

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมการจัดเก็บและรับ-ส่งสิ่งของในคลังสินค้า

Control system in the warehouse

โดย

1. นาย วิชชา จำเมือง 48015307
2. นาย วีรชัย ชีมเจริญ 48015310
3. นาย ศราวุธ สารวิงษ์ 48015312

๕๗.

๖ ๕๓๘ ๖

๒๕๕๐

เลขหมู่.....
เลขทะเบียน..... **82007**
วัน,เดือน,ปี..... **- 4 ก.ก. 2551**

b..... **11913968**
i.....

**ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท

ปีการศึกษา 2550

ภาควิชา

วิศวกรรมระบบควบคุม

คณะ

วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

ระบบควบคุมการจัดเก็บและรับ-ส่งสิ่งของในคลังสินค้า

Control system in the warehouse

ผู้จัดทำ

1. นาย วิชชา จำเมือง 48015307

2. นาย วีรชัย ชิมเจริญ 48015310

3. นาย ศราวุธ สาริวงษ์ 48015312

(อาจารย์ สุมิตร พนาอุดมทรัพย์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมการจัดเก็บและรับ-ส่งสิ่งของในคลังสินค้า

นาย วิชชา จำเมือง 48015307

นาย วีรชัย ชิมเจริญ 48015310

นาย ศราวุธ สารวิงษ์ 48015312

อาจารย์ที่ปรึกษา

อาจารย์ สุมิตร พนาอุดมทรัพย์

ปีการศึกษา 2550

บทคัดย่อ

ปริญญาานิพนธ์นี้เป็นการศึกษาระบบของคลังสินค้าเพื่อประยุกต์ใช้กับระบบคลังสินค้าอัตโนมัติ เพื่อที่จะทำการจัดเก็บ-นำสินค้าออกจากคลัง และการหมุนเวียนสินค้าอย่างมีประสิทธิภาพตามระบบที่ว่า สินค้าที่เข้าสู่คลังก่อนจะต้องหมุนเวียนออกจากคลังก่อนสินค้าที่เข้ามาสู่คลังภายหลัง โดยปริญญาานิพนธ์นี้จะประกอบด้วยส่วนของฮาร์ดแวร์ที่เป็นส่วนของตัวลิฟต์ยกสินค้า รวมถึงคลังสินค้าจำลอง และส่วนของซอฟต์แวร์ที่แบ่งออกเป็น 2 ส่วนคือ ส่วนแรกใช้ในการควบคุมการทำงานของตัวลิฟต์ยกสินค้า ซึ่งใช้ไมโครคอนโทรลเลอร์ MCS-51เบอร์ 8052 ส่วนที่สอง คือส่วนที่ใช้ในการจัดเก็บฐานข้อมูล และแสดงผลติดต่อกับผู้ใช้ ซึ่งในส่วนนี้จะใช้โปรแกรม Visual Basic 6.0

Abstract

This Project is to study about the Warehouse System, which apply to the Automatic Warehouse control. This system will provide an effective process of storage and rotating stocks as a First in First out (FIFO) system. Furthermore, this system will furnish a database of merchandise in stock, which enables the merchant to evaluate his/her need to order a new lot of merchandise. The project is composed of body warehouse and crane for lifting, simulated warehouse, microcontroller hardware, database and Visual Basic Program in User Interface

Part

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณบิดามารดาของพวกเราทุกคนที่ให้กำเนิดเลี้ยงดู และให้การศึกษาแก่พวกเรา

ขอขอบพระคุณอาจารย์สุมิตร พนาอุดมทรัพย์ ที่ให้คำแนะนำ คำปรึกษา และให้ความรู้ความช่วยเหลือแก่พวกเราทุกคนตลอดเวลา

ขอขอบพระคุณท่านอาจารย์ทุกๆท่าน และสถาบันแห่งนี้ที่ให้ความรู้ และอบรมสั่งสอนพวกเรามาโดยตลอด

ขอขอบพระคุณพี่ๆห้องธุรการ และห้องสโตร์ ที่ช่วยอำนวยความสะดวกเรื่องต่างๆโดยเฉพาะการเบิกเงิน และอุปกรณ์ต่างๆ

ขอขอบคุณเพื่อนๆทุกคนที่ให้กำลังใจและช่วยเหลือมาตลอด ทั้งเรื่องโปรแกรมและอุปกรณ์ต่างๆ

ขอขอบคุณเพื่อนนို့ดที่ช่วยทางด้านการเขียนโปรแกรมไมโครคอนโทรลเลอร์
ขอขอบคุณคอมพิวเตอร์ที่แสนจะเอาใจยากและสิ่งอำนวยความสะดวกให้พวกเราทำงานสบาย

ขอขอบคุณปริญญา นิพนธ์ของรุ่นพี่ๆที่เป็นต้นแบบให้ดู(Copy) บางส่วน

สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญภาพ	III
สารบัญตาราง	V
บทที่ 1 บทนำ	1
บทที่ 2 โครงสร้างและการออกแบบโครงการ	2
2.1 หลักการทำงานเบื้องต้น	2
2.2 การออกแบบและการสร้าง	2
2.3 ระบบSoftware	13
2.3.1 ระบบเซนเซอร์	14
2.3.2 ระบบขับเคลื่อน	17
2.3.3 ลิมิทสวิตช์	19
2.4 การออกแบบการควบคุมคลังสินค้าโดยใช้โปรแกรมVisual Basic	20
2.4.1 วิธีการออกแบบการควบคุมโดยVisual Basic 6.0	20
2.4.2 บอร์ดควบคุมรีเลย์อเนกประสงค์	21
2.4.3 รูปแสดงหน้าต่าง Visual Basic และ โปรแกรม	24
บทที่ 3 ผลการทดลอง	28
บทที่ 4 บทสรุปและวิจารณ์	30
4.1 ส่วนของHardware และ Software	30
4.2 แนวทางในการพัฒนา	30
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

สารบัญญภาพ

	หน้า
รูปที่ 1 ลักษณะของการวางสินค้าในคลังสินค้าแบบทั่วไป	3
รูปที่ 2 คลังสินค้าที่ออกแบบใหม่	4
รูปที่ 3 แสดงขนาดของฐาน 6 เหลี่ยม	5
รูปที่ 4 ขนาดของผนังที่ใช้ทำช่องคลังสินค้า	6
รูปที่ 5 เมื่อนำมาประกอบกับฐานรอง 6 เหลี่ยม	6
รูปที่ 6 แสดงถึงช่องที่ใส่แกนเหล็ก	7
รูปที่ 7 แสดงถึงการติดตั้งมอเตอร์และเฟือง	7
รูปที่ 8 แสดงเมื่อใส่สายพานแล้ว	8
รูปที่ 9 แสดงถึงการติดตั้งlimit switch	8
รูปที่ 10 แสดงการติดตั้งมอเตอร์และเฟือง	9
รูปที่ 11 แสดงถึงลักษณะของ ตัวรับ-ส่งสินค้า	9
รูปที่ 12 แสดงคลังสินค้าที่เสร็จแล้วเรียบร้อย	10
รูปที่ 13 ไฟล์ชาร์ตการควบคุมกรณีการส่งและรับสินค้า	12
รูปที่ 14 วงจรควบคุมรวมทั้งหมดของคลังสินค้าบางส่วน	13
รูปที่ 15 แสดงลักษณะของเซนเซอร์ TCRT 1000/1010	14
รูปที่ 16 ไอซี LM324N	15
รูปที่ 17 วงจรต่อเซนเซอร์ใช้งาน	16
รูปที่ 18 แสดงการต่อวงจรขั้วมอเตอร์กับ ไมโครคอนโทรลเลอร์ MCS-51	17
รูปที่ 19 ไอซี L293D	18
รูปที่ 20 แสดงการต่อสวิตช์เมตริกซ์	19
รูปที่ 21 โครงสร้างของปุ่มควบคุม	20
รูปที่ 22 หน้าต่างของปุ่มควบคุมเมื่อrun โปรแกรมแล้ว	21
รูปที่ 23 แสดงการทดลองกดปุ่มบนโปรแกรม	21
รูปที่ 24 วงจรควบคุมการทำงานของรีเลย์	22
รูปที่ 25 ส่วนควบคุมทางรีเลย์	22
รูปที่ 26 ส่วนควบคุมทางรีเลย์เมื่อrun โปรแกรมแล้ว	23
รูปที่ 27 ส่วนควบคุม โดย Visual Basic 6.0 ที่เสร็จสมบูรณ์	27

สารบัญภาพ(ต่อ)

	หน้า
รูปที่ 28 ส่วนควบคุมโดย Visual Basic 6.0	28
รูปที่ 29 ลำดับการทำงานของคลังสินค้า	29



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 1 การกำหนดค่า En, Input, Output

หน้า

18



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันนี้ประเทศของเรากำลังจะพัฒนาจากประเทศเกษตรกรรมไปสู่ประเทศอุตสาหกรรม ซึ่งระบบคลังสินค้าถือได้ว่าเป็นระบบที่สำคัญระบบหนึ่งในกระบวนการทางอุตสาหกรรม เนื่องจากระบบคลังสินค้านี้มีส่วนเกี่ยวข้องกับการจัดเก็บสินค้า การขนส่งสินค้า และการจัดจำหน่ายสินค้า และเพื่อให้ระบบต่างๆในกระบวนการทางอุตสาหกรรมสามารถดำเนินไปอย่างมีประสิทธิภาพนั้น ก็ขึ้นอยู่กับปัจจัยต่างๆเกี่ยวกับการจัดระบบคลังสินค้า ไม่ว่าจะเป็นความถูกต้องแม่นยำในการจัดเก็บสินค้า ความรวดเร็วในการจัดเก็บ แต่อย่างไรก็ตาม ในปัจจุบันยังไม่มีหรือนำเอาระบบคอมพิวเตอร์ที่ใช้ในการตรวจสอบปริมาณสินค้าคงคลัง มาใช้ร่วมกับการจัดเก็บสินค้าในคลัง ทั้งนี้อาจมีสาเหตุหลายประการ เช่น ระบบนี้อาจมีค่าใช้จ่ายสูง และอาจยังไม่สามารถลบความคิดในการควบคุมระบบจัดเก็บสินค้าแบบเก่าได้ (ระบบคลังสินค้าแบบใช้แรงงานคนในการจัดเก็บสินค้า) ดังนั้นจึงจำเป็นที่ควรจะมีการส่งเสริมให้มีแผนพัฒนาการใช้งานระบบคลังสินค้าอัตโนมัติในระบบอุตสาหกรรมมากขึ้น วิธีหนึ่งที่สามารถปฏิบัติได้ นั่นคือ นำที่จะมีการสร้างสรรค์ผลงานที่เกี่ยวข้องกับด้านนี้นำเสนอต่อธุรกิจอุตสาหกรรมมากขึ้น เมื่อเกิดผู้พัฒนาผลิตภัณฑ์นำร่องออกมา

ปัญหาที่เกี่ยวกับการขนส่งและจัดเก็บสินค้าก็เป็นสาเหตุหนึ่งที่ทำให้ต้นทุนในการผลิตสินค้าเพิ่ม ตัวอย่างของปัญหาเหล่านี้คือจะขนถ่ายสินค้านั้นอย่างไรใช้อุปกรณ์ชนิดใดจึงจะเหมาะสม ต้องใช้คนงานจำนวนเท่าไรในการขนส่ง เป็นต้น

ส่วนอีกปัญหาหนึ่งสำหรับการจัดเก็บสินค้าคือ การจัดวางที่ไม่เป็นระเบียบ ทำให้ยากต่อการค้นหาและขนส่ง ยกตัวอย่างเช่น คลังสินค้าการบินไทยในแผนกขาเข้านั้น เมื่อมีสินค้าเข้ามาเบรกข้างใน ทำให้พื้นที่ในการเก็บน้อง ทำให้ต้องวางระเกะระกะไม่เป็นระเบียบและยากต่อการค้นหา เวลาไปเบิกสินค้าออกมาเพื่อที่จะจัดการส่งออกไป

โดยโครงการนี้จัดทำเพื่อลดปัญหาดังกล่าว เหมาะสมอย่างยิ่งที่จะนำไปใช้ในพื้นที่แคบโดยสามารถที่จะจัดเก็บสินค้าได้และง่ายต่อการค้นหาเพียงแค่สั่งการป้อนข้อมูลในคอมพิวเตอร์เท่านั้น ระบบก็จะนำสินค้าที่เราจัดเก็บไว้และ ต่ำเสียงออกมาตามที่เราต้องการ

บทที่ 2

โครงสร้างและการออกแบบโรงงาน

การออกแบบโรงงานแบ่งออกเป็น 2 ส่วนคือ

1. Hardware

2. Software

ในที่นี้จะกล่าวถึงระบบ **Hardware** ก่อน

2.1 หลักการทำงานเบื้องต้น

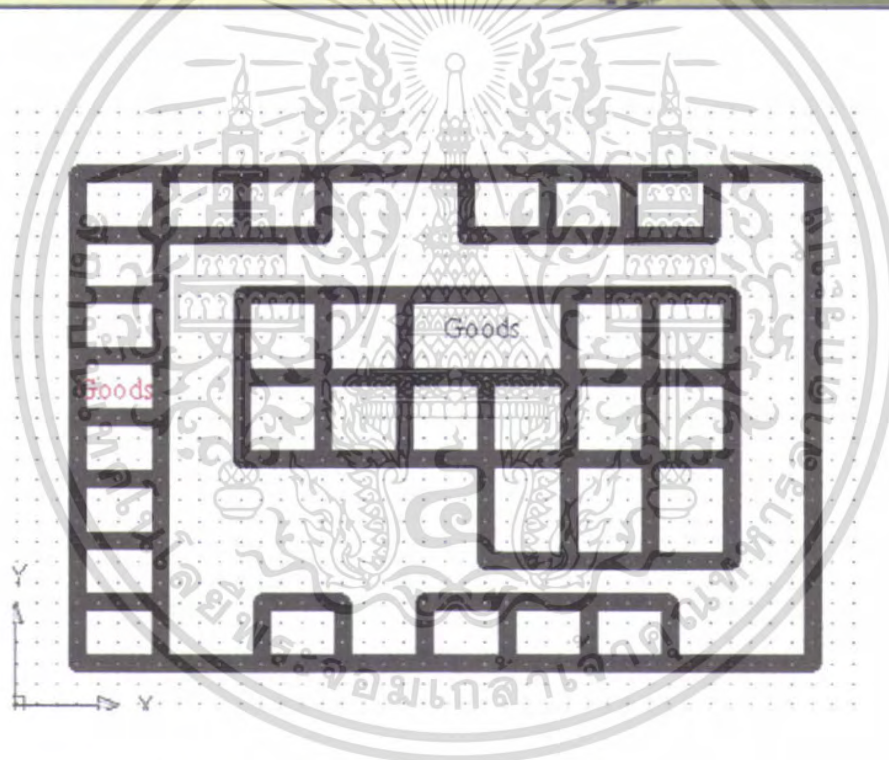
เมื่อมีสินค้าผ่านมายังระบบสายพานลำเลียงก่อนที่จะเข้าไปยังตัวคลังสินค้าจะมีเซนเซอร์คอยตรวจจับสินค้า เพื่อที่จะนำข้อมูลไปแสดงยังบนจอคอมพิวเตอร์ แสดงถึงรายละเอียดของสินค้านั้นๆ จากนั้นสินค้าก็จะถูกลำเลียงเข้าไปยังตัวรับ-ส่งสินค้าภายในคลังสินค้า เพื่อที่จะเตรียมพร้อมในการส่งสินค้าเข้าช่องเก็บเราทำการเลือกช่องเก็บและชั้นของมัน โดยส่งจากคอมพิวเตอร์โดยตรงจากนั้นตัวรับส่งสินค้าจะหมุนรอบตัวเองซึ่งจะมีระบบเซนเซอร์จับความเคลื่อนไหวอยู่ เมื่ออ่านเจอพื้นสีขาวแล้วจะหยุด จากนั้นก็ลำเลียงสินค้าเข้าไปชน **Limit switch** เพื่อสั่งให้มอเตอร์สายพานหยุดทำงานและสั่งให้ไปปรากฏบนจอคอมพิวเตอร์ด้วยว่าสินค้าถูกจัดเก็บเรียบร้อยแล้วพร้อมทั้งแสดงรายละเอียดของสินค้านั้นๆ ด้วย

