สำนักหอสมุดกลาง จิระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คลังสินค้าอัตโนมัติ

WAREHOUSE

โดย

นายจีระเดช จันทร์กระจ่าง

นายอานนท์ เปลี่ยนสกุล

นายปิยาพันธ์ กิตติสารธรรมา

เลขหมู่.....
หมายเลข 83101
วัน,เดือน,ปี - 5 ส.ค. 2551

b 11961337 2
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง คลังสินค้าอัตโนมัติ

WAREHOUSE

ผู้จัดทำ นายจีระเดช จันทร์กระจ่าง 47010112

นายอานนท์ เปลี่ยนสกุล 47010982

นายปิยาพันธ์ กิตติสารธรรมา 47011007

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ เกียรติวรรณ ทรงสัตย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลังสินค้าอัตโนมัติ

โดย

นายจีระเดช จันทร์กระจ่าง 47010112

นายอานนท์ เปลี่ยนสกุล 47010982

นายปิยาพันธ์ กิตติสารธรรมา 47011007

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ เกียรติวรรณ ทรงสัตย์

ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการสร้างแบบจำลองของคลังสินค้า โดยใช้คอมพิวเตอร์ตรวจสอบรายการสินค้าและทำการประมวลผลเพื่อสั่งการให้นำสินค้าเข้าและออกตามคำสั่ง ผู้ใช้งานสามารถเลือกการทำงานในการเก็บสินค้าเข้าและนำสินค้าออกผ่านทางโปรแกรมแอฟพลิเคชันบนคอมพิวเตอร์เพื่อให้ง่ายและสะดวกต่อการใช้งาน จากโปรแกรมบนคอมพิวเตอร์จะส่งสัญญาณควบคุมผ่านทางพอร์ทอนุกรม (Serial Port) สั่งการเครนอัตโนมัติ ซึ่งตัวเครนอัตโนมัติถูกควบคุมโดยไมโครคอนโทรลเลอร์ PIC16F877A และมีการตรวจจับตำแหน่งการเคลื่อนที่ในแต่ละแถวและแต่ละชั้นของตัวเครนอัตโนมัติจากเซนเซอร์ที่ประกอบด้วยตัวส่งและตัวรับอินฟราเรด ในส่วนการขับเคลื่อนตัวเครนอัตโนมัติจะใช้มอเตอร์กระแสตรงที่ถูกควบคุมจากตัวไมโครคอนโทรลเลอร์ผ่านทาง IC L298 และ IC L293D เป็นตัวขับเคลื่อนตัวเครนอัตโนมัติเพื่อทำการขนสินค้าเข้าและออกจากคลังสินค้าต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WARE HOUSE

By

Mr. Jeeradet Junkrajang 47010112

Mr. Anon Pliensakul 47010982

Mr. Piyapan Kittisarathamma 47011007

Advisor

Asst. Prof. Kiettiwan Songsataya

Academic Year 2007

ABSTRACT

This thesis presents about the constructive warehouse and warehouse control system by using computer which checks and compiles for trade in or trade out product. User can choose operation about trade in or trade out product by access on application. Then computer will transmit the control signal to automatic crane by serial port. Automatic crane is controlled by microcontroller PIC16F877A and check row and column by infrared sensors. Automatic crane use dc motor for movement to warehouse.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก ผศ.เกียรติวรรณ ทรงสัตย์ ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงงาน ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงงานอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงงานนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นายจีระเดช จันทร์กระจ่าง

นายอานนท์ เปลี่ยนสกุล

นายปิยาพันธ์ กิตติสารธรรมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขอบเขตของโครงงาน	2
1.4 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	3
2.1 ไมโครคอนโทรลเลอร์ PIC16F877A	3
2.1.1 โครงสร้างพื้นฐาน	3
2.1.2 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F877A	4
2.1.3 โมดูลสื่อสารข้อมูลอนุกรมในไมโครคอนโทรลเลอร์ PIC16F877A	6
2.2 การส่งข้อมูล	8
2.3 การตรวจจับตำแหน่ง	8
2.4 การควบคุมมอเตอร์	9
บทที่ 3 หลักการออกแบบ	11
3.1 การออกแบบในส่วนของฮาร์ดแวร์	11
3.1.1 โครงสร้างของคลังสินค้า	11
3.1.2 โครงสร้างของเครน	11
3.1.2.1 ชุดเคลื่อนที่ในแนวแกน X	12
3.1.2.2 ชุดเคลื่อนที่ในแนวแกน Y	12
3.1.2.3 ชุดเคลื่อนที่ในแนวแกน Z	13
3.2 การออกแบบในส่วนของซอฟท์แวร์	13
3.2.1 ส่วนที่ใช้ในการควบคุม	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.2.1.1 การนำสินค้าเข้าไปในคลังสินค้า 15

3.2.1.2 การนำสินค้าออกจากคลังสินค้า 15

3.2.2 ส่วนที่ใช้ในการแสดงผลบนหน้าจอคอมพิวเตอร์ 20

3.2.2.1 การออกแบบระบบฐานข้อมูล 20

บทที่ 4 การทดลองและผลการทดลอง 29

4.1 การเข้าสู่ระบบใช้งาน (Log in) 29

4.2 การขนถ่ายสินค้าขาเข้า 31

4.2.1 การทำงานในส่วนของโปรแกรมเพื่อนำสินค้าเข้า 31

4.2.2 การทำงานในส่วนของเครนเพื่อนำสินค้าเข้า 33

4.3 การขนถ่ายสินค้าขาออก 36

4.3.1 การทำงานในส่วนของโปรแกรมเพื่อนำสินค้าออก 36

4.3.2 การทำงานในส่วนของเครนเพื่อนำสินค้าออก 37

บทที่ 5 วิจารณ์และสรุป 39

5.1 สรุปผลการทดลอง 39

5.2 ปัญหาที่พบและแนวทางแก้ไข 39

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา 39

ภาคผนวก ก โปรแกรมควบคุม 41

ภาคผนวก ข เอกสารเครื่องมืออุปกรณ์อิเล็กทรอนิกส์ 58

เอกสารอ้างอิง 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
2.1 แสดงโครงสร้างแต่ละพอร์ตของไมโครคอนโทรลเลอร์ PIC16F877A	3
2.2 แสดงวงจร IC MAX232	8
2.3 แสดงวงจรตรวจจับตำแหน่ง	9
2.4 แสดงวงจรจรของ L293D	10
2.5 แสดงวงจรของ L298	10
3.1 แสดงโครงสร้างของคลังสินค้า	11
3.2 แสดงชุดเคลื่อนที่ในแนวแกน X	12
3.3 ชุดเคลื่อนที่ในแนวแกน Y	12
3.4 (ก) ชุดเคลื่อนที่ในแนวแกน Z เมื่อมองจากด้านบน	13
3.4 (ข) ชุดเคลื่อนที่ในแนวแกน Z เมื่อมองจากด้านข้าง	13
3.5 แสดงวงจรของ PIC16F877A	14
3.6 โฟลวชาร์ตแสดงการทำงานของระบบควบคุม	20
3.7 รูปตารางทั้งหมดที่ใช้ในคลังเก็บสินค้า	21
3.8 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลผู้ใช้งาน	22
3.9 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลที่เก็บในคลังสินค้า	23
3.10 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลที่เคยเก็บในคลังสินค้า	24
3.11 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลที่ได้ถูกนำออกมาจากคลังสินค้า	25
3.12 แสดงหน้าจอเข้าใช้งาน	25
3.13 แสดงหน้าจอระบบใช้งาน	26
3.14 หน้าจอแสดงตำแหน่งทั้ง 2 ฝั่ง	26
3.15 ข้อมูลสินค้าที่มีอยู่ในคลังเก็บสินค้า	27
3.16 แสดงการทำงานถ่ายสินค้าเข้า	27
3.17 แสดงการทำงานถ่ายสินค้าออก	28
4.1 แสดงหน้าจอเข้าสู่ระบบใช้งาน	29
4.2 แสดงกล่องข้อความรายงานความผิดพลาด	29
4.3 หน้าจอแสดงข้อความเมื่อชื่อและรหัสผ่านถูกต้อง	30
4.4 แสดงหน้าจอระบบใช้งาน	30
4.5 แสดงหน้าจอเลือกตำแหน่งเพื่อเก็บชิ้นงาน	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่	หน้า
4.6 แสดงหน้าจอระบบเตือนให้เลือก Product	31
4.7 แสดงหน้าจอระบบเตือนให้เลือก ID	32
4.8 แสดงหน้าจอการยืนยันการทำงาน	32
4.9 ข้อความแสดงการทำงานสมบูรณ์	32
4.10 แสดงหน้าจอข้อมูลที่ถูกเก็บในตาราง	33
4.11 แสดงเครนที่ตำแหน่งเริ่มต้น	33
4.12 แสดงการทำงานในแนวแกน X	34
4.13 แสดงการทำงานในแนวแกน Y	34
4.14 แสดงการทำงานในแนวแกน Z	35
4.15 เครนเคลื่อนที่กลับมาที่ตำแหน่งเริ่มต้น	35
4.16 แสดงหน้าจอเลือกตำแหน่งเพื่อนำชิ้นงานออก	36
4.17 แสดงหน้าจอการยืนยันการทำงาน	36
4.18 ข้อความแสดงการทำงานสมบูรณ์	36
4.19 แสดงหน้าจอข้อมูลที่ถูกลบออกตาราง	37
4.20 แสดงการทำงานของเครนขณะที่เข้าไปเอาชิ้นงาน	37
4.21 เครนนำชิ้นงานมาวางไว้ที่จุดเริ่มต้น	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 เอาไว้ใช้เก็บรายละเอียดของผู้เข้ามาใช้งาน	21
3.2 Products เอาไว้ใช้เก็บรายละเอียดของสินค้าที่อยู่ใน Stock	22
3.3 TradeIN เอาไว้ใช้เก็บรายละเอียดของสินค้าที่ได้ถูกเก็บใน Stock	23
3.4 TradeOUT เอาไว้ใช้เก็บรายละเอียดของสินค้าที่ได้ถูกนำออกมาจาก Stock	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 กล่าวนำ

การศึกษาในสาขาวิชาแมคคาทรอนิกส์ เป็นการศึกษาเกี่ยวกับวิชาทางด้านอิเล็กทรอนิกส์และวิชาทางด้านเครื่องกลควบคู่กันไป รวมทั้งมีการออกแบบระบบควบคุมเพื่อใช้ควบคุมระบบต่างๆ ให้สามารถทำงานได้อย่างมีเสถียรภาพ เพื่อให้เป็นไปตามวัตถุประสงค์จึงเลือกที่จะศึกษาทำโครงการ WAREHOUSE ขึ้นมา ซึ่งเป็นโครงการหนึ่งที่น่าสนใจ เพราะได้มีการใช้ความรู้ทางด้านแมคคานิกส์ในการออกแบบชิ้นงาน ความรู้ทางด้านอิเล็กทรอนิกส์ในการออกแบบวงจร และความรู้ทางด้านระบบควบคุมเพื่อใช้ในการออกแบบระบบอีกด้วย

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

ในปัจจุบันโรงงานอุตสาหกรรมทั้งขนาดใหญ่ ขนาดกลาง และขนาดเล็ก ต้องการระบบการจัดเก็บสินค้าที่ได้มาตรฐานทั้งนี้เพื่อความสะดวก รวดเร็วและมีประสิทธิภาพ ซึ่งจะส่งผลให้ต้นทุนในการดำเนินธุรกิจต่ำและยังคงความน่าเชื่อถือของธุรกิจ ส่งผลให้สามารถแข่งขันได้ในตลาดโดยยังสร้างผลกำไรให้กับเจ้าของธุรกิจได้ ในทุกวันนี้การจัดเก็บสินค้ายังใช้คนเป็นผู้จัดเก็บอยู่ ซึ่งอาจก่อให้เกิดความสับสน และความซับซ้อนในการจัดการสินค้า ทำให้เกิดการเสียเวลาและมีความล่าช้าในการเรียกใช้ เป็นการเพิ่มต้นทุนในการผลิต และไม่ส่งผลดีต่อระบบธุรกิจในปัจจุบัน และถ้าใช้ในงานที่ต้องการจัดเก็บมีขนาดใหญ่อาจจำเป็นต้องใช้คนเป็นจำนวนมาก และอาจก่อให้เกิดอันตรายและความเสียหายต่อชีวิตและทรัพย์สิน การจัดเก็บสินค้าของโรงงานในปัจจุบันมีความต้องการที่จะทำให้เกิดความสะดวก รวดเร็ว และง่ายต่อการปฏิบัติงาน

การใช้คลังสินค้าอัตโนมัติ สามารถทำงานได้อย่างรวดเร็ว ประหยัดการลงทุน ค่าใช้จ่ายในด้านพลังงาน และบุคลากร ซึ่งท่านสามารถคืนทุนได้ในเวลาที่รวดเร็ว ยิ่งกรณีที่เครื่องเดิน 3 กะ ยิ่งคืนทุนเร็ว เป็นการลดต้นทุนด้าน Logistics ทำให้เป็นข้อได้เปรียบในการแข่งขันทางการค้าได้ และในการทำงานของเครน เมื่อนำสินค้าเข้าจัดเก็บในชั้นวางสินค้า การวิ่งของเครนขาออกจะสามารถรับคำสั่งให้นำสินค้าออกมาด้วยอย่างต่อเนื่อง ทำให้การวิ่งเครน ประหยัดเวลาและพลังงานและนี่คือคอนเซปท์ของการจัดการอันทันสมัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1. ส่วนของคลังสินค้าจำลองซึ่งมีหน้าที่เก็บสินค้ามี 2 ฝั่งโดยแต่ละฝั่งมีขนาด 78x90.5x20 cm รวมทั้งหมด 3 ชั้น โดยแต่ละชั้นสามารถจัดเก็บสินค้าได้ทั้งหมด 5 ชิ้น

2. ส่วนของเครนทำหน้าที่เคลื่อนย้ายสินค้าไปยังตำแหน่งที่ต้องการเพื่อนำสินค้าเข้าหรือออกจากตำแหน่งดังกล่าว

3. ส่วนที่ใช้ในการควบคุมตัวเครนให้สามารถที่จะเคลื่อนย้ายสินค้าไปยังตำแหน่งที่ต้องการได้ โดยจะใช้ไมโครคอนโทรลเลอร์ PIC16F877A ในการควบคุมโดยใช้ภาษา C

4. ส่วนคอมพิวเตอร์ที่มีหน้าที่รับคำสั่งจากผู้ใช้แล้วนำมาประมวลผลเพื่อสั่งการให้เครนทำงาน แสดงผลการทำงาน และเก็บข้อมูลการเข้าออกของสินค้าภายในคลังสินค้าโดยใช้ภาษา Visual C#

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวนำถึงวัตถุประสงค์ ขอบเขตของโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่ใช้ การใช้งานพอร์ตอนุกรม วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้องและนำเอาความรู้ไปประยุกต์ใช้ในการจัดทำโครงงาน

บทที่ 3 หลักการออกแบบ นำเสนอการประกอบโครงสร้างของระบบ รวมถึงแนวคิดในการออกแบบระบบควบคุม

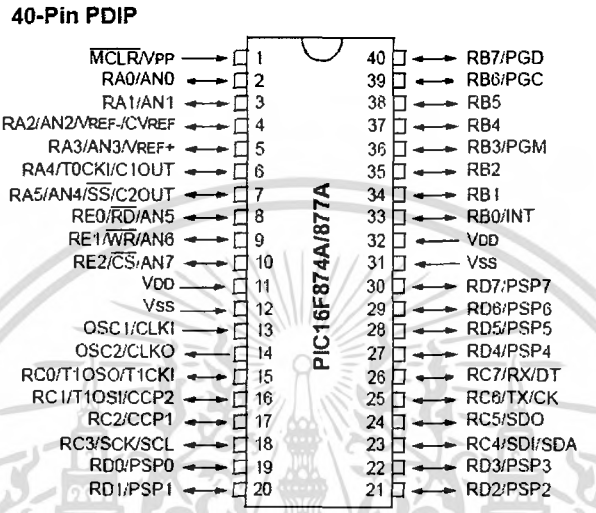
บทที่ 4 การทดลอง เป็นส่วนการทดสอบองค์ประกอบต่างๆ ในระบบ ตลอดจนการทดลองการใช้งาน WAREHOUSE

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงงานนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและความรู้ที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ PIC16F877A



รูปที่ 2.1 แสดงโครงสร้างแต่ละพอร์ตของไมโครคอนโทรลเลอร์ PIC16F877A

2.1.1 โครงสร้างพื้นฐาน

ไมโครคอนโทรลเลอร์ PIC16F877A มีคุณสมบัติหลัก ดังนี้

- ซีพียูเป็นแบบ RISC (Reduced Instruction-Set Computer) มีคำสั่งใช้งานเพียง 25 คำสั่ง
- สามารถกระทำคำสั่งโดยใช้สัญญาณเพียงหนึ่งลูก ยกเว้นคำสั่งกระโดด
- ความถี่สัญญาณนาฬิกา ตั้งแต่ไฟตรงถึง 20 MHz
- มีหน่วยความจำโปรแกรม 8 กิโลเวิร์ด
- มีหน่วยความจำข้อมูลแรมหรือรีจิสเตอร์ 368 ไบต์
- หน่วยความจำข้อมูลอีอีพรอม 256 ไบต์
- ตอบสนองแหล่งกำเนิดอินเตอร์รัปต์สูงสุดถึง 15 แหล่ง
- มีสแต็ก 8 ระดับ
- มีวงจรเพาเวอร์ออนรีเซต (POR)
- มีเพาเวอร์อัปไทเมอร์ (PWRT) และออสซิลเลเตอร์สตาร์ตอัปไทเมอร์ (OST)
- มีวงจรวอตช์ด็อกไทเมอร์ (WDT) ที่มีวงจรออสซิลเลเตอร์ในตัว
- เลือกป้องกันข้อมูลทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลสามารถเลือกระดับป้องกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีโหมดประหยัดพลังงาน
- สามารถโปรแกรมโดยใช้แรงดัน +5V ได้
- แก้ไขข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ ICD (In-circuit Debugger) ผ่านพอร์ตเพียง 2 ขา
- ซีพียูสามารถอ่านและเขียนหน่วยความจำโปรแกรมได้
- ไฟเลี้ยง +2 ถึง +5.5V
- กระแสซิงก์และซอร์สของพอร์ต 25 mA
- การใช้พลังงานไฟฟ้าในกรณีไม่ขับโหลด

  น้อยกว่า 2mA ที่ไฟเลี้ยง +5V และสัญญาณนาฬิกา 4 MHz

  20 μA ที่ไฟเลี้ยง +3V และสัญญาณนาฬิกา 32 kHz

  น้อยกว่า 1 μA ในโหมดประหยัดพลังงานหรือสแตนด์บาย

2.1.2 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F877A

พอร์ต A

มีทั้งสิ้น 6 ช่องหรือ 6 บิต กำหนดชื่อขาเป็น **RA0-RA5** รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ **PORTA** มีแอดเดรสอยู่ที่ 0x05 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานจริงเพียง 6 บิต ที่เหลือ 2 บิตต้องกำหนดให้เป็น "0" ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ **TRISA** ซึ่งมีแอดเดรสอยู่ที่ 0x85 (แบงก์ 1) มีขนาด 8 บิตและใช้เพียง 6 บิตเช่นกัน 2 บิตบนคือบิต 6 และบิต 7 ต้องกำหนดให้เป็น "0" บิต 0 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA5 ไล่เรียงลำดับจนถึงบิต 5 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA5 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล "0" ไปยังบิตนั้น

การกำหนดการทำงานและเขียนข้อมูลไปยังพอร์ต A ในขั้นแรกต้องทำการเตรียมความพร้อมให้แก่รีจิสเตอร์ PORTA โดยการเลือกแบงก์สำหรับติดต่อกับรีจิสเตอร์ PORTA จากนั้นส่งข้อมูล "0" เพื่อเคลียร์ข้อมูลทั้งหมด และเนื่องจากพอร์ต A สามารถทำงานกับสัญญาณอะนาลอกได้ และค่าเริ่มต้นของพอร์ต A ที่ทำงานกับสัญญาณอะนาลอกจะถูกกำหนดให้เป็นอินพุตอะนาลอกทั้งหมด ดังนั้นหากต้องการใช้งานเป็นพอร์ตดิจิตอลต้องกำหนดข้อมูล 006 หรือ 007 แล้วเขียนลงในรีจิสเตอร์ ADCON1 เพื่อดิสเอเบิลการทำงานกับสัญญาณอะนาลอก จากนั้นจึงเลือกแบงก์ใหม่เพื่อติดต่อกับรีจิสเตอร์ TRISA แล้วเขียนข้อมูลเพื่อกำหนดทิศทางของขาพอร์ตตามที่ต้องการลงในรีจิสเตอร์ TRISA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต B

มี 8 บิต กำหนดชื่อขาเป็น **RB0-RB7** รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ **PORTB** มีแอดเดรสอยู่ที่ 0x06 (แบงก์ 0) และ 0x106 (แบงก์ 2) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ **TRISB** ซึ่งมีแอดเดรสอยู่ที่ 0x86 (แบงก์ 1) และ 0x186 (แบงก์ 3) มีขนาด 8 บิตเช่นเดียวกับพอร์ต A บิต 0 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RB0 ไล่เรียงลำดับจนถึงบิต 7 ของ TRISB ใช้กำหนดทิศทางของขาพอร์ต RB7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล "0" ไปยังบิตนั้น

พอร์ต B สามารถใช้งานในลักษณะต่างๆ ได้ 5 แบบคือ

- เป็นขาพอร์ตอินพุตเอาท์พุตปกติ
- เป็นขาอินพุตสัญญาณอินเตอร์รัปต์จากภายนอก โดยใช้ขา RB0/INT
- เป็นขาพอร์ตอินพุตสำหรับรับแรงดันโปรแกรมระดับต่ำ (low voltage programming) โดยใช้ขา RB3/PGM
- เป็นขาข้อมูลอนุกรมและสัญญาณนาฬิกาอนุกรมสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ ซึ่งใช้ 2 ขาคือ RB7/PGD และ RB6/PGC
- ใช้เป็นแหล่งกำเนิดสัญญาณอินเตอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงข้อมูลหรือระดับสัญญาณที่ขา RB4-RB7

ในบิต RB0-RB3 ที่ขาพอร์ตจะมีวงจรพูลอัปแบบโปรแกรมได้ต่ออยู่ นั่นคือหากต้องการกำหนดให้เป็นขาอินพุตต้องทำการเขียนข้อมูล "0" ไปยังบิต RBPU ในรีจิสเตอร์ OPTION_REG เพื่อเอ็นเอเบิลวงจรพูลอัปภายในขาพอร์ต B ในขณะที่หากกำหนดเป็นเอาต์พุต การพูลอัปที่ขาพอร์ต B นี้จะถูกยกเลิกโดยอัตโนมัติ และขาพอร์ต RB4-RB7 จะมีความสามารถพิเศษพอสมควร โดยสามารถเลือกให้ทำงานเป็นขาพอร์ตอินพุตเอาต์พุตปกติ, ขาอินพุตรับแรงดันสำหรับการโปรแกรม (RB3), ขาสัญญาณสำหรับการโปรแกรม (RB6-RB7) และทำงานเป็นแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ในแบบตรวจสอบการเปลี่ยนแปลงที่ขาพอร์ต RB4-RB7

