เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่
RFID Scanner Using Aduino Microcontroller

นางสาวกรกมล ไวยวุฒิ
นายคำรณ โยธี
นายพีรวัฒน์ ทองพูล

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม
คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร
พ.ศ. 2563

RFID scanner using Aduino microcontroller

Kornkamol Waiyawut
Kramron Yotee
Peerawat Thongpool

This is Project Report Submitted in Partial Fulfillment of
The Requirement for the Degree of Bachelor of Science in Technical Education
Program (Electrical Engineering)
Department of Electronic and Telecommunication Engineering
Faculty of Industrial Education
Rajamangala University of Technology Phrnakhon
2020

ชื่อโครงการ	**เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่**
ชื่อนักศึกษา	นางสาวกรกมล ไวยวุฒิ รหัส 036160505016-6
ชื่อนักศึกษา	นายคำรณ โยธี รหัส 036160505018-2
ชื่อนักศึกษา	**นายพีรวัฒน์ ทองพูล** รหัส 036160505014-1
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม
อาจารย์ที่ปรึกษา	อาจารย์วรรณภา มโนสืบ

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร อนุมัติให้รายงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

.....
(อาจารย์ภาวนา ชูศิริ)
หัวหน้าสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

คณะกรรมการสอบโครงการ

.....ประธาน
(อาจารย์วรรณภา มโนสืบ)

.....กรรมการ
(อาจารย์ภาวนา ชูศิริ)

.....กรรมการ
(อาจารย์อนุชา ไชยชาญ)

.....กรรมการ
(อาจารย์วารินี วีระสินธุ์)

.....กรรมการ
(อาจารย์สุปัญญา สิงห์กรณ์)

.....กรรมการ
(ผศ.ดร.รุ่งอรุณ พรเจริญ)

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม
อาจารย์ที่ปรึกษา อาจารย์วรรณภา มโนสืบ

บทคัดย่อ

โครงงานนี้ มีวัตถุประสงค์เพื่อพัฒนาเครื่องสแกนอาร์เอฟไอดี โดยมีขอบเขตดังต่อไปนี้ สแกนบัตรได้อย่างน้อย 15 ใบ มีการบันทึกข้อมูลทั้งเข้าและการออกได้ มีการแสดงผล Welcome บนหน้าจอและเพิ่มหรือลดรหัสได้ด้วยตัวเอง

ผลการทดสอบพบว่าเครื่องสแกนอาร์เอฟไอดีสามารถสแกนได้ทั้งคีย์การ์ดและสแกนด้วยตัวเองทั้งเข้าและออก อีกทั้งหน้าจอยังสามารถแสดงคำว่า Welcome ได้เมื่อมีคนเข้า และแสดงคำว่า Thank you เมื่อมีคนออก แสดงให้เห็นว่าเครื่องสแกนที่พัฒนาขึ้นสามารถนำไปประยุกต์ใช้งานได้จริง

(มีจำนวนหน้าทั้งสิ้น 111 หน้า)

Project Title RFID scanner using Aduino microcontroller
Students Miss KornKamol Waiyawut No. 036160505016-6
Students Mr Kramron Yotee No. 036160505018-2
Students Mr Peerawat Thongpool No. 036160505014-1
Major Field Electronics and Telecommunication Engineering
Advisor Miss Wannapa ManoSueb

Abstract

This project aims to develop RFID reader. With the following scope can scan at least 15 cards, save both entry and exit data, display Welcome on the screen and add or reduce codes by yourself

The test results show that the RFID scanner can scan both key cards and manually scan both in and out. The screen also shows Welcome when someone enters and shows thank you when someone shows that the developed scanner can be applied really slowly

(Total 111 pages)

กิตติกรรมประกาศ

รายงานเล่มนี้สำเร็จลุล่วงไปได้ด้วยการสนับสนุนจากอาจารย์วรรณภา มโนสืบ อาจารย์ที่ปรึกษาโครงการ อาจารย์สุปัญญา สิงห์กรณ์ อาจารย์ประจำวิชาที่ได้ให้คำปรึกษา ให้ความอนุเคราะห์ด้านวัสดุอุปกรณ์ และความกรุณาจากอาจารย์ประจำสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม ที่คอยชี้แนะปัญหาและวิธีตรวจสอบข้อบกพร่อง ตลอดจนอาจารย์ทุกๆ ท่าน ที่ให้คำปรึกษา ให้ข้อคิดให้กำลังใจ และสถานที่ในการทำงาน ระหว่างการทดลอง การแก้ไขปัญหาอุปสรรคต่าง ๆ จนโครงการนี้ประสบผลสำเร็จอย่างสมบูรณ์ คณะผู้จัดทำโครงการขอขอบพระคุณเป็นอย่างสูง

โครงการนี้ได้รับทุนอุดหนุนงบประมาณจากโครงการส่งเสริมสิ่งประดิษฐ์และนวัตกรรมคนรุ่นใหม่ ประจำปีงบประมาณ พ.ศ.2563 มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

สุดท้ายนี้คุณค่าและประโยชน์ใดๆ อันพึงมีจากโครงการนี้ คณะผู้จัดทำขอกราบขอบพระคุณคุณพ่อ คุณแม่ ครอบครัว ญาติพี่น้องทุกคนที่คอยช่วยเหลือให้การสนับสนุนด้านต่างๆ และเป็นกำลังใจมาโดยตลอด

กรกมล ไวยวุฒิ
คำรณ โยธี
พีรวัฒน์ ทองพูล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญภาพ	ซ
บทที่	
1 บทนำ	1
1.1 ปัญหาและความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 แผนการดำเนินงานตลอดโครงการ	2
1.6 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ	2
2 เอกสารและทฤษฎีที่เกี่ยวข้อง	3
2.1 Arduino Mega รุ่น2560	3
2.2 RFID RC 522	5
2.3 RFID TAG	6
2.4 Keypad ขนาด 4x4 membrane	8
2.5 LCD 20x4 I2C	9
2.6 Power Supply 5V 3A	11
2.7 NodeMCU V3	12
2.8 หน้าจอ Nextion 2.8	13
2.9 ภาษา Python	14
2.10 Anto Server	14
2.11 EEPROM	15
2.12 Google API	16
2.13 Google Drive	17

สารบัญ (ต่อ)

	หน้า
2.14 Google Sheets	18
2.15 RTC Clock DS3231	19
2.16 Arduino Uno	19
3 ขั้นตอนวิธีดำเนินงานโครงการ	21
3.1 แผนการดำเนินโครงการ	21
3.2 บล็อกไดอะแกรมการทำงานเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่	22
3.3 การออกแบบวงจรเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่	24
3.4 การออกแบบซอฟต์แวร์	27
3.5 การออกแบบฮาร์ดแวร์	39
4 ผลการดำเนินงาน	42
4.1 ขั้นตอนการทดสอบ	42
4.2 สรุปผลการทดสอบ	50
5 สรุปผล ปัญหาและข้อเสนอแนะ	51
5.1 สรุปผลโครงการ	51
5.2 ปัญหาและอุปสรรคในการทำโครงการ	52
5.3 ข้อเสนอแนะในการทำโครงการ	52
บรรณานุกรม	53
ภาคผนวก	55
ภาคผนวก ก ซอร์ซโค้ด	56
ภาคผนวก ข Data Sheets	84
ภาคผนวก ค คู่มือสำหรับการใช้งาน	91
ประวัติผู้เขียน	96

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน	2
2.1 LCD 20x4 I2C	11

สารบัญภาพ

ภาพที่		หน้า
2.1	Arduino Mega รุ่น2560	4
2.2	RFID RC 522	5
2.3	Passive RFID TAG	7
2.4	Active RFID TAG	8
2.5	4x4 Matrix Membrane Keypad	9
2.6	LCD 20x4 I2C	10
2.7	LCD 20x4 I2C	10
2.8	บล็อกไดอะแกรมของโปรแกรม SOPARE	11
2.9	บอร์ด Node MCU V3 รุ่น ESP8266	12
2.10	บอร์ด NodeMCU V3 รุ่น ESP8266	13
2.11	หน้าจอ Nextion 2.8	14
2.12	Anto Server	15
2.13	EEPROM	15
2.14	Google API	16
2.15	Google Drive	17
2.16	Google Sheets	18
2.17	RTC Clock DS3231	19
2.18	Arduino Uno	20
3.1	แผนการดำเนินงาน	21
3.2	บล็อกไดอะแกรมการทำงานเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อุยโน่	22
3.3	การออกแบบการทำงานของวงจรภาพรวม	24
3.4	วงจรการทำงานในส่วนของการรับค่าอินพุต	25
3.5	วงจรการทำงานในส่วนของการส่งค่าเอาต์พุต	26
3.6	วงจรการทำงานในการส่งค่าไปเก็บที่ Cloud	26
3.7	การแสดงผลบน Google sheet	27
3.8	การเขียนโค้ดบอร์ด Arduino mega 2560	27
3.9	การเขียนโค้ดบอร์ด Arduino mega 2560	29

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.10 การเขียนโค้ดบอร์ด Arduino mega 2560	28
3.11 การเขียนโค้ดบอร์ด Arduino mega 2560	29
3.12 การเขียนโค้ดบอร์ด Arduino mega 2560	29
3.13 การเขียนโค้ดบอร์ด Arduino mega 2560	30
3.14 การเขียนโค้ดบอร์ด Arduino mega 2560	30
3.15 การเขียนโค้ดบอร์ด Arduino mega 2560	31
3.16 การเขียนโค้ดบอร์ด Arduino mega 2560	31
3.17 การเขียนโค้ดบอร์ด Arduino mega 2560	32
3.18 การเขียนโค้ดบอร์ด Arduino mega 2560	32
3.19 การเขียนโค้ดบอร์ด Arduino mega 2560	33
3.20 การเขียนโค้ดบอร์ด Arduino UNO	33
3.21 การเขียนโค้ดบอร์ด Arduino UNO	34
3.22 การเขียนโค้ดบอร์ด Arduino UNO	34
3.23 การเขียนโค้ดบอร์ด Node MCU	35
3.24 การเขียนโค้ดบอร์ด Node MCU	35
3.25 การเขียนโค้ดบอร์ด Node MCU	36
3.26 การเขียนโค้ด anto	37
3.27 การเขียนโค้ด anto	37
3.28 การเขียนโค้ด anto	38
3.29 การเขียนโค้ด anto	39
3.30 เครื่องสแกนด้านนอกกล่อง	40
3.31 อุปกรณ์ภายในกล่อง	40
3.32 ภายนอกกล่องจะเป็นชื่อโครงการและผู้จัดทำ	41
4.1 เครื่องสแกนอาร์เอฟไอดี	42
4.2 หน้าต่าง Google	43
4.3 ค้นหา Google sheet	43
4.4 หน้าต่าง Google sheet	44

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.5 ภาพตาราง Google sheet	44
4.6 การสแกนคีย์การ์ดในการเข้า	45
4.7 ผลตารางการสแกนเข้าโดยคีย์การ์ด	45
4.8 การสแกนคีย์การ์ดในการออก	46
4.9 ผลตารางการสแกนเข้า (1) สแกนออก (0) โดยคีย์การ์ด	46
4.10 การกดรหัสเข้าด้วย Manual	47
4.11 ผลตารางการแสดงการเข้าด้วย Manual	48
4.12 การกดรหัสออกด้วยManual	48
4.13 ผลตารางแสดงการ เข้า (1) แสดงการ ออก (0) โดย Manual	49
4.14 ผลตารางการสแกนเข้า (1) การสแกนออก (0) โดยคีย์การ์ดจากกลุ่มตัวอย่าง	49
4.15 ผลตารางแสดงการเข้า (1) แสดงการออก (0) โดย Manual จากกลุ่มตัวอย่าง	50
ค 1 เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่	92
ค 2 การแสดงหน้าจอและรายละเอียดอุปกรณ์	93
ค 3 การสแกนบัตรด้วยคีย์การ์ดเข้า	93
ค 4 การสแกนบัตรด้วยคีย์การ์ดออก	94
ค 5 การใช้ Manual เข้า	94
ค 6 การใช้ Manual ออก	95

บทที่ 1
บทนำ

1.1 ปัญหาและความเป็นมาของโครงการ

ปัจจุบันบ้านเมืองของเรามีการพัฒนาสิ่งต่าง ๆ มากมายทั้งการสร้างอาคารตึกสูงเป็นจำนวนมากเพื่อนำไปใช้เป็นสำนักงานอพาร์ทเม้นท์ โรงแรม ห้างสรรพสินค้า รวมถึงสถาบันการศึกษาต่าง ๆ ซึ่งการที่จะล็อคประตูด้วยลูกกุญแจปกติมีความล่าช้าและไม่สะดวกต่อการงาน บางครั้งการใช้ลูกกุญแจปกติ อาจมีความไม่ปลอดภัยเพียงพอ เนื่องจากมีข่าวในเรื่องการโจรกรรมอยู่เสมอด้วยวิธีงัดห้องและวิธีอื่น ๆ ในสถานที่ต่าง ๆ ส่งผลให้เกิดความเสียหายสูญเสียทรัพย์สินส่วนบุคคลและส่วนรวม จึงต้องหาผู้กระทำผิดมารับผิดชอบในทรัพย์สินที่เสียหาย

Radio Frequency Identification คือ ระบบเก็บข้อมูลทางอิเล็กทรอนิกส์ที่เพิ่มความสามารถในการคำนวณและการรักษาความปลอดภัยของข้อมูล และส่งกำลังโดยคลื่นแม่เหล็กหรือ คลื่นแม่เหล็กไฟฟ้าแทนการสัมผัสทางกายภาพ เป็นการเอาคลื่นวิทยุมาเป็นคลื่นพาหะ เพื่อใช้ในการสื่อสารข้อมูล RFID มีลักษณะเป็นป้ายอิเล็กทรอนิกส์ (RFID Tag) ที่สามารถอ่านค่าได้โดยผ่านคลื่นวิทยุ จากระยะห่าง

ผู้จัดทำจึงได้สร้างเครื่องควบคุมการล็อคประตูด้วยระบบ RFID ขึ้นมาเพื่อป้องกันการโจรกรรมและเพิ่มความปลอดภัยในบ้านเรือน สำนักงาน ที่อยู่อาศัย สามารถใช้ แบบคีย์การ์ดและกดปุ่มรหัสเข้าออก โดยเครื่องควบคุมการล็อคประตูนี้สามารถ เพิ่ม-ลดจำนวนคีย์การ์ดและรหัส EEprom Library ใน บอร์ดอดุยโนเมก้า รุ่น 2560 เก็บข้อมูลได้ 1,024 ไบท์ ด้วยเงื่อนไขดังกล่าว ทำให้ทราบจำนวนผู้มีสิทธิเข้า-ออก และป้องกันการโจรกรรมเพื่อความปลอดภัยในชีวิตและทรัพย์สิน

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อพัฒนาเครื่องสแกนอาร์เอฟไอดี

1.3 ขอบเขตของโครงการ

1.3.1 สามารถใช้บัตรสแกนได้ 256 ใบ

1.3.2 สามารถบันทึกข้อมูลเข้าและออกได้

1.3.3 มีหน้าจอ TFT แสดงผล Welcome ขณะบันทึกข้อมูลเข้า

1.3.4 เพิ่มหรือลดรหัสด้วย Manual

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาข้อมูลที่เกี่ยวข้อง

1.4.2 เสนอโครงการ

1.4.3 จัดหาอุปกรณ์

1.4.4 สร้างเครื่องสแกนอาร์เอฟไอดี

1.4.5 สร้างและเขียนโปรแกรม

1.4.6 ทดสอบและแก้ไขโปรแกรม

1.4.7 ทดสอบเครื่องสแกนโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่

1.4.8 จัดทำรูปเล่มและสรุปผลการดำเนินงาน

1.5 แผนการดำเนินงานตลอดโครงการ

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	พ.ศ. 2562					พ.ศ.2563			
	มิ.ย	ก.ค	ส.ค	ก.ย	ต.ค	พ.ย	ธ.ค	ม.ค	ก.พ
1. ศึกษาค้นคว้าหาข้อมูลที่เกี่ยวข้อง	←	→							
2. ศึกษาการทำงานของอุปกรณ์		←	→						
3. จัดซื้ออุปกรณ์			←	→					
4. ออกแบบระบบ/เขียนโปรแกรม				←	→				
5. ทดสอบ/หาข้อผิดพลาด/แก้ไข					←	→			
6. สรุปผลการทดลอง							←	→	
7. จัดทำรูปเล่ม	←								→

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 เครื่องสแกนอาร์เอฟไอดีสามารถบันทึกเวลาเข้าออกเพื่อเป็นการรักษาความปลอดภัยในการเข้าออกได้

บทที่ 2
เอกสารและทฤษฎีที่เกี่ยวข้อง

การศึกษาทฤษฎีที่เกี่ยวข้อง ผู้จัดทำโครงการได้ศึกษาเอกสารในหัวข้อดังต่อไปนี้

2.1 Arduino Mega รุ่น2560
2.2 RFID RC 522
2.3 RFID TAG
2.4 Keypad ขนาด 4x4 membrane
2.5 LCD 20x4 I2C
2.6 Power Supply 5V 3A
2.7 NodeMCU V3
2.8 หน้าจอ Nextion 2.8”
2.9 ภาษา Python
2.10 Anto Server
2.11 EEPROM
2.12 Google API
2.13 Google Drive
2.14 Google Sheets
2.15 RTC Clock DS3231
2.16 Arduino Uno

2.1 Arduino Mega รุ่น2560

Arduino Mega รุ่น2560 คือบอร์ดไมโครคอนโทรลเลอร์ที่พัฒนาจาก ATmega 2560 ดังภาพที่ 2.1 มี 54 Digital Input/output โดยมี 14 ขา สามารถใช้เป็น Output แบบ PWM ได้มี Analog Input 16 ขา มี UARTS (Hardware Serial Ports) 4 ขา ทำงานที่ความถี่ 16 MHz สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยสายเคเบิล USB หรือใช้ adaptor AC-TO-DC เพื่อเริ่มต้นใช้งาน และมีปุ่ม Reset สามารถต่อเข้ากับ Shields ที่ออกแบบเพื่อใช้งานกับ Arduino Duemilanove หรือ Diecimila

ภาพที่ 2.1 Arduino Mega รุ่น2560
ที่มา : https://www.gravitechthai.com/

2.1.1 ส่วนประกอบของ Arduio Mega รุ่น2560 ดังนี้

- ขา VIN เป็น Input Voltage ของบอร์ด Arduino โดยใช้แหล่งจ่ายจากภายนอก
- ขา 5V เป็น Iutput Pin ที่ควบคุม 5 V จากบอร์ด
- 3.3 Volt Supply ที่สร้างขึ้นจาก Regulator บนบอร์ด และให้กระแสได้สูงสุด 50mA
- GND เป็น Groud pin
- IOREF เป็น Pin ที่ให้ Voltage Reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ Shield ที่มาเชื่อมกับบอร์ด

2.1.2 หน่วยความจำ

ATmega 2560 มีหน่วยความจำ 256 KB (8 KB ใช้สำหรับ Boot loader) นอกจากนี้ยังมีอีก 8 KB สำหรับ SRAM และ 4 KB สำหรับ EEPROM

2.1.3 Input and Output

ในแต่ละ Digital Pins ทั้ง 54 pins บนบอร์ด Arduino Uno สามารถเป็นได้ทั้ง Input และOutput โดยจะทำงานที่แรงดัน 5 V และให้กระแสสูงสุด 40 mA

2.1.4 ฟังก์ชัน

External Interrupts: 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 5), 19 (Interrupt 4), 21 (Interrupt 2) Pins เหล่านี้สามารถที่จะกำหนดค่าที่เรียก Interrupt ในค่าต่ำ ๆ ขอบขาขึ้นและลง หรือเปลี่ยนแปลงค่า

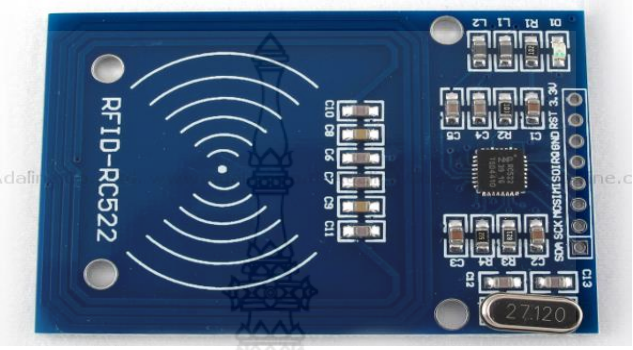
- PWM: 2 ถึง 13 และ 44 ถึง 46 ให้ Output PWM Output 8-Bits
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) ใช้สำหรับรองรับการสื่อสารแบบ SPI โดยที่ไม่เกี่ยวข้องกันกับ ICSP header ซึ่งจะมีลักษณะคล้ายกับ Uno, Duemilanove และ Diecimila

- LED 13: เป็น build-in LED ที่เชื่อมต่อกับ Digital pin 13 เมื่อ ญin มีค่าเป็น High LED จะติด แต่เมื่อ Pin เป็น Low LED จะดับ
- TWI : 20 (SDA) and 21 (SCL). รองรับการเชื่อมต่อแบบ TWI (I2C)
- บอร์ด Mega 2560 มี 16 Analog Inputs แต่ละ Pins ให้ความละเอียด 10 Bits

- AREF. แรงดันอ้างอิง สำหรับ Analog Input

- Reset ใช้ในการ Reset ไมโครคอนโทรลเลอร์ โดยทั่วไปจะใช้โดยการเพิ่มปุ่ม Reset ไว้บน Shield เพื่อป้องกันปุ่มที่อยู่บนบอร์ด

2.2 RFID RC 522



ภาพที่ 2.2 RFID RC 522
ที่มา: https://www.arduinothai.com/

RFID Module (RC522) ดังภาพที่ 2.2 เป็นชุด RFID ที่สามารถอ่านค่า และแท็บ 13.56 MHz ใช้ไฟเลี้ยง 3.3 VDC การสื่อสารเป็นแบบ SPI หรือ Serial Peripheral Interface เป็นวิธีการสื่อสารอนุกรม

คุณสมบัติของบอร์ด

MF RC522 การมอดูเลตและ Demodulation ขนาด 13.56 MHz สื่อสารและโปรโตคอล 14443A สัญญาณดาวเทียม ส่วนดิจิตอลจัดการเฟรม ISO14443A และการตรวจจับข้อผิดพลาด นอกจากนี้ยังรองรับอัลกอริทึมการเข้ารหัสด่วน CRYPTO1 ซึ่งเป็นชุดการตรวจสอบคำศัพท์ MIFARE MFRC522 รองรับชุด MIFARE ของการสื่อสารความเร็วสูงแบบไม่สัมผัสอัตราการถ่ายโอนข้อมูลแบบสองทางสูงสุด 424 kbit /s