2.2 การออกแบบและการสร้าง

ในโรงงานอุตสาหกรรมการผลิตส่วนใหญ่ จะผลิตสินค้า สิ่งของเครื่องใช้ต่างๆตามความต้องการของลูกค้า และเครื่องอุปโภคบริโภค เช่น อาหารสำเร็จรูป ยานพาหนะ อะไหล่อุปกรณ์ต่างๆ ชิ้นส่วนอิเล็กทรอนิกส์ ฯลฯ เป็นต้น ดังนั้น เมื่อสินค้าที่ผลิตได้จะต้องบรรจุลงหีบห่อหรือใส่ลังที่มีขนาดใหญ่ เพื่อที่จะได้นำส่งออกไปยังตามสถานที่ต่างๆตาม **Order** ของลูกค้าหรือเจ้าของสถานประกอบการที่สั่งเพื่อจัดจำหน่ายให้แก่ลูกค้าต่อไปในระบบการจัดวางสินค้าในโรงงานอุตสาหกรรมทั่วไปนั้น สังเกตได้ว่าจะมีการจัดวางเป็นชั้นๆเรียงติดต่อกันเป็นระยะทางที่ยาวๆ โดยที่จะมีรถโฟล์ทลิฟท์มายกสินค้าที่บรรจุลงใส่ลังเรียบร้อยแล้วออกไปวางยังที่ๆกำหนด ซึ่งส่วนใหญ่จะเป็นอย่างนี้ในทุกๆ โรงงานอุตสาหกรรมการผลิตดังรูปที่ 1

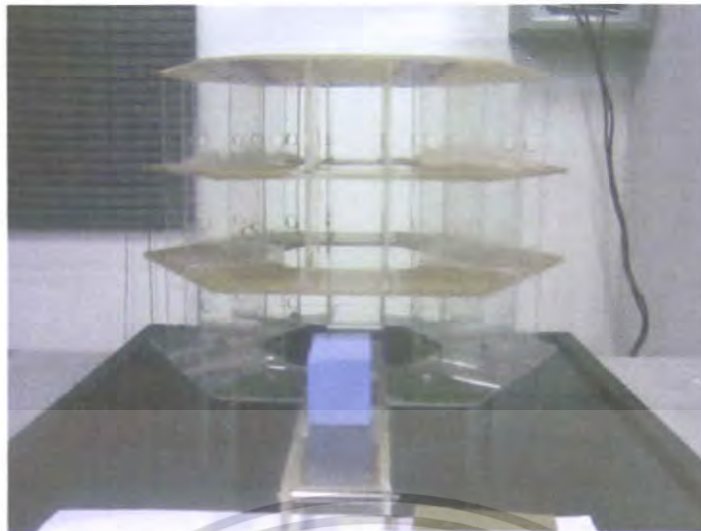
จากที่กล่าวมาการเคลื่อนย้ายสินค้าไปจัดเก็บไว้ในคลังสินค้าจะต้องเวลาและระยะทางที่มาก จะต้องขนสินค้าไปวางในพื้นที่ๆ วางเนื่องจากการวางสินค้าที่เรียงกันเป็นชั้นๆต่อกันไปจึงต้องคิดวิธีที่ว่า จะทำอย่างไรที่จะประหยัดเวลาและระยะทางในการขนถ่ายสินค้า เพื่อสะดวกในการจัดเก็บสินค้าเข้าชั้นหรือนำสินค้าออกมา ซึ่งคลังสินค้าที่ออกแบบใหม่นี้จะมีลักษณะดังรูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 ลักษณะของการวางสินค้าในคลังสินค้าแบบทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 กลังสินค้าที่ออกแบบใหม่

ในการออกแบบโครงสร้างกลังสินค้านี้จากที่กล่าวมาแล้วว่าระบบกลังสินค้าในโรงงานอุตสาหกรรมการผลิตส่วนใหญ่ จะเคลื่อนย้ายสินค้าไปจัดเก็บในพื้นที่ที่กำหนด ซึ่งมีลักษณะการวางเป็นชั้นๆและเรียงต่อกันไป ถ้าคิดดูแล้วมันจะเสียเวลาและระยะทางที่มาก จึงคิดว่าวิธีที่วิธีใดที่จะสามารถลำเลียงสินค้าเข้าที่โดยประหยัดทั้งเวลาและพื้นที่

ระบบกลังสินค้าที่ออกแบบใหม่นี้จะมีลักษณะเป็นทรงกระบอกจำนวน 3 ชั้น ชั้นละ 6 ช่อง รวมทั้งหมดจะมี 18 ช่อง แต่ว่าเนื่องจากชั้นแรกจะต้องเว้นช่องไว้เพื่อให้สามารถลำเลียงสินค้าเข้ามายังตัวรับส่งได้ จึงเหลือจำนวนช่องทั้งหมด 17 ช่อง ซึ่งกลังสินค้าที่มีลักษณะเป็นทรงกระบอกนี้ การจัดเก็บเข้า-ออกนั้น การเคลื่อนที่ของมันจะอยู่กับที่ อาศัยการหมุนรอบตัวเองและการเคลื่อนที่ขึ้นลงเท่านั้น โดยไม่มีการเคลื่อนที่ในระยะขจัดที่ไกลๆระบบกลังสินค้านี้จะทำงานด้วยคอมพิวเตอร์และควบคุมโดยไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

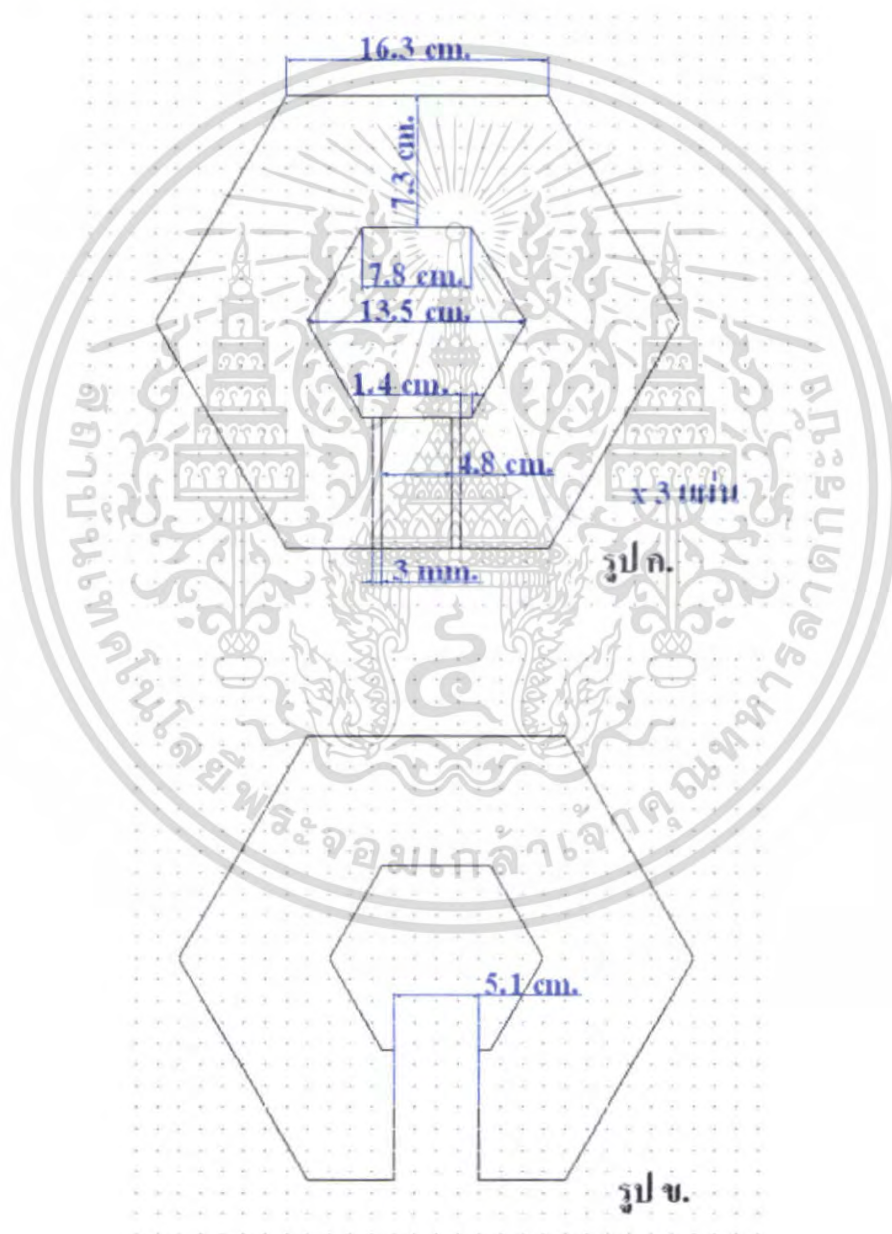
วิธีการออกแบบ

1.ออกแบบคลังสินค้าซึ่งมีลักษณะเป็นทรงกระบอกดังรูปที่ 2

2.ฐานรอง 6 เหลี่ยม

2.1) ตัดแผ่นอะคริลิกขนาด 3 mm.ให้เป็นรูป 6 เหลี่ยม จำนวน 3 แผ่นโดยขนาดทั้งหมดสังเกตุได้จากในรูปที่ 3 ก.

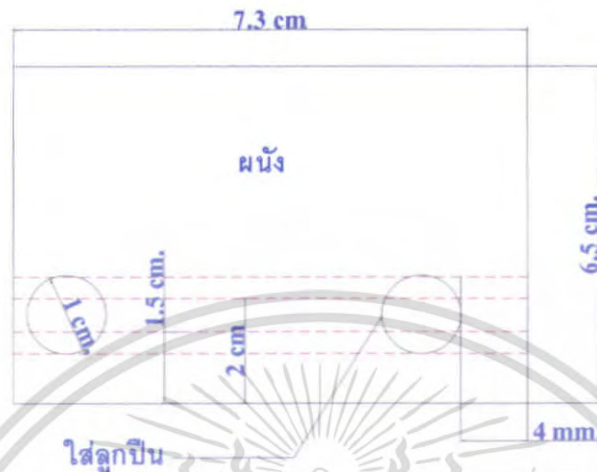
2.2) นำแผ่นอะคริลิก 1ใน 3 แผ่นนี้มาตัดช่อง 1 ด้านดังรูปที่ 3 ข. เพื่อเป็นช่องผ่านให้สามารถลำเลียงสินค้าผ่านเข้าไปได้



รูปที่ 3 แสดงขนาดของฐาน 6 เหลี่ยม

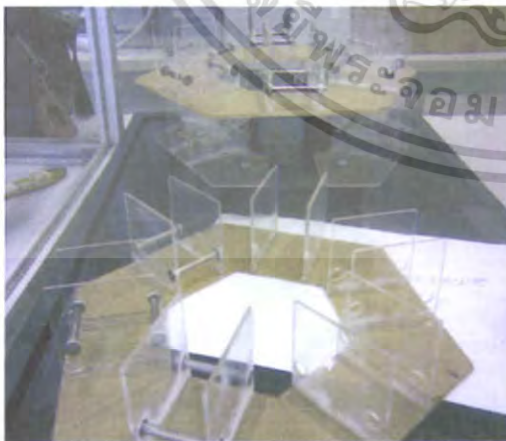
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตัดแผ่นอะคริลิกขนาด 3 mm. ให้เป็นรูปสี่เหลี่ยมผืนผ้าขนาดกว้าง 6.5 cm. ยาว 7.3 cm. เพื่อใช้ทำเป็นช่อง พร้อมทั้งเจาะรู 2 รู เพื่อที่จะสามารถใส่ลูกป็นได้ดังรูปที่ 4

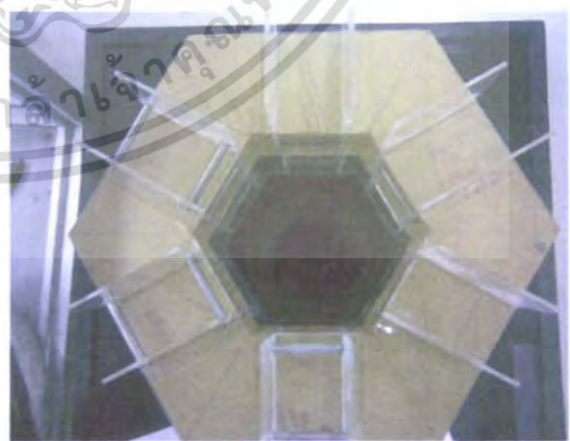


รูปที่ 4 ขนาดของแผ่นที่ใช้ทำช่องกลิ้งสินค้า

4. เมื่อตัดแผ่นอะคริลิกเป็นรูปสี่เหลี่ยมผืนผ้าและเจาะรูสำหรับใส่ลูกป็นเรียบร้อยแล้วก็นำมาประกอบรวมเข้ากับฐานรองของชั้น ซึ่งระยะห่างและความพอดีของช่องจะต้องเท่าๆกัน ถ้าไม่เท่ากันแล้วชั้นต่อไปก็จะไม่เท่ากันด้วย ดังรูปที่ 5



