu$ F .....	1 ตัว
Terminal block .....	2 ตัว
ตะกั่ว .....	1 ม้วน
สายไฟ 18 AWG .....	1 ม้วน
Module XL4015 DC-DC Buck Converter .....	1 ตัว
Module XH-M604 .....	1 อัน
แบตเตอรี่ชนิด Sealed Lead-Acid ขนาดความจุ 12V 9Ah .....	1 ก้อน
DC Converter 5 V .....	2 ตัว
ESP-32 .....	1 ตัว
ESP-32CAM .....	1 ตัว
ลิเทียมแบตเตอรี่ 18650 3.7V 3350 mAh .....	4 ก้อน
โมดูลชาร์ตแบตเตอรี่ลิเทียม CN3791 .....	1 อัน
รางถ่านลิเทียม .....	1 อัน
แผงโซลาร์เซลล์ 12 V 6 W .....	1 แผง
DHT22 .....	1 อัน
Reed switch module .....	1 อัน
Relay module .....	1 ตัว
เซนเซอร์ตรวจจับสี RGB (APDS-9960) .....	1 อัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.9 โปรแกรมที่ใช้ในการทำงาน

- Arduino IDE
- MIT App Inventor
- Visual Studio Code
- EasyEDA



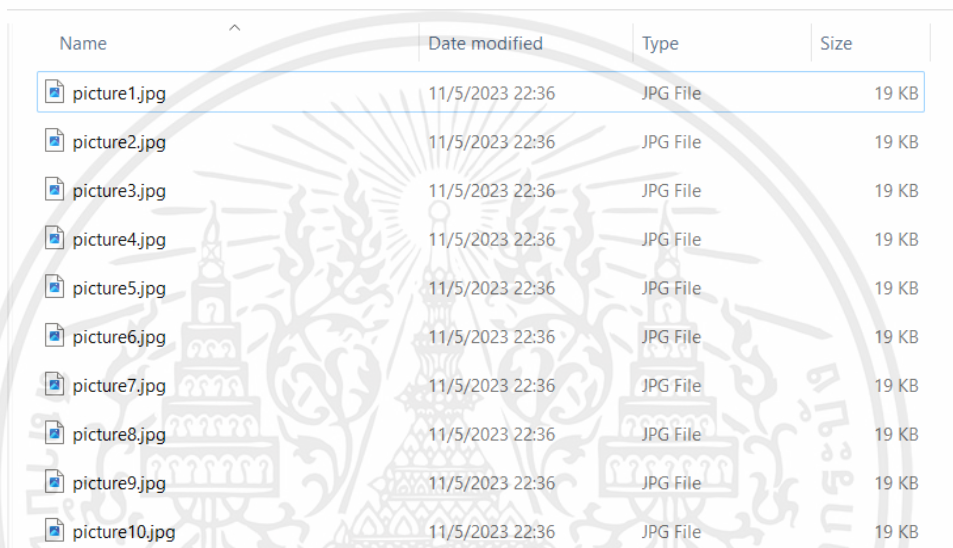
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

#### 4.1 การถ่ายภาพจาก ESP-32CAM ไปเก็บ Realtime Database

4.1.1 การถ่ายภาพเมื่อตู้ AED มีสถานะปิดอยู่จะถ่ายภาพสภาพแวดล้อมด้วยความเร็วตามที่ตั้งไว้ 1 ภาพต่อวินาทีและบันทึกภาพไว้ในการ์ดความจำ จนกว่าจะมีการเปิดตู้

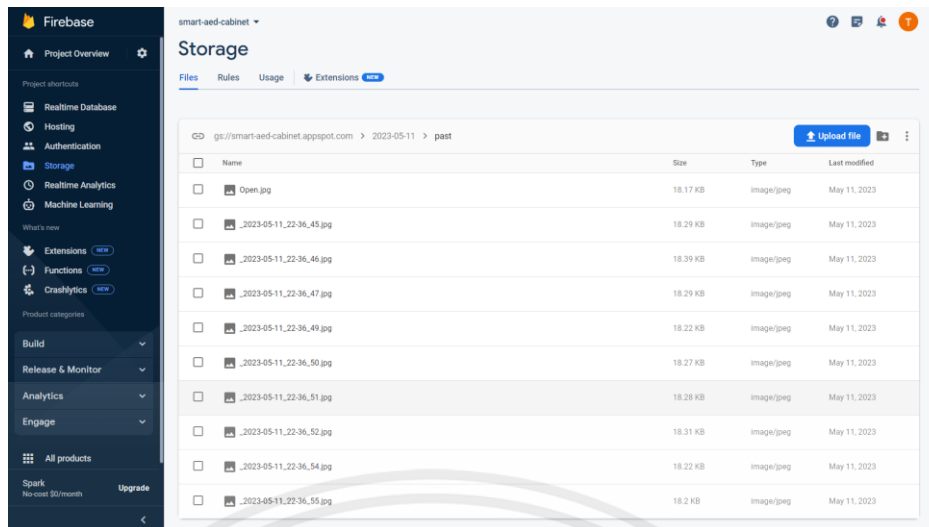


Name	Date modified	Type	Size
picture1.jpg	11/5/2023 22:36	JPG File	19 KB
picture2.jpg	11/5/2023 22:36	JPG File	19 KB
picture3.jpg	11/5/2023 22:36	JPG File	19 KB
picture4.jpg	11/5/2023 22:36	JPG File	19 KB
picture5.jpg	11/5/2023 22:36	JPG File	19 KB
picture6.jpg	11/5/2023 22:36	JPG File	19 KB
picture7.jpg	11/5/2023 22:36	JPG File	19 KB
picture8.jpg	11/5/2023 22:36	JPG File	19 KB
picture9.jpg	11/5/2023 22:36	JPG File	19 KB
picture10.jpg	11/5/2023 22:36	JPG File	19 KB

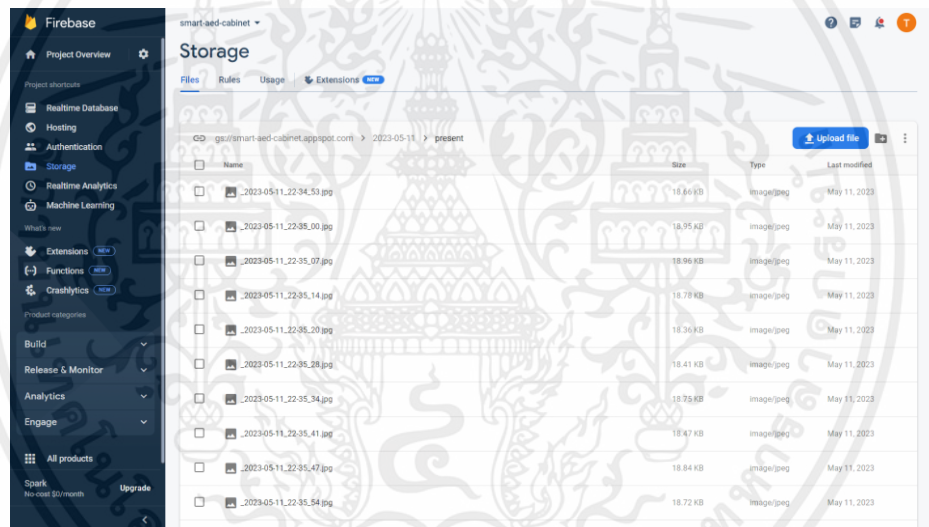
รูปที่ 4.1 ภาพที่ถูกบันทึกไว้ในการ์ดความจำ