โมดูล RFID

โมดูล MF522-AN ฟิลิปส์ MFRC522 เครื่องอ่านบัตรวงจรการออกแบบชิป โมดูลนี้สามารถโหลดโดยตรง เครื่องอ่านโมดูลใช้แรงดันไฟฟ้า 3.3V ผ่านอินเตอร์เฟซ SPI โดยตรงกับบอร์ด CPU ของผู้ใช้ ที่เชื่อมต่อกับโมดูลการสื่อสารได้อย่างมีประสิทธิภาพ

สเปคโมดูล MF522-AN

- ชื่อโมดูล: MF522-ED
- งานปัจจุบัน: 13-26mA / DC 3.3V
- ทนกระแสตั้งแต่80uA -30mA
- ความถี่ในการทำงาน: 13.56MHz
- ระยะการอ่านการ์ด: 0 ～ 60 มม. (การ์ด Mifare1)

- โปรโตคอล: SPI
- ความเร็วในการสื่อสารข้อมูล: สูงสุด 10Mbit / s
- รองรับประเภทการ์ด : Mifare1 S50, mifare1 S70, Mifare Ultra Light, Mifare Pro, Mifare Desfire
- ขนาดBoaed ขนาด 40 mm × 60 mm

สภาพแวดล้อม

- อุณหภูมิในการทำงาน: -20 - 80 องศา
- อุณหภูมิการจัดเก็บ: -40 - 85 องศา
- ความชื้น: ความชื้นที่เกี่ยวข้อง 5% - 95%
- ความเร็ว SPI สูงสุด: 10Mbit / s

2.3 RFID TAG

RFID TAG เป็นโครงสร้างภายในของแท็กประกอบด้วย 2 ส่วน ได้แก่ขดลวดขนาดเล็กซึ่งทำหน้าที่เป็นสายอากาศ (Antenna) สำหรับรับส่งสัญญาณคลื่นความถี่วิทยุ และสร้างพลังงานป้อนให้ส่วนของไมโครชิพ (Microchip) ที่ทำหน้าที่เก็บข้อมูลของวัตถุ เช่น รหัสสินค้า โดยทั่วไปตัวแท็กอยู่ในชนิดทั้งเป็นกระดาษ แผ่นฟิล์ม พลาสติก มีขนาดและรูปร่างต่างๆกัน ทั้งนี้ขึ้นอยู่กับวัสดุที่นำไปติดและมีหลายรูปแบบเช่นขนาด ขนาดเท่ากับบัตรเครดิต เหรียญ กระดุม ฉลากสินค้า แคปซูน ดังภาพที่ 2.3 เป็นต้น ส่วนในเรื่องของโครงสร้าง จะแบ่งชนิดของ Tag ได้ 2 ประเภทใหญ่ ๆ ดังนี้

2.3.1 Passive RFID TAG

คือ เป็นแท็กที่ไม่ต้องอาศัยแหล่งจ่ายไฟภายนอกใดๆ เพราะภายในจะมีวงจรกำเนิดไฟฟ้าเหนี่ยวนำขนาดเล็กอยู่ ฉะนั้นการอ่านข้อมูลได้ไม่ไกล ระยะไม่เกิน 1 เมตรขึ้นอยู่กับความแรงของเครื่องส่ง และคลื่นความถี่วิทยุ หน่วยความจำขนาดเล็ก 16 – 1,024 ไบร์ท มี IC (Integrated circuits) เป็นอุปกรณ์ควบคุมโครงสร้างหลัก 3 ส่วนคือ

ส่วนที่ 1 ส่วนควบคุมการทำงานของภาครับ-ส่งสัญญาณวิทยุ (Analog Front End)

ส่วนที่ 2 ส่วนควบคุมภาค Logic (Digital Control Unit)

ส่วนที่ 3 ส่วนของหน่วยความจำ ซึ่งอาจจะเป็นแบบ ROM หรือ EPROM

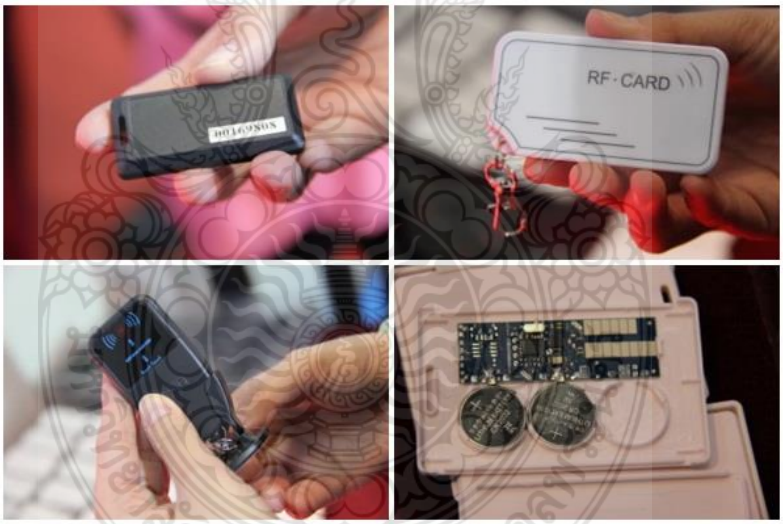
Raspbian คือ ระบบปฏิบัติการสำหรับติดตั้งใช้งานบนบอร์ดขนาดเล็กนาม Raspberry Pi พัฒนามาจากระบบ Debian Linux เหมาะสำหรับนำมาใช้ทำแล็ป และงานวิจัยเกี่ยวกับระบบคอมพิวเตอร์แบบฝังตัว (Embedded System) โดยที่ Raspbian มีแพ็กแกจให้ใช้งานกว่า 35,000 แพ็กเกจ กล่าวได้ว่าสามารถติดตั้งแพ็กเกจที่ใช้งานใน Debian Linux และ Ubuntu Linux ได้เกือบทุกแพ็กเกจ

- เครื่องคอมพิวเตอร์ windows ที่อ่าน SD card ได้
- SD card 8 GB ขึ้นไป

ขั้นตอนการลง OS Raspbian

ขั้นตอนที่ 1 ดาวน์โหลด Raspbian จาก www.raspberrypi.org/downloads/ ซึ่งเป็นไฟล์ image สำหรับติดตั้งบนบอร์ด Raspberry Pi

ภาพที่ 2.3 Passive RFID TAG
ที่มา : http://www.rfid-asia.com/



ภาพที่ 2.4 Active RFID TAG
ที่มา : http://www.rfid-asia.com

2.3.2 Active RFID TAG

Active RFID TAG คือ แท็กที่ต้องอาศัยแหล่งจ่ายไฟจากแบตเตอรี่ภายนอกเพื่อจ่ายพลังงานให้กับวงจรภายในทำงานซึ่งจะมีหน่วยความจำได้ถึง 1 เมกะไบร์ท ดังภาพที่ 2.4 การอ่านข้อมูลได้ไกลสูงสุด 10 เมตรซึ่งแท็กชนิดนี้ สามารถแบ่งประเภทการอ่านและเขียนข้อมูลได้ 2 ประเภท ดังนี้ คือ

ประเภทที่ 1 สามารถถูกอ่านและเขียนข้อมูลได้อย่างอิสระ (Read-Write)

ประเภทที่2 สามารถเขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างอิสระ (Write - Once Read- Many หรือ WORM)

2.4 Keypad ขนาด 4x4 Membrane

Keypad ขนาด 4x4 Membrane คือชุดโมดูลปุ่มกดกึ่งสำเร็จรูป สำหรับ Input ข้อมูลหรือป้อนคำสั่งไปยังไมโครคอนโทรลเลอร์ เช่น บอร์ด Arduino ที่มีการ Input ค่าต่าง ๆ เข้าไปในวงจร เพื่อให้ทำงานตามต้องการ เช่น วงจรจับเวลา, กรอกรหัสผ่าน ดังภาพที่ 2.5

ภาพที่ 2.5 4x4 Matrix Membrane Keypad
ที่มา : https://commandronestore.com

ประโยชน์ของ Matix Membrane Keypad

Matix Membrane Keypad ใช้งานง่ายและรวดเร็ว ประหยัดเวลาในการทดลองวงจร ประหยัดจำนวน Pin เพราะจำนวน Pin ที่ใช้ มักจะน้อยกว่าจำนวนปุ่มที่ได้ เช่นในที่นี้ 4x4 ใช้ 8 Pin แต่ได้ถึง 16 ปุ่ม จะได้เหลือ Pin Arduino เอาไปทำอย่างอื่นได้อีกมีLibrary ครบครันกดง่าย เนื่องจากภายในปุ่มมีพลาสติกบาง เรียกว่า Membrane ซึ่งสามารถหดและขยายตัวได้ตามแรงกด ซึ่งเป็นแผ่นโลหะเล็กๆติดอยู่ด้วย ทำหน้าที่แทนหน้าสัมผัสของ Push Button ทั่วไป

คุณสมบัติทั่วไป

Keypad สำหรับส่งค่า Input ข้อมูลไปยัง Microcontroller 16 ปุ่ม ใช้งานได้สะดวก รวดเร็ว ทนไฟเลี้ยงสูงสุด 24 V, 30 mA, Membrane แบบบางกดง่ายพกพาสะดวก

- ขนาด4x4 (16 ปุ่ม)
- ใช้งาน 8 Pin
- ขนาด Matix Membrane Keypad 6.9 cm x 7.6 cm
- ความยาวสายแพ 8.8 cm

2.5 LCD 20x4 I2C

จอแสดงผล LCD แบบ Resistive Touch ขนาด 2.8 นิ้วความละเอียด 320 x 240 พิกเซล รุ่น NX3224K028 ในตระกูล Nextion หน่วยความจำ Flash Memory 16 MB EEPROM 1 KB

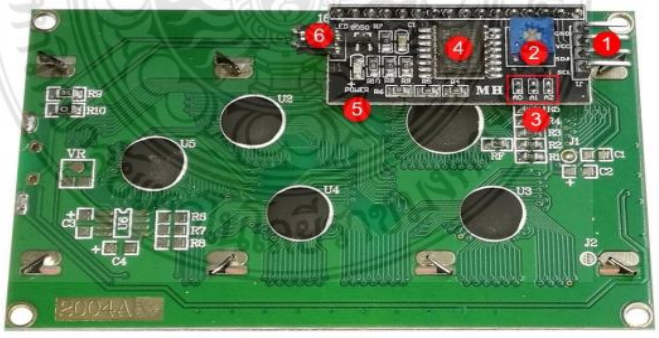
RAM 3.5 KB แสดงสีได้ 65K มาพร้อม Real-Time Clock และ GPIO บนบอร์ด สามารถประยุกต์ทำ Human Machine Interface (HMI) ได้ สื่อสารและสั่งงานผ่านพอร์ตอนุกรม UART ใช้งานร่วมกับ Arduino ดังภาพที่ 2.6

ภาพที่ 2.6 LCD 20x4 I2C
ที่มา : http://www.ett.co.th/

คุณสมบัติทางเทคนิค

- LCD แบบตัวอักษรขนาด 20 ตัวอักษร 4 บรรทัดพื้นสีน้ำเงินตัวอักษรสีขาว (STN Negative Blue)
- การเชื่อมต่อแบบ I2C-bus สามารถต่อร่วมกันได้ 8 ตัวบนบัสเดียวกัน (เลือก Address ต่างกัน)
- ใช้ไอซีขยายพอร์ตนัมเบอร์ PCF8574A หรือ PCF8574 ในการเชื่อมต่อกับจอ LCD
- ใช้แรงดันไฟเลี้ยง 5 VDC

ส่วนประกอบของแผงวงจร LCD 20 x 4 I2C ดังภาพที่ 2.7



ภาพที่ 2.7 LCD 20x4 I2C
ที่มา : http://www.ett.co.th/

หมายเลข 1 ขาต่อใช้งานมี 4 ขา ดังนี้

1.1 ขา GND ขากราวด์

1.2 ขา VCC ขาใช้แรงดันไฟเลี้ยง 5 VDC

1.3 ขา SDA ขาสัญญาณ DATA ของระบบ I2C-bus

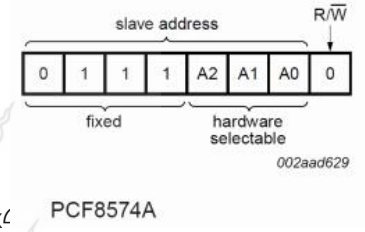
1.4 ขา SCL ขาสัญญาณ CLOCK ของระบบ I2C-bus

หมายเลข 2 ตัวต้านทานปรับค่าได้ สำหรับปรับความเข้มของจอ LCD

หมายเลข 3 Jumper เลือก Address (A0-A2) ของจอ LCD หากถ้าไม่เชื่อมต่อจะได้ค่าลอจิก 1 และถ้าเชื่อมต่อจะ ได้ค่าลอจิก 0 ซึ่งปกติจะไม่ได้บัดกรีไว้ทำให้ Address เริ่มต้นคือ 0x3F (A2=1 , A1=1 , A0=1) แต่ถ้าผู้ใช้ต้องการเปลี่ยน Address ก็สามารถเปลี่ยนได้ 8 ค่า คือ (0x38) - (0x3F) ดังตารางที่ 2.1

PCF8574A address map

Pin connectivity			Address of PCF8574A								Address byte value		7-bit hexadecimal address without R/W
A2	A1	A0	A6	A5	A4	A3	A2	A1	A0	R/W	Write	Read	
V_{SS}	V_{SS}	V_{SS}	0	1	1	1	0	0	0	-	70h	71h	38h
V_{SS}	V_{SS}	V_{DD}	0	1	1	1	0	0	1	-	72h	73h	39h
V_{SS}	V_{DD}	V_{SS}	0	1	1	1	0	1	0	-	74h	75h	3Ah
V_{SS}	V_{DD}	V_{DD}	0	1	1	1	0	1	1	-	76h	77h	3Bh
V_{DD}	V_{SS}	V_{SS}	0	1	1	1	1	0	0	-	78h	79h	3Ch
V_{DD}	V_{SS}	V_{DD}	0	1	1	1	1	0	1	-	7Ah	7Bh	3Dh
V_{DD}	V_{DD}	V_{SS}	0	1	1	1	1	1	0	-	7Ch	7Dh	3Eh
V_{DD}	V_{DD}	V_{DD}	0	1	1	1	1	1	1	-	7Eh	7Fh	3Fh



ที่มา : http://www.ett.co.th

2.6 Power supply 5V 3A

2.6.1 Switching Power supply แหล่งจ่ายไฟ 5V 3A

สวิทชิ่งเพาเวอร์ซัพพลาย (Switching Power Supply) แหล่งจ่ายไฟขนาด 5V 3A สามารถแปลงไฟจาก 220V เป็น 5V กระแสต่อเนื่อง 3A

ภาพที่ 2.8 บล็อกไดอะแกรมของโปรแกรม SOPARE

ที่มา : https://www.bishoph.org/

รายละเอียด ข้อมูล Switching Power Supply จากภาพที่ 2.8

- Switching power supply AC 100-240V to DC 5V 3A 15W Module

- มีระบบตัดไฟอัตโนมัติ เมื่อมีการช๊อตวงจร
- แรงดัน-อินพุต :100-240VAC
- แรงดันเอาต์พุต: 5Vdc
- กระแสเอาต์พุตสูงสุด: 3A
- กำลังเอาต์พุตสูงสุด : 15W
- ขนาด : 85x58x34mm

2.7 Node MCU V3

2.7.1 บอร์ด NodeMCU V3 รุ่น ESP8266

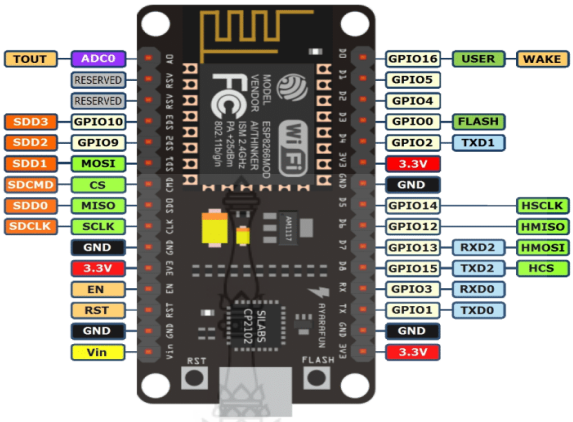
ภาพที่ 2.9 บอร์ด Node MCU V3 รุ่น ESP8266
ที่มา : https://l.facebook.com

บอร์ด NodeMCU V3 รุ่น ESP8266 ดังภาพที่ 2.9 เรียกของชิฟของโมดูล ESP8266 สำหรับติดต่อสื่อสารบนมาตรฐานทำงานที่แรงดันไฟฟ้า 3.0-3.6V ทำงานใช้กระแสโดยเฉลี่ย 80mA รองรับคำสั่ง Deep Sleep ในการประหยัดพลังงาน ใช้กระแสน้อยกว่า 10 ไมโครแอมป์ สามารถ Wake up กลับมาส่งข้อมูลใช้เวลาน้อยกกว่า 2 มิลลิวินาที ภายในมี Low Power MCU 32 bit ทำให้เราเขียนโปรแกรมสั่งงานได้ มีวงจร Analog Digital Converter ทำให้สามารถอ่านค่าจาก Analog ได้ความละเอียด 10bit ทำงานได้ที่อุณหภูมิ -40 ถึง 125 องศาเซลเซียส ESP8266 เชื่อมต่อกับ WI-FI แบบ Serial สามารถเขียนโปรแกรมลงไปในชิฟ โดยใช้ Arduino IDE ได้ ทำให้การเขียนโปรแกรมและใช้งานง่าย

ส่วนประกอบของแผงวงจร บอร์ด NodeMCU V3 รุ่น ESP8266 ดังภาพที่ 2.10

- ขา GPIO0 เป็นขาสำหรับเลือกโหมด โดยเมื่อต่อกับ GND จะเข้าโหมด
- ขา GPIO15เป็นขาที่ต้องต่อลง GND เพื่อให้โมดูลทำงาน
- ขา CH_PD หรือ EN เป็นขาที่ต้องต่อไฟ VCC เพื่อ Pull Up สัญญาณ ให้โมดูลทำงาน โมดูลบางร่นไม่มีขา Reset มาให้ เมื่อต้องการรีเซต ให้ต่อขา CH_PD กับ GND
- Reset ต่อกับไฟ VCC เพื่อ Pull Up สัญญาณ โดยเมื่อต้องการรีเซต ให้ต่อกับไฟ GND
- VCC เป็นขาสำหรับจ่ายไฟเลี้ยง ใช้ไฟเลี้ยง 3.0-3.6V
- ขา GND ต่อกับไฟ 0V
- ขา GPIO เป็นขาดิจิตอล Input/Output ทำงานที่ไฟ 3.3 V

- ขา ADC เป็นขา Analog Input รับแรงดันสูงสุด 1V ความละเอียด 10 bit



ภาพที่ 2.10 บอร์ด NodeMCU V3 รุ่น ESP8266
ที่มา : http://dtecesp8266arduino.blogspot.com/

2.8 หน้าจอ Nextion 2.8 นิ้ว

จอแสดงผล HMI TFT ขนาด 2.8 นิ้ว มีพื้นที่เก็บข้อมูลแฟลช 16MB, EEPROM 1024 ไบต์, RAM ขนาด 3584 ไบต์ ด้วยการรองรับ GPIO Nextion เป็นโซลูชั่น Human Machine Interface (HMI) ทำให้การควบคุมและการสร้างภาพข้อมูลระหว่างมนุษย์และกระบวนการเครื่องจักร แอพพลิเคชั่นหรืออุปกรณ์ Nextion ส่วนใหญ่จะใช้กับอินเทอร์เน็ตของสิ่ง (IoT) หรือเขตข้อมูลผู้บริโภคอิเล็กทรอนิกส์ มันเป็นทางออกที่ดีที่สุดในการแทนที่หลอดแอลซีดีแบบดั้งเดิมและ LED Nixie Nextion ประกอบด้วยส่วนฮาร์ดแวร์ (ชุดของแผง TFT) และส่วนซอฟต์แวร์ (ตัวแก้ไข Nextion) Nextion TFT Board ใช้พอร์ตอนุกรมเพียงพอร์ตเดียวในการสื่อสาร ดังภาพที่ 2.11

ภาพที่ 2.11 หน้าจอ Nextion 2.8
ที่มา : https://www.thaieasyelec.com/

2.9 ภาษา Python

Python คือ ชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux , Windows NT,

Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัว นี้เป็น Open Source เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีๆ โดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงาน ไวยากรณ์ของภาษา Python นั้นถูกพัฒนาขึ้นมาโดยมีความตั้งใจให้เป็นภาษาที่อ่านง่าย ออกแบบให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อน โดยมักจะใช้คำในภาษาอังกฤษ นอกจากนี้ Python มีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษา C และ Pascal Python Interpreter Python interpreter นั้นเป็นตัวแปรภาษาของภาษา Python เพื่อให้สามารถรันโค้ด Python ได้ ซึ่งได้มากับไลบรารี่มาตรฐานที่สามารถใช้งานได้ฟรี ซึ่งดาวน์โหลดได้ที่ https://www.python.org/ เป็นโปรแกรมแบบ source และ binary สำหรับแพลตฟอร์มที่ได้รับความนิยม นอกจากนี้ Interpreter สามารถเขียนโปรแกรมกับ Interactive shell ซึ่งเป็นการเขียนโค้ดของภาษา Python และเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที Python Interpreter นั้นยังสามารถนำเพิ่มความสามารถกับฟังก์ชันใหม่ที่ถูกพัฒนามาจากภาษา C และ C++ Python นั้นเหมาะสำหรับเป็นภาษาในการสร้าง Extension และแอพพลิเคชันที่ปรับแต่งได้

2.10 Anto Server

Anto Server คือสื่อกลางในการสื่อสารระหว่าง Hardware เข้าสู่ Internet และมีเครื่องมือที่ช่วยให้คุณสร้างสรรค์นวัตกรรมทางด้าน Internet of Things ได้ง่ายขึ้น ดังภาพที่ 2.12