(ก)

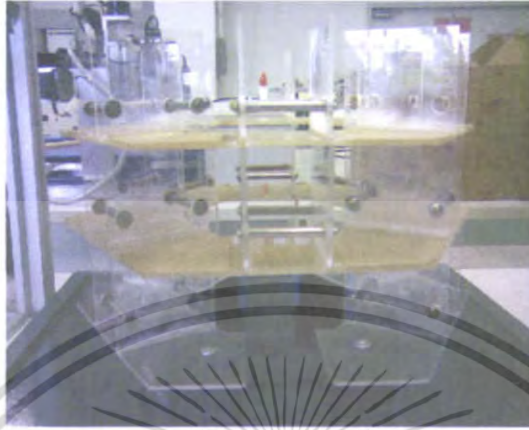


(ข)

รูปที่ 5 เมื่อนำมาประกอบกับฐานรอง 6 เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ใส่แกนเหล็ก 2 อันในช่องแต่ละช่องให้ครบทุกๆช่องดังรูปที่ 6



รูปที่ 6 แสดงถึงช่องที่ใส่แกนเหล็ก

6. ใส่เฟืองขับให้กับช่องสินค้า โดยติดกับแกนเหล็กอันหลังสุด พร้อมกับวางมอเตอร์ และทำตัวยึดมอเตอร์ให้อยู่กับที่ ดังรูปที่ 7



(ก)

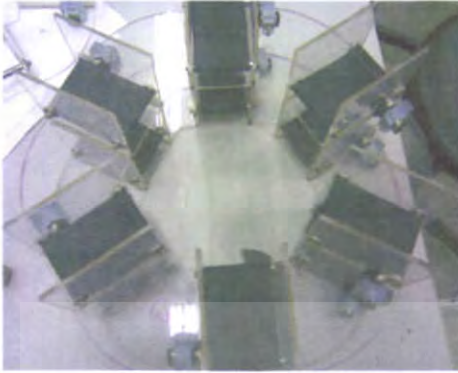


(ข)

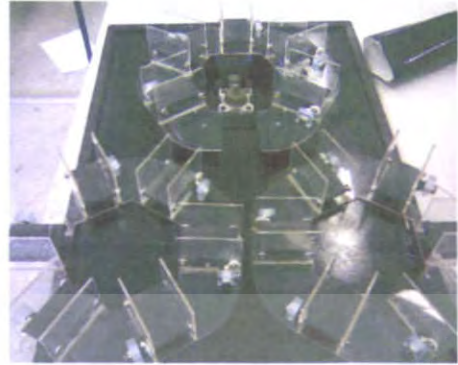
รูปที่ 7 แสดงถึงการติดตั้งมอเตอร์และเฟือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ใส່สายพานลำเลียงทุกๆช่อง ดังรูปที่ 8



(ก)



(ข)

รูปที่ 8 แสดงเมื่อใส່สายพานแล้ว

8. ติด limit switch เพื่อที่จะเป็นตัว off มอเตอร์สายพานและยังเป็นตัวที่ส่งข้อมูลว่าสินค้าได้ถูกจัดเก็บหรือลำเลียงออกไปเรียบร้อยแล้ว ดังรูปที่ 9



(ก)

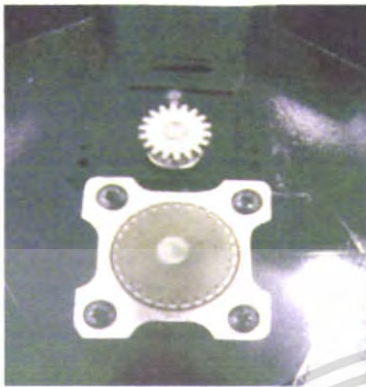


(ข)

รูปที่ 9 แสดงถึงการติดตั้ง limit switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. คิดตั้งระบบมอเตอร์เฟืองที่ฐานรองของตัวกลิ้งสินค้าเพื่อที่จะเป็นตัวหมุนตัวรับ-ส่งสินค้า ดังรูปที่10



(ก)



(ข)

รูปที่10 แสดงการติดตั้งมอเตอร์และเฟือง

10. ออกแบบและสร้างตัวรับส่งสินค้าดังรูปที่11



(ก)

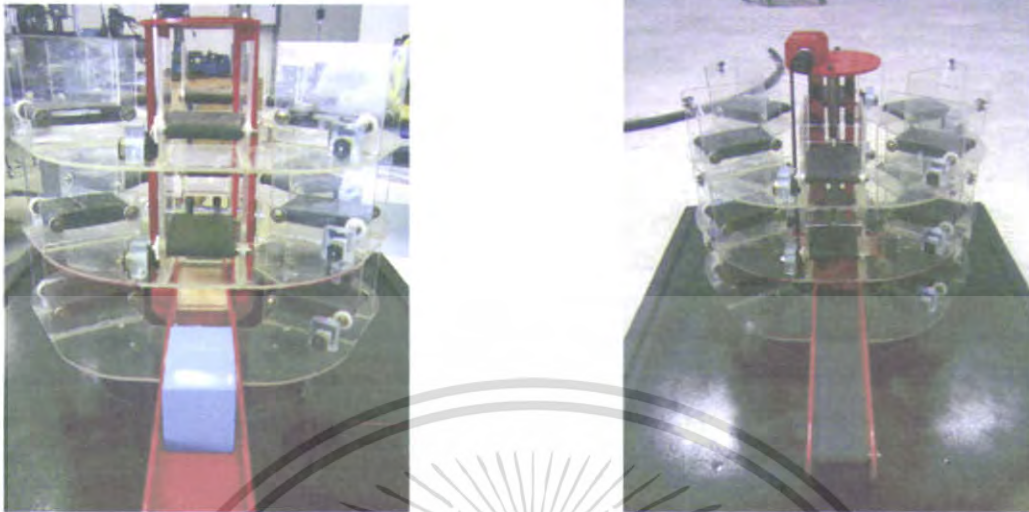


(ข)

รูปที่11-แสดงถึงลักษณะของ ตัวรับ-ส่งสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

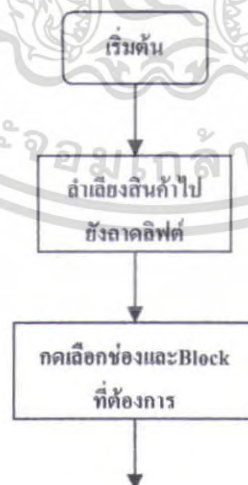
11. นำส่วนประกอบทั้งหมดมาประกอบเข้าด้วยกันเป็นอันเรียบร้อยสำหรับตัวHardwareดังรูปที่ 12



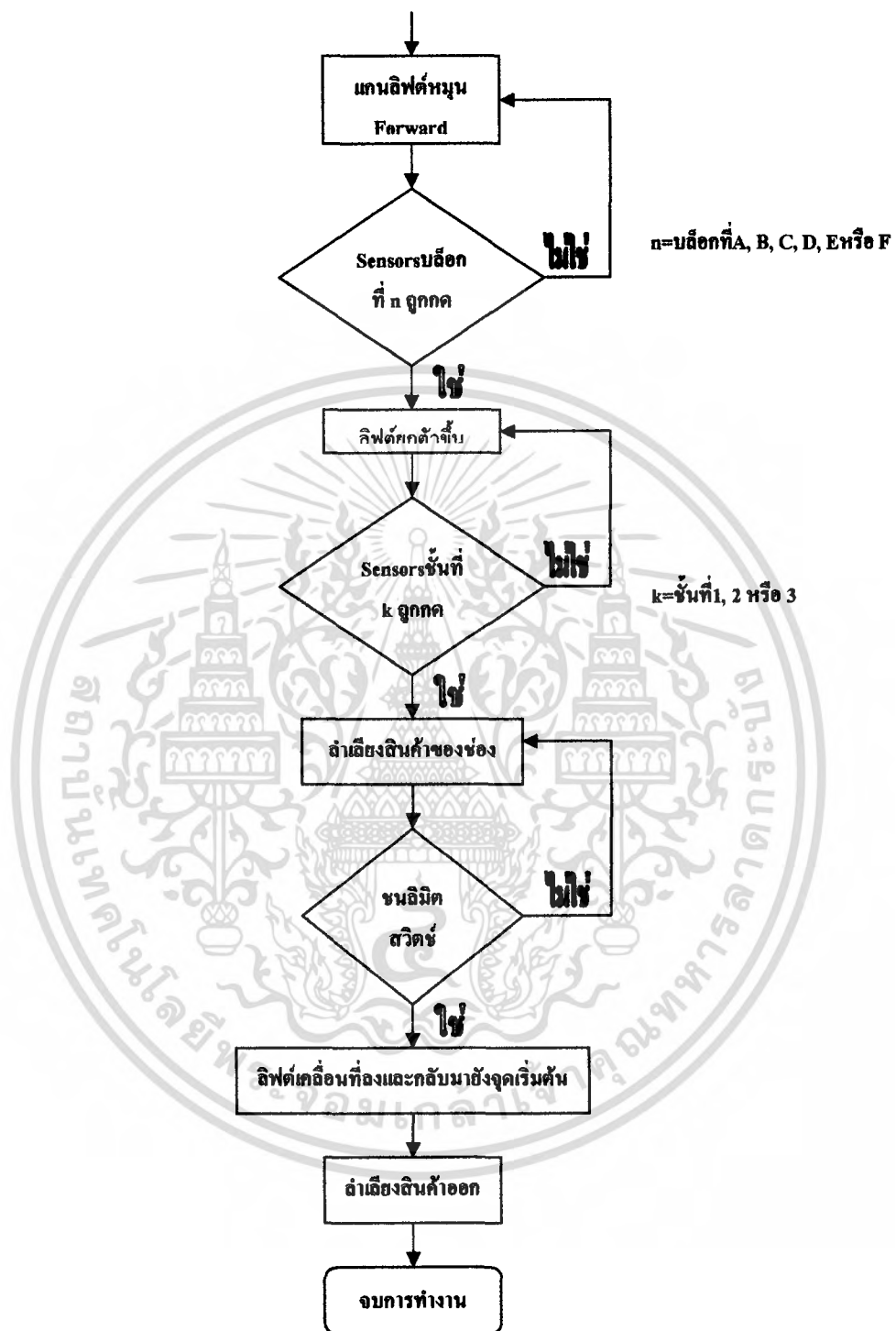
รูปที่12 แสดงคลังสินค้าที่เสร็จแล้วเรียบร้อย

การควบคุมการเคลื่อนที่ของลิฟต์ในคลังสินค้าเพื่อที่จะนำสินค้าไปจัดเก็บและการนำสินค้าออกมาแสดงได้ดัง โพลีชาร์ตดังต่อไปนี้

โพลีชาร์ตแสดงการทำงานในการจัดเก็บสินค้า

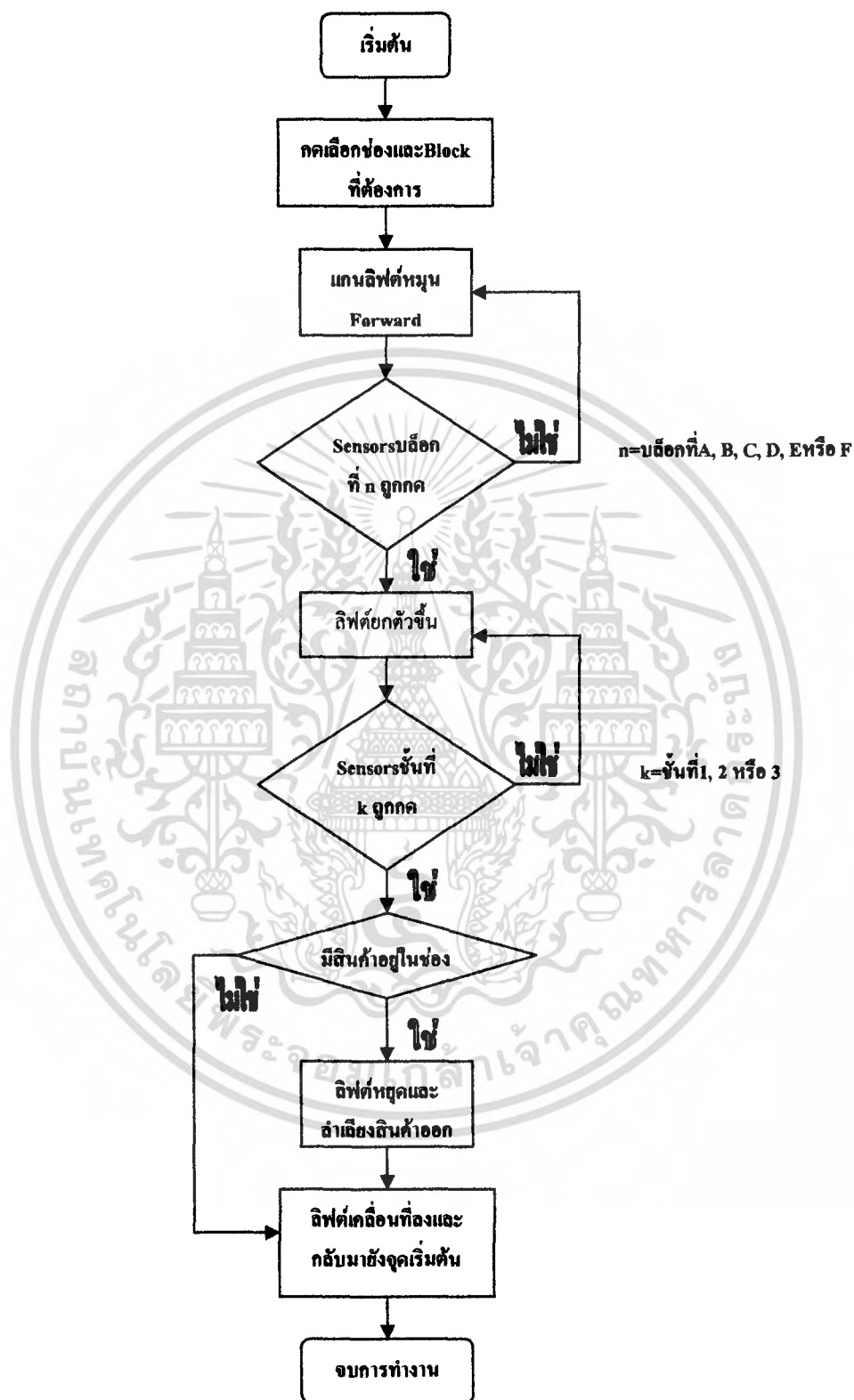


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลว์ชาร์ตแสดงการทำงานในการรับสินค้าออก

