อภินันทนาการ

สำนักหอสมุด

ระบบจัดเก็บสินค้าและนำออกจากคลัง

WAREHOUSE STOCKPILING AND FETCHING SYSTEM

นายนรินทร์ มีบุญ รหัส 55360901

นายจีรวัฒน์ ช่างเก็บ รหัส 55362974

นายกฤษฎิ์ สวัสดี รหัส 55363759

สำนักหอสมุด มหาวิทยาลัยนเรศวร
17 ต.ค. 2560
วันลงทะเบียน.....
เลขทะเบียน.....
เลขเรียกหนังสือ.....

ป
น4455
2558

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2558

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	ระบบจัดเก็บสินค้าและนำออกจากคลัง	
ผู้ดำเนินโครงงาน	นายนรินทร์ มีบุญ	รหัส 55360901
	นายจีรวัฒน์ ช่างเก็บ	รหัส 55362974
	นายกฤษฎิ์ สวัสดี	รหัส 55363759
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2558	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. ศุภวรรณ พลพิทักษ์ชัย)

.....กรรมการ
(ดร. สราวุฒิ วัฒนวงค์พิทักษ์)

ชื่อหัวข้อโครงงาน	ระบบจัดเก็บสินค้าและนำออกจากคลัง	
ผู้ดำเนินโครงงาน	นายนรินทร์ มีบุญ	รหัส 55360901
	นายจีรวัฒน์ ช่างเก็บ	รหัส 55362974
	นายกฤษฎิ์ สวัสดี	รหัส 55363759
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2558	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบระบบจัดเก็บสินค้าและนำออกจากคลังเพื่อช่วยแก้ปัญหาความล่าช้าในระบบจัดเก็บสินค้าและนำออกจากคลังรวมถึงมีความแม่นยำและง่ายต่อการตรวจสอบจำนวนสินค้าภายในคลัง และสร้างแบบจำลองคลังเก็บสินค้าจำนวน 2 ชั้น แต่ละชั้นแบ่งเป็น 3 แถว แต่ละแถวเก็บสินค้าได้ 3 ชิ้น ใช้ไมโครคอนโทรลเลอร์ในการประมวลผลและสั่งงานให้ชุดอุปกรณ์เคลื่อนที่เข้าและออกจากคลัง ซึ่งประกอบด้วยชุดการเคลื่อนที่ช่องลิฟต์เพื่อให้เคลื่อนที่ไปในแต่ละแถวหน้าคลังสินค้า ชุดลิฟต์เพื่อเคลื่อนที่ขึ้นไปยังชั้นที่ 2 และแท่นขนย้ายสินค้าใช้ในการเคลื่อนที่เข้าและออกจากคลัง โดยใช้สวิตช์จำกัดระยะเป็นตัวระบุตำแหน่งในการเคลื่อนที่ของชุดอุปกรณ์ และได้ทำการออกแบบระบบให้มีการทำงานเป็น 2 แบบวิธี คือ แบบวิธีการจัดเก็บสินค้าซึ่งใช้ตัวรับรู้แบบใช้แสงในการตรวจจับสินค้าเพื่อเริ่มทำการจัดเก็บ และแบบวิธีการนำสินค้าออก โดยผู้ใช้สามารถป้อนจำนวนสินค้าที่ต้องการนำออกผ่านแผงแป้นตัวเลข นอกจากนี้สถานะการทำงานและจำนวนสินค้าคงคลังยังถูกแสดงบนหน้าจอแอลซีดี

Project title	Warehouse Stockpiling and Fetching System		
Name	Mr. Narin	Meeboon	ID. 55360901
	Mr. Jeerawat	Changkeb	ID. 55362974
	Mr. Krit	Sawatdee	ID. 55363759
Project advisor	Asst. Prof. Niphat Jantharamin, Ph.D.		
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic year	2015		

Abstract

This thesis presents a design of a warehouse stockpiling and fetching system, which could speed the process of product stockpiling and fetching and provide accuracy and convenience of a product count in a warehouse. A model of warehouse stockpiling and fetching system is built. This warehouse model has 2 floors, each floor has 3 rows, each of which can stock 3 items. Hereby, a microcontroller controls motors that drive three transporting parts, namely a lift shaft, a lift and a transporting platform. The lift shaft is moved to each row, the platform is then lifted to the second floor if needed, and finally moves a product into or out of a programmed position. Movement of each transporting part is regulated by limit switches. The model operates in 2 modes: Stockpiling mode and Fetching mode. An infrared sensor is used for product detection which starts the stockpiling process. List of the stockpiling and fetching tasks are designed to be done in a specific order. User can enter a number of products to be fetched via a numeric keypad, and the system status is displayed through an LCD.

กิตติกรรมประกาศ

ผู้ดำเนินโครงงานขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์ อาจารย์ที่ปรึกษาโครงงาน ซึ่งเอาใจใส่ในรายละเอียดทุกขั้นตอนของการดำเนินโครงงาน โดยให้คำปรึกษาและคำแนะนำในการแก้ไขปัญหาต่าง ๆ อย่างต่อเนื่องจนกระทั่งโครงงานสำเร็จลุล่วง รวมถึงแนะนำหลักการเขียนปริญญานิพนธ์และตรวจทานแก้ไขอย่างละเอียดจนได้ปริญญานิพนธ์เป็นรูปเล่มสมบูรณ์

ขอขอบคุณกรรมการสอบโครงงาน ผู้ช่วยศาสตราจารย์ ดร. ศุภวรรณ พลพิทักษ์ชัย และ ดร. สราวุฒิ วัฒนวงค์พิทักษ์ ซึ่งกรุณาให้คำแนะนำที่เป็นประโยชน์ในการดำเนินโครงงาน

และขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ยืมเครื่องมือในการสร้างชิ้นงานขึ้นโดยเฉพาะว่าที่ร้อยตรีธานี โกสุม (พี่ต้น) ซึ่งเป็นรุ่นพี่สาขาวิชาวิศวกรรมไฟฟ้าและปัจจุบันเป็นครูช่างของภาควิชาฯที่อำนวยความสะดวกในการใช้ห้องปฏิบัติการและการยืมใช้เครื่องมือจนกระทั่งสร้างชิ้นงานเสร็จสิ้นลง

รวมทั้งขอขอบคุณรัฐบาลไทยที่จัดตั้งกองทุนเงินให้กู้ยืมเพื่อการศึกษา (กยศ.) ซึ่งสนับสนุนด้านทุนทรัพย์ให้แก่นายนรินทร์ มีบุญ และนายจีรวัฒน์ ช่างเก็บ ตลอดระยะเวลาการศึกษาในระดับปริญญาตรี

เหนือสิ่งอื่นใด ผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดาและมารดาที่ให้การสนับสนุนในทุกด้านเกี่ยวกับการศึกษาของผู้ดำเนินโครงงาน รวมทั้งมอบความรัก ความเมตตา และคอยเป็นกำลังใจให้จนประสบความสำเร็จในวันนี้

นายนรินทร์ มีบุญ
นายจีรวัฒน์ ช่างเก็บ
นายกฤษฎิ์ สวัสดี

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 4
2.1 มอเตอร์ไฟฟ้ากระแสตรง 4
2.1.1 โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง 4
2.1.2 การทำงานเป็นมอเตอร์ไฟฟ้า 6
2.1.3 การขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง 7
2.1.4 การกลับทิศทางการหมุนของมอเตอร์ 9
2.2 รีเลย์ 10
2.3 สวิตช์จำกัดระยะ 13
2.4 ตัวรับรู้แบบใช้แสง 14
2.5 แผงแป้นตัวเลข 16
2.6 ไมโครคอนโทรลเลอร์ 17
2.7 จอแสดงผลแอลซีดี 20

สารบัญ (ต่อ)

หน้า

2.8 ไอทูซี 21

บทที่ 3 การออกแบบและสร้างแบบจำลองระบบจัดเก็บสินค้าและนำออกจากคลัง 24
3.1 ขั้นตอนการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง 24
3.2 การออกแบบโครงสร้างและชุดการเคลื่อนที่ระบบจัดเก็บสินค้าและนำออกจากคลัง 26
3.3 รูปแบบการจัดเก็บสินค้าและนำออกจากคลัง 28
3.3.1 การออกแบบการจัดวางตำแหน่งในคลังสินค้า 28
3.3.2 การออกแบบลำดับการจัดเก็บสินค้าเข้าคลัง 28
3.3.3 การออกแบบลำดับการนำสินค้าออกจากคลัง 29
3.4 การออกแบบขั้นตอนการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง 31
3.5 การออกแบบการเชื่อมต่อวงจรของระบบจัดเก็บสินค้าและนำออกจากคลัง 33
3.6 การติดตั้งอุปกรณ์ของแบบจำลองระบบจัดเก็บสินค้าและนำออกจากคลัง 34
3.6.1 การติดตั้งตัวรับรู้แบบแสง 34
3.6.2 การติดตั้งสวิตช์จำกัดระยะและก้านสัมผัส 34
3.6.3 กล่องควบคุมการทำงานของระบบ 36
3.7 วิธีการใช้งานระบบจัดเก็บสินค้าและนำออกจากคลัง 38
3.8 แบบจำลองโครงสร้างระบบจัดเก็บสินค้าและนำออกจากคลัง 39

บทที่ 4 ผลการทดสอบ 40
4.1 การทดสอบการทำงานของรีเลย์ 40
4.1.1 แบบวิธีการจัดเก็บสินค้า 41
4.1.2 แบบวิธีการนำสินค้าออกจากคลัง 42
4.2 การทดสอบหาระยะเวลาที่ใช้ในการจัดเก็บและนำสินค้าออกจากคลัง 43
4.3 การทดสอบการเปลี่ยนแบบวิธี 46
4.3.1 การทดสอบเปลี่ยนแบบวิธีขณะช่องลิฟต์เคลื่อนที่ไปหน้าคลัง 46
4.3.2 การทดสอบเปลี่ยนแบบวิธีขณะลิฟต์เคลื่อนที่ขึ้น 47
4.3.3 การทดสอบเปลี่ยนแบบวิธีขณะแท่นขนย้ายเคลื่อนที่เข้าไปในคลัง 47
4.3.4 การทดสอบเปลี่ยนแบบวิธีขณะแท่นขนย้ายเคลื่อนที่ออกจากคลัง 48

สารบัญ (ต่อ)

หน้า

4.3.5 การทดสอบเปลี่ยนแบบวิธีขณะลิฟต์เคลื่อนที่ลง....49
4.3.6 การทดสอบเปลี่ยนแบบวิธีขณะช่องลิฟต์เคลื่อนที่กลับ....49

บทที่ 5 สรุปผลและข้อเสนอแนะ....50
5.1 สรุปผลการดำเนินงาน....50
5.2 ปัญหาและแนวทางการแก้ไข....51
5.3 แนวทางในการพัฒนาต่อไป....51

เอกสารอ้างอิง....52

ภาคผนวก ก รหัสต้นฉบับของโปรแกรมควบคุมระบบการจัดเก็บสินค้าและนำออกจากคลัง....53
ภาคผนวก ข รายละเอียดข้อมูลของ Arduino MEGA 2560....75
ภาคผนวก ค รายละเอียดข้อมูลของรีเลย์....79
ภาคผนวก ง รายละเอียดข้อมูลของไอซีคุมค่าแรงดัน รุ่น LM7805....82

ประวัติผู้ดำเนินโครงงาน....89

สารบัญตาราง

ตารางที่ หน้า

2.1 สัญลักษณ์ขาของไอทูซี 22
4.1 หน้าที่ของรีเลย์แต่ละตัวในระบบจัดเก็บสินค้าและนำออกจากคลัง 40
4.2 ผลการทดสอบการทำงานของรีเลย์ในแบบวิธีการจัดเก็บสินค้า 41
4.3 ผลการทดสอบการทำงานของรีเลย์ในแบบวิธีการนำสินค้าออกจากคลัง 42
4.4 ผลการทดสอบหาระยะเวลาในการจัดเก็บสินค้า 44
4.5 ผลการทดสอบหาระยะเวลาในการนำสินค้าออกจากคลัง 45

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่ หน้า

2.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง 4
2.2 ขั้วแม่เหล็กที่สร้างจากการพันขดลวด 5
2.3 โรเตอร์ 5
2.4 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง 6
2.5 การขับเคลื่อนมอเตอร์กระแสตรงด้วยตัวเรียงกระแสแบบควบคุมเฟส 7
2.6 การขับเคลื่อนมอเตอร์กระแสตรงด้วยตัวแปลงผันกำลังกระแสตรง 7
2.7 แผนภาพวงจรการกลับขั้วขดลวดอาร์เมเจอร์ 9
2.8 แผนภาพวงจรการกลับขั้วขดลวดสนาม 9
2.9 สัญลักษณ์ภายในโครงสร้างของรีเลย์ 10
2.10 สภาวะการทำงานของรีเลย์ 10
2.11 แผงวงจรรีเลย์แบบมีตัวเชื่อมต่อทางแสง 11
2.12 แผนภาพวงจรการทำงานภายในของรีเลย์ 12
2.13 สัญลักษณ์สวิตช์จำกัดระยะ 13
2.14 กลไกลการทำงานภายในของสวิตช์จำกัดระยะ 14
2.15 แผนภาพการใช้งานตัวรับรู้ 14
2.16 ตัวรับรู้แบบใช้แสง ระยะตรวจจับ 3-80 cm 15
2.17 ส่วนประกอบหลักของตัวรับรู้แบบใช้แสง 15
2.18 แผงแป้นตัวเลข แบบ 4 x 4 16
2.19 แผงวงจร Arduino Mega2560 17
2.20 หน่วยประมวลผลกลางของ ATmega2560 ขนาด 100 ขา 19
2.21 จอแสดงผลแอลซีดี Hitachi หมายเลข HD44780 20
2.22 การเชื่อมต่อไมโครคอนโทรลเลอร์กับจอแอลซีดี 21
2.23 ไอทูซี 21
2.24 การรับและส่งข้อมูลแบบ I2C BUS 22
2.25 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ กับจอแอลซีดี 23
2.26 การแสดงการเชื่อมต่อของ I2C กับจอแอลซีดี 23
3.1 ส่วนประกอบและหลักการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง 24
3.2 โครงสร้างคลังเก็บสินค้า 26

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.3 ชุดการเคลื่อนที่ของช่องลิฟต์ 26
3.4 ชุดการเคลื่อนที่ของลิฟต์ 27
3.5 ชุดการเคลื่อนที่ของแท่นขนย้ายสินค้า 27
3.6 ตำแหน่งสินค้าในคลัง 28
3.7 ลำดับการจัดเก็บสินค้าชั้นที่ 1 29
3.8 ลำดับการจัดเก็บสินค้าชั้นที่ 2 29
3.9 ลำดับการนำสินค้าออกจากคลังชั้นที่ 1 30
3.10 ลำดับการนำสินค้าออกจากคลังชั้นที่ 2 30
3.11 ขั้นตอนการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง 31
3.12 การเชื่อมต่อวงจรของระบบจัดเก็บสินค้าและนำออกจากคลัง 33
3.13 การติดตั้งตัวรับรู้แบบใช้แสง 34
3.14 การติดตั้งสวิตช์จำกัดระยะและก้านสัมผัสเพื่อควบคุมการเคลื่อนที่ของแท่นขนย้าย 35
3.15 การติดตั้งสวิตช์จำกัดระยะและก้านสัมผัสเพื่อควบคุมการเคลื่อนที่ของช่องลิฟต์ 35
3.16 การติดตั้งสวิตช์จำกัดระยะเพื่อควบคุมการเคลื่อนที่ของช่องลิฟต์ 36
3.17 หน้าจอแสดงผลและปุ่มควบคุมการทำงาน 36
3.18 การติดตั้งอุปกรณ์ภายในกล่องควบคุม 37
3.19 แผนภาพวงจรไฟเลี้ยงโดยใช้ IC 7805 37
3.20 การแสดงผลของหน้าจอเลือกแบบวิธีการทำงาน 38
3.21 การแสดงผลของหน้าจอขณะการจัดเก็บสินค้า 38
3.22 รูปแบบการแสดงผลของหน้าจอขณะนำสินค้าออกจากคลัง 39
3.23 แบบจำลองระบบจัดเก็บสินค้าและนำออกจากคลัง 39
4.1 ระยะเวลาที่ใช้ในการนำสินค้าออกจากคลัง 45
4.2 การเคลื่อนที่ของช่องลิฟต์ขณะเคลื่อนที่ไปหน้าคลัง 46
4.3 การเคลื่อนที่ของลิฟต์ขณะเคลื่อนที่ขึ้น 47
4.4 การเคลื่อนที่ของแท่นขนย้ายขณะเคลื่อนที่เข้าในคลัง 48
4.5 การเคลื่อนที่ของแท่นขนย้ายขณะเคลื่อนที่ออกจากคลัง 48
4.6 การเคลื่อนที่ของลิฟต์ขณะเคลื่อนที่ลง 49
4.7 การเคลื่อนที่ของช่องลิฟต์ขณะเคลื่อนที่กลับมาจุดเริ่มต้น 49

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากโรงงานอุตสาหกรรมหลายประเภท ที่มีระบบการผลิตและจัดเก็บสินค้า เพื่อรอเวลาส่งออก ควรมีระบบจัดเก็บสินค้าและนำออกจากคลังที่มีประสิทธิภาพทั้งในเรื่องของความเป็นระเบียบเรียบร้อย ความรวดเร็ว ง่ายต่อการตรวจสอบจำนวนสินค้า รวมถึงประหยัดเงินทุนที่ใช้ ซึ่งในโรงงานอุตสาหกรรมที่มีคลังสินค้าขนาดใหญ่และมีการใช้บุคลากรทำหน้าที่ในระบบจัดเก็บสินค้าและนำออกจากคลัง อาจมีความล่าช้าในการจัดเก็บและนำออกเนื่องจากเคลื่อนย้ายสินค้านั้นมีระยะทางไกล และร่างกายของมนุษย์มีขีดจำกัดด้านศักยภาพการทำงาน รวมถึงในระบบจัดเก็บสินค้าและนำออกจากคลังสินค้าที่มีขนาดใหญ่จำเป็นต้องใช้แรงงานจำนวนมากเพื่อเพิ่มประสิทธิภาพในการทำงานจึงเป็นการเพิ่มค่าใช้จ่ายของผู้ประกอบการ นอกจากนี้การตรวจนับจำนวนสินค้าในคลังสินค้าขนาดใหญ่โดยใช้แรงงานคนนั้น นอกจากจะเกิดความล่าช้าในการตรวจสอบจำนวนสินค้าแล้ว หากผู้ที่ทำหน้าที่บันทึกจำนวนสินค้าไม่มีความละเอียดรอบคอบเพียงพอก็อาจทำให้เกิดความผิดพลาดได้ง่าย

ในโครงงานนี้ได้เล็งเห็นถึงปัญหาดังกล่าว จึงได้พัฒนาระบบจัดเก็บสินค้าและนำออกภายในคลังสินค้าของโรงงานอุตสาหกรรม เพื่อเพิ่มประสิทธิภาพของระบบจัดเก็บสินค้าและนำออกจากคลัง โดยจะใช้เทคโนโลยีเข้ามาแทนที่การใช้บุคลากรในการทำงาน โดยควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ เพื่อช่วยแก้ปัญหาการความล่าช้าในระบบจัดเก็บสินค้าและนำออกจากคลังรวมถึงมีความแม่นยำและง่ายต่อการตรวจสอบจำนวนสินค้าภายในคลัง

1.2 วัตถุประสงค์ของโครงงาน

เพื่อพัฒนาระบบจัดเก็บสินค้าและนำออกจากคลังที่มีมากกว่าหนึ่งชั้นและสร้างแบบจำลองแสดงการทำงานของระบบ ซึ่งควบคุมด้วยไมโครคอนโทรลเลอร์

1.3 ขอบเขตของโครงงาน

1) สร้างแบบจำลองคลังสินค้าที่มีจำนวน 2 ชั้น แต่ละชั้นมี 3 แถว แต่ละแถวสามารถจัดเก็บสินค้าได้ 3 ชิ้น รวมถึงสร้างชุดการเคลื่อนที่ช่องลิฟต์ ลิฟต์ และแท่นขนย้ายสินค้า
2) ใช้ไมโครคอนโทรลเลอร์ควบคุมการทำงานของแบบจำลองและใช้สวิตช์จำกัดระยะในการกำหนดตำแหน่งการจัดเก็บและนำสินค้าออกจากคลัง
3) แบบจำลองสามารถแสดงการจัดเก็บและนำสินค้าออกจากคลังโดยเริ่มจากตำแหน่งที่ใกล้ก่อนเพื่อความรวดเร็วในการทำงานและผู้ใช้สามารถเปลี่ยนเป็นแบบวิธีนำสินค้าออกได้ทุกขณะการทำงานในแบบวิธีจัดเก็บรวมถึงแสดงจำนวนสินค้าที่มีอยู่เป็นตัวเลขบนหน้าจอแอลซีดี

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	พ.ศ. 2558					พ.ศ. 2559			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1) ออกแบบโครงสร้างและการทำงานของระบบ									
2) สร้างแบบจำลองของระบบ									
3) ใช้ไมโครคอนโทรลเลอร์ควบคุมการทำงานของแบบจำลอง									
4) ทดสอบการทำงานและปรับปรุงชิ้นงาน									
5) สรุปผลและจัดทำรูปเล่มปริญญานิพนธ์									

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ระบบจัดเก็บสินค้าและนำออกจากคลังที่พัฒนาขึ้นในโครงงานนี้เป็นระบบที่ควบคุมแบบอัตโนมัติด้วยไมโครคอนโทรลเลอร์ สามารถเพิ่มความสะดวกรวดเร็วในการจัดเก็บและนำสินค้าออก รวมถึงมีความแม่นยำและง่ายต่อการตรวจสอบจำนวนสินค้า อีกทั้งยังลดความเสี่ยงที่จะเกิดอันตรายต่อคนงานและลดต้นทุนค่าแรงงานสำหรับทำหน้าที่จัดเก็บสินค้าและนำออกจากคลัง

1.6 งบประมาณ

1) ไมโครคอนโทรลเลอร์ Arduino Mega 2560	550 บาท
2) ค่าวัสดุอุปกรณ์ในการทำโครงสร้างของแบบจำลอง	2,000 บาท
3) มอเตอร์ไฟฟ้ากระแสตรง 5 V จำนวน 3 ตัว	300 บาท
4) มอเตอร์ไฟฟ้ากระแสตรง 12 V จำนวน 2 ตัว	600 บาท
5) วงจรควบคุมมอเตอร์	300 บาท
6) จอแสดงผลแอลซีดีขนาด 20×4	350 บาท
7) ตัวรับรู้แบบใช้แสง	300 บาท
8) ค่าถ่ายเอกสารและเข้าเล่มปริญญานิพนธ์	800 บาท
รวมเป็นเงินทั้งสิ้น (ห้าพันสองร้อยบาทถ้วน)	5,200 บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

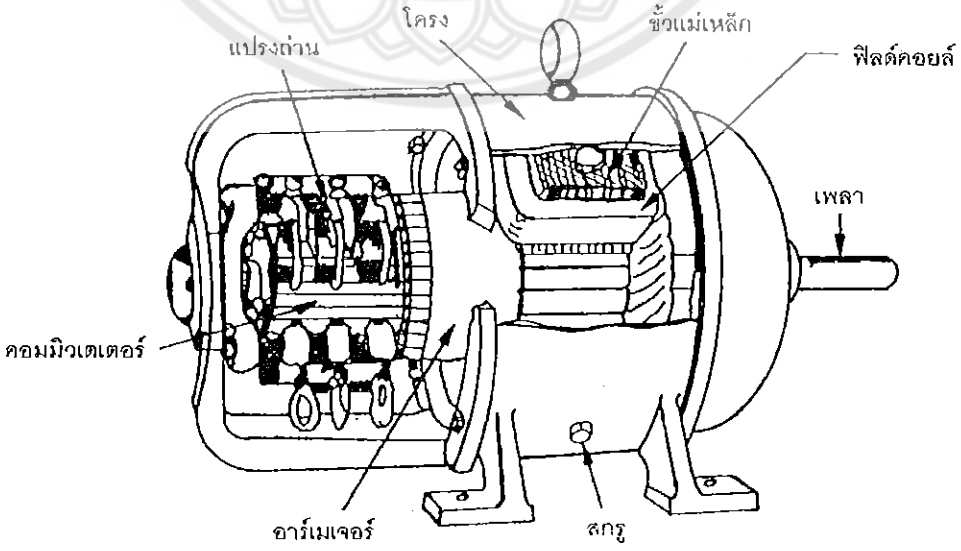
บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

เนื่องจากโครงงานนี้ได้พัฒนาระบบการจัดเก็บสินค้าและนำออกจากคลัง โดยสร้างเป็นแบบจำลองระบบการทำงาน ซึ่งมีส่วนเคลื่อนที่ 3 ส่วน คือ แท่นขนย้ายสินค้า ลิฟต์ และช่องลิฟต์ โดยกลไกการขับเคลื่อนทั้งสามส่วนดังกล่าวใช้มอเตอร์ไฟฟ้ากระแสตรง และใช้รีเลย์ทำหน้าที่ตัดต่อวงจรจ่ายไฟให้กับมอเตอร์รวมทั้งควบคุมการกลับทิศการหมุนของมอเตอร์ และใช้สวิตช์จำกัดระยะ ในการระบุตำแหน่งการเคลื่อนที่ของทั้งสามส่วน โดยการจัดเก็บสินค้าเข้าคลังจะเริ่มต้นหลังจากที่ตัวรับรู้แสงตรวจพบว่ามีสินค้าอยู่บนแท่นขนย้าย ในขณะที่จำนวนสินค้าที่จะนำออกจากคลังถูกกำหนดผู้ใช้ผ่านแผงแป้นตัวเลข ซึ่งตัวรับรู้แสงและแผงแป้นตัวเลขจะส่งสัญญาณให้ไมโครคอนโทรลเลอร์ประมวลผลเพื่อควบคุมการทำงานของส่วนเคลื่อนที่และแสดงสถานะการทำงานว่ากำลังจับเก็บหรือกำลังนำสินค้าออกจากคลังรวมทั้งแสดงจำนวนสินค้าคงคลังผ่านจอแอลซีดี

2.1 มอเตอร์ไฟฟ้ากระแสตรง

2.1.1 โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง

โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรงแสดงได้ดังรูปที่ 2.1 โดยมีส่วนประกอบดังนี้

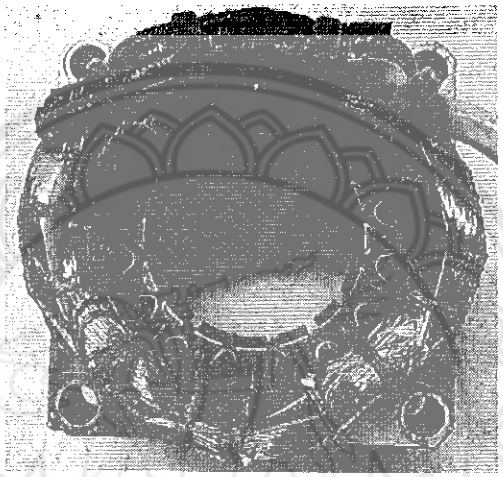


รูปที่ 2.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง [1]

1) สเตเตอร์ (Stator) เป็นส่วนของมอเตอร์ไฟฟ้าที่อยู่กับที่ ประกอบด้วย

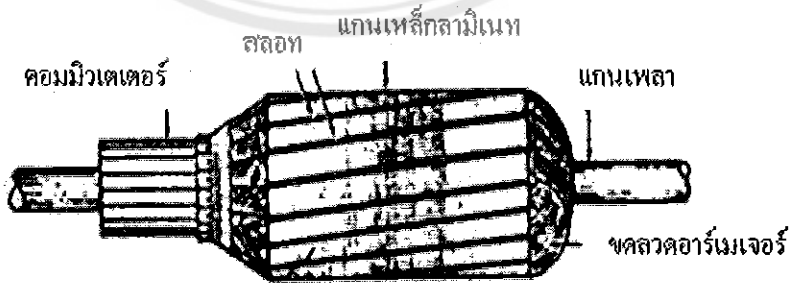
ก) เปลือกหรือโครง (Frame) เป็นทางเดินให้กับเส้นแรงแม่เหล็กจากขั้วเหนือไปยังขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่น ให้มีความแข็งแรง ทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาม้วนเป็นรูปทรงกระบอกกลม

ข) ขั้วสนามแม่เหล็ก (Field poles) เป็นส่วนที่ใช้ในการสร้างฟลักซ์แม่เหล็กแสดงได้ดังรูปที่ 2.2 เมื่อตัวนำในขดลวดอาร์เมเจอร์หมุนตัดผ่านฟลักซ์แม่เหล็กนี้จะเกิดการเหนี่ยวนำขึ้น



รูปที่ 2.2 ขั้วแม่เหล็กที่สร้างจากการพันขดลวด [2]

2) โรเตอร์ (Rotor) เป็นส่วนที่ทำให้เกิดกำลังงานมีลักษณะดังรูปที่ 2.3 ซึ่งมีแกนวางอยู่ในรองลื่น (Bearing) ประกอบอยู่ในแผ่นปิดหัวท้าย (End plate) ของมอเตอร์



รูปที่ 2.3 โรเตอร์ [1]

โรเตอร์ของมอเตอร์ไฟฟ้ากระแสตรงประกอบด้วย 4 ส่วน คือ

ก) แกนเพลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์เมเจอร์ (Armature core) ประกอบเป็นตัวโรเตอร์ แกนเพลานี้วางอยู่บนรองลื่นเพื่อบังคับให้หมุนโดยไม่มีการสั่นสะเทือน

ข) แกนเหล็กอาร์เมเจอร์ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated sheet steel) เป็นที่สำหรับพันขดลวดอาร์เมเจอร์ซึ่งสร้างแรงบิด (Torque)

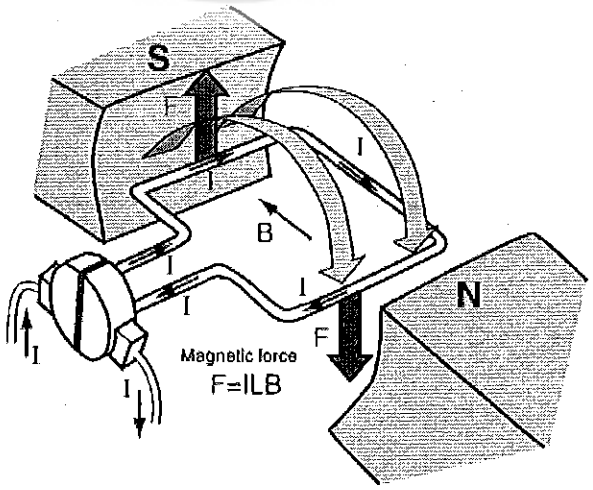
ค) คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่ แต่ละซี่มีฉนวนไมกา (Mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์มีร่องสำหรับใส่ปลายสายของขดลวดอาร์เมเจอร์ ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลาเป็นทรงกระบอกมีหน้าที่สัมผัสกับแปรงถ่าน (Carbon brushes)

ง) ขดลวดอาร์เมเจอร์ (Armature winding) เป็นขดลวดพันอยู่ในร่องสล็อต (Slot) ของแกนอาร์เมเจอร์ ขนาดของลวดจะเล็กหรือใหญ่ และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้น เพื่อให้เหมาะสมกับงานต่าง

3) แปรงถ่าน อาจทำจากส่วนผสมของคาร์บอนกับกราไฟต์ หรือคาร์บอนกับทองแดงทำหน้าที่รับกระแสไฟฟ้าจากวงจรภายนอกส่งไปยังคอมมิวเตเตอร์

2.1.2 การทำงานเป็นมอเตอร์ไฟฟ้า

มอเตอร์ไฟฟ้าคือ เครื่องจักรกลที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกลโดยอาศัยหลักการคือ เมื่อมีกระแสไหลผ่านตัวนำที่วางอยู่ในสนามแม่เหล็กจะทำให้ลวดตัวนำเกิดการเคลื่อนที่ หลักการของมอเตอร์ไฟฟ้ากระแสตรงแสดงดังรูปที่ 2.4

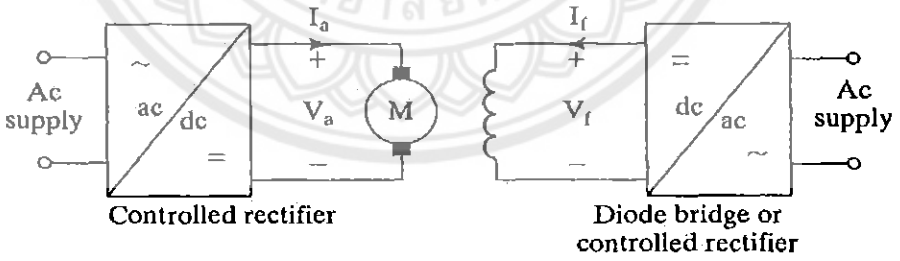


รูปที่ 2.4 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง [3]