พอร์ต C

มีทั้งสิ้น 8 บิต กำหนดชื่อขาเป็น **RC0-RC7** รีจิสเตอร์ที่ใช้เก็บข้อมูลคือ **PORTC** มีแอดเดรสอยู่ที่ 0x07 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ **TRISC** ซึ่งมีแอดเดรสอยู่ที่ 0x87 (แบงก์ 1) มีขนาด 8 บิตเช่นเดียวกับพอร์ต A และ B บิต 0 ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC0 ไล่เรียงลำดับจนถึงบิต 7 ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล "0" ไปยังบิตนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต C สามารถใช้งานในลักษณะต่างๆ ได้หลายรูปแบบ และเป็นขาพอร์ตที่มีความสามารถสูงมาก ไม่ว่าจะเป็นขาพอร์ตอินพุตเอาต์พุตปกติ, ขาอินพุตเอาท์พุตออสซิลเลเตอร์ของโมดูลไทเมอร์ 1, ขาอินพุตสำหรับรับสัญญาณนาฬิกาของโมดูลไทเมอร์ 1, ขาเชื่อมต่อระบบบัส I^2C, ขาเชื่อมต่อแบบ SPI (Serial Peripheral Interface), ขาเชื่อมต่ออนุกรมแบบ USART, ขาอินพุตวงจรแคปเจอร์ (capture) หรือวงจรตรวจจับสัญญาณ ขาเอาต์พุตของวงจรเปรียบเทียบสัญญาณ (compare) และขาเอาต์พุตวงจร PWM (Pulse Width Modulation) หรืออาจกล่าวได้ว่าพอร์ต C เป็นพอร์ตสำหรับเชื่อมต่ออุปกรณ์ภายนอก (peripheral function port) ที่มีความสมบูรณ์แบบมากที่สุด

พอร์ต D

มี 8 บิต กำหนดชื่อขาเป็น **RD0-RD7** รีจิสเตอร์ที่ใช้เก็บข้อมูลคือ **PORTD** มีแอดเดรสอยู่ที่ 0x08 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ **TRISD** ซึ่งมีแอดเดรสอยู่ที่ 0x88 (แบงก์ 1) มีขนาด 8 บิต หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล "0" ไปยังบิตนั้น

พอร์ต D สามารถใช้งานได้ 2 โหมดคือ เป็นขาพอร์ตอินพุตเอาต์พุตปกติและเป็นส่วนขยายพอร์ตแบบขนาน (Parallel Slave Port : PSP) สำหรับเชื่อมต่อกับอุปกรณ์ภายนอกที่มีการจัดระบบบัสแบบไมโครโปรเซสเซอร์คือ มีสายข้อมูล 8 เส้น สายสัญญาณควบคุม 3 เส้นคือ สายสัญญาณควบคุมการอ่าน (RD : Read), เขียน (WR : Write) และเลือกอุปกรณ์ (CS : Ship Select)

การเลือกโหมดการทำงานของพอร์ต D นี้ขึ้นกับบิต PSPMODE (บิต 4 ในรีจิสเตอร์ TRISE) ถ้าเป็น "0" เป็นการกำหนดให้พอร์ต D เป็นพอร์ตปกติ และถ้าเป็น "1" พอร์ต D จะทำงานในโหมด PSP

พอร์ต E

มี 3 บิต กำหนดชื่อขาเป็น **RE0-RE2** รีจิสเตอร์ที่ใช้เก็บข้อมูลคือ **PORTE** มีแอดเดรสอยู่ที่ 0x09 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานเพียง 3 บิตล่างคือ บิต0-บิต2 เท่านั้น ที่เหลือกำหนดให้เป็น "0" ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ **TRISE** ซึ่งมีแอดเดรสอยู่ที่ 0x89 (แบงก์ 1) มีขนาด 8 บิต โดยใช้ 3บิตล่างในการกำหนดทิศทางของพอร์ต E ส่วนที่เหลือใช้ควบคุมการทำงานในโหมด PSP ของพอร์ต D

2.1.3 โมดูลสื่อสารข้อมูลอนุกรมในไมโครคอนโทรลเลอร์ PIC16F877A

USART (Universal Synchronous Asynchronous Receiver Transmitter) เป็นโมดูลสื่อสารข้อมูลอนุกรมเอนกประสงค์ที่สามารถทำงานเป็นตัวรับหรือส่งข้อมูลในแบบอะซิงโครนัสหรือซิงโครนัสก็ได้ ทั้งยังสามารถกำหนดรูปแบบของข้อมูลได้ เลือกอัตราเร็งในการถ่ายทอดข้อมูลได้

เอกสารนี้ผลิตขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ คัดลอก ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดให้ทำงานแบบฟูลดูเพล็กซ์อะซิงโครนัส (Full duplex asynchronous)จะสามารถรับ-ส่งข้อมูล 2 ทิศทางได้ในเวลาเดียวกัน ทั้งนี้เพื่อใช้ติดต่อกับพอร์ตอนุกรมของคอมพิวเตอร์หรืออุปกรณ์เทอร์มินอลแบบอื่น ถ้ากำหนดให้ทำงานเป็นโมดูลสื่อสารข้อมูลอนุกรมแบบซิงโครนัส จะสามารถถ่ายทอดข้อมูลในลักษณะฮาล์ฟดูเพล็กซ์ กล่าวคือสามารถรับ – ส่งข้อมูล 2 ทิศทางได้ แต่ต้องสลับกันรับและส่ง การใช้งานในลักษณะนี้คือ ติดต่อกับอุปกรณ์ต่อพ่วง อาทิ หน่วยความจำอีอีพรอมอนุกรม ไอซีแปลงสัญญาณอะนาลอกเป็นดิจิตอล หรือแปลงสัญญาณดิจิตอลเป็นอะนาลอก เป็นต้น

โมดูล USART ใน PIC16F877A สามารถกำหนดการทำงานได้ 3 โหมดคือ

1. โหมดอะซิงโครนัส สามารถรับส่งข้อมูลในลักษณะฟูลดูเพล็กซ์
2. โหมดซิงโครนัส – มาสเตอร์ รับส่งข้อมูลในลักษณะฮาล์ฟดูเพล็กซ์
3. โหมดซิงโครนัส – สเลฟ รับส่งข้อมูลในลักษณะฮาล์ฟดูเพล็กซ์

การรับส่งข้อมูลในโหมดอะซิงโครนัส

สรุปขั้นตอนการกำหนดโมดูล USART ส่งข้อมูลในโหมดอะซิงโครนัสดังนี้

1. กำหนดข้อมูลสร้างบอดเรตลงใน SPBRG หากใช้บอดเรตความเร็วสูงต้องเซตบิต BRGH
2. เลือกฟังก์ชันการติดต่อผ่านพอร์ตอนุกรมและกำหนดให้ทำงานในโหมดอะซิงโครนัส โดยการเซตบิต SPEN และเคลียร์บิต SYNC
3. ถ้าต้องการให้เกิดอินเตอร์รัปต์ ต้องทำการเซตบิต TXIE
4. ถ้าเลือกการส่งข้อมูลแบบ 9 บิตให้ทำการเซตบิต TX9
5. เอ็นเอเบิลการส่งข้อมูลโดยเซตบิต TXEN ส่งผลให้แฟลก TXIF ถูกเซตตามไปด้วย
6. ถ้าหากเลือกการส่งข้อมูลแบบ 9 บิต ข้อมูล 8 บิต หรือบิตที่ 9 จะถูกส่งไปยังบิต TX9D
7. กำหนดข้อมูลลงในรีจิสเตอร์ TXREG เพื่อเริ่มต้นส่งข้อมูล

สรุปขั้นตอนการกำหนดโมดูล USART รับข้อมูลในโหมดอะซิงโครนัสดังนี้

1. กำหนดข้อมูลสร้างบอดเรตลงใน SPBRG หากใช้บอดเรตความเร็วสูงต้องเซตบิต BRGH
2. เลือกฟังก์ชันการติดต่อผ่านพอร์ตอนุกรมและกำหนดให้ทำงานในโหมดอะซิงโครนัส โดยการเซตบิต SPEN และเคลียร์บิต SYNC
3. ถ้าต้องการให้เกิดอินเตอร์รัปต์ ต้องทำการเซตบิต RCIE
4. ถ้าเลือกการรับข้อมูลแบบ 9 บิตให้ทำการเซตบิต RX9
5. เอ็นเอเบิลการรับข้อมูลโดยเซตบิต CREN
6. บิตแฟลก RCIF ถูกเซตเมื่อรับข้อมูลเสร็จสิ้น จะเกิดอินเตอร์รัปต์หากเอ็นเอเบิลไว้
7. อ่านข้อมูลของรีจิสเตอร์ RCSTA เพื่ออ่านข้อมูลบิตที่ 9 ถ้าหากเลือกการรับข้อมูลแบบ 9 บิต และตรวจสอบความผิดพลาดที่อาจเกิดขึ้นระหว่างการรับข้อมูล

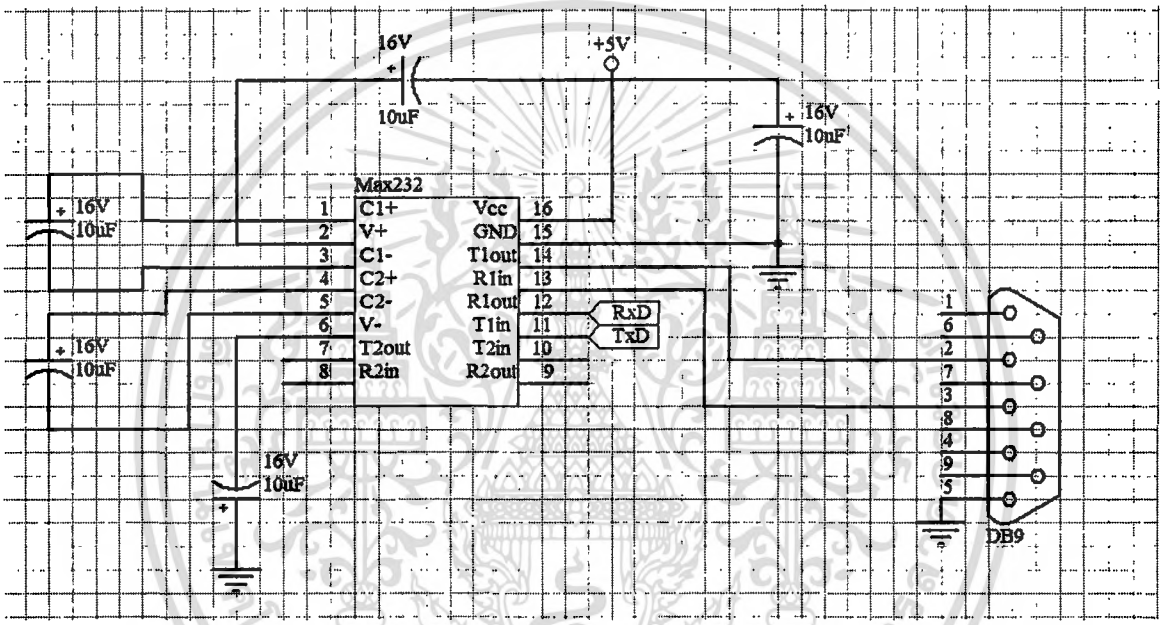
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. อ่านข้อมูล 8 บิตที่รับเข้ามา โดยผ่าทางรีจิสเตอร์ RCREG

9. ถ้าเกิดความผิดพลาดขึ้น ให้เคลียร์บิตแจ้งความผิดพลาดหรือบิต CREN

2.2 การส่งข้อมูล

การติดต่อระหว่างซอฟแวร์กับตัวเครนที่ใช้ในการเคลื่อนย้ายสินค้า จะใช้การติดต่อสื่อสารแบบอนุกรม จึงจำเป็นที่จะต้องใช้ IC MAX232 เป็นตัวกลางในการติดต่อสื่อสาร ซึ่ง IC ชนิดนี้จะทำให้อุปกรณ์รับและส่งทั้ง 2 ฝั่งเป็นค่าแรงดันที่ตรงกัน



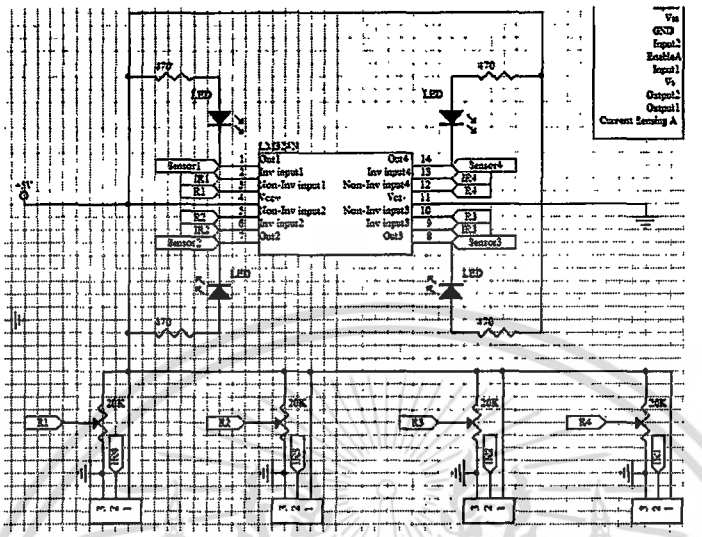
รูปที่ 2.2 แสดงวงจร IC MAX232

2.3 การตรวจจับตำแหน่ง

ส่วนตรวจจับของระบบใช้ในการตรวจจับตำแหน่งของคอลัมน์และชั้นของคลังสินค้า ทำงานโดยการกำหนดตำแหน่งโดยจะมีแถบเส้นสีดำติดไว้ที่ตัวคลังสินค้าในแต่ละคอลัมน์เพื่อใช้ในการตรวจเช็คตำแหน่งในแนวแกน X และที่เครนเพื่อใช้ในการตรวจเช็คตำแหน่งในแนวแกน Y ซึ่งแถบสีดำนี้จะถูกตรวจจับโดยเซนเซอร์ (Sensor) จะตรวจจับการสะท้อนของรังสีอินฟราเรด เซนเซอร์ตัวนี้จะประกอบด้วยส่วนรับและส่วนส่งคลื่นอินฟราเรด เมื่อเซนเซอร์ตรวจพบพื้นสีของเส้นที่ต่างกัน จะส่งสัญญาณที่แตกต่างกันออกมา แต่เนื่องจากสัญญาณที่ได้ยังไม่เห็นเป็นลอจิกที่น่าเชื่อถือ ดังนั้นต้องนำสัญญาณที่ได้ เข้าสู่วงจรเปรียบเทียบแรงดันก่อนที่จะส่งไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรเลอร์ วงจรเปรียบเทียบแรงดันนั้นจะใช้ออปแอมป์ (Op-Amp) ในไอซีเบอร์ LM324 ซึ่งมีโครงสร้างภายในดังรูปที่ 2.3



รูปที่ 2.3 แสดงวงจรตรวจจับตำแหน่ง

เมื่อเซนเซอร์ตรวจพบพื้นสีขาว แสงอินฟราเรดจากตัวส่งจะส่งสะท้อนเข้าสู่ตัวรับ ทำให้เกิดการนำกระแสแรงดันที่ขาอินเวอร์ตติ้ง(Inverting) ของออปแอมป์จะต่ำทำให้เอาต์พุทออกเป็นแรงดันสูง ซึ่งถือว่าเป็นลอจิก 1 ส่งให้กับไมโครคอนโทรเลอร์ จะทำให้หลอด LED ดับ

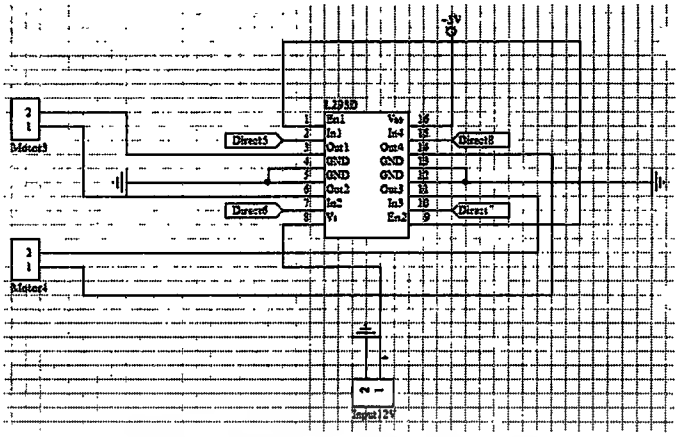
เมื่อเซนเซอร์ตรวจพบพื้นสีดำ แสงอินฟราเรดจากตัวส่งจะถูกดูดกลืนทำให้ตัวรับไม่นำกระแสแรงดันที่ขาอินเวอร์ตติ้งของออปแอมป์จะสูง ทำให้เอาต์พุตออกเป็นแรงดันต่ำ ซึ่งถือว่าเป็น Logic ศูนย์ ส่งให้กับไมโครคอนโทรเลอร์ จะทำให้หลอด LED ติด

2.4 การควบคุมมอเตอร์

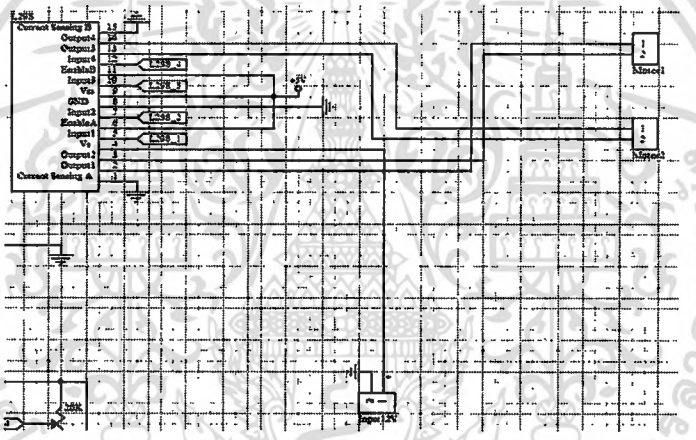
ในส่วนของเดรนจะเลือกใช้ DC Motor ในการขับเคลื่อน เนื่องจากสามารถควบความเร็วและทิศทางการหมุนได้ง่าย ซึ่งทิศทางการหมุนของมอเตอร์จะควบคุมได้โดยการควบคุมทิศทางการไหลของกระแสที่จ่ายให้กับมอเตอร์ โดยควบคุมผ่านไมโครคอนโทรลเลอร์

การควบคุมมอเตอร์โดยใช้ไมโครคอนโทรลเลอร์ ก่อนที่สัญญาณควบคุมจากไมโครคอนโทรลเลอร์จะไปควบคุมมอเตอร์ จะต้องมีการปรับขนาดโดยใช้ไอซีเบอร์ L293D สำหรับใช้ควบคุมทิศทางการหมุนของมอเตอร์ในแนวแกน Z และไอซีเบอร์ L298 สำหรับใช้ควบคุม ทิศทางการหมุนของมอเตอร์ในแกน X และ แกน Y ดังรูปที่ 2.4 และรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงวงจรจรของ L293D



รูปที่ 2.5 แสดงวงจรของ L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

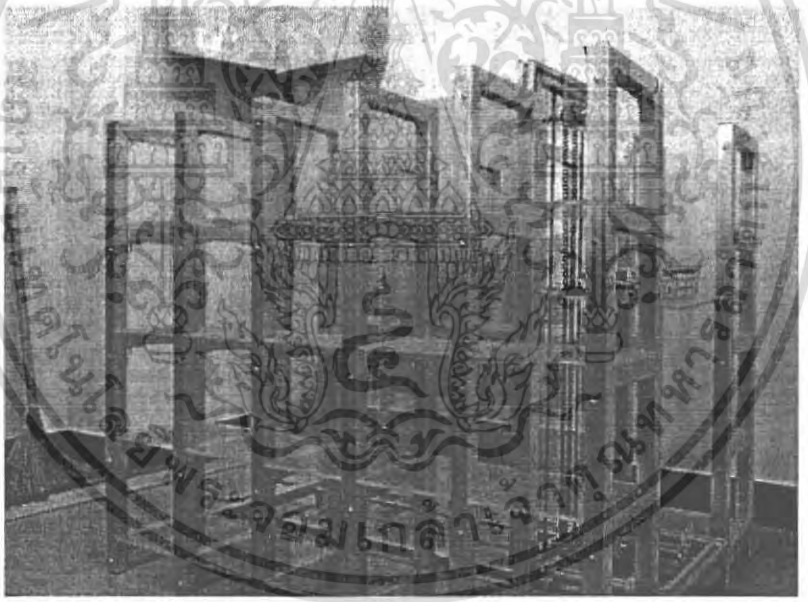
บทที่ 3
หลักการออกแบบ

3.1 การออกแบบในส่วนของฮาร์ดแวร์

ในส่วนของฮาร์ดแวร์จะประกอบไปด้วย 2 ส่วน คือ คลังสินค้า และเครน

3.1.1 โครงสร้างของคลังสินค้า

โครงสร้างของคลังสินค้า เป็นช่องสำหรับใช้เก็บชิ้นงาน โดยมีขนาด 78x90.5x20 เซนติเมตร ซึ่งทำมาจากอะลูมิเนียมกล่อง ขนาด 1x1 นิ้ว และตัวพื้นในแต่ละชั้นทำมาจากอะคิลิกใส มีความหนา 5 มิลลิเมตร แบ่งออกเป็น 3 ชั้น แต่ละชั้นมี 5 ช่อง รวมทั้งหมด 15 ช่อง ใช้เก็บชิ้นงาน ขนาด 100x120x150 มิลลิเมตร คลังสินค้าจะมี 2 ฝั่งโดยแต่ละฝั่งจะห่างกัน 350 มิลลิเมตร ดังรูปที่ 3.1



รูปที่ 3.1 แสดงโครงสร้างของคลังสินค้า

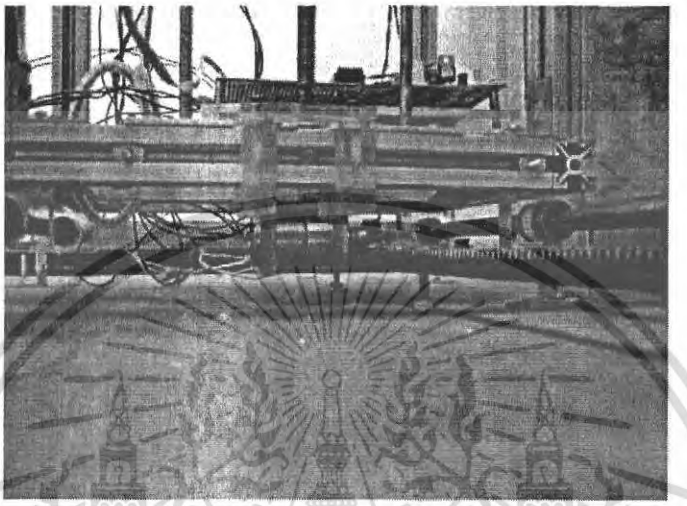
3.1.2 โครงสร้างของเครน

เครนที่ออกแบบมานั้นจะทำหน้าที่ในการนำสินค้าเข้าและออกจากคลังสินค้าตามที่ได้ป้อนคำสั่งไว้ โดยเครนจะสามารถเคลื่อนที่ได้ 3 แกน คือในแนวแกน X แกน Y และแกน Z ซึ่งในแต่ละแกนจะมีโครงสร้างดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1 ชุดเคลื่อนที่ในแนวแกน X