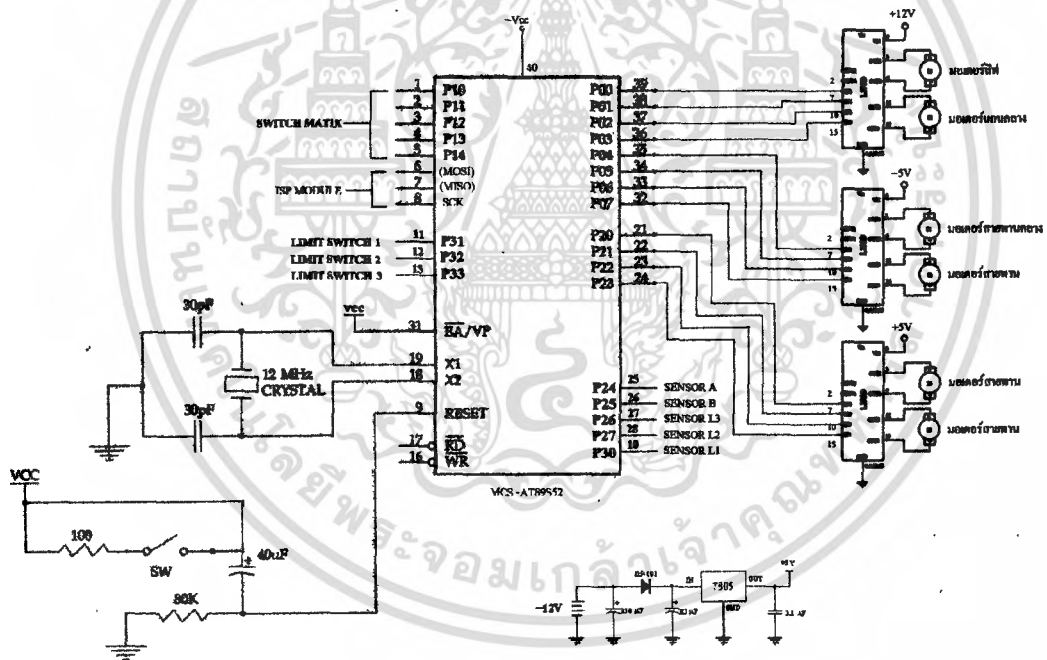


รูปที่ 13 โฟลว์ชาร์ตการควบคุมกรณีการจัดเก็บและรับสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบSoftware

ในที่นี้จะกล่าวถึงระบบควบคุมต่างๆในคลังสินค้าซึ่งในที่นี้เราจะใช้ไมโครคอนโทรลเลอร์MCS-51 ในการควบคุม โดยจะใช้ IC เบอร์AT89S52 ใช้งาน ซึ่งรายละเอียดต่างๆนั้นจะกล่าวในบทต่อไป ระบบควบคุมในคลังสินค้าแบ่งการควบคุมออกเป็น 3 ส่วน คือ1.ระบบเซนเซอร์ 2.ระบบขั้มมอเตอร์สายพาน 3.ระบบสวิตช์ โดยทั้ง 3 ระบบนั้นจะต้องเขียนโปรแกรมให้มันสัมพันธ์กัน จึงจะสามารถใช้งานได้ อย่างมีประสิทธิภาพ ซึ่งในการสังงานนั้น เราจะใช้โปรแกรมVisual Basic เข้ามาเกี่ยวข้องด้วย เพราะปัจจุบันนี้เทคโนโลยีได้ก้าวหน้าไปอย่างรวดเร็ว การสั่งการทุกอย่างจะต้องใช้คอมพิวเตอร์เป็นส่วนใหญ่ในการควบคุม และโปรแกรมภาษาC เป็นโปรแกรมที่ผู้ใช้สามารถเข้าใจได้ง่ายในการเขียนในระบบควบคุม คลังสินค้านั้น ใช้ไมโครคอนโทรลเลอร์เข้ามาประยุกต์ใช้ในการควบคุม ดังที่ได้กล่าวมาแล้วข้างต้น ดังนั้น จะขอเสนอระบบควบคุมต่างๆในคลังสินค้าแต่ละส่วนซึ่งนำมาประกอบเข้ากันโดยสมบูรณ์ ดังรูปที่ 14



รูปที่ 14 วงจรควบคุมรวมทั้งหมดของคลังสินค้าบางส่วน

จะรูปจะเห็นว่า มีส่วนควบคุมอยู่ 3 ส่วนใหญ่ๆคือ เซนเซอร์ ระบบขั้มมอเตอร์สายพาน ลิมิตสวิตช์ โดยจะอธิบายเป็นส่วนๆต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.) ระบบเซนเซอร์

เซนเซอร์เป็นอุปกรณ์ที่สำคัญในการทำงานของคลังสินค้า ซึ่งเซนเซอร์ที่เลือกใช้นั้น คือ เบอร์ TRCT1000/TRCT1010 เป็นเซนเซอร์ทำงานด้วยการส่งผ่านของแสงซึ่งมีส่วนประกอบคือแหล่งกำเนิดแสง (light source) และตัวตรวจจับแสง (light detector) ในที่นี้ LED ทำหน้าที่เป็นแหล่งกำเนิดแสง และ ไดทรานซิสเตอร์ซึ่งมีคุณสมบัติในการทำงานเปลี่ยนแปลงตามแสงที่ตกกระทบตัวมัน ทำหน้าที่เป็นตัวตรวจจับแสง ลำแสงที่ใช้เป็นตัวกลางในการส่งผ่านสื่อสารข้อมูลระหว่างตัวส่ง คือแหล่งกำเนิดแสงกับตัวตรวจจับ นิยมใช้สีแดงหรือแสงอินฟราเรด

หลักการทำงานคือ แหล่งกำเนิดแสง LED ทำการสร้างแสงและส่งผ่านไปยังวัตถุซึ่งเคลื่อนที่มาตรงกับตำแหน่งที่ได้ติดตั้งเซนเซอร์ไว้ ทำให้ลำแสงดังกล่าวตกกระทบกับวัตถุนั้นแล้วสะท้อนกลับมายังโฟโตไดทรานซิสเตอร์ เพื่อสร้างสัญญาณเอาต์พุตปรากฏให้ผู้ควบคุมกระบวนการทราบถึงตำแหน่งของวัตถุ

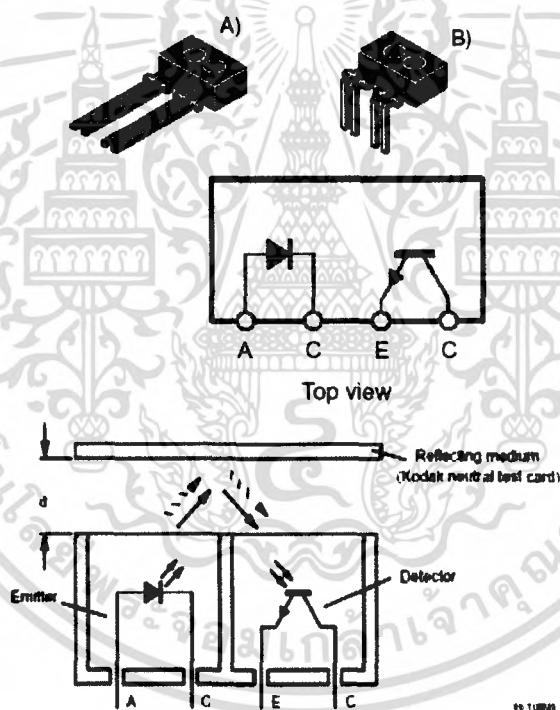


Figure 1 Test circuit

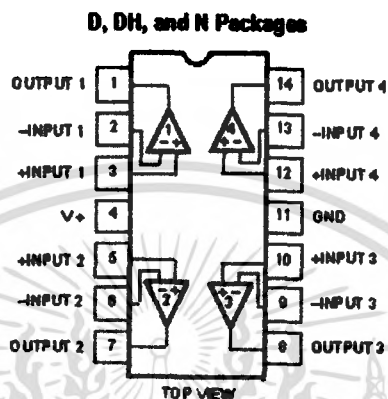
รูปที่ 15 แสดงลักษณะของเซนเซอร์ TCRT 1000/1010

สำหรับจำนวนเซนเซอร์ที่ใช้นั้นจะมีอยู่ด้วยกัน 9 ตัว ซึ่งคลังสินค้านั้นจะมีรูปแบบเป็นลักษณะทรงกระบอกมี 3 ชั้น และ ภายใน 1 ชั้นก็จะมียู่ 6 ช่อง เราได้ทำการติดตั้งเซนเซอร์ไว้ที่ชั้นแรก ถึง 6 ตำแหน่งเพื่อที่จะใช้ในการตรวจจับตำแหน่งตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

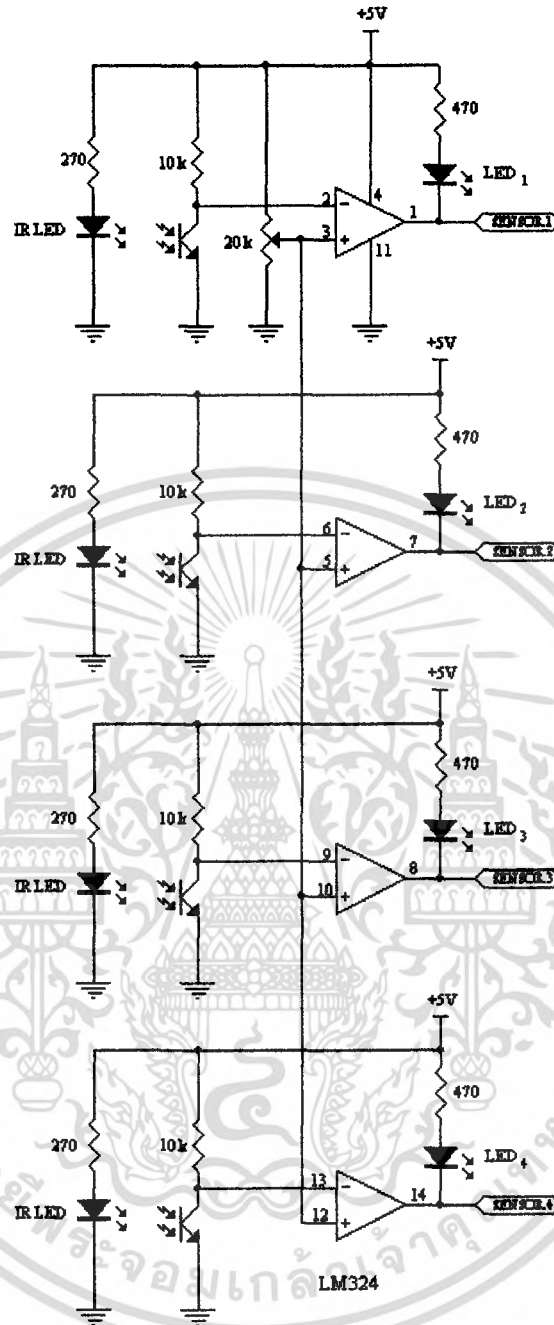
จากนั้นก็ติดตั้งเซนเซอร์ไว้วัดตรงเสาอีก 3 ตำแหน่ง เพื่อใช้ตรวจจับตำแหน่งของจัน ซึ่งในการเขียนโปรแกรมสั่งงาน เราจะใช้เซนเซอร์ทั้ง 9 ตัวนั้นเป็นหลักเพื่อใช้ในการเลือกช่องและชั้นที่ต้องการได้

ในการต่อวงจรควบคุมเซนเซอร์นั้น เราจะใช้ตัว IC เข้ามาต่อประยุกต์ใช้ ซึ่งเบอร์ที่ใช้คือ IC เบอร์ LM324N ดังรูป



รูปที่ 16 ไอซี LM324N

ภายใน IC-LM324N จะประกอบไปด้วยออปแอมป์ 4 ตัวด้วยกันบรรจุอยู่ในทำให้ภายใน 1 ชิปสามารถต่อเซนเซอร์ไว้วัดงานได้ถึง 4 ตัวด้วยกันดังรูป

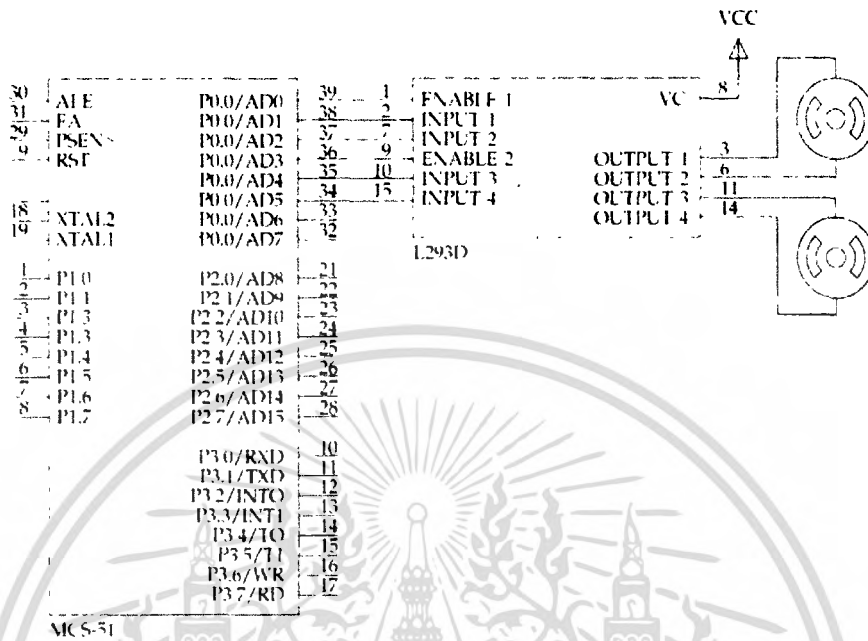


รูปที่ 17 วงจรต่อเซนเซอร์ใช้งาน

ภายในคลังสินค้านี้ใช้เซนเซอร์ 9 ตัว จำนวน IC ที่ใช้จึงเป็น 3 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.)ระบบขับเคลื่อนมอเตอร์

