

ระบบนับถาดอัตโนมัติ

Automatic Counter Tray System

บุญยง ตันปี

Punyanud Tanbee

รินรดา อังสถาพร

Rinrada Ungsathaporn

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบนับถาดอัตโนมัติ

Automatic Counter Tray System

โดย

บุญยง ชันบี

รินรดา อังสถาพร

อาจารย์ที่ปรึกษา

ผศ.ดร. แสงระวี บัวแก้ว

ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2566

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

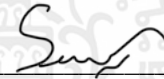
เรื่อง ระบบนับถาดอัตโนมัติ

Automatic Counter Tray System

ผู้จัดทำ นางสาวปุกฤษฏ์ ตันปี รหัสประจำตัว 63010613

นางสาวรินรดา อังสถาพร รหัสประจำตัว 63010823

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(ผศ.ดร. แสงระวี บัวแก้ว)

อาจารย์ที่ปรึกษา

หัวข้อโครงการ	ระบบนับถาดอัตโนมัติ
นักศึกษา	นางสาวปุกฤษณัฐ ต้นปี รหัสประจำตัว 63010613 นางสาวรินรดา อังสถาพร รหัสประจำตัว 63010823
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2566
อาจารย์ที่ปรึกษาโครงการ	ผศ.ดร.แสงระวี บัวแก้ว

บทคัดย่อ

โครงการนี้เป็นการศึกษาและพัฒนาเครื่องนับถาด โดยใช้ Ultrasonic Sensor ในการนับจำนวนถาดที่ซ้อนกัน ผ่านระบบสายพานจำลอง และแสดงผลบนจอ LCD รวมไปถึงศึกษาหลักการทำงานของ RFID โดยใช้ RFID Reader MFRC-522 RC เพื่ออ่านข้อมูลของวัตถุภายในถาดจาก RFID tag แบบ passive tag ซึ่งอุปกรณ์นี้ทำงานที่ความถี่ 13.56 MHz ข้อมูลที่ได้จากการอ่านจะถูกส่งไปยังโปรแกรมที่พัฒนาจาก Firebase เพื่อจัดเก็บและติดตามในโหมด real-time ซึ่งวิธีการนี้ทำให้การนับถาดมีความแม่นยำมากขึ้น และเพิ่มประสิทธิภาพในการจัดการข้อมูล

Project Title	Automatic Counter Tray System	
Student	Miss Punyanud Tanbee	Student ID 63010613
	Miss Rinrada Ungsathaporn	Student ID 63010823
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2023	
Project Advisor	Asst. Prof. Dr. Seangrawee Buakaew	

ABSTRACT

This project is a study and development of a tray counting machine using an ultrasonic sensor to count the number of nested trays. through a simulated conveyor system and display on the LCD screen, including studying the working principles of RFID by using the RFID Reader MFRC-522 RC to read information about objects inside the tray from passive RFID tags. This device works at a frequency of 13.56 MHz. The information obtained from reading will be sent to a program developed in Firebase for storage and tracking in real-time. This method makes tray counting more accurate. and increase efficiency in data management.

กิตติกรรมประกาศ

โครงการระบบนับถาดอัตโนมัติสำเร็จลุล่วงไปได้ด้วยดีโดยได้รับความอนุเคราะห์จากผู้มีพระคุณหลายท่าน โดยเฉพาะอาจารย์ แสงระวี บัวแก้ว ที่คอยช่วยเหลือ แนะนำ และให้คำปรึกษาเกี่ยวกับการส่งข้อมูลไปยัง dashboard, การออกแบบวงจรอิเล็กทรอนิกส์ รวมถึงการออกแบบหน้าเว็บไซต์ให้มีความเหมาะสม ขอขอบคุณรุ่นพี่ที่คอยช่วยชี้แนวทางต่างๆ และช่วยแก้ปัญหาที่เกิดขึ้นระหว่างการทำโครงการจนออกมาสำเร็จ ขอขอบคุณเพื่อนๆ ที่คอยช่วยเหลือในด้านอุปกรณ์ที่ใช้ในการทำโครงการ, การแสดงความคิดเห็นของการออกแบบหน้าเว็บไซต์ รวมถึงอยู่ร่วมทำงานด้วยกันตลอดระยะเวลาการทำโครงการครั้งนี้

และสุดท้ายผู้จัดทำขอขอบพระคุณ ผู้มีพระคุณทุกท่านที่ไม่ได้ออกนามไว้ ณ ที่นี้ด้วย

บุญยง ตันปี

รินดา อังสภาพร

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	i
บทคัดย่อภาษาอังกฤษ	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญภาพ	viii
สารบัญตาราง	xi
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 สมมติฐานของการศึกษา	2
1.4 ขอบเขตในการจัดทำโครงการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 ระยะเวลาในการทำโครงการ	3
บทที่ 2 ทฤษฎีและเอกสารที่เกี่ยวข้อง	
2.1 ไมโครคอนโทรลเลอร์	4
2.1.1 โครงสร้างทั่วไป	4
2.1.2 การสื่อสารระหว่างอุปกรณ์	5
2.1.3 ESP8266	6
2.2 วงจรที่ใช้ในการทำโครงการ	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.2.1 วงจรรักษาแรงดัน	7
2.2.2 วงจรขับมอเตอร์	9
2.3 อุปกรณ์ที่เกี่ยวข้อง	14
2.3.1 เซนเซอร์ตรวจจับวัตถุด้วยแสงอินฟราเรด	14
2.3.2 เซนเซอร์ชนิดอัลตราโซนิก	16
2.3.3 มอเตอร์ Stepper รุ่น NEMA 17 Stepper Motor	18
2.4 ระบบ RFID	20
2.4.1 คลื่นวิทยุที่ใช้ในระบบ RFID	21
2.4.2 RFID Tag	21
2.4.3 การเลือกใช้งาน RFID Tag	23
2.4.4 RFID Reader	24
2.5 ฐานข้อมูลแบบคลาวด์ (Cloud Database)	24
2.6 ระบบฐานข้อมูล (Database System)	25
2.7 Firebase	25
2.7.1 Cloud Firestore	25
2.7.2 Authentication	26
2.7.3 Hosting	26
2.7.4 Realtime Database – Realtime Database	26

สารบัญ (ต่อ)

	หน้า
บทที่ 3 หลักการทำงานและการออกแบบ	
3.1 ลักษณะการทำงาน	27
3.2 การออกแบบทางด้าน Hardware	28
3.2.1 การเลือกใช้อุปกรณ์ในการดำเนินสร้างชิ้นงาน	28
3.2.1.1 การคำนวณพิกัดหม้อแปลง	28
3.2.1.2 วงจรซีพพลายที่มีรักษาแรงดัน (Voltage regulator)	29
3.3 การออกแบบทางด้าน Software	31
3.4 การเขียนโปรแกรมควบคุมวงจรและแสดงผลบนเว็บไซต์	32
3.4.1 Flowchart ระบบตรวจนับจำนวนถาด	32
3.4.2 Flowchart ขั้นตอนการทำงานของ NodeMCU	33
3.4.3 Flowchart ระบบตรวจเก็บข้อมูลของถาด	34
3.4.4 Flowchart การส่งข้อมูลจาก ESP8266 ไปยัง Firebase	36
3.4.5 Flowchart การส่งข้อมูลของ Firebase ไปยัง Website	37
3.4.6 Flowchart การทำงานของ Webapp	37
3.5 ขั้นตอนการตั้งค่า	38
3.5.1 การตั้งค่า Firebase Realtime database	38
3.5.2 การตั้งค่า Firebase Storage	38
3.5.3 การตั้งค่า Firebase Hosting	38

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	
4.1 ทดสอบความถูกต้องของ Ultrasonic Sensor ขณะมีวัตถุ	42
4.1.1 วางถาดบนสายพานจำนวน 1 ถาด	42
4.1.2 วางถาดบนสายพานจำนวน 2 ถาด	43
4.1.3 วางถาดบนสายพานจำนวน 3 ถาด	44
4.1.4 วางถาดบนสายพานจำนวน 4 ถาด	45
4.1.5 วางถาดบนสายพานจำนวน 5 ถาด	46
4.2 ทดสอบการแสดงผลของ Dashboard	47
4.3 ทดสอบเวลาที่ใช้ในการอ่านค่าจาก ESP8266	48
4.4 ทดสอบเวลาที่ใช้ในการอ่านค่าจาก real-time firebase database ไปยัง Dashboard	48
4.5 สามารถแสดงผลบน Google sheet ได้	49
4.6 สามารถแสดงผลบน Dashboard ได้	50
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	
5.1 สรุปผลการทดลอง	53
5.2 ข้อเสนอแนะ	54
บรรณานุกรม	55
ภาคผนวก	57

สารบัญภาพ

ภาพที่	หน้า
ภาพที่ 2.1 รูปบอร์ด NodeMCU V3 และขาต่างๆ	7
ภาพที่ 2.2 (ก) วงจรรักษาแรงดัน (Voltage regulator)	8
ภาพที่ 2.2 (ข) วงจรพาวเวอร์ซัพพลายที่มีการควบคุม	8
ภาพที่ 2.3 (ก) การต่อวงจรขับเคลื่อนมอเตอร์กับมอเตอร์ เข้ากับบอร์ดไมโครคอนโทรลเลอร์	10
ภาพที่ 2.3 (ข) โมดูล A4988 ควบคุม Stepper Motor	10
ภาพที่ 2.4 แสดงขา Pins ของโมดูล A4988	11
ภาพที่ 2.5 แสดงDatasheet ของมอเตอร์	13
ภาพที่ 2.6 (ก) เซ็นเซอร์แสงสำหรับตรวจจับวัตถุ	15
ภาพที่ 2.6 (ข) แสดงเงื่อนไขการทำงานของเซนเซอร์ตรวจจับวัตถุ	15
ภาพที่ 2.7 การทำงานของคลื่นอัลตราโซนิก	16
ภาพที่ 2.8 โครงสร้าง Stepper Motor	19
ภาพที่ 2.9 โหมดการทำงานของ Nema 17 Stepper Motor	20
ภาพที่ 2.10 แท็กแบบ Active	22
ภาพที่ 2.11 แท็กแบบ Passive	23
ภาพที่ 3.2.1 วงจรซัพพลายที่มีรักษาแรงดัน	29
ภาพที่ 3.2.2 (ก) Block Diagram	29
ภาพที่ 3.2.2 (ข) schematic diagram	30

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 3.2.2 (ค) PCB ที่ออกแบบได้	30
ภาพที่ 3.3.1 (ก) รูปแบบการทำงาน Software	31
ภาพที่ 3.3.1 (ข) รูปแบบการทำงานของ Software ในส่วนของระบบ RFID	31
ภาพที่ 3.4.1 Flowchart แสดงขั้นตอนการทำงานของระบบตรวจนับจำนวนภาค	32
ภาพที่ 3.4.2 Flowchart แสดงขั้นตอนการทำงานของ NodeMCU	34
ภาพที่ 3.4.3 Flowchart แสดงขั้นตอนการทำงานของระบบโดยรวม	35
ภาพที่ 3.4.4 Flowchart การส่งข้อมูลจากการอ่าน RFID tag ไปยัง firebase	36
ภาพที่ 3.4.5 Flowchart แสดงขั้นตอนการส่งข้อมูลของ firebase ไปยังเว็บไซต์	37
ภาพที่ 3.4.6 Flowchart แสดงขั้นตอนการทำงานและการแสดงผลของ Webapp	37
ภาพที่ 3.5.1 การตั้งค่าหน้า user	38
ภาพที่ 3.5.2 การตั้งค่าหน้า Rules	38
ภาพที่ 3.6 คำสั่งการติดตั้ง firebase tools	39
ภาพที่ 3.7 การติดตั้ง firebase tools	39
ภาพที่ 3.8 login firebase	39
ภาพที่ 3.9 เริ่มต้น firebase	40
ภาพที่ 3.10 ตัวเลือกฟีดเจอร์ต่างๆ	40
ภาพที่ 3.11 เลือกโปรเจคที่เราต้องการใช้	40

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 3.12 Hosting Setup	41
ภาพที่ 3.13 SDK setup and configuration	41
ภาพที่ 4.2.1 การแสดงผลของ Dashboard ผ่าน Laptop	47
ภาพที่ 4.2.2 การแสดงผลของ Dashboard ผ่าน Smart Phone	47
ภาพที่ 4.5.1 การแสดงผลบน Google sheet เมื่อมี 5 ข้อมูล	49
ภาพที่ 4.5.2 การแสดงผลบน Google sheet เมื่อมี 10 ข้อมูล	49
ภาพที่ 4.5.3 การแสดงผลบน Google sheet เมื่อมี 15 ข้อมูล	49
ภาพที่ 4.5.4 การแสดงผลบน Google sheet เมื่อมี 20 ข้อมูล	50
ภาพที่ 4.5.5 การแสดงผลบน Google sheet เมื่อมี 25 ข้อมูล	50
ภาพที่ 4.6.1 การแสดงผลบน Dashboard เมื่อมี 5 ข้อมูล	51
ภาพที่ 4.6.2 การแสดงผลบน Dashboard เมื่อมี 10 ข้อมูล	51
ภาพที่ 4.6.3 การแสดงผลบน Dashboard เมื่อมี 15 ข้อมูล	51
ภาพที่ 4.6.4 การแสดงผลบน Dashboard เมื่อมี 20 ข้อมูล	52
ภาพที่ 4.6.5 การแสดงผลบน Dashboard เมื่อมี 25 ข้อมูล	52

สารบัญตาราง

ตารางที่	หน้า
4.1.1 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 1 ถาด	42
4.1.2 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 2 ถาด	43
4.1.3 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 3 ถาด	44
4.1.4 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 4 ถาด	45
4.1.5 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 5 ถาด	46
4.3 แสดงเวลาที่ใช้ในการอ่านค่าจาก ESP8266 ไปยัง real-time firebase database	48
4.4 แสดงเวลาที่ใช้ในการอ่านค่าจาก real-time firebase database ไปยัง Dashboard	48

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

จากโครงการในเทอมแรกซึ่งได้แรงบันดาลใจจากบริษัท Sony device เนื่องจากเป็น senior project ของทางบริษัท ที่ต้องการความร่วมมือกับสถาบันต่างๆ โดยผ่านระบบสายพานจำลอง ได้มีการออกแบบระบบสายพานจำลองโดยใช้ ultrasonic sensor สำหรับการนับจำนวนถาดบนสายพาน และแสดงผลบนจอ LCD ยังมีข้อจำกัดในการอ่านข้อมูลภายในถาด

ดังนั้นกลุ่มของพวกเราจึงได้มีการพัฒนาการนับจำนวนถาดโดยใช้ RFID tag และอ่านข้อมูลผ่าน RFID Reader เพื่อให้สามารถแสดงผลข้อมูลภายในถาดเพิ่มเติม จากนั้นนำข้อมูลที่ได้เก็บลงใน Firebase แบบ Realtime และใช้เวลาในการแสดงผลไม่เกิน 10 วินาทีต่อถาด

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 นำความรู้ทางด้านอิเล็กทรอนิกส์มาใช้ในการปฏิบัติงานจริง
- 1.2.2 เพื่อศึกษาหลักการและวิธีการออกแบบเครื่องนับถาดอัตโนมัติ
- 1.2.3 สามารถเลือกอุปกรณ์ที่เหมาะสมกับการใช้งาน
- 1.2.4 ฝึกการคิดวิเคราะห์และการแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างการปฏิบัติงาน
- 1.2.5 เพื่อศึกษาหลักการทำงานของ RFID และการเก็บข้อมูลลงใน Firebase

1.3 สมมติฐานของการศึกษา

สามารถออกแบบเครื่องนับถาดอัตโนมัติ โดยใช้ Sensor ตรวจจับระยะทางในการตรวจสอบความถูกต้อง เพื่อนับจำนวนถาดที่ถูกซ้อนกัน โดยมีสายพานในการส่งถาดออกมา ซึ่งจำนวนถาดที่นับได้มีความผิดพลาดไม่เกินร้อยละ 10 และสามารถเก็บข้อมูลของวัตถุภายในถาดลงใน Firebase โดยการอ่านข้อมูลจาก RFID Reader แบบ Realtime ได้อย่างถูกต้อง และใช้เวลาในการแสดงผลต่อถาดไม่เกิน 10 วินาที

1.4 ขอบเขตในการจัดทำโครงการ

- 1.4.1 สามารถนับจำนวนถาดที่ส่งมาจากสายพานได้
- 1.4.2 จำนวนถาดที่นับได้มีความผิดพลาดไม่เกินร้อยละ 10
- 1.4.3 สามารถอ่านข้อมูลจากถาดโดยใช้ RFID Reader
- 1.4.4 สามารถเก็บข้อมูลจาก RFID Reader ลงใน Firebase แบบ Realtime ได้อย่างถูกต้อง
- 1.4.5 การแสดงผลแบบ Realtime ไม่เกิน 10 วินาทีต่อถาด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 สามารถออกแบบเครื่องนับถาดอัตโนมัติโดยใช้ Sensor ตรวจจับระยะทาง
- 1.5.2 มีความเข้าใจเกี่ยวกับหลักการทำงานของเครื่องนับถาดอัตโนมัติ
- 1.5.3 มีความเข้าใจเกี่ยวกับหลักการทำงานของระบบ RFID
- 1.5.4 มีความเข้าใจเกี่ยวกับการเก็บข้อมูลลงใน Firebase
- 1.5.5 สามารถเลือกใช้อุปกรณ์ต่างๆ ในการออกแบบได้อย่างเหมาะสม
- 1.5.6 สามารถวิเคราะห์และแก้ไขปัญหาที่เกิดขึ้นระหว่างปฏิบัติงานได้

1.6 ระยะเวลาในการทำโครงการ

สิงหาคม 2566 - มีนาคม 2567

รายละเอียด	สิงหาคม	กันยายน	ตุลาคม	ธันวาคม	มกราคม	กุมภาพันธ์
วางแผนการดำเนินการ	↔					
ซื้ออุปกรณ์		↔				
วางแผนการติดตั้ง		↔				
ทดสอบวงจร		↔	↔			
ทดสอบการทำงานของ สายพาน			↔			
ทดสอบการทำงานของระบบ RFID			↔	↔		
เก็บผลการทดลอง					↔	
ตรวจสอบและแก้ไขปัญหา					↔	↔
สรุปผลการทดลอง						↔
จัดทำรูปเล่มโครงการ						↔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเอกสารที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller มักย่อว่า μC หรือ MCU) คืออุปกรณ์ควบคุมขนาดเล็กซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ ได้รวมเอาซีพียู หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน ไมโครคอนโทรลเลอร์ถ้าแปลความหมายแบบตรงตัวก็คือ ระบบคอนโทรลขนาดเล็กเรียกอีกอย่าง หนึ่งคือเป็นระบบคอมพิวเตอร์ขนาดเล็ก ที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย โดยผ่านการออกแบบ วงจรให้เหมาะกับงานต่างๆ และยังสามารถโปรแกรมคำสั่งเพื่อควบคุมขา Input / Output เพื่อสั่งงานให้ไป ควบคุมอุปกรณ์ต่างๆ ได้อีกด้วย ซึ่งก็นับว่าเป็นระบบที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย ทั้งทางด้าน Digital และ Analog ยกตัวอย่างเช่น ระบบสัญญาณตอบรับอัตโนมัติ, ระบบบัตรคิว, ระบบตอกบัตร พนักงาน และอื่นๆ ยิ่งระบบไมโครคอนโทรลเลอร์ในยุคปัจจุบันนั้นสามารถทำการเชื่อมต่อกับระบบ Network ของคอมพิวเตอร์ทั่วไปได้อีกด้วย ดังนั้นการสั่งงานจึงไม่ใช่แค่หน้าแผงวงจร แต่อาจจะเป็นการสั่งงานอยู่คนละซีกโลกผ่านเครือข่ายอินเทอร์เน็ตก็ได้

2.1.1 โครงสร้างทั่วไป

โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)
2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกับกระดานขดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Electrically Erasable Programmable Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยงก็ตาม

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะใช้การกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัว ไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) , บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา เป็นองค์ประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะ หากสัญญาณนาฬิกาที่มีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถึงขั้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

2.1.2 การสื่อสารระหว่างอุปกรณ์

การสื่อสารระหว่างอุปกรณ์สามารถแบ่งได้ 2 รูปแบบ คือ

1. การสื่อสารแบบขนาน

คือการใช้สาย 1 เส้น แขนบิตข้อมูล 1 บิต โดยทั่วไปแล้วมักจะนิยมใช้งานแบบ 4 บิต 7 บิต และ 8 บิต โดยจะมีสาย 1 เส้นเป็นสายควบคุมที่บอกว่าจะให้รับข้อมูลเขาไปเมื่อใด ข้อดีของการสื่อสารรูปแบบนี้คือ มีความเร็วที่สูงมาก และมีข้อเสียคือใช้สายจำนวนมากในการสื่อสาร การสื่อสารแบบขนานใช้ในอุปกรณ์อ่านจอ LCD ขนาดต่าง ๆ ซึ่งจะใช้งานได้ทั้งแบบ 4 บิต และ 8 บิต ใช้ขา EN ในการควบคุมการรับข้อมูล

2. การสื่อสารแบบอนุกรม

คือการใช้สาย 1 เส้นรับ-ส่งข้อมูลแบบต่อเนื่อง โดยอาศัยเทคนิคต่าง ๆ ในการสื่อสาร เช่นการใช้สัญญาณทริกเพื่อรับข้อมูลเข้า การใช้บิตเริ่มต้นกำหนดการรับข้อมูล โดยอาจจะอาศัยและไม่อาศัยเวลาในการทำงาน ทั้งนี้การสื่อสารแบบอนุกรมสามารถแบ่งได้เป็น 2 รูปแบบคือ

แบบซิงโครนัส (Synchronous) – เป็นการสื่อสารที่ใช้สายสัญญาณข้อมูลอย่างน้อย 1 เส้น และมีสายอีก 1 เส้นกำหนดจังหวะการรับข้อมูล ข้อดีของการสื่อสารแบบนี้คือการรับส่งข้อมูลมีความผิดพลาดน้อยหรือไม่มีความผิดพลาดเลย แต่ข้อเสียคือต้องใช้สายสัญญาณอย่างน้อย 2 เส้นในการสื่อสาร โดยโปรโตคอลที่ทำงานแบบซิงโครนัสได้แก่ I2C I2S และ SPI

แบบอะซิงโครนัส (Asynchronous) – เป็นการสื่อสารที่ใช้สายสัญญาณข้อมูลเพียงเส้นเดียวในการทำงาน โดยอาศัยสัญญาณจากบิตเริ่มต้น และบิตสิ้นสุดในการบอกจังหวะการรับส่งข้อมูล การสื่อสารแบบนี้จำเป็นต้องอาศัยเวลามาเป็นตัวกำหนดการรับสัญญาณเข้ามาซึ่งหากมีการตั้งค่าที่ผิดจะทำให้อ่านข้อมูลที่ส่งมาได้ผิดพลาด ข้อดีของการสื่อสารแบบนี้คือใช้สายสัญญาณเพียง 1 เส้นก็สามารถรับ-ส่งข้อมูลได้แล้ว แต่ข้อเสียคือมีความผิดพลาดในการสื่อสารได้ง่าย โดยโปรโตคอลที่ทำงานแบบอะซิงโครนัสคือ UART

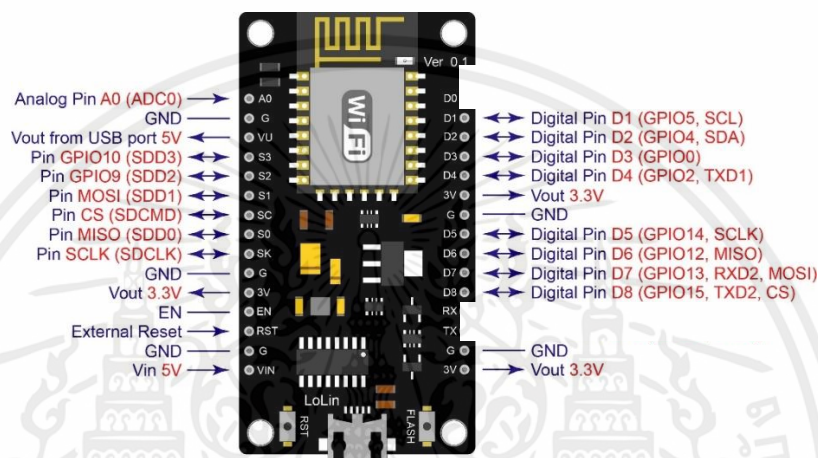
ทั้งนี้การสื่อสารแบบอนุกรมจะช้ากว่าการสื่อสารแบบขนานมาก แต่มีข้อดีคือโปรโตคอลบางตัวสามารถต่อพ่วงอุปกรณ์ได้หลายตัวในขณะที่ใช้สายเท่าเดิม หรือใช้สายเพิ่มเพียงไม่กี่เส้น