เมื่อตู้มีสถานะปิดอยู่หรือไม่ได้ใช้งานโดยรับค่าสถานะจาก Realtime database พบว่าสามารถถ่ายรูปและบันทึกรูปลงการ์ดความจำตามที่กำหนดไว้ได้ และเมื่อบันทึกจำนวนภาพตามที่กำหนดครบแล้วสามารถกลับมาบันทึกรูปภาพซ้ำตามที่กำหนดไว้ได้

4.1.2 การถ่ายภาพเมื่อตู้ AED มีสถานะเปิดอยู่จะถ่ายภาพสภาพแวดล้อมด้วยความเร็วที่ 1 ภาพต่อวินาที จนกว่าจะมีการปิดตู้



รูปที่ 4.2 รูปจากการคัดความจำที่ถูกส่งไปเก็บยัง Storage



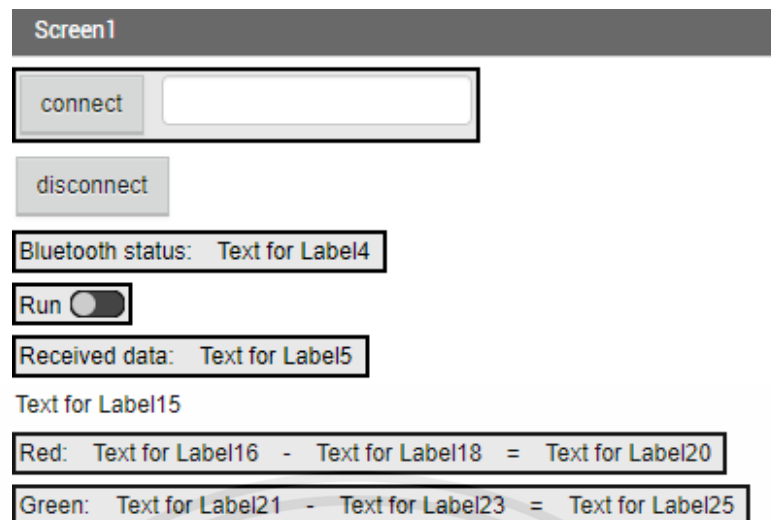
รูปที่ 4.3 รูปที่ถ่ายขณะที่ตู้เปิดถูกส่งไปเก็บยัง Storage

เมื่อตู้มีสถานะเปิดอยู่ โดยรับค่าสถานะจาก Realtime database พบว่าสามารถส่งภาพที่บันทึกไว้ในการ์ดความจำและภาพที่ถูกลำยขณะที่ตู้ถูกเปิดไปยัง Storage ได้

## 4.2 การอ่านค่าเซนเซอร์ และนำข้อมูลไปเก็บที่ Realtime Database

### 4.2.1 หน้าตาแอปที่ควบคุมการส่งข้อมูลไปเก็บที่ Realtime Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 หน้าแอปที่ควบคุมการรับส่งข้อมูลไปที่ Realtime database

เมื่อมีข้อมูลจาก ESP32 ถูกรับเข้ามา จะนำมาแสดงที่หน้าแอป จากนั้นจะมีการแปรผลจากข้อมูลของเซนเซอร์สี่เพื่อวิเคราะห์สีของสถานะของเครื่อง AED และข้อมูลก็จะถูกส่งไปยัง Realtime database

#### 4.2.2 การส่งค่าสถานะต่าง ๆ วันที่และเวลาไปเก็บที่ Realtime Database

```
bluetoothv2_2debug
  00_time: ""2023.04.29 ค.ศ. at 15:48:10""
  01_temperature: "24.3"
  02_humidity: "71"
  03_reed_switch: "Close"
  04_lock_switch: "Lock"
  11_button_one_flag: "0"
  13_aed_status: ""Flashing green""
```

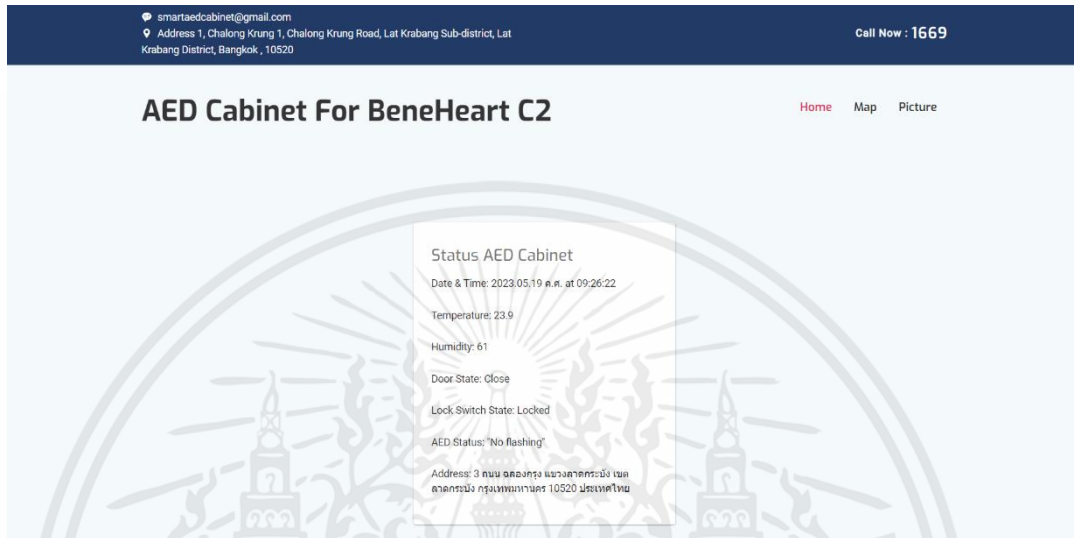
รูปที่ 4.5 ตัวอย่างการเก็บข้อมูลที่ Realtime database

รูปนี้แสดงถึงข้อมูลที่ส่งมาจากโทรศัพท์ที่ถูกควบคุมด้วยแอป โดยจะแสดงถึงเวลาของข้อมูลและค่าสถานะต่าง ๆ ภายในตู้

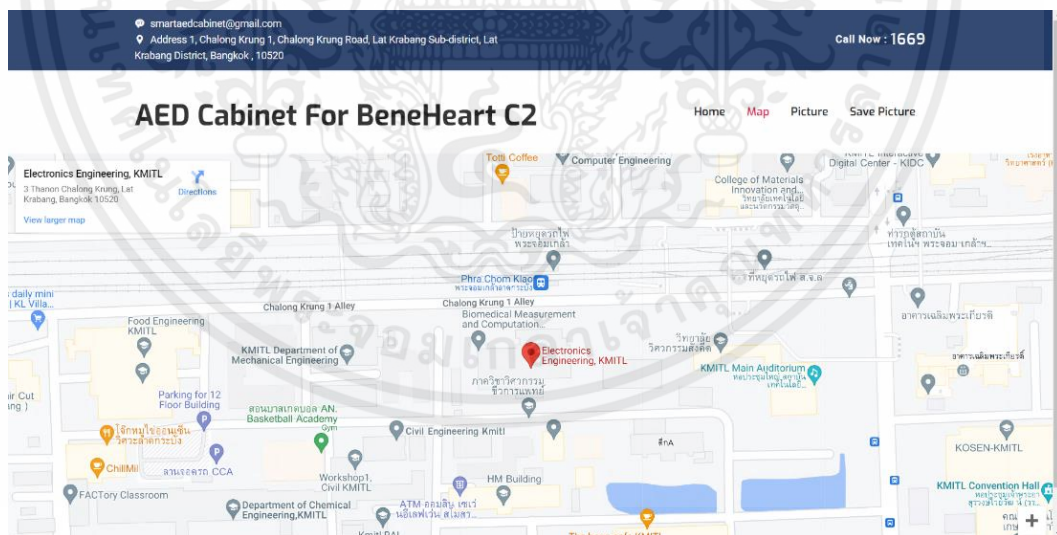
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3 การนำข้อมูลที่เก็บไว้ใน Realtime Database ไปแสดงที่ Web Hosting