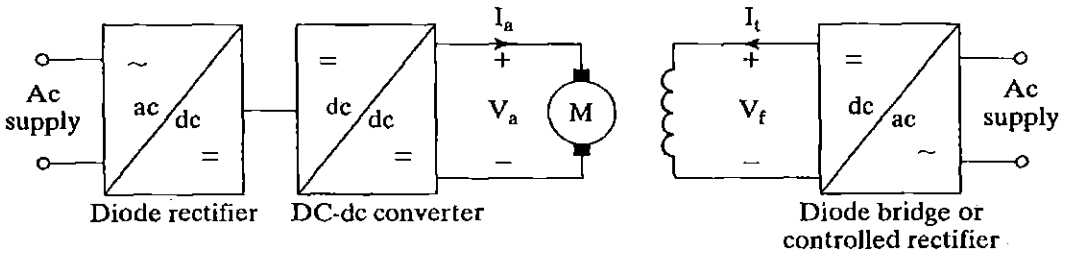
เมื่อมีแรงดันไฟฟ้าจ่ายผ่านแปรงถ่านไปคอมมิวเตเตอร์และขดลวดตัวนำที่อาร์เมเจอร์ ทำให้ขดลวดอาร์เมเจอร์เกิดสนามแม่เหล็กไฟฟ้าขึ้น ทางด้านขวามือเป็นขั้วเหนือ (N) และด้านซ้ายมือเป็นขั้วใต้ (S) เหมือนกับขั้วแม่เหล็กถาวรที่วางอยู่ใกล้ ๆ กันทำให้เกิดอำนาจแม่เหล็กผลักดันกัน อาร์เมเจอร์หมุนไปในทิศทางตามเข็มนาฬิกา พร้อมกับคอมมิวเตเตอร์หมุนตามไปด้วย แปรงถ่านสัมผัสกับส่วนของคอมมิวเตเตอร์เปลี่ยนไปในอีกปลายหนึ่งของขดลวด แต่มีผลทำให้เกิดขั้วแม่เหล็กที่อาร์เมเจอร์เหมือนกับขั้วแม่เหล็กถาวรที่อยู่ใกล้ ๆ อีกครั้ง ทำให้อาร์เมเจอร์ยังคงถูกผลักให้หมุนไปในทิศทางตามเข็มนาฬิกาตลอดเวลา ส่งผลให้เกิดการหมุนของอาร์เมเจอร์ ซึ่งหมายถึงเครื่องจักรกลกำลังทำงานเป็นมอเตอร์ไฟฟ้า

2.1.3 การขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

การขับเคลื่อนมอเตอร์กระแสตรงจากตัวเรียงกระแสแบบควบคุมเฟสสร้างแรงดันด้านออกกระแสตรงที่ปรับค่าได้จากแรงดันไฟกระแสสลับที่มีค่าคงที่ ในขณะที่ตัวแปลงผันกำลังกระแสตรงสร้างแรงดันไฟกระแสตรงที่ปรับค่าได้จากแรงดันกระแสตรงที่มีค่าคงที่ ด้วยคุณสมบัติในการสร้างแรงดันไฟกระแสตรงที่ปรับค่าได้อย่างต่อเนื่อง ตัวเรียงกระแสแบบควบคุมเฟสและตัวแปลงผันกำลังกระแสตรงจึงก่อให้เกิดวิวัฒนาการทางด้านอุปกรณ์ควบคุมและการขับเคลื่อนมอเตอร์แบบปรับความเร็วรอบได้ในอุตสาหกรรมสมัยใหม่ที่มีระดับกำลังไฟฟ้าตั้งแต่ไม่กี่แรงม้าจนถึงหลายเมกะวัตต์ ตัวเรียงกระแสแบบควบคุมเฟสนิยมใช้ในการปรับความเร็วรอบของมอเตอร์กระแสตรงดังรูปที่ 2.5 อีกหนึ่งทางเลือกคือการใช้ตัวเรียงกระแสแบบไดโอดร่วมกับตัวแปลงผันกำลังกระแสตรงดังรูปที่ 2.6 [4]



รูปที่ 2.5 การขับเคลื่อนมอเตอร์กระแสตรงด้วยตัวเรียงกระแสแบบควบคุมเฟส [4]



รูปที่ 2.6 การขับเคลื่อนมอเตอร์กระแสตรงด้วยตัวแปลงผันกำลังกระแสตรง [4]

สมการที่เกี่ยวข้องกับมอเตอร์ไฟฟ้ากระแสตรง

$$E_g = k \cdot I_f \cdot \omega \tag{2.1}$$

$$V_a = R_a I_a + E_g = R_a I_a + k \cdot I_f \cdot \omega \tag{2.2}$$

$$T_d = k \cdot I_f \cdot I_a = B\omega + T_L \tag{2.3}$$

โดยที่

E_g คือ แรงเคลื่อนไฟฟ้าตีกลับ (Back emf) มีหน่วย V

Va คือ แรงดันตกคร่อมอาร์เมเจอร์ มีหน่วย V

K คือ ค่าคงที่ของมอเตอร์ มีหน่วย V/A.rad/s

I_f คือ กระแสสนาม (Field current) มีหน่วย A

I_a คือ กระแสอาร์เมเจอร์ (Armature current) มีหน่วย A

ω คือ ความเร็วเชิงมุมของมอเตอร์ มีหน่วย rad/s

R_a คือ ความต้านทานของขดลวดอาร์เมเจอร์ มีหน่วย Ω

T_d คือ แรงบิด (Developed torque) มีหน่วย N·m

T_L คือ แรงบิดโหลด (Load torque) มีหน่วย N·m

B คือ ค่าคงที่แรงเสียดทาน มีหน่วย N·m/rad/s

กำลังไฟฟ้าที่มอเตอร์สร้างขึ้นคำนวณหาได้จาก

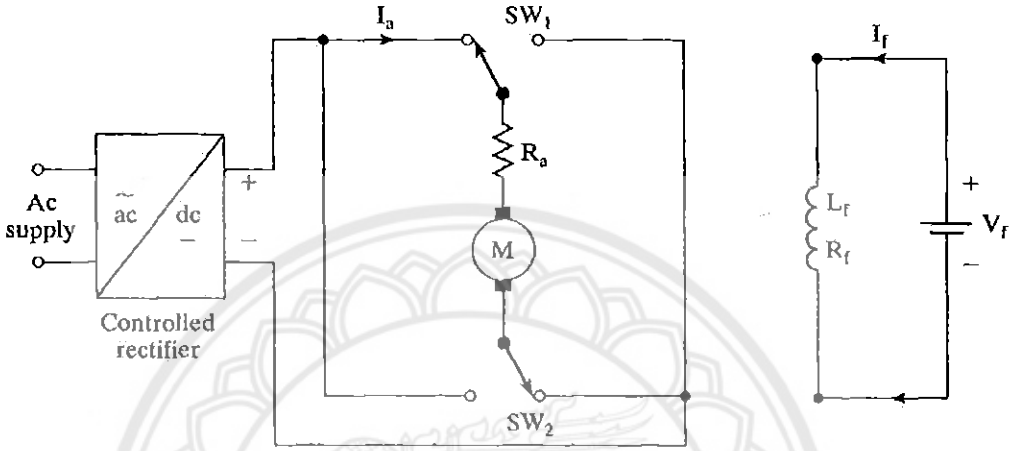
$$P_d = T_d \omega \tag{2.4}$$

จาก (2.1) เราสามารถเขียนสมการความเร็วรอบของมอเตอร์แบบกระตุ้นแยกได้ดังนี้

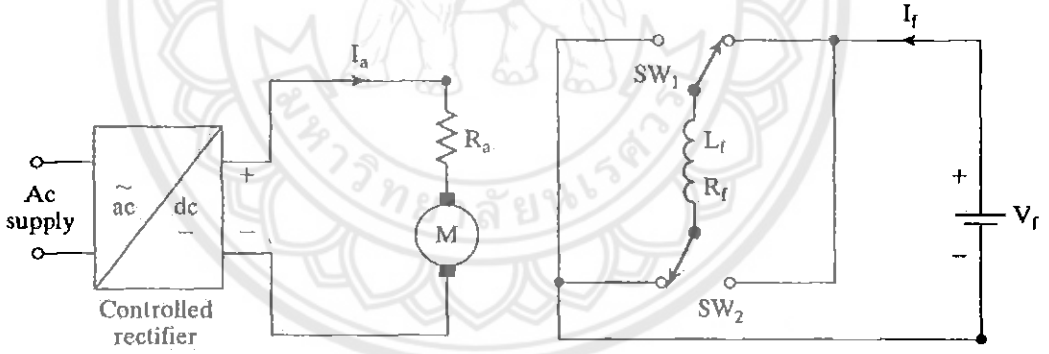
$$\omega = \frac{V_a - R_a I_a}{k \cdot I_f} \tag{2.5}$$

2.1.4 การกลับทิศทางการหมุนของมอเตอร์

การกลับทิศการหมุนของมอเตอร์กระแสตรงสามารถทำได้โดยการกลับขั้วของขดลวดอาร์เมเจอร์ดังรูปที่ 2.7 หรือในการกลับขั้วขดลวดสนามดังรูปที่ 2.8 ซึ่งควรทำในขณะที่กระแสอาร์เมเจอร์มีค่าเป็นศูนย์เพื่อเลี่ยงการกระชากของแรงดันเหนี่ยวนำ (Inductive voltage surge)



รูปที่ 2.7 แผนภาพวงจรการกลับขั้วขดลวดอาร์เมเจอร์ [4]

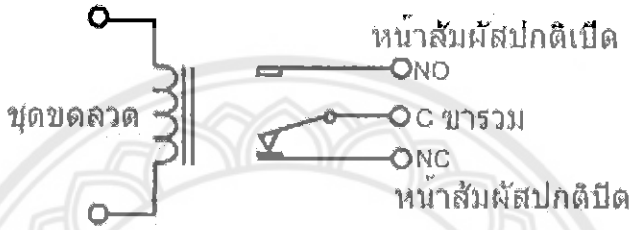


รูปที่ 2.8 แผนภาพวงจรการกลับขั้วขดลวดสนาม [5]

โดยปกติมุมยิง (Firing angle) ของตัวเรียงกระแสแบบควบคุมเฟส (Controlled rectifier) จะถูกปรับเพื่อให้เกิดจังหวะที่กระแสมีค่าเป็นศูนย์และมีช่วงเวลาไร้ผลตอบสนอง (Dead time) ประมาณ 2-10 ms เพื่อให้แน่ใจว่ากระแสอาร์เมเจอร์มีค่าเท่ากับศูนย์ เนื่องจากขดลวดสนามมีค่าคงตัวทางเวลาค่อนข้างมาก การกลับขั้วขดลวดสนามจึงใช้เวลานานกว่า โดยในการทำงานสามารถใช้รีเลย์เป็นสวิตช์ในการกลับขั้วของขดลวดอาร์เมเจอร์เพื่อกลับทิศทางการหมุนของมอเตอร์กระแสตรงได้

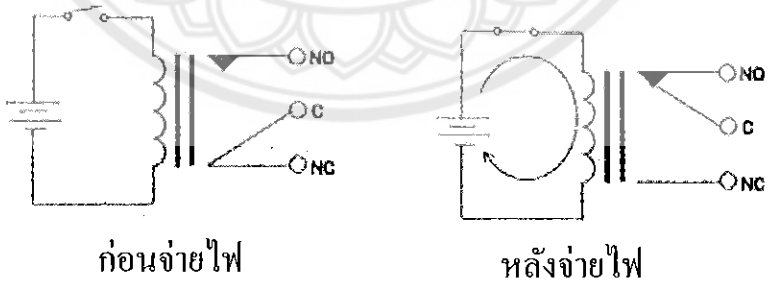
2.2 รีเลย์

รีเลย์ (Relay) มีทำหน้าที่ตัดต่อวงจร เช่นเดียวกับสวิตช์ รีเลย์มีหลายชนิดและหลายขนาด ขึ้นอยู่กับการนำไปใช้งาน เช่น รีเลย์ขนาดเล็ก ใช้ในวงจรอิเล็กทรอนิกส์ รีเลย์ขนาดใหญ่ ใช้ในระบบไฟฟ้ากำลัง เป็นต้น โครงสร้างภายในของรีเลย์โดยทั่วไปประกอบด้วยขดลวด 0 ชุดและหน้าสัมผัส ซึ่งในหน้าสัมผัส 0 ชุด ประกอบไปด้วยหน้าสัมผัสแบบปกติปิด (Normally close หรือ NC) และหน้าสัมผัสแบบปกติเปิด (Normally open หรือ NO) ในสภาวะปกติ หน้าสัมผัสแบบปกติปิดเชื่อมต่ออยู่กับขาร่วม (C) ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 สัญลักษณ์ภายในโครงสร้างของรีเลย์ [5]

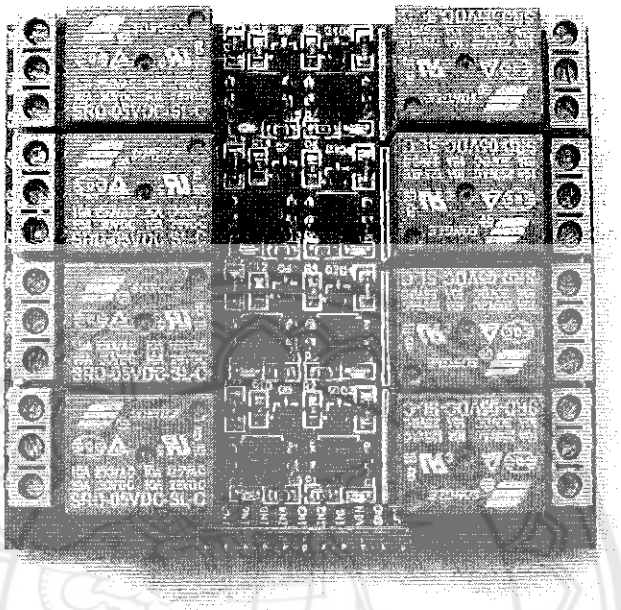
ขณะที่ยังไม่มีการจ่ายกระแสให้ขดลวดของรีเลย์ หน้าสัมผัส NC กับ C ยังต่อถึงกัน ทำให้กระแสไฟฟ้าไหลผ่านไปได้ เมื่อจ่ายกระแสไฟฟ้าให้ขดลวดของรีเลย์ อำนาจแม่เหล็กที่เกิดขึ้นจะดึงหน้าสัมผัส C มาต่อกับหน้าสัมผัส NO ทำให้กระแสไฟฟ้าไหลจาก NO ไปยัง C ได้ และเมื่อกระแสในขดลวดหยุดไหล หน้าสัมผัส C ถูกสปริงดึงกลับไปติดกับหน้าสัมผัส NC ดังเดิม [5] ดังรูปที่ 2.10



รูปที่ 2.10 สภาวะการทำงานของรีเลย์ [6]

ในโครงงานนี้ได้เลือกใช้แผงวงจรรีเลย์แบบมีตัวเชื่อมต่อทางแสง (Opto-isolator) สำหรับตัดต่อวงจรในโครงงานประกอบด้วยรีเลย์ 8 ตัวดังรูปที่ 2.11 ซึ่งแต่ละตัวสามารถรองรับกระแสได้สูงถึง 10 A และใช้งานได้ทั้งกับไฟฟ้ากระแสตรงและกระแสสลับ ซึ่งรับแรงดันกระแสตรง 5 V

และมีหลอดแอลอีดีแสดงสถานะการทำงานของรีเลย์แต่ละตัว ซึ่งมีการป้องกันวงจรควบคุมออกจากวงจรกำลังด้วยตัวเชื่อมต่อด้วยแสงเพื่อป้องกันสัญญาณรบกวนจากการทำงานของอุปกรณ์



รูปที่ 2.11 แผงวงจรรีเลย์แบบมีตัวเชื่อมต่อทางแสง

ในการใช้งานที่แผงวงจรของรีเลย์จะมีตัวเชื่อมสำหรับเชื่อมต่อไฟเลี้ยงวงจรของวงจรควบคุม (VCC) เข้ากับไฟเลี้ยงชุดรีเลย์ (JD - VCC) ซึ่งหากอุปกรณ์ที่ไปควบคุมโหลดต่าง ๆ นั้นไม่ได้สร้างสัญญาณรบกวนมากนัก สามารถใช้งานโมดูลนี้โดยตรงได้ทันทีด้วยการป้อน VCC IN1 IN2 และ GND จากวงจรควบคุมได้ทันที อย่างไรก็ตามจากการใช้ VCC ของวงจรควบคุม ป้อนให้กับ JD - VCC ทำให้ทั้งระบบยังคงต้องใช้กราวด์อ้างอิงร่วมกัน ซึ่งหากใช้งานเพื่อควบคุมไฟสูง กระแสสูง หรืออุปกรณ์ประเภทขดลวดแม่เหล็กไฟฟ้าซึ่งเกิดการรบกวนได้ง่ายควรทำการแยกไฟเลี้ยงรีเลย์ออกจากไฟเลี้ยงวงจรด้วยการปลดตัวเชื่อมต่อไฟเลี้ยง ดังกล่าว แล้วทำการจ่ายไฟเลี้ยงที่เป็นอิสระต่อวงจรควบคุมเข้าสู่ขา JD-VCC และ GND แทน โดยในการควบคุมของภาคควบคุมจะป้อนสัญญาณควบคุมผ่านขา IN1-IN8 และขา VCC โดยไม่ต้องเชื่อมต่อ GND ของฝั่งควบคุม

ในการใช้งานรีเลย์โดยทั่วไปคำนึงถึงปัจจัยต่อไปนี้

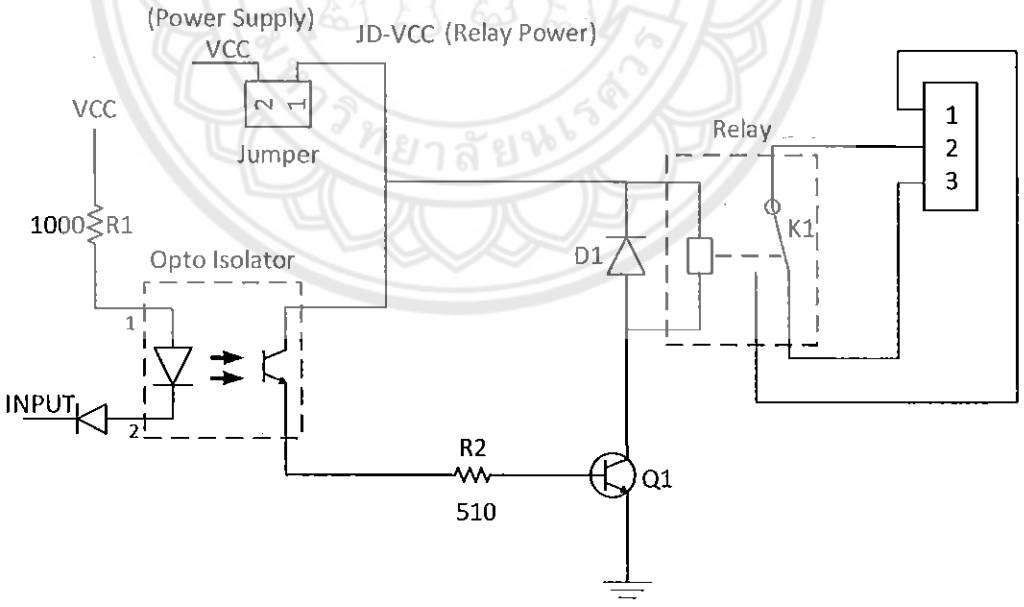
1) แรงดันใช้งานหรือแรงดันที่ทำให้รีเลย์ทำงานได้ สามารถดูได้ที่ตัวรีเลย์ ซึ่งจะระบุค่าแรงดันใช้งานไว้ เช่น 12 VDC หมายถึงต้องใช้แรงดันที่ 12 VDC เท่านั้น ซึ่งหากใช้มากกว่านี้จะส่งผลให้ขดลวดภายในตัวรีเลย์อาจขาดได้หรือหากใช้แรงดันต่ำกว่ามาก ๆ อาจทำให้รีเลย์ไม่ทำงาน ในส่วนของการต่อวงจรนั้นสามารถต่อเข้ากับขั้วใดก็ได้ เพราะรีเลย์จะไม่ระบุขั้วต่อไว้ (นอกจากรีเลย์ชนิดพิเศษ)

2) การใช้งานกระแสผ่านหน้าสัมผัส จะมีการระบุค่าไว้ที่ตัวรีเลย์ เช่น 10 A 220 VAC หมายถึง หน้าสัมผัสของรีเลย์นั้นสามารถทนกระแสได้ 10 A ที่ 220 VAC แต่ในการใช้งานจริง ควรให้รีเลย์ทำงานต่ำกว่าระดับกระแสพิกัด เนื่องจากหากมีกระแสมากจะยิ่งมีผลทำให้หน้าสัมผัสของรีเลย์นั้นเสียหายได้เร็วขึ้น

3) ชนิดและจำนวนหน้าสัมผัสภายในตัวรีเลย์

โครงสร้างภายในและการจ่ายไฟเพื่อให้รีเลย์ทำงาน

ในส่วนของโครงสร้างการทำงานภายในของรีเลย์ สามารถแสดงลักษณะและเส้นทางในการจ่ายไฟของรีเลย์ แสดงได้ดังรูปที่ 2.12



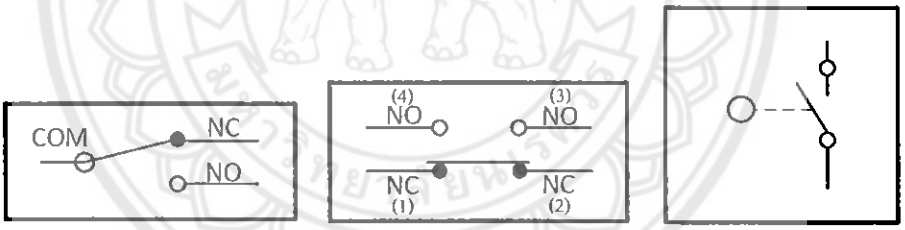
รูปที่ 2.12 แผนภาพวงจรการทำงานภายในของรีเลย์

ที่มา: https://arduino-info.wikispaces.com/RelayIsolation

การทำงานของรีเลย์ เริ่มด้วยการจ่ายไฟเลี้ยงเข้าที่ตัวเชื่อมต่อทางแสง (Opto-isolator) และขดลวดของรีเลย์ การป้อนอินพุตด้วยลอจิก 0 ทำให้วงจรภายในตัวเชื่อมต่อทางแสงเริ่มทำงาน นั่นคือมีความต่างศักย์ตกคร่อมไดโอดเปล่งแสง ทำให้เกิดการไบแอสไปหน้า ส่งผลทำให้ไดโอดเกิดการเปล่งแสงไปตกกระทบที่ขาเบสของทรานซิสเตอร์ที่อยู่ภายในตัวเชื่อมต่อทางแสง ทรานซิสเตอร์จึงนำกระแส เกิดกระแสไหลผ่านตัวต้านทาน R2 เข้าที่ขาเบสของทรานซิสเตอร์ Q1 ส่งผลให้เกิดการนำกระแส จึงเกิดความต่างศักย์ที่ขดลวดของรีเลย์ ทำให้มีกระแสไหลผ่านขดลวด ส่งผลให้รีเลย์มีการเปลี่ยนสถานะของหน้าสัมผัส

2.3 สวิตช์จำกัดระยะ

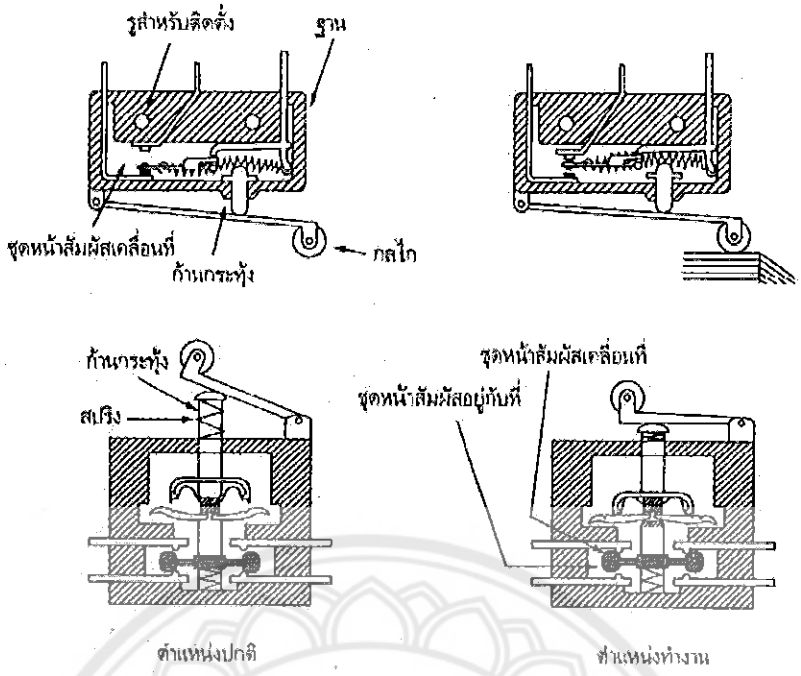
การทำงานของสวิตช์จำกัดระยะ (Limit switch) อาศัยการชนของวัตถุกับลูกล้อ (Roller) แล้วส่งผลให้หน้าสัมผัสที่ต่ออยู่กับก้านชนเปิดหรือปิดตามจังหวะของการชน จะเห็นว่าการทำงานดังกล่าวอาศัยแรงกดจากภายนอกมากระทำ เช่น วางของทับที่ปุ่มกดหรือลูกเบี้ยวมาชนที่ปุ่มกด และเป็นผลทำให้หน้าสัมผัสที่ต่ออยู่กับก้านชน เปิดและปิดตามจังหวะของการชนสัญลักษณ์ของสวิตช์จำกัดระยะแสดงดังรูปที่ 2.14 สวิตช์จำกัดระยะมีการนำไปใช้ประโยชน์ได้อย่างมากมาย เช่น ลิฟต์โดยสาร ลิฟต์ขนของ ประตูที่ทำงานด้วยไฟฟ้า ระบบสายพานลำเลียง เป็นต้น [7, 8]



รูปที่ 2.13 สัญลักษณ์สวิตช์จำกัดระยะ

ที่มา: http://fonengineering.com/our-products/19-limit-switch.html

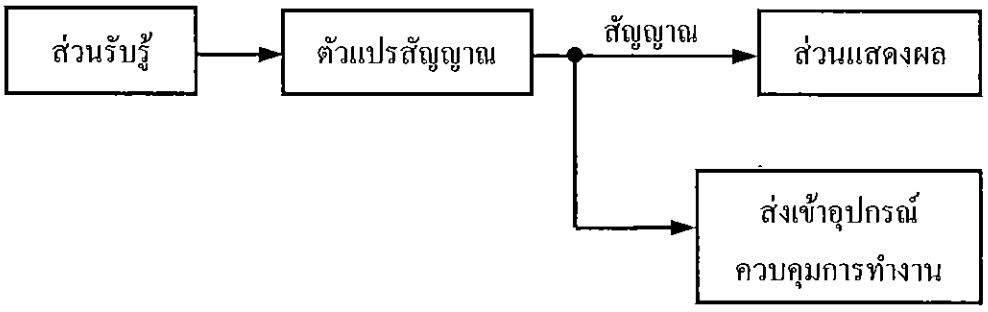
สวิตช์จำกัดระยะเปรียบได้กับสวิตช์ตัดวงจร ทำหน้าที่หลักในการหยุดการเคลื่อนที่ของมอเตอร์ในแนวแกนต่าง ของเครื่องจักรที่ถูกออกแบบมา ในกรณีที่เป็นเครื่องจักรทำงานแบบอัตโนมัติ เมื่อมีวัตถุเคลื่อนที่ไปแตะกับสวิตช์จำกัดระยะ ทำให้หน้าสัมผัสแยกออกจากกันดังแสดงในรูปที่ 2.14 วงจรจะไม่จ่ายกระแสไฟฟ้าไปที่มอเตอร์ขับเคลื่อน [9]



รูปที่ 2.14 กลไกลการทำงานภายในของสวิตช์จำกัดระยะ [9]

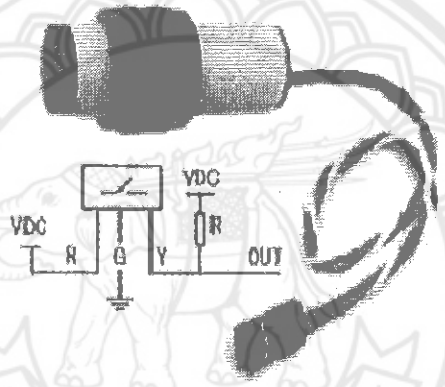
2.4 ตัวรับรู้แบบใช้แสง

ตัวรับรู้ (Sensor) ประกอบด้วยส่วนรับรู้ (Sensing part) ซึ่งทำหน้าที่ตรวจจับปริมาณของตัวแปรต่างๆที่ต้องการทราบค่า เช่น อุณหภูมิ การเคลื่อนที่ แสงสว่าง เป็นต้น แต่ส่วนรับรู้เพียงอย่างเดียวไม่สามารถบอกค่าที่ต้องการวัดได้ จึงจำเป็นต้องมีส่วนแปลงพลังงาน (Transducing part) ซึ่งทำหน้าที่แปลงสัญญาณที่ได้จากการตรวจจับมาเป็นปริมาณที่สามารถเข้าใจได้เรียกว่า ตัวแปรสัญญาณ (Signal converter) โดยทำหน้าที่แปลงพลังงานจากรูปหนึ่งให้อยู่ในอีกรูปแบบหนึ่ง เช่น แปลงพลังงานแสงเป็นพลังงานไฟฟ้า ในการนำตัวรับรู้ไปใช้งานแสดงได้ดังแผนภาพในรูปที่ 2.15 ส่วนรับรู้ทำการตรวจวัดและให้ตัวแปรสัญญาณเอาท์พุตเป็นสัญญาณไฟฟ้าซึ่งถูกส่งไปเข้ากระบวนการทางไฟฟ้าขั้นต่อไป เช่น การขยายสัญญาณ แล้วจึงได้เอาท์พุตออกมาแสดงผลหรือนำไปใช้งานในด้านอื่นๆตามต้องการ [10]



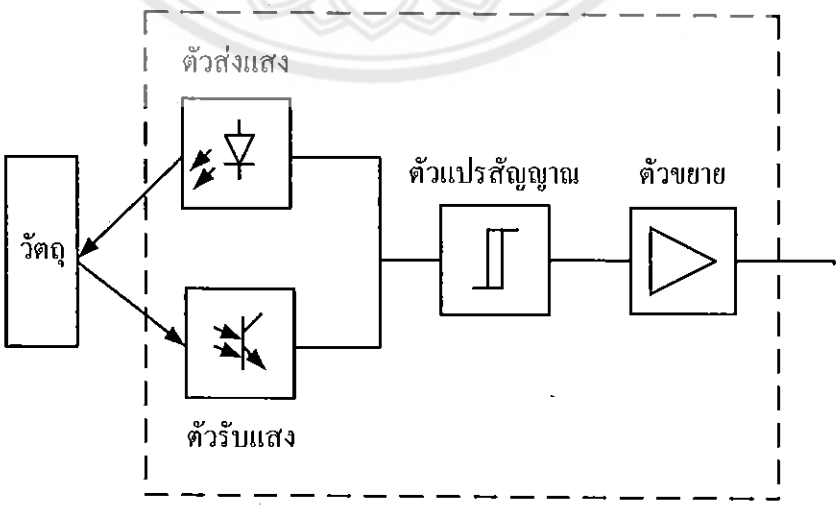
รูปที่ 2.15 แผนภาพการใช้งานตัวรับรู้ [10]

