ระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ

(Automated Small Warehouse)

บุปผา พงศ์พสุขา 63010018

Buppha Phongphasukha 63010018

กานต์ญาณัฐ มังเทศ 63010068

Ganyanut Mangtes 63010068

พรนิชา ลินทะจะกะ 63010640

Pornnicha Lintajaka 63010640

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ

(Automated Small Warehouse)

โดย

บุปผา พงศ์พสุขา 63010018

กานต์ญาณัฐ มังเทศ 63010068

พรนิชา ลินทะจะกะ 63010640

อาจารย์ที่ปรึกษา

ผศ.ดร. แสงระวี บัวแก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2566

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ

(Automated Small Warehouse)

ผู้จัดทำ นางสาว บุปผา พงศ์พสุขา รหัสประจำตัว 63010018

นางสาว กานต์ญาณัฐ มังเทศ รหัสประจำตัว 63010068

นางสาว พรนิชา ลินทะจะกะ รหัสประจำตัว 63010640

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ผศ.ดร. แสงระวี บัวแก้ว

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	ระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ
นักศึกษา	นางสาว บุปผา พงศ์พสุขา รหัสประจำตัว 63010018
	นางสาว กานต์ญาณัฐ มังเทศ รหัสประจำตัว 63010068
	นางสาว พรนิชา ลินทะจะกะ รหัสประจำตัว 63010640
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2566
อาจารย์ที่ปรึกษาโครงการ	ผศ.ดร. แสงระวี บัวแก้ว

บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อศึกษาการทำงานของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ ซึ่งมีจุดประสงค์เพื่อศึกษาเกี่ยวกับการทำงานของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ โดยใช้ระบบไมโครคอนโทรลเลอร์ (MCU) อย่าง Arduino Uno เป็นส่วนควบคุมการทำงานของระบบทั้งหมด โดยการรับค่าจาก Scanner แล้วนำไปทำการประมวลผล เพื่อควบคุมการทำงานของมอเตอร์ การแสดงผลของจอ LCD รวมไปถึงทำการออกแบบซอฟต์แวร์สำหรับการตรวจเช็คจำนวนสินค้าออก และยอดคงเหลือ โดยการใช้โปรแกรม Filemaker ที่เป็นการอัพเดตข้อมูลแบบ Real time เพื่อให้ระบบมีการทำงานแบบอัตโนมัติ และเป็นไปอย่างที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Automated Small Warehouse
Student	Miss Buppha Phongphasukha Student ID 63010018
	Miss Ganyanut Mangtes Student ID 63010068
	Miss Pornnicha Lintajaka Student ID 63010640
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2023
Project Advisor	Asst. Prof. Dr. Seangrawee Buakeaw

ABSTRACT

This project was created to study the operation a Automated Small Warehouse control system. Its purpose is to investigate how a Automated Small Warehouse control system works using a microcontroller (MCU) like the Arduino Uno as the control unit for the entire system. It receives input from a scanner, processes it to control the operation of motors, display information on an LCD screen, and design software for checking the quantity of products in and out, as well as the remaining balance. It utilizes Filemaker software to update real-time data to ensure the system operates automatically and as desired.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยความกรุณาช่วยเหลือ การแนะนำให้คำปรึกษา ตรวจสอบแก้ไขข้อพกพร่องต่างๆด้วยความเอาใจใส่อย่างดียิ่งจาก ผศ.ดร. แสงระวี บัวแก้ว และคณะอาจารย์จากภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะผู้จัดทำกราบขอบพระคุณด้วยเป็นอย่างสูง

ขอขอบคุณรุ่นพี่ รวมไปถึงเพื่อน ๆ ทุกคนที่ให้คำปรึกษา รวมทั้งกำลังใจที่ดี จึงขอขอบคุณทุกท่านเหล่านั้นไว้ ณ โอกาสนี้ด้วย

ขอขอบคุณบิดา และมารดาที่ช่วยเหลือสนับสนุนทั้งด้านกำลังใจ และกำลังทรัพย์ด้วยดีตลอดมา คุณค่าทั้งหลายที่ได้รับจากโครงงาน และรายงานฉบับนี้ ผู้เขียนขอมอบเป็นกตัญญูกตเวทีแด่บิดามารดา และบูรพาจารย์ที่เคยอบรมส่งสอน รวมทั้งผู้มีพระคุณทุกท่าน

บุปผา พงศ์พสุขา
กานต์ญาณัฐ มังเทศ
พรนิชา ลินทะจะกะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ซ
สารบัญตาราง	ญ
บทที่ 1 บทนำ	1
1.1 ความเป็นมา และความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการศึกษา	2
1.3 สมมติฐานของการศึกษา	2
1.4 ขอบเขตของการศึกษา	2
1.5 ระยะเวลาในการทำโครงงาน	2
1.5.1 ภาคการศึกษาที่ 1	2
1.5.2 ภาคการศึกษาที่ 2	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ	5
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	6
2.1 Arduino Mega 2560	6
2.1.1 คุณสมบัติจำเพาะของ Arduino Mega 2560	7
2.1.2 ฟังก์ชันของ Arduino Mega 2560	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.1.3 การสื่อสารบน Arduino Mega 2560	9
2.2 Servo motor	12
2.2.1 หลักการทำงานของเซอร์โวมอเตอร์	12
2.2.2 ชนิดของเซอร์โวมอเตอร์	13
2.2.3 การใช้งานเซอร์โวมอเตอร์	13
2.2.4 การจ่ายไฟให้เซอร์โวมอเตอร์	14
2.2.5 การเลือกใช้งานเซอร์โวมอเตอร์	15
2.3 Liquid Crystal Display (LCD)	16
2.3.1 โครงสร้างโดยทั่วไปของ LCD	18
2.3.2 การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller	18
2.3.3 การเชื่อมต่อ Arduino กับจอ Character LCD	21
2.4 MH-ET Live Scanner V3.0	22
2.4.1 คุณสมบัติของ MH-ET Live Scanner V3.0	22
2.4.2 บาร์โค้ด และประเภทของการอ่านบาร์โค้ดที่ MH-ET Live Scanner V3.0 อ่านได้	24
2.4.3 หลักการอ่านบาร์โค้ด	26
2.5 Karakuri kaizen theory	27
2.5.1 หลักการทำงานของ Karakuri kaizen	27
2.5.2 ประโยชน์ของ Karakuri kaizen	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.6 โปรแกรม Filemaker Pro	29
2.6.1 คุณสมัติของ Filemaker Pro	30
บทที่ 3 วิธีการดำเนินโครงการ	31
3.1 อุปกรณ์ เครื่องมือ หรือโปรแกรมที่ใช้ในการดำเนินโครงงาน	31
3.2 ขั้นตอนการดำเนินโครงงาน	31
3.3 การพิจารณาการใช้งานเซอร์โวมอเตอร์	32
3.4 การพิจารณาการใช้งานจอ LCD 16*2 I2C	33
3.5 พิจารณาวงจรที่ใช้	34
3.6 พิจารณาการเขียนโปรแกรมภาษาซีใน Arduino IDE	35
3.7 การออกแบบโครงของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse)	36
3.8 พิจารณาการสร้างแอพพลิเคชั่นจากโปรแกรม Filemaker	38
3.8.1 การสร้างหน้า Layout สำหรับผู้ใช้งาน	38
3.8.2 Flowchart ของ Filemaker Pro	43
บทที่ 4 ผลการทดลอง	44
4.1 การทดลองการทำงานของการแสดงผลบนจอ LCD	44
4.2 การทดลองทดสอบการทำงานของ Servo motor	45
4.3 การทดสอบการทำงานของโปรแกรม Filemaker	46
4.3.1 การทำงานบน Smartphone (Iphone)	46
4.3.2 การทำงานบน Tablet (Ipad)	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.3.3 การส่งออกข้อมูลในรูปแบบ .PDF file	48
บทที่ 5 สรุปผลการทดลอง	49
5.1 สรุปผลการทดลอง	49
5.2 ข้อเสนอแนะ	50
เอกสารอ้างอิง	ฎ
ภาคผนวก	ฐ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 Arduino Mega 2560	6
รูปที่ 2.2 โครงสร้าง Arduino Mega 2560	8
รูปที่ 2.3 Serial communication แบบ Synchronous	9
รูปที่ 2.4 Serial communication แบบ Asynchronous	10
รูปที่ 2.5 I2C Communication	10
รูปที่ 2.6 SPI Communication	11
รูปที่ 2.7 Servo motor	12
รูปที่ 2.8 สัญญาณ PWM ที่ส่งผลต่อองศาการหมุนของเซอร์โวมอเตอร์	14
รูปที่ 2.9 ตัวอย่างการต่อไฟเลี้ยงเซอร์โวมอเตอร์เพิ่มเติมโดยใช้สวิตชิ่ง 5V 5A	14
รูปที่ 2.10 Liquid Crystal Display (LCD)	16
รูปที่ 2.11 Character LCD	17
รูปที่ 2.12 Graphic LCD	17
รูปที่ 2.13 รูปแบบการเขียน/อ่านข้อมูลแบบ I2C Bus	20
รูปที่ 2.14 I2C Bus START Conditions and STOP Conditions	21
รูปที่ 2.15 MH-ET Live Scanner V3.0	22
รูปที่ 2.16 โครงสร้างของ MH-ET Live Scanner V3.0	23
รูปที่ 2.17 มุมมองการอ่านบาร์โค้ด	23
รูปที่ 2.18 องค์ประกอบของ Barcode	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 2.19 ตัวอย่าง Barcode ในรูปแบบ 1D 25

รูปที่ 2.20 ตัวอย่าง Barcode ในรูปแบบ 2D 25

รูปที่ 2.21 ขั้นตอนการอ่านบาร์โค้ด 26

รูปที่ 2.22 ตัวอย่าง Karakuri 27

รูปที่ 2.23 Filemaker Pro 29

รูปที่ 2.24 Icon Program Filemaker Pro บน PC 30

รูปที่ 3.1 Schematic ของวงจร Automated Small Warehouse 34

รูปที่ 3.2 Flowchart การทำงานของโปรแกรม 35

รูปที่ 3.3 การออกแบบโครงของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ 36

รูปที่ 3.4 ตำแหน่งการติดตั้ง Servo motor 37

รูปที่ 3.5 โครงสร้างจริงของงระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ 37

รูปที่ 3.6 Flowchart ของ Filemaker 43

รูปที่ 4.1 การแสดงผลของจอ LCD แสดงสถานะเริ่มต้น 44

รูปที่ 4.2 การแสดงผลของจอ LCD เมื่ออ่านบาร์โค้ดของช่องที่ 1 44

รูปที่ 4.3 การแสดงผลของจอ LCD เมื่ออ่านบาร์โค้ดของช่องที่ 2 44

รูปที่ 4.4 การแสดงผลของจอ LCD เมื่ออ่านบาร์โค้ดของช่องที่ 3 44

รูปที่ 4.5 การแสดงผลของจอ LCD เมื่อบาร์โค้ดไม่ถูกต้อง 44

รูปที่ 4.6 หน้าโปรแกรม Filemaker สำหรับผู้ใช้งานบนสมาร์ทโฟน 46

รูปที่ 4.7 หน้าโปรแกรม Filemaker สำหรับผู้ใช้งานบนแท็บเล็ต 47

รูปที่ 4.8 หน้าข้อมูลที่ถูกส่งออกมาเป็น .PDF 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1.5 ระยะเวลาในการทำโครงงาน	2
1.5.1 ภาคการศึกษาที่ 1	2
1.5.2 ภาคการศึกษาที่ 2	4
2.3 ตารางแสดงตัวอย่างการเชื่อมต่อขากับ Arduino	20
3.4 การพิจารณาการใช้งานจอ LCD 16*2 I2C	33
4.2 การทดลองทดสอบการทำงานของ Servo motor	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

จากการฝึกงานภาคฤดูร้อนในโรงงานอุตสาหกรรม ในการทำงานพบว่า การเบิกจ่ายของในคลังสินค้านั้นมีหลายขั้นตอน และขึ้นอยู่กับหลายหน่วยงานในโรงงานหนึ่งโรงงาน ประกอบกับการใช้บุคลากรจำนวนมากในการตรวจเช็คจำนวนของสินค้าภายในคลัง ทำให้ประสบกับปัญหาหลายประการ อาทิเช่น หากไม่มีเจ้าหน้าที่ หรือพนักงานประจำอยู่ ณ จุดที่เบิกจ่ายสินค้า จะทำให้ไม่สามารถเบิกสินค้าได้ หรือไม่ว่าจะเป็นการเบิกสินค้าเกินจำนวน ทำให้เมื่อต้องสรุปยอดเช็คจำนวนสินค้าประจำวันนั้นมีข้อผิดพลาด ส่งผลต่อค่าใช้จ่ายที่เพิ่มขึ้น และศูนย์เสียทรัพยากรโดยใช่เหตุ จึงมีการนำระบบอัตโนมัติมาใช้ในการจัดการคลังสินค้า

การประยุกต์ใช้งานระบบอัตโนมัติต่างๆ เข้ากับคลังสินค้า ได้รับความนิยม และถูกนำไปใช้งานกันอย่างแพร่หลายภายในโรงงานอุตสาหกรรมทั้งขนาดเล็ก และขนาดใหญ่ อาทิเช่น แขนกลที่ใช้ในการหยิบ หรือประกอบชิ้นส่วนต่างๆ ระบบสายพาน สำหรับการส่งสินค้าไปยังจุดต่างๆ โดยไม่ต้องใช้ทรัพยากรบุคคล และระบบ Karakuri Kaizen เป็นระบบการลำเลียงสินค้าผ่านทางลาดที่เป็นพื้นลื่น ลูกกลิ้งลื่น (Roller) หรือสายพานที่มีความลาดเอียง โดยไม่จำเป็นต้องใช้ระบบไฟฟ้าในการขับเคลื่อนระบบ ใช้เพียงหลักการของแรงโน้มถ่วง พื้นเอียง ลอก ลูกตุ้ม หรือสปริงในการทำงาน ทำให้ช่วยประหยัดพลังงานทั้งในทางธรรมชาติ และไม่จำเป็นต้องใช้ทรัพยากรมนุษย์ในการเคลื่อนย้ายสินค้า นอกจากนี้ยังเป็นระบบอัตโนมัติที่มีต้นทุนต่ำ ทำให้ได้รับความนิยมอย่างมากในโรงงานอุตสาหกรรม ตัวอย่างเช่น โรงงาน Thai Koito ที่มีนักศึกษาภายในกลุ่มโครงงานของเราได้ไปทำการฝึกงานมา

ดังเหตุที่ได้กล่าวมาข้างต้น จึงเป็นที่มาของโครงงานระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse) ซึ่งจะมุ่งเน้นในการสร้างระบบการเบิกจ่ายสินค้า ที่สามารถทำงานได้อย่างอัตโนมัติ มีการสิ้นเปลืองพลังงาน ทรัพยากรบุคคลน้อย และสามารถตรวจสอบปริมาณสินค้าในคลังสินค้าได้อย่างถูกต้อง โดยจะประกอบด้วยส่วนของฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ที่ใช้การควบคุมผ่าน MCU ให้สามารถทำงานได้ตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาการทำงานของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse)

1.2.2 เพื่อศึกษาการทำงาน และเขียนโปรแกรม (Coding) ของ MCU ชนิด Arduino

1.2.3 เพื่อศึกษาระบบ Karakuri ที่ประยุกต์ใช้ในโรงงานอุตสาหกรรม

1.2.4 เพื่อพัฒนาระบบสำหรับแก้ปัญหาภายในคลังสินค้าของโรงงานอุตสาหกรรมขนาดย่อม

1.3 สมมติฐานของการศึกษา

1.3.1 ระบบควบคุมคลังสินค้าสามารถทำงานได้อัตโนมัติ และเช็คจำนวนสินค้าได้ตามความเป็นจริง

1.3.2 สามารถเขียนโปรแกรมควบคุมการทำงาน และออกแบบระบบวงจรได้อย่างมีประสิทธิภาพ

1.3.3 นำระบบ Karakuri มาประยุกต์ใช้ในโครงงานได้เหมาะสม

1.4 ขอบเขตของการศึกษา

1.4.1 ออกแบบวงจรโดยใช้ Arduino เป็นไมโครคอนโทรลเลอร์

1.4.2 ออกแบบกลไกให้สามารถทำงานได้อัตโนมัติ

1.4.3 มุ่งเน้นการแก้ไขปัญหาจากภายในโรงงานที่เกิดขึ้นจริง

1.4.4 ออกแบบระบบซอฟต์แวร์สำหรับเช็คจำนวนสินค้าโดยโปรแกรม Filemaker

1.5 ระยะเวลาในการทำโครงงาน

1.5.1 ภาคการศึกษาที่ 1 : ตั้งแต่วันที่ 10 สิงหาคม พ.ศ. 2566 ถึงวันที่ 18 ตุลาคม พ.ศ. 2566

รายละเอียด	สัปดาห์ที่									
	สิงหาคม				กันยายน				ตุลาคม	
	1	2	3	4	1	2	3	4	1	2
1. กำหนดหัวข้อโครงงาน	←	→								
2. ศึกษาและหาข้อมูลเกี่ยวกับระบบควบคุมคลังสินค้า		←	—	→						
3. ทำความเข้าใจ ระบบ Karakuri		←	→							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียด	สัปดาห์ที่									
	สิงหาคม				กันยายน				ตุลาคม	
	1	2	3	4	1	2	3	4	1	2
4. ศึกษาโค้ดสำหรับสั่งการในกระบวนการทำงาน										
5. ออกแบบโครงสร้าง										
6. เตรียมรายการอุปกรณ์										
7. จัดซื้ออุปกรณ์										
8. ทำโครงสร้างของส่วนจัดเก็บสินค้า										
9. ศึกษาข้อมูลเกี่ยวกับ Arduino Uno และออกแบบลายวงจร										
10. ประกอบวงจร และโครงสร้าง										
11. เขียนโปรแกรมการทำงาน และปรับระบบการทำงาน										
12. ทดลองการทำงานของโครงสร้าง										
13. บันทึก และสรุปผลการทำงาน										
14. จัดทำรายงานโครงงาน										

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5.2 ภาคการศึกษาที่ 2 : ตั้งแต่วันที่ 1 ธันวาคม พ.ศ. 2566 ถึงวันที่ 14 มีนาคม พ.ศ. 2567

รายละเอียด	สัปดาห์ที่													
	ธันวาคม				มกราคม				กุมภาพันธ์				มีนาคม	
	1	2	3	4	1	2	3	4	1	2	3	4	1	2
1. ศึกษาโค้ดสำหรับสั่งการในกระบวนการทำงานเพิ่มเติม														
2. เตรียมรายการอุปกรณ์														
3. จัดซื้ออุปกรณ์														
4. แก้ไขโครงสร้างบางส่วน														
5. ออกแบบหน้าแอพพลิเคชั่น														
6. เขียนโปรแกรมการทำงาน และปรับระบบการทำงาน														
7. เขียนโปรแกรม Filemaker สำหรับแอพพลิเคชั่น														
8. ทดลองการทำงานของโครงสร้าง ควบคู่กับแอพพลิเคชั่น														
9. บันทึก และสรุปผลการทำงาน														
10. จัดทำรายงานโครงงานเพิ่มเติม														

