เครื่องควบคุมหุ่นยนต์

ROBOT CONTROLLER

โดย

นายกฤษดา	อิ้มทับ	37013327
นายชัยวัฒน์	เอี่ยมสมบูรณ์	37013333
นายพิเชษฐ	รัตนเจริญชัย	37013349

วัน เดือน ปี -1 ตค 2541
เลขทะเบียน 038355
เลขเรียกหนังสือ T ๙๑๗๕ ก 2539 ค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีอิเลคทรอนิคส์

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ เครื่องควบคุมหุ่นยนต์

ROBOT CONTROLLER

จัดทำโดย

นายกฤษดา	อิ้มทับ	37013327
นายชัยวัฒน์	เอี่ยมสมบูรณ์	37013333
นายพิเชษฐ	รัตนเจริญชัย	37013349

อาจารย์ที่ปรึกษา อาจารย์ ไพศาล สิทธิโยภาสกุล

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2539

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของขบวนการศึกษาหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการปริญญานิพนธ์

..... ประธานกรรมการ

(.....)

..... กรรมการ

(.....)

..... กรรมการ

(.....)

..... กรรมการ

(.....)

..... กรรมการ

(.....)

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมหุ่นยนต์

นายกฤษดา อิ้มทับ
นายชัยวัฒน์ เอี่ยมสมบูรณ์
นายพิเชษฐ รัตนเจริญชัย

อาจารย์ที่ปรึกษา
อาจารย์ ไพศาล สิทธิโยภาสกุล

บทคัดย่อ

โครงงานนี้มีลักษณะการทำงานคือ จะควบคุมให้หุ่นยนต์ซึ่งทำงานด้วยระบบนิวแมติกเคลื่อนไหวเป็นท่าทางที่สอดคล้องกับเสียงเพลงหรือเสียงพูด โดยที่ท่าทางและเสียงพูดเราได้จัดทำเอาไว้แล้ว หลักการควบคุมจะใช้การอ่านสัญญาณควบคุมจากเทปบันทึกเสียง เทปบันทึกเสียงที่ใช้จะเป็นแบบสเตอริโอ โดยที่เราจะบันทึกสัญญาณควบคุมไว้ที่ช่องสัญญาณช่องใดช่องหนึ่งของเทปบันทึกเสียง ส่วนอีกช่องจะบันทึกสัญญาณเสียงที่สอดคล้องกันเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROBOT CONTROLLER

MR.KRITSADA IMTUB

MR.CHAIWAT EAMSOMBOON

MR.PICHET RATTANAJARERNCHAI

ADVISOR

MR.PISAN SITTIYOPASAKUL

ABSTACT

This project has ability to control a movement of Pneumatic Robot which is correspond to a sound. However,the movement and sound is designed earlier. A principle of controlled is reading the control signal from stereo tape recording which is recorded on the left or right channel of tape while the other is recorded the sound that correspond to the movement of Robot.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ บิดา มารดา อาจารย์ ไพศาล สิทธิโยกาสกุล ซึ่งเป็นอาจารย์ที่ปรึกษา ตลอดจนอาจารย์ทุกท่าน ที่ประสิทธิประสาทวิชาความรู้ให้แก่พวกผมทุกคน อันเป็นผลให้พวกผมมีวันนี้ และทำให้ปริญญานิพนธ์นี้สำเร็จลงได้โดยสมบูรณ์

ขอขอบคุณ บริษัท อีทีที จำกัด ที่กรุณาให้ความรู้เกี่ยวกับ ไมโครโปรเซสเซอร์ Z80 และการเขียนโปรแกรมต่างๆ มากมาย ตลอดจนการนำเอาไมโครโปรเซสเซอร์มาประยุกต์ใช้งานกับฮาร์ดแวร์ต่างๆ

ขอบคุณ หนังสือทุกเล่มที่ใช้ในการจัดทำปริญญานิพนธ์เล่มนี้ ให้สำเร็จลุล่วงไปได้ดี

นายกฤษดา	อิ้มทับ	37013327
นายชัยวัฒน์	เอี่ยมสมบูรณ์	37013333
นายพิเชษฐ	รัตนเจริญชัย	37013349

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 การทำงานของชุดขับโซลีนอยด์วาล์ว	2
1.2 การทำงานของกระบอกสูบที่ตำแหน่งต่างๆ	7
1.3 การควบคุมพื้นฐาน	8
บทที่ 2 การออกแบบและหลักการทำงาน	9
2.1 การบันทึกข้อมูลลงเทป	9
2.2 หลักการทำงาน	11
2.3 การออกแบบวงจร	12
2.3.1 ภาคแสดงผลและคีย์บอร์ด	12
2.3.2 ภาคอินพุท/เอาท์พุทและอินเตอร์เฟซ	14
2.4 การทำงานของวงจร	16
บทที่ 3 การใช้งานโปรแกรม	18
3.1 การใช้งานโปรแกรม DATA	18
3.1.1 การแก้ไขข้อมูล	19
3.2 การใช้งานโปรแกรม REC	22
3.3 การใช้งานโปรแกรม DESIGN	24
3.4 การใช้งานโปรแกรม LIST	28
3.5 การใช้งานโปรแกรม SAVE	29
3.6 การใช้งานโปรแกรม LOAD	30
3.7 การใช้งานโปรแกรม RUN	31
3.8 การใช้งานโปรแกรม PRINT	31
3.9 การใช้งานโปรแกรม PCLINK	32
3.10 การใช้งานโปรแกรม MOVE	32
บทที่ 4 บทสรุป	34
4.1 การใช้งานเครื่อง	34
4.1.1 การบันทึก	34
4.1.2 การเล่นกลับ	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 อุปสรรคและปัญหา 35
4.2.1 ปัญหาจากเครื่องเล่นเทป 35
4.2.2 ความแรงของสัญญาณ 35
4.2.3 ปัญหาทั่วไป 35
หนังสืออ้างอิง 36
ภาคผนวก ก. รายละเอียดทางฮาร์ดแวร์ 37
ภาคผนวก ข. โปรแกรม 47
ภาคผนวก ค. รายละเอียดทั่วไป 81
ภาคผนวก ง. DATA SHEET 87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โครงงานนี้ชื่อ โครงงานเครื่องควบคุมหุ่นยนต์ ได้จัดทำขึ้นในปีการศึกษา 2539 มีลักษณะการทำงานคือ จะทำการควบคุมให้หุ่นยนต์ทำท่าทางเคลื่อนไหวประกอบกับเสียงเพลง หรือคำพูดที่สอดคล้องกับท่าทางที่แสดงออกมา การควบคุมทั้งหมดใช้ไมโครโปรเซสเซอร์เบอร์ Z80 ซึ่งเป็นไมโครโปรเซสเซอร์ขนาด 16 บิท เป็นตัวควบคุมการทำงานทั้งหมดของวงจร

หลักการควบคุม ใช้หลักการอ่านสัญญาณจากเทปบันทึกเสียงแบบสเตอริโอซึ่งทั้ง 2 แชนแนลของเทปบันทึกเสียงจะบันทึกสัญญาณที่แตกต่างกัน แชนแนลแรกบันทึก แอดเดรส ของข้อมูลที่จะใช้ในการควบคุมการเคลื่อนไหวของหุ่นยนต์ ส่วนอีกแชนแนลจะบันทึกสัญญาณเสียงที่สอดคล้องกับท่าทางของหุ่นยนต์ที่เคลื่อนไหวตามข้อมูลในแอดเดรสที่อ่านได้จาก แชนแนลแรก โดยโครงงานนี้ มีหน้าที่ทั้งบันทึกข้อมูลลงเทป และอ่านข้อมูลจากเทปบันทึกเสียง ซึ่งจะกล่าวในรายละเอียดต่อไป

หุ่นยนต์ที่จะควบคุมในโครงงานนี้ ทำงานด้วยระบบ นิวแมติก ซึ่งสร้างขึ้นภายใต้โครงงานชื่อ "หุ่นยนต์ควบคุมโดย PC" ซึ่งจัดทำขึ้นในปีการศึกษา 2538 พร้อมกับมีชุด ควบคุม โซลีนอยด์วาล์ว ให้มาด้วย ดังนั้นเพื่อความเข้าใจอย่างแท้จริงถึงการทำงานและวิธีการควบคุมของโครงงานนี้ จึงจะขอกล่าวถึงหลักการทำงานของ ชุด ควบคุม โซลีนอยด์วาล์ว อย่างคร่าวๆก่อน ซึ่งจะอยู่ในหัวข้อ การทำงานของชุด ควบคุม โซลีนอยด์วาล์ว และหัวข้อ การควบคุมพื้นฐาน ต่อจากนั้น จะเป็นแนวความคิดของโครงงานนี้ ซึ่งจะกล่าวถึง แนวความคิดในการเขียนหรืออ่านสัญญาณดิจิตอล ลงบนเทปบันทึกเสียง ว่ากระทำได้อย่างไร ในหัวข้อ การทำงานของเครื่องควบคุม ก็จะกล่าวถึงหลักการทำงานของโครงงานนี้อย่างละเอียด ต่อไปก็เป็น การใช้งานโปรแกรม ที่ใช้ในการจัดทำข้อมูลที่ใช้ควบคุมการเคลื่อนไหวของหุ่นยนต์ พร้อมทั้งโปรแกรมอำนวยความสะดวกต่างๆ ที่จำเป็นต้องใช้

สุดท้ายของปริญญานิพนธ์เล่มนี้ ก็จะเป็นรายละเอียดต่างๆ เกี่ยวกับเครื่องควบคุม อาทิเช่น วงจร ลายวงจร KeyCode ตำแหน่งขาคอนเนคเตอร์ ฯลฯ จากนั้นก็จะเป็น Source โปรแกรม ของเครื่องควบคุมซึ่งจะเป็นภาษา Assembly ของ Z80 โดยจะใช้โปรแกรม X80.EXE และ LINK.COM เป็นโปรแกรมคอมไฟล์ แต่เพื่อความสะดวกได้จัดทำ แบทช์ไฟล์ ที่ใช้สำหรับ คอมไพล์ และ link Source โปรแกรม เอาไว้ด้วย นอกจากนั้นยังสามารถใช้ แบทช์ไฟล์ดังกล่าว ส่งโปรแกรมที่คอมไพล์เป็น HEX ไฟล์แล้วไปให้บอร์ดได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 การทำงานของชุด ควบคุม โซลีนอยด์วาล์ว

ชุด ควบคุม โซลีนอยด์วาล์ว ได้แก่ชุดที่จะรับข้อมูลเลขฐาน 16 จากเครื่องควบคุม และทำการส่งไปขับชุดโซลีนอยด์วาล์วให้ทำงาน เพื่อปิดหรือเปิดลมที่กระบอกสูบนิวแมติก ที่ต่ออยู่กับส่วนต่างๆ ของหุ่นยนต์ ทำให้เกิดการเคลื่อนไหวของหุ่นยนต์ ชุด ควบคุม โซลีนอยด์วาล์ว ดังกล่าวประกอบด้วย พอร์ต 3 พอร์ต มีหน้าที่ รับข้อมูลเลขฐาน 16 จากเครื่องควบคุมมาพักเอาไว้ นอกจากนั้นยังมีชุด ที-ฟลิปฟลอป และ ทรานซิสเตอร์ ประกอบกันขึ้นเป็นวงจร Drive โซลีนอยด์วาล์ว อีก 16 ชุด แต่ละชุดก็ต่อกับแต่ละบิทของพอร์ต จำนวน 2 พอร์ต ส่วนอีก 1 พอร์ตสร้างเอาไว้เผื่อการเพิ่มเติมส่วนต่างๆ ของหุ่นยนต์ในอนาคต โซลีนอยด์วาล์วแต่ละตัวจะใช้ควบคุมกระบอกสูบหนึ่งอันหรือมากกว่า

วงจรของชุด ควบคุม โซลีนอยด์วาล์ว แสดงดังวงจรที่ 1.1 IC 74LS374 ทั้ง 3 ตัวทำหน้าที่เป็นพอร์ตรับข้อมูลจากเครื่องควบคุม มาพักเอาไว้ก่อนแล้วค่อยส่งต่อให้วงจร ควบคุม ในวงจรที่ 1.2 IC 74LS138 จะเป็นตัวเลือกว่าจะให้พอร์ตตัวไหนรับข้อมูลที่เข้ามาทางอินพุท โดยจะสร้าง clock ให้แก่ IC 74LS374 ตัวนั้น การสร้าง clock ก็สร้างมาจากเลขฐาน 16 ที่ทางเครื่องควบคุมส่งมาให้ เครื่องควบคุมจะส่งข้อมูลมาให้ชุด ควบคุม 2 ชุดคือ ชุดแรก เป็นข้อมูลในการปิดเปิดโซลีนอยด์วาล์วแต่ละตัว 1 ตัวต่อ 1 บิท โดยจะส่งมาทาง J1 ชุดที่สองเป็นข้อมูลที่ใช้ในการสร้าง clock ให้แก่พอร์ตตัวที่ต้องการให้รับข้อมูลที่เข้ามาทาง J1 โดยจะส่งมาทาง J2 ข้อมูลชุดที่สองแสดงดังตาราง 1.1

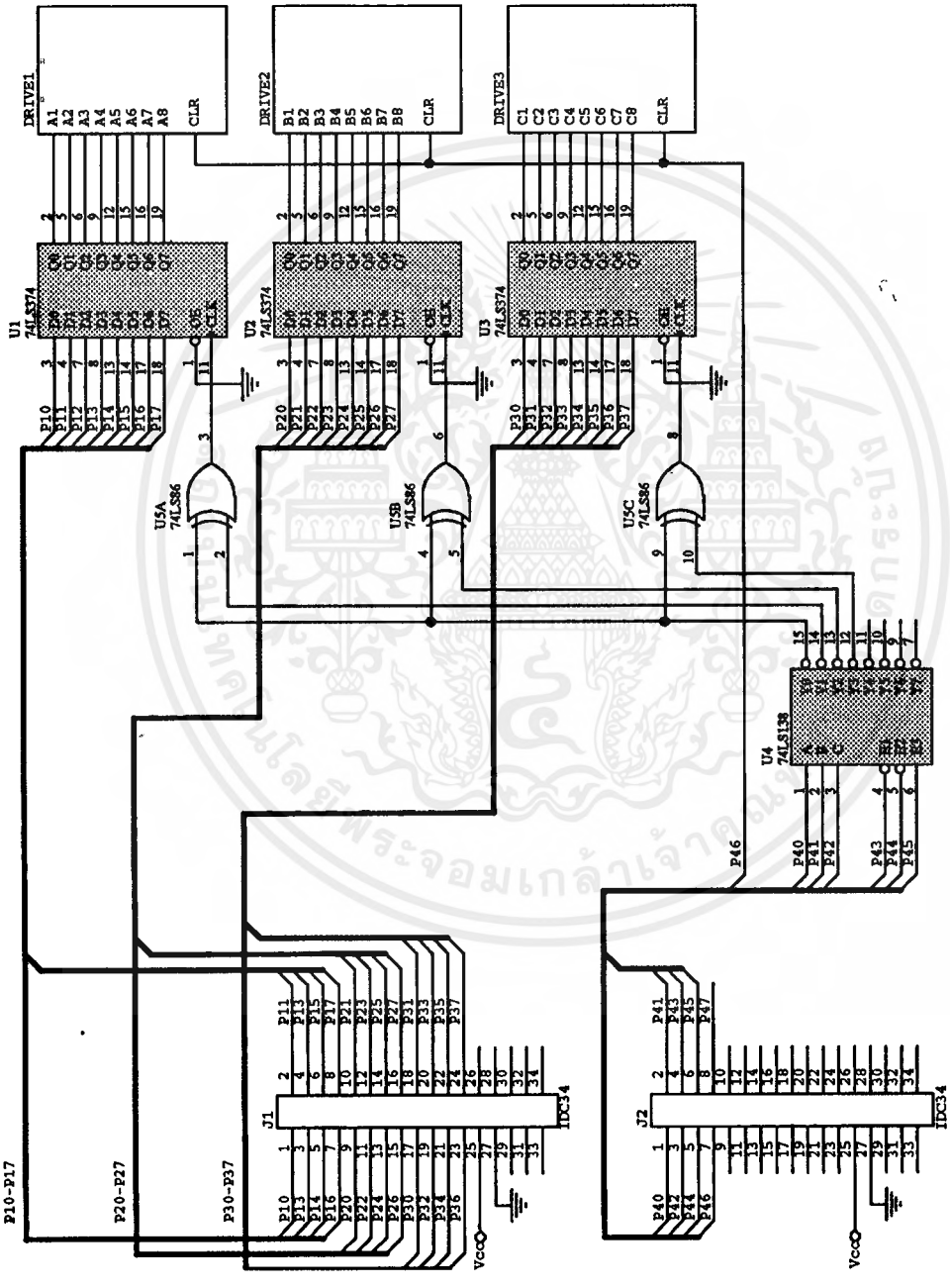
7	6	5	4	3	2	1	0	HEX	ความหมาย
-	CLR	G2B	G2A	G1	C	B	A		
0	0	0	0	0	0	0	0	00	Clear all drive
0	1	0	0	0	0	0	0	40	clock"0"
0	1	0	0	1	0	0	0	48	latch all
0	1	0	0	1	0	0	1	49	latch 1
0	1	0	0	1	0	1	0	4A	latch 2
0	1	0	0	1	0	1	1	4B	latch 3

ตาราง 1.1 แสดงข้อมูลที่ใช้ในการสร้าง clock ให้แก่พอร์ตในชุด ควบคุมโซลีนอยด์วาล์ว

จากตาราง 1.1 หากข้อมูลชุดที่สองเป็น 00H ก็จะเป็นการเคลียร์โซลีนอยด์วาล์วทุกตัวหุ่นยนต์จะกลับมาสู่ท่าเตรียม ,40H เป็นการสร้าง clock โลจิก 0 ให้แก่ IC 74LS374 หลังจากสร้าง clock โลจิก 0 แล้ว ถ้าต้องการให้พอร์ต 1 รับข้อมูลชุดแรกที่เข้ามาทาง J1 เข้ามาเก็บเอาไว้ ก็ทำได้โดยส่ง

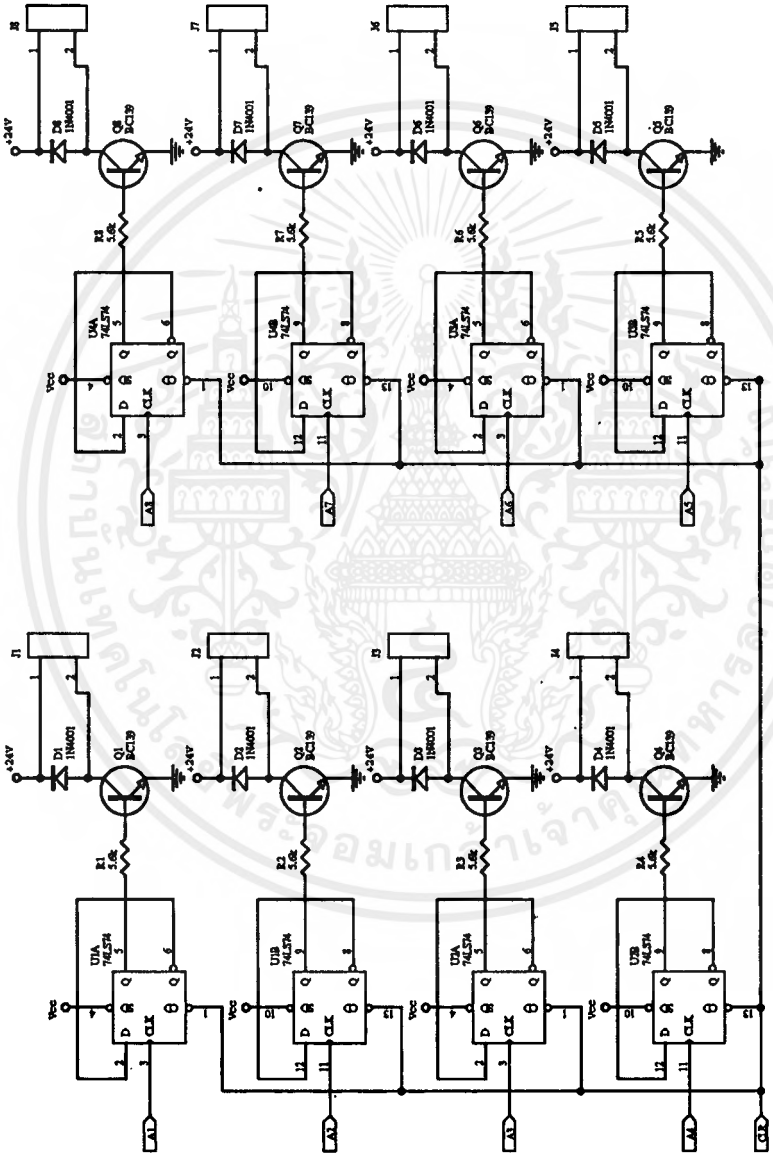
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลข 49H หลังจากส่ง 40H เสร็จ (ส่ง 40H ก่อนแล้วตามด้วย 49H) หรือ 4AH,4BH สำหรับพอร์ต 2 และพอร์ต 3 ตามลำดับ ส่วน 48H เป็นการสร้าง clock ให้แก่พอร์ตทุกตัว คือให้พอร์ตทุกตัวรับข้อมูลชุดแรกที่เข้ามาทั้งหมด



วงจร 1.1 วงจรควบคุมไซลินอยด์วาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจร 1.2 วงจร Drive ไซลินอยด์วาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นในการควบคุมให้หุ่นยนต์เคลื่อนไหวก็คือการสั่งให้กระบอกสูบที่ต่ออยู่กับส่วนต่างๆ ของหุ่นทำงาน โดยจะต้องส่งข้อมูลทั้งหมด 2 ชุดคือ

ชุดแรก เป็นข้อมูลที่ใช้สำหรับปิดหรือเปิดโซลีนอยด์วาล์วแต่ละตัว 1 บิท ต่อ 1 ตัว

ชุดสอง เป็นข้อมูลในการสร้าง clock ให้แก่พอร์ตที่ต้องการให้รับข้อมูลชุดแรก

ในการควบคุมจริงๆ ผู้ใช้สามารถออกแบบได้แค่ข้อมูลชุดแรกกับค่าหน่วงเวลาซึ่งจะกล่าวในภายหลังเท่านั้น ส่วนข้อมูลชุดที่สองที่ใช้สร้าง clock เป็นข้อมูลตายตัวซึ่งจะปล่อยให้เป็นหน้าที่ของโปรแกรมเป็นตัวจัดการแทน

รายละเอียดของโซลีนอยด์วาล์วที่ต่ออยู่กับแต่ละบิทของพอร์ตแสดงดังตาราง 1.2

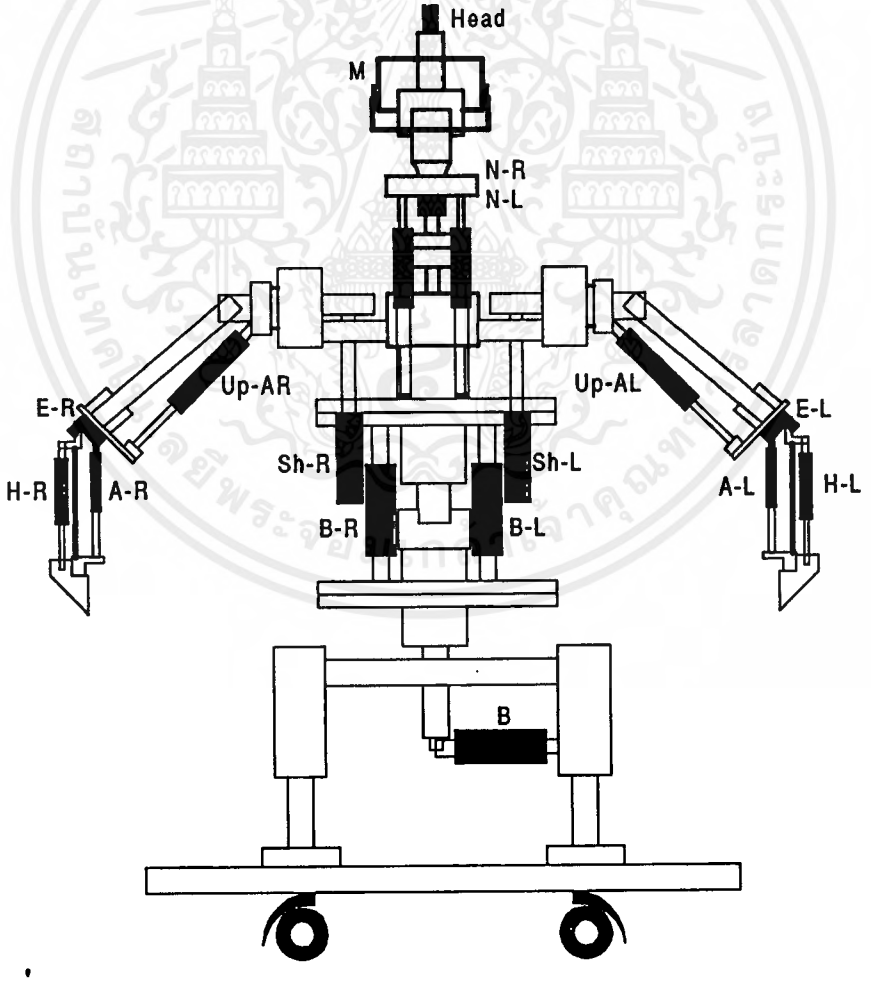
บิท	สัญลักษณ์	ตำแหน่งที่ควบคุม
P10	N_L	คอ (หันซ้าย)
P11	N_R	คอ (หันขวา)
P12	A_R	แขนท่อนล่าง ข้างขวา
P13	B_R,B_L	หน้าอก
P14	Sh_R	หัวไหล่ ข้างขวา
P15	B	ลำตัว
P16	M	ปาก
P17	Up_AL	แขนท่อนบน ข้างซ้าย
P20	E_R	ข้อศอก ข้างขวา
P21	Up_AR	แขนท่อนบน ข้างขวา
P22	H_R	ข้อมือ ข้างขวา
P23	Sh_L	หัวไหล่ ข้างซ้าย
P24	Head	หัว
P25	E_L	ข้อศอก ข้างซ้าย
P26	A_L	แขนท่อนล่าง ข้างซ้าย
P27	H_L	ข้อมือ ข้างซ้าย

ตาราง 1.2 แสดงการต่อโซลีนอยด์วาล์วเข้ากับพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของกระบอกสูบต่างๆ ที่ติดตั้งอยู่ที่แต่ละส่วนของหุ่นยนต์แสดงดังรูป 1.1 กระบอกสูบแต่ละตัวก็จะถูกควบคุมโดยโซลีนอยด์วาล์วที่ต่ออยู่กับบิทต่างๆของพอร์ตดังที่แสดงในตาราง 1.2

จากที่กล่าวมาทั้งหมดแล้วนั้นเราก็ได้ทราบว่า ถ้าเราต้องการให้กระบอกสูบตัวใดทำงานหรือในอีกทางหนึ่ง คือต้องการให้หุ่นยนต์เคลื่อนไหวส่วนใด เราก็จะต้องส่งข้อมูลเลขฐาน 16 ให้แก่พอร์ตใดในชุด ควบคุม โซลีนอยด์วาล์ว เป็นการค่อนข้างลำบากเล็กน้อย ถ้าจะต้องหาว่าส่วนของหุ่นที่เราต้องการให้เคลื่อนไหวนั้นมีกระบอกสูบที่ต่ออยู่กับโซลีนอยด์วาล์วตัวใดและโซลีนอยด์วาล์วตัวนั้นต่ออยู่กับบิทใดของพอร์ตที่เท่าไร ในโครงงานนี้ได้จัดทำโปรแกรมที่ใช้ออกแบบท่าทางของหุ่นยนต์เอาไว้แล้ว ชื่อ โปรแกรม DESIGN โปรแกรม DESIGN จะบอกค่าเลขฐาน 16 ที่จะต้องส่งให้แก่พอร์ตทั้ง 3 พอร์ตออกทางจอ 7-Segment โดยเลขฐาน 16 ดังกล่าวจะสอดคล้องกับท่าทางของหุ่นที่แสดงออกมาให้เห็น รายละเอียดการใช้งานสามารถอ่านได้ในหัวข้อ การใช้งานโปรแกรม DESIGN ในปริญญานิพนธ์เล่มนี้



รูป 1.1 แสดงตำแหน่งของกระบอกสูบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 การทำงานของกระบอกสูบที่ตำแหน่งต่างๆ

ในการทำให้หุ่นยนต์เคลื่อนไหวเป็นท่าทางต่างๆ จะต้องกำหนดว่าให้กระบอกสูบตัวใดทำงานและความเร็วของก้านสูบที่เคลื่อนที่เข้าหรือเคลื่อนที่ออก ต้องมีการควบคุมที่เหมาะสม ในการควบคุมความเร็วของก้านสูบ สามารถทำได้โดยการปรับที่วาล์วควบคุมการไหลของลม (Flow control valve) ที่อยู่ตรงรูลมเข้าและลมออกของกระบอกสูบ และอยู่ระหว่างโซลินอยด์วาล์วกับกระบอกสูบ สามารถปรับให้ความเร็วของก้านสูบเคลื่อนที่เข้าหรือเคลื่อนที่ออกไม่เท่ากันหรือเท่ากันได้ ซึ่งจะมีวาล์วที่ทำให้ลมไหลได้ทางเดียวต่อร่วมอยู่ด้วย ซึ่งถ้าไม่มีวาล์วตัวนี้จะควบคุมความเร็วของก้านสูบได้ยาก

หน้าที่ของกระบอกสูบแต่ละตัวซึ่งกำหนดเป็นอักษรย่อตามตำแหน่งต่างๆดังนี้

- Head จะทำให้หุ่นยนต์ก้มศรีษะลงมา ใช้ Cylinder Ø 10 STK 30 single acting,sping return,port M5 ใช้ Soliniod valve 3/2 DC 24 V.

- M จะทำให้หุ่นยนต์ขยับปากขึ้นลง ใช้ Cylinder Ø 10 STK 30 single acting,sping return,port M5 ใช้ Soliniod valve 3/2 DC 24 V.

- N_R และ N_L จะทำให้หุ่นยนต์หมุนคอไปทางด้านขวาและซ้าย ใช้ Rotary actuator port M5 ใช้ Solinoid valve 5/3 DC 24 V. closed center

- Up_AR และ Up_AL จะทำให้หุ่นยนต์ยกแขนในลักษณะทิศทางออกหรือเข้าหาลำตัว ใช้ Cylinder Ø 20 STK 50 double acting universal mount port 1/8 นิ้ว ใช้ Soliniod valve 5/2 DC 24 V.

- E_R และ E_L จะทำให้หุ่นยนต์หมุนปลายท่อนแขนได้ 180 องศา ใช้ Rotary actuator port M5 ใช้ Soliniod valve 5/2 DC 24 V.

- A_R และ A_L จะทำให้หุ่นยนต์ยกปลายท่อนแขนออกหรือเข้าหาลำตัว ใช้ Cylinder Ø 10 STK 50 double acting universal mount port M5 ใช้ Soliniod valve 5/2 DC 24 V.

- H_R และ H_L จะทำให้หุ่นยนต์ขยับมือขึ้นลงในทิศทางออกหรือเข้าหาลำตัว ใช้ Cylinder Ø 10 STK 25 double acting universal mount port M5 ใช้ Soliniod valve 5/2 DC 24 V.

- Sh_R และ Sh_L จะทำให้หุ่นยนต์ยกแขนในลักษณะทิศทางขึ้นลงในแนวดิ่ง ใช้ Cylinder Ø 20 STK 50 double acting port 1/8 นิ้ว ใช้ Soliniod valve 5/2 DC 24 V.

- B_R และ B_L จะทำให้หุ่นยนต์ก้มตัวไปข้างหน้า ใช้ Cylinder Ø 40 STK 20 double acting stud mout port 1/8 นิ้ว ใช้ Soliniod valve 5/2 DC 24 V.