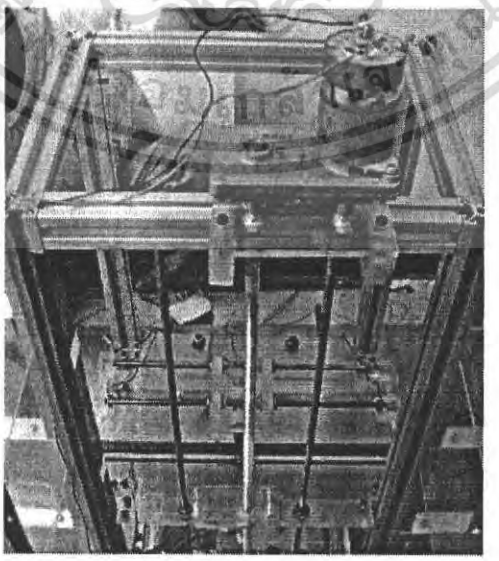
เป็นโครงสร้างของการเคลื่อนที่ไปตามรางในแนวราบ เพื่อนำสินค้าไปยังแถวที่ต้องการ โดยจะใช้มอเตอร์ไฟฟ้ากระแสตรงเป็นตัวขับตามเฟืองสะพานเพื่อให้เคลื่อนที่ไปตามตำแหน่งที่ต้องการ และจะมีการติดเซนเซอร์เพื่อใช้ในการตรวจเช็คว่าตัวเครนเคลื่อนที่ไปยังแถวใดแล้ว



รูปที่ 3.2 แสดงชุดเคลื่อนที่ในแนวแกน X

3.1.2.2 ชุดเคลื่อนที่ในแนวแกน Y

เป็นโครงสร้างของการเคลื่อนที่ให้ยกตัวขึ้นลงในแนวดิ่ง เพื่อนำสินค้าไปยังชั้นที่ต้องการ โดยใช้มอเตอร์หมุนสกรูให้ยกตัวขึ้นหรือลงตามตำแหน่งที่ต้องการ และจะมีการติดเซนเซอร์เพื่อใช้ในการตรวจเช็คว่าตัวเครนเคลื่อนที่ไปยังชั้นใดแล้ว

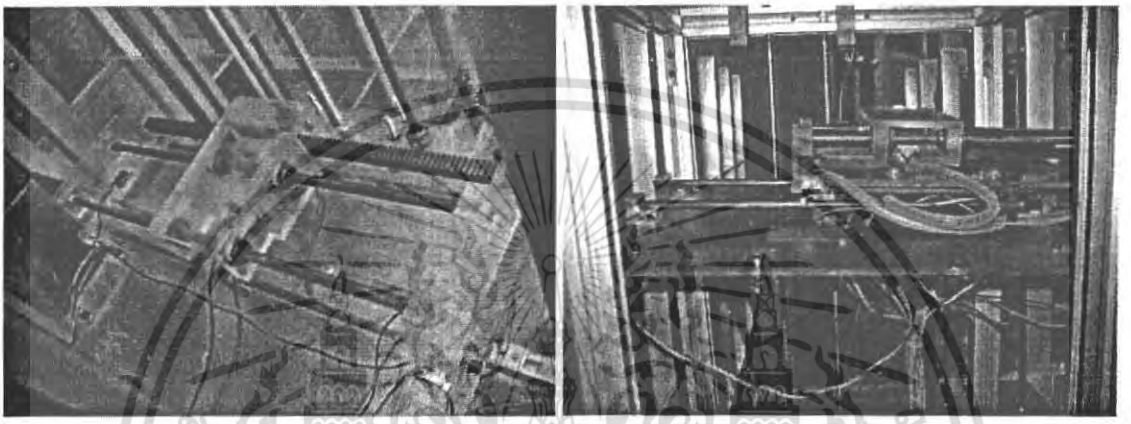


รูปที่ 3.3 ชุดเคลื่อนที่ในแนวแกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.3 ชุดเคลื่อนที่ในแนวแกน Z

เป็นโครงสร้างของการเคลื่อนที่ในแนวราบเพื่อนำสินค้าไปใส่ในช่องที่ต้องการ โดยในแนวแกน Z นี้จะมี 2 ชั้นเพื่อให้สามารถเลื่อนเข้าและออกได้ทั้งฝั่งซ้ายและฝั่งขวา ใช้มอเตอร์ดีซีขับตัวเฟืองสะพาน โดยชั้นแรกตัวมอเตอร์จะถูกยึดไม่ให้เคลื่อนที่ และชั้นที่ 2 จะให้มอเตอร์วิ่งไปบนเฟืองสะพาน และจะมีการติดลิมิตสวิตช์เพื่อใช้เป็นตัวตรวจสอบว่าชุดเคลื่อนที่ในแนวแกน Z ได้วิ่งไปสุดแล้ว



(ก) (ข)

รูปที่ 3.4 (ก) ชุดเคลื่อนที่ในแนวแกน Z เมื่อมองจากด้านบน
(ข) ชุดเคลื่อนที่ในแนวแกน Z เมื่อมองจากด้านข้าง

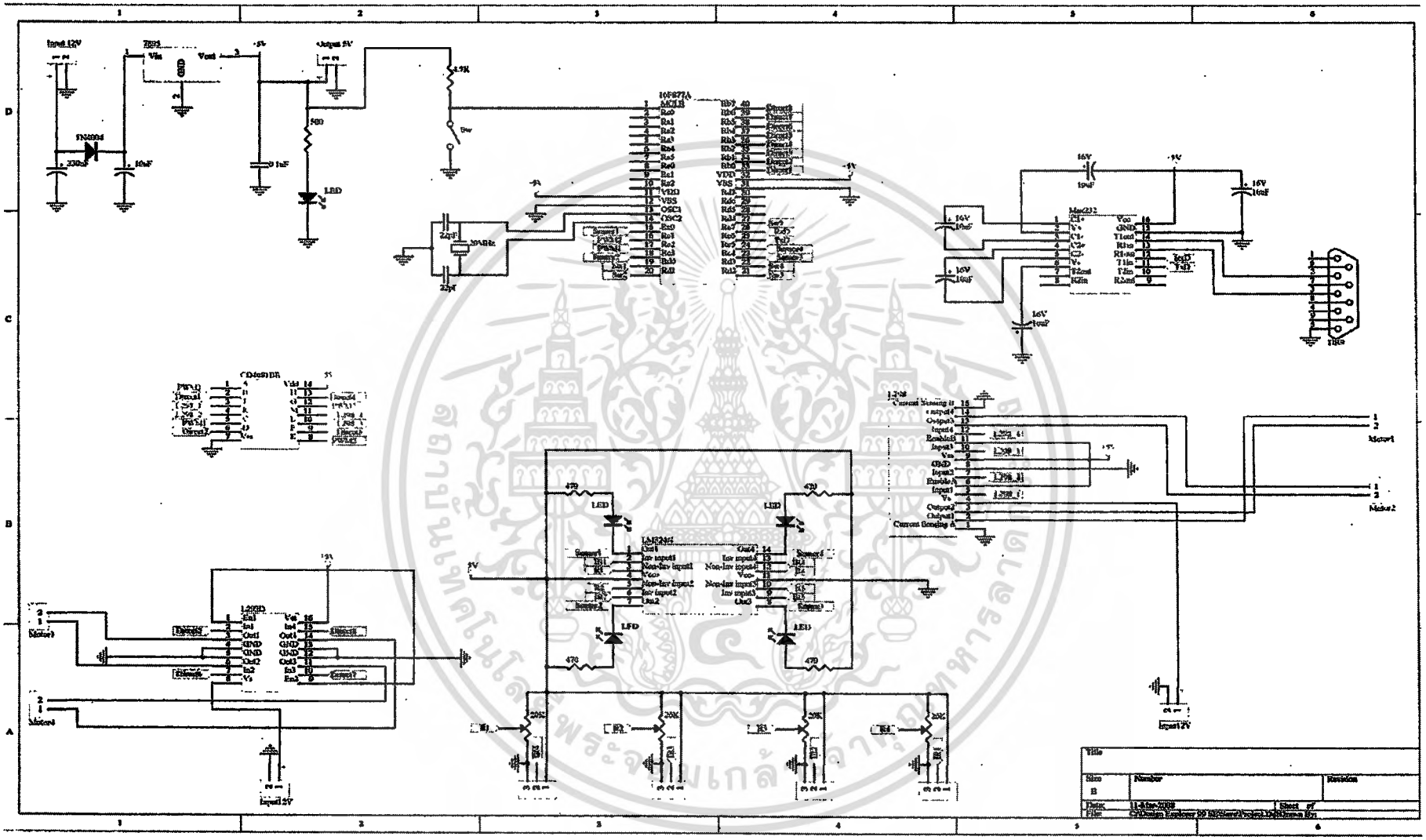
3.2 การออกแบบในส่วนของซอฟท์แวร์

ประกอบด้วย 2 ส่วนคือ ส่วนที่ใช้ในการควบคุมตัวเครนโดยไมโครคอนโทรลเลอร์ และ ส่วนที่ใช้ในการแสดงผลบนหน้าจอคอมพิวเตอร์

3.2.1 ส่วนที่ใช้ในการควบคุม

จะใช้ไมโครคอนโทรลเลอร์ PIC16F877A ในการควบคุมเครนอัตโนมัติ โดยจะนำมาต่อเข้ากับวงจรการส่งข้อมูล วงจรการตรวจจับตำแหน่ง และวงจรควบคุมดีซีมอเตอร์ ดังรูปที่ 3.5 แบ่งออกเป็น 2 ส่วน คือ การนำสินค้าเข้าไปในคลังสินค้า และการนำสินค้าออกจากคลังสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงวงจรของ PIC16F877A

3.2.1.1 การนำสินค้าเข้าไปในคลังสินค้า

เมื่อผู้ใช้งานต้องการนำสินค้าเข้าไปจัดเก็บในคลังสินค้า โดยสามารถสั่งการผ่านทางโปรแกรมแอฟพลิเคชั่นบนคอมพิวเตอร์ ซึ่งก่อนการใช้งานผู้ใช้งานจะต้องทำการ Login เข้าโปรแกรมแอฟพลิเคชันดังกล่าว เมื่อเราต้องการนำสินค้าจัดเก็บในคลังสินค้า จากโปรแกรมบนคอมพิวเตอร์จะมีช่อง ID ซึ่งให้เราเลือกว่าสินค้าชนิดนั้นๆ มี ID number อะไร และหลังจากนั้น ในช่อง Product ให้เราเลือกว่า สินค้าที่จัดเก็บเป็นสินค้าชนิดไหน และเมื่อเราทำการเลือก ID และ Product เสร็จเรียบร้อยแล้ว เราสามารถที่เลือกการจัดเก็บให้อยู่ทางคลังสินค้าทางด้านซ้ายหรือด้านขวา โดยสามารถเลือกคลิ๊กตรงตารางด้านขวามือของโปรแกรมแอฟพลิเคชั่นได้โดยตรง หลังจากเราระบุการเก็บสินค้าเรียบร้อยแล้ว ให้เราทำการคลิ๊กที่ปุ่ม Trade in โปรแกรมก็จะส่งข้อมูลผ่านทางระบบ Rs232 (serial port) เพื่อที่จะสั่งการไปยังไมโครคอนโทรลเลอร์ที่ควบคุมตัวเครนอัตโนมัติอีกต่อไป

เมื่อไมโครคอนโทรลเลอร์ได้รับข้อมูลจากคอมพิวเตอร์แล้ว ไมโครคอนโทรลเลอร์ก็จะทำการประมวลผลว่าจะให้ตัวเครนอัตโนมัติเคลื่อนที่ไปยัง ณ จุดใดของคลังสินค้า โดยไมโครคอนโทรลเลอร์จะทำการสั่งการและควบคุม IC L298 เพื่อให้มอเตอร์ในแนวแกน X และแกน Y ทำงาน โดยการเคลื่อนที่ของตัวเครนอัตโนมัตินั้นจะมีการเช็คตำแหน่งโดยตัวอินฟราเรด เมื่อตัวเครนอัตโนมัติเคลื่อนที่ผ่านแต่แถวและแต่ละชั้นของตัวคลังสินค้า ตัวตรวจจับอินฟราเรดก็จะทำการส่งข้อมูลไปให้ไมโครคอนโทรลเลอร์เพื่อให้เกิดการนับว่าตัวเครนอัตโนมัติเคลื่อนที่อยู่ ณ ตำแหน่งแถวและชั้นที่ถูกต้องแล้วหรือยัง เมื่อตัวเครนอัตโนมัติเคลื่อนที่มายังแถวและชั้นที่ผู้ใช้งานเลือกไว้ มอเตอร์ที่เคลื่อนที่ในแนวแกน X และแกน Y ก็จะหยุดทำงาน ซึ่งต่อไปจะเป็นการทำงานของแกน Z เพื่อที่จะนำสินค้าเข้าไปวาง โดยตัวไมโครคอนโทรลเลอร์จะสั่งการและควบคุม IC L293D เพื่อทำการขับมอเตอร์ในแกน Z เมื่อแกน Z ชั้นที่ 1 ได้เลื่อนเข้าไปชน Limit Switch ตัวที่ 1 จะทำให้มอเตอร์แกน Z ตัวที่ 1 หยุดทำงาน และไมโครคอนโทรลเลอร์จะสั่งให้มอเตอร์ในแกน Z ตัวที่ 2 เคลื่อนที่ เมื่อแกน Z ชั้นที่ 2 เลื่อนเข้าไปชน Limit Switch ตัวที่ 2 ซึ่งตัวเครนอัตโนมัติก็จะทำการวางสินค้าในคลังสินค้านั้น หลังจากวางสินค้าในคลังสินค้าเรียบร้อยแล้ว ตัวเครนก็จะเลื่อนกลับมายังตำแหน่งเริ่มต้นตามเดิม เพื่อคอยรับคำสั่งจากผู้ใช้งานต่อไป

3.2.1.1 การนำสินค้าออกจากคลังสินค้า

เมื่อผู้ใช้งานต้องการนำสินค้าออกจากตัวคลังสินค้า โดยผู้ใช้งานสามารถสั่งการผ่านทางโปรแกรมแอฟพลิเคชั่นบนคอมพิวเตอร์ ซึ่งก่อนการใช้งานผู้ใช้งานจะต้องทำการ Login เข้าโปรแกรมแอฟพลิเคชันดังกล่าว เมื่อเราต้องการนำสินค้าออกจากตัวคลังสินค้า จากโปรแกรมบนคอมพิวเตอร์ เราสามารถที่จะเลือกสินค้าออกโดย สามารถเลือกที่ ID ของสินค้า หรือ Product ของสินค้า หรือ โดยการคลิ๊กเลือกจากตารางที่แสดงสินค้าบนโปรแกรมแอฟพลิเคชันทางด้าน

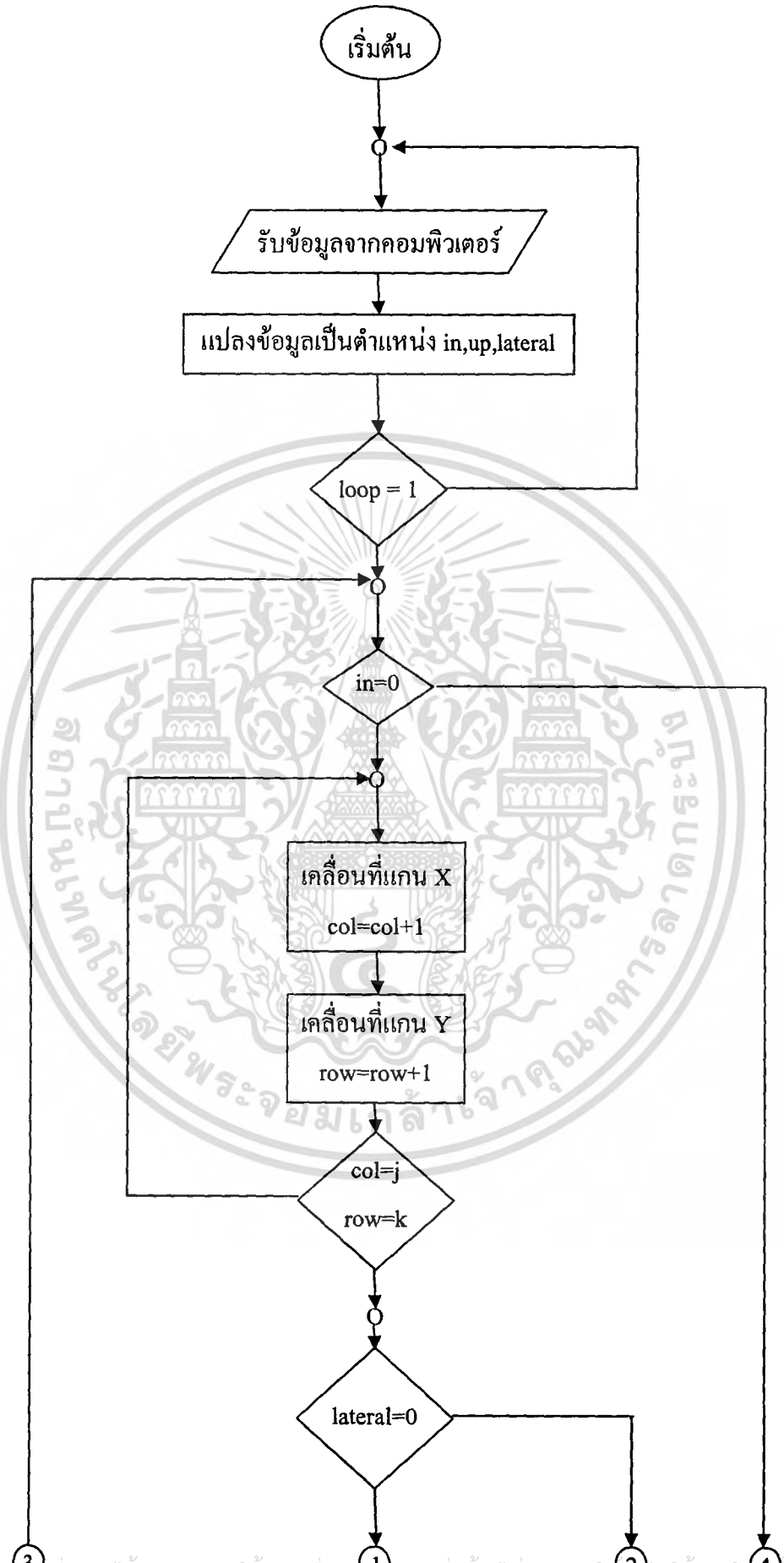
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขวามือ โดยเลือกอย่างใดอย่างหนึ่งจากโปรแกรม เมื่อเลือกสินค้าที่ต้องการนำออกมาเป็นที่เรียบร้อยแล้ว ให้เราคลิ๊กที่ปุ่ม Trade Out จากโปรแกรม โปรแกรมแอฟพลิเคชั่นก็จะส่งข้อมูลผ่านทางระบบ Rs232 (serial port) เพื่อที่จะติดต่อและสั่งการไมโครคอนโทรเลอร์ที่ควบคุมตัวเครนอัตโนมัติอีกต่อไป

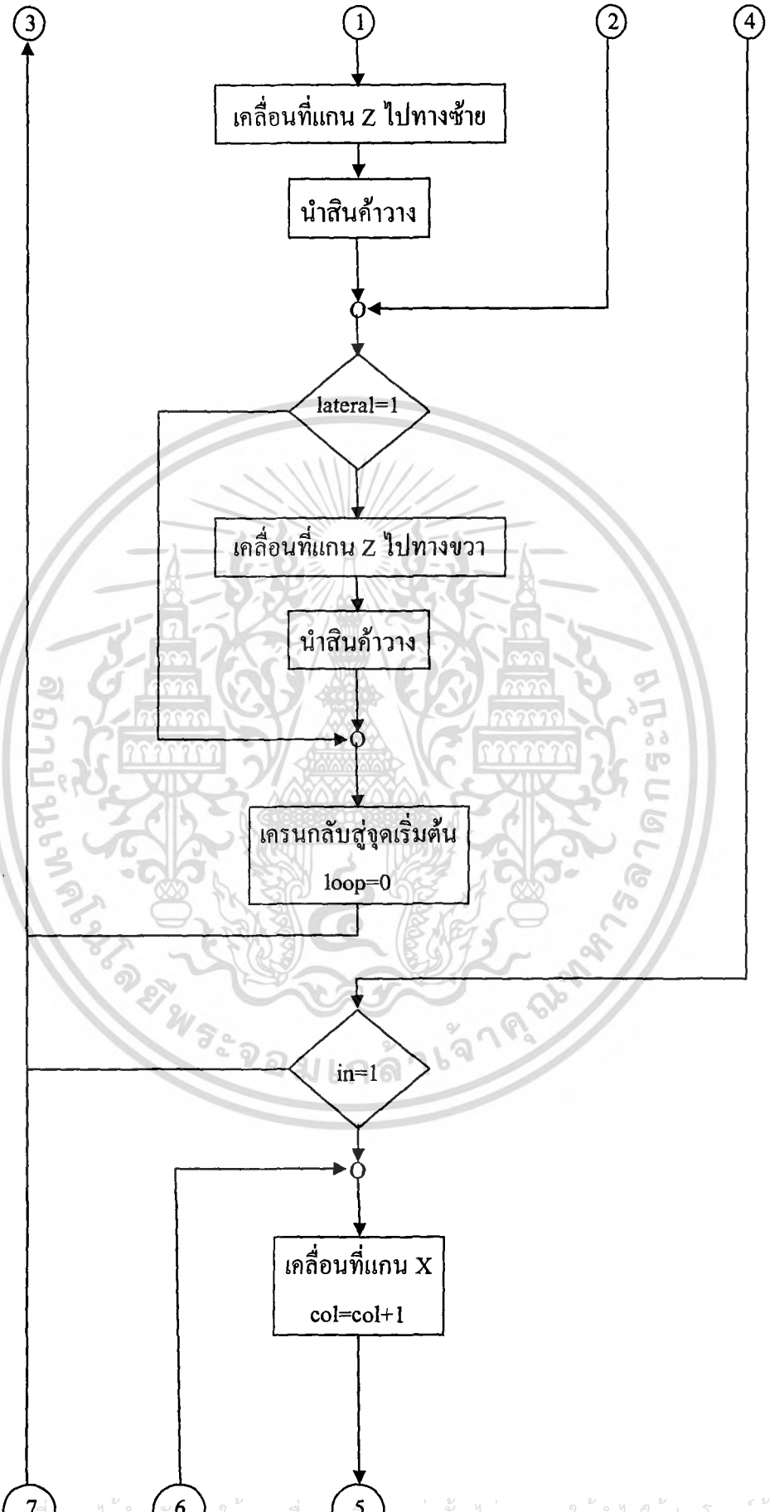
เมื่อไมโครคอนโทรลเลอร์ได้รับข้อมูลจากคอมพิวเตอร์แล้ว ไมโครคอนโทรลเลอร์ก็จะทำการประมวลผลว่าจะให้ตัวเครนอัตโนมัติเคลื่อนที่ไปยัง ณ จุดใดของคลังสินค้า โดยไมโครคอนโทรลเลอร์จะทำการสั่งการและควบคุม IC L298 เพื่อให้มอเตอร์ในแนวแกน X และแกน Y ทำงาน โดยการเคลื่อนที่ของตัวเครนอัตโนมัตินั้นจะมีการเช็คตำแหน่งโดยตัวอินฟราเรด เมื่อตัวเครนอัตโนมัติเคลื่อนที่ผ่านแต่แถวและแต่ละชั้นของตัวคลังสินค้า ตัวตรวจจับอินฟราเรดก็จะทำการส่งข้อมูลไปให้ไมโครคอนโทรลเลอร์เพื่อให้เกิดการนับว่าตัวเครนอัตโนมัติเคลื่อนที่อยู่ ณ ตำแหน่งแถวและชั้นที่ถูกต้องแล้วหรือยัง เมื่อตัวเครนอัตโนมัติเคลื่อนที่มายังแถวและชั้นที่ผู้ใช้งานเลือกไว้ มอเตอร์ที่เคลื่อนที่ในแนวแกน X และแกน Y ก็จะหยุดทำงาน ซึ่งต่อไปจะเป็นการทำงานของแกน Z เพื่อที่จะนำสินค้าออกจากตัวคลังสินค้า โดยตัวไมโครคอนโทรลเลอร์จะสั่งการและควบคุม IC L293D เพื่อทำการขับมอเตอร์ในแกน Z เมื่อแกน Z ชั้นที่ 1 ได้เลื่อนเข้าไปชน Limit Switch ตัวที่ 1 จะทำให้มอเตอร์แกน Z ตัวที่ 1 หยุดทำงาน และไมโครคอนโทรลเลอร์จะสั่งให้มอเตอร์ในแกน Z ตัวที่ 2 เคลื่อนที่ เมื่อแกน Z ชั้นที่ 2 เลื่อนเข้าไปชน Limit Switch ตัวที่ 2 เมื่อแกน Z ตัวที่ 1 และ 2 เลื่อนเข้าไปอยู่ใต้สินค้าที่จะนำออกมาแล้ว ไมโครคอนโทรลเลอร์ก็จะทำการสั่งให้แกน Y เลื่อนขึ้นเล็กน้อย เพื่อเป็นการยกสินค้าขึ้นจากตัวคลังสินค้า ซึ่งตัวเครนอัตโนมัติก็จะทำการนำสินค้าออกจากคลังสินค้าในตำแหน่งนั้นออกมา และตัวเครนก็จะเลื่อนกลับมายังตำแหน่งเริ่มต้นตามเดิมโดยนำสินค้าออกมาพร้อมกับตัวเครนอัตโนมัติ และคอยรับคำสั่งจากผู้ใช้งานต่อไป