2.1.3 ESP8266

ESP8266 คือโมดูล WiFi จากจีน ที่มีความพิเศษตรงที่ตัวมันสามารถโปรแกรมลงไปได้ ทำให้สามารถนำไปใช้งานแทนไมโครคอนโทรลเลอร์ได้เลย และมีพื้นที่โปรแกรมที่มากถึง 4MB ทำให้มีพื้นที่เหลือมากในการเขียนโปรแกรมลงไป

ESP8266 เป็นชื่อของชิปไอซีบนบอร์ดของโมดูล ซึ่งไอซี ESP8266 ไม่มีพื้นที่โปรแกรม (flash memory) ในตัว ทำให้ต้องใช้ไอซีภายนอก (external flash memory) ในการเก็บโปรแกรม ที่ใช้การเชื่อมต่อผ่านโปรโตคอล SPI ซึ่งสาเหตุนี้เองทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่าไอซีไมโครคอนโทรลเลอร์เบอร์อื่นๆ

ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3V - 3.6V การนำไปใช้งานร่วมกับเซ็นเซอร์อื่นๆที่ใช้แรงดัน 5V ต้องใช้วงจรแบ่งแรงดันมาช่วย เพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200mA ความถี่คริสตอล 40MHz ทำให้เมื่อนำไปใช้งานอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การแสดงผลข้อมูลเร็วกว่าไมโครคอนโทรลเลอร์ยี่ห้ออื่น Arduino มาก



ภาพที่ 2.1 รูปบอร์ด NodeMCU V3 และขาต่างๆ

ที่มา <https://1th.me/eNsGy>

2.2 วงจรที่ใช้ในการทำโครงการ

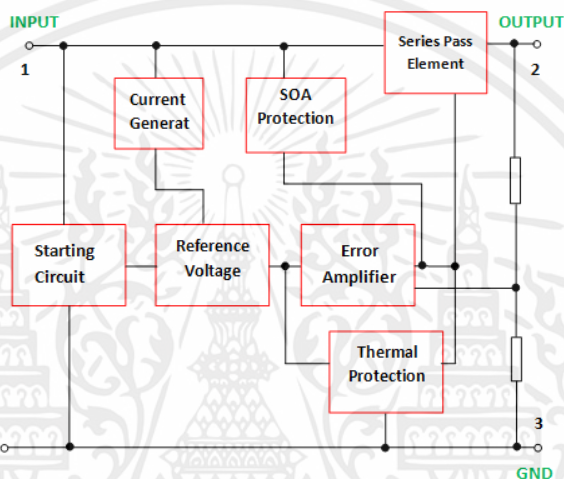
2.2.1 วงจรรักษาแรงดัน (Voltage regulator)

เลือกใช้ IC LM7805, LM7812 เพราะ IC LM78xx เป็นวงจรรวมตัวควบคุมที่ควบคุมแรงดันไฟฟ้าเชิงเส้นคงที่ xx ที่มีอยู่ใน 78xx หมายถึงค่าของแรงดันเอาต์พุตคงที่ที่เฉพาะเจาะจง โดย regulator นี้ยังเพิ่มข้อกำหนดสำหรับชุดระบายความร้อน แรงดันไฟฟ้าขาเข้าของเครื่องปรับแรงดันไฟฟ้านี้สามารถสูงถึง 35V และ IC นี้สามารถให้ 5V คงที่สำหรับค่าอินพุตใด ๆ ที่น้อยกว่าหรือเท่ากับ 35V ซึ่งเป็นขีดจำกัด

บล็อกไดอะแกรมภายในของ 78XX Voltage Regulator

บล็อกไดอะแกรมภายในของ IC 7805 แสดงในรูปด้านล่าง:

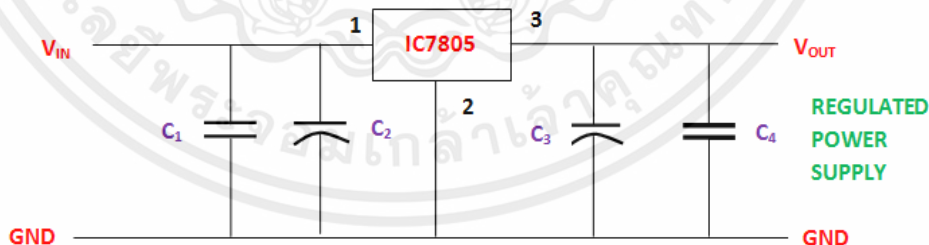
บล็อกไดอะแกรมประกอบด้วย Error Amplifier, เครื่องกำเนิดกระแสไฟฟ้า, แรงดันไฟฟ้าอ้างอิง, วงจรเริ่มต้น, การป้องกัน SOA และการป้องกันความร้อน



ภาพที่ 2.2 (ก) วงจรรักษาแรงดัน (Voltage regulator)

ที่มา <https://riverglennapts.com/th/current-voltage/234-voltage-regulator-7805.html>

วงจรพาวเวอร์ซัพพลายที่มีการควบคุม



ภาพที่ 2.2 (ข) วงจรพาวเวอร์ซัพพลายที่มีการควบคุม

ที่มา <https://riverglennapts.com/th/current-voltage/234-voltage-regulator-7805.html>

C1 เป็นตัวเก็บประจุบายพาส

C2 และ C3 เป็นตัวเก็บประจุกรอง C2 ใช้เพื่อทำให้การเปลี่ยนแปลงซ้ำในแรงดันไฟฟ้าที่ป้อนให้กับวงจรในรูปแบบคงที่

C3 ใช้เพื่อทำการเปลี่ยนแปลงที่ซ้ำในผลลัพธ์แรงดันไฟฟ้าจากตัวควบคุมในวงจรไปยังรูปแบบคงที่ เมื่อค่าของตัวเก็บประจุเหล่านี้เพิ่มขึ้นความเสถียรจะขยายใหญ่ขึ้น แต่ตัวเก็บประจุเหล่านี้โดดเดี่ยวเดียวดายไม่สามารถกรองการเปลี่ยนแปลงของแรงดันไฟฟ้าอินพุตและเอาต์พุต

C4 เป็นตัวเก็บประจุบายพาสที่ใช้ในการข้ามขอบเขตขนาดเล็กมากไปยังพื้นดินหรือดิน สิ่งนี้ทำได้โดยไม่ส่งผลกระทบต่อส่วนประกอบอื่นๆ

2.2.2 วงจรขับมอเตอร์ (Drive Motor)

ในที่นี้เลือกใช้เป็นบอร์ดโมดูล A4988 เป็นโมดูลบอร์ด สำหรับควบคุม Stepper Motor โดยเฉพาะ และสามารถควบคุม Stepper Motor แบบ Bipolar ใช้วงจร H-Bridge มาใช้ในการขับ Stepper Motor ซึ่ง Stepper Motor เป็นแบบพื้นฐานที่ได้รับความนิยมสูงเช่นเดียวกัน และสามารถนำไปประยุกต์ใช้ได้หลากหลายรูปแบบเช่น งานหุ่นยนต์ เครื่อง CNC หรือ 3D Printer เป็นต้น

หลักการทำงาน

วงจร H-Bridge จะขับกระแสเข้ามอเตอร์ ตามขั้วที่กำหนดด้วยลอจิกเพื่อควบคุมทิศทาง ส่วนความเร็วของมอเตอร์นั้นจะถูกควบคุมด้วย สัญญาณ (PWM Pulse Width Modulation) จากหลักการของ H-Bridge ทำให้สามารถใช้อุปกรณ์อิเล็กทรอนิกส์ทำหน้าที่ตัด-ต่อวงจรแทนสวิทช์ได้ ทั้งการใช้รีเลย์ และการใช้อุปกรณ์สารกึ่งตัวนำ

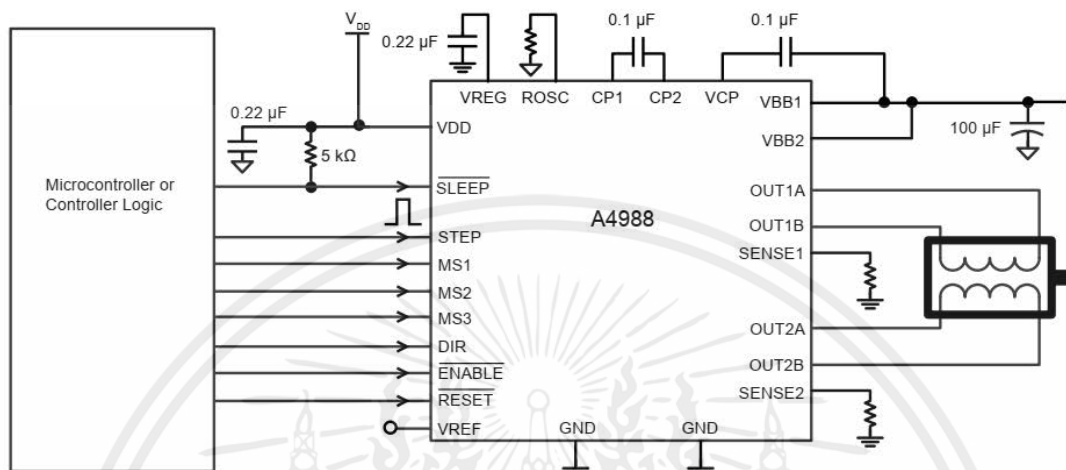
การควบคุมมอเตอร์สองทิศทางด้วยไอซี

ไอซีที่ใช้นำมาขับมอเตอร์ คือ A4988

A4988 เป็นไอซี DMOS Microstepping Driver with Translator ซึ่งคุณสมบัติของไอซีตัวนี้สามารถรองรับสัญญาณได้ทั้ง 3.3 และ 5 โวลต์ มีวงจรป้องกันต่าง ๆ เช่น ตัดการทำงานเมื่ออุณหภูมิสูงเกิน ป้องกันการ Short-to-Ground ป้องกันการ Short-Load อีกทั้งยังสามารถรองรับการทำงานใน Step Mode ได้อย่างครบถ้วนทั้ง 1/1(Full Step), 1/2(Half Step), 1/4, 1/8, 1/16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งานนั้นเราต้องต่อวงจรอิเล็กทรอนิกส์ที่แสดงใน Datasheet



ภาพที่ 2.3 (ก) แสดงการต่อวงจรขับมอเตอร์กับมอเตอร์ เข้ากับบอร์ดไมโครคอนโทรลเลอร์
ที่มา https://fu.lnwfile.com/_fu/_raw/m0/p5/22.png

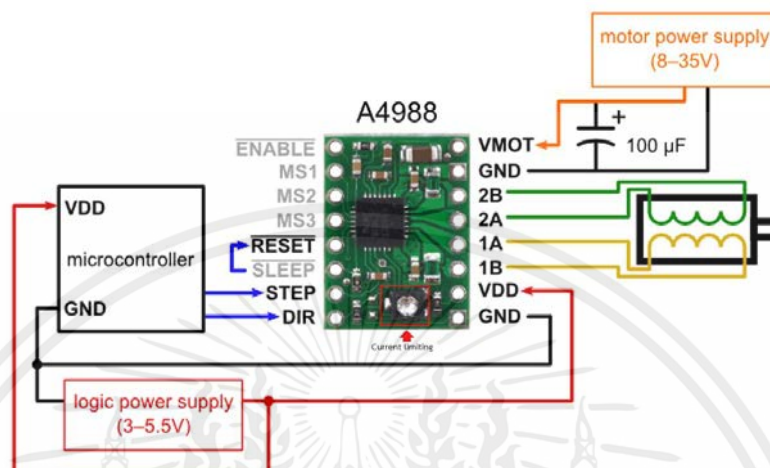
โมดูลขับมอเตอร์ A4988



ภาพที่ 2.3 (ข) โมดูล A4988 ควบคุม Stepper Motor
ที่มา https://fu.lnwfile.com/_fu/_raw/3r/lf/sx.jpg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pins ต่างๆ ของ A4988



ภาพที่ 2.4 แสดงขา Pins ของโมดูล A4988

ที่มา <https://commandronstore.com/products/cm4988.php>

- เริ่มจาก VMOT และ GND คือขาสำหรับไฟเลี้ยงมอเตอร์ 8-35 V มอเตอร์จะนำไฟเลี้ยงจากส่วนนี้ไปใช้ ซึ่งประกอบด้วย แรงดันและกระแสตามสเปคที่มอเตอร์ต้องการ ซึ่งในที่นี้ใช้มอเตอร์ NEMA 17
- Stepper ที่ใช้แรงดันต่ำกว่า 8 - 35 V ก็สามารถใช้กับโมดูลตัวนี้ได้ เนื่องจากตัวโมดูล สามารถเปลี่ยนแรงดันที่เข้ามายังขา VMOT ให้กลายเป็นกระแสที่มอเตอร์ต้องใช้ โดยการลดแรงดันและเพิ่มกระแส นั่นเอง โดยเราจะต้องปรับ Current Limit ของตัวโมดูล ซึ่งผลรวมกำลัง ($P = IV$) ของ Output (ไปยังมอเตอร์) จะน้อยกว่ากำลังที่ Input เข้ามาทาง VMOT เสมอ โดยกำลังที่หายไป จะเปลี่ยนไปเป็นความร้อน

ที่สำคัญระหว่างขา VMOT กับ GND ต้องต่อตัวเก็บประจุอย่างน้อย 47 μF ขึ้นไป (แนะนำให้ใช้ 100 - 300 μF) เพื่อป้องกันการเกิด LC Voltage Spikes

- ขา 2B - 1B ต่อเข้ากับสายไฟของมอเตอร์ตามขั้วที่บอกมาใน Datasheet ของมอเตอร์นั้นๆ
- VDD และ GND คือขาไฟเลี้ยง Logic 3 - 5.5 V ที่จะกำหนดสถานะ HIGH / LOW ให้เฟสต่างๆของมอเตอร์ ทำให้มอเตอร์เกิดการหมุนตามต้องการ โดยทั่วไปอาจจะใช้ไฟ 5 V จาก Arduino โดยตรงเลย แต่แนะนำว่า ควรนำแหล่งจ่าย 5 V ภายนอกมาต่อแทน ในกรณีที่มีวงจรมีอุปกรณ์ไฟฟ้ามากๆ เพราะไฟ 5 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ Arduino อาจจ่ายกระแสได้ไม่เพียงพอ ที่สำคัญคือ อย่าลืมต่อ GND ของ Arduino เข้ากับ GND ของวงจร Driver ด้วย

- DIR และ STEP คือขาควบคุมการหมุนของมอเตอร์ ซึ่งจะเป็นการระบุ ทิศทาง องศา และความเร็วในการหมุน โดยจะต่อเข้ากับ Pin Control ที่กำหนดไว้ใน Code ของ Arduino
- Sleep และ Reset สองขา นี้ จะต่อเข้าด้วยกัน เนื่องจากขา Sleep จะทำหน้าที่ส่งกำลังไฟฟ้าน้อยที่สุดที่ทำให้ให้มอเตอร์ ในขณะที่มอเตอร์ยังไม่หมุน เพื่อเป็นการประหยัดพลังงาน ส่วนขา Reset จะกำหนดจุดเริ่มแรกหรือ Home State ในการหมุนของมอเตอร์ ซึ่งถ้าเราปล่อยขานี้ให้ว่างไว้ หรือเป็น LOW จะทำให้มอเตอร์ไม่หมุน (Disable) ดังนั้นเราจึงต้องต่อขา Sleep เข้ากับขา Reset เพื่อให้มอเตอร์อยู่ในโหมด Enable
- MS1 - MS3 ทั้ง 3 ขานี้จะใช้ในการกำหนด Step Resolution ของ Stepper Motor ซึ่งถือเป็นอีกหนึ่งสิ่งที่น่าสนใจในการใช้งาน Stepper ให้มีประสิทธิภาพสูงสุด
- Enable ขาสุดท้ายนี้ จะถูกปล่อยให้เป็น LOW ตลอดการใช้งาน เนื่องจากถ้ามีสถานะมีสถานะเป็น HIGH จะทำให้ Output ไปยัง Motor เป็น Disable Mode

การปรับ Current limiting

ก่อนที่จะเริ่มใช้งานกับมอเตอร์นั้นเราจะต้องกำหนดกระแสก่อน เพื่อป้องกันไม่ให้เกิดกระแสไฟฟ้าไหลเข้าขดลวดของมอเตอร์มากเกินไปซึ่งจะส่งผลให้มอเตอร์เกิดความเสียหาย โดยการปรับกระแสนี้จะทำการปรับตัวต้านทานปรับค่าได้แบบ Trimmer บนตัวโมดูล โดยมี 2 วิธี

1. หา Datasheet ของมอเตอร์ที่เราจะใช้ มองหา Rated Current ตัวอย่างใช้มอเตอร์ NEMA17 1.2A

General specifications		Electrical specifications	
Step Angle (°)	1.8	Rated Current (A)	1.2
Temperature Rise (°C)	80 Max (rated current, 2 phase on)	Inductance Per Phase (±20% mH)	2.8
Ambient temperature (°C)	-20 ~ +50	Holding Torque (Kg.cm)	3.17
Number of Phase	2	Detent Torque (g.cm)	200
Insulation Resistance	100MΩ, Min (500VDC)	Rotor Inertia (g.cm ²)	68
Insulation Class	Class B	Weight (Kg)	0.365
Max radial force (N)	28 (20mm from the flange)		
Max axial force (N)	10		

● Pull out torque curve

ภาพที่ 2.5 แสดง Datasheet ของมอเตอร์
ที่มา <https://www.tme.eu/Document/.pdf>

2. คำนวณ จากสูตร

$$\text{Current Limit} = V_{\text{ref}} \times 2.5$$

ในที่นี้ Current Limit เราคือ 1.2A คำนวณ V_{ref} ได้ 0.48 เริ่มแรกตั้งโมดูลให้เป็น Full Step Mode แล้วทำการจ่ายไฟ ต่อมอเตอร์เข้าไป วัดค่า V_{ref} โดยนำโวลต์มิเตอร์จิ้มที่ตัว Trimmer เทียบกราวด์ ค่อยๆ หมุนปรับจนได้ค่าตามที่คำนวณไว้

2.3 อุปกรณ์ที่เกี่ยวข้อง

2.3.1 เซนเซอร์ตรวจจับวัตถุด้วยแสงอินฟราเรด (Infrared Proximity Sensor)

- ชนิดที่ใช้ในโครงการนี้ คือ IR Sensor Switch E18-D80NK-N

คือ Sensor Infrared ตรวจจับระยะทาง สามารถปรับย่านการตรวจจับได้ตั้งแต่ 3 cm- 80 cm

ให้ Output ออกมาเป็นแบบ Logic TTL คือ 0 (GND) และ 1 (5V)

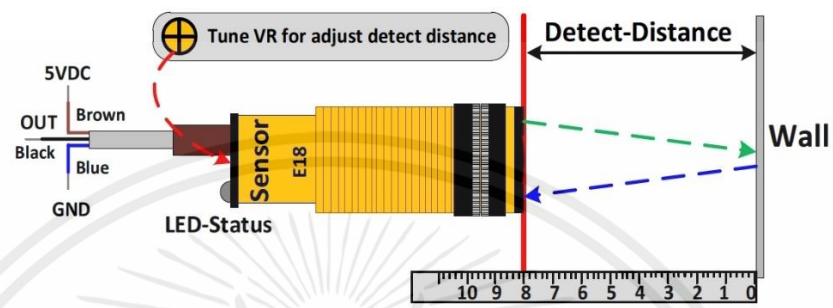
คุณสมบัติ

1. ระยะทางการตรวจจับ สามารถปรับได้ตั้งแต่ 3 cm - 80 cm ด้วย VR ปรับค่า และแสดงสถานะ ระยะตรวจจับด้วย LED
2. วัตถุที่ใช้ Sensing จะต้องเป็นแบบทึบแสงหรือแสงผ่านได้น้อยและควรเป็นสีดำจะดีที่สุด เนื่องจากตัว Sensor ทำงานโดยใช้การสะท้อนของ Infrared
 - OUTPUT เป็นแบบ Open Collector เวลาใช้งานต้องต่อ R 10 K Ω Pull Up ที่ Output ด้วย
 - และให้สัญญาณ Output แบบ ดิจิตอล TTL คือ 0= GND และ 1 = 5V
3. ใช้ไฟเลี้ยง DC 5V กระแส 100 mA

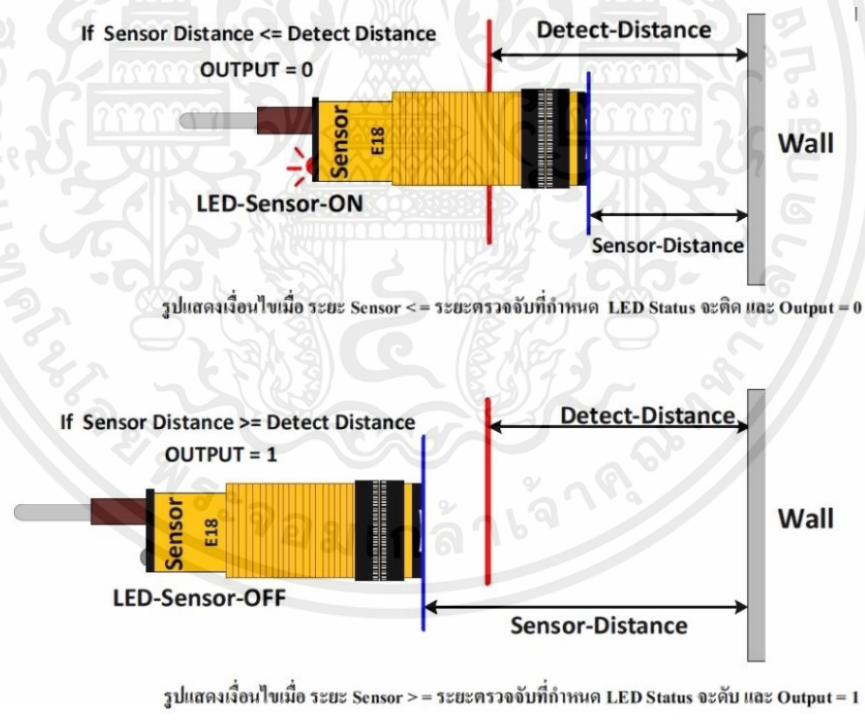
การตั้งระยะตรวจจับ ก่อนการใช้งานจะต้องตั้งระยะการตรวจจับที่ผู้ใช้ต้องการใช้งานให้กับ Sensor ดังนี้

1. จ่ายไฟเลี้ยง 5V และ GND ให้กับตัว Sensor
2. หันหัว Sensor ให้ตั้งฉากกับพื้นหรือกำแพง
3. ใช้ไม้บรรทัดวัดระยะที่ต้องการตรวจจับจากพื้น มายังหัว Sensor และจับ Sensor ค้างไว้ในระยะที่ต้องการตรวจจับ
4. ปรับ VR ที่ด้านท้ายของตัว Sensor และสังเกตการเปลี่ยนแปลงของ LED ที่ด้านท้ายของตัว Sensor ดังนี้
 - ถ้า LED ดับอยู่ (OUTPUT=1) ให้ปรับ VR ตามเข็มนาฬิกาจนเห็น LED ติด (OUTPUT=0)แล้วให้หยุดปรับจุดที่ LED เปลี่ยนสถานะนี้คือระยะ Detect ที่เราต้องการจะได้เงื่อนไขการทำงาน คือ ถ้าระยะ Sensor น้อยกว่าหรือเท่ากับระยะ Detect LED Status จะติดและจะได้ OUTPUT เป็น Logic 0 แต่ถ้าระยะ Sensor มากกว่าระยะ Detect LED Status จะดับและจะได้ OUTPUT เป็น Logic 1
 - ถ้า LED ติดอยู่ (OUTPUT=0) ให้ปรับ VR ทวนเข็มนาฬิกาจนเห็น LED ดับ (OUTPUT=1) แล้วให้หยุด ปรับจุดที่ LED เปลี่ยนสถานะนี้คือระยะ Detect ที่เราต้องการ จะได้เงื่อนไขการทำงานคือ ถ้าระยะ Sensor

มากกว่าหรือเท่ากับระยะ Detect LED Status จะดับและจะได้ OUTPUT เป็น Logic 1 แต่ถ้าระยะ Sensor น้อยกว่าระยะ Detect LED Status จะติดและจะได้ OUTPUT เป็น Logic 0



ภาพที่ 2.6 (ก) เซ็นเซอร์แสงสำหรับตรวจจับวัตถุ
ที่มา <https://www.ett.co.th/productSensor/E18-D80NK/E18-D80NK.html>



ภาพที่ 2.6 (ข) แสดงเงื่อนไขการทำงานของเซนเซอร์ตรวจจับวัตถุ
ที่มา <https://www.ett.co.th/productSensor/E18-D80NK/E18-D80NK.html>