สามารถดึงข้อมูลอุณหภูมิ ความชื้น สถานะความพร้อมใช้งานของเครื่อง AED สถานะการเปิด-ปิดของประตู วันที่และเวลา รวมไปถึงรูปภาพที่บันทึกขณะมีการเปิดปิดตู้ ที่ถูกจัดเก็บไว้ใน Realtime Database มาแสดงผลบนเว็บไซต์ และสามารถอัปเดตข้อมูลล่าสุดได้อัตโนมัติ



รูปที่ 4.6 ตัวอย่างการแสดงผลสถานะต่าง ๆ ของเครื่อง AED บนเว็บไซต์



รูปที่ 4.7 การแสดงตำแหน่งที่ตั้งของเครื่อง AED บนเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

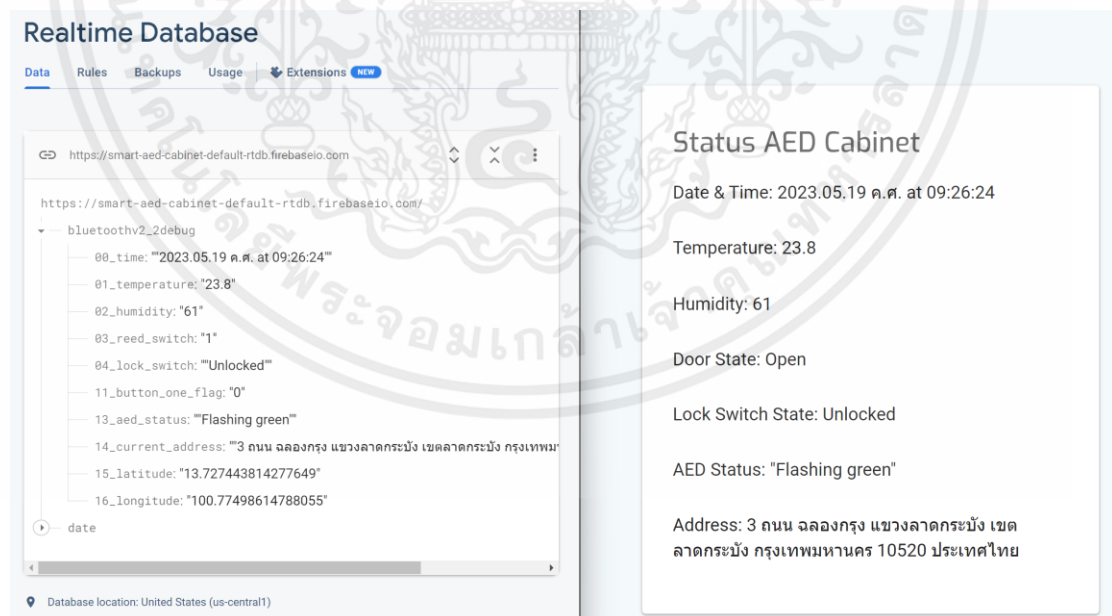


รูปที่ 4.8 ตัวอย่างการแสดงรูปภาพที่บันทึกขณะที่มีการเปิดปิดตู้บนเว็บไซต์

#### 4.4 การทำงานต่าง ๆ ของระบบ

##### 4.4.1 ตอนที่ตู้ถูกเปิด

ข้อมูล Reed switch และ Lock switch จากตู้จะถูกส่งไปยัง Realtime database และจะถูกนำไปแสดงบน webapp ซึ่งแสดงถึงว่าตู้ได้ถูกเปิดแล้ว

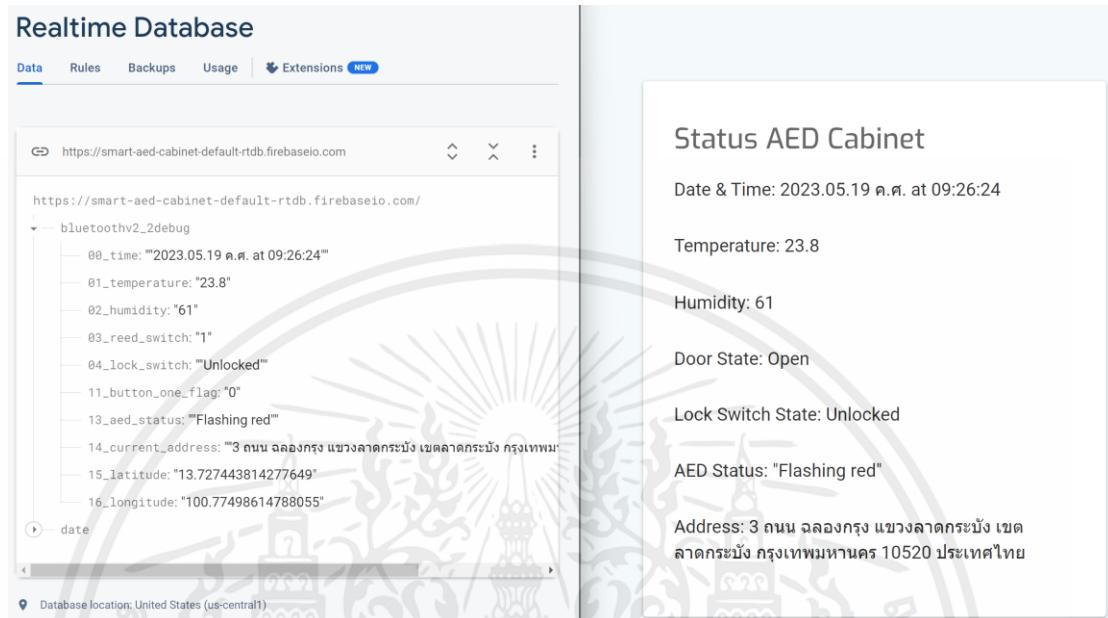


รูปที่ 4.9 Realtime database และ webapp ตอนที่ตู้ถูกเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.2 การรายงานสถานะของเครื่อง AED

ค่าสถานะของเครื่อง AED จะถูกอ่านโดยเซนเซอร์วัดสี จากนั้นจะถูกประมวลผลและส่งไปยัง Realtime database และจะถูกนำไปแสดงบน webapp



รูปที่ 4.10 Realtime database และ webapp แสดงสถานะของเครื่อง AED ว่า Flashing red