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้รับความรู้ และความเข้าใจเกี่ยวกับระบบการทำงานของควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ

1.6.2 ระบบควบคุมคลังสินค้าขนาดเล็กสามารถทำงานได้แบบอัตโนมัติ

1.6.3 สามารถออกแบบวงจร และเขียนโปรแกรมคำสั่ง เพื่อให้ระบบทำงานได้อย่างที่ต้องการ

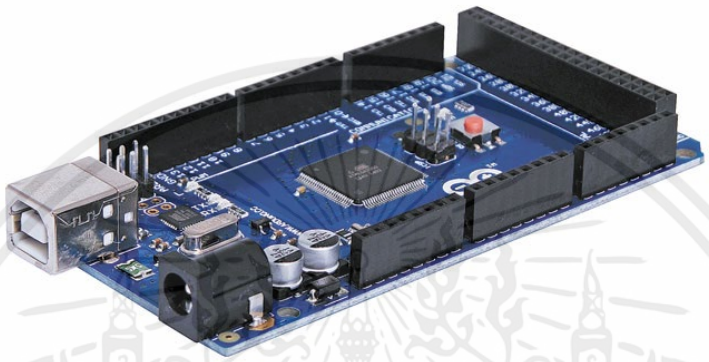
1.6.4 สามารถวิเคราะห์ และแก้ปัญหาต่าง ๆ ที่เกิดขึ้นกับวงจรอิเล็กทรอนิกส์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 Arduino MEGA 2560



รูปที่ 2.1 Arduino Mega 2560

(ที่มา : https://www.altronics.com.au/p/z6241-funduino-mega-2560-r3-compatible-development-board/)

Arduino Mega 2560 คือ บอร์ดไมโครคอนโทรลเลอร์ที่พัฒนาจาก ATmega 2560 มี 54 digital input/output โดยมี 14 ขา สามารถใช้เป็น output แบบ PWM ได้ มี analog inputs 16 ขา มี UARTs (hardware serial ports) 4 ขา ทำงานที่ความถี่ 16 MHz สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยสายเคเบิล USB หรือใช้ adaptor AC-to-DC (Wall-Wart) เพื่อเริ่มต้นใช้งาน บอร์ดสามารถทำงานได้กับแหล่งจ่ายไฟภายนอก 6 ถึง 20 โวลต์ อย่างไรก็ตาม หากจ่ายไฟน้อยกว่า 7V พิน 5V อาจจ่ายไฟน้อยกว่า 5V โวลต์และบอร์ดอาจไม่เสถียร หากใช้มากกว่า 12V ตัวควบคุมแรงดันไฟฟ้าอาจทำให้บอร์ดร้อนเกินไปและทำให้บอร์ดเสียหาย ที่แนะนำช่วง 7 ถึง 12 โวลต์ และมีปุ่ม reset สามารถต่อเข้ากับ shields ที่ออกแบบเพื่อใช้งานกับ Arduino Duemilanove หรือ Diecimila. โดย VIN เป็น input voltage ของบอร์ด Arduino โดยใช้แหล่งจ่ายจากภายนอก ขา 5V เป็น output pin ที่ควบคุม 5 V จากบอร์ด ส่วนขา 3V3 เป็น 3.3 volt supply ที่สร้างขึ้นจาก regulator บนบอร์ด และให้กระแสได้สูงสุด 50 mA GND เป็น ground pin และขา IOREF เป็น pin ที่ให้ voltage reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ shield ที่มาเชื่อมต่อกับบอร์ด สำหรับหน่วยความจำของ ATmega2560 มีหน่วยความจำ 256 KB (8 KB ใช้สำหรับ bootloader) นอกจากนี้ยังมีอีก 8 KB สำหรับ SRAM และ 4 KB สำหรับ EEPROM Arduino Mega 2560 R3 มีการออกแบบรูปแบบให้รองรับการสวมกับ Shield ต่าง ๆ ได้โดยตรง ทำให้สามารถพัฒนาระบบต่าง ๆ ได้อย่างรวดเร็ว และออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลาย ๆ ตัว ทำให้ Pin input และ Output ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่าในความเร็วของ MCU ที่เท่ากัน

2.1.1 คุณสมบัติจำเพาะของ Arduino Mega 2560

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 , 4 UART TTL)
Analog Input Pins	16
DC Current per I/O Pin	20 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
Length :	101.52 mm
Width :	53.3 mm
Weight :	37 g

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ฟังก์ชันของ ARDUINO MEGA 2560

• External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2). pins เหล่านี้สามารถที่จะกำหนดค่าที่เรียก interrupt ในค่าต่ำๆ, ขอบขาขึ้น และลง 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 SPIโดยที่ไม่เกี่ยวข้องกันกับ ICSP header ซึ่งจะมีลักษณะคล้ายกับ Uno, Duemilanove และDiecimila

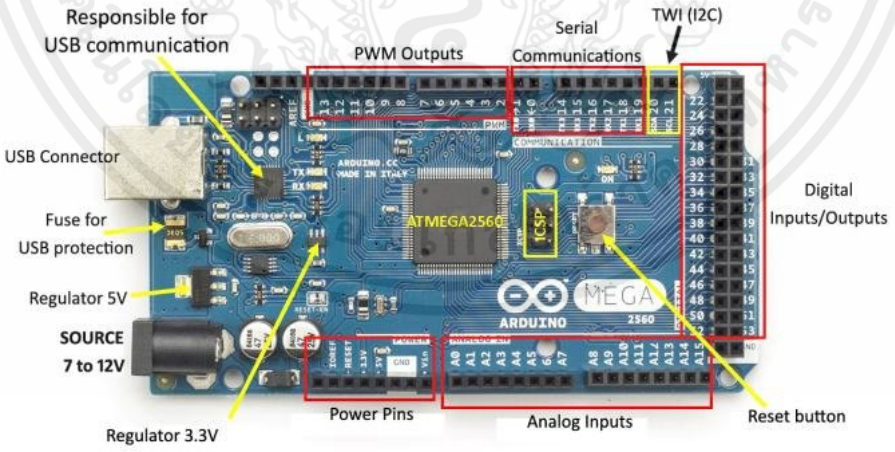
• LED 13 : เป็น build-in LED ที่เชื่อมต่อกับ digital pin 13 เมื่อ pin มีค่าเป็น HIGH LEDจะติด , แต่เมื่อ pin เป็น LOW LED จะดับ

• TWI : 20 (SDA) and 21 (SCL). รองรับการเชื่อมต่อแบบ TWI(I2C)

• บอร์ด Mega2560 มี 16 analog inputs แต่ละ pins ให้ความละเอียด 10 bits

• AREF. แรงดันอ้างอิง สำหรับ analog input

• Reset ใช้ในการ reset ไมโครคอนโทรลเลอร์ โดยทั่วไปจะใช้โดยการเพิ่มปุ่ม reset ไว้บน shield เพื่อป้องกันปุ่มที่อยู่บนบอร์ด



รูปที่ 2.2 Arduino Mega 2560

(ที่มา : https://www.researchgate.net/figure/Block-Diagram-1-ArduinoMEGA2560-The-Arduino-Mega-2560-is-a-type-of-microcontroller_fig5_281538436)

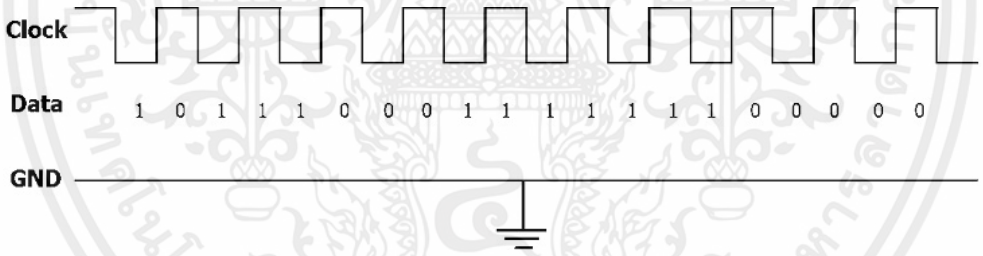
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การสื่อสารบน Arduino Mega 2560

Arduino Mega รองรับอินเทอร์เฟซการสื่อสารได้ทั้งหมด 3 รูปแบบ โดยแต่ละรูปแบบจะถูกแบ่งประเภทจากลักษณะการรับส่งข้อมูล และจะมีการแบ่งขาสำหรับการใช้งานไว้อย่างชัดเจนดังนี้

1. Serial : การสื่อสารแบบ Serial ทำได้โดยการส่งข้อมูลจากตัวส่ง (Transmitter) ไปยังตัวรับ (Receiver) ครั้งละ 1 ไบต์ ดังนั้นการส่งข้อมูลจำนวน 1 ไบต์ จำเป็นต้องใช้สายสัญญาณอย่างน้อย 2 เส้น ในการส่งข้อมูล คือต้องใช้สายสัญญาณ 1 สาย สำหรับการส่งข้อมูล และ 1 สาย สำหรับกราวน์ ซึ่งขา Digital I/O 0 และ 1 ใช้เป็นขา Serial RX0 และ TX0 เพื่อรับและส่งข้อมูล ขาเหล่านี้เชื่อมต่อกับขา Serial ของ USB on-board เป็น Serial Converter IC และสามารถแบ่งการสื่อสารแบบอนุกรมออกเป็น 2 ประเภท ได้แก่

   – แบบซิงโครนัส (Synchronous) เป็นรูปแบบหนึ่งของวิธีการส่งข้อมูล โดยใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะในการรับส่งข้อมูล การส่งข้อมูลแบบนี้ เป็นการรับส่งที่ค่อนข้างมีเสถียรภาพและส่งได้ที่ความเร็วสูง มีโอกาสที่ข้อมูลจะสูญหายระหว่างการส่งน้อย ตัวอย่างการส่งข้อมูลลักษณะนี้เช่น I2C, I2S, SPI



รูปที่ 2.3 Serial communication แบบ Synchronous

(ที่มา : https://blog.thaieasyelec.com/espino32-ch7-how-to-use-uart/1)

   – แบบอะซิงโครนัส (Asynchronous) เป็นการส่งข้อมูลที่ไม่ต้องใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะการรับส่งข้อมูล แต่ใช้วิธีกำหนดรูปแบบการรับส่งข้อมูลขึ้นมาแทน และ อาศัยการกำหนดอัตราเร็วของการรับ และส่งที่เท่ากันทั้งฝั่งรับและฝั่งส่ง ข้อดีของการใช้ Asynchronous คือสามารถสื่อสารแบบ Full Duplex รับ และ ส่งได้ในเวลาเดียวกัน แต่การสื่อสารแบบ Asynchronous มีโอกาสที่ข้อมูลจะสูญหายขณะรับหรือส่งข้อมูลมากกว่าแบบ Synchronous และ UART เป็นประเภทหนึ่งของการสื่อสารแบบอะซิงโครนัส ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter หมายถึง อุปกรณ์ทางคอมพิวเตอร์ที่ผู้ใช้

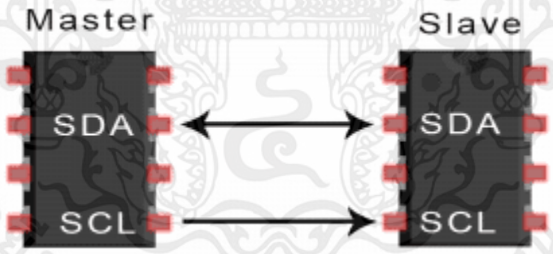
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถกำหนดรูปแบบที่ใช้ในการรับส่งข้อมูล กำหนดอัตราเร็วในการรับส่งข้อมูลในหน่วยบิตต่อวินาทีหรือบอดเรท (Baud rate) ให้แก่อุปกรณ์ที่จะติดต่อสื่อสารกันให้มีค่าตรงกันจึงจะสามารถสื่อสารกันได้ สายสัญญาณที่ใช้จะใช้สาย 2 เส้น คือ Tx ในการส่งข้อมูล และ Rx

รูปที่ 2.4 Serial communication แบบ Asynchronous

(ที่มา : https://blog.thaieasyelec.com/espino32-ch7-how-to-use-uart/1)

2. Inter-Integrated Circuit (I2C) หรือ Two Wire Interface (TWI) : เป็นการสื่อสารโดยแบ่งเป็น 2 ส่วน คือ มาสเตอร์ (Master) ซึ่งเป็นส่วนหลักในการสั่งการการควบคุม และสลาฟ (Slave) ที่จะทำงาน เมื่อมีการส่งข้อมูลจากมาสเตอร์เท่านั้น ซึ่งข้อดีของระบบนี้สามารถใช้ มาสเตอร์ 1 ตัวควบคุมการทำงานของสลาฟได้หลายตัว อาทิเช่น การสั่งการการทำงานแสดงผลบนหน้าจอ LCD และใน Arduino Mega สามารถกำหนดค่า Digital I/O Pins 20 และ 21 เป็น SDA (20) และ SCL (21) เพื่อรองรับการสื่อสาร

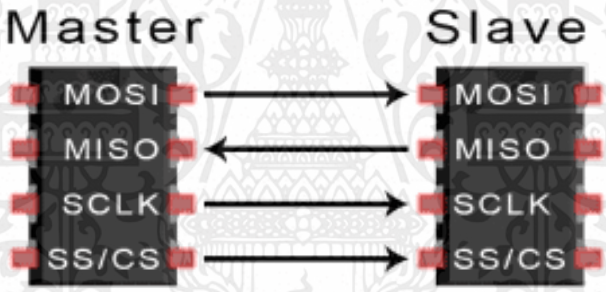


รูปที่ 2.5 I2C communication

(ที่มา : https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Serial Peripheral Interface (SPI) : เป็นโปรโตคอลการสื่อสารทั่วไปที่ใช้โดยอุปกรณ์ต่างๆ มากมาย ตัวอย่างเช่น โมดูลตัวอ่านการ์ด SD , โมดูลตัวอ่านการ์ด RFID และเครื่องส่ง/รับสัญญาณไร้สาย 2.4 GHz ล้วนใช้ SPI เพื่อสื่อสารกับไมโครคอนโทรลเลอร์ ข้อดีอย่างหนึ่งที่เป็นเอกลักษณ์ของ SPI คือความจริงที่ว่าข้อมูลสามารถถ่ายโอนได้โดยไม่หยุดชะงัก สามารถส่งหรือรับบิตจำนวนเท่าใดก็ได้ในสตรีมต่อเนื่อง ด้วย I2C และ UART ข้อมูลจะถูกส่งเป็นแพ็กเก็ต ซึ่งจำกัดจำนวนบิตที่ระบุ เงื่อนไขการเริ่มต้นและหยุดจะกำหนดจุดเริ่มต้นและจุดสิ้นสุดของแต่ละแพ็กเก็ต ดังนั้นข้อมูลจึงถูกขัดจังหวะระหว่างการส่ง อุปกรณ์ที่สื่อสารผ่าน SPI มีความสัมพันธ์แบบมาสเตอร์-สลาฟ หลักคืออุปกรณ์ควบคุม (โดยปกติคือไมโครคอนโทรลเลอร์) ในขณะที่สลาฟ (โดยปกติคือเซ็นเซอร์ จอแสดงผล หรือชิปหน่วยความจำ) รับคำสั่งจากต้นแบบ การกำหนดค่าที่ง่ายที่สุดของ SPI คือระบบมาสเตอร์เดี่ยวหรือระบบทาสเดี่ยว แต่ต้นแบบหนึ่งตัวสามารถควบคุมทาสได้มากกว่าหนึ่งตัว และภายใน Arduino Mega SPI Digital I/O Pins 50, 51 52 และ 53 สามารถกำหนดค่าเป็น SPI pins MISO, MOSI, SCK และ SS ตามลำดับ



รูปที่ 2.6 SPI communication

(ที่มา : https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Servo motor

รูปที่ 2.7 Servo motor

(ที่มา http://robo-circuit.com/product/512/mg996r-15kg-torque-servo-motor-180)

เซอร์โวมอเตอร์ (Servo Motor) เป็นมอเตอร์ที่มีการควบคุมการเคลื่อนที่ (State) ไม่ว่าจะเป็นระยะ ความเร็ว มุมการหมุน โดยใช้การควบคุมแบบป้อนกลับ (Feedback control) เป็นอุปกรณ์ที่สามารถควบคุมเครื่องจักรกล หรือระบบการทำงานนั้นๆ ให้เป็นไปตามความต้องการ เช่น ควบคุมความเร็ว (Speed), ควบคุมแรงบิด (Torque), ควบคุมแรงตำแหน่ง (Position), ระยะทางในการเคลื่อนที่การหมุน (Position Control) ของตัวมอเตอร์ได้ ซึ่งมอเตอร์ทั่วไปไม่สามารถควบคุมลักษณะงานเบื้องต้นได้ โดยให้ผลลัพธ์ตามความต้องการที่มีความแม่นยำสูง ขนาดของ Servo Motor จะมีหน่วยในการบอกขนาดเป็นวัตต์ (Watt)

2.2.1 หลักการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ประกอบด้วย 4 ส่วนหลัก คือ มอเตอร์กระแสตรง (DC Motor) ชุดเฟืองทดรอบ (Gear system) วอลุ่ม (Potentiometer หรือ VR) และวงจรควบคุม (Control Electronics) โดยหลักการทำงานของเซอร์โวมอเตอร์เริ่มที่วงจรควบคุม เมื่อวงจรควบคุมได้รับข้อมูลองศาที่ต้องการมาแล้ว วงจรควบคุมจะคำนวณว่ามอเตอร์จะต้องหมุนในทิศทางตามเข็มนาฬิกา หรือทวนเข็มนาฬิกา เพื่อให้ไปสู่องศาที่ต้องการได้ เมื่อมอเตอร์เริ่มหมุน ตัววอลุ่มที่ติดอยู่กับชุดเฟืองมอเตอร์จะตรวจสอบตำแหน่งที่มอเตอร์หมุนไป โดยหากวอลุ่มตรวจพบว่าตำแหน่งที่มอเตอร์หมุนเริ่มใกล้กับองศาที่ผู้ใช้กำหนด วงจรส่วนควบคุมจะเริ่มสั่งให้มอเตอร์หมุนช้าลงเพื่อให้หมุนเข้าใกล้องศาที่กำหนดได้มากที่สุด เมื่อมอเตอร์หมุนได้ตำแหน่งองศาที่ถูกต้องแล้ว วงจรส่วนควบคุมจะตรวจสอบตำแหน่งของมอเตอร์เป็นระยะ ๆ โดยอ่านค่าจากวอลุ่ม หากตรวจพบว่าตำแหน่งผิดเพี้ยนไปจากค่าที่ตั้งไว้ อาจเกิดจากผู้ใช้เอามือไปหมุนเล่น หรือภาระส่งผลให้ตำแหน่งเคลื่อน วงจรควบคุมจะสั่งให้มอเตอร์หมุนกลับมาให้ได้ตำแหน่งเป็นระยะ ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ชนิดของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์แบ่งออกเป็น 2 ชนิด ดังนี้