2.3.2 เซ็นเซอร์ชนิดอัลตราโซนิก (Ultrasonic sensor)

สามารถใช้ตรวจเช็ควัตถุ โดยใช้หลักการสะท้อนไป-กลับของคลื่นเสียงจากเซนเซอร์ไปยังวัตถุเป้าหมาย จากนั้นแปลงค่าการสะท้อนของเสียงออกมาในรูปแบบของสัญญาณทางไฟฟ้า คลื่นอัลตราโซนิกนั้นเดินทางรวดเร็วกว่าความเร็วของเสียงที่มนุษย์เราได้ยิน

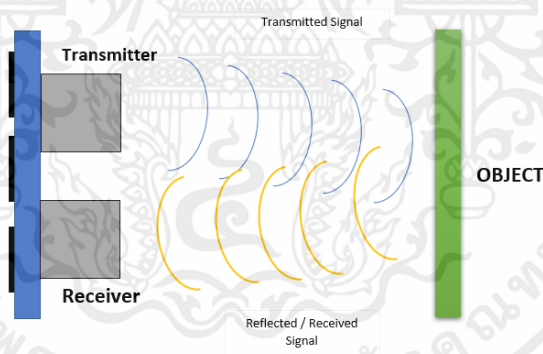
เซนเซอร์อัลตราโซนิก ประกอบไปด้วย 2 องค์ประกอบ คือ

1. ตัวส่งเสียง (Transmitter)

2. ตัวรับเสียง (Receiver)

1. ตัวส่งเสียง (Transmitter) โดยใช้ผลึกเพียโซอิเล็กทริกในการส่งสัญญาณ

2. ตัวรับเสียง (Receiver) รับค่าของสัญญาณที่เกิดจากการเดินทางไปและกลับจากวัตถุเป้าหมาย แปลงออกมาเป็นค่าทางไฟฟ้า หรือค่าระยะทาง



ภาพที่ 2.7 การทำงานของคลื่นอัลตราโซนิก

ที่มา <http://jwtech.co.th/activity/?p=881>

ในการใช้งานตัวเซ็นเซอร์วัดระยะทางแบบ Ultrasonic Sensor ควรจะต้องพิจารณาเรื่องผลกระทบทางด้านอุณหภูมิที่มีต่อการทำงานของเซ็นเซอร์ โดยอุณหภูมิที่เปลี่ยนแปลงไปจะมีผลทำให้ความเร็วในการเคลื่อนที่ของคลื่นเสียงเปลี่ยนแปลงไปด้วย ซึ่งสามารถอธิบายได้จากสมการด้านล่าง

$$C = 331 + 0.6t \text{ (m/s)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C = ความเร็วในการเคลื่อนที่ของเสียง

t = ค่าอุณหภูมิ ณ ขณะนั้น หน่วยเป็น เซลเซียส

จะเห็นได้ว่าความเร็วในการเคลื่อนที่ของเสียงนั้น จะเปลี่ยนแปลงตลอดเวลา ที่อุณหภูมิเปลี่ยนแปลง ซึ่งอัลตราโซนิกเซ็นเซอร์นั้นจะใช้หลักการการสะท้อนของคลื่นเสียงกับวัตถุ แล้วหาค่าเวลา ซึ่งถ้าความเร็วคลื่นเสียงไม่คงที่แล้ว อาจทำให้การวัดค่ามีความคลาดเคลื่อนได้

ปัจจัยที่ส่งผลต่อประสิทธิภาพของ Ultrasonic sensor

1. ระยะเวลาจำกัดของการตรวจจับ (Blind zone)

Ultrasonic sensor จะมีระยะเวลาจำกัดของการตรวจจับอยู่ ซึ่งจะทำให้เซ็นเซอร์ตรวจจับไม่ได้โดยสามารถดูค่า Blind zone ของ Ultrasonic sensor จากคู่มือของตัวเซ็นเซอร์ เช่น Ultrasonic sensor รุ่น UD800 จะมี Blind zone อยู่ที่ 0.4 ถึง 1.8 เมตร

2. ผลกระทบด้านอุณหภูมิ (Effect of Temperature)

อุณหภูมิจะส่งผลของคลื่น Ultrasonic ดังนั้นในการใช้งานของ Ultrasonic sensor จะต้องใช้อุณหภูมิที่เหมาะสม

3. มุมตกกระทบวัตถุภายใน

การใช้งานของ Ultrasonic sensor ได้ดีที่สุดที่วัตถุที่กระทบต้องมีลักษณะตั้งฉากกับ Ultrasonic sensor ดังนั้นแล้วถ้ามุมของวัตถุมีองศาเพิ่มขึ้นจะทำให้ประสิทธิภาพของการใช้งาน Ultrasonic sensor ลดลง โดยส่วนใหญ่มุมของวัตถุจะต้องมีค่าเอียงน้อยกว่า 10 องศา และวัตถุที่จะทำการสะท้อนจากคลื่น Ultrasonic ควรจะมีลักษณะเป็นแนวราบ เพื่อให้ได้ประสิทธิภาพในการตรวจจับที่ดีที่สุด ถ้าวัตถุที่ทำการตรวจจับเป็นชิ้นเล็กๆ หรือมีรูปร่างที่ไม่แน่นอน จะทำให้ความแม่นยำลดลง

4. ตำแหน่งการติดตั้งของ Ultrasonic sensor ที่เหมาะสม

การติดตั้งนั้นจะส่งผลต่อการวัดค่าและความแม่นยำ ดังนั้นตำแหน่งของการติดตั้ง Ultrasonic sensor จะมี 4 ประเภทดังนี้

4.1 การติดตั้ง Ultrasonic Sensor บริเวณที่โดนแสงแดด การติดตั้ง Ultrasonic Sensor ควรมียูปรกรณ์ป้องกันเซ็นเซอร์ที่จะได้รับความเสียหายจาก ความร้อน ซึ่งจะส่งต่อความแม่นยำ

4.2 การติดตั้ง Ultrasonic ในกรณีที่มีการเติมวัตถุ เข้าไปในถังแบบอาศัยแรงโน้มถ่วงของโลก จุดที่เหมาะสมกับการติดตั้ง Ultrasonic Sensor นั้นควรอยู่ให้ห่างจากจุดเติม เพื่อป้องกันการทํางานผิดพลาดหรือความเสียหายแก่ตัวเซ็นเซอร์

4.3 การติดตั้ง Ultrasonic ในกรณีที่มีการปล่อยของออกด้านตรงจุดกลางของถัง ควรดูค่ามุมส่งสัญญาณ (Emission Angle) ของ Ultrasonic sensor เพื่อให้การวัดมีความแม่นยำที่สุด

4.4 การติดตั้ง Ultrasonic ที่มีอุปกรณ์ประเภท Pneumatic จะสามารถรบกวนการส่งคลื่น Ultrasonic ได้ซึ่งจะทำให้ Ultrasonic sensor มีความแม่นยำลดลง ดังนั้นไม่ควรติดตั้ง Ultrasonic sensor ใกล้กับจุดที่โดนลม

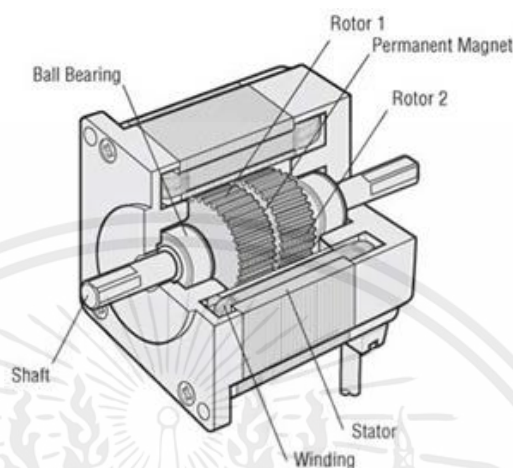
การประยุกต์ใช้ Ultrasonic sensor สามารถประยุกต์การใช้งานได้หลากหลายในอุตสาหกรรม โดยไม่ได้จำกัดการใช้ในงานวัดระดับความสูงของวัตถุเท่านั้น ดังนั้น Ultrasonic Sensor ประยุกต์ใช้งานได้ดังนี้

1. การใช้งาน Ultrasonic Sensor สำหรับงานเครน
2. เซ็นต์ตำแหน่งของรถยนต์ในลานจอดรถ
3. เซ็นต์ระดับความสูงของแผ่นไม้แต่ละกอง

2.3.3 มอเตอร์ Stepper รุ่น NEMA 17 Stepper Motor

Stepper Motor เป็นมอเตอร์ไฟฟ้าชนิดหนึ่งที่ใช้สำหรับควบคุมการหมุนเพื่อกำหนดตำแหน่งและทิศทางด้วยมุมมองให้กับกลไกการทํางานของเครื่องจักรกลที่ต้องการความแม่นยำสูง เช่น ระบบควบคุมในหุ่นยนต์, เครื่องพิมพ์ และสายพานลำเลียงในโรงงาน ซึ่งอาศัยการทํางานจากการใช้สัญญาณไฟฟ้าแบบคลื่นสี่เหลี่ยม (Pulse) ขับมอเตอร์เพื่อให้แกนหมุนแบบไม่ต่อเนื่องแต่จะขับเคลื่อนไปทีละขั้นโดยสามารถหมุนได้รอบแกน 360 องศา ทำให้กลไกการเคลื่อนที่ของแกนหมุนมีความเร็วต่ำและรักษาแรงบิดได้ในทันทีโดยไม่ทำให้มอเตอร์เสียหาย

ส่วนประกอบของสเต็ปมอเตอร์



ภาพที่ 2.8 โครงสร้าง Stepper Motor

ที่มา <https://www.orientalmotor.com/images/stepper-motors/stepper-motor-structural-diagram.jpg>

1. Stator เป็นส่วนที่ติดกับตัวถังของมอเตอร์ จะเป็นขั้วแม่เหล็กที่มีปลายเป็นฟันซี่เล็ก ๆ พันด้วยขดลวด เพื่อเหนี่ยวนำให้สนามแม่เหล็กเปลี่ยนแปลง
2. Rotor เป็นก้อนแม่เหล็กถาวร มีลักษณะคล้ายเฟือง มีฟันเป็นซี่ ๆ เพื่อสอดเข้าไปตรงกันกับซี่ฟันของ Stator โดย Rotor นี้ก็จะติดอยู่กับแกนหมุนเพื่อนำไปใช้งานตามต้องการ

โหมดการทำงาน

เนื่องจาก Stator ของ Stepper Motor ประกอบด้วยขดลวดหลายขด การจ่ายไฟฟ้าเข้าไป กระตุ้นที่ขั้วขดลวดต่าง ๆ ย่อมทำให้เกิดการทำงานที่แตกต่างกัน สามารถแยกได้ดังนี้

Full Step Mode ในโหมดนี้เมื่อจ่าย Pulse เข้าไป 1 ลูก มอเตอร์จะขยับ 1 Step เต็ม ๆ เช่น มอเตอร์ NEMA 17 มีสเปคคือ 200 Step/Revolution ฉะนั้นหากต้องการให้มอเตอร์หมุนครบรอบ ต้องจ่ายเป็นจำนวน 200 Pulse

Half Step Mode การทำงานแบบ Half Step Mode ทำให้ได้ความละเอียดเพิ่มขึ้น โดยการจ่าย Pulse แต่ละครั้ง Rotor จะหมุนครึ่งละครึ่งสเต็ป เช่น NEMA 17 หากใช้ Full Step จะหมุนครึ่งละ

1.8 องศา แต่หากใช้ Half Step จะหมุนเพียง 0.9 องศาเท่านั้น และหากต้องการให้หมุนครบรอบ 360 องศา ก็ต้องจ่าย Pulse เพิ่มขึ้นจาก 200 เป็น 400 ครั้ง

Micro Step Mode เป็นโหมดที่มีความซับซ้อนที่สุด แต่ให้ผลการหมุนของแกนที่ละเอียด ให้แรงบิดสูง โดยวิธีนี้ขดลวดจะถูกกระตุ้นด้วยคลื่นไซน์ (sin wave) จำนวนสองลูกต่างเฟสกัน 90° ด้วยวิธีนี้ทำให้ควบคุม Pulse ที่เข้าไปได้ทั้งทิศทางและขนาด ทำให้สามารถควบคุมการหมุนของแกนมอเตอร์ได้อย่างละเอียดมาก ๆ (ทำได้ถึง 0.007°/Step) ซึ่งเป็นที่นิยมนำมาใช้งานเป็นอย่างมากในเครื่องมือต่าง ๆ เช่น เครื่อง 3D Printer, เครื่อง CNC

MS3	MS2	MS1	Step Increment (°)	Steps per Revolution
0	0	0	Full Step (1.8°)	200
0	0	1	Half Step (0.9°)	400
0	1	0	1/4 Step (0.45°)	800
0	1	1	1/8 Step (0.225°)	1600
1	0	0	1/16 Step (0.1125°)	3200
1	0	1	1/32 Step (0.05625°)	6400
1	1	0	1/32 Step (0.05625°)	6400
1	1	1	1/32 Step (0.05625°)	6400

ภาพที่ 2.9 โหมดการทำงานของ Nema 17 Stepper Motor

ที่มา <https://makersportal.com/blog/raspberry-pi-stepper-motor-control-with-nema-17>

2.4 ระบบ RFID

RFID คือเทคโนโลยีในการบ่งชี้แบบหนึ่ง ย่อมาจาก Radio Frequency Identification วัตถุประสงค์หลักเพื่อนำมาใช้ติดกับวัตถุต่างๆ เพื่อบ่งบอกถึงความเป็นตัวตนเฉพาะเจาะจง หรือเป็นหมายเลขประจำตัว เช่น ใช้แทนฉลาก หรือรหัสแท่ง (Bar Code) ที่ติดกับสินค้า หรือใช้เป็นป้ายที่ติดกับใบหูของสัตว์เลี้ยง การทำงานของ RFID อาศัยคลื่นวิทยุ โดยจะมีส่วนประกอบสำคัญคือ เครื่องอ่านกับแท็ก โดยแท็กจะมีการส่งข้อมูลที่เป็นเอกลักษณ์หรือหมายเลขประจำตัวออกมาเป็นคลื่นวิทยุ เมื่อถูกกระตุ้นด้วยกระบวนกรบางอย่างจากเครื่องอ่าน สำหรับเครื่องอ่านจะมีส่วนที่รับสัญญาณคลื่นวิทยุได้เพื่อถอดรหัสข้อมูลที่ส่งมาจากแท็ก

ปัจจุบันได้มีการพัฒนาการใช้งาน RFID ให้มีความสามารถมากขึ้น เช่น การนำไปใช้งานกับบัตรสมาร์ทการ์ดเป็นบัตรสมาร์ทการ์ดแบบไร้หน้าสัมผัส (Contactless Smart Card) ทำให้ออกจากจะมีการรับส่งข้อมูลหมายเลขประจำตัวแล้ว ก็ยังสามารถติดต่อสื่อสารและเขียนอ่านข้อมูลที่มีปริมาณมากกว่าเดิมได้

2.4.1 คลื่นวิทยุที่ใช้ในระบบ RFID

RFID สามารถใช้คลื่นวิทยุได้หลากหลายช่วงความถี่ แต่ที่นิยมกันมาก มีอยู่ 3 ช่วงความถี่ต่อไปนี้

1. คลื่นความถี่ต่ำ (LF) 125 KHz เป็นความถี่ที่มีการใช้งานทั่วไปในระบบของ RFID มีระยะในการรับส่งข้อมูลใกล้ ความเร็วในการอ่านค่อนข้างช้า ใช้มากในระบบ Access Control เช่น ระบบควบคุมการเปิดปิดประตู หรือระบบคีย์การ์ด, ระบบลงเวลาทำงาน, บัตรสมาชิก และใช้งานทางด้านปศุสัตว์
2. คลื่นความถี่สูง (HF) 13.56 MHz ใช้มากในงานทางด้านไอที และงานที่ต้องการความปลอดภัยสูง เช่น ใช้งานในบัตรสมาร์ทการ์ดแบบไร้หน้าสัมผัส (Contactless Smart Card) ซึ่งนิยมนำไปใช้เป็นบัตรเงินสด, บัตรเครดิต และบัตรโดยสารรถไฟฟ้า และมีการใช้งานเป็นแท็กตามมาตรฐาน NFC สำหรับโทรศัพท์มือถือ และใช้งานเป็นป้ายสินค้า
3. คลื่นความถี่สูงมาก (UHF) 860-960 MHz สามารถใช้งานได้ในระยะที่ไกลกว่าช่วงความถี่อื่น นิยมใช้งานเป็นป้ายสินค้าตามมาตรฐาน EPC ซึ่งถูกกำหนดมาให้ใช้งานแทนระบบบาร์โค้ด ดังนั้นจึงมีการใช้งานกันมากในงานทางด้านคลังสินค้า การผลิต และการจัดจำหน่าย

2.4.2 RFID Tag

- ชนิดโมดูลที่เลือกใช้ในโครงการนี้ คือ RFID-RC522

แท็ก คือชิ้นวัสดุที่มีวงจรรีเลย์ทรานซิสเตอร์อยู่ภายในและมีคุณสมบัติสามารถรับส่งข้อมูลกับเครื่องอ่านได้ผ่านทางคลื่นวิทยุ โดยอาจใช้คลื่นความถี่เท่าใด หรือใช้กระบวนการสื่อสารแบบใดก็ได้ และจะมีแบตเตอรี่อยู่ภายในหรือไม่ก็ได้ รูปร่างมีได้หลากหลายแล้วแต่ความต้องการในการใช้งาน

แท็กสามารถแบ่งออกได้เป็น 2 ชนิด คือ Active Tags กับ Passive Tags

1. Active Tag คือแท็กที่มีแหล่งจ่ายไฟในตัวเอง เช่น มีแบตเตอรี่อยู่ภายใน ทำให้แท็กชนิดนี้โดยมากจะสามารถรับส่งข้อมูลได้ในระยะไกล ในระดับเมตรขึ้นไปจนถึงหลายสิบลเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น แท็ก Easy Pass ที่เราติดไว้ที่หน้ากระจกรถเพื่อใช้กับระบบทางด่วน จะเป็นแท็กแบบนี้ Active Tag จึงมักมีขนาดใหญ่ และมีราคาที่สูงกว่าแท็กแบบธรรมดา

ข้อดี : ระยะการอ่านได้ไกล มีหน่วยความจำภายในขนาดใหญ่ นอกจากนี้ยังทำงานในบริเวณที่มีสัญญาณรบกวนได้ดี

ข้อเสีย : ราคาแพง และตัว tag ค่อนข้างมีขนาดใหญ่



ภาพที่ 2.10 แท็กแบบ Active

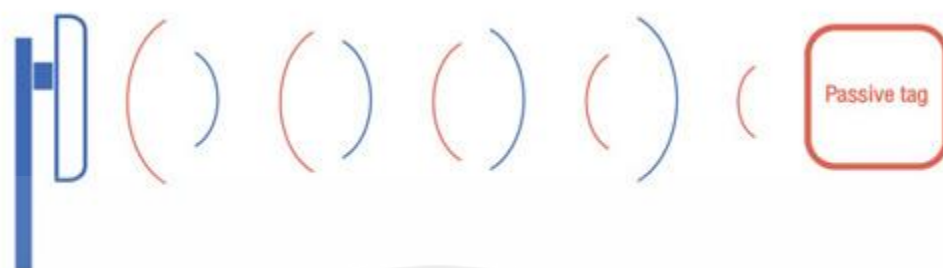
ที่มา <https://www.smartiden.com/17695828/rfid-tag>

2. แท็กแบบ Passive คือแท็กแบบทั่วๆ ไป มักมีลักษณะแบน และมีขนาดเล็ก ภายในจะไม่มีแหล่งพลังงานเป็นของตัวเอง (ไม่มีแบตเตอรี่) ต้องอาศัยพลังงานจากคลื่นวิทยุที่ส่งมาจากเครื่องอ่าน แท็กประเภทนี้มักมีระยะทำงานอยู่ในช่วง 1 cm ขึ้นไปจนถึงประมาณ 1 เมตร (แต่หากใช้เครื่องอ่านที่มีสายอากาศขนาดใหญ่ อาจใช้งานได้ไกลกว่านี้)

ตัวอย่างของแท็กแบบ Passive เช่น บัตรทาปสำหรับเปิดปิดประตู (Proximity Card) บัตรสมาร์ทการ์ดแบบไร้หน้าสัมผัส (Contactless Smart Card) และแท็กของ NFC เป็นต้น

ข้อดี : มีน้ำหนักเบา ขนาดเล็ก มีอายุการใช้งานได้ไม่จำกัด และราคาถูกกว่า tag ประเภท active tag

ข้อเสีย : ระยะการรับส่งข้อมูลใกล้ (ขึ้นอยู่กับ RFID reader ด้วย) มีหน่วยความจำขนาดเล็กประมาณ 32 ถึง 128 บิต



ภาพที่ 2.11 แท็กแบบ Passive

ที่มา <https://www.smartiden.com/17695828/rfid-tag>

2.4.3 การเลือกใช้งาน RFID Tag

การเลือก RFID tag เพื่อนำไปใช้งานนั้นเราต้องคำนึงถึง ระยะการอ่านที่ต้องการ, พื้นผิววัสดุที่ต้องการนำ RFID tag ไปติด และสภาวะแวดล้อม เป็นต้น

1. Label tag เป็น tag ที่มีต้นทุนต่ำสุดสำหรับใช้ติดสินค้าที่พื้นผิวเรียบ และวัสดุที่ไม่เป็นเหล็ก หรือโลหะ มีให้เลือกหลายขนาดตามความต้องการ รวมทั้งยังสามารถพิมพ์ Barcode ได้ลงบน tag ได้
2. Anti Metal Tag เป็น tag ที่ถูกออกแบบเพื่อให้สามารถติดลงบนพื้นผิววัสดุที่เป็นเหล็ก หรือโลหะได้ โดยไม่ถูกรบกวนสัญญาณ เช่น รถเข็นขนสินค้า ชั้น Rack ในคลังสินค้า เครื่องมือช่าง เป็นต้น
3. High Temperature Tag เป็น tag ที่เหมาะสำหรับการนำไปใช้งานในสภาวะที่มีความร้อนสูง หรือสำหรับงานที่ต้องการ tag ที่สามารถทนความร้อนได้เกิน 100 องศาขึ้นไป แต่ tag ประเภทนี้ก็จะมีราคาสูงกว่า tag ทั่วไป
4. Laundry Tag คือ tag ที่สามารถทนต่อการซักล้าง, ทนต่อความร้อนในการรีด ได้และสามารถเย็บติดกับเสื้อผ้าได้ Laundry Tag มักถูกนำมาใช้ในการตรวจนับสินค้า หรือ นับจำนวนครั้งในการซักสินค้า เพราะเสื้อผ้าบางชนิด เช่น ชุด Cleanroom จำเป็นต้องมีการตรวจสอบจำนวนครั้งในการซัก
5. Windshild Tag คือ tag ที่ใช้สำหรับติดกระจกรถยนต์ ซึ่งถูกออกแบบมาเพื่อขยายสัญญาณให้ดีขึ้นเมื่อติดเข้ากับกระจกรถยนต์ และใน Windshild Tag บางรุ่นตัว Chip จะถูกเคลือบด้วยสารกัน UV ทำให้มีความคงทนต่อการใช้งานมากยิ่งขึ้นอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Tool Tag คือ tag ประเภทเดียวกันกับ Anti Metal Tag แต่มีขนาดเล็กเพื่อสำหรับนำไปติดที่เครื่องมือประเภทต่างๆ เช่น เครื่องมือแพทย์ หรือ เครื่องมือของช่างซ่อมบำรุง เพื่อใช้ในการตรวจสอบอุปกรณ์เหล่านี้ว่ายังอยู่ครบหรือไม่

7. Active Tag เป็น tag ที่มีแบตเตอรี่ในตัว และต้องใช้งานกับเครื่องอ่านประเภท RFID Active Reader เท่านั้น การที่ tag มีแบตเตอรี่ในตัวจึงทำให้สามารถอ่านระยะได้ไกลกว่า tag แบบ Passive ส่วนใหญ่มักถูกนำไปใช้งานใน Solution ที่มีพื้นที่กว้าง และต้องการระยะการอ่านที่ไกล เช่น 20 - 30 เมตร

8. Wristband คือ Tag ประเภทหนึ่งที่ถูกออกแบบมาในรูปของ สายรัดข้อมือ เพื่อใช้ในการสวมใส่ สามารถนำไปประยุกต์ใช้งานได้หลากหลาย ในปัจจุบันที่พบเห็นได้ทั่วไป เช่น การใช้งานในระบบ E-payment, การปลดล็อคตู้ล็อคเกอร์, การใช้งานภายในสวนน้ำ และสถานที่ออกกำลังกาย

