



เครื่องบันทึกเสียงทางโทรศัพท์ แบบแยกช่อง
VOICE RECORDER FROM TELEPHONE LINE



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป (008473) การค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญาโท

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

เครื่องบันทึกเสียงทางโทรศัพท์ แบบแยกช่อง
(Voice Recorder From Telephone Line)

โดย

นายวันชัย ก้อนทอง รหัส 33.131220
นายสมิทธิ์ สุขชี รหัส 33.131229
นายอุดมศักดิ์ ชัยเพชร รหัส 33.131237

ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญาโท



ประธานกรรมการ

(-----)

กรรมการ

(-----)

กรรมการ

(-----)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องบันทึกเสียงโทรศัพท์ แบบแยกช่อง

Voice Recorder From Telephone Line



นายวันชัย ก้อนทอง รหัส 33.131220
นายสมิทธิ สุขชัย รหัส 33.131229
นายอุดมศักดิ์ ชัยเพชร รหัส 33.131237

ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษา

ตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

ภาควิชาเทคโนโลยีอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาานิพนธ์ เครื่องบันทึกเสียงทางโทรศัพท์ แบบแยกช่อง
(Voice Recorder From Telephone Line)
ชื่อผู้เขียน นายวันชัย ก้อนทอง
นายสมิทธิ์ สุขชี
นายอุดมศักดิ์ ชัยเพชร
ชื่อปริญญา อุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์
ที่ปรึกษาปริญญาานิพนธ์ อาจารย์ชวลิต เบญจางคประเสริฐ
ปีการศึกษา 2534

บทคัดย่อ

ปริญญาานิพนธ์นี้เป็นการสร้างเครื่องบันทึกเสียงโทรศัพท์ แบบแยกช่อง ควบคุมการทำงานด้วย Microprocessor โดยใช้ CPU เบอร์ Z 80 สามารถบันทึกเสียงผู้โทรศัพท์เข้ามาติดต่อลงบนเทปบันทึกเสียงขนาด 30 นาที ต่อหน้า 1 ม้วน เลือกฝากข้อความถึงผู้ที่ต้องการติดต่อด้วยได้ 5 คน โดยเลือกได้ครั้งละ 1 คน ในการฝากข้อความแต่ละครั้งสามารถฝากได้ไม่เกิน 2 นาที ซึ่งถือว่าพอเพียงสำหรับการฝากข้อความโดยทั่วไป สำหรับผู้รับข้อความ สามารถรับข้อความได้คนละ 3 ชุด และสามารถฟังข้อความได้ทั้งจากโทรศัพท์ภายนอกหรือที่ตัวเครื่อง โดยใช้รหัสผ่านเพื่อป้องกันผู้อื่นแอบฟัง เครื่องนี้สามารถอธิบายวิธีใช้เครื่องแก่ผู้ที่ต้องการฝากข้อความได้ สามารถใช้ได้กับโทรศัพท์ชนิด Tone เท่านั้น โครงการนี้คาดว่าจะประโยชน์กับการติดต่อสื่อสารในธุรกิจขนาดเล็ก และอาคารบ้านเรือนที่ต้องการเครื่องมือประเภทนี้ไว้ใช้ในราคาถูกลงต่อไป

Project Name Voice Recorder From Telephone Line
Student Name Mr. Wanchai Konthong
 Mr. Smitt Sukkhi
 Mr. Udomsuk Chaipech
Bachelor Degree in Industrial Techonology (Electronics)
Advisor Mr. Chawalit Benjangkprasurt
Year 1991

ABSTRACT

This project is the voice recorder from telephone line in multi channel type. It cotrolled by microprocessor Z-80 CPU. Voice recording of telephone user who dial in for communication will be record in the tape cassett size 30minute/page 1 set. They can select the receiver in each of five. And each receiver can get message maximun 3 time. Recording in a time not over 2 minute. Wecan get the voice massege in 2 ways bythe telephone to call in and by press the function keys of machine. Both ways have to use passwords for protect the listening from another one who are not voice message's owner. This machine have user's guide in answering voice form for explain to telephone's contactor. And using with lone type telephone only.

We hope this project can help the communication in small business or household which are want the low price of this type machine.

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ท
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนและวิธีดำเนินงาน	3
1.5 การศึกษาค้นคว้าข้อมูลในส่วนที่เกี่ยวข้องกับโครงการ	4
1.6 โครงสร้างของระบบและหลักการทำงาน	5
1.7 การทำงานของ Block diagram	6
1.8 การออกแบบโครงสร้างของระบบ	9
บทที่ 2 ทฤษฎีการทำงาน และการออกแบบวงจร	11
2.1 DTMF Decoder	11
2.2 Digital Voice Recorder	20
2.3 Tape Recorder	30
2.4 วงจรตรวจสอบการรบกวนทางหู วางหู	39
2.5 วงจรตรวจสอบสัญญาณกระดิ่ง	40
2.6 วงจรกำเนิดสัญญาณ NMI	41
2.7 Dot Matrix LCD Module	42
2.8 ชุดควบคุม Z-80 CP-A	52
2.9 วงจรตรวจสอบสัญญาณเสียง	56
2.10 วงจรตรวจสอบสัญญาณหึ่งค์	60
2.11 วงจรภาคขยายเสียง	61
2.12 วงจรกำเนิดเสียงดนตรี	62
2.13 สรุป	63
บทที่ 3 การทดสอบการทำงาน	66
3.1 ผลการทดสอบวงจรต่าง ๆ	66

3.2 การทดสอบการทำงานของฟังก์ชันต่าง ๆ	76
บทที่ 4 การใช้งานเครื่องบันทึกเสียงโทรศัพท์แบบแยกช่อง	78
4.1 รายละเอียดของฟังก์ชัน	78
4.2 ขั้นตอนการติดตั้งใช้งาน	80
บทที่ 5 สรุปผลการดำเนินงาน	90
5.1 ผลของโครงการ	90
5.2 ปัญหาและการแก้ไข	91
5.3 ข้อเสนอแนะ	93
กิตติกรรมประกาศ	95
เอกสารอ้างอิง	96
ภาคผนวก	97



บทที่ 1

บทนำ

1.1 ความเป็นมา

เนื่องจากปัจจุบันสภาพสังคมมีขนาดใหญ่ขึ้น มีความเจริญเติบโตอย่างมากในด้านเศรษฐกิจ ในสภาพดังกล่าวจะเห็นได้ว่าการติดต่อสื่อสารที่รวดเร็วถูกต้องและประหยัดค่าใช้จ่าย เป็นสิ่งสำคัญอย่างยิ่ง ซึ่งในปัจจุบันการติดต่อสื่อสารก็ได้พัฒนาไปมากเช่นกัน โดยเป็นไปในหลายรูปแบบ เช่นระบบไปรษณีย์ ระบบวิทยุสื่อสาร ระบบโทรศัพท์ ซึ่งประเภทหลังนี้มีใช้กันอย่างแพร่หลายในสังคมเมือง และกำลังขยายตัวออกสู่ชนบท จนกลายเป็นสิ่งจำเป็นในชีวิตประจำวัน และการติดต่อสื่อสารในด้านธุรกิจ ต่าง ๆ

ระบบโทรศัพท์ได้มีการพัฒนาไปมากในด้านของเทคโนโลยี เช่น โทรศัพท์เคลื่อนที่ วิทยุติดตามตัว แต่ราคาต่อหน่วยก็ยังค่อนข้างสูงสถานประกอบการต่าง ๆ ไม่สามารถซื้อให้ครบตามความต้องการได้ โดยเฉพาะธุรกิจขนาดเล็กหรือบ้านพักอาศัย ยกตัวอย่างเช่น ในบริษัทแห่งหนึ่งมีพนักงานชาย 4 คน ผู้จัดการ 1 คน ทุกคนต้องการติดต่อแลกเปลี่ยนข้อมูลกัน วิธีที่สะดวกรวดเร็วและได้ข้อมูลถูกต้องที่สุดก็คือการใช้โทรศัพท์ แต่ในบางกรณีบางคนหรือหลายคนต้องออกไปนอกสำนักงาน โทรศัพท์ที่ใช้ก็ต้องเป็นชนิดที่พกพาไปในที่ต่าง ๆ ได้ ซึ่งก็มีราคาสูง การที่จะจัดซื้อให้ครบทุกคนต้องเสียค่าใช้จ่ายสูง ซึ่งอาจไม่คุ้มค่าในการลงทุน แต่ความจำเป็นในการติดต่อสื่อสารในลักษณะดังกล่าว ก็ยังคงมีอยู่ ด้วยปัญหาดังกล่าว จึงมีแนวความคิดที่จะสร้างระบบของเครื่องที่ทำงานในลักษณะเป็นศูนย์กลางแลกเปลี่ยนข้อมูล ในรูปแบบของการบันทึกเสียงพูด โดยมีแนวทางคล้ายกับเครื่องบันทึกเสียงโทรศัพท์ที่มีใช้กันอยู่ในปัจจุบัน

1.2 วัตถุประสงค์ของโครงการ

1. เป็นการศึกษาหาหนทางใช้ระบบโทรศัพท์ ให้มีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สร้างศูนย์แลกเปลี่ยนข่าวสาร Voice message ในราคา ถูกโดยอาศัยระบบคู่สายโทรศัพท์ ขององค์การโทรศัพท์แห่งประเทศไทยที่มีอยู่
3. เพิ่มประสิทธิภาพของเครื่องบันทึกเสียงโทรศัพท์โดยดัดแปลง ให้สามารถแยกบันทึกได้
4. เป็นการเพิ่มพูนความรู้ความเข้าใจ เกี่ยวกับระบบโทรศัพท์
5. ฝึกทักษะในการนำ Microprocessor มาประยุกต์ใช้ งานกับระบบโทรศัพท์ทั้งทางด้าน Hardware และ Software

1.3 ขอบเขตของโครงการ

1. ความคุ้มค่าการทำงานของระบบด้วย Microprocessor Z-80
2. ใช้บันทึกเสียงผู้โทรเข้ามาติดต่อ ลงบนเทปบันทึกเสียงขนาด 30 นาที/หน้า 1 ม้วน
 3. การบันทึกเสียงข้อมูลของแต่ละคนจัดแบ่งเป็นบล็อก
 4. การจัดสรร การบันทึกข้อมูลลงบนเทปบันทึกเสียง
 - จำนวนบล็อกสูงสุดคือ 5 บล็อก
 - แต่ละบล็อกแบ่งเป็น 3 ช่อง ซึ่งก็คือจำนวนครั้งสูงสุด ในการบันทึกของแต่ละบล็อกโดยแต่ละบล็อกบันทึกได้นาน 6 นาที และในแต่ละ ครั้ง จะบันทึกได้สูงสุด 2 นาที ซึ่งเครื่องจะหยุดการบันทึก ถ้าไม่มีสัญญาณ เสียงพูดนานกว่า 10 วินาที
 - ข้อมูลย่อยแต่ละบล็อกจะถูกบันทึกสลับกันไป เพื่อไม่ให้ เสียเวลามากเกินไปในขณะที่ CPU กรอเทปค้นหาบล็อกข้อความ
 5. เครื่องจะเริ่มทำงานอัตโนมัติทันทีที่สัญญาณกระดิ่งดังเกินกว่า 4 ครั้ง
 6. ถ้าหากมีผู้มารับสายก่อนที่สัญญาณกระดิ่งครั้งที่ 5 จะดังขึ้น เครื่องจะไม่ทำงาน ยกเว้น ในกรณีที่ผู้รับสายสั่งให้เครื่องทำงาน ได้แก่ สั่ง ให้เครื่องบันทึกข้อความ ในบล็อกที่ต้องการ หรือจะทำการบันทึกเสียง การสนทนาระหว่างคู่สาย
 7. การสั่งงานของเครื่องในฟังก์ชันของการบันทึก หรือขอฟังข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความในเนื้อเทป ผู้ที่โทรเข้ามาสามารถสั่งงานโดยใช้วิธีกดรหัสผ่านทางคู่สาย
โทรศัพท์ โดยกดปุ่มที่หน้าปัดของเครื่องโทรศัพท์ หรือจะสั่งงานที่
Keyboard ของเครื่องโดยตรงก็ได้ ซึ่งหน้าที่ของ Keyboard มีดังนี้

- การตั้งรหัสของแต่ละบล็อก และรหัสผ่าน
- การ Playback ออกลำโพง
- การควบคุมให้เครื่องทำการบันทึก การฝากข้อความหรือ
บันทึกการสนทนา
- การบันทึกสัญญาณเสียงลงบนชุด Digital Voice
Recorder

8. ข้อความที่ผู้โทรเข้าจะได้ยินในตอนแรก จะถูกบันทึกในรูป
ของ Digital โดยเป็นคำแนะนำเบื้องต้น ให้ผู้โทรเข้ามาปฏิบัติตามขั้นตอน
ตอนอย่างถูกต้องได้

9. ผู้เป็นเจ้าของบล็อกแต่ละคน จะมี Password (4หลัก) เป็น
ของตัวเอง และสามารถเปลี่ยนแปลงได้ด้วยตัวเอง เพื่อใช้ในการขอฟังข้อ
ความในเนื้อเทป โดยโทรเข้ามาขอฟังจากเครื่องโทรศัพท์ภายนอก หรือขอฟัง
ผ่านเครื่องโดยตรง

10. ถ้าหากบล็อกใดถูกบันทึกจนเต็มแล้ว (ครบ 3 ครั้ง) เครื่อง
จะไม่ทำการบันทึกข้อความให้ ซึ่งกรณีนี้ เครื่องจะส่งข้อความบอกไปยังผู้โทร
เข้ามาทราบ ถ้าหากช่องใดที่ถูกฟังไปแล้ว เครื่องจะยอมให้บันทึกซ้ำได้
ตามลำดับของการถูกฟังก่อน-หลัง

11. ใช้ได้กับโทรศัพท์ชนิดกดปุ่ม หรือระบบ Tone เท่านั้น

1.4 ขั้นตอนและวิธีดำเนินงาน

เนื่องจากโครงการนี้เป็นโครงการต่อเนื่อง คือ Project I
และ Project II จึงแบ่งการดำเนินงานออกเป็น 2 ระยะ คือระยะแรก
(Project I) ดำเนินงานทางด้านการออกแบบ Hardware เป็นส่วนใหญ่
ส่วนในระยะที่ 2 (Project II) เป็นการดำเนินงานทางด้านการออกแบบ
Software

ขั้นตอนการดำเนินงานในระยะแรก (Project I)

1. การศึกษาค้นคว้าหาข้อมูล ในส่วนที่เกี่ยวข้องกับโครงการ
 - ระบบโทรศัพท์
 - ระบบไมโครโปรเซสเซอร์
 - เครื่องบันทึกเทปและวิธีค้นหาข้อมูลในเทปบันทึกเสียง
 - กำหนดฟังก์ชันการทำงานของเครื่อง (Block

diagram)

2. เลือกอุปกรณ์และวงจรที่ต้องการใช้
3. ออกแบบวงจรใช้งานแต่ละภาค
4. ทดลองวงจรแต่ละภาค และปรับปรุงแก้ไข

ขั้นตอนการดำเนินงานในระยะที่ 2 (Project II)

1. นำวงจรในแต่ละภาคมาต่อร่วมกัน และทดลองการทำงานร่วมกันของวงจรต่าง ๆ

2. ออกแบบลายวงจร และประกอบอุปกรณ์ลงบนแผ่นวงจรพิมพ์
3. ดำเนินงานในส่วนของ Software
4. ทดลองการทำงานของ Program ปรับปรุงแก้ไขจนเสร็จ

สมบูรณ์

5. ประกอบอุปกรณ์ทุกส่วนเข้าด้วยกัน และทดลองการทำงานร่วมกับ Program

1.5 การศึกษาค้นคว้าหาข้อมูลในส่วนที่เกี่ยวข้องกับโครงการ

ในการพัฒนาระบบโทรศัพท์ที่มีใช้อยู่ในปัจจุบัน ให้สามารถใช้งานในการสื่อสารข้อมูล ได้กว้างขึ้นโดยสามารถฝากข้อความต่าง ๆ ผ่านคู่

สายโทรศัพท์ มาบันทึกลงบนเทปบันทึกเสียงได้นั้น ปัจจุบันมีเครื่องมือชนิดนี้จำหน่ายอยู่ในท้องตลาด แต่โครงการนี้เป็นการบินบันทึกเสียงผ่านคู่สายโทรศัพท์ที่ผู้โทรเข้ามาสามารถฝากข้อความถึงผู้อื่นได้ถึง 5 คน โดยข้อความทั้ง 5 นี้จะถูกแบ่งเป็นอิสระออกจากกัน และเจ้าของเครื่องสามารถโทรเข้ามาขอรับฟังข้อความได้จากภายนอก หรือจะขอฟังข้อความได้จากการกดปุ่มบนหน้าปัทม์ของตัวเครื่องก็ได้

จากเหตุผลดังกล่าว จึงมีการศึกษาเป็น 2 ส่วน คือส่วนที่ได้รับ การออกแบบและพัฒนาจนใช้งานได้แล้ว และส่วนที่ต้องออกแบบเพิ่มเติมเอง ในส่วนแรกนั้นก็เพียงแต่ศึกษาทำความเข้าใจ การทำงานและดัดแปลงบางส่วน เพื่อให้เหมาะสมและใช้งานได้กับโครงการ ในส่วนที่ต้องออกแบบเองนั้นใช้วิธี กำหนดหน้าที่การทำงานก่อน แล้วจึงหาข้อมูลเพื่อมาออกแบบ

จากการศึกษาข้อมูลจากระบบต่าง ๆ ที่เกี่ยวข้องพบข้อสรุปว่า ควรใช้ Microprocessor มาควบคุมระบบการทำงานทั้งหมด คือควบคุม การทำงานของเครื่องบันทึกเทป และต้องทำให้ Microprocessor รับรู้สภาวะต่าง ๆ ของคู่สายโทรศัพท์ จึงต้องศึกษาข้อมูลเกี่ยวกับวงจรต่าง ๆ ที่ จะนำมาใช้ร่วมกับการทำงานของระบบ โดยการหาข้อมูลเกี่ยวกับวงจร หรือ อุปกรณ์ในท้องตลาดที่สามารถนำมาประยุกต์ใช้ได้กับโครงการนี้ ซึ่งเป็น การลดปัญหาในการหาซื้ออุปกรณ์ไปได้

1.6 โครงสร้างของระบบและหลักการทำงาน

การทำงานของระบบ แบ่งตามลักษณะการทำงานได้ 3 ส่วน คือ

1. ส่วนควบคุม (Z-80 CP-A Board)

ส่วนควบคุมทำหน้าที่รับสัญญาณ จากวงจรตรวจสอบสัญญาณ คือ วงจร Hook detection , Voice detection , Ring detection , DTMF decoder , และ Tape detection รับเข้ามาทำการประมวลผล เพื่อสั่งงานต่อไป

2. ส่วนตรวจสอบ (Sensor)

วงจรตรวจสอบ มีดังนี้

วงจร Ring detection ทำการ detect สัญญาณ
เรียกที่เข้ามา เพื่อส่งให้ CPU ทำการนับ

- วงจร Voice detection ทำหน้าที่เป็นสวิทช์สัญญาณ
เสียงพูด เพื่อส่งให้ CPU ทราบและเริ่มบันทึกเสียงลงบนเทปบันทึกเสียง

- วงจร Hook detection ทำหน้าที่ตรวจสอบการวางหู
/ยกหู โทรศัพท์ เพื่อส่งสัญญาณให้ CPU ทราบ และหยุดทำงานในกรณีที่มี
คนยกหูรับก่อนที่เครื่องบันทึกเริ่มทำงาน (ก่อนสัญญาณกระดิ่ง 5 ครั้ง)

- วงจร DTMF detection ทำหน้าที่แปลงสัญญาณความถี่
จากหน้าปัดโทรศัพท์ ให้อยู่ในรูป ดิจิตอล เพื่อส่งให้ CPU รับรู้จากการกด
Keys หน้าปัดโทรศัพท์ และสิ่งงานต่าง ๆ ในระบบ

- วงจร Tape detection ทำหน้าที่ตรวจสอบการหมุน
การหยุดหมุน และการสิ้นสุดของ Tape บันทึกเสียง เพื่อให้การทำงานของ
เทปเสียงกับ CPU สัมพันธ์กัน

3. ส่วนปฏิบัติการ ทำหน้าที่รับสัญญาณการควบคุมต่าง ๆ ที่ส่งมา
จาก CPU มาปฏิบัติงาน ประกอบด้วยการควบคุม Relays, Solinoid
และสัญญาณการควบคุมวงจรภาคต่าง ๆ

1.7 การทำงานของ Block diagram

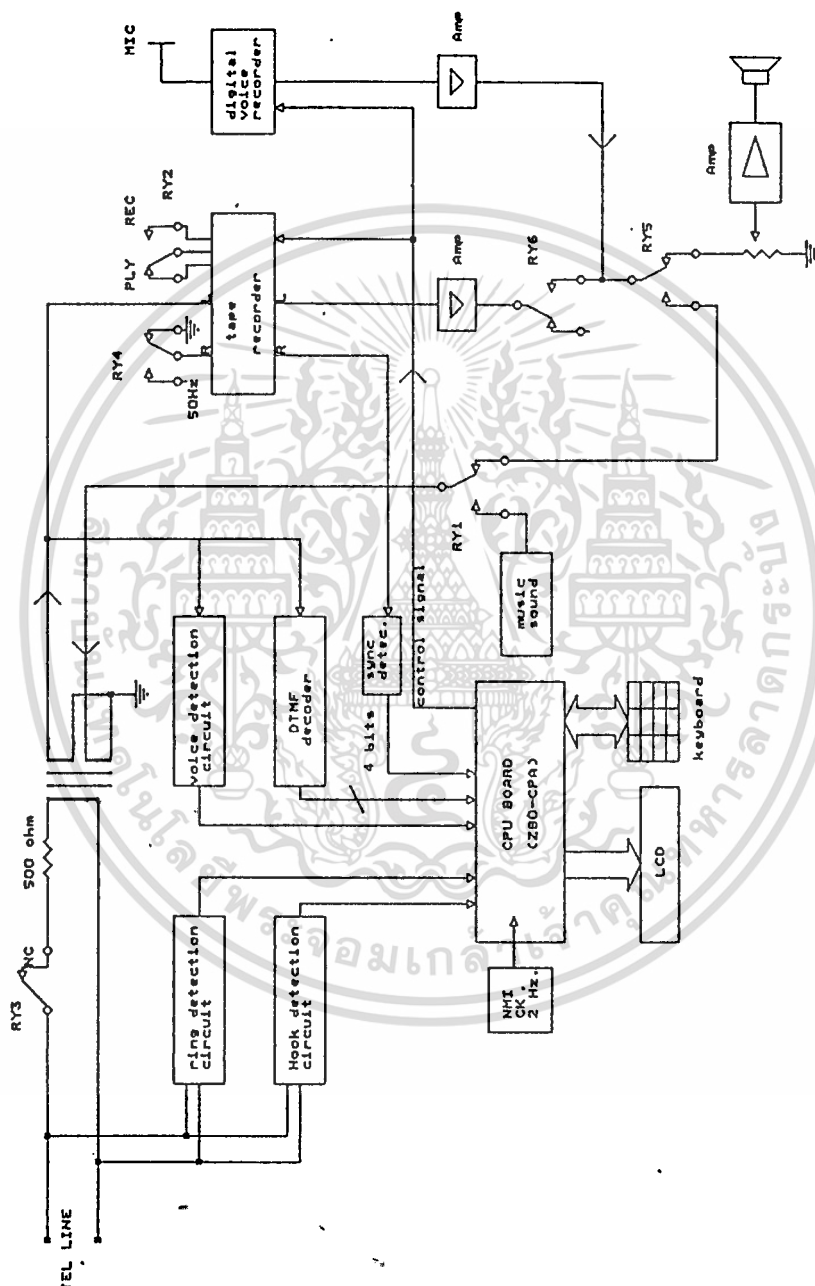
จากอินพุตที่ต่ออยู่กับ Telephone line ของ TOT และต่อ
ขนานกับเครื่องโทรศัพท์ เมื่อมีสัญญาณกระดิ่ง (Ringing Tone) เข้ามา
วงจร Ringing Detection จะทำการตรวจสอบสัญญาณกระดิ่งลูกแรก
ส่งไปบอก CPU ให้เริ่มจับเวลา เมื่อเวลาผ่านไปจนครบตามเวลาที่ได้กำ
หนดไว้ CPU ก็จะมีเริ่มสั่งงานให้ระบบทำงาน ในกรณีที่มีการยกหูโทรศัพท์
ก่อนที่จะครบกำหนดเวลา วงจร Hook Detection ซึ่งทำหน้าที่ตรวจสอบ
การยกหู ก็จะส่งสัญญาณไปบอก CPU ให้เลิกทำงาน ระบบก็จะเข้าสู่สภาวะ
Normal

ในกรณีที่ CPU จับเวลาจนครบ ก็จะสั่งให้ระบบเริ่มทำงาน
โดยขั้นแรกสั่งให้ Relay 3 ON เพื่อต่อ Line เข้ากับระบบ โดยให้

Matching Transformer T_1 หลังจากนั้นก็จะส่งให้วงจร Digital Voice Recorder ส่งข้อความตอบรับ โดยเป็นคำแนะนำเบื้องต้นเพื่อให้ผู้ที่โทรเข้ามาทราบ ซึ่งข้อความที่บันทึกอยู่ในวงจร Digital Voice Recorder ถูกบันทึกไว้เป็น Block ซึ่ง CPU สามารถที่จะเลือก Block ในการนำข้อความออกมาได้ โดยผ่านวงจร Amp, RY_0 และ RY_1 ตามลำดับ เมื่อจบข้อความตอบรับชุดแรก ผู้ที่โทรเข้ามาสามารถที่จะเลือกฝากข้อความหรือขอฟังข้อความ โดยการกด Key จากหน้าปัดโทรศัพท์ ซึ่งสัญญาณที่ถูกกดเข้ามาจากหน้าปัดโทรศัพท์ จะผ่านเข้ามาทางวงจร DTMF Decoder ทำการแปลงสัญญาณ DTMF เป็นสัญญาณ Digital เพื่อเป็น DATA ให้กับ CPU CPU ก็จะตรวจสอบและทำงานให้ตรงกับที่ได้โปรแกรมไว้

ในกรณีที่ผู้โทรเข้ามาต้องการฝากข้อความ CPU ก็จะเอาข้อความในบล็อกต่อไปซึ่งเป็นคำแนะนำให้กดเบอร์ประจำตัวของผู้ที่ต้องการจะฝากข้อความถึง เมื่อจบข้อความ CPU ก็จะรอรับรหัสจากผู้โทรเข้ามา เมื่อ CPU ได้รับรหัส ก็จะสั่งให้ชุด Tape Recorder ซึ่งได้รับการ Format ไว้ก่อนหน้านี้แล้ว เตรียมพร้อมเพื่อรอการบันทึกข้อความ หลังจากนั้น CPU ก็จะส่งข้อความตอบรับในบล็อกต่อไป ซึ่งจะบอกว่าจะเริ่มพูดฝากข้อความได้ CPU ก็จะรอรับโดยการตรวจสอบจากวงจร Voice Detection ว่าเริ่มพูดหรือยัง เมื่อเริ่มพูดวงจร Voice Det. ก็จะส่งสัญญาณไปบอก CPU CPU ก็จะสั่งให้ชุด Tape Recorder เริ่มทำการบันทึก

เมื่อผู้โทรเข้ามาต้องการขอฟังข้อความ CPU ก็จะนำข้อความตอบรับบอกให้กดรหัส เจ้าของช่องข้อความ (4 หลัก) CPU ก็จะค้นหารหัสที่ได้โปรแกรมไว้เมื่อพบว่าตรงกับรหัสใดก็จะทำการเช็คว่ามีข้อความฝากไว้หรือไม่หรือว่ามีข้อความฝากไว้กี่ช่อง ถ้าไม่มีข้อความฝากไว้เลย CPU ก็จะนำข้อความบล็อกที่บอกให้ทราบว่าไม่มีข้อความฝากไว้ ระบบก็จะกลับเข้าสู่สภาวะ NORMAL แต่ถ้ามีข้อความฝากไว้ CPU ก็จะบอกให้รอสักครู่ และทำการรอกเทปเพื่อค้นหาช่องที่ถูกบันทึกข้อความ ของเจ้าของรหัสนั้น โดยนับจากสัญญาณ Sync ที่บันทึกไว้ตอนทำการ Format เพื่อ PLAY ให้ฟังทีละช่อง เมื่อ PLAY ให้ฟังหมดทุกช่องแล้ว CPU ก็จะนำข้อความที่บอกให้ทราบว่าจบข้อความแล้ว และกลับเข้าสู่สภาวะปกติ ช่องที่ถูกขอฟังไปแล้วก็จะกลายเป็น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ช่องว่างที่สามารถฝากข้อความทับไปได้อีก

ในกรณีที่กดรหัสผิด CPU ก็จะยกเลิกการทำงานเลยและกลับเข้าสู่สภาวะปกติ จากการทำงานที่ผ่านมา CPU จะนำสถานะของช่องบันทึกข้อความแต่ละช่องมาตลอด เพื่อใช้เป็น Reference ซึ่งเป็นรายละเอียดในทาง Software

ในกรณีที่ต้องการเซทโปรแกรมต่าง ๆ ที่เครื่องก็สามารถทำได้ โดยการกด Key board จากหน้าเครื่อง รายละเอียดของ Key board และ Function การทำงานต่าง ๆ จะถูกกำหนดโดย Software และ Display จะทำหน้าที่แสดงผลเป็นข้อความซึ่งบอกถึงสถานะของระบบในสภาวะต่าง ๆ

ส่วนวงจร Music Sound จะทำหน้าที่กำเนิดเสียงดนตรีออกทางคู่สายโทรศัพท์ เมื่อมีการรอกการทำงานของระบบ โดยการสั่งจาก CPU ผ่าน RY₁

1.8 การออกแบบโครงสร้างของระบบ

จากการศึกษาข้อมูลของวงจรและอุปกรณ์ในท้องตลาด ทำให้ได้วงจรสำเร็จรูปที่สามารถนำมาใช้ได้โครงงานนี้ เพื่อทำให้การสร้างโครงงานนี้สะดวกขึ้น และเป็นการสานต่อการพัฒนาทางด้านเทคโนโลยี จึงได้วงจรสำเร็จรูปที่สามารถนำมาใช้ได้เลย ได้แก่

1. Z-80 CP-A Board ของบริษัท ETT
2. Digital Recorder
3. LCD Display and Controller

ในการนำวงจรสำเร็จรูปมาใช้กับโครงงาน ได้คำนึงถึงปัญหาที่อาจเกิดขึ้น ในการเชื่อมโยงของแต่ละส่วนเข้าด้วยกัน ซึ่งจะอาศัยการแก้ไขปัญหาด้วยการออกแบบเพิ่มเติมภายนอก และแก้ไขเล็ก ๆ น้อย ๆ ในวงจร ในการนำวงจรเหล่านี้มาใช้งานก็โดยการติดตั้งชุดสำเร็จเลย ซึ่งรายละเอียดจะได้กล่าวต่อไป

ส่วนของวงจรที่ต้องค้นคว้าและออกแบบเองเพิ่มเติม ก็คำนึงถึง

การเชื่อมต่อกับวงจรชุดสำเร็จรูปที่ได้กล่าวมาแล้วเป็นหลัก ซึ่งประกอบด้วย ส่วนต่าง ๆ ดังนี้

1. ภาควัดตรวจสอบการยกหู/วางหูโทรศัพท์ (Hook detection circuit)
2. ภาควัดตรวจสอบสัญญาณเสียง (Voice detection circuit)
3. ภาควัดตรวจสอบสัญญาณกระดิ่ง (Ring detection circuit)
4. ภาควัดถอดรหัสจากสัญญาณโทรศัพท์ (DTMF decoder)
5. ภาควัดติดต่อระหว่างอุปกรณ์ภายนอกกับ CP-A Board
6. ภาควัดกำเนิดสัญญาณการ Interrupt (NMI)
7. ภาควัดควบคุมการทำงานของเครื่องเล่นเทป
8. ภาควัดตรวจสอบการทำงานของเครื่องเล่นเทป
9. วงจรปรับเทป เล่น/บันทึก
10. ภาควัดกำเนิดเสียงดนตรี (Music sound)
11. ภาควัดขยายเสียง (Amplifier)

บทที่ 2

ทฤษฎีการทำงานและการออกแบบวงจร

2.1 DTMF Decoder

โทรศัพท์ชนิดกดปุ่มแบบ DTMF ประกอบด้วยปุ่มกดจำนวน 12 ปุ่ม มีการทำงานเป็นแบบ Dual Multi Frequency แป้นกดจะแบ่งเป็น Row 4 แถว Column 4 แถว ประกอบกันเป็นรูป Matrix ในแต่ละ Row และ Column จะมีค่าความถี่ประจำตำแหน่งอยู่ ดังรูปที่ 1

			C1	C2	C3	
			1209Hz	1336Hz	1447Hz	
1	2	3	697Hz	1	2	3
4	5	6	770Hz	4	5	6
7	8	9	852Hz	7	8	9
*	0	#	941Hz	*	0	#

รูปที่ 1 ค่าความถี่ประจำหมายเลข ในระบบโทรศัพท์

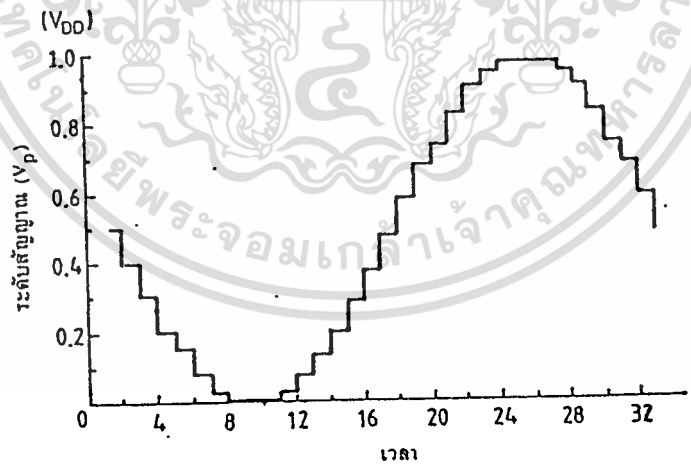
การทำงาน เมื่อกดปุ่มหมายเลขใดหมายเลขหนึ่ง จะประกอบไปด้วย Tone เสียง 2 ความถี่ด้วยกันคือ ความถี่สูงและความถี่ต่ำ ซึ่งแต่ละหมายเลขจะให้ค่าความถี่คู่ต่างกัน จากปุ่มกดจะพบว่าหมายเลข 1, 4, 7, และ * อยู่ใน Column ที่ 1 โดยหมายเลข 1, 2, 3 อยู่ใน Row ที่ 1 ตัวเลขแต่ละตัวเป็นการพบกันของความถี่ทาง Row (ความถี่ต่ำ) กับความถี่ทาง Column (ความถี่สูง) ยกตัวอย่างเช่นเมื่อกดปุ่มเลข "5" จะอยู่ใน Column ของ 1336 Hz และ Row 770 Hz ดังนั้นหากกดปุ่มเลข

"5" จะผลิตสัญญาณ Output 2 ความถี่ออกมา ซึ่งเรียกว่า DTMF

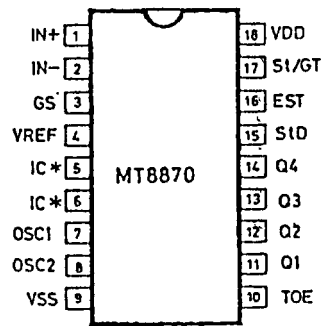
ดังนั้น ในการสร้างวงจรถอดรหัส DTMF ก็โดยการสร้างวงจรถอดรหัสค่าความถี่เหล่านี้นั่นเอง ในปัจจุบันได้มีอุปกรณ์ที่ทำหน้าที่นี้โดยเฉพาะ คือเป็น IC เบอร์ MT8870 ของบริษัท INTEL ซึ่งจะทำหน้าที่รับสัญญาณ DTMF มาแปลงให้เป็นค่าตัวเลขฐานสองขนาด 4 บิต ดังนั้นในโครงการนี้จึงได้เลือกใช้ IC เบอร์นี้มาทำเป็นวงจรถอดรหัส DTMF

2.1.1 MT8870 DTMF Decoder

IC เบอร์ MT8870 นี้ จะทำหน้าที่รับสัญญาณ DTMF ซึ่งเป็นสัญญาณ Analog 2 ความถี่ได้อย่างดีถึงแม้ว่าสัญญาณที่ได้จากการกดปุ่มของโทรศัพท์แบบทซ์โทนจะมีลักษณะไม่เป็นคลื่น Sine ที่แท้จริง ดังรูปที่ 2 IC เบอร์นี้ก็ยังสามารถยอมรับและถอดรหัสออกมาได้



รูปที่ 2 ลักษณะสัญญาณ DTMF ที่ได้จากโทรศัพท์



* ต่อกับ VSS

MT8870BE 18 PIN PLASTIC
MT8870BC 18 PIN CERDIP

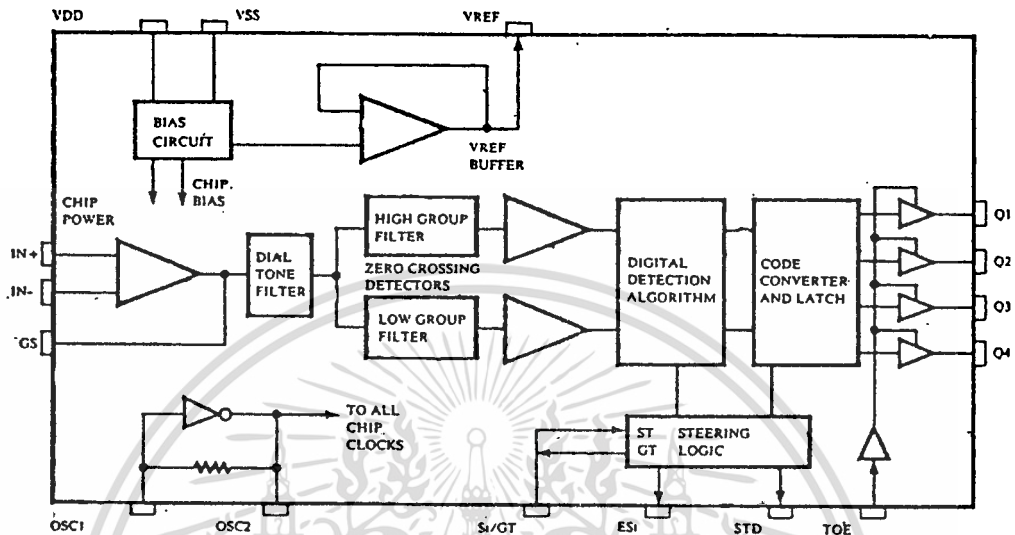
รูปที่ 3 แสดงรายละเอียดขาของ MT8870

2.1.2 คุณสมบัติของ MT8870

- เป็นตัวรับและถอดรหัสความถี่ DTMF
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดไทม์ (Guard time) ได้
- เป็นไอซีคุณภาพสูง

2.1.3 โครงสร้างภายในของ MT8870

MT8870 ประกอบไปด้วยวงจรกรองความถี่ และวงจรถอดรหัส ฟังก์ชันทางดิจิทัล ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิทช์ Capacitor Filter สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิทัล เพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ ออกเป็นเลขฐานสองขนาด 4 บิต และตรวจสอบช่วงเวลาสัญญาณเข้ามา ส่วนภาคอินพุทเป็นออปแอมป์ที่สามารถปรับอัตราขยายได้ โดยต่ออุปกรณ์ภายนอก เอาท์พุทเป็นวงจรแลทซ์ 3 สถานะ



รูปที่ 4 โครงสร้างภายในของ MT8870

2.1.4 ฟังก์ชันการทำงานภายใน

ภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วนคือ

1. ภาคกรองความถี่ (Filter section) ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่คือช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิดสวิทช์คาปาซิเตอร์ (six-order switched capacitor band pass filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วงคือช่วงความถี่สูงและช่วงความถี่ต่ำ

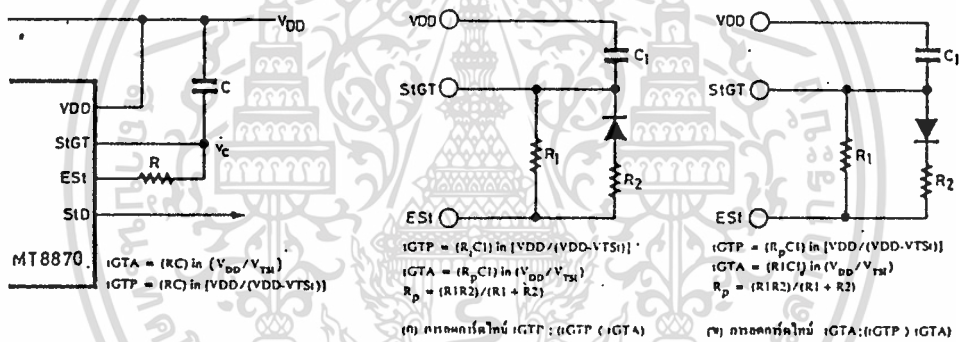
2. ภาคถอดรหัส (Decoder section)

F_{LOW}	F_{HIGH}	NO	TOE	Q_4	Q_3	Q_2	Q_1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1447	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1447	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1447	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1447	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

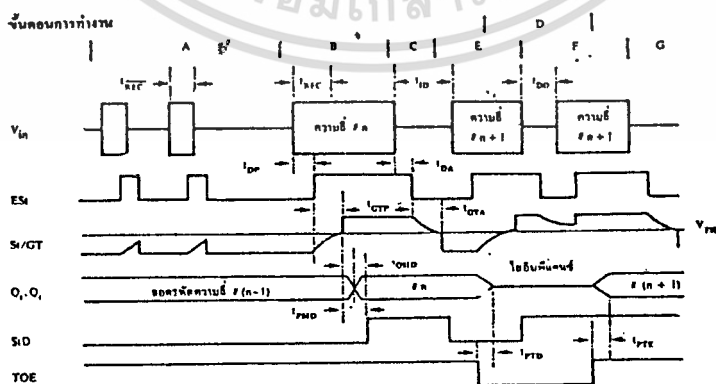
รูปที่ 5 แสดงค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ ความถี่ DTMF ที่ถูกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิทัล และมีการตรวจสอบความถี่ที่เข้ามาว่า เป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่หา Est (early steering) ก็จะถูกแอดคัพ สำหรับค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ นั้น แสดงในรูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ภาคตรวจสอบสัญญาณ (Steering circuit) ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุท จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา EST จะเป็น "high" นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 6 เมื่อขา EST เป็น "high" ทำให้ Vc สูงขึ้น ตัวเก็บประจุ C จะคายประจุ ทำให้แรงดัน Vc สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสจึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงานสามารถดูได้จาก Timing diagram ในรูปที่ 7



รูปที่ 6 แสดงวงจรตรวจสอบสัญญาณ และการกำหนดเวลาการ์ด์ไทม์



รูปที่ 7 แสดงแผนภูมิเวลา (timing diagram) ของ MT8870

อธิบายขั้นตอนการทำงาน

- A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาท์พุท
ไม่เปลี่ยน
- B - ความถี่ #n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่
ถูกถอดรหัสและแลทช์ไว้ที่เอาท์พุท
- C - จบความถี่ #n ช่วงห่างถูกต้อง เอาท์พุทยังคงแลทช์อยู่
จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่
- D - เอาท์พุทเปลี่ยนเป็นไฮอิมพีแดนซ์
- E - ความถี่ #n+1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่
ถูกถอดรหัสและแลทช์ไว้
- F - ความถี่ #n+1 หายไป ช่วงห่างไม่ถูกต้องเอาท์พุทยังคง
แลทช์อยู่
- G - จบความถี่ #n+1 ช่วงห่างถูกต้องเอาท์พุทยังคงแลทช์อยู่
จนถึงความถี่ใหม่ที่ถูกต้อง

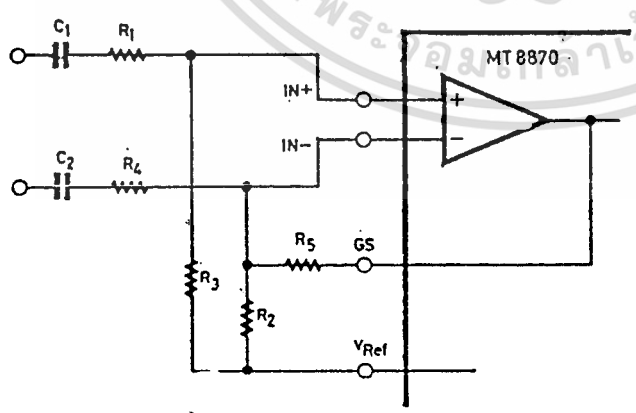
สัญลักษณ์

- V_{in} - สัญญาณความถี่ DTMF ที่เข้ามา
- EST - Early steering output ใช้แสดงความถี่ที่ถูกต้อง
- St/GT - Steering input/Guard time output สำหรับ
ต่อกับ RC ภายนอก
- Q_1-Q_4 - เอาท์พุท BCD ขนาด 4 บิต
- StD - Delayed Steering output ใช้แสดงค่าความถี่ที่
ได้รับหรือหายไป มีคาบ เวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ
- TOE - Tone output enable (input) ใช้ควบคุม Q_1-
 Q_4 ให้เป็นไฮอิมพีแดนซ์
- t_{REC} - คาบเวลานานที่สุด ที่ตรวจพบความถี่ DTMF แล้วยัง
ไม่ถูกต้อง
- t_{REC} - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง
- t_{ID} - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

- t_{DO} - เวลนานานสุด ที่ยอมให้สัญญาณหายไปได้ ในคาบเวลาความถี่ที่ถูกต้อง
- t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง
- t_{DA} - เวลาที่ใช้ในการตรวจการหายไป ของสัญญาณความถี่ DTMF ที่ถูกต้อง
- t_{GTP} - การ์ดไทม์ของการปรากฏความถี่ DTMF
- t_{GTA} - การ์ดไทม์ของการหายไปของความถี่ DTMF

สำหรับคำว่าการ์ดไทม์(Guard time) นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากันหรือมากกว่า ช่วงเวลาที่เรที่ตั้งไว้ จึงจะได้รับการยอมรับว่า สัญญาณความถี่นั้นถูกต้อง หรือพูดได้ว่าเวลาที่เรที่ตั้งไว้โดย RC ก็คือ การ์ดไทม์นั่นเอง เมื่อสัญญาณความถี่เข้ามา นานเท่าหรือมากกว่าเวลาที่ตั้งไว้ จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่เข้ามาสั้นกว่าก็ จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลาและคำนวณเวลาดูได้จากรูปที่ 6

4. ภาคขยายสัญญาณความแตกต่าง(Differential Input) วงจรส่วนอินพุทของ MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยาย โดยต่อวงจรภายนอกเพิ่มเข้าไปดังรูปที่ 8 ซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอินพุทและอิมพีแดนซ์ได้ดังนี้



ภาคขยายความแตกต่างคั้งอินพุท .