ขั้นตอนการทำควบคุมเครนอัตโนมัติโดยใช้ไมโครคอนโทรลเลอร์ PIC16F877A จะมีขั้นตอนการทำงานดังรูปที่ 3.6

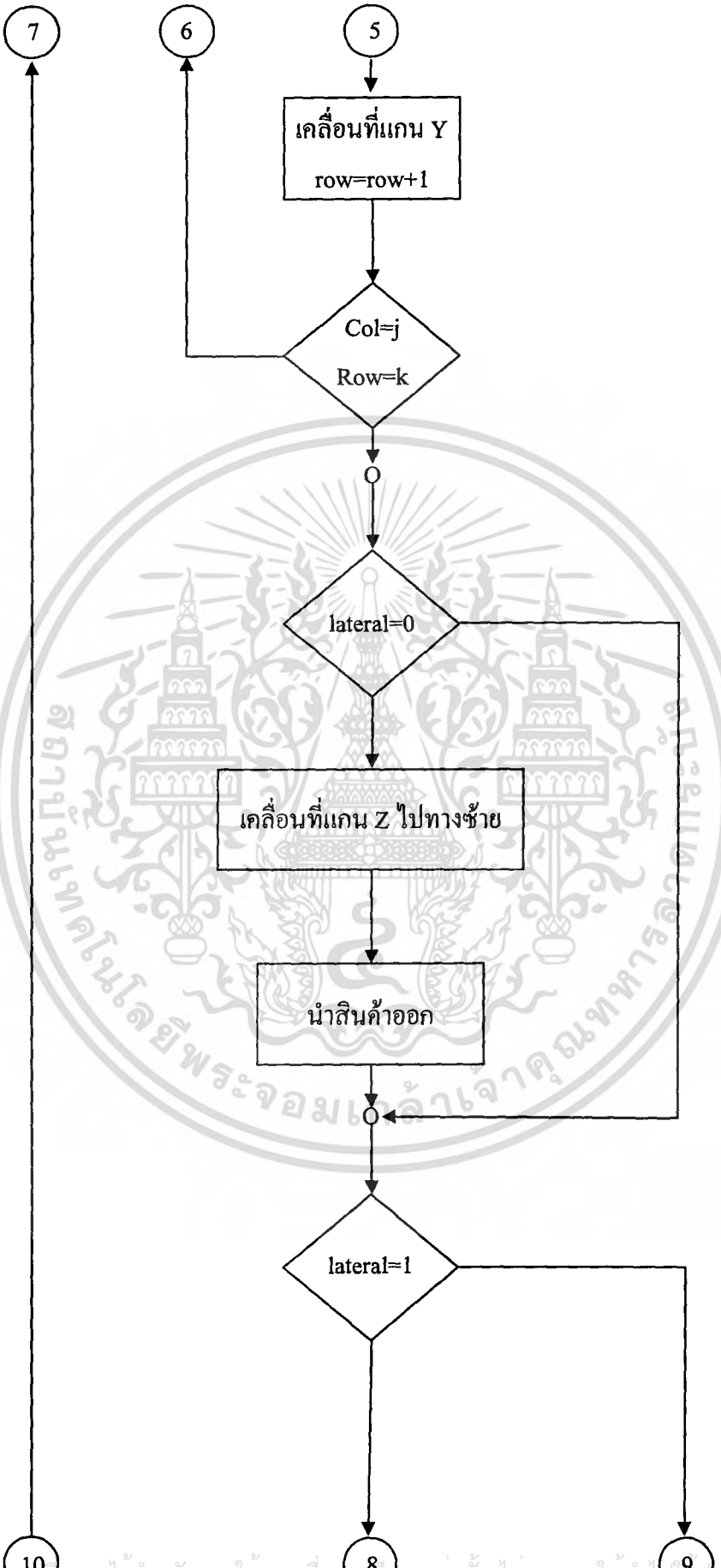
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



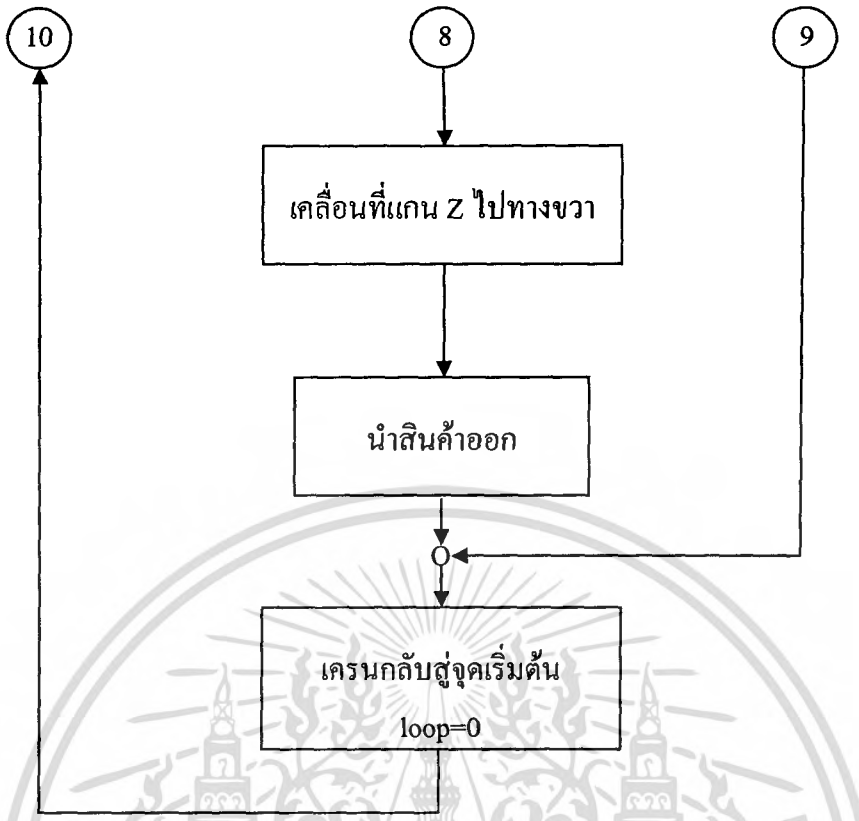
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 โฟลวชาร์ตแสดงการทำงานของระบบควบคุม

3.2.2 ส่วนที่ใช้ในการแสดงผลบนหน้าจอคอมพิวเตอร์

ในการออกแบบโครงงานนี้ได้นำการประยุกต์การใช้งานโปรแกรม Visual C# 2005 เป็นเครื่องมือที่ช่วยในการเขียนโปรแกรมในภาษา C# ซางภาษา C# เป็นภาษาโปรแกรมยุกต์ใหม่ที่ถูกสร้างขึ้นมาสำหรับการพัฒนาซอฟแวร์ภายใต้เทคโนโลยี Microsoft .NET ซึ่งเป็นเทคโนโลยี ที่ได้รับความนิยมสูงที่สุดในปัจจุบัน Visual C# นั้นถูกแนะนำครั้งแรกพร้อมกับ Visual Studio .NET ในชื่อของ Visual C# .NET ซึ่งก็ได้รับการตอบรับจากนักพัฒนาซอฟแวร์เป็นอย่างดี จนกระทั้งไมโครซอฟท์ได้พัฒนา Visual Studio 2005 ขึ้นมาทำให้ Visual C# ก็ได้รับการพัฒนาเป็น Visual C# 2005 และโปรแกรม Microsoft Access ในการจัดการด้านฐานข้อมูลของคลังสินค้าซึ่งในการออกแบบการควบคุมการจัดการระบบฐานข้อมูลจะแบ่งออกเป็นหัวข้อใหญ่ คือ

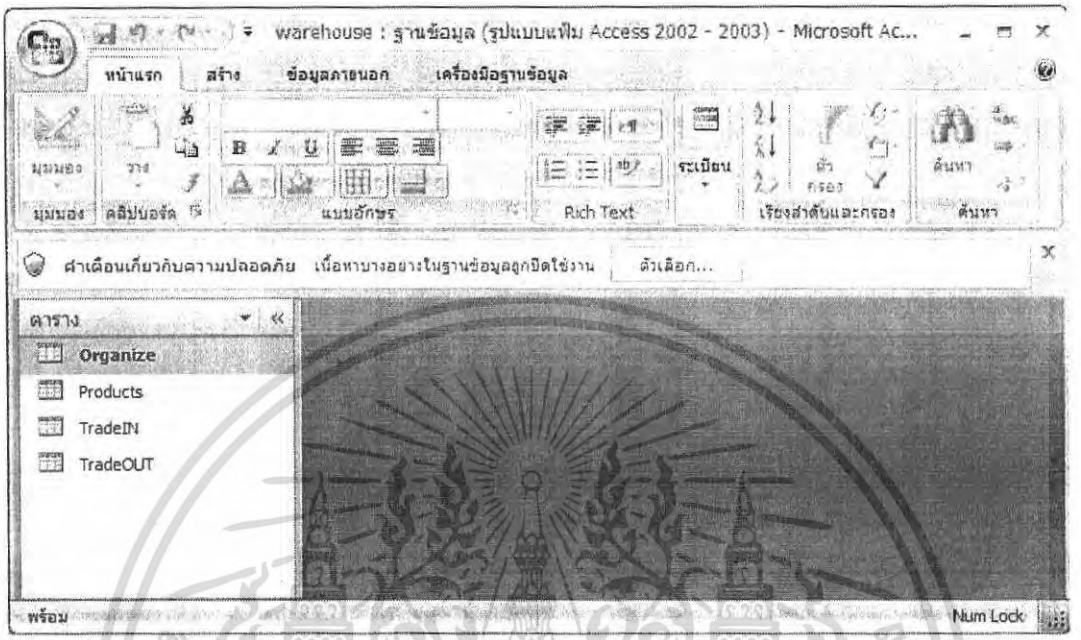
3.2.2.1 การออกแบบระบบฐานข้อมูล

การออกแบบระบบฐานข้อมูลได้มีการประยุกต์การใช้งาน โปรแกรม Microsoft Access ซึ่งมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลของคลังสินค้า

ในการออกแบบฐานข้อมูลของคลังสินค้า ผู้จัดทำได้กำหนดรูปแบบที่ใช้อ้างอิงถึงสินค้าที่สำคัญ ดังรูปที่ 3.7



รูปที่ 3.7 รูปตารางทั้งหมดที่ใช้ในคลังเก็บสินค้า

Organize เอาไว้ใช้เก็บรายละเอียดของผู้เข้ามาใช้งาน

Products เอาไว้ใช้เก็บรายละเอียดของสินค้าที่อยู่ใน Stock

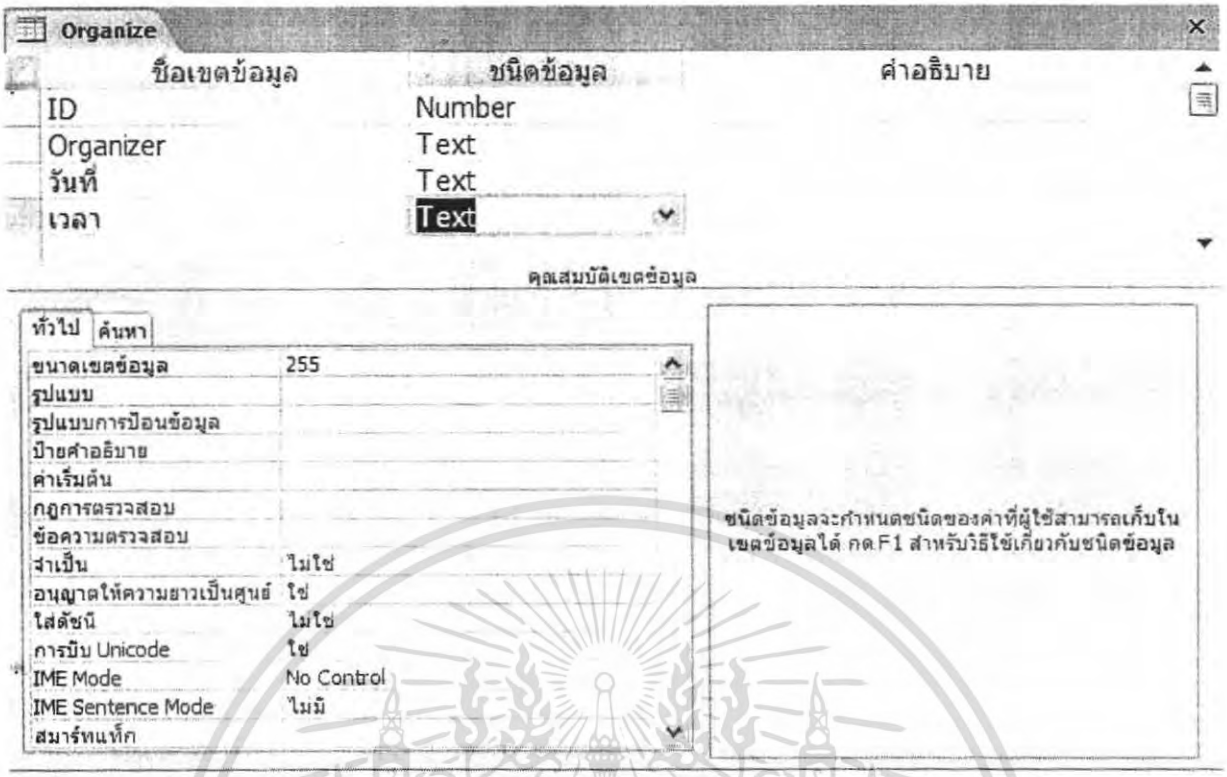
TradeIN เอาไว้ใช้เก็บรายละเอียดของสินค้าที่ได้ถูกเก็บใน Stock

TradeOUT เอาไว้ใช้เก็บรายละเอียดของสินค้าที่ได้ถูกนำออกมาจาก Stock

ตารางที่ 3.1 เอาไว้ใช้เก็บรายละเอียดของผู้เข้ามาใช้งาน

ฟิลด์	ชนิดข้อมูล	จุดประสงค์
ID	Text	ข้อมูลรหัสของผู้ใช้งาน
Organizer	Text	ข้อมูลชื่อของผู้ใช้งาน
วันที่	Text	วันที่เข้าใช้งาน
เวลา	Text	เวลาที่เข้าใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

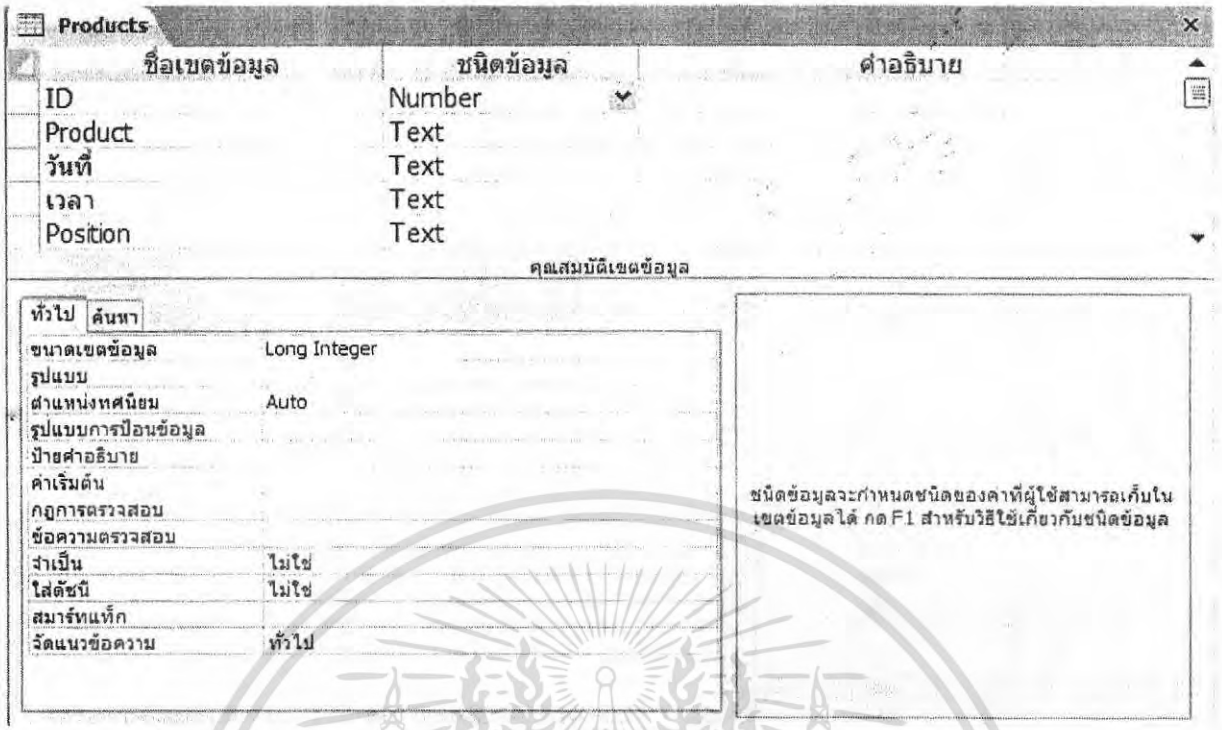


รูปที่ 3.8 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลผู้ใช้งาน

ตารางที่ 3.2 Products เอาไว้ใช้เก็บรายละเอียดของสินค้าที่อยู่ใน Stock

ฟิลด์	ชนิดข้อมูล	จุดประสงค์
ID	Text	ข้อมูลรหัสของสินค้า
Product	Text	ข้อมูลชื่อของสินค้า
วันที่	Text	วันที่นำสินค้าเข้าไปเก็บ
เวลา	Text	เวลาที่นำสินค้าเข้าไปเก็บ
Position	Text	ข้อมูลตำแหน่งที่เก็บสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

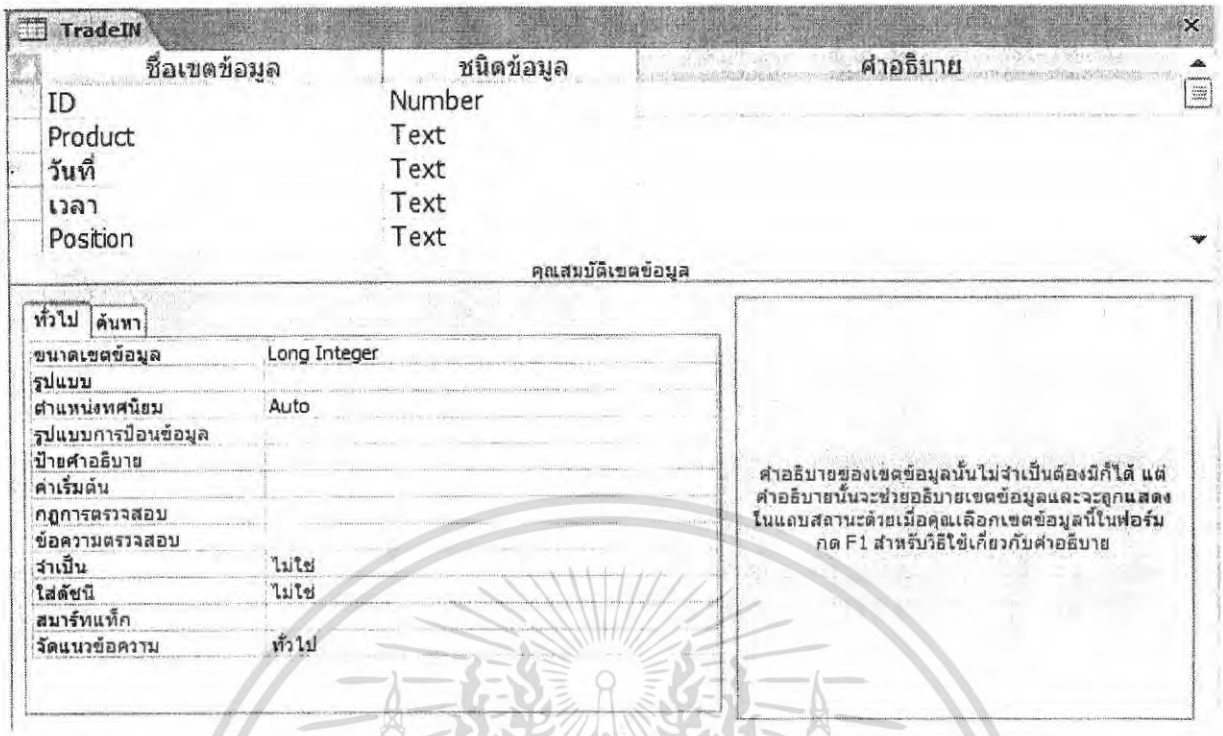


รูปที่ 3.9 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลที่เก็บในคลังสินค้า

ตารางที่ 3.3 TradeIN เอาไว้ใช้เก็บรายละเอียดของสินค้าที่ได้ถูกเก็บใน Stock

ฟิลด์	ชนิดข้อมูล	จุดประสงค์
ID	Text	ข้อมูลรหัสของสินค้า
Product	Text	ข้อมูลชื่อของสินค้า
วันที่	Text	วันที่นำสินค้าเข้าไปเก็บ
เวลา	Text	เวลาที่นำสินค้าเข้าไปเก็บ
Position	Text	ข้อมูลตำแหน่งที่เก็บสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