ในโครงงานนี้ตัวรับรู้แบบใช้แสง (Photoelectric sensor) ซึ่งต้องการไฟเลี้ยงเป็นไฟกระแสตรงขนาด 5 VDC ระยะการตรวจจับ 3-80 cm ขนาดยาว 4.5 cm เส้นผ่านศูนย์กลาง 2 cm ดังแสดงในรูปที่ 2.16 โดยมีตัวส่งแสงและตัวรับแสงอยู่ภายใน มีการตอบสนองต่อการเปลี่ยนแปลงปริมาณของแสงที่มากระทบกับตัวรับแสง และส่งสัญญาณเอาท์พุตซึ่งสัมพันธ์กับการเปลี่ยนแปลงปริมาณแสงที่ได้รับผ่านตัวรับแสง ตัวรับรู้ชนิดนี้สามารถตรวจจับการปรากฏขึ้นหรือการหายไปของวัตถุ สามารถตรวจจับขนาด รูปร่าง การสะท้อนแสง และความโปร่งแสงหรือสีของวัตถุ โดยมีส่วนประกอบหลักแสดงดังรูปที่ 2.17 ในทางปฏิบัติสามารถสร้างให้ตัวรับรู้แบบใช้แสงสามารถตรวจจับได้ในระยะไกลถึง 100 m หรือตรวจจับวัตถุขนาดเล็ก (เช่น เล็กกว่า 1 mm) ได้ [11] โดยในโครงงานนี้ใช้ตัวรับรู้ดังกล่าวในการตรวจสิ่งของ ซึ่งสามารถตรวจจับวัตถุได้แม่นยำโดยไม่ต้องสัมผัสกับสิ่งของ



รูปที่ 2.16 ตัวรับรู้แบบใช้แสง ระยะตรวจจับ 3-80 cm

ที่มา: http://www.arduino.in.th

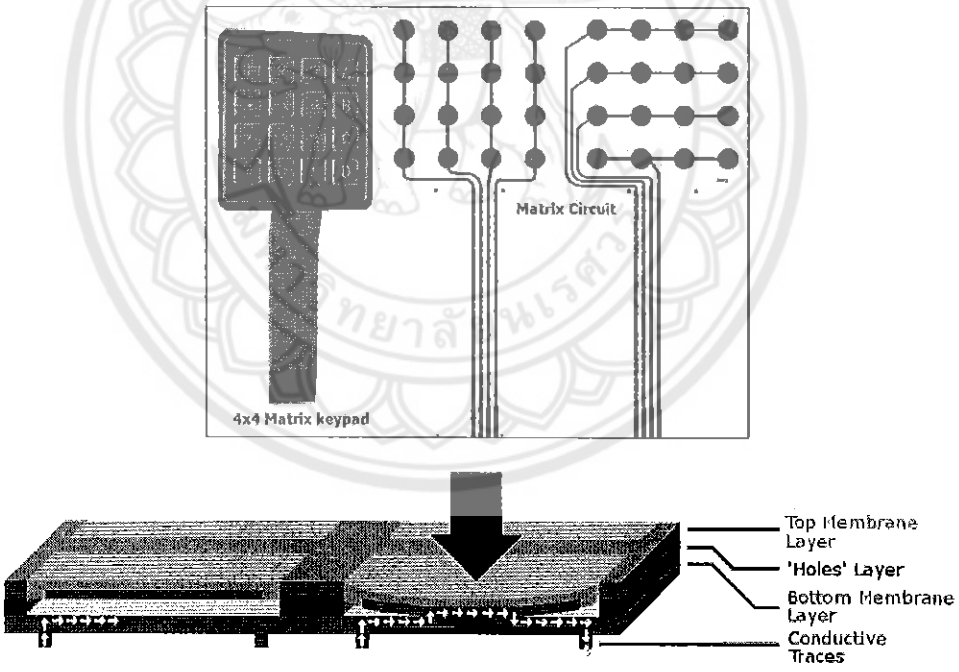


รูปที่ 2.17 ส่วนประกอบหลักของตัวรับรู้แบบใช้แสง [10]

ตัวรับรู้แบบใช้แสงสามารถตรวจจับวัตถุได้ทุกประเภท มีอายุการใช้งานได้ยาวนาน ระยะในการตรวจจับไกลที่สุดในบรรดาตัวรับรู้ชนิดอื่น เวลาในการตอบสนองดีที่สุดจึงเหมาะที่จะใช้ตรวจจับประเภทที่มีความถี่ในการตรวจจับสูง เช่น ใช้ในการวัดความเร็วในการเคลื่อนที่ทั้งเชิงเส้นและเชิงมุม อย่างไรก็ตามตัวรับรู้ชนิดนี้มีข้อจำกัดในการตรวจจับวัตถุโปร่งใสและวัตถุที่มีสีแตกต่างกันมากเนื่องจากความสามารถในการสะท้อนหรือดูดกลืนแสงในแต่ละสีแตกต่างกัน นอกจากนี้ยังต้องระวังเรื่องความสะอาดของเลนส์ของตัวรับรู้ [10]

2.5 แผงแป้นตัวเลข

แผงแป้นตัวเลข แบบ 4 x 4 ชนิดนี้ประกอบไปด้วยปุ่ม 16 ปุ่ม ที่เรียงต่อกันเป็นเมตริกซ์แบบ 4 แถว และ 4 หลัก ประกอบไปด้วย เลข 0 – 9 และตัวอักษร A – D และ * และ # ซึ่งปุ่มแต่ละปุ่ม เป็นการกดเพื่อให้หน้าสัมผัสที่เป็นชั้นสีแดงดังรูปที่ 2.18 ไปแตะกันทำให้เป็นการเชื่อมต่อกันทางไฟฟ้าไปอีกด้านหนึ่งของสวิตช์



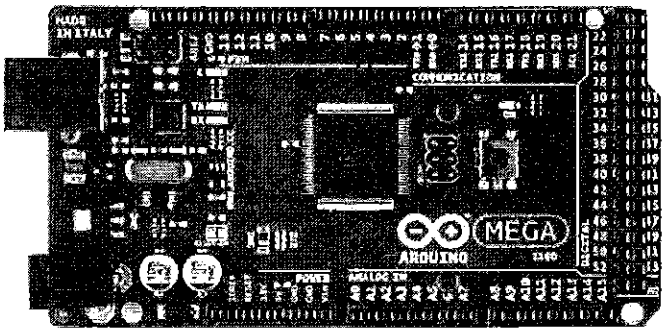
รูปที่ 2.18 แผงแป้นตัวเลข แบบ 4 x 4 [11]

ในการตรวจสอบแป้นตัวเลขการกดปุ่มของผู้ใช้ในขณะนั้น ใช้วิธีการตรวจสอบไปทีละหลัก จนครบทุกหลัก แล้วนำมาตีความว่ามีการตอบสนองออกมาเป็นแบบใดบ้าง เช่น ถ้ามีการกดเลข 1 อยู่ในขณะที่เราจ่ายแรงดัน 5 V ไปที่หลักที่ 1 จะมีเพียงแถวแรกเท่านั้นที่จะอ่านค่าแรงดันได้สูง นอกนั้นจะเป็นแรงดันต่ำ หรือถ้ามีการกดปุ่ม # อยู่ ขณะที่ตรวจสอบไปแต่ละหลักนั้นจะไม่เจอ

แรงดันสูงที่แถวใดเลยจนกว่าจะตรวจสอบไปถึงหลักที่ 3 ซึ่งจะพบว่ามีการตอบสนองกลับมาจากแถวที่ 4 นั้นเอง ดังนั้นเมื่อพบว่าเป็นการตรวจสอบหลักที่ 3 และมีแถวที่ 4 ตอบสนอง ก็คือปุ่ม # นั่นเอง [11]

2.6 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ที่ใช้ในโครงงานนี้เป็นแผงวงจร Arduino จัดอยู่ในตระกูล AVR ขนาด 100 ขา ซึ่งใช้ไมโครคอนโทรลเลอร์หมายเลข ATmega2560 แสดงดังรูปที่ 2.19 เป็นแผงวงจรArduino ที่ออกแบบมาสำหรับงานที่ต้องใช้อินพุตและเอาท์พุตมากกว่า Arduino รุ่นอื่น ๆ เช่น งานที่ต้องรับสัญญาณจากตัวรับรู้ หรือควบคุมมอเตอร์เซอร์โวหลาย ๆ ตัว โดยไมโครคอนโทรลเลอร์ Arduino เป็นแพลตฟอร์ม (Platform) ของอินพุตและเอาท์พุต (I/O) ที่เพียงพอกับการใช้งานและการเรียนรู้ และมีการพัฒนาแบบ Open source คือมีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ แผงวงจรถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นใช้งาน ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวแผงวงจร หรือโปรแกรมต่อได้ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกเข้ามาที่ขา I/O ของแผงวงจร หรือเพื่อความสะดวกสามารถเลือกต่อกับแผงวงจรเสริม (Shield) ประเภทต่าง ๆ เช่น X Bee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาเสียบกับแผงวงจร Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้โดยตัวแผงวงจรมีคำสั่งที่ใช้ควบคุมพอร์ต อินพุตและเอาท์พุต ไม่ว่าจะเป็นพอร์ตดิจิตอล พอร์ตแอนะล็อกพีดับเบิลยูเอ็มและพอร์ตอนุกรมซึ่งแผงวงจร Arduino ทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าการใช้เครื่องคอมพิวเตอร์ ตัวแผงวงจรออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยวและมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้แผงวงจร Arduino สามารถรับสัญญาณจากสวิตช์หรือตัวรับรู้และควบคุมหลอดไฟมอเตอร์หรืออุปกรณ์อื่นๆแผงวงจร Arduino สามารถทำงานอิสระหรือทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ [12]

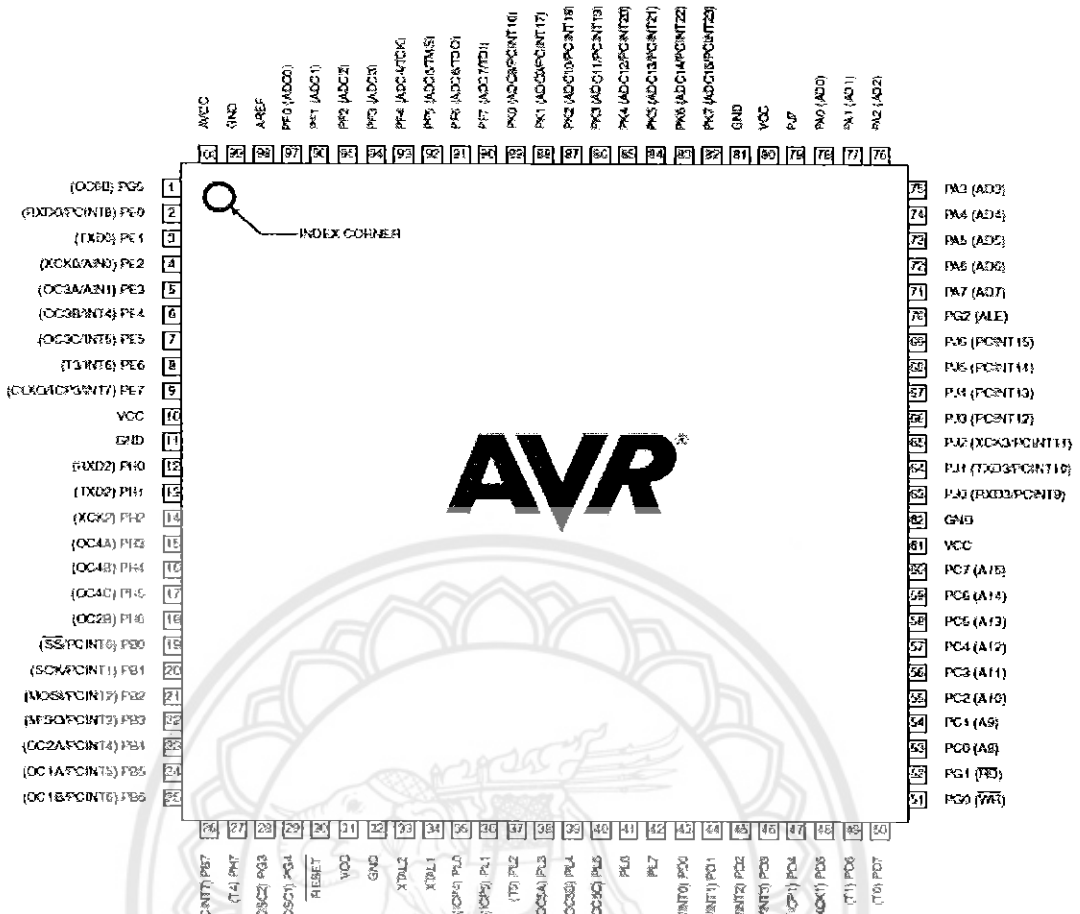


รูปที่ 2.19 แผงวงจร Arduino Mega2560

ที่มา: www.arduitronics.com

แผงวงจร Arduino ซึ่งมีจุดเด่นในเรื่องของความง่ายต่อการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่าง ๆ เพื่อสนับสนุนการใช้งานด้วยรูปแบบที่ไม่ซับซ้อนคือเน้นการโปรแกรมไมโครคอนโทรลเลอร์เป็นหลักแผงวงจร Arduino เป็นไมโครคอนโทรลเลอร์โดยใช้ AVR ขนาดเล็กซึ่งเป็นตัวประมวลผลและสั่งงานเหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์ และนำไปประยุกต์เพื่อใช้งานเกี่ยวกับการควบคุมอุปกรณ์ อินพุตและเอาท์พุตต่าง ๆ ได้มากมายทั้งในแบบที่เป็นการทำงานตัวเดียวอิสระ หรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่นเช่น คอมพิวเตอร์ทั้งนี้เนื่องมาจากว่า Arduino สนับสนุนการเชื่อมต่อกับอุปกรณ์อินพุตและเอาท์พุตได้มากมาย ทั้งแบบดิจิตอล (Digital) และแอนะล็อก (Analog) เช่น การรับค่าจากสวิตช์หรือตัวรับรู้แบบต่าง ๆ รวมไปถึงการควบคุมอุปกรณ์เอาท์พุตต่าง ๆ ส่วนภาษาในการเขียนโปรแกรมลงบนแผงวงจร Arduino นั้นจะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ทีมโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) แต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมและยังสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C โดยตรง

ตัวแผงวงจร Arduino ที่ใช้ในโครงงานนี้จะกล่าวถึงสถาปัตยกรรมของเอวีอาร์ (AVR) ขนาด 8 bits โดยเป็นหน่วยประมวลผลกลาง (Central processing unit, CPU) แบบ RISC (Reduced Instruction Set Computer) มีสถาปัตยกรรมการต่อหน่วยความจำแบบฮาวาร์ด (Harvard) ซึ่งแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกันโดยเด็ดขาดดังแสดงในรูปที่ 2.20 โดยใช้หน่วยความจำแบบแฟลช (Flash) สำหรับเป็นหน่วยความจำโปรแกรม ซึ่งมีความจุมากกว่ารุ่น Arduino Uno R3 ทำให้สามารถเขียนโปรแกรมเข้าไปได้มากกว่าในความเร็วของ MCU ที่เท่ากัน และใช้หน่วยความจำแบบ SRAM สำหรับหน่วยความจำข้อมูลและนอกจากนี้ยังมีหน่วยความจำแบบ EEPROM ซึ่งสามารถเก็บข้อมูลเอาไว้ได้โดยไม่จำเป็นต้องมีไฟเลี้ยงอีกด้วย



รูปที่ 2.20 หน่วยประมวลผลกลางของ ATmega2560 ขนาด 100 ขา

ที่มา: www.atmel.com

จากรูปที่ 2.20 แผงวงจรไมโครคอนโทรลเลอร์ ATmega2560 มีคุณสมบัติเด่น [12] ดังนี้

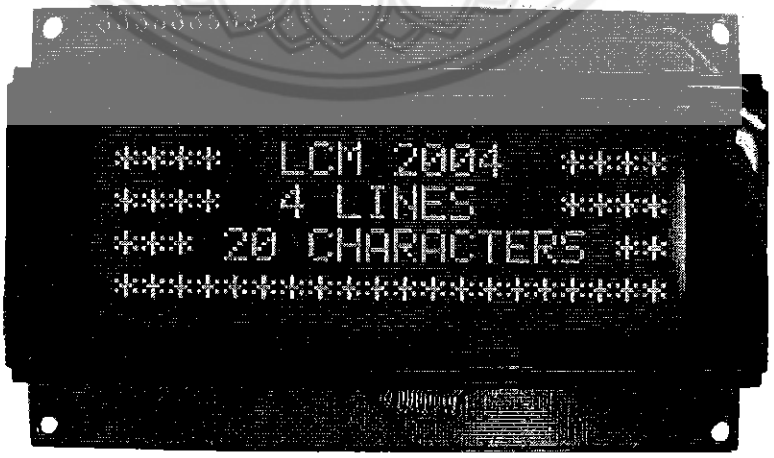
1) ทำงานได้ตั้งแต่ย่านแรงดัน 1.8-5.5 V แรงดันของระบบอยู่ที่ 5 V
2) หน่วยความจำข้อมูลแบบ SRAM ขนาด 8 kb
3) หน่วยความจำข้อมูลแบบ EEPROM ขนาด 4 kb
4) สนับสนุนการเชื่อมต่อแบบ I2C bus
5) พอร์ตอินพุตเอาท์พุตแบบดิจิตอลจำนวน 54 ช่อง
6) พอร์ตเอาท์พุตแบบแอนะล็อกจำนวน 16 ช่อง
7) วงจรสื่อสารอนุกรม
8) สนับสนุนช่องสัญญาณสำหรับสร้าง สัญญาณพีดับเบิลยูเอ็ม (PWM) จำนวน 14 ช่อง

ไมโครคอนโทรลเลอร์ ATmega2560 ที่ใช้ในโครงงานนี้ทำหน้าที่เป็นส่วนประมวลผลสัญญาณที่รับมาจากตัวรับรู้และทำงานตามโปรแกรมที่เขียนไว้ เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ผ่านทางมอเตอร์

2.7 จอแสดงผลแอลซีดี

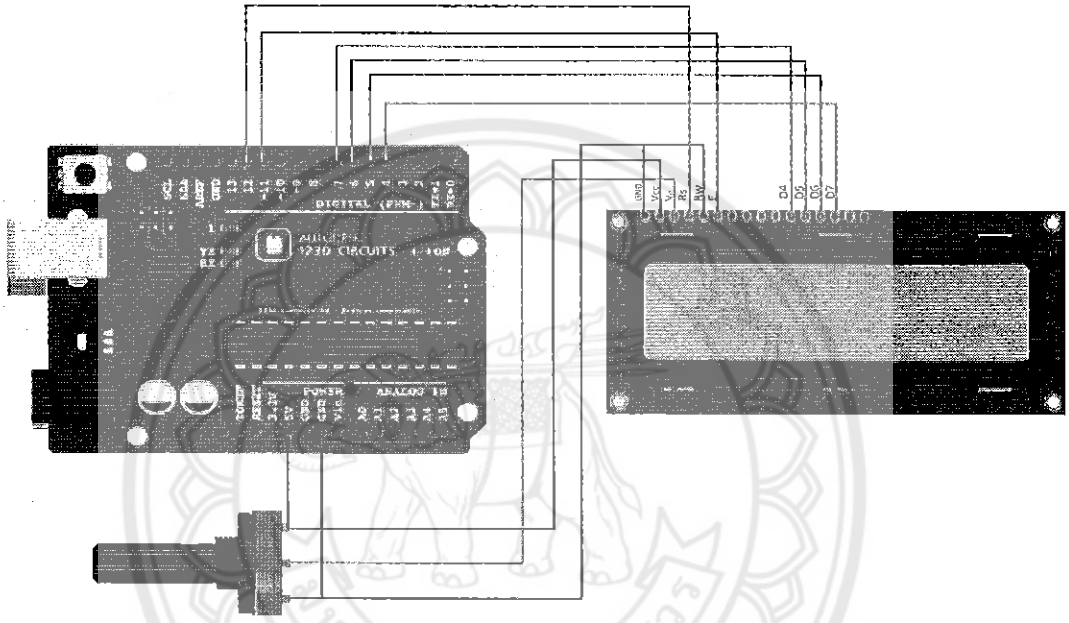
ในการควบคุมหรือสั่งงานผ่านจอแสดงผลแอลซีดี (Liquid Crystal Display, LCD) นั้นมีตัวควบคุม (Controller) รวมไว้ในตัว ซึ่งสามารถส่งรหัสคำสั่งควบคุมการทำงานของจอแสดงผลแอลซีดีผ่านไมโครคอนโทรลเลอร์ (Microcontroller) โดยในส่วนของการควบคุมจอแสดงผลแอลซีดีเป็น Hitachi หมายเลข HD44780 ดังแสดงในรูปที่ 2.21 และขาในการเชื่อมต่อระหว่างจอแสดงผลแอลซีดีกับไมโครคอนโทรลเลอร์มีดังนี้

1) GND เป็นกราวด์ใช้ต่อระหว่างกราวด์ของระบบไมโครคอนโทรลเลอร์กับแอลซีดี
2) VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับแอลซีดีขนาด +5 VDC
3) VO ใช้ปรับความสว่างของหน้าจอแอลซีดี
4) RS ใช้บอกตัวควบคุมให้ทราบว่ารหัสที่ส่งมาทางขา DB0-DB7 นั้นเป็นคำสั่งหรือข้อมูล
5) R/W ใช้เลือกระหว่างการอ่านหรือเขียนข้อมูลกับตัวควบคุม
6) E เป็นขา Enable หรือ Chips Select เพื่อกำหนดการทำงานของตัวควบคุม
7) DB0-DB7 เป็นขาสัญญาณข้อมูล (Data) ใช้สำหรับเขียนหรืออ่านข้อมูลและคำสั่งกับตัวควบคุม [13]



รูปที่ 2.21 จอแสดงผลแอลซีดี Hitachi หมายเลข HD44780 [13]

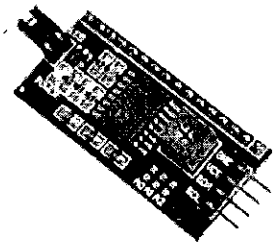
การเชื่อมต่อสัญญาณขาข้อมูลระหว่างไมโครคอนโทรลเลอร์กับจอแอลซีดี สามารถทำได้ 2 ลักษณะ คือ การเชื่อมต่อแบบ 8 bit (DB0-DB7) และการเชื่อมต่อแบบ 4 bit (DB4-DB7) ซึ่งทั้งสองแบบแตกต่างกันเพียงจำนวนขาที่ใช้คือ 8 หรือ 4 ขา โดยสามารถทำงานได้เหมือนกัน อย่างไรก็ตามการส่งข้อมูลแบบ 4 ขา ช้ากว่าแบบ 8 ขา แต่ไม่ได้ช้ามากจนสังเกตได้ด้วยสายตา ดังนั้นโดยทั่วไปการต่อกับ Arduino จึงนิยมต่อเพียง 4 ขา หรือ 4 bit เท่านั้น ดังแสดงในรูปที่ 2.22 เพื่อประหยัดขาในการต่อใช้งานไปไว้ต่อกับอุปกรณ์อื่น [14]



รูปที่ 2.22 การเชื่อมต่อไมโครคอนโทรลเลอร์กับจอแอลซีดี [14]

2.8 ไอทูซี

จอแอลซีดีที่มีการเชื่อมต่อแบบ I2C หรือเรียกอีกอย่างว่าการเชื่อมต่อแบบอนุกรม ซึ่งเป็นจอแอลซีดีธรรมดาทั่วไปที่ติดตั้งกับแผงวงจร I2C Bus ทำให้มีการใช้งานได้ที่สะดวกยิ่งขึ้น และมีตัวต้านทานปรับค่าได้เพื่อสำหรับปรับความเข้มของจอ การเชื่อมต่อระหว่าง I2C กับไมโครคอนโทรลเลอร์ใช้จำนวน 4 ขา (แบบขนานใช้ 16 ขา) แสดงดังรูปที่ 2.23



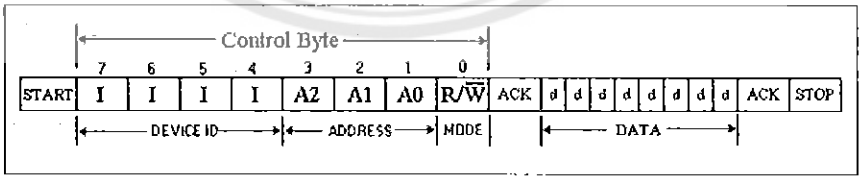
รูปที่ 2.23 ไอทูซี [15]

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอแอลซีดีจะมีส่วนควบคุม (Controller) อยู่ในตัวแล้ว สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอแอลซีดี (I2C) เช่นเดียวกันกับจอแอลซีดีแบบธรรมดาคือรหัสคำสั่งที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล สำหรับการส่งข้อมูลรูปแบบ I2C ที่ใช้ขาเพียง 4 ขาที่ใช้ในการเชื่อมต่อเท่านั้น แสดงดังตารางที่ 2.1

ตารางที่ 2.1 สัญลักษณ์ขาของไอทูซี [15]

ขาเชื่อมต่อ	สัญลักษณ์	คำอธิบาย
1	GND	กราวด์ของระบบไมโครคอนโทรลเลอร์กับจอแอลซีดี
2	VCC	ไฟเลี้ยงวงจรที่ป้อนให้กับจอแอลซีดี มีขนาด +5 V_{DC}
3	SDA	ขาที่ใช้ในการรับส่งข้อมูล
4	SCL	ขาสัญญาณนาฬิกาในการรับส่งข้อมูล

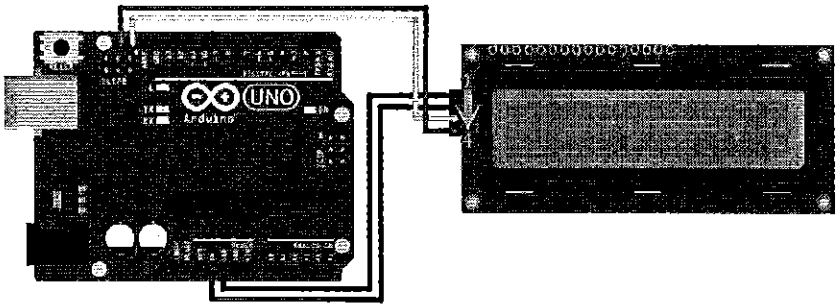
สำหรับการรับ-ส่งข้อมูลแบบ I2C BUS ไมโครคอนโทรลเลอร์จะทำการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัสแล้วตามด้วย รหัสควบคุม (Control Byte) ซึ่งประกอบด้วยรหัสประจำตัวอุปกรณ์ Device ID, Device Address และแบบวิธีในการเขียนหรืออ่านข้อมูล เมื่ออุปกรณ์รับทราบว่าไมโครคอนโทรลเลอร์ต้องการจะติดต่อด้วยต้องส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ไมโครคอนโทรลเลอร์รับรู้ว่าข้อมูลที่ส่งมามีความถูกต้อง และเมื่อสิ้นสุดการส่งข้อมูลไมโครคอนโทรลเลอร์จะต้องส่งสถานะสิ้นสุด เพื่อให้อุปกรณ์ทราบว่าสิ้นสุดการส่งสัญญาณ โดยจะส่งสัญญาณการรับ และส่งข้อมูลแบบ I2C BUS แสดงดังรูปที่ 2.24



รูปที่ 2.24 การรับและส่งข้อมูลแบบ I2C BUS [15]

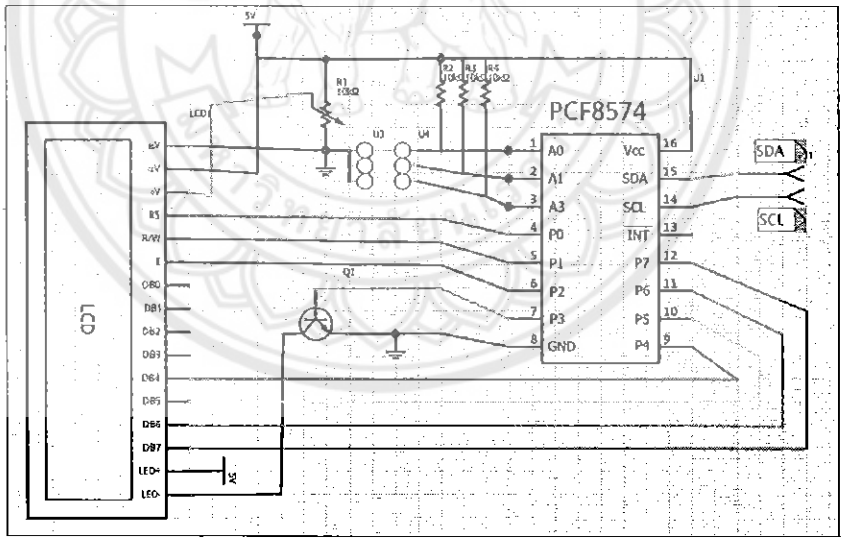
สำหรับการเชื่อมต่อสัญญาณระหว่างไมโครคอนโทรลเลอร์กับจอแอลซีดี ที่ต่อกับแผงวงจร I2C การส่งข้อมูลจากไมโครคอนโทรลเลอร์จะถูกส่งออกมาในรูปแบบ I2C ไปยังแผงวงจร I2C และแผงวงจรจะมีหน้าที่จัดการข้อมูลให้ออกมาในรูปแบบปกติหรือแบบขนาน เพื่อใช้ในการติดต่อไปยังจอแอลซีดี โดยที่รหัสคำสั่งที่ใช้ในการสั่งงานจอแอลซีดี ยังคงไม่ต่างกับจอ

แอลซีดี ที่เป็นแบบขนานโดยส่วนใหญ่แผงวงจร I2C จะเชื่อมต่อกับตัวควบคุมของจอแอลซีดีเพียง 4 บิตเท่านั้นแสดงดังรูปที่ 2.25



รูปที่ 2.25 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ กับจอแอลซีดี [15]

จากรูปที่ 2.26 จะเห็นได้ว่าจอแอลซีดี และแผงวงจร I2C ได้มีการเชื่อมต่อขาสำหรับการรับส่งข้อมูลเป็นแบบ 4 บิต ขาที่เชื่อมต่อไว้คือ ขา P4 > DB4, P5 > DB5, P6 > DB6, P7 > DB7 และขา P2 > E (Enable), P1 > R/W, P0 > RS รวมไปถึงตัวต้านทานสำหรับปรับค่าความเข้มของตัวอักษร และ Switch Blacklight



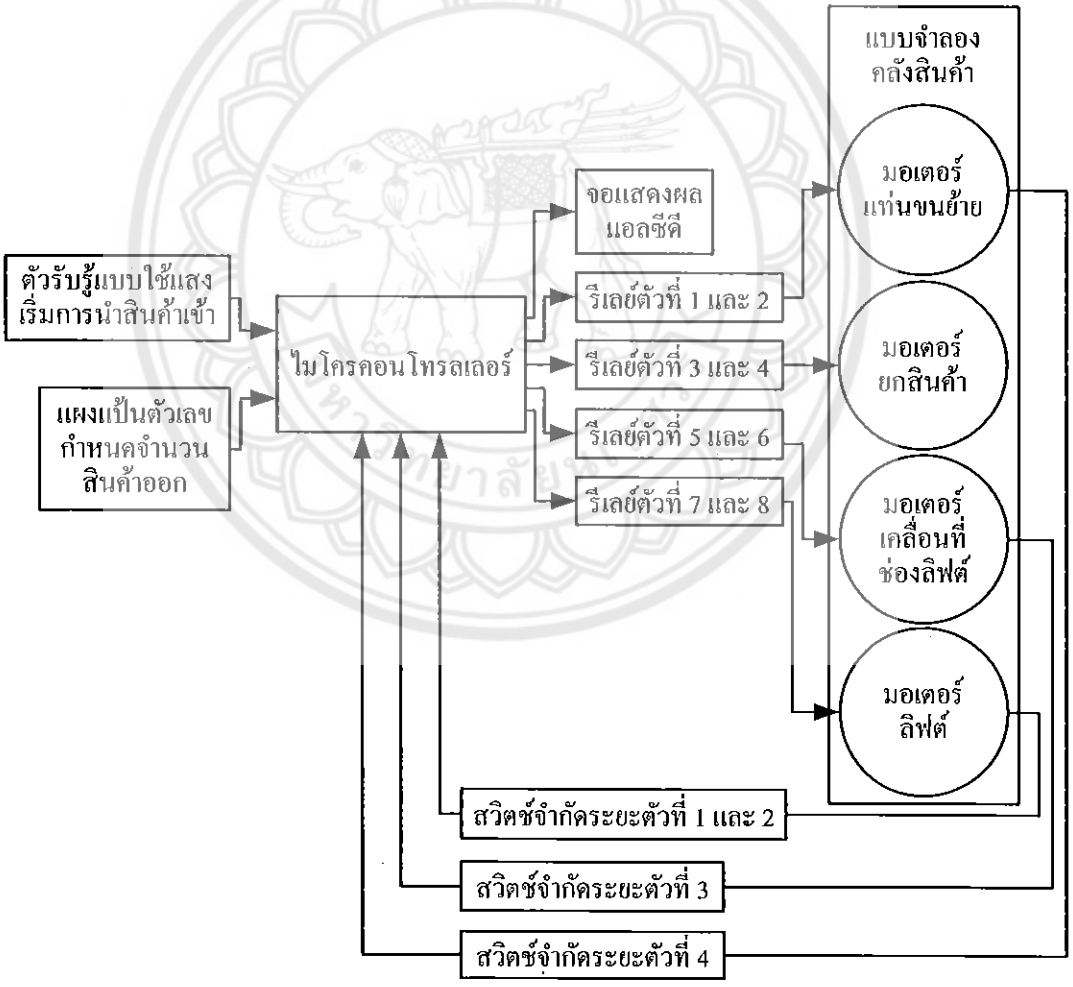
รูปที่ 2.26 การแสดงการเชื่อมต่อของ I2C กับจอแอลซีดี [15]