$C_1 = C_2 = 10 \text{ nF}$

$R_1 = R_4 = R_5 = 100 \text{ K}\Omega$ ค่าผิดพลาด $\pm 1\%$

$R_2 = 60 \text{ K}\Omega, R_3 = 37.5 \text{ K}\Omega$ ค่าผิดพลาด $\pm 5\%$

$R_2 = \frac{R_1 R_4}{R_2 + R_4}$

อัตราขยายแรงดัน ($A_v \text{ diff}$) = $-\frac{R_5}{R_1}$

อิมพีแดนซ์ที่แค้นซ์

$(Z_{INDIFF}) = 2 \sqrt{R_1^2 + \left(\frac{1}{\omega C}\right)^2}$

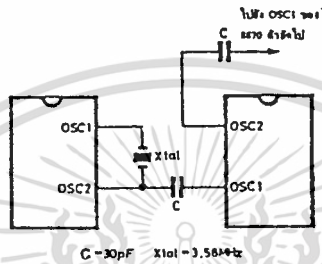
รูปที่ 8 แสดงการต่อวงจรภาคขยายสัญญาณความแตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{อัตราขยาย (} A_v \text{ diff)} = R_2/R_1$$

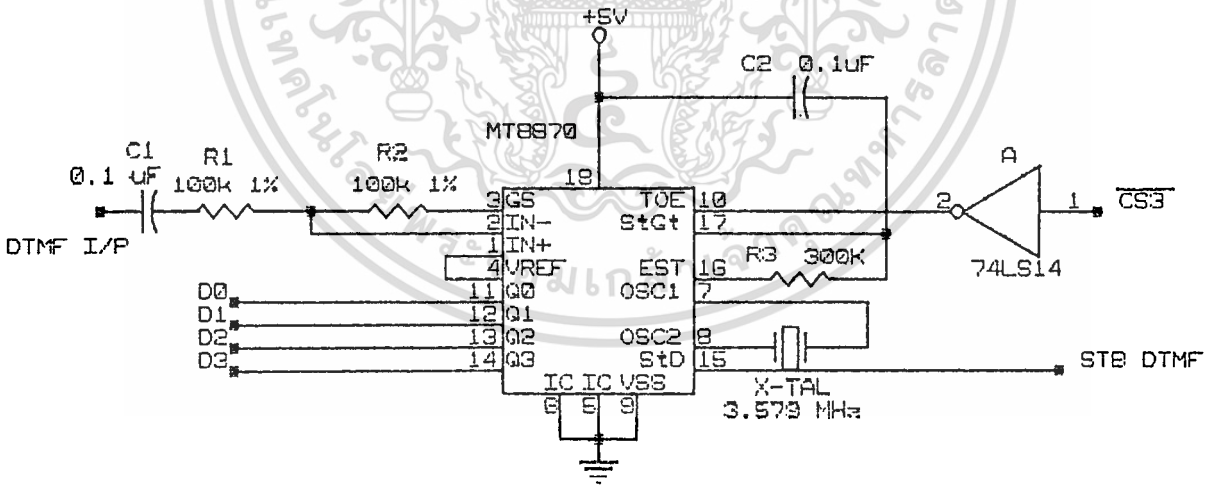
$$\text{อินพุทอิมพีแดนซ์ (} Z_{in} \text{ diff)} = 2/R_1 \text{ }^2 + (1/wc) \text{ }^2$$

5. ภาคกำเนิดความถี่ (Oscillator) ในภาคนี้ภายในไอซีมี วงจรเวลาอยู่ภายใน เพียงแต่ต่อ X-TAL ขนาด 3.58 MHz ก็สามารถใช้ งานได้ทันที ลักษณะการต่อวงจรดังแสดงในรูปที่ 9



รูปที่ 9 แสดงการต่อวงจรผลิตความถี่

2.1.5 วงจรใช้งาน



รูปที่ 10 แสดงวงจรใช้งานของ MT8870

จากคุณสมบัติและโครงสร้างของ MT8870 ที่กล่าวมา จึงได้นำ มาประกอบเป็นวงจรดังรูปที่ 10 ซึ่งจะมีการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณความถี่ DTMF จะเข้ามาทางขา non-inverting input (ขา 2) โดยผ่าน C_1 เป็นตัว coupling สัญญาณและจะมี R_1 , R_2 เป็นตัวปรับอัตราขยายให้เหมาะสม ส่วน R_3, C_2 จะเป็นตัวกำหนดการดีไทม์เอาท์พุท ในที่นี้ใช้ค่า $R_3 = 300 \text{ kohm}$, $C_2 = 0.1 \text{ uF}$ ซึ่งจะได้ค่าการดีไทม์ประมาณ 20 ms

ในขณะที่ MT8870 ทำการ Decode สัญญาณ DTMF ออกมานั้น มันจะส่งสัญญาณ STD ออกมาที่ขา 15 เพื่อไปบอกให้ CPU รู้ว่าขณะนี้ มี DATA ที่จะส่งให้ CPU แล้ว เมื่อ CPU รู้ก็จะส่งสัญญาณ CS_0 มาเข้าที่ขา TOE ของ MT8870 เพื่อเป็น output enable ให้กับ MT8870 ที่ขา TOE นี้หากได้รับ logic "0" จะทำให้ Q_1-Q_4 มีสถานะเป็น high impedance ฉะนั้นหาก CPU รับรู้ว่าการ Decode สัญญาณ DTMF และต้องการ DATA ก็ส่ง CS_0 มาให้ สัญญาณที่ได้จากการ Decode จะออกมาเป็นรหัสเลขฐานสอง โดยเป็น Data D_0-D_9 ให้กับ CPU เพื่อให้ CPU นำไปประมวลผลต่อไป

จากวงจรและการทำงานที่กล่าวมานี้ เมื่อนำมาประกอบวงจรทดลอง เพื่อเลือกค่าอุปกรณ์ RC ที่เหมาะสม ปรากฏว่าค่าของ R_3 และ C_2 นั้นมีความสำคัญมาก เพราะจะเป็นการกำหนดช่วงเวลาของสัญญาณ DTMF ที่จะยอมรับ ซึ่งค่า C_2 นั้นทางบริษัทที่ผลิตไอซีเบอร์นี้ได้แนะนำให้ใช้ค่า 0.1 uF แล้วเลือกค่า R_3 ตามต้องการ ซึ่งค่า 300 kohm นั้นเป็นค่าที่เหมาะสมที่สุด สำหรับการใช้งานโทรศัพท์ของคนทั่วไปที่จะกดปุ่มโทรศัพท์ และทำให้ไอซี MT8870 รับรหัสตัวเลขได้ทุกตัวและถูกต้อง

2.2 Digital Voice Recorder

2.2.1 แนวความคิด

1. ต้องปราศจากกลไกต่าง ๆ ในการทำงาน
2. สามารถบันทึกได้ติดต่อกันไม่ต่ำกว่า 2 นาที
3. สามารถเรียกข้อมูลกลับมาใช้ใหม่ได้ อย่างรวดเร็ว

4. สามารถกำหนดตำแหน่งในการบันทึกได้ และเรียกใช้ข้อมูลแต่ละตำแหน่งได้

5. สามารถใช้งานร่วมกับชุด CPU ได้

2.2.2 วงจรใช้งาน

การบันทึกเสียงสำหรับการนำกลับมาใช้ใหม่ โดยใช้วิธีการทาง Digital ทำให้ไม่ต้องใช้กลไกเคลื่อนไหวใด ๆ ดังเช่นแบบเดิม แต่จะใช้อุปกรณ์หน่วยความจำเก็บข้อมูลแทนเส้นเทป หรือแผ่นเสียง เทคนิคการบันทึกเสียงด้วยระบบ Digital มีด้วยกันอยู่หลายวิธี แต่ในโครงการนี้จะใช้ระบบ Delta Modulation

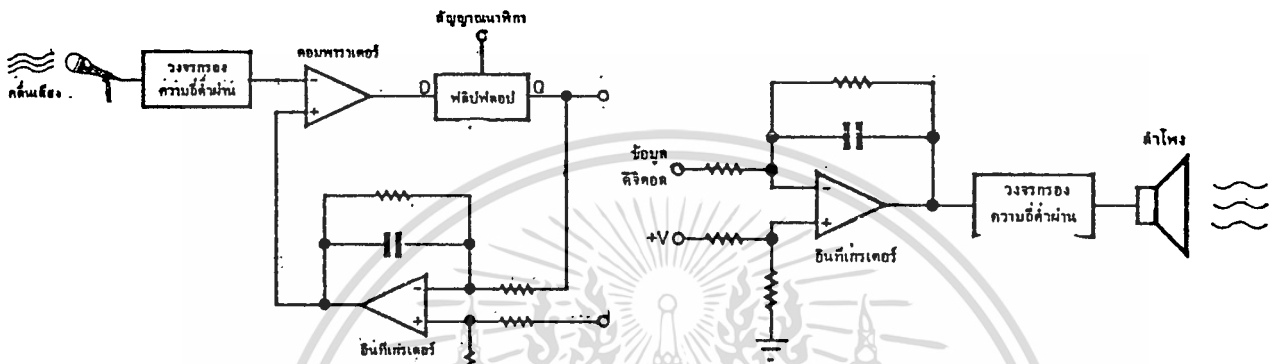
ในการบันทึกเสียงระบบดิจิทัล ไม่ว่าจะใช้วิธีการอะไรก็ตาม จะประกอบด้วย 3 ส่วนสำคัญคือ ส่วนแรกทำหน้าที่แปลงสัญญาณเสียงที่เป็นอนาลอก ให้เป็นข้อมูลดิจิทัล แล้วจึงนำไปบันทึกลงในหน่วยความจำซึ่งเป็นส่วนที่สอง และส่วนสุดท้ายคือส่วนที่ทำหน้าที่แปลงข้อมูลดิจิทัลจากหน่วยความจำออกมาเป็นสัญญาณอนาลอก

2.2.3 Delta Modulation

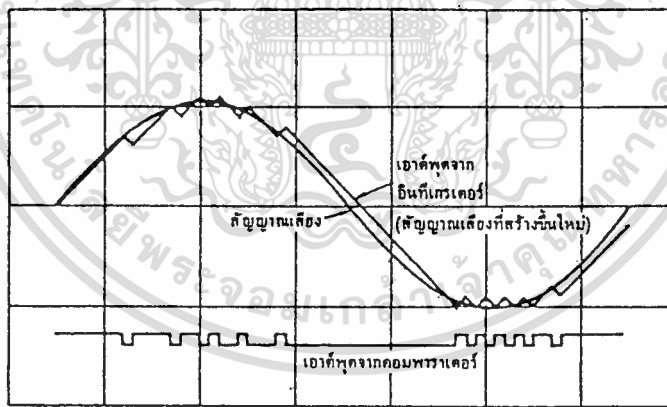
เทคนิคของ Delta Modulation จะไม่ใช้การสุ่มสัญญาณหนึ่งจุดแล้วแปลงเป็นข้อมูลดิจิทัลหนึ่งเวิร์ด ที่มีความละเอียดเป็นจำนวนบิตที่ต้องการ แต่จะใช้วิธีเปรียบเทียบความสูงหรือการเปลี่ยนแปลงของสัญญาณเสียงแทน ข้อมูลที่ได้ก็คือทิศทางของการเปลี่ยนแปลงซึ่งก็มีเพียงขึ้นหรือลงเท่านั้น ดังนั้นความกว้างของข้อมูลดิจิทัลจึงใช้เพียงบิตเดียวก็เพียงพอ ข้อดีของวิธีการนี้คือ ใช้หน่วยความจำน้อยกว่าวิธีการแบบอื่น ๆ

จากรูปที่ 11 เป็นวงจรเบื้องต้นของ Delta Modulation คอมพิวเตอร์จะทำหน้าที่เปรียบเทียบสัญญาณอินพุตปัจจุบัน กับอินพุตก่อนหน้า ซึ่งได้จากการป้อนกลับมายังอินติเกรเตอร์ เอาท์พุทจากการเปรียบเทียบถูกป้อนผ่านฟิลิปปลอปที่ควบคุมด้วยสัญญาณนาฬิกา เพื่อให้ได้เป็นข้อมูลดิจิทัล ซึ่ง

ก็คือการกำหนดอัตราการสุ่มสัญญาณนั่นเอง สัญญาณที่ได้จากตัวเปรียบเทียบและจากอินทิเกรเตอร์ เปรียบเทียบกับสัญญาณอินพุท แสดงในรูปที่ 12 ลักษณะเช่นนี้จะพบว่ายิ่งความถี่ของสัญญาณนาฬิกามีค่าสูง ก็ยังสามารถบันทึกการเปลี่ยนแปลงที่แคบได้มากขึ้น ทำให้ได้คุณภาพเสียงที่ดีขึ้น แต่ก็สิ้นเปลืองหน่วยความจำมากขึ้นตามไปด้วย



รูปที่ 11 วงจรเบื้องต้นของ Delta Modulation



รูปที่ 12 เปรียบเทียบสัญญาณอินพุทกับสัญญาณอนาลอกที่ได้จากอินทิเกรเตอร์

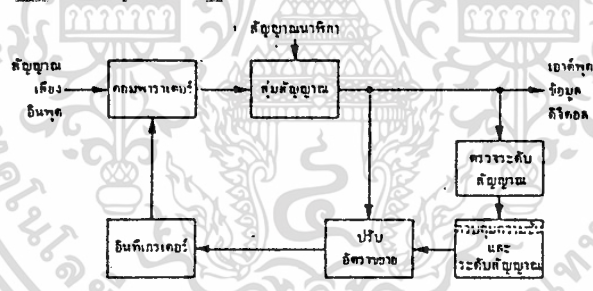
2.2.4 CVSD

ข้อจำกัดของวิธีการ Delta Modulation ก็คือ แลกกว้างความถี่ที่ใช้งาน ซึ่งถูกจำกัดโดยความถี่สัญญาณนาฬิกา และจะสูงกว่าความถี่สูงสุดของสัญญาณอินพุท มากกว่า 2 เท่าขึ้นไป อีกอย่างหนึ่งคือ ความ

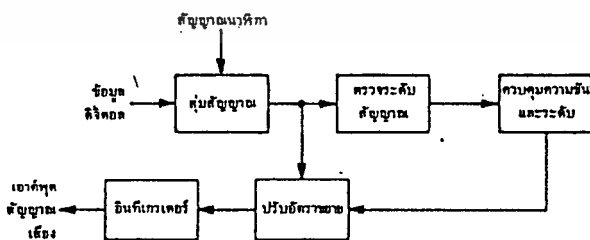
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เร็วของการเปลี่ยนแปลงความสูงของสัญญาณ หรือ ไดนามิกเรนจ์ ระบบ Delta Modulation ขรรคมามีค่าไดนามิกเรนจ์ที่แคบ จำเป็นต้องมีส่วนเพิ่มเติมทำหน้าที่ขยายไดนามิกเรนจ์ให้กว้าง โดยการควบคุมอัตราขยายของอินทิเกรเตอร์ เพื่อให้ตอบสนองต่อสัญญาณที่มีความชันมาก ๆ ได้ทัน ระบบนี้มีชื่อเรียกใหม่ว่า ระบบ Delta Modulation แบบเปลี่ยนแปลงความชันต่อเนื่อง หรือ CVSD (Continuous Variable Slope Delta modulation)

ระบบ CVSD ทั้งส่วนแปลงจากอนาลอกเป็นดิจิตอล และส่วนแปลงกลับจากดิจิตอลเป็นอนาลอก แสดงในรูปที่ 13 และ 14 ตามลำดับ วิธีการของ CVSD ก็คือ มีการตรวจระดับสัญญาณโดยอาจใช้วิธีการจัดให้มีรีจิสเตอร์สำหรับเก็บข้อมูลดิจิตอลล่าสุดจำนวน 3 ถึง 4 บิต แล้วตรวจดูว่าเป็น "0" หกหรือ "1" หกหรือไม่ ถ้าใช่แสดงว่าอัตราขยายของอินทิเกรเตอร์ต่ำเกินไป ตอบสนองต่อความชันของสัญญาณไม่ทัน ก็จะทำการเพิ่มอัตราขยายให้สูงขึ้นเฉพาะในช่วงนั้น ในส่วนของการแปลงกลับก็ต้องมีการทำงาน ในลักษณะเดียวกัน



รูปที่ 13 แผนผังการทำงานของ CVSD ในส่วนของการแปลงจากสัญญาณเสียงเป็นดิจิตอล



รูปที่ 14 แผนผังการทำงานของ CVSD ในส่วนแปลงกลับจากดิจิตอลเป็นสัญญาณเสียง

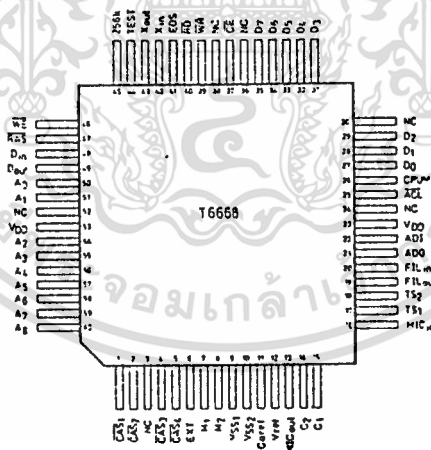
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 ไอซีบันทึกเสียงสำเร็จรูป T6668

ไอซีเบอร์ T6668 นี้เป็นของบริษัท โตชิบา มีฟังก์ชันการทำงานก่อนข้างสมบูรณ์ ประยุกต์ใช้งานได้กว้างขวางและสะดวก มีรูปร่างภายนอกเป็นแบบติดตั้งบนผิวหรือ Surface mount ขนาด 60 ขา ต่อกับหน่วยความจำชนิดไดนามิกขนาด 64 K x 1 บิต หรือ 256 K x 1 บิตได้โดยตรง 4 ตัวใช้คริสตอลควบคุมความถี่ของสัญญาณนาฬิกา เปลี่ยนบิตเรตโดยใช้ดีปสวิทช์ เลือกหน้าของหน่วยความจำแยกบันทึก/เล่นกลับได้ เมื่อใช้หน่วยความจำ 256 K x 1 (41256) จำนวน 4 ตัว ที่บิตเรต 16 K จะบันทึกได้นาน 64 วินาที

T6668 นี้สมบูรณ์ในตัว ทางด้านอินพุตสามารถต่อไมโครโฟนเข้ากับไอซีได้เลย ด้านเอาต์พุตก็เพิ่มภาคขยายอีกส่วนเดียว วงจรที่สมบูรณ์ดังแสดงในรูปที่ 16

รายละเอียดขาใช้งานของ T6668



รูปที่ 15 แสดงตำแหน่งขาของ T6668

- A₀-A₈ : ขาแอดเดรสต่อกับแรม
- D_{in}, D_{out} : ขาดาต้าต่อกับแรม
- RAS, WE : สัญญาณควบคุมแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$CAS_1 - CAS_4$: ขาเลือกแรมแต่ละตัวรวม 4 ตัว

M_1, M_2 : ใช้กำหนดจำนวนแรมที่ใช้ ดังตารางที่ 1

ตารางที่ 1 การกำหนดชนิดและจำนวนแรมที่ใช้

ชนิดและจำนวน	256 K (ขา45)	M_2	M_1
4164x1	0	0	0
4164x2	0	0	1
4164x3	0	1	0
4164x4	0	1	1
41256x1	1	0	0
41256x2	1	0	1
41256x3	1	1	0
41256x4	1	1	1

256K : เลือกขนาดแรม ต่อกราวด์ใช้ 4164 ต่อไฟบวกใช้
41256

EOS : เอาท์พุทเป็น high เมื่อจบข้อความที่บันทึก

MIC_{in}, MIC_{out} : อินพุทและเอาท์พุทของภาคขยายส่วนหน้า

AD_1, AD_0 : อินพุทของสัญญาณอนาลอกที่จะนำไปแปลงเพื่อ
บันทึก และเอาท์พุท อนาลอกที่ได้จากการอ่าน

FIL_{in}, FIL_{out} : วงจรกรองความถี่ต่ำผ่าน

C_1, C_2 : ต่อตัวเก็บประจุภายนอก

ACL : ขารีเซต แอคทีฟโลว์

X_{in}, X_{out} : คริสตัลลออสซิลเลเตอร์ ความถี่ 650 kHz

CPUM, CE : ขาสัญญาณควบคุมสำหรับอินเตอร์เฟสกับ CPU

WR : ขาอินพุทสัญญาณควบคุมสำหรับเปลี่ยนไปโหมดการบันทึก

D_4, D_5 : เริ่มต้น และหยุดการนับของเคาน์เตอร์ภายใน
สำหรับการบันทึก และการเล่นกลับ

D_6, D_7 : กำหนดบิตเรต ดังตารางที่ 2

ตารางที่ 2 การกำหนดอัตราเร็วข้อมูล

บิตเรต	D_7 (ขา 35)	D_6 (ขา 34)
8 K	0	0
11 K	0	1
16 K	1	0
32 K	1	1

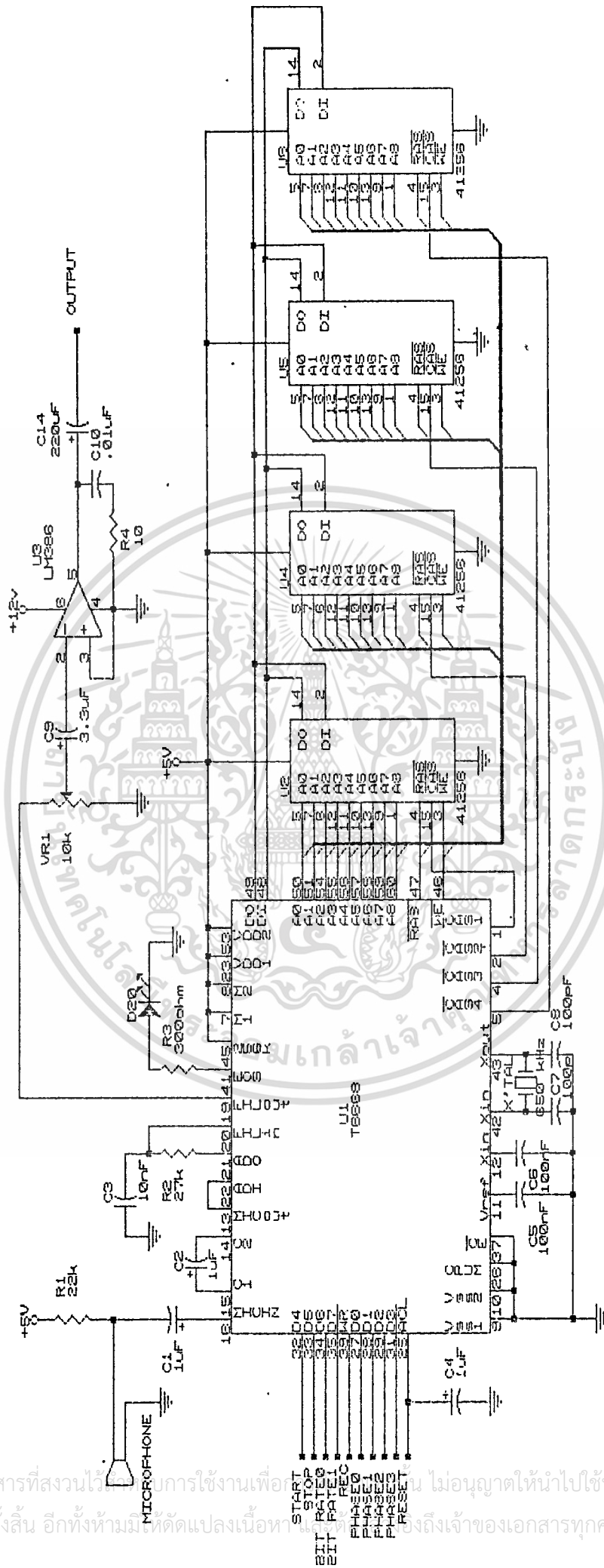
$D_0 - D_9$: เลือกหน้าของหน่วยความจำ สำหรับการบันทึก
แบ่งได้สูงสุด 16 หน้า ตามรหัสเลขฐานสอง แต่ละหน้าไม่กำหนดความยาว
แล้วแต่ จะกด stop(D_{15}) เมื่อใด ก็จะมีการบันทึกเอาไว้โดย อัตโนมัติ

V_{DD}, V_{SS} : ขาไฟเลี้ยงและกราวด์

2.2.6 วงจรการทำงาน

จากคุณสมบัติของ ไอซีเบอร์รี่ T6668 นี้เมื่อนำมาประกอบเป็น
วงจรการทำงานโดยสมบูรณ์ จะได้ดังรูปที่ 16 ซึ่งมีการทำงานดังนี้

T6668 จะทำการรับสัญญาณเสียงพูดเข้ามาทำการขยาย แล้ว
เปลี่ยนจากสัญญาณอนาลอกไปเป็นข้อมูลดิจิทัล แล้วไปเก็บไว้ที่ไดนามิกแรม
โดย CPU ภายในจะทำการเลื่อนแอดเดรสที่จะนำเข้าไปเก็บเองโดยอัตโนมัติ
เมื่อทำการแปลงข้อมูลจาก D/A จะใช้อัตรา 10 bit D/A เพื่อเปลี่ยน
กลับมาเป็นเสียงเช่นเดิม การอัดเข้าไปเราสามารถเลือก speed ได้ 4
speed โดยการเลือกที่ D_6, D_7 จากตารางที่ 1 สำหรับในโครงงานนี้
จะใช้สัญญาณจากชุด CPU มาเป็นตัวกำหนด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีการตัดแปลงเนื้อหาในเอกสารนี้โดยเด็ดขาด รวมถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเลือกให้ทำงานที่ 8 Kbps จะทำให้อัดหรือเล่นได้นาน 128 วินาที ถ้าเป็น 11 Kbps จะได้นาน 93 วินาที ถ้าเป็น 16 Kbps ก็จะได้ 64 วินาที และถ้าเลือกเป็น 32 Kbps ก็จะได้ 32 วินาที

การทดลองใช้ X-TAL 650 KHZ เป็นฐานความถี่และต่อกับ แรม 256 k จำนวน 4 ตัว ทำให้ความจุของ Memory เพิ่มขึ้นเป็น 1 Mbit การอัดเมื่อเราอัดที่ speed ใด speed หนึ่งเสร็จแล้วเราสามารถที่จะนำกลับมาเล่นที่ speed อื่นได้ ทำให้เราสามารถเร่งหรือลดความเร็วของเสียงได้ตามต้องการ ถ้าเราต้องการอัดเสียงสูง ๆ ให้ได้ผลดีควรจะใช้ speed สูง ๆ ในการอัดด้วยจึงจะทำให้เสียงที่อัดออกมาดี

T6668 สามารถเลือกใช้ Memory ได้ 2 ขนาด คือ 64K และ 256K โดยการเลือกที่ขา 45 ของไอซี และสามารถเลือกจำนวน Memory ได้ตั้งแต่ 1 ถึง 4 ตัว โดยการเลือกที่ขา 7 และขา 8 (M_1 และ M_2) ตามตารางที่ 1 สำหรับในโครงการนี้ได้เลือกแรมชนิด 256 K จำนวน 4 ตัว จึงได้ต่อขา 45 , 7 และ 8 เข้ากับ Vdd

2.2.7 การทดสอบวงจรใช้งาน

ในการทดสอบวงจรที่ออกแบบมานั้น จะใช้สวิทช์สั่งงานแทน การสั่งงานจาก CPU ลักษณะการทำงานจะเป็นดังนี้

1. เมื่อเริ่มเปิดเครื่อง LED₁ จะติด
2. กดสวิทช์ไปที่ตำแหน่ง บันทึกลับ ค้างไว้
3. เลือกช่องที่จะบันทึกโดยใช้สวิทช์ 4 ตัว ซึ่งจะไปกำหนดค่า D₀-D₃ ในลักษณะ Binary code ดังตารางที่ 3
4. เลือกสวิทช์เพื่อตั้ง speed ตามตารางที่ 2
5. กดปุ่มสวิทช์ start แล้วไฟที่ LED จะดับ แสดงว่าเครื่องกำลังอัดค่าพูดเข้าไปเก็บ เมื่อพูดจบแล้วจึงกดสวิทช์ stop ไฟที่ LED จะสว่าง ในกรณีที่เรานานเกินกว่าเวลาที่กำหนด เมื่อครบกำหนดเวลา เครื่องจะหยุดการบันทึกโดยอัตโนมัติ ไฟที่ LED จะสว่างขึ้นมาเพื่อบอกให้เรา รู้ว่าเป็นการสิ้นสุดการบันทึกใน 1 ช่อง

6. ถ้าเราต้องการบันทึกในช่องอื่น ๆ อีก ก็ทำเช่นเดียวกัน แต่รวมเวลาของแต่ละช่องต้องไม่เกินเวลาที่กำหนดไว้
7. การอ่านทำโดยการยกเลิกสวิตช์ WR
8. เลือกช่องที่จะอ่านและ speed
9. กดสวิตช์ start เครื่องจะหยุดตามที่อัดไว้ถ้าเรากดสวิตช์ซ้ำกันหลายครั้งในระหว่างหยุดเครื่องจะจำได้ว่ามีการกดสวิตช์ start ขึ้นเพียงครั้งเดียว และจะหยุดซ้ำอีกเมื่อหยุดจบ

ตารางที่ 3 การเลือกช่องสำหรับการบันทึก

D_0	D_1	D_2	D_3	ช่องที่
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เมื่อต้องการให้เครื่องพูดติดต่อกัน ทำโดยการเลือกช่องแรก กด start เสร็จแล้วกดสวิทช์ซ้ำอีกทีหนึ่ง เครื่องจะพูดซ้ำอีกตามต้องการได้ จากลักษณะการทำงานที่กล่าวมานี้ จึงทำให้สามารถกำหนดให้เครื่องตัดต่อคำพูดได้ พูดซ้ำได้ เร่งหรือลด speed คำพูดได้ ซึ่งการควบคุมการทำงานดังกล่าวนี้สามารถกำหนดได้จาก CPU โดยตรง

2.2.8 ปัญหาและวิธีการแก้ไข

ปัญหาหนึ่งที่เกิดกับชุดวิเคราะห์เสียงพูดชุดนี้ คือ เกิดเสียงหวีดขึ้นที่ลำโพง ทำให้เสียงที่บันทึกไว้ถูกรบกวน สาเหตุของเสียงหวีดเกิดจากการป้อนกลับของสัญญาณ ระหว่างไมค์กับลำโพง วิธีแก้ไขก็โดยต่อสวิทช์สำหรับตัดลำโพงออกขณะที่ทำการบันทึก

2.3 Tape Recorder

ชุด Tape Recorder ประกอบด้วย 2 ส่วนคือ

1. ชุดกลไกเทปและส่วนควบคุมกลไกเทป
2. ชุด Pre-Amp. เล่น/บันทึก

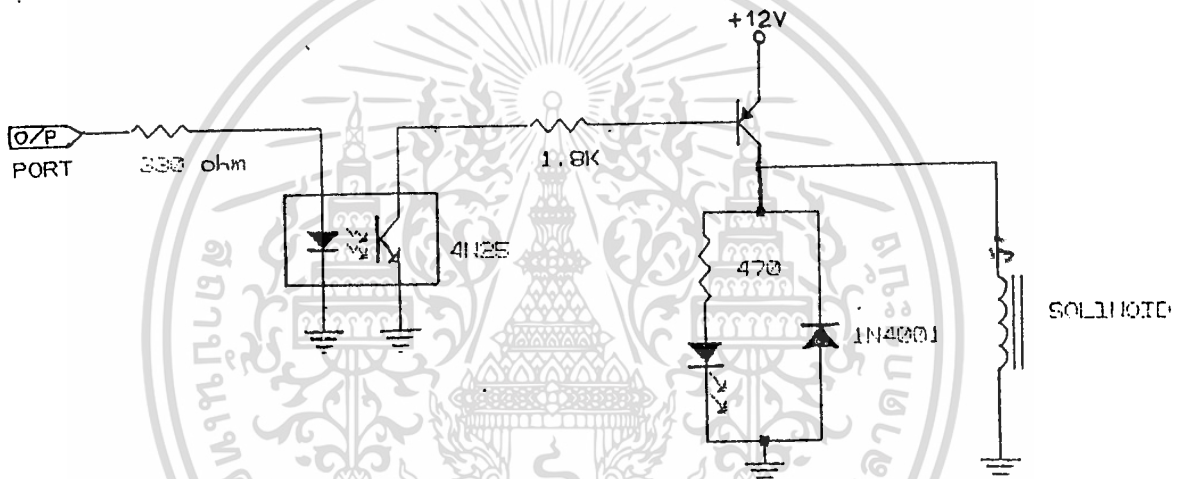
2.3.1 ชุดควบคุมกลไกเทป

โครงการนี้ได้เลือกใช้ชุดกลไกเทปที่มีการควบคุม โดยใช้โซลินอยด์ทั้งหมด ซึ่งมีขายในท้องตลาด ประกอบด้วยชุดโซลินอยด์ 4 ชุด คือ

1. ชุดโซลินอยด์สำหรับการรอกเทป (SPD)
2. ชุดโซลินอยด์สำหรับควบคุมกลไกเทปให้รอกเดินหน้า หรือถอยหลัง (F/R)
3. ชุดโซลินอยด์สำหรับควบคุม การเคลื่อนที่ของหัวเทปเพื่อให้หัวเทปสัมผัสกับเนื้อเทป (SND)

4. ชุดโซลินอยด์สำหรับควบคุมการเปิด ปิดของมอเตอร์ (MTR)

ชุด Magnetic tape ใช้หัวเทปแบบ stereo เล่นแบบทางเดียว ส่วนสัญญาณที่จะมาควบคุมการทำงานของชุดโซลินอยด์ทั้งหมด ถูกส่งมาจาก Port output ของชุด CPU โดยผ่านชุด opto isolator และชุด driver ตามลำดับ ซึ่งจะได้กล่าวต่อไป



รูปที่ 17 ชุดควบคุมกลไกของเทป

2.3.2 ชุด Pre Amplifier เล่น/บันทึก

หลักการเบื้องต้นของชุด Pre Amplifier แสดงดังรูปที่ 18 ซึ่งมีส่วนสำคัญคือ วงจรควบคุมการเล่น/บันทึก หากวงจรทำงานในการเล่นกลับก็จะนำสัญญาณที่ได้จากหัวเทป มาเข้ายังวงจรปริแอมป์หัวเทปภาคแรก และทำการขยายอีกครั้งเพื่อให้ได้สัญญาณแรงขึ้นที่ภาคขยายเอาท์พุท ก่อนส่งออกไปทางเอาท์พุท เพื่อไปเข้ายังภาคโทนคอนโทรล และเครื่อง

ทำให้สารแม่เหล็กที่ถูกฉาบอยู่บนเส้นเทป เปลี่ยนสภาพเป็นแม่เหล็กตามสัญญาณที่เหนี่ยวนำ อย่างไรก็ตาม ในการบันทึกเสียงย่อมจะต้องมีปัญหาเรื่องความเพี้ยนเกิดขึ้น ความเพี้ยนนี้เรียกว่า Hysteresis ซึ่งสามารถแก้ไขได้โดยการป้อนกระแสไฟฟ้าค่าต่ำ ๆ ให้กับหัวเทปในขณะที่มีการบันทึก พร้อม ๆ กับสัญญาณเสียงที่ป้อนเข้าไปทางอินพุท เรียกว่าวิธีการนี้ว่าการไบอัส โดยปกติแล้วมักจะใช้การไบอัสแบบกระแสสลับ โดยใช้ความถี่จากวงจรออสซิลเลเตอร์ที่มีค่าสูงกว่าความถี่เสียงประมาณ 2 ถึง 3 เท่า มีระดับสัญญาณประมาณ 10 มิลลิโวลต์ ขึ้นไป

การเล่นกลับ (Play back)

มีวิธีการที่ตรงข้ามกับการบันทึก อานาแม่เหล็กของสารแม่เหล็กที่ฉาบบนผิวของเส้นเทป เมื่อมันผ่านหัวเทป เส้นแรงแม่เหล็กที่เส้นเทปนั้นจะตัดกับขดลวดภายในหัวเทป ทำให้เกิดการเหนี่ยวนำเป็นกระแสไฟฟ้าที่มีขนาดต่ำ ๆ ขึ้นมา จากนั้นนำสัญญาณนี้ไปขยายต่อไป

วงจรควบคุมสัญญาณอัตโนมัติ (ALC)

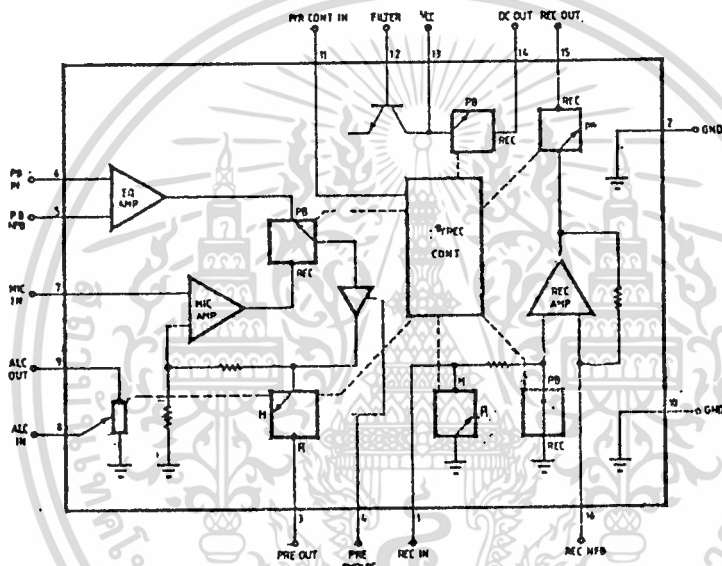
ในการบันทึกเสียงของวงจรปริเทปชุดนี้ จะเห็นว่าไม่มีตัวต้านทานปรับค่า เพื่อเพิ่ม-ลด ระดับสัญญาณอินพุทที่ป้อนเข้ามาเพื่อทำการบันทึก เพราะในวงจรมีส่วนที่เรียกว่าวงจรควบคุมระดับสัญญาณอัตโนมัติ (Automatic Level Control-ALC) ถ้าอินพุทแรงเกินไปก็จะลดระดับเองอัตโนมัติ ถ้าอินพุทมาเบา ก็จะทำการเพิ่มให้พอดี

2.3.3 BA5101 ไอซีปริเทป เล่น/บันทึก

จากรูปที่ 19 เป็นโครงสร้างภายในของไอซีเบอร์ BA5101 เป็นของบริษัทอาร์ไอเอ็ม ไอซีเบอร์นี้มี 16 ขา ภายในตัวประกอบด้วย วงจรขยายอ็ควอลไลเซอร์ (playback equalizer amplifier)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรถ่ายสัญญาณบันทึก(recording amplifier) วงจรถ่ายสัญญาณไมโครโฟน (microphone amplifier) วงจรควบคุมความแรงของสัญญาณอินพุตอัตโนมัติ (ALC) วงจรออสซิลเลเตอร์เพื่อใช้ในการไบอัสหัวเทป และวงจรถับสัญญาณเล่นกลับและบันทึก ข้อดีของไอซีตัวนี้คือ นอกจากรวมวงจรต่าง ๆ ไว้ในตัวแล้ว ในการควบคุมการเล่นกลับและบันทึกใช้การควบคุมแบบดิจิทัล คือถ้าขาควบคุม(ขา 11) เป็น "0" ไอซีจะทำหน้าที่เป็นวงจรถ่ายสัญญาณเล่นกลับ แต่ถ้าที่ขาควบคุมเป็น "1" ก็จะทำงานในการบันทึกสัญญาณ แรงดันไฟเลี้ยงของไอซีเบอร์นี้ใช้ประมาณ 6-9 โวลต์