- B จะทำให้หุ่นยนต์หมุนตัว ใช้ Cylinder Ø 32 STK 30 double acting universal mount port 1/8 นิ้ว ใช้ Soliniod valve 5/2 DC 24 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ กระบอกสูบที่ตำแหน่ง B_R และ B_L จะทำงานพร้อมกันเสมอเพราะว่าใช้โซลินอยด์วาล์วเพียง 1 ตัวเป็นตัวควบคุม ส่วนกระบอกสูบที่ตำแหน่งอื่นๆควบคุมแยกได้ทั้งขวาและซ้าย

1.3 การควบคุมพื้นฐาน

การควบคุมให้หุ่นยนต์เคลื่อนไหว เป็นท่าทางที่ต่อเนื่อง เราก็ต้องส่งข้อมูลไปให้ชุด ควบคุมโซลีนอยด์วาล์ว อย่างต่อเนื่องเช่นกัน และจะต้องมีการหน่วงเวลาเพื่อให้กระบอกสูบสามารถทำงานได้ทันเพราะเราใช้ไมโครโปรเซสเซอร์เป็นตัวส่งข้อมูล ดังนั้นข้อมูลในการควบคุมการเคลื่อนไหวก็จะประกอบด้วย 2 ส่วนด้วยกันคือ ส่วนแรกเป็นข้อมูลที่ควบคุมส่วนต่างๆของหุ่นยนต์ จำนวน 3 ไบท์ และส่วนที่สองเป็นค่าหน่วงเวลา ในที่นี้ใช้ 2 ไบท์ ทั้ง 5 ไบท์นี้ ผู้ใช้เป็นคนออกแบบเองตามต้องการส่วนข้อมูลที่ใช้ในการสร้าง clock ให้แก่ พอร์ต 74LS374 ในชุด ควบคุม โซลีนอยด์วาล์วนั้นเป็นค่าตายตัวตามตาราง 1.1 ผู้ใช้จึงไม่ต้องออกแบบ เป็นหน้าที่ของโปรแกรม

สมมุติว่าเราได้ออกแบบท่าทางการเคลื่อนไหวของหุ่นยนต์เอาไว้แล้ว และนำไปใส่ในหน่วยความจำของเครื่องโดยโปรแกรม DATA เรียบร้อยแล้วดังนี้

Address	Value				
2000	01	02	03	E8	03
2005	04	05	06	E8	03
200A	07	08	09	E8	03
200F	0A	0B	0C	E8	03
2014	0D	0E	0F	E8	03
2019	10	11	12	E8	03
	data			delay	

จากข้อมูลข้างบนจะเห็นได้ว่าในแต่ละบรรทัดจะแบ่งออกเป็นสองส่วนคือ ส่วนของ data และส่วนของ delay ส่วนของ data ก็จะเรียงกันไปเริ่มจากไบท์แรกเป็นข้อมูลสำหรับส่งให้แก่พอร์ต 1 ต่อไปก็เป็น พอร์ต 2 และ พอร์ต 3 ตามลำดับ ส่วนค่า delay ใช้ 2 ไบท์จึงมีค่าอยู่ระหว่าง 0-FFFFH หรือ 0-65535 โดย 1 หน่วยมีค่าเท่ากับ 1 mSec ดังนั้นจึงสามารถหน่วงเวลาได้สูงสุดเท่ากับ 65535 mSec หรือ 65.34 วินาที ค่า delay เมื่อออกแบบได้แล้วการใส่ลงหน่วยความจำให้ใส่สลับไบท์กันระหว่าง ไบท์สูง และ ไบท์ต่ำ

ข้อมูลแต่ละบรรทัดก็จะทำให้หุ่นยนต์เคลื่อนไหว 1 step 6 บรรทัดก็เคลื่อนไหว 6 step โดยแต่ละ step จะหน่วงเวลาเท่ากับค่า delay ในแต่ละ step

เมื่อโปรแกรมอ่านข้อมูลจากหน่วยความจำก็จะส่งข้อมูลในส่วนของ data ทั้ง 3 ไบท์ไปให้พอร์ต 1ถึง 3 ของชุด ควบคุม จากนั้นโปรแกรมจะหน่วงเวลาเท่ากับส่วนของ delay เมื่อหน่วงเวลาเสร็จโปรแกรมก็จะนำเอา 3 ไบท์ต่อไปส่งให้ชุด ควบคุม และหน่วงเวลาอีก ทำเช่นนี้ไปเรื่อยๆ ก็จะทำให้หุ่นยนต์เคลื่อนไหวในท่าทางที่ต่อเนื่องออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การออกแบบและหลักการทำงาน

จากที่กล่าวมาในบทนำว่าในโครงงานนี้จะต้องทำการบันทึกสัญญาณดิจิตอลก็คือแอดเดรสของข้อมูลที่ใช้ควบคุมการเคลื่อนไหวของหุ่น ลงบนเทปบันทึกเสียง แชนแนลใดแชนแนลหนึ่ง ส่วนอีกแชนแนลจะบันทึกสัญญาณเสียงที่สอดคล้องกับท่าทางการเคลื่อนไหวของหุ่นยนต์ที่แสดงออกมาตามข้อมูลในแอดเดรสที่อ่านได้จากแชนแนลแรก ในหัวข้อนี้จะกล่าวถึงแนวความคิดในการบันทึกสัญญาณดิจิตอลลงเทปบันทึกเสียงว่าทำได้อย่างไร ต่อจากนั้นจะเป็นการออกแบบวงจรและหลักการทำงานอย่างคร่าวๆ ของโครงงานนี้ ซึ่งมีหน้าที่อ่านแอดเดรสจากเทปบันทึกเสียงแล้วนำข้อมูลในแอดเดรสที่อ่านได้ส่งไปควบคุมหุ่นยนต์ ในขณะเดียวกันก็จะเปิดทางให้สัญญาณเสียงซึ่งอยู่อีกแชนแนลหนึ่งของเทปบันทึกเสียงผ่านออกไปสู่วงจรขยายเสียงภายนอกเพื่อใช้เป็นสัญญาณเสียงประกอบท่าทางการเคลื่อนไหวของหุ่นยนต์ต่อไป นอกจากการอ่านข้อมูลจากเทปบันทึกเสียงแล้ว โครงงานนี้ยังมีหน้าที่บันทึก ข้อมูลลงเทปบันทึกเสียงอีกด้วย ซึ่งจะกล่าวในรายละเอียดต่อไป

2.1 การบันทึกข้อมูลลงเทป

การบันทึกสัญญาณดิจิตอลลงเทปบันทึกเสียงจะไม่ใช้การบันทึกโลจิก 0 และ โลจิก 1 ไปโดยตรง เพราะในการอ่านกลับจะตัดสินใจได้ยากว่า ข้อมูลที่อ่านเข้ามาเป็น โลจิก 0 หรือ โลจิก 1 สาเหตุมาจากการที่จะต้องมีวงจรขยายสัญญาณที่อ่านออกมาจากหัวเทป เพื่อให้มีขนาดของสัญญาณแรงพอ วงจรนั้นได้แก่วงจรปรีเทป เมื่อมีวงจรขยายก็ต้องพิจารณาอีกว่าสัญญาณ output จะ out of phase กับสัญญาณ input หรือจะ inphase กันด้วยเหตุนี้เองทำให้ไม่สามารถตัดสินใจได้ว่า สัญญาณที่อ่านได้เป็นโลจิก 0 หรือ โลจิก 1 ถ้าเราบันทึกโลจิก 0 หรือ 1 นั้นลงไปโดยตรง

ทางที่จะแก้ปัญหาให้สามารถบันทึกสัญญาณดิจิตอลลงเทปบันทึกเสียง แล้วสามารถอ่านกลับมาเป็นค่าเหมือนกับตอนที่บันทึกได้นั้น สามารถทำได้โดย ใช้การบันทึกสัญญาณที่มีความถี่แตกต่างกันระหว่างโลจิก 0 และโลจิก 1 เช่น โลจิก 0 จะใช้สัญญาณความถี่เท่ากับ 1kHz บันทึกลงเทป แต่โลจิก 0 จะใช้ความถี่ 2kHz แทน จะเห็นได้ว่าถึงแม้สัญญาณที่อ่านได้จะกลับเฟสกันอย่างไรเราไม่สนใจ เราจะตรวจสอบเฉพาะความถี่ของสัญญาณที่อ่านเข้ามาเท่านั้น จึงสามารถที่จะแยกแยะได้ว่า สัญญาณที่อ่านได้จากเทปบันทึกเสียงเป็นโลจิก 0 หรือ โลจิก 1 ได้โดยการตรวจสอบความถี่ของสัญญาณที่อ่านมาได้

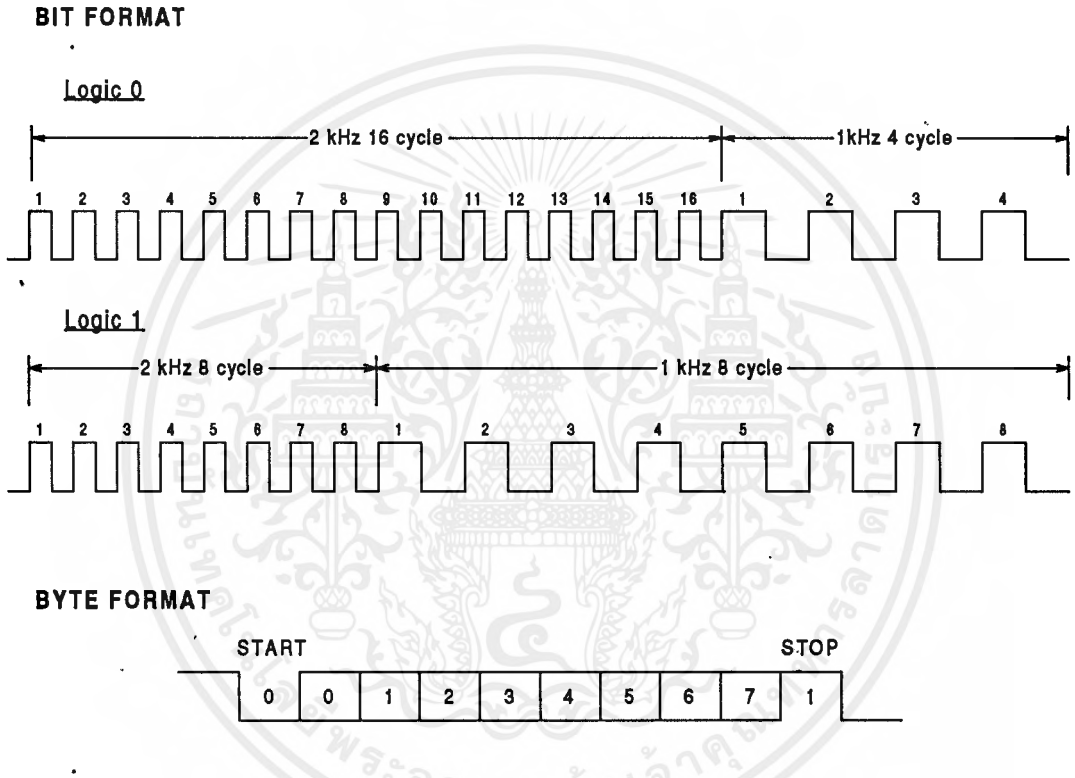
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงานนี้จะใช้หลักการบันทึกสัญญาณดิจิตอลลงเทปบันทึกเสียงเหมือนกับที่กล่าวมา คือ ใช้การบันทึกความถี่แทนโลจิก โดยในที่นี้จะใช้สัญญาณสแควร์เวฟเป็นสัญญาณในการบันทึก และให้

โลจิก 0 แทนด้วย สัญญาณสแควร์เวฟความถี่ 2kHz 16 ไซเคิล ตามด้วย 1kHz 4 ไซเคิล

โลจิก 1 แทนด้วย สัญญาณสแควร์เวฟความถี่ 2kHz 8 ไซเคิล ตามด้วย 1kHz 8 ไซเคิล

แสดงดังรูป 2.1



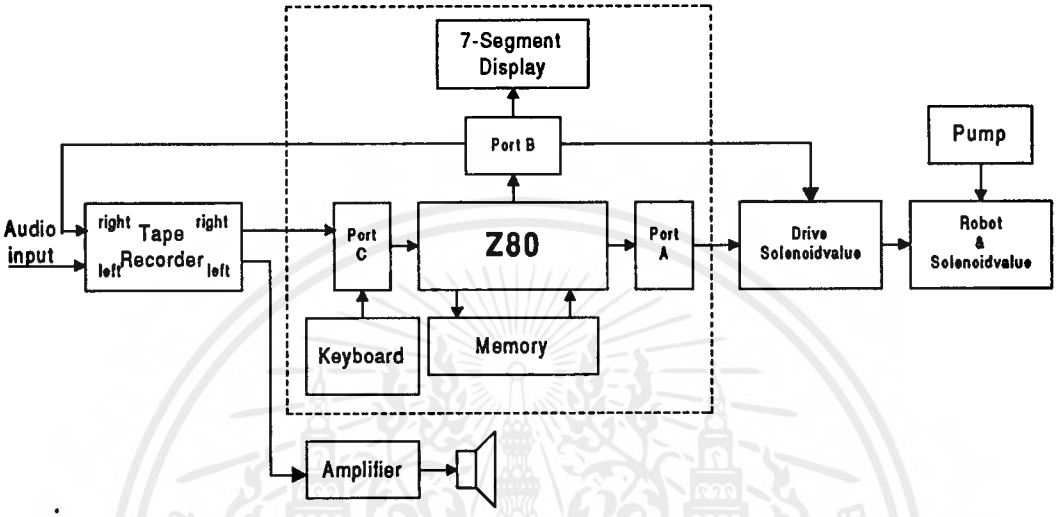
รูป 2.1 แสดงรูปแบบของสัญญาณที่ใช้แทนโลจิก 0 และ 1

ดังนั้นข้อมูล 1 บิท จะใช้เวลาในการบันทึกทั้งหมดเท่ากับ 12 mSec 1ไบท์จะใช้เวลาเท่ากับ 96 mSec แต่ในความเป็นจริงจะต้องเพิ่มบิทเพื่อตรวจสอบความถูกต้องของข้อมูลที่อ่านเข้ามาได้อีกอย่างน้อย 2 บิท คือ start bit และstop bit เพราะในการบันทึกจะเริ่มบันทึกจาก Less significant bit (LSB) ไปหา Most significant bit (MSB) เปรียบเสมือนการสื่อสารข้อมูลแบบ Asynchronous จึงต้องมี start bit และ stop bit เพิ่มเข้าไปในข้อมูล ดังนั้นในข้อมูล 1 ไบท์ที่จะบันทึกลงเทปบันทึกเสียงจะประกอบด้วยข้อมูล 8 บิท start bit กับ stop bit อีกอย่างละ 1 บิท รวมเป็น 10 บิทต่อ 1 ไบท์ จึงใช้เวลาในการบันทึกข้อมูล 1 ไบท์เท่ากับ 120 mSec และมีอัตราการบันทึกเท่ากับ 83บิทต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 หลักการทำงาน

หลักการทำงานอย่างคร่าวๆของเครื่องควบคุมแสดงดังรูปที่ 2.2 ชุดควบคุมหรือโครงงานนี้จะอยู่ภายในกรอบเส้นประ



รูปที่ 2.2 แสดงบล็อคไดอะแกรมของเครื่องควบคุม

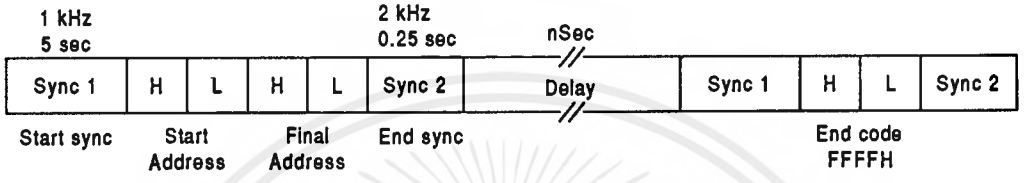
เริ่มจากการอ่านสัญญาณจากเครื่องเล่น/บันทึกเทปจะถูกอ่านเข้ามาทางพอร์ต C จากนั้นจะนำเอาข้อมูลในหน่วยความจำตรงตำแหน่งแอดเดรสที่อ่านได้จากเทปบันทึกเสียงส่งผ่านพอร์ต A ไปยังชุด Drive โซลีนอยด์วาล์ว และส่งสัญญาณควบคุมไปทางพอร์ต B (สัญญาณที่ใช้ในการสร้าง clock ให้เก่ IC 74LS374 ที่อยู่ในชุด Drive) ทำให้หุ่นยนต์เกิดการเคลื่อนไหว ในขณะเดียวกันสัญญาณเสียงอีกแชนแนลหนึ่งจะถูกส่งออกไปขยายเสียงออกทางลำโพง เมื่อครบ 1 ชุดก็จะกลับไปอ่านแอดเดรสจากพอร์ต C ใหม่ เมื่ออ่านเข้ามาแล้วก็ทำเหมือนเดิมไปเรื่อยๆจนกว่าจะอ่านเจอรหัสจบก็จะเคลียร์ให้หุ่นยนต์กลับสู่ท่าเตรียมและโปรแกรมกลับสู่มอนิเตอร์เพื่อรอรับคำสั่งต่อไป

ในการบันทึกไมโครโปรเซสเซอร์จะนำเอา แอดเดรสของข้อมูลคือ แอดเดรสเริ่มต้นและแอดเดรสสุดท้ายบันทึกลงไปบนเทปบันทึกเสียง ผ่านทางพอร์ต B ในการบันทึกจะบันทึกแอดเดรสเริ่มต้น ก่อนแล้วตามด้วย แอดเดรสสุดท้าย นอกจากนั้นทั้งสองแอดเดรสจะบันทึกไบท์สูง ลงไปก่อนแล้วค่อยบันทึกไบท์ต่ำ ตามลงไปหลังจากบันทึกแอดเดรสเสร็จแล้วก็จะนำเอาข้อมูลในแอดเดรสดังกล่าว ส่งออกไปควบคุมหุ่นยนต์เหมือนกับตอนบันทึกที่กล่าวมาแล้ว เหตุที่ทำแบบนี้เพื่อให้ผู้ใช้สามารถพากษ์เสียงได้ตรงกับท่าทางของหุ่นยนต์ที่แสดงออกมา และสัญญาณเสียงจากผู้ใช้ก็จะถูกบันทึกลงไปอีกแชนแนลหนึ่งของเทปบันทึกเสียง และจะสอดคล้องกับท่าทางที่หุ่นยนต์แสดงออกมาในขณะนั้น เมื่อเวลาเล่นกลับจะได้เหมือนกับตอนที่บันทึก เมื่อครบ 1 ชุดก็จะบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรสของข้อมูลชุดต่อไปลงไป แล้วก็นำเอาข้อมูลในแอดเดรสชุดที่สองส่งไปควบคุมเหมือนเดิม เมื่อบันทึกครบทุกแอดเดรส และข้อมูลตัวสุดท้ายของแอดเดรสชุดสุดท้าย ถูกส่งออกไปก็จะบันทึกรหัสจบคือ FFFFH ลงไปเพื่อเป็นการบอกการสิ้นสุดการบันทึก หลังจากนั้นจะเคลียร์ท่าทางของหุ่นยนต์ และโปรแกรมกลับสู่มอนิเตอร์เพื่อรอรับคำสั่งต่อไป

รูปแบบของข้อมูลที่บันทึกลงบนเทปบันทึกเสียงแสดงดังรูปที่ 2.3



รูปที่ 2.3 แสดงรูปแบบของข้อมูลที่บันทึกลงบนเทปบันทึกเสียง

จากรูป start sync จะเป็นสัญญาณสแควร์เวฟความถี่ 1kHz ยาวประมาณ 1 วินาที เพื่อใช้ในการ Synchronization ของสัญญาณ จากนั้นจะเป็น แอดเดรสเริ่มต้น โดยจะบันทึกไบท์สูง ก่อนแล้วตามด้วยไบท์ต่ำ ต่อจาก แอดเดรสเริ่มต้น ก็เป็น แอดเดรสสุดท้าย และ end sync ช่วง delay nSec คือ เวลาที่นำเอาข้อมูลในแอดเดรสที่บันทึกส่งไปควบคุมหุ่นยนต์ให้เคลื่อนไหว หลังจากบันทึกครบทุกชุดแล้วก็จะบันทึกรหัสจบ คือ FFFFH ลงไปแล้วกลับสู่มอนิเตอร์

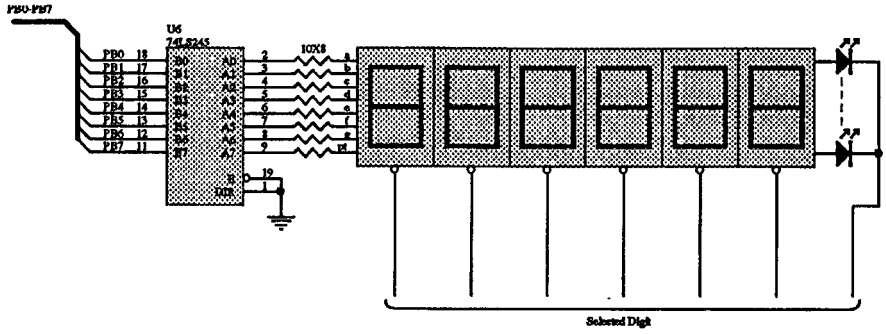
2.3 การออกแบบวงจร

จากการทำงานที่ได้กล่าวมาในข้างต้น ต่อไปนี้จะเป็นการออกแบบวงจรเพื่อให้สามารถทำงานได้ดังที่กล่าวมา โดยจะแบ่งการออกแบบออกเป็น 2 ภาคคือ ภาคแสดงผลและคีย์บอร์ด ส่วนนี้จะเป็นส่วนที่ติดต่อกับผู้ใช้ และที่เหลือคือ ภาคอินพุท/เอาท์พุทและอินเตอร์เฟซ ส่วนนี้จะเป็นส่วนที่ส่งข้อมูลให้ชุดควบคุมโซลีนอยด์วาล์ว และรับส่งข้อมูลกับเครื่องบันทึกเทป

2.3.1 ภาคแสดงผลและคีย์บอร์ด

การแสดงผลจะใช้การมัลติเพล็กซ์ เพื่อให้ง่ายแก่การออกแบบลายวงจร วงจรที่ออกแบบจะเป็นดังวงจร 2.1

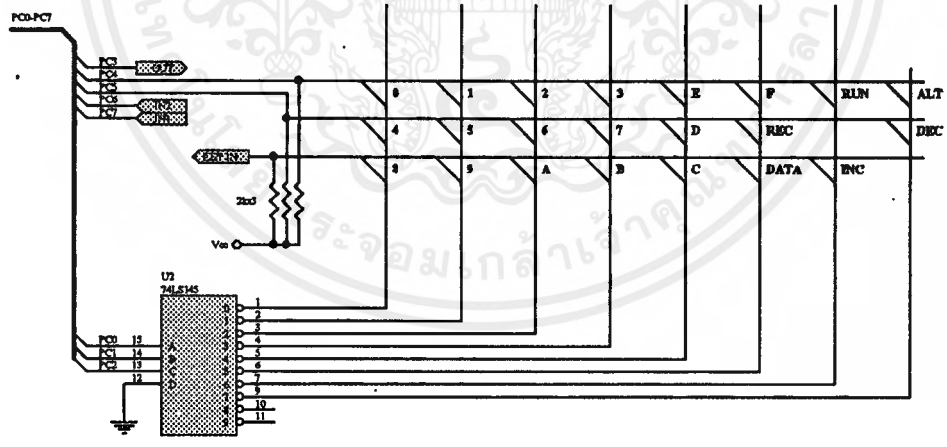
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจร 2.1 วงจรแสดงผล

เนื่องจากกระแสเอาท์พุทของพอร์ต 8255 ไม่เพียงพอที่จะทำให้ 7-Segment แต่ละส่วนสว่างพอควรจึงต้องใช้ IC 74LS245 มาเป็นบัฟเฟอร์เพื่อขับกระแส ถ้าต้องการให้ 7-Segment หลักใดติดสว่างก็ต้องให้โลจิก "0" ที่ขาคอมมอนและส่งข้อมูลควบคุมการติดดับมาทางพอร์ต B เพราะเราต่อขาของ 7-Segment ขนานกันทั้งหมดยกเว้นขาคอมมอน ดังนั้นถ้าต้องการให้ 7-Segment แสดงผลทั้งหมดก็ต้องเริ่มสแกนตั้งแต่หลักแรกจนถึงหลักสุดท้าย โดยส่งข้อมูลของแต่ละหลักมาทางพอร์ต B ก็จะได้การแสดงผลตามต้องการที่กล่าวมาทั้งหมดเป็นหลักการของการแสดงผลแบบมัลติเพล็กซ์ ซึ่งจะต้องใช้โปรแกรมเป็นตัวควบคุมเพราะโปรแกรมมีความเร็วมากเมื่อเทียบกับสายตาของมนุษย์ ดังนั้นเราจึงมองเห็น 7-Segment ติดพร้อมกับทุกหลัก รวมทั้ง LED ทั้ง 8 ดวงด้วย

ในส่วนของคีย์บอร์ดก็จะใช้หลักการคล้ายๆกับการแสดงผลแสดงดังวงจร 2.2



วงจร 2.2 คีย์บอร์ด

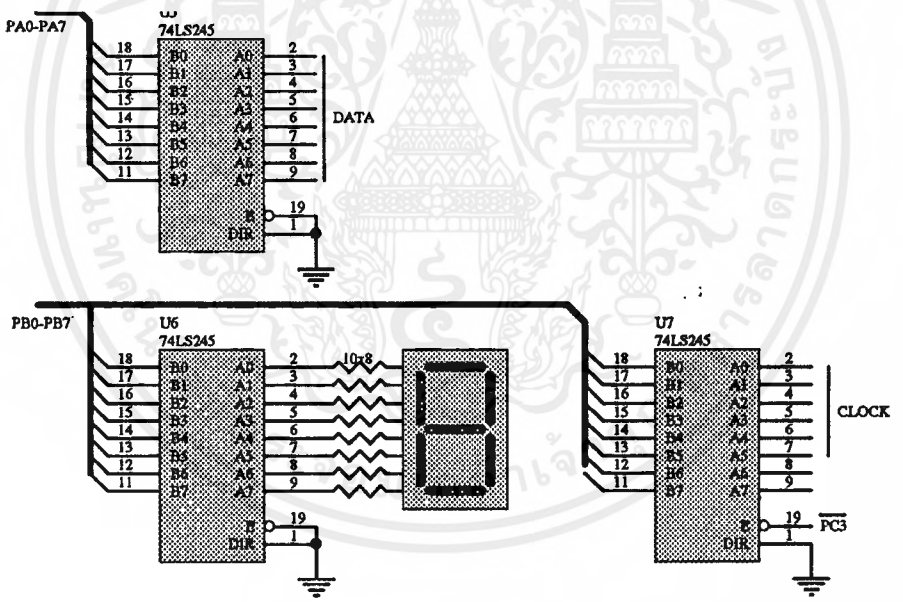
การตรวจสอบว่าคีย์ใดถูกกดจะใช้โปรแกรมคอยตรวจสอบคีย์ทีละหลักโดยจะกำหนดให้พอร์ต C บนเป็นอินพุท และพอร์ต C ล่างเป็นเอาท์พุท PC0-PC2 จะเป็นตัวเลือกหลักที่จะตรวจสอบผ่านทาง IC 74LS145 ซึ่งเป็น Decoder แบบ Open Collector และอ่านค่าบิทของคีย์ที่ถูกกดเข้ามาทาง PC4-PC6 โดยบิทของคีย์ที่ถูกกดจะมีค่าเป็นโลจิก "0" และเมื่อเอาค่าบิทที่อ่านได้นี้ไปรวมกับค่าหลักที่สแกนก็จะได้ค่ารหัสของคีย์ที่กดออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ภาคอินพุท/เอาท์พุทและอินเตอร์เฟซ

ในภาคนี้เราจะต้องใช้พอร์ตถึง 2 ตัวในการส่งข้อมูลให้ชุดควบคุมโซลีนอยด์วาล์ว พอร์ตแรกจะเป็นพอร์ตที่ใช้ส่งข้อมูลควบคุมการติดดับของโซลีนอยด์วาล์ว เราจะใช้พอร์ต A ของ พอร์ต 8255 เพราะเป็นพอร์ตเดียวที่เหลืออยู่ตอนนี้ ส่วนพอร์ตที่สองซึ่งใช้สำหรับสร้าง clock ให้แก่ชุดคบคุมโซลีนอยด์วาล์ว จะมาพิจจารณากันที่พอร์ต B ซึ่งตอนนี้ใช้สำหรับการแสดงผลของ 7-Segment จะสังเกตได้ว่าหากมีการแสดงผลที่ 7-Segment ก็แสดงว่าเรายังไม่พร้อมที่จะส่งข้อมูลให้ชุดควบคุม และในทางกลับกันหากเรากำลังส่งข้อมูลให้ชุดควบคุมเราก็ไม่มีความจำเป็นที่จะดูที่ 7-Segment ดังนั้นเราจึงใช้พอร์ต B เป็นพอร์ตที่ใช้สร้าง clock ให้แก่ชุดควบคุม โดยจะต้องมีตัวควบคุมให้พอร์ต B นี้ทำงานทั้งสอง อย่างได้โดยไม่ผิดพลาด

จากที่กล่าวมาเราสามารถใช้ IC 74LS245 เป็นตัวควบคุมข้อมูลที่พอร์ต B โดยจะต่อเพิ่มเข้ามาอีกหนึ่งตัวต่อขนานกับ A0-A7 ของ IC 74LS245 ตัวแรกที่ใช้ในภาคแสดงผลและใช้ PC3 เป็นตัวควบคุมการทำงาน วงจรที่ออกแบบแสดงดังวงจร 2.3

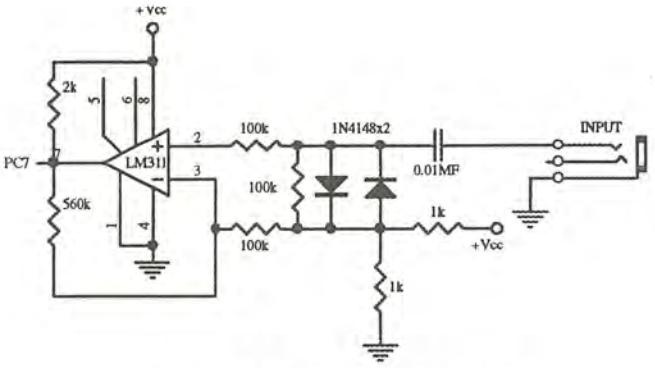


วงจร 2.3 วงจรอินเตอร์เฟซกับชุดควบคุมโซลีนอยด์วาล์ว

จากวงจร 2.3 เมื่อเราต้องการส่งข้อมูลให้ชุดควบคุมโซลีนอยด์วาล์ว เราจะต้องเซ็ทให้ PC3 = 1 จึงจะสามารถทำได้ ข้อมูลที่สร้าง clock ให้แก่ชุดควบคุมจะถูกแสดงออกทาง 7-Segment หรือแถว LED หลักใดหลักหนึ่งด้วยประโยชน์เพื่อเอาไว้ดูสถานะของเครื่องว่ากำลังทำอะไรอยู่

นอกจากจะต้องส่งข้อมูลให้ชุดควบคุมแล้ว เรายังต้องรับส่งข้อมูลกับเครื่องเล่นเทปอีกด้วย พอร์ตอินพุทจะเหลืออยู่บิทเดียวคือ PC7 ดังนั้นเราจึงจะอ่านสัญญาณจากเทปบันทึกเสียงผ่านทาง PC7 นี้โดยสัญญาณดังกล่าวจะต้องผ่านวงจรชมิททริกเกอร์ก่อน ดังวงจร 2.4

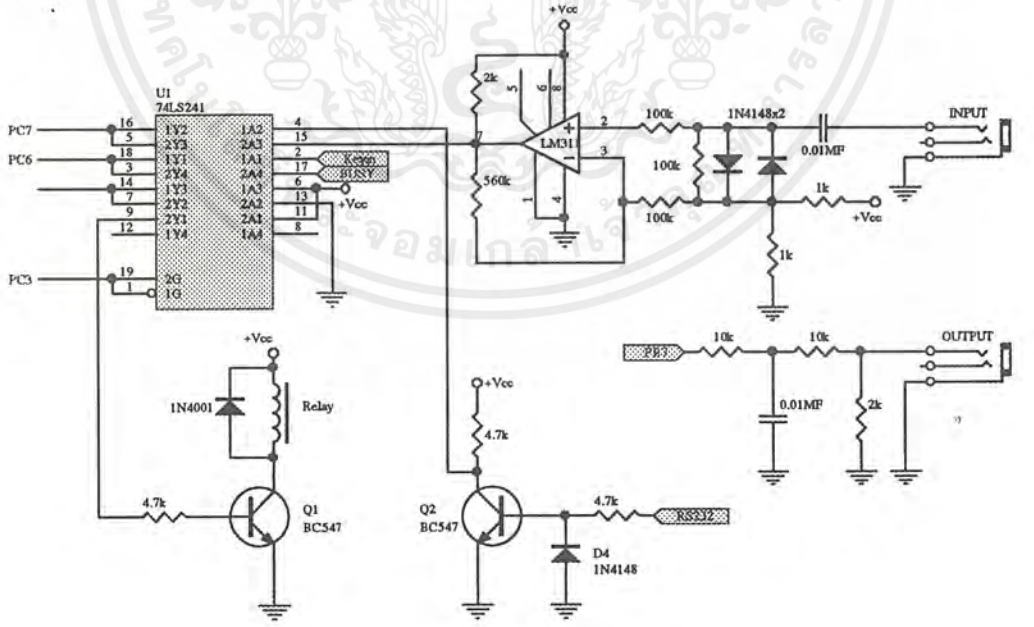
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจร 2.4 วงจรชมิททริกเกอร์

และเพื่อให้เครื่องควบคุมสามารถติดต่อกับอุปกรณ์ภายนอกได้ เช่น เครื่องพิมพ์ เครื่องไมโครคอมพิวเตอร์ เราจึงจะใช้ PC7 และ PC6 มาทำหน้าที่รับอินพุทจากอุปกรณ์ภายนอกดังกล่าว แต่ PC7 และ PC6 ถูกใช้งานหมดแล้วเราจึงต้องใช้ IC 74LS241 มาควบคุมข้อมูลที่จะเข้า PC7 และ PC6 โดยมี PC3 เป็นตัวเลือก

จากวงจรอินเตอร์เฟซเราจะส่งข้อมูลให้ชุดควบคุมโซลีนอยด์วาล์ว เมื่อ PC3=1 และ จากบล็อคไดอะแกรมแสดงการทำงาน จะเห็นได้ว่าในตอนนี้จะมีการอ่านสัญญาณจากเครื่องเล่นเทปเข้ามาด้วย ดังนั้นเราจึงกำหนดให้ PC7 และ PC6 รับข้อมูลจาก เครื่องเล่นเทปและเครื่องพิมพ์ตามลำดับ เมื่อ PC3=1 และเมื่อ PC3=0 PC7 จะรับข้อมูลจาก RS232 ส่วน PC6 จะรับค่าคีย์บอร์ด จากที่กล่าวมาวงจรของระบบอินพุท/เอาท์พุทแสดงดังวงจร 2.5



วงจร 2.5 วงจรระบบอินพุท/เอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรที่ 2.5 และ 2.3 เมื่อ PC3=1 เราจะใช้ PB7 เป็นบิทเอาท์พุทสร้างสัญญาณความถี่ส่งออกไปบันทึกลงเทปบันทึกเสียง ส่วน PB6 จะใช้เป็นสัญญาณ $\overline{STB}$ สำหรับเครื่องพิมพ์วงจรของ Q2 จะมีหน้าที่แปลงระดับแรงดันไฟของระบบ RS232 ให้เหลือ 5 โวลท์ก่อนแล้วจึงส่งให้ PC7 ส่วน PC6 จะใช้อ่านสัญญาณ BUSY จากเครื่องพิมพ์ ตารางที่ 2.1 จะแสดงโหมดของอินพุทละเอาท์พุทที่มี PC3 เป็นตัวควบคุม

Port \ PC3	0	1	Type
PC7	PC (Rx)	Tape (in)	input
PB7	-	Tape (out)	output
PB6	-	Printer($\overline{STB}$)	output
PC6	Keyboard	Printer (Busy)	input
-	Relay (off)	Relay (on)	output

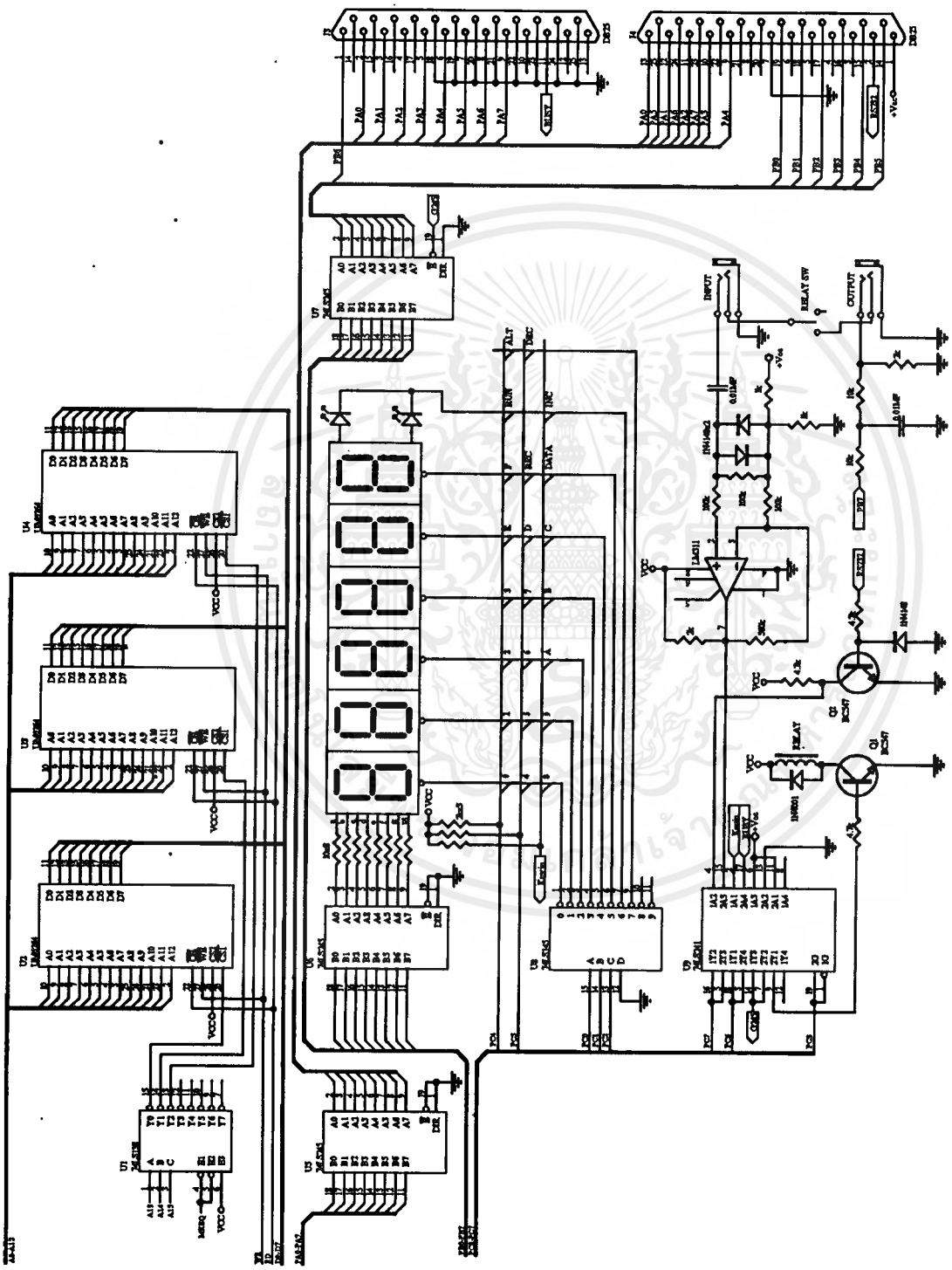
ตารางที่ 2.1 โหมดของอินพุทและเอาท์พุท

2.4 การทำงานของวงจร

วงจรที่ออกแบบมาข้างต้นจะถูกควบคุมการทำงานโดยบอร์ดสำเร็จรูปของบริษัท อีทีที รุ่น CP-Z80V1 ภายในประกอบด้วยไมโครโปรเซสเซอร์ Z80 หน่วยความจำ และพอร์ต 8255 โดยจะต่อ BUS ของ Z80 และ พอร์ตของ 8255 ออกมาทางคอนเนคเตอร์ที่มีไว้ให้วงจรโดยรวมของเครื่องควบคุมแสดงดังวงจร2.6

จากวงจร มีการขยายหน่วยความจำจากเดิมบอร์ด CP-Z80V1 มี 16 kByte ต่อเพิ่มอีก 24 kByte ทำให้มีหน่วยความจำทั้งหมด 40 kByte ในส่วนของการอินเตอร์เฟซกับชุดควบคุมโซลีนอยด์วาล์ว ก็จะใช้ IC 74LS245 คือ U5 และ U7 เป็นตัวอินเตอร์เฟซ U5 ใช้ส่งข้อมูลควบคุมโซลีนอยด์วาล์ว ส่วน U7 จะใช้ PB0-PB5 สำหรับสร้าง clock ให้แก่ชุดควบคุม โดยมี PC3 เป็นตัวควบคุมการทำงาน U6 จะเป็นบัฟเฟอร์รับข้อมูลจากพอร์ต B มาส่งให้กับชุด 7-Segment U8 จะรับข้อมูลจาก PC0-PC2 มาสแกนคีย์บอร์ดและ 7-Segment ค่าคีย์ที่อ่านได้จะเข้าทาง PC4-PC6 U9 เป็นตัวเลือกสัญญาณอินพุทให้ PC7 และ PC6 มี PC3 ควบคุม U10 ทำหน้าที่เป็นวงจรชมิททริกเกอร์รับสัญญาณจากเครื่องเล่นเทปมาส่งให้ U9 เพื่อส่งให้ PC7 ต่อไป วงจร Q2 เป็นวงจรแปลงระดับแรงดันไฟจากระบบ RS232 ให้เป็น 5 โวลท์ Q1 ควบคุมรีเลย์ให้ต่อสัญญาณเสียงจากแจ็คอินพุทไปออกที่แจ็คเอาท์พุท เมื่อ PC3=1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจร 2.6 วงจรของเครื่องควบคุมหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การใช้งานโปรแกรม

ในหัวข้อนี้จะกล่าวถึงการใช้งานโปรแกรมต่างๆที่มีอยู่ในโครงงานนี้ โปรแกรมต่างๆ สามารถเรียกใช้ได้โดยการกดปุ่มที่หน้าปัดของเครื่องได้ทันที