1. เซอร์โวมอเตอร์ 180 องศา เป็นเซอร์โวมอเตอร์ที่นิยมใช้งานทั่วไป มีหลายรุ่น หลายขนาด และหลายราคา สามารถควบคุมให้หมุนได้ตามองศาที่ต้องการ โดยหมุนได้ 0 ถึง 180 องศา แต่ในบางรุ่นหมุนได้สุดที่ประมาณ 200 องศา
2. เซอร์โวมอเตอร์ 360 องศาเป็นเซอร์โวมอเตอร์ที่ส่วนใหญ่ดัดแปลงมาจากแบบ 180 องศา โดยดัดแปลงวงจรควบคุม และตัดแกนหรือนำเอาวอลุ่มออก เพื่อให้เซอร์โวมอเตอร์สามารถหมุนได้ครบรอบ เซอร์โวมอเตอร์ชนิดนี้ไม่สามารถควบคุมองศาได้ ควบคุมได้แค่ความเร็ว และทิศทางการหมุนเท่านั้น นิยมนำมาใช้เป็นมอเตอร์สำหรับรถบังคับ รถวิ่งตามเส้น เพราะอาศัยชุดเฟืองที่เซอร์โวมอเตอร์มีอยู่แล้ว ทำให้ได้แรงบิดที่มากกว่ามอเตอร์กระแสตรงปกติ รวมทั้งการควบคุมยังไม่ต้องใช้วงจรขับมอเตอร์แยก ทำให้ลดความยุ่งยากในการต่อวงจร รวมทั้งประหยัดค่าใช้จ่ายด้วย

2.2.3 การใช้งานเซอร์โวมอเตอร์

เซอร์โวมอเตอร์มีสาย 3 เส้น ประกอบด้วย

Signal (สีส้ม หรือ สีขาว) - สายสัญญาณควบคุมการหมุนแบบ PWM

VCC (สีแดง) - สายสำหรับจ่ายไฟบวก 5V

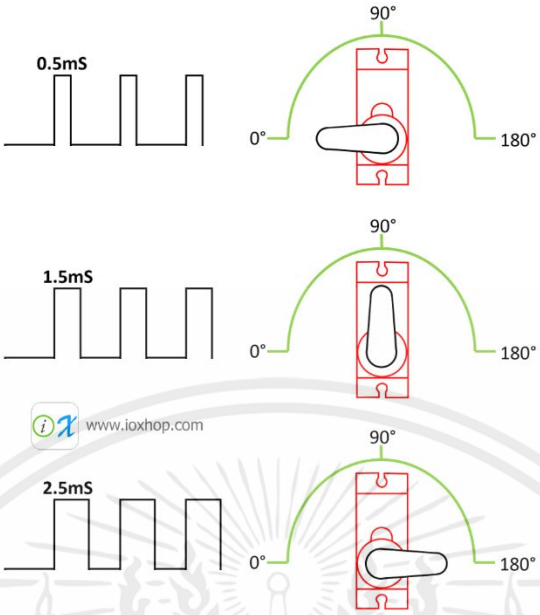
GND (สีน้ำตาล หรือ สีดำ) - สายสำหรับจ่ายไฟลบ หรือกราวด์ (GND)

การควบคุมการหมุนของเซอร์โวมอเตอร์จะทำที่สาย Signal โดยป้อนสัญญาณ PWM ความถี่ 50Hz เข้าไป โดยมีความกว้างพัสล์บวกที่ 0.5mS (ค่าต่ำสุด) ถึง 2.5mS (ค่าสูงสุด) หรือ 1mS (ค่าต่ำสุด) ถึง 2mS (ค่าสูงสุด) ตามแต่รุ่นของเซอร์โวมอเตอร์ โดยหากป้อนสัญญาณ PWM ที่มีความกว้างช่วงบวกเข้าไปเท่าค่าต่ำสุด เซอร์โวมอเตอร์จะหมุนไปที่ 0 องศา หากป้อนสัญญาณ PWM เข้าไปเท่าค่าสูงสุด เซอร์โวมอเตอร์จะหมุนไปที่ 180 องศา

โดยการหาค่าความกว้างช่วงบวกของสัญญาณ PWM จากค่าองศา สามารถหาได้จากสูตร

ความกว้างพัสล์บวก =((ค่าสูงสุด-ค่าต่ำสุด)/180) X องศาที่ต้องการ + ค่าต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

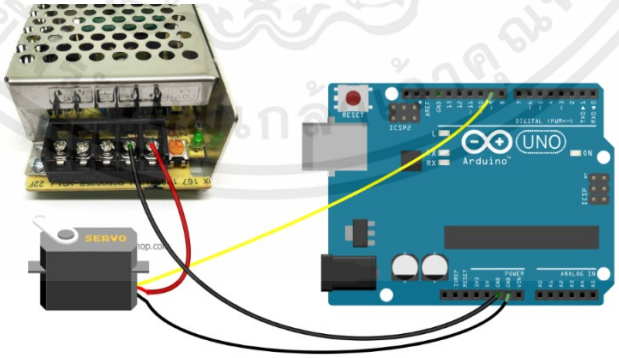


รูปที่ 2.8 สัญญาณ PWM ที่ส่งผลต่อองศาการหมุนของเซอร์โวมอเตอร์

(ที่มา http://www.advance-electronic.com/blog/detail/86/th/)

2.2.4 การจ่ายไฟให้เซอร์โวมอเตอร์

เซอร์โวมอเตอร์เป็นอุปกรณ์ที่ใช้กระแสไฟฟ้ามาก หากใช้เซอร์โวมอเตอร์รุ่นเล็กอย่าง SG90 หรือ MG90S ไฟที่มาจากช่อง USB ของคอมพิวเตอร์ ยังพอสำหรับการทำงานของเซอร์โวมอเตอร์ แต่หากเป็นเซอร์โวมอเตอร์ที่มีขนาดใหญ่ แรงบิดสูง อย่าง Futaba S3003 หรือ MG995 จำเป็นต้องใช้แหล่งจ่ายไฟภายนอกเพิ่ม โดยอาจจะเลือกใช้อแดปเตอร์ 5V มาจ่ายไฟเลี้ยงเซอร์โวมอเตอร์เพิ่ม หรือเลือกใช้สวิตชิ่ง 5V



รูปที่ 2.9 ตัวอย่างการต่อไฟเลี้ยงเซอร์โวมอเตอร์เพิ่มเติมโดยใช้สวิตชิ่ง 5V 5A

(ที่มา https://www.artronshop.co.th/article/92/)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 การเลือกใช้งานเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ส่วนใหญ่ใช้แรงดันไฟฟ้า 5V และมีองศาการหมุนที่ 0 ถึงประมาณ 200 องศา (ยกเว้นมีการดัดแปลงให้หมุน 360 องศา) นอกจาก 2 ข้อนี้ที่เซอร์โวมอเตอร์ทุกรุ่นมีเหมือนกัน ยังมีอีก 4 ข้อที่เซอร์โวมอเตอร์แต่ละรุ่นมีไม่เหมือนกัน

1. แรงบิด (Torque) - เป็นเลขบอกกำลังของเซอร์โวมอเตอร์ ยิ่งมีค่ามาก แสดงว่าเซอร์โวมอเตอร์มีแรงมาก
2. ความเร็วในการหมุน (Speed) - เป็นตัวเลขที่บอกว่าเซอร์โวมอเตอร์มีความสามารถสำหรับการเปลี่ยนตำแหน่งได้เร็วแค่ไหน ยิ่งมีค่ามาก แสดงว่าเซอร์โวมอเตอร์มีความเร็วในการเปลี่ยนตำแหน่งที่เร็วมาก
3. วัสดุที่ใช้ทำแกนหมุน โดยวัสดุที่ใช้ทำแกนหมุนของเซอร์โวมอเตอร์มี 2 ชนิด คือ พลาสติก และเหล็ก สำหรับแกนหมุนพลาสติกเมื่อใช้งานเซอร์โวมอเตอร์อย่างหนักเป็นเวลานาน จะทำให้เฟืองของเซอร์โวมอเตอร์ครูด ดังนั้นหากนำเซอร์โวมอเตอร์ไปใช้งานหนักเป็นเวลานาน จึงควรเลือกแกนเหล็ก เพราะแกนเหล็กมีโอกาศที่เฟืองครูดได้น้อยกว่า
4. ขนาดของแกนหมุน สำหรับเซอร์โวมอเตอร์แต่ละรุ่นจะมีขนาดของแกนหมุนที่แตกต่างกัน ตามแรงบิดของเซอร์โวมอเตอร์แต่ละรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Liquid Crystal Display (LCD)

รูปที่ 2.10 Liquid Crystal Display (LCD)
(ที่มา https://th.cytron.io/p-i2c-1602-serial-lcd-for-arduino-and-rpi)

LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตอลเหลว โดยมีหลักการทำงานคือ ด้านหลังจอจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight อยู่ เมื่อมีการจ่ายกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึก ตัวผลึกก็จะเกิดความโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอ ส่วนอื่นที่โดนผลึกปิดกั้นไว้จะไม่สว่าง ผลึกคริสตอลมีหลากหลายสี แต่ที่จะพบเห็นได้บ่อยในท้องตลาดส่วนใหญ่เป็นสีเขียว และสีฟ้า ทำให้เมื่อมองไปที่จอก็จะพบกับตัวหนังสือ พร้อมกับพื้นหลังสีต่างๆ

จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่นิยมนำมาใช้งานกันกับระบบสมองกลฝังตัวอย่างแพร่หลาย ซึ่งจะมีทั้งแบบแสดงผลเป็นตัวอักขระเรียกว่า Character LCD มีการกำหนดตัวอักษร หรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็นรูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD นอกจากนี้บางชนิดเป็นจอที่มีการผลิตขึ้นมาใช้เฉพาะงาน ทำให้มีรูปแบบและรูปร่างเฉพาะเจาะจงในการแสดงผล เช่น นาฬิกาดิจิตอล เครื่องคิดเลข หรือ หน้าปัดวิทยุ เป็นต้น จอ LCD แบ่งได้เป็น 2 ประเภทตามลักษณะการแสดงผลดังนี้

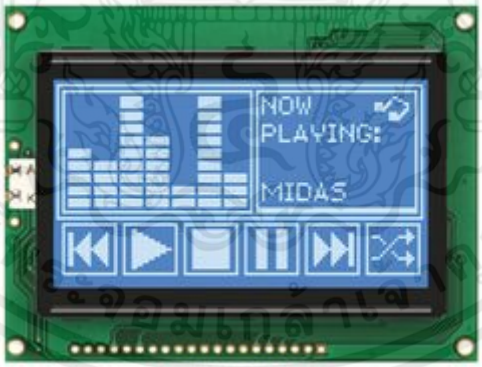
1. Character LCD เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่น จอ LCD ขนาด 16X2 หมายถึงใน 1 แถวมีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20x4 จะหมายถึงใน 1 แถวมี ตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 Character LCD

(ที่มา https://blog.thaieasyelec.com/how-to-use-character-lcd-display-arduino-ch1-parallel/)

2. Graphic LCD เป็นจอที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอกั้นแสง หรือปล่อยแสงออกไป ทำ ให้จอนี้สามารถสร้างรูปขึ้นมาบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด หลักการทำงานอาศัยของเหลวพิเศษที่มีคุณสมบัติการบิดแกนโพราไรซ์ของแสง ถ้าจ่ายแรงดันไฟฟ้าเข้าไป ระหว่างสารเหลวนี้ โมเลกุลจะบิดตัวและทำให้แสงไม่สามารถผ่านกระจกออกมาได้ ถ้าไม่มีการจ่ายแรงดันไฟฟ้าแสงจะทะลุผ่านออกมาได้



รูปที่ 2.12 Graphic LCD

(ที่มา https://th.element14.com/midas/mc128064a6w-bnmlw-v2/display-lcd-graphic-128x64-bstn/dp/2675690)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 โครงสร้างโดยทั่วไปของ LCD

โครงสร้างของ LCD ทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่ โดยเว้นช่องว่างตรงกลางไว้ 6-10 ไมโครเมตร ผิวด้านในของแผ่นแก้วจะเคลือบด้วยตัวนำไฟฟ้าแบบใสเพื่อใช้แสดงตัวอักษรตรงกลางระหว่าง ตัวนำไฟฟ้าแบบใสกับผลึกเหลว มีชั้นของสารที่ทำให้โมเลกุลของผลึกรวมตัวกันในทิศทางที่แสงส่องมากระทบ เรียกว่า Alignment Layer และผลึกเหลวที่ใช้โดยทั่วไปจะเป็นแบบ Magnetic โดย LCD สามารถแสดงผลได้ทั้งหมด 3 แบบคือ

1. แบบใช้การสะท้อนแสง (Reflective Mode) : LCD แบบนี้ใช้สารประเภทโลหะเคลือบอยู่ที่แผ่นหลังของ LCD ซึ่ง LCD ประเภทนี้เหมาะกับการนำมาใช้งานในที่ที่มีแสงสว่างเพียงพอ

2. แบบใช้การส่งผ่าน (Transitive Mode) : LCD แบบนี้จะต้องวางหลอดไฟไว้ด้านหลังจอ เพื่อทำให้การอ่านค่าที่แสดงผลทำได้ชัดเจนมากยิ่งขึ้น

3. แบบส่งผ่านและสะท้อน (Transflective Mode) : LCD แบบนี้เป็นการนำเอาข้อดีของจอแสดงผล LCD ทั้ง 2 แบบมารวมกัน

2.3.2 การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัว ผู้ใช้งานสามารถส่ง รหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกันกับจอ LCD แบบธรรมดา แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล โดยในโครงงานจะเลือกใช้จอ LCD 16x2 ที่มีการส่งข้อมูลรูปแบบ I2C ที่ใช้ขาเพียง 4 ขาที่ใช้ในการเชื่อมต่อเท่านั้น โดยการควบคุมหรือสั่งงานจอ LCD Controller และขาในการเชื่อมต่อระหว่าง LCD กับ Arduino มีดังนี้

1. GND เป็นกราวด์ใช้ต่อระหว่าง Ground ของ Arduino กับ LCD

2. V_{CC} เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD ขนาด $+5V_{DC}$

3. V_O ใช้ปรับความสว่างของหน้าจอ LCD

4. RS จะทำให้ LCD Controller ทราบว่า Code ที่ส่งมาที่ขา Data เป็นคำสั่งหรือข้อมูล

5. R/W ใช้กำหนดว่าจะอ่านหรือเขียนข้อมูลกับ LCD Controller

6. E เป็นขา Enable หรือ Chips Select ใช้เพื่อกำหนดการทำงานให้กับ LCD Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. DB0-D87 เป็นขาสัญญาณ Data ใช้สำหรับเขียนหรืออ่านข้อมูล/คำสั่ง กับ LCD Controller

วิธีการสั่งงานจะแตกต่างกันไปโดย LCD Controller สามารถรับรหัสคำสั่งจาก Arduino ได้จากสัญญาณ RS R/W และ DB0-DB7 ในขณะที่สัญญาณ E มีค่า Logic เป็น “1” ซึ่งสัญญาณเหล่านี้จะใช้ร่วมกันเพื่อกำหนด เป็นรหัสคำสั่งสำหรับสั่งงาน LCD โดยหน้าที่ของแต่ละสัญญาณสรุปได้ดังนี้

1. E เป็นสัญญาณ Enable เมื่อมีค่าเป็น

“1” เป็นการบอกให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการติดต่ออ่านหรือเขียนข้อมูล

“0” ให้ LCD ไม่สนใจสัญญาณ RS R/W และ D87-DB0

2. RS เป็นสัญญาณสำหรับกำหนดให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการติดต่อกับ LCD ในขณะนั้น เป็นรหัสคำสั่งหรือข้อมูล

3. R/W เป็นสัญญาณสำหรับบอกให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการอ่านหรือเขียนกับ LCD โดย

- RW = “0” หมายถึง เขียน
- RW = “1” หมายถึง อ่าน

4. DB0-D87 เป็นสัญญาณแบบ 2 ทิศทาง โดยจะสัมพันธ์กับสัญญาณ R/W ใช้สำหรับรับคำสั่งและข้อมูล ระหว่าง LCD กับอุปกรณ์ภายนอก โดยถ้า R/W = “0” สัญญาณ DB7-DB0 จะส่งจากอุปกรณ์ภายนอกมาที่ LCD แต่ถ้า R/W = “1” สัญญาณ DB7-DBO จะส่งจาก LCD ไปยังอุปกรณ์ภายนอก

การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller สามารถทำได้ 2 ลักษณะ คือ การ เชื่อมต่อแบบ 8 บิต (DB0-DB7) และการเชื่อมต่อแบบ 4 บิต (DB4-D87) ทั้งสองแบบแตกต่างกันเพียงจำนวนขาที่ ใช้คือ 8 หรือ 4 ขา และยังสามารถทำงานได้เหมือนกัน อย่างที่แน่นอนในการส่งข้อมูลแบบ 4 ขา ผลย่อมทำได้ช้า กว่า 8 ขา แต่ไม่ได้ช้ามากจนสังเกตุได้ด้วยสายตา โดยทั่วไปนิยมใช้การต่อแบบ 4 ขาเพื่อประหยัดจำนวนขาในการเชื่อมต่อกับ Arduino

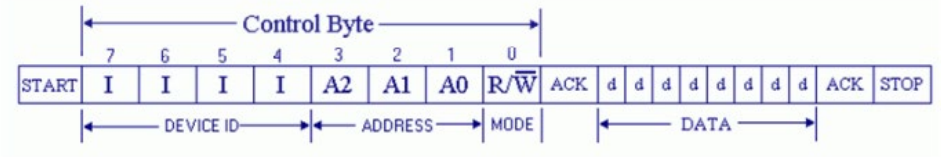
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงตัวอย่างการเชื่อมต่อขากับ Arduino

VR 10 kΩ	LCD	Arduino
GND	V_{SS}/GND	GND
V_{CC}	V_{DD}	+5 V_{DC}
Signal	V_O/V_{EE}	-
-	RS	PIN 12
-	RW	GND
-	E/EN	PIN 11
-	DB4	PIN 4
-	DB5	PIN 5
-	DB6	PIN 6
-	DB7	PIN 7
-	A	+5 V_{DC}
-	K	GND

ลำดับขั้นตอนการการรับ-ส่งข้อมูลแบบ I2C BUS มีดังนี้

1. MCU จะทำการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส
2. ตามด้วยรหัสควบคุม (Control Byte) ซึ่งประกอบ ด้วยรหัสประจำตัวอุปกรณ์ Device ID, Device Address และ Mode ในการเขียนหรืออ่านข้อมูล
3. เมื่ออุปกรณ์รับทราบว่า MCU ต้องการจะติดต่อด้วย ก็ต้องส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง
4. เมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่าสิ้นสุดการใช้บัส