2.4.4 RFID Reader

เครื่องอ่านป้าย (RFID Reader) มีหน้าที่ในการเชื่อมต่อกับป้าย (RFID Tag) เพื่อทำการอ่านหรือเขียนข้อมูลลงในป้ายโดยใช้สัญญาณวิทยุ ซึ่งภายในเครื่องอ่านจะประกอบด้วย เสาอากาศ เพื่อใช้รับ - ส่งสัญญาณ สัญญาณวิทยุ ภาครับ-ภาคส่ง วงจรควบคุม การอ่าน-เขียนข้อมูล และส่วนที่ติดต่อกับคอมพิวเตอร์ RFID Reader นั้นมีอยู่หลายประเภทแตกต่างกันไป เช่น เรื่องของขนาดเครื่องอ่าน RFID เรื่องของคลื่นความถี่ (LF, HF, UHF) เรื่องของเสาอากาศสัญญาณ (Antenna) เป็นต้น ดังนั้นเราควรเลือก RFID Reader ให้ตรงกับจุดประสงค์กับการใช้งานเพื่อให้ RFID Reader หรือเครื่องอ่าน RFID ทำงานให้มีประสิทธิภาพสูงสุด

2.5 ฐานข้อมูลแบบคลาวด์ (Cloud Database)

ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่บังคับว่าข้อมูลทั้งหมดจะต้องถูกเก็บไว้ในที่เดียวกันหรือแยกเก็บหลายๆแฟ้มข้อมูล ซึ่งข้อมูลจะถูกจัดเก็บอย่างเป็นระเบียบโดยจะมีซอฟต์แวร์เข้ามาควบคุมกระบวนการใช้งาน การทำงาน หรือการประมวลผล ทำให้ผู้ใช้บริการสามารถเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ

ฐานข้อมูลแบบคลาวด์ คือ ระบบของฐานข้อมูลอีกรูปแบบที่สามารถทำงานได้แบบออนไลน์ สามารถที่จะทำงานเชื่อมโยงกับ Web Application ได้อย่างสะดวกมากขึ้น หรืออาจจะกล่าวได้อีกอย่าง

ว่า คือ ระบบฐานข้อมูลที่ทำงานอยู่บนเทคโนโลยี Cloud ทำให้ระบบฐานข้อมูลมีความยืดหยุ่น สามารถรับส่งข้อมูลได้อย่าง Realtime อัปเดตข้อมูลได้แบบทันที และยังสามารถปรับแต่ง เพิ่มขนาดหรือขยายฐานข้อมูลและทำการแก้ไขได้อย่างอิสระ

2.6 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล (Database System) หมายถึง การรวมตัวกันของฐานข้อมูลตั้งแต่ 2 ฐานข้อมูลเป็นต้นไปที่มีความสัมพันธ์กัน โดยมีวัตถุประสงค์เพื่อเป็นการลดความซ้ำซ้อนของข้อมูล และทำให้การบำรุงรักษาตัวโปรแกรมง่ายมากขึ้น โดยผ่านระบบการจัดการฐานข้อมูล หรือ เรียกว่า DBMS

ระบบจัดการฐานข้อมูล (Database Management System หรือ DBMS) หมายถึง กลุ่มโปรแกรมหรือซอฟต์แวร์ชนิดหนึ่ง ที่สร้างขึ้นมาเพื่อทำหน้าที่บริหารฐานข้อมูลโดยตรง เป็นเครื่องมือที่ช่วยอำนวยความสะดวกให้ผู้ใช้สามารถเข้าถึงข้อมูลได้ โดยที่ผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างฐานข้อมูล พุดง่าย ๆ ก็คือ DBMS นี้ เป็นตัวกลางในการเชื่อมโยงระหว่างผู้ใช้ และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับระบบฐานข้อมูลเข้าด้วยกัน ตัวอย่างของ DBMS ที่นิยมใช้ในปัจจุบัน ได้แก่ Microsoft Access, FoxPro, SQL Server, Oracle, Informix, DB2 เป็นต้น

2.7 Firebase

Firebase คือ Platform ที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วนของ Backend หรือ Server side ซึ่งทำให้สามารถ Build Mobile Application ได้อย่างมีประสิทธิภาพ และยังคงเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูลให้อีกด้วย โดยมีทั้งเครื่องมือที่ฟรี และเครื่องมือที่มีค่าใช้จ่าย (สำหรับการ Scale) บริการต่างๆ ที่ Firebase ให้บริการมี ได้แก่

2.7.1 Cloud Firestore จัดเก็บและซิงค์ข้อมูลระหว่างผู้ใช้และอุปกรณ์ในระดับโลกโดยใช้ฐานข้อมูล NoSQL ที่โฮสต์บนคลาวด์ Cloud Firestore ให้การซิงโครไนซ์แบบออนไลน์และการสนับสนุนออฟไลน์พร้อมกับการสืบค้นข้อมูลที่มีประสิทธิภาพ การผสมรวมกับผลิตภัณฑ์ Firebase อื่น ๆ ช่วยให้สร้างแอปแบบไร้เซิร์ฟเวอร์ได้อย่างแท้จริง

2.7.2 Authentication จัดการผู้ใช้งานด้วยวิธีที่ง่ายและปลอดภัย Firebase Auth มีหลายวิธีในการตรวจสอบสิทธิ์รวมถึงอีเมลและรหัสผ่านผู้ให้บริการบุคคลที่สามเช่น Google หรือ Facebook และใช้ระบบบัญชีที่มีอยู่โดยตรง สร้างอินเทอร์เฟซเองหรือใช้ประโยชน์จากโอเพ่นซอร์ส UI ที่ปรับแต่งได้อย่างเต็มที่

2.7.3 Hosting ลดความซับซ้อนของเว็บไซต์ด้วยเครื่องมือที่สร้างขึ้นเฉพาะสำหรับเว็บแอปสมัยใหม่ เมื่ออัปโหลดเนื้อหาเว็บ จะส่งเนื้อหาเหล่านั้นไปยัง CDN ทั่วโลกโดยอัตโนมัติและมอบใบรับรอง SSL ฟรีเพื่อให้ผู้ใช้ได้รับประสบการณ์ที่ปลอดภัยเชื่อถือได้และมีเวลาแฝงต่ำไม่ว่าจะอยู่ที่ใดก็ตาม

2.7.4 Realtime Database – Realtime Database คือฐานข้อมูลดั้งเดิมของ Firebase เป็นโซลูชันที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับแอปบนอุปกรณ์เคลื่อนที่ที่ต้องการสถานะการซิงค์ระหว่างไคลเอนต์แบบเรียลไทม์

บทที่ 3

หลักการงานและการออกแบบ

3.1 ลักษณะการทำงาน

ต่อบอร์ดจ่ายไฟเข้ากับตัวไมโครคอนโทรลเลอร์ ESP8266 โดยจ่ายแรงดันกระแสไฟตรงขนาด 5 โวลต์ เพื่อให้ตัวบอร์ดทำงาน โดยส่งข้อมูลในรูปแบบการสื่อสารอนุกรม Serial และจ่ายแรงดันกระแสไฟตรงขนาด 12 โวลต์ ให้กับ Nema 17 stepper motor เมื่อวัตถุ (ภาต) เคลื่อนที่บนสายพาน และเจอ IR Sensor จะทำการตรวจจับวัตถุนั้น และส่งข้อมูลให้ ESP8266 เพื่อควบคุม Drive Motor ให้สายพานหยุดหมุนเป็นเวลา 5 วินาที จากนั้นในช่วงเวลาที่มอเตอร์หยุดหมุน ESP8266 จะสั่งการทำงานให้ Ultrasonic sensor อ่านระยะความสูงของภาตที่อยู่บนสายพาน และแสดงผลเป็นจำนวนภาตที่นับได้ บน จอ LCD จำนวนภาตที่แสดงขึ้นเกิดจากการคำนวณจากระยะความสูงที่เซนเซอร์อ่านค่าได้หารด้วย ระยะความสูงของภาต 1 ชิ้น และจากนั้นภาตที่นับจำนวนได้จะถูกนำไปตรวจสอบสิ่งของที่อยูข้างในภาต

ส่วนต่อมาก็คือ เริ่มต้นจ่ายไฟเข้าสู่บอร์ด RFID-RC522 จะทำให้ RFID Reader ทำงาน จากนั้นตัว RFID Reader จะทำการตรวจจับการมอดูเลตสัญญาณ หากมี RFID tag บริเวณ 0-1 เซนติเมตร RFID Reader จะทำการอ่านค่าของสัญญาณนั้น ในโครงการนี้ได้กำหนด RFID tag สำหรับการอ่านข้อมูลไว้ 7 ประเภท คือ Hammer, Screwdriver, Wrench, Electrical Tape, Pliers, Steel scale และ Drill ซึ่งข้อมูลที่ RFID Reader อ่านได้จะแสดงผลบนโปรแกรม Arduino IDE และมีการติดตั้ง library NTPClient.h เพื่อใช้ในการอ่านค่าวันและเวลาในขณะที่อ่านข้อมูล RFID tag โดยใช้เวลาประเทศไทย จากนั้นส่งข้อมูลจาก ESP8266 ไปยัง real-time firebase database

นอกจากนี้ยังมีการกำหนดข้อมูลอื่นๆ เช่น สถานะของระบบ (Available/Unavailable), จำนวนภาตต่อวัน และจำนวนภาตต่อสัปดาห์ ผ่านโปรแกรม Arduino IDE การส่งข้อมูลไปจะเป็นแบบ real-time firebase database ซึ่งเป็นการสร้างฐานข้อมูลเพื่อนำไปทำ dashboard โดยเป็นฐานข้อมูลชนิด NoSQL และเก็บข้อมูลเป็นชนิด JSON จากนั้นข้อมูลที่อยู่ภายใน real-time firebase database จะส่งต่อไปยัง dashboard สำหรับการแสดงข้อมูลแบบ real-time และ Google sheet เพื่อเก็บข้อมูลลงฐานข้อมูลและสามารถอ่านข้อมูลย้อนหลังได้

โดย dashboard จะประกอบไปด้วยข้อมูลที่ส่งจาก ESP8266 รวมถึงกราฟแผนภูมิแท่ง และแผนภูมิวงกลม ที่มีการอัปเดตข้อมูลทุกๆ 10 นาที ตามที่กำหนดใน trigger ของ Google sheet

3.2 การออกแบบทางด้าน Hardware

3.2.1 การเลือกใช้อุปกรณ์ในการดำเนินสร้างชิ้นงาน

3.2.1.1 การคำนวณพิกัดหม้อแปลง

ทำการคำนวณพิกัดหม้อแปลงโดย

$$V_{AC(rms)} = \frac{V_{O(DC)} + \frac{1}{2}\Delta V + V_f}{\sqrt{2}}$$

$$= \frac{19.3 + \frac{1}{2}(1) + 1.4}{\sqrt{2}}$$

$$V_{AC(rms)} = 14.99 \approx 15$$

เมื่อกำหนดให้ $V_{O(DC)}$ คือ ค่าแรงดันเอาต์พุตไฟตรง หน่วยเป็น โวลต์

ΔV คือ ค่าแรงดันกระเพื่อม หน่วยเป็น V_{pp} โดยกำหนด ให้ $\Delta V = 1.0 V_{pp}$

V_f คือ ค่าแรงดันตกคร่อมไดโอดที่ทำงานในครึ่งรอบการจ่ายไฟกระแสสลับในที่นี้เป็น 1.4V (Maximum ต่อไดโอด 1 ตัว)

พิกัดกระแสของหม้อแปลงที่ใช้มีลักษณะการต่อเป็น Full wave Bridge จะได้

$$I_{AC} = 1.8I_O = 1.8(1) = 1.8 A \approx 2 A_{rms}$$

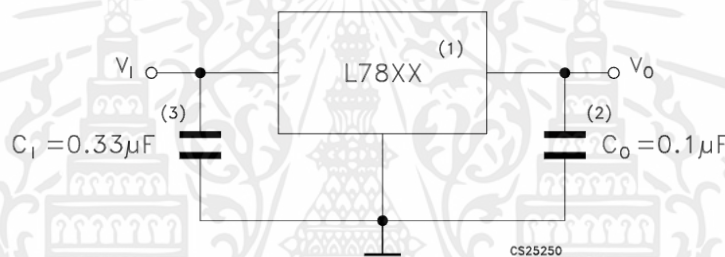
เพราะฉะนั้นจึงทำการใช้หม้อแปลงที่มีพิกัดแรงดัน 15V และมีพิกัดกระแส 2A เป็นหม้อแปลงปรับแรงดันจากไฟบ้าน 220 V_{rms} ให้มีค่าต่ำลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.2 วงจรซีพพลายที่มีรักษาแรงดัน (Voltage regulator)

เลือกใช้ IC LM7805, LM7812

จาก datasheet ของ LM78XX กล่าวว่าควรมีตัวเก็บประจุบายพาสอินพุท เพื่อให้มีการทำงานที่เสถียร โดยใช้ค่าตัวเก็บประจุขนาด $0.33 \mu\text{F}$ ขึ้นไป เป็นประเภทแทนทาลัม ไมลาร์ หรือตัวเก็บประจุอื่นๆ ที่มีค่าอิมพีแดนซ์ภายในต่ำที่ความถี่สูง นอกจากนี้ควรติดตั้งตัวเก็บประจุบายพาสให้มีระยะทางห่างจาก regulators input terminals สั้นที่สุดเท่าที่จะเป็นไปได้

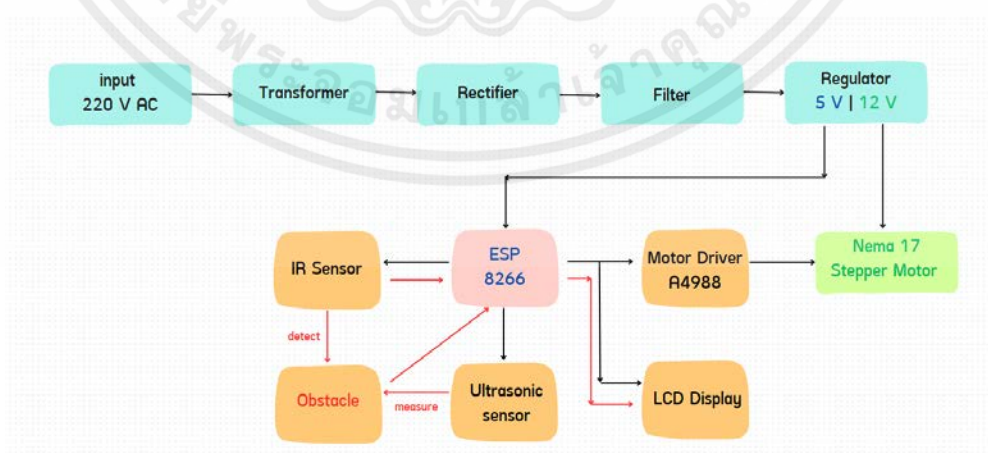


ภาพที่ 3.2.1 วงจรซีพพลายที่มีรักษาแรงดัน

ที่มา <https://www.microkitschile.com/product-page/lm7924>

3.2.2 การออกแบบ

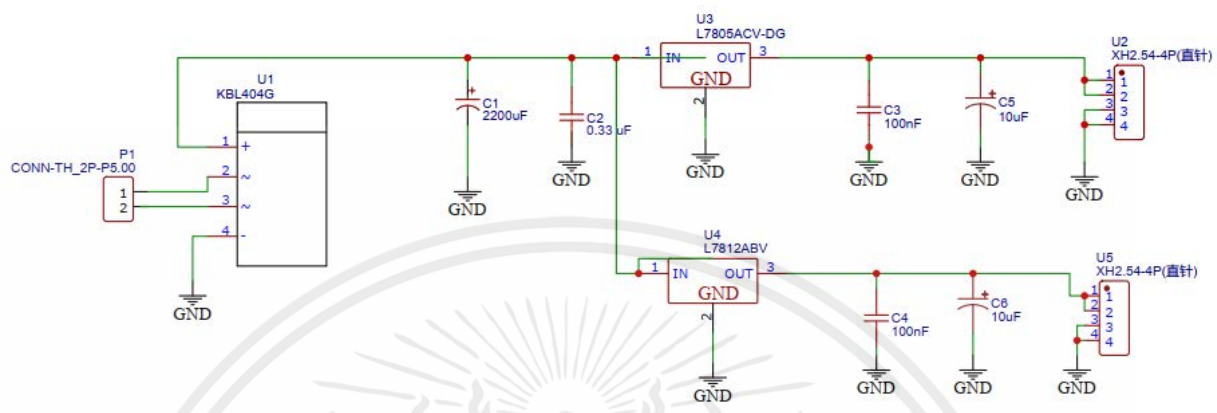
Block Diagram



ภาพที่ 3.2.2 (ก) Block Diagram

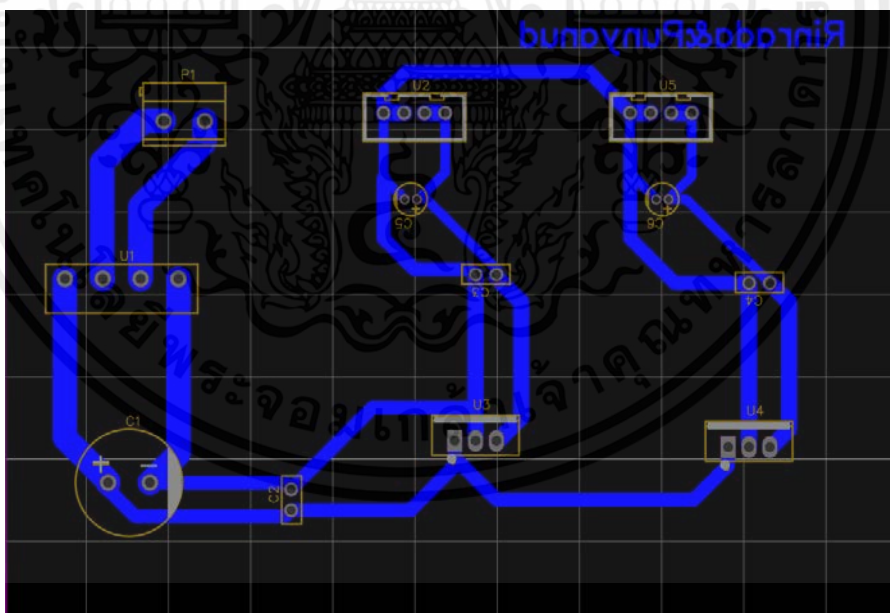
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

schematic diagram



ภาพที่ 3.2.2 (ข) schematic diagram

PCB ที่ออกแบบได้



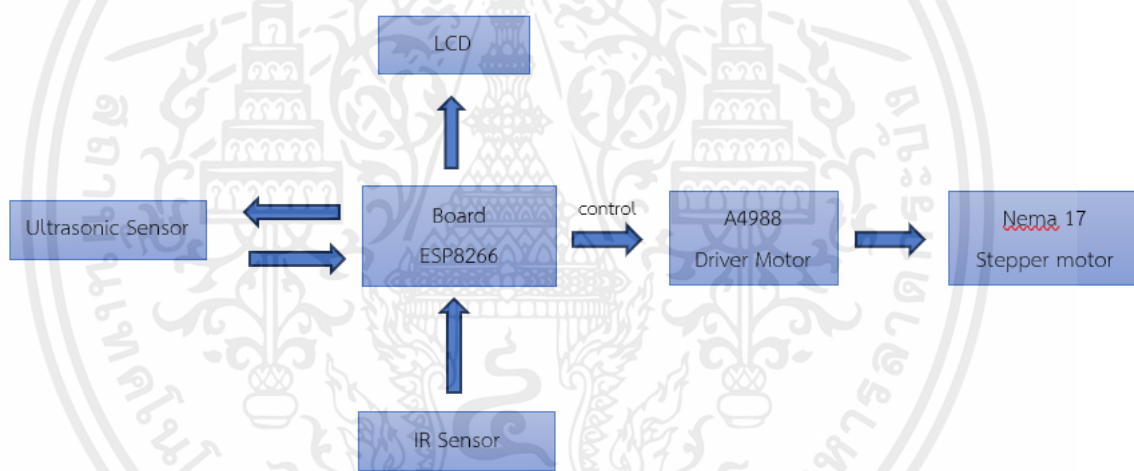
ภาพที่ 3.2.2 (ค) PCB ที่ออกแบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

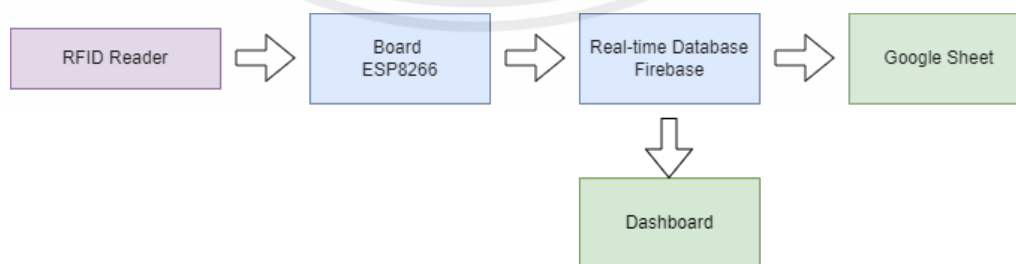
3.3 การออกแบบทางด้าน Software

ในการออกแบบด้าน Software นี้เป็นการออกแบบโปรแกรมของวงจร โดยประกอบด้วย 4 ส่วน คือ ส่วนแรกจะเป็นการเขียนโปรแกรมของวงจรด้วยโปรแกรม Arduino IDE โดยจะทำการออกแบบแล้ว จะถูก upload ใส่ ESP8266 เพื่อใช้ในการควบคุมอุปกรณ์เซนเซอร์ต่างๆ ต่อจากนั้นจะเป็นการออกแบบ โปรแกรมในการส่งข้อมูลจาก ESP8266 ไปยัง real-time database firebase โดยใช้โปรแกรม Arduino IDE, การออกแบบโปรแกรมในการส่งข้อมูลจาก real-time database firebase ไปยัง google sheet และการออกแบบในการส่งข้อมูลจาก real-time database firebase ไปยัง dashboard

3.3.1 รูปแบบการทำงาน



ภาพที่ 3.3.1 (ก) รูปแบบการทำงาน Software

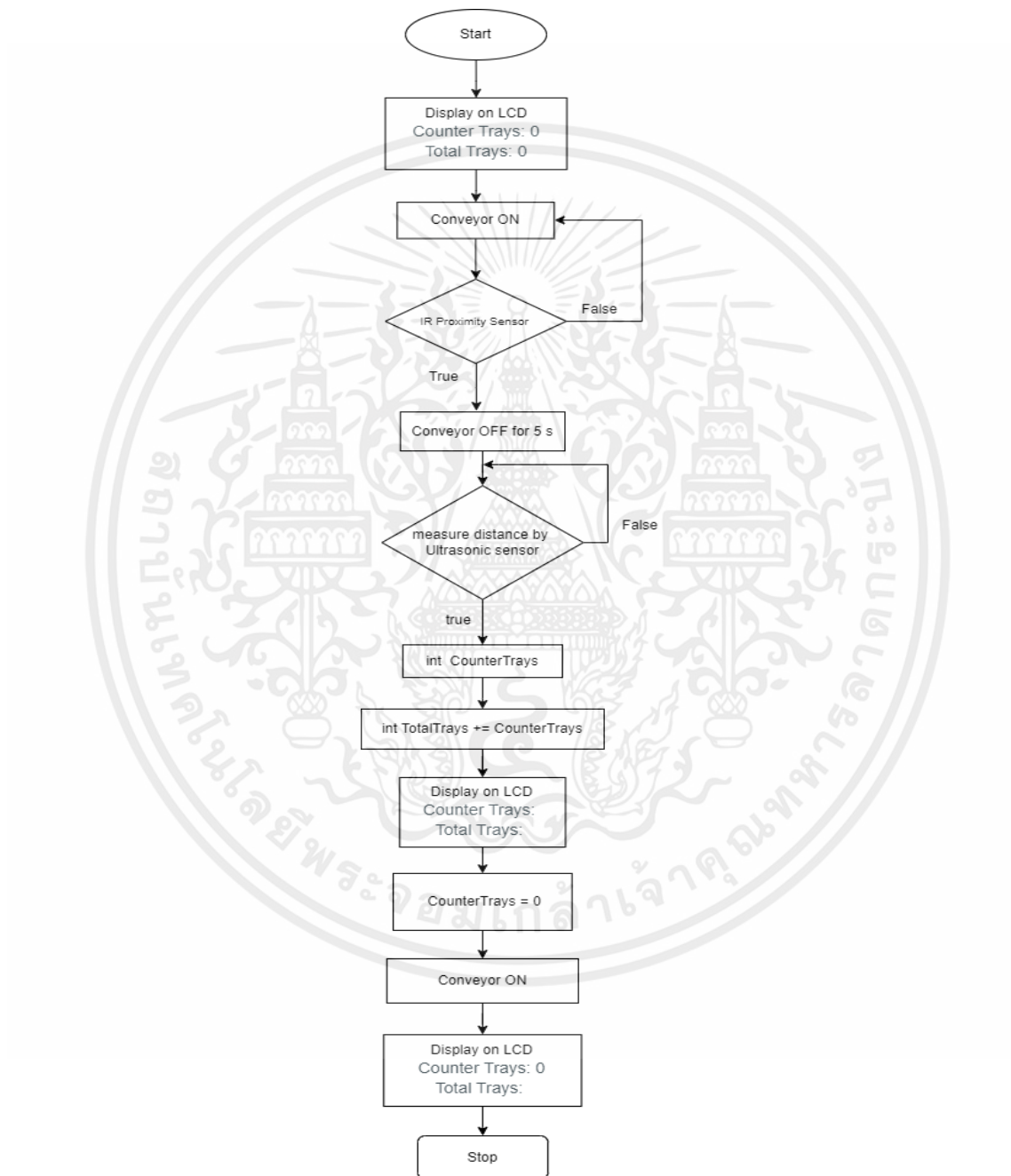


ภาพที่ 3.3.1 (ข) รูปแบบการทำงานของ Software ในส่วนของระบบ RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเขียนโปรแกรมควบคุมวงจรและแสดงผลบนเว็บไซต์