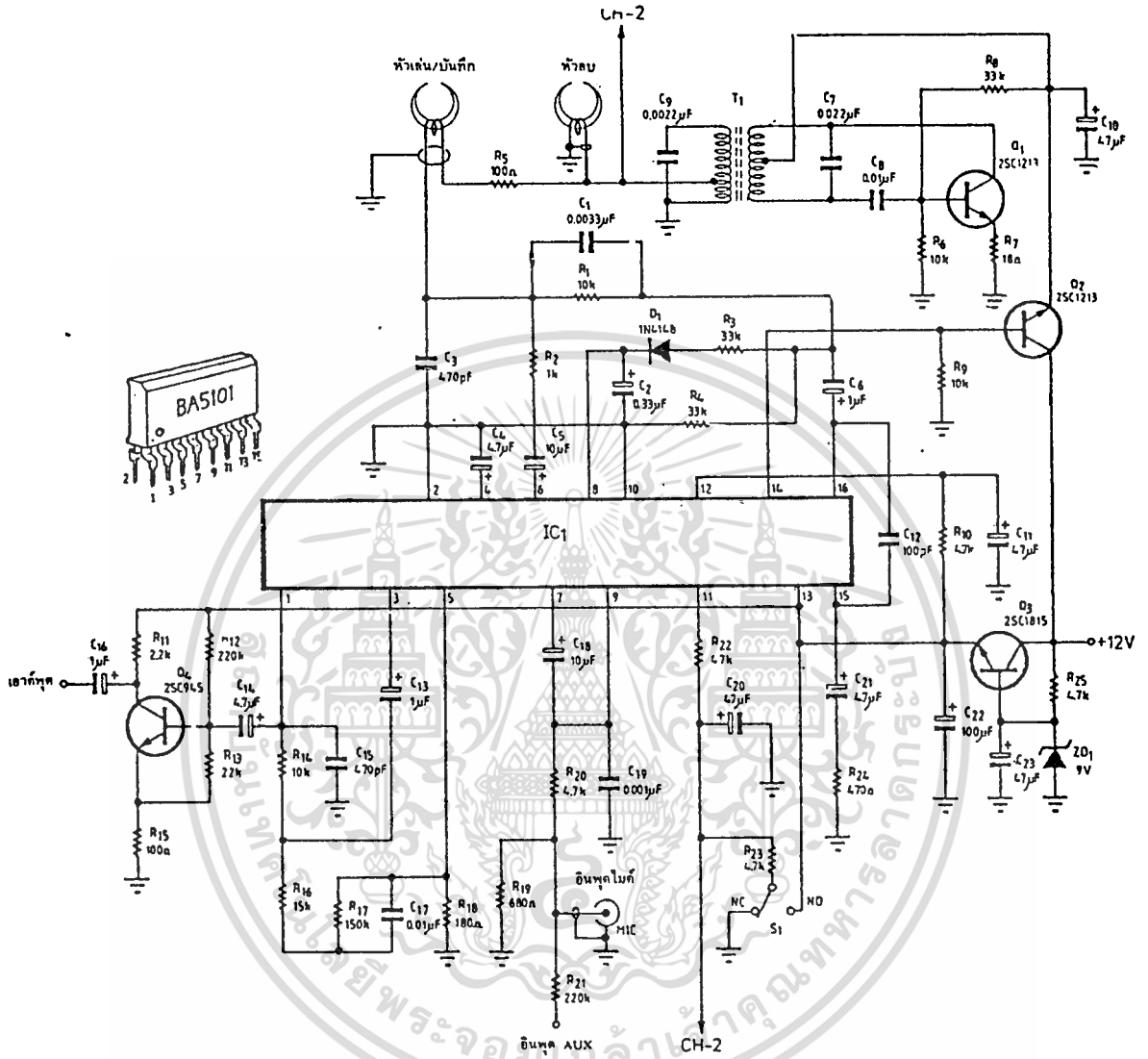
รูปที่ 19 โครงสร้างไอซี BA5101

2.3.4 การทำงานของวงจร

วงจรถ่ายสัญญาณ ดังแสดงในรูปที่ 20 เมื่อต้องการให้วงจรถ่ายงานในการเล่นกลับ สวิตช์ SW₁ จะถูกต่อลงกราวด์ ไอซีจะรับรู้ว่าจะขณะนี้ต้องการเล่นกลับ สัญญาณจากหัวเทปจะผ่านตัวต้านทาน R₂ คัปปลิ่ง เข้าไอซี โดยตัวเก็บประจุ C₅ ที่ขา 6 ซึ่งเป็นอินพุตของไอซี สัญญาณที่ผ่านการขยายแล้วจะถูกส่งออกมาที่ขา 3 ผ่านตัวเก็บประจุ C₁₀ เข้ามายังวงจรถ่ายสัญญาณเอาต์พุต ซึ่งประกอบด้วย R₁₁, R₁₂, R₁₃, R₁₅,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Q_4 และ C_{10} จะได้สัญญาณที่มีความแรงมากพอ ออกไปทางเอาต์พุตของวงจร เพื่อไปยังโทนคอนโทรลต่อไป



รูปที่ 20 วงจรปรับเทียบเล่น/บันทึก

ที่ขา 5 ของไอซี ตัวต้านทาน R_{17} และ R_{10} ต่อร่วมกับตัวเก็บประจุ C_{17} เป็นวงจรป้อนกลับอ็ควอลไลเซอร์ เพื่อให้ไอซีมีการตอบสนองความถี่ได้ดีขึ้น ตัวเก็บประจุ C_4 ที่ต่อไว้ที่ขา 4 ของไอซีทำหน้าที่ขยายพาสค่าของตัวเก็บประจุหากใช้ค่าที่ไม่ถูกต้องอาจทำให้การตอบสนองความถี่ต่ำของวงจรเสียไปได้

เมื่อทำการบันทึกสวิตซ์ SW_1 จะถูกต่อไปที่ไฟเลี้ยงของไอซี สัญญาณอินพุทจะผ่านตัวต้านทาน R_{20} และ R_{21} เพื่อลดความ

แรงลงระดับหนึ่งก่อน แล้วส่งเข้าไปยังขา 7 ซึ่งเป็นขาอินพุทของวงจรรขยายสัญญาณไมโครโฟน เอาท์พุทของสัญญาณนี้จะออกมาที่ขา 16 และผ่านตัวต้านทาน R_9 ใดโอด D_1 ซึ่งทำหน้าที่เปลี่ยนสัญญาณกระแสสลับที่ได้จากวงจรรขยายสัญญาณบันทึกในตัวไอซีมาเป็นไฟตรง เพื่อป้อนเข้ายังขา 8 ของไอซี ซึ่งเป็นขาอินพุท ALC เพื่อทำการควบคุมความแรงของสัญญาณอินพุทอย่างอัตโนมัติ

เอาท์พุทของวงจรร ALC จะออกมาที่ขา 9 และทำการป้อนกลับสัญญาณเข้าที่ขา 7 เป็นเช่นนี้เรื่อยไปเพื่อทำการควบคุมความแรงของสัญญาณอินพุท เอาท์พุทที่ขา 16 ของไอซี จะผ่านตัวต้านทาน R_1 และ C_1 ป้อนเข้าที่ขา 6 เพื่อทำการขยายสัญญาณให้พอเหมาะ แล้วป้อนเข้าที่ขา 1 ซึ่งเป็นขาอินพุทของวงจรรบันทึก สัญญาณที่ขา 16 อีกส่วนหนึ่งก็จะถูกส่งไปที่หัวเทป เพื่อทำการบันทึกสัญญาณลงบนเส้นเทป ในขณะที่เดียวกันที่ขา 14 จะเกิดสัญญาณเพื่อไปทำให้ทรานซิสเตอร์ Q_2 ทำงาน เกิดแรงดันไปไบอัสให้ทรานซิสเตอร์ Q_1 ทำงาน

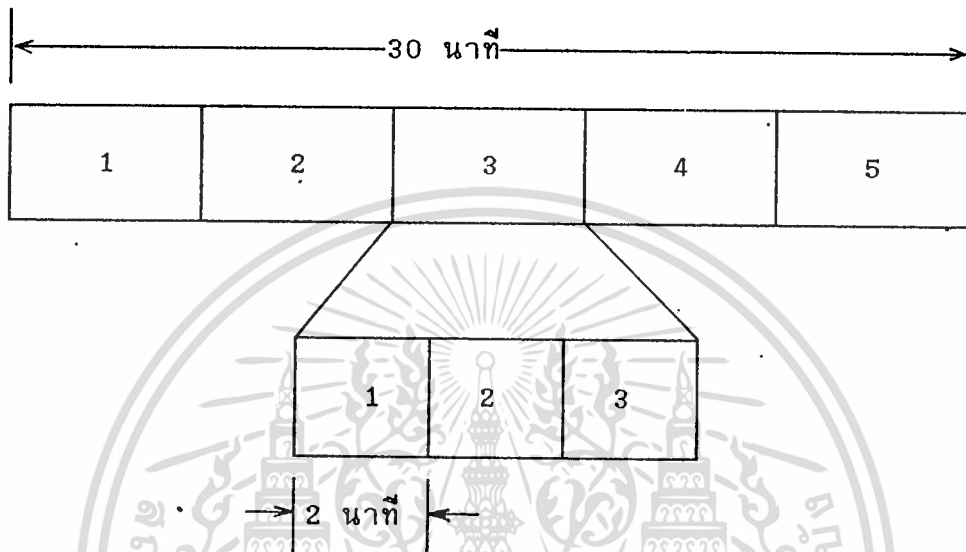
ทรานซิสเตอร์ Q_1 ต่อร่วมกับหม้อแปลง T_1 ตัวต้านทาน R_6, R_7 และ R_8 และตัวเก็บประจุ C_7, C_8 ทำหน้าที่เป็นวงจรรออสซิลเลเตอร์ความถี่สูง เพื่อส่งสัญญาณไปไบอัสที่หัวเทป และยังใช้ส่วนนี้ในการลบเทปด้วยความถี่ประมาณ 45 kHz ทรานซิสเตอร์ Q_3 และ zener-diode ZD_1 ทำหน้าที่เป็นวงจรรเรกูเลเตอร์ 9 V เพื่อจ่ายเป็นไฟเลี้ยงให้เข้าไอซี โดยมีตัวเก็บประจุ C_{21} กรองไฟให้เรียบอีกชั้นหนึ่ง

2.3.5 การบันทึกและการค้นหาข้อมูลในเทปบันทึกเสียง

การบันทึก

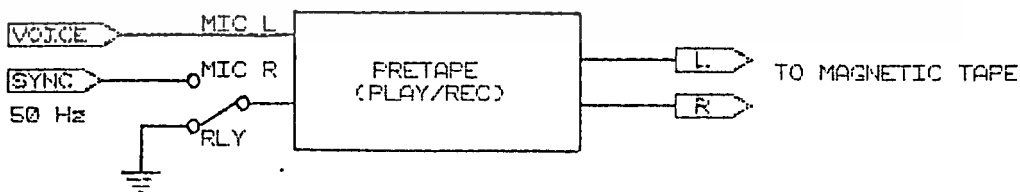
ใช้การบันทึกลงบนเทปบันทึกเสียงหน้าเดียว ความยาว 30 นาที แบ่งเนื้อที่การบันทึกออกเป็น 15 บล็อก ๆ ละ 2 นาที เมื่อมีผู้ที่โทรเข้ามาคนแรก เครื่องจะทำการบันทึกที่ตำแหน่งบล็อกที่ 1 ทันที แต่ CPU จะเป็นตัวจำไว้ว่าเป็นข้อความของใครเมื่อบันทึกเสร็จก็จะไปรออยู่ที่บล็อกที่ 2 ทันที

เมื่อมีผู้ที่โทรเข้ามาติดต่ออีก ก็สามารถบันทึกต่ออีกได้ทันที โดยไม่ต้องรอ แต่สำหรับผู้โทรเข้ามาขอฟังข้อความ ซึ่งเป็นเจ้าของบล็อก ก็จะต้องเสียเวลารอบ้างสำหรับการค้นหาข้อความ ซึ่งจะให้ความสะดวกต่อผู้ที่โทรเข้ามาบันทึก



รูปที่ 21 การแบ่งบล็อกของเทปบันทึกเสียง

การบันทึก จะบันทึกเป็น 2 channel (stereo) โดยใช้ channel ซ้าย เป็นช่องสำหรับบันทึก Voice ส่วน channel ขวา ใช้สำหรับบันทึก Sync.



รูปที่ 22 Block diagram การบันทึก Voice และ Sync.

ม้วนเทปบันทึกเสียงก่อนที่จะใช้งาน จะต้องทำการ Format ก่อน โดยการกรอเทปกลับไปหาตำแหน่งเริ่มต้น และทำการบันทึก Sync. ในช่อง R โดยการควบคุมจาก CPU ให้ตัดต่อ Relay เพื่อสร้างตำแหน่ง Blank "0" ในเนื้อเทป โดยกำหนดให้ Blank 2 นาที และใส่ Sync. 1 วินาที เมื่อทำการบันทึก Sync. จนหมดม้วนแล้ว CPU ก็จะสั่งให้เทปกรอกลับ มาอยู่ในตำแหน่งเริ่มต้น เพื่อรอการบันทึกต่อไป

การค้นหาล็อกข้อความ

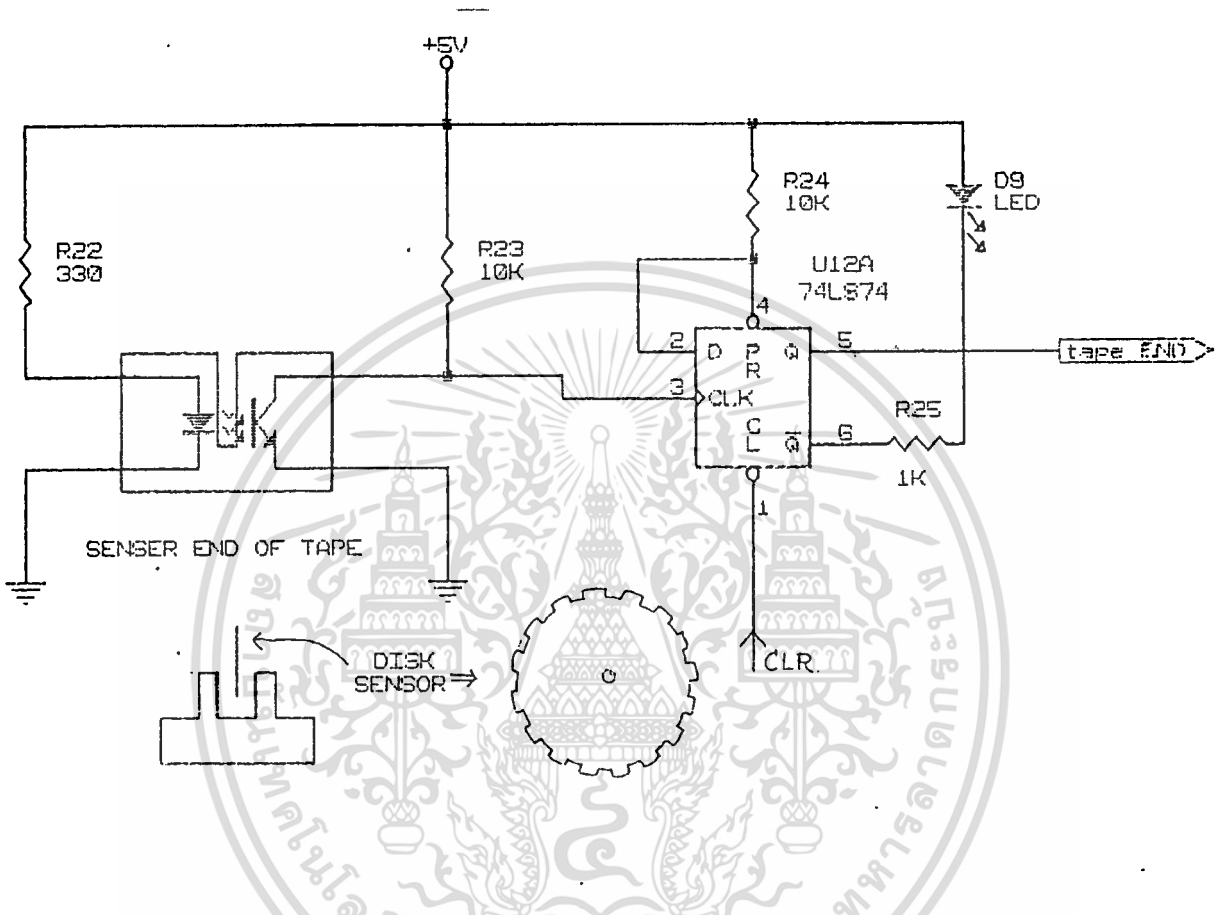
ในการค้นหาล็อกข้อความแต่ละบล็อกนั้น ใช้หลักการ APSS (Automatic Program Search System) ซึ่งจะได้กล่าวต่อไป โดยการ Detect สัญญาณ Blank "0" ใน Channel R ของเนื้อเทป เพื่อส่งไปบอก CPU ให้ทราบถึงตำแหน่งของแต่ละบล็อกได้ เพื่อเป็นตำแหน่งที่ใช้ในการค้นหาต่อไป

APSS เป็นระบบที่ใช้สำหรับการค้นหาตำแหน่ง Blank ในเนื้อเทป โดยการหาตำแหน่งที่ไม่มีการบันทึก (เงียบ) และสามารถสั่งให้ค้นหาที่ตำแหน่งใดก็ได้โดยการใส่โปรแกรม ในการค้นหาโดยการกรอพร้อมกับใช้หัวเทปอ่านไปด้วย ถ้ามีสัญญาณก็ให้เป็น "1" ถ้าไม่มีสัญญาณก็ให้เป็น "0"

2.3.6 วงจรการตรวจสอบการหยุดหมุนของเทป

การ sensor การหยุดหมุนของเทปนั้น ใช้ Opto Couple ในการตรวจจับ โดยใช้แผ่น Disk กลมตัดขอบในรูปลักษณะเป็นเฟืองเพื่อ Detect แสงที่ตกกระทบ Opto couple แล้วยึดติดกับชุดขับเทป (แทนที่ชุด counter ของเทปที่มีอยู่แล้ว) เมื่อเทปหมุนเฟืองนี้ก็จะหมุนตาม ฟันเฟืองก็จะตัดกับแสงของ Opto couple เกิดเป็น Pulse สัญญาณนาฬิกา มาให้ D flipflop (74LS74) ได้เอาที่พุก Q เป็น "1" ที่ขา 5 ไปให้ CPU โดยผ่าน Port input เมื่อ CPU รับรู้แล้วก็จะส่ง Pulse มาเคลียร์ 74LS74 เพื่อรอรับ Pulse จากชุด Opto couple ลูกลงต่อไป แต่ถ้าหากไม่มี Pulse

เกิดขึ้นอีกหลังจากรออยู่ระยะเวลาหนึ่งแล้ว ก็แสดงว่าเทปหยุดหมุนแล้ว CPU ก็จะสามารถได้ ซึ่งเราจะนำสถานะเหล่านี้ไปใช้ในการตรวจสอบว่า การกรอเทปได้หมดม้วนแล้วหรือยัง

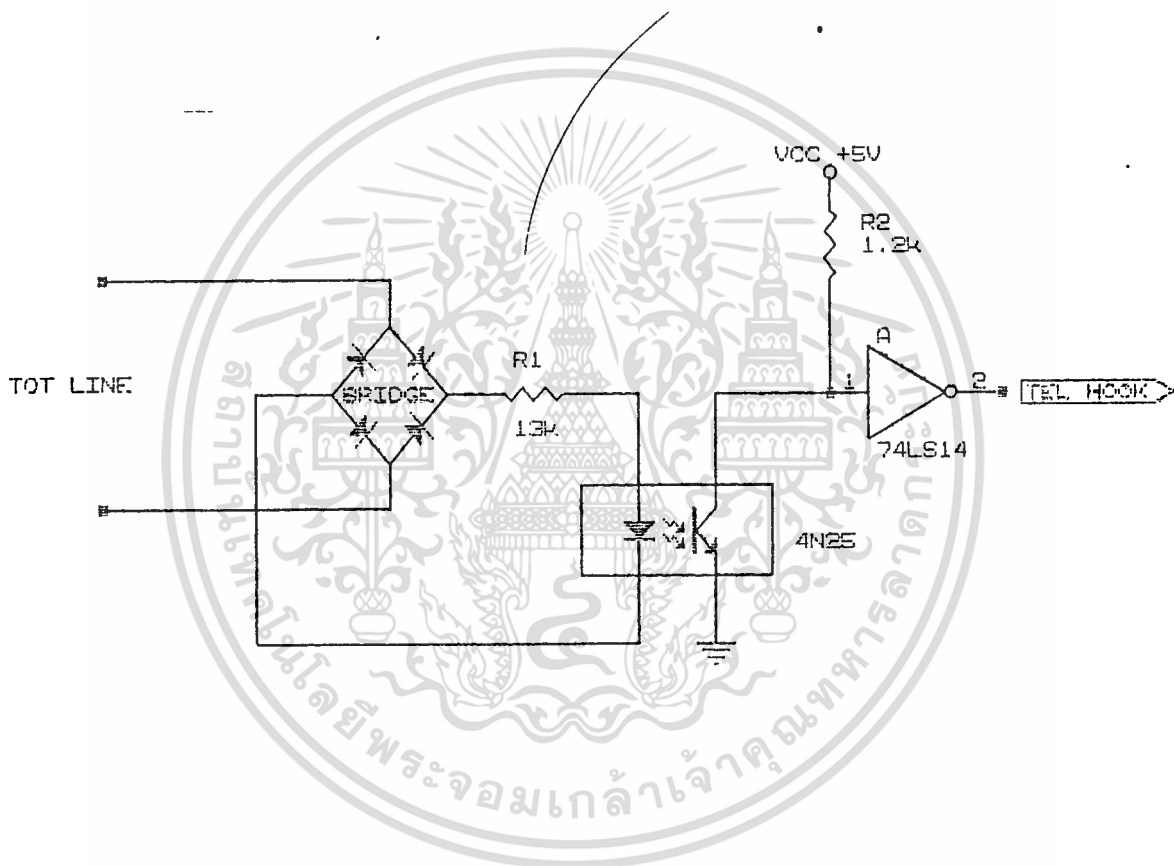


รูปที่ 23 วงจรตรวจสอบการหยุดหมุนของเทป

2.4 วงจรตรวจสอบการยกหู (Hook Detection)

คู่สายโทรศัพท์จะถูกต่อเข้ากับ วงจรตรวจสอบการยกหูโทรศัพท์ โดยอาศัยหลักการตรวจจับการเปลี่ยนแปลงระดับแรงดันในคู่สายโทรศัพท์ ซึ่งในสภาวะปกติ เมื่อไม่มีการยกหูจะมีค่าแรงดัน DC อยู่ประมาณ 48 โวลต์ ซึ่งแรงดันนี้จะทำให้ LED ภายในตัว Opto Isolator ทำงานได้ โดยมี R_1 เป็นตัวจำกัดกระแส เมื่อ LED ทำงานเป็นผลให้ทรานซิสเตอร์ภายในทำงานไปด้วย ทำให้แรงดันเอาต์พุตตกลงเป็น "0" และผ่าน NOT Gate เพื่อ

Interface เข้ากับ input port ของ CPU และเมื่อมีการยกหูโทรศัพท์ แรงดันภายในคู่สายโทรศัพท์ จะตกลงเหลือประมาณ 8 โวลต์ ซึ่งไม่เพียงพอที่จะทำให้ LED ภายใน 4N25 ทำงานได้ เนื่องจากมี R₁ เป็นตัวจำกัดกระแสอยู่ เมื่อเป็นเช่นนี้จะทำให้ทรานซิสเตอร์หยุดนำกระแส เอาท์พุทจึงเป็น "1" และผ่าน NOT Gate เพื่อเป็นสัญญาณ TEL HOOK ส่งให้ CPU โดยผ่าน Input Port CPU ก็สามารถตรวจสอบได้ว่าการยกหูรับโทรศัพท์หรือไม่ เพื่อทำการ Process ต่อไป



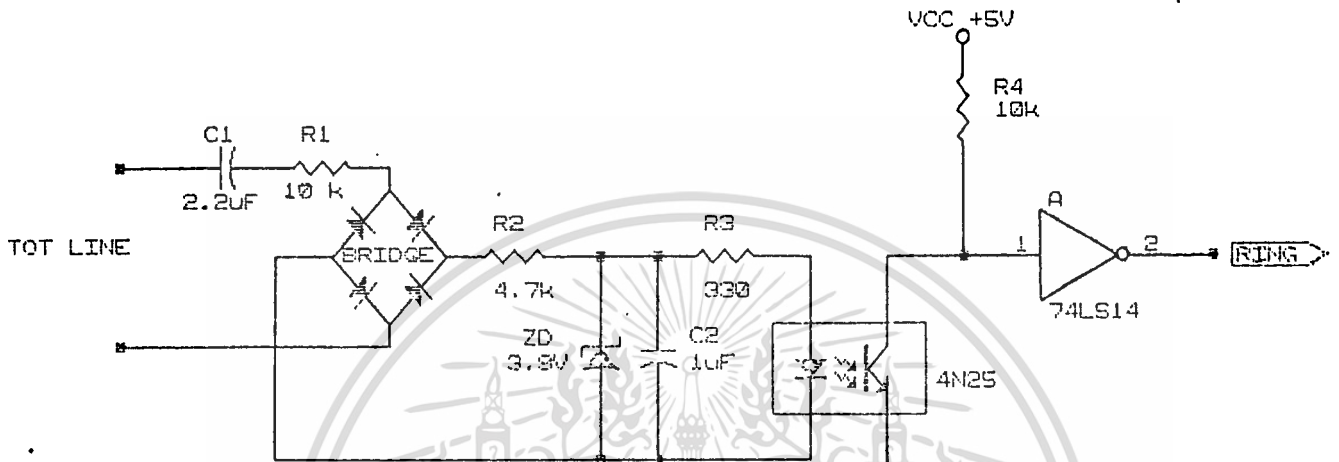
รูปที่ 24 วงจรตรวจสอบการยกหูโทรศัพท์

2.5 วงจรตรวจสอบสัญญาณกระดิ่ง (Ringing Detection)

วงจรนี้จะทำการตรวจจับสัญญาณ Ringing ซึ่งจะต่ออยู่กับคู่สายโทรศัพท์ จากวงจร C₁ จะทำหน้าที่กั้นแรงดันไฟ DC ที่มีอยู่ในคู่สายโทรศัพท์ จากนั้นจะเข้าสู่ Rectifier ทำหน้าที่เปลี่ยนสัญญาณ Ringing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งมีแรงดัน AC ประมาณ 100 โวลต์ ให้เป็นไฟ DC เมื่อมีสัญญาณ Ringing เข้ามาจะทำให้มีกระแสไหลผ่าน LED ประมาณ 10 mA โดยมี Zener ช่วยลดแรงดันที่ตกคร่อม LED



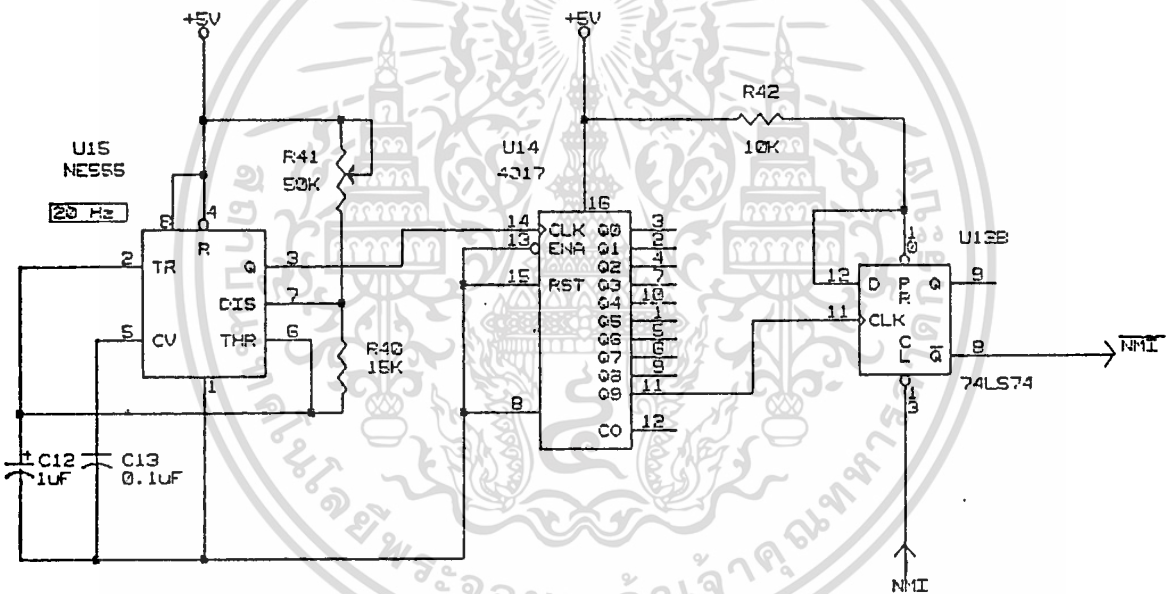
รูปที่ 25 วงจรตรวจสอบสัญญาณกระดิ่ง

เมื่อมีกระแสไหลผ่าน LED จะทำให้ทรานซิสเตอร์ทำงาน เป็นผลให้ได้ output "0" แล้วผ่าน NOT Gate จะได้เป็น Logic "1" เป็นสัญญาณ Ring ส่งไปให้ Input Port ของชุด CPU เพื่อให้ CPU ทำการตรวจสอบและนับจำนวน Ringing เพื่อทำการประมวลผลต่อไป

2.6 วงจรกำเนิดสัญญาณ MNI

เวลาต่าง ๆ ที่ใช้ในระบบ เช่นการจับเวลาการหายไปของเสียงพูด เพื่อตรวจสอบการสิ้นสุดการบันทึก เวลาต่าง ๆ เหล่านี้ CPU จะอาศัยการนับจำนวนการถูก Interrupt แบบ Nonmaskable โดยจะทำการนับทุกครั้งที่มีสัญญาณ NMI เข้ามา โดยค่าที่นับแต่ละครั้ง จะเท่ากับ 0.5 วินาที ซึ่งก็คือสัญญาณนาฬิกา 2 Hz นั้นเอง โดยใช้ไอซี NE555 ทำหน้าที่เป็นวงจร

Astable Multivibrator กำเนิดสัญญาณนาฬิกา 20 Hz ได้เอากัพท์ไปหาร 10 ด้วยไอซี #4017 หลังจากหารแล้วก็จะได้สัญญาณนาฬิกา 2 Hz ไปเป็นสัญญาณนาฬิกาให้กับ D FlipFlop ที่ขา CLK ส่วนขา D ต่ออยู่กับ "1" ดังนั้นทุกครั้งที่มีการสัญญาณนาฬิกา ที่ขา Q ก็จะเปลี่ยนจาก "1" เป็น "0" ทำให้ได้สัญญาณ NMI ไปยัง CPU เมื่อ CPU ต้องการการจับเวลา ก็จะส่งสัญญาณ "1" มาให้ Flip Flop ที่ขา CLR ซึ่งจะทำให้ Flip Flop มีสัญญาณ Pulse ออกมาเป็นสัญญาณ NMI ให้ CPU แต่ถ้า CPU ไม่ต้องการจับเวลา CPU ก็จะส่ง "0" มาเคลียร์ Flip Flop ทำให้ที่ขา Q เป็น "1" อยู่ตลอด จึงไม่มีการ Interrupt เกิดขึ้น



รูปที่ 26 วงจรกำเนิดสัญญาณ NMI

2.7 DOT MATRIX LCD MODULE

อุปกรณ์ในปัจจุบันนี้ในส่วนแสดงผลนั้นจะให้ LCD เสียเป็นส่วนใหญ่ว่าจะเป็นเครื่องเล่น VIDEO, เครื่องถ่ายภาพเอกสาร, เครื่องมือวัดคุมต่างๆ, เครื่องคอมพิวเตอร์ เราพอจะแบ่ง DOT MATRIX LCD MODULE นี้ออกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นพวก ๆ ดังนี้

1. CHARACTER LCD MODULE
2. GRAPHIC LCD MODULE
3. SEGMENT DISPLAY TYPE LCD MODULE

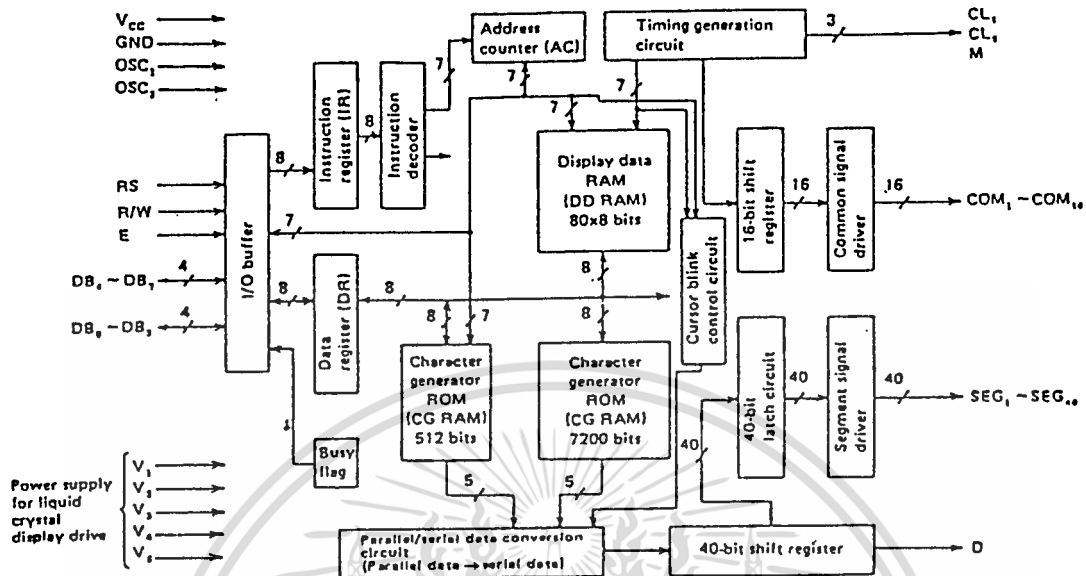
โดยในแต่ละแบบนี้ก็จะมีส่วนประกอบใหญ่ๆแบ่งได้เป็น

1. DOT MATRIX LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการปิดและเปิดตัวเองกับแสงก็คือ ส่วนของที่เป็นตัวกระจกบรจุผลึก
2. DRIVER เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD อีกทีหนึ่งโดยมีเบอร์ที่นิยมใช้ใน LCD MODULE เช่น HD 44780H, MSM5259
3. CONTROLLER เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุม LCD MODULE ให้ทำงานแสดงผลต่างๆเช่น การลบจอภาพ, การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ IC ที่นิยมใช้กันคือ HDM20216M ซึ่งจะใช้ในแบบ CHARACTER LCD MODULE เป็นส่วนใหญ่ เบอร์ HD61830 จะใช้ในแบบ GRAPHIC LCD MODULE

ในการศึกษาการทำงานและใช้งาน LCD MODULE นั้นไม่ใช่เรื่องยากเลยถ้าเราสามารถทำความเข้าใจในส่วนของ CONTROLLER ได้ก็เพียงพอแล้วและโดยมาก LCD MODULE ในแต่ละบริษัทแล้วจะใช้ตัว CONTROLLER ที่มีหลักการทำงานเหมือนกัน เป็นส่วนใหญ่และใน LCD MODULE แต่ละขนาดจำนวนตัวอักษร หรือจำนวนบรรทัดก็มีหลักการทำงานแบบเดียวกันซึ่งในโครงการนี้ได้เลือกใช้ เบอร์ HDM 20216H การแสดงผลเป็นแบบ 20 ตัวอักษร จำนวน 2 บรรทัด

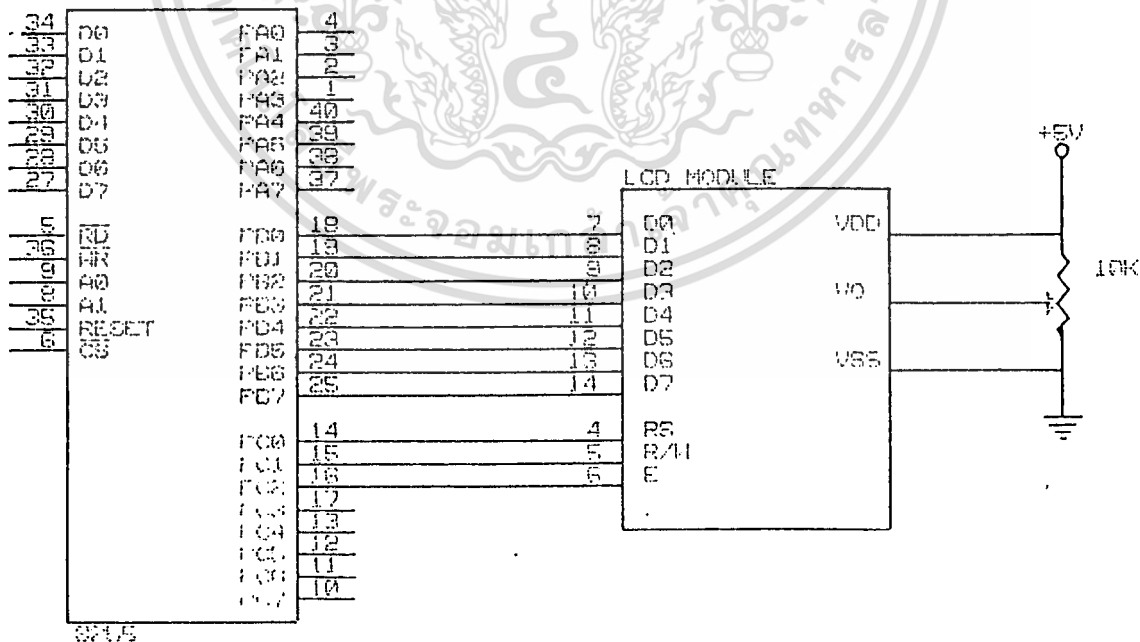
HDM 20216H เป็นไอซี LSI ควบคุมการแสดงผลของ LCD ในรูปตัวอักษรหรือสัญลักษณ์ต่างๆ ตัวมันเองสามารถต่อใช้งานแบบ 4 BIT หรือ 8 BIT ก็ได้ โดยถ้าเราต่อแบบ 4 BIT จะต่อใช้งาน DB7-DB4 เท่านั้น โดยข้อมูลครั้งแรกที่ส่งนั้นจะถือเป็นข้อมูล 4 BIT บน และข้อมูลที่ส่งต่อมานั้นเป็นข้อมูล 4 BIT ล่าง

Block diagram of HD44780 interior



รูปที่ 27 Block diagram ของ HD44780

2.7.1 วงจรใช้งาน



รูปที่ 28 วงจรการต่อใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

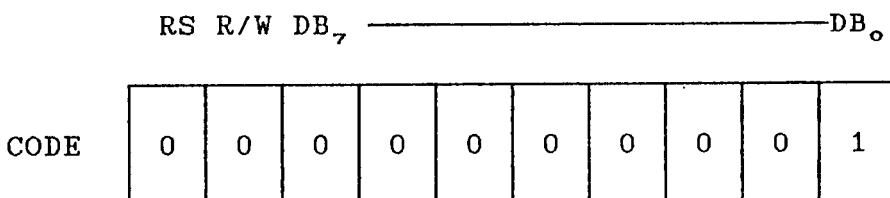
จากวงจรเป็นการต่อ 8255 ให้เข้าใช้กับ LCD โดยเราจะจำลองสัญญาณต่างๆขึ้นมาโดยการใช้ PORT A และ PORT B โดย PORT A นั้นเราให้เป็น DATA PORT และ PORT B นั้นเราให้เป็นสัญญาณควบคุมไปใช้ เมื่อเราเริ่มเปิดไฟป้อนให้ HD44780 นั้นก็จะทำการ RESET ตัวมันเองโดยจะใช้เวลาประมาณ 10 ms หลังจากไฟ VDD ถึง 4.5 VOLT แล้ว โดยจะ SET ตัวเองดังนี้ :-

1. DISPLAY CLEAR จะทำการลบข้อมูลจอภาพ LCD
2. FUNCTION SET โดยจะ SET ค่าภายใน
 - DL = 1 : เป็นการ SET ให้การติดต่อแบบ 8 BIT
 - N = 0 : SET เป็น 1 บรรทัดการแสดงผล
 - F = 0 : 5X7 DOT ต่อหนึ่งตัวอักษร
3. DISPLAY ON/OFF D = 0 : DISPLAY OFF
 - C = 0 : CURSOR OFF
 - B = 0 : BLINK OFF
4. ENTRY MODE SET I/D = 1:+ (เพิ่มค่า COUNTER ขึ้น 1)
 - S = 0 : NO SHIFT

เมื่อเราเริ่มเปิดเครื่องทำงานแล้วก็ต้องส่งคำสั่งควบคุมให้มันเริ่มทำงาน

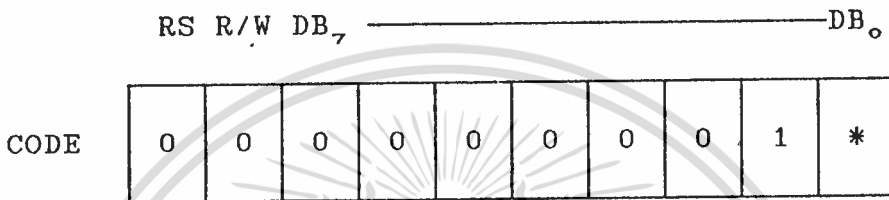
2.7.2 รายละเอียดคำสั่งของ HD 44780

1. CLEAR DISPLAY



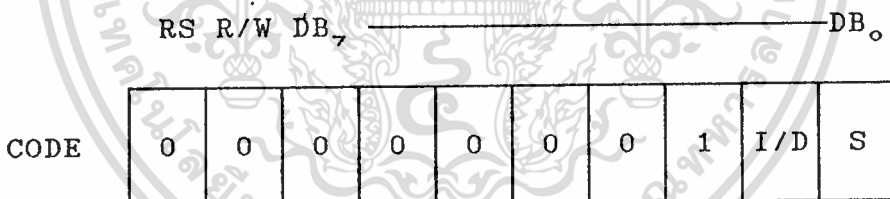
คำสั่งนี้จะเป็นการเขียนช่องว่างหรือ SPACE (ASCII 20H) เข้าไปใน DD RAM ทั้งหมดและทำการ SET DD RAM ADDRESSER เป็นศูนย์ตัว CURSOR จะกลับ ไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ SET I/D = 1,S ไม่มีการเปลี่ยน

2. RETURN HOME



คำสั่งนี้จะทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปยังตำแหน่งบนสุดซ้ายมือของจอภาพข้อมูลในจอภาพไม่เปลี่ยน

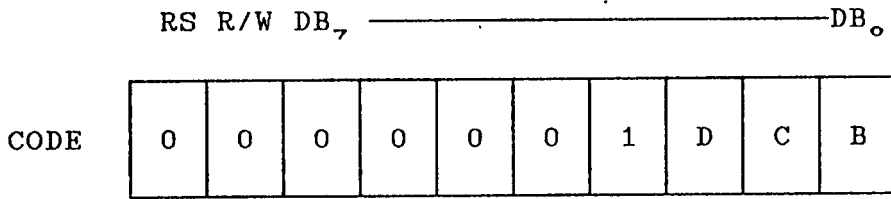
3. ENTRY MODE SET



BIT I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM ADDRESS เพิ่มขึ้นหนึ่งหรือลดลงหนึ่งโดย 1 = เพิ่ม 0 = ลดลงหนึ่ง

BIT S : เป็นตัวกำหนดแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัว CURSOR อยู่ที่ที่ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่ที่ที่ตัว CURSOR จะถูกดันไปทางขวามือ

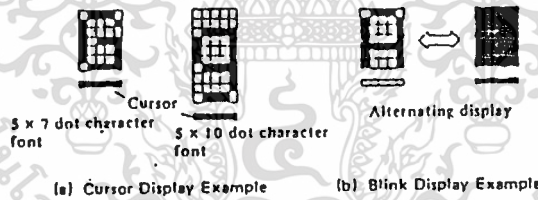
4. DISPLAY ON/OFF CONTROL



BIT D : เป็น BIT ให้อเปิดปิดหน้าจจอภาพโดยถ้า D = 1 จะ ON และ D = 0 จะ OFF

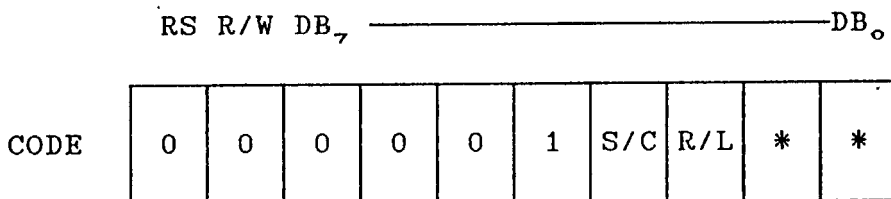
BIT C : จะให้แสดง CURSOR ให้ BIT C = 1 และถ้ายังไม่ต้องการแสดง CURSOR BIT C = 0. โดยตัว CURSOR จะอยู่ที่ LINE ที่ 8 ในแบบ 5X7 และจะอยู่ที่ LINE ที่ 11 ในแบบ 5X10 DOT

BIT B : เป็น BIT SET การกระพริบของ CURSOR โดย B = 1 มีการกระพริบ B=0 ไม่มีการกระพริบ โดยมีระยะเวลาการกระพริบประมาณ 379.2 ms



รูปที่ 29 แสดงตัวอย่างการแสดงผลของ LCD

5. CURSOR หรือ DISPLAY SHIFT



* No effect

เป็นคำสั่งกำหนดให้ตำแหน่ง CURSOR หรือข้อมูลไปเกิดทางซ้าย หรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่าน โดย

ตารางที่ 4 คำสั่งกำหนดตำแหน่ง CURSOR

S/C	R/L	การทำงาน
0	0	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปขวามือ 1 ตำแหน่ง
1	0	เป็นการดันตัวอักษรที่เกิดไปทางซ้ายมือ
1	1	เป็นการดันตัวอักษรที่เกิดไปทางขวามือ

6. FUNCTION SET

RS R/W DB₇ _____ DB₀

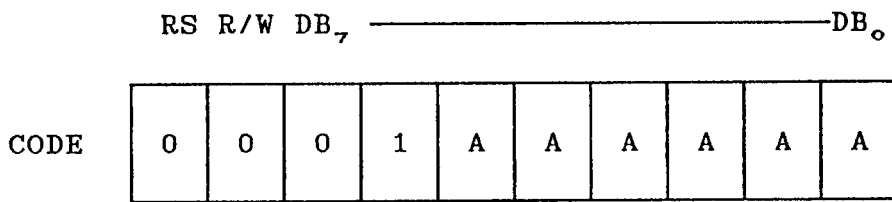
CODE	0	0	0	0	1	DL	N	F	*	*
------	---	---	---	---	---	----	---	---	---	---

BIT DL : เป็นการ SET การติดต่อกว่าจะให้เป็นแบบ 8 BIT หรือ 4 BIT โดยถ้าต้องการติดต่อ 4 BIT DL = 0 และ 8 BIT DL = 1

N : เป็นการ SET บรรทัดการแสดงผล N = 0 แสดง 1 บรรทัด N = 1 แสดงผล 2 บรรทัด ในกรณี มากกว่า 2 บรรทัด ก็ให้ SET N = 1

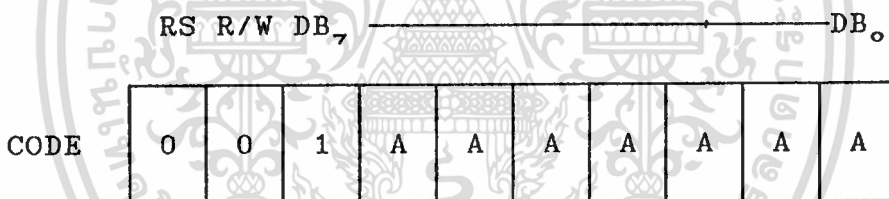
F : เป็นการ SET ขนาด DOT การแสดงผล 5X7 หรือ 5X10 โดย F = 0 เป็นแบบ 5X7 และ F = 1 เป็นแบบ 5X10

7. SET CG RAM ADDRESS



ใน HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือ DISPLAY DATA RAM (DD RAM) จำนวน 80X8 BIT และ CHARACTER GENERATOR RAM(CG RAM) จำนวน 512 BIT และ 7200 BIT คำสั่งนี้จะเป็นการ SET ADDRESS ใน CG RAM โดยต้องทำการ SET ADDRESS ก่อนเขียนหรืออ่านข้อมูลจาก CG RAM ด้วย

8. SET DD RAM ADDRESS



เป็นคำสั่ง SET ค่า ADDRESS ใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM (DD RAM คือ ส่วนที่จะแสดงผลหน้าจอ LCD) โดยจำนวน ADDRESS ที่จะเกิดขึ้นบนจอ LCD จะอยู่กับ SET ค่า N ด้วย

ถ้า N = 0 (1 บรรทัด) ADDRESS จะอยู่ 00H-4FH

ถ้า N = 1 (2 บรรทัด) ADDRESS จะอยู่ 00H-27H สำหรับบรรทัดที่ 1 และ 40H-67H สำหรับบรรทัดที่ 2

ตัวอย่างการจัด ADDRESS ของ DD RAM หน้าจอ LCD แบบ 16 ตัวอักษร 4 บรรทัดและ 20 ตัวอักษร 2 บรรทัด HDM-16416H, HDM-20216H

ตารางที่ 5 การจัดหน้าจอของ HDM-16416H

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	-- display position
1-line	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	-- DD RAM address
2-line	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
3-line	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
4-line	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	

ตารางที่ 6 การจัดหน้าจอของ HDM-20216H

	1hr	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	-- display position
1-line	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	-- DD RAM address
2-line	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	
3-line	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	
4-line	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	

(Note) Shift display is as same as 2-line type.