บทที่ 3
การออกแบบและสร้างแบบจำลองระบบจัดเก็บสินค้าและนำออกจากคลัง

ในบทนี้จะอธิบายถึงการควบคุมการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง การออกแบบกลไกการขับเคลื่อน และการออกแบบโครงสร้างของคลังสินค้า

3.1 ขั้นตอนการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง

ส่วนประกอบและหลักการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลังซึ่งควบคุมด้วยไมโครคอนโทรลเลอร์แสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 ส่วนประกอบและหลักการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง

ว 17 ต.ค. 2560
25
สำนักหอสมุด

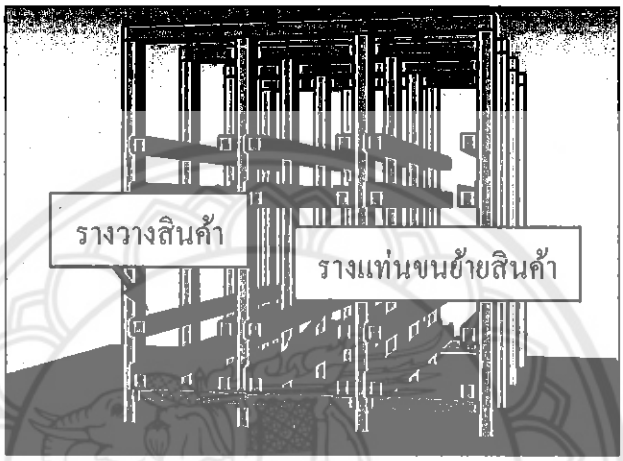
การทำงานของระบบจัดเก็บสินค้าและนำออกจากคลังมีการทำงานหลักแบ่งออกเป็น 2 ระบบคือ ระบบการจัดเก็บสินค้าและระบบการนำสินค้าออกจากคลัง ซึ่งถูกควบคุมด้วยไมโครคอนโทรเลอร์สั่งรีเลย์ตัดต่อวงจรให้มอเตอร์ของชุดการเคลื่อนที่ทำงาน

ระบบการจัดเก็บสินค้าเริ่มทำงานเมื่อตัวรับรู้แสงทำการตรวจสอบพบสินค้ารอการจัดเก็บอยู่จึงส่งสัญญาณไปยังไมโครคอนโทรลเลอร์เพื่อสั่งให้รีเลย์ทำงานส่งผลให้มอเตอร์ของชุดการเคลื่อนที่นำสินค้าเข้าจัดเก็บในคลัง โดยการทำงานเริ่มจากรีเลย์ตัวที่ 3 ต่อวงจรให้มอเตอร์ยกสินค้าขึ้น จากนั้นรีเลย์ตัวที่ 5 ต่อวงจรให้มอเตอร์ช่องลิฟต์เคลื่อนที่ไปหน้าคลังถูกกำหนดโดยสวิตช์จำกัดระยะตัวที่ 3 เมื่อถึงแนวแถวของคลังสินค้าที่ต้องการจัดเก็บ รีเลย์ตัวที่ 7 ทำงานส่งผลให้มอเตอร์ลิฟต์เคลื่อนที่ขึ้น (ถ้ามีการเก็บสินค้าชั้นที่ 2) ถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 1 เมื่อเคลื่อนที่ถึงชั้นที่ 2 รีเลย์ตัวที่ 1 ต่อวงจรให้มอเตอร์แท่นขนย้ายเคลื่อนที่เข้าไปในคลังถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 4 หลังจากนั้นรีเลย์ตัวที่ 4 จึงต่อวงจรให้มอเตอร์วางสินค้าลงตามตำแหน่งในคลัง และไมโครคอนโทรลเลอร์ประมวลผลจำนวนสินค้าคงคลังแล้ว แสดงผลผ่านจอแอลซีดี จากนั้นชุดการเคลื่อนที่กลับมายังจุดเริ่มต้นโดยเริ่มการทำงานจากรีเลย์ตัวที่ 2 ต่อวงจรส่งผลแท่นขนย้ายเคลื่อนที่ออกจากคลังถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 4 แล้ว รีเลย์ตัวที่ 8 ต่อวงจรให้ลิฟต์เคลื่อนที่ลง (ถ้ามีการเก็บสินค้าชั้นที่ 2) สวิตช์จำกัดระยะตัวที่ 2 เป็นตัวกำหนดให้ลิฟต์หยุดที่ชั้น 1 หลังจากนั้นรีเลย์ตัวที่ 6 ต่อวงจรให้มอเตอร์เคลื่อนที่ช่องลิฟต์กลับมายังจุดเริ่มต้นถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 3 พร้อมรอการจัดเก็บสินค้าชิ้นถัดไป

ระบบการนำสินค้าออกจากคลังเริ่มการทำงานเมื่อผู้ใช้ระบุจำนวนสินค้าที่ต้องการนำออกผ่านแผงแป้นตัวเลขสัญญาณถูกส่งไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผลและสั่งให้รีเลย์ตัดต่อวงจรกับมอเตอร์ของชุดการเคลื่อนที่ โดยการทำงานเริ่มจากรีเลย์ตัวที่ 5 ต่อวงจรให้ช่องลิฟต์เคลื่อนที่ไปหน้าคลังเมื่อถึงแนวแถวของคลังสินค้าที่ต้องการนำออกถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 3 จากนั้นรีเลย์ตัวที่ 7 ต่อวงจรให้มอเตอร์ลิฟต์ทำงานส่งผลให้ลิฟต์เคลื่อนที่ขึ้น (ถ้ามีการนำสินค้าออกชั้นที่ 2) ถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 1 เมื่อเคลื่อนที่ถึงชั้นที่ 2 รีเลย์ตัวที่ 1 ต่อวงจรให้มอเตอร์แท่นขนย้ายเคลื่อนที่เข้าไปในคลังถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 4 หลังจากนั้นรีเลย์ตัวที่ 3 ต่อวงจรให้มอเตอร์ยกสินค้าขึ้น จากนั้นชุดการเคลื่อนที่จะกลับมายังจุดเริ่มต้นโดยเริ่มการทำงานจากรีเลย์ตัวที่ 2 ต่อวงจรส่งผลให้มอเตอร์แท่นขนย้ายเคลื่อนที่ออกจากคลังถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 4 แล้ว รีเลย์ตัวที่ 8 ต่อวงจรให้มอเตอร์ลิฟต์ทำงานกลับทิศการหมุน (ถ้ามีการเก็บสินค้าชั้นที่ 2) มีสวิตช์จำกัดระยะตัวที่ 2 เป็นตัวกำหนดให้ลิฟต์หยุดเมื่อถึงชั้นที่ 1 รีเลย์ตัวที่ 6 ต่อวงจรให้มอเตอร์เคลื่อนที่ช่องลิฟต์ย้อนกลับมายังจุดเริ่มต้นถูกกำหนดด้วยสวิตช์จำกัดระยะตัวที่ 3 หลังจากนั้นรีเลย์ตัวที่ 4 ต่อวงจรให้มอเตอร์วางสินค้าลง และไมโครคอนโทรลเลอร์ประมวลผลจำนวนสินค้าคงคลังและแสดงผลผ่านจอแอลซีดี

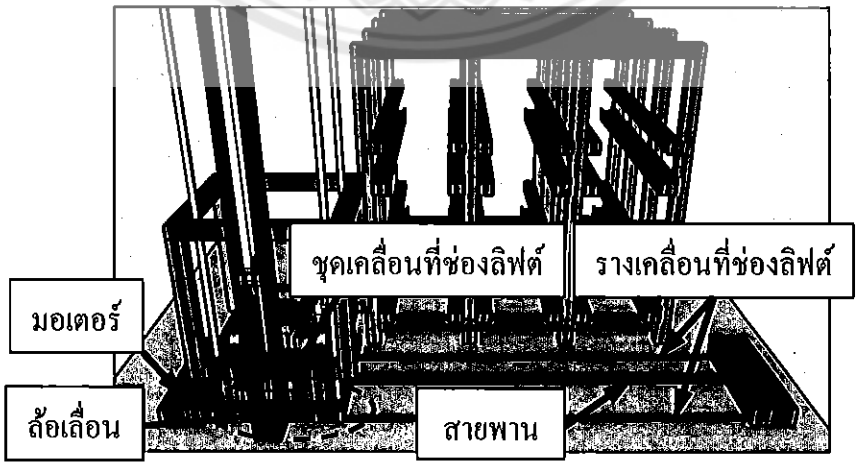
3.2 การออกแบบโครงสร้างและชุดการเคลื่อนที่ระบบจัดเก็บสินค้าและนำออกจากคลัง

การออกแบบโครงสร้างคลังเก็บสินค้าแสดงได้ดังรูปที่ 3.2 ซึ่งแบ่งออกเป็น 2 ชั้น แต่ละชั้นแบ่งเป็น 3 แถว แต่ละแถวสามารถเก็บสินค้าได้ 3 ชิ้น โดยโครงสร้างของคลังเก็บสินค้านี้สร้างขึ้นจากไม้มีขนาดกว้าง 52 cm ยาว 52 cm และสูง 55 cm ซึ่งประกอบด้วยรางวางสินค้าและรางแท่นขนย้ายสินค้า ในส่วนของรางแท่นขนย้ายสินค้าประกอบด้วยไม้เพื่อใช้เป็นโครงสร้างของราง และแผ่นอะคริลิคใช้สำหรับกั้นล้อของแท่นขนย้ายให้อยู่ในรางขณะเคลื่อนที่หน้าคลังสินค้า



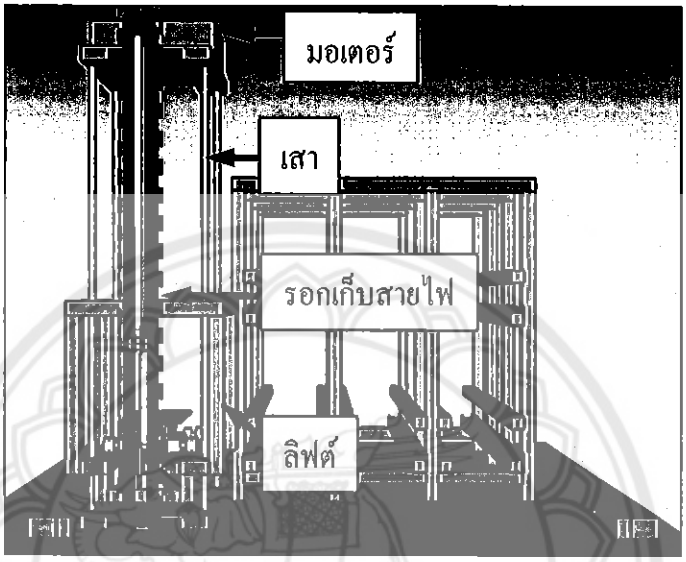
รูปที่ 3.2 โครงสร้างคลังเก็บสินค้า

การออกแบบโครงสร้างชุดการเคลื่อนที่ของช่องลิฟต์แสดงได้ดังรูปที่ 3.3 โดยมีส่วนประกอบหลักคือ มอเตอร์เกียร์ไฟฟ้ากระแสตรง 12 V สายพาน ล้อเลื่อน และรางพลาสติก โดยการทำงานของมอเตอร์เกียร์ไฟฟ้ากระแสตรงทำหน้าที่ดึงสายพานที่ติดกับช่องลิฟต์ให้เคลื่อนที่ไปกับล้อเลื่อนที่ติดอยู่ใต้ช่องลิฟต์ให้เคลื่อนที่ไปตามราง



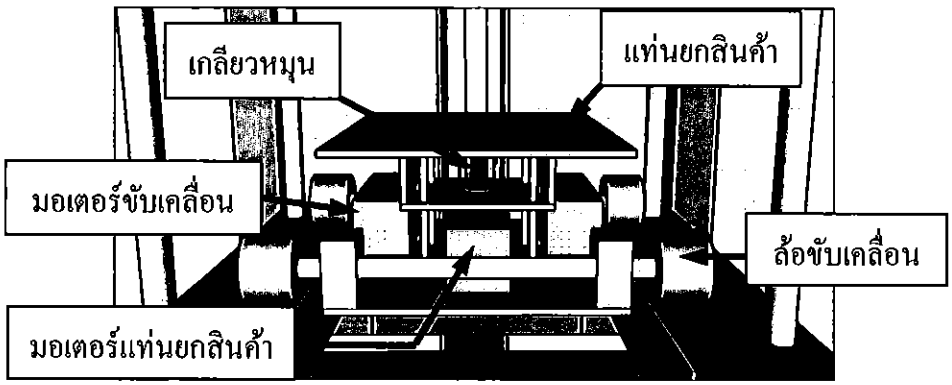
รูปที่ 3.3 ชุดการเคลื่อนที่ของช่องลิฟต์

การออกแบบโครงสร้างชุดการเคลื่อนที่ของลิฟต์แสดงได้ดังรูปที่ 3.4 ประกอบด้วยลิฟต์และมอเตอร์เกียร์ไฟฟ้ากระแสตรงขนาด 12 V โครงสร้างของลิฟต์ทำจากไม้และเสาอะลูมิเนียมทรงกระบอก 4 เสา โดยการทำงานของชุดการเคลื่อนที่ของลิฟต์เริ่มจากมอเตอร์หมุนเพื่อดึงเชือกที่ติดกับลิฟต์ให้เกิดการเคลื่อนที่ขึ้นลงตามแนวดิ่งตามเสาอะลูมิเนียม 4 เสา นอกจากนี้ช่องลิฟต์ยังมีรอกเพื่อใช้สำหรับการเก็บสายไฟที่จ่ายไฟให้มอเตอร์แท่นขนย้ายสินค้าและแท่นยกสินค้า



รูปที่ 3.4 ชุดการเคลื่อนที่ของลิฟต์

การออกแบบโครงสร้างชุดการเคลื่อนที่ของแท่นขนย้ายสินค้าแสดงได้ดังรูปที่ 3.5 ซึ่งโครงสร้างหลักทำจากอะคริลิค และประกอบด้วยมอเตอร์เกียร์ไฟฟ้ากระแสตรง 5 V จำนวน 2 ตัว ทำหน้าที่ขับเคลื่อนล้อของแท่นขนย้าย และมอเตอร์เกียร์ไฟฟ้ากระแสตรง 5 V จำนวน 1 ตัว ทำหน้าที่ยกสินค้า โดยการยกสินค้าจะใช้การหมุนเกลียวเพื่อเคลื่อนแท่นยกขึ้นลงตามเกลียวและมีแท่งเหล็ก 4 แท่ง ช่วยประคองให้แท่นยกสินค้าขึ้นลงตามแนวดิ่งได้แนวเดียว



รูปที่ 3.5 ชุดการเคลื่อนที่ของแท่นขนย้ายสินค้า

3.3 รูปแบบการจัดเก็บสินค้าและนำออกจากคลัง

3.3.1 การออกแบบการจัดวางตำแหน่งในคลังสินค้า

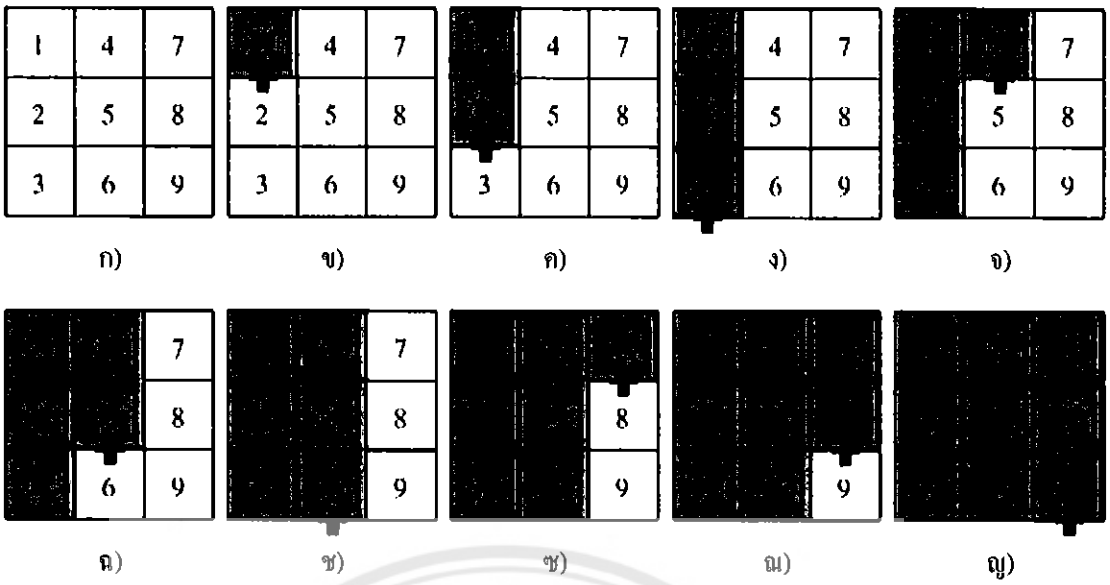
โครงสร้างของคลังสินค้ามีจำนวน 2 ชั้น โดยแต่ละชั้นมีการกำหนดตำแหน่งสินค้าในคลังและตำแหน่งเริ่มต้นที่รอการจัดเก็บสินค้าหรือตำแหน่งเตรียมนำสินค้าออกจากคลัง การออกแบบการจัดวางตำแหน่งสินค้าในคลังชั้นที่ 1 เริ่มจากแถวแรกซึ่งเรียงจากตำแหน่งที่ 1 ถึง 3 แถวที่ 2 ตำแหน่งที่ 4 ถึง 6 และแถวที่ 3 ตำแหน่งที่ 7 ถึง 9 เรียงตามลำดับ แสดงได้ดังรูปที่ 3.6 ก) และตำแหน่งสินค้าในคลังชั้นที่ 2 เริ่มจากแถวแรกซึ่งเรียงจากตำแหน่งที่ 10 ถึง 12 แถวที่ 2 ตำแหน่งที่ 13 ถึง 15 และแถวที่ 3 ตำแหน่งที่ 16 ถึง 18 เรียงตามลำดับ แสดงได้ดังรูปที่ 3.6 ข)

ก) ตำแหน่งสินค้าในคลังชั้นที่ 1 ข) ตำแหน่งสินค้าในคลังชั้นที่ 2

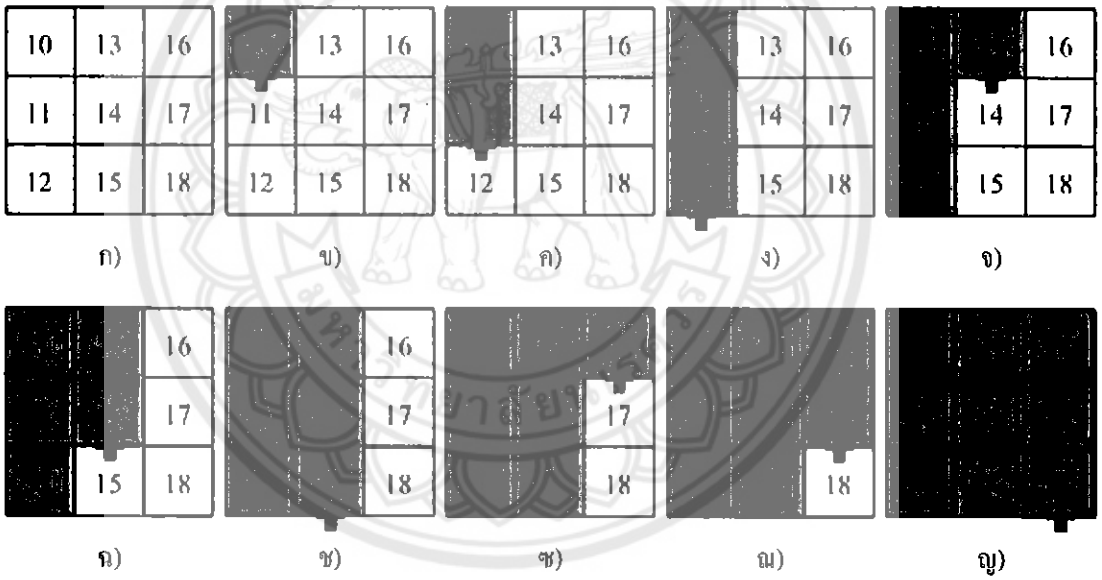
รูปที่ 3.6 ตำแหน่งสินค้าในคลัง

3.3.2 การออกแบบลำดับการจัดเก็บสินค้าเข้าคลัง

การนำสินค้าเข้าไปจัดเก็บในคลังออกแบบให้มีการจัดเก็บที่เป็นระเบียบง่ายต่อการตรวจสอบ และใช้ระยะเวลาในการจัดเก็บรวดเร็ว โดยเริ่มจากการนำสินค้าเข้าจัดเก็บชั้นที่ 1 ที่แถวแรกเรียงจากตำแหน่งที่ 1 จนถึงตำแหน่งที่ 3 จากนั้นจัดเก็บแถวที่ 2 และแถวที่ 3 ตามลำดับ โดยการจัดเก็บสินค้าเรียงจากตำแหน่งที่ 1 ถึง 9 แสดงลำดับการจัดเก็บดังรูปที่ 3.7 ก) ถึง ญ) เมื่อสินค้าชั้นที่ 1 เต็มแล้วจึงนำสินค้าไปเก็บที่ชั้น 2 โดยระบบลิฟต์ทำงานเคลื่อนย้ายสินค้าขึ้นไปที่ชั้น 2 ของคลัง ซึ่งมีลำดับการจัดเก็บสินค้าเหมือนกับชั้นที่ 1 โดยเรียงลำดับการจัดเก็บสินค้าจากตำแหน่งที่ 10 จนถึงตำแหน่งที่ 12 ต่อจากนั้นจึงนำไปจัดเก็บในแถวที่ 2 และแถวที่ 3 ตามลำดับ โดยการจัดเก็บสินค้าที่ชั้น 2 เรียงจากตำแหน่งที่ 10 ถึง 18 จนครบตามตำแหน่งที่กำหนดไว้ในคลัง แสดงลำดับการจัดเก็บสินค้าชั้น 2 ดังรูปที่ 3.8 ก) ถึง ญ) ซึ่งโครงสร้างของระบบจัดเก็บสินค้าและนำออกจากคลังมีจำนวน 2 ชั้นสามารถจัดเก็บสินค้าทั้งหมดได้ 18 ชิ้น



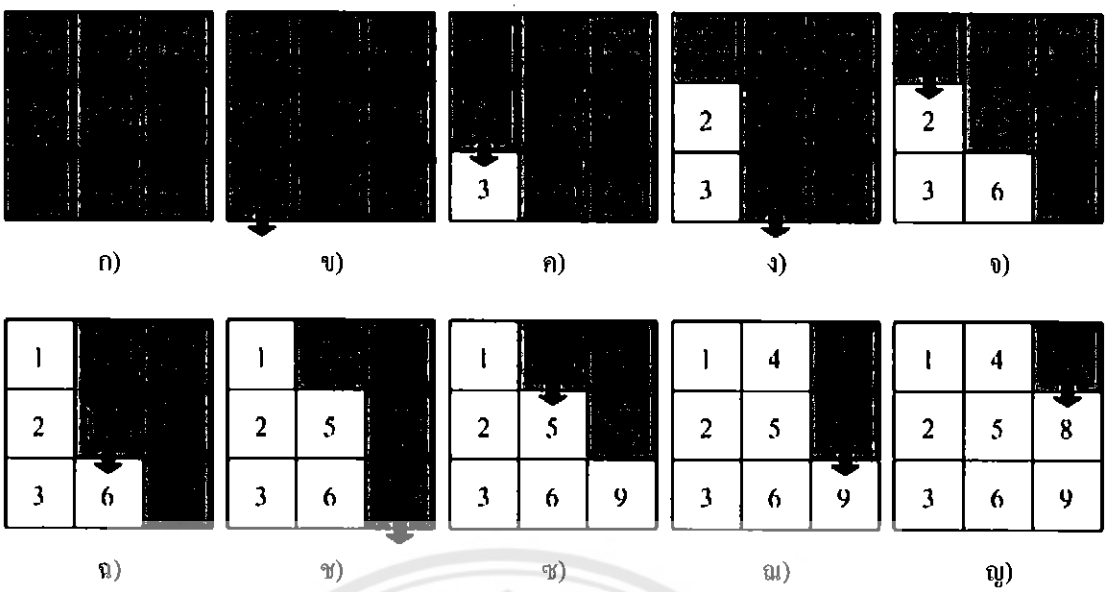
รูปที่ 3.7 ลำดับการจัดเก็บสินค้าชั้นที่ 1



รูปที่ 3.8 ลำดับการจัดเก็บสินค้าชั้นที่ 2

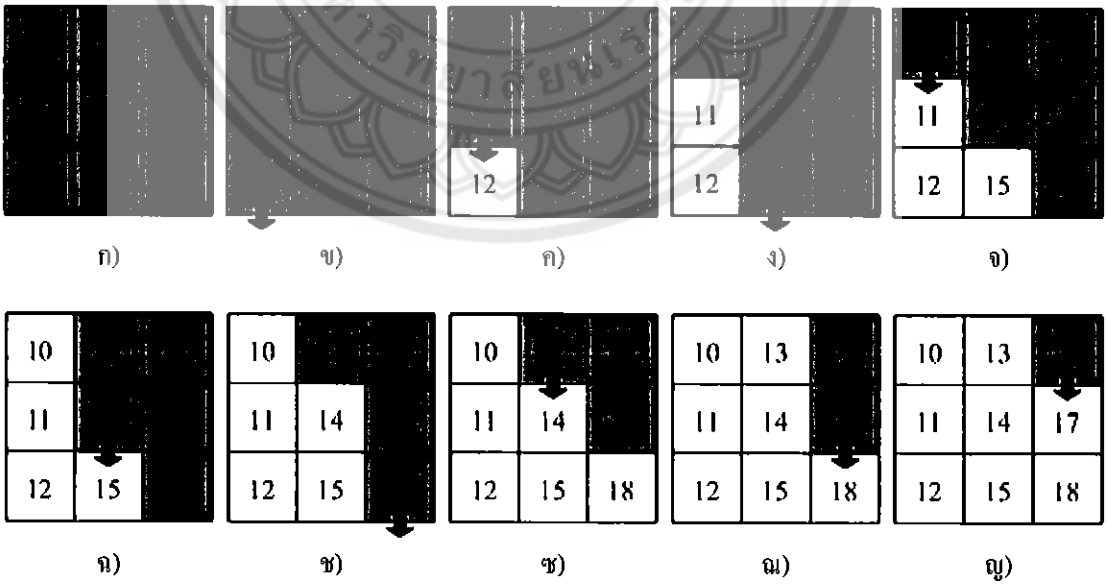
3.3.3 การออกแบบลำดับการนำสินค้าออกจากคลัง

การนำสินค้าออกจากคลังออกแบบให้มีการนำสินค้าในตำแหน่งที่ใกล้ออกก่อนเพื่อเป็นการลดเวลาในการนำสินค้าออกและลดพลังงานไฟฟ้าที่จ่ายให้มอเตอร์ของชุดการเคลื่อนที่ โดยเริ่มต้นจากการนำสินค้าออกชั้นที่ 1 โดยการนำสินค้าออกจากคลังจะเรียงลำดับตามตำแหน่งในคลังที่ 3, 2, 6, 1, 5, 9, 4, 8 และ 7 จนครบตามจำนวนสินค้าที่ต้องการนำออก แสดงลำดับการนำสินค้าออกดังรูปที่ 3.9 ก) ถึง ญ) จากนั้นจึงนำสินค้าออกจากชั้นที่ 2



รูปที่ 3.9 ลำดับการนำสินค้าออกจากคลังชั้นที่ 1

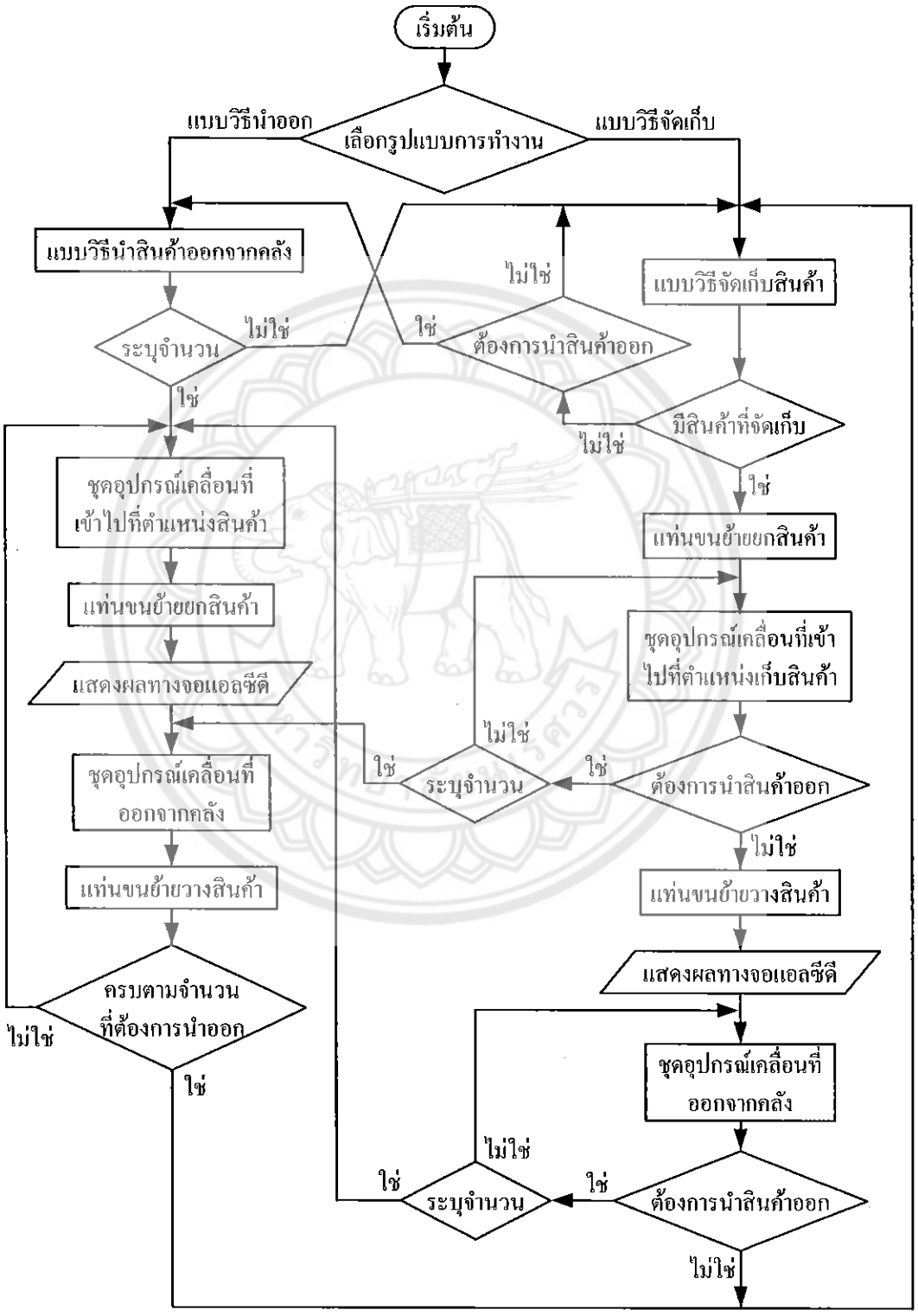
การนำสินค้าออกจากคลังของชั้นที่ 2 นั้นออกแบบให้มีการนำสินค้าในตำแหน่งที่ใกล้ออกก่อนเพื่อเป็นการลดเวลาในการนำสินค้าออกและลดพลังงานไฟฟ้าที่จ่ายให้มอเตอร์ของชุดการเคลื่อนที่ โดยมีลิฟต์ทำหน้าที่เคลื่อนที่ในแนวดิ่งเพื่อนำสินค้าออกจากคลัง โดยการนำสินค้าออกจะเรียงจากตำแหน่งในคลังที่ 12, 11, 15, 10, 14, 18, 13, 17 และ 16 แสดงลำดับการนำสินค้าออกดังรูปที่ 3.10 ก) ถึง ญ)