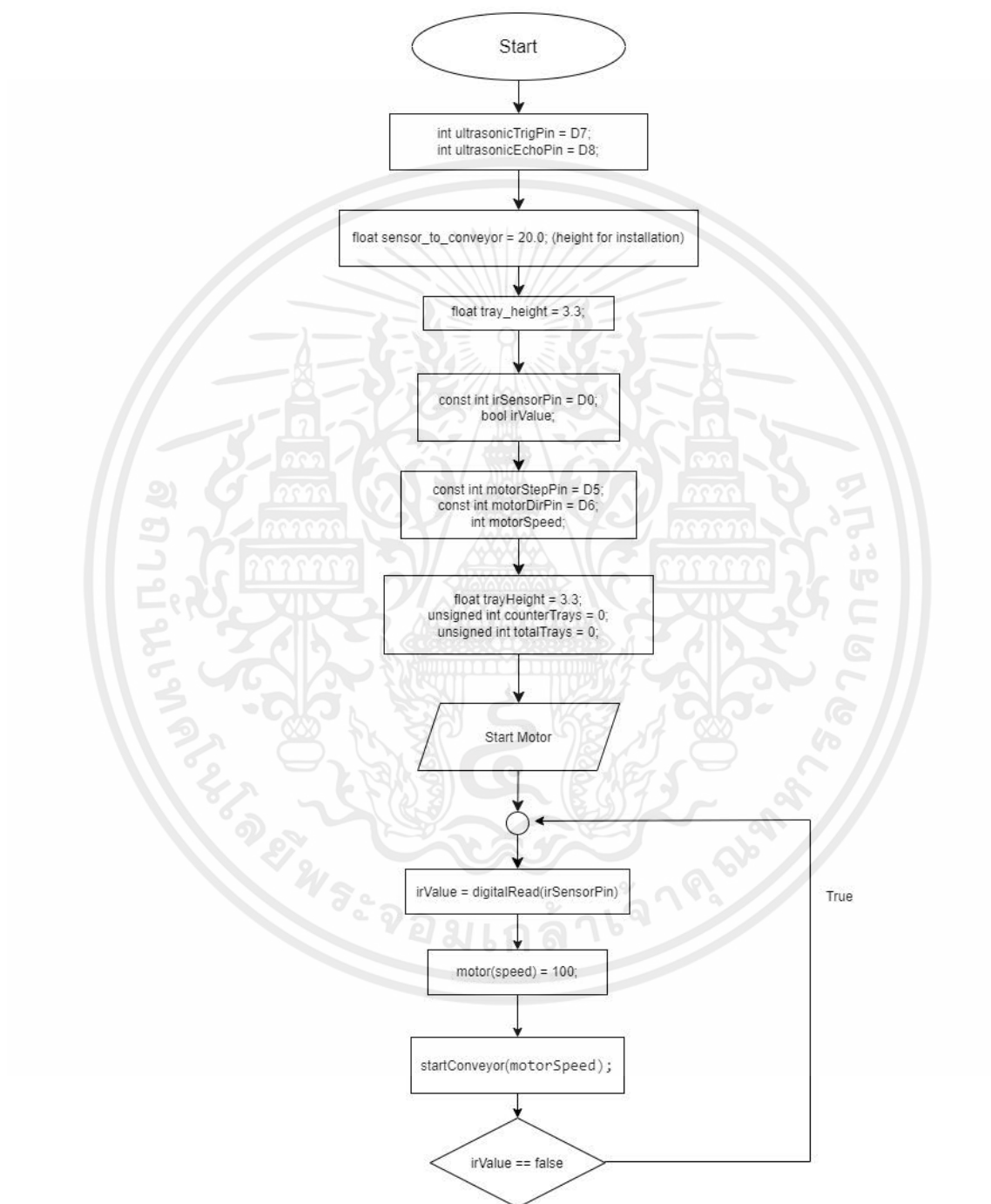
3.4.1 แผนภาพแสดงขั้นตอนการทำงานของระบบตรวจนับจำนวนถาด (Flowchart)



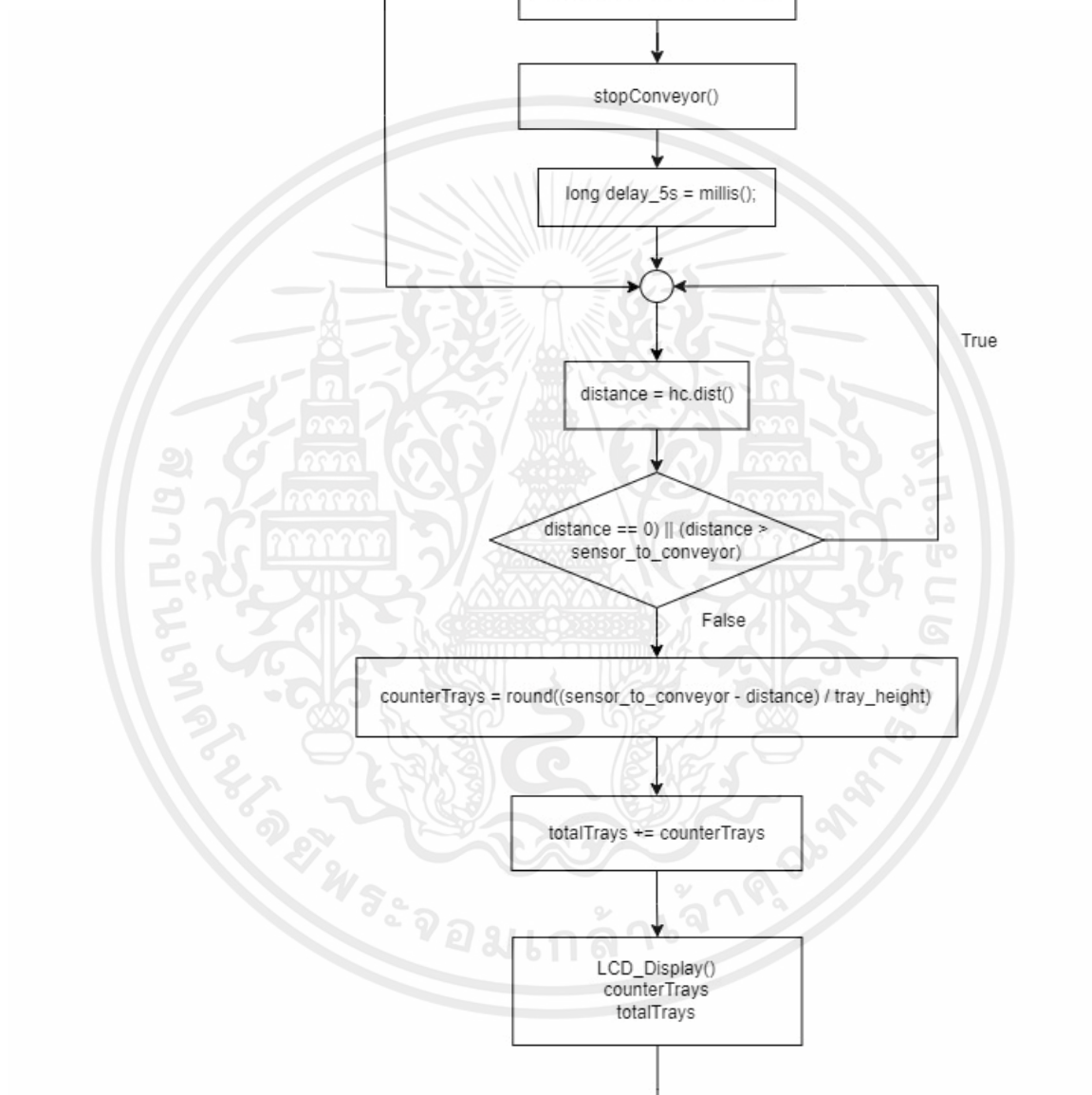
ภาพที่ 3.4.1 Flowchart แสดงขั้นตอนการทำงานของระบบตรวจนับจำนวนถาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 แผนภาพแสดงขั้นตอนการทำงานของ NodeMCU (Flowchart)



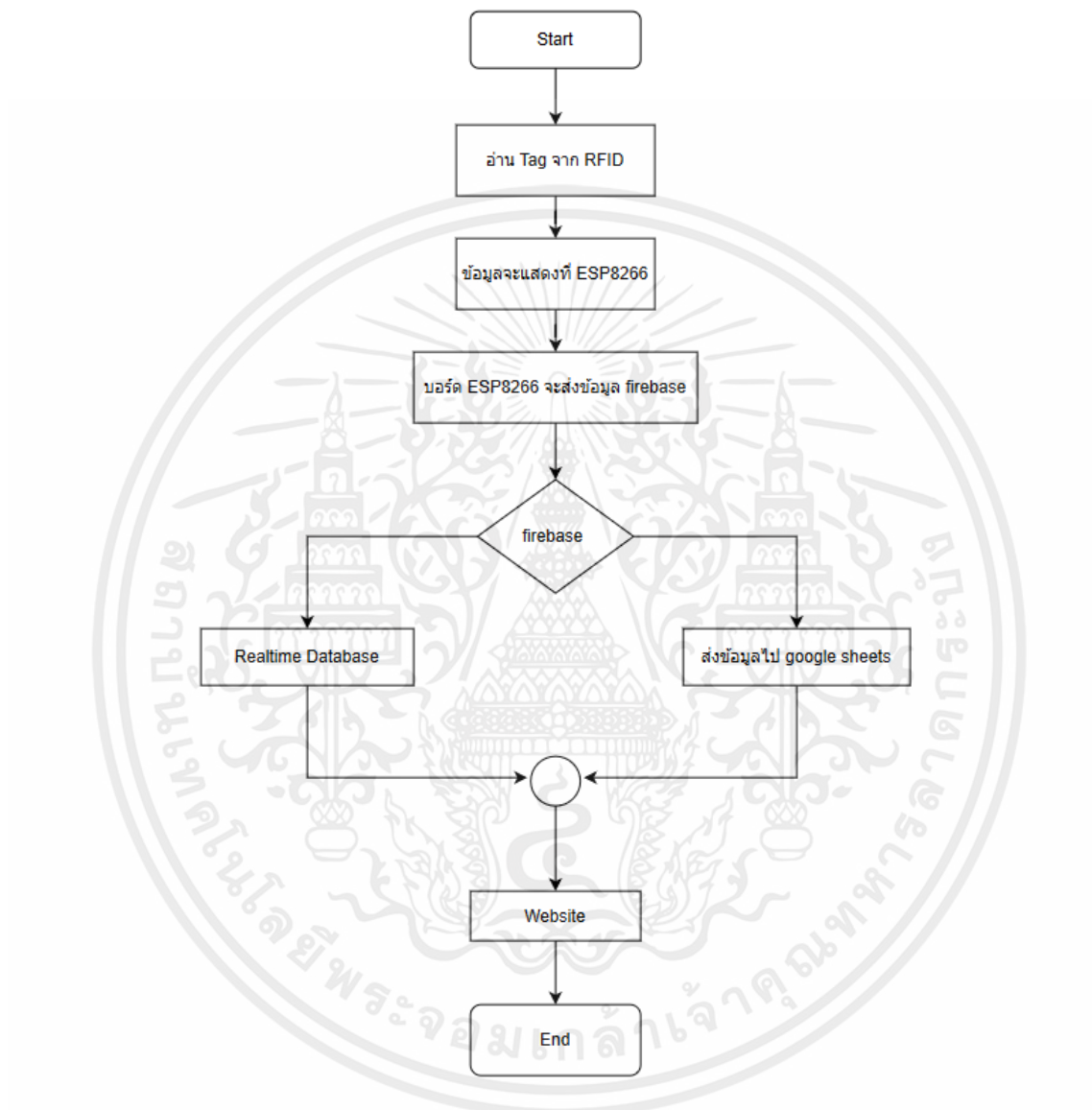
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4.2 Flowchart แสดงขั้นตอนการทำงานของ NodeMCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

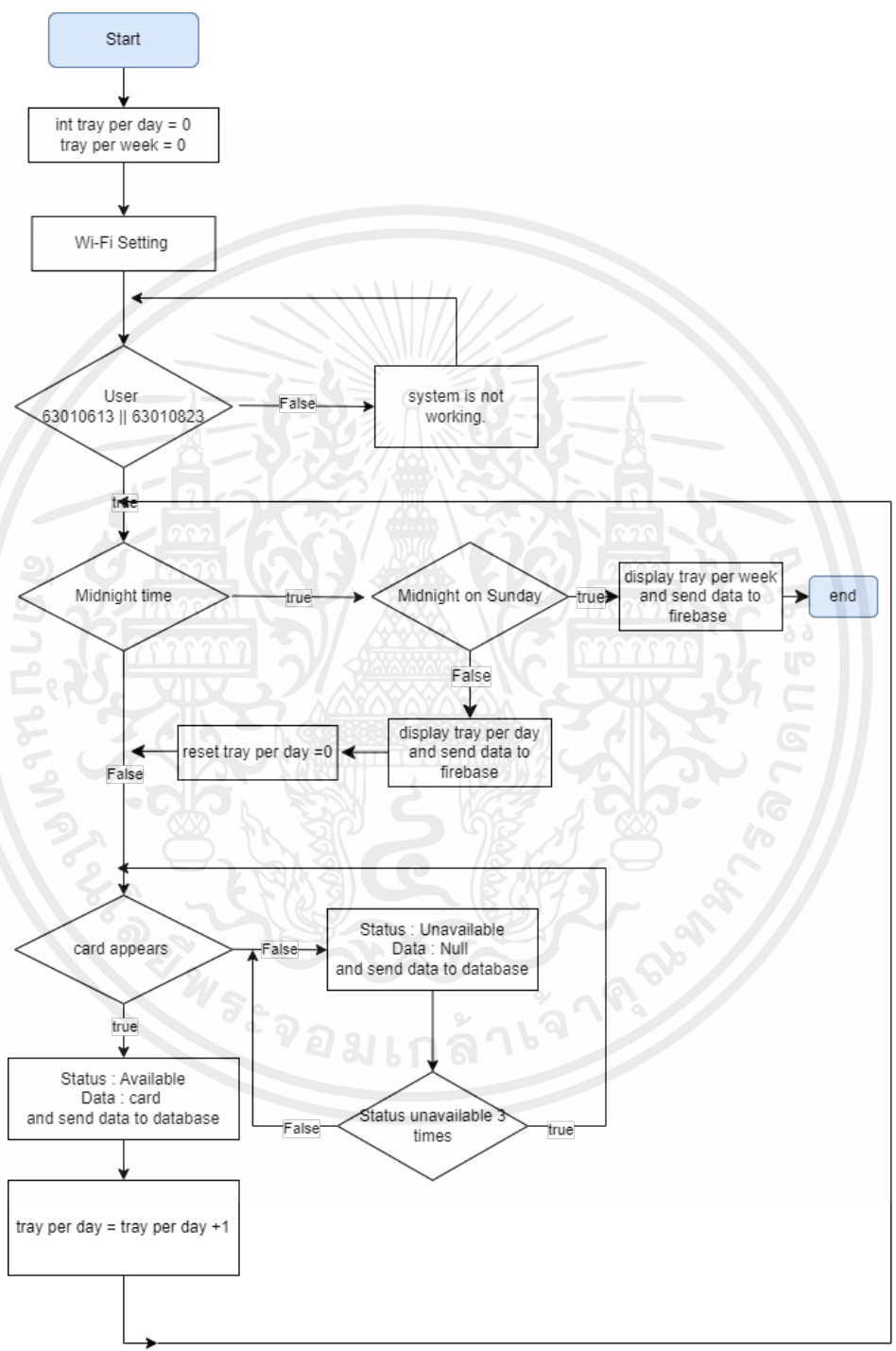
3.4.3 แผนภาพแสดงขั้นตอนการทำงานของระบบตรวจเก็บข้อมูลของภาค (Flowchart)



ภาพที่ 3.4.3 Flowchart แสดงขั้นตอนการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

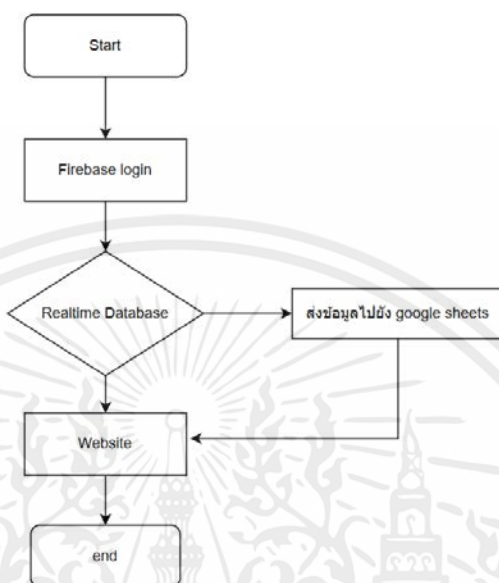
3.4.4 แผนภาพแสดงขั้นตอนการทำงานของ การส่งข้อมูลจาก ESP8266 ไปยัง firebase



ภาพที่ 3.4.4 Flowchart แสดงขั้นตอนการส่งข้อมูลจากการอ่าน RFID tag ไปยัง firebase

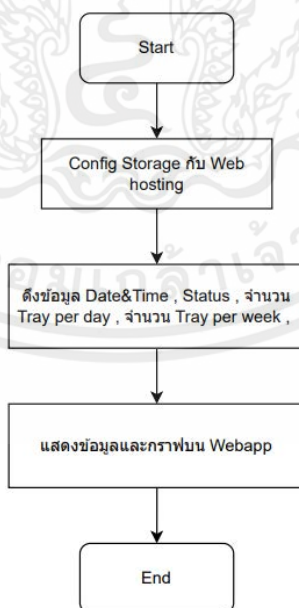
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 แผนภาพแสดงขั้นตอนการทำงานการส่งข้อมูลของ firebase ไปยังเว็บไซต์



ภาพที่ 3.4.5 Flowchart แสดงขั้นตอนการส่งข้อมูลของ firebase ไปยังเว็บไซต์

3.4.6 แผนภาพแสดงการทำงานของ Webapp



ภาพที่ 3.4.6 Flowchart แสดงขั้นตอนการทำงานและการแสดงผลของ Webapp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ขั้นตอนการตั้งค่า

3.5.1 การตั้งค่า Firebase Realtime database

- หน้า Rules ให้ทำการตั้งค่าดังนี้



ภาพที่ 3.5.1 การตั้งค่าหน้า user

3.5.2 การตั้งค่า Firebase Storage

- หน้า Rules ให้ทำการตั้งค่าดังนี้



ภาพที่ 3.5.2 การตั้งค่าหน้า Rules

3.5.3 การตั้งค่า Firebase Hosting

- ทำการติดตั้งโปรแกรม Visual Studio Code และ Node.js (เวอร์ชัน LTS) หลังจากนั้น

ติดตั้งส่วนขยาย node.js extension pack ลง VS code

- ติดตั้ง Firebase tool โดยใช้คำสั่ง `npm -g install firebase-tools`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Set up Firebase Hosting

1 Install Firebase CLI

To host your site with Firebase Hosting, you need the Firebase CLI (a command line tool).

Run the following [npm](#) command to install the CLI or update to the latest CLI version.

```
$ npm install -g firebase-tools
```

Doesn't work? Take a look at the [Firebase CLI reference](#) or change your [npm permissions](#)

Also show me the steps to add the Firebase JavaScript SDK to my web app

The SDK includes Cloud Firestore, Authentication, Performance Monitoring and more. It can be added now or later.

Next

ภาพที่ 3.6 คำสั่งการติดตั้ง firebase tools

```
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>
```

ภาพที่ 3.7 การติดตั้ง firebase tools

- หลังจากนั้น ใช้คำสั่ง firebase login

```
PS C:\Users\66922\Downloads\test> firebase login
Already logged in as rfidproject2024@gmail.com
PS C:\Users\66922\Downloads\test>
```

ภาพที่ 3.8 login firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากนั้น ใช้คำสั่ง firebase init

```

PS C:\Users\66922\Downloads\test> firebase login
Already logged in as rfidproject2024@gmail.com
PS C:\Users\66922\Downloads\test> firebase init

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

```

You're about to initialize a Firebase project in this directory:

ภาพที่ 3.9 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

```

? 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)

```

ภาพที่ 3.10 ตัวเลือกฟีเจอร์ต่างๆ

- 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

```

ภาพที่ 3.11 เลือกโปรเจกต์ที่เราต้องการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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!
  