9. READ BUSY FLAG AND ADDRESS

RS R/W DB₇ _____ DB₀

CODE

0	1	BF	A	A	A	A	A	A	A
---	---	----	---	---	---	---	---	---	---

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

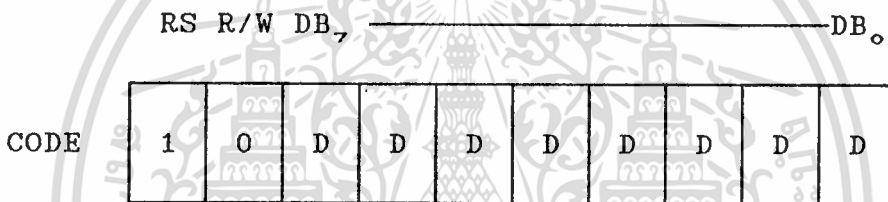
เป็นคำสั่งอ่านค่า BUSY FLAG ซึ่งจะเป็นตัวบอกว่า HD44780 นี้
อยู่ในขบวนการทำงานภายในหรืออยู่ในสภาพพร้อมจะรับข้อมูล โดย

BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือ
คำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่ง

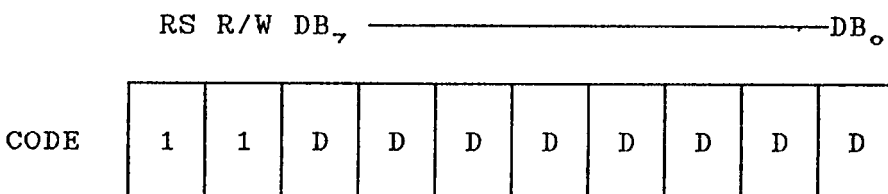
และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล ADDRESS ของ CG RAM
หรือ DD RAM ด้วย

10. WRITE DATA TO CG หรือ DD RAM



เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อ
เขียนข้อมูล และ ADDRESS จะเพิ่มหรือลดโดยอัตโนมัติตามคำสั่งที่ SET ใน
ENTRY MODE ข้อกำหนดที่จะรู้ว่าเป็นการเขียนข้อมูลของ CG RAM หรือ DD
RAM ทำได้โดยการ SET ADDRESS ของ CG RAM หรือ DD RAM ขึ้นมา
ก่อนจะเขียนข้อมูล

11. READ DATA FROM CG OR DD RAM



เป็นคำสั่งอ่านค่าข้อมูลจาก CG RAM หรือ DD RAM โดยก่อนอ่านค่าจาก DD RAM หรือ CG RAM นี้ควรจะใช้คำสั่ง SET ADDRESS ก่อน เพื่อให้รู้ว่าข้อมูลที่อ่านได้นั้นเป็น DD หรือ CG RAM

จากตารางการทำงานจะเห็นว่าการทำงาน LCD MODULE นั้นง่ายเพียงแต่เราส่งคำสั่งเริ่มแรกและ SET ความต้องการในขนาดตัวอักษร, CURSOR หลังจากนั้นเราก็สามารถเขียนตัวอักษรเข้าไปใน DD RAM ตามตารางตัวอักษรที่ให้มานั้นก็จะเกิดอักษรในจอภาพ LCD เรายังสามารถกำหนดตำแหน่งตัวอักษรที่จะให้เกิดบนจอได้โดยการ SET DD RAM ADDRESS ตามตารางที่ให้มาในหัวข้อ SET DD RAM ADDRESS

2.8 ชุดควบคุม Z80 CP-A

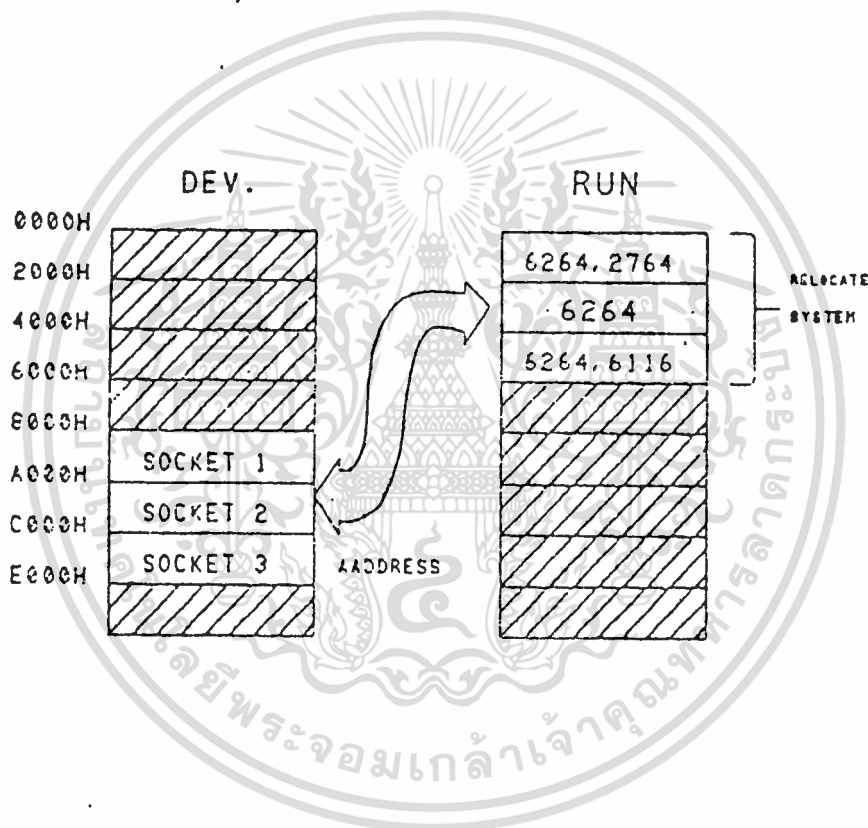
Z 80 CP-A เป็นบอร์ดไมโครโพรเซสเซอร์ขนาดเล็ก ที่ประกอบด้วยระบบพื้นฐานในการทำงาน เช่น CPU , I/O Port , หน่วยความจำ , Clock , วงจรควบคุมแรงดัน , รวมทั้งการจัด Decode ไว้ อย่างเรียบร้อย วงจรสมบูรณ์ของ Z 80 CP-A แสดงไว้ในรูปที่ 31

Z 80 CP-A เป็นโครงสร้างทาง Hardware อย่างเดียว ซึ่งจะต้องมีการพัฒนาโปรแกรมเพื่อให้มันทำงานตามที่ต้องการโดยจะขึ้นอยู่กับลักษณะของงานนั้น ๆ ที่นำไปใช้

Z 80 CP-A มีโหมดในการทำงาน 2 ลักษณะคือ การพัฒนาและการนำโปรแกรมไปใช้งาน ในขั้นของการพัฒนาโปรแกรมนั้น จะทำได้โดยการเสียบเข้ากับบอร์ดได้เลยหรือจะทำโดยใช้เครื่องอื่น ๆ ก็ได้ แล้วจึงนำโปรแกรมมาทำงาน โดยผ่านอีพรอมหรือแรมแพค ซึ่งผู้ใช้จะต้องเขียนโปรแกรมโดยอ้างแอสเตอรตามตำแหน่งที่กำหนดไว้

2.8.1 รายละเอียดในการจัดหน่วยความจำและพอร์ต

การจัดหน่วยความจำของ CP-A นี้จะอ้างแอสแอดเรสเริ่มต้นที่ 8000H เพื่อให้ทำโปรแกรมได้สะดวกจากเครื่องอีทีบอร์ด และเมื่อนำโปรแกรมไปใช้งาน CP-A มีระบบ Relocate เพื่อให้หน่วยความจำไปเริ่มที่ 0000H ได้ (ในโหมด RUN) เพื่อให้เป็นแอสแอดเรสที่ทำงานเมื่อเริ่มเปิดเครื่องหรือกด Reset



รูปที่ 30 การจัดหน่วยความจำใน CP-A

ระบบพอร์ตจะทำกา Decode ไว้ทั้งหมด 8 ช่วง และจะสามารถเลือกเบอร์ได้ 7 ตำแหน่ง โดยการ Jump สายบนปรีน ซึ่งมีเบอร์พอร์ตดังตารางที่ 3 ส่วนเบอร์พอร์ต 40 - 5F นั้น ไว้สำหรับการขยายได้อีกทางด้าน 8255 Connector

COO - 073

IN4148 X 2

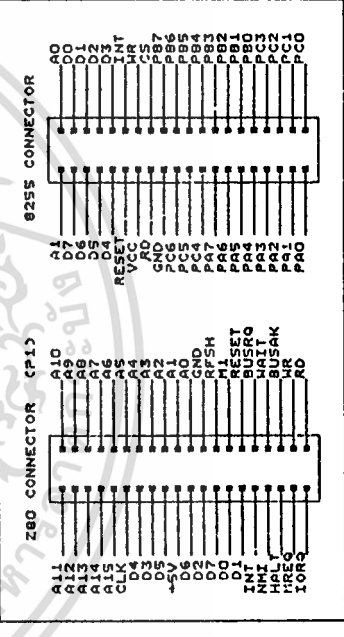
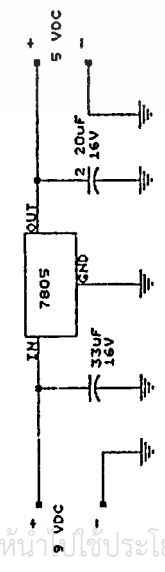
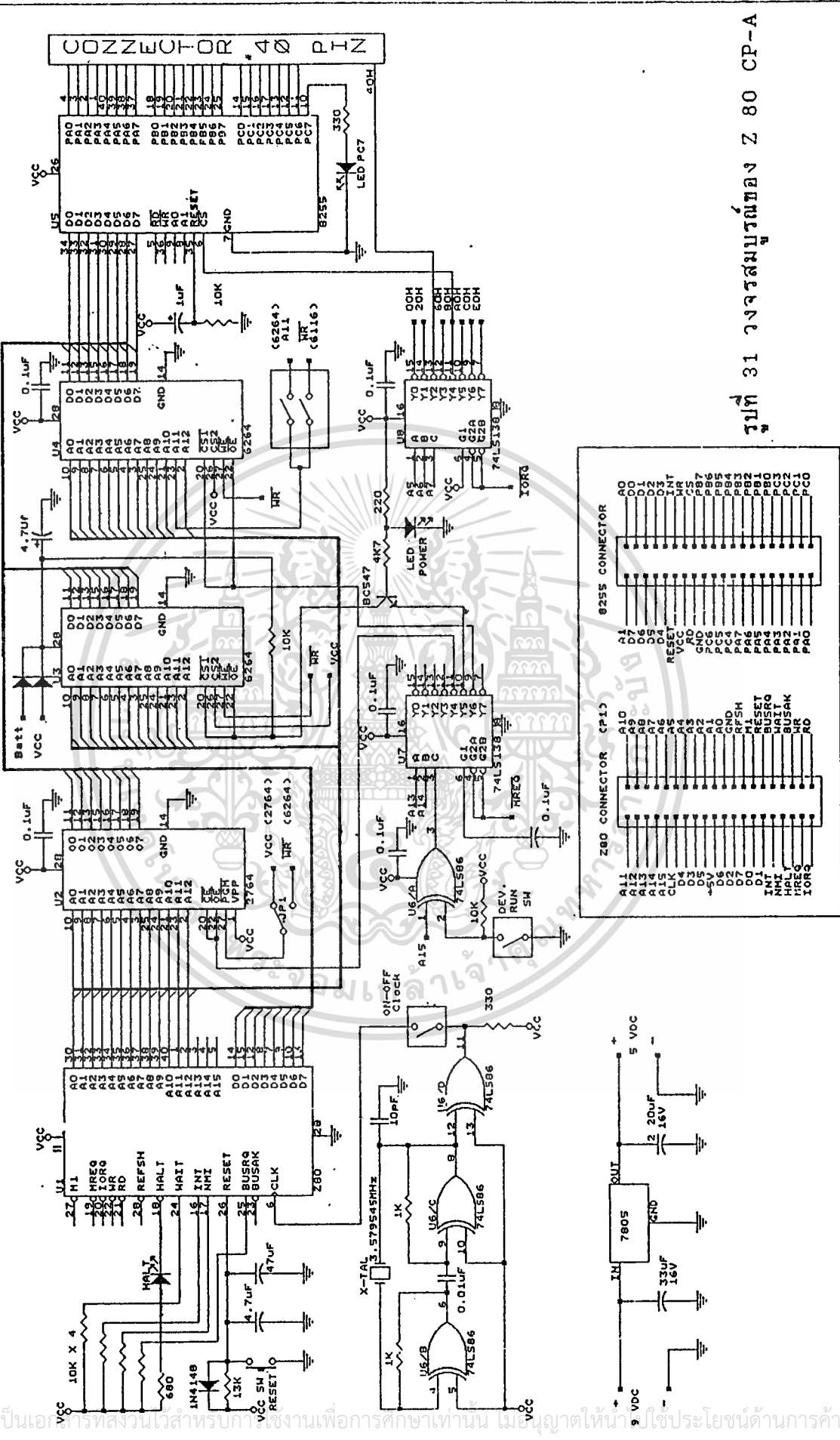
Batt

VCC

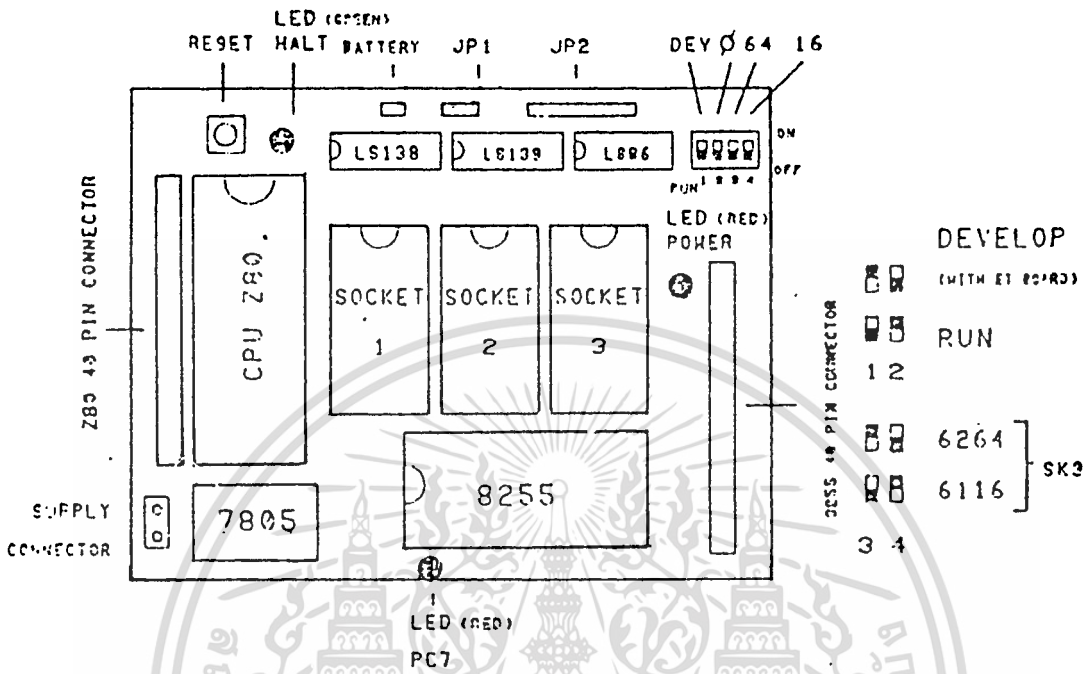
VCC

VCC

VCC



รูปที่ 31 วงจรสมมุติของ Z 80 CP-A



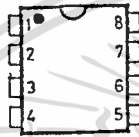
รูปที่ 32 แสดงลักษณะและตำแหน่งอุปกรณ์ของ CP-A

2.9 วงจรตรวจสอบสัญญาณเสียง

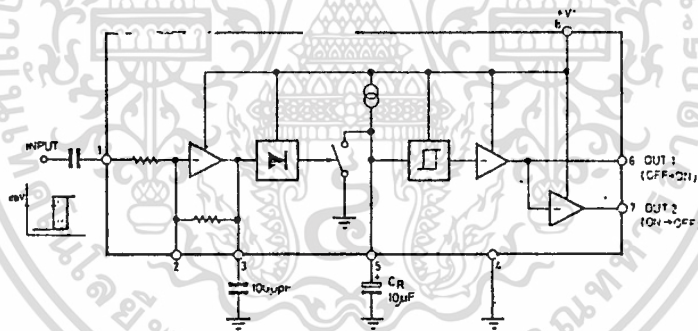
ในการฝากข้อความแต่ละครั้ง หลังจากที่เราบอกให้เริ่มพูดได้ คนที่โทรเข้ามาอาจจะยังไม่พูดในทันที ซึ่งถ้าไม่มีการตรวจสอบสัญญาณเสียง ก็อาจทำให้เกิดการสูญเสียเวลาในการบันทึกไป ดังนั้น ในโครงการงานจึงได้ออกแบบวงจรโดยใช้ไอซีเบอร์ NJM2072D ซึ่งรายละเอียดของไอซีและการนำไปใช้งานจะได้กล่าวดังต่อไปนี้

NJM2072 เป็นไอซีขนาด 8 ขา มีให้เลือก 2 ลักษณะคือ NJM2072D เป็นตัวถังแบบ DIP และ NJM2072M เป็นตัวถังแบบติดตั้งผิวหน้า เป็นไอซีที่ตรวจสอบระดับสัญญาณให้เอาท์พุทเปลี่ยนแปลงจาก ON-OFF หรือ OFF-

ON เมื่อมีสัญญาณอินพุตเข้ามา โดยภาคอินพุตมีอัตราขยายสัญญาณสูงมาก และสามารถปรับอัตราขยายได้ด้วยตัวต้านทาน 2 ตัว (ความไวอินพุตสูงสุด -36 dBV) เอาท์พุทมีการหน่วงเวลาหลังจากไม่มีอินพุตแล้ว โดยเวลาที่กำหนดได้ด้วยตัวเก็บประจุเพียงตัวเดียว เหมาะสำหรับการทำงานจรสวิทช์เสียง หรือ VOX (voice operate switch) เช่น ในเทปบันทึกเสียง, วิทยุรับ/ส่ง, สวิทช์สั่งงานด้วยเสียง, วงจรตัดเสียงรบกวน, วงจรตัดต่อเสียงพากย์, ระบบกันขโมย เป็นต้น

- 
1. อินพุต (INPUT)
 2. ควบคุมอัตราขยาย (ความไว)
 3. เอาท์พุทของออพแอมป์
 4. ทรานซ์ (GND)
 5. ตัวเก็บประจุหน่วงเวลา
 6. เอาท์พุทที่ 1 (OUT 1)
 7. เอาท์พุทที่ 2 (OUT 2)
 8. แรงดันซัพพลาย (V⁺)

รูปที่ 33 แสดงการจัดขาภายนอกของ NJM2072

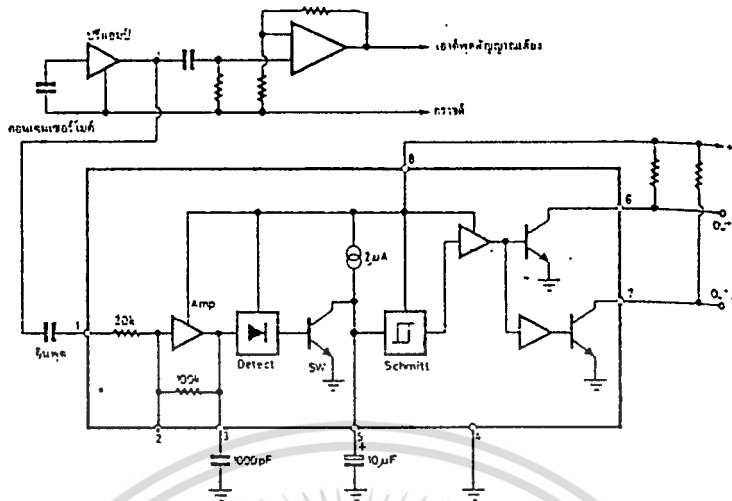


รูปที่ 34 ลักษณะของวงจรภายใน NJM2072

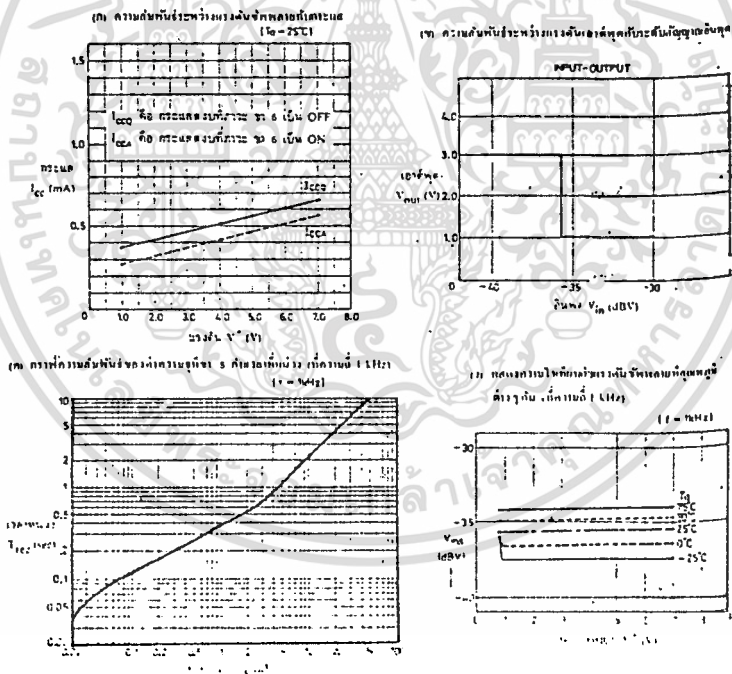
การจัดขาภายนอกแสดงในรูปที่ 33 และลักษณะการจัดวงจรภายใน แสดงในรูปที่ 34 ข้อกำหนดในการใช้งานมีดังนี้ คือ ใช้กับแรงดันได้ตั้งแต่ 0.9-7 โวลท์, กระแสภาวะสงบ 0.55 mA, ความไวอินพุต -36 dBV รูปที่ 36 เป็นกราฟแสดงความสัมพันธ์ของสภาวะต่างๆ

รูปที่ 35 เป็นตัวอย่างวงจรใช้งานสวิทช์ทำงานด้วยเสียง ซึ่งด้านเอาท์พุท OUT1 และ OUT2 ต้องมีตัวต้านทานพูลอัพ โดยคำนวณจากสูตร

$$R_p = (+V_{m.in} - 0.2) / 0.3 \quad \text{kohm.}$$



รูปที่ 35 ลักษณะวงจรใช้งานเบื้องต้นของ NJM2072



รูปที่ 36 กราฟแสดงคุณสมบัติต่างๆ

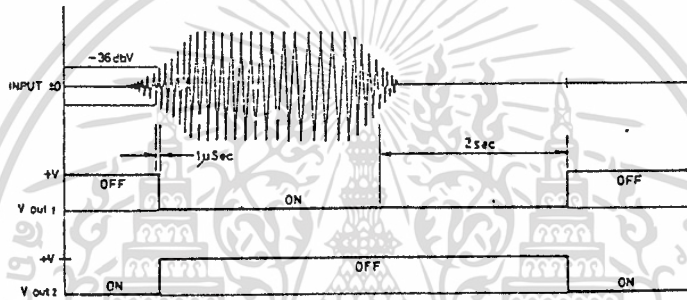
เมื่อ $+V_{m, in}$ หมายถึงแรงดันไฟเลี้ยงต่ำสุดที่กำหนดให้วงจรยังคงทำงานได้ตามปกติ ลักษณะของสัญญาณเอาต์พุตของวงจรเทียบกับสัญญาณอินพุต แสดงในรูปที่ 37 ซึ่งการหน่วงเวลาหลังจากที่สัญญาณอินพุตหมดไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

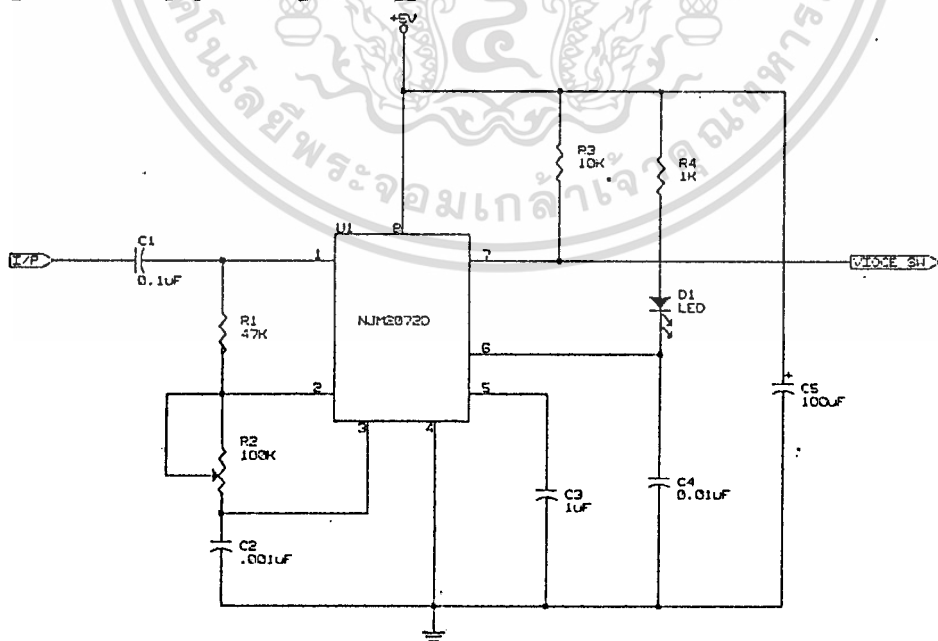
กำหนดได้ด้วยตัวเก็บประจุ ที่ต่อกับขา 5 การคำนวณการหน่วงเวลา ให้ดู จากกราฟในรูปที่ 36 (ค)

2.9.1 วงจร VOX (Voice Operate Switch)

วงจรสวิตช์ทำงานด้วยเสียง (VOX) จะมีลักษณะรูปร่างของ สัญญาณต่างๆ ดังที่แสดงในรูปที่ 37 และมีลักษณะของวงจรดังในรูปที่ 38 ซึ่ง การทำงานจะได้กล่าวถึงดังต่อไปนี้



รูปที่ 37 รูปเปรียบเทียบแสดงการทำงานของสัญญาณอินพุตและเอาต์พุต



รูปที่ 38 วงจรสวิตช์ทำงานด้วยเสียง

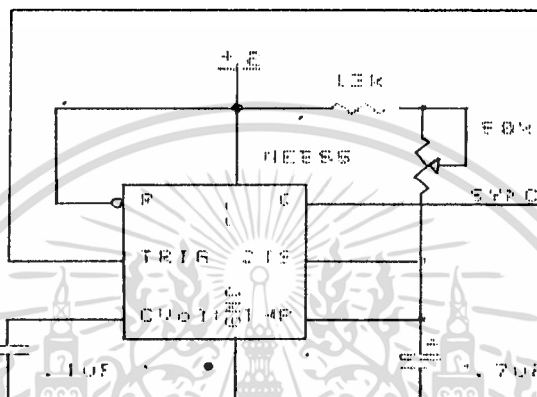
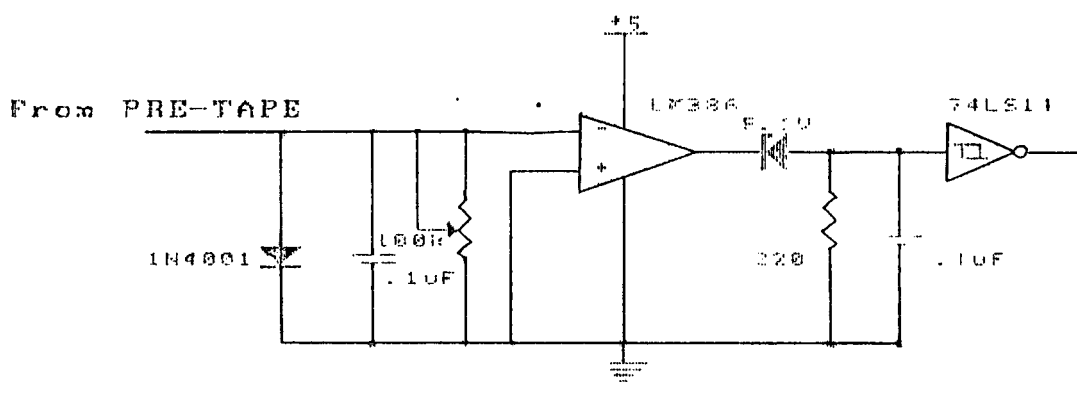
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 การทำงานของวงจร

เมื่อมีสัญญาณเสียง เข้ามาที่ขา 1 ซึ่งเป็นขาอินพุทของไอซี โดยมี R_1 และ R_2 เป็นตัวปรับความไวในการเปลี่ยนสถานะของเอาต์พุท (ในที่นี้ใช้ R_2 เป็นตัวปรับ) ถ้าสัญญาณที่เข้ามามีระดับสัญญาณเพียงพอ จะทำให้ขาเอาต์พุท OUT1 ที่ขา 6 เปลี่ยนสถานะจาก OFF-ON (1-0) ทำให้มีกระแสไหลผ่าน LED ทำให้ LED ติด (LED จะเป็นตัวแสดงสถานะการทำงานของวงจร) ในขณะที่ขา 7 (OUT2) ก็จะเปลี่ยนจากสถานะ ON-OFF (0-1) สัญญาณจะถูกส่งให้กับ CPU นำไปควบคุมวงจรควบคุมการบันทึกให้เริ่มการบันทึกต่อไป และเมื่อหมดสัญญาณอินพุทที่จะส่งเข้ามา วงจรจะรองนกว่าจะครบค่าในการหน่วงเวลา โดย C_3 จะเป็นตัวกำหนดค่าในการหน่วงเวลา ของการเปลี่ยนสถานะของเอาต์พุท (ค่าของ C_3 กับเวลาในการหน่วงให้ดูจากกราฟในรูปที่ 33(ค)) เมื่อไม่มีสัญญาณอินพุทเข้ามาในเวลาที่กำหนดไว้ในการหน่วงเวลาแล้ว วงจรจะเปลี่ยนสถานะที่ขา 6 (OUT1) และที่ขา 7 (OUT2) โดยที่ขา 6 (OUT1) จะกลับเข้าสู่สถานะ OFF (1) ทำให้ LED ดับ และที่ขา 7 (OUT2) จะกลับเข้าสู่สถานะ ON (0) ทำให้ CPU จะควบคุมให้หยุดการบันทึกเทป จนกว่าจะมีสัญญาณอินพุทเข้ามาอีกครั้งหนึ่ง

2.10 วงจรตรวจสอบสัญญาณซิงค์

ในการกำหนดบล็อดข้อความนั้น เราจะกำหนดโดยการบันทึกสัญญาณซิงค์ เข้าไปในเนื้อเทปที่ Channel ขวา ส่วน Channel ซ้ายจะเป็นการบันทึกของสัญญาณเสียง ในการบันทึกสัญญาณซิงค์นั้น เนื่องจากเราต้องการแบ่งเนื้อเทปที่มีความยาว 30 นาที ออกเป็น 15 บล็อค ๆ ละ 2 นาที จึงทำการบันทึกสัญญาณซิงค์ทุก ๆ 2 นาที โดยมีความยาวของสัญญาณซิงค์ประมาณ 1 วินาที สัญญาณที่นำมาใช้บันทึกเป็นสัญญาณซิงค์นี้ก็คือ สัญญาณความถี่ 50 Hz ที่ได้จาก Transformer ของ Power Supply และการตรวจสอบสัญญาณซิงค์นี้ ก็จะได้จากวงจรตรวจสอบสัญญาณซิงค์ ดังรูปที่ 39



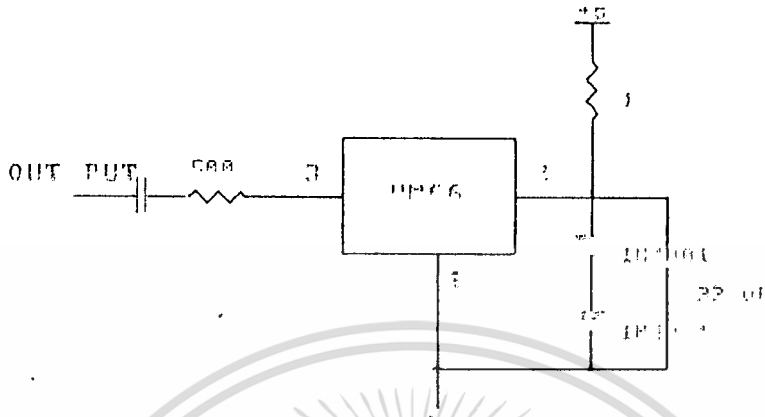
รูปที่ 39 วงจรตรวจสอบสัญญาณซิงค์

ในขณะที่ทำการ Play back สัญญาณจาก Channel ขวาของ เทป ซึ่งบันทึกเป็นสัญญาณซิงค์ไว้จะผ่านวงจร Pre-Tape มาเข้าสู่ IC LM386 ซึ่งจะขยายและส่งผ่านสัญญาณนี้ไป Trig ให้ IC NE555 ซึ่งเป็น วงจร Monostable Multivibrator IC NE555 นี้จะผลิตความถี่ขึ้นมาใน ช่วงจังหวะที่มีการ Trig ซึ่งก็คือช่วงที่มีสัญญาณซิงค์ และ Output ของวง จรก็จะใช้เป็นสัญญาณซิงค์ ส่งไปเข้า Input Port (IC 8255) ต่อไป

ในขณะที่ระบบทำงาน CPU จะตรวจสอบสัญญาณซิงค์ และจะจำ จำนวนซิงค์นี้ไว้เพื่อจะได้รู้ว่าเป็นบล็อกข้อความบล็อกใด และทำงานได้ถูกต้อง

2.11 วงจรภาคขยายเสียง

วงจรมีหน้าที่ในการขยายเสียงข้อความที่บันทึกอยู่ในเทป หรือ ข้อความจากชุด Voice Recorder เพื่อให้มีเสียงดังที่เหมาะสมกับการรับฟัง

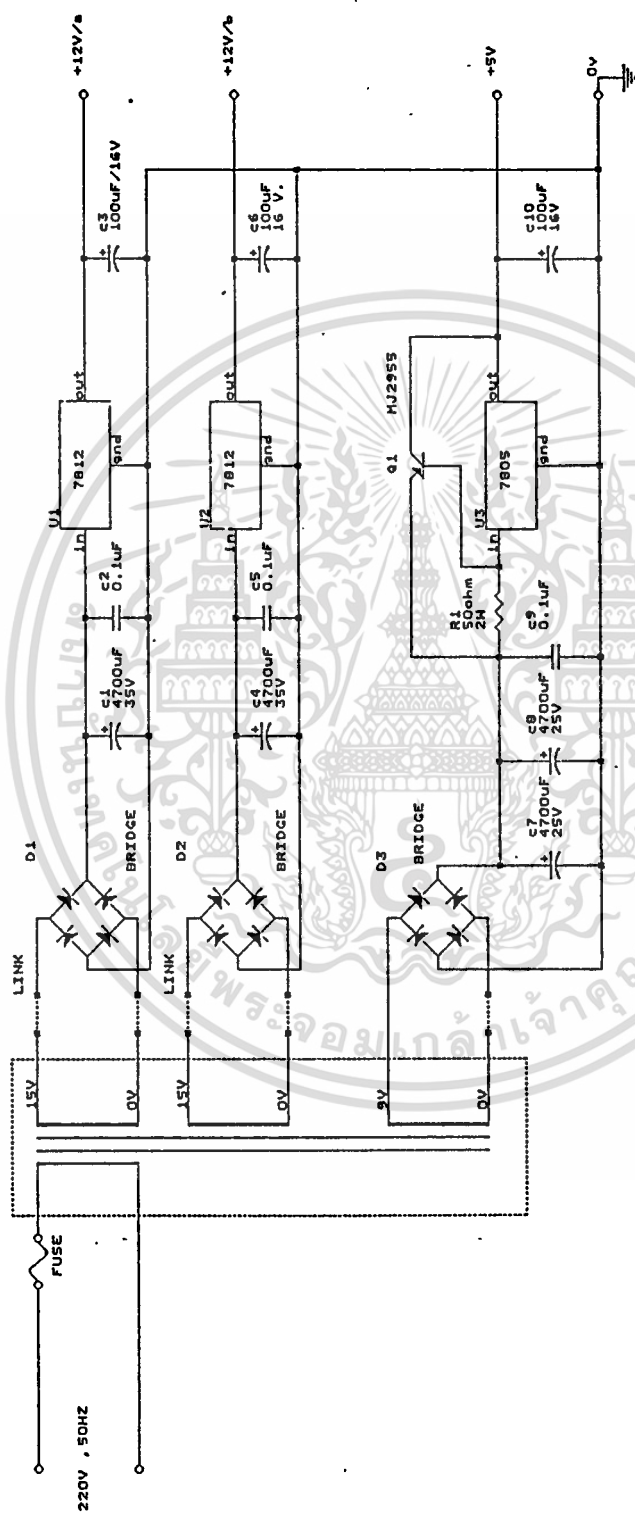


รูปที่ 41 วงจรกำเนิดเสียงดนตรี

2.13 สรุป

จากการที่ได้ศึกษา ทดลอง วงจรในส่วนต่าง ๆ แล้ว ก็ได้ นำ วงจรต่าง ๆ ทั้งหมดนี้ มาประกอบเข้าด้วยกัน เพื่อให้ทำงานร่วมกัน ซึ่ง นอกจากวงจรที่ได้กล่าวไว้แล้วนั้น ก็ยังต้องเพิ่มเติมวงจร และอุปกรณ์บางส่วน เข้าไป เพื่อให้วงจรทำงานได้โดยสมบูรณ์ เช่น Keyboard Coupling Transformer และอื่น ๆ นอกจากนี้ในการทดลองการทำงานของวงจร ยัง พบปัญหาบางอย่าง จึงต้องเพิ่มวงจรบางส่วนเข้าไป ได้แก่ วงจร Low Pass Filter ซึ่งปัญหาและการแก้ไขนั้น จะได้กล่าวต่อไปในบทที่ 5

เมื่อปรับปรุง แก้ไข และทดลองวงจรจนเสร็จเรียบร้อยแล้ว ก็จะได้วงจรสมบูรณ์ของเครื่องบันทึกโทรศัพท์แบบแยกช่อง ดังรูปที่ 42 การทำงานของวงจรได้อธิบายไว้แล้วในบทที่ 1



รูปที่ 43 วงจร Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การทดสอบการทำงาน