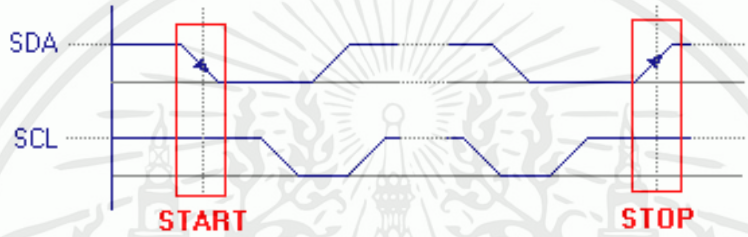
รูปที่ 2.13 รูปแบบการเขียน/อ่านข้อมูลแบบ **I2C Bus**

(ที่มา https://www.thaimicrotron.com/CCS-628/Referrence/I2CBUS.htm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดสถานะต่างๆของ I2C Bus

1. สถานะว่าง : บัสไม่ได้ถูกใช้งาน ทั้ง SCL และ SDA จะเป็น 1 ทั้งคู่
2. สถานะเริ่มต้น (Start condition) : เมื่อต้องการส่งข้อมูล MCU จะต้องส่งสถานะเริ่มต้น (START Conditions) คือให้ SDA เปลี่ยนจาก 1 มาเป็น 0 ในขณะที่ SCL มีค่าเป็น 1
3. สถานะสิ้นสุด (Stop condition) : เมื่อสิ้นสุดการการใช้บัส MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) คือให้ SDA เปลี่ยนจาก 0 มาเป็น 1ในขณะที่ SCL มีค่าเป็น 1



รูปที่ 2.14 I2C Bus START Conditions and STOP Conditions

(ที่มา https://www.thaimicrotron.com/CCS-628/Referrence/I2CBUS.htm)

2.3.3 การเชื่อมต่อ Arduino กับจอ Character LCD

1. การเชื่อมต่อแบบขนาน - เป็นการเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง โดยจะแบ่งเป็น เชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ใน Arduino จะนิยมเชื่อมต่อแบบ 4 บิต เนื่องจากใช้การต่อน้อยกว่า

2. การเชื่อมต่อแบบอนุกรม - เป็นการเชื่อมต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบการเชื่อมต่อกับจอ LCD จากแบบขนานมาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface จะเป็นการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ด Arduino เชื่อมต่อกับบอร์ดโมดูลผ่านโปรโตคอล I2C ทำให้ใช้สายเพียง 4 เส้นก็สามารถทำให้หน้าจอแสดงผลข้อความต่างๆออกมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 MH-ET LIVE Scanner V3.0

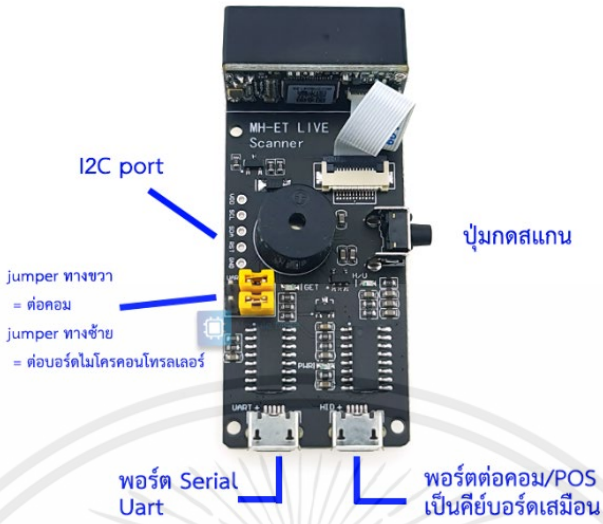
รูปที่ 2.15 MH-ET LIVE Scanner v3.0
(ที่มา https://www.mikroelec.com/product/2075/barcode-and-qr-code-scanner-module-v3-0-1d-2d-codes-reader-two-dimensional-scanning-engine-barcode-s)

MH-ET LIVE Scanner v3.0 เป็นโมดูลที่ใช้สำหรับการสแกนบาร์โค้ดและรหัส QR ใช้ชิปประมวลผลภาพที่เชี่ยวชาญเพื่อการจำบาร์โค้ด สามารถอ่านได้อย่างรวดเร็วและแม่นยำในสภาพแวดล้อมที่ซับซ้อน สามารถอ่านบาร์โค้ดแบบหนึ่งมิติและสองมิติที่ปรากฏบนพื้นผิวของกระดาษ หน้าจอ พลาสติก และอื่น ๆ โมดูลนี้มีขนาดเล็ก มีอินเตอร์เฟซ USB และ UART บนบอร์ด สามารถเชื่อมต่อโดยตรงกับคอมพิวเตอร์หรือผสมผสานในอุปกรณ์ต่าง ๆ มีการใช้พลังงานต่ำและไม่ก่อให้เกิดความร้อนมาก สามารถเริ่มต้นการทำงานได้อย่างรวดเร็ว และสามารถรองรับหลายโหมดในการเริ่มไว้สำหรับการพร้อมใช้งานโดยไม่ต้องรอ พร้อมใช้งานทันทีที่เปิดใช้งาน MH-ET LIVE Scanner v3.0 เป็นเครื่องแสกนประเภท CCD ซึ่งมีการทำงาน โดยไฟในตัว เครื่องสแกนฉายแสงนี้ไปที่บาร์โค้ด และการสะท้อนของบาร์โค้ดจะถูกบันทึกผ่าน CCD เพื่ออ่านบาร์โค้ดจะถูกจับเพียงครั้งเดียว ช่วยให้อ่านได้อย่างรวดเร็ว ไม่มีชิ้นส่วนที่สามารถเคลื่อนย้ายได้และทนต่อแรงกระแทกได้ดีเยี่ยม

2.4.1 คุณสมบัติของ MH-ET LIVE Scanner V3.0

1. ง่ายต่อการใช้งาน : ไม่ต้องเข้าใจอัลกอริทึมการรู้จำภาพที่ซับซ้อน ทำให้ง่ายต่อการใช้งาน
2. รองรับการอ่านบาร์โค้ดแบบทั่วไป : สนับสนุนในการอ่านบาร์โค้ดแบบหนึ่งมิติและสองมิติ เช่น บาร์โค้ดและรหัส QR ที่พบบ่อย
3. มีพอร์ต micro USB และ UART บนบอร์ด : สามารถเข้าถึงคอมพิวเตอร์หรืออุปกรณ์ได้ผ่านพอร์ต micro USB และ UART ที่มีอยู่บนบอร์ด

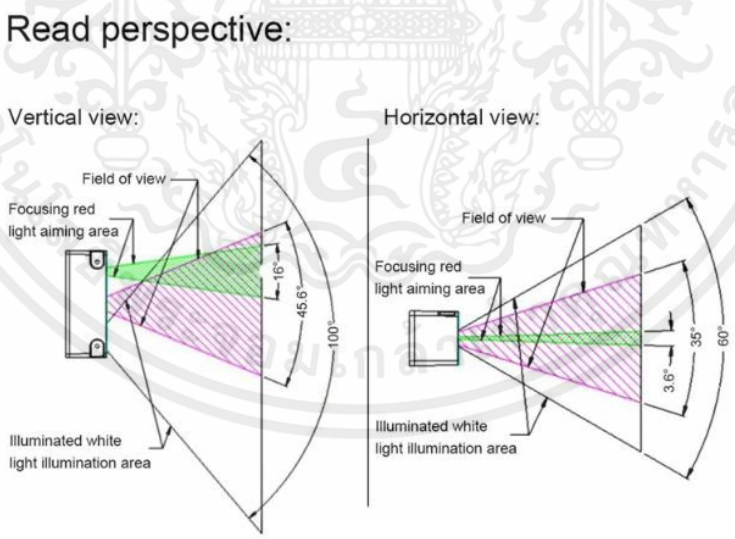
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 โครงสร้าง MH-ET LIVE Scanner v3.0

(ที่มา https://www.modulemore.com/en/product/2094/mh-et-live-scanner-v3-barcode-reading-board-qr-code-scan)

4. มีแหล่งแสงช่วยอยู่บนบอร์ด : มีแหล่งแสงช่วยที่ติดอยู่บนบอร์ด เพื่อการอ่านในสภาพแวดล้อมที่มืด โดยแหล่งแสงจะช่วยในมุมมองการอ่าน



รูปที่ 2.17 มุมมองการอ่านบาร์โค้ด

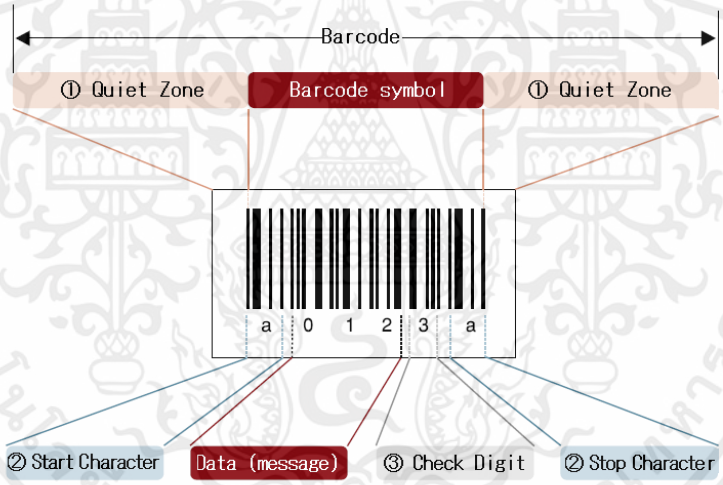
(ที่มา https://www.mikroelec.com/product/2075/barcode-and-qr-code-scanner-module-v3-0-1d-2d-codes-reader-two-dimensional-scanning-engine-barcode-s)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. รองรับการตั้งค่าพารามิเตอร์ของโมดูลผ่านการสแกนโค้ดตั้งค่า : สนับสนุนการตั้งค่าพารามิเตอร์ของโมดูลโดยการสแกนโค้ดตั้งค่า
6. พารามิเตอร์ที่สามารถตั้งค่าได้ประกอบด้วย (โหมดอ่าน, การตั้งค่าอัตราบอด, โหมดคำสั่ง): สามารถตั้งค่าโหมดการอ่าน, อัตราบอดการสื่อสาร, และโหมดคำสั่งได้ตามที่ต้องการ

2.4.2 บาร์โค้ด และ ประเภทของการอ่านบาร์โค้ดที่ MH-ET LIVE Scanner V3.0 อ่านได้

บาร์โค้ดที่ประกอบด้วยแท่งและช่องว่าง คือการแสดงตัวเลขและอักขระที่เครื่องสามารถอ่านได้ ปัจจุบัน ลายทางที่แสดงด้านล่างบนบรรจุภัณฑ์ของผลิตภัณฑ์ที่จำหน่ายในซูเปอร์มาร์เก็ต ร้านสะดวกซื้อ และร้านค้าอื่นๆ แพร่หลาย เหล่านี้คือบาร์โค้ด บาร์โค้ดประกอบด้วยแท่งและช่องว่างที่มีความกว้างต่างกันซึ่งสามารถอ่านได้ด้วยเครื่องสแกนบาร์โค้ดแบบออปติคัล และมีส่วนประกอบดังนี้



รูปที่ 2.18 องค์ประกอบของ Barcode

(ที่มา https://www.denso-wave.com/en/adcd/fundamental/barcode/barcode/index.html)

1. Quiet Zone คือ ระยะขอบว่างที่ปลายทั้งสองด้านของบาร์โค้ด ขอบขั้นต่ำระหว่างบาร์โค้ด (ระยะห่างจากแถบด้านนอกสุดของบาร์โค้ดหนึ่งไปยังแถบด้านนอกสุดของอีกบาร์โค้ดหนึ่ง) คือ 2.5 มม. หากความกว้างของเขตเงียบไม่เพียงพอ เครื่องสแกนจะอ่านบาร์โค้ดได้ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Start Character and the Stop Character คือ เป็นอักขระที่แสดงถึงจุดเริ่มต้นและจุดสิ้นสุดของข้อมูลตามลำดับ อักขระจะแตกต่างกันไปขึ้นอยู่กับประเภทบาร์โค้ด
3. Check Digit คือ ตัวเลขสำหรับตรวจสอบว่าข้อมูลบาร์โค้ดที่เข้ารหัสนั้นถูกต้องหรือไม่

นอกจากนี้ยังแบ่งประเภทของบาร์โค้ดที่สารอ่านได้ออกเป็น 2 ประเภทสำหรับ MH-ET LIVE Scanner V3.0

1. บาร์โค้ดประเภท 1 มิติ (1D) : UPC/EAN, UPC/EAN with supplementals, BooklandEAN, ISSN, UCC Coupon Extended Code, Code 128, GS1-128, ISBT 128, Code 39, Code 39 Full ASCII, Trioptic Code 39, Code 32, Code 93, Code 11, Matrix 2 of 5, Interleaved 2 of 5, Discrete 2 of 5, Codabar, MSI, Chinese 2 of 5, GS1 DataBar variants, Korean 3 of 5, ISBT Concat

รูปที่ 2.19 ตัวอย่าง Barcode ในรูปแบบ 1D
(ที่มา https://www.tiit.co.th/th/articles/105787-1d-barcode-and-2d-barcode)

2. **บาร์โค้ดประเภท 2 มิติ (2D) : QR Code, PDF417, Data Matrix**

รูปที่ 2.20 ตัวอย่าง Barcode ในรูปแบบ 2D
(ที่มา https://www.tiit.co.th/th/articles/105787-1d-barco2de-and-2d-barcode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 หลักการอ่านบาร์โค้ด

รูปที่ 2.21 ขั้นตอนการอ่านบาร์โค้ด

(ที่มา https://www.denso-wave.com/en/adcd/fundamental/barcode/scan/index.html)

จากภาพข้างต้นมีขั้นตอนดังนี้

1. บาร์โค้ดประกอบด้วยแถบสีขาวและสีดำ การดึงข้อมูลทำได้สำเร็จเมื่อเครื่องสแกนบาร์โค้ดฉายแสงไปที่บาร์โค้ด จับแสงที่สะท้อน และแทนที่แถบขาวดำด้วยสัญญาณดิจิทัลไบนารี
2. แสงสะท้อนจะชัดเจนในพื้นที่สีขาว และแสงสะท้อนจะอ่อนในพื้นที่สีดำ เซ็นเซอร์จะรับการสะท้อนเพื่อให้ได้รูปคลื่นแอนะล็อก
3. สัญญาณอะนาล็อกจะถูกแปลงเป็นสัญญาณดิจิตอลผ่านตัวแปลง A/D (ไบนาไรเซชัน)
4. การดึงข้อมูลทำได้เมื่อระบบรหัสถูกกำหนดจากสัญญาณดิจิทัลที่ได้รับ (กระบวนการถอดรหัส)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Karakuri kaizen theory

รูปที่ 2.22 ตัวอย่าง Karakuri

(ที่มา https://www.haberkorn.cz/en/data/uploads/Banner/Stavebnicove_systemy/ss_karakuri.jpg)

Karakuri เป็นวิธีการที่โดดเด่นที่ช่วยปรับปรุงประสิทธิภาพการทำงานและลดต้นทุนของบริษัท เทคนิคนี้มาจากญี่ปุ่นในการแปลตามตัวอักษรหมายถึง "กลไกที่มีทักษะ" เป็นคำภาษาญี่ปุ่นที่หมายถึงวิธีการแบบง่ายที่เกี่ยวข้องกับอุปกรณ์จัดการวัสดุเชิงกลที่เรียบง่ายซึ่งอาศัยแรงโน้มถ่วง และความเฉื่อยในการเคลื่อนย้ายผลิตภัณฑ์ผ่านโรงงาน แทนที่จะใช้คอมพิวเตอร์ ไฮดรอลิก แก๊ส หรือแหล่งพลังงานราคาแพงอื่นๆ karakuri ใช้กลไกต้นทุนต่ำ เช่น สปริง คันโยก เกียร์ และลูกตุ้มในการทำงาน Karakuri นำเสนอการแก้ไขปัญหา โดยการทำให้เพรียวลมที่ราคาไม่แพงสำหรับผู้ผลิต ในขณะเดียวกันก็นำเสนอผลิตภัณฑ์ที่เน้นผู้คนเป็นศูนย์กลางซึ่งส่งเสริมการยศาสตร์ (Ergonomics) และการปรับปรุงอย่างต่อเนื่อง แม้ว่าคาราคุริอาจเป็นเรื่องใหม่สำหรับผู้ผลิตหลายราย แต่ประโยชน์ของมันมีมากมาย ทำให้เป็นอุดมการณ์ที่คุ้มค่าที่จะนำมาใช้ และหนึ่งในคาราคุริตัวแรกถูกใช้ในโรงละครหุ่นกระบอก เป้าหมายของ karakuri สมัยใหม่ในการใช้งานในอุตสาหกรรมคือการลดเวลามาตรฐาน ลดความผันผวนของรันไทม์ ลดเวลาการสร้างใหม่ และปรับปรุงการยศาสตร์ ซึ่งหลังจากการใช้งานสำเร็จแล้ว จะนำไปสู่การประหยัดต้นทุน

2.5.1 หลักการทำงานของ Karakuri kaizen

1. เป็นการย้ายวัตถุโดยอัตโนมัติ ไม่ใช้มือมนุษย์

2. ต้นทุนต่ำ

3. ใช้แรงจากอุปกรณ์ ไม่ใช่แรงมนุษย์

4. เป็นอุปกรณ์ที่สร้างด้วยภูมิปัญญา และความคิดสร้างสรรค์ของตนในโรงงาน

5. เพื่อความปลอดภัยจำเป็นต้องสร้างอุปกรณ์ให้สามารถหยุดการทำงานได้ด้วยตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ประโยชน์ของ Karakuri kaizen

1. ต้นทุนต่ำ : เวลาในการพัฒนาสั้นลง วัสดุมีค่าใช้จ่ายน้อยลง และต้นทุนพลังงานจะน้อยกว่ามาก (ถ้ามี)

2. ดูแลรักษาง่ายกว่า : ด้วยชิ้นส่วนที่มีค่าใช้จ่ายน้อยกว่า และความรู้เพิ่มเติมเกี่ยวกับวิธีการทำงาน การทำให้อุปกรณ์เหล่านี้ทำงานได้ง่ายขึ้น

3. ปรับปรุงได้ง่ายขึ้น : เนื่องจากสร้างขึ้นด้วยมือ สามารถเปลี่ยนได้อย่างรวดเร็วซึ่งส่งเสริมการปรับปรุงอย่างต่อเนื่อง

4. การมีส่วนร่วมกับทีมมากขึ้น – ผู้คนจำนวนมากขึ้นสามารถช่วยสร้างอุปกรณ์ karakuri แทนที่จะเอาต์ซอร์ซไปยังบริษัทอื่น การมีส่วนร่วมของพนักงานมากขึ้น