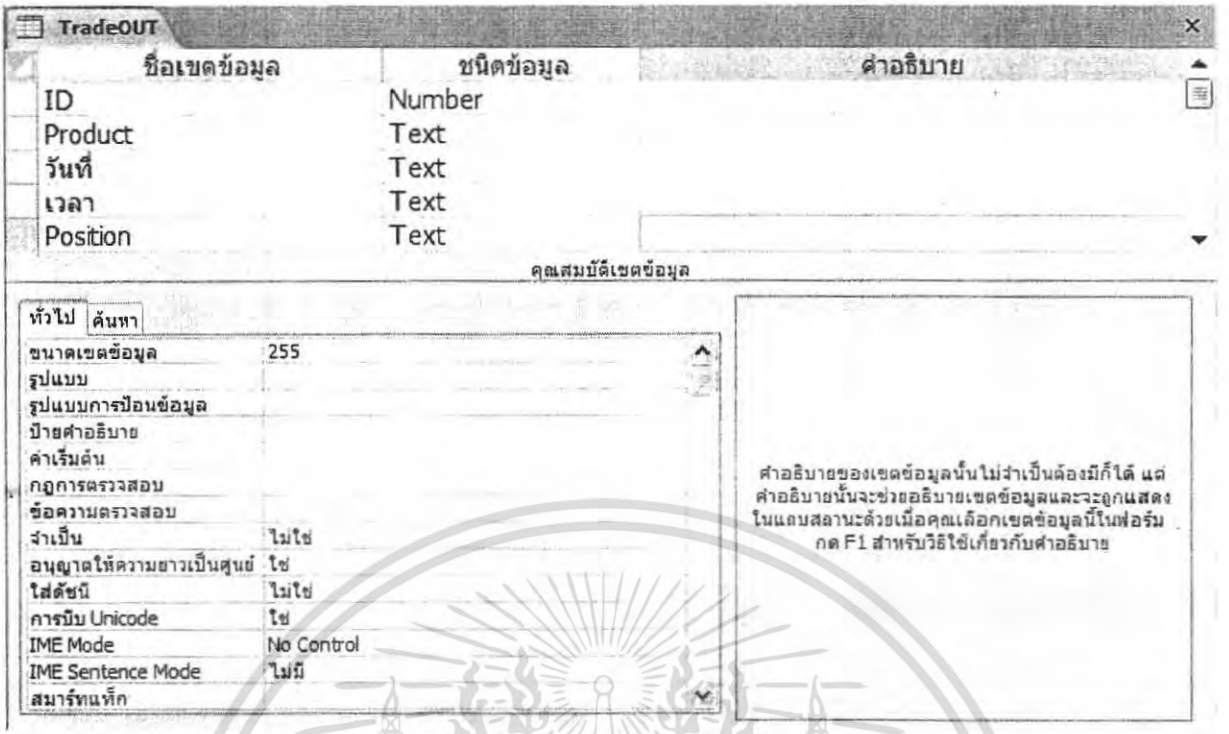


รูปที่ 3.10 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลที่เคยเก็บในคลังสินค้า

ตารางที่ 3.4 TradeOUT เอาไว้ใช้เก็บรายละเอียดของสินค้าที่ได้ถูกนำออกมาจาก Stock

ฟิลด์	ชนิดข้อมูล	จุดประสงค์
ID	Text	ข้อมูลรหัสของสินค้า
Product	Text	ข้อมูลชื่อของสินค้า
วันที่	Text	วันที่นำสินค้าเข้าไปเก็บ
เวลา	Text	เวลาที่นำสินค้าเข้าไปเก็บ
Position	Text	ข้อมูลตำแหน่งที่เก็บสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 รูปแบบฟิลด์ที่ใช้อ้างอิงถึงข้อมูลที่ได้ถูกนำออกมาจากคลังสินค้า

ส่วนของการเข้าสู่ระบบการเข้าไปใช้งาน

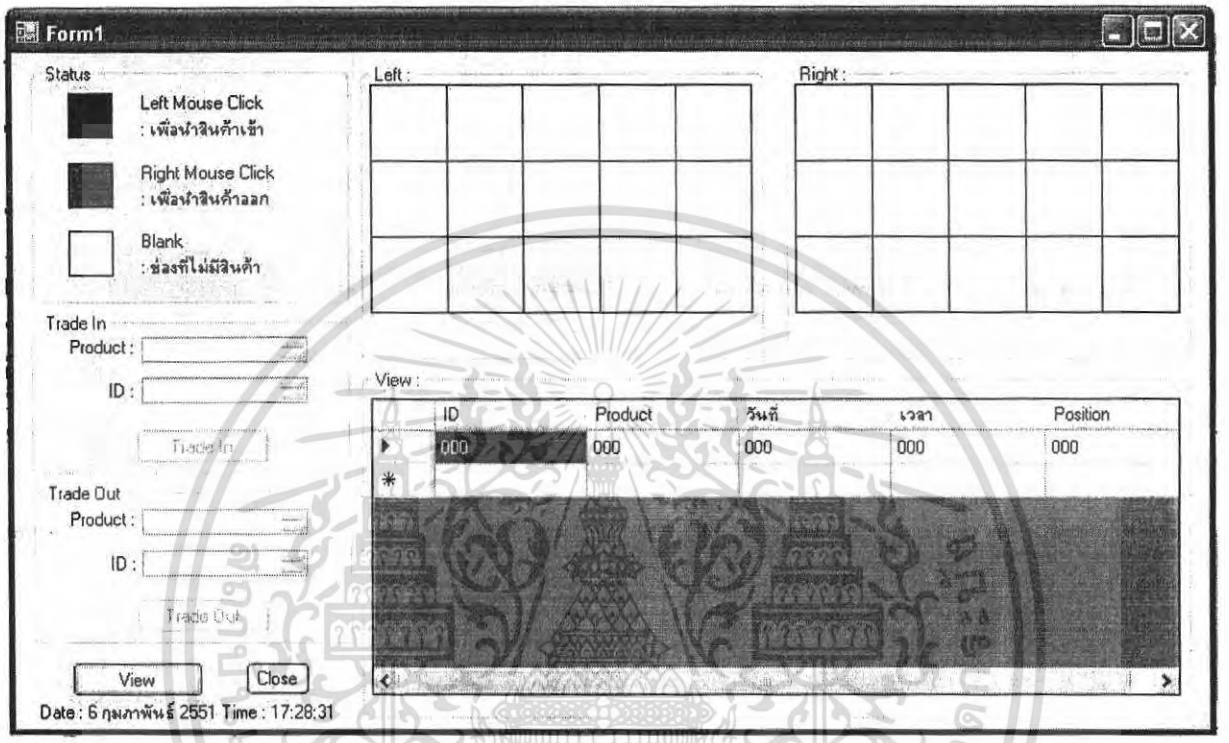
โปรแกรมได้ออกแบบให้ระบบมีการเก็บข้อมูลของผู้ใช้ก่อนใช้งานทุกครั้ง เพื่อเอาไว้ตรวจสอบผู้เข้าใช้ การเข้าสู่ระบบผู้ใช้จำเป็นต้องใส่ชื่อและรหัสพนักงานของตนพร้อมกันในการใช้ ซึ่งมีลักษณะดังรูปที่ 3.12

รูปที่ 3.12 แสดงหน้าจอเข้าใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

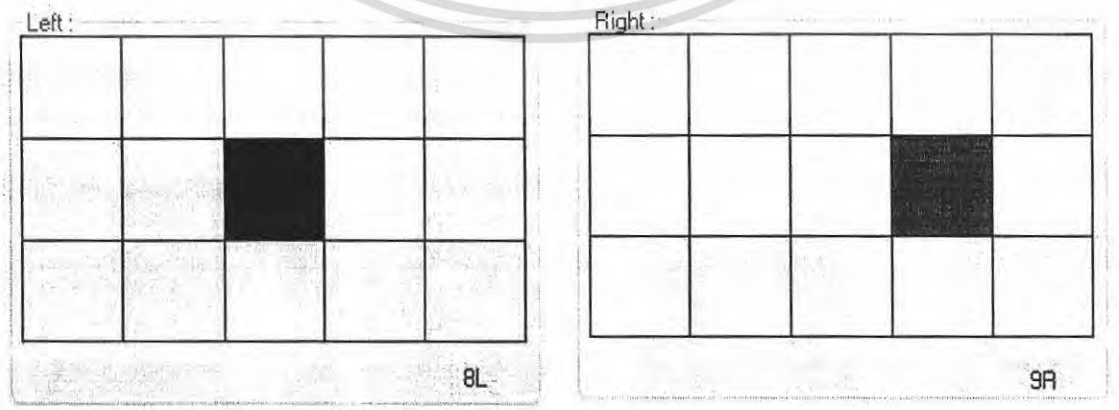
ส่วนการใช้งาน

โปรแกรมนี้ได้ถูกออกแบบมาเพื่อง่ายต่อการใช้งานจึงได้นำทั้งส่วนการขนถ่ายสินค้าเข้าและการขนถ่ายสินค้าออก รวมเอาไว้ด้วยกัน และมีการแสดงสถานะของสินค้าที่มีอยู่ในคลังสินค้าเพื่อง่ายต่อการเรียกใช้ โดยมีการแยกระบบการใช้งานโดยอธิบายในหัวข้อต่อไป



รูปที่ 3.13 แสดงหน้าจอระบบใช้งาน

ส่วนแสดงตำแหน่งของตารางที่มี 2 ฝั่ง ซ้ายและขวา เมื่อเลือกตำแหน่งใดจะมีข้อความระบุตำแหน่ง



รูปที่ 3.14 หน้าจอแสดงตำแหน่งทั้ง 2 ฝั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแสดงข้อมูลที่มีอยู่ในคลังเก็บสินค้า

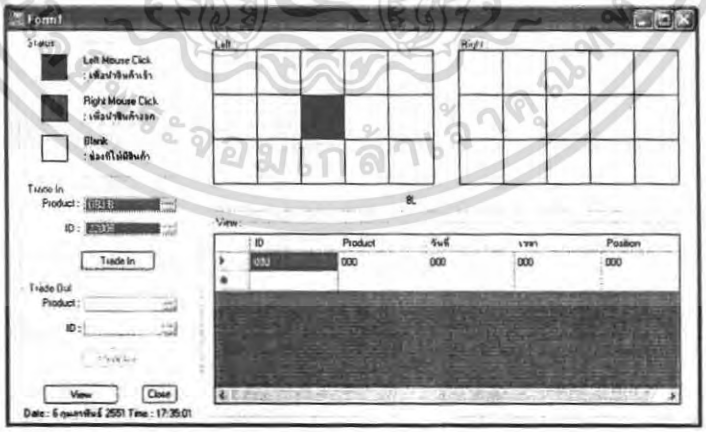
View :

	ID	Product	วันที่	เวลา	Position
▶	0	000	000	000	000
	12008	OBJ A	6 กุมภาพันธ์ 2551	22:03:36	EL
	12008	OBJ A	7 กุมภาพันธ์ 2551	4:25:02	FL
	12008	OBJ A	7 กุมภาพันธ์ 2551	4:25:08	7L
	22008	OBJ B	7 กุมภาพันธ์ 2551	4:27:19	CL
*					

รูปที่ 3.15 ข้อมูลสินค้าที่มีอยู่ในคลังเก็บสินค้า

ส่วนของการขนถ่ายสินค้าเข้า

ผู้ใช้งานจำเป็นต้องใส่ข้อมูล รหัสสินค้า ประเภทสินค้า และตำแหน่งของสินค้า โดยกดปุ่มเลือกตำแหน่งของสินค้า และได้แสดงผลสถานะของคลังสินค้าเอาไว้ทั้ง 2 ส่วน เพื่อให้ง่ายต่อผู้ใช้ในการเลือกตำแหน่ง จะมี Graphic แสดงสถานะ พร้อมทั้งยังเลือกตำแหน่งของสินค้าได้อีกด้วย ถ้าต้องการนำสินค้าเข้าไปยังตำแหน่งที่ต้องการให้คลิ๊กเม้าส์ซ้ายไปยังช่องที่เป็นสีเทา 1ครั้ง เป็นการแสดงสถานะ ว่าตำแหน่งนั้นยังว่างอยู่ และช่องที่ได้ทำการเลือกจะเปลี่ยนจากสีเทา เป็นสีดำ



รูปที่ 3.16 แสดงการทำงานถ่ายสินค้าเข้า

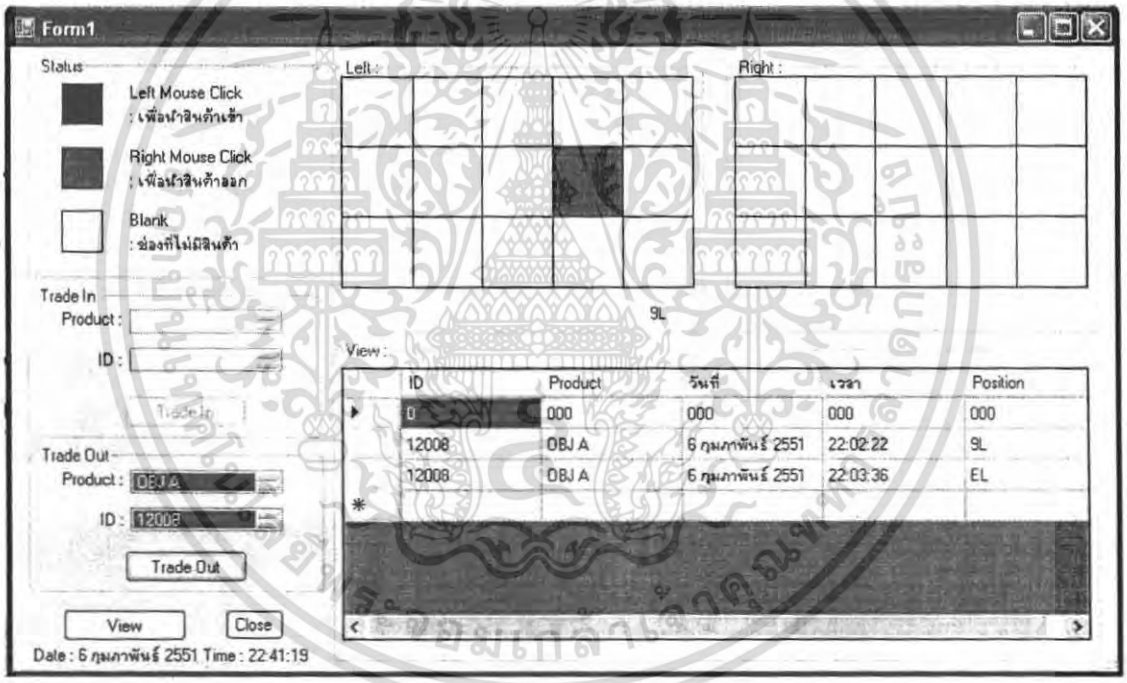
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : โปรแกรมได้ถูกออกแบบมาให้มีการรายงานความผิดเมื่อต้องการขนถ่ายสินค้าเข้าดังนี้

- ในกรณีผู้ใช้ระบุข้อมูลไม่ครบจะมีข้อความเตือน ให้เลือกข้อมูลให้ครบถ้วน
- ในกรณีผู้ใช้ไม่เลือกตำแหน่งการใช้งาน จะไม่สามารถใช้งานใดๆได้
- ผู้ใช้งานสามารถทำงานได้ครั้งละ 1 ตำแหน่งเท่านั้น

ส่วนของการขนถ่ายสินค้าออก

ผู้ใช้งานจำเป็นต้องใส่ข้อมูล รหัสสินค้า ประเภทสินค้า และตำแหน่งของสินค้า โดยกดปุ่มเลือกตำแหน่งของสินค้า และได้แสดงผลสถานะของคลังสินค้าเอาไว้ทั้ง 2 ส่วน เพื่อให้ง่ายต่อผู้ใช้ในการเลือกตำแหน่ง จะมี Graphic แสดงสถานะ พร้อมทั้งยังเลือกตำแหน่งของสินค้าได้อีกด้วย ถ้าต้องการนำสินค้าเข้าไปยังตำแหน่งที่ต้องการให้คลิ๊กเม้าส์ขวาไปยังช่องที่เป็นสีดำ 1ครั้ง เป็นตำแหน่งที่มีสินค้าเก็บอยู่ และช่องที่ได้ทำการเลือกจะเปลี่ยนจากสีดำ เป็นสีเขียว และเมื่อเสร็จกระบวนการทำงานจะกลับมาเป็นสีเทาเพื่อแสดงว่าช่องนั้นว่างอยู่



รูปที่ 3.17 แสดงการทำงานถ่ายสินค้าออก

หมายเหตุ : โปรแกรมได้ถูกออกแบบมาให้มีการรายงานความผิดเมื่อต้องการขนถ่ายสินค้าออกดังนี้

- ในกรณีผู้ใช้ระบุข้อมูลไม่ครบจะมีข้อความเตือน ให้เลือกข้อมูลให้ครบถ้วน
- ในกรณีผู้ใช้ไม่เลือกตำแหน่งการใช้งาน จะไม่สามารถใช้งานใดๆได้
- ผู้ใช้งานสามารถทำงานได้ครั้งละ 1 ตำแหน่งเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การเข้าสู่ระบบใช้งาน (Log in)

1. เปิดโปรแกรมเพื่อเริ่มต้นในส่วนการเข้าสู่ระบบใช้งาน
2. ใส่ชื่อและรหัสเพื่อตรวจสอบสิทธิ์ใช้งาน
3. คลิ๊กที่ปุ่ม "Enter"

รูปที่ 4.1 แสดงหน้าจอเข้าสู่ระบบใช้งาน

ในกรณีที่ผู้ใช้ไม่ใส่ชื่อหรือ รหัสผ่าน จะปรากฏกล่องข้อความดังรูป เพื่อให้ผู้ใช้ใส่ชื่อหรือรหัสผ่านให้ครบ

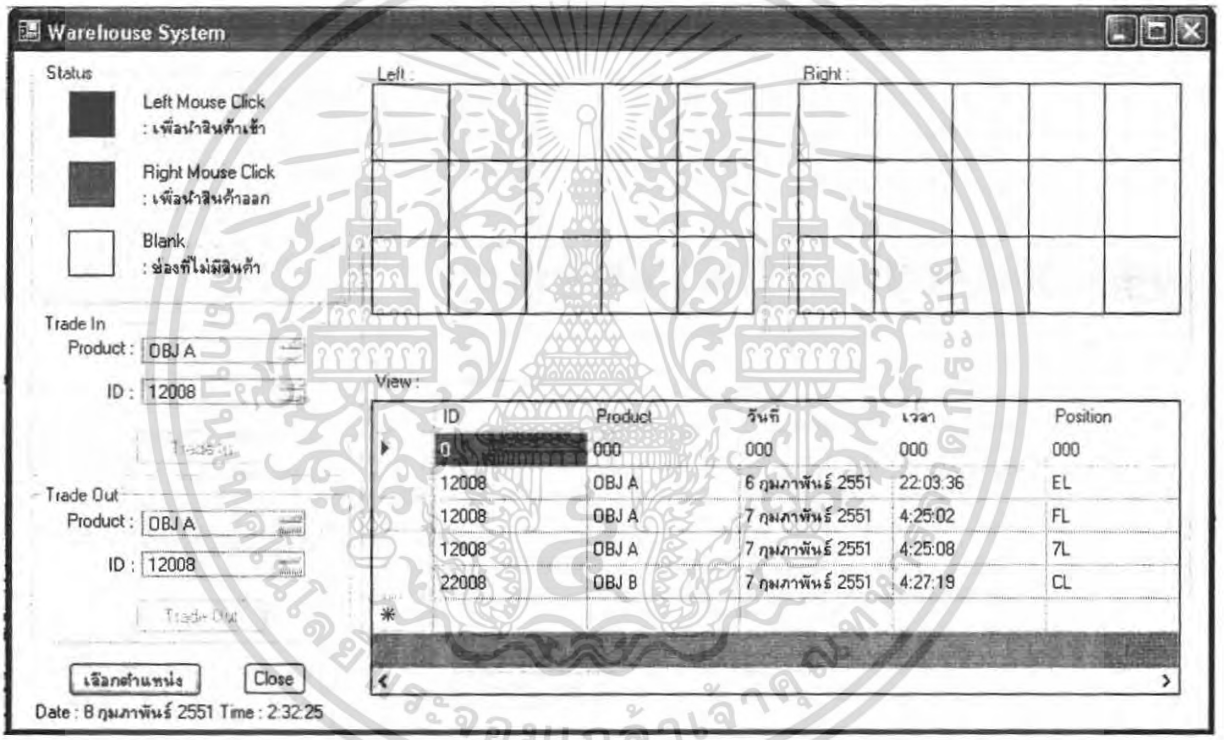
รูปที่ 4.2 แสดงกล่องข้อความรายงานความผิดพลาด

เมื่อชื่อและรหัสผ่านถูกต้องจะมีกล่องข้อความแสดงก่อนเข้าสู่ระบบใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 หน้าจอแสดงข้อความเมื่อชื่อและรหัสผ่านถูกต้อง

เข้าสู่ระบบการใช้งานโดยกดปุ่ม “เลือกตำแหน่ง” จะมีตารางขึ้นมาเพื่อให้เลือกตำแหน่งที่ต้องการเก็บชิ้นงาน



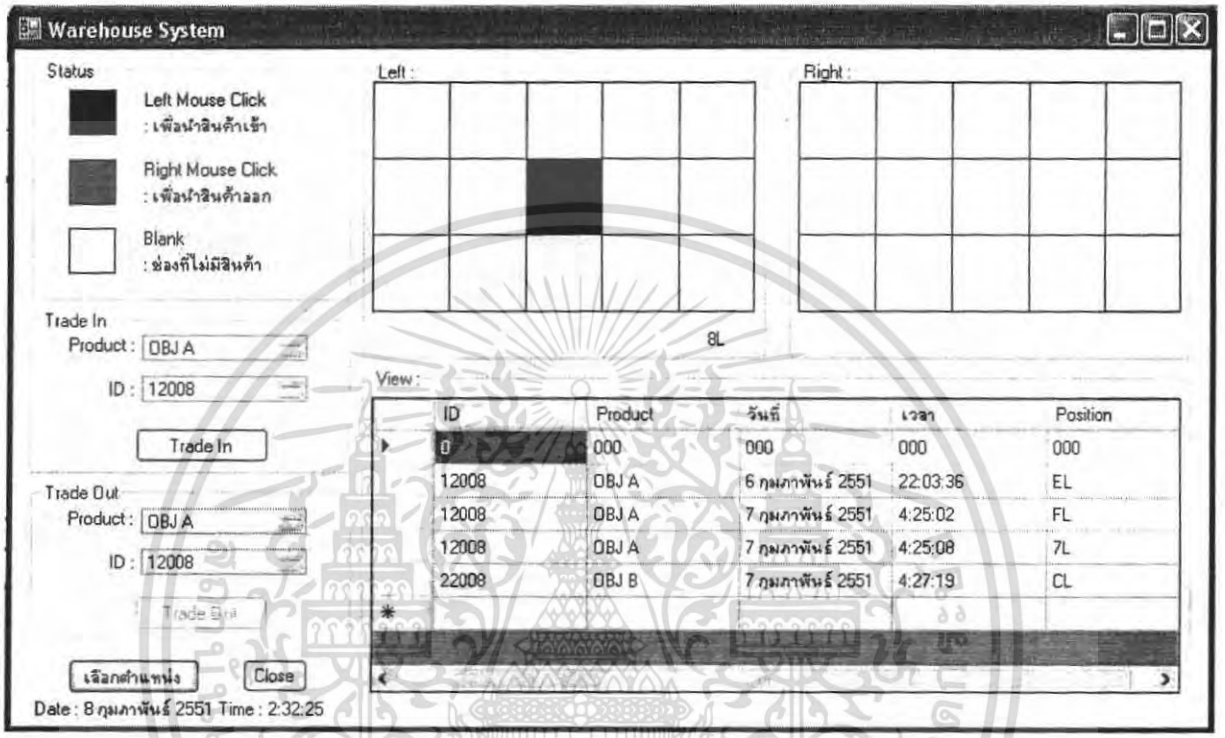
รูปที่ 4.4 แสดงหน้าจอระบบใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การขนถ่ายสินค้าขาเข้า