หลังจากประกอบวงจรลงบนแผ่นวงจรมพิมพ์เรียบร้อยแล้ว ก็จะทำ การทดสอบการทำงานของวงจร ในการทดสอบนี้จะเริ่มจากการวัดสัญญาณ ตามจุดต่าง ๆ ของวงจร โดยจะทำการทดสอบทีละภาค เพื่อหาลักษณะสมบัติ ของวงจร เช่น ค่าความถี่คัทออฟ ของวงจรรองความถี่ ค่าระดับแรงดันต่ำ สุดของเสียงพูด ที่วงจรตรวจจับสัญญาณเสียงเริ่มทำงาน เหล่านี้เป็นต้น หลังจากนั้นจึงทดลองนำไปใช้งานจริง เมื่อผ่านกระบวนการทดสอบแล้ว เราจะสามารถระบุลักษณะสมบัติของโครงการได้ว่า เป็นไปตามขอบเขตที่ได้วางไว้ แต่แรก มากน้อยเพียงใด โดยมีรายละเอียดของการทดสอบดังต่อไปนี้

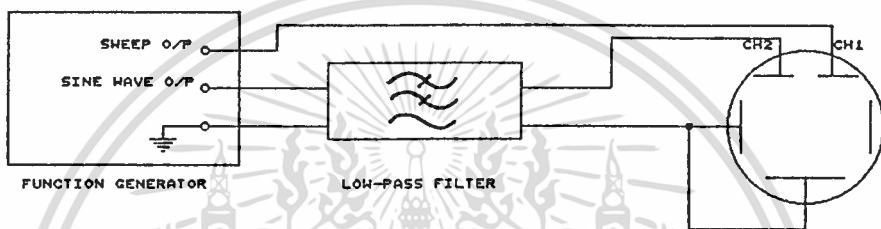
3.1 ผลการทดสอบวงจรต่าง ๆ

3.1.1 วงจรรองความถี่ต่ำผ่าน

ในวงจรส่วนนี้จะทำการทดสอบว่า มีค่าของ High Frequency Cutoff เป็นเท่าไร โดยจะใช้ตัวกำเนิดความถี่ ที่สามารถกำเนิดความถี่ชนิด ถูกควบคุมการกำเนิดด้วยแรงดัน (Voltage Control Oscillator) โดยสัญญาณเอาต์พุตที่เราต้องการคือ สัญญาณแรงดันควบคุมการกำเนิดความถี่ ซึ่งมีลักษณะเป็นสัญญาณรูปฟันเลื่อย และสัญญาณความถี่รูปคลื่นไซน์ ซึ่งค่า ของความถี่ถูกควบคุมให้ เปลี่ยนแปลงตามแรงดันควบคุมจากต่ำไปสูง โดยเรา จะนำสัญญาณความถี่นี้ไปป้อนเข้าที่อินพุตของวงจรรองความถี่ต่ำผ่าน ส่วน แรงดันควบคุมจะต่อเข้า Oscilloscope ที่ CH.1 เพื่อเบี่ยงเบนลำอิลีค ตรอนทางแกน X และเอาต์พุตของวงจรรองความถี่ต่ำผ่าน จะต่อเข้า Oscilloscope ที่ CH.2 เพื่อเบี่ยงเบนลำอิลีค ตรอนทางแกน Y

ลำดับขั้นการทดสอบ

1. ต่อดังรูปที่ 44



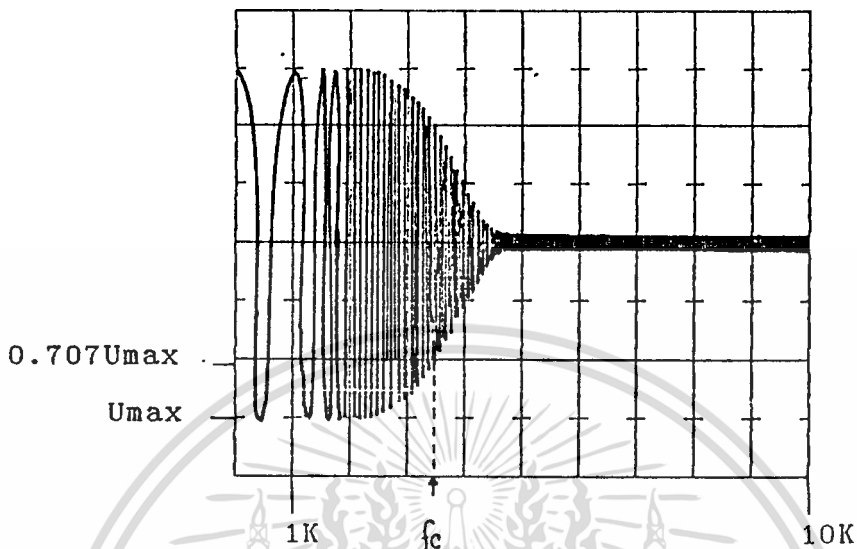
รูปที่ 44 การทดสอบวงจร LOW PASS FILTER

2. ปรับ Function Generator ไปที่ Sweep Mode ตั้งค่าความถี่เริ่มต้น 1 Hz ตั้งค่าความถี่ Stop = 10 KHz ปรับแรงดันไว้ที่ 3 Vp-p เลือกชนิดสัญญาณไปที่รูปคลื่น Sine

3. ปรับ Oscilloscope ดังนี้

- Time Base ปรบไว้ที่ตำแหน่ง EXT HORIZ
- ปุ่ม Volts/DIV ปรับไว้ที่ 0.5 Volts/DIV

4. บันทึกรูปคลื่นสัญญาณที่วัดได้จากเอาท์พุท ของวงจรรอง
ความถี่ต่ำผ่าน



รูปที่ 45 สัญญาณเอาต์พุตของวงจร LOW PASS FILTER

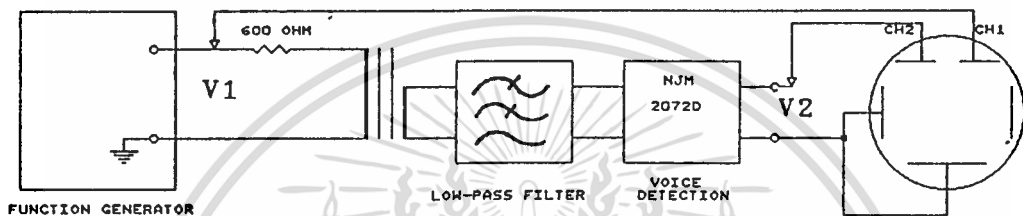
จากรูปจะเห็นว่าค่าความถี่คutoff สูงสุด ประมาณ 3.5 KHz ซึ่งก็คือ ตำแหน่งที่ค่าของแรงดันเท่ากับ 0.707 Umax

3.1.2 วงจรตรวจสอบสัญญาณเสียงพูด

หลังจากที่ได้ทดสอบวงจรกรองความถี่ต่ำผ่าน เสร็จเรียบร้อยแล้ว เราก็จะนำมาใช้เป็นวงจรกรองความถี่ให้กับ วงจรตรวจสอบเสียงพูด โดยในภาคนี้จะทดสอบว่า วงจรจะเริ่มตรวจจับสัญญาณเสียงพูดที่ระดับความแรง ของสัญญาณเสียงความถี่ต่าง ๆ ได้ที่ระดับเท่าไร

ลำดับขั้นการทดลอง

1. ต่อดังรูปที่ 46 .



รูปที่ 46 การทดสอบวงจร VOICE DETECTION

2. ที่ Function Gen. ปรับเลือกเอาท์พุทรูปคลื่น Sine ทดลองปรับค่าความถี่และแรงดัน V1 ค่าต่าง ๆ ตามตารางที่ 8 แล้วบันทึกค่าแรงดัน V2 ที่ได้

จากตารางบันทึกผลการทดลองจะเห็นว่า วงจรตรวจสอบสัญญาณเสียงพูด จะเริ่มทำงานเมื่อระดับ แรงดันของสัญญาณอินพุทที่เข้ามามีค่าประมาณ 200 mVp ที่ความถี่ 5 KHz, 500 mVp ที่ความถี่ 5 KHz และที่ความถี่สูงขึ้น วงจรกรองความถี่ต่ำผ่านจะตัดสัญญาณเหล่านั้นออกไป ทำให้วงจรตรวจสอบสัญญาณเสียงพูดไม่ทำงาน ซึ่งโดยปกติแล้วสัญญาณเสียงพูดในคู่สายโทรศัพท์ จะมีค่าประมาณ 300 mVp

ตารางที่ 9 ผลการทดสอบวงจรตรวจสอบเสียงพูด

1KHz	V1 (Vp)	.02	.04	.06	.08	.10	.20	.30	.40	.60	.70	.80
	V2 (Vp)	0	0	0	0	0	5	5	5	5	5	5
2KHz	V1 (Vp)	.02	.04	.06	.08	.10	.20	.30	.40	.60	.70	.80
	V2 (Vp)	0	0	0	0	0	5	5	5	5	5	5
3KHz	V1 (Vp)	.02	.04	.06	.08	.10	.20	.30	.40	.60	.70	.80
	V2 (Vp)	0	0	0	0	0	5	5	5	5	5	5
3.5KHz	V1 (Vp)	.02	.04	.06	.08	.10	.20	.30	.40	.60	.70	.80
	V2 (Vp)	0	0	0	0	0	0	5	5	5	5	5
4KHz	V1 (Vp)	.02	.04	.06	.08	.10	.20	.30	.40	.50	.60	.70
	V2 (Vp)	0	0	0	0	0	0	0	0	5	5	5
5KHz	V1 (Vp)	.02	.04	.06	.08	.10	.20	.30	.40	.50	.60	.70
	V2 (Vp)	0	0	0	0	0	0	0	0	5	5	5
10KHz	V1 (Vp)	.02	.04	.06	.08	.20	.30	.40	.60	.80	1	1.2
	V2 (Vp)	0	0	0	0	0	0	0	0	0	5	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 วงจรตรวจสอบการยกหูโทรศัพท์

ลำดับขั้นตอนการทดสอบ

1. ต่อคู่สายโทรศัพท์เข้ากับ อินพุทของวงจรตรวจสอบ การยกหูโทรศัพท์ โดยมีเครื่องโทรศัพท์ชนิด TONE ต่อพ่วงอยู่
2. ใช้ Voltmeter วัดค่าแรงดันของคู่สายโทรศัพท์ ได้ 48 V.
ใช้ Voltmeter วัดค่าแรงดันที่ขา 5 ของไอซีได้ 0 V.
3. ยกหูโทรศัพท์
4. ใช้ Voltmeter วัดค่าแรงดันของคู่สายโทรศัพท์ ได้ 7 V.
5. ใช้ Voltmeter วัดค่าแรงดันที่ขา 5 ของไอซีได้ 5 V.

3.1.4 วงจรตรวจสอบสัญญาณกระดิ่ง

ลำดับขั้นตอนการทดสอบ

1. ต่อคู่สายโทรศัพท์เข้ากับอินพุทของวงจรตรวจสอบสัญญาณกระดิ่ง
2. ใช้ Volt meter วัดแรงดันไฟสลัปที่คู่สายโทรศัพท์ ขณะที่ไม่มีสัญญาณกระดิ่งเข้ามาได้ 0 Volt และได้ 50 Volt เมื่อมีสัญญาณกระดิ่งเข้ามา
3. ใช้ Volt meter วัดแรงดันที่ขา 5 ของไอซี ได้ 5 Volt เมื่อไม่มีสัญญาณกระดิ่งเข้ามา และได้ 0 Volt เมื่อมีสัญญาณกระดิ่งเข้ามา

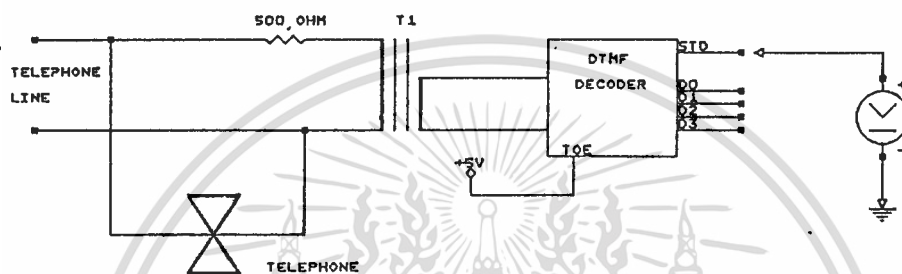
3.1.5 วงจรถอดรหัสสัญญาณ DTMF

เราจะทำการทดสอบการถอดรหัสสัญญาณความถี่ DTMF โดยอาศัยสัญญาณ DTMF จากเครื่องโทรศัพท์ที่ต่อพ่วงอยู่กับวงจรถอดรหัส ซึ่งเมื่อมีสัญญาณ DTMF เข้ามา และวงจรถอดรหัสสามารถถอดรหัสได้ มันจะให้เอาท์พุท ออกมา 5 เอาท์พุท ด้วยค่าแรงดันขนาดประมาณ 5 โวลต์ ซึ่งประกอบไป

ด้วยค่าของคีย์ ที่กด 4 เอาท์พุท และสัญญาณ STD อีก 1 เอาท์พุท

ลำดับขั้นตอนการทดสอบ

1. ประกอบวงจรตามรูปที่ 47



รูปที่ 47 การทดสอบวงจร DTMF Decoder

2. ใช้มือกดปุ่มหมายเลข "1" ที่หน้าปัทม์ของเครื่องโทรศัพท์ แล้วใช้มิเตอร์วัดแรงดันที่ขา 8 (STD) และที่ขา 11, 12, 13, 14 (DO-D3) แล้วบันทึกค่าที่ได้ลงในตารางที่ 10
3. ทำซ้ำข้อ 2 แต่เปลี่ยนไปกดหมายเลขที่เหลือคือ 2 ถึง 9 และ 0, *, #

2.1.6 วงจรปริเทปและวงจรตรวจสอบสัญญาณซิงค์

ในวงจรส่วนนี้ จะทดสอบการบันทึกเสียงลงบนเทปคาสเซ็ท โดยจะทดสอบการบันทึกสัญญาณซิงค์ ลงที่ช่องสัญญาณข้างขวา โดยใช้สื่อยุขรความถี่ 50 Hz ที่ได้จาก Transformer ของวงจร Power Supply ส่วนช่องสัญญาณข้างซ้าย จะใช้สัญญาณความถี่ 1 KHz ที่ได้จาก

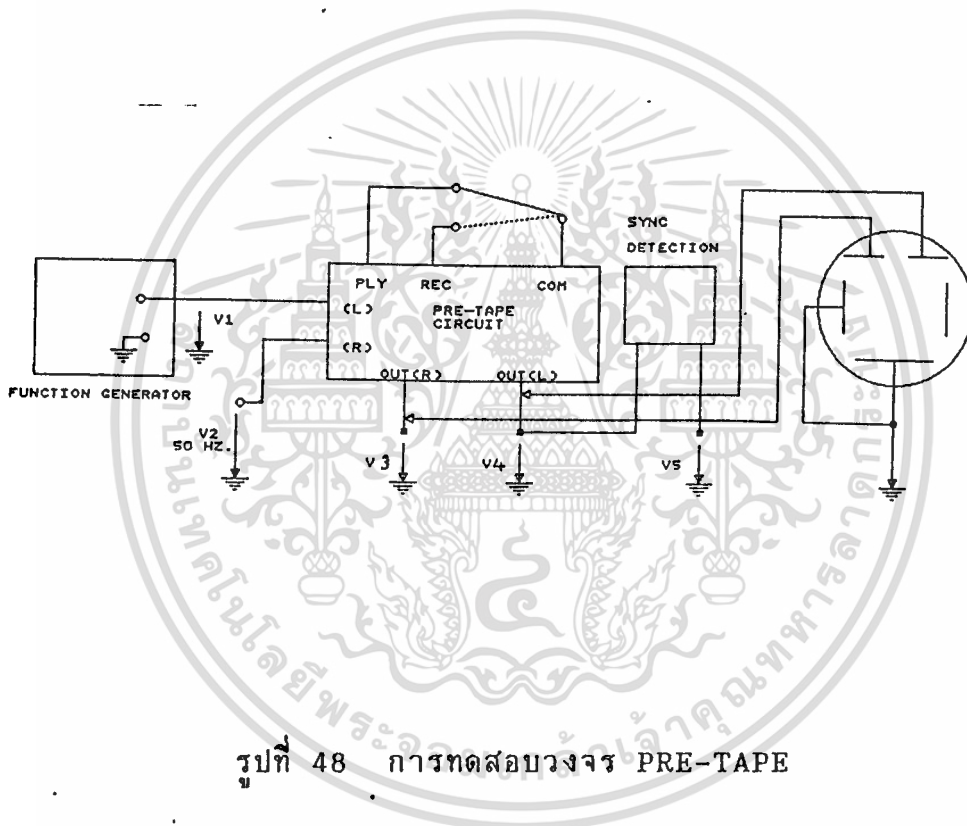
KEY	D3(V)	D2(V)	D1(V)	D0(1)	STD(V)
0	0	0	0	5	5
1	0	0	5	0	5
2	0	0	5	0	5
3	0	0	5	5	5
4	0	5	0	0	5
5	0	5	0	5	5
6	0	5	5	0	5
7	0	5	5	5	5
8	5	0	0	0	5
9	5	0	0	5	5
0	5	0	5	0	5
*	5	0	5	5	5
#	5	5	0	0	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Generator ซึ่งโดยปกติของสัญญาณข้างนี้ จะใช้บันทึกสัญญาณเสียงพูด

ลำดับขั้นการทดสอบ

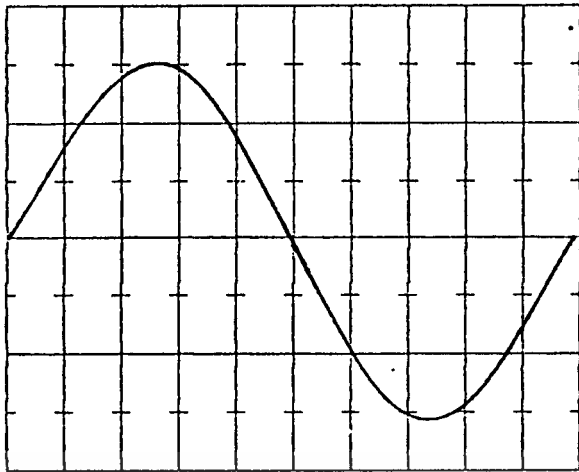
1. ต่อวงจรดังรูปที่ 48



รูปที่ 48 การทดสอบวงจร PRE-TAPE

2. ปรับ Function Generator ให้ได้เอาต์พุตรูปคลื่น Sine ที่ความถี่ 1 KHz ขนาด 1.2 Vp
3. ใช้ Oscilloscope วัดสัญญาณอินพุตของวงจรปริ๊เทป ทั้งสองข้าง (L,R) แล้วบันทึกผล

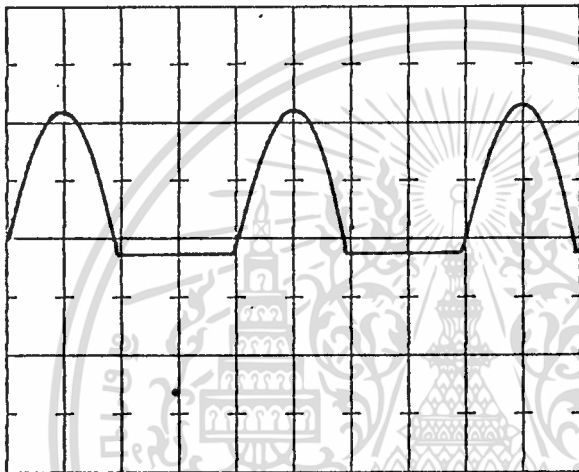
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



input (L)

.1 mS/DIV

.5 Volts/DIV



input (R)

5 mS/DIV

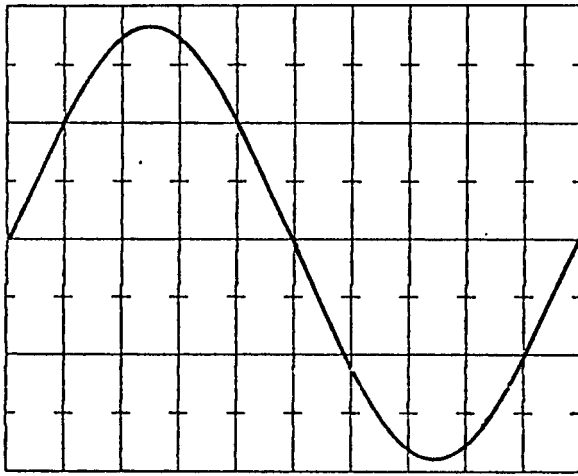
5 Volts/DIV

รูปที่ 49 รูปคลื่นสัญญาณอินพุทของวงจรถ่าย PRE-TAPE

4. วัดแรงดันเอาต์พุทของวงจรถ่ายตรวจสอบสัญญาณซึ่ง V5 ได้เท่ากับ 0 โวลต์

5. เริ่มบันทึกสัญญาณ โดยที่วงจรถ่ายให้ต่อจุด REC ไปที่ COM และที่ชุดกลไกเทป ให้ต่อไฟ 12 Vdc เข้าที่ขั้วลวดโซลินอยด์ชุด Head, Rec, Motor เป็นเวลาประมาณ 1 นาที

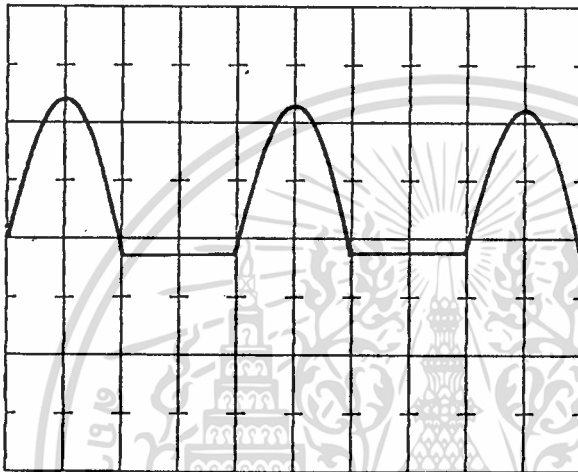
6. กรอเทปกลับ แล้วทดลอง Playback สัญญาณที่บันทึกไว้ โดยที่วงจรถ่ายให้ต่อจุด Play ไปที่ COM และที่ชุดกลไกเทป ให้ต่อไฟ 12 Vdc เข้าที่ขั้วลวดโซลินอยด์ ชุด Head, Motor แล้วใช้ Oscilloscope วัดสัญญาณเอาต์พุทของวงจรถ่ายทั้งสองข้าง แล้วบันทึกผล



input (L)

.1 mS/DIV

1 Volts/DIV



input (R)

5 mS/DIV

5 Volts/DIV

รูปที่ 50 รูปคลื่นสัญญาณเอาต์พุทของวงจรปริเทป

7. วัดแรงดันเอาต์พุทของวงจรตรวจสอบสัญญาณซึ่ง V5 ได้เท่ากับ 5 โวลต์

เมื่อผ่านการทดสอบทุกส่วนแล้ว จึงนำไปทดลองติดตั้งใช้งาน โดยขั้นตอนของการใช้งานได้กล่าวไว้ในหัวข้อถัดไป ในการทดสอบใช้งาน ได้ทดลองติดตั้งเครื่อง เข้ากับคู่สายโทรศัพท์ของสำนักงานแห่งหนึ่ง ซึ่งเป็นคู่สายที่ผ่านเครื่องชุมสายขนาดเล็ก (PABX) ของ PANASONIC

3.2 การทดสอบการทำงานของฟังก์ชันต่าง ๆ

1. ทดสอบการโปรแกรมหมายเลข Block code และ Password ด้วยฟังก์ชันที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทดสอบการบันทึกเสียง ลงเครื่องบันทึกเสียงระบบดิจิทัล ด้วยฟังก์ชันที่ 3
3. ทดสอบการ Playback ข้อความในเครื่องบันทึกเสียงระบบดิจิทัล ด้วยฟังก์ชันที่ 4
4. ทดสอบการบันทึกสัญญาณเชิงค้ด้วยฟังก์ชันที่ 7
5. ทดสอบการบันทึกเสียง ลงบนเครื่องบันทึกเสียงคาสเซ็ตเทป แบบ Manual ด้วยฟังก์ชันที่ 5
6. ทดสอบการเล่นกลับ (Playback) ข้อความจากเครื่องบันทึกเสียงคาสเซ็ตเทป ด้วยฟังก์ชันที่ 6
7. ทดสอบการบันทึกเสียง และเล่นกลับ เครื่องบันทึกเสียงคาสเซ็ตเทปแบบอัตโนมัติ ด้วยฟังก์ชันที่ 1



บทที่ 4

การใช้งานเครื่องบันทึกเสียงโทรศัพท์แบบแยกช่อง

เครื่องบันทึกเสียงโทรศัพท์แยกช่องนี้ มีการทำงานเปรียบเสมือนการรวมเอาเครื่องบันทึกเสียงโทรศัพท์ธรรมดา 5 เครื่องมารวมเข้าไว้ด้วยกัน (สำหรับคู่สายโทรศัพท์ 1 คู่สาย) และใช้คาสเซ็ทเทปเพียงตลับเดียว เมื่อนำมารวมเข้าด้วยกันแล้ว เราก็ใช้รหัสตัวเลขเป็นตัวแบ่งแยกเครื่องทั้งห้าดังกล่าว ซึ่งต่อไปนี้จะแทนชื่อโทรศัพท์แต่ละเครื่องด้วยคำว่า "บล็อก (block)" และในการบันทึกเสียง (Record) แต่ละบล็อกจะถูกแทนด้วยตัวเลขหนึ่งหลัก ตั้งแต่ 1 ถึง 5 ตามลำดับ ส่วนการเล่นกลับ (Playback) จะใช้ตัวเลข 4 หลักแทนบล็อกดังกล่าว และนอกจากนี้ในแต่ละบล็อกยังแบ่งออกเป็น 3 ส่วน (3 subblocks) ซึ่งก็คือการเตรียมเนื้อที่สำหรับการบันทึกเสียง (Record) หรือการเล่นกลับ (Play back) สลับกันไปมาอยู่แค่สามส่วนนี้เท่านั้นในแต่ละบล็อก โดยที่แต่ละส่วนสามารถบันทึกได้นานที่สุด 2 นาที ในการบันทึกเสียงนั้นสามารถทำได้สองกรณี คือ

- กรณีโทรเข้า ถ้าหากไม่มีผู้รับสายภายใน 25 วินาที หรือหลังจากสัญญาณกระดิ่งครั้งที่ห้าดังขึ้นเครื่องก็จะเริ่มทำงานเอง โดยอัตโนมัติ แต่ถ้าต้อง การให้เครื่องเริ่มทำงานให้เร็วขึ้น ก็สามารถสั่งได้ในฟังก์ชันที่ 1

- ในระหว่างการสนทนาระหว่างคู่สาย (ใช้ Function 5)

4.1 รายละเอียดของ Function

ผู้สร้างได้พยายามออกแบบสร้างเครื่องนี้ให้สะดวก และง่ายต่อการใช้งาน โดยแบ่งการทำงานออกเป็นฟังก์ชันต่างๆ 7 ฟังก์ชัน ผู้ใช้สามารถติดต่อกับเครื่องโดยการป้อนคำสั่งผ่านทางคีย์บอร์ด ซึ่งมีอยู่ 12 คีย์ และมีจอแสดงผล LCD ขนาด 20 ตัวอักษร 2 บรรทัด ซึ่งจะช่วยให้การติดต่อสั่งงานระหว่างผู้ใช้กับเครื่องเป็นไปอย่างมีประสิทธิภาพมากขึ้น โดยแต่ละฟังก์ชัน

มีรายละเอียดหน้าที่การทำงานดังนี้

- Function1 "System Run" มีหน้าที่ดังนี้

สอบถามผู้ใช้ว่า ต้องการให้เครื่องรอรับสัญญาณกระดิ่ง ก่อนที่เครื่องจะ เริ่มทำงาน นานสองครั้ง หรือห้าครั้ง จากนั้นจะสอบถามว่าต้องการกรอเทปไปตำแหน่งที่ว่าง (เพื่อให้พร้อมบันทึกข้อมูลได้ตลอดเวลา) หรือไม่ จากนั้นก็จะคอยตรวจสอบสัญญาณกระดิ่ง เพื่อทำหน้าที่เป็นเครื่องบันทึกเสียง โทรศัพท้อัดโน้มนัดต่อไป

- Function2 "Program block code and password"

รับการโปรแกรมหมายเลขที่ส่วาน (Password) ของเครื่อง และการโปรแกรมหมายเลขของบล็อกทั้งห้า (Block code) โดยเป็นเลขสี่หลักค่าที่โปรแกรมใหม่จะคงอยู่จนกว่าจะมีการปิดไฟ (Power off)

- Function3 "Digital Voice Record"

เป็นการบันทึกข้อความที่จะให้โต้ตอบกับผู้ใช้ขณะโทรเข้า โดยข้อความ ดังกล่าวจะถูกบันทึกลงเครื่องบันทึกเสียงระบบดิจิทัล ข้อความที่ต้องบันทึกมีอยู่ทั้งหมดสิบข้อความ ข้อความจะคงอยู่จนกว่าจะปิดไฟ (Power off)

- Function4 "Digital voice Play back"

ทำการเล่นกลับข้อความที่บันทึกด้วย Function3

- Function5 "Manual Record"

เป็นการสั่งให้เครื่องบันทึกเสียงจากคู่สายโทรศัพทค์ ปกติจะใช้ในระหว่างการสนทนาระหว่างคู่สายโดยการสั่งงานผ่านทางคีย์บอร์ดที่ตัวเครื่อง

- Function6 "Tape Play back"

เป็นการเล่นกลับข้อความที่ได้ถูกบันทึกไว้ในเทป ทั้งจากฟังก์ชันที่ 1 และฟังก์ชันที่ 5

- Function7 "Tape Format"

ทำการบันทึกสัญญาณซิงค์ ซึ่งจะถูกใช้เป็นเครื่องหมายในการแบ่งแยกตำแหน่งของข้อมูล ฟังก์ชันนี้จะทำเพียงครั้งเดียวเมื่อเริ่มนำตลับเทปใหม่มาใช้เท่านั้น

4.2 ขั้นตอนการติดตั้งใช้งาน

1. ต่อสายไฟ AC ที่ด้านหลังเครื่อง เข้ากับแหล่งจ่ายไฟ AC 220 V.

2. เมื่อเปิดสวิตช์ จะปรากฏข้อความ "Muti - channel recorder for a telephone line " โดยข้อความนี้จะวิ่งจากขอบด้านขวาของจอแสดงผล LCD ในบรรทัดที่1 และข้อความ "MAIL -BOX TELEPHONE" วิ่งสวนมาจากขอบด้านซ้ายไปขวาในบรรทัดที่2 และจบด้วยข้อความดังนี้

:System ready: ----- (1)
:Function Request:

ข้อความข้างบนนี้จะปรากฏให้เห็นทุกครั้งที่ระบบต้องการ การป้อนคำสั่งจากผู้ใช้ ซึ่งเกิดได้สองกรณี คือ

- หลังจากเริ่มเปิดเครื่อง
- หลังจากจบการทำงานบางฟังก์ชัน

ซึ่งเป็นการบอกว่าระบบพร้อมแล้วที่จะรับคำสั่งต่อไป ให้ผู้ใช้กดคีย์ "Funct." แล้วตามด้วยหมายเลขของฟังก์ชันที่ต้องการ

3. ต่อจากนั้นก็จะเป็นการบันทึกสัญญาณซิงค์ เพื่อกับหนดตำแหน่งบล็อกข้อมูล โดยใช้ฟังก์ชันที่ 7 ซึ่งการบันทึกซิงค์นี้จะใช้เวลาประมาณ 1/2 ชั่วโมง ระยะห่างระหว่างซิงค์ 2 นาที การบันทึกทำเพียงครั้งเดียว สามารถนำไปใช้ได้ตลอดไปจนกว่าจะเกิดการขัดข้องหรือชำรุดของเนื้อเทป

ขั้นตอนการบันทึกซิงค์

- เริ่มจากข้อความที่ (1) ให้กดคีย์ "FUNCT." ที่ LCD จะ

ปรากฏข้อความ

```
:System ready:
[OK! Select<1-7>] _
```

จากนี้ให้กดหมายเลข "7" แล้วตามด้วย "ENTER" ที่ LCD จะปรากฏข้อความ

```
Funct.7 Tape format
PASSWORD IS :
```

ให้ใส่หมายเลข รหัสผ่าน (PASSWORD) ซึ่งถ้าไม่ได้เปลี่ยนแปลงค่า (ฟังก์ชันที่ 2) ค่ารหัสผ่าน ก็จะเป็น "5555" จากนั้นให้กด "ENTER" จะปรากฏข้อความดังนี้

```
Funct.7 Tape format
Press ENTER to start
```

จากนี้ถ้าเรากด "ENTER" เครื่องก็จะเริ่มทำการ บันทึกสัญญาณซิงค์ โดยในระหว่างการบันทึก จะทำการแสดงหมายเลขของบล็อกที่กำลังทำการบันทึกอยู่ และให้เครื่องหมาย ">" เคลื่อนที่จากซ้ายไปขวาแสดงว่าเครื่องกำลังทำงาน

Funct.7 Tape format
Block No. n :>>>>>>

เมื่อบันทึกสัญญาณซึ่งครบแล้ว (16 ซึ่งค์) เครื่องก็จะทำการตรวจสอบว่าการบันทึกสมบูรณ์หรือไม่ โดยจะทำการอ่านจำนวนซึ่งค์ว่าครบ 16 ซึ่งค์ หรือไม่ โดยจะแสดงข้อความระหว่างการตรวจสอบดังนี้

Funct.7 Tape format
format in verify

ถ้าจำนวนซึ่งค์ไม่ครบก็จะแสดงข้อความดังนี้

verify error!
tape too short

บอกให้ผู้ใช้งานทราบเพื่อที่จะได้ทำฟังก์ชันนี้ใหม่ สาเหตุที่ผิดพลาดมักมาจากการใช้ชนิดของเทปคาสเซตที่สั้นเกินไป (ชนิดที่ถูกต้องคือเทปที่มีความยาว 1 ช.ม. หรือ 30 นาทีต่อหน้า) แต่การผิดพลาดนี้ไม่มีผลต่อการทำงานของระบบมากนัก ถึงแม้จำนวนซึ่งค์ที่บันทึกจะไม่ครบ การทำงานของเครื่องก็ยังคงทำงานได้ เพราะปกติแล้วข้อมูลมีการบันทึก และเล่นกลับอยู่เสมอ ตำแหน่งที่ถูกเล่นกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วจะถูกนำมาบันทึกอีก ทำให้การใช้เนื้อเทป วนอยู่ตอนต้นๆ วนเท่านี้ ยกเว้นมีการบันทึกเข้ามามาก (ซึ่งต้องมากจนเต็มเนื้อเทป) โดยที่ไม่มีการขอฟังข้อความที่บันทึกไว้เลย ซึ่งมีโอกาสเป็นไปไปได้้น้อยมาก

4. ใช้ฟังก์ชันที่ 2 ทำการโปรแกรมหมายเลขของรหัสผ่าน (PASSWORD) และหมายเลขรหัสของแต่ละบล็อก ซึ่งเป็นเลขสี่หลักเพื่อใช้ในการขอฟังข้อความที่ถูกบันทึกไว้ (ในการบันทึกแต่ละบล็อกแทนด้วยตัวเลข 1 หลัก ตั้งแต่ 1 ถึง 5 ตามลำดับ) โดยมีขั้นตอนดังนี้

- เริ่มจาก การแสดงผลข้อความหมายเลขที่ (1) ให้กดคีย์ "FUNCT." และหมายเลข "2" จะปรากฏข้อความดังนี้

Funct.2 Program Code
PASSWORD IS:

ให้ใส่รหัสผ่านค่าเดิม ซึ่งเป็นตัวเลขสี่หลัก แล้วตามด้วย "ENTER" ถ้าถูกต้อง จะปรากฏข้อความดังนี้

PASSWORD OK! (XXXX)
new Password :

"XXXX" = ค่า Password เดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนี้ให้ใส่ค่า PASSWORD ใหม่แล้วตามด้วย "ENTER" หรือถ้าไม่ต้องการเปลี่ยน ก็ให้กด "ENTER" จากนั้นจะเข้าสู่การโปรแกรมหมายเลขรหัสของแต่ละบล็อกเพื่อใช้ในการ ขอฟังข้อความ โดยต่อจากข้างต้น ที่ LCD จะแสดงข้อความต่อไปนี้

Block #n CODE (XXXX)
fill new code:

"n" = หมายเลขของบล็อก (1-5) ซึ่งใช้สำหรับการบันทึก
"XXXX" = หมายเลขเดิมของบล็อก ซึ่งใช้สำหรับการขอฟัง
ข้อความที่ได้บันทึกไว้

ถ้าต้องการเปลี่ยนรหัส ก็ให้ใส่ตัวเลขหลักได้เลย แล้วตามด้วย "Entr" แต่ถ้าไม่ต้องการ ก็ให้กด "Entr" แต่เพียงอย่างเดียว ให้ทำไปจนครบทั้งห้าบล็อก โดยเมื่อครบทั้งห้าบล็อกแล้ว จะปรากฏข้อความดังนี้

Funct.2 complete

จากนี้เครื่องจะเข้าสู่การตรวจสอบสัญญาณกระดิ่งโดยอัตโนมัติ

5. ใช้ฟังก์ชันที่ 3 บันทึกข้อความลงในส่วนของการบันทึกเสียงระบบดิจิทัลซึ่งข้อความที่บันทึก จะเป็นข้อความที่จะแนะนำขั้นตอนการ

ใช้เครื่องทดลองจนความผิดพลาดต่างๆที่เกิดขึ้น ให้นักับผู้โทรเข้าได้ทราบ โดยมีขั้นตอนดังนี้

ถ้าทำต่อจากฟังก์ชันที่ 2 เครื่องจะอยู่ในระหว่างการตรวจสอบสัญญาณ ภารกิจ ให้นักกดคีย์ใดๆเพื่อหยุดการทำงานนั้นไว้ก่อน จากนั้นเครื่องจะแสดงข้อความแบบที่ (1) ให้นักกด คีย์ "Funct." และตามด้วยเลข 3 ที่จอแสดงผลจะปรากฏข้อความดังนี้

Funct.3 Instruction
voic record

Funct.3 Instruction
Password is:

ให้ใส่ค่า PASSWORD แล้วตามด้วย ENTER ที่จอแสดงผลจะแสดงข้อความดังนี้

Digital record # n
ENTER for start/stop

"n" คือลำดับของข้อความที่บันทึก (ตั้งแต่ 0-9)

จากฟังก์ชันคีย์ "ENTER" เพื่อเริ่มบันทึกข้อความ โดยที่ LCD จะแสดงข้อความ

Record to CH. # n
to be recording

เมื่อสิ้นสุดการบันทึกข้อความในแต่ละครั้งให้กด "ENTER" และให้กด "ENTER" อีกครั้งเพื่อเริ่มบันทึกข้อความต่อไป ให้ทำไปจนถึงข้อความที่ 9

ข้อความที่ต้องบันทึกมีดังต่อไปนี้

ข้อความที่ 0. "เสียงที่ท่านได้ยินอยู่นี้เป็นเสียงจากเครื่องบันทึกเสียงโทรศัพท์อัตโนมัติ กรุณากดเครื่องหมาย "*" เพื่อฝากข้อความ หรือกดเครื่องหมาย "#" เพื่อขอฟังข้อความ ขอขอบคุณ"

ข้อความที่ 1. "กรุณากรอกหมายเลขช่องที่ท่านต้องการฝากข้อความ ถ้าไม่ทราบให้กดเครื่องหมาย "*" "

ข้อความที่ 2. "ขอภัย ไม่พบเนื้อที่ว่างในช่องนี้ "

ข้อความที่ 3. "กรุณารอสักครู่"

ข้อความที่ 4. "ระบบพร้อมแล้ว เครื่องจะเริ่มบันทึกเสียงของท่านเป็นเวลาไม่เกินสองนาทีนับจากนี้ไป กรุณากรอกเครื่องหมาย "*" เมื่อเลิกบันทึกข้อความ"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Quick Anser! or not?
Press 1=yes/0=no

ถ้ากด "1" จะเป็นการ RUN ในโหมด QUICK ANSER ซึ่งเครื่องจะรอสัญญาณกระดิ่งเพียงลูกที่สองเท่านั้น แล้วจึงเข้าสู่การยกหูโทรศัพท์โดยอัตโนมัติ แต่ถ้ากด "0" จะเป็นการรอสัญญาณกระดิ่งถึงลูกที่ห้า จึงจะทำการยกหู จากนั้นเครื่องจะถามต่อดังนี้

tape Adj. or not?
Press 1=yes/0=no

ถ้ากด "1" ก็จะเป็นการสั่งให้เครื่องทำการกรอเทปไปหยุดอยู่ที่ตำแหน่งที่ว่าง พร้อมทั้งจะบันทึกข้อมูลครั้งต่อไปได้ตลอดเวลา แต่ถ้ากด "0" ก็จะเป็นการตรวจสอบสัญญาณกระดิ่งต่อไป โดยมีรูปแบบการแสดงผลดังนี้

<MAIL BOX TELEPHONE>

สลักับข้อความ

-B1--B2--B3--B4--B5--
000 000 000 000 000

----- (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงข้อความข้างต้นจะมีอยู่สองข้อความสลับกันไป ข้อความชุดที่สอง จะเป็นการบอกให้ผู้ใช้ทราบว่าในแต่ละบล็อกมีข้อความถูกบันทึกไว้หรือไม่ โดย ถ้ามีข้อความถูกบันทึกอยู่จะแทนด้วยเลข "1" ถ้าไม่มีก็จะแทนด้วยเลข "0" จะเห็นว่าแต่ละบล็อกมีบล็อกย่อยอยู่สามบล็อกย่อย ซึ่งก็จะแทนสถานะของ ข้อมูลด้วยตัวเลขสามตัวในบันทึกที่สอง และถ้ามีข้อความใดถูกฟังไปแล้ว ตัวเลขก็จะเปลี่ยนจาก "1" เป็น "0" และพื้นที่ของเนื้อเทปในบริเวณนั้นก็จะ กลายเป็นเนื้อที่ว่าง พร้อมทั้งจะถูกบันทึกข้อมูลใหม่ในครั้งต่อๆมา ซึ่งในระหว่างที่ ยังไม่มีการบันทึกข้อความทับลงไป เราก็ยังสามารถขอฟังข้อความเดิมได้

นอกจากนี้รูปแบบการแสดงผลแบบที่ (2) ดังข้างต้นยังแสดงให้เห็น ทราบว่า ระบบพร้อมแล้วที่จะทำหน้าที่เป็นเครื่องตอบรับและบันทึกเสียง ในคู่ สายโทรศัพท์โดยอัตโนมัติ โดยจะเริ่มทำงานทันที ที่มีสัญญาณกระดิ่งเข้ามา เมื่อมาถึงตรงนี้ ก็เป็นอันว่าจบขั้นตอนการโปรแกรมสั่งงานเครื่อง ถ้าเรา ต้องการจะหยุดการทำงานนี้ก็ให้กดคีย์ใดๆ เครื่องก็จะกลับไปรอรับคำสั่งเพื่อ เข้าสู่การทำงานในฟังก์ชันต่าง ๆ ตามต้องการต่อไป