```

ภาพที่ 3.12 Hosting Setup

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

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/1
  // 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: "AIzaSyCXY45ZAvHkcQZ9rR3mHQyQah0k96ChB8",
    authDomain: "rfdttest-36d7d.firebaseio.com",
    databaseURL: "https://rfdttest-36d7d-default-rtdb.asia-southeast
    projectId: "rfdttest-36d7d",
    storageBucket: "rfdttest-36d7d.appspot.com",
    messagingSenderId: "629535582652",
    appId: "1:629535582652:web:a3a1b37194322bbf88e3b9",
    measurementId: "G-SQD0FG7EZ7"
  };

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

ภาพที่ 3.13 SDK setup and configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ทดสอบความถูกต้องของ Ultrasonic Sensor ขณะมีวัตถุ

การทดสอบความถูกต้องของการอ่านค่า Ultrasonic Sensor ขณะมีวัตถุ แบ่งเป็นการทดสอบทั้งหมด 5 ค่า คือ การวางถาดบนสายพานจำนวน 1 ถาด, 2 ถาด, 3 ถาด, 4 ถาด และ 5 ถาด จากนั้นทำการทดสอบแต่ละค่าทั้งหมด 10 ครั้ง เพื่อหาเปอร์เซ็นต์ความคลาดเคลื่อนจากการวัด

4.1.1 วางถาดบนสายพานจำนวน 1 ถาด

ตารางที่ 4.1.1 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 1 ถาด

ครั้งที่	จำนวนถาดที่อ่าน	จำนวนถาดจริง
1	1	1
2	1	1
3	1	1
4	1	1
5	2	1
6	1	1
7	2	1
8	1	1
9	1	1
10	1	1
เปอร์เซ็นต์ความคลาดเคลื่อน		10%

4.1.2 วางถาดบนสายพานจำนวน 2 ถาด

ตารางที่ 4.1.2 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 2 ถาด

ครั้งที่	จำนวนถาดที่อ่าน	จำนวนถาดจริง
1	2	2
2	2	2
3	2	2
4	2	2
5	2	2
6	2	2
7	2	2
8	2	2
9	2	2
10	2	2
เปอร์เซ็นต์ความคลาดเคลื่อน		0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 วางลาดบนสายพานจำนวน 3 ถาด

ตารางที่ 4.1.3 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางลาดบนสายพานจำนวน 3 ถาด

ครั้งที่	จำนวนถาดที่อ่าน	จำนวนถาดจริง
1	3	3
2	3	3
3	3	3
4	3	3
5	3	3
6	3	3
7	3	3
8	3	3
9	3	3
10	3	3
เปอร์เซ็นต์ความคลาดเคลื่อน		0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 วางถาดบนสายพานจำนวน 4 ถาด

ตารางที่ 4.1.4 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 4 ถาด

ครั้งที่	จำนวนถาดที่อ่าน	จำนวนถาดจริง
1	4	4
2	4	4
3	4	4
4	4	4
5	4	4
6	4	4
7	4	4
8	4	4
9	4	4
10	4	4
เปอร์เซ็นต์ความคลาดเคลื่อน		0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

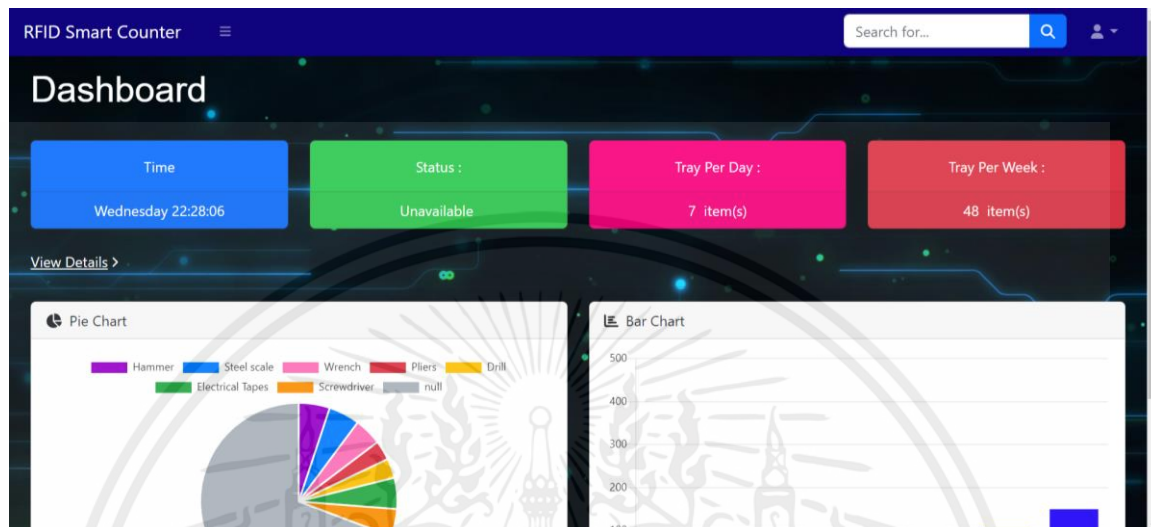
4.1.5 วางถาดบนสายพานจำนวน 5 ถาด

ตารางที่ 4.1.5 แสดงเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางถาดบนสายพานจำนวน 5 ถาด

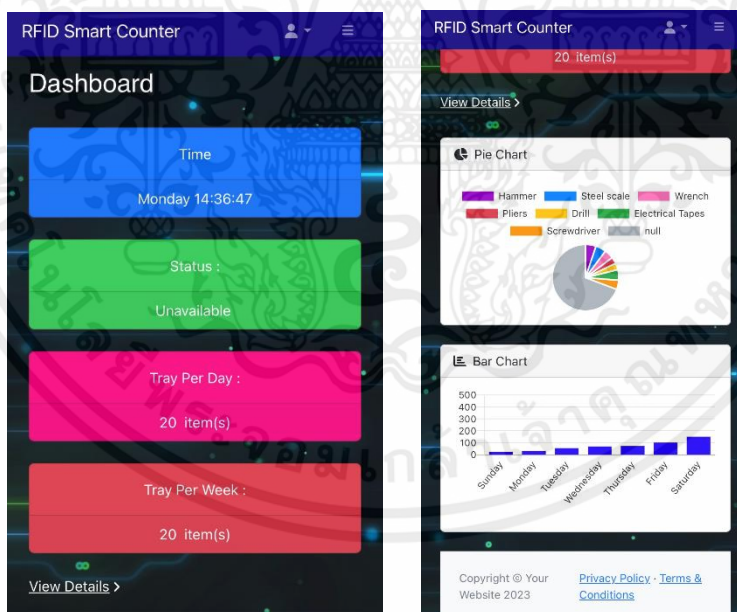
ครั้งที่	จำนวนถาดที่อ่าน	จำนวนถาดจริง
1	5	5
2	5	5
3	5	5
4	5	5
5	5	5
6	5	5
7	5	5
8	5	5
9	5	5
10	5	5
เปอร์เซ็นต์ความคลาดเคลื่อน		0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ทดสอบการแสดงผลของ Dashboard



ภาพที่ 4.2.1 การแสดงผลของ Dashboard ผ่าน Laptop



ภาพที่ 4.2.2 การแสดงผลของ Dashboard ผ่าน Smart Phone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดสอบเวลาที่ใช้ในการอ่านค่าจาก ESP8266 ไปยัง real-time firebase database

ตารางที่ 4.3 แสดงเวลาที่ใช้ในการอ่านค่าจาก ESP8266 ไปยัง real-time firebase database

RFID Tag ชั้นที่	เวลาที่ใช้ในการแสดงผล
1	2.87 วินาที
2	2.45 วินาที
3	2.01 วินาที
4	2.53 วินาที
5	4.53 วินาที
6	2.23 วินาที
7	2.45 วินาที
8	2.47 วินาที
9	2.47 วินาที
10	2.01 วินาที
เวลาเฉลี่ยที่ใช้ในการแสดงผล 2.602 วินาที	

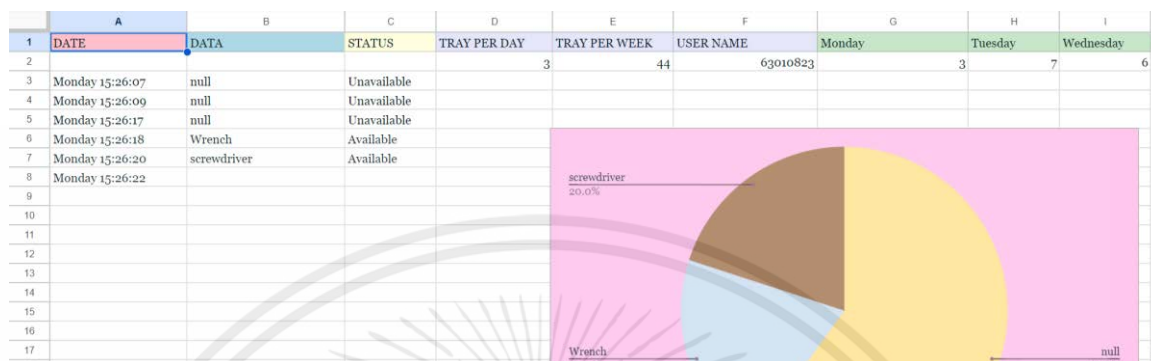
4.4 ทดสอบเวลาที่ใช้ในการอ่านค่าจาก real-time firebase database ไปยัง Dashboard

ตารางที่ 4.4 แสดงเวลาที่ใช้ในการอ่านค่าจาก real-time firebase database ไปยัง Dashboard

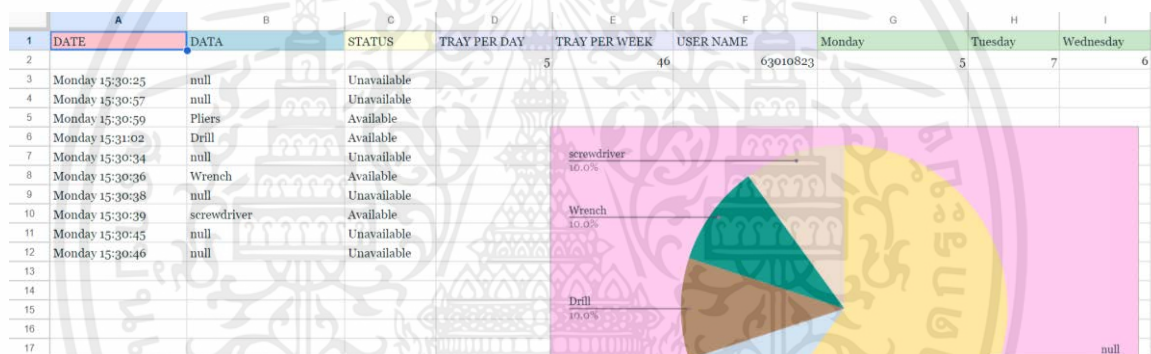
RFID Tag ชั้นที่	เวลาที่ใช้ในการแสดงผล
1	0.24 วินาที
2	0.25 วินาที
3	0.25 วินาที
4	0.26 วินาที
5	0.24 วินาที
6	0.25 วินาที
7	0.31 วินาที
8	0.24 วินาที
9	0.24 วินาที
10	0.24 วินาที
เวลาเฉลี่ยที่ใช้ในการแสดงผล 0.252 วินาที	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

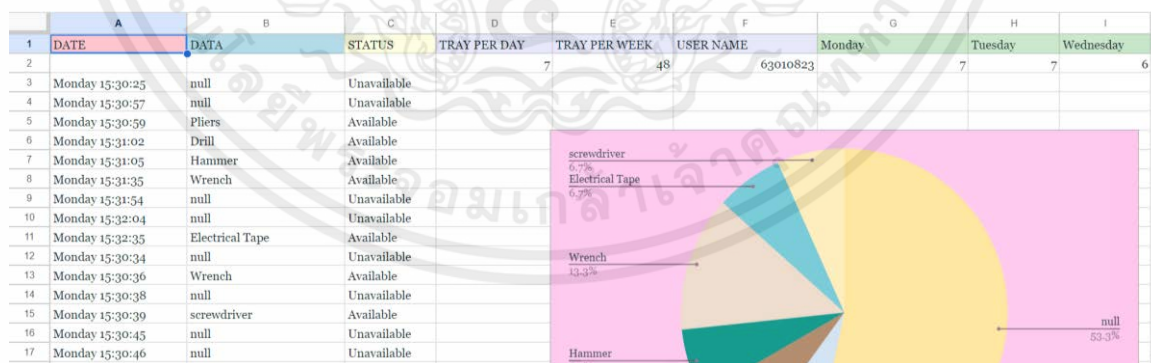
4.5 สามารถแสดงผลบน Google sheet ได้



ภาพที่ 4.5.1 การแสดงผลบน Google sheet เมื่อมี 5 ข้อมูล

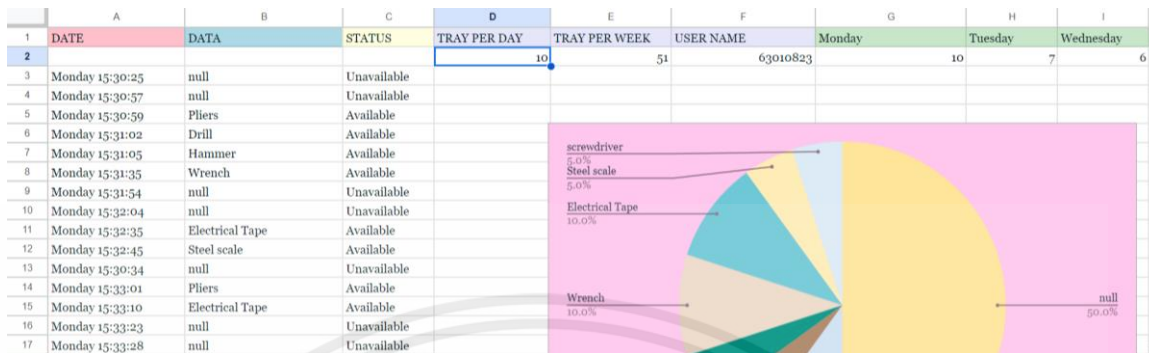


ภาพที่ 4.5.2 การแสดงผลบน Google sheet เมื่อมี 10 ข้อมูล

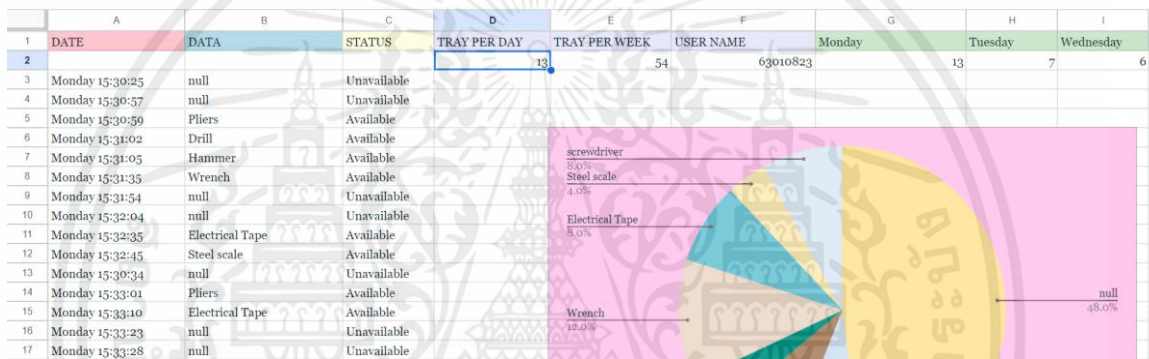


ภาพที่ 4.5.3 การแสดงผลบน Google sheet เมื่อมี 15 ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



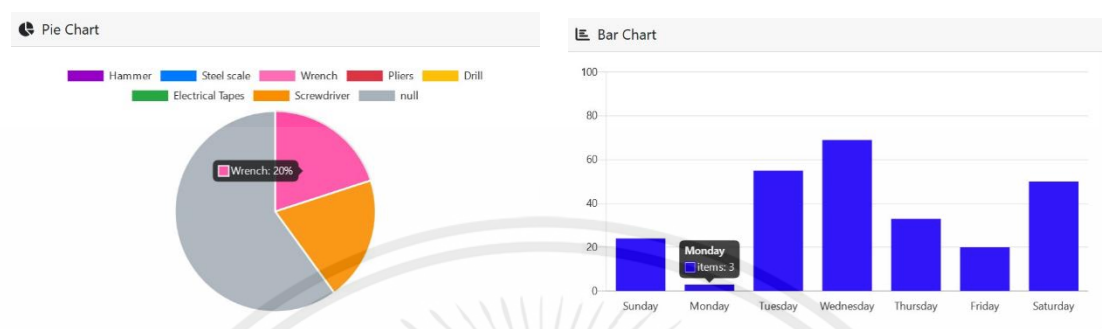
ภาพที่ 4.5.4 การแสดงผลบน Google sheet เมื่อมี 20 ข้อมูล



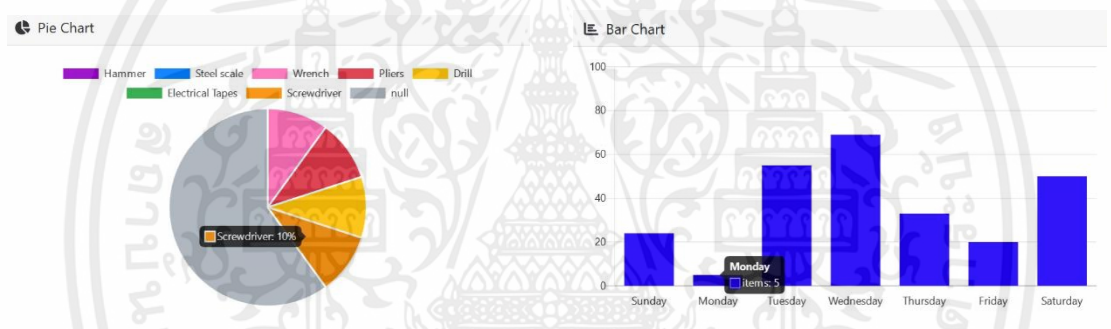
ภาพที่ 4.5.5 การแสดงผลบน Google sheet เมื่อมี 25 ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

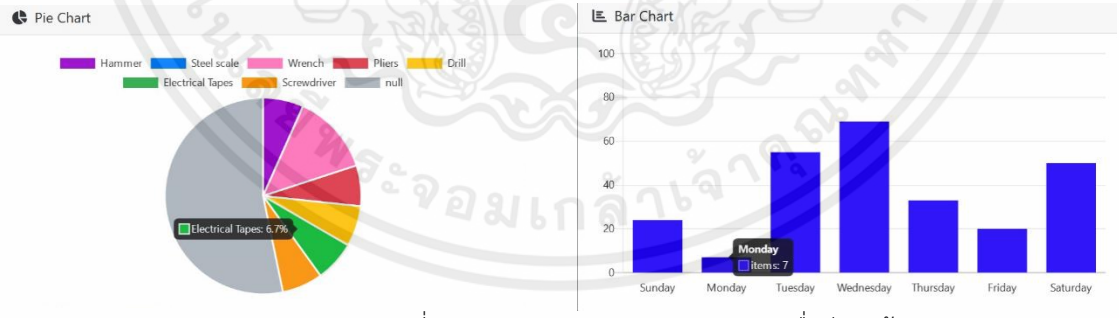
4.6 สามารถแสดงผลบน Dashboard ได้



ภาพที่ 4.6.1 การแสดงผลบน Dashboard เมื่อมี 5 ข้อมูล

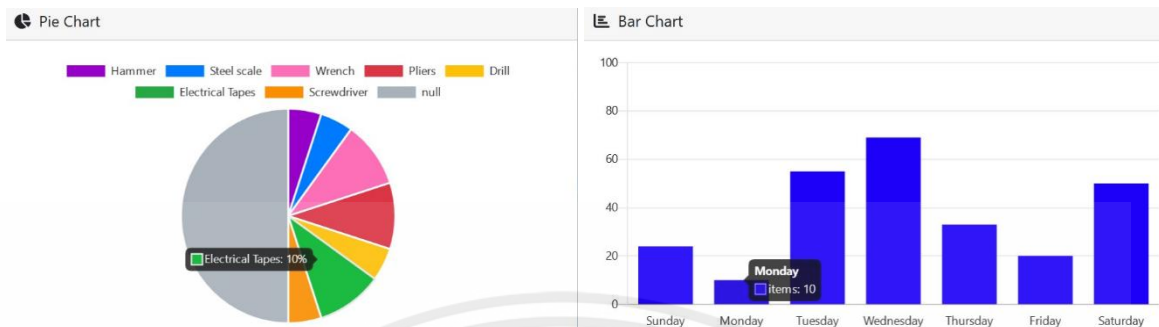


ภาพที่ 4.6.2 การแสดงผลบน Dashboard เมื่อมี 10 ข้อมูล

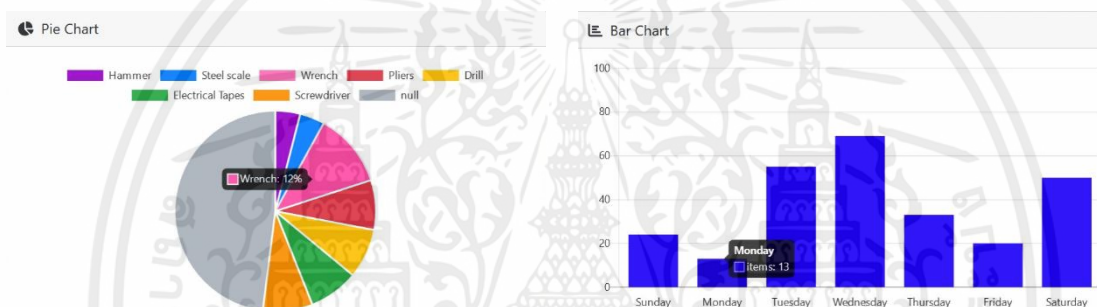


ภาพที่ 4.6.3 การแสดงผลบน Dashboard เมื่อมี 15 ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.6.4 การแสดงผลบน Dashboard เมื่อมี 20 ข้อมูล



ภาพที่ 4.6.5 การแสดงผลบน Dashboard เมื่อมี 25 ข้อมูล

จากผลการทดลองที่ 4.1 การหาเปอร์เซ็นต์ความคลาดเคลื่อนจากการวางแถบสายพาน พบว่า Ultrasonic Sensor มีการทำงานที่ผิดพลาดในบางครั้ง ซึ่งอาจเกิดจากความผิดพลาดจากการวาง sensor รวมถึงเปอร์เซ็นต์ความผิดพลาดของการอ่านค่าจาก Ultrasonic Sensor

จากผลการทดลองที่ 4.2 ทดสอบการแสดงผลของ Dashboard ผ่าน Laptop และ Smart phone พบว่าหน้าจอมีการแสดงผลที่ถูกต้องทั้ง 2 อุปกรณ์ ตัวอักษรไม่ขาดหาย

จากผลการทดลองที่ 4.3 ทดสอบเวลาที่ใช้ในการอ่านค่าจาก ESP8266 ไปยัง real-time firebase database และผลการทดลองที่ 4.4 ทดสอบเวลาที่ใช้ในการอ่านค่าจาก real-time firebase database ไปยัง Dashboard พบว่าเวลาเฉลี่ยที่อ่านค่าจาก ESP8266 ไปยัง real-time firebase database คือ 2.602 วินาที และเวลาเฉลี่ยที่อ่านค่าจาก real-time firebase database ไปยัง Dashboard คือ 0.252 วินาที ซึ่งรวมระยะเวลาในการแสดงผลทั้งหมด 2.854 วินาทีต่อภาค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

เมื่อจ่ายแหล่งจ่ายไฟให้กับบอร์ด ESP8266 และมอเตอร์ เริ่มแรกตัวมอเตอร์จะทำให้สายพานเคลื่อนที่ไปข้างหน้าพร้อมกับถาดที่วางบนสายพาน จากนั้นเมื่อเซ็นเซอร์แสงอินฟราเรด ที่ใช้ในการตรวจจับวัตถุจะเริ่มตรวจจับถาดที่เคลื่อนที่บนสายพาน และเมื่อเซ็นเซอร์แสงอินฟราเรดสามารถตรวจจับถาดบนสายพานได้ จะสั่งให้บอร์ด ESP8266 ควบคุมมอเตอร์ให้หยุดทำงานเป็นเวลา 5 วินาที จากนั้นสายพานหยุดเคลื่อนที่ และทำการวัดระยะทางจากเซ็นเซอร์ Ultrasonic แล้วคำนวณค่าระยะทางที่วัดได้ของความสูงถาดที่ได้จากการวัดของเซ็นเซอร์ Ultrasonic เพื่อให้ออกมาเป็นจำนวนถาดที่นับได้ แสดงออกมาทางจอ LCD จากนั้นนำถาดซึ่งมี RFID tag ที่ติดอยู่ด้านล่าง ไปอ่านค่าผ่าน RFID Reader พบว่าสามารถอ่านข้อมูลบนถาดได้ และสามารถส่งข้อมูลวันที่และเวลาที่อ่านถาด, สถานะของระบบ (Available/Unavailable), ประเภทของข้อมูล, จำนวนถาดต่อวัน และจำนวนถาดต่อสัปดาห์ ไปยัง real-time database firebase ได้ โดยสามารถแสดงผลบน google sheet ผ่านการดึงข้อมูลใน App Script รวมถึงสามารถแสดงผลบน dashboard ได้ โดยประกอบด้วยข้อมูลที่ส่งมาจาก ESP8266, กราฟ แผนภูมิแท่ง และแผนภูมิวงกลม

จากผลการทดลองในบทที่ 4 พบว่าเป็นไปตามวัตถุประสงค์ของโครงการ โดยที่จำนวนถาดที่นับได้จากการใช้เซ็นเซอร์ Ultrasonic มีค่าความผิดพลาดไม่เกินร้อยละ 10 และเวลาในการแสดงผลจาก ESP8266 ไปยัง real-time database firebase ใช้เวลา 2.602 วินาที และการแสดงผลจาก real-time database firebase ไปยัง dashboard ใช้เวลา 0.252 วินาที ซึ่งรวมระยะเวลาในการแสดงผลทั้งหมดไม่เกิน 10 วินาทีต่อถาด

5.2 ข้อเสนอแนะ

จากผลการทดลองพบว่าในการทดสอบนับจำนวนถาดผ่านระบบสายพานจำลองยังคงมีความผิดพลาด เนื่องจากความสูงของแต่ละด้านของถาดไม่มีความสม่ำเสมอ หากนำไปใช้งานควรเลือกใช้กับถาดที่มีความสูงเท่ากัน รวมถึงระยะที่ใช้ในการอ่านวัตถุภายในถาดผ่านระบบ RFID มีค่าน้อย ซึ่งมีระยะ 0-1 เซนติเมตร ทำให้หากนำไปใช้งานอาจจะทำให้ส่งผลต่อความคลาดเคลื่อนได้ จึงควรมีการเพิ่มอุปกรณ์เพื่อใช้ในการแสดงผลว่าได้มีการอ่านถาดแล้ว เพื่อช่วยเพิ่มความแม่นยำให้กับระบบมากขึ้น นอกจากนี้ การแสดงผลใน google sheet มีการส่งข้อมูลทุกๆ 1 นาที ทำให้การอ่านข้อมูลจากใน google sheet มีความล่าช้า ซึ่งเป็นข้อจำกัดของ trigger

บรรณานุกรม

- [1] Smart identify. 2562. RFID Desktop Reader /เครื่องอ่านอาร์เอฟไอดี
จาก : <https://www.smartiden.com/17224323/rfid-desktop-reader>
สืบค้นเมื่อ 20 ธันวาคม 2566
- [2] รศ.ดร.พนัส นัถฤทธิ. 2563. ไมโครคอนโทรลเลอร์
จาก : <https://www.nupress.grad.nu.ac.th/ไมโครคอนโทรลเลอร์>
สืบค้นเมื่อ 25 ธันวาคม 2567
- [3] NPRU Admin. 2563. การรับส่งข้อมูลแบบอนุกรม
จาก : <http://courseware.npru.ac.th/admin/files/การรับส่งข้อมูลแบบอนุกรม.pdf>
สืบค้นเมื่อ 19 มกราคม 2567
- [4] Admin. 2564. ESP8266 คือ อะไร และตัวอย่างการใช้งาน
จาก : <https://techtalk2apply.com/what-is-esp8266/>
สืบค้นเมื่อ 20 มกราคม 2567
- [5] Admin. 2564. RFID สุดยอดเทคโนโลยีทั้งในปัจจุบัน และอนาคต
จาก : https://www.stj1972.com/เกี่ยวกับ_RFID/
สืบค้นเมื่อ 20 มกราคม 2566
- [6] Admin. 2565. มารู้อีกกับ RFID (อาร์เอฟไอดี) เทคโนโลยีสารพัดประโยชน์
จาก : <https://www.pospak.com/th/what-is-rfid-blog>
สืบค้นเมื่อ 21 มกราคม 2567

- [7] Admin. 2565. ทำความรู้จักกับ Cloud Database คืออะไรกัน
จาก : <https://acccloud.tech/clouddb/>
สืบค้นเมื่อ 28 มกราคม 2567
- [8] Meow. 2565. Dashboard คืออะไร พร้อมประโยชน์และตัวอย่าง
จาก : <https://blog.datath.com/dashboard-data-analysis/>
สืบค้นเมื่อ 29 มกราคม 2566
- [9] Opendevelopmentthailand. 2566. หลักการออกแบบ Dashboard
จาก : <https://data.opendevelopmentmekong.net/dataset/resource/3.7-how-to-design-dashboards.pdf>
สืบค้นเมื่อ 29 มกราคม 2567
- [10] Jirawatee. 2559. รู้จัก Firebase Realtime Database
จาก : <https://medium.com/firebasethailand/รู้จัก-Firebase-Realtime-Database->
สืบค้นเมื่อ 13 กุมภาพันธ์ 2566
- [11] Handson Technology. 2564. RC522 RFID Development Kit
จาก : <http://www.handsontec.com/dataspecs/RC522.pdf>
สืบค้นเมื่อ 15 กุมภาพันธ์ 2567



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Ordering information

Table 2. Ordering information

Type number	Package		Version
	Name	Description	
MFRC52201HN1/TRAYB ^[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1
MFRC52201HN1/TRAYBM ^[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1

[1] Delivered in one tray.