4.2.1 การทำงานในส่วนของโปรแกรมเพื่อนำสินค้าเข้า

คลิกซ้ายเลือกตำแหน่งเพื่อเก็บชิ้นงาน ปุ่ม "Trade In" จะสามารถใช้งาน และเลือก Product และ ID



รูปที่ 4.5 แสดงหน้าจอเลือกตำแหน่งเพื่อเก็บชิ้นงาน

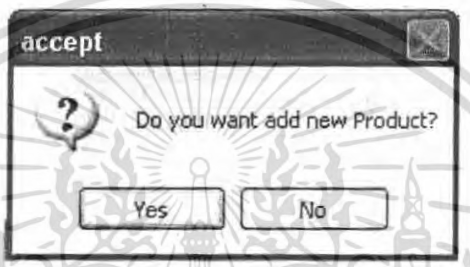
ในกรณีที่ยังไม่ได้เลือก Product และ ID จะมีข้อความขึ้นเตือนให้เลือกข้อมูลให้ครบ

รูปที่ 4.6 แสดงหน้าจอระบบเตือนให้เลือก Product

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 แสดงหน้าจอระบบเตือนให้เลือก ID

เมื่อเลือกข้อมูลครบถ้วนแล้วกดปุ่ม Trade In จะมีข้อความยืนยันขึ้นให้ยืนยันอีกครั้ง

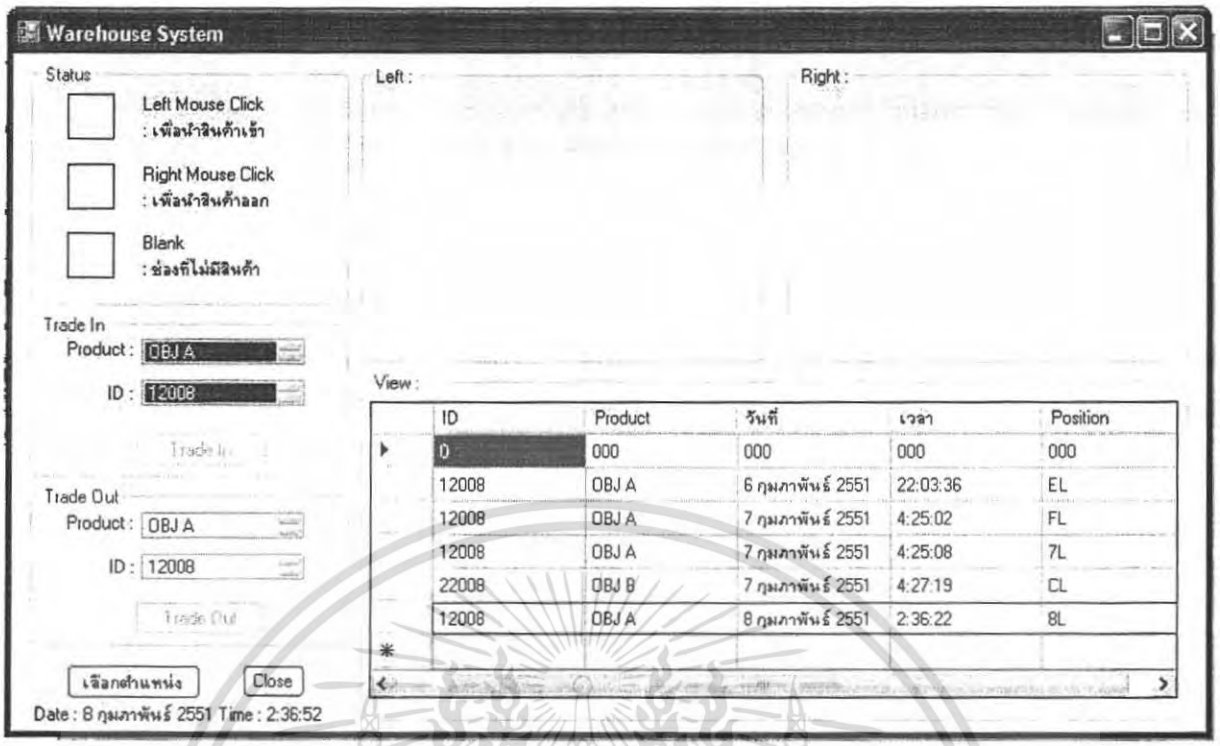


รูปที่ 4.8 แสดงหน้าจอการยืนยันการทำงาน

รูปที่ 4.9 ข้อความแสดงการทำงานสมบูรณ์

ข้อมูลจะถูกนำเก็บในตารางอย่างที่แสดงในกรอบสี่เหลี่ยมสีแดงดังรูป

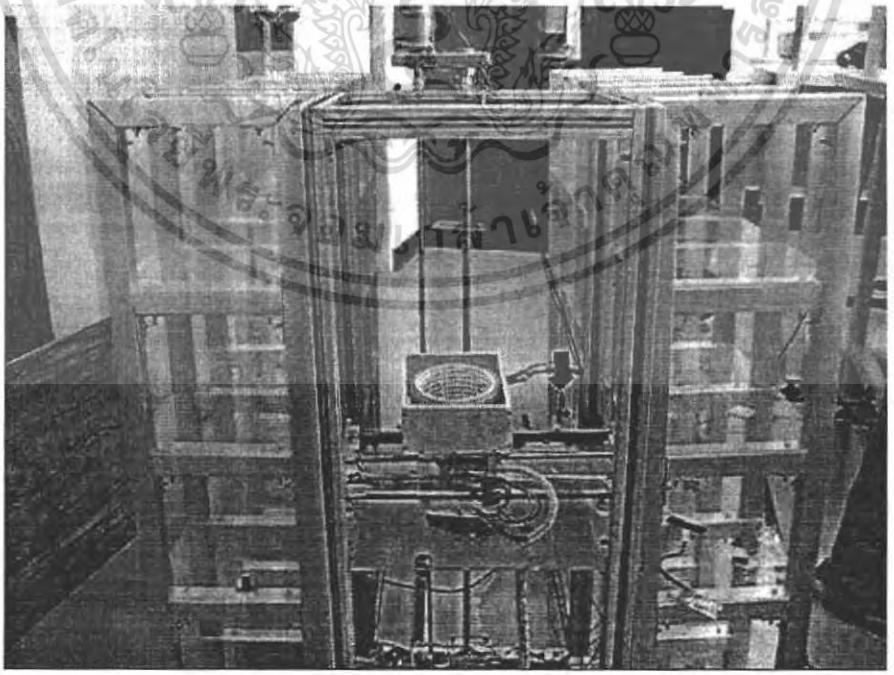
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงหน้าจอข้อมูลที่ถูกเก็บในตาราง

4.2.2 การทำงานในส่วนของเครนเพื่อนำสินค้าเข้า

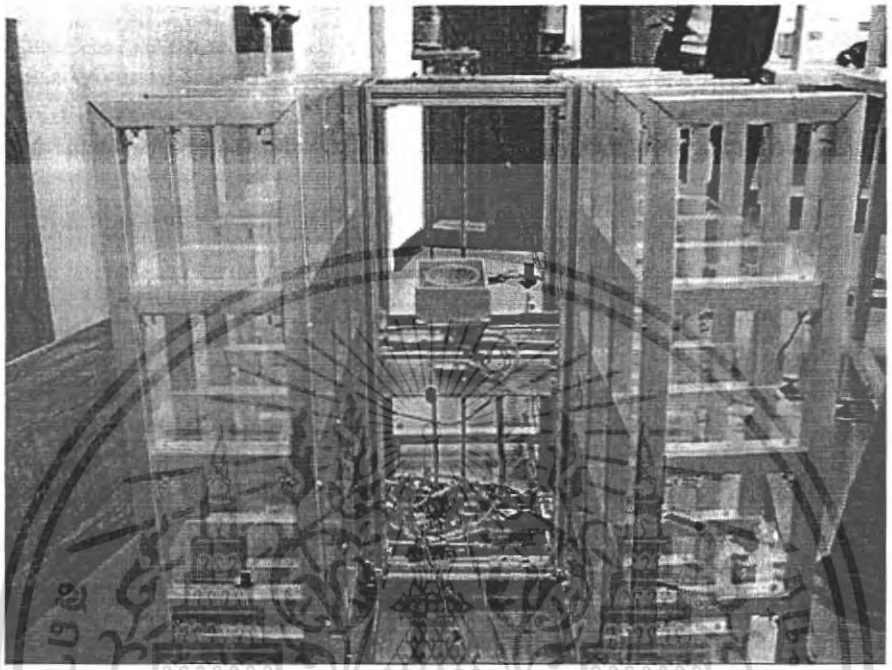
ในตอนแรกตัวเครนจะอยู่ที่ตำแหน่งเริ่มต้น เมื่อนำชิ้นงานมาวางแล้วทำการเลือกว่าจะนำชิ้นงานไปวางไว้ที่ตำแหน่งไหนของคลังสินค้าโดยผ่านตัวโปรแกรมข้างต้นดังรูปที่ 4.11



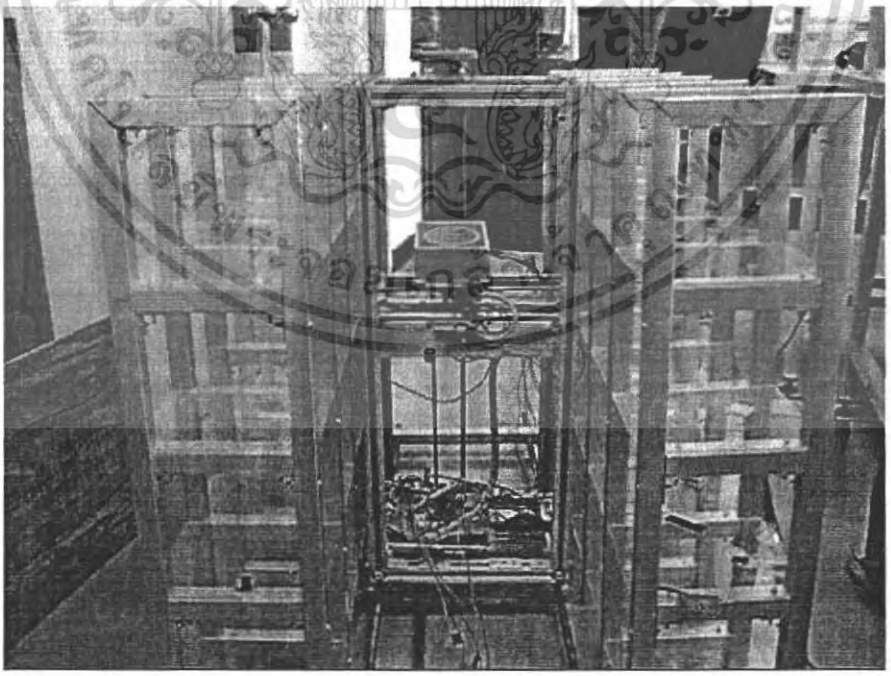
รูปที่ 4.11 แสดงเครนที่ตำแหน่งเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นไมโครคอนโทรลเลอร์จะทำการประมวลผลคำสั่งที่ได้มาว่าจะให้นำชิ้นงานไปไว้ที่ตำแหน่งไหน โดยจะเริ่มเคลื่อนที่ในแนวแกน X ก่อนเพื่อไปยังแถวที่ต้องการ จากนั้นจะเคลื่อนที่ในแนวแกน Y เพื่อไปยังชั้นที่ต้องการ แล้วแกน Z จะทำการเคลื่อนที่นำชิ้นงานไปยังคลังสินค้า ดังรูป

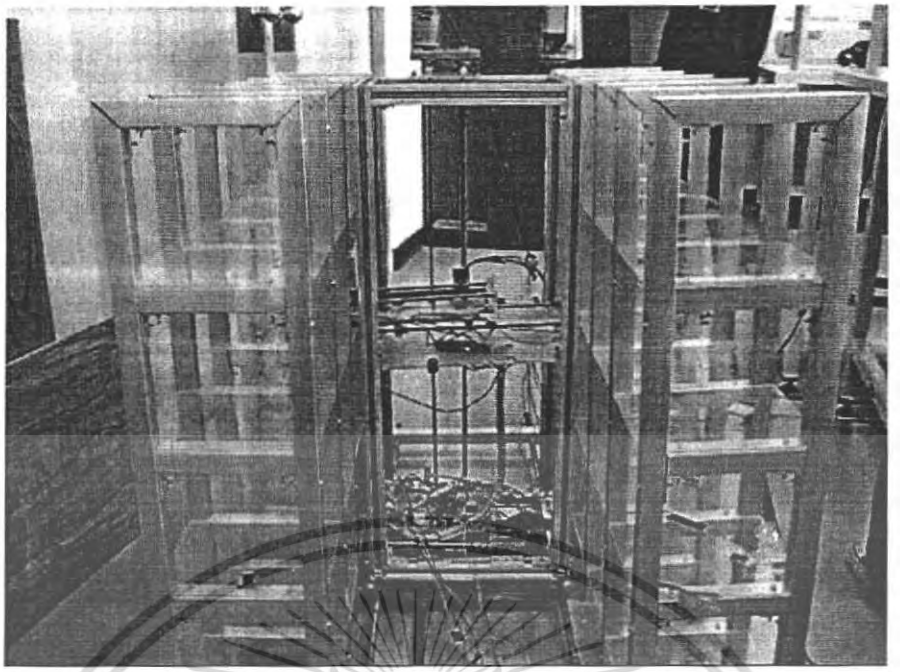


รูปที่ 4.12 แสดงการทำงานในแนวแกน X



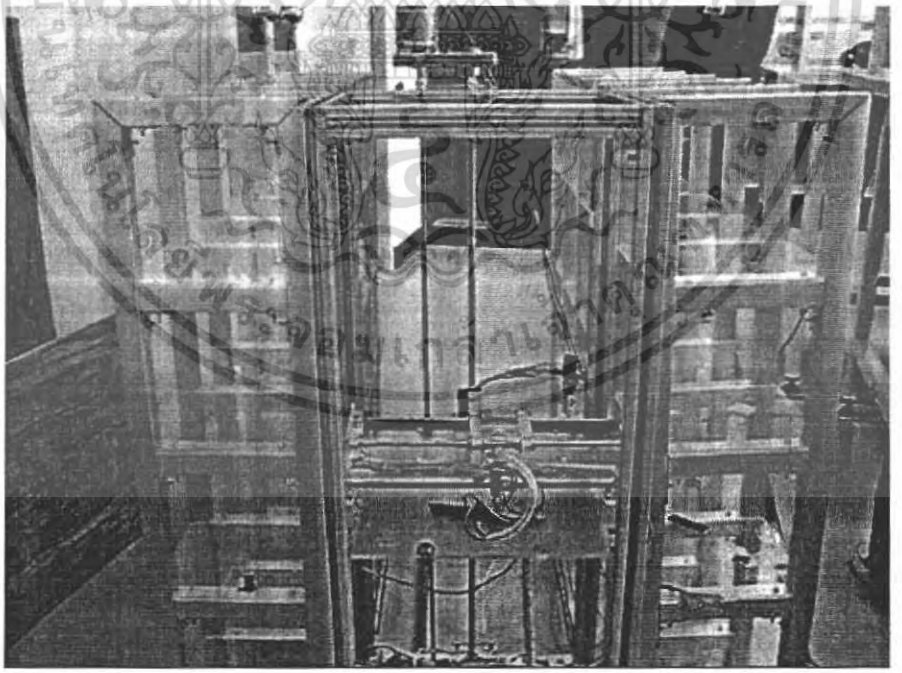
รูปที่ 4.13 แสดงการทำงานในแนวแกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงการทำงานในแนวแกน Z

เมื่อนำชิ้นงานเข้าไปเก็บที่คลังสินค้าเสร็จแล้ว เครนก็จะกลับมาที่ตำแหน่งเริ่มต้นเพื่อรอรับชิ้นงานอื่น



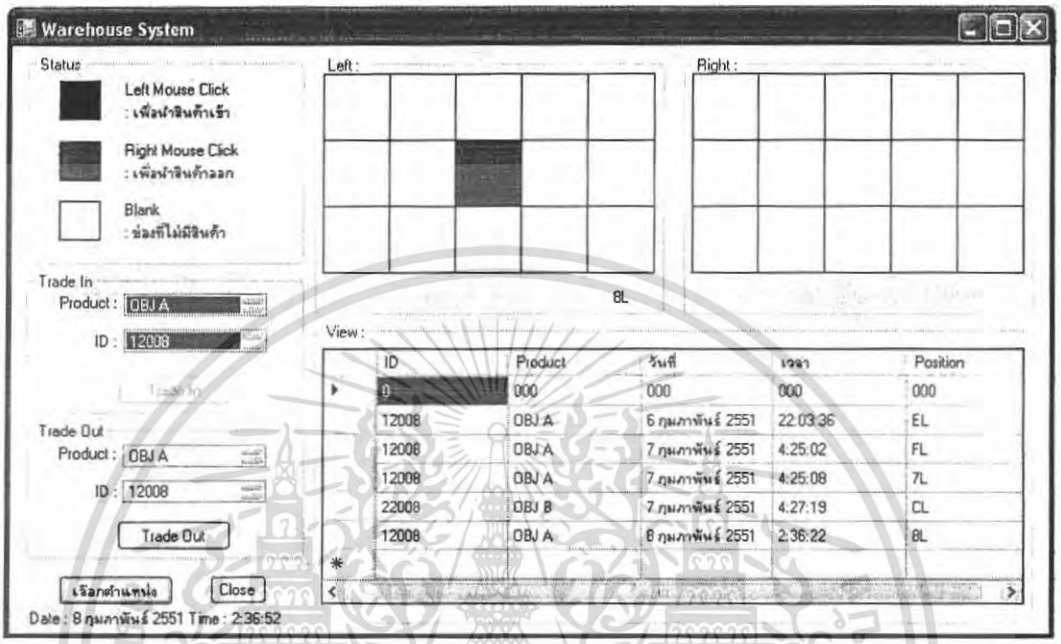
รูปที่ 4.15 เครนเคลื่อนที่กลับมาที่ตำแหน่งเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การขนถ่ายสินค้าขาออก

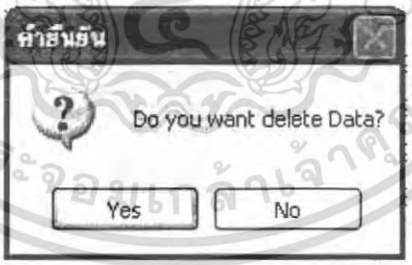
4.3.1 การทำงานในส่วนของโปรแกรมเพื่อนำสินค้าออก

คลิกขวาเลือกตำแหน่งหรือ เลือก Product และ ID เพื่อเก็บชิ้นงาน ปุ่ม "Trade Out" จะสามารถใช้งาน



รูปที่ 4.16 แสดงหน้าจอเลือกตำแหน่งเพื่อนำชิ้นงานออก

เมื่อคลิกขวาเลือกตำแหน่ง หรือ เลือก Product และ ID แล้วกดปุ่ม "Trade Out" จะมีข้อความยืนยันขึ้น ให้ยืนยันอีกครั้ง

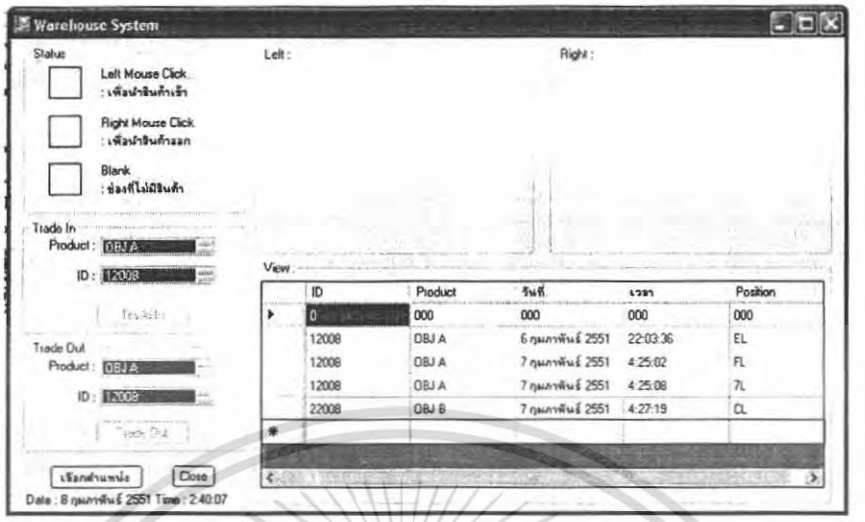


รูปที่ 4.17 แสดงหน้าจอการยืนยันการทำงาน

รูปที่ 4.18 ข้อความแสดงการทำงานสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

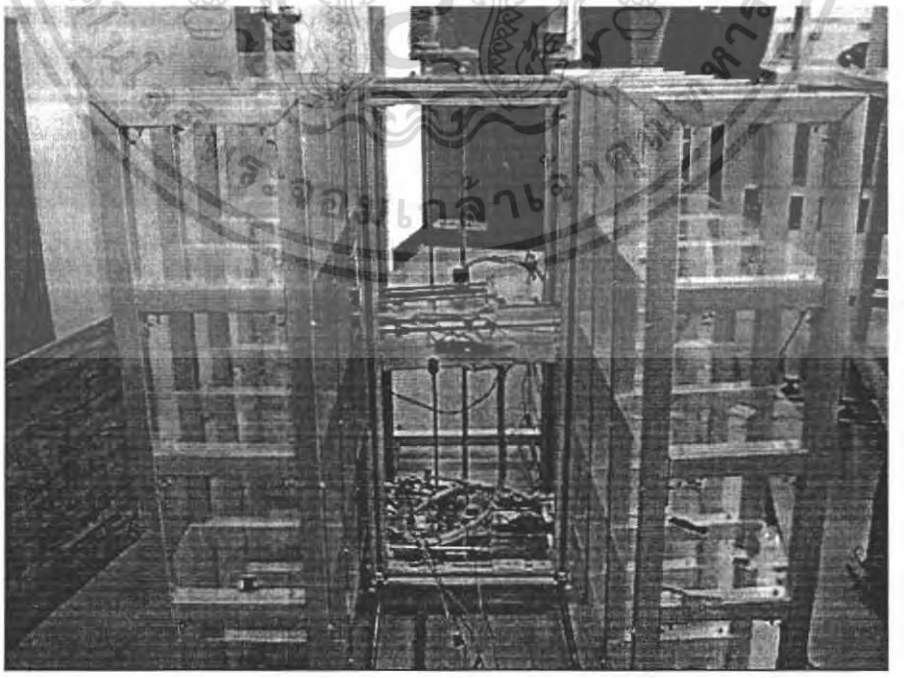
ข้อมูลจะถูกลบออกจากตารางอย่างที่แสดงในกรอบสี่เหลี่ยมสีแดงดังรูป