3.1 การใช้งานโปรแกรม DATA

โปรแกรม DATA เป็นโปรแกรมที่ใช้สำหรับป้อนหรือแก้ไขข้อมูลในหน่วยความจำ ณ ตำแหน่ง แอดเดรสต่างๆที่ต้องการ สามารถเรียกใช้โปรแกรม DATA ได้โดยการกดคีย์ DATA หลังจากนั้น ที่ 7-Segment จะแสดงดังรูป 3.1

รูป 3.1

ตัวเลข 4 หลักทางด้านซ้ายของจุดแสดงถึงตำแหน่ง แอดเดรส ของหน่วยความจำที่เราต้องการจะป้อนหรือแก้ไขข้อมูล ส่วนตัวเลข 2 หลักทางขวามือของจุดแสดงถึงข้อมูลที่อยู่ในหน่วยความจำตรง แอดเดรส ที่ระบุทางซ้ายมือ ดังนั้นจากรูปจึงหมายความว่า ที่แอดเดรส 2000H มีข้อมูล 3FH อยู่ การเรียกใช้โปรแกรม DATA โดยการกดปุ่ม DATA โดยที่เราไม่ได้ระบุ แอดเดรสใดๆ เครื่องจะแสดงแอดเดรสล่าสุดที่มีการแก้ไขออกมา แต่ถ้าเราต้องการที่จะเข้าถึงหน่วยความจำในตำแหน่ง แอดเดรสที่ต้องการ ก็สามารถทำได้โดยใส่ค่าแอดเดรส ที่ต้องการก่อนแล้วกดคีย์ DATA เช่นถ้าเราต้องการใส่ค่า AFH ในแอดเดรส 2200H ก็สามารถทำได้ดังรูป 3.2

รูป 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตได้ว่าในการใส่ค่า แอดเดรส เลขศูนย์ 2 ตัวสุดท้ายจะใส่หรือไม่ใส่ก็ได้ เครื่องสามารถที่จะรับรู้เองอัตโนมัติ หลังจากนั้นก็ใส่ค่าที่ต้องการลงไปดังรูป 3.3

รูป 3.3

หลังจากที่เราใส่ค่า AFH ลงไปแล้ว เครื่องควบคุมจะตรวจสอบว่าการเขียนข้อมูลลงหน่วยความจำสำเร็จหรือไม่ ถ้าสำเร็จก็จะแสดงแอดเดรสต่อไปโดยอัตโนมัติ แต่ถ้าไม่สำเร็จเช่น ในกรณีที่เราพยายามที่จะเขียนข้อมูลลงหน่วยความจำในตำแหน่งที่เป็น ROM เครื่องจะไม่แสดงแอดเดรสถัดไป แต่จะแสดงแอดเดรสเดิม

เราสามารถเลื่อนไปดูค่าที่แอดเดรส ต่อไปด้วยการกดคีย์ INC และในทำนองเดียวกันก็สามารถดู ค่าที่แอดเดรสก่อนหน้าได้ด้วยคีย์ DEC ทั้งสองคีย์นี้จะเพิ่มหรือลดค่าแอดเดรสทีละตำแหน่ง นอกจากนี้ยังสามารถกดค้างไว้ได้ซึ่งจะทำให้ แอดเดรสเพิ่มหรือลดอย่างต่อเนื่อง รวมทั้งจะเพิ่มความเร็วในการลดหรือเพิ่มแอดเดรสได้โดยกดคีย์ที่ตรงข้ามควบคู่กันไป เช่นหากกด INC ค้างสามารถเพิ่มความเร็วได้โดยการกดคีย์ DEC และในทำนองเดียวกันหากกด DEC ค้างก็ต้องกด INC

3.1.1 การแก้ไขข้อมูล

มี 2 กรณีคือ การแทรกข้อมูลสามารถทำได้โดยกดคีย์ ALT+INC (กดคีย์ ALT ค้างไว้แล้วกดคีย์INC) และการลบข้อมูลโดยการกดคีย์ ALT+DEC

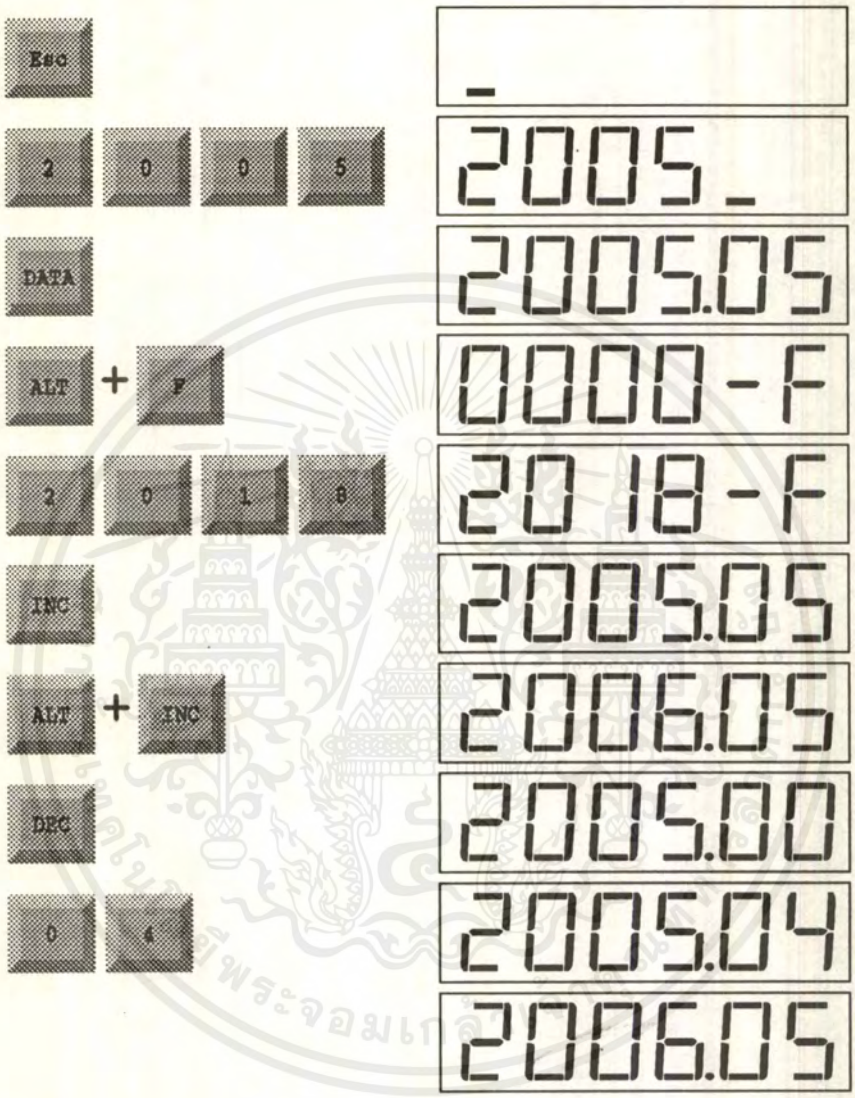
<u>การแทรกข้อมูล</u>

หากเรามีข้อมูลในหน่วยความจำดังนี้

```
Address   Value
  2000    01 02 03 E8 03
  2005    05 06 E8 03 07
  200A    08 09 E8 03 0A
  200F    0B 0C E8 03 0D
  2014    0E 0E 0F E8 03
  2019    AF 3E 00 ED 6F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038355

เราต้องการแทรกข้อมูลเลข 04H ลงไป ณ แอดเดรส 2005H และข้อมูลที่เราต้องการแทรกจะอยู่ระหว่างแอดเดรส 2000H-2018H เท่านั้น ก็สามารถทำได้โดยดังรูป 3.4



รูป 3.4

การกดคีย์ ALT+F เป็นการใส่ค่าแอดเดรสสุดท้ายของข้อมูลที่เราต้องการ ในกรณียกตัวอย่างแอดเดรสสุดท้ายของข้อมูลคือ 2018H หลังจากใส่ค่าแอดเดรสสุดท้ายแล้วก็กดคีย์ INC เครื่องจะกลับมาสู่โหมดการใส่ข้อมูล และการกด ALT+INC 1 ครั้งเป็นการแทรกข้อมูล 1 ไบท์ จะเห็นได้จากจอ 7-Segment จะแสดงให้เห็นว่า ข้อมูล 05H ได้เลื่อนมาอยู่ ณ แอดเดรส 2006H หลังจากทำการแทรกข้อมูลแล้วข้อมูลในหน่วยความจำจะกลายเป็น

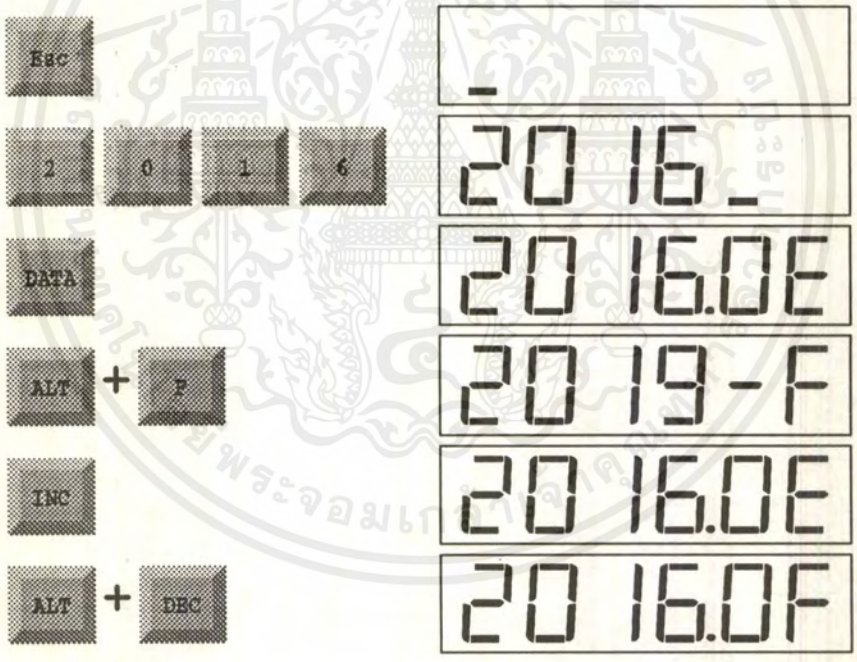
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Address  Value
  2000   01 02 03 E8 03
  2005   04 05 06 E8 03
  200A   07 08 09 E8 03
  200F   0A 0B 0C E8 03
  2014   0D 0E 0E 0F E8
  2019   03 3E 00 ED 6F
```

ถึงตอนนี้ข้อมูลที่เราต้องการจะกลายเป็นตั้งแต่แอดเดรส 2000H-2019H ในตอนนี้หากกดคีย์ ALT+F ก็จะได้ค่า แอดเดรสสุดท้ายเพิ่มเป็น 2019H

<u>การลบข้อมูล</u>

จะกระทำคล้ายๆกับการแทรกข้อมูลคือเข้าไปที่แอดเดรสที่ต้องการลบ ใส่ค่าแอดเดรสสุดท้ายของกลุ่มข้อมูล แล้วค่อยทำการลบ เช่นในกรณีตัวอย่างที่แล้วเราต้องการลบข้อมูลที่แอดเดรส 2016H ซึ่งก็สามารถทำได้ดังรูป 3.5



รูป 3.5

ในกรณีตัวอย่างจะต่อเนื่องจากตัวอย่างการแทรกข้อมูล ดังนั้นเมื่อกดคีย์ ALT+F เพื่อใส่ค่าแอดเดรสสุดท้ายก็จะได้ค่าเป็น 2019H อยู่แล้วจึงกด คีย์ INC กลับเข้าสู่โหมดใส่ข้อมูลเลยไม่ต้องใส่ค่า การกด ALT+DEC 1 ครั้งเป็นการลบค่าในแอดเดรสปัจจุบัน 1 ไบท์ หลังจากทำการลบข้อมูลแล้วข้อมูลในหน่วยความจำจะกลายเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Address      Value
 2000   01 02 03 E8 03
 2005   04 05 06 E8 03
 200A   07 08 09 E8 03
 200F   0A 0B 0C E8 03
 2014   0D 0E 0F E8 03
 2019   00 3E 00 ED 6F
```

และค่าแอดเดรสสุดท้ายจะกลายเป็น 2018H แทน ค่าในแอดเดรส 2019H จะกลายเป็น 00H ซึ่งเราไม่สนใจเพราะอยู่นอกกลุ่มข้อมูลของเรา

จะเห็นได้ว่าไม่ว่าเราจะทำการลบหรือแทรกข้อมูลก็ตามค่าแอดเดรสสุดท้ายของกลุ่มข้อมูลจะเปลี่ยนไปตามการกระทำนั้นๆ ดังนั้นในกรณีตัวอย่างการลบเราไม่ต้องกดคีย์ ALT+F ก็ได้เพราะอย่างไรก็ต้องมีค่าเท่ากับแอดเดรสสุดท้ายที่เราต้องการอยู่แล้ว แต่หากเราทำการแทรกหรือลบข้อมูลในแอดเดรสที่สูงกว่า แอดเดรสสุดท้าย โดยไม่มีการเปลี่ยนแปลงค่าแอดเดรสสุดท้าย เครื่องจะแสดงให้เราใส่ค่าแอดเดรสสุดท้ายเอง เช่น ในกรณีของตัวอย่างที่ยกมาหากเราพยายามที่จะแทรกหรือลบข้อมูลที่แอดเดรส 201DH ผลที่ได้จะแสดงดังรูป 3.6

รูป 3.6

ดังนั้นในการใส่ค่าแอดเดรสสุดท้ายสามารถใส่ได้ในครั้งแรกครั้งเดียว หลังจากนั้นหากต้องการแทรกหรือลบข้อมูลก็ทำได้เลย ไม่ต้องใสค่าแอดเดรสสุดท้ายอีก โดยมีเงื่อนไขว่า แอดเดรสของข้อมูลที่จะทำการแก้ไขจะต้องมีค่าต่ำกว่า ค่าแอดเดรสสุดท้าย

3.2 การใช้งานโปรแกรม REC

โปรแกรม REC เป็นโปรแกรมที่ใช้สำหรับบันทึกแอดเดรสของข้อมูล ที่ใช้ควบคุมการเคลื่อนไหวของหุ่นยนต์ ลงเทปบันทึกเสียง โดยจะบันทึกค่า แอดเดรสเริ่มต้นของกลุ่มข้อมูล (start address) และค่า แอดเดรสสุดท้ายของกลุ่มข้อมูล (final address) และหาก start address เท่ากับ FFFFH เครื่องจะเริ่มทำการบันทึกทันทีที่กดคีย์ INC ตัวอย่างเช่น เรามีกลุ่มของข้อมูลที่จะใช้ในการควบคุมการเคลื่อนไหวของหุ่นยนต์ อยู่ 3 ชุด รายละเอียดคือ

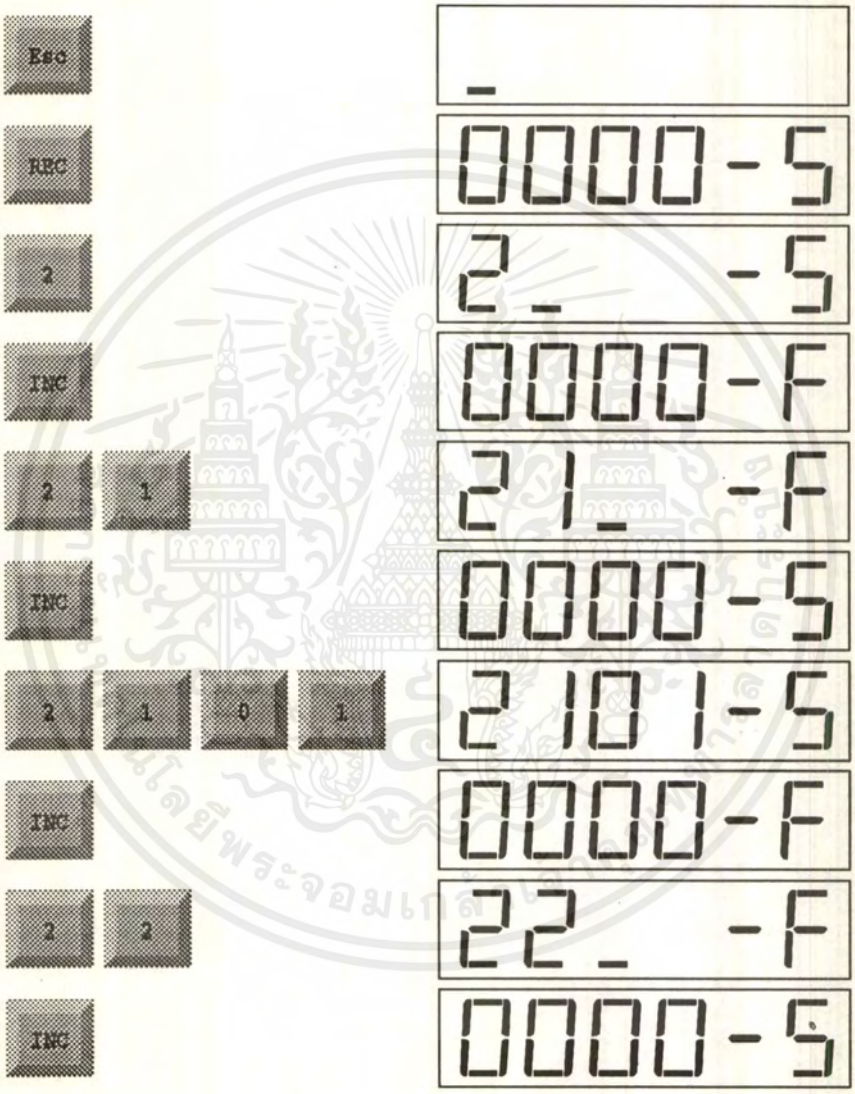
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดที่ 1 อยู่ที่แอดเดรส 2000H ถึง 2100H

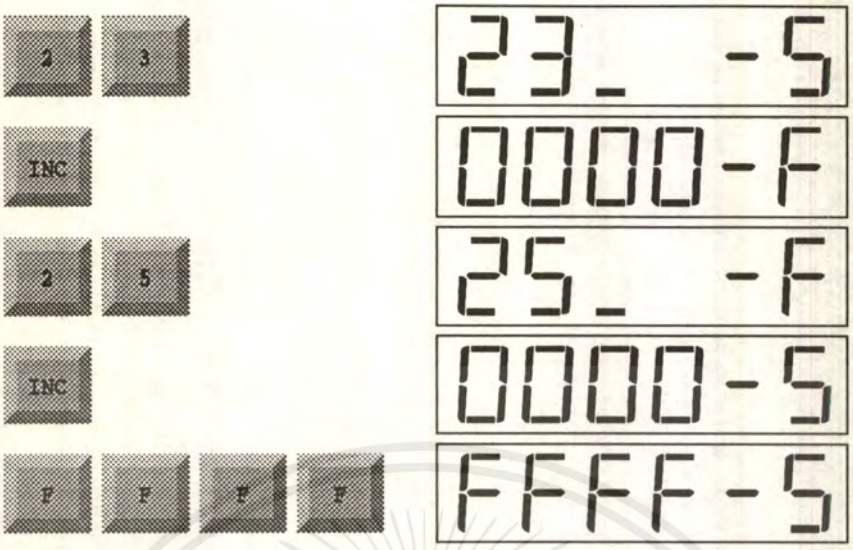
ชุดที่ 2 อยู่ที่แอดเดรส 2101H ถึง 2200H

ชุดที่ 3 อยู่ที่แอดเดรส 2300H ถึง 2500H

จะบันทึกลงเทปสามารถทำได้ดังรูป 3.7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.7

ต่อจากนี้ก็กดปุ่มบันทึกที่เครื่องบันทึกเสียงเทปที่ได้ต่อไว้กับ แจ็คเอาท์พุทของเครื่องแล้วรอสักครู่แล้วกดคีย์ INC เครื่องจะบันทึก start address และ final address ของข้อมูลชุดแรกลงเทปบันทึกเสียง หลังจากนั้นก็จะนำเอาข้อมูลในแอดเดรสดังกล่าวส่งออกไปควบคุมให้หุ่นยนต์เคลื่อนไหว ในขณะนี้ผู้ใช้สามารถ พากย์เสียงตามท่าทางที่ออกแบบไว้ เสียงที่พากย์ก็จะถูกบันทึกลงเทปโดยผ่านทางแจ็คเอาท์อินพุท . ของเครื่องเข้าไปสู่เครื่องบันทึกเทปทางแจ็คเอาท์พุท หลังจากหมดข้อมูลชุดแรก เครื่องจะบันทึก start address และ final address ของข้อมูลชุดที่สองลงไปและนำเอาข้อมูลชุดที่สอง ส่งออกไปควบคุมหุ่นยนต์อีก ทำเช่นนี้จนครบสามชุด ก็จะบันทึกค่า FFFFH ลงไปปิดท้ายเพื่อเป็นการบอกโปรแกรม RUN ว่าสิ้นสุดข้อมูลแล้ว

ในการบันทึกค่า start address และ final address ของข้อมูลแต่ละชุดเครื่องจะบันทึก highbyte ลงเทปก่อนแล้วค่อยบันทึก lowbyte ตามลงไป

3.3 การใช้งานโปรแกรม DESIGN

โปรแกรม DESIGN เป็นโปรแกรมที่ใช้สำหรับออกแบบท่าทางการเคลื่อนไหวของหุ่นยนต์ โดยจะใช้คีย์ตัวเลข 0-F ควบคุมส่วนต่างๆของหุ่นยนต์ 1 คีย์ต่อ 1 ส่วน และจะแสดงเลขฐาน 16 ที่สอดคล้องกับท่าทางที่หุ่นยนต์แสดงออกทาง 7-Segment

การเคลื่อนไหวของหุ่นยนต์จะเป็นไปตามการทำงานของกระบอกสูบ นิวแมติกที่ต่ออยู่กับส่วนต่างๆของหุ่น เมื่อกระบอกสูบทำงานก็จะทำให้ส่วนที่ต่ออยู่เคลื่อนไหว การทำงานของกระบอกสูบจะถูกควบคุมโดยโซลีนอยด์วาล์ว ซึ่งมีหน้าที่ควบคุมการเข้าออกของลมที่กระบอกสูบ ซึ่งโซลีนอยด์วาล์วจะถูกควบคุมโดย T-flipflop ที่ต่ออยู่กับแต่ละบิทของพอร์ตอีกที ดังนั้นเลขฐาน 16 ที่ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้พอร์ตก็จะไปทำให้ T-flipflop ที่ต่ออยู่กับบิทของพอร์ตนั้นทำงาน ซึ่งจะส่งผลให้โซลีนอยด์วาล์ว และกระบอกสูบทำงานทันที

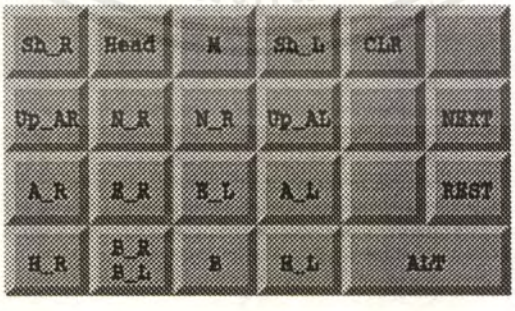
เนื่องจากใช้โซลีนอยด์วาล์ว ถึง 16 ตัวในการควบคุมกระบอกสูบที่ส่วนต่างๆของหุ่นยนต์ ดังนั้นจึงต้องใช้พอร์ตสำหรับข้อมูลถึง 2 พอร์ต จึงจะควบคุมได้ทุกส่วนของหุ่นยนต์

การเรียกใช้โปรแกรม DESIGN ทำได้โดยการกดคีย์ FUNC ตามด้วยคือ DSIGN (คีย์เลข 0) ที่ 7-Segment จะแสดงเลข 3 ไบท์โดยจะมีจุดคั่นระหว่างไบท์ ดังรูป 3.8

รูป 3.8

ไบท์ทางซ้ายสุดจะเป็นค่าที่จะส่งไปให้ พอร์ตที่ 1,พอร์ตที่ 2และ3 ตามลำดับ พอร์ตที่ 3 เป็นพอร์ตขยายมีไว้เผื่อที่จะเพิ่มเติมส่วนของหุ่นยนต์ เลข 3 ไบท์ที่ได้จากโปรแกรม DESIGN จะต้องนำไปใช้รวมกับค่าหน่วงเวลาอีก 2 ไบท์ เพื่อเป็นข้อมูลที่สมบูรณ์ (ค่าหน่วงเวลาต้องทดลองหาเอง) เพราะเมื่อเครื่องส่งข้อมูลไปให้พอร์ตทั้งสามพอร์ตแล้วจะต้องทำการหน่วงเวลาระยะหนึ่งเพื่อให้กระบอกสูบทำงานได้ทัน หลังจากนั้นจึงค่อยส่งข้อมูลชุดต่อไปออกไป เพื่อให้ได้การเคลื่อนไหวที่ต่อเนื่อง

หน้าที่ของคีย์ตัวเลข 0-F ที่ใช้สำหรับควบคุมส่วนต่างๆของหุ่นยนต์และคีย์ที่ใช้ในโปรแกรม DESIGN มีดังนี้คือ



รูป 3.9

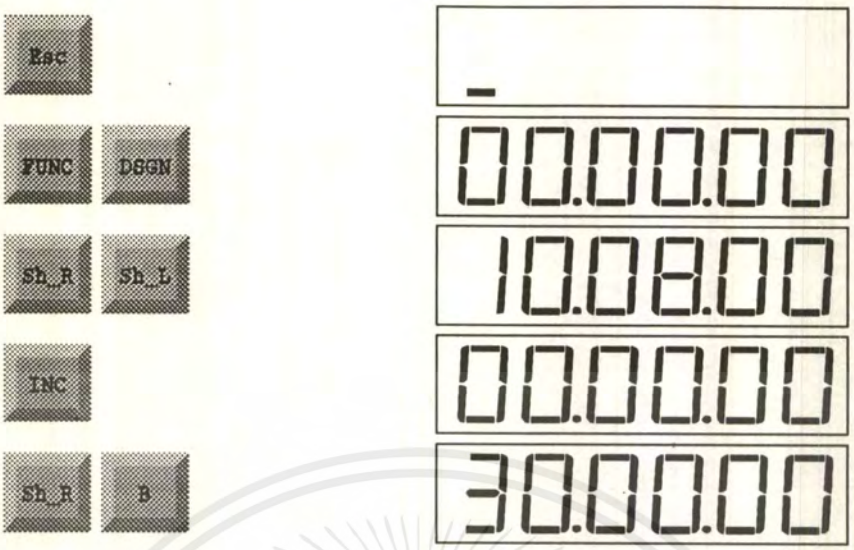
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์	หน้าที่
H_R	ข้อมือข้างขวา (Right Hand)
B_R , B_L	ลำตัวด้านหน้า (Front Body)
B	หมุนลำตัว (Rotate Body)
H_L	ข้อมือข้างซ้าย (Left Hand)
A_R	แขนข้างขวา (Right Arm)
E_R	ข้อศอกข้างขวา (Right Elbow)
E_L	ข้อศอกข้างซ้าย (Left Elbow)
A_L	แขนข้างซ้าย (Left Arm)
Up_AR	แขนท่อนบนข้างขวา (Right Up Arm)
N_R	หันคอขวา (Rotate Neck Right)
N_L	หันคอซ้าย (Rotate Neck Left)
Up_AL	แขนท่อนบนข้างซ้าย (Left Up Arm)
Sh_R	หัวไหล่ข้างขวา (Right Shoulder)
Head	หัว (Head)
M	ปาก (Mouse)
Sh_L	หัวไหล่ข้างซ้าย (Left Shoulder)
CLR	เคลียร์ท่าทางของหุ่นยนต์กลับสู่ท่าเตรียมพร้อมกับเคลียร์ 7-Segment
NEXT	เคลียร์ 7-Segment แต่หุ่นยนต์ยังคงท่าเดิมเอาไว้ ใช้สำหรับออกแบบท่าทางต่อไปของหุ่นยนต์
REST	นำค่าของเลขฐาน 16 ทั้ง 3 ไบท์ และท่าทางของหุ่นก่อนการกดคีย์ INC กลับมา

ตารางที่ 3.1 แสดงหน้าที่ของคีย์ต่างๆที่ใช้ในโปแกรม DESIGN

เช่น ถ้าเราต้องการออกแบบให้หุ่นยนต์ ยกแขนทั้งสองข้างขึ้น ต่อจากนั้นให้ลดแขนขวาลง พร้อมกับหันตัว ก็สามารถทำได้ ดังรูป 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.10

และสมมุติให้ค่าหน่วงเวลาเท่ากับ 1 วินาที = 1000 mSec = 03E8H ก็จะได้ข้อมูลควบคุมในหน่วยความจำ คือ

```
2000   10 08 00 E8 03
2005   30 00 00 E8 03
```

การใส่ค่าหน่วงเวลาจะต้องใส่สลับค่ากันระหว่าง highbyte กับ lowbyte ด้วยในกรณีที่มีการเพิ่มเติมส่วนของหุ่นยนต์ สามารถควบคุมส่วนที่เพิ่มเติมได้โดยใช้คีย์เลข 0-7 ร่วมกับคีย์ ALT ดังแสดงในตารางที่ 3.2

ALT	Port
0	P30
1	P31
2	P32
3	P33
4	P34
5	P35
6	P36
7	P37

ตารางที่ 3.2 หน้าที่ของคีย์ที่ใช้ควบคุมส่วนเพิ่มเติมของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

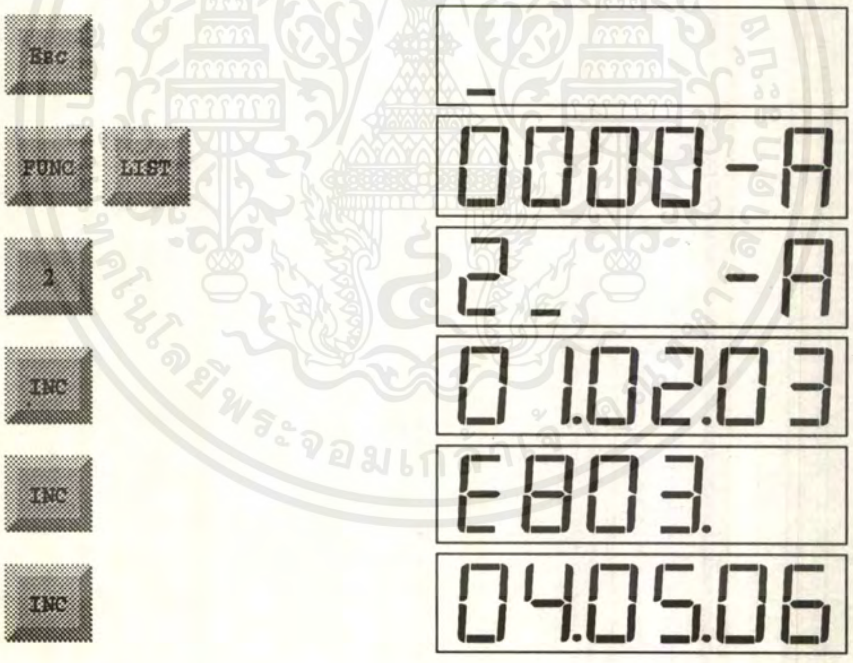
3.4 การใช้งานโปรแกรม LIST

โปรแกรม LIST เป็นโปรแกรมที่ใช้สำหรับดูข้อมูลในหน่วยความจำเป็นกลุ่มๆแตกต่างจากโปรแกรม DATA ตรงที่ โปรแกรม LIST จะไม่แสดงค่าแอดเดรส

สมมุติว่าเรามีข้อมูลสำหรับควบคุมหุ่นยนต์ ในหน่วยความจำ คือ