รูปที่ 3.10 ลำดับการนำสินค้าออกจากคลังชั้นที่ 2

3.4 การออกแบบขั้นตอนการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง

ในการออกแบบขั้นตอนการทำงานของระบบนั้น เริ่มจากออกแบบระบบควบคุมการทำงานตั้งแต่เริ่มต้น แสดงในรูปที่ 3.11



รูปที่ 3.11 ขั้นตอนการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง

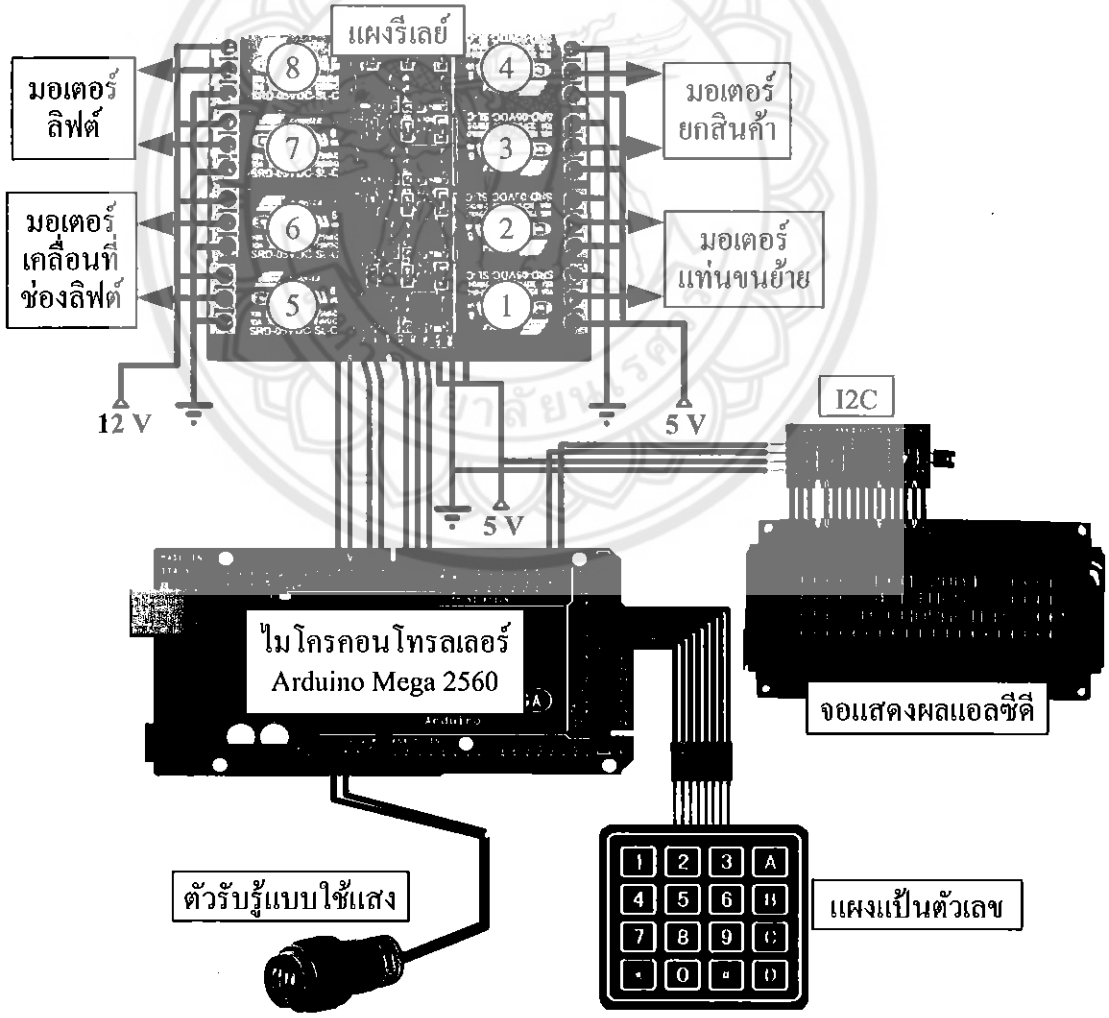
การทำงานของระบบจัดเก็บสินค้าและนำออกจากคลังแบ่งออกเป็น 2 การทำงานหลักคือ การจัดเก็บสินค้าและการนำสินค้าออกจากคลัง โดยสามารถเลือกแบบวิธีการทำงานผ่านแผงแป้นตัวเลข

เมื่อเลือกแบบวิธีเป็นการจัดเก็บสินค้าสามารถเปลี่ยนแบบวิธีเป็นการนำสินค้าออกหรือรอสินค้ามาจัดเก็บได้ โดยการทำงานเริ่มจากตัวรับรู้แบบใช้แสงตรวจพบสินค้า จากนั้นแท่นขนย้ายสินค้าจึงยกสินค้าขึ้นแล้วชุดอุปกรณ์เคลื่อนที่เคลื่อนไปยังตำแหน่งสินค้าภายในคลัง โดยมีลำดับการจัดเก็บสินค้าแสดงในหัวข้อ 3.3.2 ในระหว่างการเคลื่อนที่เข้าไปจัดเก็บสินค้าหากมีความต้องการนำสินค้าออก สามารถกดแผงแป้นตัวเลขเพื่อเปลี่ยนแบบวิธีเป็นการนำสินค้าออกได้ จากนั้นจึงระบุจำนวนสินค้าที่ต้องการนำออกแล้วกดปุ่มยืนยัน ชุดอุปกรณ์เคลื่อนที่นำสินค้าที่กำลังจัดเก็บออกมาเป็นชิ้นแรกแล้วจึงทำการนำสินค้าออกชิ้นถัดไป โดยลำดับการนำสินค้าออกจากคลังแสดงในหัวข้อ 3.3.3 แต่ในกรณีที่ไม่ต้องการนำสินค้าออกหรือไม่มีการระบุจำนวนสินค้าแล้วกดปุ่มยืนยัน ระบบทำงานแบบวิธีจัดเก็บสินค้าต่อไปจนวางสินค้าลงในคลังแล้ว โปรแกรมทำการประมวลผลระบุจำนวนสินค้าคงคลังและแสดงผลออกทางจอแอลซีดี ต่อจากนั้นชุดอุปกรณ์การเคลื่อนที่จะย้อนกลับมายังตำแหน่งเริ่มต้นเพื่อรอสินค้าที่จัดเก็บหรือรอการเปลี่ยนแบบวิธีอีกครั้ง ในขณะชุดอุปกรณ์เคลื่อนที่กำลังเคลื่อนที่กลับ หากมีความต้องการนำสินค้าออก ยังสามารถกดแผงแป้นตัวเลขเพื่อเปลี่ยนแบบวิธีเป็นการนำสินค้าออกได้ หลังจากนั้นจึงระบุจำนวนสินค้าที่ต้องการนำออกแล้วปุ่มยืนยัน ชุดอุปกรณ์เคลื่อนที่กลับออกมาตำแหน่งเริ่มต้นก่อนจึงทำการนำสินค้าออกตามลำดับถัดไปจนครบตามจำนวนที่ระบุไว้

หากเลือกแบบวิธีการนำออก กรณีที่ใส่จำนวนสินค้าที่ต้องการนำออกแล้วกดปุ่มยืนยัน ชุดอุปกรณ์เคลื่อนที่เข้าไปยังตำแหน่งภายในคลังสินค้า โดยมีลำดับการนำสินค้าออกจากคลังที่แสดงในหัวข้อ 3.3.3 เมื่อถึงตำแหน่งสินค้าในคลังแท่นขนย้ายสินค้ายกสินค้าขึ้น จากนั้นโปรแกรมประมวลผลจำนวนสินค้าคงคลังและแสดงผลออกทางจอแอลซีดี หลังจากนั้นชุดอุปกรณ์เคลื่อนที่ออกจากคลังกลับยังตำแหน่งเริ่มต้นแล้ว แท่นขนย้ายวางสินค้าลงเพื่อรอการนำออก และตรวจสอบว่านำสินค้าออกครบตามจำนวนที่ระบุไว้หรือไม่ ถ้าหากยังนำสินค้าออกไม่ครบตามจำนวน ชุดอุปกรณ์เคลื่อนที่เข้าไปในคลังสินค้าเพื่อนำสินค้าชิ้นถัดไปออกมาจนครบตามจำนวนที่ระบุไว้เสร็จเมื่อครบตามจำนวนแล้วจึงกลับสู่แบบวิธีการจัดเก็บสินค้า ในส่วนกรณีที่ไม่ระบุจำนวนแล้วกดปุ่มยืนยัน ระบบเปลี่ยนเป็นแบบวิธีการจัดเก็บสินค้า

3.5 การออกแบบการเชื่อมต่อวงจรของระบบจัดเก็บสินค้าและนำออกจากคลัง

การควบคุมการทำงานของระบบจัดเก็บสินค้าและนำออกจากคลังมีการเชื่อมต่อของอุปกรณ์หลักประกอบด้วยไมโครคอนโทรลเลอร์ แผงรีเลย์ จอแสดงผลแอลซีดี I2C แผงแป้นตัวเลข และตัวรับรู้แบบใช้แสง โดยการทำงานของระบบจะใช้ไมโครคอนโทรลเลอร์ในการรับค่าเพื่อประมวลผลและสั่งงานให้อุปกรณ์ทำงาน ซึ่งรับค่าจากตัวรับรู้แบบใช้แสงเพื่อใช้ตรวจสอบสินค้าที่ต้องการนำเข้าไปเก็บในคลังเก็บสินค้า และแผงแป้นตัวเลขเพื่อใช้รับค่าในควบคุมการทำงาน รวมถึงระบุจำนวนสินค้าที่ต้องการนำออกจากคลัง จากนั้นส่งสัญญาณให้รีเลย์ทำงานตัดต่อวงจรของมอเตอร์กระแสตรงที่ระดับแรงดัน 12 V (มอเตอร์ลิฟต์และมอเตอร์เคลื่อนที่ช่องลิฟต์) และมอเตอร์กระแสตรงที่ระดับแรงดัน 5 V (มอเตอร์ยกสินค้าและมอเตอร์แท่นขนย้าย) นอกจากนั้นยังส่งสัญญาณให้จอแสดงผลแอลซีดีผ่าน I2C เพื่อแสดงสถานะการทำงานและจำนวนสินค้าคงคลัง แสดงการต่อวงจรดังรูปที่ 3.12

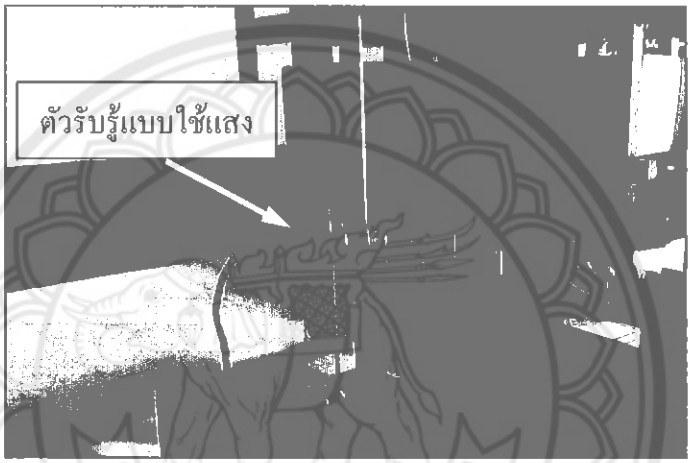


รูปที่ 3.12 การเชื่อมต่อวงจรของระบบจัดเก็บสินค้าและนำออกจากคลัง

3.6 การติดตั้งอุปกรณ์ของแบบจำลองระบบจัดเก็บสินค้าและนำออกจากคลัง

3.6.1 การติดตั้งตัวรับรู้แบบใช้แสง

การทำงานของระบบจัดเก็บสินค้า เริ่มทำงานเมื่อมีตัวรับรู้แบบใช้แสงตรวจพบสินค้าที่รอการจัดเก็บ จากนั้นจะส่งสัญญาณให้ไมโครคอนโทรลเลอร์ประมวลผลและสั่งงานให้รีเลย์ตัดต่อวงจรให้ชุดการเคลื่อนที่ทำงาน นอกจากนี้ตัวรับรู้แบบใช้แสงยังใช้ตรวจสอบสินค้าที่ถูกนำออกจากคลังหากยังมีสินค้าอยู่บนแท่นขนย้ายที่ตำแหน่งเริ่มต้นระบบจะยังไม่ทำงานในขั้นตอนต่อไป โดยตัวรับรู้แบบใช้แสงติดตั้งที่บริเวณตำแหน่งเริ่มต้นแสดงดังรูปที่ 3.13



รูปที่ 3.13 การติดตั้งตัวรับรู้แบบใช้แสง

3.6.2 การติดตั้งสวิตช์จำกัดระยะและก้านสัมผัส

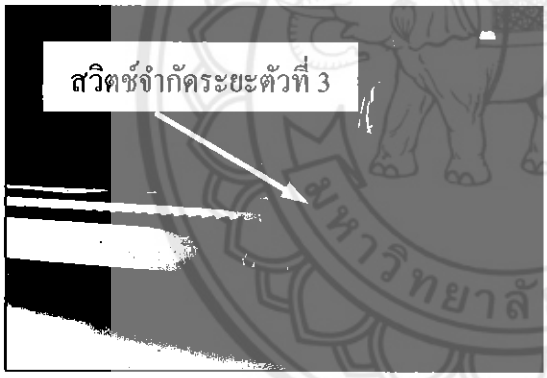
การเคลื่อนที่ของแท่นขนย้ายสินค้าในการจัดเก็บหรือนำสินค้าออกจากคลังนั้น ระบบทำการระบุตำแหน่งการเคลื่อนที่ของแท่นขนย้ายสินค้าได้โดยไมโครคอนโทรเลอร์นับสัญญาณของสวิตช์จำกัดระยะตัวที่ 4 ซึ่งติดตั้งอยู่ใต้แท่นขนย้ายสินค้า เมื่อแท่นขนย้ายสินค้าเคลื่อนที่เข้าหรือออกตามรางของแท่นขนย้ายในคลังสินค้า สวิตช์จำกัดระยะจะถูกกดโดยก้านสัมผัสที่ติดไว้ตามแนวรางของแท่นขนย้าย ซึ่งการติดตั้งสวิตช์จำกัดระยะตัวที่ 4 แสดงดังรูปที่ 3.14 ก) และก้านสัมผัสแสดงดังรูปที่ 3.14 ข)

ก) ตำแหน่งของสวิตช์จำกัดระยะตัวที่ 4

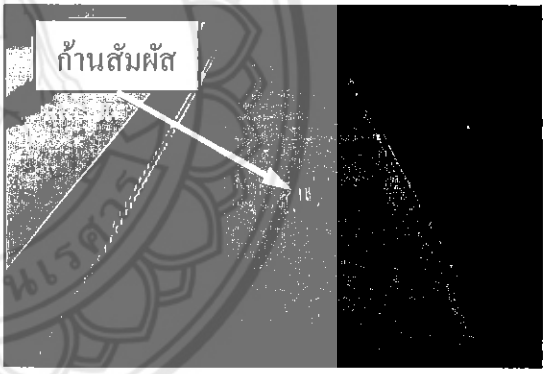
ข) ตำแหน่งของก้านสัมผัส

รูปที่ 3.14 การติดตั้งสวิตช์จำกัดระยะและก้านสัมผัสเพื่อควบคุมการเคลื่อนที่ของแท่นขนย้าย

การเคลื่อนที่ของช่องลิฟต์เพื่อไปยังแต่ละแถวหน้าคลังสินค้านั้น ระบบจะระบุตำแหน่งของแต่ละแถวหน้าคลังสินค้าจากการนับของสวิตช์จำกัดระยะที่ติดอยู่ใต้ช่องลิฟต์ เมื่อช่องลิฟต์เคลื่อนที่ไปหรือกลับสวิตช์จำกัดระยะจะถูกกดโดยก้านสัมผัสที่ติดไว้ตามแนวการเคลื่อนที่ของช่องลิฟต์ ซึ่งการติดตั้งสวิตช์จำกัดระยะแสดงดังรูปที่ 3.15 ก) และก้านสัมผัสแสดงดังรูปที่ 3.15 ข)



ก) ตำแหน่งของสวิตช์จำกัดระยะตัวที่ 3



ข) ตำแหน่งของก้านสัมผัส

รูปที่ 3.15 การติดตั้งสวิตช์จำกัดระยะและก้านสัมผัสเพื่อควบคุมการเคลื่อนที่ของช่องลิฟต์

การเคลื่อนที่ของลิฟต์ระหว่างชั้นที่ 1 และชั้นที่ 2 ระบบทำการระบุตำแหน่งลิฟต์ โดยสวิตช์จำกัดระยะที่ถูกติดตั้งไว้ด้านบนและด้านล่างของช่องลิฟต์ เมื่อลิฟต์เคลื่อนที่ขึ้นไปยังชั้นที่ 2 สวิตช์จำกัดระยะที่ติดไว้ด้านบนจะถูกกด และเมื่อลิฟต์เคลื่อนที่ลงมาชั้นที่ 1 สวิตช์จำกัดระยะที่ติดไว้ด้านล่างจะถูกกด ซึ่งการติดตั้งสวิตช์จำกัดระยะด้านบนแสดงดังรูปที่ 3.16 ก) และด้านล่างของช่องลิฟต์แสดงดังรูปที่ 3.16 ข)

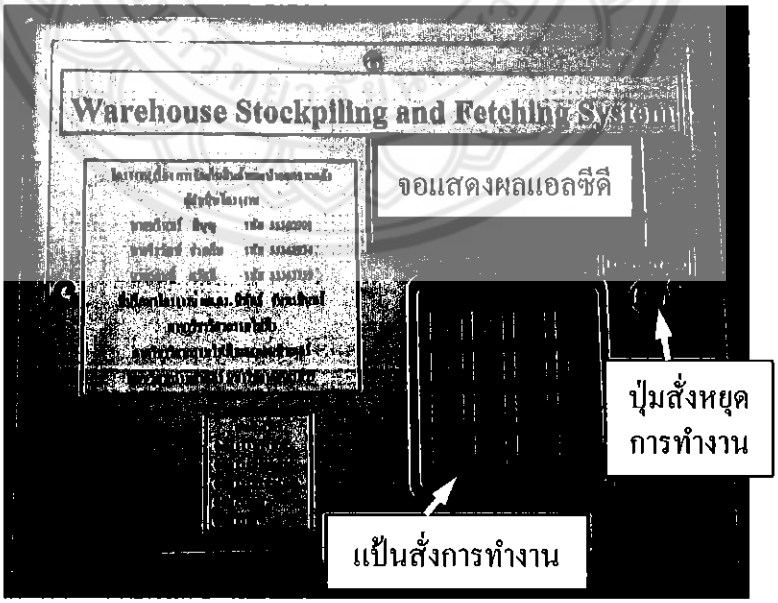
ก) สวิตช์จำกัดระยะด้านบนของช่องลิฟต์ ข) สวิตช์จำกัดระยะด้านล่างของช่องลิฟต์

รูปที่ 3.16 การติดตั้งสวิตช์จำกัดระยะเพื่อควบคุมการเคลื่อนที่ของช่องลิฟต์

3.6.3 กล่องควบคุมการทำงานของระบบ

ก) การติดตั้งอุปกรณ์หน้ากล่องควบคุมการทำงาน

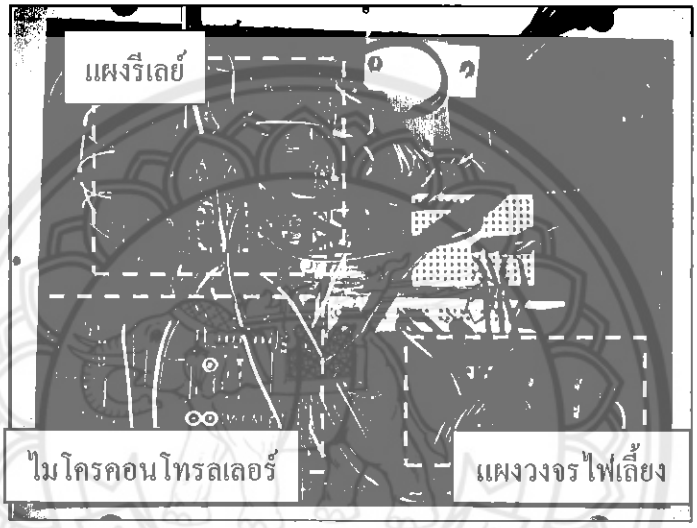
ในส่วนการใช้งานของระบบควบคุมนี้ประกอบด้วยแป้นสั่งการทำงาน และจอแอลซีดีถูกติดตั้งอยู่ด้านหน้าของกล่องควบคุม โดยมีแป้นสั่งการทำงานเป็นอุปกรณ์ที่ใช้กำหนดแบบวิธีการทำงานและกำหนดจำนวนสินค้าที่ต้องการนำออกจากคลังสินค้า สำหรับหน้าจอแสดงผล ใช้แสดงสถานะการทำงาน จำนวนสินค้าคงคลัง และจำนวนสินค้าที่ต้องการนำออกจากคลังสินค้า นอกจากนี้ยังมีปุ่มสั่งหยุดการทำงานเพื่อหยุดการทำงานชั่วคราวดังรูปที่ 3.17



รูปที่ 3.17 หน้าจอแสดงผลและปุ่มควบคุมการทำงาน

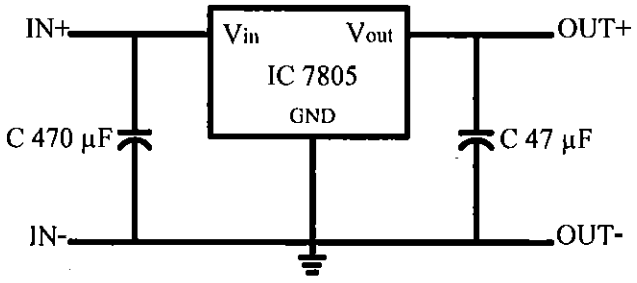
ข) การติดตั้งอุปกรณ์ภายในกล่องควบคุมการทำงาน

ภายในกล่องควบคุมแสดงในรูปที่ 3.18 ประกอบด้วยไมโครคอนโทรลเลอร์ Arduino รุ่น MEGA 2560 แผงรีเลย์ และแผงวงจรไฟเลี้ยง โดยแผงวงจรไฟเลี้ยงรับแรงดันกระแสตรง 12 V จากอะแคปเตอร์มาจ่ายให้กับมอเตอร์และลดระดับแรงดันจาก 12 V ให้เป็นแรงดันระดับ 5 V เพื่อจ่ายให้กับไมโครคอนโทรลเลอร์ ตัวรับรู้แบบใช้แสง และจอแสดงผลแอลซีดี ส่วนแผงรีเลย์เริ่มงานโดยรับคำสั่งจากไมโครคอนโทรลเลอร์เพื่อตัดต่อวงจรให้กับขับมอเตอร์หรือกลับทิศทางของมอเตอร์กระแสตรง



รูปที่ 3.18 การติดตั้งอุปกรณ์ภายในกล่องควบคุม

สำหรับแผงวงจรไฟเลี้ยงประกอบด้วย IC 7805 ตัวเก็บประจุขนาด 470 µF และ 47 µF จำนวนทั้งหมด 3 ชุด ซึ่งทำหน้าที่ลดระดับแรงดันจาก 12 V เป็นระดับแรงดัน 5 V โดยการใช้งานของวงจรไฟเลี้ยงจะจ่ายไฟกระแสตรง 12 V เข้าทางด้าน IN+ และ IN- และต่อทางด้าน OUT+ และ OUT- ที่ระดับแรงดัน 5 V กระแส 1 A แสดงดังรูปที่ 3.29 เพื่อนำไปจ่ายไฟเลี้ยงให้ระบบ



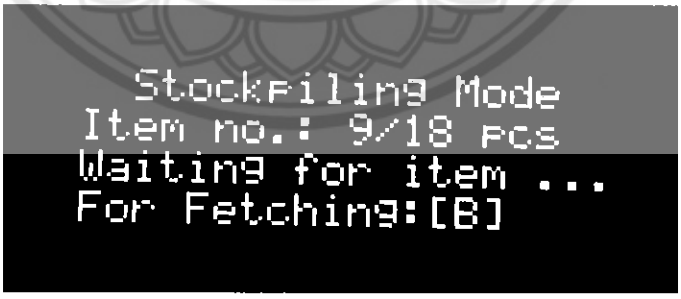
รูปที่ 3.19 แผนภาพวงจรไฟเลี้ยงโดยใช้ IC 7805

3.7 วิธีการใช้งานระบบจัดเก็บสินค้าและนำออกจากคลัง

ผู้ใช้สามารถกำหนดแบบวิธีการทำงานของระบบโดยทำการเลือกกดปุ่มที่แผงแป้นตัวเลข ซึ่งมีแบบวิธีการทำงานอยู่ 2 แบบวิธี คือแบบวิธีการจัดเก็บสินค้าและแบบวิธีการนำสินค้าออกจากคลัง เมื่อผู้ใช้เลือกแบบวิธีการจัดเก็บสินค้าโดยทำการกดปุ่ม A หน้าจอแอลซีดีปรากฏคำว่า For Stockpiling หรือเลือกแบบวิธีการนำสินค้าออกจากคลังโดยทำการกดปุ่ม B หน้าจอแอลซีดีปรากฏคำว่า For Fetching แสดงดังรูปที่ 3.20

รูปที่ 3.20 การแสดงผลของหน้าจอเลือกแบบวิธีการทำงาน

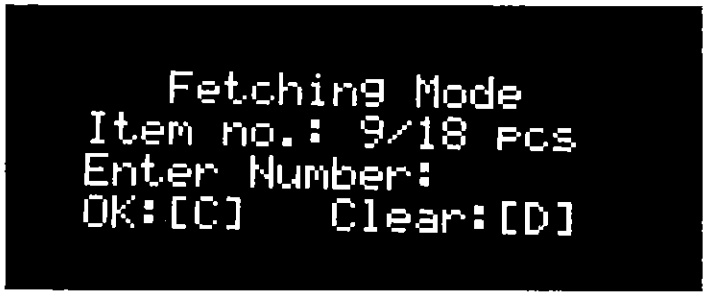
เมื่อผู้ใช้เลือกแบบวิธีการจัดเก็บสินค้าโดยทำการกดปุ่ม A หน้าจอแอลซีดีปรากฏดังรูปที่ 3.21 เมื่อรอสินค้าเข้าจัดเก็บ หน้าจอแอลซีดีปรากฏคำว่า Waiting for item เมื่อระบบการจัดเก็บสินค้าเริ่มทำงาน หน้าจอแอลซีดีปรากฏคำว่า Item no. เพื่อแสดงจำนวนของสินค้าที่มีอยู่ในคลัง นอกจากนี้ขณะที่ระบบทำงานอยู่ยังสามารถเปลี่ยนแบบวิธีการทำงานเป็นการนำสินค้าออกจากคลังได้โดยทำงานกดปุ่ม B



รูปที่ 3.21 การแสดงผลของหน้าจอขณะการจัดเก็บสินค้า

เมื่อผู้ใช้เลือกแบบวิธีการนำสินค้าออกจากคลังโดยทำการกดปุ่ม B หน้าจอแอลซีดีปรากฏดังรูปที่ 3.22 ซึ่งการนำสินค้าออกจากคลังหน้าจอแอลซีดีปรากฏคำว่า Enter Number เพื่อให้ผู้ใช้งานระบุจำนวนสินค้าที่ต้องการนำออกจากคลัง เมื่อระบุจำนวนสินค้าที่ต้องการนำออกแล้วจะต้องยืนยันด้วยการกดปุ่ม C หรือถ้าต้องการแก้ไขจำนวนสามารถทำได้โดยกดปุ่ม D ขณะที่

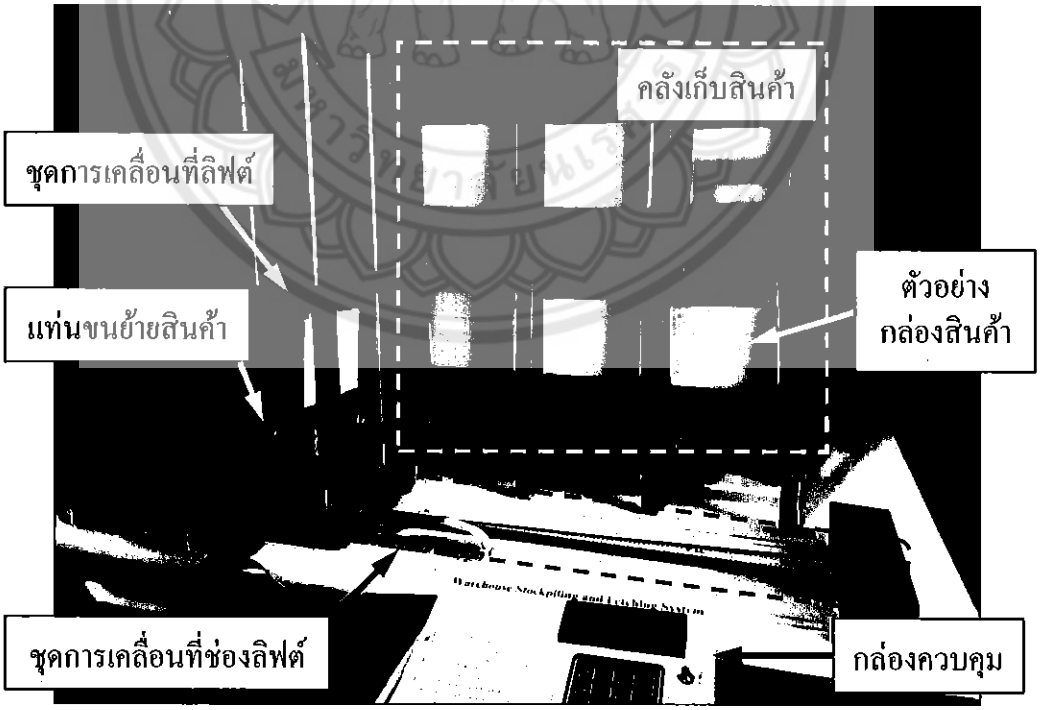
ระบบการนำสินค้าออกจากคลังเริ่มทำงาน หน้าจอแอลซีดีปรากฏคำว่า Item no. เพื่อแสดงจำนวนของสินค้าคงคลัง และแสดงจำนวนสินค้าที่ต้องการนำออก



รูปที่ 3.22 รูปแบบการแสดงผลของหน้าจอขณะนำสินค้าออกจากคลัง

3.8 แบบจำลองโครงสร้างระบบจัดเก็บสินค้าและนำออกจากคลัง

หลังจากออกแบบและสร้างแบบจำลองคลังเก็บสินค้า ชุดการเคลื่อนที่ช่องลิฟต์ ชุดการเคลื่อนที่ลิฟต์ แท่นขนย้ายสินค้า และกล่องควบคุมเสร็จแล้ว จึงได้ประกอบแต่ละส่วนเข้าด้วยกัน นอกจากนี้ยังสร้างตัวอย่างกล่องสินค้าทำจากกระดาษแข็งมีขนาดกว้าง 12 cm ยาว 12 cm และสูง 10 cm ซึ่งแบบจำลองโครงสร้างระบบจัดเก็บสินค้าและนำออกจากคลังแสดงได้ดังรูปที่ 3.23



รูปที่ 3.23 แบบจำลองระบบจัดเก็บสินค้าและนำออกจากคลัง

บทที่ 4
ผลการทดสอบ

4.1 การทดสอบการทำงานของรีเลย์

ในระบบจัดเก็บสินค้าและนำออกจากคลังมีชุดการเคลื่อนที่ต่างๆสำหรับนำสินค้าเข้าจัดเก็บหรือนำสินค้าออกจากคลัง โดยใช้รีเลย์ในการตัดต่อวงจรเพื่อขับเคลื่อนและกลับทิศมอเตอร์กระแสตรงให้ชุดอุปกรณ์เคลื่อนที่ต่าง ๆ เคลื่อนที่ไปยังตำแหน่งในคลังสินค้าตามที่ออกแบบไว้ จึงได้มีการทดสอบการตัดต่อวงจรของรีเลย์ เพื่อตรวจสอบการทำงานของรีเลย์แต่ละตัวที่ใช้ควบคุมมอเตอร์ โดยรีเลย์แต่ละตัวมีหน้าที่การทำงานแสดงดังตารางที่ 4.1