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลการทดลอง

### 5.1 สรุปผลการทดลอง

จากผลการทดลองที่ 4.1 พบว่าเมื่อประตูตู้ใส่เครื่อง AED ปิดอยู่สามารถถ่ายภาพในทุก 1 วินาที แล้วส่งไปเก็บที่การ์ดความจำได้อย่างปกติ แต่เมื่อมีการเปิดประตูตู้ใส่เครื่อง AED จะเปลี่ยนเป็นถ่ายรูปทุก 1 วินาทีและส่งรูปภาพไปพร้อมกับส่งรูปภาพที่เก็บไว้ในการ์ดความจำไปเก็บที่ storage ไปเรื่อย ๆ ได้จนกว่าประตูจะถูกปิด และเมื่อปิดแล้วสามารถกลับมาถ่ายรูปและบันทึกไว้ในการ์ดความจำได้

จากผลการทดลองที่ 4.2 พบว่าสามารถอ่านค่าสถานะที่ตัวเครื่อง AED แสดง สามารถอ่านค่าเซนเซอร์ต่าง ๆ พร้อมกับเวลาที่อ่านข้อมูล แล้วส่งไปเก็บใน Realtime database ได้

จากผลการทดลองที่ 4.3 พบว่าสามารถดึงข้อมูลล่าสุดจาก Firebase ในแต่ละส่วนที่ถูกเก็บไว้ที่ Realtime Database ให้ไปแสดงที่ Web Hosting ได้ โดยที่สามารถอัปเดตข้อมูลล่าสุดได้แบบอัตโนมัติ และระยะเวลาที่อัปเดตข้อมูลที่ Web Hosting ใกล้เคียงกับเวลาที่อัปเดตใน Realtime Database มาก ทำให้สามารถดูข้อมูลปัจจุบันของแต่ละส่วนได้อย่างต่อเนื่อง

### 5.2 วิจารณ์ผลการทดลอง

ในการถ่ายภาพในทุก ๆ 1 วินาที แล้วทำการส่งจะทำให้การส่งล่าช้าเนื่องจากรูปยังมีขนาดใหญ่ก็จะใช้เวลาในการส่งค่อนข้างช้า จึงต้องลดขนาดรูปภาพเพื่อการส่งข้อมูลที่เร็วขึ้น

ในการรับค่าเซนเซอร์ด้วยบลูทูธบางที่จะเกิดความผิดพลาดโดยบางรอบการทำงานอาจจะอ่านข้อมูลเข้ามาไม่ครบ ทำให้ต้องอ่านข้อมูลรอบถัดไปเข้ามาประกอบกันให้สมบูรณ์ หรืออาจจะทิ้งข้อมูลที่เวลานั้นไป

ในการเรียกข้อมูลที่ถูเก็บไว้ที่ Realtime Database ไปแสดงที่ Web Hosting สามารถแสดงได้แค่ข้อมูลล่าสุดที่อัปเดตเข้ามาแล้วเท่านั้น ยังไม่ได้สามารถแสดงข้อมูลที่รับมาก่อนหน้าได้ หากต้องการเรียกดูข้อมูลย้อนหลังต้องเข้ามาดูที่ Realtime Database ทำให้มีขั้นตอนยุ่งยากและใช้เวลาในการเรียกดูข้อมูลมากยิ่งขึ้น

### 5.3 ข้อเสนอแนะ

หากพื้นที่ที่ติดตั้งกล่อง ESP-32CAM อยู่ห่างจากอุปกรณ์ที่ใช้ส่งสัญญาณ Wi-Fi จะทำให้ความเสถียรในการส่งข้อมูลลดน้อยลง อาจต้องมีการติดตั้งเสาในการช่วยส่งสัญญาณเพิ่ม เพื่อให้การส่งข้อมูลมีความเสถียรมากขึ้น

หากสามารถเพิ่มฟังก์ชันการใช้งานที่หน้า Web Hosting ให้สามารถดูข้อมูลที่ถูกเก็บไว้ก่อนหน้ามาแสดงบนหน้า Web Hosting ได้โดยไม่ต้องเข้าไปเรียกดูข้อมูลย้อนหลังที่ Realtime Database จะทำให้มีความสะดวกในการเรียกดูข้อมูลมากยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## อ้างอิง

- [1] Gump. 2562. มาทำความเข้าใจจักชนิดต่าง ๆ ของแผงโซลาร์เซลล์กัน. จาก : <https://www.gump.in.th/article/535>. สืบค้นเมื่อ 25 พฤศจิกายน 2565
- [2] Randomnerdtutorials. 2563. ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained. จาก : <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>. สืบค้นเมื่อ 21 ตุลาคม 2565
- [3] Mindphp. 2560. วิธีการเข้ารหัส และถอดรหัส BASE64. จาก : <https://www.mindphp.com/developer/80-php-security/4082-encoding-decoding-base64.html>. สืบค้นเมื่อ 20 พฤศจิกายน 2565
- [4] Thanop. 2565. แบตเตอรี่ลิเธียมไอออน (Lithium-Ion Battery). จาก : <https://www.thanop.com/tag/lithium-ion-battery/page/4/>. สืบค้นเมื่อ 25 พฤศจิกายน 2565
- [5] Wikipedia. 2565. Base64. จาก : <https://en.wikipedia.org/wiki/Base64>. สืบค้นเมื่อ 20 พฤศจิกายน 2565
- [6] Ian Buckley. 2560. 6 Easy Ways to Connect Arduino to Android. จาก : <https://www.makeuseof.com/tag/6-easy-ways-connect-arduino-android/>. สืบค้นเมื่อ 17 ตุลาคม 2565
- [7] Jacob Beningo. 2557. 10 Steps to Selecting a Microcontroller. จาก : <https://community.arm.com/arm-community-blogs/b/embedded-blog/posts/10-steps-to-selecting-a-microcontroller>. สืบค้นเมื่อ 23 พฤศจิกายน 2565
- [8] Roboindia. 2565. Bluetooth Communication using MIT App Inventor. จาก : <https://roboindia.com/tutorials/bluetooth-communication-mit-app-inventor-2/>. สืบค้นเมื่อ 16 พฤศจิกายน 2565
- [9] Romin Irani. 2559. Tutorial : MIT App Inventor + Firebase. จาก : <https://rominirani.com/tutorial-mit-app-inventor-firebase-4be95051c325>. สืบค้นเมื่อ 15 พฤศจิกายน 2565
- [10] Randomnerdtutorials. 2565. ESP32 with Firebase – Creating a Web App. จาก : <https://randomnerdtutorials.com/esp32-firebase-web-app/>. สืบค้นเมื่อ 11 พฤศจิกายน 2565