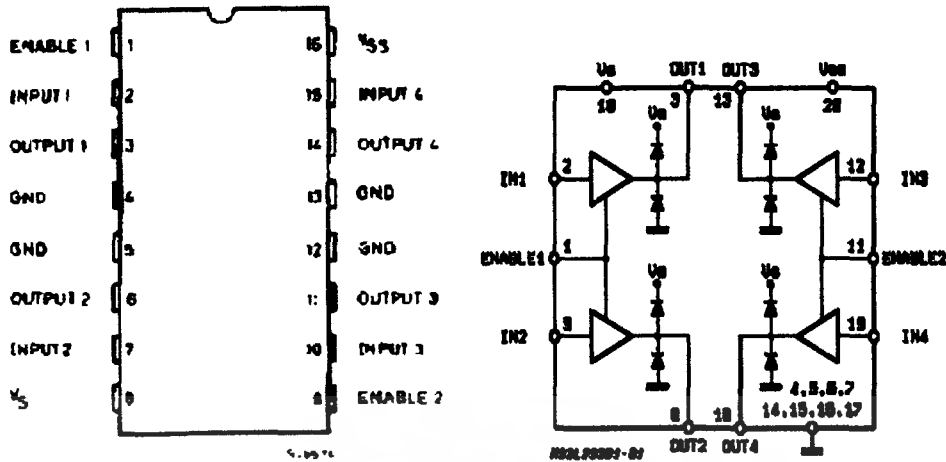


รูปที่ 18 แสดงการต่อวงจรขับเคลื่อนมอเตอร์กับไมโครคอนโทรลเลอร์MCS-51

ในระบบคลั่งสินค้านั้น อุปกรณ์หลักที่สำคัญอีกอย่างหนึ่งก็คือมอเตอร์ มอเตอร์ที่ใช้ในที่นี้คือ DC มอเตอร์ จะทำหน้าที่เปลี่ยนพลังงานจากพลังงานไฟฟ้าเป็นพลังงานกลโดยประกอบด้วย 2 ส่วนคือส่วนที่อยู่กับที่ (stator) และส่วนที่เคลื่อนที่ (rotor) ซึ่งเราจะใช้มอเตอร์อยู่ 2 ลักษณะคือ DC มอเตอร์ 12V. และ DC มอเตอร์ 3V. มอเตอร์ที่ใช้ในการขับเคลื่อนตามช่องคือ DC มอเตอร์ 3V. ส่วนที่ใช้ในการขับเคลื่อนลิฟต์และการหมุนจะใช้ DC มอเตอร์ 12V. และทั้งสองชนิดนี้จะต้องมีความเร็วรอบต่ำๆจึงจะสามารถใช้งานได้ดี เพื่อที่เซนเซอร์สามารถตรวจจับวัตถุได้ทัน

ในการต่อ DC มอเตอร์ไปใช้งานนั้นเราจะใช้ IC ต่อร่วมด้วยเหมือนกับเซนเซอร์ซึ่งตัว IC ที่ใช้คือ เมอร์ L293D ดังรูป

82007



รูปที่ 19 ไอซี L293D

IC293D จะมีอยู่ทั้งหมด 16 ขา ซึ่งภายใน IC 1 ตัวสามารถขับมอเตอร์ได้ถึง 4 ตัวเพราะมี Input/Output อย่างละ 2 ขา สามารถขับให้หมุนตามเข็มนาฬิกาหรือหมุนทวนเข็มนาฬิกาได้ จากรูปเป็นตัวอย่างของขา IC โดยจะมีขา Input และ Output อย่างละ 2 ขา เนื่องจาก IC นี้สามารถต่อมอเตอร์ DC ได้ทั้งหมด 4 ตัว เพื่อให้ได้ใช้งานได้ถูกต้องจึงมีหมายเลขกำกับไว้ เช่น input1, input2 สำหรับมอเตอร์ตัวที่ 1 หรือ input3, input4 สำหรับมอเตอร์ตัวที่ 2 เป็นต้นสำหรับการควบคุมการทำงาน กำหนดที่ขา EN, Input และ Output ซึ่งการกำหนดค่าของขาเหล่านี้แสดงได้ดังตารางต่อไปนี้

ตารางที่ 1 การกำหนดขา En, Input, Output

En(Enable)	Input	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

จากตาราง ขา En จะทำการ Enable IC L293D เมื่อป้อนลอจิกเป็น 1 และถ้าป้อนลอจิกเป็น 0 IC จะไม่ทำงานในส่วนของอินพุตและเอาต์พุต ถ้าป้อนลอจิก 1 ที่อินพุต เอาต์พุตก็จะออกลอจิก 1 และถ้าป้อนลอจิก 0 ที่อินพุต เอาต์พุตก็จะออกลอจิก 0 ด้วยเช่นกัน ซึ่งขึ้นอยู่กับการนำไปใช้งานว่าจะขับมอเตอร์ด้วยลอจิก 0 หรือลอจิก 1

2.3.3.) ลิมิตสวิตช์

คือ สวิตช์ขนาดเล็ก การทำงานอาศัยการเคลื่อนที่เล็กๆมาๆของหน้าสัมผัส ทำให้มีความไวในการทำงาน ซึ่งได้ติดตั้งไว้ที่ด้านท้ายของช่องเพื่อใช้เป็นตัวตัดต่อในการทำงาน และยังทำหน้าที่เป็นเซนเซอร์อย่างหนึ่งที่จะตรวจสอบว่าสินค้าได้เข้าหรือออกไปเรียบร้อยแล้ว ซึ่งการต่อใช้งานจะใช้การต่อเป็นแบบเมตริกซ์ ดังรูป



จากรูปแสดงถึงการต่อสวิตซ์เมตริกซ์ เป็นการ ใช้แบบสวิตซ์กดคิดป้อนต่อกันมาต่อกันในแบบเมตริกซ์ คือ ขาด้านหนึ่งจะต่อในแนวหลักหรือคอลัมน์ (Column) และขาด้านหนึ่งจะต่ออยู่ในแนวแถวหรือ (Row) หลักการเขียนโปรแกรมควบคุมสวิตซ์เมตริกซ์นั้น ต้องกำหนดค่าคงที่ประจำตัวสวิตซ์แต่ละตัวขึ้นมาก่อน การตรวจสอบค่าของสวิตซ์เมตริกซ์ดังรูปจะเห็นได้ว่า พอร์ต P2.0 - P2.3 ต่อกับสวิตซ์เมตริกซ์ตามแนวแถว (Row) และ P2.4 - P2.6 ต่อกับสวิตซ์ในแนวหลัก (Column) ทำการอ่านค่าจากแนวแถวเมื่อมีการกำหนดให้หลักใดหลักหนึ่งทำงานด้วยการส่งค่า 0 ไปที่หลักนั้น แล้วตรวจสอบว่าแถวใดมีค่าลอจิก 0 เช่น เมื่อแถวที่ 1 เท่ากับ 0 และหลักที่ 1 เท่ากับ 0 แสดงว่าปุ่มแรกถูกกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การออกแบบการควบคุมคลังสินค้าโดยใช้โปรแกรม Visual Basic 6.0

2.4.1 วิธีการออกแบบการควบคุมโดย Visual Basic 6.0

เริ่มต้นการสร้างบอร์ดทดลอง

การทดลองนี้เป็นพื้นฐานที่สำคัญมากเพราะถ้าผู้อ่านมีปัญหาเกี่ยวกับการเขียน โปรแกรมเพื่อควบคุมผ่านพอร์ตขนานหรือสงสัยว่าสัญญาณบางสัญญาณ ไม่สามารถส่งหรือรับข้อมูลส่งผ่านผ่านออกทางพอร์ตขนานได้ยังสามารถนำบอร์ดทดลองนี้และ โปรแกรมนี้มาตรวจสอบและยืนยันการทำงานของพอร์ตขนานว่ายังปรกติอยู่หรือไม่

เริ่มเขียนโปรแกรมร่วมกับบอร์ด

สำหรับการสร้าง โปรแกรมนี้มีขั้นตอนดังนี้

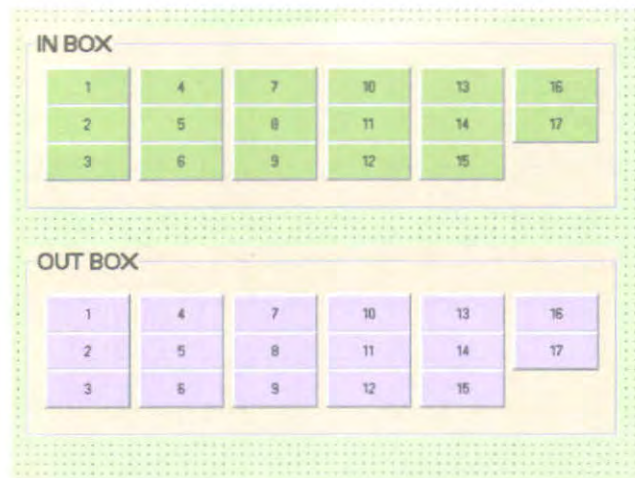
- 1.เปิด Visual Basic ขึ้นมาแล้วเลือกสร้างแอปพลิเคชันแบบ Standard ExE
- 2.เมื่อปรากฏฟอร์มขึ้นมาแล้วให้เพิ่มคอนโทรลต่างลงบนฟอร์มดังนี้



รูปที่ 21 โครงสร้างของปุ่มควบคุม

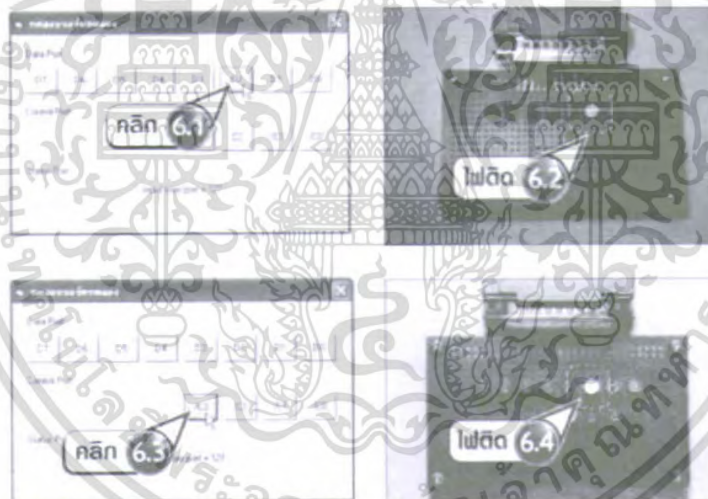
- 3.เขียน โค้ดคำสั่งเพื่อควบคุมการทำงานของ โปรแกรมดังนี้
- 4.เมื่อทุกอย่างเรียบร้อยแล้วต่อไปนี้จะเป็นการทดสอบการทำงานของ โปรแกรมร่วมกับบอร์ดก่อนอื่นจะต้องต่อบอร์ดกับกับพอร์ตขนานก่อน
- 5.กดปุ่ม F5 เพื่อรัน โปรแกรมหน้าจอจะแสดงผลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 22 หน้าต่างของปุ่มควบคุมเมื่อ run โปรแกรมแล้ว

6. ทดลองกดปุ่มต่างๆบน โปรแกรม จะเห็นว่าเวลาคลิกปุ่มใดๆก็ตามบนตัว โปรแกรมแล้วหลอด LED ที่อยู่บนบอร์ดทดลอง ณ ตำแหน่งเดียวกับปุ่มที่ถูกคลิกบนตัว โปรแกรมก็จะติดสว่างตามจังหวะการคลิกที่ตัว โปรแกรมด้วย



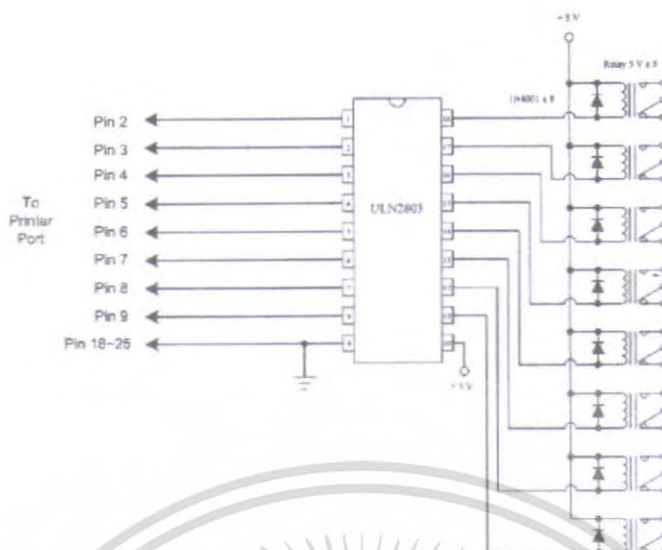
รูปที่ 23 แสดงการทดลองกดปุ่มบนโปรแกรม

2.4.2 บอร์ดควบคุมรีเลย์อเนกประสงค์

วงจรควบคุมการทำงานของรีเลย์

รีเลย์เป็นสวิตช์ชนิดหนึ่งซึ่งกลไกแม่เหล็กไฟฟ้าในการควบคุมการเปิด-ปิดของมัน (จะเรียกว่าเป็นสวิตช์แม่เหล็กไฟฟ้าขนาดเล็กก็ได้) โดยการจ่ายกระแสไฟฟ้าเข้าไปที่ตัวสวิตช์เพื่อทำให้กานสวิตช์เปิด-ปิดตามต้องการ ซึ่งมักจะใช้รีเลย์ในการควบคุมการทำงานอัตโนมัติที่มีการเปิด-ปิดการทำงาน สำหรับในโปรเจกนี้ เราจะสร้างวงจรควบคุมการทำงานของรีเลย์เพื่อนำไปควบคุมการส่งและรับของในคลังสินค้า (Ware house) นั้นเอง โดยมีรูปวงจรดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 24 วงจรควบคุมการทำงานของรีเลย์

ในวงจรจะมีการใช้งานไอซี **ULN 2803** ให้รับสัญญาณ จากเครื่องคอมพิวเตอร์ ผ่านพอร์ตขนาน ทางขาที่ 2,3,4,5,6,7,8 และ 9 และส่งเอาต์พุต ไปควบคุมการทำงานของรีเลย์อีกครั้งหนึ่ง รีเลย์ทั้ง 8 ตัวนี้จะเป็นตัวตัดต่อวงจรเพื่อควบคุมการทำงานของอุปกรณ์ไฟฟ้าแต่ละตัว ควบคุมการทำงานของรีเลย์ตามเวลา เราจะมาประยุกต์ใช้งานรีเลย์ผ่านพอร์ตขนานตัวอย่างแรกด้วยการควบคุมการเปิด-ปิด ตามเวลาที่กำหนด โดยจะเขียน โปรแกรมดังนี้