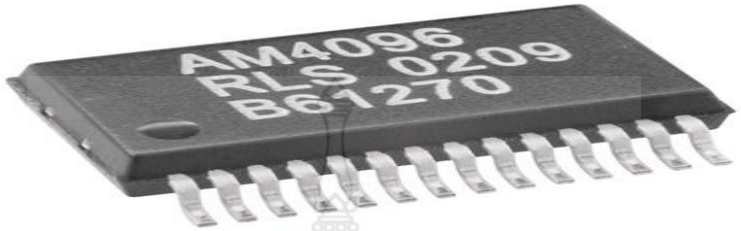
ภาพที่ 2.12 Anto Server
ที่มา : https://antoiot.gitbooks.io/

Anto Server เป็นสื่อกลางในการสื่อสาร เป็นเสมือนตัวกลางในการสื่อสารระหว่างสิ่งต่าง ๆ บนโลกอินเตอร์เน็ต เช่น การสั่งงานบอร์ดไมโครคอนโทรลเลอร์ผ่านอินเตอร์เน็ตใช้โทรศัพท์มือถือ เป็นต้น ซึ่งปัจจุบันระบบรองรับการติดต่อสื่อสารผ่าน HTTP, HTTPS, MQTT, MQTTS, Web socket ไม่จำเป็นต้องติดตั้งเซิร์ฟเวอร์ รวมถึงดูแลรักษาระบบ

2.11 EEPROM

EEPROM คือการทำงานของเครื่องมืออิเล็กทรอนิกส์ที่มีเทคโนโลยีสูงจะต้องมีหน่วยความจำที่เรียกว่ารอม (ROM : Read-only Memory) อยู่ในเครื่องมือเหล่านั้นเสมอ รอมนั้นจะมีโปรแกรมและข้อมูลที่คอยสั่งการณ์อุปกรณ์ต่างๆให้ทำงานตามคำสั่ง ซึ่ง ROM นี้จะมีความสามารถที่เก็บข้อมูลหรือโปรแกรมได้โดยไม่จำเป็นต้องมีไฟเลี้ยง

EEPROM ย่อมาจาก Electrical Erasable Programmable Read Only Memory เป็น ROM ชนิดหนึ่ง ซึ่งถูกพัฒนาและออกแบบมาให้สามารถใช้งานได้ง่าย และสามารถใช้งานแทน EPROM ได้ดี ดังภาพที่ 2.13



ภาพที่ 2.13 EEPROM
ที่มา : https://www.comgeeks.net/

EPROM (Electrical Erasable Programmable Read Only Memory) เป็นรอมที่พัฒนาจาก EPROM แต่การใช้งานง่าย เนื่องจากเป็นการลงโปรแกรมและการเพิ่มโปรแกรมใหม่ด้วยกระแสไฟฟ้า ซึ่งการใช้งานง่ายกว่า ในปัจจุบันมีการใช้งาน EEPROM แทน EPROM อย่างแพร่หลาย เพราะความสะดวกในการใช้งานพร้อมกันนั้นยังมีคุณสมบัติที่ไม่แตกต่างจาก EPROM สิ่งที่ทำให้ EEROM โดดเด่นกว่า EPROM คือความสามารถในการลงโปรแกรมแก้ไขและลบโปรแกรมที่มีความเร็วและสะดวกกว่าแบบ EPROM การลบข้อมูลของ EEPROM นั้นจะเป็นการลบข้อมูลทั้งหมด ไม่สามารถเลือกลบโปรแกรมบ้างส่วนได้ โดยอายุการใช้งานของ EEPROM นั้นจะขึ้นอยู่กับข้อกำหนดของ EEPROM ตัวนั้นว่าจะกำหนดให้สามารถลบและเขียนข้อมูลได้สูงสุดเท่าไร อาทิ 10 ครั้งหรือ 100 ครั้ง ซึ่งความสามารถของ EEPROM คือสามารถที่จะใช้ไฟฟ้าที่มีอยู่ในคอมพิวเตอร์ลบและลงโปรแกรมใหม่ได้

2.12 Google API

API ย่อมาจาก Application Programming Interface คือ คำสั่ง (Code) ที่อนุญาตให้ Software Program สื่อสารระหว่างกันได้ API เป็นช่องทางสำหรับใช้บริการคำสั่ง จาก Operation System (OS) หรือ Application ซึ่งใช้งานโดยติดตั้ง Function และเรียกใช้งานตาม Doncument ที่เขียนไว้ และ API เป็นอีกหนึ่งช่องทางที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API เช่น BOT และจากที่อื่นและเป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่อมการทำงานเข้ากับระบบปฏิบัติการได้

API ถือเป็นกลุ่มของฟังชั่นขั้นตอนหรือคลาส (Class) ที่ระบบปฏิบัติการ (OS) หรือผู้ให้บริการ สร้างขึ้นมา เพื่อรองรับการเรียกขอข้อมูล โปรแกรมอื่น ทั้งนี้ API สามารถใช้งานได้กับภาษาในการเขียนโปรแกรมที่รองรับเท่านั้น ซึ่งจะถูกจัดทำให้อยู่ในรูปแบบ Syntax หรือ element ที่นำไปใช้ได้อย่างสะดวกสบาย ดังภาพที่ 2.14

ภาพที่ 2.14 Google API

ที่มา : https://www.dailytech.in.th/

การใช้งาน APIs (Application Programming Interface)

ในปัจจุบันนี้ API ถูกใช้งานใน application เพื่อสื่อสารกับ User อย่างแพร่หลาย เช่น Facebook, Google, Twitter เป็นต้น และสามารถนำเอารูปแบบ API ขยายงานออกไปใช้ได้ ประโยชน์ของ API

- สามารถรับส่งข้อมูลข้าม Server ได้
- ไม่จำเป็นต้องเข้าหน้าเว็บหลัก ก็มีข้อมูลของเว็บหลัก จากเว็บที่ดึง API
- API ที่ขึ้นกับภาษา (language-dependent API) คือ เอพีไอ ที่สามารถการเรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาเพียงภาษาใดภาษาหนึ่ง

2.13 Google Drive

Google Drive เป็น Online Service ประเภท Cloud Technology ที่มีไว้สำหรับให้ผู้ใช้จัดเก็บข้อมูลลงไป สามารถใช้ได้ฟรีแต่ว่าต้องมี G-mail Account เท่านั้น การใช้งานประมาณ 5GB ซึ่งก็ถือว่ามีขนาดใหญ่ใช้ได้หากต้องการมีเนื้อที่เพิ่มเติมมากกว่านั้น สามารถทำได้ โดยการเสียค่าบริการเป็นรายเดือนหรือรายปี ดังภาพที่ 2.15

ภาพที่ 2.15 Google Drive

ที่มา : https://wikichaidan.wordpress.com/

การใช้งาน Google Drive

- ไฟล์ข้อมูลต่างๆ ที่เก็บไว้ใน Google Drive นั้น สามารถแชร์ (Share) ให้ผู้ใช้คนอื่น เช่น สาระการเรียนรู้ กลุ่มเพื่อน ครอบครัว มาเข้าถึงข้อมูลเหล่านั้นได้ และทำงานไปพร้อมกันในเวลาเดียวกันได้

- ไฟล์ข้อมูลต่างๆ ที่เก็บไว้ใน Google Drive เราสามารถค้นหาข้อมูลได้เต็มรูปแบบ (Full Text Search) หมายความว่าสามารถค้นหาสิ่งที่อยู่ในเนื้อภายในไฟล์นั้นด้วยเช่นเดียวกัน ความสามารถทางด้านการค้นหา (Search) นั้นทำได้เร็ว ไฟล์ใหม่ๆ ที่เพิ่งจะนำเข้า Google Drive นั้น ในเวลาไม่นานก็สามารถ Search ได้ทันที

ประโยชน์ Google Drive

- เข้าถึงไฟล์จากที่ใดก็ได้ Google Drive บน Mac, คอมพิวเตอร์พีซี, แอนดรอยด์ หรืออุปกรณ์ iOS จะทำให้มีพื้นที่เดียวสำหรับไฟล์ปัจจุบันจากทุกที่

- พกไฟล์ไปใช้พร้อมแบ่งปันไฟล์เดียวหรือทั้งโฟลเดอร์กับบุคคลที่กำหนด หรือทีมงานทั้งหมด หรือกระทั่งพนักงานชั่วคราว พันธมิตร และผู้มีสิทธิ์ สร้างและตอบกลับความคิดเห็นในไฟล์เพื่อรับข้อเสนอแนะและเพิ่มแนวคิดใหม่ ๆ

2.14 Google Sheets

Google Sheets คือ เป็นแอปพลิเคชันในกลุ่มของ Google Drive ซึ่งเป็นนวัตกรรมของ Google มีลักษณะการทำงานคล้ายกันกับ Microsoft Excel คือสามารถสร้าง Column, Row สามารถใส่ข้อมูลต่างๆ ลงไปใน Cell ได้ และคำนวณสูตรต่างๆได้ ดังภาพที่ 2.16

ภาพที่ 2.16 Google Sheets

ที่มา : https://antoiot.gitbooks.io/

ข้อดีของการใช้ Google Sheets

- เป็นบริการให้ใช้ฟรีจาก Google

- สามารถทำงานเป็นทีมได้สามารถทำงานร่วมกันในสเปรดชีท (Spread Sheet) ได้ในเวลาเดียวกัน

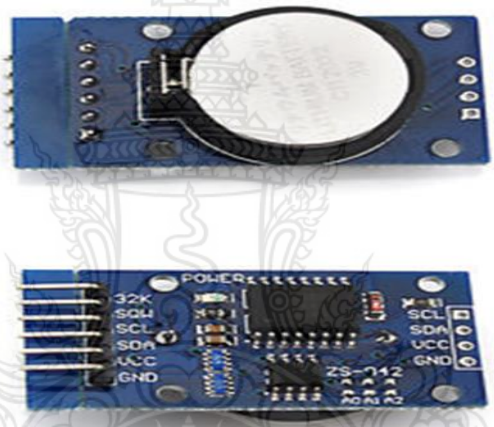
- นอกจากนี้ยังสามารถแชร์งาน แก้ไขแบบเรียลไทม์ หรือแม้กระทั่งแชทและแสดงความคิดเห็นกับบุคคลใดก็ได้

- ไม่ต้องกด "บันทึก" อีกเลย เมื่อมีการทำงานเกิดขึ้นในสเปรดชีท ทุกการพิมพ์จะถูกบันทึกไว้ทั้งหมดโดยอัตโนมัติ และยังสามารถใช้ประวัติการแก้ไขเพื่อดูเวอร์ชั่นเก่าของสเปรดชีทเดียวกัน โดยจัดเรียงตามวันที่และคนที่แก้ไข

- สามารถทำงานได้กับ Microsoft Excell สามารถเปิด แก้ไข และบันทึกเป็นไฟล์ Microsoft Excell

2.15 RTC Clock DS3231

DS3231 module โมดูนาฬิกาแบบเวลาจริง Real Time Clock (RTC) ค่าความแม่นยำสูง เนื่องภายในตัวโมดูลมีวงจรวัดอุณหภูมิเพื่อชดเชยความถี่ที่ Crytal เสียไปจากการที่อุณหภูมิเปลี่ยนแปลง หากใส่แบตสามารถทำงานได้เองโดยที่ไม่ต้องมีไฟเลี้ยง (เหมือนนาฬิกา) สามารถต่อกับ Arduino ผ่าน I2C เพื่ออ่านค่า วันที่ เวลาปัจจุบันได้ นอกจากนั้นยังมี function ตั้งค่า ตั้งปลุก AlarmRTC เก็บรักษาข้อมูลวินาที, นาที, ชั่วโมง, วัน, วันที่, เดือนและปี น้อยกว่า 31 วันของเดือน วันที่สิ้นสุดจะถูกปรับโดยอัตโนมัติรวมถึงการแก้ไขสำหรับปีอธิกสุรทิน นาฬิกาทำงานในรูปแบบ 24 ชั่วโมงหรือ band / AM / PM ทั้งในรูปแบบ 12 ชั่วโมง มีนาฬิกาปลุกที่กำหนดค่าได้สองแบบและปฏิทินสามารถตั้งค่าเป็นเอาต์พุตคลื่นสี่เหลี่ยม ที่อยู่และข้อมูลจะถูกถ่ายโอนอย่างต่อเนื่องผ่านบัส I2C สองทิศทาง ดังภาพที่ 2.17



ภาพที่ 2.17 RTC Clock DS3231
ที่มา : https://www.mosfex.com/

2.16 Arduino Uno

Arduino Uno เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้ ATmega328P (แผ่นข้อมูล) มี 14 ขา เข้า / ขาออกดิจิตอล (สามารถใช้เป็นสัญญาณ PWM), 6 อินพุตแบบอะนาล็อก, 6 แร่เซรามิก 16 เมกะเฮิรตซ์ (CSTCE16M0V53-R0), การเชื่อมต่อ USB, แจ็คไฟ, หัว ICSP และปุ่มรีเซ็ต มีทุกสิ่งที่จำเป็นในการรองรับไมโครคอนโทรลเลอร์ เพียงเชื่อมต่อกับคอมพิวเตอร์ด้วยสายเคเบิล USB หรือใช้พลังงานจากอะแดปเตอร์ AC-TO-DC หรือแบตเตอรี่เพื่อเริ่มต้นกับ UNO ดังภาพที่ 2.18

ภาพที่ 2.18 Arduino Uno
ที่มา : https://store.arduino.cc/

ข้อมูลทางเทคนิค

- แรงดันไฟฟ้า 5V
- แรงดันไฟฟ้าขาเข้า 7-12V
- แรงดันไฟฟ้าขาเข้า 6-20V
- Digital I / O Pins 14 (ซึ่ง 6 มีเอาต์พุต PWM)
- PWM Digital I / O Pins 6
- อินพุตแบบอะนาล็อก 6
- DC ต่อ I / O Pin 20 mA
- กระแสตรง 3.3V Pin 50 mA
- หน่วยความจำแฟลช 32 KB (ATmega328P) ซึ่ง 0.5 KB ที่ใช้ bootloader
- SRAM 2 KB (ATmega328P)
- EEPROM 1 KB (ATmega328P)
- ความเร็วสัญญาณนาฬิกา 16 MHz
- LED_BUILTIN 13
- ความยาว 68.6 mm
- กว้าง 53.4 mm
- น้ำหนัก 25 กรัม

บทที่ 3
ขั้นตอนวิธีดำเนินงานโครงการ

การจัดทำโครงการ เรื่อง เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ โดยทางผู้จัดทำได้จัดลำดับการดำเนินการตามขั้นตอนและวิธีการดังนี้

3.1 แผนการดำเนินโครงการ

3.2 บล็อกไดอะแกรมการทำงานเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่

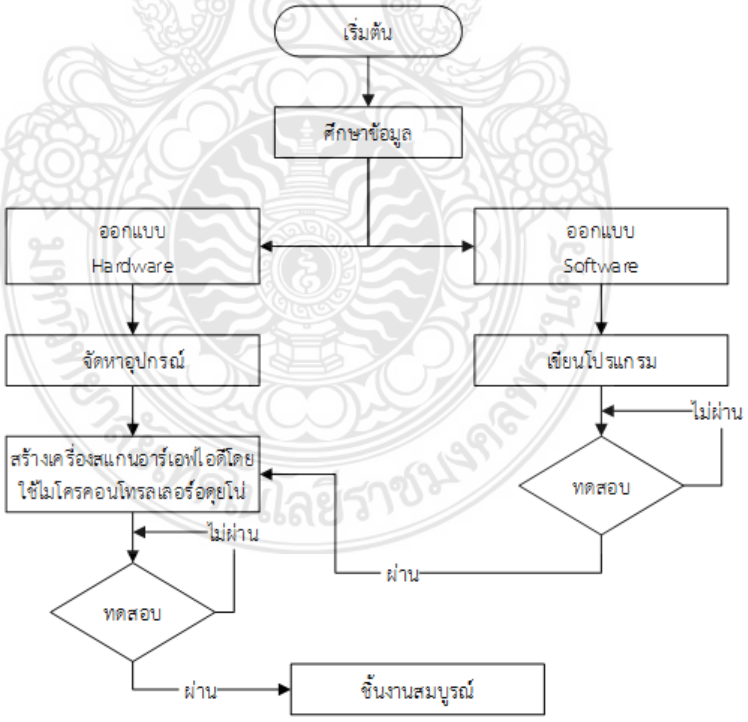
3.3 การออกแบบวงจรเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่

3.4 การออกแบบซอฟต์แวร์

3.5 การออกแบบฮาร์ดแวร์

3.1 แผนการดำเนินโครงการ

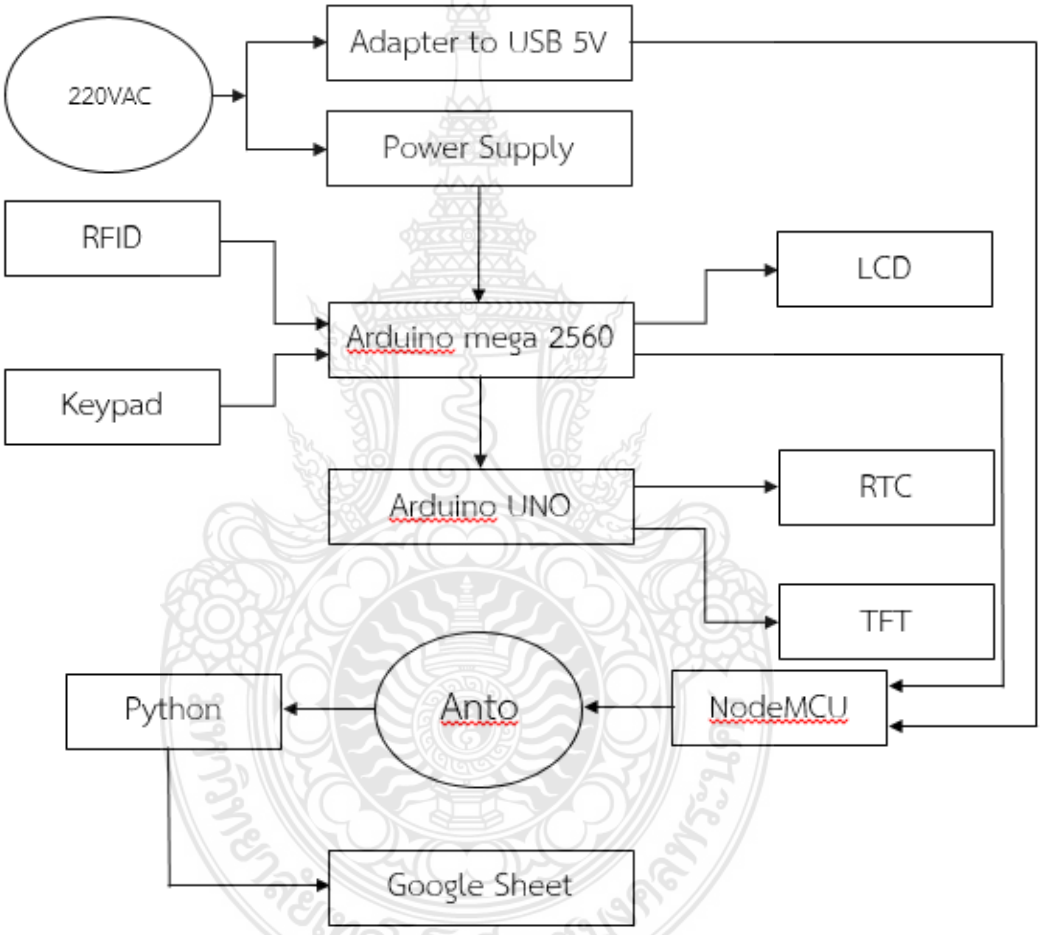
จากการศึกษาทฤษฎีที่เกี่ยวข้องสามารถเขียนแผนผังแสดงขั้นตอนและวิธีการดำเนินงานต่างๆ เป็นแผนภาพการทำงานได้ ดังภาพที่ 3.1



ภาพที่ 3.1 แผนการดำเนินงาน

จากภาพที่ 3.1 ศึกษาข้อมูลของการเขียนโปรแกรมและอุปกรณ์ที่ใช้ในโครงการนี้แล้วนำข้อมูลที่ศึกษาทั้งหมดมาสรุปและทำการออกแบบการทำงานทั้งหมดโดยแบ่งออกเป็น ซอฟต์แวร์ ฮาร์ดแวร์และการเขียนโปรแกรม หากเกิดข้อผิดพลาดให้ทำการแก้ไขและทดสอบใหม่ หากทดสอบผ่านทำการประกอบอุปกรณ์ทั้งหมดลงกล่อง จัดพิมพ์คู่มือการใช้งานและทำรายงานนำเสนอสอบเป็นการเสร็จสิ้นแผนการดำเนินโครงการ

3.2 บล็อกไดอะแกรมการทำงานของเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่

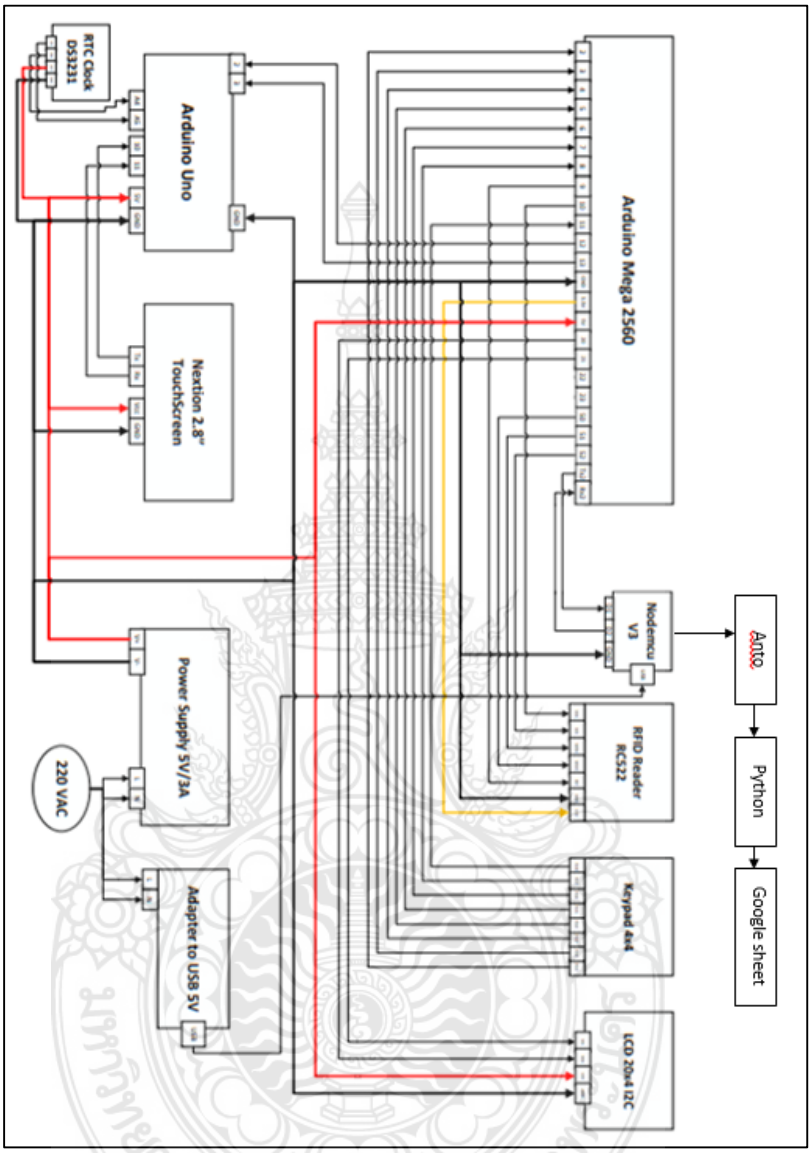


ภาพที่ 3.2 บล็อกไดอะแกรมการทำงานเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่