- [11] Mehedi Sharif. 2565. Web Application Templates For A Powerful Backend. จาก : <https://themefisher.com/web-application-templates> สืบค้นเมื่อ 11 พฤศจิกายน 2565
- [12] สุวัฒน์ คงดำรงเกียรติ. 2564. ภาวะหัวใจหยุดเต้นเฉียบพลันอันตรายที่ไร้สัญญาณเตือน. จาก : [https://www.phyathai.com/article\\_detail/1572/th/ภาวะหัวใจหยุดเต้นเฉียบพลัน...อันตรายที่ไร้สัญญาณเตือน](https://www.phyathai.com/article_detail/1572/th/ภาวะหัวใจหยุดเต้นเฉียบพลัน...อันตรายที่ไร้สัญญาณเตือน) สืบค้นเมื่อ 21 ตุลาคม 2565
- [13] บริษัท อิมเม้นซ์ คอร์ปอเรชั่น จำกัด. 2562. มาทำความเข้าใจจักชนิดต่างๆ ของแผงโซลาร์เซลล์กัน. จาก : <https://www.gump.in.th/article/535> สืบค้นเมื่อ 14 พฤศจิกายน 2565
- [14] SP Micro Tech. 2565. เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22 / AM2302. จาก : <https://www.spmicrotech.com/product/เซ็นเซอร์วัดอุณหภูมิแล/> สืบค้นเมื่อ 21 ตุลาคม 2565
- [15] เจ้าของร้าน ArtronShop. 2565. แนะนำ ESP32. จาก : <https://www.artronshop.co.th/article/51/esp32เบื้องต้น-บทที่-1-แนะนำ-esp32> สืบค้นเมื่อ 21 ตุลาคม 2564
- [16] คิดสร้างสรรค์. 2565. MIT App Inventor เขียนโปรแกรมบนมือถือ. จาก : <https://kidsangsan.com/2012/07/04/mit-app-inventor-เขียนโปรแกรมบนมือถือ/> สืบค้นเมื่อ 20 ตุลาคม 2565
- [17] บริษัท โฟร์เอ็กซ์ตรีม จำกัด. 2564. Firebase คืออะไร. จาก : <https://www.4xtreme.com/2020/11/20/firebase-คืออะไร/> สืบค้นเมื่อ 28 ตุลาคม 2565
- [18] CyberTice. 2565. GY-9960 RGB เซนเซอร์ตรวจจับสี RGB. จาก : <https://www.cybertice.com/product/1012/gy-9960-rgb-and-gesture-sensor-เซนเซอร์ตรวจจับสี-rgb-และท่าทาง-mcu-9900-apds9900> สืบค้นเมื่อ 21 ตุลาคม 2566
- [19] บริษัท ควิกเซอร์ฟ โพรไวเดอร์ จำกัด. 2566. Cloud database คืออะไร ผู้ที่จะดูแลควรมีทักษะอะไรบ้าง. จาก : <https://www.quickserv.co.th/knowledge-base/solutions/Cloud-database-คืออะไร-ผู้ที่จะดูแลควรมีทักษะอะไรบ้าง/> สืบค้นเมื่อ 21 ตุลาคม 2566

ภาคผนวก ก  
โปรแกรมถ่ายรูปด้วย ESP32-CAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

===== Code การทำงานของ ESP32-CAM ===== Start =====

// Take picture and upload to firebase //

#include "WiFi.h"
#include "esp_camera.h"
#include "Arduino.h"
#include "soc/soc.h" // Disable brownour problems (protect esp32_cam
for reset when voltage drop low than 2.4V because use USB port from
computer)
#include "soc/rtc_cntl_reg.h" // Disable brownour problems
#include "driver/rtc_io.h"
#include <EEPROM.h>
#include <SPIFFS.h>
#include <Firebase_ESP_Client.h> // Connect to Firebase
#include <addons/SDHelper.h> // Provide the SD card interfaces
setting and mountingu
#include "FS.h" // SD Card ESP32
#include "SD_MMC.h" // SD Card ESP32
#include "SPI.h"
#include "SD.h"
#include "time.h"
#include "addons/RTDBHelper.h"
#include "addons/TokenHelper.h"

// Insert Firebase project API Key
#define API_KEY "AIzaSyCgW0CsODlAqLBBAEXBWWt71nt_002iVVI"
#define DATABASE_URL "https://smart-aed-cabinet-default-
rtddb.firebaseio.com/" //Dont use because use firebaestorage
#define USER_EMAIL "smartaedcabinet@gmail.com"
#define USER_PASSWORD "aed114118357"

// Insert Firebase storage bucket ID e.g bucket-name.appspot.com
#define STORAGE_BUCKET_ID "smart-aed-cabinet.appspot.com"

File file;
File file1;

// Boolean check
bool taskCompleted = false;
bool function_check = true;
bool signupOK = false;

//Define Firebase Data objects
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//Constant parameter
const char* ssid = "Forddrof";
const char* password = "12345678";

const int timerInterval = 6000;
unsigned long previousMillis = 0;

int i = 0;
int pictureNumber = 0;
int LED_FLASH = 4;
int read_data;
int count_number_picture = 0;

String path;
String read_data_1;
String S1 = "0";
String S2 = "1";

String StrAB[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10" }; //
String myTime;

// OV2640 camera module pins (CAMERA_MODEL_AI_THINKER)
#define CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

#define FILE_PHOTO "/photoSpiffs.jpg"

bool checkPhoto(fs::FS& fs) {
  File f_pic = fs.open(FILE_PHOTO);
  unsigned int pic_sz = f_pic.size();
  return (pic_sz > 100);
}

//=====
===== WiFi Connect
void initWiFi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//=====
===== SD CARD
void init_SD_card() {
  Serial.println("Starting SD Card");
  delay(1);
  if (!SD_MMC.begin()) {
    Serial.println("SD fail to begin");
    ESP.restart();
  }
  uint8_t cardType = SD_MMC.cardType();
  if (cardType == CARD_NONE) {
    Serial.println("No SD Card attached");
    return;
  }
}

//=====
===== take picture
void take_send_sd_to_fb_present() {
  delay(1);
  bool ok = 0;
  camera_fb_t* fb = NULL;
  do {
    fb = esp_camera_fb_get();

    File file1 = SPIFFS.open(FILE_PHOTO, FILE_WRITE);

    if (!file1) {
      Serial.println("Failed to open file in writing mode");
    } else {
      file1.write(fb->buf, fb->len); // payload (image), payload length
      Serial.printf("Saved file to path: %s\n", FILE_PHOTO);
      Serial.print("Saved file ");
    }
    file1.close();
    esp_camera_fb_return(fb);
    ok = checkPhoto(SPIFFS);
  } while (!ok);

  String path_name_storage_past = "/" + printLocalTime_tag() + "/present" +
  "/" + printLocalTime() + ".jpg";

  if (Firebase.Storage.upload(&fbdo, STORAGE_BUCKET_ID, FILE_PHOTO,
mem_storage_type_flash, path_name_storage_past, "image/jpeg")) {
    Serial.printf("\nDownload URL");
    Serial.printf("\nDownload URL: %s\n", fbdo.downloadURL().c_str());
  } else {
    Serial.println("Upload fail");
    Serial.println(fbdo.errorReason());
  }
}

void send_sd_to_fb_past(fs::FS& fs, const char* dirname, uint8_t levels) {
  Serial.printf("Listing directory: %s\n", dirname);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

File root = fs.open(dirname);
if (!root) {
  Serial.println("Failed to open directory");
  return;
}
if (!root.isDirectory()) {
  Serial.println("Not a directory");
  return;
}

File file = root.openNextFile();
while (file && (count_number_picture <= 10)) {
  if (file.isDirectory()) {
    Serial.print("  DIR : ");
    Serial.println(file.name());
    if (levels) {
      send_sd_to_fb_past(fs, file.name(), levels - 1);
    }
  } else {
    Serial.print("  FILE: ");
    Serial.println(file.name());

    String path_name = file.name();
    Serial.println(path_name);

    String path_name_storage = "/" + printLocalTime_tag() + "/past/" +
StrAB[i] + ".jpg";
    i++;
    count_number_picture++;

    delay(1);

    if (Firebase.Storage.upload(&fbdo, STORAGE_BUCKET_ID, path_name,
mem_storage_type_sd, path_name_storage, "image/jpeg")) {
      Serial.printf("\nDownload URL");
      Serial.printf("\nDownload URL: %s\n", fbdo.downloadURL().c_str());
    } else {
      Serial.println("Upload fail");
      Serial.println(fbdo.errorReason());
    }
    delay(100);
    take_send_sd_to_fb_present();
  }
  file = root.openNextFile();
}
count_number_picture = 0;
}