1. เปิด **Visual Basic 6.0** ขึ้นมาแล้วสร้างแอปพลิเคชันแบบ **Standard.EXE**
2. เพิ่มคอนโทรลต่างๆลงบนฟอร์มดังนี้



รูปที่ 25 ส่วนควบคุมทางรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เขียนโค้ดคำสั่งเพื่อควบคุมการทำงานของโปรแกรมดังนี้
4. เมื่อรันโปรแกรมจะได้นาฬการปร่างของโปรแกรมดังรูป

	Date Alarm	Alarm Time	Alarm Delay(sec)
Relay1	02-07-2008	00:00:00	0
Relay2	02-07-2008	00:00:00	0
Relay3	02-07-2008	00:00:00	0

รูปที่ 26 ส่วนควบคุมทางรีเลย์เมื่อรันโปรแกรมแล้ว

อธิบายการทำงานของโปรแกรม

จากโค้ดควบคุมการทำงานของโปรแกรมคำสั่งต่างๆ มีหลักการทำงานดังนี้

1. การทำงานใน Sub Form Load () เป็นการกำหนดค่าเริ่มต้น ให้กับคอนโทรลต่างๆ และ กำหนดแอดเดรสสำหรับส่งข้อมูลออกจากพอร์ตไว้ด้วยคำสั่ง
2. การทำงานใน Sub Timer 9 นั้นเป็นการตรวจสอบเวลาปัจจุบันเปรียบเทียบกับเวลาที่ตั้งไว้ ซึ่ง การตั้งเวลาที่ตั้งไว้ ซึ่งการตั้งเวลาในการควบคุมรีเลย์ในแต่ละตัวนั้นเราจะแบ่งออกเป็น 2 ส่วน คือ
 - ส่วนแรก เป็นการตั้งเวลาให้รีเลย์ทำงาน ในส่วนนี้เราจะตั้งเวลาทำงานในส่วนนี้เราจะตั้งเวลาในรูปแบบชั่วโมง นาที วินาที (HH:MM:SS) ซึ่งเป็นเวลาที่ทำให้รีเลย์เริ่มต้นทำงานซึ่งจะเรียก ในส่วนของเวลาที่ตั้งส่วนนี้ว่า Alarm Time
 - ส่วนที่ 2 เป็นการกำหนดช่วงระยะเวลาที่เราต้องการให้รีเลย์ทำงานซึ่งเรียกเวลาที่ตั้งส่วนนี้ว่า Alarm Delay โดยการตั้งเวลาส่วนนี้ค่าตัวเลขที่ใส่ไปนั้นจะมีหน่วยเป็นวินาที
3. การทำงานจะเริ่มจากการแสดงเวลาปัจจุบันบนคอนโทรล Label9 จากนั้นตรวจสอบเวลาที่ตั้งใน ส่วนของ Alarm Time ถ้าเวลาในส่วนที่ตั้งไว้ กับ เวลาที่ตั้งไว้ตรงกันก็จะส่งเอาต์พุตออกไปที่ขา 2 ของพอร์ตขนาน รีเลย์ตัวที่หนึ่งทำงาน ซึ่งเมื่อรีเลย์ทำงานตามคำสั่งที่ Sub Timer1_Timer
4. ต่อมาเป็นการตรวจสอบไว้ในส่วนของ Alarm Time ถ้าเวลาปัจจุบันกับในส่วนของเวลาที่ตั้งไว้ ตรงกันก็จะส่งเอาต์พุต ไปที่ขา 3 ของพอร์ตขนาน รีเลย์ตัวที่ 2 ทำงาน โปรแกรมก็จะทำงานตาม คำสั่งที่ Sub Timer2_Timer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การทำงานของคำสั่งอื่นๆที่เหลือก็จะมีการทำงานคล้ายๆกับการควบคุมรีเลย์ตัวที่ 1 และตัวที่ 2 นั่นคือ โปรแกรมจะตรวจสอบค่าเวลาปัจจุบัน เปรียบเทียบกับส่วนของเวลาที่ตั้งไว้ในส่วนของ Alarm Time ถ้าตรงกันก็จะส่งเอาต์พุตออกไปยังรีเลย์ตัวนั้นๆ ใช้งาน และจะเรียกใช้งานตามคำสั่งที่ Timer แต่ละตัวนั่นเอง
6. สำหรับคำสั่งในการทำงาน ของเวลาที่เรที่ตั้งไว้ในส่วนของ Alarm Delay นั้นจะอยู่ในคำสั่งของ Timer แต่ละตัว ซึ่งจะถูกใช้งานจากคำสั่งหลักที่อยู่ใน Sub Timer9_Timer () โดยที่ลักษณะการทำงานของโปรแกรมนั้นก็คือ โปรแกรมจะลดค่าที่เราได้ตั้งไว้ในส่วนของ Alarm Delay ลงทีละหนึ่ง จนกระทั่ง ค่าเวลาในส่วนของ Alarm Delay จะมีค่าน้อยกว่า 0 โปรแกรมก็จะสั่งให้รีเลย์ตัวนั้น หยุดทำงานตัวอย่างเช่น ในการควบคุมการทำงานของกรรณังการงานของ รีเลย์ตัวที่ 1 จะเริ่มจากการลดค่า ที่ตั้งไว้ในส่วนของ Alarm delay ลงทีละ 1 แล้วตรวจสอบค่าที่ถูกลดลงน้อยกว่า 0 หรือไม่ ถ้าค่าที่ลดลงมีค่าน้อยกว่า 0 ให้รีเลย์ตัวที่ 1 หยุดทำงาน แล้วตั้งค่าเริ่มต้นในส่วนของ Alarm delay เท่ากับ 0 หลังแล้วจึงหยุดทำงานของ Timer

2.4.3 รูปแสดงหน้าต่าง Visual Basic และโปรแกรม

ส่วนของโปรแกรม Visual Basic

```
Private Declare Function Imp Lib "inpout32.dll" Alias "Inp32" (By Val Port Address
as Integer) As Integer
```

```
Private Declare Sub out Lib "inpout32.dll" _
Alias "Out32" (By Val Port Address as Integer, By Val Value as Integer)
```

```
Public pwrite1 As Integer
```

```
Public pwrite2 As Integer
```

```
Public pared As Integer
```

```
Private Sub Command1_Click ()
```

```
Out pwrite1, &H1
```

```
End Sub
```

```
Private Sub Command2_Click ()
```

```
Out pwrite1, &H2
```

```
End Sub
```

```
Private Sub Command3_Click ()
```

```
Out pwrite1, &H4
```

```
End Sub
```

```
Private Sub Command4_Click ()
```

```
Out pwrite1, &H8
```

```
End Sub
```

```
Private Sub Command5_Click ()
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Out pwrite1, &H10
End Sub
```

```
Private Sub Command6_Click ()
Out pwrite1, &H20
End Sub
```

```
Private Sub Command7_Click ()
Out pwrite1, &H40
End Sub
```

```
Private Sub Command8_Click ()
Out pwrite1, &H80
End Sub
```

```
Private Sub Form_Load ()
Frame1.Caption = "IN BOX"
```

```
Label1.Caption = "Relay1"
Label2.Caption = "Relay2"
Label3.Caption = "Relay3"
Label5.Caption = "Alarm Time"
Label6.Caption = "Alarm Delay (sec)"
Label8.FontBold = True
```

```
Text4.Alignment = 2
Text5.Alignment = 2
Text6.Alignment = 2
Text7.Alignment = 2
Text8.Alignment = 2
Text9.Alignment = 2
```

```
Text4.Text = "00:00:00"
Text5.Text = "00:00:00"
Text6.Text = "00:00:00"
```

```
Text7.Text = "0"
Text8.Text = "0"
Text9.Text = "0"
```

```
Timer1.Interval = 1000
Timer2.Interval = 1000
Timer3.Interval = 1000
```

```
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
```

```
Label7.FontBold = True
```

```
Label4.Caption = "Date Alarm"
```

```
Text1.Alignment = 2
```

```
Text2.Alignment = 2
```

```
Text3.Alignment = 2
```

```
Text1.Text = Date$
```

```
Text2.Text = Date$
```

```
Text3.Text = Date$
```

```
pwrite1 = &H378
```

```
pwrite2 = &H37A
```

```
Out p write, &H0
```

```
End Sub
```

```
Private Sub Timer1_Timer ()
```

```
Text7.Text = Text7.Text - 1
```

```
If Val (Text7.Text) < 0 Then
```

```
Out pwrite1, Imp (pwrite1) X or &H1
```

```
Text7.Text = "0"
```

```
Timer1.Enabled = False
```

```
End If
```

```
End Sub
```

```
Private Sub Timer2_Timer ()
```

```
Text8.Text = Text8.Text - 1
```

```
If Val (Text8.Text) < 0 Then
```

```
Out pwrite1, Imp (pwrite1) X or &H2
```

```
Text8.Text = "0"
```

```
Timer2.Enabled = False
```

```
End If
```

```
End Sub
```

```
Private Sub Timer3_Timer ()
```

```
Text9.Text = Text9.Text - 1
```

```
If Val (Text9.Text) < 0 Then
```

```
Out pwrite1, Imp (pwrite1) X or &H4
```

```
Text9.Text = "0"
```

```
Timer3.Enabled = False
```

```
End If
```

```
End Sub
```

```
Private Sub Timer4_Timer()
```

```
Label7.Caption = Date$
```

```
Label8.Caption = Time$
```

```
If (Label8.Caption = Text4.Text) And (Label7.Caption = Text1.Text) Then
```

```
Out pwrite1, Imp (pwrite1) Xor H1
```

```
Timer1.Enabled = True
```

```
End If
```

```

If (Label8.Caption = Text5.Text) And (Label7.Caption = Text2.Text) Then
Out pwrite1, Inp(pwrite1) Xor H2
Timer2.Enabled = True
End If

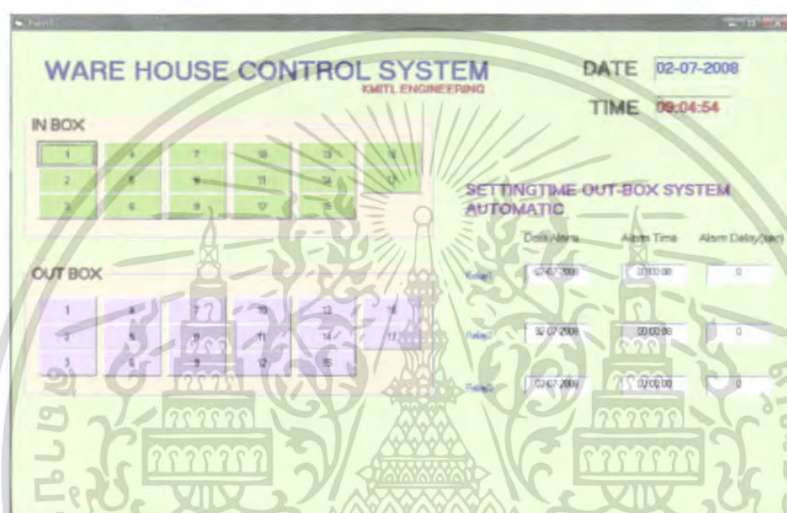
```

```

If (Label8.Caption = Text6.Text) And (Label7.Caption = Text3.Text) Then
Out pwrite1, Inp(pwrite1) Xor H4
Timer3.Enabled = True
End If
End Sub

```

ส่วนของหน้าต่างที่ใช้ควบคุมคลังสินค้า



รูปที่ 27 ส่วนควบคุมโดย Visual Basic 6.0 ที่เสร็จสมบูรณ์

INBOX - ในส่วนของปุ่มต่างๆใน in box นั้นจะเป็นคำสั่งของการกดปุ่มเพื่อนำสินค้า เข้าไปเก็บใน ส่วนของตัว คลังสินค้าเอง โดยแต่ละปุ่มนั้นจะมี หมายเลขที่ปุ่มกำกับ โดยหมายเลขที่ปุ่มนั้นจะตรงกับช่อง ต่างๆในคลังสินค้า หรือ ตัว ware house นั้นเอง

OUT BOX- ในส่วนของปุ่มต่างๆใน out box นั้นจะเป็นคำสั่งของการกดปุ่มเพื่อนำสินค้า ออกไป นอกส่วนของตัว คลังสินค้าเอง โดยแต่ละปุ่มนั้นจะมี หมายเลขที่ปุ่มกำกับ โดยหมายเลขที่ปุ่มนั้นจะตรง กับช่องต่างๆในคลังสินค้า หรือ ตัว ware house นั้นเอง

DATE, TIME -บอกค่าเวลาและวันเดือนปี ปัจจุบัน เพื่อใช้อย่างอิงกับค่าในส่วนของกาตั้งวันเวลาของ สินค้าขาออกอัตโนมัติ

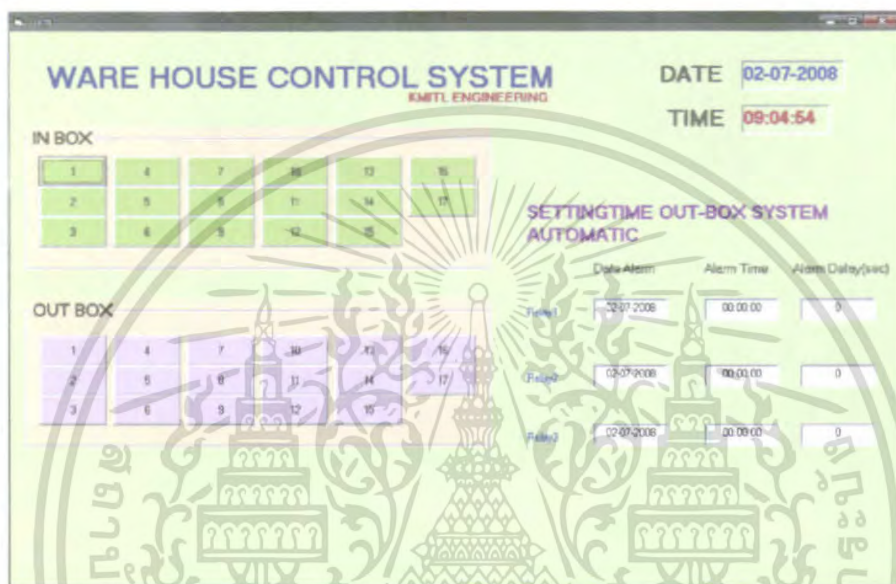
SETTING TIME OUT- BOX SYSTEM AUTOMATIC ในส่วนของหน้าต่างของ **SETTING TIME OUT- BOX SYSTEM AUTOMATIC** จะทำหน้าที่ตรวจสอบวันเดือนปีเพื่อ กำหนดการส่งสินค้าขาออกโดยอัตโนมัติ ตามวันและเวลาที่ผู้ใช้กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ผลการทดลอง