จากภาพที่ 3.2 บล็อกไดอะแกรมการทำงานเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ ระบบนี้จะใช้ CPU อดุยโน่เมกา)Arduino Mega (โดยระบบนี้จะสามารถตรวจเช็ครายชื่อ สถานการณ์การเข้าและออก สามารถรับอินพุตได้ 2 ช่องทาง คือ จาก

RFID Tag ซึ่งจะเป็นบัตร userนั้นก็จะทำการบันทึกไปใน EEPROM ที่อยู่ในอดุยโน่เมกาเพื่อจำว่าบุคคลที่ถือบัตรนี้เข้าห้องมาแล้ว กรณีที่ลืมบัตรจะมีการรับค่า อินพุตผ่านคีย์แพด ซึ่งคีย์แพต ของuser สามารถตั้งรหัสได้ 4 หลัก ไฟดับก็จะจำรหัสนี้ได้ แต่ว่าผู้ที่มีสิทธิพิเศษในการที่จะอนุญาตให้ยูเซอร์แต่ละคนมาทำการตั้งรหัสได้ก็คือแอดมิน โดยจะมีระบบของแอดมินที่แอดมินสามารถกดยืนยันว่าเป็นแอดมิน โดยทำการล็อกอินเข้าไปโดยใช้พาสเวิร์ด 4 หลักเหมือนกันเพื่อไปอีกชั้น กล่าวคือแอดมินมี 2 ชั้น พอล็อกอินชั้นที่ 1 เข้าไปแล้วก็สามารถเลือกได้ว่ายูเซอร์ 1 ใช้รหัสอะไรและยูเซอร์ 2 ใช้รหัสอะไร โดยแอดมินนี้จะทำได้ โดยระบบในการแสดงผลจะมี จอLiquid Crystal Display) LCD (เพื่อให้ทางยูเซอร์ได้เห็นว่าระบบนี้กำลังทำอยู่โหมดไหนจากนั้น Arduino Mega จะสื่อสารแบบดิจิตอลไปที่ Arduino UNO เพื่อมาสื่อสารกับ Real Time Clock) RTC (และ จอ Thin Film Transistor) TFT (ทำงานต่อ ในส่วนของ RTC จะทำหน้าที่แสดงเวลาปัจจุบันไว้ที่ หน้าจอ TFT และหน้าจอ TFT แสดงผลข้อความต้อนรับด้วยเช่น ถ้าสแกนเข้า จะมีคำว่า “Welcome” และเมื่อสแกน ออกจะมีคำว่า“ Thank You” เพื่อให้แสดงถึงระบบที่น่าสนใจมากขึ้น จะมีระบบเก็บข้อมูลรหัสของผู้เข้าออกโดยใช้ NodeMCU ในการดึงข้อมูลออกมาจาก Arduino Mega แล้วส่งค่าผ่านไวไฟ)Wifi(เพราะฉะนั้นระบบนี้จะเก็บข้อมูลของ ผู้เข้า-ออกผ่าน Cloud เพราะถ้าไฟดับหรือข้อมูลที่เป็นฮาร์ดไดฟ์พังก็จะไม่มีผลอะไร เพราะสามารถเก็บข้อมูลได้บน Cloud ซึ่งการจัดเก็บบน Cloud และไฟล์ที่จะจัดเก็บจะจัดเก็บอยู่ใน Google sheet หรือ Excel ในการแสดงผล NodeMCU จะไม่สามารถส่งข้อมูลไปที่ Google Sheet แต่จะใช้ Platform anto มาเป็นสื่อกลางที่จะนำข้อมูลจาก Nodemcu ขึ้นบน Cloud ไปที่ Anto ก่อน เนื่องจากว่าตัว Anto เป็น Platform ที่ถูกสร้างขึ้นมาเพื่อรองรับ NodeMCU และบอร์ดตระกูล Wifi แต่จะจัดเก็บไม่ได้ Anto จะทำหน้าที่บริการคือช่วยดึงข้อมูลซึ่งจำเป็นต้องใช้โปรแกรม Python ขึ้นมาซึ่งโปรแกรมนี้ต้องเขียนอยู่บนโน้ตบุ๊กหรือ PC โดยทำการดึงข้อมูลจาก Anto มา แล้วส่งไปที่Google sheet เพราะฉะนั้นการจะเก็บข้อมูลต้องเปิดไฟล์ Python ทิ้งเอาไว้

3.3 การออกแบบวงจรภาพรวมเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่



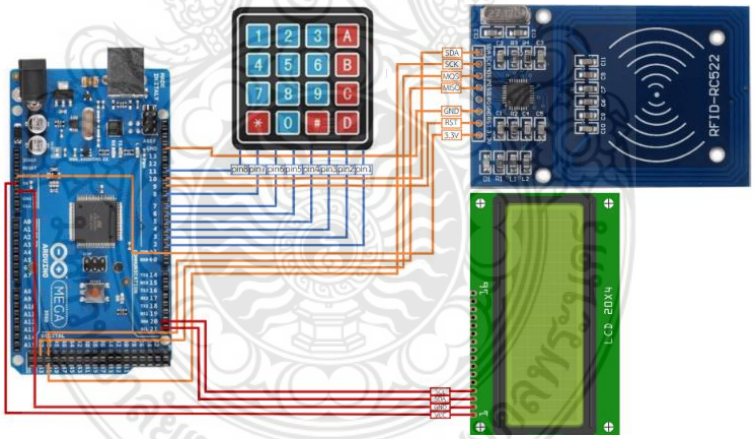
ภาพที่ 3.3 การออกแบบการทำงานของวงจรภาพรวม

จากภาพที่ 3.3 การออกแบบวงจรภาพรวมเครื่องสแกนอาร์เ อฟไอดีโดยใช้ไมโครคอนโทรลเลอร์ อดุยโน่เมื่อจ่ายกระแสแรงดันไฟ 220 VAC ไปยัง Power Supply และ Adapter เพื่อแปลงไปเป็น 5 VDC ป้อนให้กับบอร์ด Arduino mega 2560 และ NodeMCU ESP8266 หลังจากนั้นจะทำการรับอินพุต 2 ช่องทาง คือ RFID และ Keypad จะนำค่าที่รับเข้ามาไปเก็บไว้ใน EEPROM ของ Arduino mega 2560 และแสดงผลไปที่ จอLCD 20x4 จากนั้น Arduino mega 2560 จะให้ ขา 12 ,13 ส่งเอาต์พุตไปให้ ขา 2,3 ของ Arduino uno เพื่อให้ทราบว่ามีการเข้าหรือออก จะแทน LOW และ HIGH จากนั้น RTC จะเริ่มทำงานโดยแสดงค่าเวลาตามจริงโดยจะ

ดึงเวลาจากคอมพิวเตอร์ ส่วน หน้าจอ TFT เมื่อรับข้อมูลเข้ามาแล้วจะมีหน้าที่ แสดง “Welcome”และ “Thank you จะเชื่อมต่อกับบล็อกๆไดอะแกรมส่วนแรกและส่วนที่สอง เป็นการทำงานระหว่าง Arduino mega 2560 กับ NodeMCU ESP8266 ซึ่ง NodeMCU ESP8266 ได้ไปรับค่าที่อยู่ใน EEPROM ของ Arduino mega 2560 เพื่อส่งค่าผ่าน Wifi ไปเก็บที่ Cloud แต่จะใช้ Platform anto เป็นสื่อกลางที่จะนำข้อมูลจาก Node MCU ขึ้น Cloud เมื่อ anto ดึงข้อมูลจาก Node MCU มาแล้ว จะใช้โปรแกรม Python ซึ่งเขียนอยู่บน โน้ตบุ๊ก หรือ PC โดยทำการดึงข้อมูลจาก anto และส่งข้อมูลไปที่ Google sheet เพื่อแสดงผลรายงานการเข้า (1) การออก (0) RFID และ Keypad จำเป็นต้อง เปิดโปรแกรม Python ทิ้งไว้เสมอ
สามารถออกแบบวงจรการทำงานในแต่ละส่วน ได้ดังนี้

3.3.1 วงจรการทำงานในส่วนของการรับค่าอินพุต

Arduino mega 2560 จะทำการรับอินพุตสองช่องทาง คือ ช่องทางที่หนึ่ง RFID โดยต่อไฟเลี้ยง 3.3V และ GND จาก Arduino mega 2560 มาให้ RFID และในช่องทางที่สอง Keypad 4x4 โดยการต่อขา Pin1 ต่อเข้าขา 2 ของArduino mega 2560 ตามลำดับ จนถึงขา Pin8 ให้ต่อเข้าขาที่ ของ 11Arduino mega 2560 และนำไปแสดงผลใน LCD 20x4 I2C โดยมีการต่อขา VCC, GND ของ LCD 20x4 I2C เข้ากับขาไฟบวก 5V และ GND ของ Arduino mega 2560 และขาของ SDA, SCL ของ LCD 20x4 I2C ต่อเข้าขา 20, 21 ของ Arduino mega 2560 ตามลำดับ ดังภาพที่ 3.4

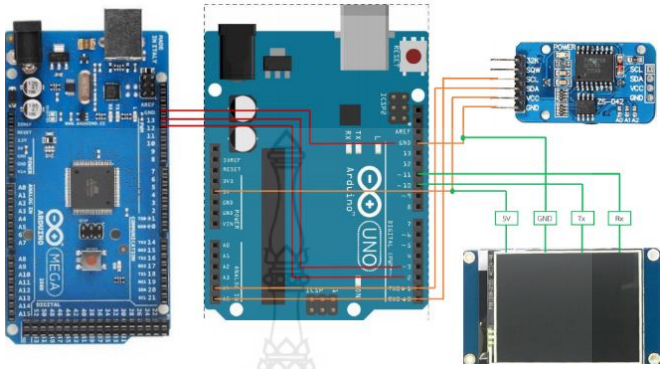


ภาพที่ 3.4 วงจรการทำงานในส่วนของการรับค่าอินพุต

3.3.2 วงจรการทำงานในส่วนของการส่งค่าเอาต์พุต

การต่อขา 12, 13 และ GND ของ Arduino mega 2560 ส่งเอาต์พุตไปให้ ขา 2,3 และ GND ของ Arduino Uno ตามลำดับ เพื่อให้ทราบว่ามีการเข้าหรือออก จากนั้น RTC จะเริ่มการทำงานโดยแสดงค่าเวลาตามจริง โดยต่อขา SDA, SCL, VCC และ GND เข้าที่ขา A4, A5, 5V และ GND ตามลำดับ ส่วนหน้าจอ TFT ทำการต่อไฟเลี้ยงขา 5V และ GND ของ Arduino Uno และรับ

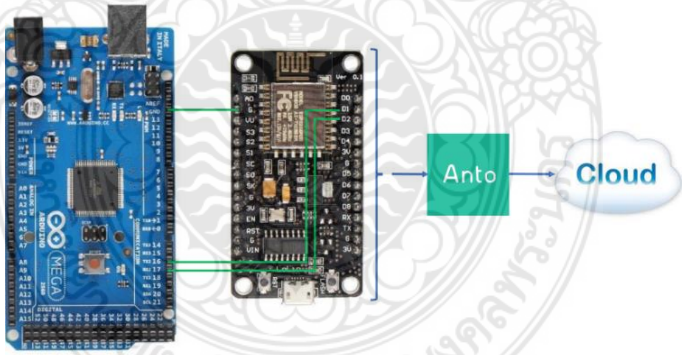
ข้อมูลจากขา ของ 11 Arduino Uno มายังขา Rx ของ TFT และส่งข้อมูลจากขา Tx ของ TFT มายังขา ของ 10Arduino Uno ดังภาพที่ 3.5



ภาพที่ 3.5 วงจรการทำงานในส่วนของการส่งค่าเอาต์พุต

3.3.3 วงจรการทำงานในการส่งค่าไปเก็บที่ Cloud

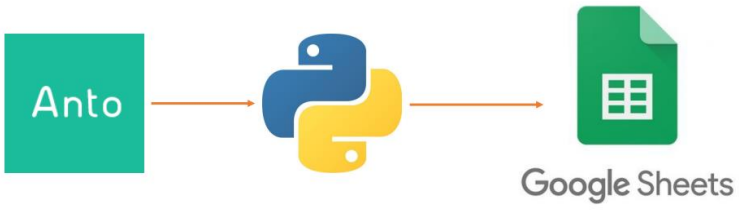
ส่วนนี้จะเชื่อมต่อกับวงจรส่วนแรกและวงจรส่วนที่สอง เป็นการทำงานระหว่าง Arduino mega 2560 กับ Nodemcu ESP8266 โดยต่อขา GND, Tx2 และ Rx2 ของ Arduino mega 2560 ไปยังขา GND, D1 และ D2 ของ NodeMCU V3 ตามลำดับ ซึ่งNodemcuESP8266ได้ไปรับค่าที่อยู่ใน EEPROM ขอArduino mega 2560 เพื่อส่งค่าผ่านWifi ไปเก็บที่ Cloudแต่จะใช้ Platform anto เป็นสื่อกลางที่จะนำข้อมูลจาก Nodemcuขึ้น Cloud ดังภาพที่ 3.6



ภาพที่ 3.6 วงจรการทำงานในการส่งค่าไปเก็บที่ Cloud

3.3.4 การแสดงผลบน Google sheet

ขั้นตอนนี้ เมื่อ anto ดึงข้อมูลจาก Nodemcu ESP8266 มาแล้ว จะใช้โปรแกรม Python ซึ่งเขียนอยู่บน โน๊ตบุ๊ค หรือ PC โดยทำการดึงข้อมูลจาก anto และส่งข้อมูลไปที่ Google sheet เพื่อแสดงผลรายงาน การเข้าและการออก จากการรับค่า ทางRFID และ Keypad จำเป็นต้องเปิดโปรแกรม Python ทิ้งไว้เสมอ ดังภาพที่ 3.7



ภาพที่ 3.7 การแสดงผลบนGoogle sheet

3.4 การออกแบบซอฟต์แวร์

3.4.1 Arduino mega 2560

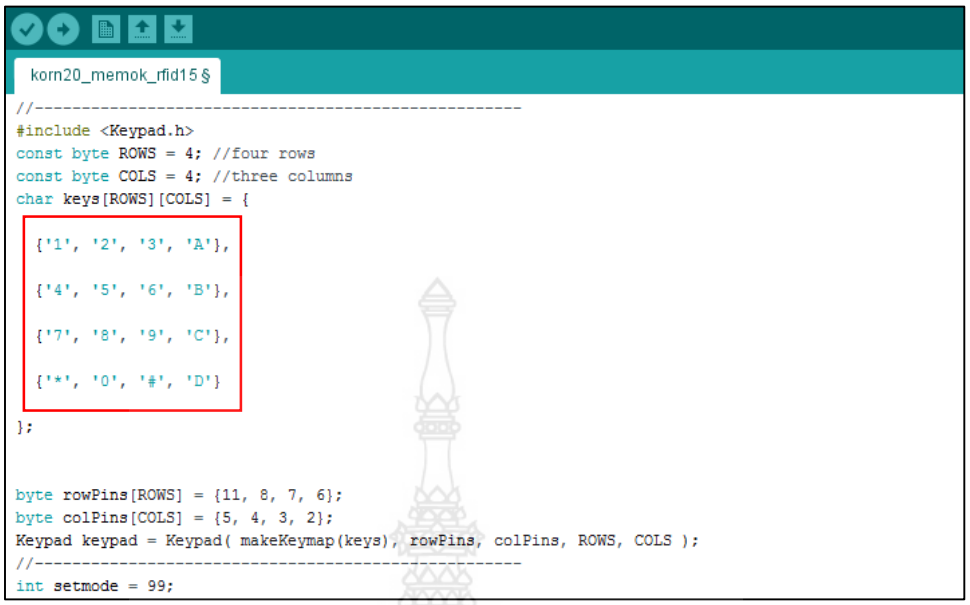
ขั้นตอนนี้จะเป็นการเขียนโค้ดโดยมีขั้นตอนการเขียนโค้ด ดังนี้

ขั้นตอนที่ 1. เป็นส่วนแรกของโค้ด ในส่วนของ RFID จะใช้ 2 ขานี้รวมกันตั้งชื่อเป็น RFID (SS_PIN, RST_PIN); ส่วนตัวแปล int sernum 0,1,2,3,4 เป็นตัวแปลที่เก็บค่าของตัวเลข RFID เวลาสแกนจะเป็นชุดตัวเลข 4 หลัก ต้องทำการสร้างตัวแปลตัวเลขไว้ ในส่วนของ LiquidCrystal_I2C lcd (0x27, 20, 4); 0x27 คือ Address ตั้งต้นของ LCD, 20 คือ จำนวนตัวอักษร, 4 คือ จำนวนแถว ดังภาพที่ 3.8

```
RFID rfid(SS_PIN, RST_PIN);
// Setup variables:
int serNum0;
int serNum1;
int serNum2;
int serNum3;
int serNum4;

String senum;
int xm;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
//--------------------------------------------------
```

ภาพที่ 3.8 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 2. เป็นในส่วนของ Keypad แบบ 4x4 การวางจะวางตามรูปแบบที่ตีกรอบไว้ ดังภาพที่จะเรียงขาที่ใช้คือ ขา 2,3,4,5,6,7,8,11 ต่อเรียงกัน ในส่วนของKeypad จะนำ rowPins, colPins, ROWS, COLS เรียงการสื่อสารนี้ว่า Keypad ดังภาพที่ 3.9



korn20_memok_rfid15 §

```
//---------------------------------------------------
#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {11, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
//---------------------------------------------------
int setmode = 99;
```

ภาพที่ 3.9 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 3. เป็นส่วนของ EEPROM เป็นหน่วยในการเก็บความจำ ยกตัวอย่างการเก็บUser เช่น User 1 รหัส 1111 ; 1 ตัวแรก จะอยู่ที่ 0, รหัส 1 ตัวที่ 2 จะอยู่ที่ 1 ,รหัส 1 ตัวที่ 3 จะอยู่ที่ 2 ,รหัส 1 ตัวที่ 4 จะอยู่ที่ 3, ส่วนสถานะ การเข้าออก จะอยู่ที่ 4. แสดงว่า User 1 ได้เก็บความจำเรียบร้อยแล้วหน่วยความจำนี้ยังสามารถเก็บ Use rตัวที่ 2 และคนอื่นได้ตามที่ต้องการ ดังภาพที่ 3.10

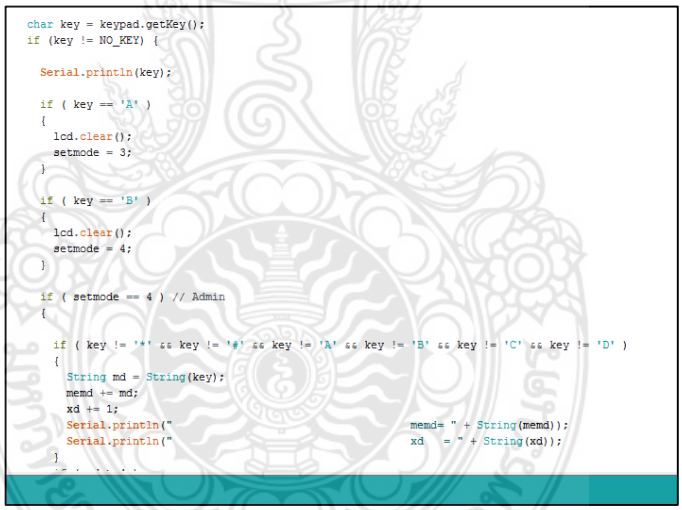
```
#include <EEPROM.h>

/** the current address in the EEPROM (i.e. which byte we're going to write to next) **/
int user1_adr1 = 0;
int user1_adr2 = 1;
int user1_adr3 = 2;
int user1_adr4 = 3;
int user1_adr5 = 4;

int user2_adr1 = 5;
int user2_adr2 = 6;
int user2_adr3 = 7;
int user2_adr4 = 8;
int user2_adr5 = 9;
```

ภาพที่ 3.10 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 4. เป็นส่วนที่ 2 ในการทำงานคือจะใช้ ขา 12,13 เป็น Output เพื่อที่จะส่งไปหา บอร์ด Arduino Uno ถ้า “IN” ขา 12 จะเป็น “LOW” ขา 13 เป็น HIGH จะแสดงคำว่า “Welcome” ถ้า “OUT” ขา 13 จะเป็น “LOW” ขา 12 จะเป็น HIGH จะโชว์ว่า “Thank you” ถ้าไม่มีอะไรเกิดขึ้นจะเป็น HIGH ทั้ง 2 ขา ดังภาพที่ 3.11