ตารางที่ 4.1 หน้าที่ของรีเลย์แต่ละตัวในระบบจัดเก็บสินค้าและนำออกจากคลัง

รีเลย์ (ตัวที่)	หน้าที่การทำงาน
1	ควบคุมมอเตอร์ของแท่นขนย้ายสินค้าเคลื่อนที่เข้าในคลัง
2	ควบคุมมอเตอร์ของแท่นขนย้ายสินค้าเคลื่อนที่ออกจากคลัง
3	ควบคุมมอเตอร์ของแท่นยกสินค้าเคลื่อนที่ยกสินค้าขึ้น
4	ควบคุมมอเตอร์ของแท่นยกสินค้าเคลื่อนที่วางสินค้าลง
5	ควบคุมมอเตอร์ของชุดการเคลื่อนที่ช่องลิฟต์เคลื่อนที่ไปหน้าคลัง
6	ควบคุมมอเตอร์ของชุดการเคลื่อนที่ช่องลิฟต์เคลื่อนที่กลับ
7	ควบคุมมอเตอร์ของชุดการเคลื่อนที่ลิฟต์เคลื่อนที่ขึ้น
8	ควบคุมมอเตอร์ของชุดการเคลื่อนที่ลิฟต์เคลื่อนที่ลง

โดยชุดการเคลื่อนที่ของระบบจัดเก็บสินค้าและนำออกจากคลังมีการติดตั้งสวิตช์จำกัดระยะทั้งหมด 4 ตัว ทำหน้าที่ส่งสัญญาณให้ไมโครคอนโทรลเลอร์ประมวลและสั่งงานให้รีเลย์ตัดต่อวงจรมอเตอร์เพื่อควบคุมการทำงานของชุดการเคลื่อนที่ต่างๆ คือ แท่นขนย้ายสินค้า แท่นยกสินค้า ชุดการเคลื่อนที่ช่องลิฟต์ และชุดการเคลื่อนที่ลิฟต์ ซึ่งได้ทำการทดสอบการทำงานของรีเลย์แต่ละตัวในแบบวิธีการจัดเก็บสินค้า และแบบวิธีการนำสินค้าออกจากคลัง

4.1.1 แบบวิธีการจัดเก็บสินค้า

การทดสอบการทำงานของรีเลย์ในแบบวิธีการจัดเก็บสินค้า เพื่อตรวจสอบการต่อวงจรของรีเลย์ให้ทำงานถูกต้องตามที่ออกแบบไว้ โดยใช้รีเลย์ทั้งหมด 8 ตัว สำหรับการเคลื่อนที่ 4 ชุด ได้แก่ ชุดการเคลื่อนที่ช่องลิฟต์ ชุดการเคลื่อนที่ลิฟต์ ชุดการเคลื่อนที่แท่นขนย้าย และแท่นยกสินค้า โดยในแต่ละชุดการเคลื่อนที่ประกอบด้วยมอเตอร์ ซึ่งมีหน้าที่ขับเคลื่อนชุดอุปกรณ์เพื่อนำสินค้าเข้าไปจัดเก็บในคลังสินค้า

การทำงานในแบบวิธีการจัดเก็บสินค้าจะมีรีเลย์ทำงานเพียงตัวเดียวในแต่ละลำดับการทำงาน โดยเริ่มการทำงานจากรีเลย์ตัวที่ 3 ต่อวงจรให้แท่นขนย้ายยกสินค้าขึ้น จากนั้นรีเลย์ตัวที่ 5 ต่อวงจรให้ช่องลิฟต์เคลื่อนที่ไปหน้าคลัง หลังจากนั้นรีเลย์ตัวที่ 7 ทำงานส่งผลให้ลิฟต์เคลื่อนที่ขึ้นไปชั้นที่ 2 (ถ้ามีการเก็บสินค้าชั้นที่ 1 รีเลย์ตัวที่ 7 จะไม่ทำงาน) รีเลย์ตัวที่ 1 ต่อวงจรให้แท่นขนย้ายเคลื่อนที่เข้าในคลังแล้ว รีเลย์ตัวที่ 4 ต่อวงจรให้แท่นขนย้ายวางสินค้าลง จากนั้นรีเลย์ตัวที่ 2 ต่อวงจรให้แท่นขนย้ายเคลื่อนที่ออกจากคลัง รีเลย์ตัวที่ 8 จึงทำงานส่งผลให้ลิฟต์เคลื่อนที่ลงมายังชั้นที่ 1 (ถ้ามีการเก็บสินค้าชั้นที่ 1 รีเลย์ตัวที่ 8 จะไม่ทำงาน) หลังจากนั้นรีเลย์ตัวที่ 6 ต่อวงจรให้ช่องลิฟต์เคลื่อนที่กลับมายังจุดเริ่มต้นเพื่อรอการจัดเก็บสินค้าชิ้นถัดไป โดยการทำงานของรีเลย์ในแบบวิธีการจัดเก็บสินค้าแสดงดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดสอบการทำงานของรีเลย์ในแบบวิธีการจัดเก็บสินค้า

ลำดับการทำงาน	หน้าสัมผัสของรีเลย์							
	ตัวที่ 1	ตัวที่ 2	ตัวที่ 3	ตัวที่ 4	ตัวที่ 5	ตัวที่ 6	ตัวที่ 7	ตัวที่ 8
แท่นขนย้ายยกสินค้า	X	X	√	X	X	X	X	X
ช่องลิฟต์เคลื่อนที่ไปหน้า	X	X	X	X	√	X	X	X
ลิฟต์เคลื่อนที่ขึ้น (ชั้นที่ 2)	X	X	X	X	X	X	√	X
แท่นขนย้ายเคลื่อนที่เข้า	√	X	X	X	X	X	X	X
แท่นขนย้ายวางสินค้า	X	X	X	√	X	X	X	X
แท่นขนย้ายเคลื่อนที่ออก	X	√	X	X	X	X	X	X
ลิฟต์เคลื่อนที่ลง (ชั้นที่ 2)	X	X	X	X	X	X	X	√
ช่องลิฟต์เคลื่อนที่กลับ	X	X	X	X	X	√	X	X

หมายเหตุ: เครื่องหมาย √ หมายถึง หน้าสัมผัสต่อวงจร และ X หมายถึง หน้าสัมผัสเปิดวงจร

4.1.2 แบบวิธีการนำสินค้าออกจากคลัง

การทดสอบการทำงานของรีเลย์ในแบบวิธีการนำสินค้าออกจากคลัง เพื่อตรวจสอบการต่อวงจรของรีเลย์มีให้ทำงานถูกต้องตามที่ออกแบบไว้ โดยมีรีเลย์ทั้งหมด 8 ตัว ใช้สำหรับการเคลื่อนที่ 4 ชุด ได้แก่ ชุดการเคลื่อนที่ช่องลิฟต์ ชุดการเคลื่อนที่ลิฟต์ ชุดการเคลื่อนที่แท่นขนย้าย และแท่นยกสินค้าโดยในแต่ละชุดการเคลื่อนที่ประกอบด้วยมอเตอร์ซึ่งมีหน้าที่ขับเคลื่อนชุดอุปกรณ์นำสินค้าออกจากคลัง

การทำงานในแบบวิธีการนำสินค้าออกจากคลังจะมีรีเลย์ทำงานเพียงตัวเดียวในแต่ละลำดับการทำงาน โดยเริ่มการทำงานจากรีเลย์ตัวที่ 5 ต่อวงจรให้ช่องลิฟต์เคลื่อนที่ไปหน้าคลัง จากนั้นรีเลย์ตัวที่ 7 ทำงานเพื่อให้ลิฟต์เคลื่อนที่ขึ้นชั้นที่ 2 (ถ้ามีการนำสินค้าออกในชั้นที่ 1 รีเลย์ตัวที่ 7 จะไม่ทำงาน) หลังจากนั้นรีเลย์ตัวที่ 1 ต่อวงจรให้แท่นขนย้ายเคลื่อนที่เข้าไปในคลังแล้วรีเลย์ตัวที่ 3 จึงต่อวงจรให้แท่นขนย้ายยกสินค้าขึ้น จากนั้นรีเลย์ตัวที่ 2 ต่อวงจรให้แท่นขนย้ายเคลื่อนที่ออกจากคลัง รีเลย์ตัวที่ 8 จึงทำงานส่งผลให้ลิฟต์เคลื่อนที่ลงชั้น 1 (ถ้ามีการเก็บสินค้าชั้นที่ 1 รีเลย์ตัวที่ 8 จะไม่ทำงาน) หลังจากรีเลย์ตัวที่ 6 ต่อวงจรให้ช่องลิฟต์เคลื่อนที่กลับมาจุดเริ่มต้นแล้ว รีเลย์ตัวที่ 4 จึงต่อวงจรให้แท่นขนย้ายวางสินค้าลง เป็นอันเสร็จสิ้นกระบวนการนำสินค้าแต่ละชิ้นออกจากคลัง โดยการทำงานของรีเลย์ในแบบวิธีการนำสินค้าออกแสดงดังตารางที่ 4.3

ตารางที่ 4.3 ผลการทดสอบการทำงานของรีเลย์ในแบบวิธีการนำสินค้าออกจากคลัง

ลำดับการทำงาน	หน้าสัมผัสของรีเลย์							
	ตัวที่ 1	ตัวที่ 2	ตัวที่ 3	ตัวที่ 4	ตัวที่ 5	ตัวที่ 6	ตัวที่ 7	ตัวที่ 8
ช่องลิฟต์เคลื่อนที่ไปหน้า	X	X	X	X	√	X	X	X
ลิฟต์เคลื่อนที่ขึ้น (ชั้นที่ 2)	X	X	X	X	X	X	√	X
แท่นขนย้ายเคลื่อนที่เข้า	√	X	X	X	X	X	X	X
แท่นขนย้ายยกสินค้า	X	X	√	X	X	X	X	X
แท่นขนย้ายเคลื่อนที่ออก	X	√	X	X	X	X	X	X
ลิฟต์เคลื่อนที่ลง (ชั้นที่ 2)	X	X	X	X	X	X	X	√
ช่องลิฟต์เคลื่อนที่กลับ	X	X	X	X	X	√	X	X
แท่นขนย้ายวางสินค้า	X	X	X	√	X	X	X	X

หมายเหตุ: เครื่องหมาย √ หมายถึง หน้าสัมผัสต่อวงจร และ X หมายถึง หน้าสัมผัสเปิดวงจร

4.2 การทดสอบหาระยะเวลาที่ใช้ในการจัดเก็บและนำสินค้าออกจากคลัง

เนื่องจากความเร็วในการจัดเก็บและนำสินค้าออกจากคลังนั้นเป็นการบอกถึงประสิทธิภาพการทำงานของระบบ จึงได้มีการทดสอบเพื่อหาเวลาในการจัดเก็บสินค้าและนำสินค้าออกจากคลัง โดยทำการทดสอบหาเวลาแต่ละตำแหน่ง ซึ่งทำการทดสอบตำแหน่งละ 3 ครั้ง และหาค่าเฉลี่ยของเวลาที่ใช้ รวมทั้งหาเวลารวมในการจัดเก็บสินค้าหรือนำสินค้าออก

ก) การทดสอบหาระยะเวลาในการจัดเก็บสินค้า

จากการทดสอบหาระยะเวลาในการจัดเก็บสินค้าตามลำดับ พบว่าการจัดเก็บสินค้าลำดับที่ 1, 4, 7, 10, 13 และ 16 ซึ่งเป็นตำแหน่งที่อยู่ในสุดของแต่ละแถวจึงใช้เวลาจัดเก็บมากที่สุดในแถวนั้น ลำดับการจัดเก็บที่ 3, 6, 9, 12, 15 และ 18 ใช้เวลาในการจัดเก็บน้อยที่สุดในแต่ละแถวเนื่องจากเป็นตำแหน่งที่อยู่นอกสุดของแถว นอกจากนี้เมื่อเปรียบเทียบตำแหน่งที่ลึกสุดของแต่ละแถว หรือตำแหน่งกลางของแต่ละแถว หรือตำแหน่งนอกสุดของแต่ละแถว โดยแถวที่อยู่ใกล้จุดเริ่มต้นใช้เวลาน้อยกว่าแถวที่อยู่ไกลออกไป และชั้นที่ 1 ใช้ในการจัดเก็บเวลาน้อยกว่าชั้นที่ 2 โดยการจัดเก็บสินค้าทั้ง 2 ชั้น จำนวน 18 ชิ้น ใช้ระยะเวลาทั้งหมด 395.16 s หรือประมาณ 6.35 นาที ระยะเวลาที่ใช้ในการจัดเก็บแสดงดังตารางที่ 4.4 โดยการจัดเก็บสินค้าเก็บในคลังมีการเรียงลำดับตามหัวข้อที่ 3.3.2

ข) การทดสอบหาระยะเวลาในการนำสินค้าออกจากคลัง

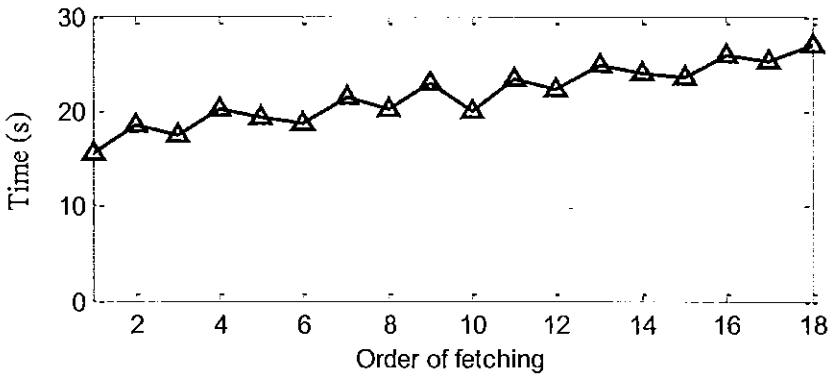
จากการทดสอบหาระยะเวลาในการนำสินค้าออกจากคลัง พบว่าการนำสินค้าออกจากตำแหน่งที่ 3 ใช้เวลาน้อยที่สุด เนื่องจากเป็นตำแหน่งที่อยู่ใกล้จุดเริ่มต้นมากที่สุด แล้วยิ่งตำแหน่งอยู่ไกลจากจุดเริ่มต้นยิ่งใช้เวลานาน ซึ่งสอดคล้องกับลำดับการนำสินค้าออกที่ออกแบบไว้ตามหัวข้อที่ 3.3.3 โดยการนำสินค้าออกจากคลังทั้งหมด จำนวน 18 ชิ้น ใช้ระยะเวลารวม 390.34 s หรือประมาณ 6.30 นาที ระยะเวลาที่ในการนำสินค้าออกแสดงดังตารางที่ 4.5 นอกจากนั้นผลการทดสอบยังพบว่าเวลาที่ใช้ในการนำสินค้าออกแต่ละตำแหน่งไม่ได้เพิ่มสอดคล้องกับลำดับของการนำสินค้าออก เนื่องจากความเร็วของมอเตอร์ที่ใช้ในเคลื่อนช่องลิฟต์มีค่าสูงกว่าความเร็วของมอเตอร์ที่ใช้ขับเคลื่อนแท่นขนย้าย ถึงอย่างไรก็ตามแนวโน้มของระบบเวลาที่ใช้ดังกล่าวยังคงสอดคล้องกับแนวคิดที่ใช้ในการออกแบบ นั่นคือมีค่าเพิ่มขึ้นตามลำดับการนำสินค้าออกจากคลัง แสดงดังรูปที่ 4.1

ตารางที่ 4.4 ผลการทดสอบหาระยะเวลาในการจัดเก็บสินค้า

ลำดับการจัดเก็บสินค้า	ตำแหน่งสินค้า	เวลา (s)			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
1	1	18.89	19.03	19.04	18.99
2	2	18.27	18.56	18.45	18.43
3	3	16.48	16.55	16.54	16.52
4	4	21.37	21.74	21.50	21.54
5	5	19.44	19.71	19.57	19.57
6	6	17.76	18.00	17.55	17.77
7	7	22.59	22.51	22.70	22.60
8	8	21.61	20.73	20.67	21.00
9	9	18.87	18.56	18.79	18.74
10	10	25.17	24.30	25.36	24.94
11	11	23.32	21.95	23.22	22.83
12	12	21.52	21.44	21.44	21.47
13	13	26.37	26.89	26.33	26.53
14	14	24.82	24.19	24.66	24.56
15	15	22.27	22.89	22.59	22.58
16	16	28.03	29.11	28.18	28.44
17	17	25.38	25.94	24.37	25.23
18	18	22.30	24.01	23.96	23.42
รวม		394.46	396.11	394.92	395.16

ตารางที่ 4.5 ผลการทดสอบหาระยะเวลาในการนำสินค้าออกจากคลัง

ลำดับการนำสินค้าออก	ตำแหน่งสินค้า	เวลา (s)			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
1	3	15.36	15.94	15.40	15.57
2	2	18.30	18.31	18.63	18.41
3	6	17.57	17.56	17.30	17.48
4	1	20.11	20.22	20.16	20.16
5	5	19.40	19.39	19.36	19.38
6	9	18.80	18.61	18.79	18.73
7	4	21.30	21.51	21.34	21.38
8	8	20.63	19.28	20.64	20.18
9	7	22.52	23.74	22.47	22.91
10	12	20.16	20.18	19.85	20.06
11	11	23.05	23.80	23.14	23.33
12	15	22.24	22.36	22.22	22.21
13	10	25.00	24.96	24.75	24.90
14	14	24.10	24.11	23.93	24.05
15	18	23.54	23.49	23.34	23.46
16	13	25.89	25.86	25.69	25.81
17	17	25.33	25.22	25.09	25.21
18	16	27.21	27.05	26.83	27.03
รวม		390.51	391.59	388.93	390.34



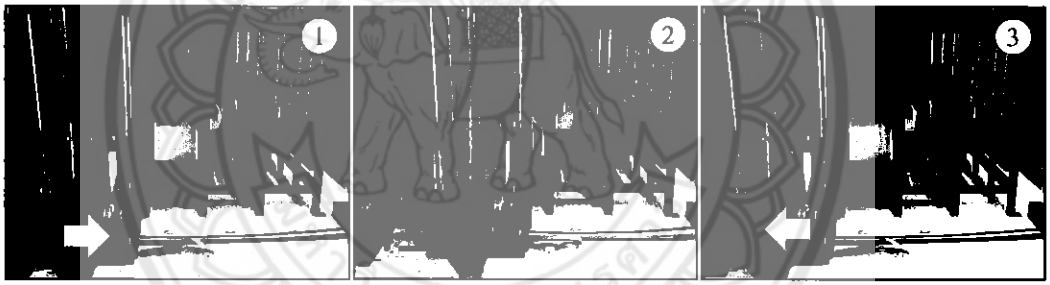
รูปที่ 4.1 ระยะเวลาที่ใช้ในการนำสินค้าออกจากคลัง

4.3 การทดสอบการเปลี่ยนแบบวิธี

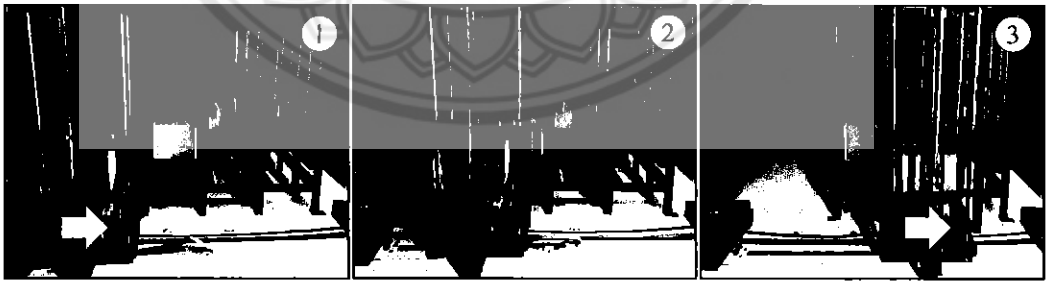
การทำงานของระบบจัดเก็บสินค้าและนำออกจากคลัง ผู้ใช้สามารถนำสินค้าออกได้ทันทีด้วยการกดปุ่มเปลี่ยนแบบวิธีขณะที่ระบบทำงานจัดเก็บสินค้า โดยการทดสอบการนำสินค้าออกขณะที่ชุดการเคลื่อนที่แต่ชุดทำงานนำสินค้าเข้าไปเก็บ แบ่งได้ 6 กรณีดังนี้

4.3.1 การทดสอบเปลี่ยนแบบวิธีขณะช่องลิฟต์เคลื่อนที่ไปหน้าคลัง

ในระหว่างที่ระบบทำงานในแบบวิธีการจัดเก็บสินค้า ขณะที่ชุดการเคลื่อนที่ช่องลิฟต์กำลังนำสินค้าเข้าไปจัดเก็บในคลังสินค้าดังรูปหมายเลข 1 ของรูปที่ 4.2 ก) และ ข) หากผู้ใช้ต้องการให้นำสินค้าออกจากคลังทันทีโดยการกดปุ่ม B เพื่อเปลี่ยนแบบวิธี ส่งผลให้ชุดการเคลื่อนที่ช่องลิฟต์หยุดทันทีดังรูปหมายเลข 2 ของรูปที่ 4.2 ก) และ ข) เพื่อรอให้ผู้ใช้ป้อนจำนวนสินค้าที่ต้องการนำออกจากคลัง หลังจากผู้ใช้ป้อนจำนวนและกดยืนยันแล้วช่องลิฟต์จึงเคลื่อนที่ย้อนกลับมาที่ตำแหน่งเริ่มต้นดังรูปหมายเลข 3 ของรูปที่ 4.2 ก) แต่ถ้าผู้ใช้ไม่ป้อนจำนวนสินค้าแล้วกดยืนยันช่องลิฟต์จะเคลื่อนที่ไปหน้าคลังเพื่อจัดเก็บสินค้าต่อไปดังรูปหมายเลข 3 ของรูปที่ 4.2 ข)



ก) กรณีที่ป้อนจำนวนแล้วกดยืนยัน

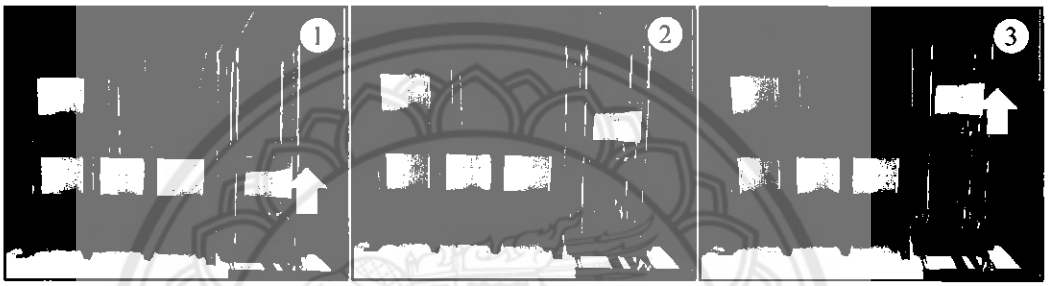


ข) กรณีที่ไม่ป้อนจำนวนแล้วกดยืนยัน

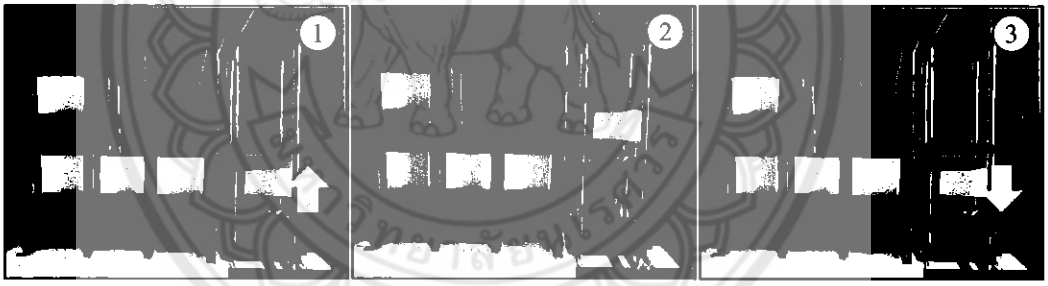
รูปที่ 4.2 การเคลื่อนที่ของช่องลิฟต์ขณะเคลื่อนที่ไปหน้าคลัง

4.3.2 การทดสอบเปลี่ยนแบบวิธีขณะลิฟต์เคลื่อนที่ขึ้น

ในระหว่างที่ระบบทำงานในแบบวิธีการจัดเก็บสินค้า ขณะที่ชุดการเคลื่อนที่ลิฟต์กำลังเคลื่อนที่ดังรูปหมายเลข 1 ของรูปที่ 4.3 ก) และ ข) หากผู้ใช้ต้องการให้นำสินค้าออกจากคลังทันทีโดยการกดปุ่ม B เพื่อเปลี่ยนแบบวิธี ส่งผลให้ชุดการเคลื่อนที่ลิฟต์หยุดทันทีดังรูปหมายเลข 2 ของรูปที่ 4.3 ก) และ ข) เพื่อรอให้ผู้ใช้ป้อนจำนวนสินค้าที่ต้องการนำออกจากคลังหลังจากผู้ใช้ป้อนจำนวนและกดยืนยันแล้วลิฟต์จึงเคลื่อนที่กลับลงมาที่ชั้นล่างดังรูปหมายเลข 3 ของรูปที่ 4.3 ก) จากนั้นช่องลิฟต์จึงเคลื่อนที่กลับมาตำแหน่งเริ่มต้น แต่ถ้าผู้ใช้ไม่ป้อนจำนวนแล้วกดยืนยัน ลิฟต์จะเคลื่อนที่จัดเก็บสินค้าต่อไปดังรูปหมายเลข 3 ของรูปที่ 4.3 ข)



ก) กรณีที่ไม่ป้อนจำนวนแล้วกดยืนยัน



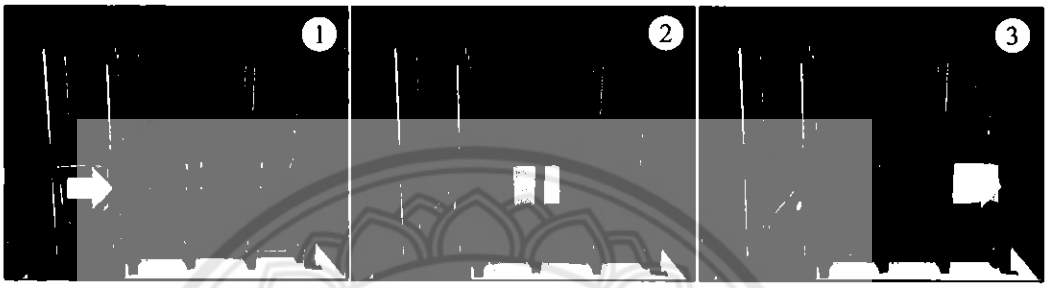
ข) กรณีที่ป้อนจำนวนแล้วกดยืนยัน

รูปที่ 4.3 การเคลื่อนที่ของลิฟต์ขณะเคลื่อนที่ขึ้น

4.3.3 การทดสอบเปลี่ยนแบบวิธีขณะแท่นขนย้ายเคลื่อนที่เข้าไปในคลัง

ในระหว่างที่ระบบทำงานในแบบวิธีการจัดเก็บสินค้า ขณะที่ชุดการเคลื่อนที่แท่นขนย้ายกำลังเคลื่อนที่ดังแสดงในรูปหมายเลข 1 ของรูปที่ 4.4 ก) และ ข) หากผู้ใช้ต้องการให้นำสินค้าออกจากคลังทันทีโดยการกดเปลี่ยนแบบวิธี ส่งผลให้ชุดการเคลื่อนที่แท่นขนย้ายหยุดทันทีดังแสดงในรูปหมายเลข 2 ของรูปที่ 4.4 ก) และ ข) เพื่อรอให้ผู้ใช้ป้อนจำนวนสินค้าที่ต้องการนำออกจากคลังหลังจากผู้ใช้ป้อนจำนวนและกดยืนยันแล้ว แท่นขนย้ายจึงเคลื่อนที่ย้อนกลับมาที่ลิฟต์ดังแสดงในรูปหมายเลข 3 ของรูปที่ 4.4 ก) ต่อมาลิฟต์จะเคลื่อนที่กลับลงมาที่ชั้นล่าง จากนั้นช่องลิฟต์จึงเคลื่อนที่กลับมาตำแหน่งเริ่มต้น แต่ถ้าผู้ใช้ไม่ป้อนจำนวนสินค้าแล้วกดยืนยัน แท่นขนย้ายเคลื่อนที่เข้าไปในคลังเพื่อจัดเก็บสินค้าต่อไปดังแสดงในรูปหมายเลข 1 ของรูปที่ 4.4 ข)

ก) กรณีที่ป้อนจำนวนแล้วกดยืนยัน



ข) กรณีที่ไม่ป้อนจำนวนแล้วกดยืนยัน

รูปที่ 4.4 การเคลื่อนที่ของแท่นขนย้ายขณะเคลื่อนที่เข้าในคลัง

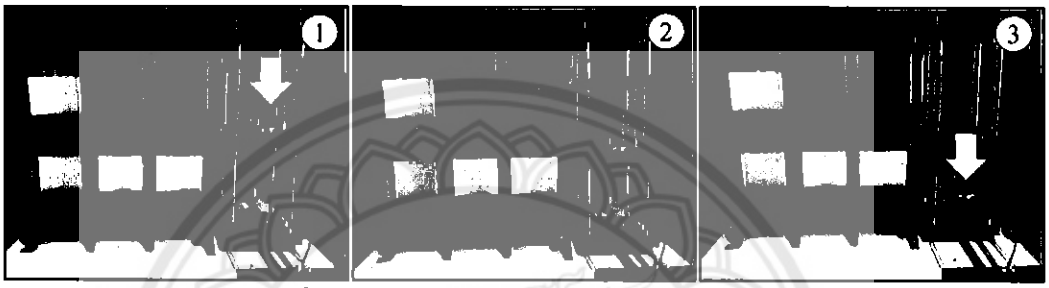
4.3.4 การทดสอบเปลี่ยนแบบวิธีขณะแท่นขนย้ายเคลื่อนที่ออกจากคลัง

ในระหว่างที่ระบบทำงานในแบบวิธีการจัดเก็บสินค้า หลังจากวางสินค้าลงแล้ว ในขณะที่แท่นขนย้ายสินค้ากำลังเคลื่อนที่ออกจากคลังดังรูปหมายเลข 1 ของรูปที่ 4.5 หากผู้ใช้ต้องการให้นำสินค้าออกจากคลังโดยการกดปุ่ม B เพื่อเปลี่ยนแบบวิธี ส่งผลให้ชุดการเคลื่อนที่แท่นขนย้ายหยุดทันทีดังรูปหมายเลข 2 ของรูปที่ 4.5 เพื่อรอให้ผู้ใช้ป้อนจำนวนสินค้าที่ต้องการนำออกจากคลัง หลังจากผู้ใช้ป้อนจำนวนและกดยืนยันแล้วแท่นขนย้ายจะกลับมายังลิฟต์ดังรูปหมายเลข 3 ของรูปที่ 4.5 ต่อมาลิฟต์จะเคลื่อนที่กลับลงมาที่ชั้นล่าง จากนั้นช่องลิฟต์เคลื่อนที่กลับมาตำแหน่งเริ่มต้นแล้วจึงไปนำสินค้าออกตามลำดับที่ออกแบบไว้ในหัวข้อที่ 3.3.3

รูปที่ 4.5 การเคลื่อนที่ของแท่นขนย้ายขณะเคลื่อนที่ออกจากคลัง

4.3.5 การทดสอบเปลี่ยนแบบวิธีขณะลิฟต์เคลื่อนที่ลง

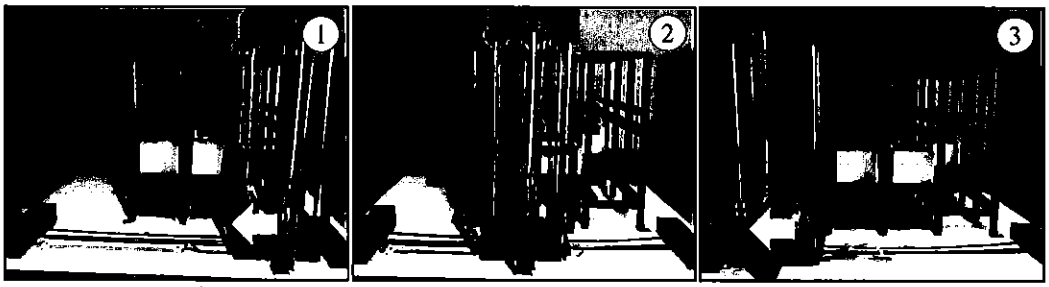
ในระหว่างที่ระบบทำงานในแบบวิธีการจัดเก็บสินค้า หลังจากวางสินค้าลงแล้ว ในขณะที่ลิฟต์กำลังเคลื่อนที่ลงดังรูปหมายเลข 1 ของรูปที่ 4.6 หากผู้ใช้ต้องการให้นำสินค้าออกจากคลังโดยการกดปุ่ม B เพื่อเปลี่ยนแบบวิธี ส่งผลให้ชุดการเคลื่อนที่ลิฟต์หยุดทันทีดังรูปหมายเลข 2 ของรูปที่ 4.6 เพื่อรอให้ผู้ใช้ป้อนจำนวนสินค้าที่ต้องการนำออกจากคลัง หลังจากผู้ใช้ป้อนจำนวนและกดยืนยันแล้วลิฟต์จะลงมายังชั้นล่างดังรูปหมายเลข 3 ของรูปที่ 4.6 จากนั้นช่องลิฟต์เคลื่อนที่กลับมาตำแหน่งเริ่มต้น แล้วจึงไปนำสินค้าออกตามลำดับที่ออกแบบไว้ในหัวข้อที่ 3.3.3