```
2000   01 02 03 E8 03
2005   04 05 06 E8 03
200A   07 08 09 E8 03
200F   0A 0B 0C E8 03
```

หากเราใช้โปรแกรม DATA ดูข้อมูลดังกล่าวอาจจะดูยาก หากใช้โปรแกรม LIST ดูก็จะสามารถดูได้ง่ายกว่าเพราะโปรแกรม LIST จะแสดงเลข 2 ชุดคือ ชุดแรก จะแสดง3 ไบท์เป็นค่าที่ใช้สำหรับส่งให้พอร์ต และ ชุดที่สองเป็นค่า หน่วงเวลา 2 ไบท์ การใช้งานโปรแกรม LIST ทำได้โดยการกดคีย์ FUNC , LIST (คีย์เลข 3) เช่นเราต้องการดูข้อมูลที่ยกมาข้างบนโดยโปแกรม LIST ทำได้ดังรูป 3.11

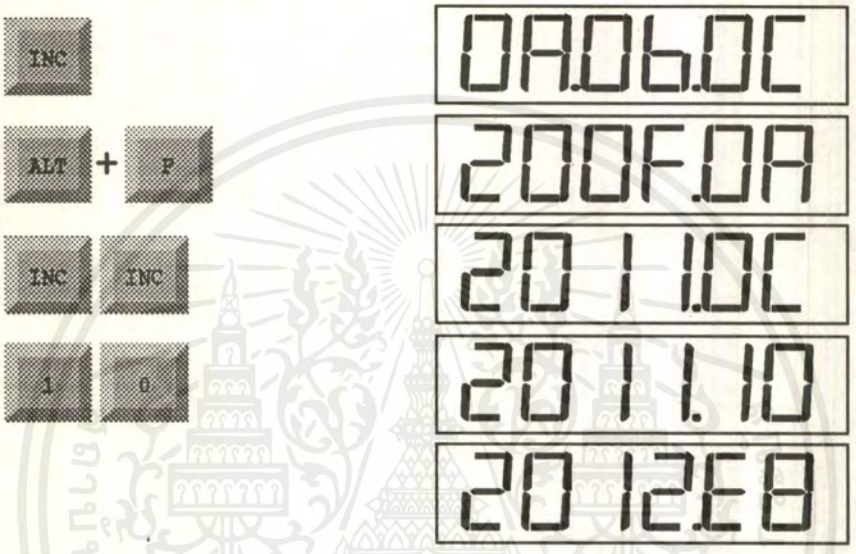


รูป 3.11

ในตอนแรกโปรแกรมจะถาม ค่าแอดเดรสที่ต้องการจะดู ในกรณีนี้คือ 2000H หลังจากนั้นจะแสดงเลขกลุ่มแรก และเมื่อกด INC จะแสดงชุดที่สอง จะแสดงสลับกันไปเรื่อยๆ การใช้งานปุ่ม INC และ DEC เหมือนกับโปรแกรม DATA คือสามารถกดค้างได้เพื่อการดูที่ต่อเนื่องและเพิ่มความเร็วได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเราต้องการแก้ไขข้อมูลในขณะที่อยู่ในโปรแกรม LIST ก็สามารถทำได้โดยกดคีย์ ALT+F โปรแกรมจะไปเรียกใช้งานโปรแกรม DATA ขึ้นมาทำงาน โดยจะแสดงค่าแอดเดรสของข้อมูลที่อยู่ทางด้านซ้ายสุดของหน้าจอ 7-Segment เช่น จากข้อมูลที่ยกมาข้างต้นหากเราต้องการแก้ไขข้อมูลในตำแหน่งแอดเดรส 2011H จาก 00H เป็น 10H ขณะที่อยู่ในโปรแกรม LIST ก็สามารถทำได้ดังรูป 3.12



รูป 3.12

หลังจากนั้นก็เรียกเข้าโปรแกรม LIST ใหม่เพื่อดูข้อมูลต่อไป โปรแกรมจะแสดงแอดเดรสล่าสุดก่อนที่จะไปเรียกใช้โปรแกรม DATA ออกมา

3.5 การใช้งานโปรแกรม SAVE

โปรแกรม SAVE เป็นโปรแกรมบันทึกข้อมูลในหน่วยความจำ ลงบนเทปบันทึกเสียงเพื่อเก็บเอาไว้ใช้ในครั้งต่อไป เหมือนกับการบันทึกข้อมูลลงแผ่น ดิสก์ ของเครื่องคอมพิวเตอร์ แต่ในที่นี้เป็นการบันทึกลงเทปบันทึกเสียงแทน สามารถเรียกใช้โปรแกรมได้โดยการกดคีย์ FUNC , SAVE (คีย์เลข 1) เช่นเราต้องการบันทึกข้อมูลในหน่วยความจำตำแหน่ง 2000H-2100H ลงเทปทำได้ดังรูป 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.13

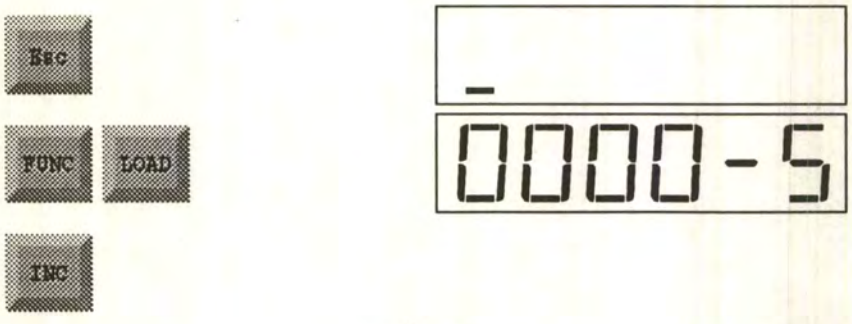
โปรแกรมจะถาม start address และ final address หลังจากใส่ค่า final address แล้วกดคีย์ INC หลังจากบันทึกเสร็จก็จะกลับเข้าสู่ มอนิเตอร์ อีกครั้ง

รูปแบบของข้อมูลที่บันทึกลงบนเทปบันทึกเสียงโดยโปรแกรม SAVE เป็นดังรูป 3.14

รูป 3.14

3.6 การใช้งานโปแกรม LOAD

โปรแกม LOAD เป็นโปรแกรมโหลดข้อมูลจากเทปบันทึกเสียงลงหน่วยความจำ โดยข้อมูลดังกล่าวถูกบันทึกโดยโปรแกรม SAVE การเรียกใช้งาน คือ กดคีย์ FUNC, LOAD (คีย์เลข 2)



รูป 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

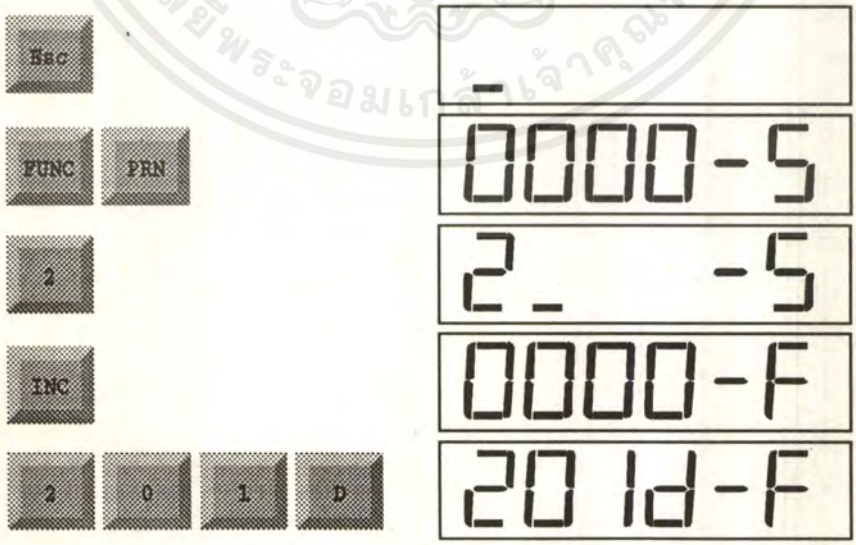
โปรแกรมจะถามว่าข้อมูลที่โหลดมาได้จะนำไปไว้ที่แอดเดรสใด ถ้ากดคีย์ INC ผ่านเลยไป โปรแกรมจะใช้ค่าแอดเดรสที่โหลดมาได้จากเทป โปรแกรมจะแสดงข้อมูลที่โหลดมาได้ที่แถวของ LED ข้างล่าง 7-Segment ในขณะที่กำลังโหลดด้วย หลังจากโหลดเสร็จก็จะกลับเข้ามอนิเตอร์เพื่อรอรับคำสั่งต่อไป

3.7 การใช้งานโปรแกรม RUN

โปรแกรม RUN เป็นโปรแกรมที่ใช้สำหรับโหลดค่าแอดเดรสจากเทปบันทึกเสียง ที่บันทึกโดยโปรแกรม REC แล้วนำเอาข้อมูลในแอดเดรสที่อ่านได้ส่งออกไปควบคุมให้หุ่นยนต์เกิดการเคลื่อนไหว พร้อมกับเปิดทางให้สัญญาณเสียงอีกช่องสัญญาณของเทปผ่านแจ็คเอาท์พุทออกไปขยายเสียง สามารถเรียกใช้โปรแกรมได้โดยการกดคีย์ RUN โปรแกรมจะแสดงค่าแอดเดรสที่อ่านได้ออกทางแถว LED ข้างใต้ 7-Segment ถ้าค่า start address ที่มาได้มีค่าเท่ากับ FFFFH ซึ่งเป็นรหัสจบก็จะกลับเข้าสู่มอนิเตอร์ทันที เพื่อรอรับคำสั่งต่อไป

3.8 การใช้งานโปรแกรม PRINT

โปรแกรม PRINT เป็นโปรแกรมที่ใช้พิมพ์ข้อมูลที่อยู่ในหน่วยความจำออกทางเครื่องพิมพ์ โดยจะแสดงบรรทัดละ 5 ไบท์ 3 ไบท์แรกเป็นข้อมูลที่ส่งให้พอร์ต 2 ไบท์หลังเป็นค่าหน่วงเวลา การเรียกใช้งานโปรแกรมก็กดคีย์ FUNC,PRN (คีย์เลข 4) โปรแกรมจะถามค่าแอดเดรสเริ่มต้นและแอดเดรสสุดท้ายของกลุ่มข้อมูลที่เราต้องการพิมพ์ออกทางเครื่องพิมพ์ เมื่อป้อนเสร็จก็จะสั่งพิมพ์ เช่น หากเราต้องการพิมพ์ข้อมูลตั้งแต่แอดเดรส 2000H ถึง 201DH ออกทางเครื่องพิมพ์สามารถทำได้ดังรูป 3.16



รูป 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 การใช้งานโปรแกรม PCLINK

โปรแกรม PCLINK เป็นโปรแกรมที่ใช้สำหรับรับข้อมูลเลขฐาน 16 ที่ส่งมาทางพอร์ตอนุกรมของเครื่องไมโครคอมพิวเตอร์ จะใช้รูปแบบของข้อมูลที่ส่งออกมาเป็นแบบ INTEL HEX FORMAT ซึ่งมีส่วนขยายของไฟล์เป็น .HEX การส่งข้อมูลจะใช้อัตราบอดเรท (Buad Rate) เท่ากับ 9600 บิทต่อวินาที ดังนั้นก่อนที่จะใช้งานโปรแกรมควรตั้งโหมดของพอร์ตอนุกรม ที่เครื่องไมโครคอมพิวเตอร์ ให้มีอัตราบอดเรทเท่ากับ 9600 บิทต่อวินาทีก่อนโดยใช้คำสั่ง

MODE COM2:9600,n,8,1,p

หลังจากนั้น เรียกใช้งานโปรแกรม PCLINK โดยกดคีย์ FUNC,PCL (คีย์เลข 5) โปรแกรมจะถามค่า offset แอดเดรส (โปรแกรมจะนำเอาค่านี้ไปบวกกับแอดเดรสเริ่มต้นของข้อมูลที่ส่งมาทางพอร์ตอนุกรม) หลังจากนั้นก็กดคีย์ INC โปรแกรมจะรอรับข้อมูลที่ส่งเข้ามาและจะแสดงข้อมูลที่อ่านได้ออกทางแถว LED ด้วย ดังแสดงในรูป 3.17

รูป 3.17

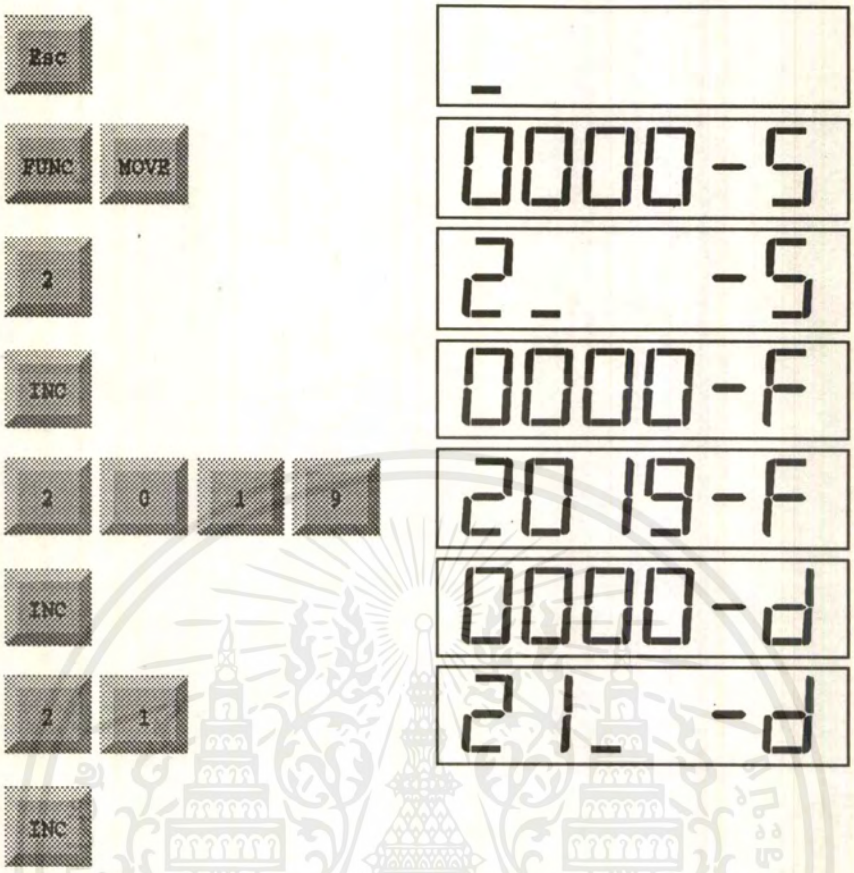
การส่งข้อมูลจากเครื่องไมโครคอมพิวเตอร์สามารถทำได้โดยพิมพ์คำสั่งต่อไปนี้ที่ บรรทัดคำสั่ง

COPY FILES.HEX COM2

3.10 การใช้งานโปรแกรม MOVE

โปรแกรม MOVE เป็นโปรแกรมที่ใช้คัดลอกข้อมูลภายในแอดเดรสที่กำหนดไปยังแอดเดรสปลายทาง การเรียกใช้งานโปรแกรม MOVE ทำได้คือ กดคีย์ FUNC,MOVE (คีย์เลข 6) หลังจากนั้นโปรแกรมจะถามค่า แอดเดรสเริ่มต้น แอดเดรสสุดท้ายของข้อมูล และแอดเดรสปลายทาง เช่นเราต้องการคัดลอกข้อมูลในแอดเดรส 2000H-2019H ไปที่แอดเดรส 2100H จะต้องทำดังรูป 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
บทสรุป

4.1 การใช้งานเครื่อง

4.1.1 การบันทึก

1. ออกแบบท่าทางของหุ่นยนต์ โดยใช้โปรแกรม DESIGN จดข้อมูลทั้ง 3 ไบท์ที่จอ 7-Segment เอาไว้

2. เมื่อออกแบบท่าทางการเคลื่อนไหวเสร็จก็นำเอาข้อมูลที่จดได้มาใส่ในหน่วยความจำโดยจะต้องใส่ค่าหน่วงเวลาในแต่ละ step ลงไปด้วยในตอนแรกใส่ค่าเท่าไรก็ได้ลงไปก่อน (แนะนำให้ใส่ค่า 03E8H)

3. ทดลองส่งข้อมูลให้หุ่นเคลื่อนไหวตามที่ได้ออกแบบเพื่อหาค่าหน่วงเวลาที่เหมาะสมโดยใช้โปรแกรม REC แต่ยังไม่ต้องกดปุ่มบันทึกที่เครื่องบันทึกเทป แก้ไขค่าหน่วงเวลาให้ได้การเคลื่อนไหวที่ต่อเนื่องและสวยงาม

4. ต่อสายสัญญาณจากแจ็คเอาท์พุทของเครื่องไปเข้าที่เครื่องบันทึกเสียง เพื่อบันทึกข้อมูลที่ได้ออกแบบในข้อ 1. ลงเทปบันทึกเสียงโดยใช้โปรแกรม SAVE ในการบันทึก โปรแกรมจะบันทึกข้อมูลลงเทปแค่แชนแนลเดียว จึงแนะนำว่าควรบันทึกสัญญาณเสียงอาจเป็นเพลงหรืออะไรก็ได้ลงไปบนแชนแนลที่เหลือพร้อมกัน ประโยชน์ก็เพื่อให้ทราบว่าตอนนี้ข้อมูลที่เราบันทึกอยู่ตรงไหนเวลาเราย้อนกลับมาจะได้ไม่งง

5. ใช้โปรแกรม REC อีกครั้งเพื่อบันทึกแอดเดรสของข้อมูลและสัญญาณเสียงที่สอดคล้องลงเทป โดยทำเหมือนกับข้อ 3. แต่ตอนนี้เราจะกดปุ่มบันทึกที่เครื่องบันทึกเทปด้วย ในขั้นตอนนี้ก็คือขั้นตอนการพากย์เสียงให้ตรงกับท่าทางที่หุ่นแสดงออกมานั่นเอง (การบันทึกในขั้นตอนนี้ควรเว้นระยะให้ห่างจากข้อ 4 พอสมควรเพื่อความสะดวกเวลาเราเล่นกลับ)

หากเรามีข้อมูลที่ใช้ควบคุมการเคลื่อนไหวหลายชุดและต้องการบันทึกให้แต่ละชุดต่อเนื่องกัน เราก็ต้องบันทึกข้อมูลดังกล่าวลงเทปก่อนดังข้อ 4 หลังจากนั้นก็ทำตามข้อ 5 โดยใส่ค่าแอดเดรสของแต่ละชุดเรียงกันไปตามที่เราต้องการ

4.1.2 การเล่นกลับ

1. โหลดข้อมูลควบคุมการเคลื่อนไหวจากเทปบันทึกเสียงที่บันทึกในข้อ 4. ลงหน่วยความจำ โดยใช้โปรแกรม LOAD (ในกรณีที่เครื่องมีข้อมูลอยู่แล้วก็ไม่ต้องโหลด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรอเทปมาตรงตำแหน่งก่อนที่จะบันทึกข้อ 5. เล็กน้อย เสียบสัญญาณจากเครื่องเล่นเทปเข้าที่แจ็คอินพุทของเครื่องควบคุม โดยเอามาแต่แชนแนลที่มีแอดเดรสของข้อมูล ส่วนสัญญาณเสียงที่อยู่อีกแชนแนลส่งออกไปขยายเสียงออกทางลำโพง กดปุ่ม RUN แล้วกดปุ่ม play ที่เครื่องเล่นเทป

4.2 อุปสรรคและปัญหา

จากการทดลองควบคุมให้หุ่นยนต์เคลื่อนไหวประกอบกับเสียง พบว่าสามารถทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ อย่างไรก็ตามพบว่าเกิดปัญหาขึ้นบ้างคือ

4.2.1 ปัญหาจากเครื่องเล่นเทป

พบว่าถ้าหากใช้เครื่องเล่นและบันทึกเทปคนละเครื่องกันอาจเกิดปัญหาขึ้นบ้าง เนื่องจากความเร็วของเส้นเทปแต่ละเครื่องต่างกัน ทำให้สัญญาณที่ได้ออกมา มีความถี่ไม่ตรงกับค่าที่ต้องการคือ 1kHz และ 2kHz เพราะจะใช้ความถี่เหล่านี้แทนสถานะของโลจิกในการบันทึกข้อมูลดิจิตอลลงบนเทปบันทึกเสียง (ดูรายละเอียดลักษณะของสัญญาณในหัวข้อแนวความคิด) แต่ปัญหานี้พบน้อยมากเพราะส่วนใหญ่เครื่องเล่นเทปจะมีความเร็วที่ได้มาตรฐานอยู่แล้ว อาจจะผิดพลาดบ้างแต่ก็ยังสามารถใช้งานได้ นอกจากปัญหาความเร็วของเส้นเทปแล้ว ตำแหน่งของหัวเทปก็เป็นสิ่งที่ก่อให้เกิดปัญหาแบบเดียวกัน

4.2.2 ความแรงของสัญญาณ

สัญญาณเสียงที่ป้อนให้เครื่องควบคุมเพื่อใช้อ่านแอดเดรส จะต้องมีความแรงพอสมควรที่จะทำให้วงจรชมิททริกเกอร์ สามารถตัดสินใจสถานะของระดับสัญญาณได้ ถ้าระดับของสัญญาณแรงไม่พอ การอ่านข้อมูลเกิดความผิดพลาดขึ้น เครื่องจะแสดงรหัสความผิดพลาดออกมาที่ 7-Segment สามารถดูความหมายของความผิดพลาดได้ในภาคผนวก ค.

4.2.3 ปัญหาทั่วไป

ปัญหาที่เกิดขึ้นในลักษณะนี้จะไม่กีดขวางการทำงานของเครื่องแต่จะทำให้เกิดความล่าช้าบ้าง เช่น เครื่องไม่มีระบบสำรองข้อมูล ดังนั้นทุกครั้งที่เริ่มต้นการทำงานจะต้องป้อนข้อมูลใหม่ทั้งหมดอีกอย่างคือในตอนเริ่มต้น การทำงานของกระบอกสูบจะยังไม่สามารถทำงานได้ทันที อาจจะต้องรอสักครู่โดยเฉพาะกระบอกสูบทางข้อศอกข้างขวาต้องใช้แรงดันลมพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

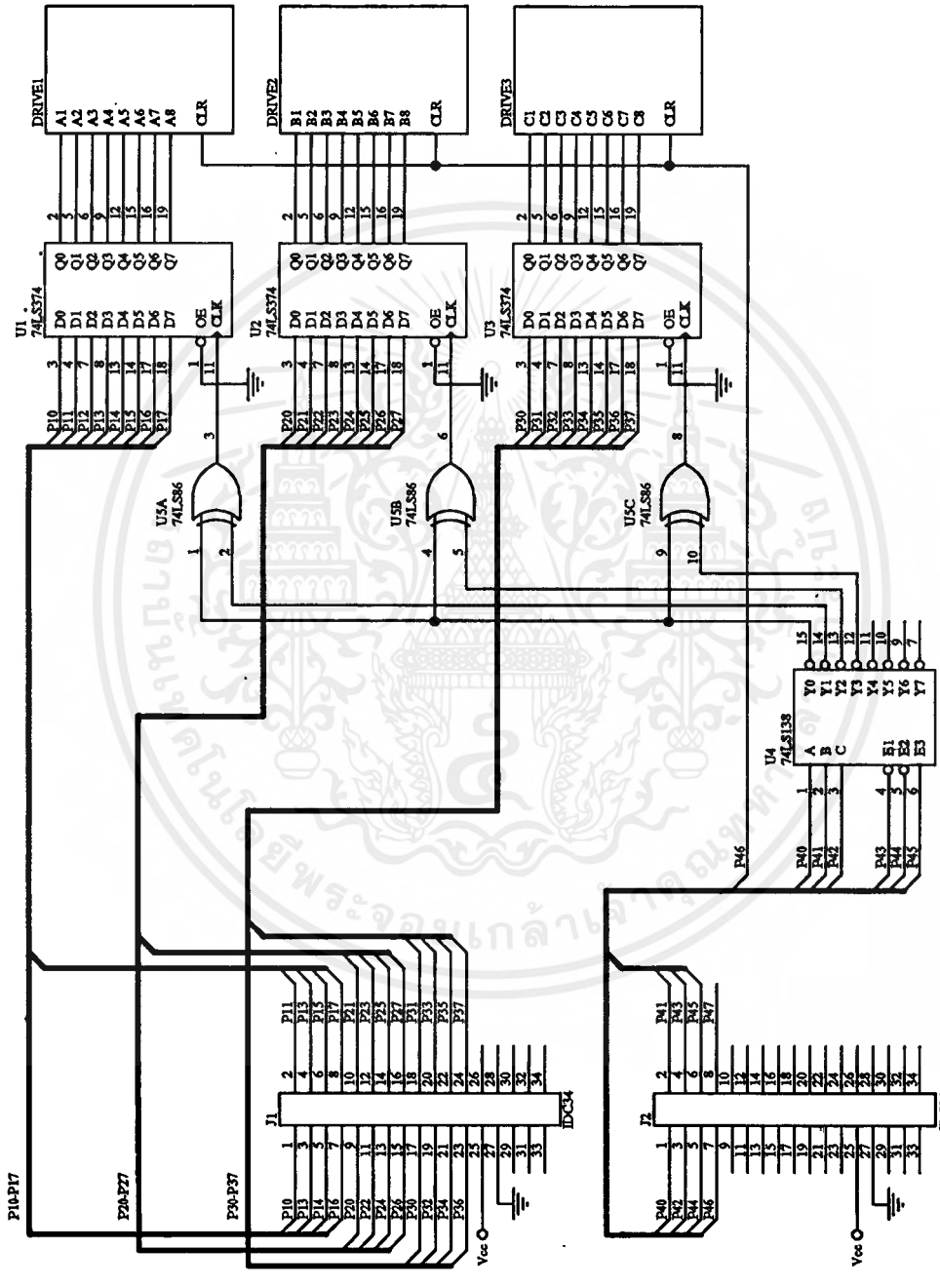
หนังสืออ้างอิง

1. บริษัท อีทีที จำกัด , คู่มือการใช้งานอีทีบอร์ดรุ่น 3.0 , 2532
2. บริษัท อีทีที จำกัด , คู่มือการใช้งานชุดทดลอง Micro Training ET-01 , 2532
3. รศ.ยืน ภู่วรวรรณ และ วัฒนา เชียงกูล , ไมโครโปรเซสเซอร์ ไมโครคอมพิวเตอร์ Z80 microprocessor , 2538
4. วารสาร เซมิคอนดัคเตอร์อิเลคทรอนิคส์ เล่มที่ 113 มกราคม 2535
5. ปริญญานิพนธ์ "หุ่นยนต์ควบคุมโดย PC" , 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

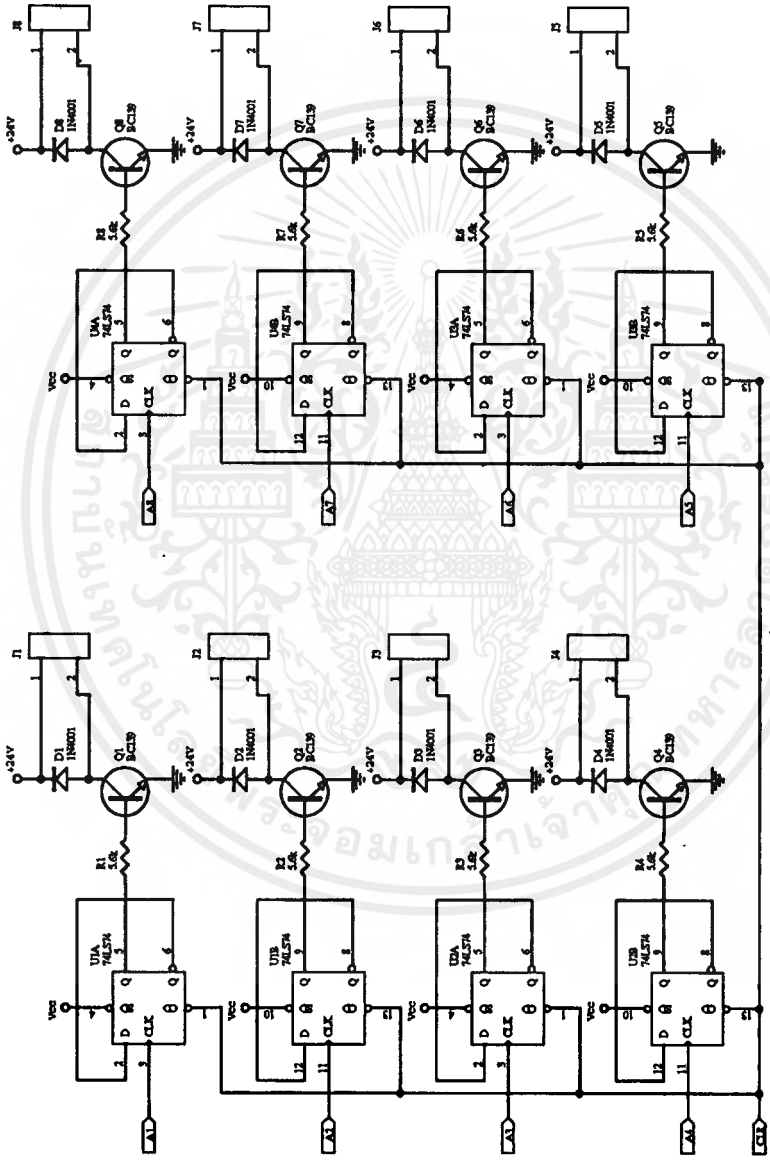
ภาคผนวก ก.
รายละเอียดทางฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



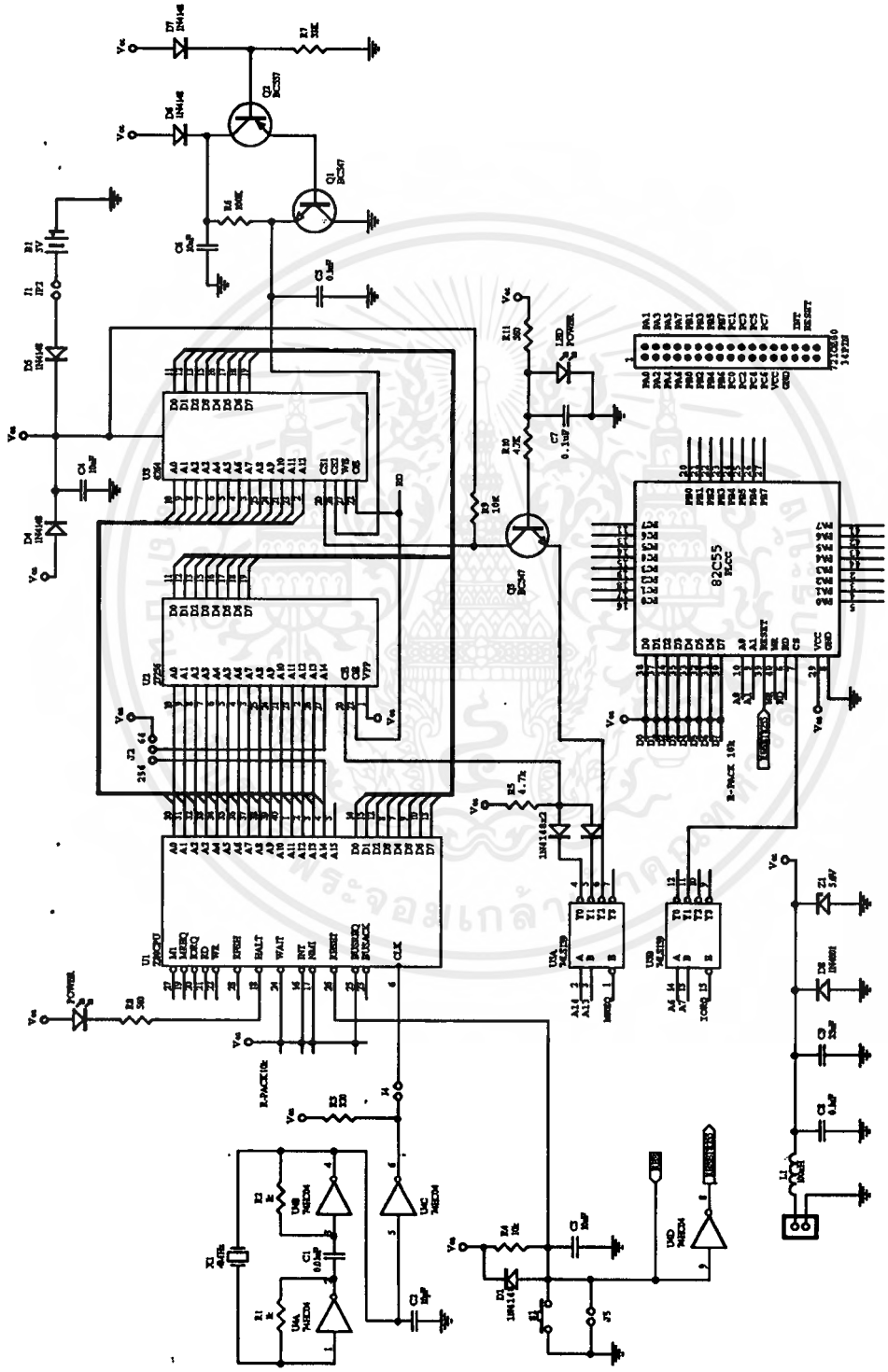
รูปแสดงวงจรชุดควบคุมไซลินเดอร์วาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



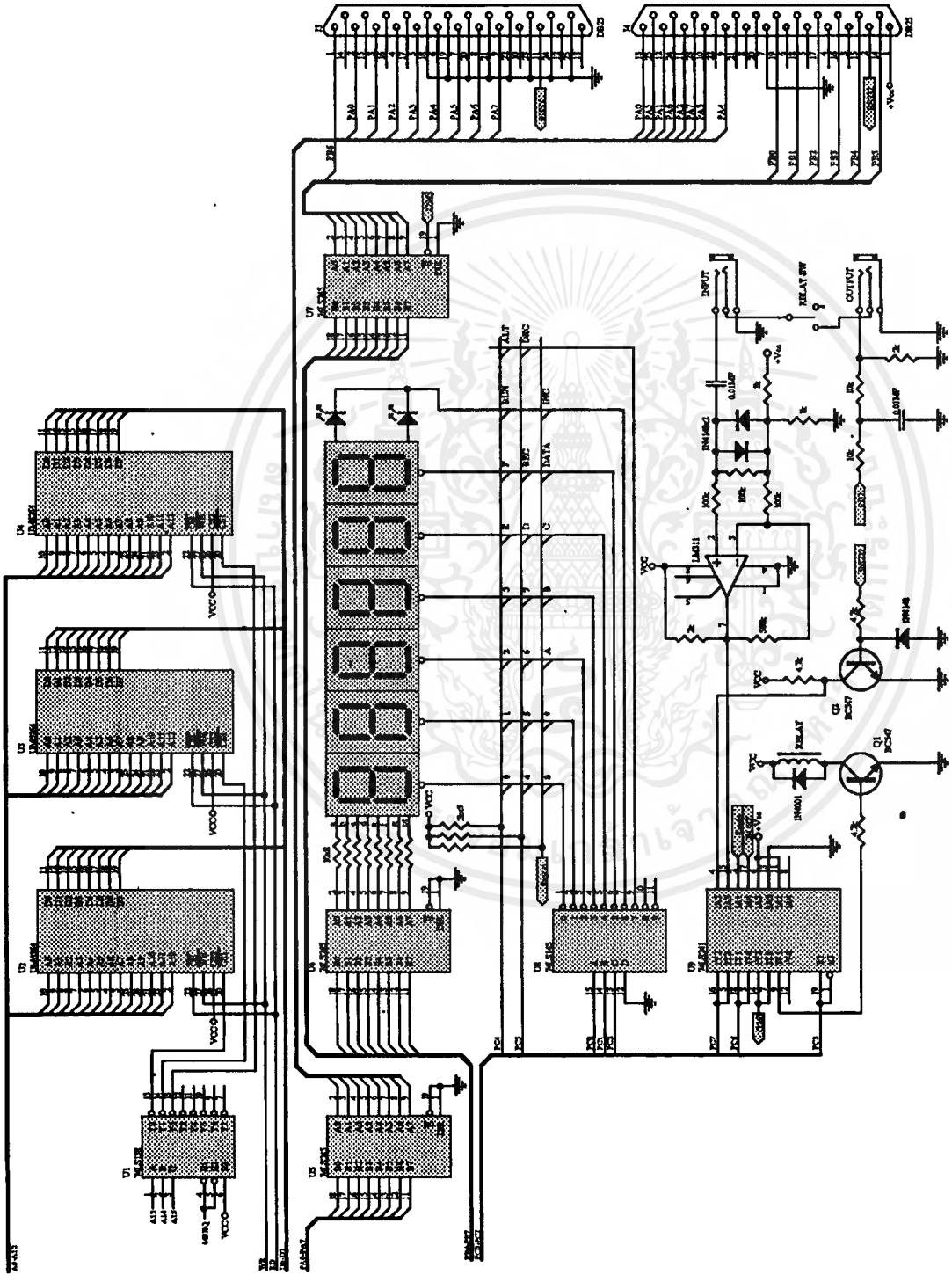
รูปแสดงวงจรขับไซลินอยด์วาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



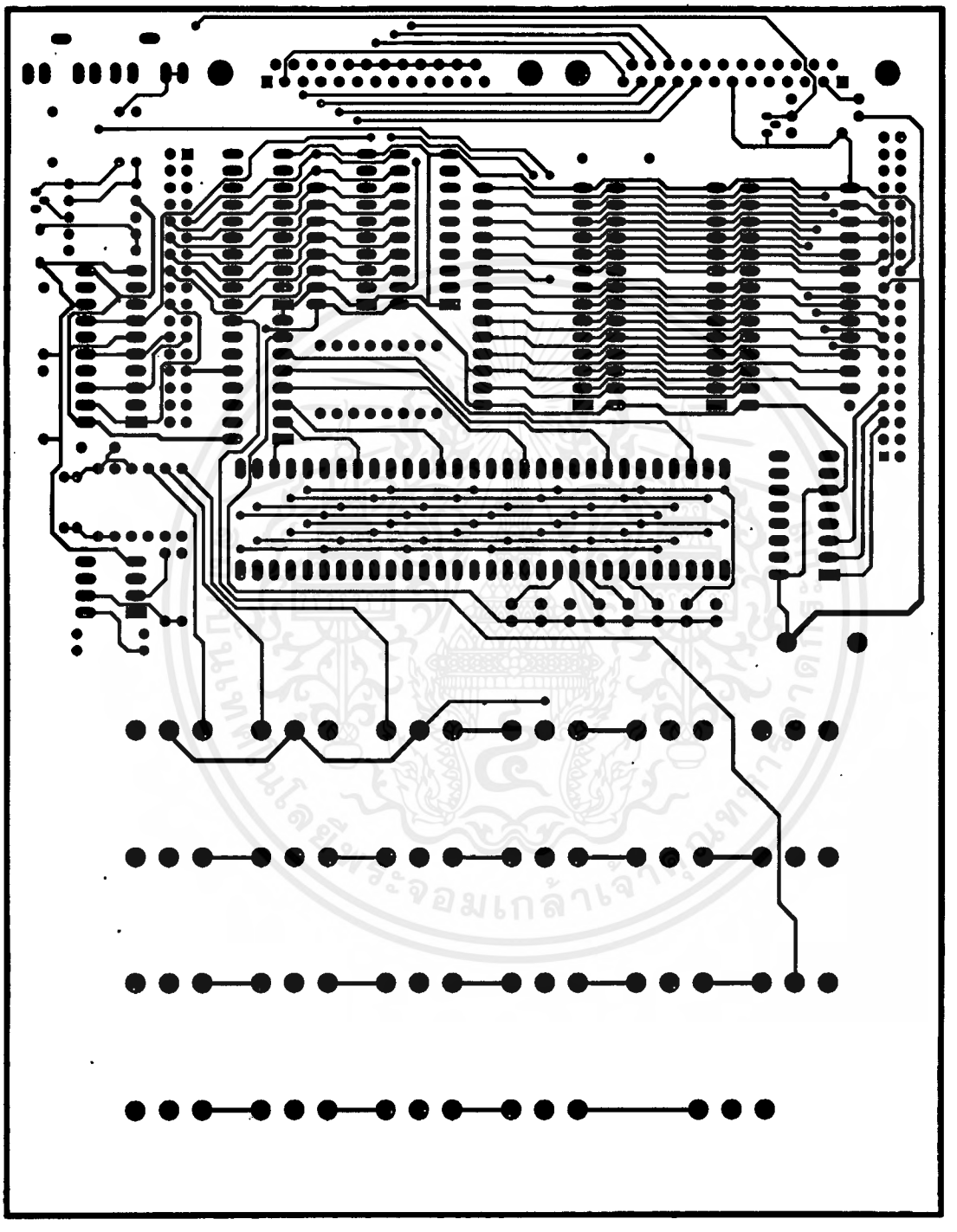
รูปแสดงวงจรของบอร์ด CP-Z80V1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



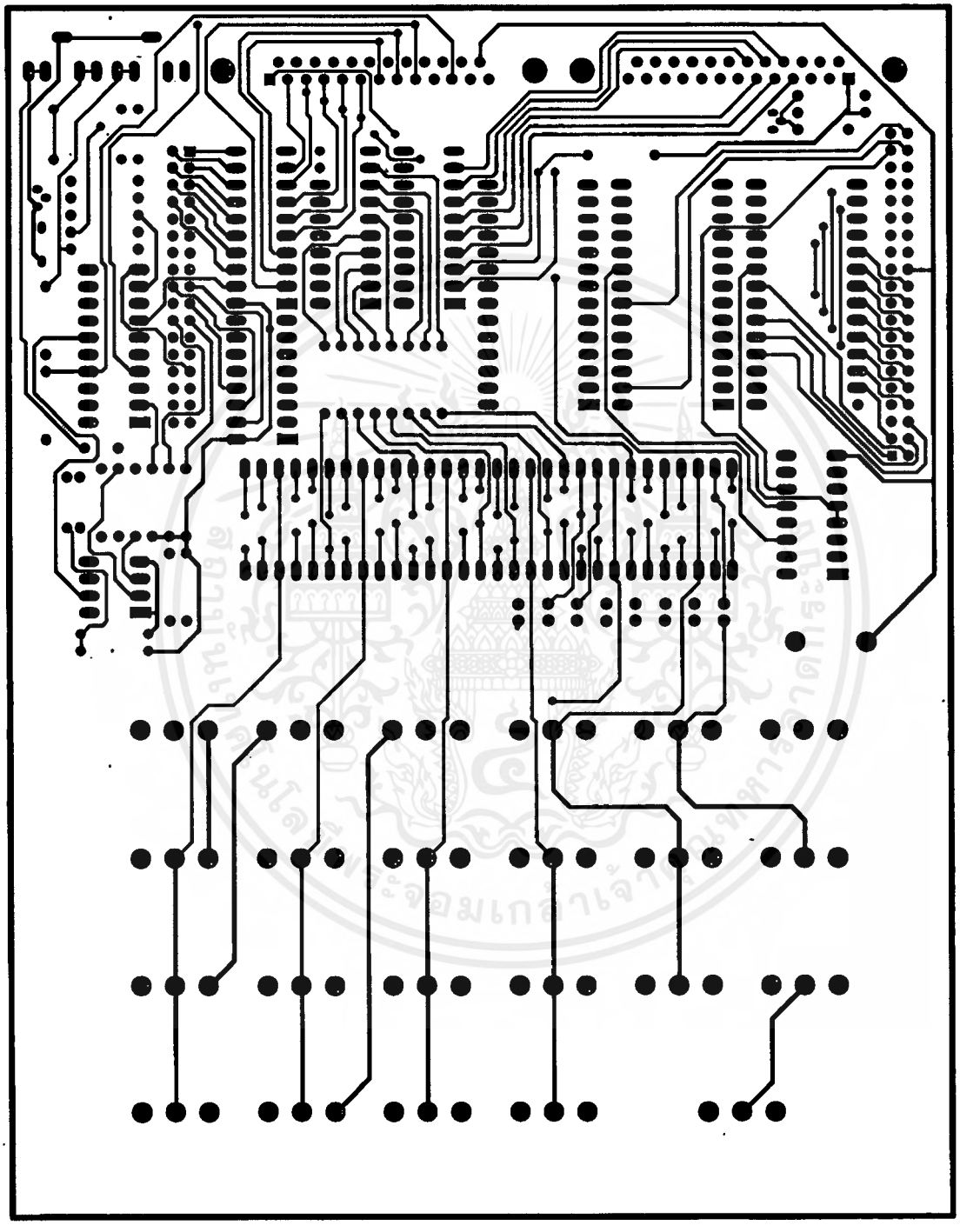
รูปแสดงวงจรของเครื่องควบคุมหุ่นยนต์ ตอนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



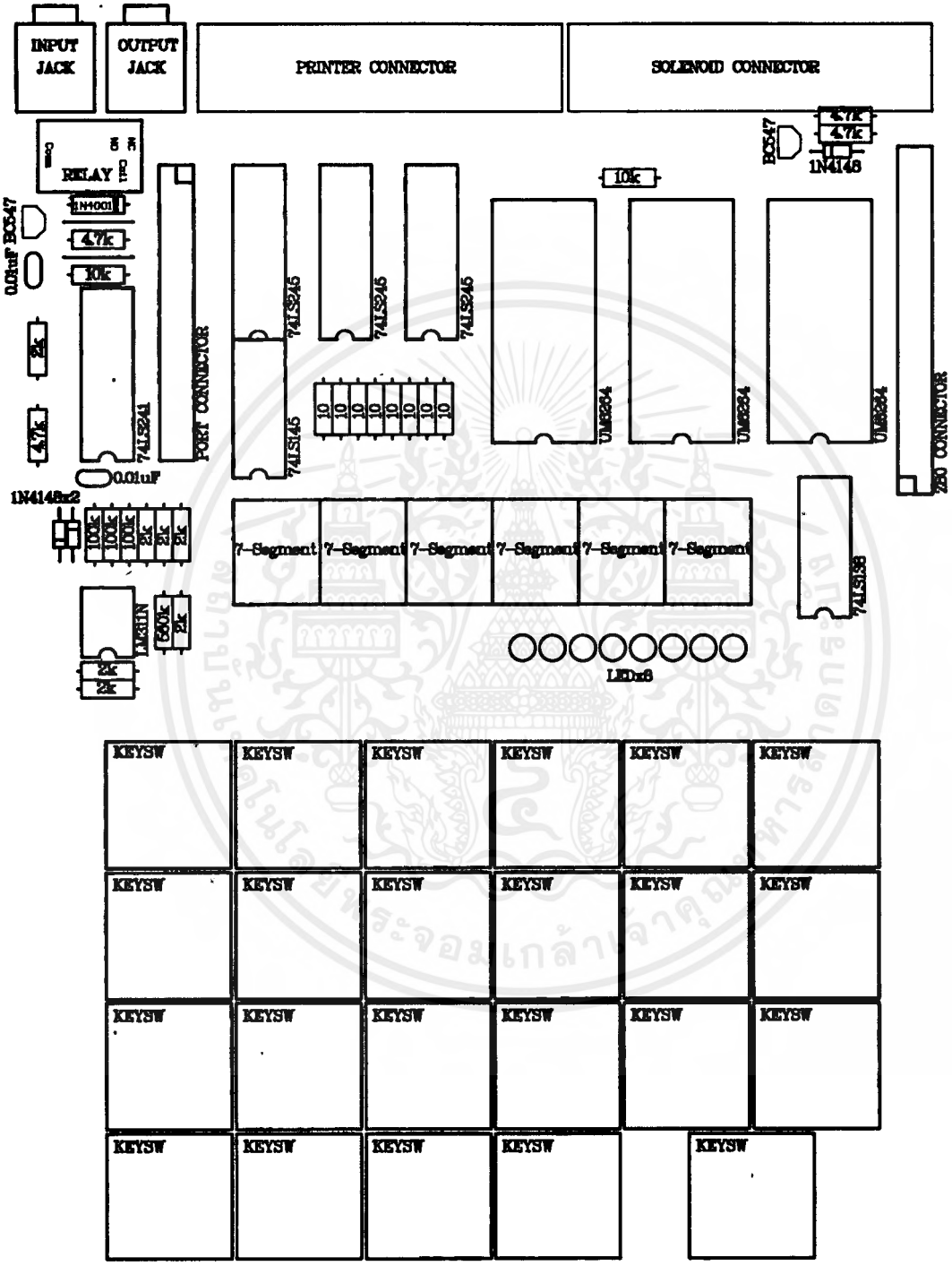
รูปแสดงลายวงจรด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



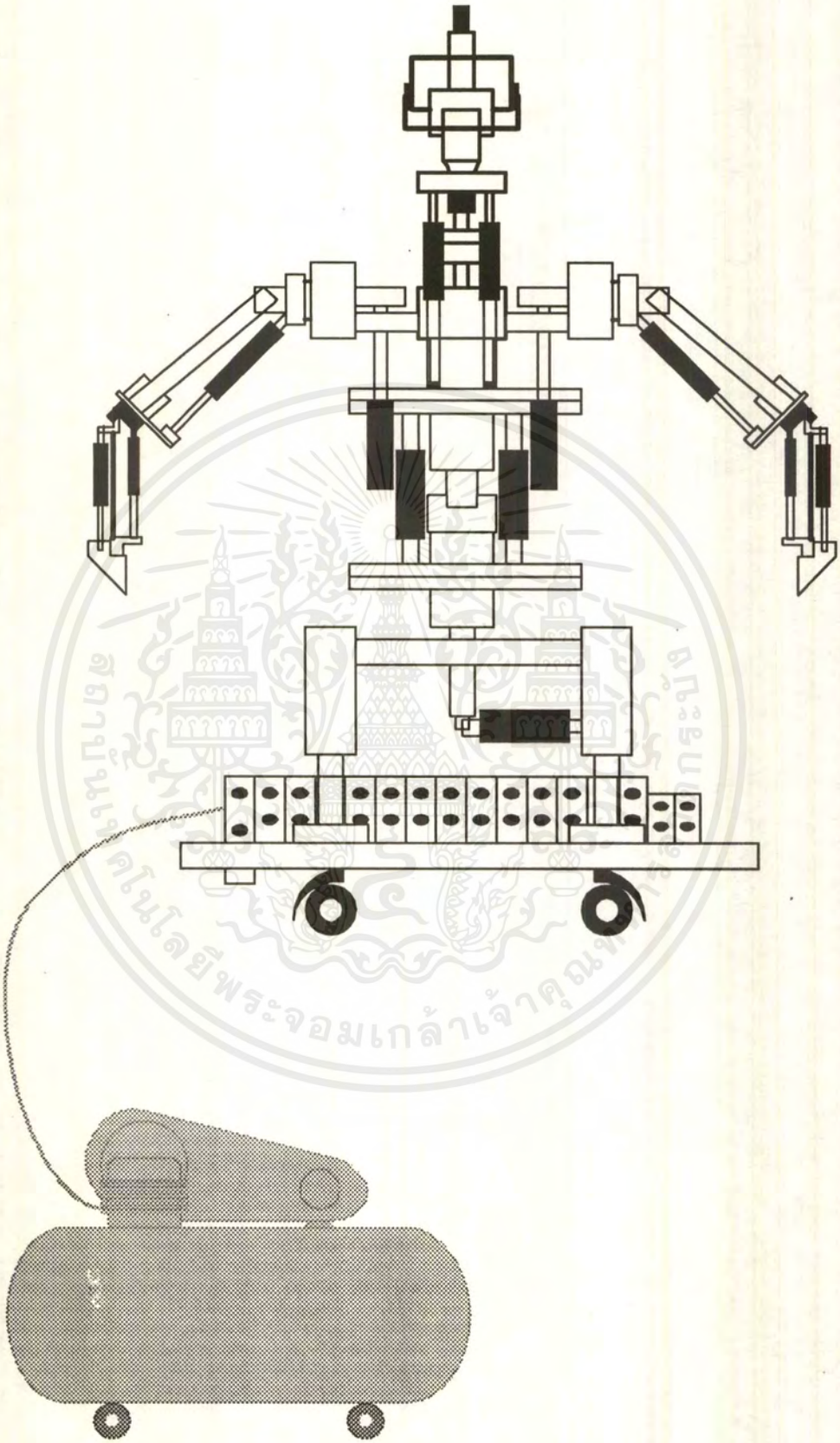
รูปแสดงลายวงจรด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



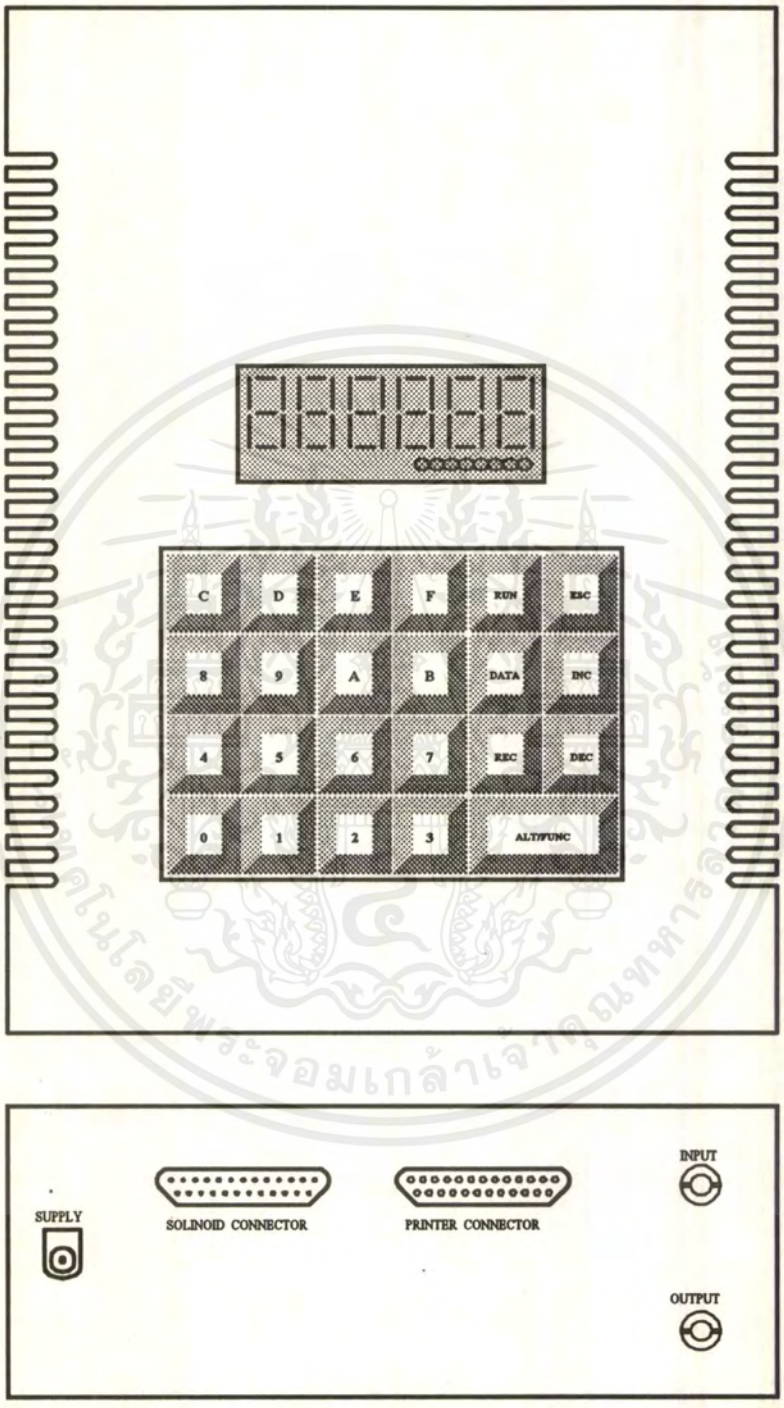
รูปแสดงการวางอุปกรณ์บนแผ่นวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง โครงสร้างของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง รูปร่างของโครงงานด้านหน้าและด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                2500 A.D. Z80 Macro Assembler  -  Version 4.02a
                ------------------------------------------------

                        Input  Filename : rbt.asm
                        Output Filename : rbt.obj

        0000                            ORG 0000H

        0040            DATA_           EQU 40H
        0041            SEGM            EQU 41H
        0042            DIGIT           EQU 42H
        0043            CTRL            EQU 43H

        0040            PRNDAT          EQU 40H
        0041            PRNSTB          EQU 41H
        0042            PRNBUSY         EQU 42H

        03E8            SYNC1           EQU 3E8H
        0320            SYNC2           EQU 320H
        0060            REP1            EQU 60H
        0016            REP2            EQU 16H

        9FCF            STACK           EQU 9FCFH
        9E00            SAVMEM          EQU 9E00H
        9FE0            RAM_AREA        EQU 9FE0H

0000    21 00 40        MON:            LD HL,4000H         ; Start up delay
0003    2B              MON1:           DEC HL
0004    7D                              LD A,L
0005    B4                              OR H
0006    20 FB                           JR NZ,MON1
0008    3E 88                           LD A,88H            ; Initial 8255
000A    D3 43                           OUT (CTRL),A

000C    31 CF 9F                        LD SP,STACK
000F    CD F9 03        MON11:          CALL OUTPE          ; Clear port
0012    CD F1 01                        CALL CLR0           ; Clear buffer
0015    D3 41                           OUT (SEGM),A
0017    36 08                           LD (HL),08H
0019    21 E0 9F        MON2:           LD HL,SYSFLG
001C    CB D6                           SET 2,(HL)
001E    CD 67 00                        CALL SCAN
0021    CB 96                           RES 2,(HL)
0023    FE 15                           CP 15H
0025    CA 77 04                        JP Z,RUN            ; Run
0028    FE 14                           CP 14H
002A    CA 66 02                        JP Z,DATA0          ; Data editor
002D    FE 12                           CP 12H
002F    CA 1E 03                        JP Z,REC            ; Record
0032    21 E8 9F                        LD HL,DISPX+5
0035    FE 10                           CP 10H
0037    28 09                           JR Z,MON3           ; Function key
0039    CB 46                           BIT 0,(HL)
003B    20 0B                           JR NZ,MONF
003D    CD BE 01                        CALL CHK4           ; Number key
0040    18 D7                           JR MON2

0042    7E              MON3:           LD A,(HL)
0043    EE 01                           XOR 1               ; Toggle function
0045    77                              LD (HL),A
0046    18 D1                           JR MON2
0048    FE 07           MONF:           CP 7
004A    30 CD                           JR NC,MON2
004C
004C    87                              ADD A,A
004D    5F                              LD E,A
004E    16 00                           LD D,0
0050    21 59 00                        LD HL,FUNCTAB
0053    19                              ADD HL,DE
0054    5E                              LD E,(HL)
0055    23                              INC HL
0056    56                              LD D,(HL)

0057    EB                              EX DE,HL
0058    E9                              JP (HL)             ; Jump to function

0059    3F05            FUNCTAB         DW DSIGN            ; Design
005B    5E06                            DW SAVE             ; Save
005D    B106                            DW LOAD             ; Load
005F    F305                            DW LIST             ; List
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0061   2807                          DW PRN              ; Print
0063   6908                          DW PCL              ; PC link
0065   A701                          DW MOVES            ; Move data

                        ;********** SCAN KEYBOARD **********
                        ; Scan keyboard
                        ; in  (DISPX)    = Segment code
                        ; out A,(KEYIN) = Key code
                        ; reg ABCDEHL

0067   01 08 08         SCAN:         LD BC,0808H
006A   1E 00                         LD E,0
006C   21 E3 9F                      LD HL,DISPX

006F   CD 01 01         SCAN1:        CALL SCANS
0072   22 F0 9F                      LD (MEMX),HL
0075   DB 42                         IN A,(DIGIT)
0077   E6 70                         AND 70H
0079   FE 70                         CP  70H
007B   20 17                         JR NZ,SCAN3

007D   0D               SCAN11:       DEC C
007E   20 0C                         JR NZ,SCAN2
0080   21 E0 9F                      LD HL,SYSFLG
0083   CB 86                         RES 0,(HL)
0085   CB BE                         RES 7,(HL)
0087   21 E2 9F         SCAN12:       LD HL,REPDLY
008A   36 60                         LD (HL),REP1

008C   2A F0 9F         SCAN2:        LD HL,(MEMX)
008F   1C                           INC E
0090   10 DD                         DJNZ SCAN1
0092   18 D3                         JR SCAN

0094   B3               SCAN3:        OR E                ; Key pressed
0095   C5                           PUSH BC             ; Decode key code
0096   21 25 01                      LD HL,KEYTAB+16H
0099   06 16                         LD B,16H
009B   BE               SCAN31:       CP (HL)
009C   28 03                         JR Z,SCAN32
009E   2B                            DEC HL
009F   10 FA                         DJNZ SCAN31
00A1   50               SCAN32:       LD D,B
00A2   C1                            POP BC

00A3   21 E0 9F                      LD HL,SYSFLG
00A6   AF                            XOR A
00A7   CB 56                         BIT 2,(HL)
00A9   CC E5 00                      CALL Z,SCANA        ; ALT check
00AC   38 CF                         JR C,SCAN11
00AE   CB 46                         BIT 0,(HL)
00B0   28 24                         JR Z,SCAN5

00B2   7A                            LD A,D              ; Repeated key check
00B3   E6 7F                         AND 7FH
00B5   CD F6 00                      CALL SCANC
00B8   38 D2                         JR C,SCAN2
00BA   7E               SCAN4:        LD A,(HL)
00BB   AA                            XOR D
00BC   E6 80                         AND 80H
00BE   20 C7                         JR NZ,SCAN12
00C0   3A E1 9F                      LD A,(KEYIN)
00C3   CD F6 00                      CALL SCANC
00C6   38 C4                         JR C,SCAN2

00C8   57                            LD D,A              ; Repeat key
00C9   7E                            LD A,(HL)
00CA   E6 80                         AND 80H
00CC   B2                            OR D
00CD   21 E2 9F                      LD HL,REPDLY
00D0   35                            DEC (HL)
00D1   20 B9                         JR NZ,SCAN2
00D3   36 16                         LD (HL),REP2
00D5   C9                            RET

00D6   7A               SCAN5:        LD A,D              ; New press
00D7   E6 7F                         AND 7FH
00D9   32 E1 9F                      LD (KEYIN),A
00DC   CB C6                         SET 0,(HL)
00DE   CB 7A                         BIT 7,D
00E0   C8                            RET Z
00E1   CB FE                         SET 7,(HL)
00E3   7A                            LD A,D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
00E4   C9                              RET

00E5   3E 10          SCANA:           LD A,10H            ; Check ALT key
00E7   AA                              XOR D
00E8   37                              SCF
00E9   C8                              RET Z
00EA   3E 07          SCANA1:          LD A,7
00EC   D3 42                           OUT (DIGIT),A
00EE   DB 42                           IN A,(DIGIT)
00F0   E6 10                           AND 10H
00F2   C0                              RET NZ
00F3   CB FA                           SET 7,D
00F5   C9                              RET

00F6   FE 13          SCANC:           CP 13H              ; Check repeat key
00F8   C8                              RET Z
00F9   FE 11                           CP 11H
00FB   C8                              RET Z
00FC   FE 16                           CP 16H
00FE   C8                              RET Z
00FF   37                              SCF
0100   C9                              RET

0101   7B             SCANS:          LD A,E              ; Scan display
0102   D3 42                          OUT (DIGIT),A
0104   7E                             LD A,(HL)
0105   D3 41                          OUT (SEGM),A
0107   AF                             XOR A
0108   3D             SCANS1:         DEC A
0109   20 FD                          JR NZ,SCANS1
010B   D3 41                          OUT (SEGM),A
010D   23                             INC HL
010E   C9                             RET

010F   60 61 62 63     KEYTAB          DB 60H,61H,62H,63H
0113   50 51 52 53                    DB 50H,51H,52H,53H
0117   30 31 32 33                    DB 30H,31H,32H,33H
011B   34 54 64 65                    DB 34H,54H,64H,65H
011F   67 57 55 36                    DB 67H,57H,55H,36H
0123   35 66 47                       DB 35H,66H,47H

                      ;******* HEX TO SEGMENT CODE *******
                      ; Convert HEX to Segment code
                      ; in  (DISPY)    = HEX
                      ; out (DISPX)    = Segment code
                      ; reg ABCDEHL

0126   21 EC 9F       H2SEG:          LD HL,DISPY+1
0129   11 E3 9F                       LD DE,DISPX
012C   CD 35 01                       CALL H2SEGC
012F   2B                             DEC HL
0130   CD 35 01                       CALL H2SEGC
0133   23                             INC HL
0134   23                             INC HL

0135   AF             H2SEGC:         XOR A
0136   06 02                          LD B,02H
0138   ED 6F          H2SEGC1:        RLD
013A   4F                            LD C,A
013B   CD 46 01                      CALL H2SEGT
013E   12                            LD (DE),A
013F   13                            INC DE
0140   79                            LD A,C
0141   10 F5                         DJNZ H2SEGC1
0143   ED 6F                         RLD
0145   C9                            RET

0146   E5             H2SEGT:         PUSH HL             ; Look up table
0147   21 7E 01                       LD HL,SEGTAB
014A   85                             ADD A,L
014B   6F                             LD L,A
014C   3E 00                          LD A,0
014E   8C                             ADC A,H
014F   67                             LD H,A
0150   7E                             LD A,(HL)
0151   E1                             POP HL
0152   C9                             RET

                      ;******* SEGMENT CODE TO HEX *******
                      ; Convert Segment code to HEX
                      ; in  (DISPX)    = Segment code
                      ; out (DISPY)    = HEX
                      ; reg ABCDEHL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0153   21 EC 9F          SEG2H:          LD HL,DISPY+1
0156   11 E3 9F                          LD DE,DISPX
0159   CD 62 01                          CALL SEG2HC
015C   2B                                DEC HL
015D   CD 62 01                          CALL SEG2HC
0160   23                                INC HL
0161   23                                INC HL

0162   0E 02             SEG2HC:         LD C,02H
0164   1A                SEG2HC1:        LD A,(DE)
0165   CD 6F 01                          CALL SEG2HT
0168   ED 6F                             RLD
016A   13                                INC DE
016B   0D                                DEC C
016C   20 F6                             JR NZ,SEG2HC1
016E   C9                                RET

016F   E5                SEG2HT:         PUSH HL              ; Look up table
0170   21 8D 01                          LD HL,SEGTAB+0FH
0173   06 0F                             LD B,0FH
0175   BE                SEG2HT1:        CP (HL)
0176   28 03                             JR Z,SEG2HT2
0178   2B                                DEC HL
0179   10 FA                             DJNZ SEG2HT1
017B   78                SEG2HT2:        LD A,B
017C   E1                                POP HL
017D   C9                                RET

017E   3F 06 5B 4F       SEGTAB          DB 3FH,06H,5BH,4FH
0182   66 6D 7D 07                       DB 66H,6DH,7DH,07H
0186   7F 6F 77 7C                       DB 7FH,6FH,77H,7CH
018A   39 5E 79 71                       DB 39H,5EH,79H,71H

                         ;************ MOVES DATA *************
                         ; MOVEs block of data to destination
                         ; in  DE = Start address
                         ;     HL = Final address
                         ;     BC = Destination address
                         ; reg ABCDEHL

018E   AF                MOVE:           XOR A
018F   ED 52                             SBC HL,DE
0191   C5                                PUSH BC              ; BC=DEST
0192   E5                                PUSH HL
0193   19                                ADD HL,DE            ; HL=FINAL-START
0194   EB                                EX DE,HL             ; HL=START,DE=FINAL
0195   ED 42                             SBC HL,BC            ; HL=START-DEST
0197   C1                                POP BC               ; BC=lenght-1
0198   30 07                             JR NC,MOVE1

019A   E1                                POP HL               ; HL=DEST
019B   09                                ADD HL,BC            ; HL=DEST+lenght-1
019C   EB                                EX DE,HL             ; HL=FINAL,DE=DEST+lenght-1
019D   03                                INC BC
019E   ED B8                             LDDR
01A0   C9                                RET

01A1   D1                MOVE1:          POP DE               ; DE=DEST
01A2   19                                ADD HL,DE            ; HL=START,DE=DEST
01A3   03                                INC BC
01A4   ED B0                             LDIR
01A6   C9                                RET

                         ;**** MOVE DATA TO DESTINATION *****
                         ; Copy data to destination address
                         ; reg BCDEHL

01A7   CD 26 02          MOVES:          CALL INSF
01AA   CD 3D 02                          CALL IND

01AD   ED 5B F9 9F                       LD DE,(START)
01B1   2A F7 9F                          LD HL,(FINAL)
01B4   ED 4B FB 9F                       LD BC,(DEST)
01B8   CD 8E 01                          CALL MOVE

01BB   C3 00 00                          JP MON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;********** CHECK 4 DIGIT **********
                        ; Check 4 digit input
                        ; in  A = Keycode
                        ; reg ABCHL

01BE   FE 11            CHK4:           CP 11H
01C0   28 1C                            JR Z,CHK43          ; Is DEC key?
01C2   FE 10                            CP 10H
01C4   D0                               RET NC              ; Is number key?
01C5   CD 46 01                         CALL H2SEGT
01C8   4F                               LD C,A
01C9   06 04                            LD B,04H
01CB   21 E3 9F                         LD HL,DISPX
01CE   7E               CHK41:          LD A,(HL)