[2] Delivered in five trays.

6. Block diagram

The analog interface handles the modulation and demodulation of the analog signals.

The contactless UART manages the protocol requirements for the communication protocols in cooperation with the host. The FIFO buffer ensures fast and convenient data transfer to and from the host and the contactless UART and vice versa.

Various host interfaces are implemented to meet different customer requirements.

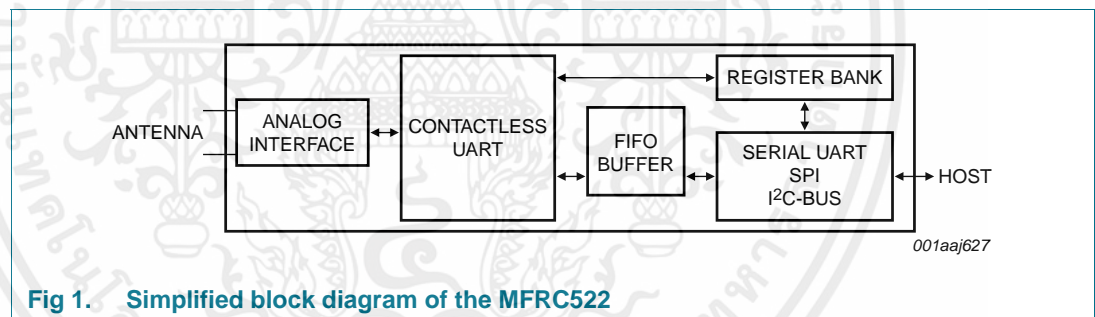


Fig 1. Simplified block diagram of the MFRC522

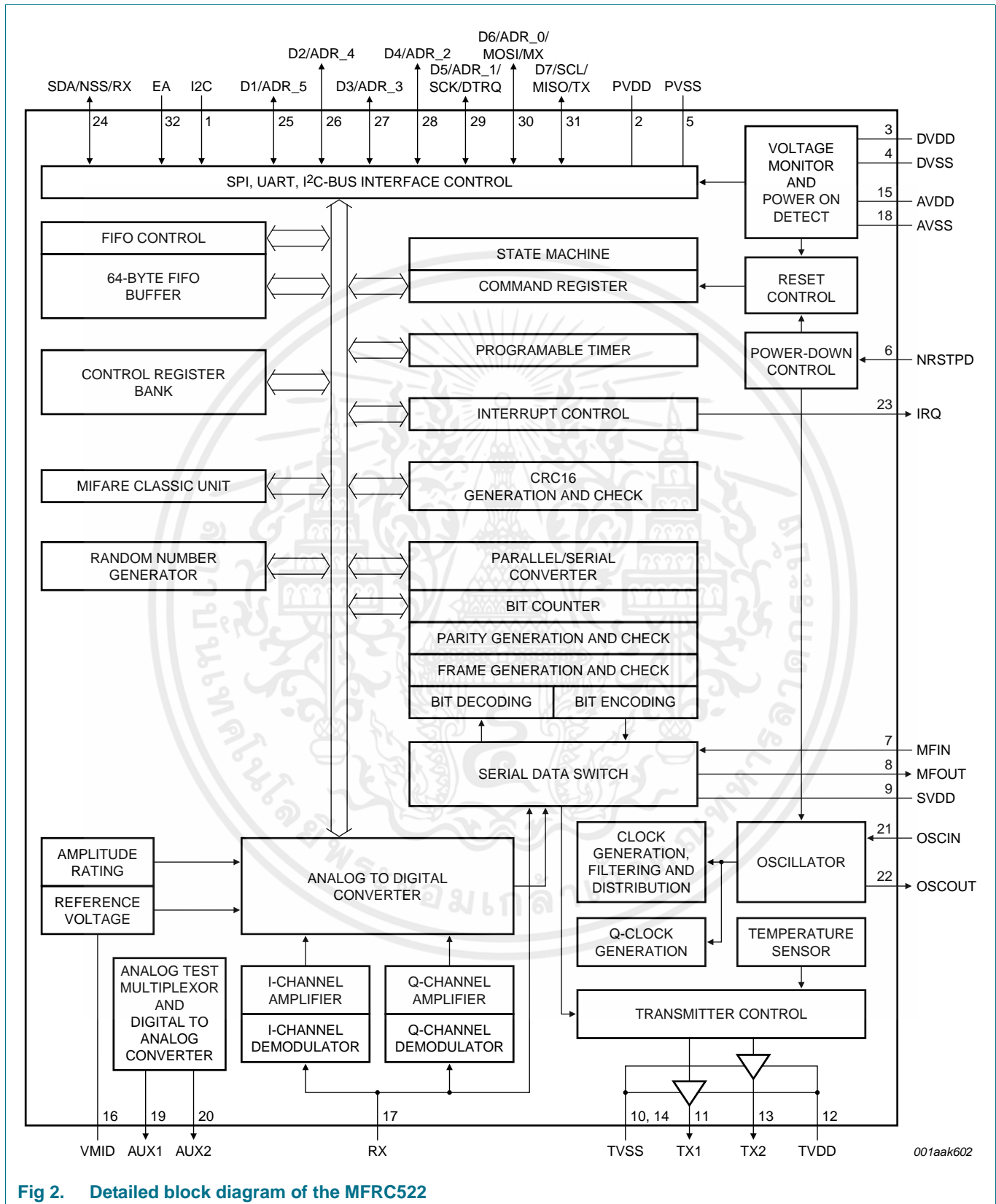


Fig 2. Detailed block diagram of the MFRC522

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

7. Pinning information

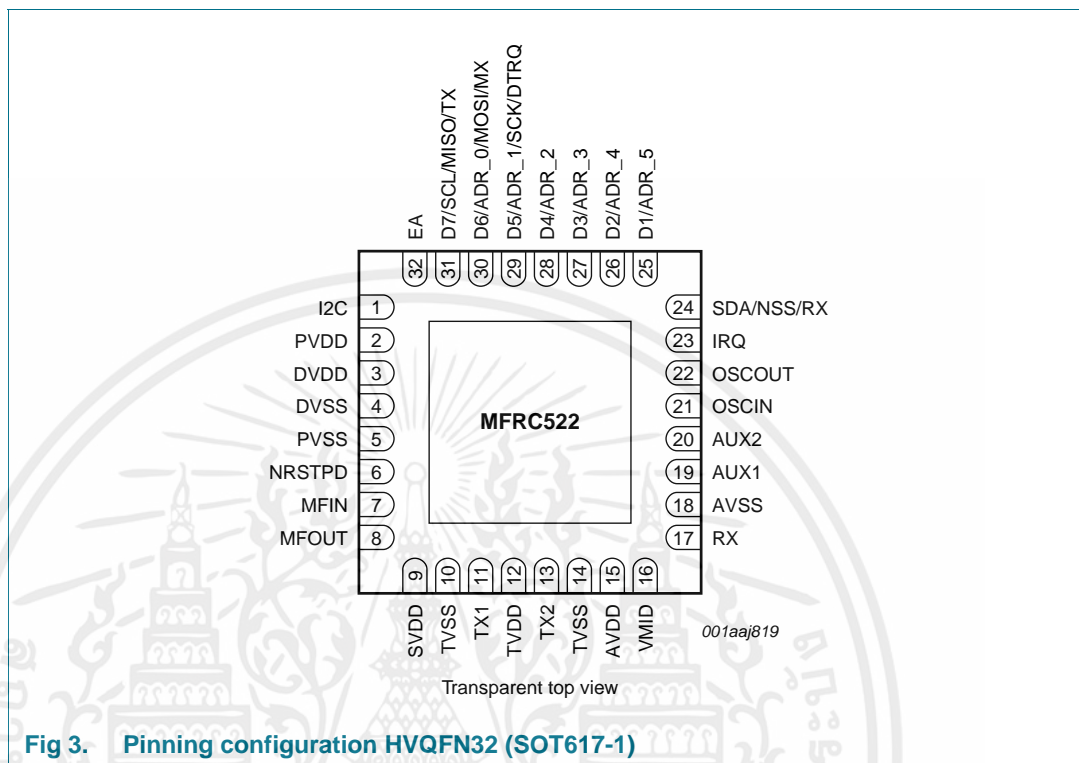


Fig 3. Pinning configuration HVQFN32 (SOT617-1)

7.1 Pin description

Table 3. Pin description

Pin	Symbol	Type ^[1]	Description
1	I2C	I	I ² C-bus enable input ^[2]
2	PVDD	P	pin power supply
3	DVDD	P	digital power supply
4	DVSS	G	digital ground ^[3]
5	PVSS	G	pin power supply ground
6	NRSTPD	I	reset and power-down input: power-down: enabled when LOW; internal current sinks are switched off, the oscillator is inhibited and the input pins are disconnected from the outside world reset: enabled by a positive edge
7	MFIN	I	MIFARE signal input
8	MFOUT	O	MIFARE signal output
9	SVDD	P	MFIN and MFOUT pin power supply
10	TVSS	G	transmitter output stage 1 ground
11	TX1	O	transmitter 1 modulated 13.56 MHz energy carrier output
12	TVDD	P	transmitter power supply: supplies the output stage of transmitters 1 and 2
13	TX2	O	transmitter 2 modulated 13.56 MHz energy carrier output
14	TVSS	G	transmitter output stage 2 ground
15	AVDD	P	analog power supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

8. Functional description

The MFRC522 transmission module supports the Read/Write mode for ISO/IEC 14443 A/MIFARE using various transfer speeds and modulation protocols.

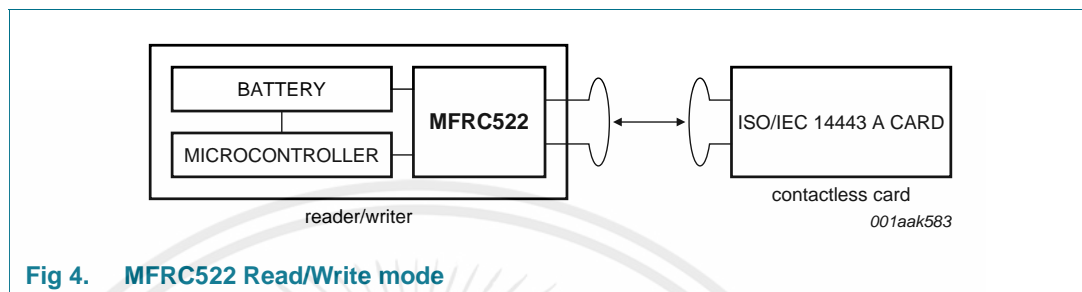


Fig 4. MFRC522 Read/Write mode

The physical level communication is shown in Figure 5.

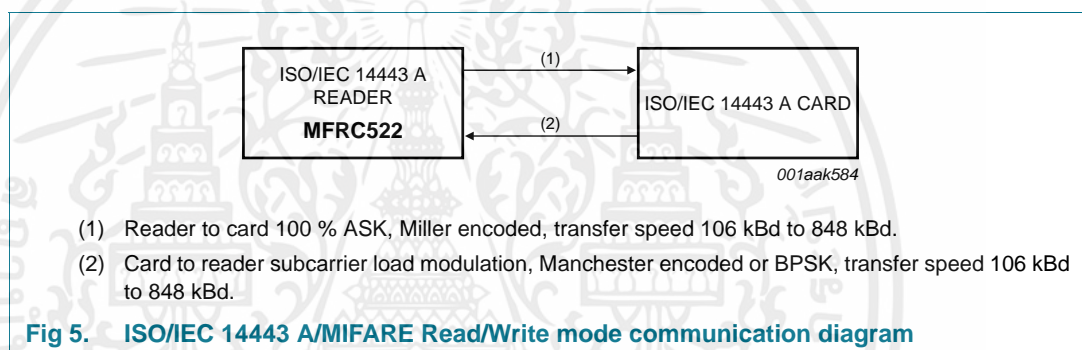


Fig 5. ISO/IEC 14443 A/MIFARE Read/Write mode communication diagram

The physical parameters are described in Table 4.

Table 4. Communication overview for ISO/IEC 14443 A/MIFARE reader/writer

Communication direction	Signal type	Transfer speed			
		106 kBd	212 kBd	424 kBd	848 kBd
Reader to card (send data from the MFRC522 to a card)	reader side modulation	100 % ASK	100 % ASK	100 % ASK	100 % ASK
	bit encoding	modified Miller encoding	modified Miller encoding	modified Miller encoding	modified Miller encoding
	bit length	128 (13.56 μs)	64 (13.56 μs)	32 (13.56 μs)	16 (13.56 μs)
Card to reader (MFRC522 receives data from a card)	card side modulation	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation
	subcarrier frequency	13.56 MHz / 16	13.56 MHz / 16	13.56 MHz / 16	13.56 MHz / 16
	bit encoding	Manchester encoding	BPSK	BPSK	BPSK

The MFRC522's contactless UART and dedicated external host must manage the complete ISO/IEC 14443 A/MIFARE protocol. Figure 6 shows the data coding and framing according to ISO/IEC 14443 A/MIFARE.

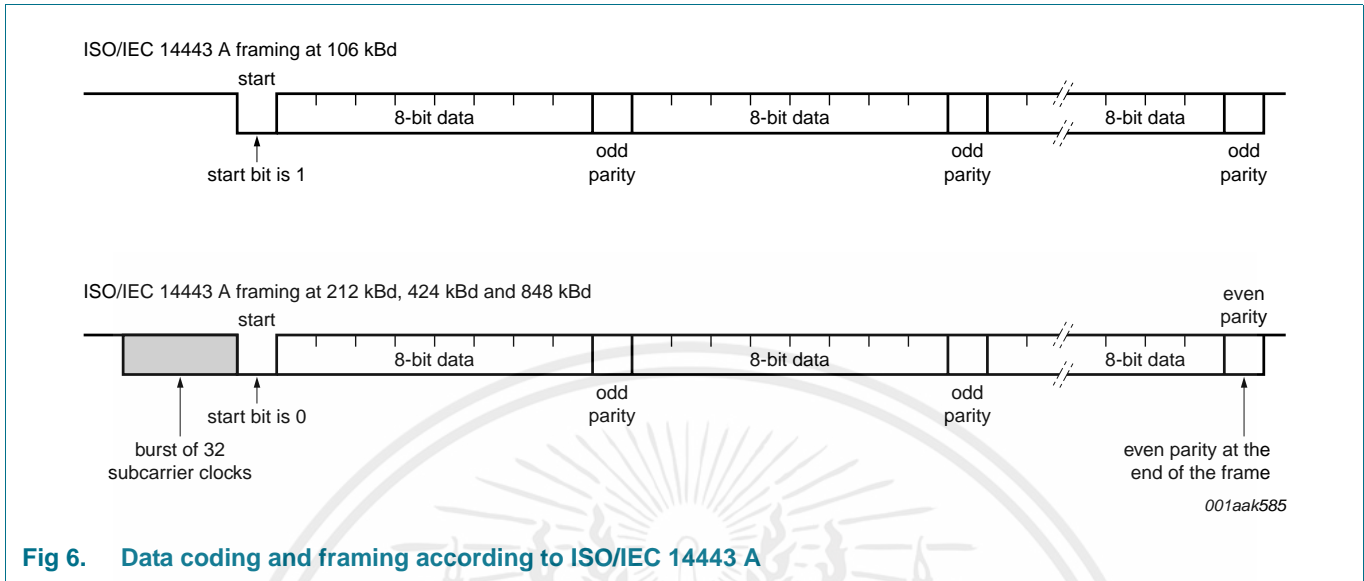


Fig 6. Data coding and framing according to ISO/IEC 14443 A

The internal CRC coprocessor calculates the CRC value based on ISO/IEC 14443 A part 3 and handles parity generation internally according to the transfer speed. Automatic parity generation can be switched off using the MfRxReg register's ParityDisable bit.

8.1 Digital interfaces

8.1.1 Automatic microcontroller interface detection

The MFRC522 supports direct interfacing of hosts using SPI, I²C-bus or serial UART interfaces. The MFRC522 resets its interface and checks the current host interface type automatically after performing a power-on or hard reset. The MFRC522 identifies the host interface by sensing the logic levels on the control pins after the reset phase. This is done using a combination of fixed pin connections. [Table 5](#) shows the different connection configurations.

Table 5. Connection protocol for detecting different interface types

Pin	Interface type		
	UART (input)	SPI (output)	I ² C-bus (I/O)
SDA	RX	NSS	SDA
I2C	0	0	1
EA	0	1	EA
D7	TX	MISO	SCL
D6	MX	MOSI	ADR_0
D5	DTRQ	SCK	ADR_1
D4	-	-	ADR_2
D3	-	-	ADR_3
D2	-	-	ADR_4
D1	-	-	ADR_5

8.1.2 Serial Peripheral Interface

A serial peripheral interface (SPI compatible) is supported to enable high-speed communication to the host. The interface can handle data speeds up to 10 Mbit/s. When communicating with a host, the MFRC522 acts as a slave, receiving data from the external host for register settings, sending and receiving data relevant for RF interface communication.

An interface compatible with SPI enables high-speed serial communication between the MFRC522 and a microcontroller. The implemented interface is in accordance with the SPI standard.

The timing specification is given in [Section 14.1 on page 75](#).

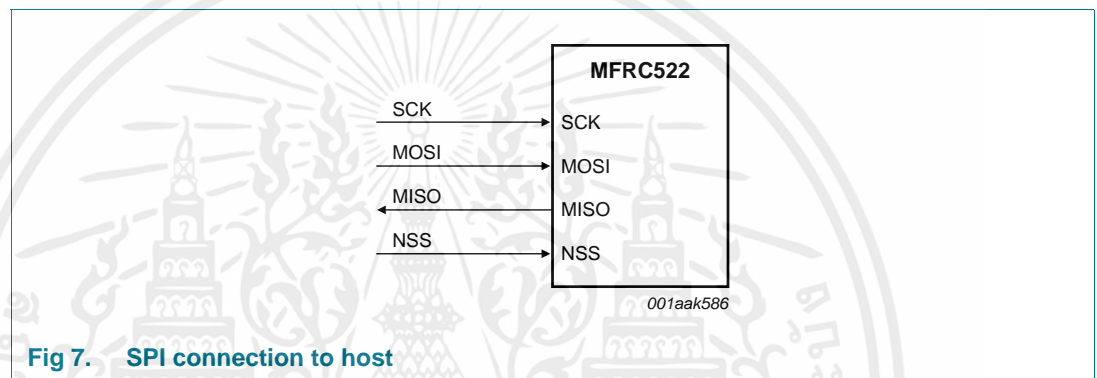


Fig 7. SPI connection to host

The MFRC522 acts as a slave during SPI communication. The SPI clock signal SCK must be generated by the master. Data communication from the master to the slave uses the MOSI line. The MISO line is used to send data from the MFRC522 to the master.

Data bytes on both MOSI and MISO lines are sent with the MSB first. Data on both MOSI and MISO lines must be stable on the rising edge of the clock and can be changed on the falling edge. Data is provided by the MFRC522 on the falling clock edge and is stable during the rising clock edge.

8.1.2.1 SPI read data

Reading data using SPI requires the byte order shown in [Table 6](#) to be used. It is possible to read out up to n-data bytes.

The first byte sent defines both the mode and the address.

Table 6. MOSI and MISO byte order

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	address 1	address 2	...	address n	00
MISO	X ^[1]	data 0	data 1	...	data n – 1	data n

[1] X = Do not care.

Remark: The MSB must be sent first.



1.

Overview

Espressif's ESP8266EX delivers highly integrated Wi-Fi SoC solution to meet users' continuous demands for efficient power usage, compact design and reliable performance in the Internet of Things industry.

With the complete and self-contained Wi-Fi networking capabilities, ESP8266EX can perform either as a standalone application or as the slave to a host MCU. When ESP8266EX hosts the application, it promptly boots up from the flash. The integrated high-speed cache helps to increase the system performance and optimize the system memory. Also, ESP8266EX can be applied to any microcontroller design as a Wi-Fi adaptor through SPI/SDIO or UART interfaces.

ESP8266EX integrates antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules. The compact design minimizes the PCB size and requires minimal external circuitries.

Besides the Wi-Fi functionalities, ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor and on-chip SRAM. It can be interfaced with external sensors and other devices through the GPIOs. Software Development Kit (SDK) provides sample codes for various applications.