รูปที่ 4.19 แสดงหน้าจอข้อมูลที่ถูกลบออกจากตาราง

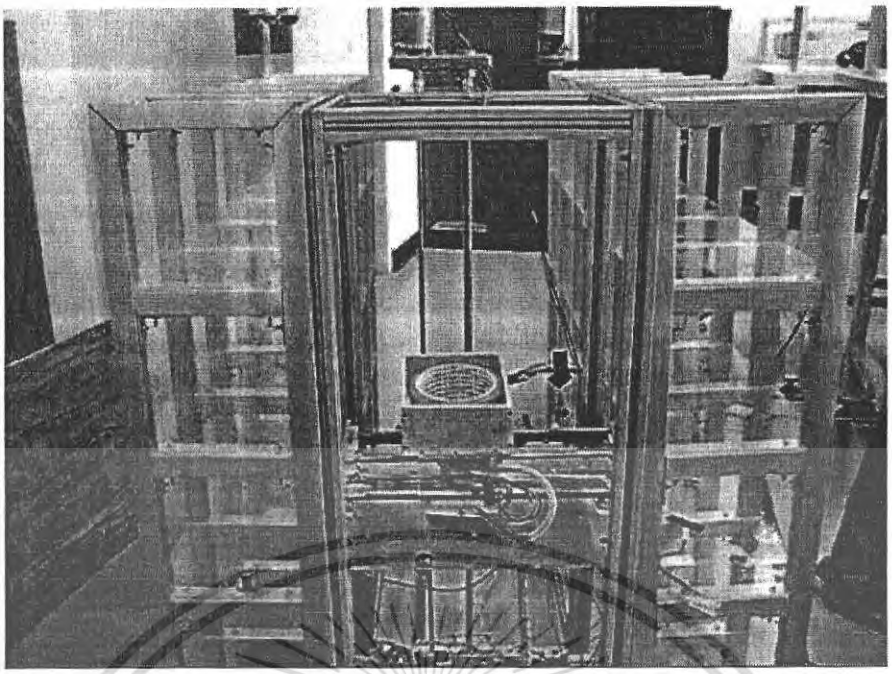
4.3.2 การทำงานในส่วนของเครนเพื่อนำสินค้าออก

ในตอนแรกตัวเครนจะอยู่ที่ตำแหน่งเริ่มต้น เมื่อรับคำสั่งจากผู้ใช้งานว่าต้องการนำชิ้นงานใดออกจากคลังสินค้าโดยผ่านตัวโปรแกรมข้างต้น ไมโครคอนโทรลเลอร์จะทำการประมวลผลคำสั่งที่ได้สั่งการให้เครนเคลื่อนที่ไปยังตำแหน่งนั้น โดยจะเคลื่อนที่ในแนวแกน X แกน Y และแกน Z ตามลำดับ แล้วนำสินค้ามาไว้ที่ตำแหน่งเริ่มต้น



รูปที่ 4.20 แสดงการทำงานของเครนขณะที่เข้าไปเอาชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 เครนนำชิ้นงานมาวางไว้ที่จุดเริ่มต้น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ณัฏฐพล วงศ์สุนทรชัย, ชัยวัฒน์ ลิ้มพรจิตรวิไล. **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ PIC16F877.** กรุงเทพมหานคร : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

[2] สัจจะ จรัสรุ่งรวีวร. **คู่มือ Visual C# 2005 ฉบับสมบูรณ์.** นนทบุรี : บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด. 2550.

[3] ประจิน พลังสันติกุล. **เรียนรู้และใช้งาน CCS C คอมไพล์เลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC.** พิมพ์ครั้งที่1. กรุงเทพนคร : อินโนเวตีฟ เอ็กเพอริเมนต์. 2547.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมควบคุม

```
#include<16F877A.h>
#define TxD Pin_C6
#define RxD Pin_C7
#fuses HS,NOLVP,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=TxD,rcv=RxD)
#byte porta=0x05
#byte portb=0x06
#byte portc=0x07
#byte portd=0x08
#byte porte=0x09
#bit Direct8=portb.7
#bit Direct7=portb.6
#bit Direct6=portb.5
#bit Direct5=portb.4
#bit Direct4=portb.3
#bit Direct3=portb.2
#bit Direct2=portb.1
#bit Direct1=portb.0
#bit sensor1=portc.0
#bit sensor2=portc.3
#bit sensor3=portc.4
#bit sensor4=portc.5

#bit pinc1=portc.1
#bit pinc2=portc.2

#bit Sw1=portd.0
#bit Sw2=portd.1
#bit Sw3=portd.2
#bit Sw4=portd.3
#bit Sw5=portd.4

int i,flag1,flag2,col,row,loop;
int in,back,up,lateral,flag_col,flag_row;
int flag_Sw1,flag_Sw2,home,sub_col,sub_row;
char test1,test2;
char stock[3];
char Trad[2] = {'I','O'};
char Block[15] = {'1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
char side[2] = {'L','R'};


void Getcode()
{
 printf("Please Choose Position\n\r");
    for(i=0;i<3;i++)
    {
      stock[i] = getc();
    }
       printf("Recive Complete\n\r");
       loop = 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Check()
{
 for(i=0;i<2;i++)
     {
       if(Trad[i]==stock[0])
       {
         in = i;
          printf("Trad = %d\n\r",in);
       }
     }
     for(i=0;i<15;i++)
     {
       if(Block[i]==stock[1])
       {
         up = i;
          printf("Block = %d\n\r",up);
       }
     }
      for(i=0;i<2;i++)
     {
       if(side[i]==stock[2])
       {
         lateral = i;
          printf("Side = %d\n\r",lateral);
       }
     }
}

void Motor_on()
{
Direct1=0;
Direct2=1;
Direct3=0;
Direct4=1;
}

void Crane_in()
{
  switch(up)
  {
  case 0: if(col<1)
          {
            if(sensor1==1&&flag1==0)
            {
              flag1 = 1;
              col = col+1;
              sub_col = col;
              printf("Col = %d\n\r",col);
            }
            if(sensor1==0)
              flag1 = 0;
          }
          if(col>=1)
          {
            Direct1 = 1;
            Direct2 = 1;
            flag1 = 1;
            flag_col = 1;
          }

          if(row<3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
         if(sensor2==1&&flag2==0)
         {
           flag2 = 1;
           row = row+1;
           sub_row = row;
           printf("Row = %d\n\r",row);
         }
         if(sensor2==0)
           flag2 = 0;
        }
        if(row>=3)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
    break;

    case 1:if(col<2)
          {
            if(sensor1==1&&flag1==0)
            {
              flag1 = 1;
              col = col+1;
              sub_col = col;
              printf("Col = %d\n\r",col);
            }
            if(sensor1==0)
              flag1 = 0;
          }
          if(col>=2)
          {
            Direct1 = 1;
            Direct2 = 1;
            flag1 = 1;
            flag_col = 1;
          }

        if(row<3)
        {
         if(sensor2==1&&flag2==0)
         {
           flag2 = 1;
           row = row+1;
           sub_row = row;
           printf("Row = %d\n\r",row);
         }
         if(sensor2==0)
           flag2 = 0;
        }
        if(row>=3)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
    break;
    case 2:if(col<3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=3)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

       if(row<3)
       {
        if(sensor2==1&&flag2==0)
        {
          flag2 = 1;
          row = row+1;
          sub_row = row;
          printf("Row = %d\n\r",row);
        }
        if(sensor2==0)
          flag2 = 0;
       }
       if(row>=3)
       {
         Direct3 = 1;
         Direct4 = 1;
         flag2 = 1;
         flag_row = 1;
       }

break;
case 3:if(col<4)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=4)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

        if(row<3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
         if(sensor2==1&&flag2==0)
         {
           flag2 = 1;
           row = row+1;
           sub_row = row;
           printf("Row = %d\n\r",row);
         }
         if(sensor2==0)
           flag2 = 0;
        }
        if(row>=3)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }

break;
case 4:if(col<5)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=5)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

        if(row<3)
        {
         if(sensor2==1&&flag2==0)
         {
           flag2 = 1;
           row = row+1;
           sub_row = row;
           printf("Row = %d\n\r",row);
         }
         if(sensor2==0)
           flag2 = 0;
        }
        if(row>=3)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
break;
case 5:if(col<1)
```

เอกสารนี้จัดเตรียมไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=1)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

       if(row<2)
       {
        if(sensor2==1&&flag2==0)
        {
          flag2 = 1;
          row = row+1;
          sub_row = row;
          printf("Row = %d\n\r",row);
        }
        if(sensor2==0)
          flag2 = 0;
       }
       if(row>=2)
       {
         Direct3 = 1;
         Direct4 = 1;
         flag2 = 1;
         flag_row = 1;
       }
break;
case 6: if(col<2)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=2)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

       if(row<2)
       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(sensor2==1&&flag2==0)
        {
          flag2 = 1;
          row = row+1;
          sub_row = row;
          printf("Row = %d\n\r",row);
        }
        if(sensor2==0)
          flag2 = 0;
      }
      if(row>=2)
      {
        Direct3 = 1;
        Direct4 = 1;
        flag2 = 1;
        flag_row = 1;
      }
break;
case 7:if(col<3)
      {
        if(sensor1==1&&flag1==0)
        {
          flag1 = 1;
          col = col+1;
          sub_col = col;
          printf("Col = %d\n\r",col);
        }
        if(sensor1==0)
          flag1 = 0;
      }
      if(col>=3)
      {
        Direct1 = 1;
        Direct2 = 1;
        flag1 = 1;
        flag_col = 1;
      }

      if(row<2)
      {
        if(sensor2==1&&flag2==0)
        {
          flag2 = 1;
          row = row+1;
          sub_row = row;
          printf("Row = %d\n\r",row);
        }
        if(sensor2==0)
          flag2 = 0;
      }
      if(row>=2)
      {
        Direct3 = 1;
        Direct4 = 1;
        flag2 = 1;
        flag_row = 1;
      }
break;
case 8: if(col<4)
      {
        if(sensor1==1&&flag1==0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
              flag1 = 1;
              col = col+1;
              sub_col = col;
              printf("Col = %d\n\r",col);
            }
            if(sensor1==0)
              flag1 = 0;
          }
          if(col>=4)
          {
            Direct1 = 1;
            Direct2 = 1;
            flag1 = 1;
            flag_col = 1;
          }

          if(row<2)
          {
           if(sensor2==1&&flag2==0)
           {
             flag2 = 1;
             row = row+1;
             sub_row = row;
             printf("Row = %d\n\r",row);
           }
           if(sensor2==0)
             flag2 = 0;
          }
          if(row>=2)
          {
            Direct3 = 1;
            Direct4 = 1;
            flag2 = 1;
            flag_row = 1;
          }
break;
case 9: if(col<5)
          {
            if(sensor1==1&&flag1==0)
            {
              flag1 = 1;
              col = col+1;
              sub_col = col;
              printf("Col = %d\n\r",col);
            }
            if(sensor1==0)
              flag1 = 0;
          }
          if(col>=5)
          {
            Direct1 = 1;
            Direct2 = 1;
            flag1 = 1;
            flag_col = 1;
          }

          if(row<2)
          {
           if(sensor2==1&&flag2==0)
           {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            flag2 = 1;
            row = row+1;
            sub_row = row;
            printf("Row = %d\n\r",row);
          }
          if(sensor2==0)
            flag2 = 0;
        }
        if(row>=2)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
break;
case 10:if(col<1)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=1)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

        if(row<1)
        {
          if(sensor2==1&&flag2==0)
          {
            flag2 = 1;
            row = row+1;
            sub_row = row;
            printf("Row = %d\n\r",row);
          }
          if(sensor2==0)
            flag2 = 0;
        }
        if(row>=1)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
break;
case 11:if(col<2)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=2)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

        if(row<1)
        {
         if(sensor2==1&&flag2==0)
         {
           flag2 = 1;
           row = row+1;
           sub_row = row;
           printf("Row = %d\n\r",row);
         }
         if(sensor2==0)
           flag2 = 0;
        }
        if(row>=1)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
break;
case 12:if(col<3)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=3)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

        if(row<1)
        {
         if(sensor2==1&&flag2==0)
         {
           flag2 = 1;
           row = row+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            sub_row = row;
            printf("Row = %d\n\r",row);
          }
          if(sensor2==0)
            flag2 = 0;
        }
        if(row>=1)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
break;
case 13:if(col<4)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=4)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

        if(row<1)
        {
          if(sensor2==1&&flag2==0)
          {
            flag2 = 1;
            row = row+1;
            sub_row = row;
            printf("Row = %d\n\r",row);
          }
          if(sensor2==0)
            flag2 = 0;
        }
        if(row>=1)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
break;
case 14:if(col<5)
        {
          if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col+1;
            sub_col = col;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
        }
        if(col>=5)
        {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          flag_col = 1;
        }

        if(row<1)
        {
         if(sensor2==1&&flag2==0)
         {
           flag2 = 1;
           row = row+1;
           sub_row = row;
           printf("Row = %d\n\r",row);
         }
         if(sensor2==0)
           flag2 = 0;
        }
        if(row>=1)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          flag_row = 1;
        }
  break;
  }
}

void Go_side1()
{
  if(flag_col==1&&flag_row==1)
  {
    switch(lateral)
    {
    case 0:  Direct5 = 1;
             Direct6 = 0;
             Direct7 = 1;
             Direct8 = 0;
             flag_Sw1 = 1;
             printf("Axis Z Go Left\n\r");

    break;

    case 1:  Direct5 = 0;
             Direct6 = 1;
             Direct7 = 0;
             Direct8 = 1;
             flag_Sw1 = 1;
             printf("Axis Z Go Right\n\r");
    break;
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Stop_Z1()
{
  if((Sw1==0||Sw2==0)&&flag_Sw1==1)
  {
  flag_col = 0;
  flag_row = 0;
  up = 15;
  Direct5 = 1;
  Direct6 = 1;
  flag_Sw2 = 1;
  printf("Stop Floor1\n\r");
  }
  if((Sw3==0||Sw4==0)&&flag_Sw2==1)
  {
  Direct7 = 1;
  Direct8 = 1;
  printf("Stop Floor2\n\r");
  back = 1;
  flag_Sw1 = 0;
  }
}

void Backward_in()
{
  if(back==1)
  {
    Direct3 = 1;
    Direct4 = 0;
    flag_Sw1 = 1;
    flag_Sw2 = 0;
    printf("Little move Y\n\r");
  }
    if(Sw5==0&&flag_Sw1==1)
    {
      Direct3 = 1;
      Direct4 = 1;
      back = 0;
      printf("Stop Y\n\r");
      switch(lateral)
      {
        case 0:    Direct5 = 0;
                   Direct6 = 1;
                   Direct7 = 0;
                   Direct8 = 1;
                   flag_Sw2 = 1;
          printf("Axis Z Go Back Left\n\r");
        break;

        case 1:  Direct5 = 1;
                 Direct6 = 0;
                 Direct7 = 1;
                 Direct8 = 0;
                 flag_Sw2 = 1;
          printf("Axis Z Go back Right\n\r");
        break;
      }
    }
   if(Sensor4==1&&flag_Sw2==1)
   {
     Direct5 = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Direct6 = 1;
      flag_Sw1 =0;
      printf("Axis Z1 initial\n\r");
    }
    if(Sensor3==1&&flag_Sw2==1)
    {
      Direct7 = 1;
      Direct8 = 1;
      flag_Sw2 = 0;
      printf("Axis Z2 inital\n\r");
      back = 0;
      home = 1;
    }


}

void Go_Home()
{
  if(home==1)
  {
   Direct1 = 1;
   Direct2 = 0;
   Direct3 = 1;
   Direct4 = 0;
   flag1 = 0;
   flag2 = 0;
   col = sub_col;
   row = sub_row;
   home = 2;
   printf("Col = %d\n\r",col);
   printf("Row = %d\n\r",row);
  }

  if(home==2)
  {
    if(sensor1==1&&flag1==0)
          {
            flag1 = 1;
            col = col-1;
           // sub_col = col;
            printf("Col = %d\n\r",col);
          }
          if(sensor1==0)
            flag1 = 0;
    if(col==0)
    {
          Direct1 = 1;
          Direct2 = 1;
          flag1 = 1;
          //flag_col = 1;
    }
    if(sensor2==1&&flag2==0)
        {
          flag2 = 1;
          row = row-1;
          //sub_row = row;
          printf("Row = %d\n\r",row);
        }
        if(sensor2==0)
          flag2 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(row==0)
        {
          Direct3 = 1;
          Direct4 = 1;
          flag2 = 1;
          loop = 0;
        }
      }
}

void Little_Down_Y()
{
  for(i=0;i<10;i++)
  {
    Direct3 = 1 ;
    Direct4 = 0;
    delay_ms(1000);
  }
  if(i==9)
  {
    Direct3 = 1;
    Direct4 = 1;
  }
}
void Backward_out()
{
  if(back==1)
  {
    Direct3 = 0;
    Direct4 = 1;
    flag_Sw1 = 1;
    flag_Sw2 = 0;
    printf("Little move Y\n\r");
  }
    if(Sw5==1&&flag_Sw1==1)
    {
      Direct3 = 1;
      Direct4 = 1;
      back = 0;
      printf("Stop Y\n\r");
      switch(lateral)
      {
        case 0:    Direct5 = 0;
                   Direct6 = 1;
                   Direct7 = 0;
                   Direct8 = 1;
                   flag_Sw2 = 1;
          printf("Axis Z Go Back Left\n\r");
        break;

        case 1:  Direct5 = 1;
                 Direct6 = 0;
                 Direct7 = 1;
                 Direct8 = 0;
                 flag_Sw2 = 1;
          printf("Axis Z Go back Right\n\r");
        break;
      }
    }
  if(Sensor3==1&&flag_Sw2==1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      Direct5 = 1;
      Direct6 = 1;
      flag_Sw1 =0;
      printf("Axis Z1 initial\n\r");
    }
    if(Sensor4==1&&flag_Sw2==1)
    {
      Direct7 = 1;
      Direct8 = 1;
      flag_Sw2 = 0;
      printf("Axis Z2 inital\n\r");
      back = 0;
      home = 1;
    }

}

void main()
{
set_tris_b(0x00);
set_tris_c(0xB9);
set_tris_d(0xFF);
portb = 0;
//col = 0;
//row = 0;
loop = 0;
//flag1 = 0;
//flag2 = 0;
//flag_col = 0;
//flag_row = 0;
lateral = 2;
//flag_Sw1 = 0;
//flag_Sw2 = 0;
pinc1=1;
pinc2=1;

  printf("Automatic Warehouse System\r\n");
  while(1)
  {
  flag1 = 0;
  flag2 = 0;
  flag_col = 0;
  flag_row = 0;
  flag_Sw1 = 0;
  flag_Sw2 = 0;
  col = 0;
  row = 0;
  back = 0;
  home = 0;
  sub_col = 0;
  sub_row = 0;
  Getcode();
  Check();
  Motor_on();
   while(loop==1)
   {
    if(in==0)
    {
     Crane_in();
     Go_side1();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Stop_Z1();
      Backward_in();
      Go_Home();
     }
     if(in==1)
     {
       Crane_in();
       Little_Down_Y();
       Go_side1();
       Stop_Z1();
       Backward_out();
       Go_Home();

     }
    }
   }//end while(1)
}//end main
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

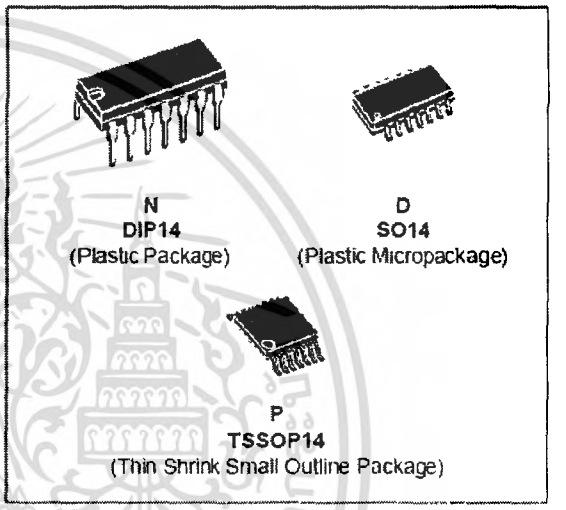
ภาคผนวก ข

เอกสารเครื่องมืออุปกรณ์อิเล็กทรอนิกส์

LM124
LM224 - LM324

LOW POWER QUAD OPERATIONAL AMPLIFIERS

- WIDE GAIN BANDWIDTH : 1.3MHz
- INPUT COMMON-MODE VOLTAGE RANGE INCLUDES GROUND
- LARGE VOLTAGE GAIN : 100dB
- VERY LOW SUPPLY CURRENT/AMPLI : 375µA
- LOW INPUT BIAS CURRENT : 20nA
- LOW INPUT OFFSET VOLTAGE : 5mV max.
  (for more accurate applications, use the equivalent parts LM124A-LM224A-LM324A which feature 3mV max)
- LOW INPUT OFFSET CURRENT : 2nA
- WIDE POWER SUPPLY RANGE :
  SINGLE SUPPLY : +3V TO +30V
  DUAL SUPPLIES : ±1.5V TO ±15V



N
DIP14
(Plastic Package)

D
SO14
(Plastic Micropackage)

P
TSSOP14
(Thin Shrink Small Outline Package)

DESCRIPTION

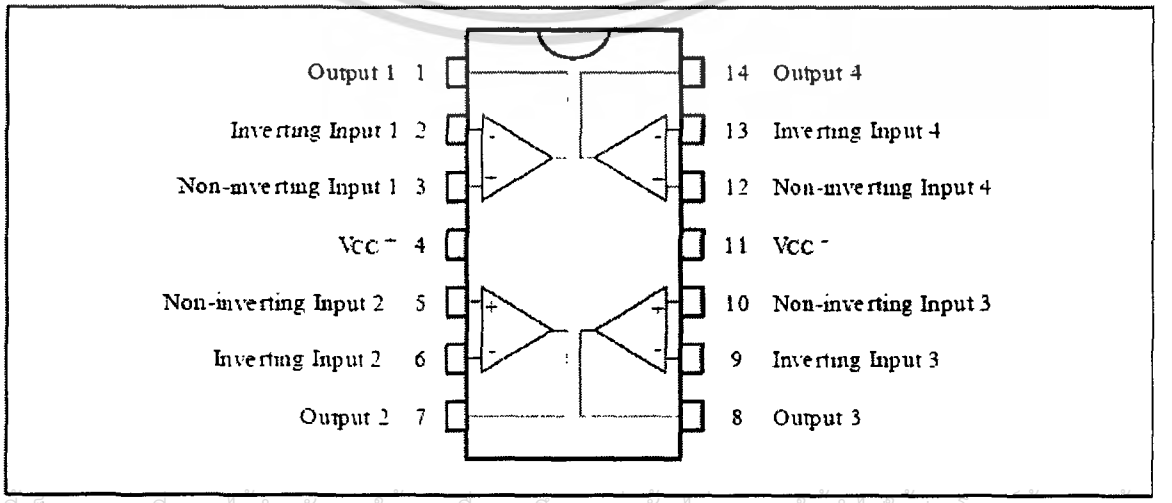
These circuits consist of four independent, high gain, internally frequency compensated operational amplifiers. They operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