บทที่ 5

สรุปผลการดำเนินงาน

ในโครงการนี้ได้ทำการออกแบบให้มีประสิทธิภาพ มากกว่า เครื่องที่มีอยู่โดยทั่วไป กล่าวคือ ในระบบเดิมจะเป็นการบันทึกเสียงแบบ เรียงลำดับของการบันทึกก่อนหลัง และเมื่อต้องการฟังข้อความ ก็จะทำใน ลักษณะก่อนหลังเช่นกัน แต่ในบางครั้งพบว่า ข้อความบางอย่างเป็นความลับที่ ไม่ต้องการเปิดเผยต่อผู้อื่น ซึ่งเครื่องแบบเดิมนั้นจะไม่สามารถเก็บเป็น ความลับได้เลย จากปัญหานี้จึงได้สร้างเครื่องที่สามารถกำหนดว่าจะบันทึกข้อ ความถึงใคร โดยใช้รหัสตัวเลข 1 หลัก เป็นการระบุถึงบุคคลใด ๆ ซึ่ง บุคคลนั้น ๆ จะใช้รหัสตัวเลข 4 หลัก ในการฟังข้อความ

5.1 ผลของโครงการ

การควบคุมการทำงานของระบบในโครงการนี้ใช้ Micro-processor Z-80 เป็นตัวควบคุมการทำงาน ซึ่งจะต้องใช้ Software มากำหนดการทำงาน จึงได้เขียน Program ขึ้น และบันทึกไว้ใน RAM 2 ทาง คือสามารถบันทึกข้อมูลเข้าไปและเก็บไว้ได้ โดยมี Battery Backup ข้อมูลอยู่ และสามารถนำข้อมูลมากำหนดการทำงานของ Microprocessor Z-80 ซึ่งการใช้ RAM ชนิดนี้จะช่วยให้การทดลองโครงการ เป็นไปด้วย ความสะดวก เพราะสามารถจะเปลี่ยนแปลงแก้ไข Program ได้ง่ายกว่า การใช้ EPROM

ในการทดลองนั้น ผลการทำงานของโครงการ ในส่วนของ Function การทำงาน การสั่งงาน และการควบคุมต่าง ๆ เป็นไปตามที่กำหนดไว้ ซึ่งจะมีลักษณะการใช้งานและการทำงานดังที่กล่าวไว้ในบทที่ 4 ยก เว้นในส่วนของการบันทึกเสียงของเทปนั้น ยังมีปัญหาที่ไม่สามารถจะบันทึกซิงค์ และข้อความได้ จึงทำให้ผลของโครงการนี้ การทำงานยังไม่สมบูรณ์ในส่วน ของการบันทึกเสียงดังกล่าว แต่ในส่วนอื่น ๆ ก็มีผลเป็นที่น่าพอใจ ในการค่า

เนื้องานขั้นต่อไปนั้น เมื่อพิจารณาในส่วนของ Program แล้วเห็นว่า Program ทำงานได้ถูกต้อง สมบูรณ์ จึงได้นำ Program บันทึกลงเข้าไปใน EPROM เพื่อใช้งานจริง แต่เมื่อใช้ EPROM ปรากฏว่า Microprocessor ไม่สามารถทำงานได้ ซึ่งอาจจะมีปัญหาต่าง ๆ เกิดขึ้น จะได้อธิบายถึงปัญหาต่อไป

5.2 ปัญหาและการแก้ไข

จากการดำเนินงาน ได้มีปัญหาเกิดขึ้นในขณะที่ทำการทดลองวงจร และปัญหาในการทำงานของโครงการ ซึ่งสามารถแก้ไขปัญหาก็ได้บางส่วน และในบางปัญหาก็ยังไม่สามารถแก้ไขได้ทัน เนื่องจากเวลาจำกัด ปัญหาและการแก้ไขต่าง ๆ มีดังต่อไปนี้

1. การจัดสรรเนื้อที่ภายในเทป ในครั้งแรกใช้การจัดสรรเนื้อที่ภายในเทปแบบ 5 บล็อกใหญ่ ในแต่ละบล็อกใหญ่ประกอบด้วย 3 บล็อกย่อย ก็พบปัญหาว่า ถ้าหัวเทปอยู่ที่ตำแหน่งแรกของบล็อกที่ 1 เมื่อมีผู้โทรเข้ามาต้องการติดต่อกับบล็อกที่ 5 เขาจะต้องเสียเวลารอถึง 1.5 นาที ซึ่งเป็นเวลานานมากสำหรับการรอการติดต่อในโทรศัพท์ จึงแก้ปัญหาด้วยการเลื่อนหัวเทปไปที่ตำแหน่งของบล็อกที่ 3 เสมอ จะทำให้การรอในการค้นหาใช้เวลา นานที่สุดประมาณ 45 วินาที ซึ่งก็ยังไม่เป็นที่พอใจ จึงตัดสินใจใช้วิธีแบ่งบล็อก ออกเป็น 15 บล็อก ๆ ละ 2 นาทีเมื่อมีผู้โทรเข้ามาคนแรกจะทำการบันทึกที่ บล็อกที่ 1 ทันที แต่ CPU จะเป็นตัวจำไว้ว่าเป็นข้อความของใคร และจะไปรออยู่ที่บล็อกที่ 2 ทันที เมื่อมีผู้โทร เข้ามาติดต่ออีกก็สามารถบันทึกต่อได้ทันที แสดงการจัดสรรเนื้อที่ในเทปโดยไม่ต้องรอ แต่สำหรับผู้โทรเข้ามาขอฟังข้อความก็ต้องเสียเวลารอบ้างสำหรับการค้นหาข้อความ

2. ในเรื่องของการบินปีกเสียงของเทปนั้น ส่วนของสัญญาณซิงค์ ไม่สามารถบันทึกได้ สัญญาณซิงค์นี้ได้มาจาก Transformer ของภาคจ่ายไฟ เมื่อนำมาใช้ในการบันทึก ทำให้เกิดการรบกวนกันของภาคจ่ายไฟ เนื่องจากใช้ Ground ร่วมกัน ฉะนั้นในการแก้ปัญหาจึงควรใช้ Transformer ที่แยกชุดขดลวดต่างหาก เพื่อแยกระบบ Ground และในส่วน

ของวงจร Pre tape นั้น จะต้องมีประสิทธิภาพสูง มีสัญญาณรบกวนที่ต่ำ เพราะเป็นเครื่องของสัญญาณที่มีขนาดเล็ก ฉะนั้นในโครงการนี้ปัญหาของการบันทึก จึงอาจจะมีผลมาจากเครื่องของสัญญาณรบกวน และประสิทธิภาพของวงจร Pre tape

3. การใช้ EPROM เก็บ Program เพื่อให้ Microprocessor ทำงาน ไม่สามารถจะใช้ได้สำหรับโครงการนี้ ผู้จัดทำก็ได้พยายามค้นคว้าทดลอง และปัญหาด้วยวิธีต่าง ๆ แต่ก็ยังไม่สามารถสรุปผลได้ แต่ก็มีความเห็นว่าปัญหาอย่างหนึ่งอาจเป็นเพราะสัญญาณรบกวนของระบบ ไปมีผลกระทบต่อ การอ่านข้อมูลในตัว EPROM จึงทำให้ Program บางส่วนอาจผิดพลาดไป Microprocessor จึงไม่สามารถทำงานได้

2. สัญญาณรบกวนในสายโทรศัพท์ ทำให้การตรวจจับเสียงพูดทำงานผิดพลาด คือเมื่อผู้โทรเข้ามาฝากข้อความหยุดพูด และวางสายแล้ว วงจรตรวจจับเสียงพูดยังทำงานอยู่ คือตรวจจับได้สัญญาณรบกวนแทน จึงต้องแก้ปัญหาด้วยการสร้างวงจรกรองความถี่ต่ำผ่าน ซึ่งมี High Frequency Cut off 3.5 kHz ซึ่งครอบคลุมสัญญาณเสียงในสายโทรศัพท์ ก็สามารถแก้ปัญหาได้

3. จากการทดลองนำโครงการ ไปต่อผ่านระบบชุมสายอัตโนมัติขนาดเล็ก (PABX) พบว่าถ้าเป็นการโทรติดต่อกันภายในชุมสาย PABX สัญญาณต่างๆจะถูกสร้างจากตัว PABX เอง และจะมีการตรวจสอบการวางหูของ คู่สายภายในด้วย โดยถ้าปลายสายด้านหนึ่งวางหู คู่สนทนาจะได้ยินสัญญาณเตือน ซึ่งในกรณีนี้ทำให้วงจรตรวจสอบเสียงพูดทำงาน และส่งผลให้ CPU ไม่ทราบการหยุดพูด และจะไม่หยุดบันทึกแม้ว่าจะไม่มีเสียงพูดนานกว่า 10 วินาที แต่จะหยุดบันทึกเมื่อการบันทึกครบ 2 นาที ซึ่งในภายหลังได้เพิ่มเงื่อนไขเพื่อแก้ปัญหาโดยให้ผู้ใช้กดเครื่องหมาย "*" เมื่อเลิกบันทึกข้อความ ซึ่งปัญหาเช่นนี้จะไม่เกิดกับคู่สายโทรศัพท์ ที่ต่อโดยตรงกับชุมสายโทรศัพท์ขององค์การโทรศัพท์

4. การทำงานของวงจร DTMF Decoder ทำงานผิดพลาด เนื่องจากไม่ได้ใช้วงจร Hy-Bridge มาเป็นตัวแยกสัญญาณ เอาท์พุทและ อินพุทของระบบออกจากกัน ทำให้เกิดสัญญาณรบกวนเนื่องจาก สัญญาณเอาท์

พุ่มของวงจรมหาสัญญาณ ของเครื่องบันทึกเสียงดิจิตอล ลักษณะของความผิดพลาดที่เกิดขึ้น จะทำให้การตรวจสอบสัญญาณ DTMF ใช้ได้กับโทรศัพท์ไม่ทุกเครื่อง ซึ่งได้แก้ปัญหาโดยการเขียนโปรแกรม ให้ Relay ที่มีอยู่แต่เดิมแล้ว ตัดเอาพุ่มของวงจรมหาสัญญาณออกไป ทุกครั้งที่ต้องการตรวจสอบสัญญาณ DTMF

5.3 ข้อเสนอแนะ

ผู้จัดทำโครงการมีความหวังเป็นอย่างยิ่งว่า โครงการงานนี้จะ เป็นประโยชน์กับการติดต่อสื่อสารในปัจจุบัน สำหรับผู้ที่มีความประสงค์จะนำโครงการนี้มาพัฒนาให้ดียิ่งขึ้น ผู้จัดทำมีความยินดีเป็นอย่างยิ่ง และขอฝากแนวทางในการพัฒนาเครื่องบันทึกเสียงโทรศัพท์เครื่องนี้ไว้ดังนี้

1. ในส่วนที่โครงการนี้ยังไม่สามารถทำงานได้ สาเหตุหนึ่งที่ได้กล่าวไว้แล้วคือ อาจเนื่องมาจากสัญญาณรบกวน ซึ่งการออกแบบลายวงจรในโครงการนี้ ไม่ได้แยกระบบ Ground ของวงจรต่าง ๆ ออกจากกัน แต่ในวงจรแต่ละส่วนนั้นมีหน้าที่ที่แตกต่างกัน บางส่วนเป็นเรื่องของสัญญาณ Digital บางส่วนเป็นสัญญาณ Analog ในโครงการนี้ได้คำนึงถึงในขณะทำการออกแบบ จึงได้รวมวงจรต่าง ๆ ไว้ด้วยกัน และไม่มีการป้องกันการรบกวนกัน จึงอาจเป็นสาเหตุหนึ่ง

ผู้จัดทำจึงขอเสนอแนะในส่วนนี้ว่า ควรจะมีการแยกระบบ Ground ของชุดต่าง ๆ ออกจากกัน จัดวางอุปกรณ์ให้เป็นสัดส่วน ในขณะออกแบบลายวงจรพิมพ์ควรคำนึงถึงเรื่องของการรบกวนกัน และสัญญาณรบกวนจากภายนอกให้มาก และอาจจะเพิ่มระบบการป้องกันสัญญาณรบกวนเข้าไปด้วย นอกจากนี้ในเรื่องของการเดินสายสัญญาณ และส่วนของภาค Power Supply ควรพิจารณาด้วย

2. ควรปรับปรุงประสิทธิภาพ และขนาดของเครื่อง ให้ดีขึ้น มีขนาดเล็ก ใช้งานง่าย และอาจพัฒนาให้ใช้กับอุปกรณ์ในการบันทึกเสียงที่ต่ำกว่าเทปคาสเซ็ท เพื่อให้การทำงานในการค้นหาข้อความที่ต้องการได้เร็วขึ้น

3. สำหรับผู้ที่ต้องการใช้เทปคาสเซ็ทในการสร้างเครื่องนี้ ถ้า

หาเนื้อเทพและหัวเทพที่สามารถตอบสนองความถี่ได้ดีกว่านี้ ควรที่จะทดลอง
การนำมาเข้ารหัสเพื่อจะได้ใช้ในการป้องกันความลับได้ดียิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการปริญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์ จากท่านอาจารย์ที่ปรึกษาปริญานิพนธ์ คืออาจารย์ชวลิต เบญจางคประเสีฐ และคณาจารย์ประจำภาควิชาเทคนิคอุตสาหกรรมหลายท่าน ที่ให้ความอนุเคราะห์ในด้านคำปรึกษา เครื่องมือ อุปกรณ์ และสถานที่เป็นอย่างดี ซึ่งเป็นประโยชน์ต่อการจัดทำในครั้งนี้ คณะผู้จัดทำใคร่ขอขอบพระคุณทุกท่านอย่างสูงมา ณ โอกาสนี้

นายวันชัย ก้อนทอง
นายสมิทธิ์ สุขขี
นายอุดมศักดิ์ ชัยเพชร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- พันธ์ศักดิ์ ศรีกรรพย์. เครื่องบันทึกเทป. กรุงเทพมหานคร : บริษัทศูนย์
การพิมพ์ดวงกมล จำกัด, 2530.
- วิบูลย์ ชื่นแขก. ไมโครโปรเซสเซอร์. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร :
สำนักพิมพ์สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ, 2532.
- ประเสริฐ วรรณสุธีวัฒน์. "เครื่องบันทึกเสียงพูดระบบดิจิทัล". วารสารเซ
มิคอนดักเตอร์ อิเลคทรอนิกส์ ฉบับที่ 84 (กุมภาพันธ์ 2531):
144-149.
- สมยศ โลหะวิทยวิกรานต์, บรรณาธิการ. "MT 8870 ไอซีถอดรหัสความถี่
โทรศัพท์". วารสารเซมิคอนดักเตอร์ อิเลคทรอนิกส์ ฉบับที่ 88
(กันยายน 2531): 210-213
- สมยศ โลหะวิทยวิกรานต์, บรรณาธิการ. "NJM2072 ไอซีสวิตช์เสียง".
วารสารเซมิคอนดักเตอร์ อิเลคทรอนิกส์ ฉบับที่ 89 (ธันวาคม
2533): 209-211.
- พิชวัฒน์ ลิ้มพรจิตรวิไล. "ปริทัศน์เล่น-บันทึก". วารสารเซมิคอนดักเตอร์
อิเลคทรอนิกส์ ฉบับที่ 103 (ธันวาคม 2533) : 238-246.
- JOHN L. HILBURN, AND DAVID E. JOHNSON. Manual of
Active filter design. LOUISIANA: Magraw-Hill
Inc, 1973
- RODNAY ZAKS, AND AUSTIN LESEA. Microprocessor Interface
Techniques. Third Edition. United State of
America: Sybex Inc.
- RODNEY ZAKS. Programing The Z80. Third Edition. United
State of America: Sybex Inc, 1980



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80-CPU ™ Z80A-CPU



Product Specification

The Zilog Z80 product line is a complete set of micro-computer components, development systems and support software. The Z80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

The Z80 and Z80A CPU's are third generation single chip microprocessors with unrivaled computational power. This increased computational power results in higher system through-put and more efficient memory utilization when compared to second generation microprocessors. In addition, the Z80 and Z80A CPU's are very easy to implement into a system because of their single voltage requirement plus all output signals are fully decoded and timed to control standard memory or peripheral circuits. The circuit is implemented using an N-channel, ion implanted, silicon gate MOS process.

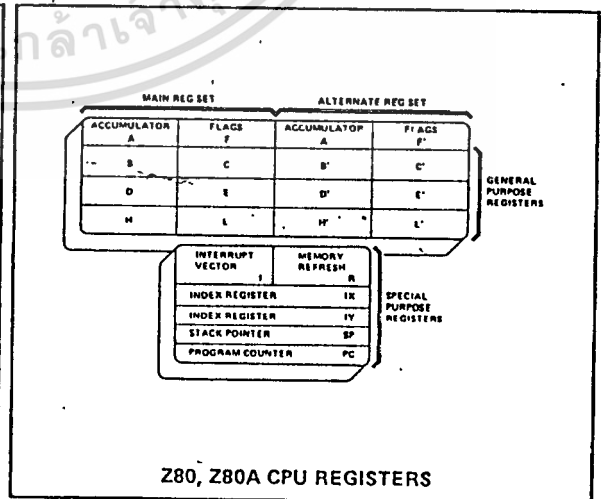
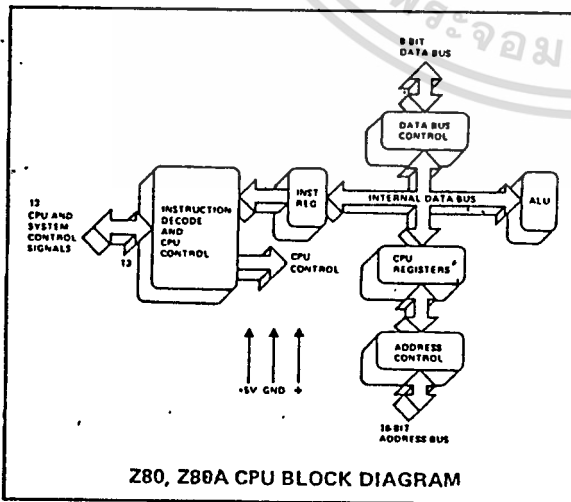
Figure 1 is a block diagram of the CPU, Figure 2 details the internal register configuration which contains 208 bits of Read/Write memory that are accessible to the programmer. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or as 16-bit register pairs. There are also two sets of accumulator and flag registers. The programmer has access to either set of main or alternate registers through a group of exchange instructions. This alternate set allows foreground/background mode of operation or may be reserved for very fast Interrupt response. Each CPU also contains a 16-bit stack pointer which permits simple implementation of

multiple level interrupts, unlimited subroutine nesting and simplification of many types of data handling.

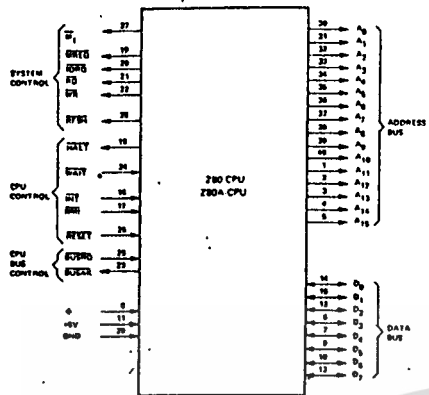
The two 16-bit index registers allow tabular data manipulation and easy implementation of relocatable code. The Refresh register provides for automatic, totally transparent refresh of external dynamic memories. The I register is used in a powerful interrupt response mode to form the upper 8 bits of a pointer to an interrupt service address table, while the interrupting device supplies the lower 8 bits of the pointer. An indirect call is then made to this service address.

FEATURES

- Single chip, N-channel Silicon Gate CPU.
- 158 instructions—includes all 78 of the 8080A instructions with total software compatibility. New instructions include 4-, 8- and 16-bit operations with more useful addressing modes such as indexed, bit and relative.
- 17 internal registers.
- Three modes of fast interrupt response plus a non-maskable interrupt.
- Directly interfaces standard speed static or dynamic memories with virtually no external logic.
- 1.0 μ s instruction execution speed.
- Single 5 VDC supply and single-phase 5 volt Clock.
- Out-performs any other single chip microcomputer in 4-, 8-, or 16-bit applications.
- All pins TTL Compatible
- Built-in dynamic RAM refresh circuitry.



Z80, Z80A-CPU Pin Description



Z80, Z80A CPU PIN CONFIGURATION

A₀-A₁₅
(Address Bus)

Tri-state output, active high. A₀-A₁₅ constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges.

D₀-D₇
(Data Bus)

Tri-state input/output, active high. D₀-D₇ constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

M₁
(Machine Cycle one)

Output, active low. M₁ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution.

MREQ
(Memory Request)

Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

IORQ
(Input/Output Request)

Tri-state output, active low. The IORQ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An IORQ signal is also generated when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus.

RD
(Memory Read)

Tri-state output, active low. RD indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

WR
(Memory Write)

Tri-state output, active low. WR indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.

RFSH
(Refresh)

Output, active low. RFSH indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the current MREQ signal should be used to do a refresh read to all dynamic memories.

HALT
(Halt state)

Output, active low. HALT indicates that the CPU has executed a HALT software instruction and is awaiting either a non-maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.

WAIT
(Wait)

Input, active low. WAIT indicates to the Z-80 CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active.

INT
(Interrupt Request)

Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled.

NMI
(Non Maskable Interrupt)

Input, active low. The non-maskable interrupt request line has a higher priority than INT and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. NMI automatically forces the Z-80 CPU to restart to location 0066H.

RESET

Input, active low. RESET initializes the CPU as follows: reset interrupt enable flip-flop, clear PC and registers I and R and set interrupt to 8080A mode. During reset time, the address and data bus go to a high impedance state and all control output signals go to the inactive state.

BUSRQ
(Bus Request)

Input, active low. The bus request signal has a higher priority than NMI and is always recognized at the end of the current machine cycle and is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these busses.

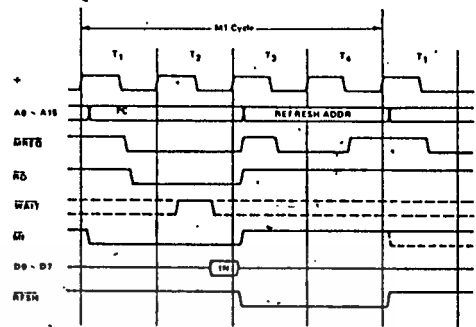
BUSAK
(Bus Acknowledge)

Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

Timing Waveforms

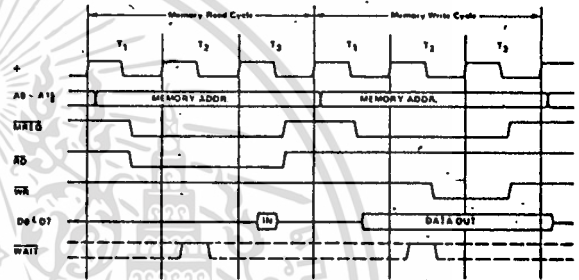
INSTRUCTION OP CODE FETCH

The program counter content (PC) is placed on the address bus immediately at the start of the cycle. One half clock time later MREQ goes active. The falling edge of MREQ can be used directly as a chip enable to dynamic memories. RD when active indicates that the memory data should be enabled onto the CPU data bus. The CPU samples data with the rising edge of the clock state T₃. Clock states T₃ and T₄ of a fetch cycle are used to refresh dynamic memories while the CPU is internally decoding and executing the instruction. The refresh control signal RFSH indicates that a refresh read of all dynamic memories should be accomplished.



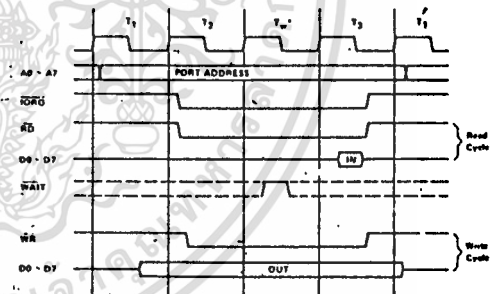
MEMORY READ OR WRITE CYCLES

Illustrated here is the timing of memory read or write cycles other than an OP code fetch (M₁ cycle). The MREQ and RD signals are used exactly as in the fetch cycle. In the case of a memory write cycle, the MREQ also becomes active when the address bus is stable so that it can be used directly as a chip enable for dynamic memories. The WR line is active when data on the data bus is stable so that it can be used directly as a R/W pulse to virtually any type of semiconductor memory.



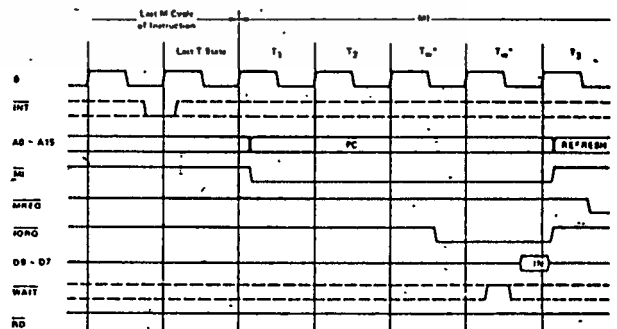
INPUT OR OUTPUT CYCLES

Illustrated here is the timing for an I/O read or I/O write operation. Notice that during I/O operations a single wait state is automatically inserted (Tw*): The reason for this is that during I/O operations this extra state allows sufficient time for an I/O port to decode its address and activate the WAIT line if a wait is required.



INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

The interrupt signal is sampled by the CPU with the rising edge of the last clock at the end of any instruction. When an interrupt is accepted, a special M₁ cycle is generated. During this M₁ cycle, the IORQ signal becomes active (instead of MREQ) to indicate that the interrupting device can place an 8-bit vector on the data bus. Two wait states (Tw*) are automatically added to this cycle so that a ripple priority interrupt scheme, such as the one used in the Z80 peripheral controllers, can be easily implemented.



Z80, Z80A Instruction Set

The following is a summary of the Z80, Z80A instruction set showing the assembly language mnemonic and the symbolic operation performed by the instruction. A more detailed listing appears in the Z80-CPU technical manual, and assembly language programming manual. The instructions are divided into the following categories:

- | | |
|---|-------------------------|
| 8-bit loads | Miscellaneous Group |
| 16-bit loads | Rotates and Shifts |
| Exchanges | Bit Set, Reset and Test |
| Memory Block Moves | Input and Output |
| Memory Block Searches | Jumps |
| 8-bit arithmetic and logic | Calls |
| 16-bit arithmetic | Restarts |
| General purpose Accumulator & Flag Operations | Returns |

In the table the following terminology is used.

- b ≡ a bit number in any 8-bit register or memory location
- cc ≡ flag condition code
- NZ ≡ non zero
- Z ≡ zero
- NC ≡ non carry
- C ≡ carry
- PO ≡ Parity odd or no over flow
- PE ≡ Parity even or over flow
- P ≡ Positive
- M ≡ Negative (minus)

- d ≡ any 8-bit destination register or memory location
- dd ≡ any 16-bit destination register or memory location
- e ≡ 8-bit signed 2's complement displacement used in relative jumps and indexed addressing
- L ≡ 8 special call locations in page zero. In decimal notation these are 0, 8, 16, 24, 32, 40, 48 and 56
- n ≡ any 8-bit binary number
- nn ≡ any 16-bit binary number
- r ≡ any 8-bit general purpose register (A, B, C, D, E, H, or L)
- s ≡ any 8-bit source register or memory location
- sb ≡ a bit in a specific 8-bit register or memory location
- ss ≡ any 16-bit source register or memory location
- subscript "L" ≡ the low order 8 bits of a 16-bit register
- subscript "H" ≡ the high order 8 bits of a 16-bit register
- () ≡ the contents within the () are to be used as a pointer to a memory location or I/O port number
- 8-bit registers are A, B, C, D, E, H, L, I and R
- 16-bit register pairs are AF, BC, DE and HL
- 16-bit registers are SP, PC, IX and IY

Addressing Modes implemented include combinations of the following:

Immediate	Indexed
Immediate extended	Register
Modified Page Zero	Implied
Relative	Register Indirect
Extended	Bit

Mnemonic	Symbolic Operation	Comments
LD r, s	r ← s	s ≡ r, n, (HL), (IX+e), (IY+e)
LD d, r	d ← r	d ≡ (HL), r (IX+e), (IY+e)
LD d, n	d ← n	d ≡ (HL), (IX+e), (IY+e)
LD A, s	A ← s	s ≡ (BC), (DE), (nn), I, R
LD d, A	d ← A	d ≡ (BC), (DE), (nn), I, R
LD dd, nn	dd ← nn	dd ≡ BC, DE, HL, SP, IX, IY
LD dd, (nn)	dd ← (nn)	dd ≡ BC, DE, HL, SP, IX, IY
LD (nn), ss	(nn) ← ss	ss ≡ BC, DE, HL, SP, IX, IY
LD SP, ss	SP ← ss	ss = HL, IX, IY
PUSH ss	(SP-1) ← ss _H ; (SP-2) ← ss _L	ss = BC, DE, HL, AF, IX, IY
POP dd	dd _L ← (SP); dd _H ← (SP+1)	dd = BC, DE, HL, AF, IX, IY
EX DE, HL EX AF, AF' EXX	DE ↔ HL AF ↔ AF' $\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	
EX (SP), ss	(SP) ↔ ss _L ; (SP+1) ↔ ss _H	ss ≡ HL, IX, IY

Mnemonic	Symbolic Operation	Comments
LDI	(DE) ← (HL), DE ← DE+1 HL ← HL+1, BC ← BC-1	
LDIR	(DE) ← (HL), DE ← DE+1 HL ← HL+1, BC ← BC-1 Repeat until BC = 0	
LDD	(DE) ← (HL), DE ← DE-1 HL ← HL-1, BC ← BC-1	
LDDR	(DE) ← (HL), DE ← DE-1 HL ← HL-1, BC ← BC-1 Repeat until BC = 0	
CPI	A-(HL), HL ← HL+1 BC ← BC-1	
CPIR	A-(HL), HL ← HL+1 BC ← BC-1, Repeat until BC = 0 or A = (HL)	A-(HL) sets the flags only. A is not affected
CPD	A-(HL), HL ← HL-1 BC ← BC-1	
CPDR	A-(HL), HL ← HL-1 BC ← BC-1, Repeat until BC = 0 or A = (HL)	
ADD s	A ← A + s	
ADC s	A ← A + s + CY	CY is the carry flag
SUB s	A ← A - s	
SBC s	A ← A - s - CY	s ≡ r, n, (HL) (IX+e), (IY+e)
AND s	A ← A ∧ s	
OR s	A ← A ∨ s	
XOR s	A ← A ⊕ s	

Mnemonic	Symbolic Operation	Comments
8-BIT ALU CP s INC d DEC d	A ← s d ← d + 1 d ← d - 1	s = r, n (HL) (IX+e), (IY+e) d = r, (HL) (IX+e), (IY+e)
16-BIT ARITHMETIC ADD HL, ss ADC HL, ss SBC HL, ss ADD IX, ss ADD IY, ss INC dd DEC dd	HL ← HL + ss HL ← HL + ss + CY HL ← HL - ss - CY IX ← IX + ss IY ← IY + ss dd ← dd + 1 dd ← dd - 1	ss ≡ BC, DE HL, SP ss ≡ BC, DE, IX, SP ss ≡ BC, DE, IY, SP dd ≡ BC, DE, HL, SP, IX, IY dd ≡ BC, DE, HL, SP, IX, IY
CP ACC. & FLAG DAA CPL NEG CCF SCF	Converts A contents into packed BCD following add or subtract. A ← \bar{A} A ← 00 - A CY ← \bar{CY} CY ← 1	Operands must be in packed BCD format
MISCELLANEOUS NOP HALT DI EI IM 0 IM 1 IM 2	No operation Halt CPU Disable Interrupts Enable Interrupts Set interrupt mode 0 Set interrupt mode 1 Set interrupt mode 2	8080A mode Call to 0038H Indirect Call
ROTATES AND SHIFTS RLC s RL s RRC s RR s SLA s SRA s SRL s RLD RRD	 	s ≡ r, (HL) (IX+e), (IY+e)

Mnemonic	Symbolic Operation	Comments
BIT S. R. & T BIT b, s SET b, s RES b, s	Z ← \bar{s}_b $s_b \leftarrow 1$ $s_b \leftarrow 0$	Z is zero flag s ≡ r, (HL) (IX+e), (IY+e)
INPUT AND OUTPUT IN A, (n) IN r, (C) INI INIR IND INDR OUT(n), A OUT(C), r OUTI OTIR OUTD OTDR	A ← (n) r ← (C) (HL) ← (C), HL ← HL + 1 B ← B - 1 (HL) ← (C), HL ← HL + 1 B ← B - 1 (HL) ← (C), HL ← HL - 1 B ← B - 1 (HL) ← (C), HL ← HL - 1 B ← B - 1 (n) ← A (C) ← r (C) ← (HL), HL ← HL + 1 B ← B - 1 (C) ← (HL), HL ← HL + 1 B ← B - 1 (C) ← (HL), HL ← HL - 1 B ← B - 1 (C) ← (HL), HL ← HL - 1 B ← B - 1	Set flags Repeat until B = 0 Repeat until B = 0 Repeat until B = 0
JUMPS JP nn JP cc, nn JR e JR kk, e JP (ss) DJNZ e	PC ← nn If condition cc is true PC ← nn, else continue PC ← PC + e If condition kk is true PC ← PC + e, else continue PC ← ss B ← B - 1, if B = 0 continue, else PC ← PC + e	cc { NZ PO Z PE NC P C M kk { NZ NC Z C ss = HL, IX, IY
CALLS CALL nn CALL cc, nn	(SP-1) ← PC _H (SP-2) ← PC _L , PC ← nn If condition cc is false continue, else same as CALL nn	cc { NZ PO Z PE NC P C M
RESTARTS RST L	(SP-1) ← PC _H (SP-2) ← PC _L , PC _H ← 0 PC _L ← L	
RETURNS RET RET cc RETI RETN	PC _L ← (SP), PC _H ← (SP+1) If condition cc is false continue, else same as RET Return from interrupt, same as RET Return from non- maskable interrupt	cc { NZ PO Z PE NC P C M

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. Characteristics Z80-CPU

T_A = 0°C to 70°C, V_{CC} = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
φ	t _c	Clock Period	4	1121	μsec	
	t _{w(φH)}	Clock Pulse Width, Clock High	180	[E]	nsec	
	t _{w(φL)}	Clock Pulse Width, Clock Low	180	2000	nsec	
	t _{r, f}	Clock Rise and Fall Time		30	nsec	
A ₀₋₁₅	t _{D(AD)}	Address Output Delay		145	nsec	C _L = 50pF
	t _{F(AD)}	Delay to Float		110	nsec	
	t _{acm}	Address Stable Prior to MREQ (Memory Cycle)	[1]		nsec	
	t _{aci}	Address Stable Prior to IORQ, RD or WR (I/O Cycle)	[2]		nsec	
	t _{ca}	Address Stable from RD, WR, IORQ or MREQ	[3]		nsec	
D ₀₋₇	t _{D(D)}	Data Output Delay		230	nsec	C _L = 50pF
	t _{F(D)}	Delay to Float During Write Cycle		90	nsec	
	t _{Sφ(D)}	Data Setup Time to Rising Edge of Clock During M1 Cycle	50		nsec	
	t _{Sφ(D)}	Data Setup Time to Falling Edge of Clock During M2 to M5	60		nsec	
	t _{dcm}	Data Stable Prior to WR (Memory Cycle)	[5]		nsec	
	t _{dcl}	Data Stable Prior to WR (I/O Cycle)	[6]		nsec	
	t _{cdf}	Data Stable From WR	[7]		nsec	
	t _H	Any Hold Time for Setup Time	0		nsec	
MREQ	t _{DLφ(MR)}	MREQ Delay From Falling Edge of Clock, MREQ Low		100	nsec	C _L = 50pF
	t _{DHφ(MR)}	MREQ Delay From Rising Edge of Clock, MREQ High		100	nsec	
	t _{DHφ(MR)}	MREQ Delay From Falling Edge of Clock, MREQ High		100	nsec	
	t _{w(MRL)}	Pulse Width, MREQ Low	[8]		nsec	
	t _{w(MRH)}	Pulse Width, MREQ High	[9]		nsec	
IORQ	t _{DLφ(IR)}	IORQ Delay From Rising Edge of Clock, IORQ Low		90	nsec	C _L = 50pF
	t _{DHφ(IR)}	IORQ Delay From Falling Edge of Clock, IORQ Low		110	nsec	
	t _{DHφ(IR)}	IORQ Delay From Rising Edge of Clock, IORQ High		100	nsec	
	t _{DHφ(IR)}	IORQ Delay From Falling Edge of Clock, IORQ High		110	nsec	
RD	t _{DLφ(RD)}	RD Delay From Rising Edge of Clock, RD Low		100	nsec	C _L = 50pF
	t _{DHφ(RD)}	RD Delay From Falling Edge of Clock, RD Low		130	nsec	
	t _{DHφ(RD)}	RD Delay From Rising Edge of Clock, RD High		100	nsec	
	t _{DHφ(RD)}	RD Delay From Falling Edge of Clock, RD High		110	nsec	
WR	t _{DLφ(WR)}	WR Delay From Rising Edge of Clock, WR Low		80	nsec	C _L = 50pF
	t _{DHφ(WR)}	WR Delay From Falling Edge of Clock, WR Low		90	nsec	
	t _{DHφ(WR)}	WR Delay From Rising Edge of Clock, WR High		100	nsec	
	t _{w(WRL)}	Pulse Width, WR Low	[10]		nsec	
M1	t _{DL(M1)}	M1 Delay From Rising Edge of Clock, M1 Low		130	nsec	C _L = 50pF
	t _{DH(M1)}	M1 Delay From Rising Edge of Clock, M1 High		130	nsec	
RFSH	t _{DL(RF)}	RFSH Delay From Rising Edge of Clock, RFSH Low		180	nsec	C _L = 50pF
	t _{DH(RF)}	RFSH Delay From Rising Edge of Clock, RFSH High		150	nsec	
WAIT	t _{s(WT)}	WAIT Setup Time to Falling Edge of Clock	70		nsec	
HALT	t _{D(HT)}	HALT Delay Time From Falling Edge of Clock		300	nsec	C _L = 50pF
INT	t _{s(IT)}	INT Setup Time to Rising Edge of Clock	80		nsec	
NMI	t _{w(NML)}	Pulse Width, NMI Low	80		nsec	
BUSRQ	t _{s(BQ)}	BUSRQ Setup Time to Rising Edge of Clock	80		nsec	
BUSAK	t _{DL(BA)}	BUSAK Delay From Rising Edge of Clock, BUSAK Low		120	nsec	C _L = 50pF
	t _{DH(BA)}	BUSAK Delay From Falling Edge of Clock, BUSAK High		110	nsec	
RESET	t _{s(RS)}	RESET Setup Time to Rising Edge of Clock	90		nsec	
	t _{F(C)}	Delay to Float (MREQ, IORQ, RD and WR)		100	nsec	
	t _{mr}	M1 Stable Prior to IORQ (Interrupt Ack.)	[11]		nsec	

[12] t_c = t_{w(φH)} + t_{w(φL)} + t_r + t_f

[1] t_{acm} = t_{w(φH)} + t_r - 75

[2] t_{aci} = t_c - 80

[3] t_{ca} = t_{w(φL)} + t_r - 40

[4] t_{cai} = t_{w(φL)} + t_r - 60

[5] t_{dcm} = t_c - 210

[6] t_{dcl} = t_{w(φL)} + t_r - 210

[7] t_{cdf} = t_{w(φL)} + t_r - 80

[8] t_{w(MRL)} = t_c - 40

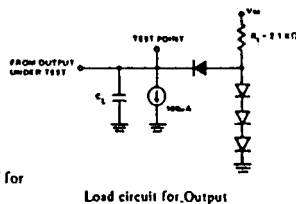
[9] t_{w(MRH)} = t_{w(φH)} + t_r - 30

[10] t_{w(WRL)} = t_c - 40

[11] t_{mr} = 2t_c + t_{w(φH)} + t_r - 80

NOTES:

- A. Data should be enabled onto the CPU data bus when \overline{RD} is active. During interrupt acknowledge data should be enabled when $\overline{M1}$ and \overline{IORQ} are both active.
- B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
- C. The \overline{RESET} signal must be active for a minimum of 3 clock cycles.
- D. Output Delay vs. Loaded Capacitance
 T_A = 70°C, V_{CC} = +5V ± 5%
 Add 10nsec delay for each 50pf increase in load up to a maximum of 200pf for the data bus & 100pf for address & control lines
- E. Although static by design, testing guarantees t_{w(φH)} of 200 μsec maximum



Load circuit for Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Temperature Under Bias	Specified operating range.
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	1.5W

***Comment**

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note: For Z80-CPU all AC and DC characteristics remain the same for the military grade parts except I_{CC} .

$$I_{CC} = 200 \text{ mA}$$

Z80-CPU D.C. Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5\text{V} \pm 5\%$ unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	$V_{CC} - 0.6$		$V_{CC} + 0.3$	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 1.8 \text{ mA}$
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -250 \mu\text{A}$
I_{CC}	Power Supply Current			150	mA	
I_{LI}	Input Leakage Current			10	μA	$V_{IN} = 0$ to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	$V_{OUT} = 2.4$ to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	$V_{OUT} = 0.4\text{V}$
I_{LD}	Data Bus Leakage Current in Input Mode			± 10	μA	$0 < V_{IN} < V_{CC}$

Capacitance

$T_A = 25^\circ\text{C}$, $f = 1 \text{ MHz}$,
-unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80-CPU

Ordering Information

- C - Ceramic
- P - Plastic
- S - Standard 5V $\pm 5\%$ 0° to 70°C
- E - Extended 5V $\pm 5\%$ -40° to 85°C
- M - Military 5V $\pm 10\%$ -55° to 125°C

Z80A-CPU D.C. Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5\text{V} \pm 5\%$ unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	$V_{CC} - 0.6$		$V_{CC} + 0.3$	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 1.8 \text{ mA}$
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -250 \mu\text{A}$
I_{CC}	Power Supply Current		90	200	mA	
I_{LI}	Input Leakage Current			10	μA	$V_{IN} = 0$ to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	$V_{OUT} = 2.4$ to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	$V_{OUT} = 0.4\text{V}$
I_{LD}	Data Bus Leakage Current in Input Mode			± 10	μA	$0 < V_{IN} < V_{CC}$

Capacitance

$T_A = 25^\circ\text{C}$, $f = 1 \text{ MHz}$,
-unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80A-CPU

Ordering Information

- C - Ceramic
- P - Plastic
- S - Standard 5V $\pm 5\%$ 0° to 70°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. Characteristics