Espressif Systems' Smart Connectivity Platform (ESCP) enables sophisticated features including:

- Fast switch between sleep and wakeup mode for energy-efficient purpose;
- Adaptive radio biasing for low-power operation
- Advance signal processing
- Spur cancellation and RF co-existence mechanisms for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation

1.1. Wi-Fi Key Features

- 802.11 b/g/n support
- 802.11n support (2.4 GHz), up to 72.2 Mbps
- Defragmentation
- 2 x virtual Wi-Fi interface
- Automatic beacon monitoring (hardware TSF)
- Support Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- Antenna diversity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



1.2. Specifications

Table 1-1. Specifications

Categories	Items	Parameters	
Wi-Fi	Certification	Wi-Fi Alliance	
	Protocols	802.11 b/g/n (HT20)	
	Frequency Range	2.4G ~ 2.5G (2400M ~ 2483.5M)	
	TX Power		802.11 b: +20 dBm
			802.11 g: +17 dBm
			802.11 n: +14 dBm
			802.11 b: -91 dbm (11 Mbps)
	Rx Sensitivity		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)	
Antenna		PCB Trace, External, IPEX Connector, Ceramic Chip	
Hardware	CPU	Tensilica L106 32-bit processor	
	Peripheral Interface		UART/SDIO/SPI/I2C/I2S/IR Remote Control
			GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V ~ 3.6V	
	Operating Current	Average value: 80 mA	
	Operating Temperature Range	-40°C ~ 125°C	
	Package Size	QFN32-pin (5 mm x 5 mm)	
	External Interface	-	
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station	
	Security	WPA/WPA2	
	Encryption	WEP/TKIP/AES	
	Firmware Upgrade	UART Download / OTA (via network)	
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming	
	Network Protocols	IPv4, TCP/UDP/HTTP	
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App	

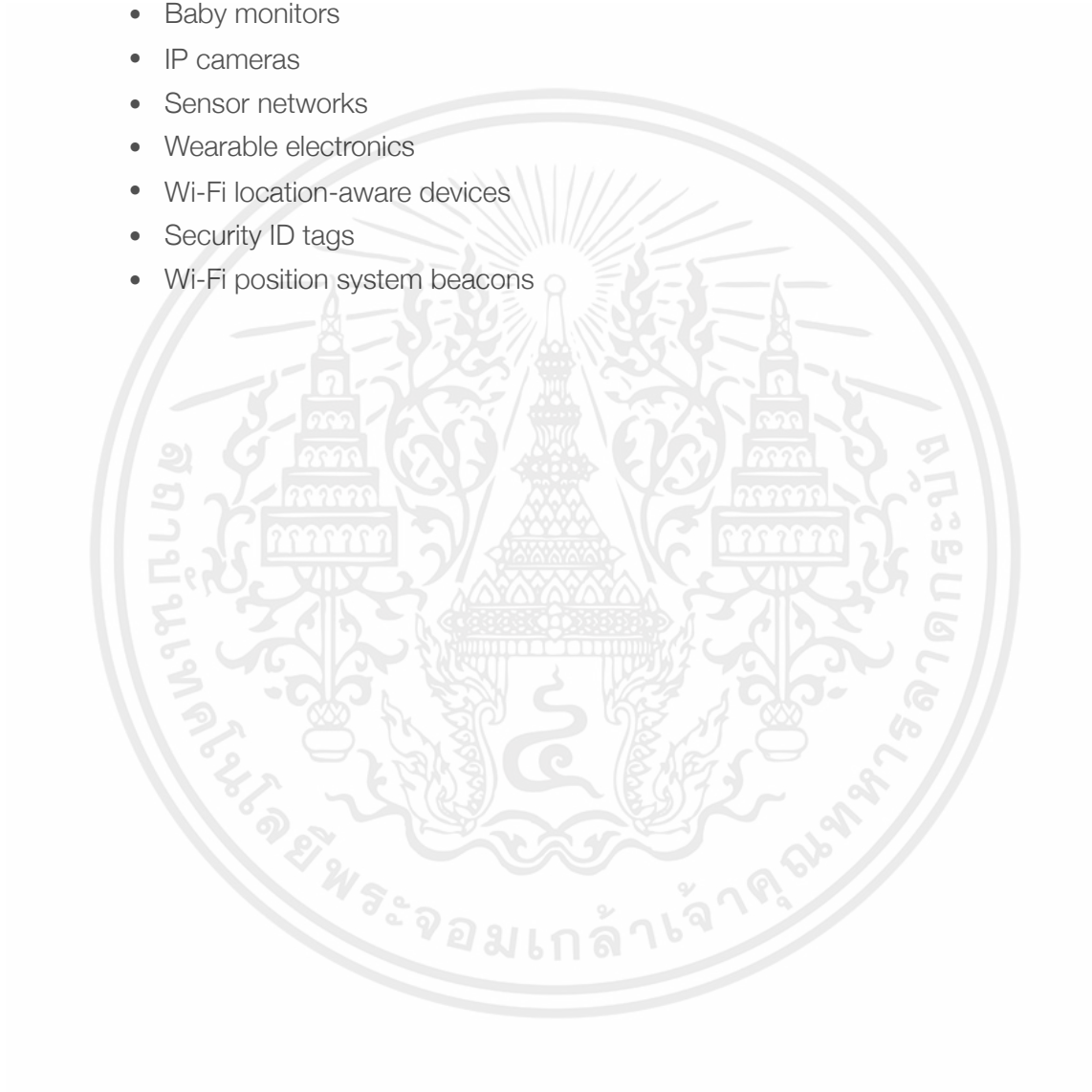
Note:

The TX power can be configured based on the actual user scenarios.



1.3. Applications

- Home appliances
- Home automation
- Smart plugs and lights
- Industrial wireless control
- Baby monitors
- IP cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Wi-Fi position system beacons





2.

Pin Definitions

Figure 2-1 shows the pin layout for 32-pin QFN package.

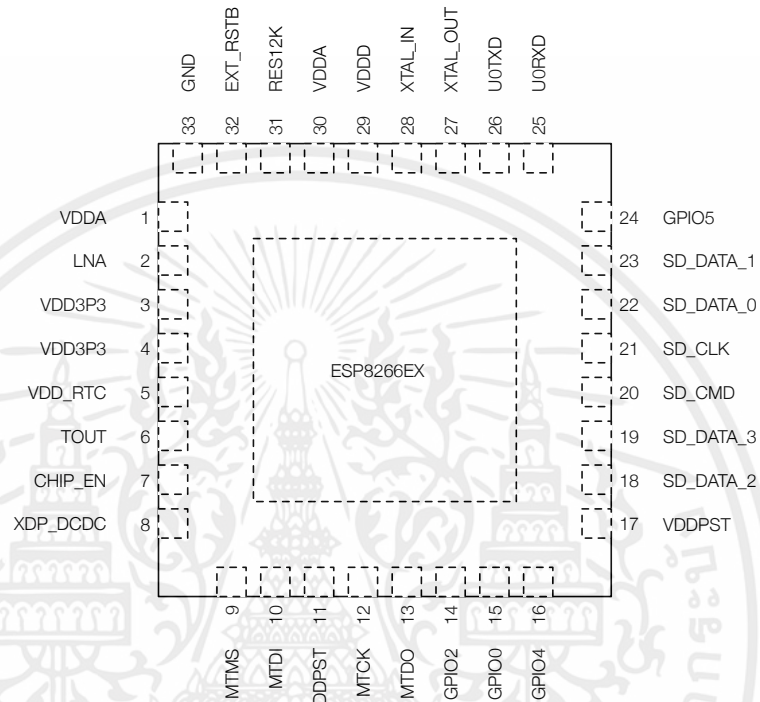


Figure 2-1. Pin Layout (Top View)

Table 2-1 lists the definitions and functions of each pin.

Table 2-1. ESP8266EX Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 2.5V ~ 3.6V
2	LNA	I/O	RF antenna interface Chip output impedance= $-39+j6 \Omega$. It is suggested to retain the π -type matching network to match the antenna.
3	VDD3P3	P	Amplifier Power 2.5V ~ 3.6V
4	VDD3P3	P	Amplifier Power 2.5V ~ 3.6V
5	VDD_RTC	P	NC (1.1V)
6	TOUT	I	ADC pin. It can be used to test the power-supply voltage of VDD3P3 (Pin3 and Pin4) and the input power voltage of TOUT (Pin 6). However, these two functions cannot be used simultaneously.



Pin	Name	Type	Function
7	CHIP_EN	I	Chip Enable High: On, chip works properly Low: Off, small current consumed
8	XPD_DCDC	I/O	Deep-sleep wakeup (need to be connected to EXT_RSTB); GPIO16
9	MTMS	I/O	GPIO 14; HSPI_CLK
10	MTDI	I/O	GPIO 12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V ~ 3.6V)
12	MTCK	I/O	GPIO 13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO 15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART TX during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V ~ 3.6V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MISO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART TX during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 2.5V ~ 3.6V
30	VDDA	P	Analog Power 2.5V ~ 3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: active)

Note:

1. GPIO2, GPIO0, and MTDO are used to select booting mode and the SDIO mode;
2. U0TXD should not be pulled externally to a low logic level during the powering-up.



3. Functional Description

The functional diagram of ESP8266EX is shown as in Figure 3-1.

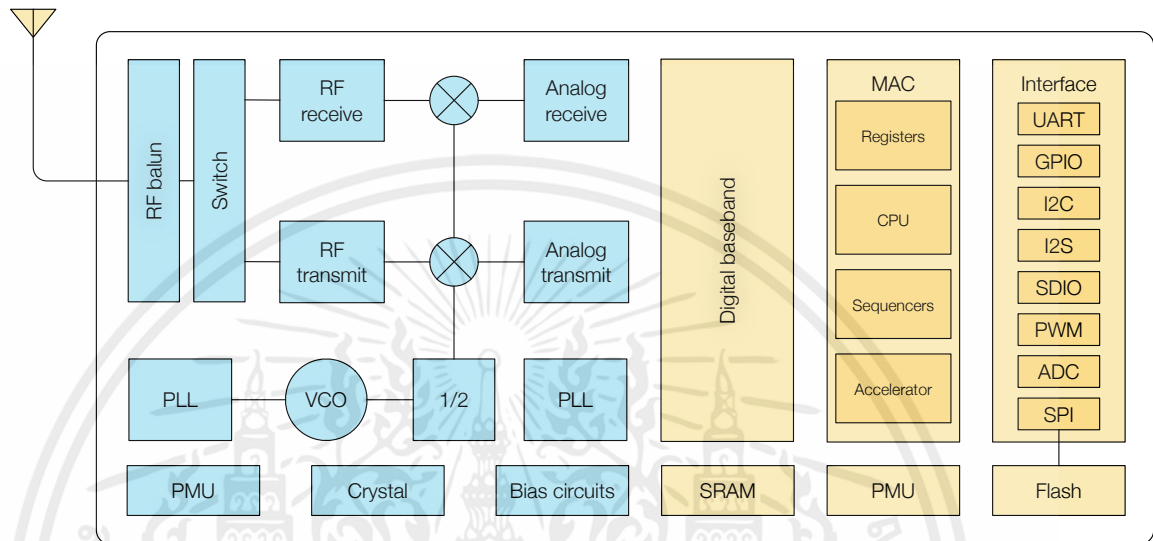


Figure 3-1. Functional Block Diagram

3.1. CPU, Memory, and Flash

3.1.1. CPU

The ESP8266EX integrates a Tensilica L106 32-bit RISC processor, which achieves extra-low power consumption and reaches a maximum clock speed of 160 MHz. The Real-Time Operating System (RTOS) and Wi-Fi stack allow 80% of the processing power to be available for user application programming and development. The CPU includes the interfaces as below:

- Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller, and can also be used to visit flash.
- Data RAM interface (dBus), which can connected with memory controller.
- AHB interface which can be used to visit the register.

3.1.2. Memory

ESP8266EX Wi-Fi SoC integrates memory controller and memory units including SRAM and ROM. MCU can access the memory units through iBus, dBus, and AHB interfaces. All memory units can be accessed upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor.

According to our current version of SDK, SRAM space available to users is assigned as below.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



- RAM size < 50 kB, that is, when ESP8266EX is working under the Station mode and connects to the router, the maximum programmable space accessible in Heap + Data section is around 50 kB.
- There is no programmable ROM in the SoC. Therefore, user program must be stored in an external SPI flash.

3.1.3. External Flash

ESP8266EX uses external SPI flash to store user programs, and supports up to 16 MB memory capacity theoretically.

The minimum flash memory of ESP8266EX is shown below:

- OTA disabled: 512 kB at least
- OTA enabled: 1 MB at least

! Notice:

SPI mode supported: Standard SPI, Dual SPI and Quad SPI. The correct SPI mode should be selected when flashing bin files to ESP8266. Otherwise, the downloaded firmware/program may not be working properly.

3.2. Clock

3.2.1. High Frequency Clock

The high frequency clock on ESP8266EX is used to drive both transmit and receive mixers. This clock is generated from internal crystal oscillator and external crystal. The crystal frequency ranges from 24 MHz to 52 MHz.

The internal calibration inside the crystal oscillator ensures that a wide range of crystals can be used, nevertheless the quality of the crystal is still a factor to consider to have reasonable phase noise and good Wi-Fi sensitivity. Refer to Table 3-1 to measure the frequency offset.

Table 3-1. High Frequency Clock Specifications

Parameter	Symbol	Min	Max	Unit
Frequency	FXO	24	52	MHz
Loading capacitance	CL	-	32	pF
Motional capacitance	CM	2	5	pF
Series resistance	RS	0	65	Ω
Frequency tolerance	Δ FXO	-15	15	ppm
Frequency vs temperature (-25°C ~ 75°C)	Δ FXO,Temp	-15	15	ppm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



3.2.2. External Clock Requirements

An externally generated clock is available with the frequency ranging from 24 MHz to 52 MHz. The following characteristics are expected to achieve good performance of radio.

Table 3-2. External Clock Reference

Parameter	Symbol	Min	Max	Unit
Clock amplitude	VXO	0.8	1.5	Vpp
External clock accuracy	Δ FXO,EXT	-15	15	ppm
Phase noise @1-kHz offset, 40-MHz clock	-	-	-120	dBc/Hz
Phase noise @10-kHz offset, 40-MHz clock	-	-	-130	dBc/Hz
Phase noise @100-kHz offset, 40-MHz clock	-	-	-138	dBc/Hz

3.3. Radio

ESP8266EX radio consists of the following blocks.

- 2.4 GHz receiver
- 2.4 GHz transmitter
- High speed clock generators and crystal oscillator
- Bias and regulators
- Power management

3.3.1. Channel Frequencies

The RF transceiver supports the following channels according to IEEE802.11b/g/n standards.

Table 3-3. Frequency Channel

Channel No.	Frequency (MHz)	Channel No.	Frequency (MHz)
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484



3.3.2. 2.4 GHz Receiver

The 2.4 GHz receiver down-converts the RF signals to quadrature baseband signals and converts them to the digital domain with 2 high resolution high speed ADCs. To adapt to varying signal channel conditions, RF filters, automatic gain control (AGC), DC offset cancelation circuits and baseband filters are integrated within ESP8266EX.

3.3.3. 2.4 GHz Transmitter

The 2.4 GHz transmitter up-converts the quadrature baseband signals to 2.4 GHz, and drives the antenna with a high-power CMOS power amplifier. The function of digital calibration further improves the linearity of the power amplifier, enabling a state of art performance of delivering +19.5 dBm average TX power for 802.11b transmission and +18 dBm for 802.11n (MCS0) transmission.

Additional calibrations are integrated to offset any imperfections of the radio, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities

These built-in calibration functions reduce the product test time and make the test equipment unnecessary.

3.3.4. Clock Generator

The clock generator generates quadrature 2.4 GHz clock signals for the receiver and transmitter. All components of the clock generator are integrated on the chip, including all inductors, varactors, loop filters, linear voltage regulators and dividers.

The clock generator has built-in calibration and self test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms to ensure the best performance of the receiver and transmitter.

3.4. Wi-Fi

ESP8266EX implements TCP/IP and full 802.11 b/g/n WLAN MAC protocol. It supports Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF). Power management is handled with minimum host interaction to minimize active-duty period.

3.4.1. Wi-Fi Radio and Baseband

The ESP8266EX Wi-Fi Radio and Baseband support the following features:

- 802.11b and 802.11g
- 802.11n MCS0-7 in 20 MHz bandwidth
- 802.11n 0.4 μ s guard-interval
- up to 72.2 Mbps of data rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



- Receiving STBC 2x1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity

3.4.2. Wi-Fi MAC

The ESP8266EX Wi-Fi MAC applies low-level protocol functions automatically, as follows:

- 2 × virtual Wi-Fi interfaces
- Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- Request To Send (RTS), Clear To Send (CTS) and Immediate Block ACK
- Defragmentation
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WEP (RC4) and CRC
- Automatic beacon monitoring (hardware TSF)
- Dual and single antenna Bluetooth co-existence support with optional simultaneous receive (Wi-Fi/Bluetooth) capability

3.5. Power Management

ESP8266EX is designed with advanced power management technologies and intended for mobile devices, wearable electronics and the Internet of Things applications.

The low-power architecture operates in the following modes:

- Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
- Modem-sleep mode: The CPU is operational. The Wi-Fi and radio are disabled.
- Light-sleep mode: The CPU and all peripherals are paused. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
- Deep-sleep mode: Only the RTC is operational and all other part of the chip are powered off.

Table 3-4. Power Consumption by Power Modes

Power Mode	Description	Power Consumption
Active (RF working)	Wi-Fi TX packet	Please refer to 5-2.
	Wi-Fi RX packet	
Modem-sleep ^①	CPU is working	15 mA
Light-sleep ^②	-	0.9 mA
Deep-sleep ^③	Only RTC is working	20 uA
Shut down	-	0.5 uA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



4. Peripheral Interface

4.1. General Purpose Input/Output Interface (GPIO)

ESP8266EX has 17 GPIO pins which can be assigned to various functions by programming the appropriate registers.

Each GPIO PAD can be configured with internal pull-up or pull-down (XPD_DCDC can only be configured with internal pull-down, other GPIO PAD can only be configured with internal pull-up), or set to high impedance. When configured as an input, the data are stored in software registers; the input can also be set to edge-trigger or level trigger CPU interrupts. In short, the IO pads are bi-directional, non-inverting and tristate, which includes input and output buffer with tristate control inputs.

These pins, when working as GPIOs, can be multiplexed with other functions such as I2C, I2S, UART, PWM, and IR Remote Control, etc.

For low power operations, the GPIOs can also be set to hold their state. For instance, when the IOs are not driven by internal and external circuits, all outputs will hold their states before the chip entered the low power modes.

The required drive strength is small— 5 μ A or more is enough to pull apart the latch.

4.2. Secure Digital Input/Output Interface (SDIO)

ESP8266EX has one Slave SDIO, the definitions of which are described as Table 4-1, which supports 25 MHz SDIO v1.1 and 50 MHz SDIO v2.0, and 1 bit/4 bit SD mode and SPI mode.

Table 4-1. Pin Definitions of SDIOs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SDIO_CLK
SDIO_DATA0	22	IO7	SDIO_DATA0
SDIO_DATA1	23	IO8	SDIO_DATA1
SDIO_DATA_2	18	IO9	SDIO_DATA_2
SDIO_DATA_3	19	IO10	SDIO_DATA_3
SDIO_CMD	20	IO11	SDIO_CMD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



4.3. Serial Peripheral Interface (SPI/HSPI)

ESP8266EX has two SPIs.

- One general Slave/Master SPI
- One general Slave HSPI

Functions of all these pins can be implemented via hardware.

4.3.1. General SPI (Master/Slave)

Table 4-2. Pin Definitions of SPIs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SPICLK
SDIO_DATA0	22	IO7	SPIQ/MISO
SDIO_DATA1	23	IO8	SPID/MOSI
SDIO_DATA_2	18	IO9	SPIHD
SDIO_DATA_3	19	IO10	SPIWP
U0TXD	26	IO1	SPICS1
GPIO0	15	IO0	SPICS2
SDIO_CMD	20	IO11	SPICS0

Note:

SPI mode can be implemented via software programming. The clock frequency is 80 MHz at maximum when working as a master, 20 MHz at maximum when working as a slave.

4.3.2. HSPI (Slave)

Table 4-3. Pin Definitions of HSPI (Slave)

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	HSPICLK
MTDI	10	IO12	HSPIQ/MISO
MTCK	12	IO13	HSPID/MOSI
MTDO	13	IO15	HPSICS

Note:

SPI mode can be implemented via software programming. The clock frequency is 20 MHz at maximum.



4.4. I2C Interface

ESP8266EX has one I2C, which is realized via software programming, used to connect with other microcontrollers and other peripheral equipments such as sensors. The pin definition of I2C is as below.

Table 4-4. Pin Definitions of I2C

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	I2C_SCL
GPIO2	14	IO2	I2C_SDA

Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized via software programming, and the clock frequency is 100 kHz at maximum.

4.5. I2S Interface

ESP8266EX has one I2S data input interface and one I2S data output interface, and supports the linked list DMA. I2S interfaces are mainly used in applications such as data collection, processing, and transmission of audio data, as well as the input and output of serial data. For example, LED lights (WS2812 series) are supported. The pin definition of I2S is shown in Table 4-5.

Table 4-5. Pin Definitions of I2S

Pin Name	Pin Num	I2S Data Input		Function Name
		IO		
MTDI	10	IO12		I2SI_DATA
MTCK	12	IO13		I2SI_BCK
MTMS	9	IO14		I2SI_WS
MTDO	13	IO15		I2SO_BCK
U0RXD	25	IO3		I2SO_DATA
GPIO2	14	IO2		I2SO_WS

4.6. Universal Asynchronous Receiver Transmitter (UART)

ESP8266EX has two UART interfaces UART0 and UART1, the definitions are shown in Table 4-6.



Table 4-6. Pin Definitions of UART

Pin Type	Pin Name	Pin Num	IO	Function Name
UART0	U0RXD	25	IO3	U0RXD
	U0TXD	26	IO1	U0TXD
	MTDO	13	IO15	U0RTS
	MTCK	12	IO13	U0CTS
UART1	GPIO2	14	IO2	U1TXD
	SD_D1	23	IO8	U1RXD

Data transfers to/from UART interfaces can be implemented via hardware. The data transmission speed via UART interfaces reaches 115200 x 40 (4.5 Mbps).

UART0 can be used for communication. It supports flow control. Since UART1 features only data transmit signal (TX), it is usually used for printing log.

Note:

By default, UART0 outputs some printed information when the device is powered on and booting up. The baud rate of the printed information is relevant to the frequency of the external crystal oscillator. If the frequency of the crystal oscillator is 40 MHz, then the baud rate for printing is 115200; if the frequency of the crystal oscillator is 26 MHz, then the baud rate for printing is 74880. If the printed information exerts any influence on the functionality of the device, it is suggested to block the printing during the power-on period by changing (U0TXD, U0RXD) to (MTDO, MTCK).

4.7. Pulse-Width Modulation (PWM)

ESP8266EX has four PWM output interfaces. They can be extended by users themselves. The pin definitions of the PWM interfaces are defined as below.

Table 4-7. Pin Definitions of PWM

Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	PWM0
MTDO	13	IO15	PWM1
MTMS	9	IO14	PWM2
GPIO4	16	IO4	PWM3

The functionality of PWM interfaces can be implemented via software programming. For example, in the LED smart light demo, the function of PWM is realized by interruption of the timer, the minimum resolution reaches as high as 44 ns. PWM frequency range is adjustable from 1000 μ s to 10000 μ s, i.e., between 100 Hz and 1 kHz. When the PWM frequency is 1 kHz, the duty ratio will be 1/22727, and a resolution of over 14 bits will be achieved at 1 kHz refresh rate.



4.8. IR Remote Control

ESP8266EX currently supports one infrared remote control interface. For detailed pin definitions, please see Table 4-8 below.

Table 4-8. Pin Definitions of IR Remote Control

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	IR TX
GPIO5	24	IO 5	IR Rx

The functionality of Infrared remote control interface can be implemented via software programming. NEC coding, modulation, and demodulation are supported by this interface. The frequency of modulated carrier signal is 38 kHz, while the duty ratio of the square wave is 1/3. The transmission range is around 1m which is determined by two factors: one is the maximum current drive output, the other is internal current-limiting resistance value in the infrared receiver. The larger the resistance value, the lower the current, so is the power, and vice versa.

4.9. ADC (Analog-to-Digital Converter)

ESP8266EX is embedded with a 10-bit precision SAR ADC. TOUT (Pin6) is defined as below:

Table 4-9. Pin Definition of ADC

Pin Name	Pin Num	Function Name
TOUT	6	ADC Interface

The following two measurements can be implemented using ADC (Pin6). However, they cannot be implemented at the same time.

- Measure the power supply voltage of VDD3P3 (Pin3 and Pin4).

Hardware Design	TOUT must be floating.
RF Initialization Parameter	The 107th byte of <i>esp_init_data_default.bin</i> (0 ~ 127 bytes), <i>vdd33_const</i> must be set to <i>0xFF</i> .
RF Calibration Process	Optimize the RF circuit conditions based on the testing results of VDD3P3 (Pin3 and Pin4).
User Programming	Use <i>system_get_vdd33</i> instead of <i>system_adc_read</i> .

- Measure the input voltage of TOUT (Pin6).

Hardware Design	The input voltage range is 0 to 1.0V when TOUT is connected to external circuit.
-----------------	--



RF Initialization Parameter	The value of the 107th byte of esp_init_data_default.bin (0 ~ 127 bytes), vdd33_const must be set to the real power supply voltage of Pin3 and Pin4. The unit and effective value range of vdd33_const is 0.1V and 18 to 36, respectively, thus making the working power voltage range of ESP8266EX between 1.8V and 3.6V,
RF Calibration Process	Optimize the RF circuit conditions based on the value of vdd33_const . The permissible error is $\pm 0.2V$.
User Programming	Use <code>system_adc_read</code> instead of <code>system_get_vdd33</code> .

Notes:

esp_init_data_default.bin is provided in SDK package which contains RF initialization parameters (0 ~ 127 bytes). The name of the 107th byte in **esp_init_data_default.bin** is **vdd33_const**, which is defined as below:

- When **vdd33_const** = 0xff, the power voltage of Pin3 and Pin4 will be tested by the internal self-calibration process of ESP8266EX itself. RF circuit conditions should be optimized according to the testing results.
- When $18 \leq \text{vdd33_const} \leq 36$, ESP8266EX RF Calibration and optimization process is implemented via ($\text{vdd33_const}/10$).
- When **vdd33_const** < 18 or $36 < \text{vdd33_const} < 255$, **vdd33_const** is invalid. ESP8266EX RF Calibration and optimization process is implemented via the default value 3.3V.