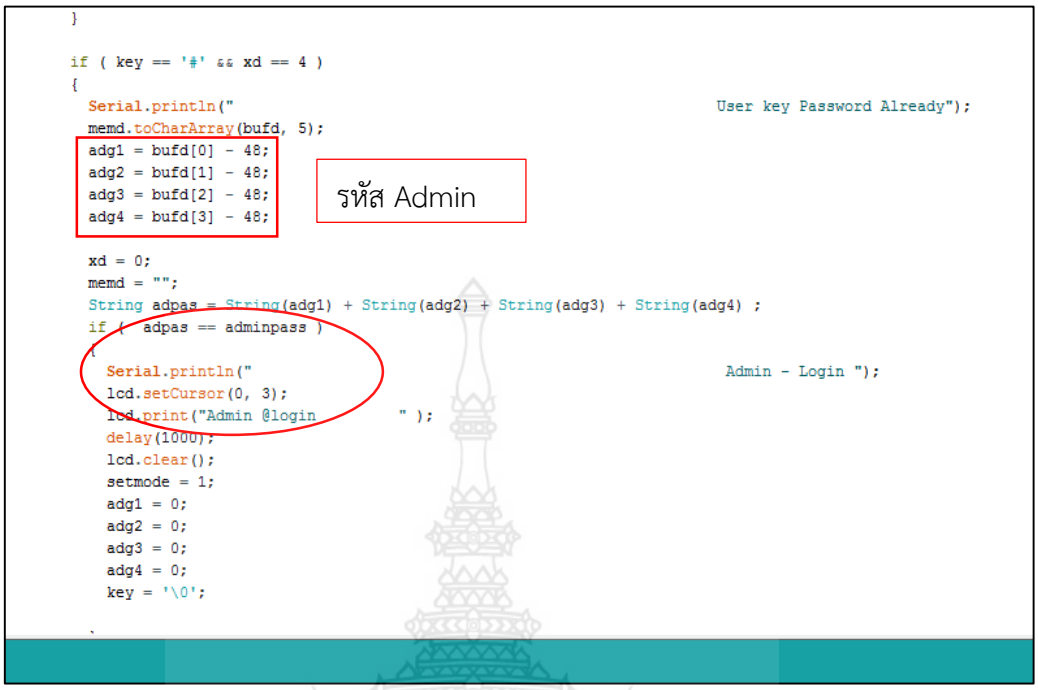
ภาพที่ 3.11 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 5. ส่วนที่ 3 จะเป็นส่วนของ Keypad จะแบ่งเป็น 2 ส่วน คือ Setmode 3 หรือ “A” จะเป็นในส่วนของ User ส่วน Setmode 4 หรือ “B” จะเป็นส่วนของ Admin ขั้นตอนนี้คือ ถ้า กด “B” มาแล้วถือว่าเป็นส่วนของ Admin จะไม่อนุญาตให้กด “* # A B C D”จะต้องทำการกดรหัส loginของ Admin เท่านั้น คือรหัส “1234” ดังภาพที่ 3.12



ภาพที่ 3.12 การเขียนโค้ดบอร์ด Arduino mega 2560

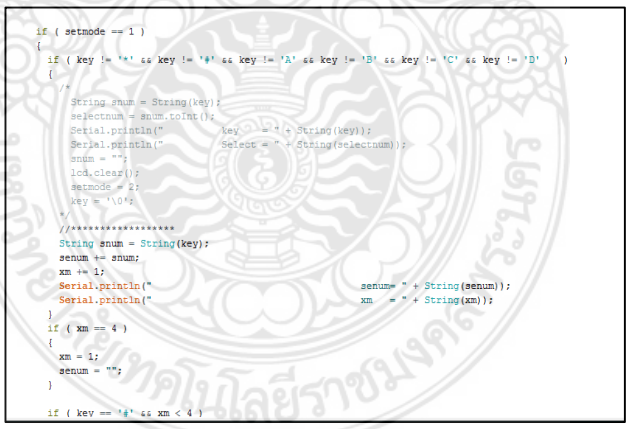
ขั้นตอนที่ 6. ส่วนที่ตีกรอบไว้เป็นในส่วนของรหัส Admin ถ้าตัวแปล adpass ตรงกับ adminpass ด้านบน ให้โชว์ว่า “Admin login” จากโหมด Setmode 4 จะเปลี่ยน เป็น Setmode 1 ทันที ดังภาพที่ 3.13




```
}

if ( key == '#' && xd == 4 )
{
  Serial.println("                                                User key Password Already");
  memd.toCharArray(bufd, 5);
  adg1 = bufd[0] - 48;
  adg2 = bufd[1] - 48;
  adg3 = bufd[2] - 48;
  adg4 = bufd[3] - 48;

  xd = 0;
  memd = "";
  String adpas = String(adg1) + String(adg2) + String(adg3) + String(adg4) ;
  if (  adpas == adminpass )
  {
   Serial.println("                                                 Admin - Login ");
   lcd.setCursor(0, 3);
   lcd.print("Admin @login          " );
   delay(1000);
   lcd.clear();
   setmode = 1;
   adg1 = 0;
   adg2 = 0;
   adg3 = 0;
   adg4 = 0;
   key = '\0';
```

ภาพที่ 3.13 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 7. ขั้นที่นี้มาที่ Setmode 1 ห้ามกด “* # A B C D” ทำการสร้างตัวแปลใหม่ เป็น string snum = syring (key); กดเลือก user 1-15 แล้วกด # จะขึ้นว่า selectnum ของใคร จาก setmode 1 จะเป็น setmode 2 แต่ถ้า กด * จะขึ้น “Admin logout”



```
if ( setmode == 1 )
{
 if ( key != '*' && key != '#' && key != 'A' && key != 'B' && key != 'C' && key != 'D'    )
 {
  /*
   String snum = String(key);
   selectnum = snum.toInt();
   Serial.println("          key     = " + String(key));
   Serial.println("          Select = " + String(selectnum));
   snum = "";
   lcd.clear();
   setmode = 2;
   key = '\0';
  */
  //******************
  String snum = String(key);
  senum += snum;
  xm += 1;
  Serial.println("                                    senum= " + String(senum));
  Serial.println("                                    xm   = " + String(xm));
 }
 if ( xm == 4 )
 {
  xm = 1;
  senum = "";
 }

 if ( key == '#' && xm < 4 )
```

ภาพที่ 3.14 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 8. มาที่ Setmode 2 จะเป็นการตั้งรหัสให้ User ห้ามกด “* # A B C D” กดรหัสUser เช่นเลือก selectnum 1 ตั้งรหัส 1111 แล้วกด # จะทำการ save รหัสของ selectnum 1 ไป ทำจนครบจำนวนที่กำหนดของกลุ่มตัวอย่าง พอครบจะทำการ write ลงไปในหน่วยความจำ EEPROM ดังภาพที่ 3.15

```
}
if ( i > 4 )
{
  i = 0;
  mid = "";
}
if ( key == '*' && i <= 4 )
{
  i = 0;
  memid = "";
}
if ( key == '#' && i < 4 )
{
  Serial.println("                                        Please key password more !!");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Plse.KeyPasswordMore");
  delay(1000);
  lcd.clear();
}

if ( key == '#' && i == 4 )
{
  //-------------------------------------------------------------------------------------
  Serial.println("                                        Remember User" + String(selectnum));

  if ( selectnum == 1)
  {
```

ภาพที่ 3.15 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 9. ข้อมูลจะถูกส่งไปทีจอ LCD ให้ทำการ Remember จอLCD จะแสดง SuccessFull จากนั้นจะไปที่ Setmode 1 จากนั้นก็ตั้งรหัสจนคทุกคน ดังภาพที่ 3.16

```
    //
    //        Serial.println( String(digit1) + String(digit2) + String(digit3) + String(digit4));
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("RememberUser:" + String(selectnum));
    lcd.setCursor(0, 1);
    lcd.print("SuccessFull!!!");
    delay(3000);
    lcd.clear();
    i = 0;
    memid = "";
    setmode = 1;
  }


}// admin mem
```

ภาพที่ 3.16 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 10. ถ้ากดรหัสถูก User 1 จะทำการ login แล้วส่ง สัญญาณ LOW-HIGH ไปที่บอร์ด Arduino UNO แสดงสถานะเป็นการเข้า เมื่อกดรหัส User 1อีกครั้ง จะทำการ logout จะแสดงสถานะเป็น HIGH-LOW คือสถานะเป็นการออก ดังภาพที่ 3.17

```
}


if ( EEPROM.read(user1_adr1) == dg1 && EEPROM.read(user1_adr2) == dg2 && EEPROM.read(user1_adr3) == dg3 && EEPROM.read(user1_adr4) == dg4 && ck == 1 )
{
  status1 += 1;
  if ( status1 == 1 )
  {
    Serial.println("                                                User1 - Login !!!");
    Serial2.println("1=1");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("User1 - login" );
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    door_open ();
    delay(1000);
    door_lock ();
    digitalWrite(12, HIGH);
    digitalWrite(13, HIGH);
    lcd.clear();
  }
  if (status1 == 2)
  {
    status1 = 0;
  }
  if ( status1 == 0 )
  {
    Serial.println("                                                User1 - Logout !!!");
```

ภาพที่ 3.17 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 11. ในส่วนของ RFID เมื่อนำการ์ดมาแตะ สัญญาณ RFID จะอ่านรหัสจากการ์ด เจอ sernum ที่ตั้งไว้ซึ่ง มี 5 หลัก คือ 0 1 2 3 4

```
if (rfid.isCard()) {
  if (rfid.readCardSerial()) {
    if (rfid.serNum[0] != serNum0
        && rfid.serNum[1] != serNum1
        && rfid.serNum[2] != serNum2
        && rfid.serNum[3] != serNum3
        && rfid.serNum[4] != serNum4
       ) {

      Serial.println(" ");
      Serial.println("Card found");
      serNum0 = rfid.serNum[0];
      serNum1 = rfid.serNum[1];
      serNum2 = rfid.serNum[2];
      serNum3 = rfid.serNum[3];
      serNum4 = rfid.serNum[4];
```

```
//Serial.println(" ");
Serial.println("Cardnumber:");
Serial.print("Dec: ");
Serial.print(rfid.serNum[0], DEC);
Serial.print(", ");
Serial.print(rfid.serNum[1], DEC);
Serial.print(", ");
Serial.print(rfid.serNum[2], DEC);
Serial.print(", ");
Serial.print(rfid.serNum[3], DEC);
Serial.print(", ");
Serial.print(rfid.serNum[4], DEC);
Serial.println(" ");
//1Dec: 51, 94, 124, 3, 18
//2Dec: 19, 46, 108, 3, 82
//3Dec: 183, 55, 61, 78, 243
//4Dec: 23, 56, 202, 77, 168
```

ภาพที่ 3.18 การเขียนโค้ดบอร์ด Arduino mega 2560

ขั้นตอนที่ 12.ในการเช็ครหัส ถ้ารหัสที่ RFID.sernum เป็นตามในกรอบสีแดงดังภาพที่ 3.14 แสดงว่า status จะเป็น 1 (user login) ถ้าstatusจะเป็น (user logout) จากนั้น serial 2 จะไปสื่อสารกับ Nodemcu ส่งไปที่ Anto ว่ามีรหัสคนที่ 1 เข้าและออก ดังภาพที่ 3.19

```
if ( rfid.serNum[0] == 51 && rfid.serNum[1] == 94 && rfid.serNum[2] == 124 && rfid.serNum[3] == 3 && rfid.serNum[4] == 18 )
{
  status1 += 1;
  if ( status1 == 1 )
  {
    Serial.println("                                                User1 - Login !!!");
    Serial2.println("1=1");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("User1 - login" );

    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    door_open ();
    delay(1000);
    door_lock ();
    digitalWrite(12, HIGH);
    digitalWrite(13, HIGH);
    lcd.clear();

  }
  if (status1 == 2)
  {
    status1 = 0;
  }
  if ( status1 == 0 )
  {
    Serial.println("                                                User1 - Logout !!!");
```

ภาพที่ 3.19 การเขียนโค้ดบอร์ด Arduino mega 2560

3.4.2 Arduino UNO

มีขั้นตอนการเขียนโค้ด ดังนี้

ขั้นตอนที่ 1 จะสื่อสารกับ Arduino mega แบบดิจิตอล จะสื่อสารกับ RTC และ TFT ในส่วนที่ 1 การประกาศตัวแปล Wire เกี่ยวข้องกับอักษรจะคู่กับ RTC ต่อมา TFT เนื่องจากบอร์ด Arduino UNO จะไม่มีขา RX2 ,TX2 จะเป็นเหมือน Nodemcu ต้องใช้ Softwareserial เข้ามาตั้งชื่อขา โดยการให้ขา 10 เป็น RX ขา 11 เป็น TX ขา TX ของ TFT จะมาต่อ ขา RX ของ Arduino UNO โดยการต่อไขว้ ดังภาพที่ 3.20

```
#include <Wire.h>
#include <SPI.h>  // not used here, but needed to prevent a RTClib compile error
#include <RTClib.h>
RTC_DS3231 RTC;
#include <SoftwareSerial.h>
#include <Nextion.h>
//--------------------------------  Communication with TouchScreen
SoftwareSerial nextion(10, 11);// Nextion TX to pin 10 and RX to pin 11 of Arduino
Nextion myNextion(nextion, 9600); //create a Nextion object named myNextion using the nextion serial port @ 9600bps
```

ภาพที่ 3.20 การเขียนโค้ดบอร์ด Arduino UNO

ขั้นตอนที่ 2 ขา 12,13 ของ Arduino mega จะมาต่อขา 2,3 ของ Arduino uno โดยขา 12 จะทำการต่อ ขา 2 และขา 13 จะทำกาต่อกับ ขา 3 ให้เป็น INPUT_PULLUP เปรียบเสมือน HIGH ตลอดเวลา จากนั้นก็จะเริ่มการทำงานของTFT จะเริ่มการทำงานของ RTC ดังภาพที่ 3.21

```
void setup() {
  Serial.begin(9600);


  pinMode(2, INPUT_PULLUP); // Setup for leads off detection LO +
  pinMode(3, INPUT_PULLUP); // Setup for leads off detection LO -


  myNextion.init();

  Wire.begin();
  RTC.begin();
```

ภาพที่ 3.21 การเขียนโค้ดบอร์ด Arduino UNO

ขั้นตอนที่ 3 ในโค้ดส่วน DateTime now = RTC.now คือการเริ่มอ่านเวลา วัน เดือน ปี ชั่วโมง นาที วินาที ถ้าเข้า LOW HIGH Serial.println “IN” Nextion.setcomponenttext “t2” “Welcome” ถ้าออก HIGH LOW Serial.println “OUT” Nextion.setcomponenttext “t2” “Thank you” ดังภาพที่ 3.22

```
String nows = String(now.hour()) + ":" + String(now.minute());

if ( digitalRead(2) == LOW && digitalRead(3) == HIGH  )
{
  Serial.println("  IN ");
  myNextion.setComponentText("t2", "Welcome");
  delay(5000);

}

if ( digitalRead(2) == HIGH && digitalRead(3) == HIGH  )
{
  Serial.println("  Normal ");
  myNextion.setComponentText("t2", nows);
  delay(1000);

}

if ( digitalRead(2) == HIGH && digitalRead(3) == LOW  )
{
  Serial.println("  OUT ");
  myNextion.setComponentText("t2", "Thank you.");
  delay(5000);
}
```

ภาพที่ 3.22 การเขียนโค้ดบอร์ด Arduino UNO

3.4.3 Node MCU

ขั้นตอนที่ 1 ในส่วนของ Node MCU จะต่อกับ Arduino mega แบบ Serial ในส่วนแรกนี้จะเลือกใช้ SoftwareSerial จะให้ D1 เป็น RX ,TX2 จะไปต่อกับ D1 ที่เป็น RX ของNode ส่วน TX ก็คือ D2 จะต่อเข้ากับ RX2 ของ Arduino mega ต่อไขว้กัน ส่วนNodemcu จะส่งข้อมูลไปที่ Anto ซึ่งเป็น Platform ตัวกลาง จึงต้องมี ไลฟ์บารี่ Anto จากนั้น Nodemcu จะต้องเชื่อมต่อ Wifi เพราะหน้าที่ของ Nodemcu คือรับข้อมูลจาก Arduino mega แล้วส่งไปที่ Anto เท่านั้น ดังภาพที่ 3.23

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(D1, D2); // RX, TX
#include <AntoIO.h>
const char *ssid = "iPhone Kornkamol";
const char *pass = "kornkamol";
const char *user = "rmutp";
const char *token = "HTSci920ghYHgpjAB3d2ZFQAMADSq15N32OQqj4h";
const char *thing = "rfid_server";
AntoIO anto(user, token, thing);

//---------------------------------------------- -

int id;
int datamega;
char data[8];
```

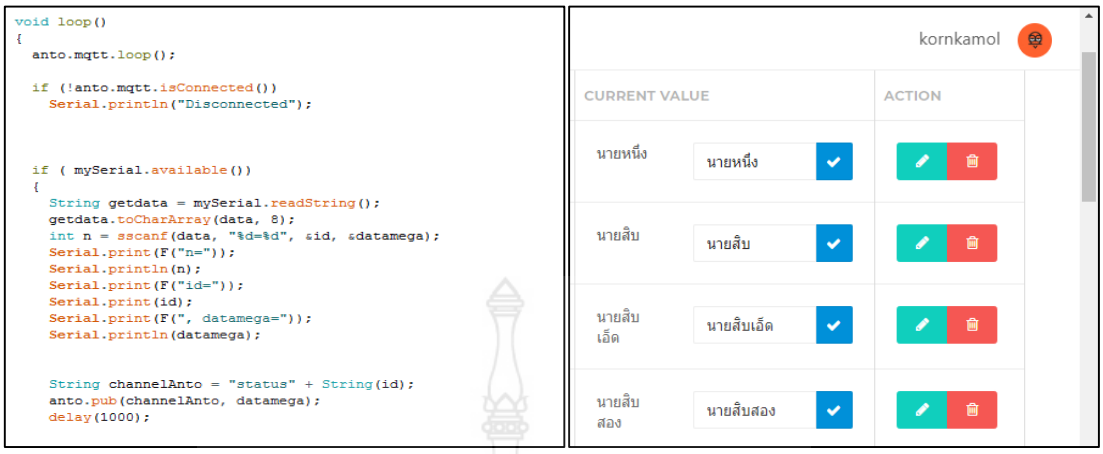
ภาพที่ 3.23 การเขียนโค้ดบอร์ด Node MCU

ขั้นตอนที่ 2 ในส่วนที่มีกรอบสีแดงครอบไว้ เป็นส่วนที่ แสดงสถานะไฟของ Nodemcu จะปิดไว้ก่อน ส่วนในกรอบสีน้ำเงิน จะเป็นการ Connected ไวไฟ ถ้าเชื่อมต่อได้จะขึ้นคำว่า “done” ไฟสีฟ้าใน Nodemcu จะสว่างขึ้น digitalWrite (D4, LOW); ดังภาพที่ 3.24

```
void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600);
  delay(10);
  pinMode(D4, OUTPUT);
  digitalWrite(D4, HIGH);

  Serial.println();
  Serial.println();
  Serial.print("Anto library version: ");
  Serial.println(anto.getVersion());

  Serial.print("\nTrying to connect ");
  Serial.print(ssid);
  Serial.println("...");

  anto.begin(ssid, pass, messageReceived);
  Serial.println("\nConnected Anto done");

  // anto.sub("mode");

  digitalWrite(D4, LOW);

} // end setup
```

ภาพที่ 3.24 การเขียนโค้ดบอร์ด Node MCU

ขั้นตอนที่ 3 จะเป็นส่วนของ Anto ถ้าเชื่อมต่อได้จะขึ้นว่า “Disconnected Anto” จะมีรายชื่อที่สร้างไว้ ในส่วนนี้จะแทน ID และ สถานะเข้าออกคือ 1 กับ 0 (data, "%d=%d", &id, &datamega); ดังภาพที่ 3.25



```
void loop()
{
  anto.mqtt.loop();

  if (!anto.mqtt.isConnected())
    Serial.println("Disconnected");


  if ( mySerial.available())
  {
    String getdata = mySerial.readString();
    getdata.toCharArray(data, 8);
    int n = sscanf(data, "%d=%d", &id, &datamega);
    Serial.print(F("n="));
    Serial.println(n);
    Serial.print(F("id="));
    Serial.print(id);
    Serial.print(F(", datamega="));
    Serial.println(datamega);


    String channelAnto = "status" + String(id);
    anto.pub(channelAnto, datamega);
    delay(1000);
```


ภาพที่ 3.25 การเขียนโค้ดบอร์ด Node MCU

3.4.4 Python

ขั้นตอนที่ 1 เป็นการเลือกใช้ antolib จะให้โปรแกรมนี้ส่งข้อมูลไปหา Anto ได้ต้องมี Library ส่วน Library ที่จะส่งข้อมูลไปที่ Google Sheet จะมี 3 ตัวหลัก ๆ คือ gspread ย่อมาจาก Google spread Sheet ทำหน้าที่ในการสร้างเซล ต่อมาคือ oauth2client.service_account เป็น Library สำหรับการขอไฟล์ยืนยันระหว่างเครือข่าย และ pprint คือการแสดงค่า ส่วนลิงค์ที่จะส่งข้อมูลไปให้ Google อยู่ในกรอบสีแดงดังภาพที่ 3.21 พอส่งข้อมูลเข้าไปแล้วจะเป็นการยืนยันตัวโดยใช้ file.json แล้วแนบไฟล์โดยนำลิงค์มารวมกัน (scope) จากนั้นจะตั้งชื่อว่า creds (gspread) เป็นการสร้าง sheet จากนั้นจะรวมข้อมูลกับลิงค์ กลายเป็น client และนำไปสร้างไฟล์ Datalogger โดยสร้างใน sheet 1 ดังภาพที่ 3.26

```
Kornpython1.py - C:\Users\Administrator\Desktop\Kornpython1.py (3.7.4)
File  Edit  Format  Run  Options  Window  Help

import antolib
import requests
import datetime
import time
#-------------------------------------------------------------------------
import gspread
from oauth2client.service_account import ServiceAccountCredentials
import pprint
x=0
name=""
scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']
creds = ServiceAccountCredentials.from_json_keyfile_name('credskorn.json',scope)
client = gspread.authorize(creds)
sheet1 = client.open('datalogger').sheet1


pp = pprint.PrettyPrinter()
```

ภาพที่ 3.26 การเขียนโค้ด anto

ขั้นตอนที่ 2 เป็นการรับข้อมูลจาก anto หรือเรียกว่าการ sub ในกรอบสีน้ำเงินดังภาพที่ 3.22 จะแทน 1 กับ 0 ส่วน Nodemcu จะเป็นตัวส่งข้อมูลไปบอกสถานะ1และ0 ส่วนในกรอบสีแดง จะแทน ชื่อของกลุ่มตัวอย่างทั้ง 15 คน python จะเป็นตัวดึงข้อมูลชื่อมาจาก anto ดังภาพที่ 3.27

ภาพที่ 3.27 การเขียนโค้ด anto

ขั้นตอนที่ 3 เป็นการ Connet Internet ในส่วน loopfunction คือ เป็นการแสดงให้ดูว่าชื่อใคร สถานะ 1 หรือ 0 ทุกๆ 1 วินาที ดังภาพที่ 3.28