01CF   FE 08                            CP 08H              ; Find cursor
01D1   28 06                            JR Z,CHK42
01D3   23                               INC HL
01D4   10 F8                            DJNZ CHK41
01D6   CD F8 01                         CALL CLR1           ; Clear buffer
01D9   71               CHK42:          LD (HL),C           ; Fill number
01DA   23                               INC HL
01DB   36 08                            LD (HL),08H
01DD   C9                               RET

01DE   06 04            CHK43:          LD B,04H            ; Delete last digit
01E0   21 E7 9F                         LD HL,DISPX+4
01E3   2B               CHK44:          DEC HL
01E4   7E                               LD A,(HL)
01E5   E6 F7                            AND 0F7H
01E7   20 02                            JR NZ,CHK45
01E9   10 F8                            DJNZ CHK44
01EB   36 08            CHK45:          LD (HL),08H
01ED   23                               INC HL
01EE   36 00                            LD (HL),00H
01F0   C9                               RET

                        ;*************** CLEAR **************
                        ; Clear display bufffer & HEX buffer
                        ; reg ABHL

01F1   06 0B            CLR0:           LD B,0BH            ; Clear all
01F3   21 EE 9F                         LD HL,DISPY+3
01F6   18 05                            JR CLR2

01F8   06 08            CLR1:           LD B,08H            ; Clear display buffer
01FA   21 EB 9F                         LD HL,DISPX+8

01FD   AF               CLR2:           XOR A
01FE   2B               CLR21:          DEC HL
01FF   77                               LD (HL),A
0200   10 FC                            DJNZ CLR21
0202   C9                               RET

                        ;********** INPUT 4 DIGIT ***********
                        ; Recieve HEX 4 digit input
                        ; in HL = HEX
                        ;    B  = Segment code
                        ; out (DISPY) = HEX
                        ; reg BHL

0203   22 EB 9F         IN4:            LD (DISPY),HL
0206   60                               LD H,B
0207   2E 40                            LD L,40H
0209   22 EE 9F                         LD (MEMY),HL
020C   CD 26 01                         CALL H2SEG

020F   2A EE 9F         IN41:           LD HL,(MEMY)
0212   22 E7 9F                         LD (DISPX+4),HL
0215   CD 67 00                         CALL SCAN
0218   CD BE 01                         CALL CHK4
021B   FE 13                            CP 13H
021D   20 F0                            JR NZ,IN41

021F   CD 53 01                         CALL SEG2H
0222   2A EB 9F                         LD HL,(DISPY)
0225   C9                               RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;******* INPUT START & FINAL *******
                        ; Recieve Start & Final address
                        ; in  (START),(FINAL) = Old address
                        ; out (START),(FINAL) = New address
                        ; reg BHL

0226   2A F9 9F          INSF:           LD HL,(START)
0229   06 6D                            LD B,6DH
022B   CD 03 02                         CALL IN4
022E   22 F9 9F                         LD (START),HL

0231   2A F7 9F          INF:            LD HL,(FINAL)
0234   06 71                            LD B,71H
0236   CD 03 02                         CALL IN4
0239   22 F7 9F                         LD (FINAL),HL
023C   C9                               RET

                        ;******** INPUT DESTINATION ********
                        ; Recieve Destination address
                        ; in  (DEST) = Old Destination addr
                        ; out (DEST) = New Destination addr
                        ; reg BHL

023D   2A FB 9F          IND:            LD HL,(DEST)
0240   06 5E                            LD B,5EH
0242   CD 03 02                         CALL IN4
0245   22 FB 9F                         LD (DEST),HL
0248   C9                               RET

                        ;********** CHECK 2 DIGIT **********
                        ; Check 2 digit input
                        ; in A = Key code
                        ; reg ACHL

0249   CD 46 01          CHK2:           CALL H2SEGT
024C   4F                               LD C,A
024D   7E                               LD A,(HL)
024E   B7                               OR A
024F   28 03                            JR Z,CHK21
0251   36 00                            LD (HL),0
0253   2B                               DEC HL

0254   71                CHK21:          LD (HL),C
0255   C9                               RET

                        ;************ SCAN DISPLAY ************
                        ; Display data in display buffer
                        ; in (DISPX) = Segment code
                        ; reg DEHL

0256   21 E3 9F          SCAND:          LD HL,DISPX
0259   11 00 08          SCAND1:         LD DE,0800H
025C   CD 01 01          SCAND2:         CALL SCANS
025F   1C                               INC E
0260   15                               DEC D
0261   20 F9                            JR NZ,SCAND2
0263   10 F1                            DJNZ SCAND
0265   C9                               RET

                        ;************ DATA EDITOR ************
                        ; Edit data in memory
                        ; reg ADHL,IX

0266   21 E3 9F          DATA0:          LD HL,DISPX
0269   7E                               LD A,(HL)
026A   FE 08                            CP 08H
026C   28 09                            JR Z,DATA1
026E   CD 53 01                         CALL SEG2H
0271   2A EB 9F                         LD HL,(DISPY)
0274   22 F5 9F                         LD (ADDR),HL

0277   DD 2A F5 9F       DATA1:          LD IX,(ADDR)        ; Load address
027B   DD 22 EB 9F       DATA12:         LD (DISPY),IX
027F   DD 7E 00                         LD A,(IX+0)
0282   32 ED 9F                         LD (DISPY+2),A
0285   CD 26 01                         CALL H2SEG
0288   21 E6 9F                         LD HL,DISPX+3
028B   CB FE                            SET 7,(HL)

028D   CD 67 00          DATA2:          CALL SCAN
0290   FE 13                            CP 13H              ; Next
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0292   28 31                          JR Z,DATA3
0294   FE 11                          CP 11H              ; Previous
0296   28 31                          JR Z,DATA4
0298   FE 93                          CP 93H              ; Insert
029A   28 31                          JR Z,DATA5
029C   FE 96                          CP 96H              ; Delete
029E   28 2D                          JR Z,DATA5
02A0   FE 8F                          CP 8FH              ; Final address
02A2   28 2E                          JR Z,DATA6
02A4   FE 10                          CP 10H              ; Is number key?
02A6   30 E5                          JR NC,DATA2
02A8
02A8   21 E8 9F                       LD HL,DISPX+5
02AB   CD 49 02                       CALL CHK2           ; Get 2 digit
02AE   20 DD                          JR NZ,DATA2
02B0   CD 53 01                       CALL SEG2H
02B3   3A ED 9F                       LD A,(DISPY+2)
02B6   DD 77 00                       LD (IX+00),A

02B9   57                             LD D,A
02BA   DD 7E 00                       LD A,(IX+00)
02BD   BA                             CP D                ; Check data writing
02BE   20 BB                          JR NZ,DATA12

02C0   06 16                          LD B,REP2
02C2   CD 56 02                       CALL SCAND
02C5   DD 23                DATA3:    INC IX              ; Change to next
02C7   18 B2                          JR DATA12

02C9   DD 2B                DATA4:    DEC IX              ; Change to previous
02CB   18 AE                          JR DATA12
02CD   CD D7 02             DATA5:    CALL EDIT
02D0   30 A9                          JR NC,DATA12
02D2   CD 31 02             DATA6:    CALL INF
02D5   18 A4                          JR DATA12

                            ;***** EDIT DATA (INS or DEL) ******
                            ; Insert or Delete data 1 byte
                            ; in IX = Start address
                            ;    (FINAL) = Final address
                            ; reg ABCDEHL,IX

02D7   DD E5                EDIT:     PUSH IX
02D9   D1                             POP DE
02DA   2A F7 9F                       LD HL,(FINAL)
02DD   ED 52                          SBC HL,DE
02DF   D8                             RET C
02E0   19                             ADD HL,DE
02E1   D5                             PUSH DE
02E2   C1                             POP BC
02E3   FE 93                          CP 93H
02E5   20 0E                          JR NZ,EDIT1

02E7   23                             INC HL              ; Insert data
02E8   22 F7 9F                       LD (FINAL),HL
02EB   2B                             DEC HL
02EC   03                             INC BC
02ED   CD 8E 01                       CALL MOVE
02F0   AF                             XOR A
02F1   12                             LD (DE),A
02F2   DD 23                          INC IX
02F4   C9                             RET

02F5   2B                   EDIT1:    DEC HL              ; Delete data
02F6   22 F7 9F                       LD (FINAL),HL
02F9   23                             INC HL
02FA   23                             INC HL
02FB   13                             INC DE
02FC   CD 8E 01                       CALL MOVE
02FF   1B                             DEC DE
0300   AF                             XOR A
0301   12                             LD (DE),A
0302   C9                             RET

                            ;************** DIVIDE *************
                            ; Divide HEX number
                            ; in  HL = tuatang
                            ;     C  = tuahan
                            ; out HL = pollub
                            ; reg ABCHL

0303   AF                   DIV:      XOR A
0304   06 10                          LD B,10H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0306  ED 6A          DIV1:       ADC HL,HL          ; Result in
0308  8F                        ADC A,A
0309  91                        SUB C
030A  30 01                     JR NC,DIV2
030C  81                        ADD A,C
030D  3F              DIV2:     CCF
030E  10 F6                     DJNZ DIV1
0310  ED 6A                     ADC HL,HL
0312  C9                       RET

                      ;************** DELAY **************
                      ; Time Delay
                      ; in BC = Delay value
                      ; reg ABC

0313  3E F8           DELAY:     LD A,0F8H
0315  3D              DELAY1:    DEC A
0316  20 FD                     JR NZ,DELAY1
0318  0B                       DEC BC
0319  78                       LD A,B
031A  B1                       OR C
031B  20 F6                    JR NZ,DELAY
031D  C9                       RET

                      ;************** RECORD **************
                      ; Recieve block address
                      ; reg ABDEHL

031E  DD 21 00 9E     REC:      LD IX,SAVMEM       ; Load buffer
0322  CD F1 01                  CALL CLR0
0325  32 EE 9F                 LD (MEMY),A

0328  3E 6D           REC1:     LD A,6DH
032A  32 EF 9F        REC2:     LD (MEMY+1),A
032D  DD 66 00                  LD H,(IX+0)
0330  DD 6E FF                  LD L,(IX-1)        ; Decrease count
0333  22 EB 9F                  LD (DISPY),HL
0336  CD 26 01                  CALL H2SEG

0339  2A EE 9F        REC3:     LD HL,(MEMY)
033C  7D                        LD A,L
033D  2E 40                     LD L,40H
033F  22 E7 9F                  LD (DISPX+4),HL
0342  32 E9 9F                  LD (DISPX+6),A
0345  CD 67 00                  CALL SCAN
0348  CD BE 01                  CALL CHK4
034B  FE 13                     CP 13H
034D  20 EA                     JR NZ,REC3

034F  CD 53 01                  CALL SEG2H
0352  2A EB 9F                  LD HL,(DISPY)
0355  DD 74 00                  LD (IX+0),H
0358  DD 75 FF                  LD (IX-1),L
035B  DD 2B                     DEC IX
035D  DD 2B                     DEC IX
035F  EB                        EX DE,HL
0360  21 EF 9F                  LD HL,MEMY+1
0363  7E                        LD A,(HL)
0364  2B                        DEC HL
0365  34                        INC (HL)           ; indicate counter
0366  28 B6                     JR Z,REC

0368  06 71                     LD B,71H
036A  B8                        CP B
036B  28 BB                     JR Z,REC1          ; Jump for Final addr
036D  35                        DEC (HL)
036E  13                        INC DE
036F  7A                        LD A,D
0370  B3                        OR E
0371  78                        LD A,B
0372  20 B6                     JR NZ,REC2
0374  CD 7A 03                  CALL RECS          ; Save to tape
0377  C3 00 00                  JP MON

                      ;************ RECORD SAVE ************
                      ; Save Address to tape recorder
                      ; in HL = Counter address
                      ; reg ABCDEHL,IX

037A  34              RECS:     INC (HL)
037B  DD 21 00 9E               LD IX,SAVMEM
037F  01 E8 03        RECS1:    LD BC,SYNC1        ; Start sync signal
0382  CD 49 04        RECS2:    CALL TONE1K
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0385   0B                              DEC BC
0386   78                              LD A,B
0387   B1                              OR C
0388   20 F8                           JR NZ,RECS2

038A   1E 04                           LD E,4
038C   DD 56 00          RECS3:       LD D,(IX+00)        ; High byte first
038F   CD 16 04                        CALL WRBYTE
0392   DD 2B                           DEC IX
0394   1D                              DEC E
0395   20 F5                           JR NZ,RECS3

0397   06 FF                           LD B,0FFH
0399   CD 60 04          RECS4:       CALL TONE2K         ; End sync signal
039C   10 FB                           DJNZ RECS4
039E   21 EE 9F                        LD HL,MEMY
03A1   35                              DEC (HL)
03A2   B6                              OR (HL)
03A3   C8                              RET Z               ; End process

03A4   DD 6E 01                        LD L,(IX+01)
03A7   DD 66 02                        LD H,(IX+02)
03AA   22 F7 9F                        LD (FINAL),HL
03AD   DD 6E 03                        LD L,(IX+03)
03B0   DD 66 04                        LD H,(IX+04)
03B3   22 F9 9F                        LD (START),HL
03B6   CD C1 03                        CALL SHOW           ; Sent data out

03B9   01 70 00                        LD BC,70H
03BC   CD 13 03                        CALL DELAY
03BF   18 BE                           JR RECS1

                       ;********** SHOW MOVEMENT **********
                       ; Show movement of robot
                       ; in (START),(FINAL) = address
                       ; reg ABCDEHL

03C1   3E 0E           SHOW:           LD A,0EH            ; Select I/O mode
03C3   D3 42                           OUT (DIGIT),A
03C5   CD F9 03                        CALL OUTPE          ; Clear port
03C8   AF                              XOR A
03C9   D3 41                           OUT (SEGM),A
03CB   2A F7 9F                        LD HL,(FINAL)
03CE   ED 5B F9 9F                     LD DE,(START)
03D2   23                              INC HL
03D3   ED 52                           SBC HL,DE

03D5   0E 05                           LD C,05H
03D7   CD 03 03                        CALL DIV            ; Divide by 5
03DA   EB                              EX DE,HL
03DB   CD ED 03        SHOW1:          CALL OUTP           ; Sent to port
03DE   4E                              LD C,(HL)
03DF   23                              INC HL
03E0   46                              LD B,(HL)
03E1   CD 13 03                        CALL DELAY

03E4   23                              INC HL
03E5   1B                              DEC DE
03E6   7A                              LD A,D
03E7   B3                              OR E
03E8   20 F1                           JR NZ,SHOW1
03EA   D3 41                           OUT (SEGM),A
03EC   C9                              RET

                       ;************** OUTPORT **************
                       ; Sent data to Solenoid driver
                       ; in HL = Address of first port data
                       ; reg ABCHL

03ED   01 29 03        OUTP:           LD BC,0329H
03F0   7E              OUTP1:          LD A,(HL)
03F1   CD 04 04                        CALL OUTPC
03F4   0C                              INC C
03F5   23                              INC HL
03F6   10 F8                           DJNZ OUTP1
03F8   C9                              RET

03F9   01 29 03        OUTPE:          LD BC,0329H
03FC   AF              OUTP2:          XOR A
03FD   CD 04 04                        CALL OUTPC
0400   0C                              INC C
0401   10 F9                           DJNZ OUTP2
0403   C9                              RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0404   CD 08 04          OUTPC:          CALL OUTP3          ; data out
0407   AF                                XOR A               ; Clear buffer

0408   D3 40             OUTP3:          OUT (DATA_),A
040A   79                                LD A,C
040B   D3 41                             OUT (SEGM),A        ; Select port
040D   3E 0E                             LD A,0EH
040F   D3 42                             OUT (DIGIT),A       ; Select I/O mode
0411   3E 20                             LD A,20H
0413   D3 41                             OUT (SEGM),A        ; Disable port
0415   C9                                RET

                         ;********** WRITE 1 BYTE ***********
                         ; Write 1 byte to tape
                         ; in D = HEX
                         ; reg ADC

0416   AF                WRBYTE:         XOR A               ; Start bit = 0
0417   CD 29 04                          CALL WRBIT
041A   0E 08                           LD C,8              ; Data 8 bit
041C   CB 1A             WRBYT1:         RR D
041E   CD 29 04                          CALL WRBIT
0421   0D                                DEC C
0422   20 F8                             JR NZ,WRBYT1
0424   37                                SCF                 ; Stop bit = 1
0425   CD 29 04                          CALL WRBIT
0428   C9                                RET

                         ;*********** WRITE 1 BIT ***********
                         ; Write 1 bit to tape
                         ; in Carry Flag
                         ; reg BC

0429   38 0F             WRBIT:          JR C,WRBIT3
042B   06 10                           LD B,10H            ; Logic 0
042D   CD 60 04           WRBIT1:         CALL TONE2K
0430   10 FB                             DJNZ WRBIT1
0432   06 04                           LD B,4
0434   CD 49 04          WRBIT2:         CALL TONE1K
0437   10 FB                             DJNZ WRBIT2
0439   C9                                RET
043A   06 08             WRBIT3:         LD B,8              ; Logic 1
043C   CD 60 04           WRBIT4:         CALL TONE2K
043F   10 FB                             DJNZ WRBIT4
0441   06 08                           LD B,8
0443   CD 49 04          WRBIT5:         CALL TONE1K
0446   10 FB                             DJNZ WRBIT5
0448   C9                                RET

                         ; Create Frequency

0449   3E 0F             TONE1K:         LD A,0FH            ; 1 kHz
044B   D3 42                             OUT (DIGIT),A
044D   3E A0                             LD A,0A0H
044F   D3 41                             OUT (SEGM),A
0451   3E 7B                             LD A,7BH
0453   3D                TONE11:         DEC A
0454   20 FD                             JR NZ,TONE11
0456   3E 00                             LD A,0
0458   D3 41                             OUT (SEGM),A
045A   3E 7B                             LD A,7BH
045C   3D                TONE12:         DEC A
045D   20 FD                             JR NZ,TONE12
045F   C9                                RET

0460   3E 0F             TONE2K:         LD A,0FH            ; 2 kHz
0462   D3 42                             OUT (DIGIT),A
0464   3E A0                             LD A,0A0H
0466   D3 41                             OUT (SEGM),A
0468   3E 3C                             LD A,3CH
046A   3D                TONE21:         DEC A
046B   20 FD                             JR NZ,TONE21
046D   3E 00                             LD A,0
046F   D3 41                             OUT (SEGM),A
0471   3E 3C                             LD A,3CH
0473   3D                TONE22:         DEC A
0474   20 FD                             JR NZ,TONE22
0476   C9                                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;*************** RUN ***************
                        ; Load address from tape and transfer
                        ;  to solenoid driver
                        ; reg ABCDHL

0477   3E 0E            RUN:            LD A,0EH            ; Select I/O mode
0479   D3 42                           OUT (DIGIT),A
047B   AF                              XOR A
047C   D3 41                           OUT (SEGM),A        ; Clear Port

047E   CD EE 04         RUN1:           CALL CYCLE          ; Wait for first signal
0481   38 FB                           JR C,RUN1
0483   3E 18                           LD A,18H
0485   D3 41                           OUT (SEGM),A        ; Sync Display

0487   01 20 03                        LD BC,SYNC2
048A   CD EE 04         RUN2:           CALL CYCLE          ; Read start sync
048D   38 FB                           JR C,RUN2
048F   0B                              DEC BC
0490   78                              LD A,B
0491   B1                              OR C
0492   20 F6                           JR NZ,RUN2

0494   CD EE 04         RUN3:           CALL CYCLE
0497   30 FB                           JR NC,RUN3          ; Wait for Start Bit

0499   06 04                           LD B,04H
049B   21 FA 9F                        LD HL,START+1       ; Read address
049E   C5               RUN4:           PUSH BC
049F   CD B5 04                        CALL RDBYTE
04A2   C1                              POP BC
04A3   72                              LD (HL),D
04A4   2B                              DEC HL
04A5   10 F7                           DJNZ RUN4

04A7   2A F9 9F                        LD HL,(START)
04AA   23                              INC HL              ; Check end code FFFFH
04AB   7C                              LD A,H
04AC   B5                              OR L
04AD   CA 00 00                        JP Z,MON
04B0   CD C1 03                        CALL SHOW           ; Transfer
04B3   18 C2                           JR RUN

                        ;************ READ 1 BYTE ************
                        ; Read 1 byte from tape
                        ; reg ADE

04B5   CD D3 04         RDBYTE:         CALL RDBIT          ; Read Start Bit
04B8   DA 26 05                        JP C,ERR00

04BB   1E 08                           LD E,08H
04BD   D5               RDBYT1:         PUSH DE
04BE   CD D3 04                        CALL RDBIT
04C1   D1                              POP DE
04C2   CB 1A                           RR D
04C4   1D                              DEC E
04C5   20 F6                           JR NZ,RDBYT1

04C7   D5                              PUSH DE             ; Read Stop Bit
04C8   CD D3 04                        CALL RDBIT
04CB   D1                              POP DE
04CC   D2 29 05                        JP NC,ERR01

04CF   7A                              LD A,D
04D0   D3 41                           OUT (SEGM),A
04D2   C9                              RET

                        ;************ READ 1 BIT *************
                        ; Read 1 bit from tape
                        ; reg BCD

04D3   01 00 00         RDBIT:          LD BC,0
04D6   CD EE 04         RDBIT1:         CALL CYCLE
04D9   14                              INC D
04DA   15                              DEC D
04DB   C2 2D 05                        JP NZ,ERR02
04DE   38 06                           JR C,RDBIT2
04E0   0D                              DEC C
04E1   0D                              DEC C
04E2   CB C0                           SET 0,B
04E4   18 F0                           JR RDBIT1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
04E6   0C              RDBIT2:         INC C
04E7   CB 40                           BIT 0,B
04E9   28 EB                           JR Z,RDBIT1
04EB   CB 11                           RL C
04ED   C9                              RET

04EE   11 00 00        CYCLE:          LD DE,0
04F1   DB 42           CYCLE1:         IN A,(DIGIT)
04F3   CB 7F                           BIT 7,A
04F5   13                              INC DE
04F6   20 F9                           JR NZ,CYCLE1
04F8   DB 42           CYCLE2:         IN A,(DIGIT)