//=====
===== SPIFFS
void initSPIFFS() {
  if (!SPIFFS.begin(true)) {
    Serial.println("An Error has occurred while mounting SPIFFS");
    ESP.restart();
  } else {
    delay(500);
    Serial.println("SPIFFS mounted successfully");
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

//===== CAMERA
void initCamera() {
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;

    if (psramFound()) {
        config.frame_size = FRAMESIZE_UXGA;
        config.jpeg_quality = 5;
        config.fb_count = 1;
    } else {
        config.frame_size = FRAMESIZE_SVGA;
        config.jpeg_quality = 10;
        config.fb_count = 1;
    }
    // Camera init
    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {
        Serial.printf("Camera init failed with error 0x%x", err);
        ESP.restart();
    }
    sensor_t* s = esp_camera_sensor_get();
    s->set_framesize(s, FRAMESIZE_CIF);
}

void take_photo() {

    delay(1);
    bool ok = 0;
    camera_fb_t* fb = NULL;

    do {
        fb = esp_camera_fb_get();

        Serial.println("Take Photo");

        if (!fb) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return;
}

pictureNumber = pictureNumber + 1;

String path = "/picture" + String(pictureNumber) + ".jpg";

File file = SD.open(path.c_str(), FILE_WRITE);

if (!file) {
    Serial.println("Failed to open file in writing mode");
} else {
    file.write(fb->buf, fb->len);
    Serial.printf("Saved file to path: %s\n", path.c_str());
    Serial.print("Saved file ");
}
file.close();

File file1 = SPIFFS.open(FILE_PHOTO, FILE_WRITE);
if (!file1) {
    Serial.println("Failed to open file in writing mode");
} else {
    file1.write(fb->buf, fb->len);
    Serial.printf("Saved file to path: %s\n", FILE_PHOTO);
    Serial.print("Saved file ");
}
file1.close();
esp_camera_fb_return(fb);
ok = checkPhoto(SPIFFS);
} while (!ok);

String path_name_storage_current = "/current";

if (Firebase.Storage.upload(&fbdo, STORAGE_BUCKET_ID, FILE_PHOTO,
mem_storage_type_flash, path_name_storage_current, "image/jpeg")) {
    Serial.printf("\nDownload URL");
    Serial.printf("\nDownload URL: %s\n", fbdo.downloadURL().c_str());
} else {
    Serial.println("Upload fail");
    Serial.println(fbdo.errorReason());
}
}

//=====
===== Timezone
void setTimezone(String timezone) {
    Serial.printf(" Setting Timezone to %s\n", timezone.c_str());
    setenv("TZ", timezone.c_str(), 1);
    tzset();
}

void initTime(String timezone) {
    struct tm timeinfo;

    Serial.println("Setting up time");
    configTime(0, 0, "pool.ntp.org");
    if (!getLocalTime(&timeinfo)) {
        Serial.println(" Failed to obtain time");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return;
}
Serial.println(" Got the time from NTP");
// Now we can set the real timezone
setTimezone(timezone);
}

String printLocalTime() {
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) {
        Serial.println("Failed to obtain time 1");
        return "";
    }
    char timeStringBuffer[60]; //จองพื้นที่เก็บ array
    strftime(timeStringBuffer, sizeof(timeStringBuffer), "%Y-%m-%d_%H-%M-%S",
&timeinfo);
    String Time_X = String(timeStringBuffer);
    return Time_X;
}

String printLocalTime_tag() {
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) {
        Serial.println("Failed to obtain time 1");
        return "";
    }
    char timeStringBuffer_tag[60]; //จองพื้นที่เก็บ array
    strftime(timeStringBuffer_tag, sizeof(timeStringBuffer_tag), "%F",
&timeinfo);
    String Time_tag = String(timeStringBuffer_tag);
    return Time_tag;
}

void setTime(int yr, int month, int mday, int hr, int minute, int sec, int
isDst) {
    struct tm tm;

    tm.tm_year = yr - 1900; // Set date
    tm.tm_mon = month - 1;
    tm.tm_mday = mday;
    tm.tm_hour = hr; // Set time
    tm.tm_min = minute;
    tm.tm_sec = sec;
    tm.tm_isdst = isDst; // 1 or 0
    time_t t = mktime(&tm);
    Serial.printf("Setting time: %s", asctime(&tm));
    struct timeval now = { .tv_sec = t };
    settimeofday(&now, NULL);
}

void check_function_take_photo() {
    Serial.println("check_function_take_photo");
    if (Firebase.RTDB.getString(&fbdo, "/bluetoothv2_2debug/03_need_switch"))
    {
        if (fbdo.dataTypeEnum() == fb_esp_rtdb_data_type_string) {
            read_data_1 = fbdo.to<String>();
            // int read_data = read_data_1.toInt();
            Serial.println(read_data_1);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (read_data_1 == S1) {

        function_check = true;

        Serial.println("XXXXXXXXXXXXX");
        Serial.println("XXXXXXXXXXXXX");
    }
}
}

void Task1code(void* parameter) {
    for (;;) {
    }
}

//=====
===== void setup
void setup() {

    Serial.begin(115200);

    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

    SD_Card_Mounting();
    initSPIFFS();

    Serial.println(SD.totalBytes());

    initWiFi();

    initCamera();
    initTime("<+07>-7");

    Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);

    // FIREBASE STORAGE
    config.api_key = API_KEY;
    config.database_url = DATABASE_URL;
    auth.user.email = USER_EMAIL;
    auth.user.password = USER_PASSWORD;
    config.fcs.upload_buffer_size = 512;

    if (Firebase.signUp(&config, &auth, "", "")) {
        Serial.println("ok");
        signupOK = true;
    } else {
        Serial.printf("%s\n", config.signer.signupError.message.c_str());
    }

    //Assign the callback function for the long running token generation task
    config.token_status_callback = tokenStatusCallback;
    Firebase.begin(&config, &auth);
    Firebase.reconnectWiFi(true);
}

//=====
===== void loop

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void loop() {

    if (Firebase.ready() && signupOK) {

        if (Firebase.RTDB.getString(&fbdo,
            "/bluetoothv2_2debug/03_reed_switch")) {

            if (fbdo.dataTypeEnum() == fb_esp_rtdb_data_type_string) {

                String read_data_1 = fbdo.to<String>();

                read_data_1.trim();

                unsigned long currentMillis = millis();

                if ((read_data_1 == S1) && (currentMillis - previousMillis >=
                    timerInterval) && function_check) {

                    previousMillis = currentMillis;
                    Serial.println(StrAB[i]);
                    myTime = printLocalTime();
                    StrAB[i] = String(myTime);
                    take_photo();
                    Serial.println(StrAB[i]);
                    i++;

                    if (i >= 10) {
                        i = 0;
                    }

                    if (pictureNumber >= 10) {
                        pictureNumber = 0;
                    }
                }

                if ((read_data_1 == S2) && function_check) { //cabinet open

                    send_sd_to_fb_past(SD, "/", 0);
                    function_check = false;
                    Serial.println("false");

                    while (!function_check) {
                        take_send_sd_to_fb_present();
                        Serial.println("true");
                        check_function_take_photo();
                    }
                }
            }
        }
    }
}

===== Code การทำงานของ ESP32-CAM ===== End =====

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข  
โปรแกรมการทำงานของ ESP32