5. ปรับปรุงความปลอดภัยและการยศาสตร์ - ลดโอกาสการบาดเจ็บของคนงานและการสูญเสียวันทำงาน

6. ปรับปรุงคุณภาพ - อุปกรณ์ลดข้อผิดพลาดและความผิดพลาดของมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 โปรแกรม Filemaker Pro

รูปที่ 2.23 Filemaker Pro

(ที่มา https://shop.thaiware.com/3129-Claris-FileMaker-Pro-2023.html)

FileMaker Pro เป็นโปรแกรมจัดการฐานข้อมูล มีลักษณะการใช้งานคล้ายกับโปรแกรมจัดการฐานข้อมูลที่มีใช้กันอยู่ทั่วไป เช่น MS Access, FoxPro เป็นต้น มีสามารถใช้งานได้ทั้งในระบบปฏิบัติการ Windows และ Macintosh มีคุณสมบัติในการสร้างฟอร์ม การสร้างรายงานโดยดึงผลมาจากฐานข้อมูล และการค้นหาข้อมูลตามเงื่อนไขหลายๆ อย่างพร้อมๆ กัน สามารถใช้งานในเครือข่ายภายในองค์กร หรือเครือข่ายอินเตอร์เน็ตได้โดยวิธีง่ายๆ FileMaker Pro เป็นโปรแกรมที่เรียกว่า cross-platform application ไฟล์ที่สร้างขึ้นจากเครื่องที่ใช้งานระบบปฏิบัติการ Windows สามารถนำไปใช้งานบนเครื่องที่ใช้งานระบบปฏิบัติการ Mac หรือไฟล์ที่สร้างขึ้นจากเครื่องที่ใช้งานระบบปฏิบัติการ Mac ก็สามารถนำไปใช้งานบนเครื่องที่ใช้งานระบบปฏิบัติการ Windows ได้ ช่วยให้สร้างแอปการดำเนินงานที่มีประสิทธิภาพสูงสุด และเชื่อถือได้ซึ่งปรับขนาดตามธุรกิจได้ สร้างแอปเพื่อจัดการรายชื่อติดต่อ ติดตามสินค้าคงคลัง จัดระเบียบโครงการ และอื่นๆ นอกจากนี้ FileMaker Pro ยังมีชุดเครื่องมือการพัฒนา และวินิจฉัยขั้นสูงเพื่อช่วยให้คุณออกแบบ และพัฒนาแอปได้อย่างง่ายดาย

การใช้งาน FileMaker Pro ไม่เหมือนกับการใช้งานโปรแกรมพิมพ์เอกสาร (word processing) หรือโปรแกรมข้อมูลตาราง (spreadsheet programs) FileMaker Pro จะทำการบันทึกงานทั้งหมดแบบอัตโนมัติในขณะที่กำลังทำงาน จึงเป็นสิ่งที่ควรระวังในการทำงานในขณะที่มีการเปลี่ยนแปลงไฟล์ข้อมูล ตัวอย่างเช่น ในการลบข้อมูล ข้อมูลที่ถูกลบจะหายไปจากฐานข้อมูลทันที โดยไม่สามารถเรียกกลับคืนมาใหม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 คุณสมบัติของ FileMaker Pro

1. ใช้งานได้กับระบบการจัดการในระบบคอมพิวเตอร์ได้หลายระบบในไฟล์เดียว ได้แก่ Windows 11 Enterprise และ Pro Editions, Windows 10 Enterprise และ Pro Editions; (22H2, 64 บิตเท่านั้น), แนะนำให้ใช้ RAM 2GB, 4GB ขึ้นไป และ แนะนำ macOS Ventura 13.0 , macOS Monterey 12.0 , 4GB RAM, 8GB ขึ้นไป

2. ใช้งานง่ายมีตัวอย่างระบบฐานข้อมูลเริ่มต้นไว้ให้ใช้งานมากมายในหลายๆด้าน

3. จัดการข้อมูลกับซอฟต์แวร์ที่คุ้นเคย แล้วนำเข้าในระบบ FileMaker Pro 2023 ได้โดยง่าย ซอฟต์แวร์ที่คุ้นเคยได้แก่ Microsoft Excel, Microsoft Words, Notepad, WordPad เป็นต้น

4. มีความสามารถในการขอดูข้อมูลเพิ่มเติม ข้อมูลแก้ไข ข้อมูลลบ ข้อมูลค้นหา ข้อมูลตามหัวข้อเฉพาะ

5. แสดงผลรายงานได้ง่ายได้หลายรูปแบบทั้งพิมพ์เป็นเอกสารกระดาษ เอกสารอิเล็กทรอนิกส์ หรือเอกสารที่ใช้ในระบบเครือข่ายอินเทอร์เน็ต

6. ใช้ข้อมูลร่วมกันได้ในระบบเครือข่ายได้ทันที

รูปที่ 2.24 Icon Program Filemaker Pro บน PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินโครงการ

สำหรับการออกแบบระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse) จำเป็นต้องคำนึงถึงขนาดของสินค้า การทำงานของโรลเลอร์ การทำงานของดีซีมอเตอร์ ส่วนของการรับข้อมูลผ่านสแกนเนอร์ การแสดงผลหน้าจอ รวมไปถึงการเขียนคำสั่งโปรแกรมสำหรับควบคุมการทำงานให้เป็นไปตามที่ต้องการ

3.1 อุปกรณ์ เครื่องมือ หรือโปรแกรมที่ใช้ในการดำเนินโครงงาน

3.1.1 Arduino Mega 2560

3.1.2 Servo Motor MG995

3.1.3 LCD 16*2

3.1.4 MH-ET Scanner Module V3.0

3.1.5 ท่อ PVC

3.1.6 ฟิวเจอร์บอร์ด

3.1.7 สายไฟ

3.1.8 Adapter 3-12 โวลต์ 5 แอมป์

3.1.9 แผ่นไม้อัด

3.1.10 เครื่องมือบัดกรี

3.1.11 โปรแกรม Arduino IDE

3.1.12 สวิตช์ 3 ทาง

3.1.13 โปรแกรม Filemaker

3.2 ขั้นตอนการดำเนินโครงงาน

3.2.1 ศึกษาเกี่ยวกับ ระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse)

3.2.2 ศึกษาการทำงานแบบ Karakuri

3.2.3 สรุปหลักการทำงานระบบการทำงานของ ระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ

3.2.4 ออกแบบโครงสร้าง และกำหนดอุปกรณ์ที่ต้องใช้สำหรับการทำระบบ

3.2.5 จัดซื้ออุปกรณ์

3.2.6 ทำโครงสร้างส่วนของโรลเลอร์ และทำการทดสอบ

3.2.7 ประกอบโครงสร้างส่วนโรลเลอร์ เข้ากับโครงสร้างหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8 ออกแบบ Schematic สำหรับวงจรภายในระบบ

3.2.9 ประกอบอุปกรณ์ต่าง เพื่อควบคุม และทดสอบระบบการทำงาน

3.2.10 เขียนโปรแกรมเพื่อระบุเงื่อนไขการทำงาน และการสั่งการทำงานของระบบ โดยใช้โปรแกรม Arduino IDE

3.2.11 เขียนโปรแกรม Filemaker สำหรับการลงทะเบียน และสร้างบาร์โค้ด

3.2.12 ทดลองประสิทธิภาพการทำงานของควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse)

3.2.13 บันทึก และสรุปผลการทดลองประสิทธิภาพการทำงาน

3.3 การพิจารณาการใช้งานเซอร์โวมอเตอร์

เนื่องจากโครงสร้างส่วนของแผ่นกั้นทำมาจากอะคลิลิคซึ่งต้องรับน้ำหนักของกล่องอุปกรณ์ที่มีน้ำหนักมาก จึงจำเป็นต้องเลือกใช้เซอร์โวมอเตอร์รุ่น MG996R ที่มีกำลังสำหรับการหมุนที่มากและทำมาจากวัสดุที่แข็งแรง โดยเป็นเซอร์โวมิเตอร์ที่ทำงานโดยการจ่ายไฟขนาด 6 โวลต์ เพื่อให้ได้กำลังหมุนสูงสุด และมีวัสดุแกนหมุนเป็นโลหะ

ลักษณะเฉพาะของเซอร์โวมอเตอร์รุ่น MG995

- Dimension : 40mm x 19mm x 43mm
- Weight : 55g
- Operating Speed : 0.17sec / 60 degrees (4.8V no load)
- Operating Speed : 0.13sec / 60 degrees (6.0V no load)
- Stall Torque : 9 kg-cm (180.5 oz-in) at 4.8V
- Stall Torque : 12 kg-cm (208.3 oz-in) at 6V
- Operation Voltage : 4.8 - 7.2Volts
- Gear Type: All Metal Gears
- Original box: NO
- Color: Black
- Connector Wire: Heavy Duty, 11.81" (300mm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

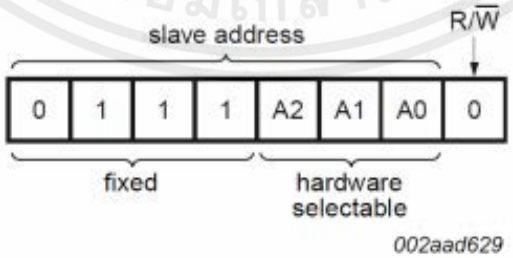
3.4 การพิจารณาการใช้งานจอ LCD 16*2 I2C

LCD เป็นจอแสดงผลแบบตัวอักษรที่มีขนาด 16 ตัวอักษร 2 บรรทัด มีพื้นสีน้ำเงิน ตัวอักษรสีขาว โดยมีการเชื่อมต่อแบบ I2C-Bus ทำให้สามารถต่อร่วมกันได้ 8 ขาในบัสตัวเดียวกัน แต่จำเป็นที่จะต้องเลือก Address ที่แตกต่างกัน IC ที่ใช้คือเบอร์ PCF8574A ซึ่งช่วยในการลดจำนวนขาที่ใช้ในการเชื่อมต่อกับไมโครคอลโทรลเลอร์ และใช้ไฟเลี้ยง 5 VDC

การเชื่อมต่อเลือก Address (A0-A2) ของจอ LCD โดยการเชื่อมต่อ ถ้าไม่เชื่อมต่อจะได้ค่าลอจิก 1 และถ้าเชื่อมต่อจะได้ค่าลอจิก 0 ซึ่งปกติจะไม่ได้บัดกรีไว้ทำให้ Address เริ่มต้นคือ 0x3F (A2=1 , A1=1 , A0=1) แต่ถ้าผู้ใช้ ต้องการเปลี่ยน Address ก็สามารถเปลี่ยนได้ 8 ค่า คือ 0x38-0x3F ดังตาราง

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

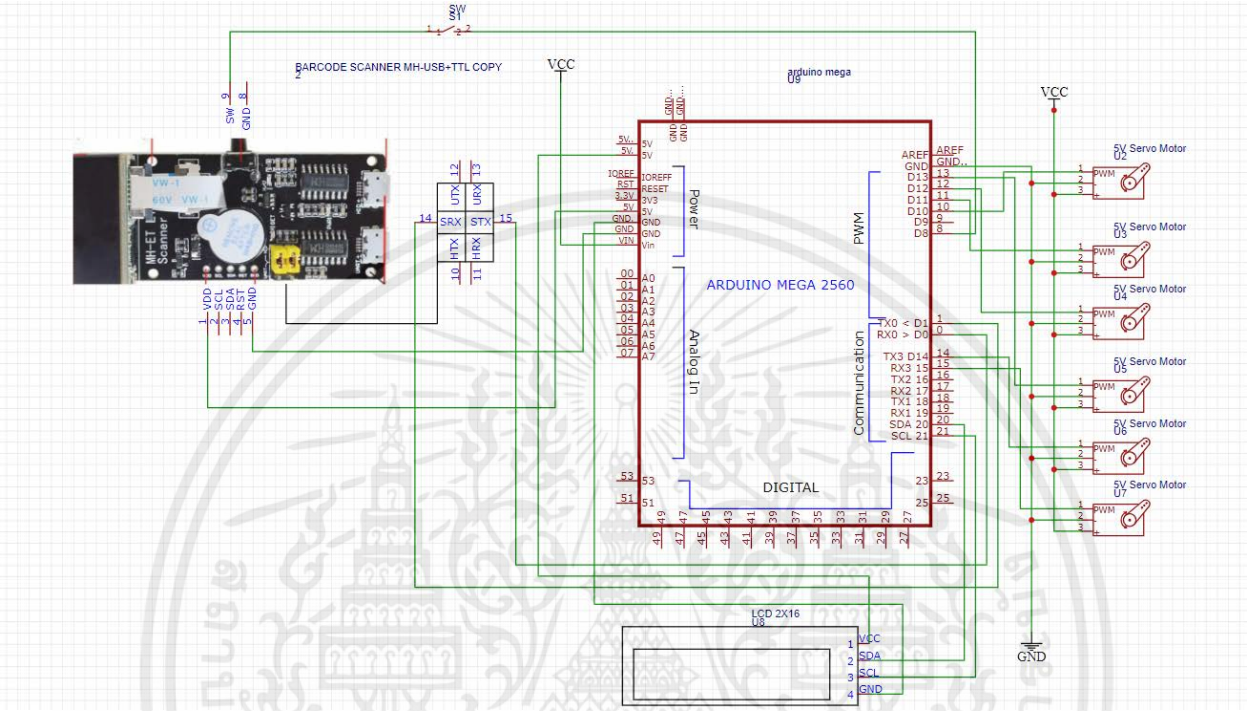


PCF8574A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 พิจารณาวงจรที่ใช้

ประกอบไปด้วย Servo motor เข้ากับ Arduino Mega 2560 ที่ I/O Port ตั้งแต่ขา D10-D15 Scanner ที่ขา RX0 และ TX0 และประกอบจอ LCD สำหรับใช้ในการแสดงผล

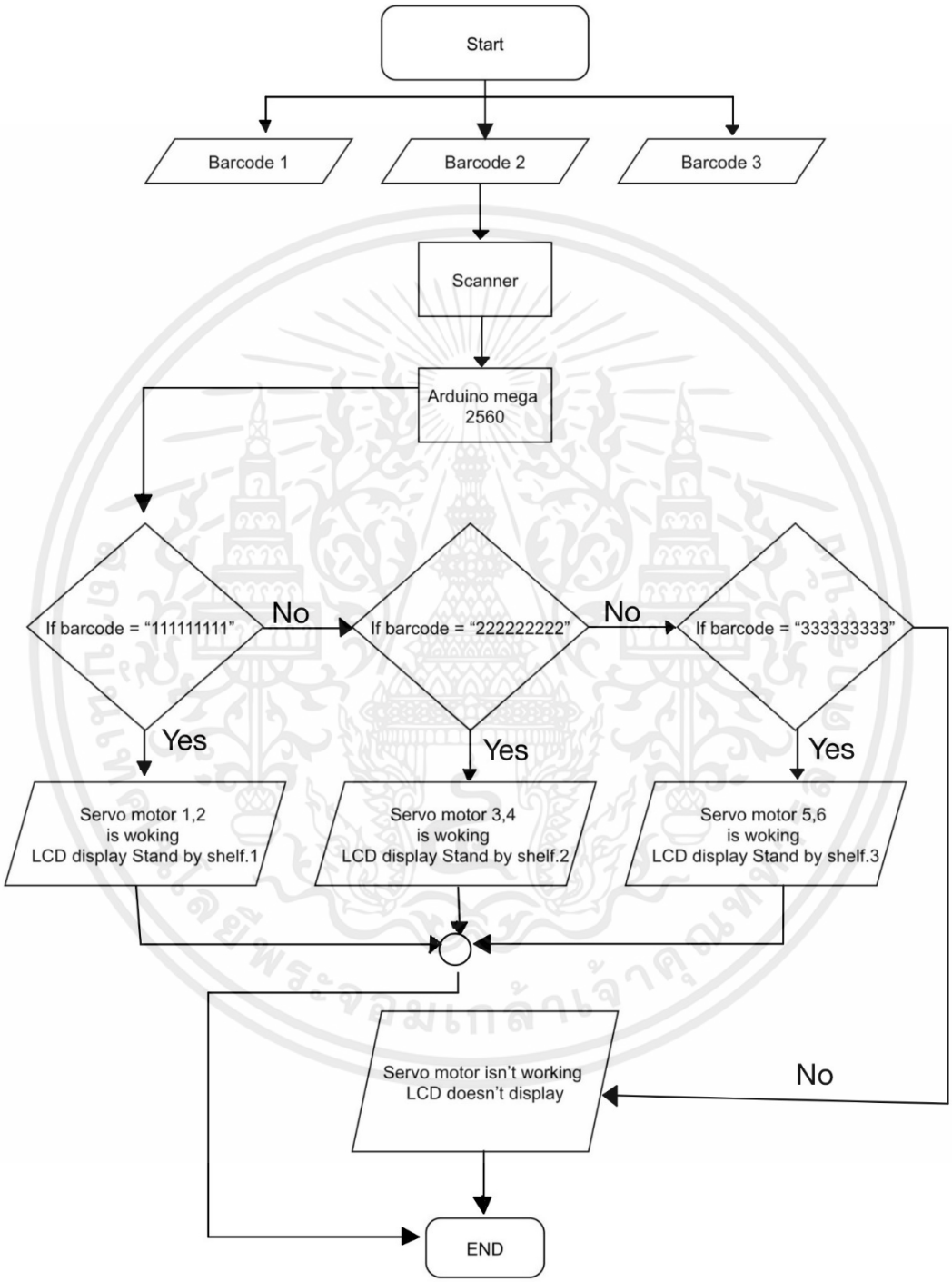


รูปที่ 3.1 Schematic ของวงจร Automated Small Warehouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 พิจารณาการเขียนโปรแกรมภาษาซีใน Arduino IDE

3.6.1 Flowchart การทำงานของโปรแกรม



รูปที่ 3.2 Flowchart การทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การออกแบบโครงสร้างของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse)

การออกแบบโครงสร้างของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ ควรคำนึงถึงวัสดุที่เลือกใช้ โดยควรเลือกวัสดุที่มีความแข็งแรง สามารถรับน้ำหนักของสินค้า เช่น Servo motor , กล่องบรรจุสินค้า เป็นต้น การออกแบบขนาดของช่องให้มีขนาดที่แตกต่างกัน เพื่อความหลากหลายของสินค้าที่สามารถบบรรจุเข้าไปได้ และอุปกรณ์เสริมส่วนอื่นๆได้ และคำนึงถึงความสะดวกสบายของผู้ใช้งานที่ไม่จำเป็นต้องใช้ทรัพยากรมนุษย์