```
def setup():
    anto.mqtt.onConnected(connectedCB)
    anto.mqtt.onData(dataCB)
    anto.mqtt.connect()


def myLoopFunction():


    global name_server1
    global name_server2
    global name_server3
    global name_server4
    global name_server5
    global name_server6
    global name_server7
    global name_server8
    global name_server9
    global name_server10
    global name_server11
    global name_server12
    global name_server13
    global name_server14
    global name_server15

    global status1
    global status2
    global status3
    global status4
    global status5
    global status6
    global status7
    global status8
    global status9
    global status10
    global status11
    global status12
    global status13
    global status14
    global status15
```

```
print('                user1 : ',name_server1,' status : ',status1)
print('                user2 : ',name_server2,' status : ',status2)
print('                user3 : ',name_server3,' status : ',status3)
print('                user4 : ',name_server4,' status : ',status4)
print('                user5 : ',name_server5,' status : ',status5)
print('                user6 : ',name_server6,' status : ',status6)
print('                user7 : ',name_server7,' status : ',status7)
print('                user8 : ',name_server8,' status : ',status8)
print('                user9 : ',name_server9,' status : ',status9)
print('                user10 : ',name_server10,' status : ',status10)
print('                user11 : ',name_server11,' status : ',status11)
print('                user12 : ',name_server12,' status : ',status12)
print('                user13 : ',name_server13,' status : ',status13)
print('                user14 : ',name_server14,' status : ',status14)
print('                user15 : ',name_server15,' status : ',status15)

print('')
print('')
time.sleep(1)
```

ภาพที่ 3.28 การเขียนโค้ด anto

ขั้นตอนที่ 4 ในส่วนนี้จะแสดงให้เห็น เช่น status 1 เป็น 1 now จะแสดงเวลาปัจจุบัน ในส่วนกรอบสีแดงดังภาพที่ 3.29 คือการส่งข้อมูลไปให้ Google sheet ส่วน global x แทนบรรทัดที่ข้อมูลต้องไปแสดงบน Google sheet

```
if(channel == 'status1'):
    global status1
    status1=msg.decode("utf-8")
    print("                                    status1= ",status1)
    global x
    now=datetime.datetime.now()
    x+=1
    sheet1.update_cell(x,1,name_server1)
    sheet1.update_cell(x,2,status1)
    sheet1.update_cell(x,3,str(now))

if(channel == 'status2'):
    global status2
    status2=msg.decode("utf-8")
    print("                                    status2= ",status2)
    now=datetime.datetime.now()
    x+=1
    sheet1.update_cell(x,1,name_server2)
    sheet1.update_cell(x,2,status2)
    sheet1.update_cell(x,3,str(now))
```

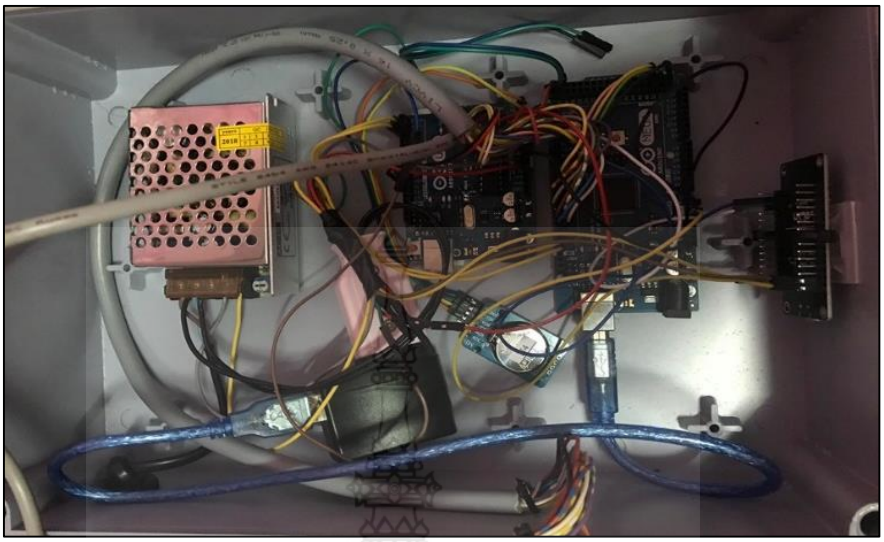
ภาพที่ 3.29 การเขียนโค้ด anto

3.5 การออกแบบฮาร์ดแวร์

การออกแบบกล่องด้านนอกของเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ ได้ทำการออกแบบด้านหน้าของกล่องให้มี หน้าจอ TFT เพื่อใช้สำหรับแสดงข้อความ “Welcome”เมื่อมีการเข้า และ “Thank you” เมื่อมีการออก ส่วน RFID จะไว้ใช้สแกนบัตร ส่วนหน้าจอ LCD ใช้สำหรับแสดงการทำงานของ Admin และ User และ Keypad ไว้สำหรับการใช้เครื่องแบบ Manual ดังภาพที่ 3.30

ภาพที่ 3.30 เครื่องสแกนด้านนอกกล่อง

ภายในกล่องประกอบไปด้วยบอร์ด Arduino mega, บอร์ด Arduino UNO, Nodemcu ESP8266, RTC, Power supply , Adapter และสายไฟที่ทำหน้าที่ทำการเชื่อมต่อสายไฟให้แต่ละบอร์ดและอุปกรณ์ต่างๆ ดังภาพที่ 3.31



ภาพที่ 3.31 อุปกรณ์ภายในกล่อง

ภาพที่ 3.32 ภายนอกกล่องจะเป็นชื่อโครงการและผู้จัดทำ

บทที่ 4
ผลการดำเนินงาน

บทนี้เป็นการทดสอบการทำงานของเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ จะทำให้ทราบถึงประสิทธิภาพของเครื่องตลอดจนปัญหาที่เกิดขึ้นในการดำเนินงาน ทำให้สามารถนำไปแก้ไขปรับปรุงและพัฒนาโครงการให้ดีขึ้น การทดสอบสามารถแบ่งออกได้ ดังนี้

4.1 ขั้นตอนการทดสอบ

4.2 สรุปผลการทดสอบ

4.1 ขั้นตอนการทดสอบ

4.1.1 ขั้นตอนการใช้เครื่องสแกนอาร์เอฟไอดีโดยการตั้งรหัสของ Admin

ภาพที่ 4.1 เครื่องสแกน อาร์เอฟไอดี

มีรายละเอียดดังนี้

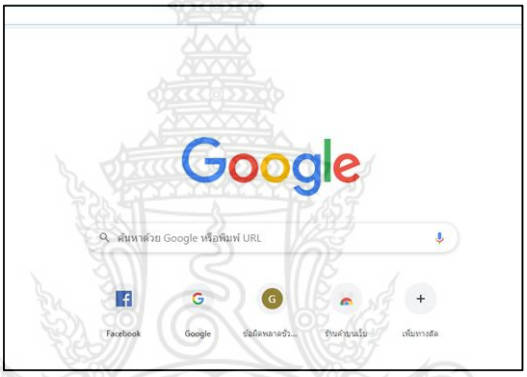
1. แอดมินทำการ Login โดยการกดปุ่ม B ตามด้วยรหัส 1234 แล้ว กดปุ่ม #

2. จากนั้นแอดมินจะทำการลงรหัส User ของคนที่จะเข้าห้องนี้ จากกลุ่มตัวอย่าง 15 คน โดยตั้งรหัส User 1-15 ให้ลงรหัสครั้งละ 1 User เช่น ลงรหัส User ที่ 1 แล้วกดปุ่ม # จากนั้นกดรหัส 4 หลัก ที่ แอดมินทำการตั้งให้ เช่น 1111 แล้วกดปุ่ม# จะได้รหัสเรียบร้อย ทำแบบนี้ให้ครบจำนวน 15 คน

3. แอดมินทำการ Logout โดยกดปุ่ม * หน้าจอ LCD จะแสดงข้อความ “Admin Logout” จากนั้น User ที่จะเข้าจะล็อกอินได้ โดยการกดปุ่ม A แล้วกดรหัส 4 หลัก ที่แอดมินตั้งรหัสไว้ให้

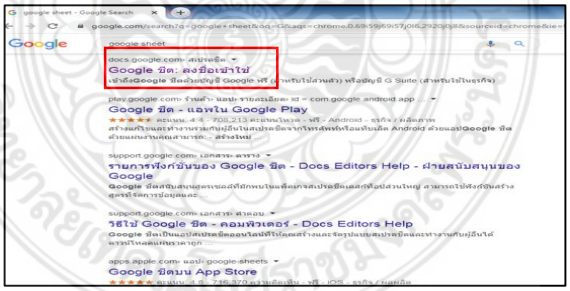
4.1.2 ขั้นตอนการเข้าใช้ Google sheet

1. เข้าไปที่ หน้าต่าง Google แล้วพิมพ์คำว่า Google Sheet ดังภาพที่ 4.2



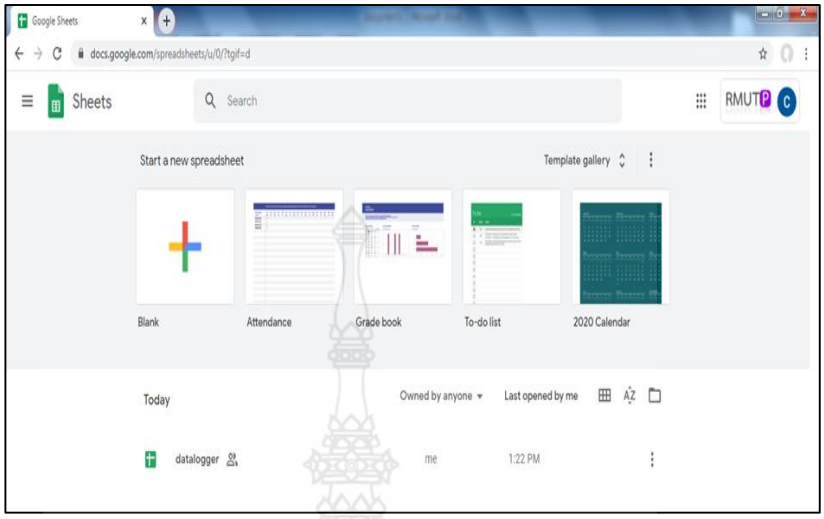
ภาพที่ 4.2 หน้าต่าง Google

2. คลิกที่ Google sheet ลงชื่อเข้าใช้ ดังภาพที่4.3

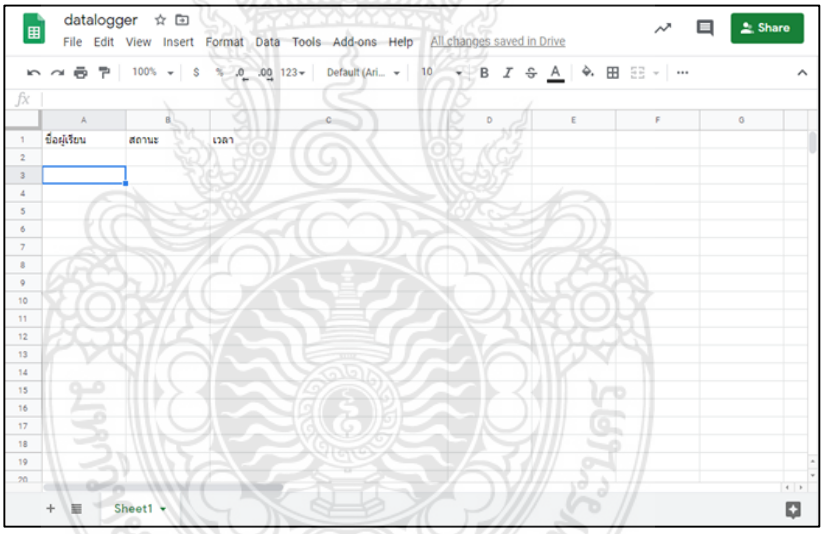


ภาพที่ 4.3 ค้นหา Google sheet

3. คลิกที่ Datalogger ดังภาพที่ 4.4

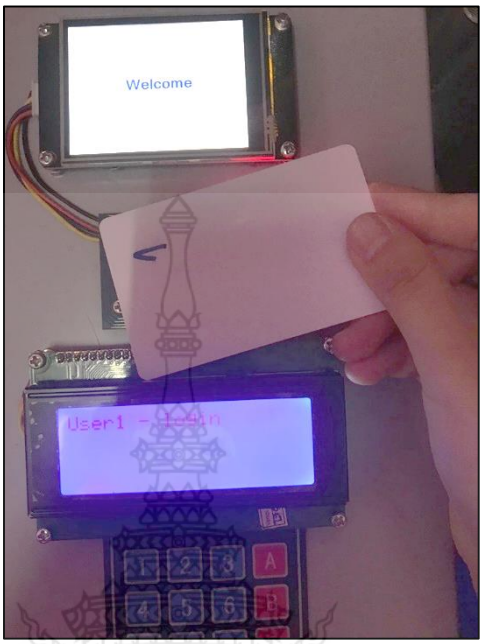


4. เข้าสู่หน้าของ Google Sheet ดังภาพที่ 4.5



ภาพที่ 4.5 ตาราง Google sheet

4.1.3 ขั้นตอนการสแกนคีย์การ์ดเข้า

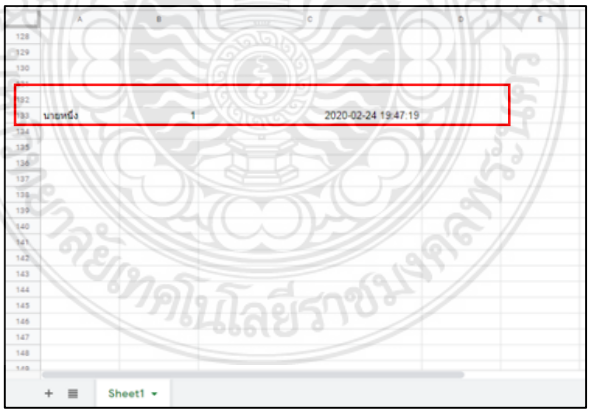


ภาพที่ 4.6 การสแกนคีย์การ์ดในการเข้า

มีรายละเอียดดังนี้

1. นำคีย์การ์ดที่มีรหัสโค้ดแล้ว ไปสแกนที่ตัว RFID จากนั้นหน้าจอTFT จะแสดงคำว่า “Welcome” ส่วนหน้าจอ LCD จะแสดงคำว่า “User 1 - login”

2. ข้อมูลจะไปปรากฏที่ Google sheet ว่า นายหนึ่ง มีสถานะเป็น 1 คือการเข้า Google sheet จะแสดงวันเดือนปีและเวลาที่เข้าให้ทราบในตารางดังภาพที่ 4.7



ภาพที่ 4.7 ผลตารางการสแกนเข้าโดยคีย์การ์ด

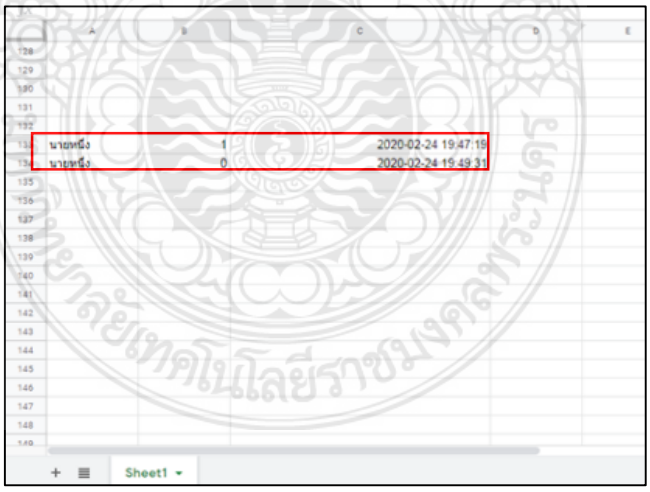
4.1.4 ขั้นตอนการสแกนคีย์การออก

ภาพที่ 4.8 การสแกนคีย์การ์ดในการออก

มีรายละเอียดดังนี้

1. นำคีย์การ์ดที่มีรหัสโค้ดแล้ว ไปสแกนที่ตัว RFID จากนั้นหน้าจอTFT จะแสดงคำว่า “Thank you” ส่วนหน้าจอ LCD จะแสดงคำว่า “User 1 - logot”

2. ข้อมูลจะไปปรากฏที่ Google sheet ว่า นายหนึ่ง มีสถานะเป็น 0 คือการออก Google sheet จะแสดงวันเดือนปีและเวลาที่ออกให้ทราบในตารางดังภาพที่ 4.9



ภาพที่ 4.9 ผลตารางการสแกนเข้า (1) สแกนออก (0) โดยคีย์การ์ด

4.1.5 ขั้นตอนการกดรหัสเข้าโดย Manual

(ก) กดปุ่ม A

(ข) กดรหัส 4 หลัก เช่น รหัส 1111

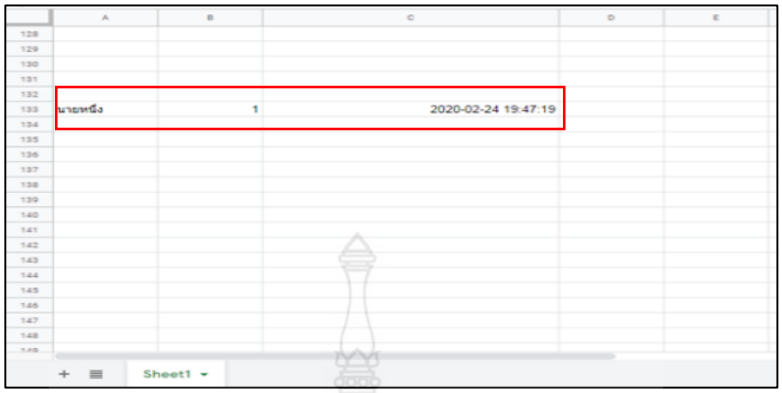
(ค) กดปุ่ม # หน้าจอ LCD จะแสดงคำว่า “User 1- login”
ส่วนจอ TFT จะแสดงคำว่า “Welcome”

ภาพที่ 4.10 การกดรหัสเข้าด้วย Manual

มีรายละเอียดดังนี้

1. กดปุ่ม A ดังภาพที่ 4.10 (ก.) ตามด้วยรหัส 4 หลัก เช่น รหัส 1111 ดังภาพที่ 4.10 (ข.) แล้วกดปุ่ม # หน้าจอ LCD จะแสดงคำว่า “User 1- login” ส่วนจอ TFT จะแสดงคำว่า “Welcome” ดังภาพที่ 4.10 (ค.)

2. ข้อมูลจะไปปรากฏที่ Google sheet ว่า นายหนึ่ง มีสถานะเป็น 1 คือการเข้า Google sheet จะแสดงวันเดือนปีและเวลาที่เข้าให้ทราบดังภาพที่ 4.11



ภาพที่ 4.11 ผลตารางการแสดงการเข้าด้วย Manual

4.1.5 ขั้นตอนการกดรหัสออกแบบManual

(ก) กดปุ่ม A

(ข) กดรหัส 4 หลัก เช่น รหัส 1111

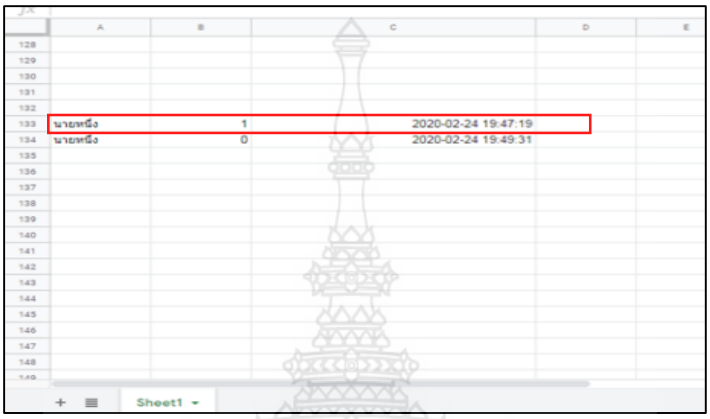
(ค) กดปุ่ม # หน้าจอของ LCD จะแสดงคำว่า“User 1- logout”
ส่วนจอ TFT จะแสดงคำว่า “Thank you”

ภาพที่ 4.12 การกดรหัสออกด้วยManual

มีรายละเอียดดังนี้

1. กดปุ่ม A 4.12 (ก.) ตามด้วยรหัส 4 หลัก เช่น รหัส 1111 ดังภาพที่ 4.12 (ข.) แล้วกดปุ่ม # หน้าจอ LCD จะแสดงคำว่า “User 1- logout” ส่วนจอ TFT จะแสดงคำว่า “Thank you” ดังภาพที่ 4.12 (ค.)

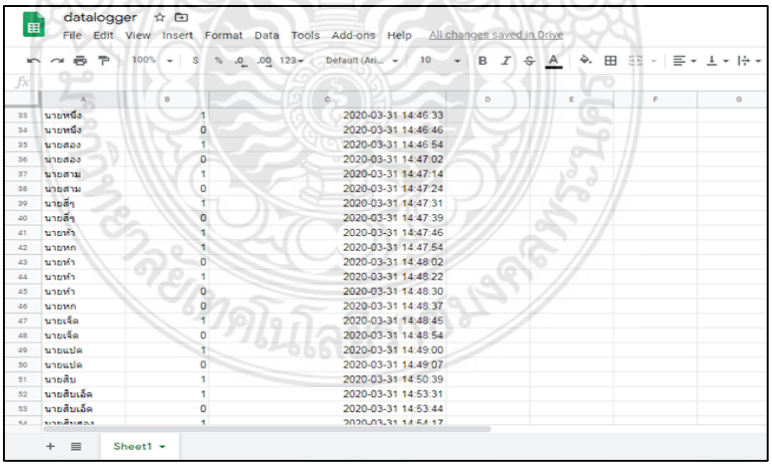
2. ข้อมูลจะไปปรากฏที่ Google sheet ว่า นายหนึ่ง มีสถานะเป็น 0 คือการออก Google sheet จะแสดงวันเดือนปีและเวลาที่ออกให้ทราบในตารางดังภาพที่ 4.13



ภาพที่ 4.13 ผลตารางแสดงการ เข้า (1) แสดงการ ออก (0) โดย Manual

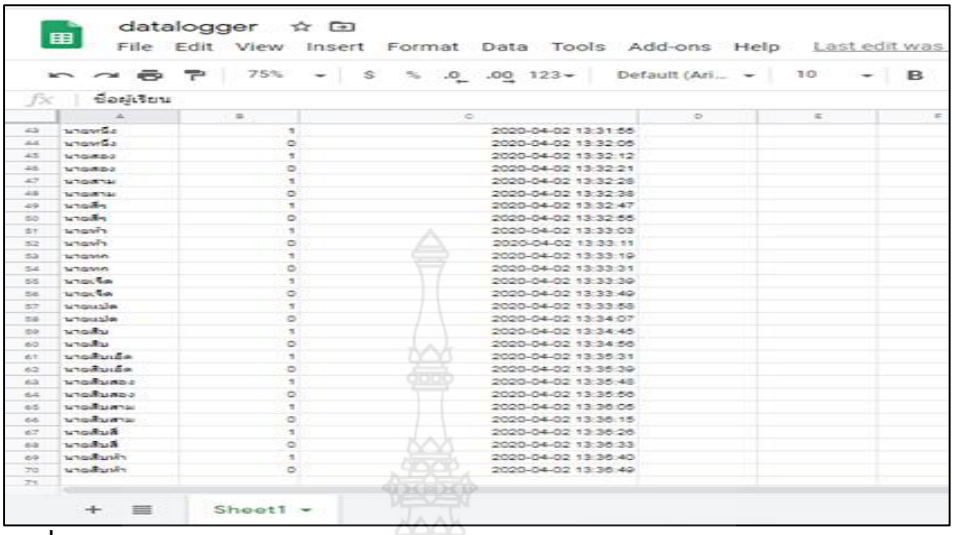
ผลการทดสอบจากกลุ่มตัวอย่าง

จากผลการทดสอบกลุ่มตัวอย่าง 15 คน โดยใช้คีย์การ์ดมาสแกนเข้า-ออก ครบตามจำนวนที่กำหนด โดยเมื่อมีการเข้า หน้าจอ TFT จะแสดงคำว่า “Welcome” Google Sheet จะแสดงผลเป็น 1 เมื่อมีการออก หน้าจอ TFT จะแสดงคำว่า “Thank you” Google Sheet จะแสดงผลเป็น 0 ดังภาพที่ 4.14



ภาพที่ 4.14 ผลตารางการสแกนเข้า (1) การสแกนออก (0) โดยคีย์การ์ดจากกลุ่มตัวอย่าง

จากผลการทดสอบกลุ่มตัวอย่าง 15 คน โดยManual เข้า-ออก ครบตามจำนวนที่กำหนด โดยเมื่อมีการเข้า หน้าจอ TFT จะแสดงคำว่า “Welcome” Google Sheet จะแสดงผลเป็น 1 เมื่อมีการออก หน้าจอ TFT จะแสดงคำว่า “Thank you” Google Sheet จะแสดงผลเป็น 0 ดังภาพที่ 4.15



ภาพที่ 4.15 ผลตารางแสดงการเข้า (1) แสดงการออก (0) โดย Manual จากกลุ่มตัวอย่าง

4.2 สรุปผลการทดสอบ

จากผลการทดสอบกลุ่มตัวอย่าง 15 คน โดยใช้คีย์การ์ดและManual เข้า-ออก ครบตามจำนวนที่กำหนด โดยเมื่อมีการเข้า หน้าจอ TFT จะแสดงคำว่า “Welcome” Google Sheet จะแสดงผลเป็น 1 เมื่อมีการออก หน้าจอ TFT จะแสดงคำว่า “Thank you” Google Sheet จะแสดงผลเป็น 0

การทำงานของเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ มีค่าความถูกต้องของการสแกนคีย์การ์ด 100% ไม่มีค่าความผิดพลาด การทดสอบการทำงานของRFID หน้าจอTFT และ หน้าจอLCD แสดงข้อความได้ถูกต้อง มีค่าความถูกต้องของการแสดงผลข้อความ 100% ไม่มีค่าความผิดพลาด และการทดสอบการทำงานของ Manual มีค่าความถูกต้อง 100% ไม่มีค่าความผิดพลาด จากผลการทดสอบการทำงานของเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ พบว่าสามารถนำไปใช้งานได้จริง มีประสิทธิภาพน่าเชื่อถือและสามารถรักษาความปลอดภัยในการการเข้าออก

บทที่ 5
สรุปผลปัญหาและข้อเสนอแนะ

โครงการเครื่องสแกนอาร์เอฟไอดีโยใช้ไมโครคอนโทรลเลอร์ดุยโน่ จัดสร้างขึ้นมาเพื่อประสงค์ป้องกันการโจรกรรมและเพิ่มความปลอดภัยในบ้านเรือน สำนักงาน ที่อยู่อาศัย สามารถใช้แบบคีย์การ์ดและแบบ Manual โดยโครงการนี้ มีผลสรุปดังต่อไปนี้

5.1 สรุปผลโครงการ

5.2 ปัญหาและอุปสรรคในการทำโครงการ

5.3 ข้อเสนอแนะในการทำโครงการ

5.1 สรุปผลโครงการ

การจัดทำโครงการ เรื่อง เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครลคอนโทรลเลอร์ดุยโน่ มาประยุกต์ใช้งานในระบบการควบคุมการเข้า-ออก RFIDเป็นที่ยอมรับว่าเป็นเทคโนโลยีที่เอื้ออำนวยต่อการใช้งาน จึงได้นำเทคโนโลยี RFID มาใช้งานกับเครื่องสแกน จากผลการทดสอบกลุ่มตัวอย่าง 15 คน โดยใช้คีย์การ์ดและManual เข้า-ออก ครบตามจำนวนที่กำหนด เมื่อมีการเข้า หน้าจอ TFT จะแสดงคำว่า “Welcome” Google Sheet จะแสดงผลเป็น 1 เมื่อมีการออก หน้าจอ TFT จะแสดงคำว่า “Thank you” Google Sheet จะแสดงผลเป็น 0

การทำงานของเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์ดุยโน่ มีค่าความถูกต้องของการสแกนคีย์การ์ด 100% ไม่มีค่าความผิดพลาด การทดสอบการทำงานของ RFID หน้าจอTFT และ หน้าจอ LCD แสดงข้อความได้ถูกต้อง มีค่าความถูกต้องของการแสดงผลข้อความ 100% ไม่มีค่าความผิดพลาด และการทดสอบการทำงานของ Manual มีค่าความถูกต้อง 100% ไม่มีค่าความผิดพลาด จากผลการทดสอบการทำงานของเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์ดุยโน่ พบว่าสามารถนำไปใช้งานได้จริง มีประสิทธิภาพน่าเชื่อถือและสามารถรักษาความปลอดภัยในการการเข้าออก

5.2 ปัญหาและอุปสรรคในการทำโครงการ

จากการดำเนินงานในการทำโครงการพัฒนาเครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ ได้พบปัญหาและอุปสรรคในการทำโครงการ ดังนี้

5.2.1 ในขณะทดสอบเครื่องสแกนอาร์เอฟไอดี เป็นระยะเวลานาน เครื่องจะเกิดการ Error เป็นครั้งคราว

5.2.2 อุปกรณ์บางตัวเกิดความเสียหาย จึงต้องมีระยะเวลาในการจัดซื้อและนำมาทดสอบใหม่

5.3 ข้อเสนอแนะในการทำโครงการ

จากการทำโครงการ เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่ เพื่อให้งานมีประสิทธิภาพในการทำงานเพิ่มมากขึ้นจึงมีข้อเสนอแนะ ดังนี้

5.3.1 ในการต่อสายสัญญาณในกล่องอุปกรณ์ ควรติดตั้งสายสัญญาณให้เรียบร้อย และหาตัวเชื่อมสายสัญญาณที่แน่นหนา เพื่อไม่ให้สายหลุด มิฉะนั้นจะทำให้เกิดปัญหาการทำงานของอุปกรณ์ และการแสดงผลของโปรแกรมได้

5.3.2 เนื่องจากในการใช้เครื่องสแกนอาร์เอฟไอดีมีการเก็บข้อมูลค่อนข้างมาก ดังนั้นจึงควรเพิ่มหน่วยความจำ เพื่อรองรับข้อมูลที่สแกนเข้าทั้งคีย์การ์ดและ Manual ให้เพียงพอเพื่อจะไม่ให้เกิดปัญหาในการประมวลผล

บรรณานุกรม

ปาณิสรา. เครื่องทาบบัตร คีย์การ์ด (Tag reader). (ออนไลน์). 2554. แหล่งที่มา : https://www.เครื่องทาบบัตร. net สืบค้นเมื่อ (3 ธันวาคม 2562)

บริษัท วีนัส ซัพพลาย จำกัด. บทความ Arduino. (ออนไลน์). 11 พฤศจิกายน 2560 แหล่งที่มา : https://www.thaieasyelec.com/article-wiki/latest-blogs/what-is-arduino-ch1.html สืบค้นเมื่อ (4 ธันวาคม 2562)

Arduino DUE. บอร์ดไมโครคอนโทรลเลอร์. (ออนไลน์). (16 มิถุนายน 2558) แหล่งที่มา : https://www.mosfex.com/_ สืบค้นเมื่อ (3 ธันวาคม 2562)

RFID RC 522 (RFID Module) (ออนไลน์). (15 กันยายน 2557) แหล่งที่มา : https://www.Arduinothai.co สืบค้นเมื่อ (13 ธันวาคม 2562)

Passive RFID TAG (ออนไลน์). (14 พฤศภาคม 2558) แหล่งที่มา : http://www.rfid-asia.com/ สืบค้นเมื่อ (14 ธันวาคม 2562).

4x4 Matrix Membrane Keypad. (ออนไลน์). (5 กุมภาพันธ์ 2560) แหล่งที่มา : https://commandronestore.com_ สืบค้นเมื่อ (17 ธันวาคม 2562)

หน้าจอ LCD 20x4. (ออนไลน์). (11 กรกฎาคม 2559). แหล่งที่มา : http://www.ett.co.th/.สืบค้นเมื่อ (3 ธันวาคม 2562)

Power supply 5V 3A I2C. (ออนไลน์). (23 เมษายน 2558). แหล่งที่มา : https://www.myarduino.net/ สืบค้นเมื่อ (22 ธันวาคม 2562).

บอร์ด NodeMCU V3 รุ่น ESP8266. (ออนไลน์). (10 มีนาคม 2560) แหล่งที่มา : https://l.facebook.com สืบค้นเมื่อ (27 ธันวาคม 2562)

หน้าจอ Nextion. (ออนไลน์). (18 มกราคม 2561) แหล่งที่มา : https://www.thaieasyelec.com/_ สืบค้น เมื่อ (5 มกราคม 2563)

Anto Server. (ออนไลน์). (21 กุมภาพันธ์ 2563). แหล่งที่มา : https://antoiot.gitbooks.io/.สืบค้นเมื่อ (8 มกราคม2563)

EPROM (ROM : Read-only Memory). (ออนไลน์). (7 มีนาคม 2559) แหล่งที่มา : https://www.comgeeks.net/_สืบค้นเมื่อ (13 มกราคม 2563).

Google API. (ออนไลน์). (17 พฤศจิกายน 2561). แหล่งที่มา : https://www.dailytech.in.th/_สืบค้นเมื่อ (19 มกราคม 2563)

Google Drive. (ออนไลน์). (26 ตุลาคม 2556). แหล่งที่มา : https://wikichaidan.wordpress.com/ สืบค้นเมื่อ (23 มกราคม 2563)

Google Sheets. (ออนไลน์). (11 สิงหาคม 2557). แหล่งที่มา : https://antoiot.gitbooks.io/_สืบค้นเมื่อ (29 มกราคม 2563)

RTC Clock DS3231. (14 ธันวาคม 2558). แหล่งที่มา : https://www.mosfex.cm/ สืบค้นเมื่อ (3 กุมภาพันธ์2563)

Arduino Uno. (ออนไลน์). (13 มีนาคม 2562). แหล่งที่มา : https://store.arduino.cc/

สืบค้นเมื่อ (9 ธันวาคม 2562)
Relay 5v 1 Channel. (ออนไลน์). (12 กุมภาพันธ์ 2557). แหล่งที่มา :
http://www.Mltelectronic.com สืบค้นเมื่อ (7 ธันวาคม 2562)

ภาคผนวก

ภาคผนวก ก
ซอร์ซโค้ด

ซอร์ซโค้ด