รูปที่ 4.6 การเคลื่อนที่ของลิฟต์ขณะเคลื่อนที่ลง

4.3.6 การทดสอบเปลี่ยนแบบวิธีขณะช่องลิฟต์เคลื่อนที่กลับ

ในระหว่างที่ระบบทำงานในแบบวิธีการจัดเก็บสินค้า หลังจากวางสินค้าลงแล้ว ในขณะที่ช่องลิฟต์กำลังเคลื่อนที่กลับดังรูปหมายเลข 1 ของรูปที่ 4.7 หากผู้ใช้ต้องการให้นำสินค้าออกจากคลังโดยการกดปุ่ม B เพื่อเปลี่ยนแบบวิธี ส่งผลให้ชุดการเคลื่อนที่ช่องลิฟต์หยุดทันทีดังรูปหมายเลข 2 ของรูปที่ 4.7 เพื่อรอให้ผู้ใช้ป้อนจำนวนสินค้าที่ต้องการนำออกจากคลัง หลังจากผู้ใช้ป้อนจำนวนและกดยืนยันแล้วช่องลิฟต์จะกลับมาตำแหน่งเริ่มต้นดังรูปหมายเลข 3 ของรูปที่ 4.7 จากนั้นจึงไปนำสินค้าออกตามลำดับที่ออกแบบไว้ในหัวข้อที่ 3.3.3



รูปที่ 4.7 การเคลื่อนที่ของช่องลิฟต์ขณะเคลื่อนที่กลับมาจุดเริ่มต้น

บทที่ 5
สรุปผลและข้อเสนอแนะ

จากการดำเนินโครงงาน สามารถสรุปผล ชี้แจงปัญหาในการดำเนินงาน รวมทั้งเสนอแนะแนวทางการแก้ปัญหา และให้ข้อเสนอแนะในการพัฒนาต่อไปดังนี้

5.1 สรุปผลการดำเนินงาน

ในโครงงานนี้ได้ทำการออกแบบและสร้างแบบจำลองโครงสร้างคลังเก็บสินค้าขนาดกว้าง 52 cm ยาว 52 cm และสูง 55 cm เป็นจำนวน 2 ชั้น แต่ละชั้นสามารถเก็บสินค้าได้ 9 ชิ้น รวมทั้งยังสร้างชุดการเคลื่อนที่ช่องลิฟต์เพื่อใช้ในการเคลื่อนที่ไปแต่ละแถวหน้าคลังสินค้า ชุดการเคลื่อนที่ลิฟต์เพื่อใช้ในการเคลื่อนที่ขึ้นลงชั้นที่ 2 และชุดการเคลื่อนที่แท่นขนย้ายสินค้าเพื่อใช้ในการเคลื่อนที่นำสินค้าเข้าจัดเก็บหรือออกจากคลัง ซึ่งเป็นระบบที่ควบคุมแบบอัตโนมัติด้วยไมโครคอนโทรลเลอร์ เพื่อความสะดวกรวดเร็วในการจัดเก็บและนำออก รวมถึงมีความแม่นยำและง่ายต่อการตรวจสอบจำนวนสินค้าในคลัง การทำงานของระบบนั้นแบ่ง 2 แบบวิธี คือ แบบวิธีการจัดเก็บสินค้าโดยมีรูปแบบจัดเก็บที่เรียงตามลำดับตำแหน่งสินค้าในคลังที่ออกแบบไว้ และแบบวิธีการนำสินค้าออกมีรูปแบบการนำสินค้าที่ตำแหน่งใกล้จุดเริ่มต้นออกก่อนโดยคำนึงถึงระยะทางในการเคลื่อนที่จากจุดเริ่มต้นถึงตำแหน่งของสินค้าในคลัง อีกทั้งยังแสดงผลการทำงานผ่านหน้าจอแอลซีดี โดยผู้ใช้สามารถกำหนดการทำงานของระบบได้ตามความต้องการ

จากการทดสอบการทำงานของแบบจำลองระบบจัดเก็บสินค้าและนำออกจากคลังที่สร้างขึ้นพบว่าในการจัดเก็บสินค้าเข้าคลังทั้งหมด 18 ชิ้น ใช้เวลาเฉลี่ย 395.16 s หรือ 6.35 นาที และการนำสินค้าออกจากคลังทั้งหมด 18 ชิ้น ใช้เวลาเฉลี่ย 390.34 s หรือ 6.30 นาที โดยผลการทดสอบดังกล่าวนั้น แสดงให้เห็นว่าระบบจัดเก็บสินค้าและนำออกจากคลังสามารถทำงานได้ตามที่ออกแบบไว้

จากการทดสอบการเปลี่ยนแบบวิธีระหว่างที่ระบบทำงานในแบบวิธีจัดเก็บสินค้า ขณะที่ชุดอุปกรณ์เคลื่อนที่เข้าไปจัดเก็บสินค้าสามารถเปลี่ยนแบบวิธีได้ทันทีโดยการกดปุ่ม B แล้วป้อนจำนวน จากนั้นกดปุ่มยืนยันระบบจะเปลี่ยนเป็นแบบวิธีการนำสินค้าออกจากคลัง หรือหากมีการกดเปลี่ยนแบบวิธีหลังจากจัดเก็บสินค้าแล้ว ขณะชุดอุปกรณ์เคลื่อนที่ออกจากคลัง ชุดอุปกรณ์จะกลับมายังจุดเริ่มต้นก่อนจึงเริ่มทำงานในแบบวิธีการนำสินค้าออก ผลการทดสอบดังกล่าวนั้นแสดงให้เห็นว่าระบบจัดเก็บสินค้าและนำออกจากคลัง สามารถทำงานได้ตามที่ออกแบบไว้

5.2 ปัญหาและแนวทางการแก้ไข

1) ในการเคลื่อนที่ของช่องลิฟต์เกิดการสั่นเมื่อหยุดการเคลื่อนที่แบบทันที จึงไม่เหมาะกับโครงสร้างที่มีจำนวนหลายชั้น เนื่องจากมีจุดศูนย์ถ่วงอยู่สูง ทำให้ในขณะเคลื่อนที่ช่องลิฟต์อาจล้มได้ ซึ่งสามารถแก้ปัญหาโดยสร้างโครงสร้างของช่องลิฟต์ให้คงที่ติดกับคลังสินค้า
2) การสร้างแบบจำลองแท่นขนย้ายมีขนาดเล็กจึงมีข้อจำกัดในการเลือกรูปแบบการยกหรือวางสินค้าบนแท่น ซึ่งรูปแบบที่เลือกใช้ในโครงงานนี้คือใช้แกนที่ทำจากน็อตเพื่อหมุนเกลียวให้เกิดการขึ้นลงตามแนวดิ่ง ซึ่งทำให้เกิดปัญหาการโคลงเคลงขณะยกหรือวางสินค้า เนื่องจากเครื่องเจาะมีขีดจำกัดจึงไม่สามารถเจาะแกนหมุนให้ตรงได้ แต่ถ้าสร้างแบบจำลองที่มีขนาดใหญ่ขึ้นจะสามารถใช้น็อตที่มีขนาดใหญ่ขึ้นได้ทำให้ความคลาดเคลื่อนในการเจาะน้อยลงแล้วการโคลงเคลงจะลดลงหรือถ้าต้องการไม่ให้มีการโคลงเคลงเลย ให้สร้างแบบจำลองที่มีขนาดใหญ่พอที่จะติดตั้งระบบนิวเมติกส์ได้ แต่ทำให้ระบบที่ออกแบบมีราคาสูงขึ้น
3) ในการเคลื่อนที่ของชุดอุปกรณ์มีการกำหนดตำแหน่งโดยใช้สวิตช์จำกัดระยะและก้านสัมผัส ซึ่งบางครั้งอาจทำงานผิดพลาดเนื่องจากการสั่นของหน้าสัมผัสในขณะเปลี่ยนสถานะ จึงจำเป็นต้องมีการเขียนโปรแกรมให้หน่วงเวลาในการนับจำนวนการชนก้านสัมผัสแต่ละครั้งหรือการใช้สวิตช์จำกัดระยะที่มีคุณภาพสูงซึ่งมีราคาแพง

5.3 แนวทางในการพัฒนาต่อไป

1) ชุดการเคลื่อนที่แท่นขนย้ายสามารถเปลี่ยนเป็นการทำงานแบบไร้สาย โดยติดตั้งแบตเตอรี่ไว้ที่แท่นขนย้ายสำหรับเป็นแหล่งพลังงานในการขับเคลื่อนมอเตอร์ ซึ่งช่วยลดปัญหาในการจัดเก็บสายไฟ
2) ออกแบบให้ระบบเลือกตำแหน่งสินค้าที่ต้องการจัดเก็บหรือนำออกได้ตามต้องการ ทำให้คลังสินค้าสามารถจัดเก็บสินค้าที่มีความแตกต่างกันได้ทั้งในเรื่องชนิดและอายุของสินค้า นอกจากนี้ยังประยุกต์ใช้งานในรูปแบบอื่นได้ เช่น ระบบโรงจอดรถในอาคารที่ต้องการประหยัดพื้นที่
3) ในการใช้งานจริงสามารถเพิ่มช่องทางเพื่อให้มีการจัดเก็บหรือนำสินค้าออกได้หลายทางหรือหลายชิ้นพร้อมกัน ทำให้ระบบมีการทำงานรวดเร็วและตอบสนองความต้องการของผู้ใช้ได้มากขึ้น
4) สามารถสั่งให้ระบบทำงานจากคอมพิวเตอร์หรือสั่งงานผ่านเครือข่ายอินเตอร์เน็ตได้

เอกสารอ้างอิง

[1] ไชชาญ หินเกิด "เครื่องกลไฟฟ้า 1", บริษัทประชาชน, กรุงเทพฯ,2537.

[2] http://www.ps-thai.org, สืบค้นเมื่อ พฤศจิกายน 2558.

[3] ซ่อมมอเตอร์.com/การทำงานของมอเตอร์, สืบค้นเมื่อ พฤศจิกายน 2558.

[4] Muhammad H. Rashid "Power Electronics Circuits, Devices and Applications", Pearson Education, Inc., Third Edition, 2004.

[5] Smart Learning, "หลักการทางานของรีเลย์ (Relay)", สืบค้นเมื่อพฤศจิกายน 2558 จาก http://www.smartlearningweb.com/knowledge/relay/relay.htm

[6] บ้านอิเล็กทรอนิกส์, "อุปกรณ์ ตอน รีเลย์", สืบค้นเมื่อพฤศจิกายน 2558 จาก http://www.semi-shop.com/knowledge/knowledge_detail.php?sk_id=28

[7] http://www.lpc.rmutl.ac.th/elcen/elearning/motorcontrol/module2/symbole.html, สืบค้นเมื่อ พฤศจิกายน 2558

[8] http://fonengineering.com/our-products/19-limit-switch.html, สืบค้นเมื่อ พฤศจิกายน 2558

[9] http://epautomation.blogspot.com/2011/11/blog-post.html, สืบค้นเมื่อ พฤศจิกายน 2558

[10] ทีมงานสมาร์ทเลิร์นนิ่ง. "เซนเซอร์ ทรานสดิวเซอร์และการใช้งาน." กรุงเทพฯ: สำนักพิมพ์ห้างหุ่นส่วนสามัญสมาร์ทเลิร์นนิ่ง.2552.

[11] http://www.arduitronics.com,สืบค้นเมื่อ พฤศจิกายน 2558

[12] ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา. สืบค้นเมื่อ 6 พฤศจิกายน 2558 จาก http://www.ee.buu.ac.th

[13] บริษัท วีนัสซัพพลาย จากัด, "การใช้งาน Character LCD Display กับ Arduino", สืบค้นเมื่อ พฤศจิกายน 2558 จาก http://thaieasyelec.com/article-wiki/review-product-article

[14] http://www.thaieasyelec.com/, สืบค้นเมื่อ พฤศจิกายน 2558

[15] http://www.thaieasyelec.com/article-wiki/review-product-article, สืบค้นเมื่อ มีนาคม 2559

ภาคผนวก ก

รหัสต้นฉบับของโปรแกรมควบคุมระบบการจัดเก็บสินค้าและนำออกจากคลัง

```
#include <Keypad.h> //เรียกใช้งานแผงแป้นตัวเลข
#include <Wire.h> //เรียกใช้งานไอทูซี
#include <LiquidCrystal_I2C.h> //เรียกใช้งานจอแอลซีดี
LiquidCrystal_I2C lcd(0x3F, 20, 4); //กำหนดแอดเดสและขนาดของจอแอลซีดี
const byte ROWS = 4; //กำหนดตัวแปร ROWS เท่ากับ 4
const byte COLS = 4; //กำหนดตัวแปร COLS เท่ากับ 4
char keys[ROWS][COLS] = { //กำหนดตัวแปรอาเรย์ keys ขนาด ROWS คูณ COLS
 {'1','2','3','A'},
 {'4','5','6','B'},
 {'7','8','9','C'},
 {'*','0','#','D'}
};
byte rowPins[ROWS] = {30,31,32,33}; //เชื่อมต่อแถวของแป้นตัวเลข
byte colPins[COLS] = {34,35,36,37}; //เชื่อมต่อหลักของแป้นตัวเลข
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
const int MBoard1 = 2; //กำหนดตัวแปรควบคุมมอเตอร์แท่นขนย้าย
const int MBoard2 = 3; //กำหนดตัวแปรควบคุมมอเตอร์แท่นขนย้าย
const int MLiftBoard1 = 4; //กำหนดตัวแปรควบคุมมอเตอร์ยกสินค้า
const int MLiftBoard2 = 5; //กำหนดตัวแปรควบคุมมอเตอร์ยกสินค้า
const int MShaft1 = 6; //กำหนดตัวแปรควบคุมมอเตอร์เคลื่อนที่ช่องลิฟต์
const int MShaft2 = 7; //กำหนดตัวแปรควบคุมมอเตอร์เคลื่อนที่ช่องลิฟต์
const int MLift1 = 8; //กำหนดตัวแปรควบคุมมอเตอร์ลิฟต์
const int MLift2 = 9; //กำหนดตัวแปรควบคุมมอเตอร์ลิฟต์
const int Sensor = 22; //กำหนดตัวแปรรับค่าจากตัวรับรู้แบบใช้แสง
const int LimShaftPin = 24; //กำหนดตัวแปรรับค่าจากสวิตช์จำกัดระยะใช้ควบคุมช่องลิฟต์
const int LimliftDownPin = 25; //กำหนดตัวแปรรับค่าจากสวิตช์จำกัดระยะใช้ควบคุมลิฟต์ขึ้น
const int LimliftUpPin = 26; //กำหนดตัวแปรรับค่าจากสวิตช์จำกัดระยะใช้ควบคุมลิฟต์ลง
const int LimBoardPin = 27; //กำหนดตัวแปรรับค่าจากสวิตช์จำกัดระยะใช้ควบคุมแท่นขนย้าย
const int Emergency = 19; //กำหนดตัวแปรรับค่าจากปุ่มกดฉุกเฉิน
char key, KeY; //กำหนดตัวแปรรับค่าจากแป้นตัวเลข
int i, j, k, w, x, emer=1; //กำหนดตัวแปรใช้นับการวนรอบและตัวแปรบอกสถานะการทำงาน
int Shaftlocation, Boardlocation, Liftstate; //กำหนดตัวแปรบอกตำแหน่งคลังสินค้า
```

```
int count1=0, count2=0; //กำหนดตัวแปรใช้นับการเคลื่อนที่
int memTotal=0; //กำหนดตัวแปรเก็บค่าจำนวนสินค้าคงคลัง
int Allitemout=0, itemout=0; //กำหนดตัวแปรเก็บค่าจำนวนสินค้านำออกทั้งหมดและที่เหลืออยู่
int Movelocation[5][10] = { //กำหนดเมทริกซ์อาเรย์เก็บค่าตำแหน่งคลังสินค้า
  {0,0,0,0,0,0,0,0,0,0},
  {0,2,2,2,3,3,3,4,4,4},
  {0,4,3,2,4,3,2,4,3,2},
  {0,2,2,3,2,3,4,3,4,4},
  {0,2,3,2,4,3,2,4,3,4}
};
int Memorylocation[3][19] = { //กำหนดเมทริกซ์อาเรย์เก็บค่าจำนวนสินค้าคงคลัง
  {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
  {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
  {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
};
```

```
void setup() {
  Serial.begin(9600); //เปิดการใช้พอร์ตอนุกรม
  lcd.begin(); //เปิดการใช้จอแอลซีดี
  pinMode(MBoard1, OUTPUT); //กำหนดตัวแปร MBoard1 เป็นเอาท์พุต
  pinMode(MBoard2, OUTPUT); //กำหนดตัวแปร MBoard2 เป็นเอาท์พุต
  pinMode(MLiftBoard1, OUTPUT); //กำหนดตัวแปร MLiftBoard1 เป็นเอาท์พุต
  pinMode(MLiftBoard2, OUTPUT); //กำหนดตัวแปร MLiftBoard2 เป็นเอาท์พุต
  pinMode(MShaft1, OUTPUT); //กำหนดตัวแปร MShaft1 เป็นเอาท์พุต
  pinMode(MShaft2, OUTPUT); //กำหนดตัวแปร MShaft2 เป็นเอาท์พุต
  pinMode(MLift1, OUTPUT); //กำหนดตัวแปร MLift1 เป็นเอาท์พุต
  pinMode(MLift2, OUTPUT); //กำหนดตัวแปร MLift2 เป็นเอาท์พุต
  pinMode(Sensor, INPUT_PULLUP); //กำหนดตัวแปร Sensor เป็นอินพุต
  pinMode(LimShaftPin, INPUT_PULLUP); //กำหนดตัวแปร LimShaftPin เป็นอินพุต
  pinMode(LimliftUpPin, INPUT_PULLUP); //กำหนดตัวแปร LimliftUpPin เป็นอินพุต
  pinMode(LimliftDownPin, INPUT_PULLUP); //กำหนดตัวแปร LimliftDownPin เป็นอินพุต
  pinMode(LimBoardPin, INPUT_PULLUP); //กำหนดตัวแปร LimliftBoardPin เป็นอินพุต
```

```
 pinMode(Emergency, INPUT_PULLUP);  //กำหนดตัวแปร Emergency เป็นอินพุต
 digitalWrite(MBoard1, HIGH);
 digitalWrite(MBoard2, HIGH);
 digitalWrite(MLiftBoard1, HIGH);
 digitalWrite(MLiftBoard2, HIGH);
 digitalWrite(MShaft1, HIGH);
 digitalWrite(MShaft2, HIGH);
 digitalWrite(MLift1, HIGH);
 digitalWrite(MLift2, HIGH);
 Lcd_Mode_Selection();
 Memory();
 in();
 out();
}
```

```
void in(){  //ฟังก์ชันตั้งค่าจำนวนสินค้าในเมทริกซ์ Memorylocation
 Memorylocation[2][4] = Memorylocation[1][1];
 Memorylocation[2][2] = Memorylocation[1][2];
 Memorylocation[2][1] = Memorylocation[1][3];
 Memorylocation[2][7] = Memorylocation[1][4];
 Memorylocation[2][5] = Memorylocation[1][5];
 Memorylocation[2][3] = Memorylocation[1][6];
 Memorylocation[2][9] = Memorylocation[1][7];
 Memorylocation[2][8] = Memorylocation[1][8];
 Memorylocation[2][6] = Memorylocation[1][9];

 Memorylocation[2][13] = Memorylocation[1][10];
 Memorylocation[2][11] = Memorylocation[1][11];
 Memorylocation[2][10] = Memorylocation[1][12];
 Memorylocation[2][16] = Memorylocation[1][13];
 Memorylocation[2][14] = Memorylocation[1][14];
 Memorylocation[2][12] = Memorylocation[1][15];
```

```
  Memorylocation[2][18] = Memorylocation[1][16];
  Memorylocation[2][17] = Memorylocation[1][17];
  Memorylocation[2][15] = Memorylocation[1][18];
}
void out(){  //ฟังก์ชันตั้งค่าจำนวนสินค้าในเมทริกซ์ Memorylocation
  Memorylocation[1][1] = Memorylocation[2][4];
  Memorylocation[1][2] = Memorylocation[2][2];
  Memorylocation[1][3] = Memorylocation[2][1];
  Memorylocation[1][4] = Memorylocation[2][7];
  Memorylocation[1][5] = Memorylocation[2][5];
  Memorylocation[1][6] = Memorylocation[2][3];
  Memorylocation[1][7] = Memorylocation[2][9];
  Memorylocation[1][8] = Memorylocation[2][8];
  Memorylocation[1][9] = Memorylocation[2][6];

  Memorylocation[1][10] = Memorylocation[2][13];
  Memorylocation[1][11] = Memorylocation[2][11];
  Memorylocation[1][12] = Memorylocation[2][10];
  Memorylocation[1][13] = Memorylocation[2][16];
  Memorylocation[1][14] = Memorylocation[2][14];
  Memorylocation[1][15] = Memorylocation[2][12];
  Memorylocation[1][16] = Memorylocation[2][18];
  Memorylocation[1][17] = Memorylocation[2][17];
  Memorylocation[1][18] = Memorylocation[2][15];
}
void readIn(){  //ฟังก์ชันหาตำแหน่งเมื่อต้องการจัดเก็บสินค้า
  int Numberlocation;  //กำหนดตัวแปรบอกตำแหน่งสินค้า
  Shaftlocation=0;
  Boardlocation=0;
  Liftstate=0;
  for(i=1;i<19;i++){  //วนรอบหาตำแหน่งที่ไม่มีสินค้า 18 รอบ
    Numberlocation = Memorylocation[1][i];
```

```
  if(Numberlocation==0){  //ตรวจสอบสถานะสินค้า
    j=i;
    if(i>9){ตรวจสอบเงื่อนไขว่าตำแหน่งจัดเก็บสินค้าอยู่ชั้น 2
      Liftstate = 1;
      j = i-9;
    }
    Shaftlocation = Movelocation[1][j];
    Boardlocation = Movelocation[2][j];
    k=1;
    break ;
   }
  }
}
void readOut(){  //ฟังก์ชันหาตำแหน่งเมื่อต้องการนำสินค้าออก
  int Numberlocation;  //กำหนดตัวแปรบอกตำแหน่งสินค้า
  Shaftlocation=0;
  Boardlocation=0;
  Liftstate=0;
  for(i=1;i<19;i++){  //วนรอบหาตำแหน่งที่มีสินค้า 18 รอบ
    Numberlocation = Memorylocation[2][i];
    if(Numberlocation==1){  //ตรวจสอบสถานะสินค้า
      j=i;
      if(i>9){  //ตรวจสอบเงื่อนไขว่าตำแหน่งนำสินค้าออกอยู่ชั้น 2
        Liftstate = 1;
        j = i-9;
      }
      Shaftlocation = Movelocation[3][j];
      Boardlocation = Movelocation[4][j];
      k=2;
      break ;
    }
  }
```

```
}
void Switch(){  //ฟังก์ชันเปลี่ยนสถานะสินค้า ณ ตำแหน่งสินค้าที่มีการจัดเก็บหรือนำสินค้าออก
 if(w==0){
  if(k==1){  //ตรวจสอบเงื่อนไขว่ามีการจัดเก็บสินค้า
   Memorylocation[1][i] = 1;
   in();
  }
  if(k==2){  //ตรวจสอบเงื่อนไขว่ามีการนำสินค้าออก
   Memorylocation[2][i] = 0;
   out();
  }
 }
}
void Memory() {  //ฟังก์ชันคำนวณจำนวนสินค้าคงคลัง
 memTotal=0;
 for(int i=1;i<19;i++){
  memTotal=memTotal+Memorylocation[1][i];
 }
}
```

```
void Lift_Item(){  //ฟังก์ชันยกสินค้า
 if(w==0){
  delay(1000);
  digitalWrite(MLiftBoard1, LOW);
  digitalWrite(MLiftBoard2, HIGH);
  delay(3000);
  digitalWrite(MLiftBoard1, HIGH);
  digitalWrite(MLiftBoard2, HIGH);
 }
}
void Put_Item(){  //ฟังก์ชันวางสินค้า
 if(w==0){
```

```
    delay(1000);
    digitalWrite(MLiftBoard1, HIGH);
    digitalWrite(MLiftBoard2, LOW);
    delay(3200);
    digitalWrite(MLiftBoard1, HIGH);
    digitalWrite(MLiftBoard2, HIGH);
  }
}
void ShaftForward(){  //ฟังก์ชันการเคลื่อนที่ไปของฟังก์ชัน
  if(w==0){
    delay(1000);
    digitalWrite(MShaft1, LOW);
    digitalWrite(MShaft2, HIGH);
    while(1){
      key = keypad.getKey();
      if(key=='B'){Stop(); KeY=key; x=1; goto A;}  //ตรวจสอบเงื่อนไขมีการเปลี่ยนแบบวิธี
      if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขมีการกดปุ่มหยุดฉุกเฉิน
      if(emer==0){
        EmergencyStop();
        if(KeY=='A') Lcd_Stockpiling_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีจัดเก็บ
        else if(KeY=='B') Lcd_Fetching_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำออก
        delay(1000);
        digitalWrite(MShaft1, LOW);
        digitalWrite(MShaft2, HIGH);
      }
      if(digitalRead(LimShaftPin) == LOW){  //ตรวจสอบเงื่อนไขมีการกดของสวิตช์จำกัดระยะ
        count1++;
        if(count1 == Shaftlocation){
          delay(50);
          digitalWrite(MShaft1, HIGH);
          digitalWrite(MShaft2, HIGH);
          count1 = 0;
```

```
        A:
        break;
       }
      delay(100);
     }
    }
   }
}
void ShaftBackward(){  //ฟังก์ชันการเคลื่อนที่กลับของฟังก์ชัน
 if(w==0){
  delay(1000);
  digitalWrite(MShaft1, HIGH);
  digitalWrite(MShaft2, LOW);
  while(1){
   key = keypad.getKey();
   if(key=='B'){Stop(); KeY=key; x=6; goto A;}  //ตรวจสอบเงื่อนไขมีการเปลี่ยนแบบวิธี
   if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขมีการกดปุ่มหยุดฉุกเฉิน
   if(emer==0){
    EmergencyStop();
    if(KeY=='A') Lcd_Stockpiling_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีจัดเก็บ
    else if(KeY=='B') Lcd_Fetching_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำออก
    delay(1000);
    digitalWrite(MShaft1, HIGH);
    digitalWrite(MShaft2, LOW);
   }
   if(digitalRead(LimShaftPin) == LOW){  //ตรวจสอบเงื่อนไขมีการกดของสวิตช์จำกัดระยะ
    count1++;
    if(count1 == Shaftlocation){
     delay(50);
     digitalWrite(MShaft1, HIGH);
     digitalWrite(MShaft2, HIGH);
     count1 = 0;
```

```
      A:
      break;
     }
     delay(100);
    }
   }
  }
}
void BoardForward(){  //ฟังก์ชันการเคลื่อนที่เข้าคลังของแท่นขนย้ายสินค้า
 if(w==0){
  delay(1000);
  digitalWrite(MBoard1, LOW);
  digitalWrite(MBoard2, HIGH);
  while(1){
   key = keypad.getKey();
   if(key=='B'){Stop(); KeY=key; x=3; goto A;}  //ตรวจสอบเงื่อนไขมีการเปลี่ยนแบบวิธี
   if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขมีการกดปุ่มหยุดฉุกเฉิน
   if(emer==0){
    EmergencyStop();
    if(KeY=='A') Lcd_Stockpiling_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีเก็บ
    else if(KeY=='B') Lcd_Fetching_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำออก
    delay(1000);
    digitalWrite(MBoard1, LOW);
    digitalWrite(MBoard2, HIGH);
   }
   if(digitalRead(LimBoardPin) == LOW){  //ตรวจสอบเงื่อนไขมีการกดของสวิตช์จำกัดระยะ
    count2++;
    if(count2 == Boardlocation){
     delay(50);
     digitalWrite(MBoard1, HIGH);
     digitalWrite(MBoard2, HIGH);
     count2 = 0;
```

```
        A:
        break;
       }
       delay(400);
     }
    }
  }
}
void BoardBackward(){  //ฟังก์ชันการเคลื่อนที่ออกคลังของแท่นขนย้ายสินค้า
 if(w==0){
   delay(1000);
   digitalWrite(MBoard1, HIGH);
   digitalWrite(MBoard2, LOW);
   while(1){
    key = keypad.getKey();
    if(key=='B'){Stop(); KeY=key; x=4; goto A;}  //ตรวจสอบเงื่อนไขมีการเปลี่ยนแบบวิธี
    if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขมีการกดปุ่มหยุดฉุกเฉิน
    if(emer==0){
     EmergencyStop();
     if(KeY=='A') Lcd_Stockpiling_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีจัดเก็บ
     else if(KeY=='B') Lcd_Fetching_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำออก
     delay(1000);
     digitalWrite(MBoard1, HIGH);
     digitalWrite(MBoard2, LOW);
    }
    if(digitalRead(LimBoardPin) == LOW){  //ตรวจสอบเงื่อนไขมีการกดของสวิตช์จำกัดระยะ
     count2++;
     if(count2 == Boardlocation){
      delay(50);
      digitalWrite(MBoard1, HIGH);
      digitalWrite(MBoard2, HIGH);
      count2 = 0;
```

```
        A:
        break;
       }
       delay(400);
     }
    }
  }
}
void LiftUp(){  //ฟังก์ชันการเคลื่อนที่ขึ้นของลิฟต์
  if(Liftstate==1&&w==0){
    delay(1000);
    digitalWrite(MLift1, LOW);
    digitalWrite(MLift2, HIGH);
    while(1){
     key = keypad.getKey();
     if(key=='B'){Stop(); KeY=key; x=2; goto A;}  //ตรวจสอบเงื่อนไขมีการเปลี่ยนแบบวิธี
     if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขมีการกดปุ่มหยุดฉุกเฉิน
     if(emer==0){
      EmergencyStop();
      if(KeY=='A') Lcd_Stockpiling_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีจัดเก็บ
      else if(KeY=='B') Lcd_Fetching_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำออก
      delay(1000);
      digitalWrite(MLift1, LOW);
      digitalWrite(MLift2, HIGH);
     }
     if(digitalRead(LimliftUpPin) == LOW){  //ตรวจสอบเงื่อนไขมีการกดของสวิตช์จำกัดระยะ
      digitalWrite(MLift1, HIGH);
      digitalWrite(MLift2, HIGH);
      A:
      break;
     }
    }
```

```
 }
}
void LiftDown(){  //ฟังก์ชันการเคลื่อนที่ลงของลิฟต์
 if(Liftstate==1&&w==0){
  delay(1000);
  digitalWrite(MLift1, HIGH);
  digitalWrite(MLift2, LOW);
  while(1){
   key = keypad.getKey();
   if(key=='B'){Stop(); KeY=key; x=5; goto A;}  //ตรวจสอบเงื่อนไขมีการเปลี่ยนแบบวิธี
   if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขมีการกดปุ่มหยุดฉุกเฉิน
   if(emer==0){
    EmergencyStop();
    if(KeY=='A') Lcd_Stockpiling_Run();  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีจัดเก็บ
    else if(KeY=='B') Lcd_Fetching_Run();//ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำออก
    delay(1000);
    digitalWrite(MLift1, HIGH);
    digitalWrite(MLift2, LOW);
   }
   if(digitalRead(LimliftDownPin) == LOW){  //ตรวจสอบเงื่อนไขการกดของสวิตช์จำกัดระยะ
    digitalWrite(MLift1, HIGH);
    digitalWrite(MLift2, HIGH);
    A:
    break;
   }
  }
 }
}
```

```
void EmergencyStop(){  //ฟังก์ชันหยุดการทำงานฉุกเฉิน
 Lcd_EmergencyStop();
 digitalWrite(MLift1, HIGH);
```

```
 digitalWrite(MLift2, HIGH);
 digitalWrite(MShaft1, HIGH);
 digitalWrite(MShaft2, HIGH);
 digitalWrite(MBoard1, HIGH);
 digitalWrite(MBoard2, HIGH);
 delay(550);
 while(1){
  if(digitalRead(19)==LOW){ emer=1; break;}
 }
}
void Stop(){  //ฟังก์ชันหยุดการทำงาน
 digitalWrite(MLift1, HIGH);
 digitalWrite(MLift2, HIGH);
 digitalWrite(MShaft1, HIGH);
 digitalWrite(MShaft2, HIGH);
 digitalWrite(MBoard1, HIGH);
 digitalWrite(MBoard2, HIGH);
 w=1;
}
void re(){ //ฟังก์ชันสำหรับกลับไปทำงานต่อ
 if(x==1){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน ShaftForward()
  ShaftForward();
  LiftUp();
  BoardForward();
  Put_Item();
  Switch();
  Memory();
  Lcd_Stockpiling_Mode();
  BoardBackward();
  LiftDown();
  ShaftBackward();
 }
```

```
if(x==2){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน LiftUp()
 LiftUp();
 BoardForward();
 Put_Item();
 Switch();
 Memory();
 Lcd_Stockpiling_Mode();
 BoardBackward();
 LiftDown();
 ShaftBackward();
}
if(x==3){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน BoardForward()
 BoardForward();
 Put_Item();
 Switch();
 Memory();
 Lcd_Stockpiling_Mode();
 BoardBackward();
 LiftDown();
 ShaftBackward();
}
if(x==4){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน BoardBackward()
 BoardBackward();
 LiftDown();
 ShaftBackward();
}
if(x==5){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน LiftDown()
 LiftDown();
 ShaftBackward();
}
if(x==6){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน ShaftBackward()
 ShaftBackward();
```

```
 }
}
void prcout(){  //ฟังก์ชันสำหรับนำสินค้าชิ้นที่กำลังจัดเก็บออก หลังมีการเปลี่ยนแบบวิธี
 if(x==1){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน ShaftForward()
  Shaftlocation=count1;
  count1=0;
  ShaftBackward();
  Put_Item();
 }
 if(x==2){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน liftUp()
  LiftDown();
  ShaftBackward();
  Put_Item();
 }
 if(x==3){  //ตรวจสอบเงื่อนไขว่ามีการขัดจังหวะมาจากฟังก์ชัน BoardForward()
  Boardlocation=count2;
  count2=0;
  BoardBackward();
  LiftDown();
  ShaftBackward();
  Put_Item();
 }
}
```

```
void GetNo(){  //ฟังก์ชันใส่ค่าจำนวนสินค้าที่ต้องการนำออก
 int num, a;  //กำหนดตัวแปรรับค่าตัวเลขและตัวแปรบอกลำดับของตัวเลข
 a=0;
 while(1){
  num=0;
  if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขว่ามีการกดปุ่มหยุดฉุกเฉิน
  if(emer==0){EmergencyStop(); delay(500); Lcd_Fetching_Mode();}
  char key = keypad.getKey();
```

```
  if(key=='C') break;  //ตรวจสอบเงื่อนไขว่ามีการกดยืนยัน
  else if(key=='D'){  //ตรวจสอบเงื่อนไขว่ามีการกดล้างค่าจำนวนสินค้านำออก
   a=0;
   itemout=0;
   lcd.setCursor(0,2);
   lcd.print("Enter Number:   ");
  }
  else if((key=='1'||key=='2'||key=='3'||key=='4'||key=='5'||key=='6'||key=='7'||key=='8'||
  key=='9')&&a==0){  //ตรวจสอบเงื่อนไขว่ามีการกดตัวเลข 1 ถึง 9 และเป็นตัวเลขตัวที่หนึ่ง
   a=1;
   num=atoi(&key);
   itemout=num;
   Allitemout=itemout;
   lcd.setCursor(14,2);
   lcd.print(itemout);
  }
  else if((key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5'||key=='6'||key=='7'||key=='8'||
  key=='9')&&a==1){  //ตรวจสอบเงื่อนไขว่ามีการกดตัวเลข 0 ถึง 9 และเป็นตัวเลขตัวที่สอง
   a=0;
   num=atoi(&key);
   itemout=(itemout*10)+num;
   Allitemout=itemout;
   lcd.setCursor(14,2);
   lcd.print(itemout);
  }
 }
}
```

```
void Lcd_Mode_Selection(){  //ฟังก์ชันแสดงหน้าจอการเลือกแบบวิธี
 lcd.clear();
 lcd.setCursor(3,0);
 lcd.print("Mode Selection");
```

```
  lcd.setCursor(0,1);
  lcd.print("For Stockpiling: [A]");
  lcd.setCursor(0,2);
  lcd.print("For Fetching   : [B]");
}
void Lcd_Stockpiling_Mode(){  //ฟังก์ชันแสดงหน้าจอแบบวิธีจัดเก็บสินค้า
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Stockpiling Mode");
  lcd.setCursor(0,1);
  lcd.print("Item no.: ");
  lcd.print(memTotal);
  lcd.print("/18 pcs");
  lcd.setCursor(0,2);
  lcd.print("Waiting for item ...");
  if(memTotal==18){  //ตรวจสอบเงื่อนไขว่ามีสินค้าเต็มคลัง
    lcd.setCursor(0,2);
    lcd.print("Full Item!!");
  }
  lcd.setCursor(0,3);
  lcd.print("For Fetching:[B]");
}
void Lcd_Stockpiling_Run(){  //ฟังก์ชันแสดงหน้าจอขณะจัดเก็บสินค้า
  lcd.clear();
  lcd.setCursor(3,0);
  lcd.print("Stockpiling...");
  lcd.setCursor(0,1);
  lcd.print("Item no.: ");
  lcd.print(memTotal);
  lcd.print("/18 pcs");
  if(memTotal==18){  //ตรวจสอบเงื่อนไขว่ามีสินค้าเต็มคลัง
    lcd.setCursor(0,2);
```

```
    lcd.print("Full Item!!");
  }
  lcd.setCursor(0,3);
  lcd.print("For Fetching:[B]");
}
void Lcd_Fetching_Mode(){  //ฟังก์ชันแสดงหน้าจอแบบวิธีนำสินค้าออก
  lcd.clear();
  lcd.setCursor(3,0);
  lcd.print("Fetching Mode");
  lcd.setCursor(0,1);
  lcd.print("Item no.: ");
  lcd.print(memTotal);
  lcd.print("/18 pcs");
  lcd.setCursor(0,2);
  lcd.print("Enter Number: ");
  lcd.setCursor(0,3);
  lcd.print("OK:[C]   Clear:[D]");
}
void Lcd_Fetching_Run(){  //ฟังก์ชันแสดงหน้าจอขณะนำสินค้าออก
  lcd.clear();
  lcd.setCursor(4,0);
  lcd.print("Fetching...");
  lcd.setCursor(0,1);
  lcd.print("Item no.: ");
  lcd.print(memTotal);
  lcd.print("/18 pcs");
  lcd.setCursor(0,2);
  lcd.print("Fetched: ");
  lcd.print(Allitemout-itemout);
  lcd.print("/");
  lcd.print(Allitemout);
  lcd.print(" pcs");
```

```
}
void Lcd_EmergencyStop(){  //ฟังก์ชันแสดงหน้าจอขณะกดปุ่มหยุดฉุกเฉิน
 lcd.clear();
 lcd.setCursor(2,0);
 lcd.print("EMERGENCY STOP!");
 lcd.setCursor(5,2);
 lcd.print("To resume,");
 lcd.setCursor(3,3);
 lcd.print("press it again");
}
```

```
void loop() {
 while(1){  //วนรอบรอการเลือกแบบวิธี
  key = keypad.getKey();
  if(key=='A'||key=='B'){KeY=key; break;}
  if(digitalRead(Emergency) == LOW) emer=0;  //ตรวจสอบเงื่อนไขว่ามีการกดปุ่มหยุดฉุกเฉิน
  if(emer==0){EmergencyStop(); delay(500); Lcd_Mode_Selection();}
 }
 while(1){
  if(KeY=='A'){  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีจัดเก็บสินค้า
   Lcd_Stockpiling_Mode();
   while(w==0){
    key = keypad.getKey();
    if(key=='B'){KeY=key; break;}  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำสินค้าออก
    if(digitalRead(Emergency) == LOW) emer=0;
    //ตรวจสอบเงื่อนไขว่ามีการกดปุ่มหยุดฉุกเฉิน
    if(emer==0){EmergencyStop(); delay(500); Lcd_Stockpiling_Mode();}
    if(digitalRead(Sensor)==LOW&&memTotal<18){
    //ตรวจสอบเงื่อนไขว่าตัวรับรู้แบบใช้แสงพบสินค้าและมีสินค้าคงคลังน้อยกว่า 18 ชิ้น
     delay(5000);
     if(digitalRead(Sensor)==LOW){  //ตรวจสอบเงื่อนไขว่าตัวรับรู้แบบใช้แสงพบสินค้า
      Lcd_Stockpiling_Run();
```