รูปที่ 3.3 การออกแบบโครงของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ต้องคำนึงถึงการติดตั้งมอเตอร์ โครงงานระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ ได้เลือกใช้ Servo motor MG995 180 องศา จำนวน 2 ตัวต่อ 1 ช่องปล่อยสินค้า สำหรับทำหน้าที่เป็นตัวกั้นเปิดและปิดช่องสินค้า โดยมอเตอร์จะถูกติดเข้ากับแผ่นไม้ที่ได้ทำการวัดขนาด และตัดให้เหมาะสมกับการใช้งานแล้ว ลักษณะการติดตั้งมอเตอร์จำเป็นต้องคำนึงถึงขนาดของกล่องสินค้าแต่ละชิ้น และระยะห่างระหว่างมอเตอร์ 2 ตัว มอเตอร์ตัวที่ 1 จะถูกติดตั้งไว้ที่ด้านหน้าสุดของชุดโรลเลอร์ และตัวที่ 2 จะอยู่ห่างจากตัวที่ 1 โดยประมาณ 20 เซนติเมตร หรือเท่ากับครึ่งหนึ่งของความยาวโรลเลอร์ ซึ่งจะถูกติดตั้งอยู่ที่ตำแหน่งกึ่งกลางของชุดโรลเลอเลอร์แต่ละช่อง เพื่อป้องกันความผิดพลาดจากการที่สินค้าไหลมาชนกัน หรือเกินจำนวนที่กำหนดไว้ และสามารถควบคุมการทำงานได้อย่างมีประสิทธิภาพ

รูปที่ 3.4 ตำแหน่งการติดตั้งของ Servo motor

รูปที่ 3.5 โครงสร้างจริงของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

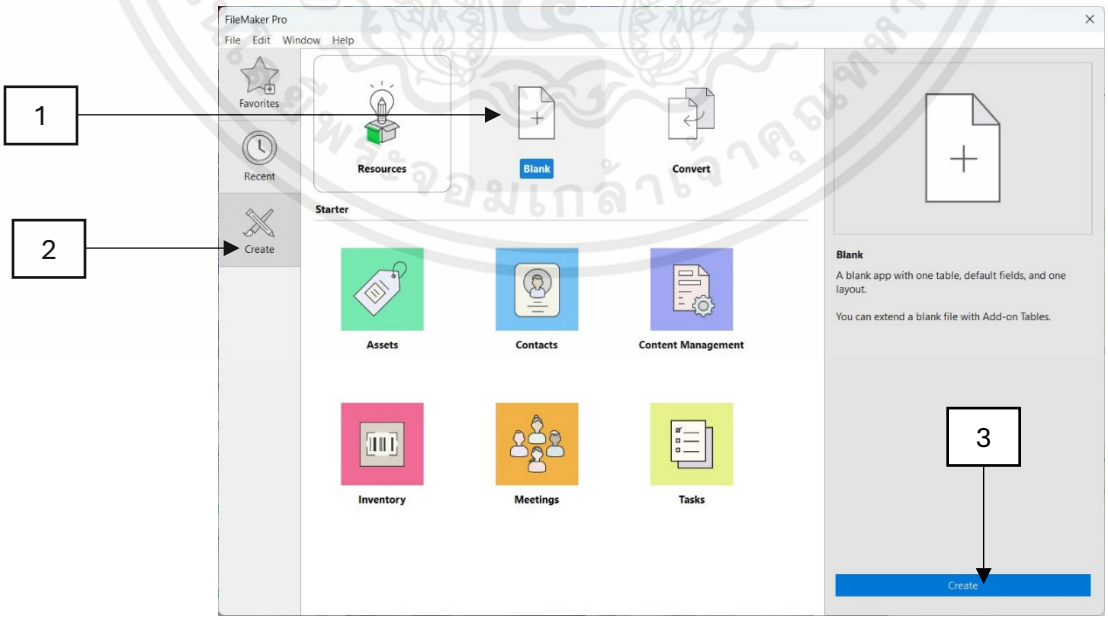
3.8 พิจารณาการสร้างแอพพลิเคชั่นจากโปรแกรม Filemaker

3.8.1 การสร้างหน้า Layout สำหรับผู้ใช้งาน

การออกแบบแอพพลิเคชั่นสำหรับโครงงานระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ โดยการสร้างจากโปรแกรม Filemaker ต้องคำนึงถึงความเหมาะสมสำหรับผู้ใช้งาน และข้อมูลพื้นฐานที่ต้องการจัดเก็บเป็น Database สำหรับการทำรายการ Stock สินค้าหลังบ้าน เพื่อที่จะใช้สำหรับการสร้าง Layout ที่เหมาะสมแก่โครงงาน และจำเป็นต้องทำให้ข้อมูลที่จัดเก็บสามารถส่งออกมาในรูปแบบของไฟล์ .pdf และ Excel ได้ ข้อมูลพื้นฐานสำหรับใช้ภายในแอพพลิเคชั่นประกอบด้วย ชื่อของผู้เบิกสินค้า แผนก วัน เวลาในการเบิก และสินค้าที่เบิก นอกจากนี้ยังต้องออกแบบระบบสำหรับการคำนวณสินค้าเข้า-ออก รวมถึงสินค้าที่คงเหลือภายในคลัง เพื่อให้ระบบทำงานได้แบบ Real-time ตามวัตถุประสงค์ของโครงงาน โดยขั้นตอนเบื้องต้นสำหรับการสร้าง Layout มีดังนี้

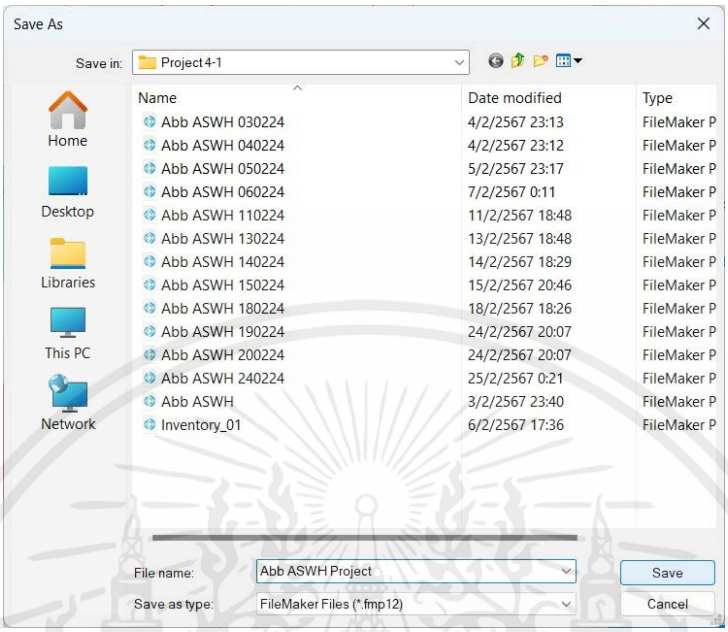
1. ดับเบิลคลิกที่ Icon FileMaker Pro ที่หน้า Desktop

2. คลิก Create (1) > Blank (2) > Create (3)

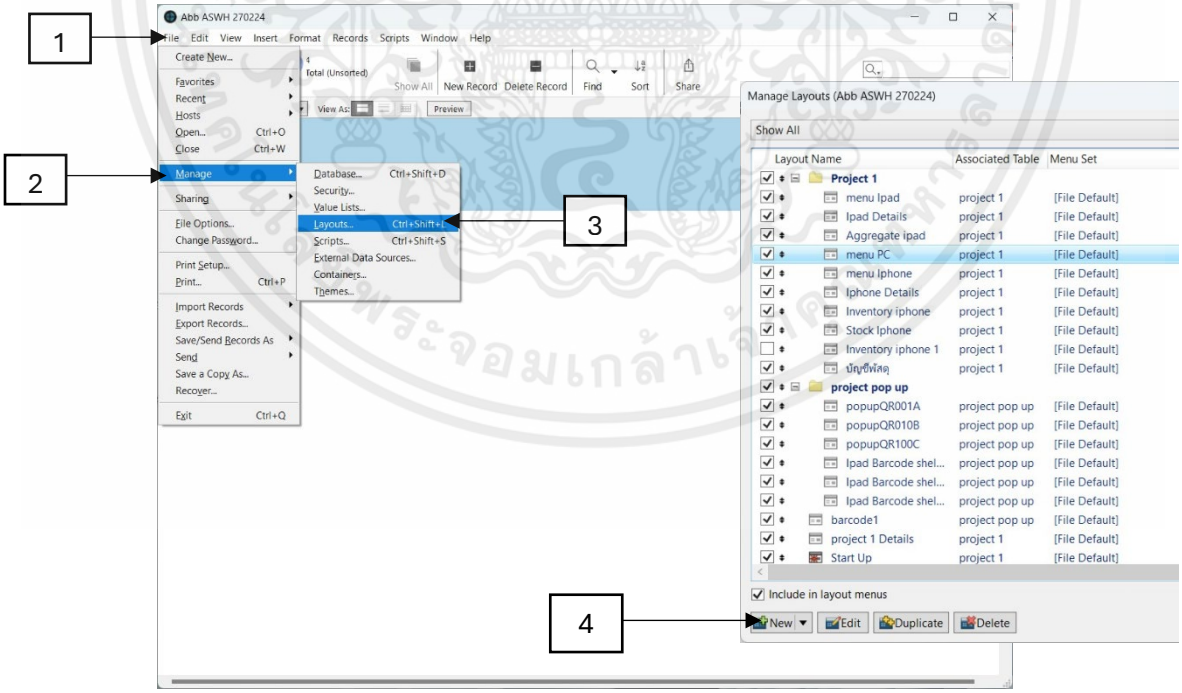


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกที่อยู่ที่จัดเก็บพร้องตั้งชื่อไฟล์แล้วกด Save

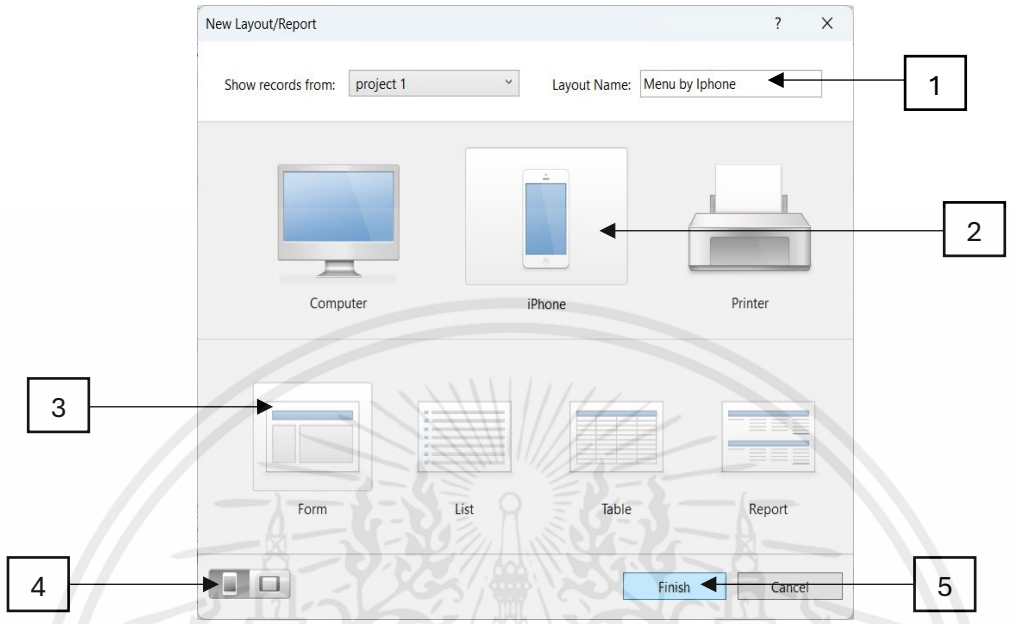


4. ทำการสร้าง Layout โดยคลิก File (1) > Manage (2) > Layouts (3) > New (4)

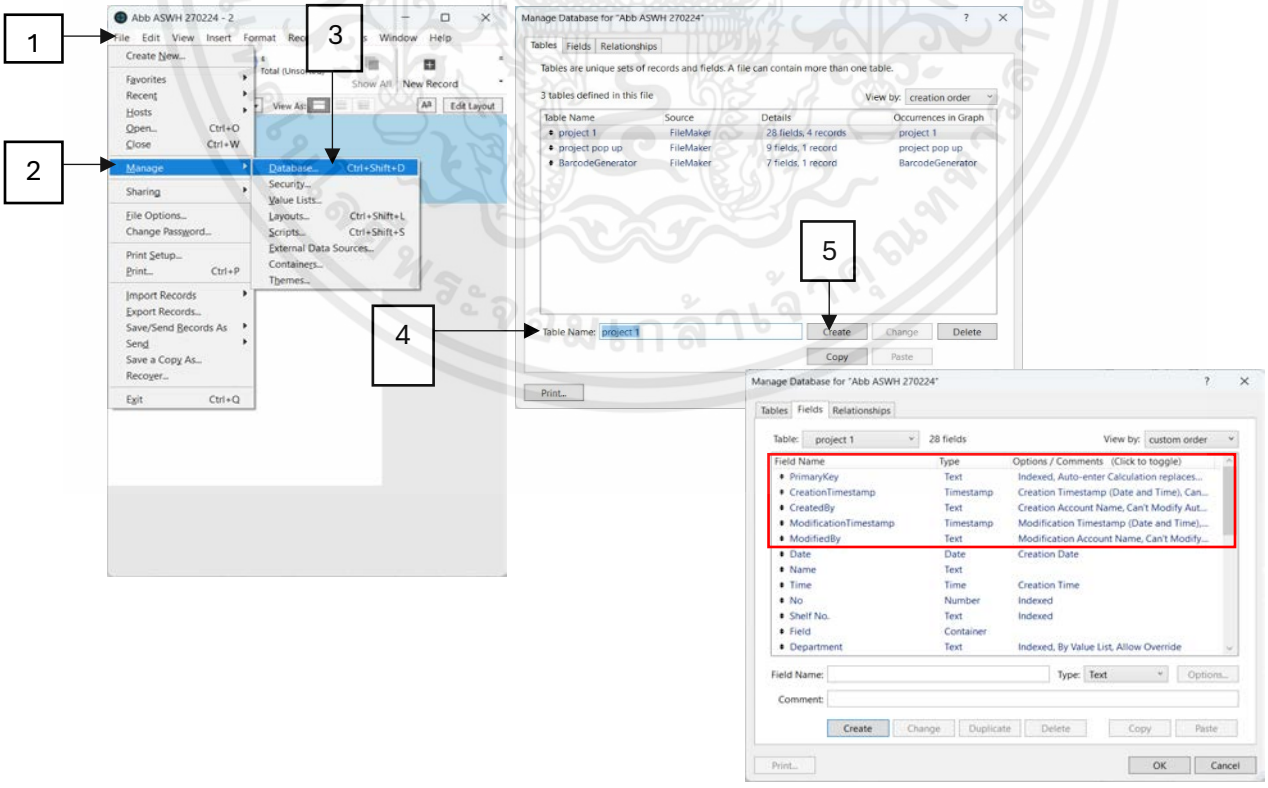


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ตั้งชื่อ Layout (1) > คลิกที่ Iphone (2) > Form(3) > เลือกรูปแบบแนวตั้ง (4) > Finish (5)

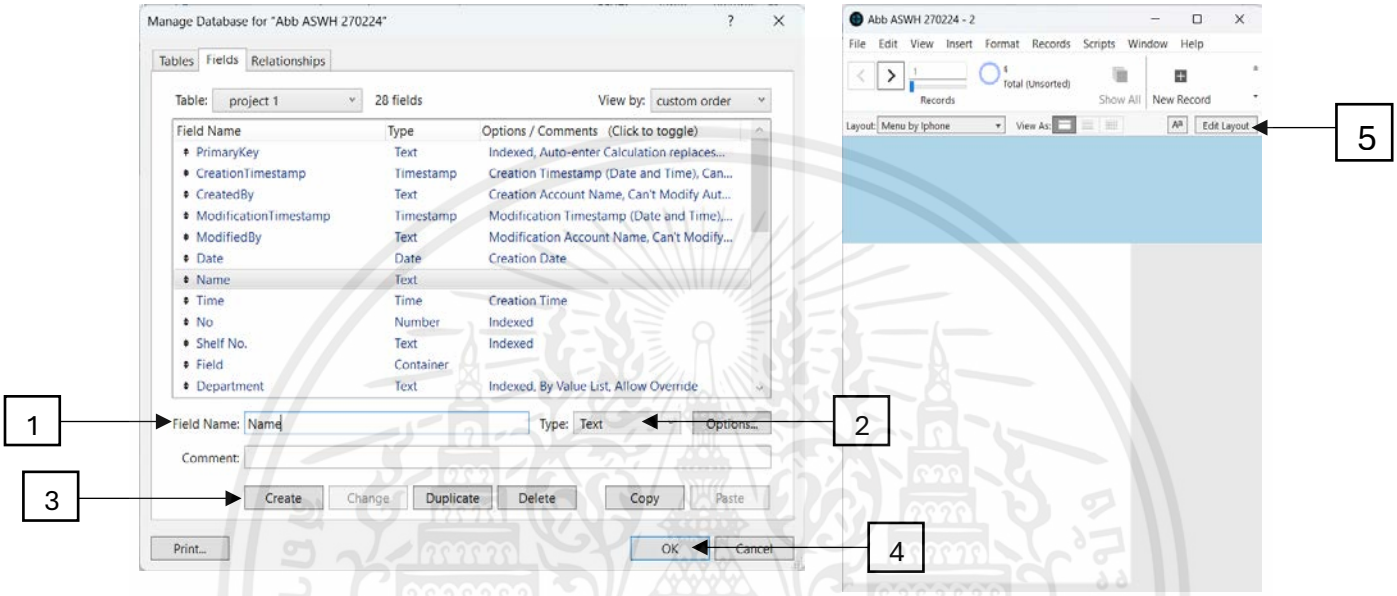


6. จากนั้นสร้าง Tables โดยคลิก File(1) > Manage(2) > Database(3) > ตั้งชื่อ(4) > Create(5) จากนั้นจะมี Fields เริ่มต้นมาให้ 5 Fields อยู่ใน Tables ที่เราสร้าง ใน Project นี้เราจะสร้าง 3 Tables