เมื่อทำการติดตั้งระบบควบคุมคลังสินค้ากับคอมพิวเตอร์แล้วก็เปิดหน้าต่างควบคุมโดยโปรแกรม Visual Basic 6.0 -7 ขึ้นมาจะมีลักษณะดังรูป



รูปที่ 28 ส่วนควบคุมโดย Visual Basic 6.0

จากรูป ปุ่มกดจะแบ่งออกเป็น 2 ส่วนคือ ส่วนแรกไว้ใช้ในการสั่งเก็บสินค้า ส่วนที่สองไว้ใช้สั่งในการรับสินค้าและลำเลียงออกมาโดยจะทดลองในส่วนแรกคือ การสั่งเก็บสินค้า

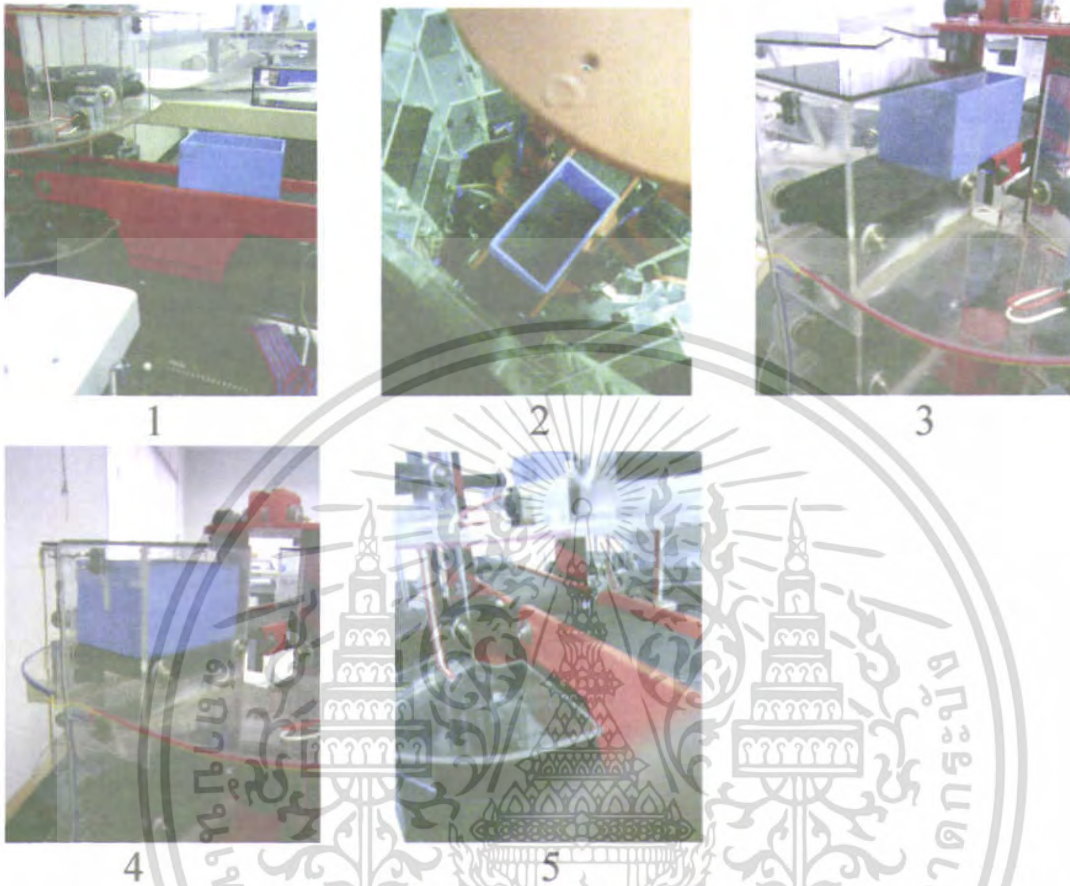
ยกตัวอย่างเช่นหากเราต้องการเก็บสินค้าในชั้นที่ 3 แถว B ก็คลิกที่ปุ่มหมายเลข 6 จากนั้นระบบลิฟต์ก็จะลำเลียงสินค้าขึ้นไปเก็บยังช่องที่เราเลือก ตามที่เราเขียนโปรแกรมสั่งงานไว้ เมื่อสินค้าเคลื่อนที่เข้าไปในช่องแล้วแตะกับลิมิตสวิทช์ ระบบลิฟต์ก็จะกลับมายังตำแหน่งเริ่มต้นเพื่อที่จะรอรับสินค้าต่อไป ซึ่งทางขวามือของส่วนควบคุมสั่งการนั้นจะมีเวลาบอกเพื่อที่จะบันทึกทั้งเวลาและวันที่ใช้งานด้วย เพื่อเป็นข้อมูลในด้านอื่นๆ

ข้อสำคัญอีกอย่างหนึ่งก็คือ เราสามารถที่จะกำหนดเวลาในการจัดเก็บสินค้าแบบอัตโนมัติได้อีกด้วยหรือว่าสั่งการให้ลำเลียงสินค้าออกมาได้ในวันถัดไป ซึ่งทำได้โดยการตั้งวันเวลาทางขวามือ

ส่วนที่สอง การรับสินค้าและลำเลียงออกมา ซึ่งการสั่งงานก็คล้ายกับการสั่งเก็บสินค้า แต่จะต่างกันตรงที่เมื่อลิฟต์ขึ้นไปยังช่องที่เราต้องการแล้ว การสั่งงานของมอเตอร์ก็จะเปลี่ยนจาก Forward เป็น Reward เพื่อที่จะลำเลียงสินค้าออกมา ดังนั้นเวลาเราจะสั่งเก็บสินค้าหรือว่าจะสั่งให้ไปรับสินค้าออกมานั้น จะมีวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเวลาบออกอยู่เสมอว่าเราได้ทำการจัดเก็บสินค้าหรือว่ารับออกมาเมื่อไหร่ ทั้งนี้ขึ้นอยู่กับความต้องการของผู้สั่งงาน



รูปที่ 29 ลำดับการทำงานของคลังสินค้า

หากในกรณีเมื่อมีสินค้าอยู่ในช่องแล้วเมื่อลิฟต์ได้ขึ้นไปยังช่องที่เราสั่งงาน ลิฟต์ก็จะดึงกลับมาจุดเริ่มต้นทันที เนื่องจากในช่องสินค้าที่เราสั่งขึ้นไปนั้น ได้มีสินค้าอยู่ในช่องเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

บทสรุปและวิจารณ์

4.1 ส่วนของ Hardware และ Software

1. ช่องสินค้าในแต่ละช่องมีความกว้างที่แคบเนื่องจากความยาวของแกนเหล็กบังคับ จึงทำให้เวลาส่งของนั้นค่อนข้างไม่สะดวก
2. เซนเซอร์ที่ใช้เป็นเซนเซอร์แสงที่ใช้กับหุ่นยนต์เดินตามเส้นต่างๆ ไปนั้นมีความไวแสงมาก ทำให้เวลาส่งงานไปค่อนข้างที่จะ Error ยังไม่ถึงช่องดี ลิฟต์ก็ทำงานแล้ว
3. เนื่องจากจำนวนช่องที่กำหนดคนนั้นมีมากจึงทำให้พอร์ตไม่เพียงพอเวลาต่อใช้งาน
4. ในแต่ละช่องจะมีมอเตอร์ควบคุมทุกช่อง ทำให้ต้องใช้ไอซีหลายตัว โดยเฉพาะ 8052
5. ช่องสินค้า 1 ช่องสามารถเก็บสินค้าได้ชิ้นเดียวเท่านั้น
6. ถ้าเราส่งงานให้ไปยังช่องๆหนึ่ง ถ้าหากช่องนั้นมีสินค้าอยู่แล้ว ระบบของลิฟต์ก็จะกระเด็นกลับมาที่จุดเริ่มต้นเหมือนเดิม
7. การควบคุมการทำงานของมอเตอร์โดยการหนดเวลาตอนลำเลียงสินค้าออกมายังถาดกลางของลิฟต์ ต้องคั้งช่วงเวลาให้เหมาะสมที่สุด
8. เนื่องจากในทุกๆช่องติดตั้งมอเตอร์สายพานทุกช่อง ทำให้เปลืองอุปกรณ์มากพอสมควร
9. ทำให้มีพื้นที่ว่างเกิดขึ้นจากช่องระหว่างช่องทำให้เสียพื้นที่โดยสิ้นเปลือง
10. ในแต่ละครั้งที่จัดเก็บหรือรับออกมาสามารถทำได้ครั้งละ 1 ชิ้นเท่านั้น

4.2 แนวทางในการพัฒนา

1. ออกแบบให้มีความทนทานมากขึ้นและพยายามใช้อุปกรณ์ไม่ให้สิ้นเปลืองมากที่สุด
2. สามารถที่จะขยับความกว้างของช่องให้กว้างขึ้นได้ โดยขยายความยาวของแกนเหล็ก
3. ขยายการควบคุมให้ได้ครบทุกช่องและสามารถขยายเพิ่มขั้นขึ้นไปได้อีก
4. ออกแบบวงจรใหม่โดยการใช้ 8052 ต่อ เชื่อมและติดต่อกันเพื่อที่จะสามารถเพิ่มพอร์ตในการต่อใช้งานได้มากขึ้น
5. ออกแบบเขียนโปรแกรมควบคุมให้ทำงานโดยอัตโนมัติ
6. สามารถเขียนโปรแกรมให้เป็นคลังสินค้าอัจฉริยะได้โดยการเก็บสินค้าเมื่อมีช่องไหนว่างก็ไปจัดเก็บให้เต็มโดยอัตโนมัติ
7. อาจเลือกใช้เซนเซอร์ตัวใหม่ที่มีความเสถียรภาพกว่าเดิม
8. สามารถที่จะพัฒนาให้จัดเก็บสินค้าได้ครั้งละหลายๆชิ้น
9. พัฒนาในส่วนของโปรแกรม Visual Basic ให้มีการควบคุมที่หลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง (Reference)

1. เฉษฐิธี มณีธรรม, สำเรึง เต็มราม, “คัมภีร์ไมโครคอนโทรลเลอร์ MCS-51”, บริษัทเคทีพี คอมพ์ แอน คอนซัล จำกัด, 2548
2. อุดม รานอก , “ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51”, บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซนเตอร์ จำกัด, ธันวาคม 2548
3. รศ.ดร.วรวงศ์ ตั้งศรีรัตน์, “เซนเซอร์และทรานสดิวเซอร์”, สมาคมส่งเสริมเทคโนโลยี [ไทย-ญี่ปุ่น], เมษายน 2550
4. ประจัน พลังสันติกุล, ชัยวัฒน์ ลิ้มพรจิตวิไล, “ไมโครคอนโทรลเลอร์ MCS-51กับ Keil C51”, บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด
5. อภิชาติ ภู่วลัย, “เขียนโปรแกรม Hardware Interface ด้วย Visual Basic 6.0”, บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซนเตอร์ จำกัด, ตุลาคม 2548

ภาคผนวก ก

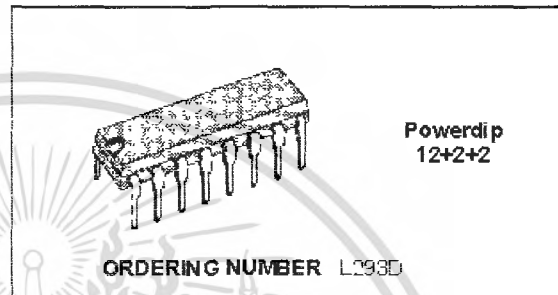


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

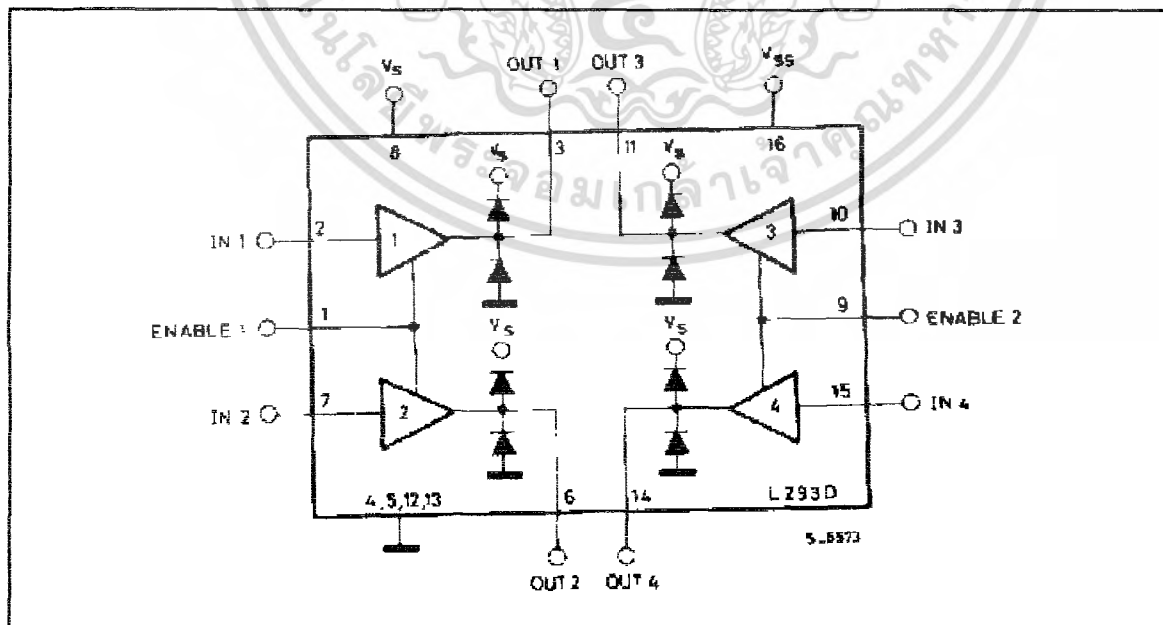
PRELIMINARY DATA

- 800mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (NON REPETITIVE) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES


DESCRIPTION

The L293D is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays, solenoids, DC and stepping motors) and switching power transistors. To simplify use as two bridges a pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a low voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz. The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking.

BLOCK DIAGRAM


Low power quad op amps

LM124/224/324/324A/ SA534/LM2902

DESCRIPTION

The LM124/SA534/LM2902 series consists of four independent, high-gain, internally frequency-compensated operational amplifiers designed specifically to operate from a single power supply over a wide range of voltages.