```
String adminpass = "1234";

#include <SPI.h>
#include <RFID.h>

#define SS_PIN 10
#define RST_PIN 9

RFID rfid(SS_PIN, RST_PIN);
// Setup variables:
int serNum0;
int serNum1;
int serNum2;
int serNum3;
int serNum4;

String senum;
int xm;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
//--------------------------------------------
#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {

  {'1', '2', '3', 'A'},

  {'4', '5', '6', 'B'},

  {'7', '8', '9', 'C'},

  {'*', '0', '#', 'D'}
```

```
};


          lcd.clear();
        }
        EEPROM.write(user6_adr5, status5);
        delay(1000);

        serNum0 = 0;
        serNum1 = 0;
        serNum2 = 0;
        serNum3 = 0;
        serNum4 = 0;
      }
      //============================================== 7

//================================================================
================================================================
=================
      else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] == 202
&& rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
      {
        status7 += 1;
        if ( status7 == 1 )
        {
          Serial.println("                                        User7 - Login !!!");
          Serial2.println("7=1");
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("User7 - login" );
          delay(3000);
          lcd.clear();
        }
        if (status7 == 2)
        {
```

```
        status7 = 0;
      }
      if ( status7 == 0 )
      {
        Serial.println("                                                User7 - Logout !!!");
        Serial2.println("7=0");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("User7 - logout" );
        delay(3000);
        lcd.clear();
      }
      EEPROM.write(user7_adr5, status7);
      delay(1000);

      serNum0 = 0;
      serNum1 = 0;
      serNum2 = 0;
      serNum3 = 0;
      serNum4 = 0;
    }
    //============================================== 8

//=============================================================
============================================================
=================
    else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] == 202
&& rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
    {
      status8 += 1;
      if ( status8 == 1 )
      {
        Serial.println("                                                User8 - Login !!!");
        Serial2.println("8=1");
        lcd.clear();
        lcd.setCursor(0, 0);
```

```
        lcd.print("User8 - login" );
        delay(3000);
        lcd.clear();
      }
      if (status8 == 2)
      {
        status8 = 0;
      }
      if ( status8 == 0 )
      {
        Serial.println("                                    User8 - Logout !!!");
        Serial2.println("8=0");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("User8 - logout" );
        delay(3000);
        lcd.clear();
      }
      EEPROM.write(user8_adr5, status5);
      delay(1000);

      serNum0 = 0;
      serNum1 = 0;
      serNum2 = 0;
      serNum3 = 0;
      serNum4 = 0;
    }
    //============================================== 9

//============================================================
============================================================
=================
    else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] == 202
&& rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
    {
      status9 += 1;
```

```
  if ( status9 == 1 )
  {
    Serial.println("                                        User9 - Login !!!");
    Serial2.println("9=1");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("User9 - login" );
    delay(3000);
    lcd.clear();
  }
  if (status9 == 2)
  {
    status9 = 0;
  }
  if ( status9 == 0 )
  {
    Serial.println("                                        User9 - Logout !!!");
    Serial2.println("9=0");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("User9 - logout" );
    delay(3000);
    lcd.clear();
  }
  EEPROM.write(user9_adr5, status9);
  delay(1000);

  serNum0 = 0;
  serNum1 = 0;
  serNum2 = 0;
  serNum3 = 0;
  serNum4 = 0;
 }
 //============================================== 10

//============================================================
```

```
================================================================
=================
      else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] == 202
&& rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
      {
        status10 += 1;
        if ( status10 == 1 )
        {
          Serial.println("                                        User10 - Login !!!");
          Serial2.println("10=1");
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("User10 - login" );
          delay(3000);
          lcd.clear();
        }
        if (status10 == 2)
        {
          status10 = 0;
        }
        if ( status10 == 0 )
        {
          Serial.println("                                        User10 - Logout !!!");
          Serial2.println("10=0");
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("User10 - logout" );
          delay(3000);
          lcd.clear();
        }
        EEPROM.write(user10_adr5, status10);
        delay(1000);

        serNum0 = 0;
        serNum1 = 0;
        serNum2 = 0;
```

```
        serNum3 = 0;
        serNum4 = 0;
      }

      //============================================== 11

//=============================================================
=============================================================
=================
      else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] == 202
&& rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
      {
        status11 += 1;
        if ( status11 == 1 )
        {
          Serial.println("                                    User11 - Login !!!");
          Serial2.println("11=1");
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("User11 - login" );
          delay(3000);
          lcd.clear();
        }
        if (status11 == 2)
        {
          status11 = 0;
        }
        if ( status11 == 0 )
        {
          Serial.println("                                    User11 - Logout !!!");
          Serial2.println("11=0");
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("User11 - logout" );
          delay(3000);
          lcd.clear();
```

```
        }
        EEPROM.write(user11_adr5, status11);
        delay(1000);

        serNum0 = 0;
        serNum1 = 0;
        serNum2 = 0;
        serNum3 = 0;
        serNum4 = 0;
      }
      //================================================ 12

//================================================================
================================================================
=================
      else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] == 202
&& rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
      {
        status12 += 1;
        if ( status12 == 1 )
        {
          Serial.println("                                                User12 - Login !!!");
          Serial2.println("12=1");
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("User12 - login" );
          delay(3000);
          lcd.clear();
        }
        if (status12 == 2)
        {
          status12 = 0;
        }
        if ( status12 == 0 )
        {
          Serial.println("                                                User12 - Logout !!!");
```

```
          Serial2.println("12=0");
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("User12 - logout" );
          delay(3000);
          lcd.clear();
        }
        EEPROM.write(user12_adr5, status12);
        delay(1000);

        serNum0 = 0;
        serNum1 = 0;
        serNum2 = 0;
        serNum3 = 0;
        serNum4 = 0;
      }
      /*
            //================================================ 13

//=================================================================
================================================================
=================
        else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] ==
202 && rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
            {
              status13 += 1;
              if ( status13 == 1 )
              {
                Serial.println("                                        User13 - Login
!!!");
                Serial2.println("13=1");
                lcd.clear();
                lcd.setCursor(0, 0);
                lcd.print("User13 - login" );
                delay(3000);
                lcd.clear();
```

```
                }
                if (status13 == 2)
                {
                  status13 = 0;
                }
                if ( status13 == 0 )
                {
                  Serial.println("                                                User13 - Logout
!!!");
                  Serial2.println("13=0");
                  lcd.clear();
                  lcd.setCursor(0, 0);
                  lcd.print("User13 - logout" );
                  delay(3000);
                  lcd.clear();
                }
                EEPROM.write(user13_adr5, status13);
                delay(1000);

                serNum0 = 0;
                serNum1 = 0;
                serNum2 = 0;
                serNum3 = 0;
                serNum4 = 0;
              }
        /*
              //============================================== 14

//=============================================================
=============================================================
=================
        else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] ==
202 && rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
              {
                status14 += 1;
                if ( status14 == 1 )
```

```
    {
      Serial.println("                                    User14 - Login
!!!");
      Serial2.println("14=1");
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("User14 - login" );
      delay(3000);
      lcd.clear();
    }
    if (status14 == 2)
    {
      status14 = 0;
    }
    if ( status14 == 0 )
    {
      Serial.println("                                    User14 - Logout
!!!");
      Serial2.println("14=0");
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("User14 - logout" );
      delay(3000);
      lcd.clear();
    }
    EEPROM.write(user14_adr5, status14);
    delay(1000);

    serNum0 = 0;
    serNum1 = 0;
    serNum2 = 0;
    serNum3 = 0;
    serNum4 = 0;
  }

  //============================================ 15
```

```
//=============================================================
=============================================================
=================
        else if ( rfid.serNum[0] == 23 && rfid.serNum[1] == 56 && rfid.serNum[2] ==
202 && rfid.serNum[3] == 77 && rfid.serNum[4] == 168 )
              {
                status15 += 1;
                if ( status15 == 1 )
                {
                  Serial.println("                                        User15 - Login
!!!");
                  Serial2.println("15=1");
                  lcd.clear();
                  lcd.setCursor(0, 0);
                  lcd.print("User15 - login" );
                  delay(3000);
                  lcd.clear();
                }
                if (status15 == 2)
                {
                  status15 = 0;
                }
                if ( status15 == 0 )
                {
                  Serial.println("                                        User15 - Logout
!!!");
                  Serial2.println("15=0");
                  lcd.clear();
                  lcd.setCursor(0, 0);
                  lcd.print("User15 - logout" );
                  delay(3000);
                  lcd.clear();
                }
                EEPROM.write(user15_adr5, status15);
                delay(1000);
```

```
                serNum0 = 0;
                serNum1 = 0;
                serNum2 = 0;
                serNum3 = 0;
                serNum4 = 0;
              }
       */
     } else {

       Serial.print(".");
     }
   }
  }

  rfid.halt();


//=================================================================
==========================================
  if ( setmode == 99 ) // normal
  {
    lcd.setCursor(0, 0);
    lcd.print("Main Menu (Welcome)" );
    lcd.setCursor(0, 1);
    lcd.print("Select Mode" );
    lcd.setCursor(0, 2);
    lcd.print("Press A = login" );
    lcd.setCursor(0, 3);
    lcd.print("Press B = Admin" );
  }

  if ( setmode == 4 )
  {
    lcd.setCursor(0, 0);
    lcd.print("@Admin" );
```

```
    lcd.setCursor(0, 1);
    lcd.print("key Admin Password" );

    if ( xd == 1 )
    {
      lcd.setCursor(0, 2);
      lcd.print("*");
    }
    if ( xd == 2 )
    {
      lcd.setCursor(0, 2);
      lcd.print("**");
    }
    if ( xd == 3 )
    {
      lcd.setCursor(0, 2);
      lcd.print("***");
    }
    if ( xd == 4 )
    {
      lcd.setCursor(0, 2);
      lcd.print("****");
    }
    if ( xd == 0 )
    {
      lcd.setCursor(0, 2);
      lcd.print("    ");
    }

  }//end setmode4

//============================================================
==========================================
  if ( setmode == 1 )
  {
    lcd.setCursor(0, 0);
```

```
  lcd.print("@Admin - login" );
  lcd.setCursor(0, 1);
  lcd.print("Select number user" );

  lcd.setCursor(0, 2);
  lcd.print("User:" + String(senum) + " ");
}

if ( setmode == 3 )
{
  lcd.setCursor(0, 0);
  lcd.print("@User - login" );
  lcd.setCursor(0, 1);
  lcd.print("Key Your Password" );

  lcd.setCursor(0, 2);
  lcd.print(String(memc) + "     ");
}


if ( setmode == 2 )
{

  lcd.setCursor(0, 0);
  lcd.print("@Admin - login" );
  lcd.setCursor(0, 1);
  lcd.print("Set Your Password    " );

  lcd.setCursor(0, 2);
  lcd.print("User:" + String(selectnum) + " ");

  if ( selectnum == 1 )
  {
    lcd.setCursor(0, 3);
    lcd.print("Pass:" + String(mem1) + "     ");
  }
```