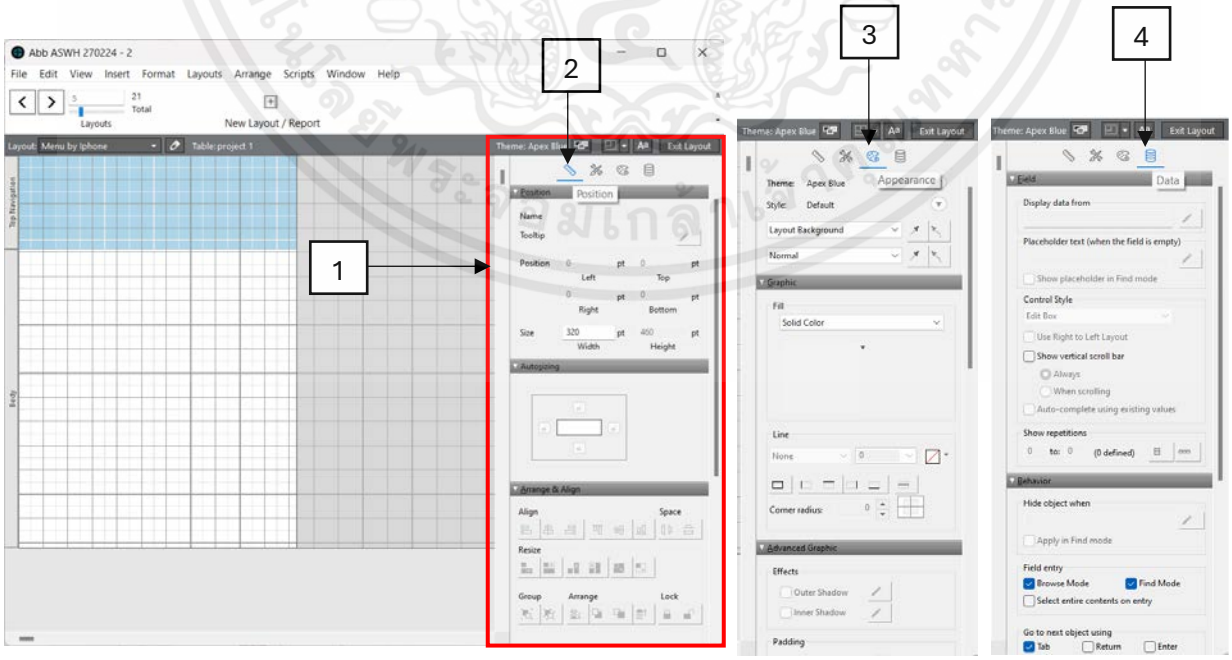


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. สร้าง Fields ที่ต้องการให้ตรงตาม Project ที่ทำ ตั้งชื่อ(1) > เลือกType(2) ตามความเหมาะสม เช่น Name – Type Text , Date – Type Date เป็นต้น > Create(3) > OK(4) ใน Project นี้เราใช้ Fields ทั้งหมด 44 Fields เมื่อได้ Fields ที่จำเป็นแล้วนั้นก็ทำการออกแบบหน้า Layout ได้เลยโดย คลิกEdit Layout(5)

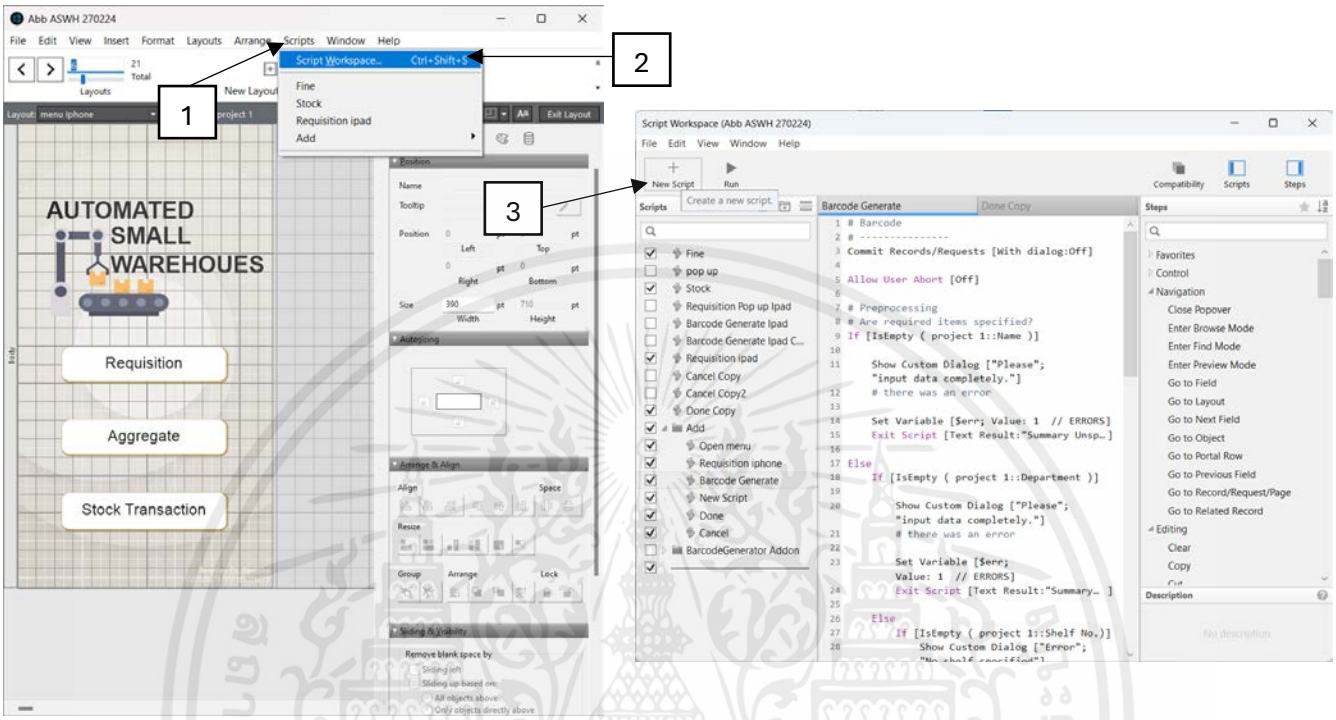


8. จัดหน้า Layout ให้เท่ากับขนาดหน้าจอของ IPhone ด้วยแถบ Status Toolbar(1) ที่ Position(2) ตกแต่งสีให้กับหน้า Layout ที่ Appearance(3) ออกแบบ Fields ได้ที่ Data(4)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เขียน Script คำสั่งการทำงานเป็นขั้นตอนได้ที่ Scripts(1) > Script Workspace(2) > New Script(3) เพื่อสร้างเงื่อนไขตามการใช้งาน

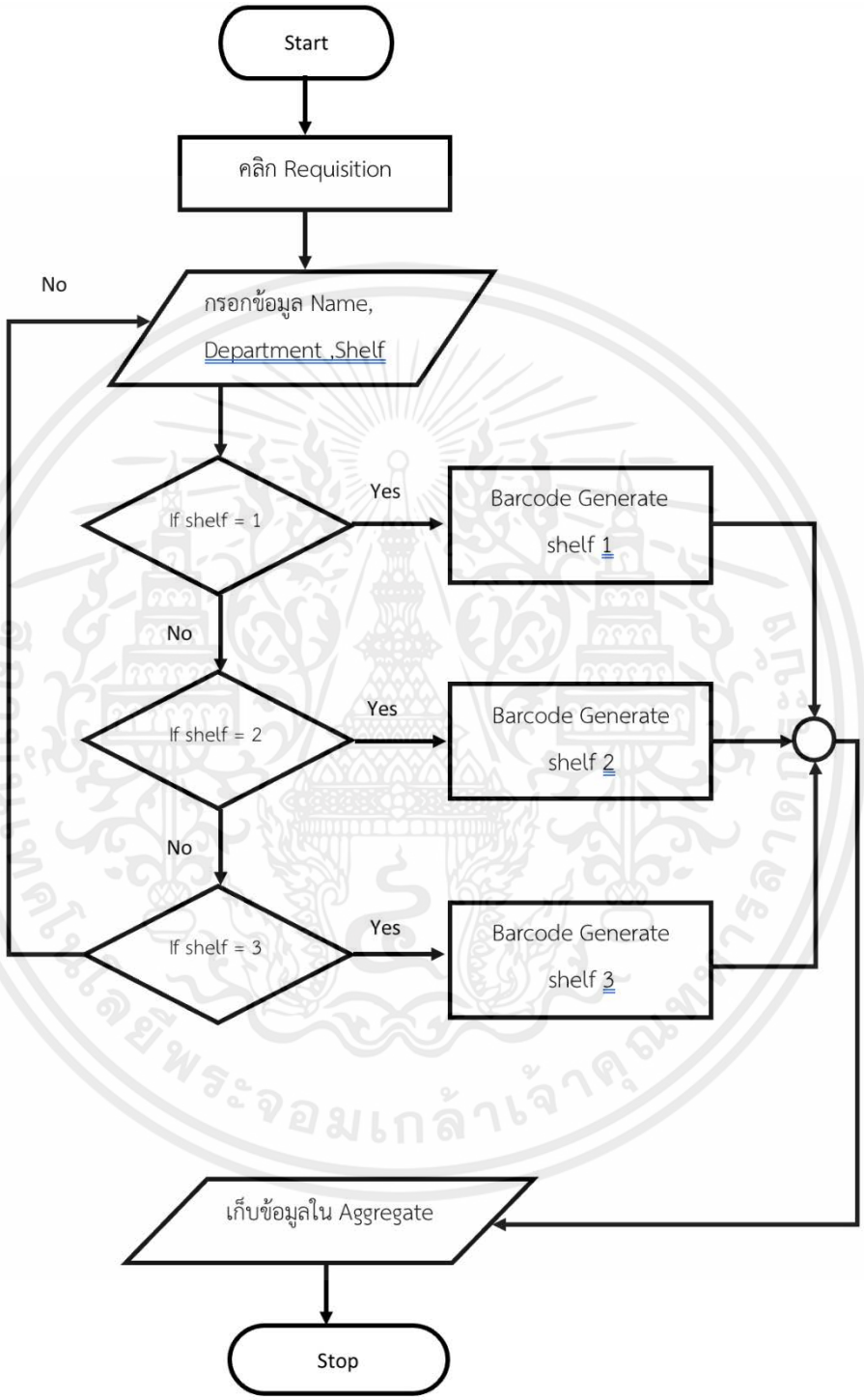


การแสดงผลในรูปแบบฟอร์มที่สร้างไว้ เรียกว่า Layout สำหรับโครงงานนี้มีจำนวน Layout จำนวน 14 Layout โดยแบ่งการแสดงผลอยู่ใน 2 Platform ได้แก่ Ipad กับ Iphone ดังนี้

1. menu Ipad
2. Ipad Details
3. Aggregate Ipad
4. Ipad Barcode shelf 1
5. Ipad Barcode shelf 2
6. Ipad Barcode shelf 3
7. menu Iphone
8. Iphone Details
9. Inventory iPhone
10. Stock Iphone
11. popupQR001A
12. popupQR010B
13. popupQR100C
14. บัญชีพัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.2 Flowchart สำหรับ Application



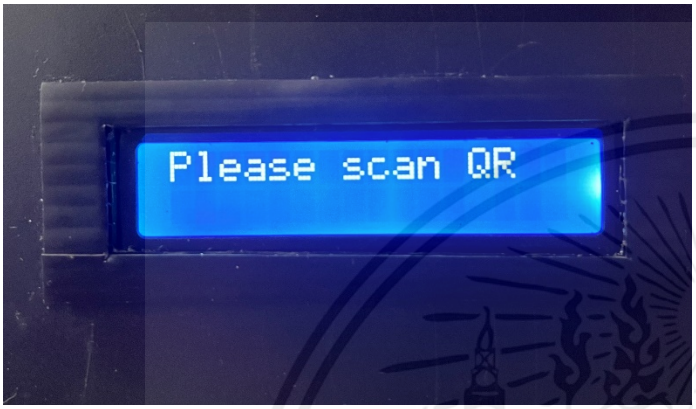
รูปที่ 3.6 Flowchart Filemaker Programs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 การทดลองการทำงานของการแสดงผลบนจอ LCD เมื่อ Scanner อ่านบาร์โค้ด



รูปที่ 4.1 การแสดงผลของจอ LCD แสดงสถานะเริ่มต้น

รูปที่ 4.2 การแสดงผลของจอ LCD เมื่อบาร์โค้ดของช่องที่ 1 ถูกแสกนเข้าระบบ

รูปที่ 4.3 การแสดงผลของจอ LCD เมื่อบาร์โค้ดของช่องที่ 2 ถูกแสกนเข้าระบบ

รูปที่ 4.4 การแสดงผลของจอ LCD เมื่อบาร์โค้ดของช่องที่ 3 ถูกแสกนเข้าระบบ

รูปที่ 4.5 การแสดงผลของจอ LCD เมื่อบาร์โค้ดไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

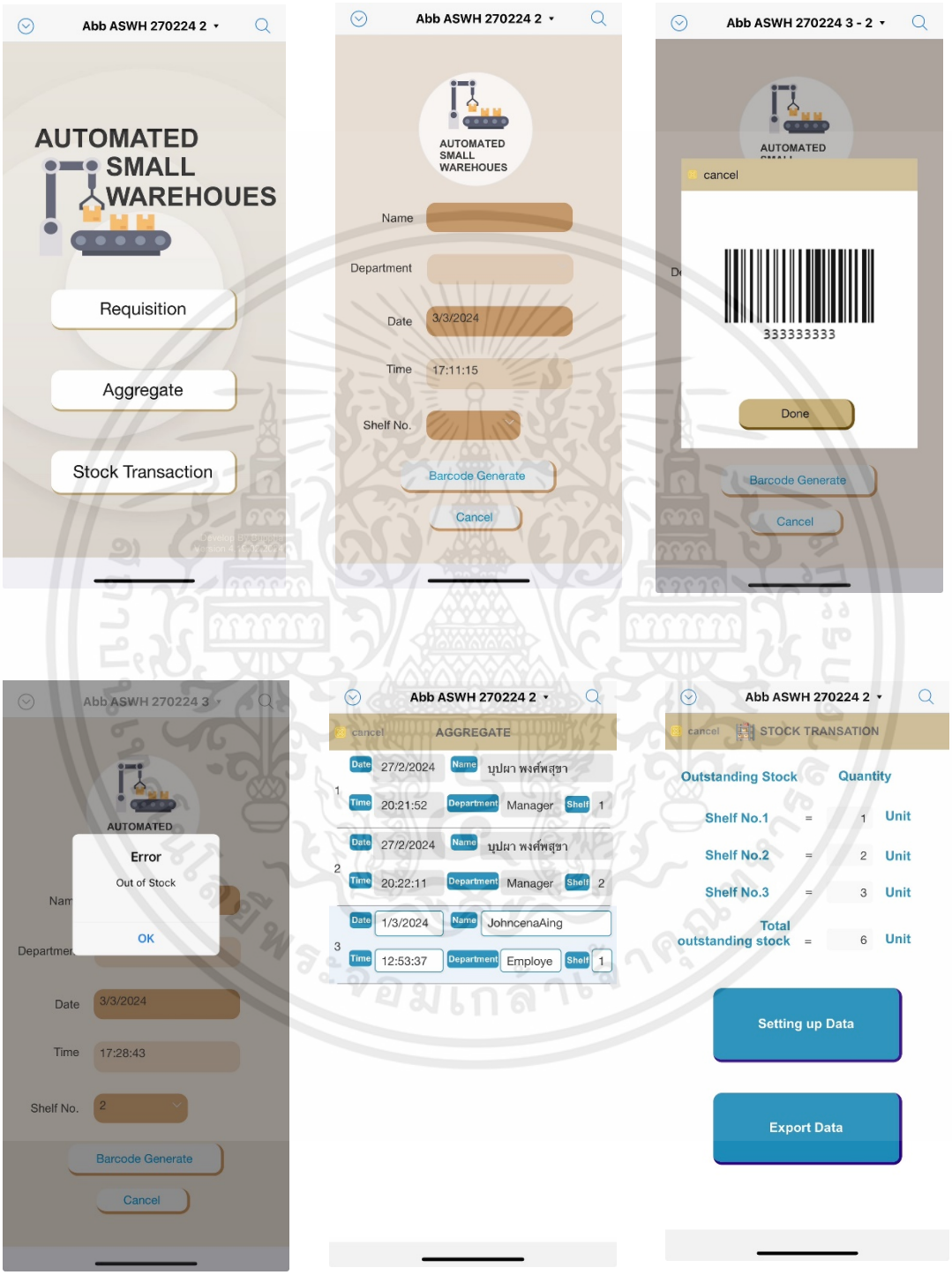
4.2 การทดลองทดสอบการทำงานของ Servo motor เมื่อ Scanner อ่านบาร์โค้ด

Barcode	แสดงผล LCD	มอเตอร์ 1	มอเตอร์ 2
111111111	Stand by Shelf.1	ทำงาน	ทำงาน
222222222	Stand by Shelf.2	ทำงาน	ทำงาน
333333333	Stand by Shelf.3	ทำงาน	ทำงาน
165873402	Incorrect	ไม่ทำงาน	ไม่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการทำงานของโปรแกรม Filemaker

4.3.1 การทำงานบน Smartphone (Iphone)



รูปที่ 4.6 หน้าโปรแกรม Filemaker สำหรับผู้ใช้งานบนสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การทำงานบน Tablet (Ipad)

รูปที่ 4.7 หน้าโปรแกรม Filemaker สำหรับผู้ใช้งานบนแท็บเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การส่งออกข้อมูลออกในรูปแบบ .PDF File

บัญชีพัสดุ อุปกรณ์

ประจำปี.....