04FA   CB 7F                           BIT 7,A
04FC   13                              INC DE
04FD   28 F9                           JR Z,CYCLE2
04FF   7B                              LD A,E
0500   FE 3A                           CP 3AH
0502   C9                              RET

                       ;************** PRINT **************
                       ; Display message on 7-Segment
                       ; in HL = Address of message
                       ;    C  = Digit
                       ; reg BCDEHL

0503   C5              PRINT:          PUSH BC
0504   11 E3 9F                       LD DE,DISPX
0507   06 00                          LD B,0
0509   ED B0                          LDIR
050B   C1                              POP BC
050C   CD 56 02                       CALL SCAND
050F   C9                              RET

                       ;************ ERROR PRINT ************
                       ; Display error message
                       ; in A = Error code
                       ; reg ABCHL

0510   32 ED 9F        ERRP:           LD (DISPY+2),A
0513   CD 26 01                       CALL H2SEG
0516   21 20 05                       LD HL,ERRTAB
0519   01 04 1B                       LD BC,1B04H
051C   CD 03 05                       CALL PRINT
051F   C9                              RET
0520   79 50 50 00 00  ERRTAB         DB 79H,50H,50H,00,00,00
0525   00

                       ;************ ERROR DISPLAY *********
                       ; Display error message
                       ; reg A

0526   AF              ERR00:          XOR A
0527   18 0E                          JR ERRST
0529   3E 01           ERR01:          LD A,1
052B   18 0A                          JR ERRST
052D   3E 02           ERR02:          LD A,2
052F   18 06                          JR ERRST
0531   3E 03           ERR03:          LD A,3
0533   18 02                          JR ERRST
0535   3E 04           ERR04:          LD A,4
0537   CD 10 05        ERRST:          CALL ERRP
053A   CD 67 00        ERRS1:          CALL SCAN
053D   18 FB                          JR ERRS1

                       ;********** DESIGN MOVEMENT ********
                       ; Design movement of robot
                       ; reg ABCDEHL

053F   CD F9 03        DSIGN:          CALL OUTPE          ; Clear port
0542   AF                             XOR A
0543   D3 41                          OUT (SEGM),A
0545   CD F1 01        DSIG1:          CALL CLR0
0548   67                             LD H,A
0549   6F                             LD L,A
054A   22 EE 9F                       LD (MEMY),HL

054D   2A EE 9F        DSIG2:          LD HL,(MEMY)
0550   22 EB 9F                       LD (DISPY),HL
0553   CD 26 01                       CALL H2SEG
0556   21 E4 9F                       LD HL,DISPX+1
0559   CB FE                          SET 7,(HL)
055B   23                             INC HL
055C   23                             INC HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
055D   CB FE                        SET 7,(HL)
055F   CD 67 00          DSIG21:    CALL SCAN
0562   06 00                       LD B,0
0564   FE 15                       CP 15H              ; Clear
0566   28 D7                       JR Z,DSIGN
0568   FE 13                       CP 13H              ; Next
056A   28 4B                       JR Z,DSIG5
056C   FE 11                       CP 11H              ; Restore
056E   28 2D                       JR Z,DSIG4
0570   FE 80                       CP 80H              ; Extention  check
0572   38 04                       JR C,DSIG3

0574   06 10                       LD B,10H
0576   E6 7F                       AND 7FH
0578   FE 10            DSIG3:     CP 10H
057A   30 E3                       JR NC,DSIG21
057C   80                          ADD A,B
057D   FE 18                       CP 18H
057F   30 DE                       JR NC,DSIG21

0581   87                          ADD A,A             ; Look up table
0582   4F                          LD C,A
0583   06 00                       LD B,0
0585   21 C3 05                    LD HL,PORTAB
0588   09                          ADD HL,BC
0589   56                          LD D,(HL)           ; Load data
058A   23                          INC HL
058B   4E                          LD C,(HL)           ; Load port number
058C   7A                          LD A,D
058D   CD 04 04                    CALL OUTPC          ; Transfer
0590   21 ED 9F                    LD HL,DISPY+2
0593   3E 2B                       LD A,2BH
0595   91                          SUB C
0596   4F                          LD C,A
0597   09                          ADD HL,BC
0598   7E                          LD A,(HL)
0599   AA                          XOR D
059A   77                          LD (HL),A           ; Update display
059B   18 B0                       JR DSIG2

059D   01 29 03          DSIG4:     LD BC,0329H         ; Restore movement
05A0   21 EF 9F                    LD HL,MEMY+1
05A3   7E                DSIG41:    LD A,(HL)
05A4   CD 04 04                    CALL OUTPC
05A7   0C                          INC C
05A8   2B                          DEC HL
05A9   10 F8                       DJNZ DSIG41

05AB   21 F2 9F                    LD HL,MEMZ          ; Restore display
05AE   11 ED 9F                    LD DE,DISPY+2
05B1   0E 03                       LD C,03H
05B3   ED B0                       LDIR
05B5   18 96                       JR DSIG2

05B7   21 ED 9F          DSIG5:     LD HL,DISPY+2       ; Save display
05BA   11 F2 9F                    LD DE,MEMZ
05BD   0E 03                       LD C,3
05BF   ED B0                       LDIR
05C1   18 82                       JR DSIG1

05C3   04 2A 08 29       PORTAB     DB 04H,2AH,08H,29H  ; H_R,B_R,B_L
05C7   20 29 80 2A                  DB 20H,29H,80H,2AH  ; B,H_L
05CB   04 29 01 2A                  DB 04H,29H,01H,2AH  ; A_R,E_R
05CF   20 2A 40 2A                  DB 20H,2AH,40H,2AH  ; E_L,A_L
05D3   02 2A 02 29                  DB 02H,2AH,02H,29H  ; Up_AR,N_R
05D7   01 29 80 29                  DB 01H,29H,80H,29H  ; N_L,Up_AL
05DB   10 29 10 2A                  DB 10H,29H,10H,2AH  ; Sh_R,Head
05DF   40 29 08 2A                  DB 40H,29H,08H,2AH  ; M,Sh_L

05E3   01 2B 02 2B                  DB 01H,2BH,02H,2BH  ; Extention
05E7   04 2B 08 2B                  DB 04H,2BH,08H,2BH
05EB   10 2B 20 2B                  DB 10H,2BH,20H,2BH
05EF   40 2B 80 2B                  DB 40H,2BH,80H,2BH

                         ;************** LIST DATA ************
                         ; List data by group
                         ; reg ABDEHL,IX

05F3   06 77             LIST:      LD B,77H
05F5   2A FB 9F                     LD HL,(DEST)
05F8   CD 03 02                     CALL IN4
05FB   22 FB 9F                     LD (DEST),HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
05FE   E5                              PUSH HL
05FF   DD E1                           POP IX
0601   21 E0 9F                        LD HL,SYSFLG
0604   CB CE                           SET 1,(HL)          ; Display 3 byte first
0606   DD 7E 02          LIST0:        LD A,(IX+02)
0609   32 ED 9F                        LD (DISPY+2),A
060C   DD 66 00          LIST01:       LD H,(IX+00)
060F   DD 6E 01                        LD L,(IX+01)
0612   22 EB 9F                        LD (DISPY),HL
0615   CD 26 01                        CALL H2SEG

0618   21 E6 9F                        LD HL,DISPX+3
061B   CB FE                           SET 7,(HL)
061D   2B                              DEC HL
061E   2B                              DEC HL
061F   CB FE                           SET 7,(HL)

0621   11 E0 9F                        LD DE,SYSFLG
0624   1A                              LD A,(DE)
0625   E6 02                           AND 02H
0627   20 07                           JR NZ,LIST1
0629   CB BE                           RES 7,(HL)
062B   67                              LD H,A
062C   6F                              LD L,A
062D   22 E7 9F                        LD (DISPX+4),HL

0630   CD 67 00          LIST1:        CALL SCAN
0633   11 02 00                        LD DE,2
0636   21 E0 9F                        LD HL,SYSFLG
0639   FE 8F                           CP 8FH              ; Edit
063B   28 18                           JR Z,LIST22
063D   FE 13                           CP 13H              ; INC
063F   28 07                           JR Z,LIST2
0641   FE 11                           CP 11H              ; DEC
0643   20 EB                           JR NZ,LIST1

0645   11 FD FF                        LD DE,-3
0648   7E                LIST2:        LD A,(HL)
0649   CB 4F                          BIT 1,A
064B   28 01                           JR Z,LIST21
064D   1C                              INC E
064E   DD 19              LIST21:      ADD IX,DE
0650   EE 02                           XOR 2               ; Toggle display
0652   77                              LD (HL),A
0653   18 B1                           JR LIST0

0655   DD E5              LIST22:      PUSH IX             ; Save address
0657   E1                              POP HL
0658   22 FB 9F                        LD (DEST),HL
065B   C3 7B 02                        JP DATA12

                          ;************ SAVE DATA *************
                          ; Save data to tape recorder
                          ; in (START),(FINAL) = Address
                          ; reg ABCDEHL

065E   CD 26 02           SAVE:        CALL INSF
0661   2A F7 9F                        LD HL,(FINAL)
0664   ED 5B F9 9F                     LD DE,(START)
0668   AF                              XOR A
0669   ED 52                           SBC HL,DE
066B   23                              INC HL
066C   ED 53 EE 9F                     LD (MEMY),DE        ; Start addr
0670   22 F0 9F                        LD (MEMX),HL        ; Lenght

0673   01 00 15                        LD BC,1500H
0676   CD 49 04           SAVE1:       CALL TONE1K         ; Start sync
0679   0B                              DEC BC
067A   78                              LD A,B
067B   B1                              OR C
067C   20 F8                           JR NZ,SAVE1

067E   21 EE 9F                        LD HL,MEMY
0681   1E 04                           LD E,4
0683   56                SAVE2:       LD D,(HL)           ; Save address
0684   CD 16 04                        CALL WRBYTE
0687   23                              INC HL
0688   1D                              DEC E
0689   20 F8                           JR NZ,SAVE2

068B   2A EE 9F                        LD HL,(MEMY)        ; Start addr
068E   ED 4B F0 9F                     LD BC,(MEMX)        ; Lenght
0692   56                SAVE3:       LD D,(HL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0693   7B                          LD A,E
0694   82                          ADD A,D
0695   5F                          LD E,A              ; Check sum
0696   C5                          PUSH BC
0697   CD 16 04                    CALL WRBYTE
069A   C1                          POP BC
069B   23                          INC HL
069C   0B                          DEC BC
069D   78                          LD A,B
069E   B1                          OR C
069F   20 F1                       JR NZ,SAVE3

06A1   7B                          LD A,E
06A2   2F                          CPL
06A3   57                          LD D,A
06A4   CD 16 04                    CALL WRBYTE         ; Save check sum

06A7   06 80                       LD B,80H
06A9   CD 60 04           SAVE4:   CALL TONE2K         ; End sync
06AC   10 FB                       DJNZ SAVE4
06AE   C3 00 00                    JP MON

                          ;*********** LOAD DATA *************
                          ; Load data from tape recorder
                          ; reg ABCDHL

06B1   21 00 00           LOAD:    LD HL,0
06B4   22 FB 9F                    LD (DEST),HL
06B7   06 6D                       LD B,6DH
06B9   CD 03 02                    CALL IN4
06BC   22 FB 9F                    LD (DEST),HL
06BF   AF                          XOR A
06C0   D3 41                       OUT (SEGM),A
06C2   3E 0E                       LD A,0EH
06C4   D3 42                       OUT (DIGIT),A

06C6   CD EE 04           LOAD1:   CALL CYCLE          ; Wait for first signal
06C9   38 FB                       JR C,LOAD1

06CB   3E 18                       LD A,18H
06CD   D3 41                       OUT (SEGM),A        ; Sync display
06CF   01 00 10                    LD BC,1000H
06D2   CD EE 04           LOAD2:   CALL CYCLE          ; Make sure
06D5   38 FB                       JR C,LOAD2
06D7   0B                          DEC BC
06D8   78                          LD A,B
06D9   B1                          OR C
06DA   20 F6                       JR NZ,LOAD2

06DC   CD EE 04           LOAD3:   CALL CYCLE          ; Wait for start bit
06DF   30 FB                       JR NC,LOAD3

06E1   21 EE 9F                    LD HL,MEMY          ; Load start & lenght
06E4   06 04                       LD B,4
06E6   C5                 LOAD4:   PUSH BC
06E7   CD B5 04                    CALL RDBYTE
06EA   C1                          POP BC
06EB   72                          LD (HL),D
06EC   23                          INC HL
06ED   10 F7                       DJNZ LOAD4

06EF   2A FB 9F                    LD HL,(DEST)
06F2   7C                          LD A,H
06F3   B5                          OR L
06F4   28 03                       JR Z,LOAD5
06F6   22 EE 9F                    LD (MEMY),HL
06F9   2A EE 9F           LOAD5:   LD HL,(MEMY)        ; Start addr
06FC   ED 4B F0 9F                 LD BC,(MEMX)        ; Lenght
0700   AF                          XOR A
0701   32 F0 9F                    LD (MEMX),A         ; Check sum

0704   C5                 LOAD6:   PUSH BC             ; Load data
0705   CD B5 04                    CALL RDBYTE
0708   C1                          POP BC
0709   3A F0 9F                    LD A,(MEMX)
070C   82                          ADD A,D
070D   32 F0 9F                    LD (MEMX),A
0710   72                          LD (HL),D
0711   23                          INC HL
0712   0B                          DEC BC
0713   78                          LD A,B
0714   B1                          OR C
0715   20 ED                       JR NZ,LOAD6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0717   AF                         XOR A
0718   D3 41                      OUT (SEGM),A
071A   CD B5 04                   CALL RDBYTE
071D   3A F0 9F                   LD A,(MEMX)
0720   2F                         CPL
0721   BA                         CP D                ; Check sum
0722   C2 35 05                   JP NZ,ERR04

0725   C3 00 00                   JP MON

                        ;************** PRINT ***************
                        ; Print data to Printer
                        ; reg ABCDEHL

0728   CD 26 02         PRN:      CALL INSF

072B   3E 0E                      LD A,0EH            ; Select I/O mode
072D   D3 42                      OUT (42H),A
072F   3E 40                      LD A,40H
0731   D3 41                      OUT (PRNSTB),A      ; STB high

0733   21 FB 07                   LD HL,HEAD1TAB      ; Header 1
0736   CD 99 07                   CALL PRN_S

0739   21 1C 08                   LD HL,STARTAB       ; Start address
073C   CD 99 07                   CALL PRN_S
073F   2A F9 9F                   LD HL,(START)
0742   CD A4 07                   CALL PRN_W

0745   21 2F 08                   LD HL,FNALTAB       ; Final address
0748   CD 99 07                   CALL PRN_S
074B   2A F7 9F                   LD HL,(FINAL)
074E   CD A4 07                   CALL PRN_W

0751   21 43 08                   LD HL,HEAD2TAB      ; Header 2
0754   CD 99 07                   CALL PRN_S

0757   2A F9 9F         PRN_L:    LD HL,(START)       ; Data
075A   1E 20             PRN_L1:   LD E,' '
075C   CD D1 07                   CALL PRN_C
075F   CD A4 07                   CALL PRN_W
0762   1E 09                      LD E,09H            ; Horizontal tab
0764   CD D1 07                   CALL PRN_C
0767   06 03                      LD B,03H
0769   C5               PRN_L3:   PUSH BC
076A   7E                         LD A,(HL)
076B   CD AD 07                   CALL PRN_H
076E   1E 20                      LD E,' '
0770   CD D1 07                   CALL PRN_C
0773   23                         INC HL
0774   C1                         POP BC
0775   10 F2                      DJNZ PRN_L3
0777   06 02                      LD B,02H
0779   C5               PRN_L4:   PUSH BC
077A   7E                         LD A,(HL)
077B   CD AD 07                   CALL PRN_H
077E   23                         INC HL
077F   C1                         POP BC
0780   10 F7                      DJNZ PRN_L4

0782   1E 0D                      LD E,0DH
0784   CD D1 07                   CALL PRN_C
0787   1E 0A                      LD E,0AH
0789   CD D1 07                   CALL PRN_C

078C   ED 5B F7 9F                LD DE,(FINAL)       ; Check end
0790   EB                         EX DE,HL
0791   ED 52                      SBC HL,DE
0793   EB                         EX DE,HL
0794   30 C4                      JR NC,PRN_L1

0796   C3 00 00                   JP MON

                        ;********** PRINT STRINGS **********
                        ; Print strings
                        ; in HL = Address of string
                        ; reg AE

0799   5E               PRN_S:    LD E,(HL)
079A   7B                         LD A,E
079B   FE 24                      CP '$'              ; Check end charactor
079D   C8                         RET Z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
079E   CD D1 07                        CALL PRN_C
07A1   23                              INC HL
07A2   18 F5                           JR PRN_S

                        ;************ PRINT WORD *************
                        ; Print HEX in word
                        ; in HL = HEX in word
                        ; reg AHL

07A4   7C               PRN_W:         LD A,H
07A5   CD AD 07                        CALL PRN_H
07A8   7D                              LD A,L
07A9   CD AD 07                        CALL PRN_H
07AC   C9                              RET

                        ;************ PRINT HEX **************
                        ; Print HEX number
                        ; in A = HEX
                        ; reg DE

07AD   CD B8 07         PRN_H:         CALL H2ASC
07B0   CD D1 07                        CALL PRN_C
07B3   5A                              LD E,D
07B4   CD D1 07                        CALL PRN_C
07B7   C9                              RET

                        ;****** CONVERT HEX TO ASCII *******
                        ; Convert HEX to ASCII code
                        ; in A = HEX
                        ; out DE = ASCII (in reverse order)
                        ; reg ADE

07B8   5F               H2ASC:         LD E,A
07B9   CD C7 07                        CALL H2ASC1
07BC   57                              LD D,A                  ; Low byte
07BD   7B                              LD A,E
07BE   0F                              RRCA
07BF   0F                              RRCA
07C0   0F                              RRCA
07C1   0F                              RRCA
07C2   CD C7 07                        CALL H2ASC1
07C5   5F                              LD E,A                  ; High byte
07C6   C9                              RET
07C7
07C7   E6 0F            H2ASC1:        AND A,0FH
07C9   C6 30                           ADD A,30H
07CB   FE 3A                           CP 3AH
07CD   D8                              RET C
07CE   C6 07                           ADD A,07H
07D0   C9                              RET

                        ;********* PRINT CHARACTOR *********
                        ; Print 1 charactor
                        ; in E = ASCII To Print
                        ; reg ABE

07D1   CD EC 07         PRN_C:         CALL BUSY
07D4   7B                              LD A,E
07D5   D3 40                           OUT (PRNDAT),A
07D7   CD F6 07                        CALL DEL1
07DA   3E 00                           LD A,0
07DC   D3 41                           OUT (PRNSTB),A
07DE   CD F6 07                        CALL DEL1
07E1   3E 40                           LD A,40H
07E3   D3 41                           OUT (PRNSTB),A
07E5   CD F6 07                        CALL DEL1
07E8   AF                              XOR A
07E9   D3 40                           OUT (PRNDAT),A
07EB   C9                              RET

07EC   DB 42            BUSY:          IN A,(PRNBUSY)          ; Read busy
07EE   CB 77                           BIT 6,A
07F0   20 FA                           JR NZ,BUSY
07F2   CD F6 07                        CALL DEL1
07F5   C9                              RET

07F6   06 10            DEL1:          LD B,10H
07F8   10 FE            DEL11:         DJNZ DEL11
07FA   C9                              RET

07FB   52 4F 42 4F 54   HEAD1TAB       DB 'ROBOT CONTROLLER DATA
                                           PRINTING',0AH,0DH,'$'
8000   20 43 4F 4E 54
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0805   52 4F 4C 4C 45
080A   52 20 44 41 54
080F   41 20 50 52 49
0814   4E 54 49 4E 47
0819   0A 0D 24
081C   0A 0D 20 53 54    STARTAB         DB 0AH,0DH,' START ADDRESS: $'
0821   41 52 54 20 41
0826   44 44 52 45 53
082B   53 3A 20 24
082F   48 0A 0D 20 46    FNALTAB         DB 'H',0AH,0DH,' FINAL ADDRESS: $'
0834   49 4E 41 4C 20
0839   41 44 44 52 45
083E   53 53 3A 20 24
0843   48 0A 0D          HEAD2TAB        DB 'H',0AH,0DH
0846   1B 2D 01                          DB 1BH,2DH,01H          ;underline
0849   0A 0D 41 44 44                    DB 0AH,0DH,'ADDRESS   DATA   DELAY',0AH,0DH
084E   52 45 53 53 20
0853   20 20 44 41 54
0858   41 20 20 20 44
085D   45 4C 41 59 0A
0862   0D
0863   1B 2D 00                          DB 1BH,2DH,00H          ;cancel underline
0866   0A 0D 24                          DB 0AH,0DH,'$'

                       ;************* PC LINK **************
                       ; Recieve data from PC
                       ; ABCDEHL,IX

0869   06 5C             PCL:            LD B,5CH
086B   21 00 00                          LD HL,0
086E   CD 03 02                          CALL IN4
0871   22 FB 9F                          LD (DEST),HL
0874   21 00 00                          LD HL,0
0877   22 FB 9F                          LD (DEST),HL

087A   3E 06                             LD A,6
087C   D3 42                             OUT (DIGIT),A

087E   DB 42             PCL2:           IN A,(DIGIT)        ; Wait for signal
0880   CB 7F                             BIT 7,A
0882   20 FA                             JR NZ,PCL2
0884   CD 2E 09                          CALL PCLBY
0887   3E 3A                             LD A,':'
0889   BA                                CP D
088A   C2 31 05                          JP NZ,ERR03         ; Missing error
088D   0E 00                             LD C,0
088F   CD 09 09                          CALL PCLBY2
0892   42                                LD B,D
0893   CD 09 09                          CALL PCLBY2
0896   D5                                PUSH DE
0897   CD 09 09                          CALL PCLBY2
089A   E1                                POP HL
089B   6A                                LD L,D              ; Load ADDR of first byte
089C   E5                                PUSH HL
089D   DD E1                             POP IX
089F   CD 09 09                          CALL PCLBY2
08A2   ED 5B FB 9F                       LD DE,(DEST)
08A6   DD 19                             ADD IX,DE           ; Add offset
08A8   DD 22 F9 9F                       LD (START),IX

08AC   CD 09 09           PCL3:          CALL PCLBY2         ; Load data
08AF   7A                                LD A,D
08B0   D3 41                             OUT (SEGM),A        ; Show to LED
08B2   DD 77 00                          LD (IX+0),A
08B5   DD 23                             INC IX
08B7   10 F3                             DJNZ PCL3

08B9   79                                LD A,C
08BA   ED 44                             NEG
08BC   47                                LD B,A
08BD   CD 09 09                          CALL PCLBY2
08C0   7A                                LD A,D
08C1   B8                                CP B                ; Check sum
08C2   C2 35 05                          JP NZ,ERR04         ; Check sum error

08C5   CD 2E 09                          CALL PCLBY          ; LF
08C8   CD 2E 09                          CALL PCLBY          ; CR
08CB   CD 2E 09                          CALL PCLBY          ; Start charactor
08CE   3E 3A                             LD A,':'
08D0   BA                                CP D
08D1   C2 31 05                          JP NZ,ERR03
08D4   0E 00                             LD C,0
08D6   CD 09 09                          CALL PCLBY2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

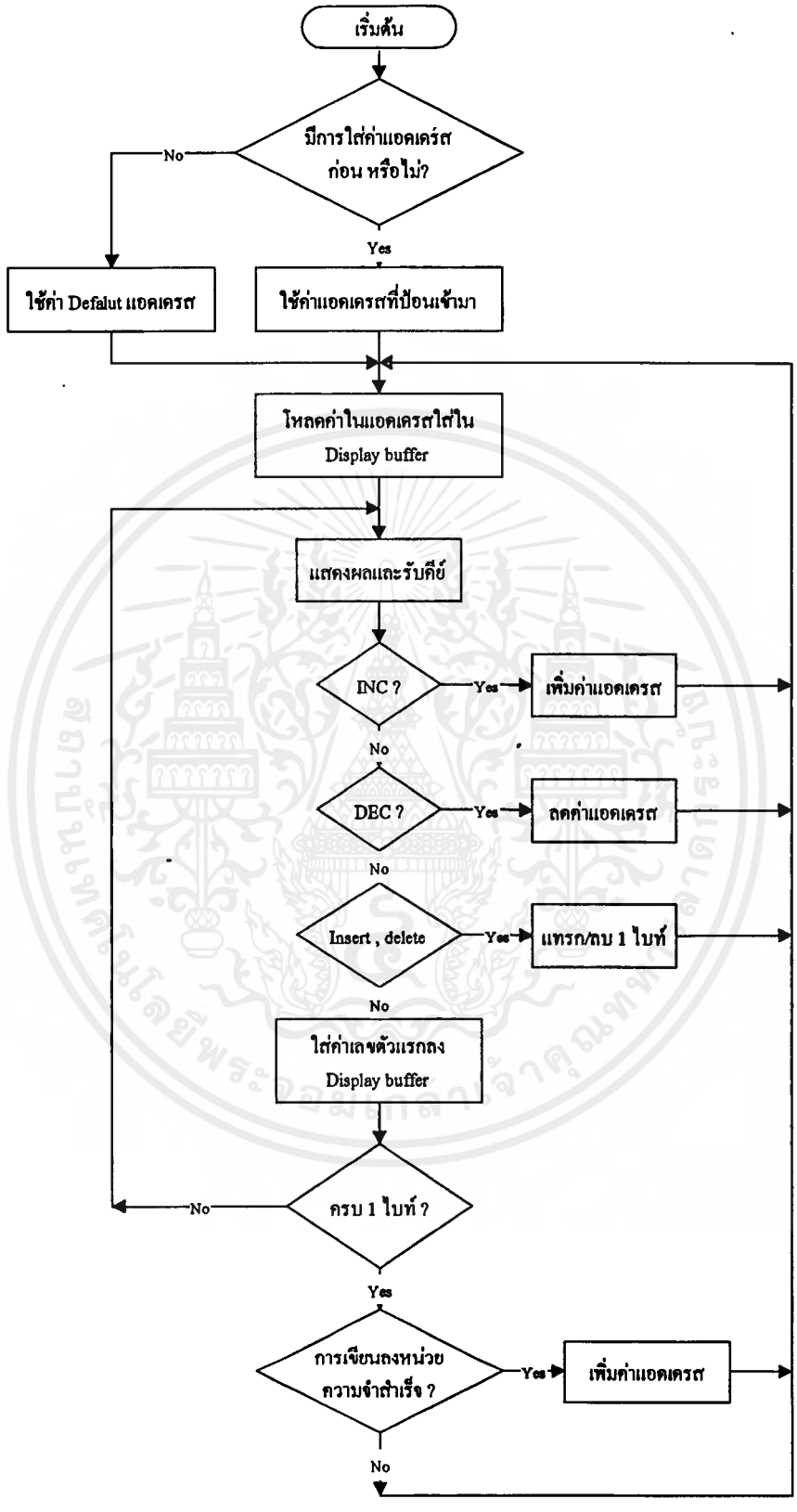
```
08D9   7A                              LD A,D
08DA   FE 00                           CP 0                    ; Check end
08DC   28 18                           JR Z,PCLEND
08DE   47                              LD B,A