```
if ( selectnum == 2 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem2) + "     ");
}
if ( selectnum == 3 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem3) + "     ");
}
if ( selectnum == 4 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem4) + "     ");
}
if ( selectnum == 5 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem5) + "     ");
}
if ( selectnum == 6 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem6) + "     ");
}
if ( selectnum == 7 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem7) + "     ");
}
if ( selectnum == 8 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem8) + "     ");
}
if ( selectnum == 9 )
```

```
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem9) + "     ");
}
if ( selectnum == 10 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem10) + "     ");
}
if ( selectnum == 11 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem11) + "     ");
}
if ( selectnum == 12 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem12) + "     ");
}
if ( selectnum == 13 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem13) + "     ");
}
if ( selectnum == 14 )
{
  lcd.setCursor(0, 3);
  lcd.print("Pass:" + String(mem14) + "     ");
}
//if ( selectnum == 15 )
// {
//   lcd.setCursor(0, 3);
//   lcd.print("Pass:" + String(mem15) + "     ");
// }
```

```
  }
}// end loop
```

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(D1, D2); // RX, TX
#include <AntoIO.h>
const char *ssid = "ECONTEC DESIGN";
const char *pass = "0879863226";
const char *user = "rmutp";
const char *token = "HTSci920ghYHgpjAB3d2ZFQAMADSq15N32OQqj4h";
const char *thing = "rfid_server";
AntoIO anto(user, token, thing);

//---------------------------------------- -
```

```
import antolib
import requests
import datetime
import time
#-------------------------------------------------------------------------
import gspread
from oauth2client.service_account import ServiceAccountCredentials
import pprint
x=0
name=""
scope = ['https://spreadsheets.google.com/feeds',
'https://www.googleapis.com/auth/drive']
```

```
creds = ServiceAccountCredentials.from_json_keyfile_name('credskorn.json',scope)
client = gspread.authorize(creds)
sheet1 = client.open('datalogger').sheet1
pp = pprint.PrettyPrinter()

name_server1=""
name_server2=""
name_server3=""
name_server4=""
name_server5=""
name_server6=""
name_server7=""
name_server8=""
name_server9=""
name_server10=""
name_server11=""
name_server12=""
name_server13=""
name_server14=""
name_server15=""
status1=""
status2=""
status3=""
status4=""
status5=""
status6=""
status7=""
status8=""
status9=""
status10=""
status11=""
status12=""
status13=""
status14=""
status15=""
# username of anto.io account
```

```
user = 'rmutp'
# key of permission, generated on control panel anto.io
key = 'HTSci920ghYHgpjAB3d2ZFQAMADSq15N32OQqj4h'
# your default thing.
thing = 'rfid_server'

anto = antolib.Anto(user, key, thing)
def connectedCB():
    anto.sub("status1");
    anto.sub("status2");
    anto.sub("status3");
    anto.sub("status4");
    anto.sub("status5");
    anto.sub("status6");
    anto.sub("status7");
    anto.sub("status8");
    anto.sub("status9");
    anto.sub("status10");
    anto.sub("status11");
    anto.sub("status12");
    anto.sub("status13");
    anto.sub("status14");
    anto.sub("status15");
    anto.sub("name_server1");
    anto.sub("name_server2");
    anto.sub("name_server3");
    anto.sub("name_server4");
    anto.sub("name_server5");
    anto.sub("name_server6");
    anto.sub("name_server7");
    anto.sub("name_server8");
    anto.sub("name_server9");
    anto.sub("name_server10");
    anto.sub("name_server11");
    anto.sub("name_server12");
    anto.sub("name_server13");
```

```
    anto.sub("name_server14");
    anto.sub("name_server15");



def dataCB(channel, msg):


    #-----------------------------------------------------------------------
    if(channel == 'name_server1'):
        global name_server1
        name_server1=msg.decode("utf-8")
        print("                                name_server1= ",name_server1)
    if(channel == 'name_server2'):
        global name_server2
        name_server2=msg.decode("utf-8")
        print("                                name_server2= ",name_server2)
    if(channel == 'name_server3'):
        global name_server3
        name_server3=msg.decode("utf-8")
        print("                                name_server3= ",name_server3)
    if(channel == 'name_server4'):
        global name_server4
        name_server4=msg.decode("utf-8")
        print("                                name_server4= ",name_server4)
    if(channel == 'name_server5'):
        global name_server5
        name_server5=msg.decode("utf-8")
        print("                                name_server5= ",name_server5)
    if(channel == 'name_server6'):
        global name_server6
        name_server6=msg.decode("utf-8")
        print("                                name_server6= ",name_server6)

    if(channel == 'name_server7'):
```

```
    global name_server7
    name_server7=msg.decode("utf-8")
    print("                              name_server7= ",name_server7)
if(channel == 'name_server8'):
    global name_server8
    name_server8=msg.decode("utf-8")
    print("                              name_server8= ",name_server8)
if(channel == 'name_server9'):
    global name_server9
    name_server9=msg.decode("utf-8")
    print("                              name_server9= ",name_server9)
if(channel == 'name_server10'):
    global name_server10
    name_server10=msg.decode("utf-8")
    print("                              name_server10= ",name_server10)
if(channel == 'name_server11'):
    global name_server11
    name_server11=msg.decode("utf-8")
    print("                              name_server11= ",name_server11
if(channel == 'name_server12'):
    global name_server12
    name_server12=msg.decode("utf-8")
    print("                              name_server12= ",name_server12)
if(channel == 'name_server13'):
    global name_server13
    name_server13=msg.decode("utf-8")
    print("                              name_server13= ",name_server13)
if(channel == 'name_server14'):
    global name_server14
    name_server14=msg.decode("utf-8")
    print("                              name_server14= ",name_server14)
if(channel == 'name_server15'):
    global name_server15
    name_server15=msg.decode("utf-8")
    print("                              name_server15= ",name_server15)
```

```
    if(channel == 'status1'):
        global status1
        status1=msg.decode("utf-8")
        print("                                status1= ",status1)
        global x
        now=datetime.datetime.now()
        x+=1
        sheet1.update_cell(x,1,name_server1)
        sheet1.update_cell(x,2,status1)
        sheet1.update_cell(x,3,str(now))


def setup():
    anto.mqtt.onConnected(connectedCB)
    anto.mqtt.onData(dataCB)
    anto.mqtt.connect()


def myLoopFunction():


    global name_server1
    global name_server2
    global name_server3
    global name_server4
    global name_server5
    global name_server6
    global name_server7
    global name_server8
    global name_server9
    global name_server10
    global name_server11
    global name_server12
    global name_server13
    global name_server14
    global name_server15
```

global status1
global status2
global status3
global status4
global status5
global status6
global status7
global status8

ภาคผนวก ข
Data Sheets

RC522 RFID Module

12 June 2019 - 0 Comments

RC522 RFID Module

RC522 RFID Module Pinout

Pin Number	Pin Name	Description
1	Vcc	Used to Power the module, typically 3.3V is used
2	RST	Reset pin – used to reset or power down the module
3	Ground	Connected to Ground of system
4	IRQ	Interrupt pin – used to wake up the module when a device comes into range
5	MISO/SCL/Tx	MISO pin when used for SPI communication, acts as SCL for I2c and Tx for UART.
6	MOSI	Master out slave in pin for SPI communication
7	SCK	Serial Clock pin – used to provide clock source
8	SS/SDA/Rx	Acts as Serial input (SS) for SPI communication, SDA for IIC and Rx during UART

4x4 Matrix Membrane Keypad (#27899)

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.

Features

- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

Key Specifications

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions:
  Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
  Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

Application Ideas

- Security systems
- Menu selection
- Data entry for embedded systems

How it Works

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in Figure 1.

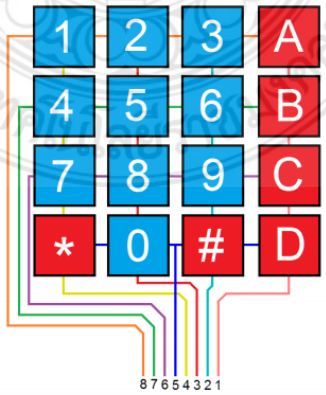


Figure 1: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed.

For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1 high. This means that a contact has been made between column 4 and row 1, so button 'A' has been pressed.

Systronix 20x4 LCD
Brief Technical Data
July 31, 2000

Here is brief data for the Systronix 20x4 character LCD. It is a DataVision part and uses the Samsung KS0066 LCD controller. It's a clone of the Hitachi HD44780. We're not aware of any incompatabilities between the two - at least we have never seen any in all the code and custom applications we have done.

This 20x4 LCD is electrically and mechanically interchangeable with 20x4 LCDs from several other vendors. The only differences we've seen among different 20x4 LCDs are:

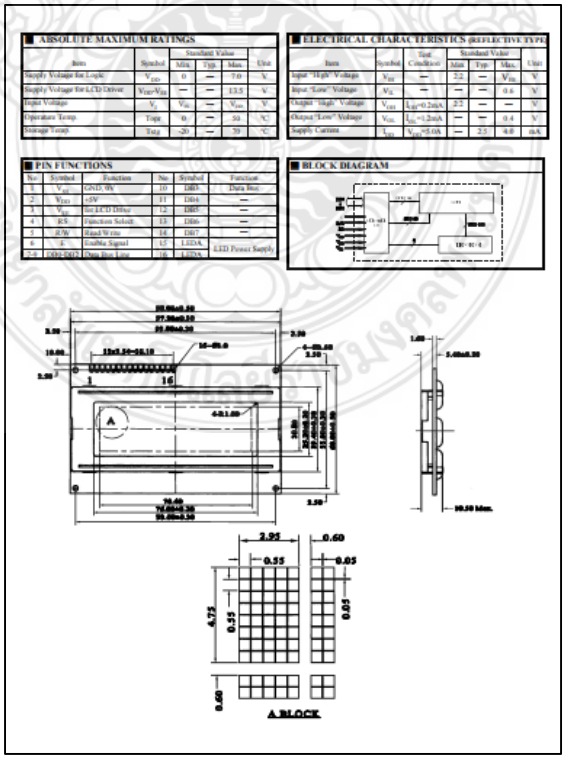
1) LED backlight brightness, voltage and current vary widely, as does the quality of the display

2) There is a resistor "Rf" which sets the speed of the LCD interface by controlling the internal oscillator frequency. Several displays we have evaluated have a low resistor value. This makes the display too slow. Looking at the Hitachi data sheet page 56, it appears that perhaps the "incorrect" resistor is really intended for 3V use of the displays.

At 5V the resistor Rf should be 91 Kohms. At 3V it should be 75 Kohms. Using a 3V display at 5V is acceptable from a voltage standpoint (the display can operate on 3-5V) but the oscillator will then be running too slowly. One fix is to always check the busy flag and not use a fixed time delay in your code, then it will work regardless of the LCD speed. The other option is to always allow enough delay for the slower display.

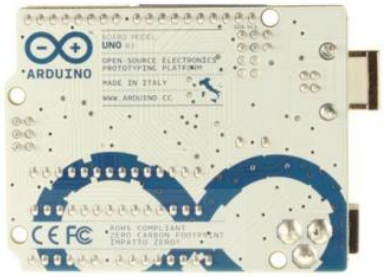
All Systronix 20x4 LCDs have the 91 Kohm resistor and are intended for 5V operation.

Thank you for purchasing Systronix embedded control products and accessories. If you have any other questions please email to support@systronix.com or phone +1-801-534-1017, fax +1-801-534-1019.



Arduino Uno

Arduino Uno R3 Front

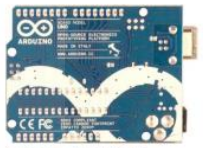


Arduino Uno R3 Back

Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front



Arduino Uno Back

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

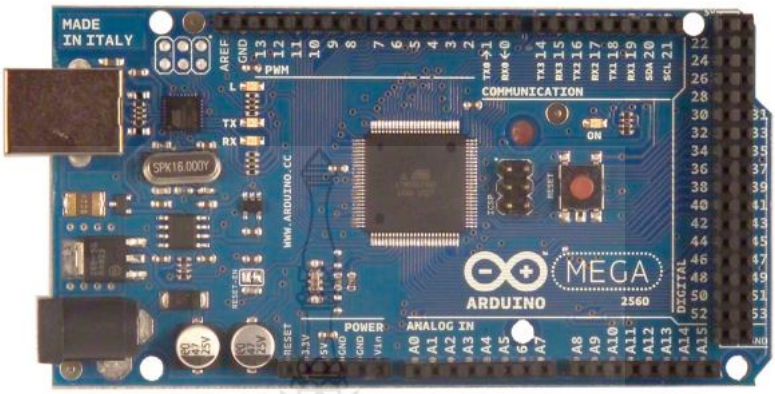
EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Arduino Mega 2560 Datasheet



Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

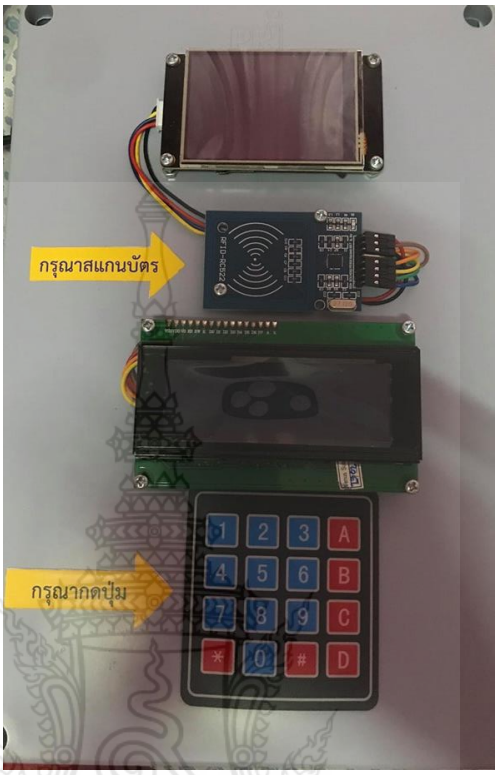
Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode() , digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH

ภาคผนวก ค
คู่มือสำหรับการใช้งาน

คู่มือสำหรับการใช้งาน เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลอดุยเลอร์โน่



ภาพที่ ค.1 เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครคอนโทรลเลอร์อดุยโน่

เครื่องสแกนอาร์เอฟไอดีโดยใช้ไมโครลคอนโทรลเลอร์อดุยโน่ จะแบ่งการใช้งานเป็น 2 รูปแบบ ได้แก่ แบบการสแกนบัตร และ แบบการใช้ Manual

1. แบบสแกนบัตร

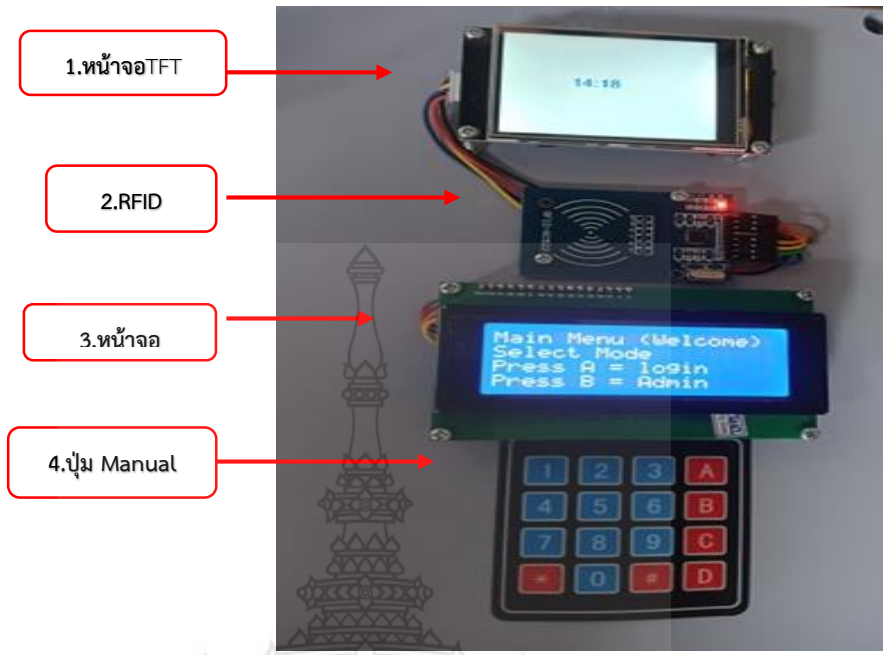
ขั้นตอนการใช้งานเครื่องสแกนอาร์เอฟไอดีโดยใช้แบบสแกนบัตร

ขั้นตอนที่ 1 ดูความพร้อมของเครื่องสแกนว่าพร้อมใช้งานหรือไม่ โดยดูจากหน้าจอ TFT จะแสดงเวลาให้ทราบ ดังภาพที่ ค.2 (1.หน้าจอ TFT)

ขั้นตอนที่ 2 นำบัตรมาสแกนที่ RFID ดังภาพที่ ค.2 (2.RFID)

ขั้นตอนที่ 3 ถ้าต้องการเข้า ให้เอาคีย์การ์ดไปสแกนที่RFID จะแสดงคำว่า “Welcome” บนหน้าจอ TFT ส่วนหน้าจอ LCD จะขึ้นคำว่า “Login” ดังภาพที่ ค.3

ขั้นตอนที่ 4 ถ้าต้องการออก ให้เอาคีย์การ์ดไปสแกนที่RFID จะแสดงคำว่า “Thank you" บนหน้าจอ TFT ส่วนหน้าจอ LCD จะขึ้นคำว่า “Logout” ดังภาพที่ ค.4



ภาพที่ ค.2 การแสดงหน้าจอและรายละเอียดอุปกรณ์

ภาพที่ ค.3 การสแกนบัตรด้วยคีย์การ์ดเข้า

ภาพที่ ค.4 การสแกนบัตรด้วยคีย์การ์ดออก

2. แบบการใช้Manual

ขั้นตอนการใช้งานเครื่องสแกนอาร์เอฟไอดีโดยใช้แบบManual

ขั้นตอนที่ 1 ดูความพร้อมของเครื่องสแกนว่าพร้อมใช้งานหรือไม่ โดยดูจากหน้าจอ TFT จะแสดงเวลาให้ทราบ ดังภาพที่ ค.2

ขั้นตอนที่ 2 เมื่อต้องการเข้า กดปุ่ม A ตามด้วยรหัสส่วนตัว 4 หลัก เช่น รหัส1111 แล้วกดปุ่ม # เพื่อ Login หน้าจอ TFT จะ แสดง“Welcome” ส่วนหน้าจอ LCD จะขึ้นคำว่า “Login” ดังภาพที่ ค.5

ขั้นตอนที่ 3 เมื่อต้องการออก ให้กดปุ่ม A แล้วตามด้วยกดรหัส 1111 แล้วกดปุ่ม # จะขึ้นคำว่า “Logout” ที่หน้าจอLCD ส่วนหน้าจอ TFT จะแสดง “Thank you” ดังภาพที่ ค.6

ภาพที่ ค.5 การใช้ Manual เข้า

ภาพที่ ค.6 การใช้ Manual ออก

คู่มือสำหรับการตั้งรหัสโดยAdmin

ขั้นตอนการตั้งรหัส Code ดังนี้

1. แอดมินทำการ Login โดยการกดปุ่ม B ตามด้วยรหัส “1234” แล้ว กดปุ่ม #

2. จากนั้นเลือก User ของคนที่ทำการเพิ่มรหัส จากกลุ่มตัวอย่าง 15คน โดยระบุคนที่ 1 ให้เป็นหมายเลข 1 จนครบ User 15 คน เช่น User 1 กดปุ่ม 1 แล้วกดปุ่ม # จากนั้นกดรหัสที่ แอดมินทำการตั้งให้ เช่น 1111 แล้วกดปุ่ม # จะได้รหัสเรียบร้อย แสดงว่า User 1จะได้รหัสเข้า “1111”

3. แอดมินทำการ Logout โดยกดปุ่ม * เพื่อให้ User สามารถเข้าล็อกอินได้เพื่อให้ User สามารถเข้า - ออก โดย Login และ logout ตามปกติ

ประวัติผู้เขียน

ชื่อ-สกุล นางสาวกรกมล ไวยวุฒิ

วัน เดือน ปีเกิด 9 มิถุนายน พ.ศ. 2540

สถานที่เกิด จังหวัดสระบุรี

ที่อยู่ปัจจุบัน 47 หมู่ 1 ตำบลปากข้าวสาร อำเภอเมือง จังหวัด สระบุรี 18000

เบอร์โทรศัพท์ 0631611329

E-mail kornkamol-w@rmutp.ac.th

ประวัติการศึกษา

พ.ศ. 2561 ระดับชั้นประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)
สาขาวิชา อิเล็กทรอนิกส์
วิทยาลัยเทคนิคสระบุรี

กำลังศึกษา ระดับปริญญาตรี อุตสาหกรรมบัณฑิต (อ.ส.บ.)
สาขาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร (วิทยาเขตเทเวศร์)

ประวัติผู้เขียน

ชื่อ-สกุล นายคำรณ โยธี

วัน เดือน ปีเกิด 23 สิงหาคม พ .ศ.2539

สถานที่เกิด จังหวัดศรีสะเกษ

ที่อยู่ปัจจุบัน 17 หมู่ 10 ตำบลปราสาทเยอ อำเภอไพรบึง จังหวัดศรีสะเกษ 33180

เบอร์โทรศัพท์ 0634131047

E-mail khamron-y@rmutp.ac.th

ประวัติการศึกษา

พ.ศ. 2560 ระดับชั้นประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)
สาขาวิชา อิเล็กทรอนิกส์

กำลังศึกษา ระดับปริญญาตรี อุตสาหกรรมบัณฑิต (อ.ส.บ.)
สาขาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร (วิทยาเขตเทเวศร์)

ประวัติผู้เขียน

ชื่อ-สกุล นาย พีรวัฒน์ ทองพูล

วัน เดือน ปีเกิด 30 ตุลาคม พ.ศ. 2540

สถานที่เกิด จังหวัด อ่างทอง

ที่อยู่ปัจจุบัน 131 หมู่ 3 ตำบล ยางช้าย อำเภอโพธิ์ทอง จังหวัดอ่างทอง 14120

เบอร์โทรศัพท์ 0632476661

E-mail phongnapee.thongpool@gmail.com

ประวัติการศึกษา

พ.ศ. 2561 ระดับชั้นประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)
สาขาวิชา อิเล็กทรอนิกส์

กำลังศึกษา ระดับปริญญาตรี อุตสาหกรรมบัณฑิต (อ.ส.บ.)
สาขาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร (วิทยาเขตเทเวศร์)