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

===== Code การทำงานของ ESP32 ===== Start =====
#include "BluetoothSerial.h"
BluetoothSerial ESP_BT;
char text[70];
unsigned long nowTime = 0;

#include "DHT.h"
#define DHTPIN 19 // DHT22 pin dht22
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
float temperature = 0;
float humidity = 0;

#include <Wire.h>
#include <SparkFun_APDS9960.h>
SparkFun_APDS9960 apds = SparkFun_APDS9960();
uint16_t ambientLight = 0;
uint16_t redLight = 0;
uint16_t greenLight = 0;
uint16_t blueLight = 0;
#define APDS9960_INT 23 // APDS9960 interrupt pin
#define LIGHT_INT_HIGH 350 // High light level for interrupt
#define LIGHT_INT_LOW 1 // Low light level for interrupt
int isr_flag = 0;
uint16_t threshold = 0;
uint16_t ambientLightInt = 0;
uint16_t redLightInt = 0;
uint16_t greenLightInt = 0;
uint16_t blueLightInt = 0;
uint8_t proximityData = 0;

#define REEDPIN 4 // Reed switch pin
int reedSwitch = 0;

#define LOCKPIN 18 // Relay control electric bolt pin
#define LOCKSWITCHPIN 14 // Electric bolt micro switch pin
int lockSwitch = 0;

#define VOLTPIN 34
float batteryVoltage = 0.0;

struct Button {
  const uint8_t PIN;
  uint32_t numberKeyPresses;
  bool pressed;
  int flag;
  unsigned long lastButtonTime;
};

Button button = {26, 0, false, 0, 0};
unsigned long buttonTime = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void IRAM_ATTR isrButton() {
  buttonTime = millis();
  if (buttonTime - button.lastButtonTime > 500) {
    button.numberKeyPresses++;
    button.pressed = true;
    button.flag = 1;
    button.lastButtonTime = buttonTime;
  }
}

int isAedStatusBlink = 0;
unsigned long lastAmbientIntTime = 0;
void IRAM_ATTR isrAmbientLight() {
  isr_flag = 1;
  lastAmbientIntTime = millis();
  isAedStatusBlink = 1;
}

void setup() {
  ESP_BT.begin("ESP32_AEDcabinetProject"); // Name of your Bluetooth
  interface -> will show up on your phone
  Serial.begin(115200);

  dht.begin();

  pinMode(APDS9960_INT, INPUT);
  attachInterrupt(APDS9960_INT, isrAmbientLight, FALLING);

  pinMode(REEDPIN, INPUT_PULLUP);

  pinMode(LOCKPIN, OUTPUT);
  digitalWrite(LOCKPIN, HIGH);

  pinMode(button.PIN, INPUT_PULLUP);
  attachInterrupt(button.PIN, isrButton, FALLING);

  // Initialize APDS-9960 (configure I2C and initial values)
  if ( apds.init() ) {
    Serial.println(F("APDS-9960 initialization complete"));
  } else {
    Serial.println(F("Something went wrong during APDS-9960 init!"));
  }

  // Set high and low interrupt thresholds
  if ( !apds.setLightIntLowThreshold(LIGHT_INT_LOW) ) {
    Serial.println(F("Error writing low threshold"));
  }
  if ( !apds.setLightIntHighThreshold(LIGHT_INT_HIGH) ) {
    Serial.println(F("Error writing high threshold"));
  }

  // Start running the APDS-9960 light sensor (no interrupts)
  if ( apds.enableLightSensor(false) ) {
    Serial.println(F("Light sensor is now running"));
  } else {
    Serial.println(F("Something went wrong during light sensor init!"));
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Read high and low interrupt thresholds
if ( !apds.getLightIntLowThreshold(threshold) ) {
  Serial.println(F("Error reading low threshold"));
} else {
  Serial.print(F("Low Threshold: "));Serial.println(threshold);
}
if ( !apds.getLightIntHighThreshold(threshold) ) {
  Serial.println(F("Error reading high threshold"));
} else {
  Serial.print(F("High Threshold: "));
  Serial.println(threshold);
}

// Enable interrupts
if ( !apds.setAmbientLightIntEnable(1) ) {
  Serial.println(F("Error enabling interrupt`s"));
}

// Adjust the Proximity sensor gain
if ( !apds.setProximityGain(PGAIN_2X) ) {
  Serial.println(F("Something went wrong trying to set PGAIN"));
}

// Start running the APDS-9960 proximity sensor (no interrupts)
if ( apds.enableProximitySensor(false) ) {
  Serial.println(F("Proximity sensor is now running"));
} else {
  Serial.println(F("Something went wrong during sensor init!"));
}

// Wait for initialization and calibration to finish
delay(500);
}

void loop() {

  if (ESP_BT.available() > 0) {

    static int bluetoothInput = 0; // input from phone
    bluetoothInput = ESP_BT.read(); // read

    if (bluetoothInput == '1') {

      temperature = dht.readTemperature();
      humidity = dht.readHumidity();

      reedSwitch = digitalRead(REEDPIN); // 0 = closed door, 1 = open door
      lockSwitch = digitalRead(LOCKSWITCHPIN); // 0 = locked door, 1 =
      unlocked door

      batteryVoltage = analogRead(VOLTYPIN);

      if (!apds.readAmbientLight(ambientLight) ||
!apds.readRedLight(redLight) || !apds.readGreenLight(greenLight) ||
!apds.readBlueLight(blueLight)) {
        Serial.println("Error reading light values");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if (!apds.readProximity(proximityData)) {
        Serial.println("Error reading proximity value");
    }

    sprintf(text, "%.2f %.2f %d %d %d %d %d %d %d %.1f %d %d %d %d %d",
temperature, humidity, reedSwitch, lockSwitch,
        ambientLight, redLight, greenLight, blueLight, proximityData,
batteryVoltage, button.flag,
        ambientLightInt, redLightInt, greenLightInt, blueLightInt,
isAedStatusBlink);
        Serial.println(text);
        ESP_BT.print(text);
        button.flag = 0;
    }
}

if (button.pressed) {
    Serial.printf("Button A has been pressed %u times\n",
button.numberKeyPresses);
    digitalWrite(LOCKPIN, LOW);
    button.pressed = false;
}

nowTime = millis();
if (nowTime - button.lastButtonTime > 1000) {
    digitalWrite(LOCKPIN, HIGH);
}
if (nowTime - lastAmbientIntTime > 5000) {
    isAedStatusBlink = 0;
}

if (isr_flag == 1) {
    if (!apds.readAmbientLight(ambientLightInt) ||
!apds.readRedLight(redLightInt) || !apds.readGreenLight(greenLightInt) ||
!apds.readBlueLight(blueLightInt)) {
        Serial.println("Error reading light values");
    } else {
        Serial.println("Ambient interrupted!");
    }
    isr_flag = 0;
    if (!apds.clearAmbientLightInt()) {
        Serial.println("Error clearing interrupt");
    }
}
}

===== Code การทำงานของ ESP32 ===== End =====

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค  
โปรแกรมการทำงานของโทรศัพท์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