Z80A-CPU

T_A = 0°C to 70°C, V_{CC} = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
φ	t _c	Clock Period	.25	1121	μsec	
	t _{w(φH)}	Clock Pulse Width, Clock High	110	11	nsec	
	t _{w(φL)}	Clock Pulse Width, Clock Low	110	2000	nsec	
	t _{r,f}	Clock Rise and Fall Time		30	nsec	
A ₀₋₁₅	t _{D(AD)}	Address Output Delay		110	nsec	C _L = 50pF
	t _{F(AD)}	Delay to Float		90	nsec	
	t _{acm}	Address Stable Prior to MREQ (Memory Cycle)	111		nsec	
	t _{ad}	Address Stable Prior to IORQ, RD or WR (I/O Cycle)	121		nsec	
	t _{ca}	Address Stable From RD, WR, IORQ or MREQ	131		nsec	
D ₀₋₇	t _{D(D)}	Data Output Delay		150	nsec	C _L = 50pF
	t _{F(D)}	Delay to Float During Write Cycle		90	nsec	
	t _{Sφ(D)}	Data Setup Time to Rising Edge of Clock During M1 Cycle	35		nsec	
	t _{Sφ(D)}	Data Setup Time to Falling Edge of Clock During M2 to M5	50		nsec	
	t _{dcm}	Data Stable Prior to WR (Memory Cycle)	131		nsec	
	t _{dcl}	Data Stable Prior to IORQ (I/O Cycle)	161		nsec	
	t _{cdf}	Data Stable From WR	171		nsec	
t _H	Any Hold Time for Setup Time		0	nsec		
MREQ	t _{DLφ(MR)}	MREQ Delay From Falling Edge of Clock, MREQ Low		85	nsec	C _L = 50pF
	t _{DHφ(MR)}	MREQ Delay From Rising Edge of Clock, MREQ High		85	nsec	
	t _{DHφ(MR)}	MREQ Delay From Falling Edge of Clock, MREQ High		85	nsec	
	t _{w(MRL)}	Pulse Width, MREQ Low	181		nsec	
	t _{w(MRH)}	Pulse Width, MREQ High	191		nsec	
IORQ	t _{DLφ(IR)}	IORQ Delay From Rising Edge of Clock, IORQ Low		75	nsec	C _L = 50pF
	t _{DLφ(IR)}	IORQ Delay From Falling Edge of Clock, IORQ Low		85	nsec	
	t _{DHφ(IR)}	IORQ Delay From Rising Edge of Clock, IORQ High		85	nsec	
	t _{DHφ(IR)}	IORQ Delay From Falling Edge of Clock, IORQ High		85	nsec	
	t _{w(MRL)}	Pulse Width, IORQ Low		85	nsec	
RD	t _{DLφ(RD)}	RD Delay From Rising Edge of Clock, RD Low		85	nsec	C _L = 50pF
	t _{DLφ(RD)}	RD Delay From Falling Edge of Clock, RD Low		95	nsec	
	t _{DHφ(RD)}	RD Delay From Rising Edge of Clock, RD High		85	nsec	
	t _{DHφ(RD)}	RD Delay From Falling Edge of Clock, RD High		85	nsec	
	t _{w(WRL)}	Pulse Width, WR Low	1101		nsec	
WR	t _{DLφ(WR)}	WR Delay From Rising Edge of Clock, WR Low		65	nsec	C _L = 50pF
	t _{DLφ(WR)}	WR Delay From Falling Edge of Clock, WR Low		80	nsec	
	t _{DHφ(WR)}	WR Delay From Rising Edge of Clock, WR High		80	nsec	
	t _{w(WRL)}	Pulse Width, WR Low		80	nsec	
M1	t _{DL(M1)}	M1 Delay From Rising Edge of Clock, M1 Low		100	nsec	C _L = 50pF
	t _{DH(M1)}	M1 Delay From Rising Edge of Clock, M1 High		100	nsec	
RFSH	t _{DL(RF)}	RFSH Delay From Rising Edge of Clock, RFSH Low		130	nsec	C _L = 50pF
	t _{DH(RF)}	RFSH Delay From Rising Edge of Clock, RFSH High		120	nsec	
WAIT	t _{s(WT)}	WAIT Setup Time to Falling Edge of Clock	70		nsec	
HALT	t _{D(HT)}	HALT Delay Time From Falling Edge of Clock		300	nsec	C _L = 50pF
INT	t _{s(IT)}	INT Setup Time to Rising Edge of Clock	80		nsec	
NMI	t _{w(NML)}	Pulse Width, NMI Low	80		nsec	
BUSRQ	t _{s(BQ)}	BUSRQ Setup Time to Rising Edge of Clock	50		nsec	
BUSAK	t _{DL(BA)}	BUSAK Delay From Rising Edge of Clock, BUSAK Low		100	nsec	C _L = 50pF
	t _{DH(BA)}	BUSAK Delay From Falling Edge of Clock, BUSAK High		100	nsec	
RESET	t _{s(RS)}	RESET Setup Time to Rising Edge of Clock	60		nsec	
	t _{F(C)}	Delay to Float (MREQ, IORQ, RD and WR)		80	nsec	
	t _{mi}	M1 Stable Prior to IORQ (Interrupt Ack.)	1111		nsec	

[12] t_c = t_{w(φH)} + t_{w(φL)} + t_r + t_f

[1] t_{acm} = t_{w(φH)} + t_r - 65

[2] t_{dcl} = t_c - 70

[3] t_{ca} = t_{w(φL)} + t_r - 50

[4] t_{caf} = t_{w(φL)} + t_r - 45

[5] t_{dcm} = t_c - 170

[6] t_{dcl} = t_{w(φL)} + t_r - 170

[7] t_{cdf} = t_{w(φL)} + t_r - 70

[8] t_{w(MRL)} = t_c - 30

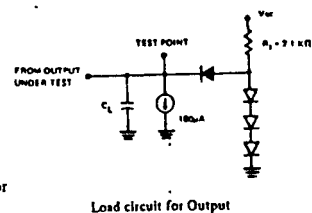
[9] t_{w(MRH)} = t_{w(φH)} + t_r - 20

[10] t_{w(WRL)} = t_c - 30

[11] t_{mi} = 2t_c + t_{w(φH)} + t_r - 65

NOTES:

- A. Data should be enabled onto the CPU data bus when RD is active. During interrupt acknowledge data should be enabled when M1 and IORQ are both active.
- B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
- C. The RESET signal must be active for a minimum of 3 clock cycles.
- D. Output Delay vs. Loaded Capacitance
 T_A = 70°C V_{CC} = +5V ± 5%
 Add 10nsec delay for each 50pf increase in load up to maximum of 200pf for data bus and 100pf for address & control lines.
- E. Although static by design, testing guarantees t_{w(φH)} of 200 μsec maximum



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCD HDM 20216H INSTRUCTION TABLE

Instruction	Code										Description	Execution time (when fosc is 250 kHz) Note 1	Execution time (when fosc is 160 kHz) Note 2	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0				
Clear display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0).	82 μ s ~ 1.64 ms	120 μ s ~ 4.8 ms	
Return home	0	0	0	0	0	0	0	0	1	•	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	40 μ s ~ 1.6 ms	120 μ s ~ 4.8 ms	
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40 μ s	120 μ s	
Display ON/OFF control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D), cursor ON/OFF (C), and blink of cursor position/character (B).	40 μ s	120 μ s	
Cursor and display shift	0	0	0	0	0	1	S/C	R/L	•	•	Moves the cursor and shifts the display without changing DD RAM contents	40 μ s	120 μ s	
Function set	0	0	0	0	1	DL	N	F	•	•	Sets interface data length (DL) number of display lines (L) and character font (F).	40 μ s	120 μ s	
Set CG RAM address	0	0	0	1	ACG					•	•	Sets the CG RAM address. CG RAM data is sent and received after this setting.	40 μ s	120 μ s
Set DD RAM address	0	0	1	ADD					•	•	Sets the DD RAM address. DD RAM data is sent and received after this setting.	40 μ s	120 μ s	
Read busy flag & address	0	1	BF	AC					•	•	Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.	1 μ s	1 μ s	
Write data to CG or DD RAM	1	0	Write Data					•	•	Writes data into DD RAM or CG RAM.	40 μ s	120 μ s		
Read data to CG or DD RAM	1	1	Read Data					•	•	Reads data from DD RAM or CG RAM.	40 μ s	120 μ s		
	I/D = 1: Increment (+1) I/D = 0: Decrement (-1) S = 1: Accompanies display shift. S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right. R/L = 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5 x 10 dots F = 0: 5 x 7 dots BF = 1: Internally operating BF = 0: Can accept instruction										DD RAM: Display data RAM CG RAM: Character generator RAM ACG: CG RAM address ADD: DD RAM address Corresponds to cursor address. AC: Address counter used for both of DD and CG RAM address.	Execution time changes when frequency changes. (Example) When fosc is 270 kHz: $40 \mu\text{s} \times \frac{250}{270} = 37 \mu\text{s}$		

• No effect

Notes 1. Applied to models driven by 1/8 duty or 1/11 duty.

2. Applied to models driven by 1/16 duty.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAXIMUM RATINGS

Electric maximum ratings

Item	Symbol	Min.	Max.	Unit	Remarks
Power supply for logic	$V_{DD} - V_{SS}$			V	
Power supply for LCD drive	$V_{DD} - V_0$	Refer to individual specifications		V	
Input voltage	V_i			V	
Static electricity			100	V	See note

Note Electro-static discharge resistance is tested by charging a condenser with a capacity of 200pF and discharging it by contact with an interface connector pin.

Environmental conditions

Item	Operating		Non-operating		Remarks
	Min.	Max.	Min.	Max.	
Ambient temperature	Refer to individual specifications				
Humidity	Note				No dew
Vibration		4.9 m/s ² (0.5G)		19.6 m/s ² (2G)	
Shock		29.4 m/s ² (3G)		490 m/s ² (50G)	XYZ 3 directions
Corrosion gas	No corrosion gas				

Note Humidity conditions are as follows.

Number of dots	Under 128 × 240		128 × 240 or over	
	Ambient temperature (Ta)	95% RH max.		85% RH max.
Ta ≤ 40°C	Below maximum absolute humidity of 40°C 95% RH		Below maximum absolute humidity of 40°C 85% RH	
Ta > 40°C (Below maximum temperature)				

RELIABILITY CONDITIONS

LCD MODULE (Consumer Type)		Conditions	Evaluation
High Temperature Operation	Operating 96 ~ 100 Hrs at 50 ± 2°C surrounding temp.		No change is visible in appearance nor function
Low Temperature Operation	Operating 96 ~ 100 Hrs at 0 ± 2°C surrounding temp.		
High Temperature Storage	Storage 96 ~ 100 Hrs at 60 ± 2°C surrounding temp. then storage 4 Hrs at normal condition (Power Off)		
Low Temperature Storage	Storage 96 ~ 100 Hrs at -20 ± 2°C surrounding temp. then storage 4 Hrs at normal condition (Power Off) No dew to be found		
Damp Proof	Storage 96 ~ 100 Hrs at 40 ± 2°C and 90 ~ 95% RH surrounding condition, then storage 4 Hrs at normal condition (Power Off) No dew to be found.		

Note The above condition is only representative, and may, differ in case of customized specifications.

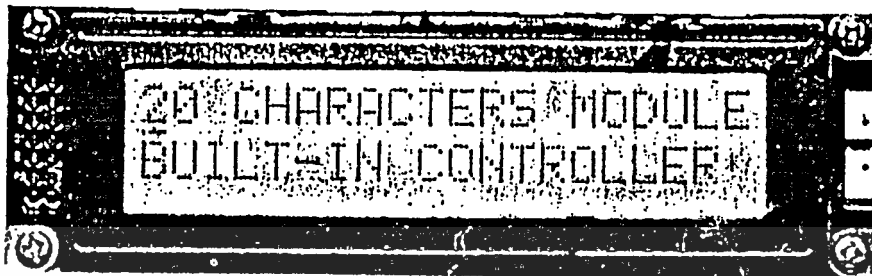
OPTICAL DATA

Ta = 25°C

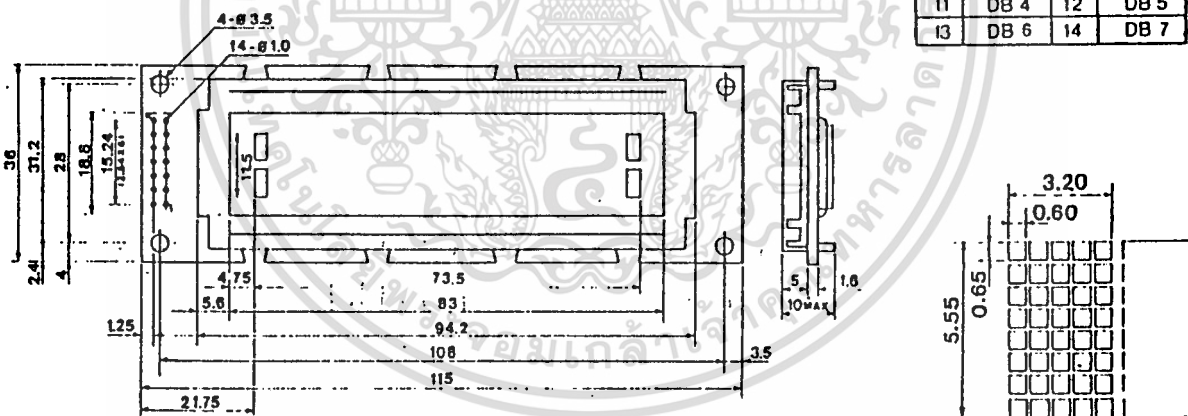
Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Notes to see
Viewing angle	$\phi 2 - \phi 1$	K = 1.4	—	20	—	deg.	—
Contrast ratio	K	$\phi = 25^\circ$ $\theta = 0^\circ$	—	2	—	—	—
Response time (rise)	t _r	$\phi = 25^\circ$ $\theta = 0^\circ$	—	250	400	ms	—
Response time (fall)	t _f	$\phi = 25^\circ$ $\theta = 0^\circ$	—	250	400	ms	—
				150	250		



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



DV-2002



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHARACTER FONT TABLE

Higher bit Lower bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CGRAM (RAM II)	ก	ข	ฅ	ด	ต	จ	ช	ซ	ฌ	ฎ	ฏ	ฐ
xxxx0001	(2)	ค	ฌ	ด	ต	จ	ช	ซ	ฌ	ฎ	ฏ	ฐ	ฑ
xxxx0010	(3)	ด	ต	จ	ช	ซ	ฌ	ฎ	ฏ	ฐ	ฑ	ฒ	ณ
xxxx0011	(4)	น	ด	ต	จ	ช	ซ	ฌ	ฎ	ฏ	ฐ	ฑ	ฒ
xxxx0100	(5)	บ	ป	ผ	ฝ	ภ	ม	ย	ร	ล	ว	ศ	ษ
xxxx0101	(6)	ห	ฬ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx0110	(7)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx0111	(8)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1000	(9)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1001	(10)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1010	(11)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1011	(12)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1100	(13)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1101	(14)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1110	(15)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ
xxxx1111	(16)	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ	อ	ฮ

NOTE: CGRAM is a CHARACTER GENERATOR RAM having a storage function of character pattern which enable to change freely by user's program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range
- 40 Pin DIP Package or 44 Lead PLCC
(See Intel Packaging; Order Number: 231369)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

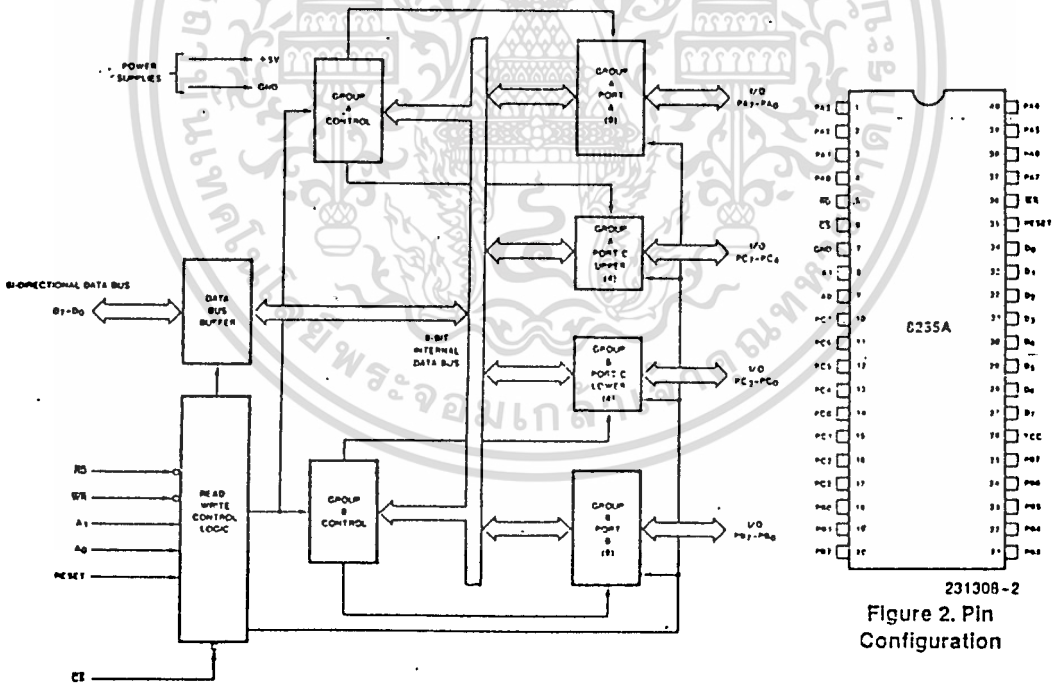


Figure 1. 8255A Block Diagram

Figure 2. Pin Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



8255A FUNCTIONAL DESCRIPTION

CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

(CS)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

(RD)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

(WR)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A₀ and A₁)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A₀ and A₁).

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the

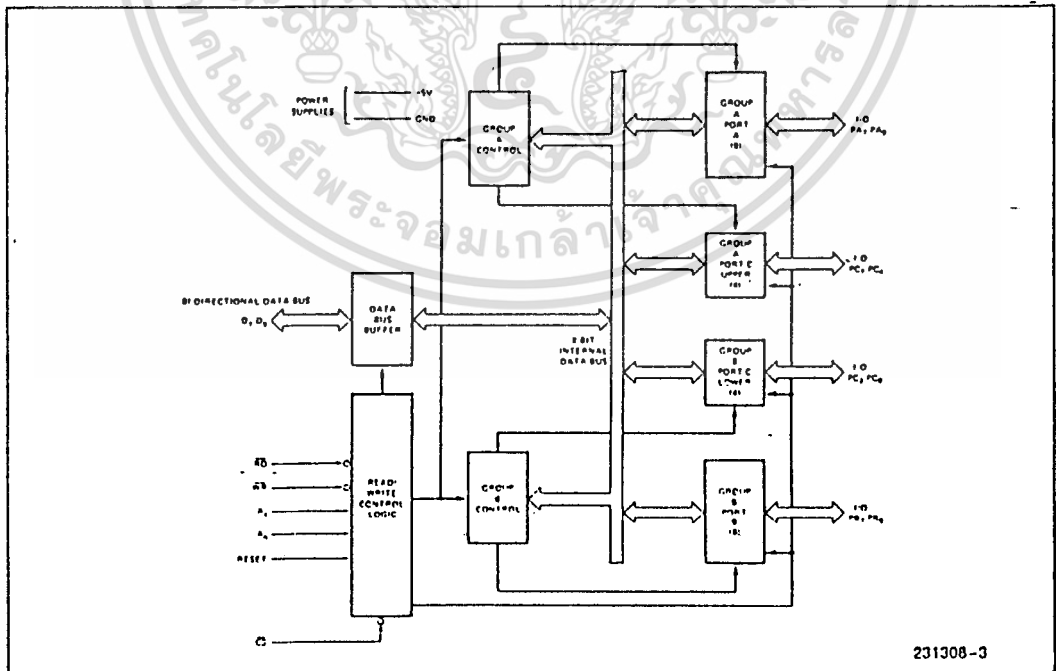


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

8255A/8255A-5

8255A BASIC OPERATION

A ₁	A ₀	\overline{RD}	\overline{WR}	\overline{CS}	Input Operallon (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					Output Operation (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A—Port A and Port C upper (C7–C4)

Control Group B—Port B and Port C lower (C3–C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 6-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.



8255A/8255A-5

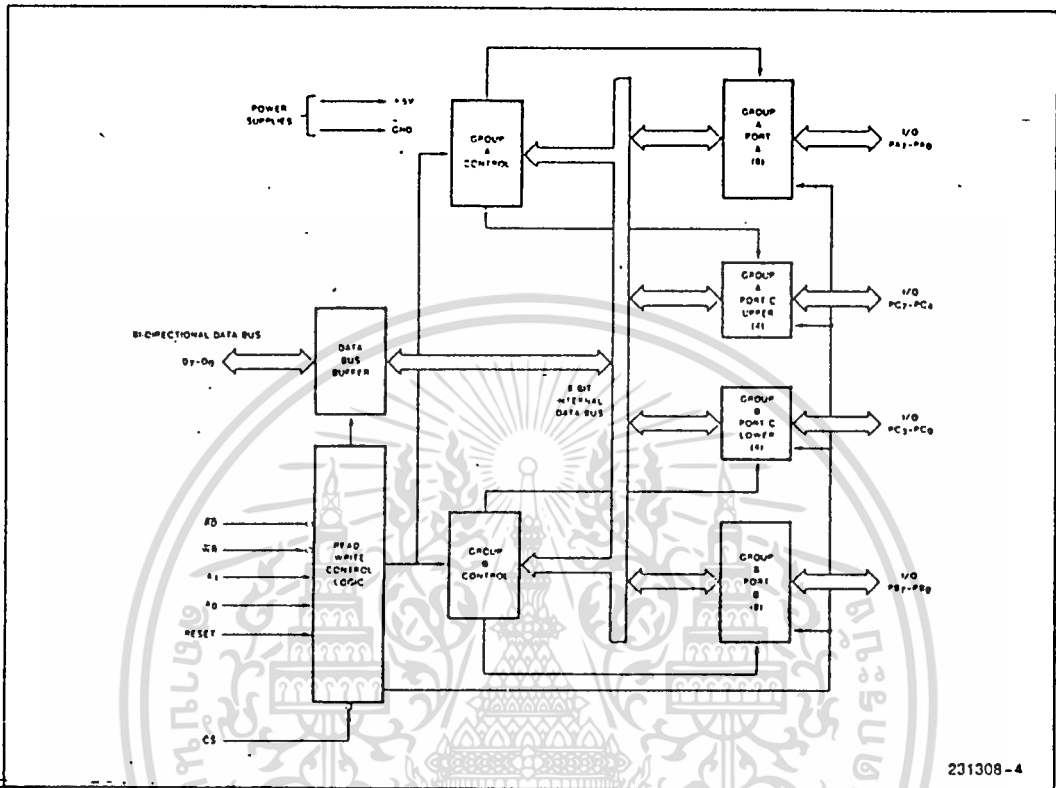
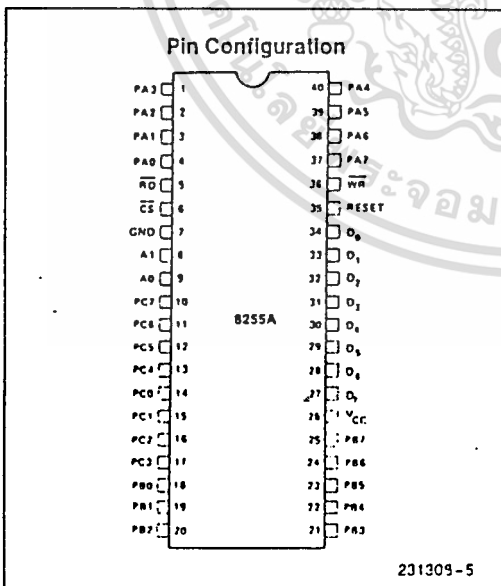


Figure 4. 8255A Block Diagram Showing Group A and Group B Control Functions



Pin Names	
D7-D0	Data Bus (Bi-Directional)
RESET	Reset Input
CS	Chip Select
RD	Read Input
WR	Write Input
A0, A1	Port Address
PA7-PA0	Port A (BIT)
PB7-PB0	Port B (BIT)
PC7-PC0	Port C (BIT)
Vcc	+ 5 Volts
GND	0 Volts

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Mode 0—Basic Input/Output

Mode 1—Strobed Input/Output

Mode 2—Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

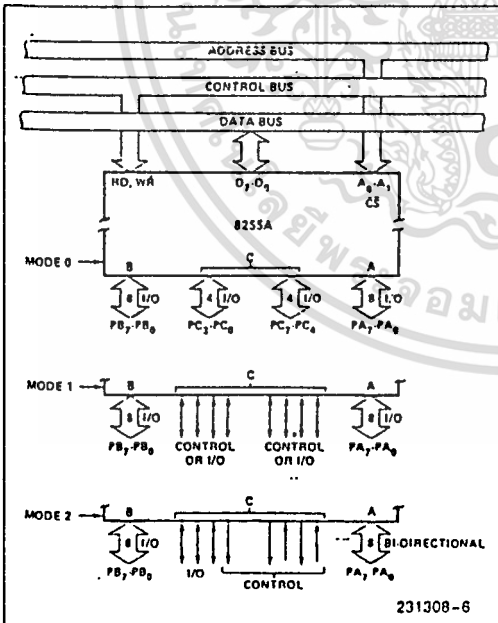


Figure 5. Basic Mode Definitions and Bus Interface

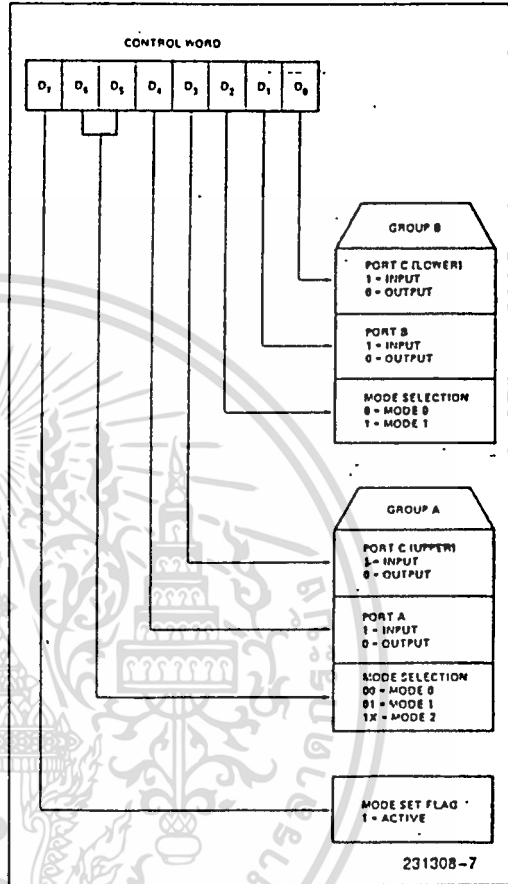


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

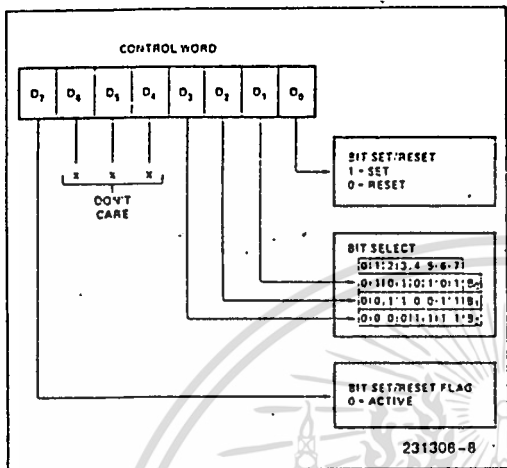


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is set—Interrupt enable

(BIT-RESET)—INTE is RESET—Interrupt disable

NOTE:

All Mask flip-flops are automatically reset during mode selection and device Reset.

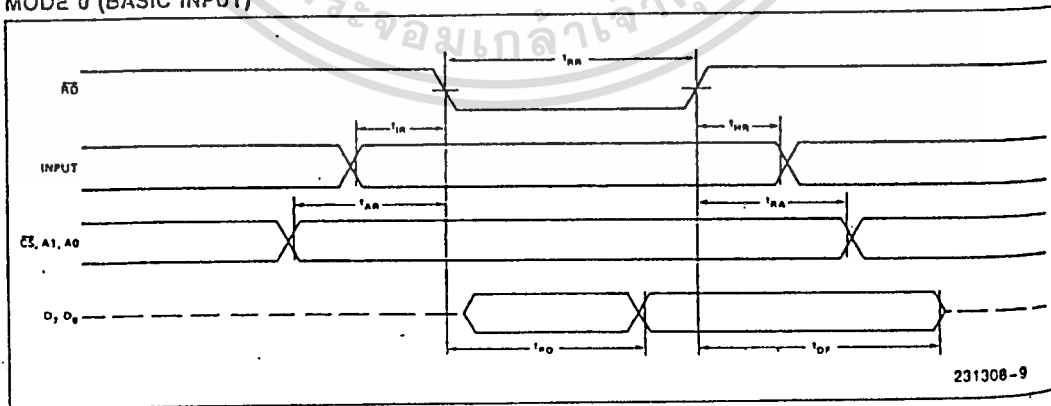
Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to-or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

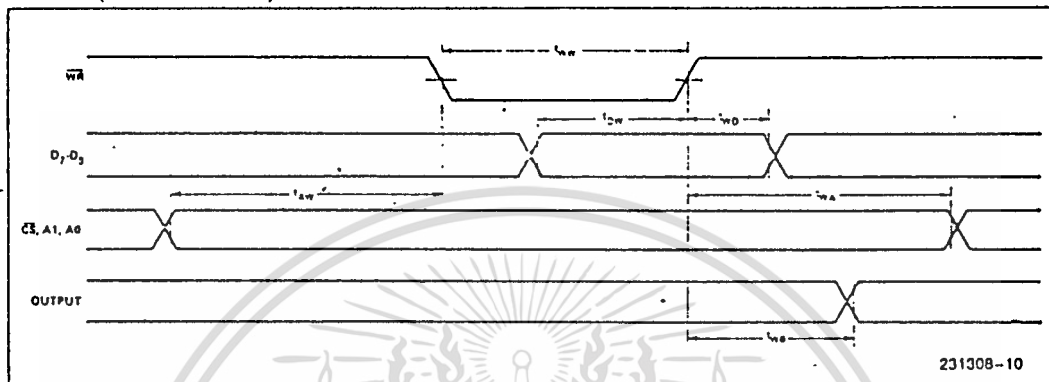
MODE 0 (BASIC INPUT)





8255A/3255A-5

MODE 0 (BASIC OUTPUT)

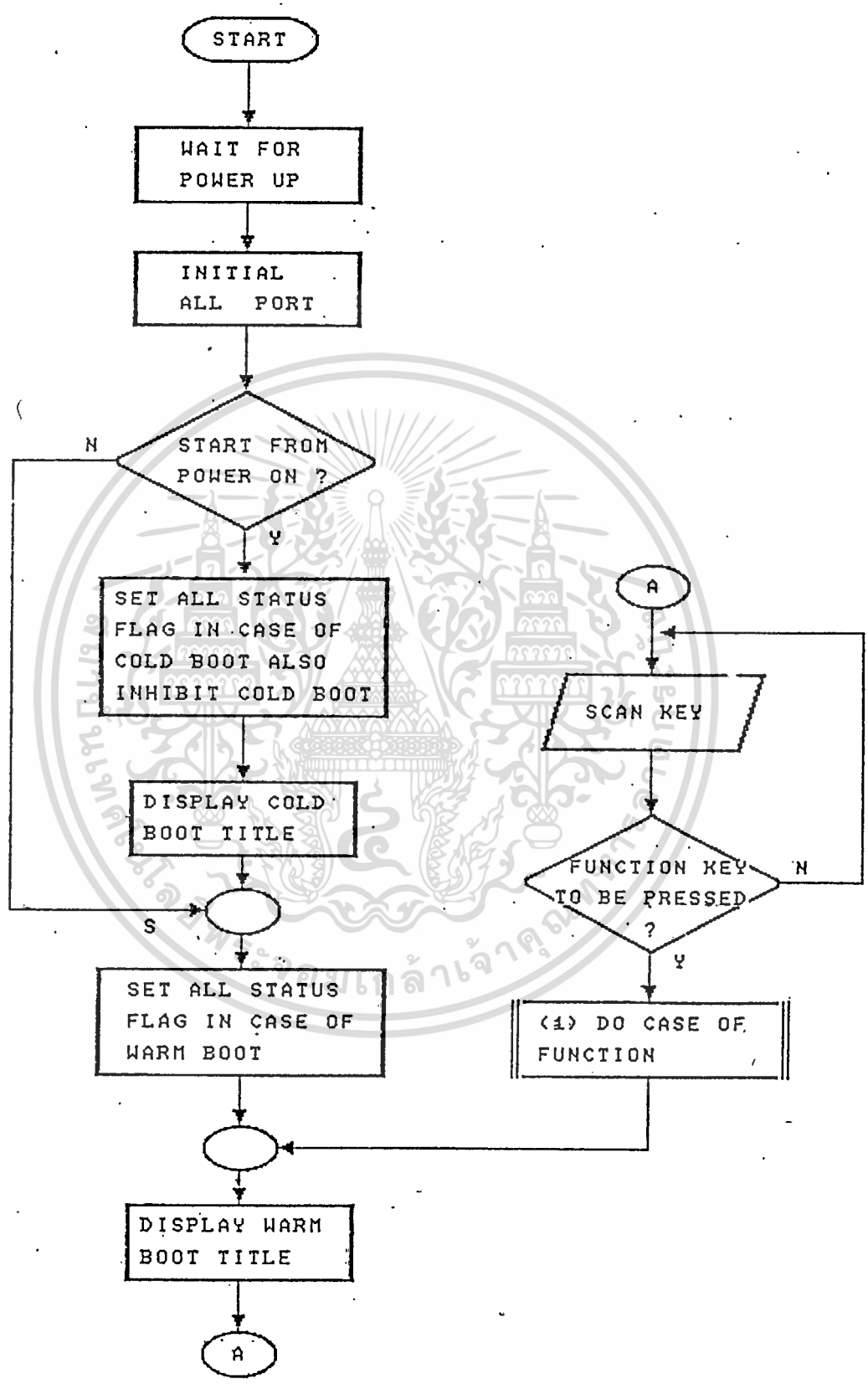


231308-10

MODE 0 PORT DEFINITION

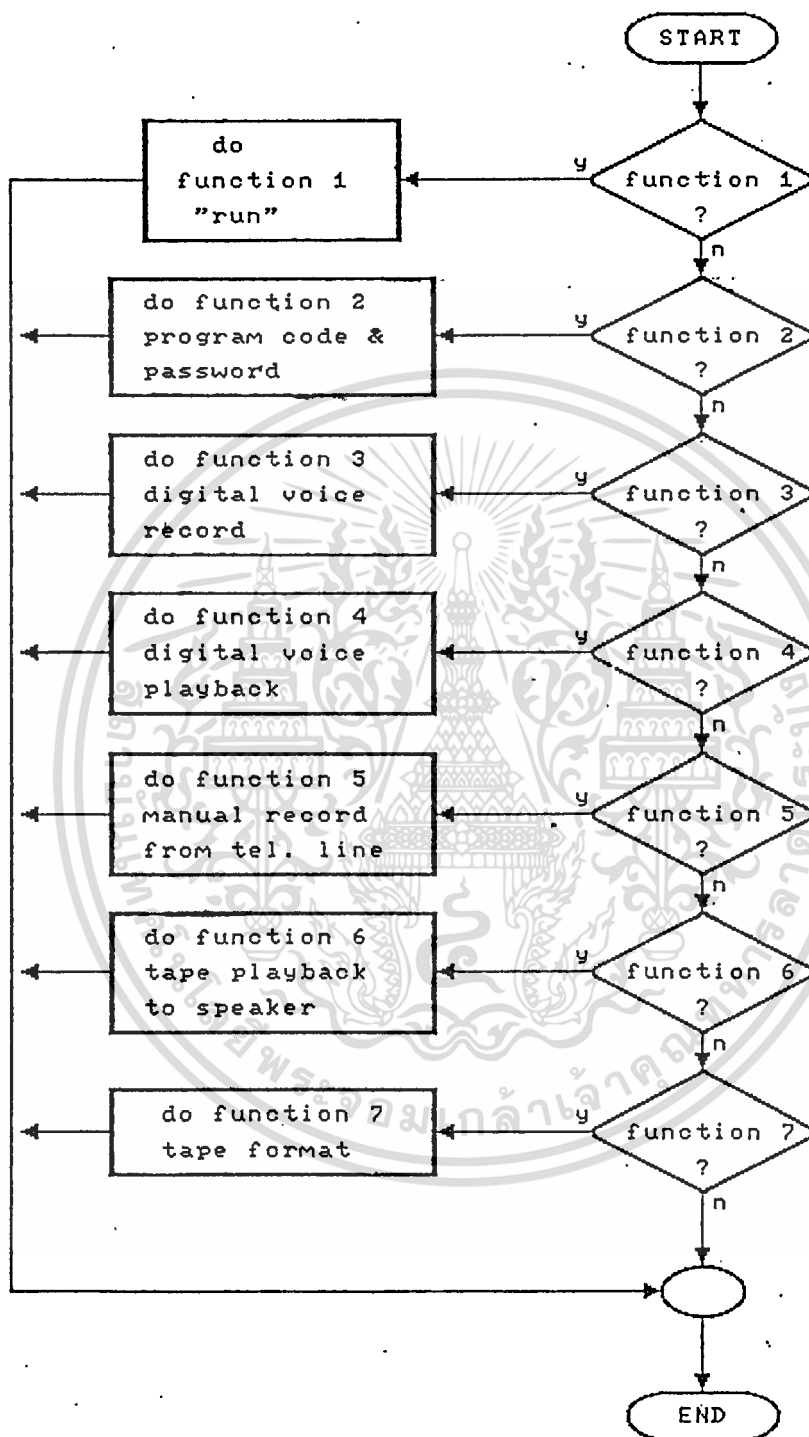
A		B		Group A			Group B		
D ₄	D ₃	D ₁	D ₀	Port A	Port C (Upper)	#	Port B	Port C (Lower)	
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT	
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT	
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT	
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT	
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT	
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT	
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT	
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT	
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT	
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT	
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT	
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT	
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT	
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT	
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT	
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



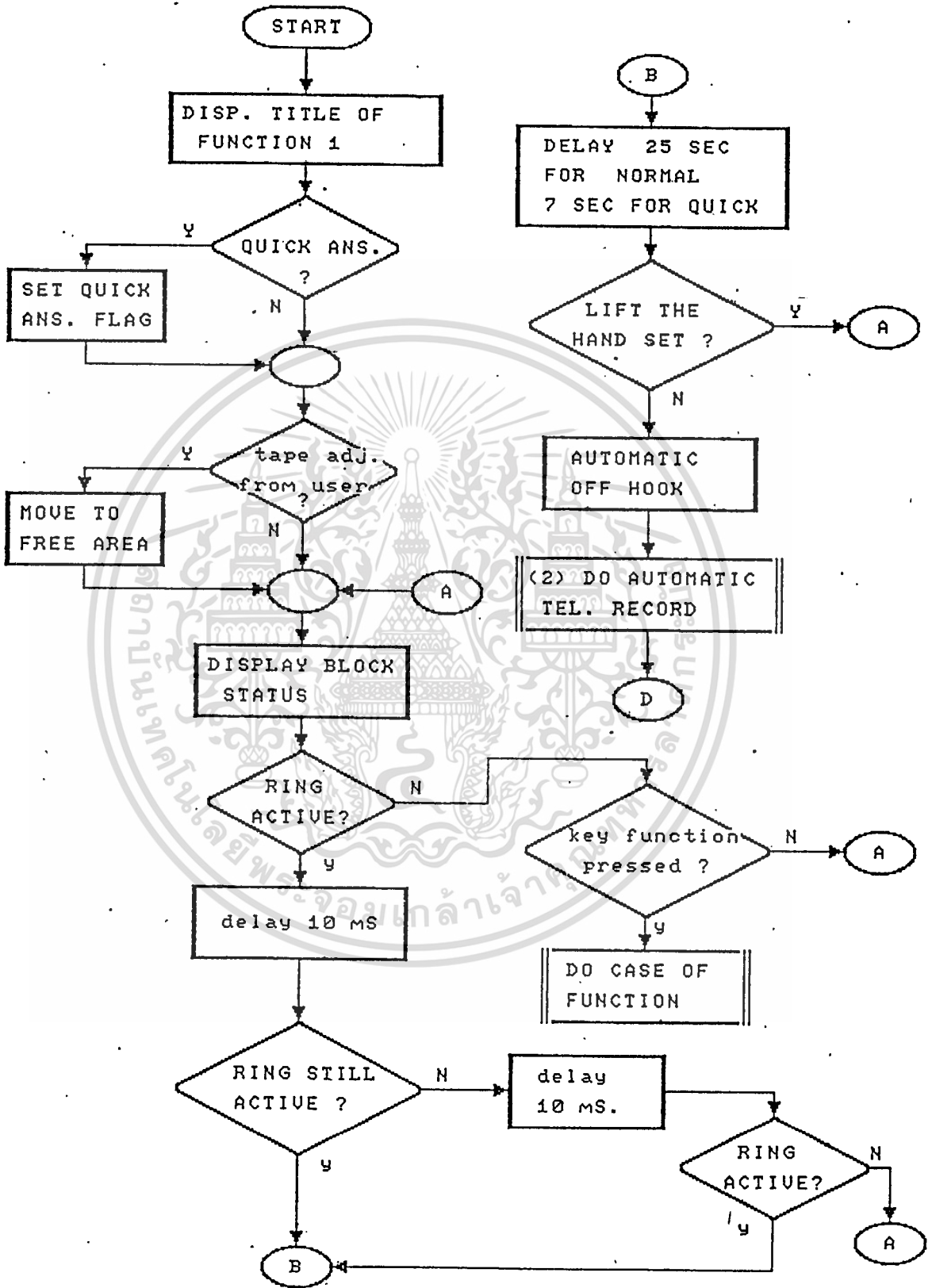
MAIN PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