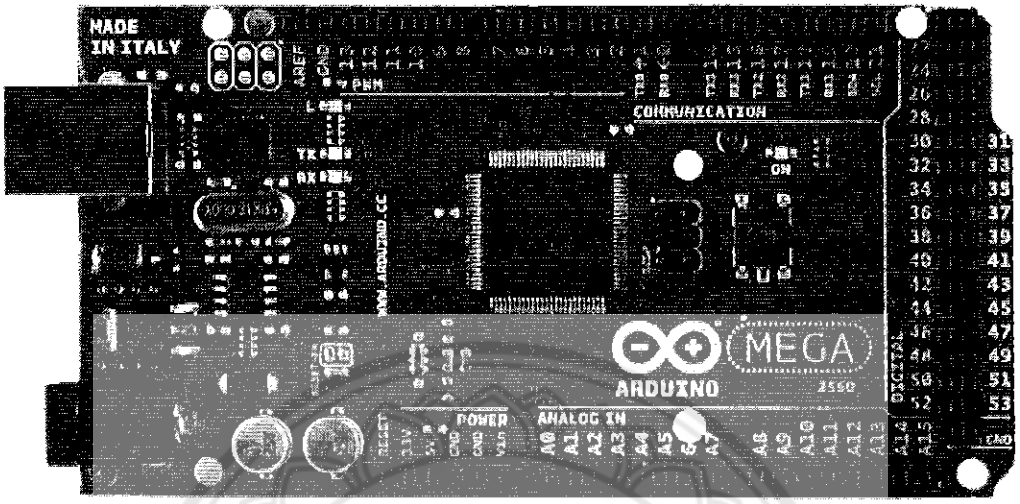
```
      readIn();
      Lift_Item();
      ShaftForward();
      LiftUp();
      BoardForward();
      Put_Item();
      Switch();
      Memory();
      Lcd_Stockpiling_Run();
      BoardBackward();
      LiftDown();
      ShaftBackward();
      Lcd_Stockpiling_Mode();
     }
    }
   }
}
if(KeY=='B'){  //ตรวจสอบเงื่อนไขว่าเป็นแบบวิธีนำสินค้าออก
 Lcd_Fetching_Mode();
 GetNo();
 if(w==1){  //ตรวจสอบเงื่อนไขว่ามีการเปลี่ยนจากแบบวิธีจัดเก็บเป็นแบบวิธีนำสินค้าออก
  w=0;
  if(x>3||itemout==0){
 //ตรวจสอบเงื่อนไขว่ามีการสั่งนำสินค้าออกขณะแท่นขนย้ายเคลื่อนที่กลับหรือไม่มีการใส่
  จำนวนสินค้า
   KeY='A';
   Lcd_Stockpiling_Run();
   re();
  }
  else if(itemout>0){  //ตรวจสอบเงื่อนไขว่ามีการใส่จำนวนออก
   Lcd_Fetching_Run();
   preout();
```

```
        itemout--;
        KeY='A';
       }
      }
      while(itemout>0&&memTotal>0){
      //ทำงานวนรอบเมื่อมีจำนวนสินค้านำออกและสินค้าคงคลัง
       if(digitalRead(Sensor)==HIGH){  //ตรวจสอบเงื่อนไขว่าตัวรับรู้แบบใช้แสงไม่พบสินค้า
        Lcd_Fetching_Run();
        readOut();
        ShaftForward();
        LiftUp();
        BoardForward();
        Lift_Item();
        itemout--;
        Switch();
        Memory();
        Lcd_Fetching_Run();
        BoardBackward();
        LiftDown();
        ShaftBackward();
        Put_Item();
        if(itemout==0)KeY='A';
       }
      }
     }
    }
   }
```

ภาคผนวก ข

รายละเอียดข้อมูลของ Arduino MEGA 2560

Arduino MEGA 2560



CE

Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications — Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials — Page 6

Terms & Conditions — Page 7

Enviromental Policies
half sqm of green via Impatto Zero® — Page 7

radiospares RADIONICS

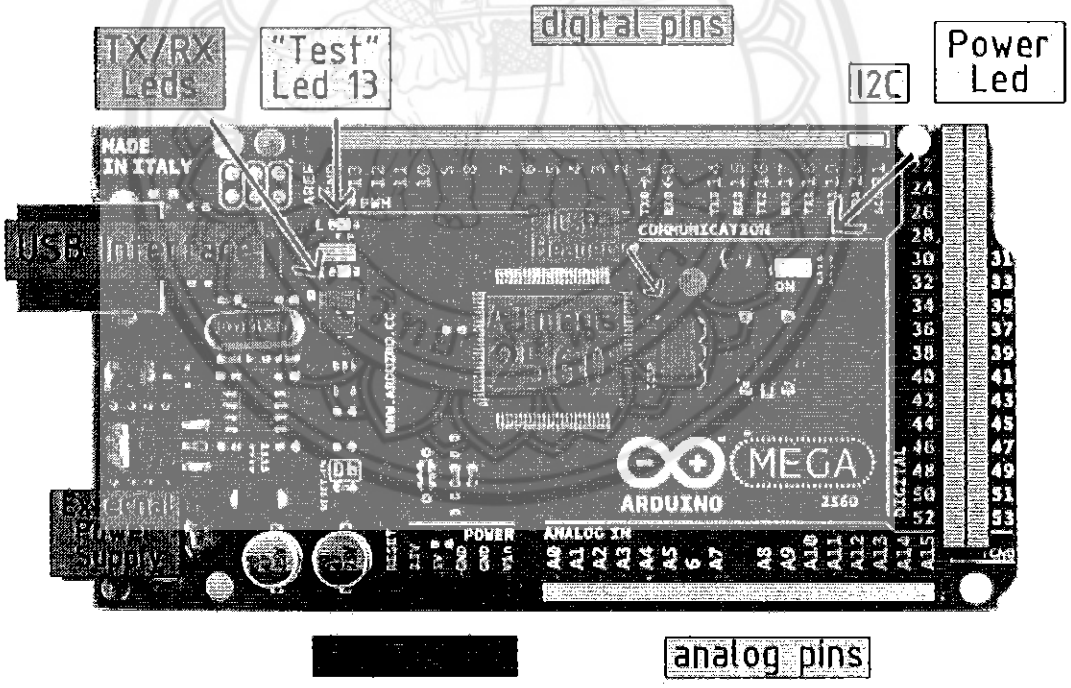
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

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

the board



radiospares RADIONICS

Power

The Arduino Mega2560 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 Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

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.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

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, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

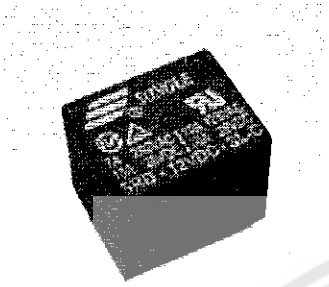
- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

RS radiospares RADIONICS

ภาคผนวก ค
รายละเอียดข้อมูลของรีเลย์

SONGLE RELAY

松乐继电器 ®SONGLE RELAY	RELAY ISO9002	SRD



1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.

(Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

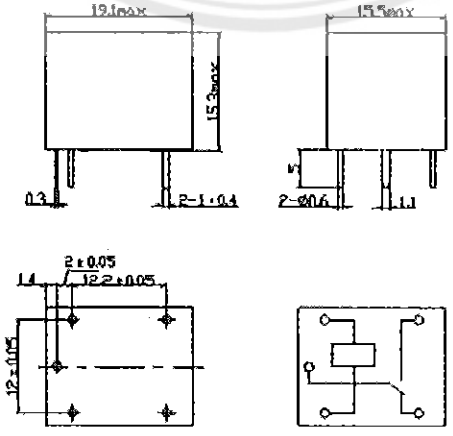
3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03、05、06、09、12、24、48VDC	S:Sealed type	L:0.36W	A:1 form A
				B:1 form B
		F:Flux free type	D:0.45W	C:1 form C

4. RATING

CCC	FILE NUMBER:CH0052885-2000	7A/240VDC
CCC	FILE NUMBER:CH0036748-99	10A/250VDC
UL /CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

5. DIMENSION(unit:mm) DRILLING(unit:mm) WIRING DIAGRAM



6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) ±10%	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
	48	48	7.5	6400				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.51W			

7. CONTACT RATING

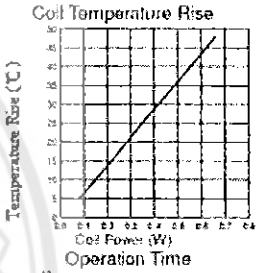
Item \ Type	SRD	
	FORM C	FORM A
Contact Capacity Resistive Load (cosΦ=1)	7A 28VDC 10A 125VAC 7A 240VAC	10A 28VDC 10A 240VAC
Inductive Load (cosΦ=0.4 L/R=7msec)	3A 120VAC 3A 28VDC	5A 120VAC 5A 28VDC
Max. Allowable Voltage	250VAC/110VDC	250VAC/110VDC
Max. Allowable Power Force	800VAC/240W	1200VA/300W
Contact Material	AgCdO	AgCdO

8. PERFORMANCE (at initial value)

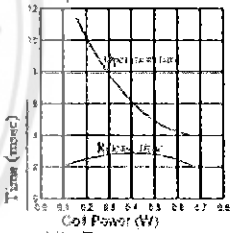
Item \ Type	SRD
Contact Resistance	100mΩ Max.
Operation Time	10msec Max.
Release Time	5msec Max.
Dielectric Strength Between coil & contact Between contacts	 1500VAC 50/60HZ (1 minute) 1000VAC 50/60HZ (1 minute)
Insulation Resistance	100 MΩ Min. (500VDC)
Max. ON/OFF Switching Mechanically Electrically	 300 operation/min 30 operation/min
Ambient Temperature	-25°C to +70°C
Operating Humidity	45 to 85% RH
Vibration Endurance Error Operation	 10 to 55Hz Double Amplitude 1.5mm 10 to 55Hz Double Amplitude 1.5mm
Shock Endurance Error Operation	 100G Min. 10G Min.
Life Expectancy Mechanically Electrically	 10^7 operations. Min. (no load) 10^5 operations. Min. (at rated coil voltage)
Weight	abt. 10grs.

9.REFERENCE DATA

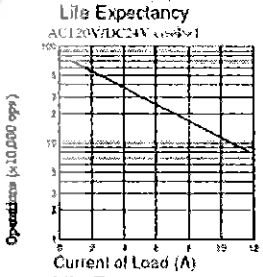
Coil Temperature Rise



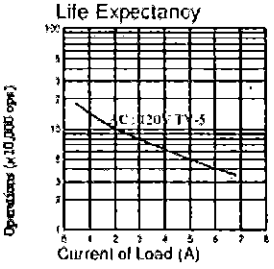
Operation Time



Life Expectancy



Life Expectancy



ภาคผนวก ง

รายละเอียดข้อมูลของไอซีคุมค่าแรงดัน รุ่น LM7805

µA7800 SERIES
POSITIVE-VOLTAGE REGULATORS

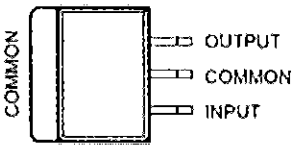
SLVS054J – MAY 1976 – REVISED MAY 2003

- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection
- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

KC (TO-220) PACKAGE
(TOP VIEW)

KCS (TO-220) PACKAGE
(TOP VIEW)

KTE PACKAGE
(TOP VIEW)



description/ordering information

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

ORDERING INFORMATION

T_J	$V_{O(NOM)}$ (V)	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 125°C	5	POWER-FLEX (KTE)	Reel of 2000	µA7805CKTER	µA7805C
		TO-220 (KC)	Tube of 50	µA7805CKC	µA7805C
		TO-220, short shoulder (KCS)	Tube of 20	µA7805CKCS	
	8	POWER-FLEX (KTE)	Reel of 2000	µA7808CKTER	µA7808C
		TO-220 (KC)	Tube of 50	µA7808CKC	µA7808C
		TO-220, short shoulder (KCS)	Tube of 20	µA7808CKCS	
	10	POWER-FLEX (KTE)	Reel of 2000	µA7810CKTER	µA7810C
		TO-220 (KC)	Tube of 50	µA7810CKC	µA7810C
	12	POWER-FLEX (KTE)	Reel of 2000	µA7812CKTER	µA7812C
		TO-220 (KC)	Tube of 50	µA7812CKC	µA7812C
		TO-220, short shoulder (KCS)	Tube of 20	µA7812CKCS	
	15	POWER-FLEX (KTE)	Reel of 2000	µA7815CKTER	µA7815C
		TO-220 (KC)	Tube of 50	µA7815CKC	µA7815C
		TO-220, short shoulder (KCS)	Tube of 20	µA7815CKCS	
	24	POWER-FLEX (KTE)	Reel of 2000	µA7824CKTER	µA7824C
		TO-220 (KC)	Tube of 50	µA7824CKC	µA7824C

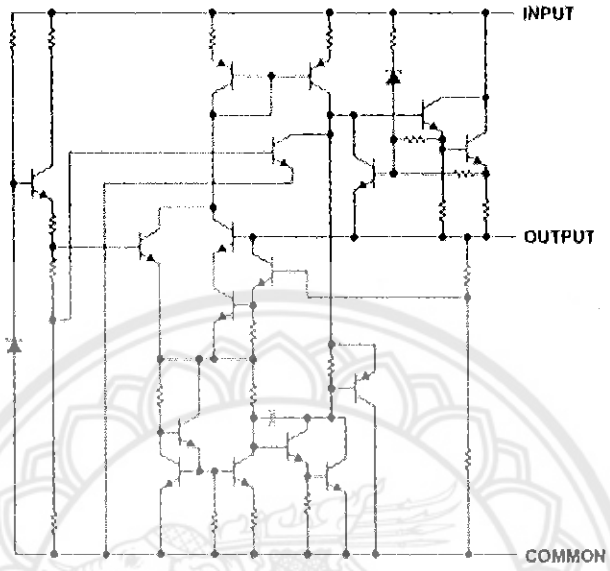
† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2003, Texas Instruments Incorporated

schematic



absolute maximum ratings over virtual junction temperature range (unless otherwise noted)†

Input voltage, V_I: μA7824C	40 V
All others	35 V
Operating virtual junction temperature, T_J	150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	–65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

package thermal data (see Note 1)

PACKAGE	BOARD	θ_{JC}	θ_{JA}
POWER-FLEX (KTE)	High K, JESD 51-5	3°C/W	23°C/W
TO-220 (KC/KCS)	High K, JESD 51-5	3°C/W	19°C/W

NOTE 1: Maximum power dissipation is a function of $T_J(max)$, θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.

recommended operating conditions

			MIN	MAX	UNIT
V_I	Input voltage	μA7805C	7	25	V
		μA7808C	10.5	25	
		μA7810C	12.5	28	
		μA7812C	14.5	30	
		μA7815C	17.5	30	
		μA7824C	27	38	
I_O	Output current			1.5	A
T_J	Operating virtual junction temperature	μA7800C series	0	125	°C

electrical characteristics at specified virtual junction temperature, V_I = 10 V, I_O = 500 mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J†	μA7805C			UNIT
			MIN	TYP	MAX	
Output voltage	I_O = 5 mA to 1 A, V_I = 7 V to 20 V, $P_D \le 15$ W	25°C	4.8	5	5.2	V
		0°C to 125°C	4.75		5.25	
Input voltage regulation	V_I = 7 V to 25 V	25°C		3	100	mV
	V_I = 8 V to 12 V			1	50	
Ripple rejection	V_I = 8 V to 18 V, f = 120 Hz	0 C to 125°C	62	78		dB
Output voltage regulation	I_O = 5 mA to 1.5 A	25°C		15	100	mV
	I_O = 250 mA to 750 mA			5	50	
Output resistance	f = 1 kHz	0 C to 125°C		0.017		Ω
Temperature coefficient of output voltage	I_O = 5 mA	0°C to 125°C		–1.1		mV/°C
Output noise voltage	f = 10 Hz to 100 kHz	25 C		40		μV
Dropout voltage	I_O = 1 A	25 C		2		V
Bias current		25°C		4.2	8	mA
Bias current change	V_I = 7 V to 25 V	0°C to 125°C			1.3	mA
	I_O = 5 mA to 1 A				0.5	
Short-circuit output current		25°C		750		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

μA7800 SERIES
POSITIVE-VOLTAGE REGULATORS

SLVS056J – MAY 1976 – REVISED MAY 2003

electrical characteristics at specified virtual junction temperature, V_I = 14 V, I_O = 500 mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J†	μA7808C			UNIT
			MIN	TYP	MAX	
Output voltage	I_O = 5 mA to 1 A, V_I = 10.5 V to 23 V, $P_D \le$ 15 W	25°C	7.7	8	8.3	V
		0°C to 125°C	7.6		8.4	
Input voltage regulation	V_I = 10.5 V to 25 V	25°C		6	160	mV
	V_I = 11 V to 17 V			2	80	
Ripple rejection	V_I = 11.5 V to 21.5 V, f = 120 Hz	0°C to 125°C	55	72		dB
Output voltage regulation	I_O = 5 mA to 1.5 A	25°C		12	160	mV
	I_O = 250 mA to 750 mA			4	80	
Output resistance	f = 1 kHz	0°C to 125°C		0.016		Ω
Temperature coefficient of output voltage	I_O = 5 mA	0°C to 125°C		–0.8		mV/°C
Output noise voltage	f = 10 Hz to 100 kHz	25°C		52		μV
Dropout voltage	I_O = 1 A	25°C		2		V
Bias current		25°C		4.3	8	mA
Bias current change	V_I = 10.5 V to 25 V	0°C to 125°C			1	mA
	I_O = 5 mA to 1 A				0.5	
Short-circuit output current		25°C		450		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

electrical characteristics at specified virtual junction temperature, V_I = 17 V, I_O = 500 mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J†	μA7810C			UNIT
			MIN	TYP	MAX	
Output voltage	I_O = 5 mA to 1 A, V_I = 12.5 V to 25 V, $P_D \le$ 15 W	25°C	9.6	10	10.4	V
		0°C to 125°C	9.5	10	10.5	
Input voltage regulation	V_I = 12.5 V to 28 V	25°C		7	200	mV
	V_I = 14 V to 20 V			2	100	
Ripple rejection	V_I = 13 V to 23 V, f = 120 Hz	0°C to 125°C	55	71		dB
Output voltage regulation	I_O = 5 mA to 1.5 A	25°C		12	200	mV
	I_O = 250 mA to 750 mA			4	100	
Output resistance	f = 1 kHz	0°C to 125°C		0.018		Ω
Temperature coefficient of output voltage	I_O = 5 mA	0°C to 125°C		–1		mV/°C
Output noise voltage	f = 10 Hz to 100 kHz	25°C		70		μV
Dropout voltage	I_O = 1 A	25°C		2		V
Bias current		25°C		4.3	8	mA
Bias current change	V_I = 12.5 V to 28 V	0°C to 125°C			1	mA
	I_O = 5 mA to 1 A				0.5	
Short-circuit output current		25°C		400		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

electrical characteristics at specified virtual junction temperature, V_I = 19 V, I_O = 500 mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J†	μA7812C			UNIT
			MIN	TYP	MAX	
Output voltage	I_O = 5 mA to 1 A, V_I = 14.5 V to 27 V, $P_D \le$ 15 W	25°C	11.5	12	12.5	V
		0°C to 125°C	11.4		12.6	
Input voltage regulation	V_I = 14.5 V to 30 V	25°C		10	240	mV
	V_I = 16 V to 22 V			3	120	
Ripple rejection	V_I = 15 V to 25 V, f = 120 Hz	0°C to 125°C	55	71		dB
Output voltage regulation	I_O = 5 mA to 1.5 A	25°C		12	240	mV
	I_O = 250 mA to 750 mA			4	120	
Output resistance	f = 1 kHz	0°C to 125°C		0.018		Ω
Temperature coefficient of output voltage	I_O = 5 mA	0°C to 125°C		–1		mV/°C
Output noise voltage	f = 10 Hz to 100 kHz	25°C		75		μV
Dropout voltage	I_O = 1 A	25°C		2		V
Bias current		25°C		4.3	8	mA
Bias current change	V_I = 14.5 V to 30 V	0°C to 125°C			1	mA
	I_O = 5 mA to 1 A				0.5	
Short-circuit output current		25°C		350		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

electrical characteristics at specified virtual junction temperature, V_I = 23 V, I_O = 500 mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J†	μA7815C			UNIT
			MIN	TYP	MAX	
Output voltage	I_O = 5 mA to 1 A, V_I = 17.5 V to 30 V, $P_D \le$ 15 W	25°C	14.4	15	15.6	V
		0°C to 125°C	14.25		15.75	
Input voltage regulation	V_I = 17.5 V to 30 V	25°C		11	300	mV
	V_I = 20 V to 26 V			3	150	
Ripple rejection	V_I = 18.5 V to 28.5 V, f = 120 Hz	0°C to 125°C	54	70		dB
Output voltage regulation	I_O = 5 mA to 1.5 A	25°C		12	300	mV
	I_O = 250 mA to 750 mA			4	150	
Output resistance	f = 1 kHz	0°C to 125°C		0.019		Ω
Temperature coefficient of output voltage	I_O = 5 mA	0°C to 125°C		–1		mV/°C
Output noise voltage	f = 10 Hz to 100 kHz	25°C		90		μV
Dropout voltage	I_O = 1 A	25°C		2		V
Bias current		25°C		4.4	8	mA
Bias current change	V_I = 17.5 V to 30 V	0°C to 125°C			1	mA
	I_O = 5 mA to 1 A				0.5	
Short-circuit output current		25°C		230		mA
Peak output current		25°C		2.1		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

SLVS056J – MAY 1976 – REVISED MAY 2003

electrical characteristics at specified virtual junction temperature, V_I = 33 V, I_O = 500 mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J†	μA7824C			UNIT
			MIN	TYP	MAX	
Output voltage	I_O = 5 mA to 1 A, V_I = 27 V to 38 V, $P_D \le 15$ W	25°C	23	24	25	V
		0°C to 125°C	22.8		25.2	
Input voltage regulation	V_I = 27 V to 38 V	25°C		18	480	mV
	V_I = 30 V to 36 V			6	240	
Ripple rejection	V_I = 28 V to 38 V, f = 120 Hz	0°C to 125°C	50	66		dB
Output voltage regulation	I_O = 5 mA to 1.5 A	25°C		12	480	mV
	I_O = 250 mA to 750 mA			4	240	
Output resistance	f = 1 kHz	0°C to 125°C		0.028		Ω
Temperature coefficient of output voltage	I_O = 5 mA	0°C to 125°C		−1.5		mV/°C
Output noise voltage	f = 10 Hz to 100 kHz	25°C		170		μV
Dropout voltage	I_O = 1 A	25°C		2		V
Bias current		25°C		4.6	8	mA
Bias current change	V_I = 27 V to 38 V	0°C to 125°C			1	mA
	I_O = 5 mA to 1 A				0.5	
Short-circuit output current		25°C		150		mA
Peak output current		25°C		2.1		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายนรินทร์ มีบุญ
ภูมิลำเนา 76 หมู่ 11 ต.แม่ต๋ำ อ.พญาเม็งราย จ.เชียงราย
ประวัติการศึกษา

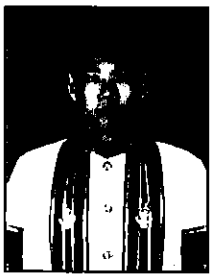
- จบระดับมัธยมศึกษาจากโรงเรียนแม่ต๋ำตาดควันวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **duang-28537@hotmail.com**

ชื่อ นายจีรวัฒน์ ช่างเก็บ
ภูมิลำเนา 88 หมู่ 9 ต.แม่ต๋ำ อ.พญาเม็งราย จ.เชียงราย
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนแม่ต๋ำตาดควันวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **jeerawat3673@gmail.com**



ชื่อ นายกฤษฎิ์ สวัสดี
ภูมิลำเนา 154 หมู่ 10 ต.วังซ่าน อ.แม่วงก์ จ.นครสวรรค์
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนนครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **k.sawatdee55@gmail.com**