08DF   CD 09 09                        CALL PCLBY2
08E2   D5                              PUSH DE
08E3   CD 09 09                        CALL PCLBY2
08E6   E1                              POP HL
08E7   6A                              LD L,D
08E8   E5                              PUSH HL
08E9   DD E1                           POP IX
08EB   CD 09 09                        CALL PCLBY2
08EE   ED 5B FB 9F                     LD DE,(DEST)
08F2   DD 19                           ADD IX,DE
08F4   18 B6                           JR PCL3

08F6   06 04           PCLEND:         LD B,4                  ; End of data
08F8   CD 09 09        PCLEND1:        CALL PCLBY2
08FB   10 FB                           DJNZ PCLEND1
08FD   CD 2E 09                        CALL PCLBY              ; LF
0900   CD 2E 09                        CALL PCLBY              ; CR
0903   AF                              XOR A
0904   D3 41                           OUT (SEGM),A
0906   C3 00 00                        JP MON

                       ;************** PCLBY2 ***************
                       ; Read one byte in ASCII
                       ; out = D Data
                       ;     = C Checksum
                       ; reg = ACDEHL

0909   CD 2E 09        PCLBY2:         CALL PCLBY
090C   7A                              LD A,D
090D   FE 41                           CP 'A'
090F   38 02                           JR C,PCLBY21
0911   C6 09                           ADD A,9                 ; A-F
0913   E6 0F           PCLBY21:        AND 0FH
0915   07                              RLCA                    ; Shift to high digit
0916   07                              RLCA
0917   07                              RLCA
0918   07                              RLCA
0919   F5                              PUSH AF
091A   CD 2E 09                        CALL PCLBY
091D   7A                              LD A,D
091E   FE 41                           CP 'A'
0920   38 02                           JR C,PCLBY22
0922   C6 09                           ADD A,9
0924   E6 0F           PCLBY22:        AND 0FH
0926   57                              LD D,A
0927   F1                              POP AF
0928   B2                              OR D
0929   57                              LD D,A
092A   79                              LD A,C
092B   82                              ADD A,D                 ; Check sum
092C   4F                              LD C,A
092D   C9                              RET

                       ;************** PCLBY ****************
                       ; Read one byte from PC
                       ; out = D Data
                       ; reg = ADEHL

092E   21 00 00        PCLBY:          LD HL,0
0931   00              PCLBY1:         NOP
0932   00                             NOP
0933   2B                             DEC HL
0934   7C                             LD A,H
0935   B5                             OR L
0936   CA 2D 05                       JP Z,ERR02              ; Long time error
0939   DB 42                          IN A,(DIGIT)
093B   CB 7F                          BIT 7,A
093D   20 F2                          JR NZ,PCLBY1            ; Start bit
093F   00                             NOP
0940   00                             NOP
0941   00                             NOP
0942   00                             NOP
0943   CD 57 09                       CALL PCLDLY

0946   C5                             PUSH BC
0947   16 00                          LD D,0
0949   06 08                          LD B,8
094B   DB 42           PCLBY12:       IN A,(DIGIT)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
094D   17                              RLA
094E   CB 1A                          RR D
0950   CD 57 09                       CALL PCLDLY
0953   10 F6                          DJNZ PCLBY12
0955   C1                             POP BC
0956   C9                             RET

0957   3E 17           PCLDLY:        LD A,17H          ; One bit speed delay
0959   7F                             LD A,A
095A   3D              PCLDLY1:       DEC A
095B   20 FD                         JR NZ,PCLDLY1
095D   C9                             RET