===== Code การทำงานของแอปโทรศัพท์ ===== Start =====

```

when connectButton .Click
do
  evaluate but ignore result call BluetoothClient1 .Connect
  address TextBox1 .Hint
  call updateBluetoothStatus

when disconnectButton .Click
do
  call BluetoothClient1 .Disconnect
  call updateBluetoothStatus

to updateBluetoothStatus
do
  if BluetoothClient1 .IsConnected
  then set Label4 .Text to "Connected"
  else set Label4 .Text to "Disconnected"

initialize global receivedData to ""
initialize global listReceivedData to create empty list
initialize global temperature to 0
initialize global humidity to 0
initialize global reedSwitch to 0
initialize global lockSwitch to 0
initialize global ambient to 0
initialize global ambientInt to 0
initialize global red to 0
initialize global redInt to 0
initialize global green to 0
initialize global greenInt to 0
initialize global blue to 0
initialize global blueInt to 0
initialize global proximity to 0
initialize global redDiff to 0
initialize global batteryVoltage to 0
initialize global greenDiff to 0
initialize global buttonOneFlag to 0
initialize global aedStatus to "None"
initialize global buttonTwoFlag to 0
initialize global isAedStatusBlink to 0

initialize global nowTime to 0
initialize global tagTime to ""

```

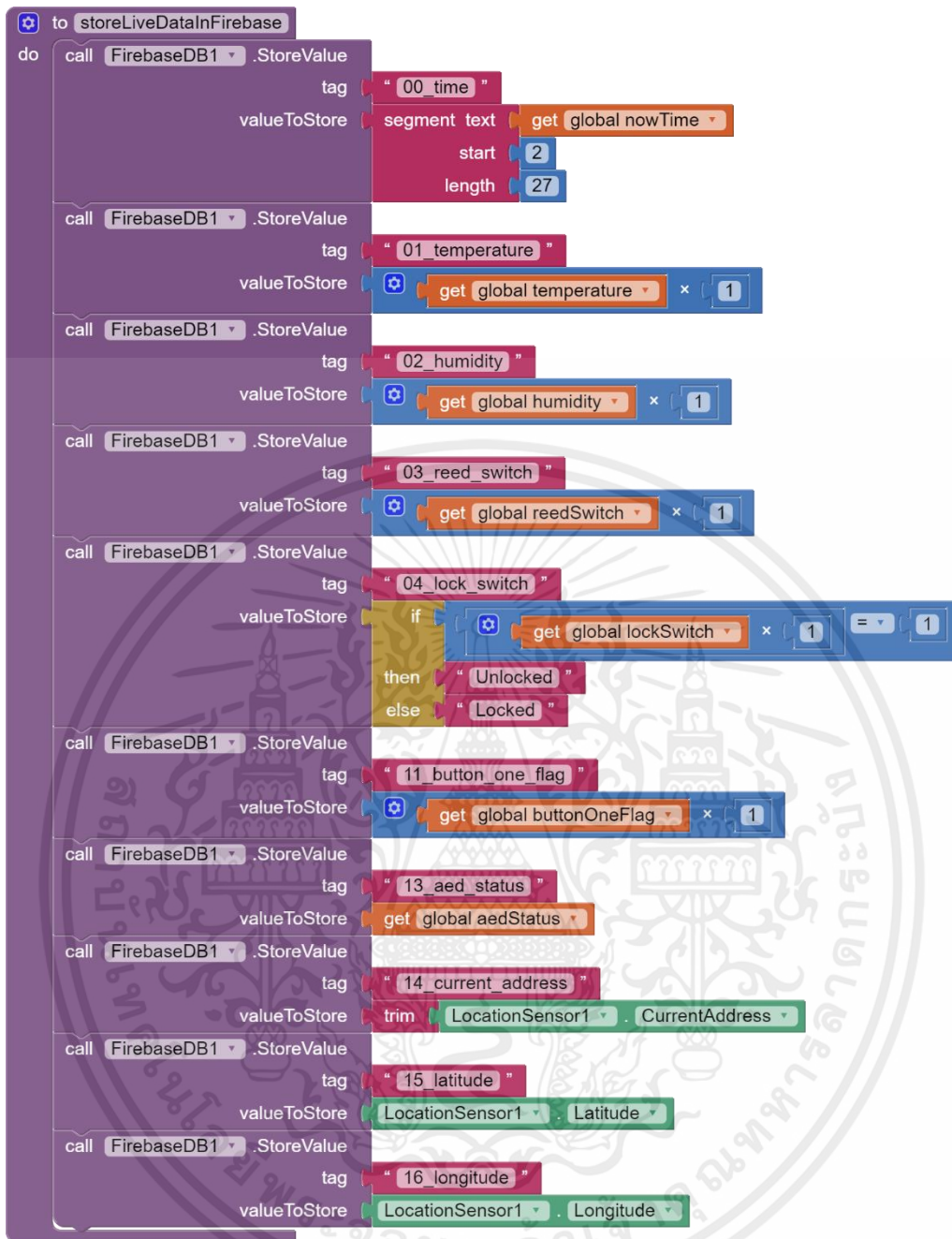
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

to divideData
do
  set global temperature to select list item list get global listReceivedData
  index 1
  set global humidity to select list item list get global listReceivedData
  index 2
  set global reedSwitch to select list item list get global listReceivedData
  index 3
  set global lockSwitch to select list item list get global listReceivedData
  index 4
  set global ambient to select list item list get global listReceivedData
  index 5
  set global red to select list item list get global listReceivedData
  index 6
  set global green to select list item list get global listReceivedData
  index 7
  set global blue to select list item list get global listReceivedData
  index 8
  set global proximity to select list item list get global listReceivedData
  index 9
  set global batteryVoltage to select list item list get global listReceivedData
  index 10
  set global buttonOneFlag to select list item list get global listReceivedData
  index 11
  set global ambientInt to select list item list get global listReceivedData
  index 12
  set global redInt to select list item list get global listReceivedData
  index 13
  set global greenInt to select list item list get global listReceivedData
  index 14
  set global blueInt to select list item list get global listReceivedData
  index 15
  set global isAedStatusBlink to select list item list get global listReceivedData
  index 16
  if get global isAedStatusBlink = "1"
  then
    set global redDiff to get global redInt - get global red
    set global greenDiff to get global greenInt - get global green
    if get global greenDiff > get global redDiff
    then
      set global aedStatus to "Flashing green"
    else
      set global aedStatus to "Flashing red"
  else
    set global aedStatus to "No flashing"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

to createTagTime
do
  set global tagTime to
  join
    segment text get global nowTime
    start 2
    length 4
    segment text get global nowTime
    start 7
    length 2
    segment text get global nowTime
    start 10
    length 2
    " "
    segment text get global nowTime
    start 21
    length 2
    segment text get global nowTime
    start 24
    length 2
    segment text get global nowTime
    start 27
    length 2

when isAutoSwitch .Changed
do
  if isAutoSwitch . On
  then
    set periodClock . TimerEnabled to true
    set periodClock . TimerInterval to 2000
  else
    set periodClock . TimerEnabled to false
    set periodClock . TimerInterval to 2000

when periodClock .Timer
do
  call BluetoothClient1 .SendText
  text "1"
  set global nowTime to call Clock1 .FormatDateTime
  instant call Clock1 .Now
  pattern "yyyy.MM.dd G 'at' HH:mm:ss z"
  set waitClock . TimerEnabled to true

when waitClock .Timer
do
  set waitClock . TimerEnabled to false
  set waitClock . TimerInterval to 200
  set global receivedData to call BluetoothClient1 .ReceiveText
  numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
  set global listReceivedData to split at spaces get global receivedData
  call createTagTime
  call divideData
  call storeLiveDataInFirebase

```

===== Code การทำงานของแอปโทรศัพท์ ===== End =====

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้