ครั้งที่	วัน / เดือน / ปี	เวลา	ชื่อผู้รับของ	ตำแหน่ง	ประเภท	จำนวน
1	1/3/2024	11:38:25	Nun	Manager	3	1
2	1/3/2024	11:44:34	Pichayatida	Manager	3	1
3	1/3/2024	11:51:14	Nithipoom	Employee	3	1
4	1/3/2024	11:53	Cha	Manager	1	1
5	1/3/2024	11:54:30	Ff	Manager	3	1
6	1/3/2024	11:59:14	Punny	Employee	1	1
7	1/3/2024	12:18:58	Natsu	Employee	1	1
8	1/3/2024	12:28:18	Aye	Manager	1	1
9	1/3/2024	12:30:44	Pachi	Employee	3	1
10	1/3/2024	12:37:45	Fearce	Manager	1	1
11	1/3/2024	12:38:38	Cha	QA	3	1
12	1/3/2024	13:17:01	คริส	Manager	1	1
13	1/3/2024	13:30:04	ชา	Manager	1	1
14	1/3/2024	13:42:56	Prysol	QA	1	1
15	1/3/2024	13:42:58	พี	Employee	1	1
16	1/3/2024	13:43:45	Prysol	QA	3	1
17	1/3/2024	14:03:15	Cream	Manager	1	1
18	1/3/2024	14:12:44	A	Manager	3	1

รูปที่ 4.8 หน้าข้อมูลที่ถูกส่งออกมาเป็น .PDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

การทำการทดลองโครงงานระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติมีวัตถุประสงค์เพื่อศึกษาการทำงานของระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติ (Automated Small Warehouse) ศึกษาการทำงานและเขียนโปรแกรม (Coding) ของ MCU ชนิด Arduino ศึกษาระบบ Karakuri ที่ประยุกต์ใช้ในโรงงานอุตสาหกรรม และพัฒนาระบบสำหรับแก้ปัญหาภายในคลังสินค้าของโรงงานอุตสาหกรรมขนาดย่อม จากการทำโครงงานพบว่า เมื่อทำการออกแบบโครงสร้างชุดของโรลเลอร์โดยใช้ท่อ PVC และทำการติดตั้งในลักษณะเอียงโดยอาศัยหลักการ Karakuri ที่เป็นหลักของการใช้พื้นฐานฟิสิกส์ อย่างพื้นลื่น และแรงโน้มถ่วงช่วยในการเคลื่อนที่ของวัตถุ แล้วทำการสร้างระบบอัตโนมัติโดยใช้การเขียนโปรแกรมผ่าน Arduino IDE รับค่าจากการอ่านบาร์โค้ดผ่าน Scanner จากนั้นนำไปประมวลผลที่ไมโครคอลโทรลเลอร์ (Arduino Mega 2560) ให้ทำการสั่งการขับมอเตอร์ และแสดงผลผ่านหน้าจอ LCD ซึ่งรหัสผ่านที่กำหนดขึ้นจะใช้สำหรับการควบคุมมอเตอร์ที่ติดอยู่ด้านหน้าของชุดโรลเลอร์ เพื่อทำการเปิดปิดช่องสินค้าของระบบ หากบาร์โค้ดที่ได้รับเข้ามาในระบบถูกต้อง มอเตอร์จะทำการหมุนเพื่อเปิดช่องให้สินค้าไหลลงมา ส่วน LCD จะแสดงผลตามช่องที่สินค้าที่กำหนดไว้ อาทิเช่น หากอ่านบาร์โค้ดได้ว่าเป็นช่องที่ 1 จะแสดงคำว่า “Stand by Shelf.1” จากนั้นส่วนของมอเตอร์ตัวที่ 2 จะทำงานหลังจากมอเตอร์ตัวแรกหมุนกลับ (ปิดช่องสินค้า) แต่หากบาร์โค้ดที่ได้รับเข้าไปผิดพลาด หรือเป็นบาร์โค้ดอื่นๆที่ไม่ได้มาจากการสร้างโดย Filemaker มอเตอร์ทั้งสองจะไม่มีการทำงานเกิดขึ้น ส่วนการแสดงผลบนจอ LCD จะแสดงคำว่า “Incorrect” ในการทดลองสำหรับทดสอบอุปกรณ์ที่ และแอพพลิเคชั่นที่ใช้ในการจัดทำโครงงานถูกแบ่งเป็น 3 ตอนดังนี้

ในการทดลองตอนที่ 1 การทดลองการทำงานของการแสดงผลบนจอ LCD โดยการเขียนโปรแกรมชุดคำสั่งการทำงานผ่านโปรแกรม Arduino IDE ซึ่งจะแสดงผลเมื่อได้รับการแสกนบาร์โค้ดผ่าน Scanner การแสดงผลในส่วนแรก จะแสดงคำว่า Please Scan QR เพื่อเป็นการบอกสถานะเริ่มต้นการทำงาน จากนั้นเมื่อทำการอ่านบาร์โค้ด แล้วส่งข้อมูลไปประมวลผลผ่าน Arduino Mega 2560 หากบาร์โค้ดถูกต้องจะแสดงผลตามค่าที่กำหนดไว้ ในกรณีที่เป็นช่องสินค้าที่ 1 จะแสดงว่า “Stand by Shelf.1” ช่องสินค้าเป็นช่องที่ 2 จะแสดงว่า “Stand by Shelf.2” และในกรณีที่เป็นช่องที่ 3 จะแสดงว่า “Stand by Shelf.3” แต่หากบาร์โค้ดไม่ตรงกับที่กำหนดจะแสดงผลว่า “Incorrect”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองตอนที่ 2 การทดลองทดสอบการทำงานของ Servo motor โดยการเขียนโปรแกรมชุดคำสั่งการทำงานผ่านโปรแกรม Arduino IDE เมื่อค่าที่ได้รับจาก Scanner ถูกต้องเป็นไปตามเงื่อนไข มอเตอร์ตัวแรกจะทำงาน โดยจะหมุนจากตำแหน่งแรกที่กำหนดไว้ที่ 10 องศาไปที่ 90 องศา เพื่อเปิดช่องสินค้า และหมุนกลับมาที่เดิม เพื่อปิดช่อง จากนั้นมอเตอร์ตัวที่ 2 จะเริ่มทำงาน เพื่อปล่อยสินค้าชิ้นถัดไปลงมา และในกรณีที่บาร์โค้ดที่ได้รับจาก Scanner นั้นไม่ตรงตามที่กำหนดไว้ในโปรแกรม มอเตอร์ทั้งสองตัวนั้นจะไม่ทำงาน จึงไม่มีการเปิดช่องสินค้า

ในการทดลองตอนที่ 3 การทดสอบระบบแอพพลิเคชั่นที่สร้างขึ้นจากโปรแกรม Filemaker โดยเมื่อทำการลงทะเบียนในระบบ หากข้อมูลถูกต้อง และครบถ้วนตามที่กำหนด บาร์โค้ดจะแสดงขึ้นบนหน้าจอ เมื่อนำไปแสกนจะสามารถสั่งการมอเตอร์ได้ หากในกรณีที่สินค้าในช่องนั้นๆหมด จะไม่มีการแสดงบาร์โค้ด และระบบสามารถ นำข้อมูลออกมาในรูปแบบของ .pdf ได้ เพื่อใช้สำหรับเก็บเป็น Database

5.2 ข้อเสนอแนะ

1. ตัวโครงสร้าง และการออกแบบมีผลอย่างมากต่อการเคลื่อนที่ของวัสดุภายใน เนื่องจากไม่มีการใช้หลักการทางไฟฟ้าเข้ามาช่วย จึงต้องออกแบบให้เหมาะสม
2. โครงสร้างมีผลต่อการรับน้ำหนัก และความหลากหลายของขนาดช่องใส่สินค้าหากมีการปรับเปลี่ยนจะทำงานได้เต็มประสิทธิภาพยิ่งขึ้น
3. เนื่องจากน้ำหนักวัสดุแต่ละช่องไม่เท่ากัน เลยทำให้เวลาของมอเตอร์ตัวที่ 2 เริ่มการทำงานแตกต่างกัน จึงต้องกำหนดดีเลย์ใหม่ทุกครั้งเมื่อเปลี่ยนวัสดุสินค้า หากทำให้สามารถปรับดีเลย์ได้ที่ตัวอุปกรณ์จะช่วยให้การทำงานได้ประสิทธิภาพ
4. เพิ่มขนาดของสินค้าในคลังให้มีความแตกต่างกัน และบาร์โค้ดต่างกัน เพื่อความหลากหลายของฟังก์ชันการใช้งาน
5. พัฒนาแอพพลิเคชั่นให้สามารถใช้ได้ในหลากหลายแพลตฟอร์มเพิ่มขึ้น

โดยโครงงานระบบควบคุมคลังสินค้าขนาดเล็กแบบอัตโนมัติสามารถนำมาพัฒนาเพื่อเพิ่มประสิทธิภาพการทำงาน หรือการพัฒนาให้สามารถเช็คจำนวนสินค้าออก และยอดสินค้าคงเหลือภายใน โดยการทำผ่านแอพพลิเคชั่นบนสมาร์ทโฟนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ARDUINO MEGA 2560. [ออนไลน์]. เข้าถึงได้จาก : https://5735512085.medium.com/ . (วันที่ค้นหาข้อมูล : 23 มกราคม 2567).

[2] ฟังก์ชันของ ARDUINO MEGA 2560. [ออนไลน์]. เข้าถึงได้จาก : https://lungmaker.com/arduino-mega-2560-. (วันที่ค้นหาข้อมูล : 23 มกราคม 2567).

[3] คุณสมบัติเฉพาะของ Arduino Mega. [ออนไลน์]. เข้าถึงได้จาก : https://www.researchgate.net/figure/Block-Diagram-1-ArduinoMEGA2560-The-Arduino-Mega-2560-is-a-type-of-microcontroller. (วันที่ค้นหาข้อมูล : 25 มกราคม 2567).

[4] การสื่อสารบน Arduino Mega. [ออนไลน์]. เข้าถึงได้จาก : https://blog.thaieasyelec.com/espino32-ch7-how-to-use-uart/1). (วันที่ค้นหาข้อมูล : 29 มกราคม 2567).

[5] การจ่ายไฟให้เซอร์โวมอเตอร์. [ออนไลน์]. เข้าถึงได้จาก : https:/www.artronshop.co.th/article/92. (วันที่ค้นหาข้อมูล : 23 สิงหาคม 2566).

[6] การสื่อสารบน Arduino Mega. [ออนไลน์]. เข้าถึงได้จาก : https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/. (วันที่ค้นหาข้อมูล : 29 มกราคม 2567).

[7] Filemaker Pro. [ออนไลน์]. เข้าถึงได้จาก : http://library.ricethailand.go.th/document/E-book/brrd5306004c0.pdf. (วันที่ค้นหาข้อมูล : 26 มีนาคม 2567).

[8] เซอร์โวมอเตอร์. [ออนไลน์]. เข้าถึงได้จาก : http://www.9engineer.com. (วันที่คันหาข้อมูล : 26 สิงหาคม 2566).

[9] เซอร์โวมอเตอร์. [ออนไลน์]. เข้าถึงได้จาก : http:/www.sangchaimeter.com. (วันที่ค้นหาข้อมูล : 26 สิงหาคม 2566).

[10] เซอร์โวมอเตอร์. [ออนไลน์]. เข้าถึงได้จาก : http:/www.stcontrol.com. (วันที่ค้นหาข้อมูล : 26 สิงหาคม 2566).

[11] เซอร์โวมอเตอร์. [ออนไลน์]. เข้าถึงได้จาก : http://www.stcontrol.com. (วันที่ค้นหาข้อมูล : 26 สิงหาคม 2566).

[12] ลักษณะเฉพาะของเซอร์โวมอเตอร์รุ่น MG995 [ออนไลน์]. เข้าถึงได้จาก : https:/www.aecexport.com. (วันที่ค้นหาข้อมูล : 26 สิงหาคม 2566).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

[13] หลักการทำงานของเซอร์โวมอเตอร์. [ออนไลน์]. เข้าถึงได้จาก : https:/www.artronshop.co.th/article (วันที่ค้นหาข้อมูล : 26 สิงหาคม 2566).

[14] LCD. [ออนไลน์]. เข้าถึงได้จาก : https://th.cytron.io/p-i2c-1602-serial-lcd-for-arduino-and-rpi (วันที่ค้นหาข้อมูล : 27 สิงหาคม 2566).

[15] LCD. [ออนไลน์]. เข้าถึงได้จาก : https://blog.thaieasyelec.com/how-to-use-character-lcd-display-arduino-ch1-parallel/. (วันที่ค้นหาข้อมูล : 27 สิงหาคม 2566).

[16] LCD. [ออนไลน์]. เข้าถึงได้จาก : https://th.element14.com/midas/mc128064a6w-bnmlw-v2/display-lcd-graphic-128x64-bstn/dp/2675690. (วันที่ค้นหาข้อมูล : 27 สิงหาคม 2566).

[17] LCD. [ออนไลน์]. เข้าถึงได้จาก : https://www.thaimicrotron.com/CCS628/Referrence/I2CBUS.htm. (วันที่ค้นหาข้อมูล : 27 สิงหาคม 2566).

[18] การสื่อสารบน Arduino Mega. [ออนไลน์]. เข้าถึงได้จาก : https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/. (วันที่ค้นหาข้อมูล : 29 มกราคม 2567).

[19] Karakuri. [ออนไลน์]. เข้าถึงได้จาก : https://www.haberkorn.cz/en/data/uploads/Banner/Stavebnicove_systemy/ss_karakuri.jpg. (วันที่ค้นหาข้อมูล : 27 สิงหาคม 2566).

[20] MH-ET LIVE SCANNER. [ออนไลน์]. เข้าถึงได้จาก : https://www.mikroelec.com/product/2075/barcode-and-qr-code-scanner-module-v3-0-1d-2d-codes-reader-two-dimensional-scanning-engine-barcode-s. (วันที่ค้นหาข้อมูล : 15 กุมภาพันธ์ 2567).

[21] โครงสร้าง MH-ET LIVE SCANNER. [ออนไลน์]. เข้าถึงได้จาก : https://www.modulemore.com/en/product/2094/mh-et-live-scanner-v3-barcode-reading-board-qr-code-scan. (วันที่ค้นหาข้อมูล : 15 กุมภาพันธ์ 2567).

[22] บาร์โค้ด. [ออนไลน์]. เข้าถึงได้จาก : https://www.denso-wave.com/en/adcd/fundamental/barcode/barcode/index.html. (วันที่ค้นหาข้อมูล : 15 กุมภาพันธ์ 2567).

[23] ชนิดบาร์โค้ด. [ออนไลน์]. เข้าถึงได้จาก : https://www.tiit.co.th/th/articles/105787-1d-barcode.2d-barcode. (วันที่ค้นหาข้อมูล : 15 กุมภาพันธ์ 2567).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดโปรแกรม

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <Servo.h> // Include the Servo library


String inputString = "";         // Variable to store data from the scanner

String DataScanner = "";         // Variable for processing data

bool stringComplete = false;     // Flag to check if data reception is complete


// Define LCD properties

LiquidCrystal_I2C lcd(0x27, 16, 2);  // Set the LCD address to 0x27 for a 16 chars and 2 line display

Servo ServoMotor1; // กำหนด ServoMotor1 object

Servo ServoMotor2; // กำหนด ServoMotor2 object

Servo ServoMotor3; // กำหนด ServoMotor1 object

Servo ServoMotor4; // กำหนด ServoMotor2 object

Servo ServoMotor5; // กำหนด ServoMotor1 object

Servo ServoMotor6; // กำหนด ServoMotor2 object


int position = 0; // ตัวแปรสถานะที่ใช้ควบคุม Servo

int ServoPosition1 = 10;  // ตำแหน่งเริ่มต้นของ ServoMotor1
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int ServoPosition2 = 10;  // ตำแหน่งเริ่มต้นของ ServoMotor2

int ServoPosition3 = 10;  // ตำแหน่งเริ่มต้นของ ServoMotor3

int ServoPosition4 = 10;  // ตำแหน่งเริ่มต้นของ ServoMotor4

int ServoPosition5 = 10;  // ตำแหน่งเริ่มต้นของ ServoMotor5

int ServoPosition6 = 10;  // ตำแหน่งเริ่มต้นของ ServoMotor6


void setup() {

    pinMode(8, OUTPUT); // Pin for triggering the scanner


    // Initialize serial communication

    Serial.begin(9600);

    while (!Serial) {

        ; // Wait for serial port to connect

    }


    // Initialize the LCD

    lcd.begin();

    lcd.backlight();  // Turn on the backlight


    ServoMotor1.attach(10); // แนบ ServoMotor1 ที่ขา 10

    ServoMotor2.attach(11); // แนบ ServoMotor2 ที่ขา 11
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ServoMotor3.attach(12); // แนบ ServoMotor1 ที่ขา 12
    ServoMotor4.attach(13); // แนบ ServoMotor2 ที่ขา 13
    ServoMotor5.attach(14); // แนบ ServoMotor1 ที่ขา 14
    ServoMotor6.attach(15); // แนบ ServoMotor2 ที่ขา 15

    // Reserve memory for the inputString
    inputString.reserve(200);

}

void loop() {
    digitalWrite(8, HIGH);   // Trigger the scanner
    delay(1000);
    digitalWrite(8, LOW);
    delay(1000);
    if (ServoPosition1 == 90 && ServoPosition3 == 90 && ServoPosition5 == 90 && position == 0 )
{
    ServoPosition1 = 10;
    ServoPosition2 = 0;
    ServoPosition3 = 10;
    ServoPosition4 = 0;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ServoPosition5 = 10;

ServoPosition6 = 0;

}


ServoMotor1.write(ServoPosition1); // กำหนดตำแหน่ง ServoMotor1

ServoMotor2.write(ServoPosition2); // กำหนดตำแหน่ง ServoMotor2

ServoMotor3.write(ServoPosition3); // กำหนดตำแหน่ง ServoMotor3

ServoMotor4.write(ServoPosition4); // กำหนดตำแหน่ง ServoMotor4

ServoMotor5.write(ServoPosition5); // กำหนดตำแหน่ง ServoMotor5

ServoMotor6.write(ServoPosition6); // กำหนดตำแหน่ง ServoMotor6


 lcd.clear();  // Clear the LCD screen

 lcd.setCursor(0, 0);  // Set the cursor to the beginning of the first line

 lcd.print("Please scan QR");


if (stringComplete) {

    if (DataScanner == "111111111\r\n")

    {

        Serial.println("Shelf.1");

        lcd.clear();  // Clear the LCD screen

        lcd.setCursor(0, 0);  // Set the cursor to the beginning of the first line
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lcd.print("Stand by Shelf.1");
    ServoPosition1 = 90;
    ServoMotor1.write(ServoPosition1); // กำหนดตำแหน่ง ServoMotor1
    position = 0;
    delay(1000); // Wait for the servo to reach the position
    ServoPosition1 = 10;
    ServoMotor1.write(ServoPosition1); // กำหนดตำแหน่ง ServoMotor1
    delay(1000); // Wait for the servo to return to initial position
    ServoMotor2.write(90); // กำหนดตำแหน่ง ServoMotor2
    position = 0;
    delay(200); // Wait for the servo to reach the position

}
else if (DataScanner == "222222222\r\n")
{
    Serial.println("Shelf.2");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Stand by Shelf.2");
    ServoPosition3 = 90;
    ServoMotor3.write(ServoPosition3); // กำหนดตำแหน่ง ServoMotor3
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    position = 0;
    delay(1000); // Wait for the servo to reach the position
    ServoPosition3 = 10;
    ServoMotor3.write(ServoPosition3); // กำหนดตำแหน่ง ServoMotor3
    delay(1000); // Wait for the servo to return to initial position
    ServoMotor4.write(180); // กำหนดตำแหน่ง ServoMotor4
    position = 0;
    delay(1000); // Wait for the servo to reach the position


    delay(200);
}
else if (DataScanner == "333333333\r\n")
{
    Serial.println("Shelf.3");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Stand by Shelf.3");
    ServoPosition5 = 100;
    ServoMotor5.write(ServoPosition5); // กำหนดตำแหน่ง ServoMotor5
    position = 0;
    delay(1000); // Wait for the servo to reach the position
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ServoPosition5 = 10;
    ServoMotor5.write(ServoPosition5); // กำหนดตำแหน่ง ServoMotor5
    delay(1000); // Wait for the servo to return to initial position
    ServoMotor6.write(180); // กำหนดตำแหน่ง ServoMotor6
    position = 0;
    delay(1000); // Wait for the servo to reach the position


    delay(200);
}
else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Incorrect");
    Serial.println("Incorrect");
    delay(200);
}


delay(1000);
lcd.clear();  // Clear the LCD screen
lcd.setCursor(0, 0);  // Set the cursor to the beginning of the first line
lcd.print("Please scan QR");
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // Clear variables for new data
        inputString = "";
        DataScanner = "";
        stringComplete = false;
    }
}

void serialEvent() {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        inputString += inChar;
        if (inChar == '\n') {
            stringComplete = true;
            DataScanner = String(inputString);
        }
    }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้