9FE0                                  ORG RAM_AREA

9FE0                   SYSFLG         DS 1
9FE1                   KEYIN          DS 1
9FE2                   REPDLY         DS 1
9FE3                   DISPX          DS 8
9FEB                   DISPY          DS 3
9FEE                   MEMY           DS 2
9FF0                   MEMX           DS 2
9FF2                   MEMZ           DS 3
9FF5                   ADDR           DS 2
9FF7                   FINAL          DS 2
9FF9                   START          DS 2
9FFB                   DEST           DS 2
```

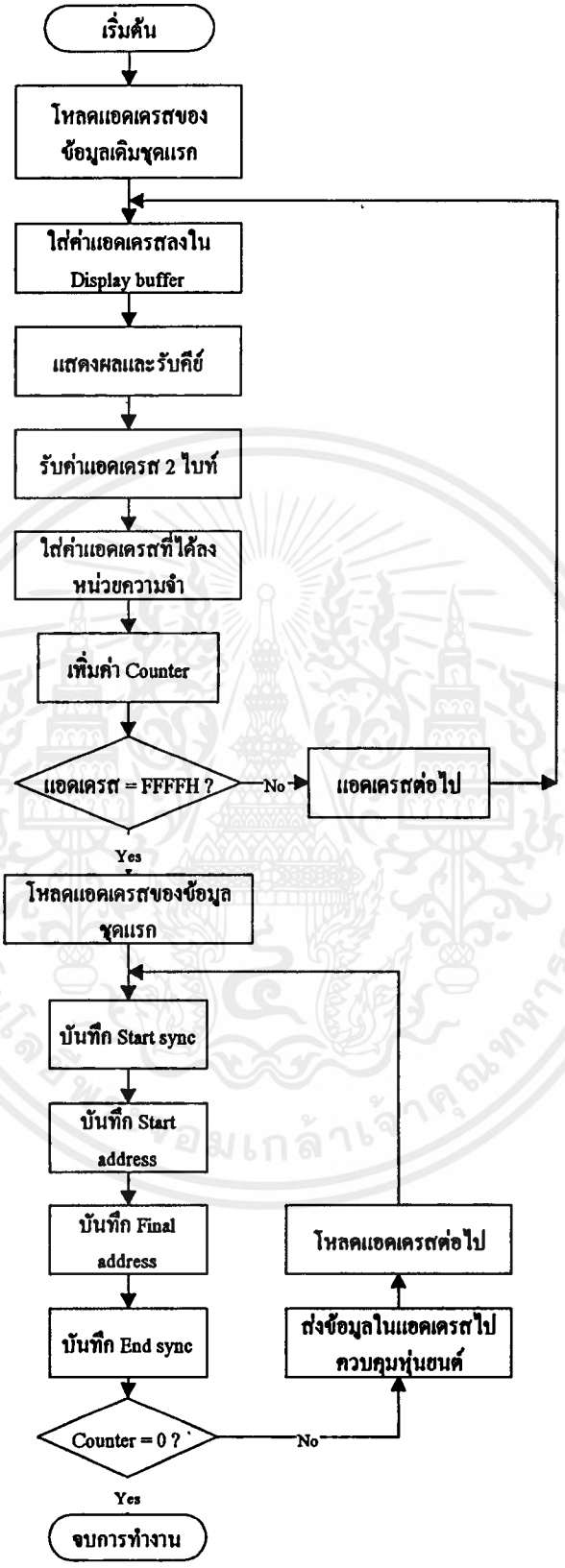
Lines Assembled : 1565 Assembly Errors : 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



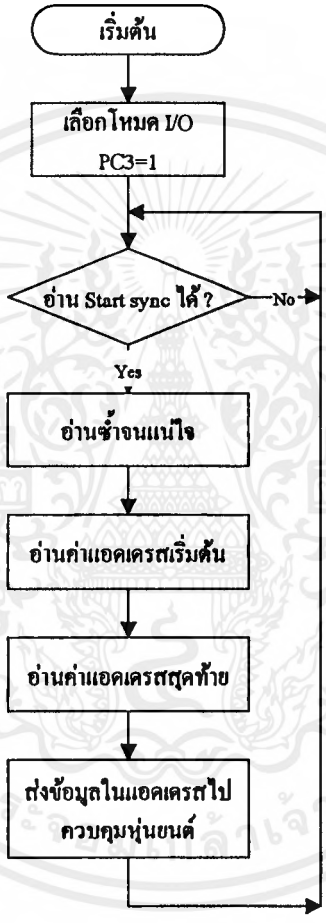
รูปแสดง Flowchart ของ โปรแกรม DATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



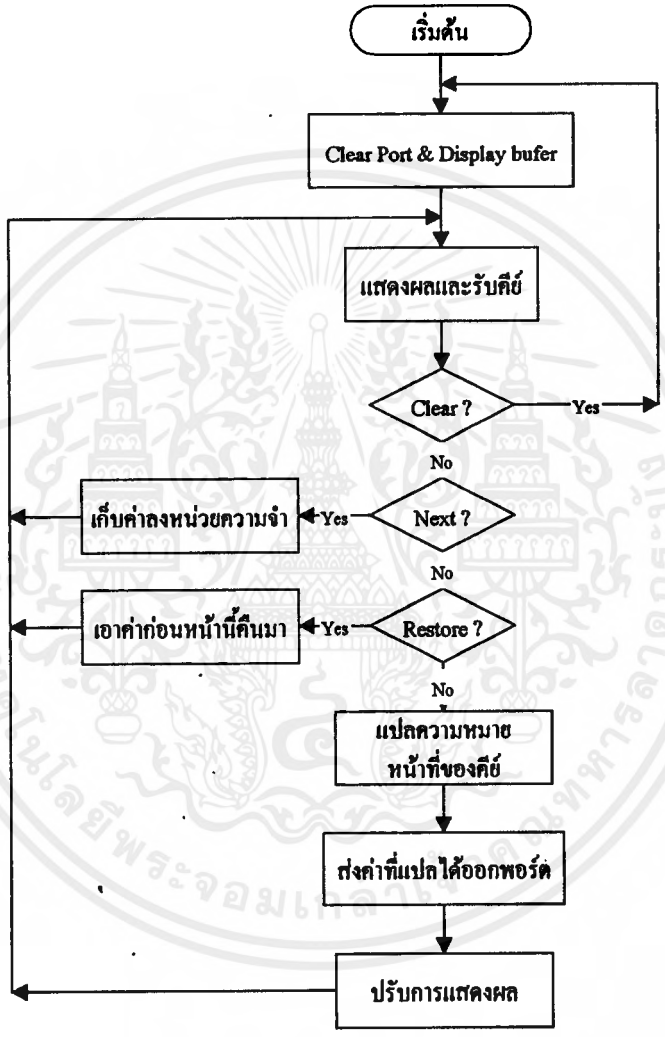
รูปแสดง Flowchart ของ โปรแกรม REC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



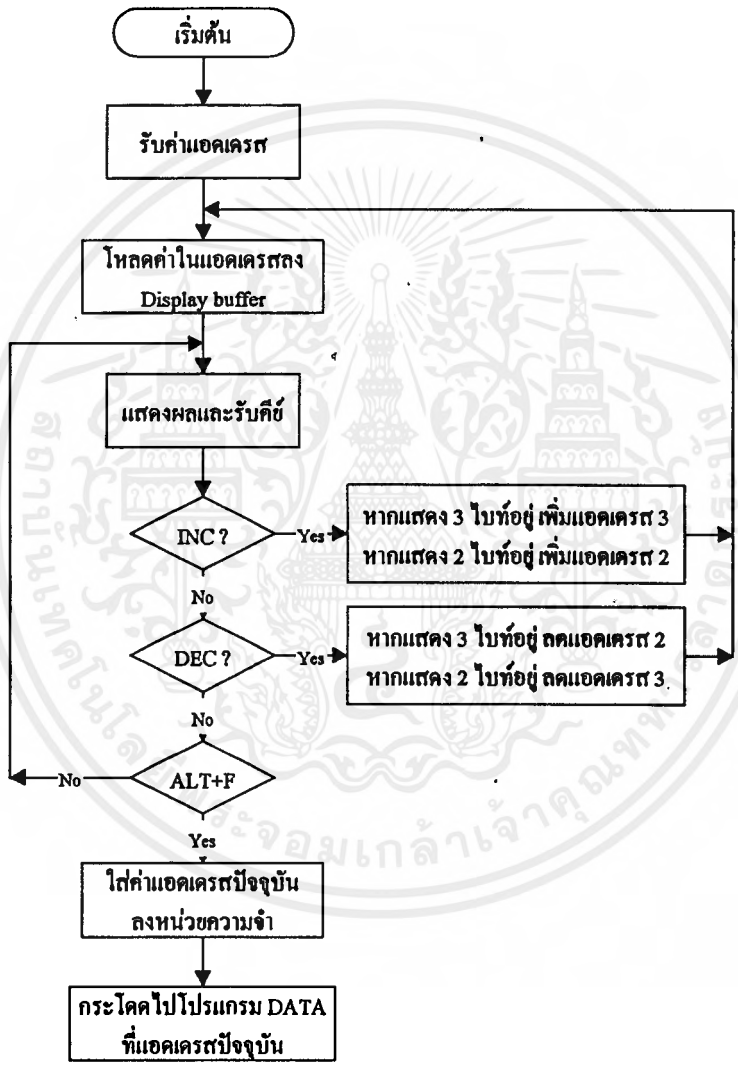
รูปแสดง Flowchart ของโปรแกรม RUN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



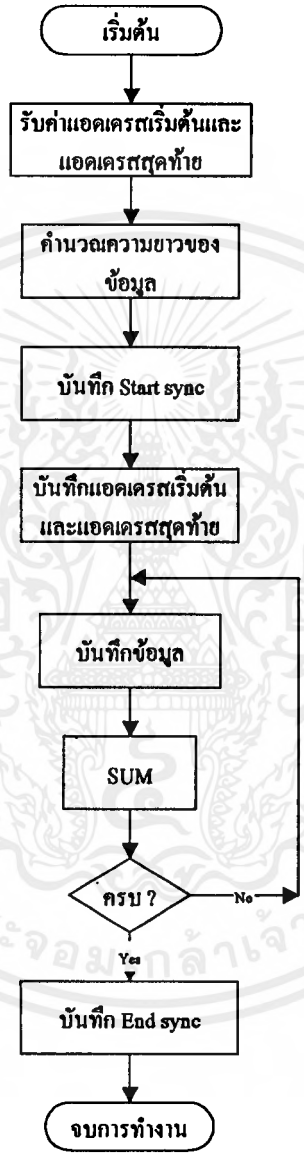
รูปแสดง Flowchart ของโปรแกรม DESIGN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



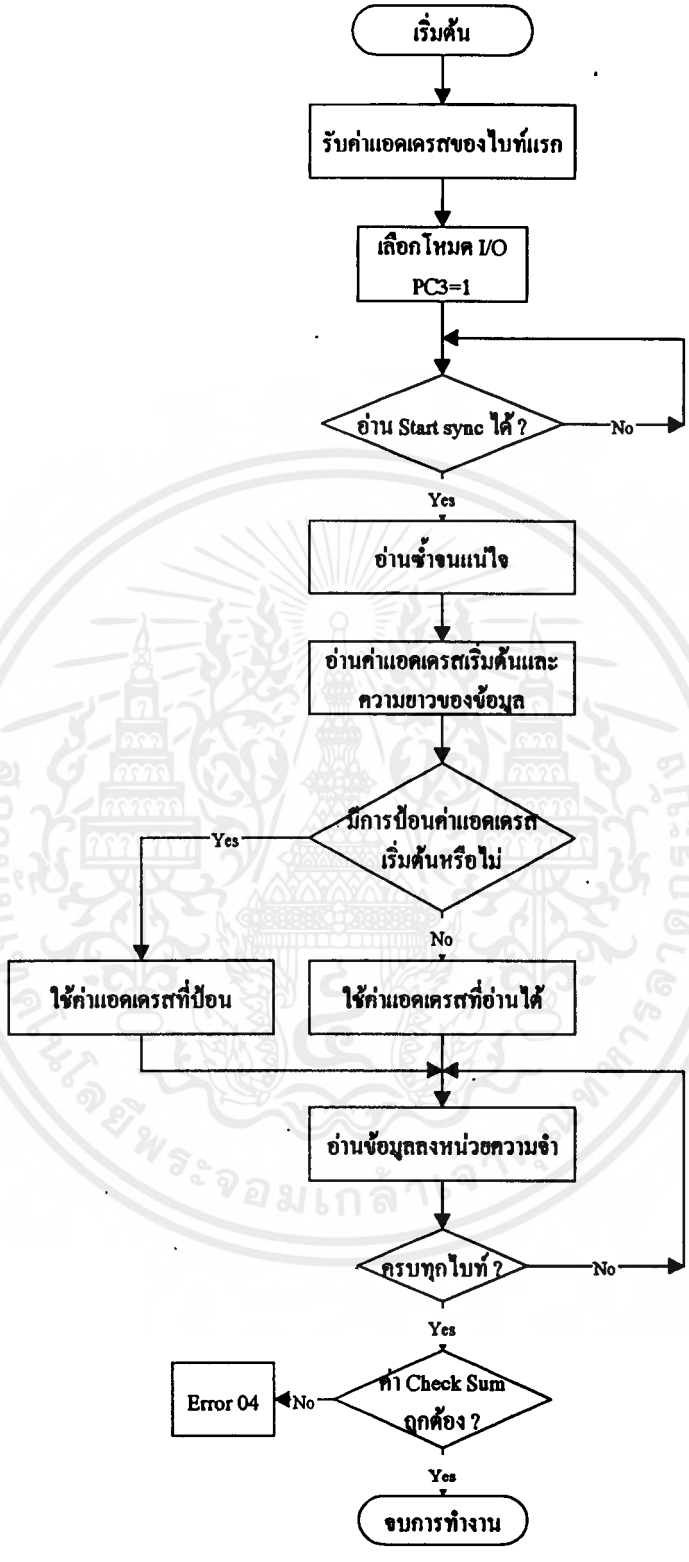
รูปแสดง Flowchart ของโปรแกรม LIST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



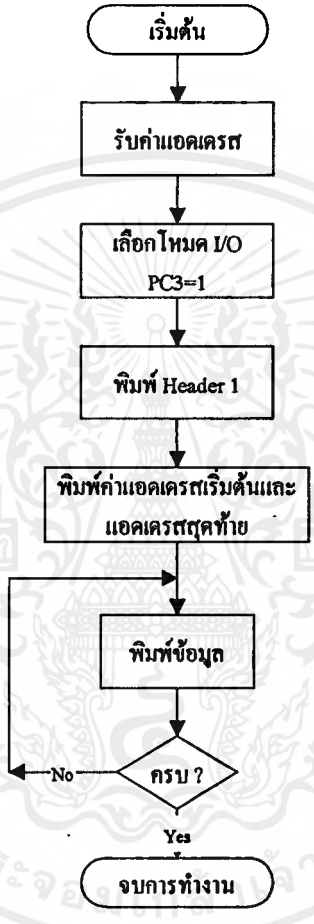
รูปแสดง Flowchart ของโปรแกรม SAVE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



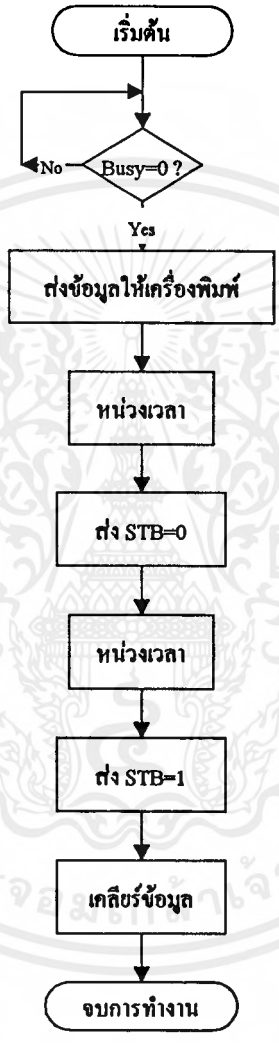
รูปแสดง Flowchart ของโปรแกรม LOAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



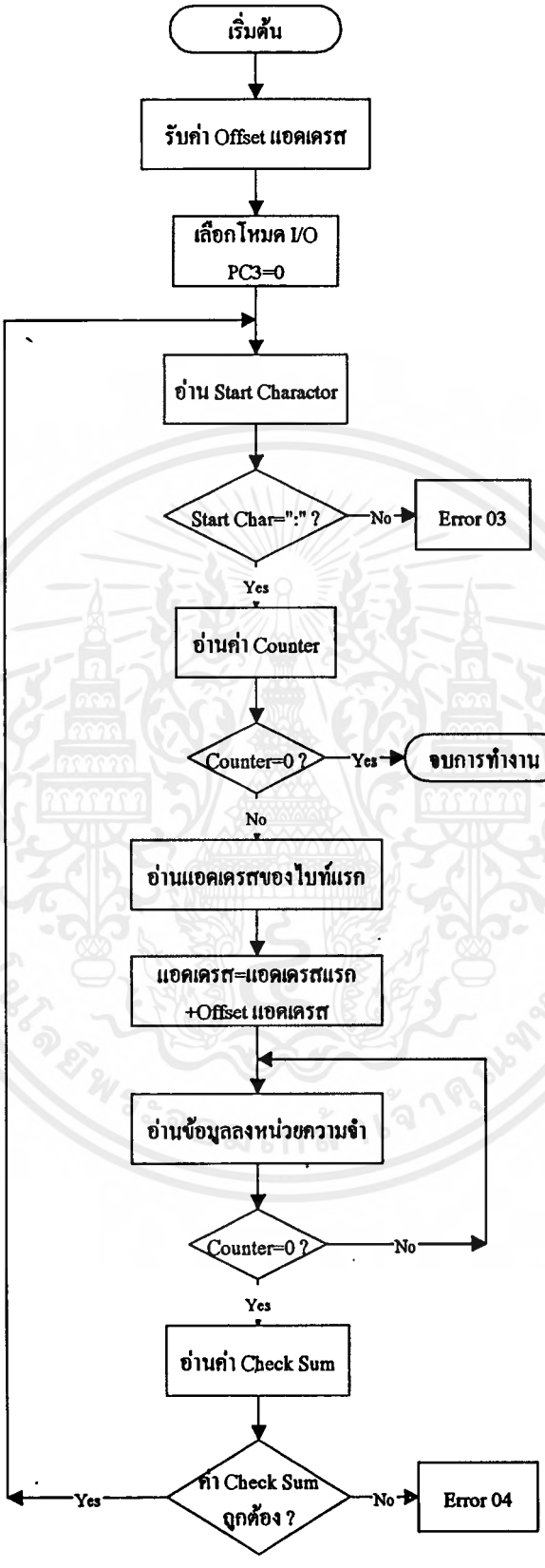
รูปแสดง Flowchart ของโปรแกรม PRINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



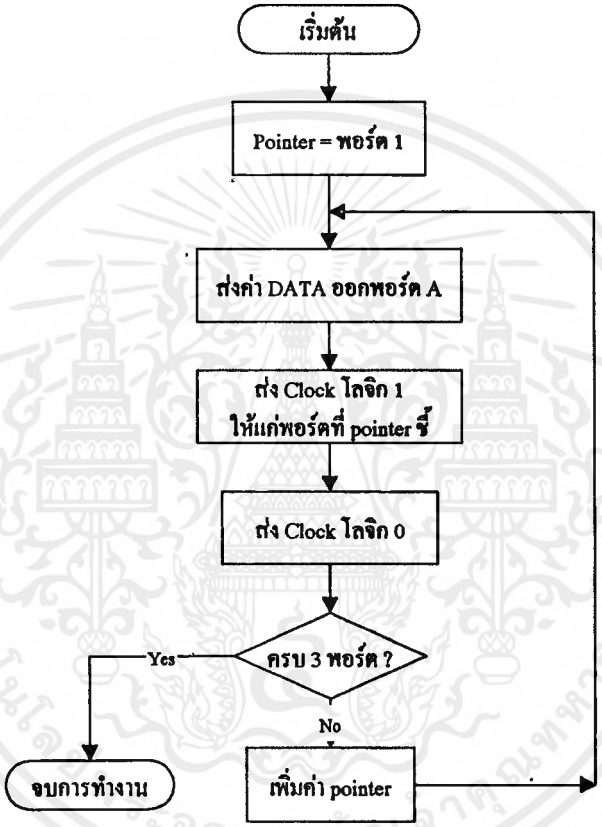
รูปแสดง Flowchart ของโปรแกรมย่อย PRN_C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



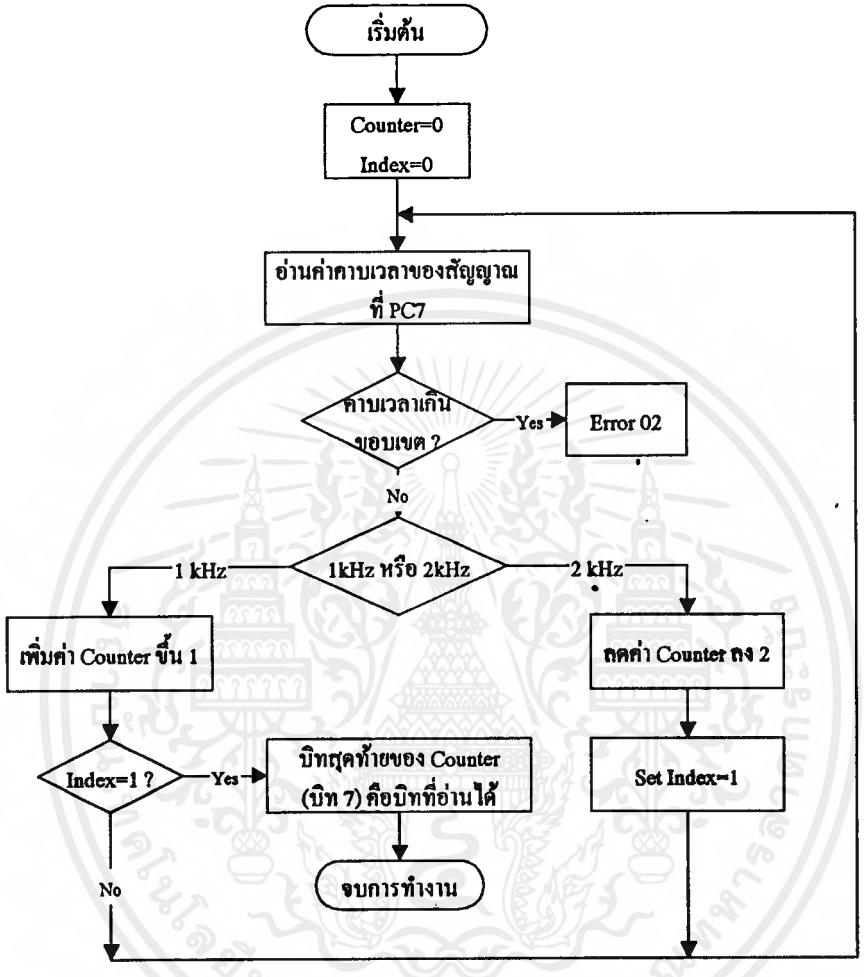
รูปแสดง Flowchart ของโปรแกรม PCLINK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง Flowchart ของโปรแกรมย่อย OUTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง Flowchart ของโปรแกรมย่อย RDBIT

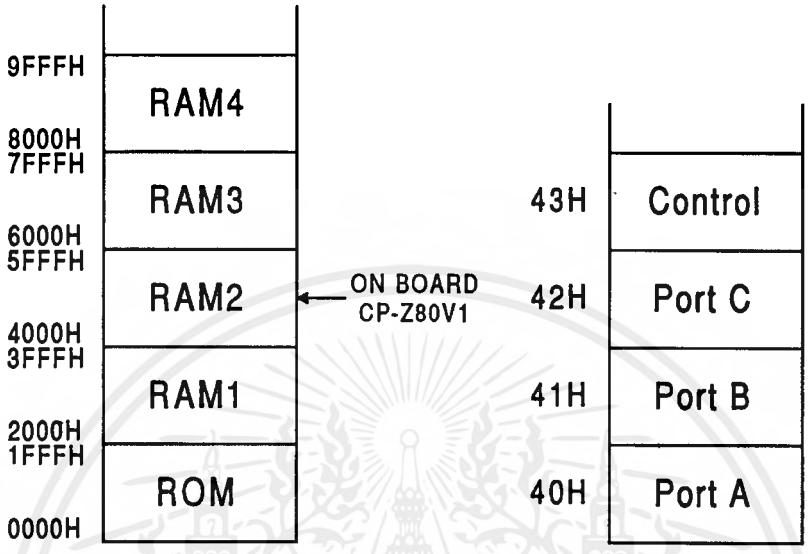
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
@echo off
echo Batch file Compile & Link version 3.0
if not exist x80.exe goto no_incl
if not exist link.exe goto no_incl
if "%1"=="" goto err0
for %%v in ( -m -M ) do if "%1"=="%%v" goto setmode
for %%v in ( -l -L ) do if "%2"=="%%v" goto link
for %%v in ( -t -T ) do if "%2"=="%%v" goto transfer
:complie
if not exist %1.asm goto err1
x80 %1.asm -d -l
if errorlevel 1 goto end
:link
if not exist %1.obj goto err2
echo %1 > dum.txt
for %%a in (1 2 3 4 5) do echo.>> dum.txt
echo     Link %1.obj into %1.hex
link <dum.txt >nul
del dum.txt
if errorlevel 1 goto end
for %%v in ( -t -T ) do if "%2"=="%%v" goto transfer
goto end
:transfer
if not exist %1.hex goto complie
echo     Transfer %1.hex to COM2
copy %1.hex com2
goto end
:setmode
mode com2:9600,n,8,1,p
goto end
:err0
echo.
echo No files exist
echo.
echo Usege -.
echo       %0 -m   :Initial mode COM2
echo  or   %0 file [-l -t]
echo         file  :filename only
echo         -l    :Link object file
echo         -t    :Transfer hex file to COM2
goto end
:no_incl
echo.
echo This program use -.
echo                    X80.EXE
echo                    LINK.EXE
goto end
:err1
echo.
echo %1.asm not found
goto end
:err2
echo.
echo %1.obj not found
:end
echo.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.
รายละเอียดทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Memory map

Port Decode

34	54	64	65	66	
30	31	32	33	35	36
50	51	52	53	55	57
60	61	62	63	67	

ALT+INC = 47

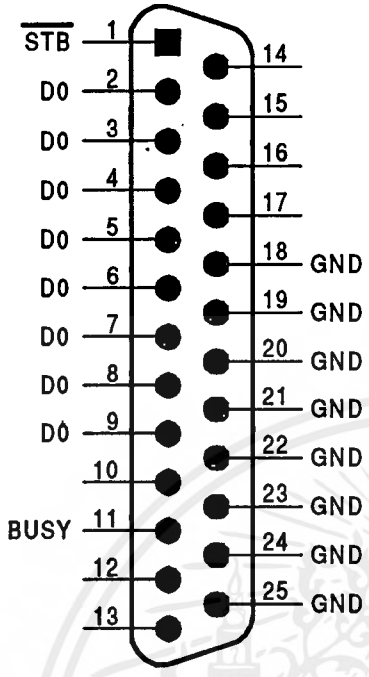
Scan Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

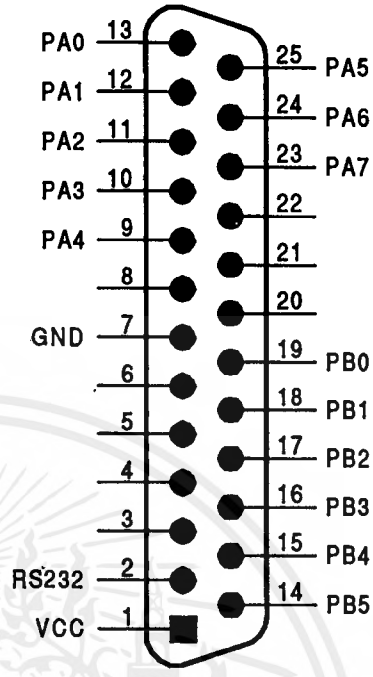
Key	Normal	ALT
0	0	80
1	1	81
2	2	82
3	3	83
4	4	84
5	5	85
6	6	86
7	7	87
8	8	88
9	9	89
A	A	8A
B	B	8B
C	C	8C
D	D	8D
E	E	8E
F	F	8F
ALT/FUN	10	-
DEC	11	96
REC	12	92
INC	13	93
DATA	14	94
RUN	15	95

ตารางแสดงค่า คีย์โค้ด ของคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

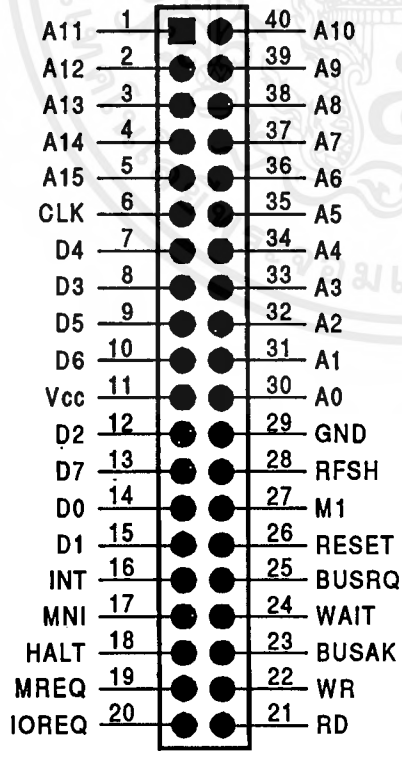


Printer Connector

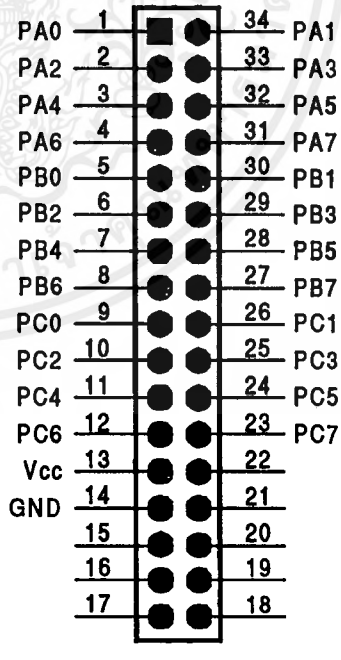


Solenoid Connector

รายละเอียดตำแหน่งขาคอนเนคเตอร์ที่ด้านหน้าของเครื่อง



Z80 CONNECTOR



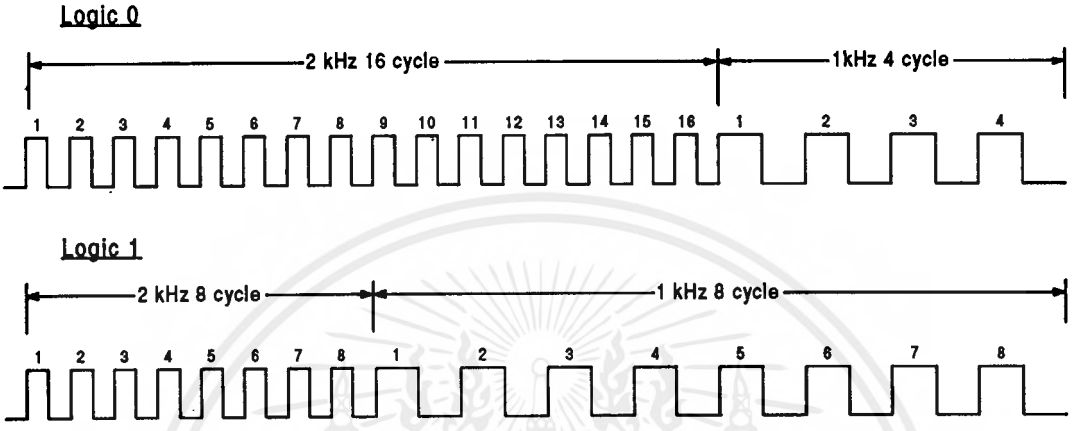
8255 CONNECTOR

รายละเอียดตำแหน่งขาคอนเนคเตอร์ของบอร์ด CP-Z80V1

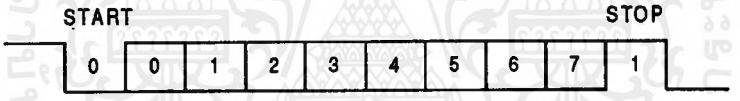
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของสัญญาณ ที่ใช้บันทึกลงเทปบันทึกเสียง

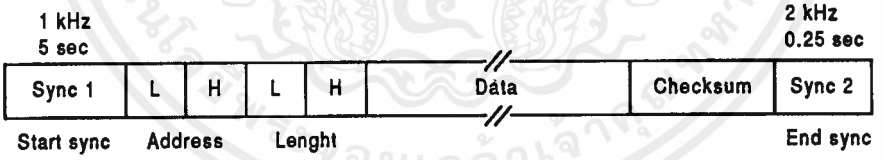
BIT FORMAT



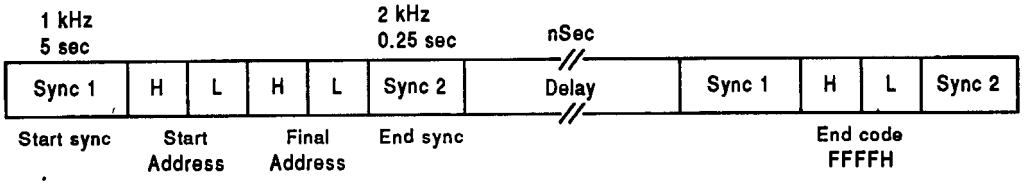
BYTE FORMAT



FILE FORMAT

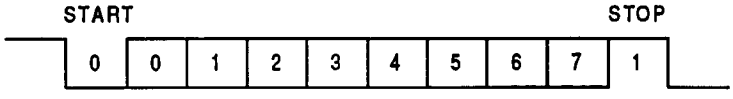


RECORD FORMAT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบข้อมูลของ RS232



รูปแบบไฟล์ข้อมูลเลขฐาน 16 ของ INTEL

: NNAAAARRHHHHHHHHHHHHCCTT

: = อักขระ เริ่มต้นของข้อมูล

NN = เลขแสดงจำนวนไบท์ที่ส่งมา

AAAA = ค่าแอดเดรสของไบท์แรกที่ส่งมา

RR = ชนิดของข้อมูล (สำหรับเลขฐาน 16 = 00H)

HH = ข้อมูล

CC = ค่าตรวจสอบความถูกต้อง (CHECKSUM)

SUM = เลขจำนวนไบท์+แอดเดรสไบท์ต่ำ+แอดเดรสไบท์สูง+เลขชนิดของข้อมูล
+ผลบวกของข้อมูล

CHECKSUM = (-SUM) AND 0FFH

TT = อักขระจบบรรทัด (CARRIGE RETURN,LINE FEED)

ความผิดพลาด	ความหมาย	แสดง
Error 00	การอ่าน Start bit ผิดพลาด	Err 00
Error 01	การอ่าน Stop bit ผิดพลาด	Err 01
Error 02	ความถี่ที่อ่านได้ผิดพลาด	Err 02
Error 03	การอ่านอักขระผิดพลาด	Err 03
Error 04	ค่า Check sum ผิดพลาด	Err 04

ตาราง แสดงความหมายของการผิดพลาดแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

DATA SHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54/74LS138

1-OF-8-DECODER/ DEMULTIPLEXER

LOW POWER SCHOTTKY

DESCRIPTION — The LSTTL/MSI SN54LS/74LS138 is a high speed 1-of-8 Decoder/Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three LS138 devices or to a 1-of-32 decoder using four LS138s and one inverter. The LS138 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

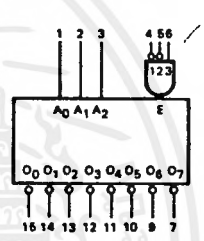
- DEMULTIPLEXING CAPABILITY
- MULTIPLE INPUT ENABLE FOR EASY EXPANSION
- TYPICAL POWER DISSIPATION OF 32 mW
- ACTIVE LOW MUTUALLY EXCLUSIVE OUTPUTS
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

PIN NAMES

		LOADING (Note a)	
		HIGH	LOW
$A_0 - A_2$	Address Inputs	0.5 U.L.	0.25 U.L.
$\overline{E}_1, \overline{E}_2$	Enable (Active LOW) Inputs	0.5 U.L.	0.25 U.L.
E_3	Enable (Active HIGH) Input	0.5 U.L.	0.25 U.L.
$\overline{O}_0 - \overline{O}_7$	Active LOW Outputs (Note b)	10 U.L.	5(2.5) U.L.

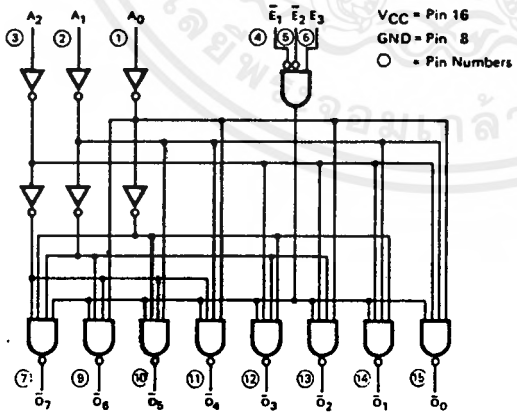
NOTES:
a. 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.
b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

LOGIC SYMBOL

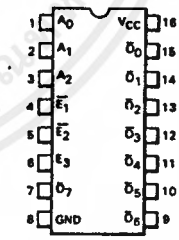


V_{CC} = Pin 16
GND = Pin 8

LOGIC DIAGRAM



CONNECTION DIAGRAM DIP (TOP VIEW)



J Suffix — Case 620-09 (Ceramic)
N Suffix — Case 648-08 (Plastic)

NOTE
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTIONAL DESCRIPTION – The LS138 is a high speed 1-of-8 Decoder/Demultiplexer fabricated with the low power Schottky barrier diode process. The decoder accepts three binary weighted inputs (A_0, A_1, A_2) and when enabled provides eight mutually exclusive active LOW outputs ($\overline{O}_0$-$\overline{O}_7$). The LS138 features three Enable inputs, two active LOW ($\overline{E}_1$, $\overline{E}_2$) and one active HIGH (E_3). All outputs will be HIGH unless $\overline{E}_1$ and $\overline{E}_2$ are LOW and E_3 is HIGH. This multiple enable function allows easy parallel expansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four LS138s and one inverter. (See Figure a.)

The LS138 can be used as an 8-output demultiplexer by using one of the active LOW Enable inputs as the data input and the other Enable inputs as strobes. The Enable inputs which are not used must be permanently tied to their appropriate active HIGH or active LOW state.

TRUTH TABLE

INPUTS						OUTPUTS							
$\overline{E}_1$	$\overline{E}_2$	E_3	A_0	A_1	A_2	$\overline{O}_0$	$\overline{O}_1$	$\overline{O}_2$	$\overline{O}_3$	$\overline{O}_4$	$\overline{O}_5$	$\overline{O}_6$	$\overline{O}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

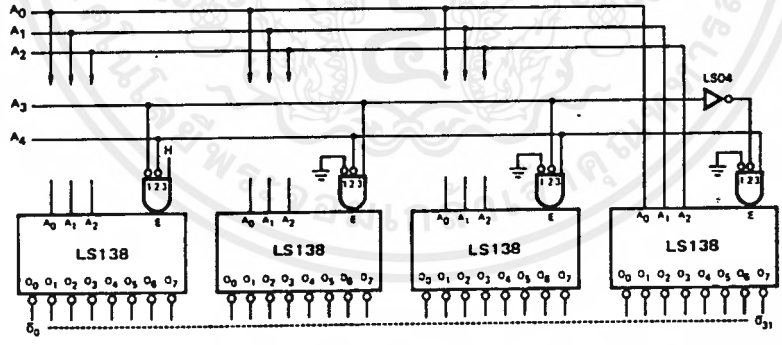


Fig. a.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.7	3.5		V		
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	I_{OL} = 4.0 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Short Circuit Current		−20		−100	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current				10	mA	V_{CC} = MAX	

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LEVEL OF DELAY	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH} t_{PHL}	Propagation Delay Address to Output	2 2		13 27	20 41	ns	V_{CC} = 5.0 V C_L = 15 pF
t_{PLH} t_{PHL}	Propagation Delay Address to Output	3 3		18 26	27 39	ns	
t_{PLH} t_{PHL}	Propagation Delay $\overline{E_1}$ or $\overline{E_2}$ Enable to Output	2 2		12 21	18 32	ns	
t_{PLH} t_{PHL}	Propagation Delay E_3 Enable to Output	3 3		17 25	26 38	ns	

AC WAVEFORMS

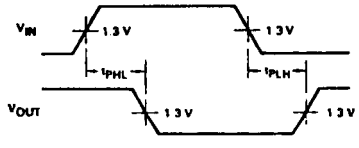


Fig. 1

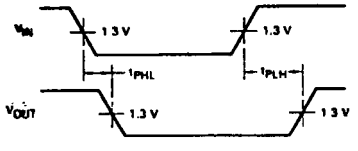


Fig. 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54/74LS145

1-OF-10 DECODER/DRIVER OPEN-COLLECTOR

LOW POWER SCHOTTKY

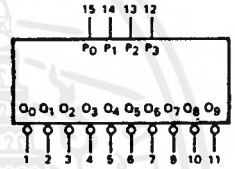
DESCRIPTION — The SN54LS/74LS145, 1-of-10 Decoder/Driver, is designed to accept BCD inputs and provide appropriate outputs to drive 10-digit incandescent displays. All outputs remain off for all invalid binary input conditions. It is designed for use as indicator/relay drivers or as an open-collector logic circuit driver. Each of the high breakdown output transistors will sink up to 80 mA of current. Typical power dissipation is 35 mW. This device is fully compatible with all TTL families.

- LOW POWER VERSION OF 54/74145
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

PIN NAMES		LOADING (Note a) HIGH	LOW
P_0, P_1, P_2, P_3	BCD Inputs	0.5 U.L.	0.25 U.L.
$\overline{Q}_0$ to $\overline{Q}_9$	Outputs (Note b)	Open Collector	15 (7.5) U.L.

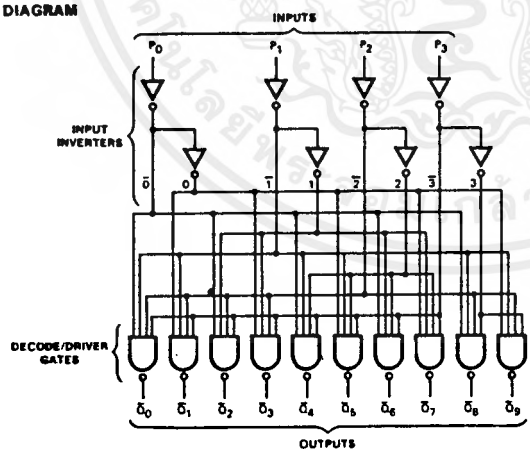
NOTES:
a. 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.
b. The Output LOW drive factor is 7.5 U.L. for Military (54) and 15 U.L. for Commercial (74) Temperature Ranges.

LOGIC SYMBOL

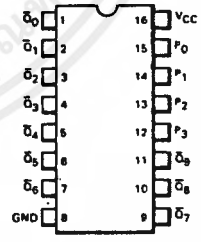


V_{CC} = Pin 16
GND = Pin 8

LOGIC DIAGRAM



CONNECTION DIAGRAM
DIP (TOP VIEW)



J Suffix — Case 620-09 (Ceramic)
N Suffix — Case 648-08 (Plastic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54/74LS145

TRUTH TABLE

INPUTS				OUTPUTS									
P_3	P_2	P_1	P_0	$\bar{Q}_0$	$\bar{Q}_1$	$\bar{Q}_2$	$\bar{Q}_3$	$\bar{Q}_4$	$\bar{Q}_5$	$\bar{Q}_6$	$\bar{Q}_7$	$\bar{Q}_8$	$\bar{Q}_9$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	H	H	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = HIGH Voltage Level
L = LOW Voltage Level

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
V_{OH}	Output Voltage — High	54, 74			15	V
I_{OL}	Output Current — Low	54 74			12 24	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74			0.8		
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA
I_{OH}	Output HIGH Current	54,74			250	μA	V_{CC} = MIN, V_{OH} = MAX
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	I_{OL} = 12 mA; V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA
		54,74		2.3	3.0	V	I_{OL} = 80 mA
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{CC}	Power Supply Current				13	mA	V_{CC} = MAX, V_{IN} = GND

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PHL} t_{PLH}	Propagation Delay P_n Input to Q_n Output			50 50	ns	V_{CC} = 5.0 V C_L = 45 pF

AC WAVEFORMS

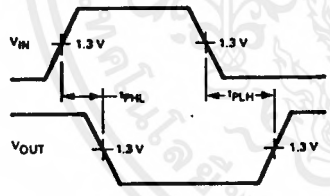


Fig. 1

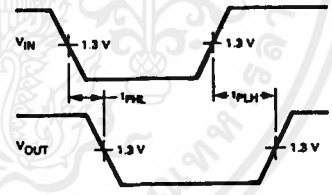


Fig. 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54/74LS240
SN54/74LS241
SN54/74LS244

OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS

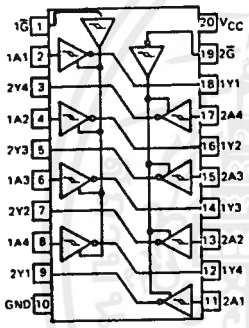
LOW POWER SCHOTTKY

DESCRIPTION — The SN54LS/74LS240, 241 and 244 are Octal Buffers and Line Drivers designed to be employed as memory address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC board density.

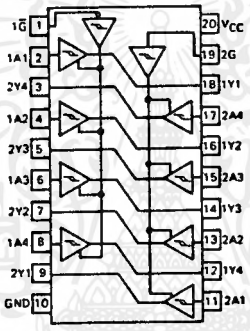
- HYSTERESIS AT INPUTS TO IMPROVE NOISE MARGINS
- 3-STATE OUTPUTS DRIVE BUS LINES OR BUFFER MEMORY ADDRESS REGISTERS
- INPUT CLAMP DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)

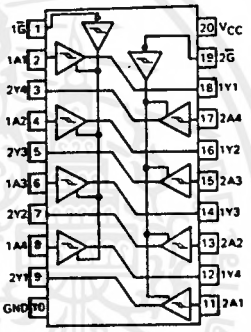
SN54LS/74LS240



SN54LS/74LS241



SN54LS/74LS244



TRUTH TABLES

SN54LS/74LS240

INPUTS		OUTPUT
1G̅,2G̅	D	
L	L	H
L	H	L
H	X	(Z)

SN54LS/74LS244

INPUTS		OUTPUT
1G̅,2G̅	D	
L	L	L
L	H	H
H	X	(Z)

SN54LS/74LS241

INPUTS		OUTPUT	INPUTS		OUTPUT
1G̅	D		2G	D	
L	L	L	H	L	L
L	H	H	H	H	H
H	X	(Z)	L	X	(Z)

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = HIGH Impedance

J Suffix — Case 732-03 (Ceramic)
N Suffix — Case 738-03 (Plastic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54,74			−3.0	mA
		54 74			−12 −15	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
$V_{T+} - V_{T-}$	Hysteresis		0.2	0.4		V	V_{CC} = MIN	
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54,74	2.4	3.4		V	V_{CC} = MIN, I_{OH} = −3.0 mA	
		54,74	2.0			V	V_{CC} = MIN, I_{OH} = MAX	
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA	
I_{OZH}	Output Off Current HIGH				20	μA	V_{CC} = MAX, V_{OUT} = 2.7 V	
I_{OZL}	Output Off Current LOW				−20	μA	V_{CC} = MAX, V_{OUT} = 0.4 V	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.2	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Output Short Circuit Current		−40		−225	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current Total, Output HIGH				27	mA	V_{CC} = MAX	
	Total, Output LOW	LS240			44			
		LS241/244			46			
	Total at HIGH Z	LS240			50			
		LS241/244			54			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC CHARACTERISTICS: $T_A = 25°C$, $V_{CC} = 5.0\ V$

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS240		9.0 12	14 18	ns	$C_L = 45\ pF$, $R_L = 667\ \Omega$
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS241/244		12 12	18 18	ns	
t_{PZH}	Output Enable Time to HIGH Level		15	23	ns	
t_{PZL}	Output Enable Time to LOW Level		20	30	ns	
t_{PLZ}	Output Disable Time from LOW Level		15	25	ns	$C_L = 5.0\ pF$ $R_L = 667\ \Omega$
t_{PHZ}	Output Disable Time from HIGH Level		10	18	ns	

AC WAVEFORMS

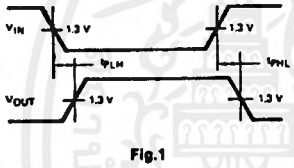


Fig. 1

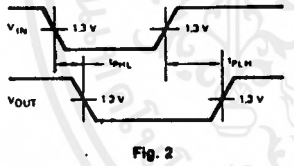


Fig. 2

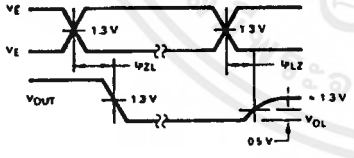


Fig. 3

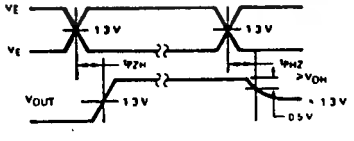
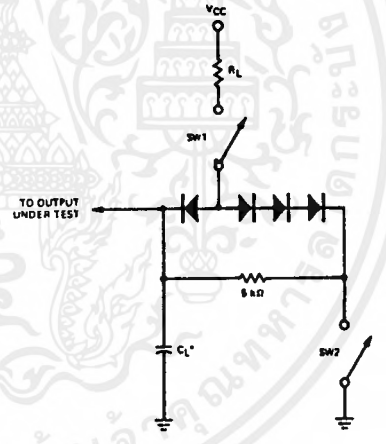


Fig. 4



SWITCH POSITIONS

SYMBOL	SW1	SW2
t_{PZH}	Open	Closed
t_{PZL}	Closed	Open
t_{PLZ}	Closed	Closed
t_{PHZ}	Closed	Closed

Fig. 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54/74LS245

OCTAL BUS TRANSCEIVER

LOW POWER SCHOTTKY

DESCRIPTION — The SN54LS/74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input ($\overline{E}$) can be used to isolate the buses.

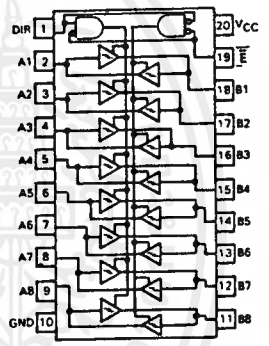
- HYSTERESIS INPUTS TO IMPROVE NOISE IMMUNITY
- 2-WAY ASYNCHRONOUS DATA BUS COMMUNICATION
- INPUT DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS

TRUTH TABLE

INPUTS		OUTPUT
$\overline{E}$	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

LOGIC AND CONNECTION DIAGRAM
DIP (TOP VIEW)



J Suffix — Case 732-03 (Ceramic)
N Suffix — Case 738-03 (Plastic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54,74			−3.0	mA
		54 74			−12 −15	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
$V_{T+}-V_{T-}$	Hysteresis		0.2	0.4		V	V_{CC} = MIN	
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54,74	2.4	3.4		V	V_{CC} = MIN, I_{OH} = −3.0 mA	
		54,74	2.0			V	V_{CC} = MIN, I_{OH} = MAX	
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA	
I_{OZH}	Output Off Current HIGH				20	μA	V_{CC} = MAX, V_{OUT} = 2.7 V	
I_{OZL}	Output Off Current LOW				−200	μA	V_{CC} = MAX, V_{OUT} = 0.4 V	
I_{IH}	Input HIGH Current	A or B, DR or $\overline{E}$			20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
		DR or $\overline{E}$			0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
		A or B			0.1	mA	V_{CC} = MAX, V_{IN} = 5.5 V	
I_{IL}	Input LOW Current				−0.2	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Output Short Circuit Current		−40		−225	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current Total, Output HIGH				70	mA	V_{CC} = MAX	
	Total, Output LOW				90			
	Total at HIGH Z				95			

AC CHARACTERISTICS: T_A = 25°C, V_{CC} = 5.0 V

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH} t_{PHL}	Propagation Delay, Data to Output		8.0 8.0	12 12	ns	C_L = 45 pF R_L = 667 Ω
t_{PZH}	Output Enable Time to HIGH Level		25	40	ns	
t_{PZL}	Output Enable Time to LOW Level		27	40	ns	
t_{PLZ}	Output Disable Time from LOW Level		15	25	ns	C_L = 5.0 pF R_L = 667 Ω
t_{PHZ}	Output Disable Time from HIGH Level		15	25	ns	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MCM6264

8K × 8 Bit Fast Static RAM

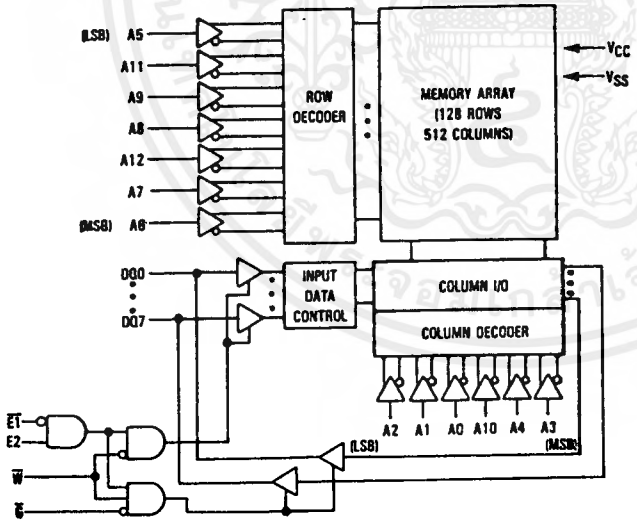
The MCM6264 is a 65,536 bit static random access memory organized as 8192 words of 8 bits, fabricated using Motorola's high-performance silicon-gate CMOS technology. Static design eliminates the need for external clocks or timing strobes, while CMOS circuitry reduces power consumption which provides greater reliability.

The chip enable pins ($\overline{E1}$ and E2) are not clocks. Either pin, when asserted false, causes the part to enter a low power standby mode. The part will remain in standby mode until both pins are asserted true again. The availability of active high and active low chip enable pins provides more system design flexibility than single chip enable devices.

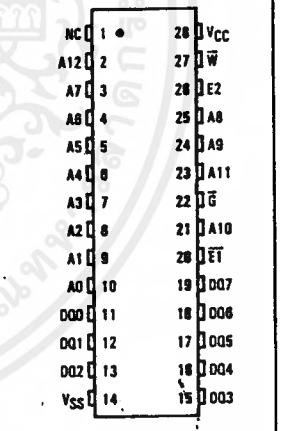
The MCM6264 is available in 300 and 600 mil, 28 pin plastic dual-in-line packages and 300 and 400 mil, 28 pin plastic SOJ packages. All packages feature the JEDEC standard pinout.

- Single 5 V Supply, ±10%
- 8K × 8 Organization
- Fully Static—No Clock or Timing Strobes Necessary
- Fast Access Time—30, 35, 45, 55 ns (Maximum)
- Low Power Operation—105, 100, 90, 80 mA (Maximum, Active)
- Three State Outputs
- All Inputs and Outputs are TTL Compatible
- Output Enable ($\overline{G}$) Feature for Increased System Flexibility and to Eliminate Bus Contention Problems

BLOCK DIAGRAM



PIN ASSIGNMENT



PIN NAMES	
A0-A12	Address
$\overline{W}$	Write Enable
$\overline{E1}$, E2	Chip Enable
$\overline{G}$	Output Enable
DQ0-DQ7	Data Input/Output
V_{CC}	+5 V Power Supply
V_{SS}	Ground
NC	No Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRUTH TABLE

$\overline{E1}$	E2	$\overline{G}$	$\overline{W}$	Mode	Supply Current	I/O Pin
H	X	X	X	Not Selected	I_{SB}	High Z
X	L	X	X	Not Selected	I_{SB}	High Z
L	H	H	H	Output Disabled	I_{CC}	High Z
L	H	L	H	Read	I_{CC}	D_{out}
L	H	X	L	Write	I_{CC}	D_{in}

X = don't care

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit.

ABSOLUTE MAXIMUM RATINGS (See Note)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	−0.5 to +7.0	V
Voltage Relative to V_{SS} for Any Pin Except V_{CC}	V_{in}, V_{out}	−0.5 to V_{CC} + 0.5	V
Output Current (per I/O)	I_{out}	±20	mA
Power Dissipation (T_A = 25°C)	P_D	1.0	W
Temperature Under Bias	T_{bias}	−10 to +85	°C
Operating Temperature	T_A	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C

NOTE: Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device reliability.

DC OPERATING CONDITIONS AND CHARACTERISTICS

(V_{CC} = 5.0 V ± 10%, T_A = 0 to 70°C, Unless Otherwise Noted)

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage (Operating Voltage Range)	V_{CC}	4.5	5.0	5.5	V
Input High Voltage	V_{IH}	2.2	–	V_{CC} + 0.3	V
Input Low Voltage	V_{IL}	−0.3*	–	0.8	V

*V_{IL} (min) = −0.3 V dc; V_{IL} (min) = −3.0 V ac (pulse width ≤20 ns)

DC CHARACTERISTICS

Parameter		Symbol	Min	Max	Unit
Input Leakage Current (All Inputs, V_{in} = 0 to V_{CC})		$I_{lkg(I)}$	–	±1.0	μA
Output Leakage Current ($\overline{E1}$ = V_{IH}, E2 = V_{IL}, or $\overline{G}$ = V_{IH}, V_{out} = 0 to V_{CC})		$I_{lkg(O)}$	–	±1.0	μA
Power Supply Current ($\overline{E1}$ = V_{IL}, E2 = V_{IH}, I_{out} = 0)	(t_{AVAV} = 30 ns)	I_{CC}	–	105	mA
	(t_{AVAV} = 35 ns)		–	100	
	(t_{AVAV} = 45 ns)		–	90	
	(t_{AVAV} = 55 ns)		–	80	
Standby Current ($\overline{E1}$ = V_{IH} or E2 = V_{IL})		I_{SB1}	–	10	mA
Standby Current ($\overline{E1}$ ≥ V_{CC} − 0.2 V or E2 ≤ 0.2 V, V_{in} = V_{IH} or V_{in} = V_{IL})		I_{SB2}	–	5	mA
Output Low Voltage (I_{OL} = 8.0 mA)		V_{OL}	–	0.4	V
Output High Voltage (I_{OH} = −4.0 mA)		V_{OH}	2.4	–	V

CAPACITANCE (f = 1.0 MHz, dV = 3.0 V, T_A = 25°C, Periodically Sampled Rather Than 100% Tested)

Characteristic		Symbol	Max	Unit
Input Capacitance	All Inputs Except DQ	C_{in}	6	pF
I/O Capacitance	DQ	$C_{I/O}$	8	pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC OPERATING CONDITIONS AND CHARACTERISTICS

(V_{CC} = 5.0 V ±10%, T_A = 0 to 70°C, Unless Otherwise Noted)

Input Pulse Levels . . . . . 0 to 3.0 V
Input Rise/Fall Time . . . . . 5 ns
Input Timing Measurement Reference Levels . . . . . 1.5 V
Output Timing Measurement Reference Levels . . . . . 0.8 and 2.0 V
Output Load . . . . . See Figure 1

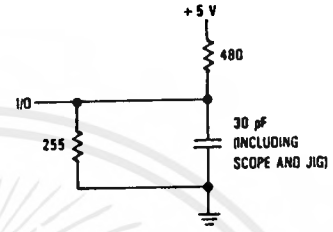
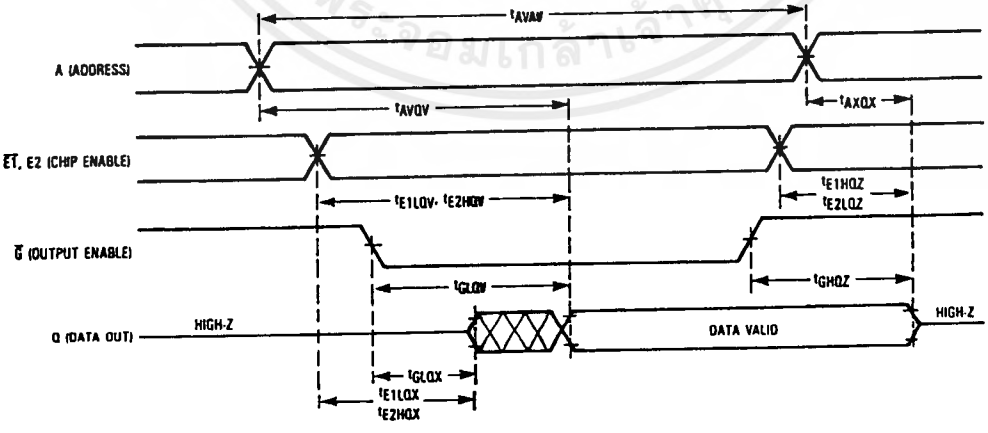


Figure 1. Test Load

READ CYCLE (See Note 1)

Parameter	Symbol	Alt Symbol	MCM6264-30		MCM6264-35		MCM6264-45		MCM6264-55		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
Read Cycle Time	t_{AVAV}	t_{RC}	30	—	35	—	45	—	55	—	ns	—
Address Cycle Time	t_{AVQV}	t_{AA}	—	30	—	35	—	45	—	55	ns	—
$\overline{E1}$ Access Time	t_{E1LQV}	t_{AC1}	—	30	—	35	—	45	—	55	ns	—
E2 Access Time	t_{E2HQV}	t_{AC2}	—	30	—	35	—	45	—	55	ns	—
$\overline{G}$ Access Time	t_{GLQV}	t_{OE}	—	12.5	—	15	—	20	—	25	ns	—
Output Hold from Address Change	t_{AXQX}	t_{OH}	5	—	5	—	5	—	5	—	ns	—
Chip Enable to Output Low-Z	t_{E1LQX}, t_{E2HQX}	t_{CLZ}	5	—	5	—	5	—	5	—	ns	2, 3
Output Enable to Output Low-Z	t_{GLQX}	t_{OLZ}	0	—	0	—	0	—	0	—	ns	2, 3
Chip Enable to Output High-Z	t_{E1HQZ}, t_{E2LQZ}	t_{CHZ}	0	15	0	15	0	15	0	15	ns	2, 3
Output Enable to Output High-Z	t_{GHQZ}	t_{OHZ}	0	15	0	15	0	15	0	15	ns	2, 3

NOTES:
1. $\overline{W}$ is high at all times for read cycles.
2. All high-Z and low-Z parameters are considered in a high or low impedance state when the output has made a 500 mV transition from the previous steady state voltage.
3. These parameters are periodically sampled and not 100% tested.

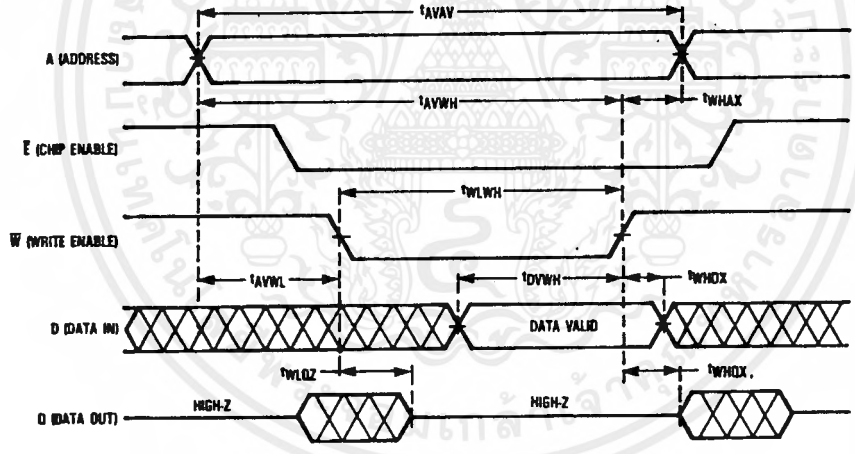


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE CYCLE 1 (W CONTROLLED) (See Note 1)

Parameter	Symbol	Alt Symbol	MCM6264-30		MCM6264-35		MCM6264-45		MCM6264-55		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
Write Cycle Time	t_{AVAV}	t_{WC}	30	–	35	–	45	–	55	–	ns	–
Address Setup Time	t_{AVWL}	t_{AS}	0	–	0	–	0	–	0	–	ns	–
Address Valid to End of Write	t_{AVWH}	t_{AW}	22.5	–	25	–	35	–	45	–	ns	–
Write Pulse Width	t_{WLWH}	t_{WP}	17.5	–	20	–	25	–	30	–	ns	3
Data Valid to End of Write	t_{DVWH}	t_{DW}	12.5	–	15	–	20	–	25	–	ns	–
Data Hold Time	t_{WHDX}	t_{DH}	0	–	0	–	0	–	0	–	ns	3
Write Low to Output in High-Z	t_{WLQZ}	t_{WHZ}	0	15	0	15	0	15	0	15	ns	4, 5
Write High to Output Low-Z	t_{WHQX}	t_{OW}	5	–	5	–	5	–	5	–	ns	4, 5
Write Recovery Time	t_{WHAX}	t_{WR}	0	–	0	–	0	–	0	–	ns	–

NOTES:
1. A write cycle starts at the latest transition of a low $\overline{E1}$, low $\overline{W}$, or high E2. A write cycle ends at the earliest transition of a high $\overline{E1}$, high $\overline{W}$, or low E2.
2. If $\overline{W}$ goes low coincident with or prior to $\overline{E1}$ low or E2 high then the outputs will remain in a high impedance state.
3. During this time the output pins may be in the output state. Signals of opposite phase to the outputs must not be applied at this time.
4. All high-Z and low-Z parameters are considered in a high or low impedance state when the output has made a 500 mV transition from the previous steady state voltage.
5. These parameters are periodically sampled and not 100% tested.



TIMING PARAMETER ABBREVIATIONS

t X X X X

signal name from which interval is defined
transition direction for first signal
signal name to which interval is defined
transition direction for second signal

The transition definitions used in this data sheet are:
- H = transition to high
- L = transition to low
- V = transition to valid
- X = transition to invalid or don't care
- Z = transition to off (high impedance)

TIMING LIMITS

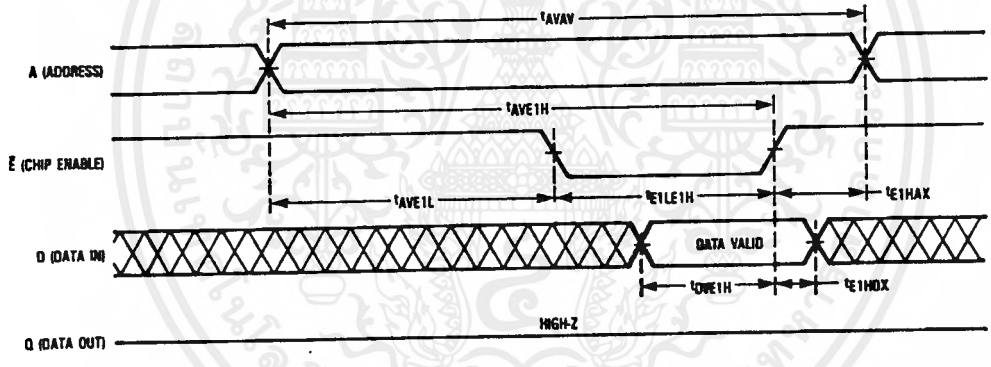
The table of timing values shows either a minimum or a maximum limit for each parameter. Input requirements are specified from the external system point of view. Thus, address setup time is shown as a minimum since the system must supply at least that much time (even though most devices do not require it). On the other hand, responses from the memory are specified from the device point of view. Thus, the access time is shown as a maximum since the device never provides data later than that time.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE CYCLE 2 (ENABLE CONTROLLED) (See Note 1)

Parameter	Symbol	Alt Symbol	MCM6264-30		MCM6264-35		MCM6264-45		MCM6264-55		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
Write Cycle Time	t_{AVAV}	t_{WC}	30	—	35	—	45	—	55	—	ns	—
Address Setup Time	t_{AVE1L}	t_{AS}	0	—	0	—	0	—	0	—	ns	2
Address Valid to End of Write	t_{AVE1H}	t_{AW}	22.5	—	25	—	35	—	45	—	ns	2
Chip Enable to End of Write	t_{E1LE1H}	t_{CW}	22.5	—	25	—	35	—	45	—	ns	2, 3
Data Valid to End of Write	t_{DVE1H}	t_{DW}	12.5	—	15	—	20	—	25	—	ns	2
Data Hold Time	t_{E1HDX}	t_{DH}	0	—	0	—	0	—	0	—	ns	2, 4
Write Recovery Time	t_{E1HAX}	t_{WR}	0	—	0	—	0	—	0	—	ns	2

NOTES:
1. A write cycle starts at the latest transition of a low $\overline{E1}$, low $\overline{W}$, or high E2. A write cycle ends at the earliest transition of a high $\overline{E1}$, high $\overline{W}$, or low E2.
2. $\overline{E1}$ and E2 timings are identical when E2 signals are inverted.
3. If $\overline{W}$ goes low coincident with or prior to $\overline{E1}$ low or E2 high then the outputs will remain in a high impedance state.
4. During this time the output pins may be in the output state. Signals of opposite phase to the outputs must not be applied at this time.



ORDERING INFORMATION
(Order by Full Part Number)

MCM 6264 X XX XX

- MCM — Motorola Memory Prefix
- 6264 — Part Number
- X — Package (P = 300 mil Plastic DIP, WP = 600 mil Plastic DIP, J = 400 mil Plastic SOJ, NJ = 300 mil Plastic SOJ)
- XX — Speed (30 = 30 ns, 35 = 35 ns, 45 = 45 ns, 55 = 55 ns)
- XX — Shipping Method (R2 = Tape & Reel, Blank = Rails)

Full Part Numbers—

MCM6264P30	MCM6264WP30	MCM6264J30	MCM6264J30R2	MCM6264NJ30	MCM6264NJ30R2
MCM6264P35	MCM6264WP35	MCM6264J35	MCM6264J35R2	MCM6264NJ35	MCM6264NJ35R2
MCM6264P45	MCM6264WP45	MCM6264J45	MCM6264J45R2	MCM6264NJ45	MCM6264NJ45R2
MCM6264P55	MCM6264WP55	MCM6264J55	MCM6264J55R2	MCM6264NJ55	MCM6264NJ55R2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้