UNIQUE FEATURES

In the linear mode, the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.

The unity gain crossover frequency and the input bias current are temperature-compensated.

FEATURES

- Internally frequency-compensated for unity gain.
- Large DC voltage gain: 100 dB.
- Wide bandwidth (unity gain): 1 MHz (temperature-compensated).
- Wide power supply range: Single supply: 3 V_{DC} to 30 V_{DC} or dual supplies: ±1.5 V_{DC} to ±15 V_{DC}.
- Very low supply current drain: essentially independent of supply voltage (1 mW/op amp at +5 V_{DC}).
- Low input biasing current: 45 nA_{DC} (temperature-compensated).
- Low input offset voltage: 2 mV_{DC} and offset current: 5 nA_{DC}.
- Differential input voltage range equal to the power supply voltage.
- Large output voltage: 0V_{DC} to V_{DC}-1.5 V_{DC} swing.

PIN CONFIGURATION

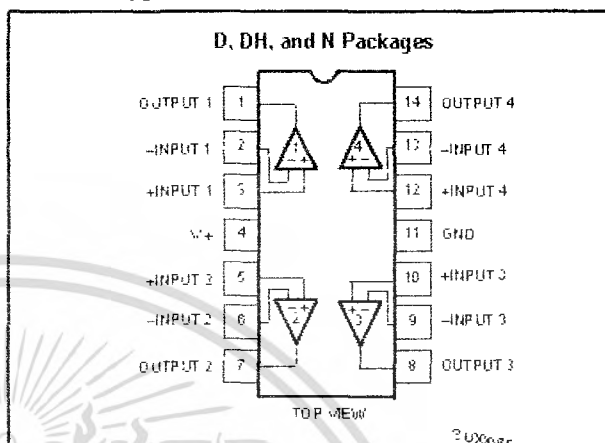


Figure 1. Pin configuration.

ORDERING INFORMATION

DESCRIPTION	TEMPERATURE RANGE	ORDER CODE	DWG #
14-Pin Plastic Dual In-Line Package (DIP)	-55°C to +125°C	LM124N	SOT27-1
14-Pin Plastic Small Outline (SO) Package	-25°C to +85°C	LM224D	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	-25°C to +85°C	LM224N	SOT27-1
14-Pin Plastic Small Outline (SO) Package	0°C to +70°C	LM324D	SOT108-1
14-Pin Plastic Thin Shrink Small Outline Package (TSSOP)	0°C to +70°C	LM324DH	SOT402-1
14-Pin Plastic Dual In-Line Package (DIP)	0°C to +70°C	LM324N	SOT27-1
14-Pin Plastic Small Outline (SO) Package	0°C to +70°C	LM324AD	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	0°C to +70°C	LM324AN	SOT27-1
14-Pin Plastic Small Outline (SO) Package	-40°C to +85°C	SA534D	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	-40°C to +85°C	SA534N	SOT27-1
14-Pin Plastic Small Outline (SO) Package	-40°C to +125°C	LM2902D	SOT108-1
14-Pin Plastic Dual In-Line Package (DIP)	-40°C to +125°C	LM2902N	SOT27-1
14-Pin Plastic Thin Shrink Small Outline Package (TSSOP)	-40°C to +125°C	LM2902DH	SOT402-1



Reflective Optical Sensor with Transistor Output

Description

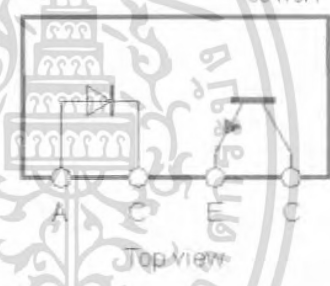
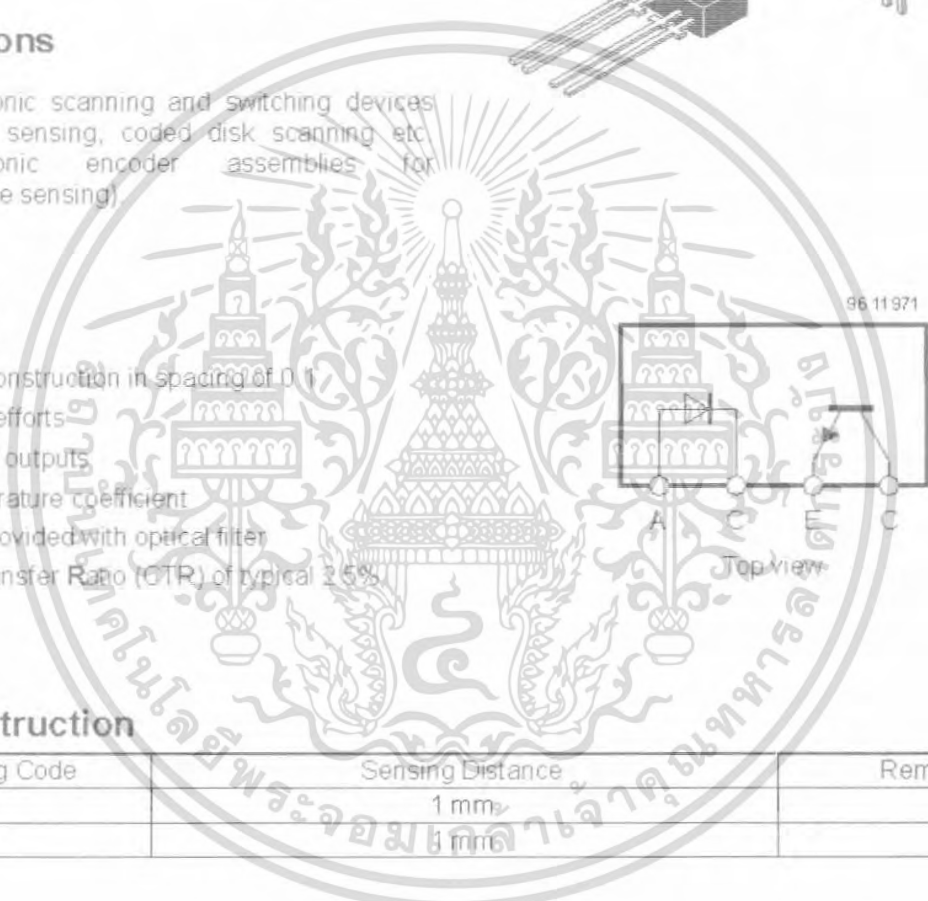
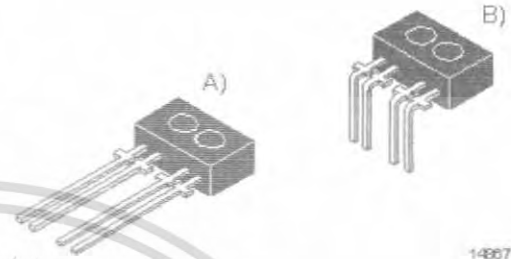
The TCRT1000/ TCRT1010 have a compact construction where the emitting-light source and the detector are arranged in the same direction to sense the presence of an object by using the reflective IR-beam from the object. The operating wavelength is 950 nm. The detector consists of a phototransistor.

Applications

- Optoelectronic scanning and switching devices i.e., index sensing, coded disk scanning etc (optoelectronic encoder assemblies for transmissive sensing)

Features

- Compact construction in spacing of 0.1
- No setting efforts
- High signal outputs
- Low temperature coefficient
- Detector provided with optical filter
- Current Transfer Ratio (CTR) of typical 25%



Order Instruction

Ordering Code	Sensing Distance	Remarks
TCRT1000A)	1 mm	
TCRT1010B)	5 mm	



ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการควบคุมคลังสินค้าที่ใช้สาธิตทั้งหมด 3 ช่อง

```
#pragma SMALL
#include<reg52.h>
//*****Switch Function*****//
sbit row1    = P1^2;
sbit column1 = P1^0;
sbit column2 = P1^1;
sbit column3 = P1^4;
//*****Sensor Function*****//
sbit sensorA = P0^3;
sbit sensorB = P0^7;
sbit sensor1 = P0^0;
sbit sensor2 = P0^1;
sbit sensor3 = P0^2;
//*****motor Function*****//
sbit M1CF = P3^0;
sbit M1CR = P3^5;
sbit MCf  = P2^5;
sbit MCr  = P2^6;
sbit LiftU = P3^4;
sbit LiftD = P3^3;
sbit MCH3f = P2^0;
sbit MCH2f = P2^2;
sbit MCH1f = P2^3;
sbit MCH3r = P3^6;
sbit MCH2r = P2^1;
sbit MCH1r = P2^4;
//*****Limit Switch*****//
sbit LSR1 = P0^5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sbit LSC1 = P3^1;
sbit LSC2 = P3^2;
sbit LSC3 = P2^7;
//*****//

    unsigned int k;
    unsigned int l;
    int t;

//*****//
//*****Timer Function*****//
//*****//
void Wait_TM(unsigned int sec)
{
    unsigned int i,j;
    for(i=0;i<=sec;i++)
    {
        for(j=0;j<=20;j++)
        {
            TH0 = 0x4B;
            TL0 = 0xFF;
            TF0 = 0;
            TR0 = 1;
            while (TF0==0);
            TR0 = 0;
        }
    }
}

//*****//
//*****Function CALL*****//
//*****//

void motor_cen_for(void)
{
    MICF = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

//*****//

void motor_cen_re(void)
{
  MICR = 0;
}

//*****//

void motor_lift_up(void)
{
  LiftU = 0;
}

//*****//

void motor_lift_do(void)
{
  LiftD = 0;
}

//*****//

void motor_MC_for(void)
{
  MCf = 0;
}

//*****//

void motor_MCH1_for(void)
{
  MCH1f = 0;
}

//*****//

void motor_MCH2_for(void)
{
  MCH2f = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//*****//
void motor_MCH3_for(void)
{
MCH3f = 0;
}
//*****//
void motor_MCH3_re(void)
{
MCH3r = 0;
}
//*****//
void motor_MCH2_re(void)
{
MCH2r = 0;
}
//*****//
void motor_MC_re(void)
{
MCr = 0;
}
//*****//
void motor_MCH1_re(void)
{
MCH1r = 0;
}
//*****//
void motor_stop_all(void)
{
M1CF = 1;
M1CR = 1;
LiftU= 1;
LiftD= 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MCf = 1;
MCH1f = 1;
MCH2f = 1;
MCH3f = 1;
MCH1r = 1;
MCH2r = 1;
MCH3r = 1;
MCr = 1;
}
//*****//
//*****Main Function*****//
void main (void)
{
    row1 = 0;
    column1 = 1;
    column2 = 1;
    column3 = 1;
    LSR1=0;
    LSC1=1;
    LSC2=1;
    LSC3=1;
    k = 0;
    l = 0;
    t = 0;

//*****//
//*****Send Goods*****//
//*****CH3 BlockB*****//
while(1)
{
    if(!column1)&&(k==0)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(!sensorA==1)
{
motor_cen_for();
}

motor_stop_all();

```

```

while(!sensor3==1)
{
motor_lift_up();
}
motor_stop_all();
while(LSC1==1)
{
motor_MC_for();
motor_MCH3_for();
}
motor_stop_all();
if(LSC1==0)
{
while(sensor1==0)
{
motor_lift_do();
}
motor_stop_all();
}
while(sensorB==0)
{
motor_cen_re();
}

motor_stop_all();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
//*****//
//*****Receive*****//
if(!column1)&&(k==1))
{
while(!sensorA==1)
{
motor_cen_for();
}
motor_stop_all();
while(!sensor3==1)
{
motor_lift_up();
}
motor_stop_all();
while(LSC1==0)
{
motor_MC_re();
Wait_TM(40);
motor_MCH3_re();
Wait_TM(70);
Wait_TM(8);
}
motor_stop_all();
if(LSC1==1)
{
Wait_TM(8);
Wait_TM(8);
while(sensor1==0)
{
motor_lift_do();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    motor_stop_all();
}
while(sensorB==0)
{
    motor_cen_re();
}
motor_stop_all();
Wait_TM(8);
MCf = 0;
Wait_TM(200);
MCf = 1;
}
//*****Send Goods*****//
//*****CH2 BlockB*****//
if(!column2)&&(l==0))
{
    while(!sensorA==1)
    {
        motor_cen_for();
    }
    motor_stop_all();

while(!sensor2==1)
    {
        motor_lift_up();
    }
    motor_stop_all();
while(LSC2==1)
    {
        motor_MC_for();
        motor_MCH2_for();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    motor_stop_all();
    if(LSC2==0)
    {
while(sensor1==0)
    {
        motor_lift_do();
    }
    motor_stop_all();
    }
    while(sensorB==0)
    {
        motor_cen_re();
    }
    motor_stop_all();
    l = 1;
}
//*****//
//*****Receive*****//
if(!column2)&&(l==1)
{
    while(!sensorA==1)
    {
        motor_cen_for();
    }

    motor_stop_all();

while(!sensor2==1)
    {
        motor_lift_up();
    }

    motor_stop_all();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(LSC2==0)
{
    motor_MC_re();
    Wait_TM(40);
    motor_MCH2_re();
    Wait_TM(70);
    Wait_TM(8);
}
motor_stop_all();

if(LSC2==1)
{
    Wait_TM(8);
    Wait_TM(8);
}

while(sensor1==0)
{
    motor_lift_do();
}
motor_stop_all();

while(sensorB==0)
{
    motor_cen_re();
}
motor_stop_all();
Wait_TM(8);
MCf = 0;
Wait_TM(200);
MCf = 1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//*****Send Goods*****//
//*****CH1 BlockF*****//
if(!column3)&&(t==0)
{
    while(!sensor3==1)
    {
        motor_lift_up();
    }
    motor_stop_all();

    while(LSC3==1)
    {
        motor_MC_for();
        motor_MCH1_for();
    }
    motor_stop_all();

    if(LSC3==0)
    {
        while((!sensorB)&&(!sensor1)==1)
        {
            motor_lift_do();
        }

        motor_stop_all();
    }

    t = 1;
}
//*****Receive*****//
if(!column3)&&(t==1)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(!sensor3==1)
{
    motor_lift_up();
}
motor_stop_all();

while(LSC3==0)
{
    motor_MC_re();
Wait_TM(40);
    motor_MCH1_re();
Wait_TM(70);
    Wait_TM(8);
}
    motor_stop_all();
    if(LSC3==1)
    {
while(!(sensorB)&&(!sensor1)==1)
    {
        motor_lift_do();
    }

        motor_stop_all();
        Wait_TM(8);
        MCf = 0;
        Wait_TM(200);
MCf = 1;
    }
}
}
}
}

```

//*****//

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้