ORDER CODES

Part Number	Temperature Range	Package N	D	P
LM124	-55°C, +125°C	•	•	•
LM224	-40°C, +105°C	•	•	•
LM324	0°C, +70°C	•	•	•

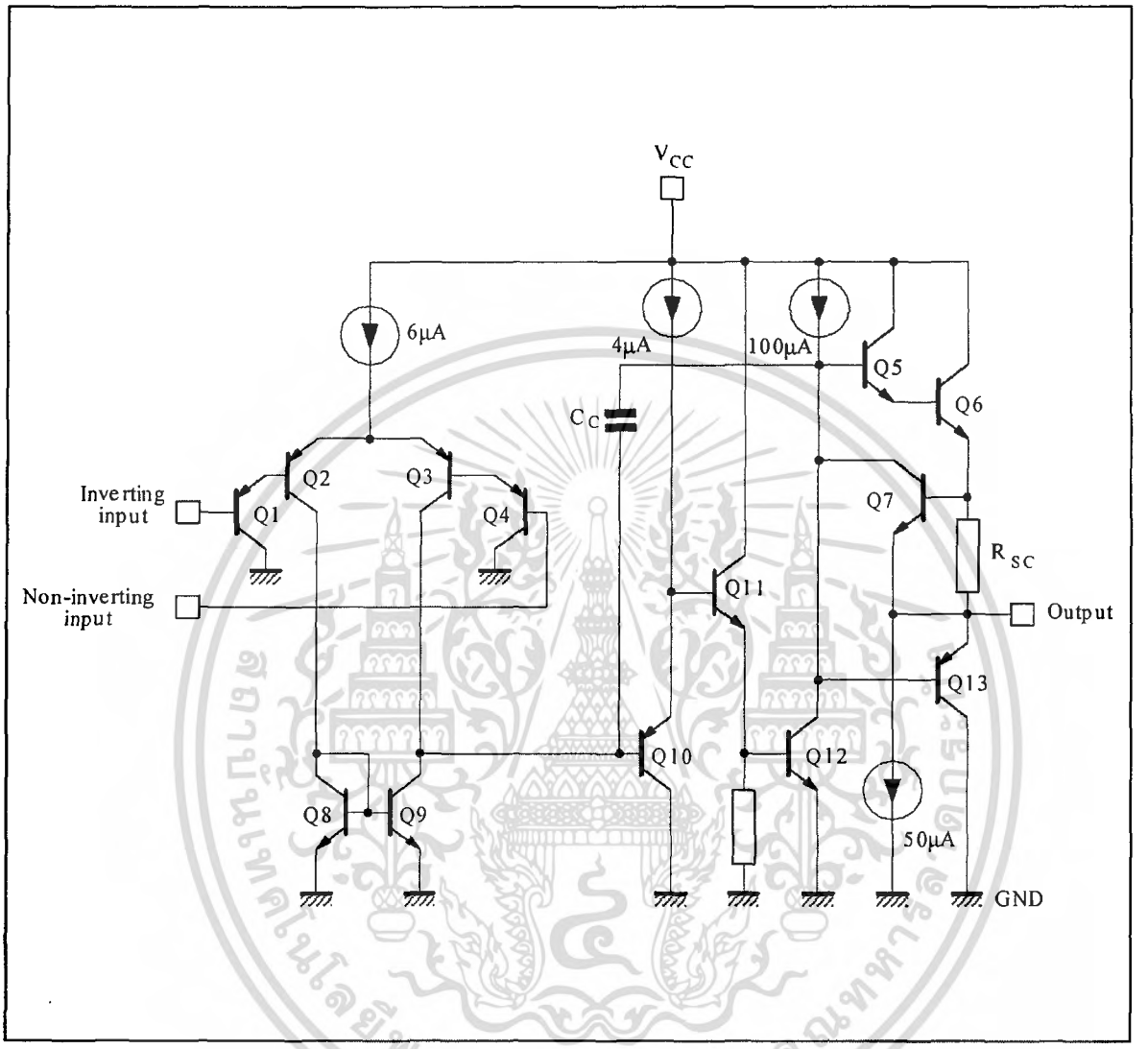
Example : LM224N

PIN CONNECTIONS (top view)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCHEMATIC DIAGRAM (1/4 LM124)



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	LM124	LM224	LM324	Unit
V_{cc}	Supply Voltage	±16 or 32			V
V_i	Input Voltage	-0.3 to +32			V
V_{id}	Differential Input Voltage - (*)	+32	+32	+32	V
P_{tot}	Power Dissipation N Suffix D Suffix	500 -	500 400	500 400	mW mW
-	Output Short-circuit Duration - (note 1)	Infinite			
I_{in}	Input Current – (note 6)	50	50	50	mA
T_{oper}	Operating Free Air Temperature Range	-55 to +125	-40 to +105	0 to +70	°C
T_{stg}	Storage Temperature Range	-65 to +150	-65 to +150	-65 to +150	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS

V_{CC}^+ = +5V, V_{CC}^- = Ground, V_O = 1.4V, T_{amb} = +25°C (unless otherwise specified)

Symbol	Parameter	LM124 - LM224 - LM324			Unit
		Min.	Typ.	Max.	
V_{io}	Input Offset Voltage (note 3) T_{amb} = +25°C LM324 $T_{min} \le T_{amb} \le T_{max}$ LM324		2	5 7 7 9	mV
I_{io}	Input Offset Current T_{amb} = +25°C $T_{min} \le T_{amb} \le T_{max}$.		2	30 100	nA
I_{ib}	Input Bias Current (note 2) T_{amb} = +25°C $T_{min} \le T_{amb} \le T_{max}$.		20	150 300	nA
A_{vd}	Large Signal Voltage Gain (V_{CC}^+ = +15V, R_L = 2kΩ, V_O = 1.4V to 11.4V) T_{amb} = +25°C $T_{min} \le T_{amb} \le T_{max}$.	 50 25	 100		V/mV
SVR	Supply Voltage Rejection Ratio ($R_S \le 10k\Omega$) (V_{CC}^+ = 5V to 30V) T_{amb} = +25°C $T_{min} \le T_{amb} \le T_{max}$	 65 65	 110		dB
I_{CC}	Supply Current, all Amp, no load T_{amb} = +25°C V_{CC} = +5V V_{CC} = +30V $T_{min} \le T_{amb} \le T_{max}$. V_{CC} = +5V V_{CC} = +30V		 0.7 1.5 0.8 1.5	 1.2 3 1.2 3	mA
V_{icm}	Input Common Mode Voltage Range (V_{CC} = +30V) - (note 4) T_{amb} = +25°C $T_{min} \le T_{amb} \le T_{max}$	 0 0		 V_{CC} -1.5 V_{CC} -2	V
CMR	Common-mode Rejection Ratio ($R_S \le 10k\Omega$) T_{amb} = +25°C $T_{min} \le T_{amb} \le T_{max}$	 70 60	 80		dB
I_{source}	Output Current Source (V_{id} = +1V) V_{CC} = +15V, V_o = +2V	20	40	70	mA
I_{sink}	Output Sink Current (V_{id} = -1V) V_{CC} = +15V, V_o = +2V V_{CC} = +15V, V_o = +0.2V	 10 12	 20 50		 mA µA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	LM124 - LM224 - LM324			Unit
		Min.	Typ.	Max.	
V_{OH}	High Level Output Voltage (V_{CC} = +30V) T_{amb} = +25°C R_L = 2kΩ $T_{min.} \le T_{amb} \le T_{max.}$ T_{amb} = +25°C R_L = 10kΩ $T_{min.} \le T_{amb} \le T_{max.}$ (V_{CC} = +5V, R_L = 2kΩ) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$	 26 26 27 27 3.5 3	 27 28		V
V_{OL}	Low Level Output Voltage (R_L = 10kΩ) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$		5	20 20	mV
SR	Slew Rate V_{CC} = 15V, V_i = 0.5 to 3V, R_L = 2kΩ, C_L = 100pF, unity gain)		0.4		V/µs
GBP	Gain Bandwidth Product V_{CC} = 30V, f = 100kHz, V_{in} = 10mV R_L = 2kΩ, C_L = 100pF		1.3		MHz
THD	Total Harmonic Distortion f = 1kHz, A_V = 20dB, R_L = 2kΩ, V_O = $2V_{pp}$ C_L = 100pF, V_{CC} = 30V		0.015		%
e_n	Equivalent Input Noise Voltage f = 1kHz, R_s = 100Ω, V_{CC} = 30V		40		$\frac{nV}{\sqrt{Hz}}$
DV_{io}	Input Offset Voltage Drift		7	30	µV/°C
DI_{IO}	Input Offset Current Drift		10	200	pA/°C
V_O1/V_O2	Channel Separation (note 5) 1kHz ≤ f ≤ 20kHz		120		dB

Notes :

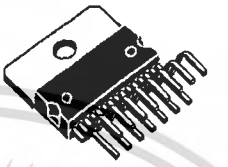
1. Short-circuits from the output to V_{CC} can cause excessive heating if V_{CC} > 15V. The maximum output current is approximately 40mA independent of the magnitude of V_{CC}. Destructive dissipation can result from simultaneous short-circuit on all amplifiers
2. The direction of the input current is out of the IC. This current is essentially constant, independent of the state of the output so no loading change exists on the input lines.
3. V_o = 1.4V, R_s = 0Ω, 5V < V_{CC}^+ < 30V, 0 < V_{ic} < V_{CC}^+ - 1.5V
4. The input common-mode voltage of either input signal voltage should not be allowed to go negative by more than 0.3V. The upper end of the common-mode voltage range is V_{CC}^+ - 1.5V, but either or both inputs can go to +32V without damage.
5. Due to the proximity of external components insure that coupling is not originating via stray capacitance between these external parts. This typically can be detected as this type of capacitance increases at higher frequences
6. This input current only exists when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistor becoming forward biased and thereby acting as input diodes clamps. In addition to this diode action, there is also NPN parasitic action on the IC chip. this transistor action can cause the output voltages of the Op-amps to go to the V_{CC} voltage level (or to ground for a large overdrive) for the time duration than an input is driven negative.
   This is not destructive and normal output will set up again for input voltage higher than -0.3V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

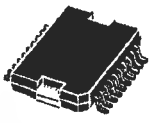
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



Multiwatt15



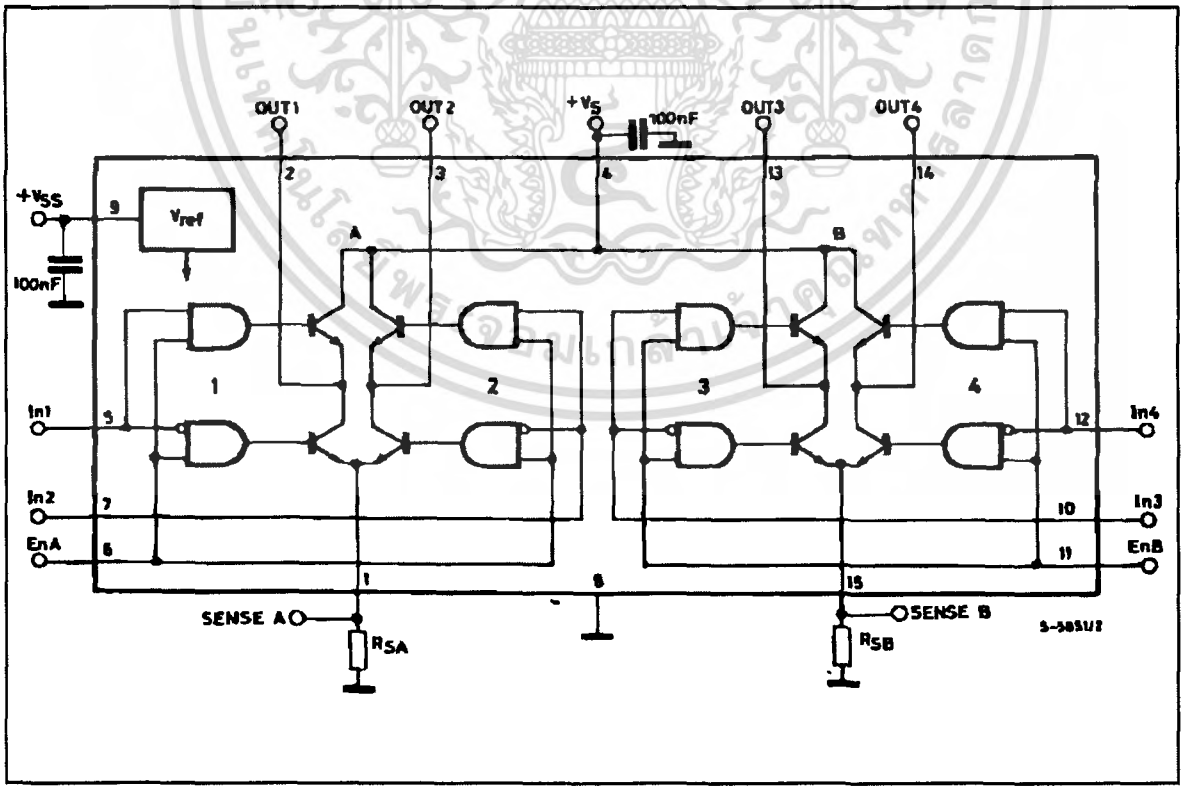
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

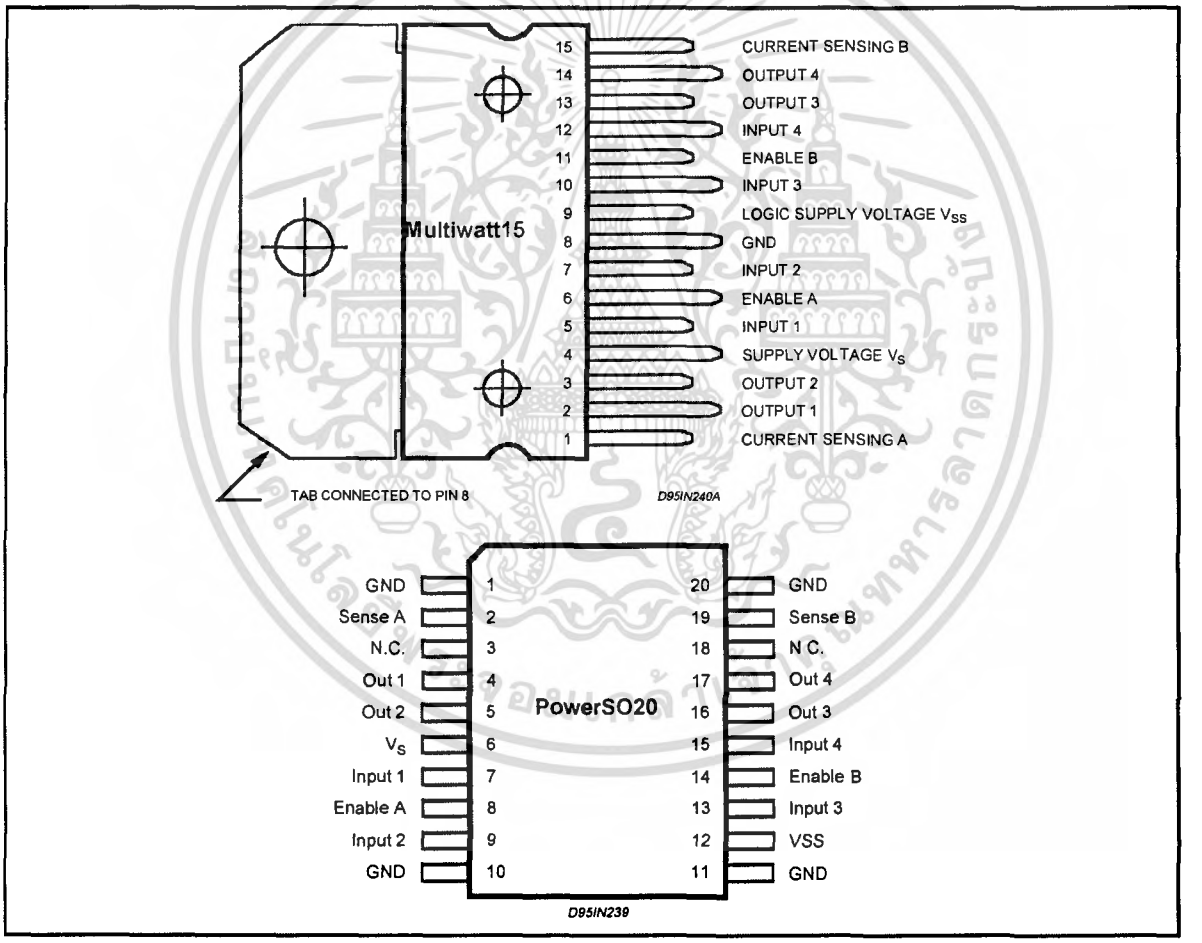


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13,15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_i = L		13	22	mA
		V_i = H		50	70	mA
		V_{en} = L V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_i = L		24	36	mA
		V_i = H		7	12	mA
		V_{en} = L V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L			–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V		30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	µA
$V_{CEsat(H)}$	Source Saturation Voltage	I_L = 1A	0.95	1.35	1.7	V
		I_L = 2A		2	2.7	V
$V_{CEsat(L)}$	Sink Saturation Voltage	I_L = 1A (5)	0.85	1.2	1.6	V
		I_L = 2A (5)		1.7	2.3	V
V_{CEsat}	Total Drop	I_L = 1A (5)	1.80		3.2	V
		I_L = 2A (5)			4.9	V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		µs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		µs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		µs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		µs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		µs
T_6 (V_i)	Sink Current Fall Time	0 9 I_L to 0.1 I_L (3); (4)		0.25		µs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3), (4)		1.6		µs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0 2		µs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		µs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		µs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		µs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		µs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		µs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		µs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		µs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		µs

1) 1)Sensing voltage can be –1 V for t ≤ 50 µsec; in steady state V_{sens} min ≥ –0.5 V
2) See fig. 2
3) See fig. 4.
4) The load must be a pure resistor

Figure 1 : Typical Saturation Voltage vs. Output Current.

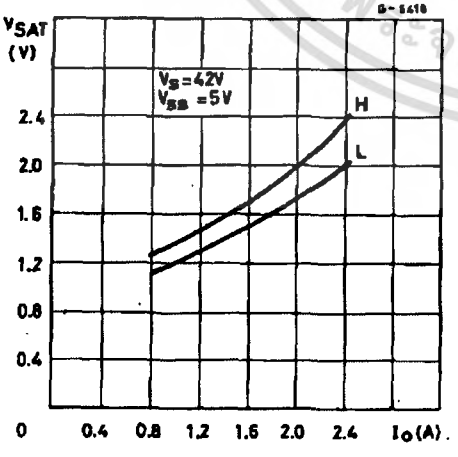
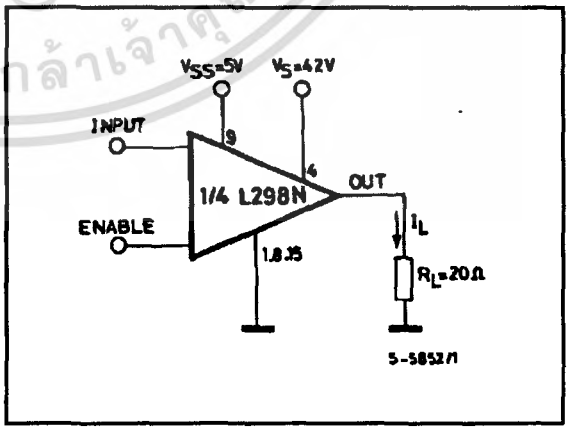


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19-4323, Rev 11; 2/03

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multidrop RS-232 Networks

Features

Superior to Bipolar

- ♦ **Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)**
- ♦ **Low-Power Receive Mode in Shutdown (MAX223/MAX242)**
- ♦ **Meet All EIA/TIA-232E and V.28 Specifications**
- ♦ **Multiple Drivers and Receivers**
- ♦ **3-State Driver and Receiver Outputs**
- ♦ **Open-Line Detection (MAX243)**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

**Contact factory for dice specifications.*

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}) ..... -0.3V to +6V
Input Voltages
T_{IN} ..... -0.3V to (V_{CC} - 0.3V)
R_{IN} (Except MAX220) ..... ±30V
R_{IN} (MAX220) ..... ±25V
T_{OUT} (Except MAX220) (Note 1) ..... ±15V
T_{OUT} (MAX220) ..... ±13.2V
Output Voltages
T_{OUT} ..... ±15V
R_{OUT} ..... -0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND ..... Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) ... 842mW
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C) ... 889mW
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) .. 440mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C) ... 696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
18-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C) 800mW
20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
18-Pin CERDIP (derate 10.53mW/°C above +70°C) 842mW
Operating Temperature Ranges
MAX2_ _AC_ _, MAX2_ _C_ _ ..... 0°C to +70°C
MAX2_ _AE_ _, MAX2_ _E_ _ ..... -40°C to +85°C
MAX2_ _AM_ _, MAX2_ _M_ _ ..... -55°C to +125°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10s) ..... +300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, $\overline{SHDN}$ or V_{CC} = 0V.
Note 2: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220: V_{CC} = 5.0V		2.4			
Logic Pull-Up/Input Current	All except MAX220, normal operation			5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/242			±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V, V_{OUT} = ±15V			±0.01	±10	
Data Rate				200	116	kbps
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V		±7	±22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V_{CC} = 5V	All except MAX243 $R2_{IN}$	0.8	1.3		V
		MAX243 $R2_{IN}$ (Note 2)	-3			
RS-232 Input Threshold High	V_{CC} = 5V	All except MAX243 $R2_{IN}$		1.8	2.4	V
		MAX243 $R2_{IN}$ (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, V_{CC} = 5V, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND		-2	-10		mA
	Shrinking V_{OUT} = V_{CC}		10	30		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
TTL/CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC} ($\overline{SHDN}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA
$\overline{EN}$ Input Threshold Low	MAX242			1.4	0.8	V
$\overline{EN}$ Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V_{CC} Supply Current ($\overline{SHDN}$ = V_{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
		MAX222/232A/233A/242/243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/232A/233A/242/243		15		
Shutdown Supply Current	MAX222/242	T_A = +25°C		0.1	10	µA
		T_A = 0°C to +70°C		2	50	
		T_A = -40°C to +85°C		2	50	
		T_A = -55°C to +125°C		35	100	
$\overline{SHDN}$ Input Leakage Current	MAX222/242				±1	µA
$\overline{SHDN}$ Threshold Low	MAX222/242			1.4	0.8	V
$\overline{SHDN}$ Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t_{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	µs
		MAX220		4	10	
	t_{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Receiver Propagation Delay RS-232 to TLL (Normal Operation), Figure 2	t_{PHLR}	MAX222/232A/233A/242/243		0.5	1	µs
		MAX220		0.6	3	
	t_{PLHR}	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (Shutdown), Figure 2	t_{PHLS}	MAX242		0.5	10	µs
	t_{PLHS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t_{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time ($\overline{SHDN}$ Goes High), Figure 4	t_{ET}	MAX222/242, 0.1µF caps (includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time ($\overline{SHDN}$ Goes Low), Figure 4	t_{DT}	MAX222/242, 0.1µF caps		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t_{PHLT} - t_{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	t_{PHLR} - t_{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

Note 3: MAX243 $R2_{OUT}$ is guaranteed to be low when $R2_{IN}$ is ≥ 0V or is floating.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้