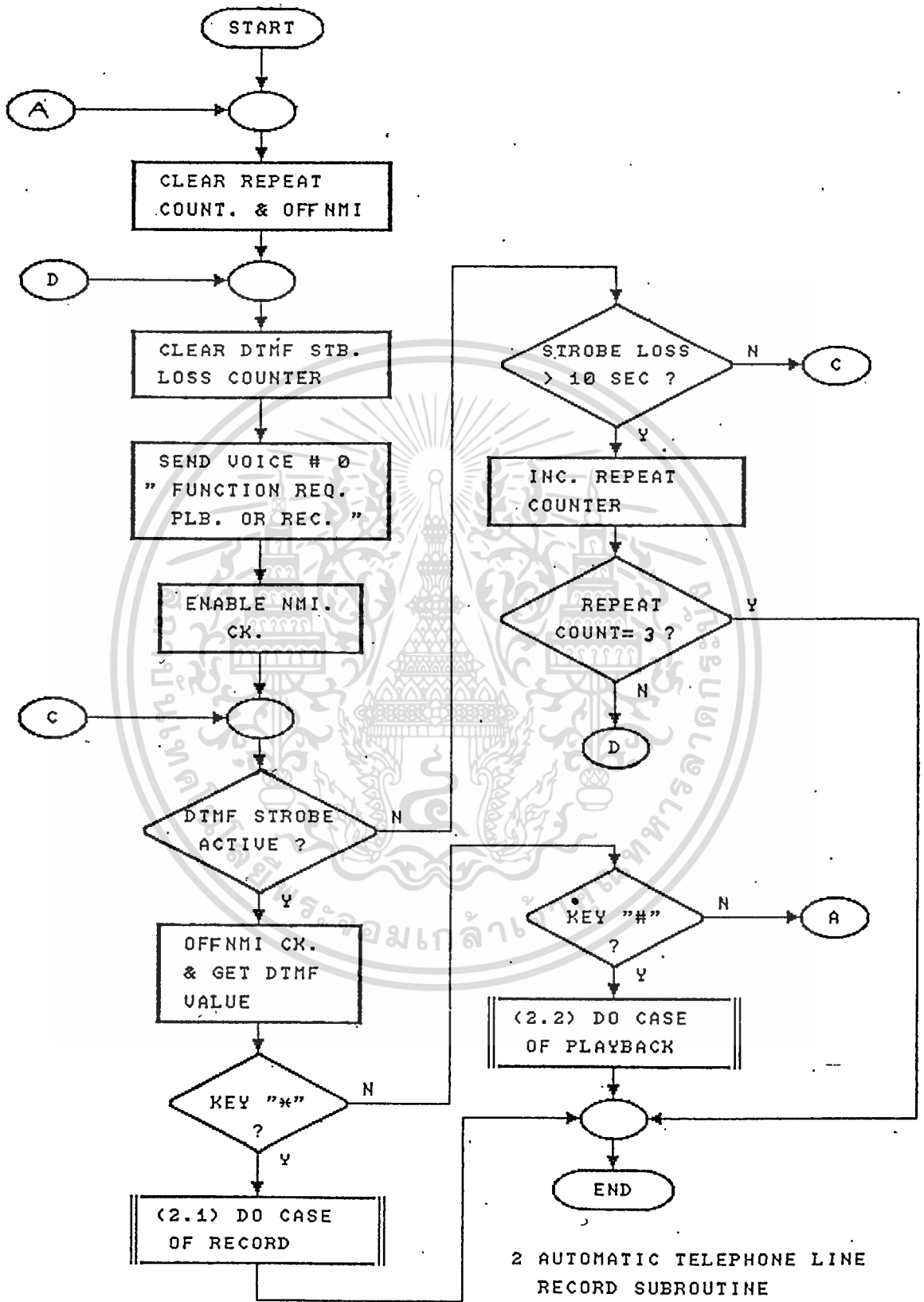


1 CASE OF FUNCTION

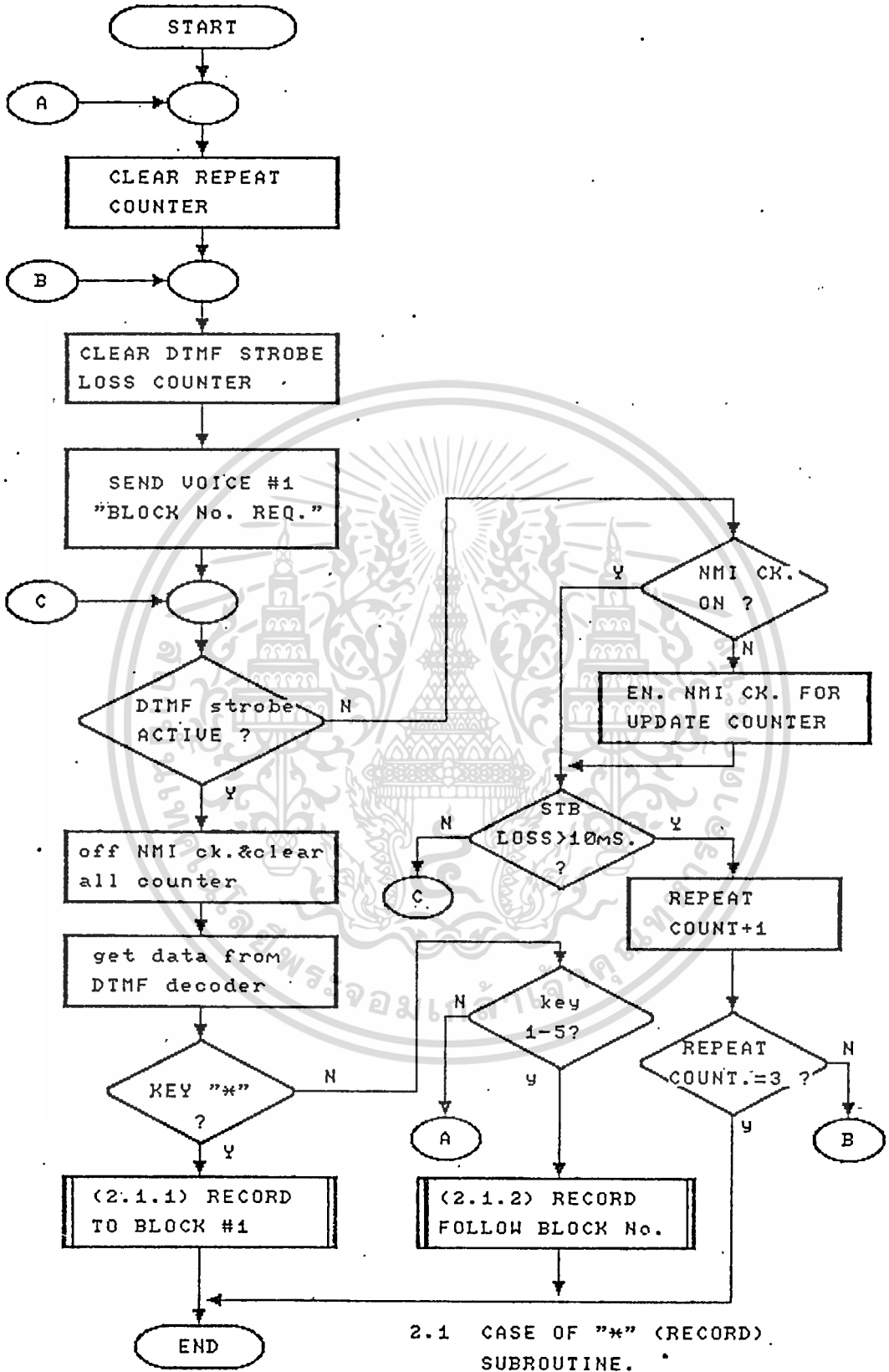
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



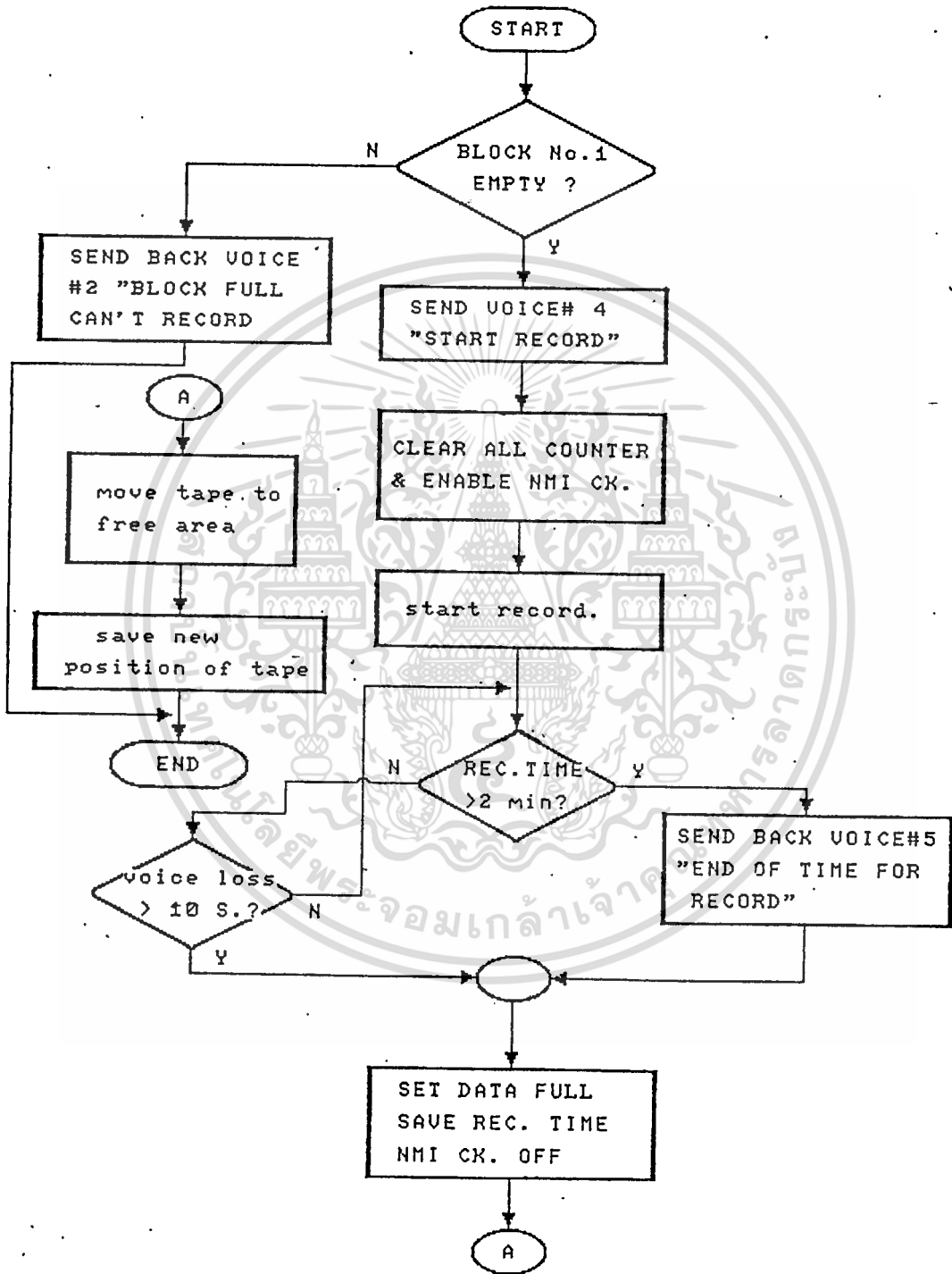
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน FUNCTION 1 RUN ROUTINE เท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

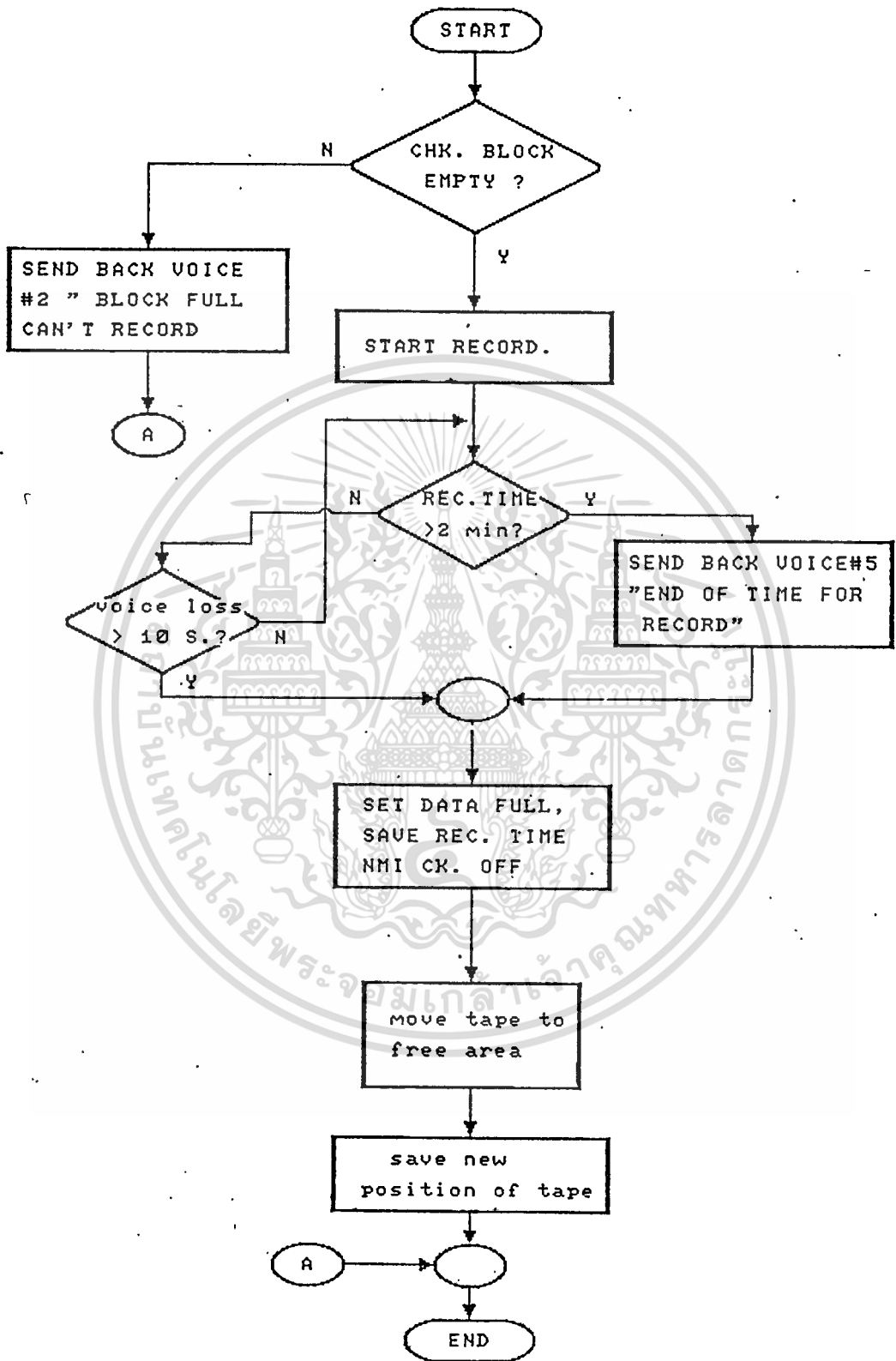


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



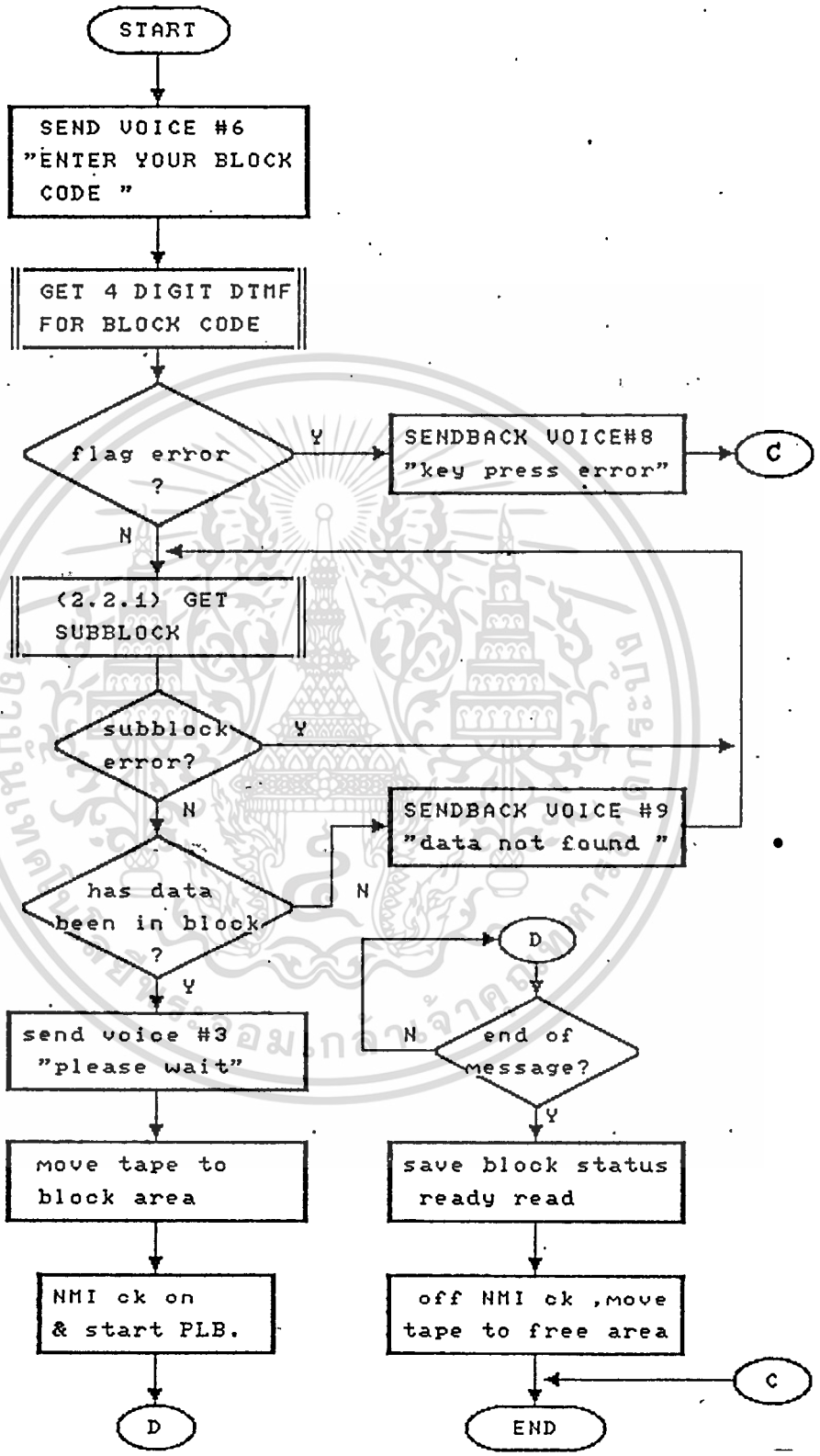
2.1.1 RECORD TO BLOCK No. 1 (HIGH PRIORITY BLOCK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



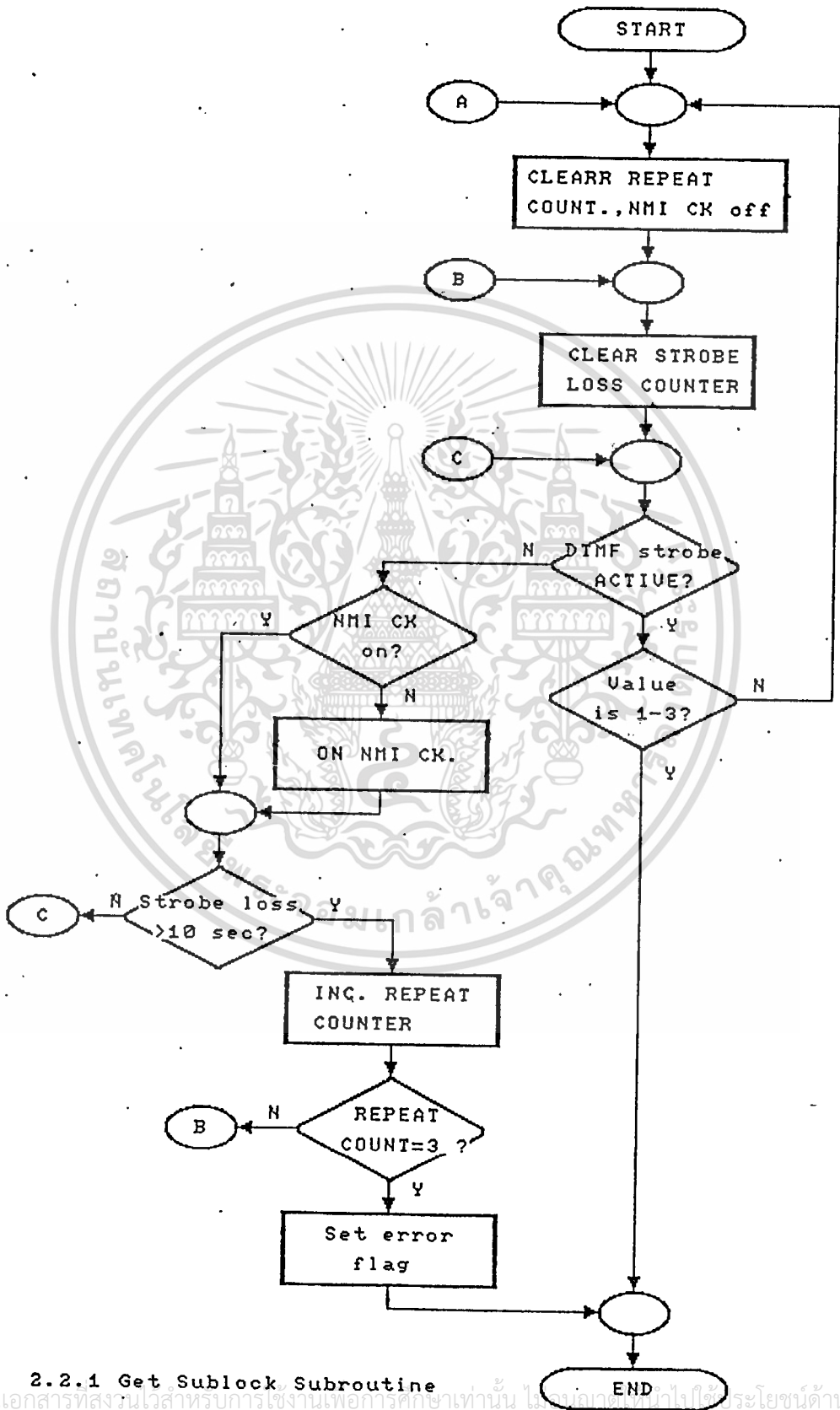
2.1.2 RECORD FOLLOW THE BLOCK NUMBER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



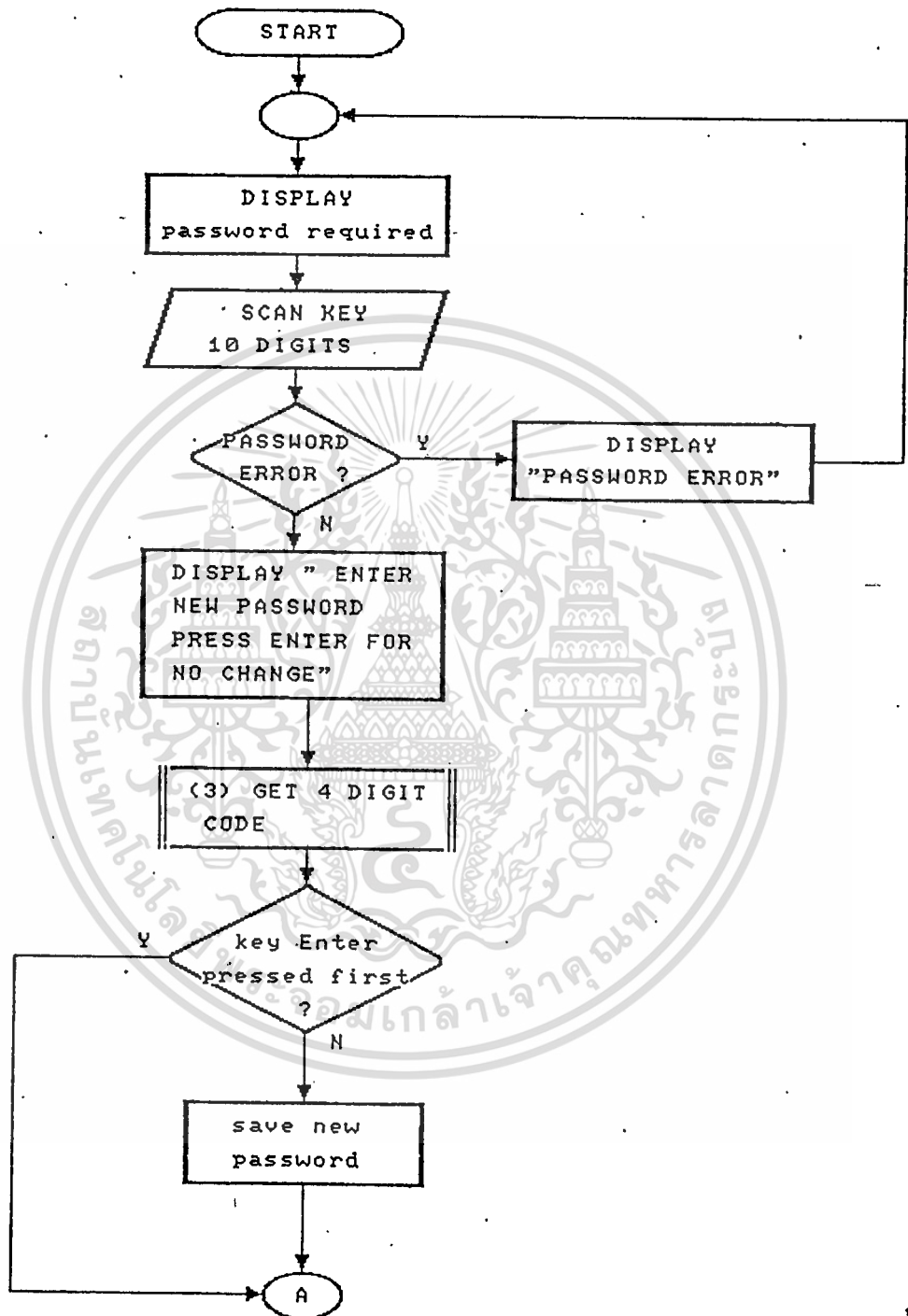
2.2 DO CASE OF '#' PLAYBACK SUBROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

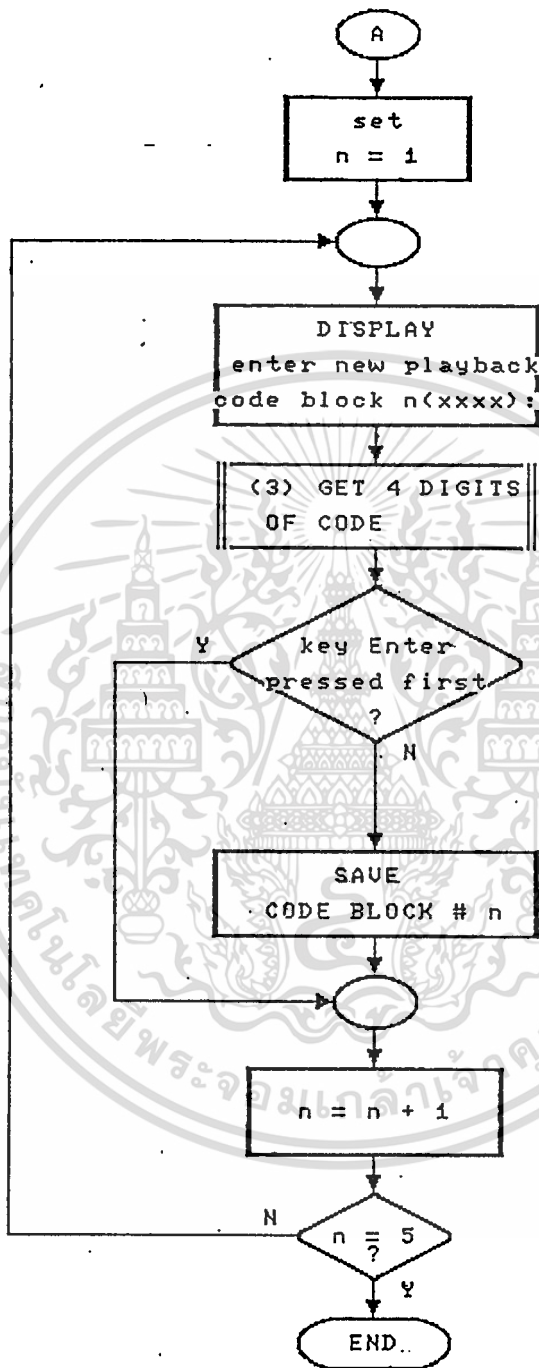


2.2.1 Get Sublock Subroutine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

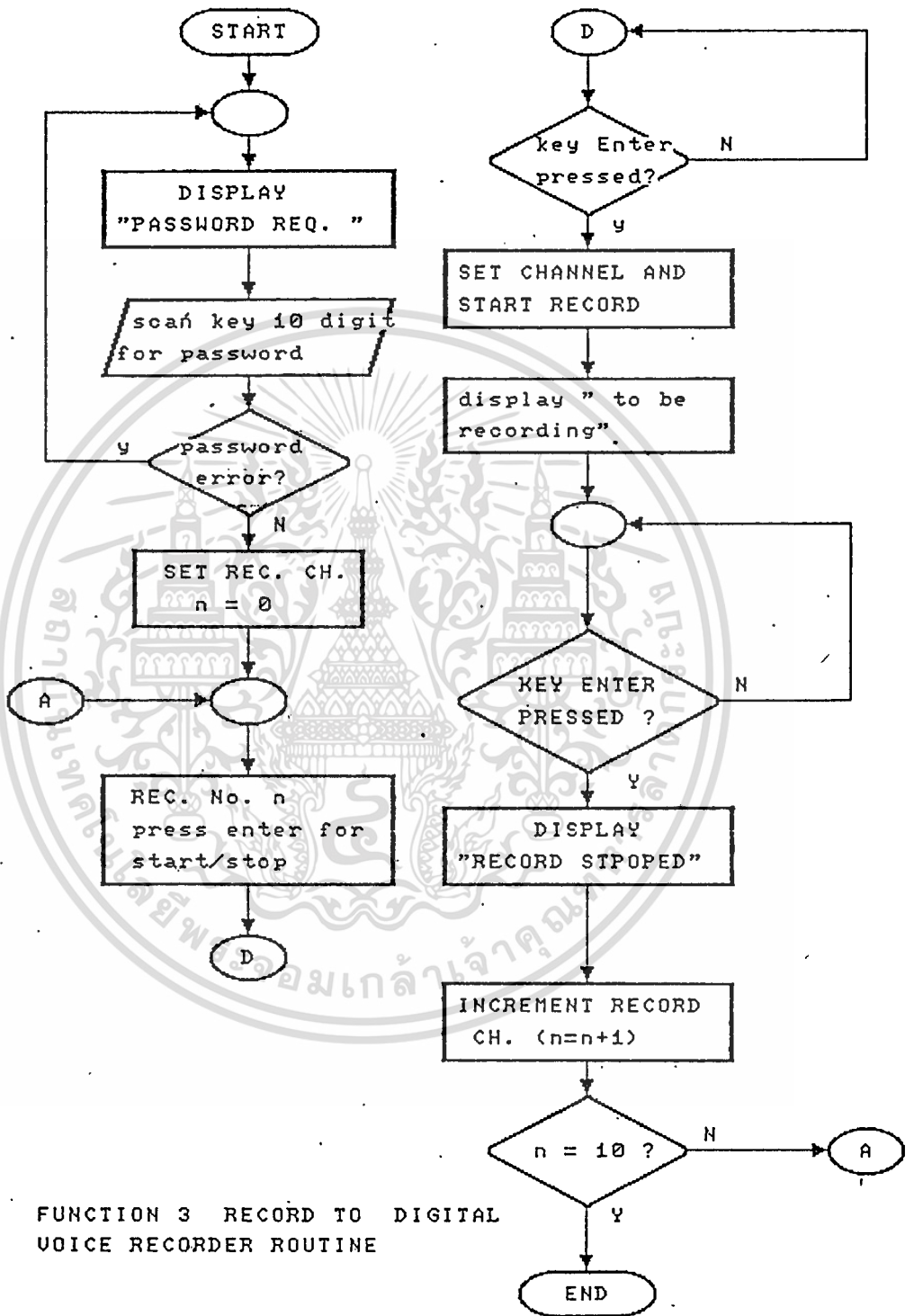


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

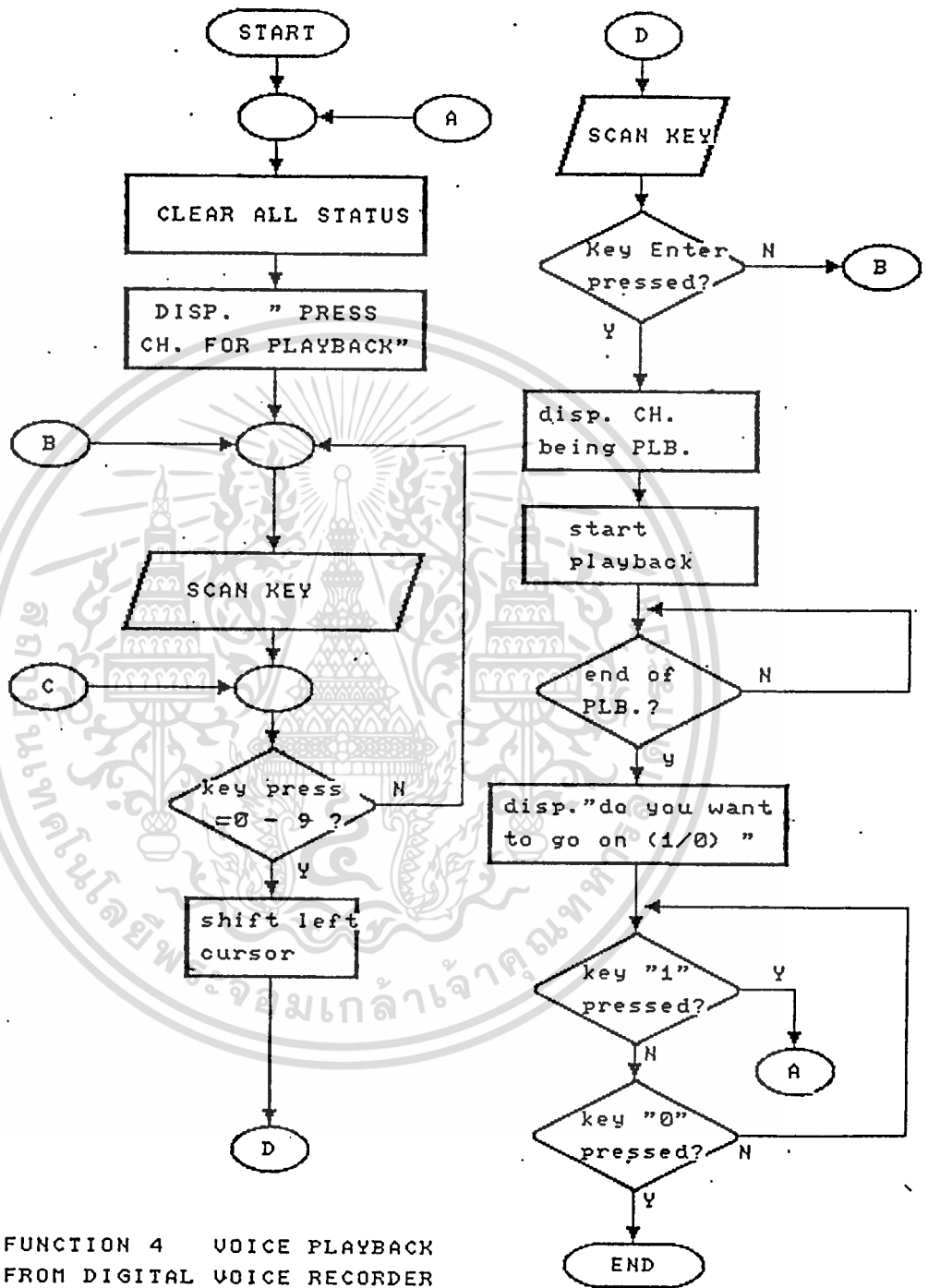


FUNCTION 2 PROGRAM CODE & PASSWORD ROUTINE

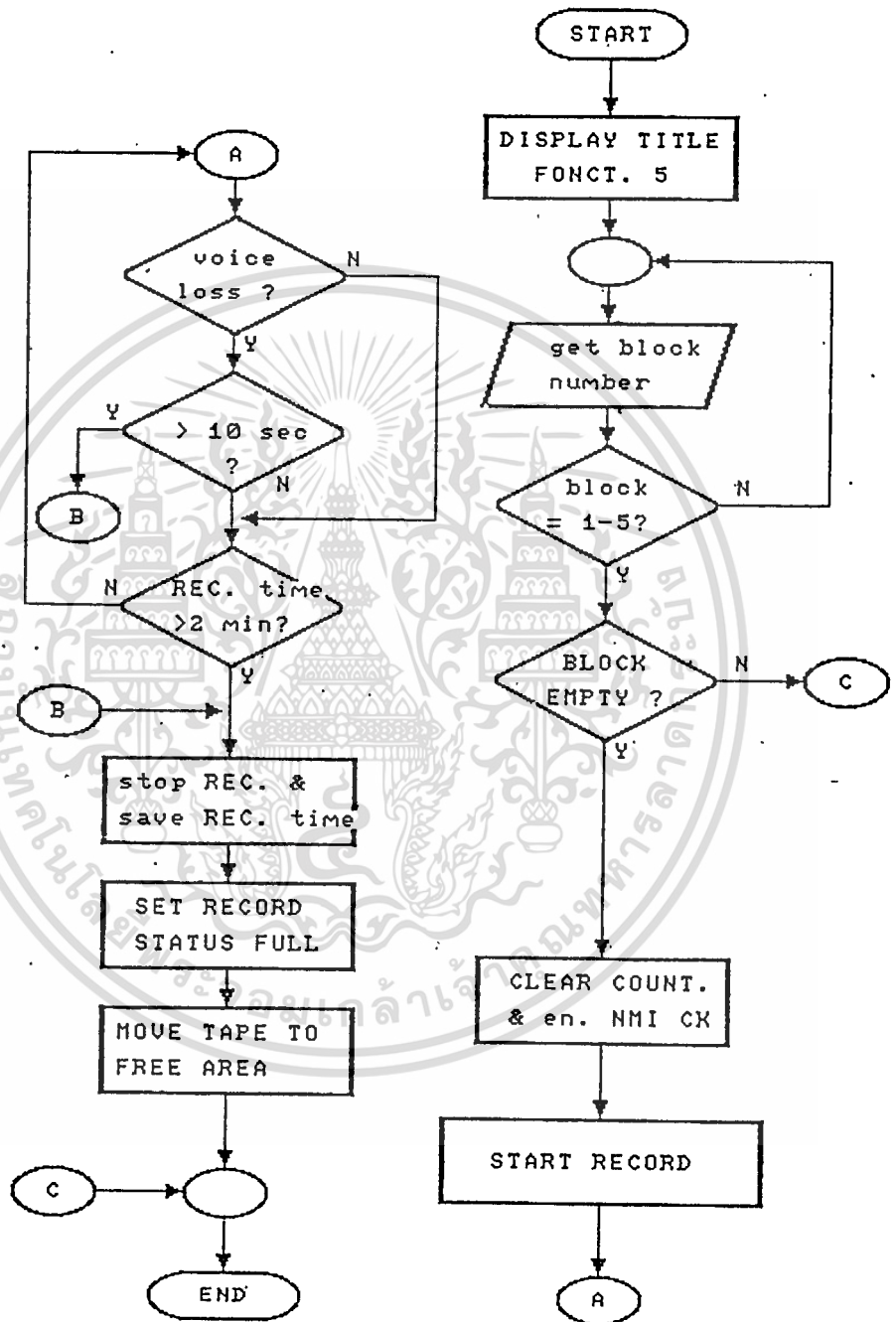
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FUNCTION 3 RECORD TO DIGITAL VOICE RECORDER ROUTINE

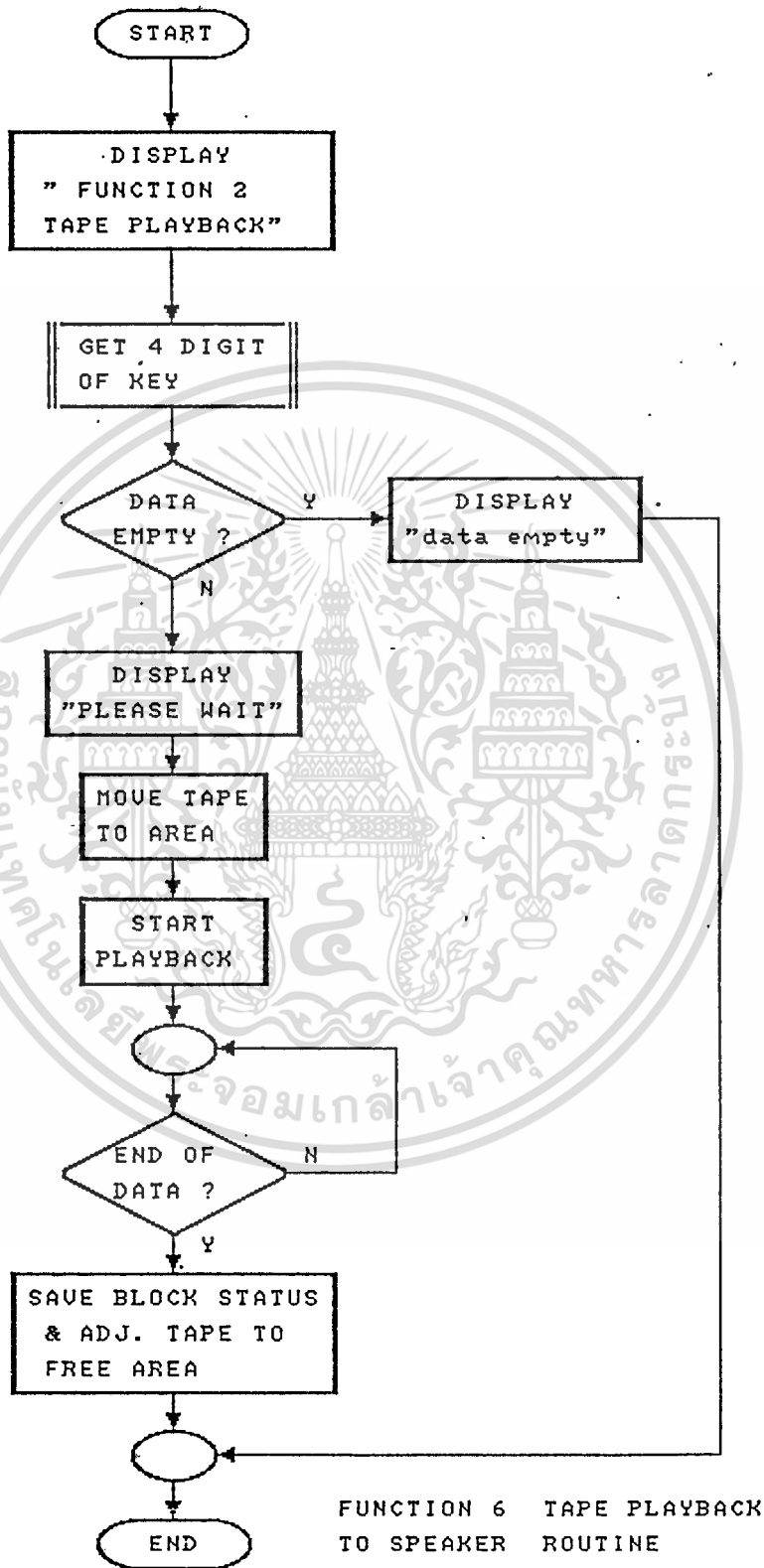


FUNCTION 4 VOICE PLAYBACK FROM DIGITAL VOICE RECORDER

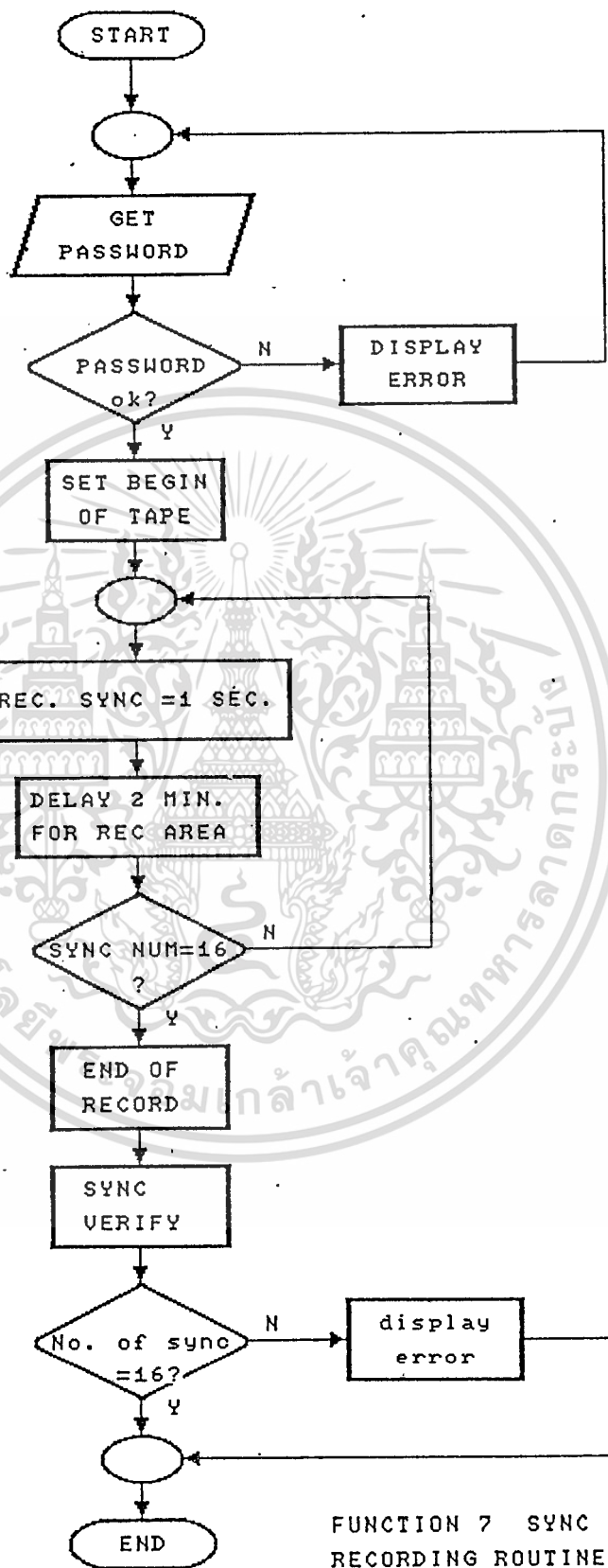


FUNCTION 5 MANUAL RECORD ROUTINE

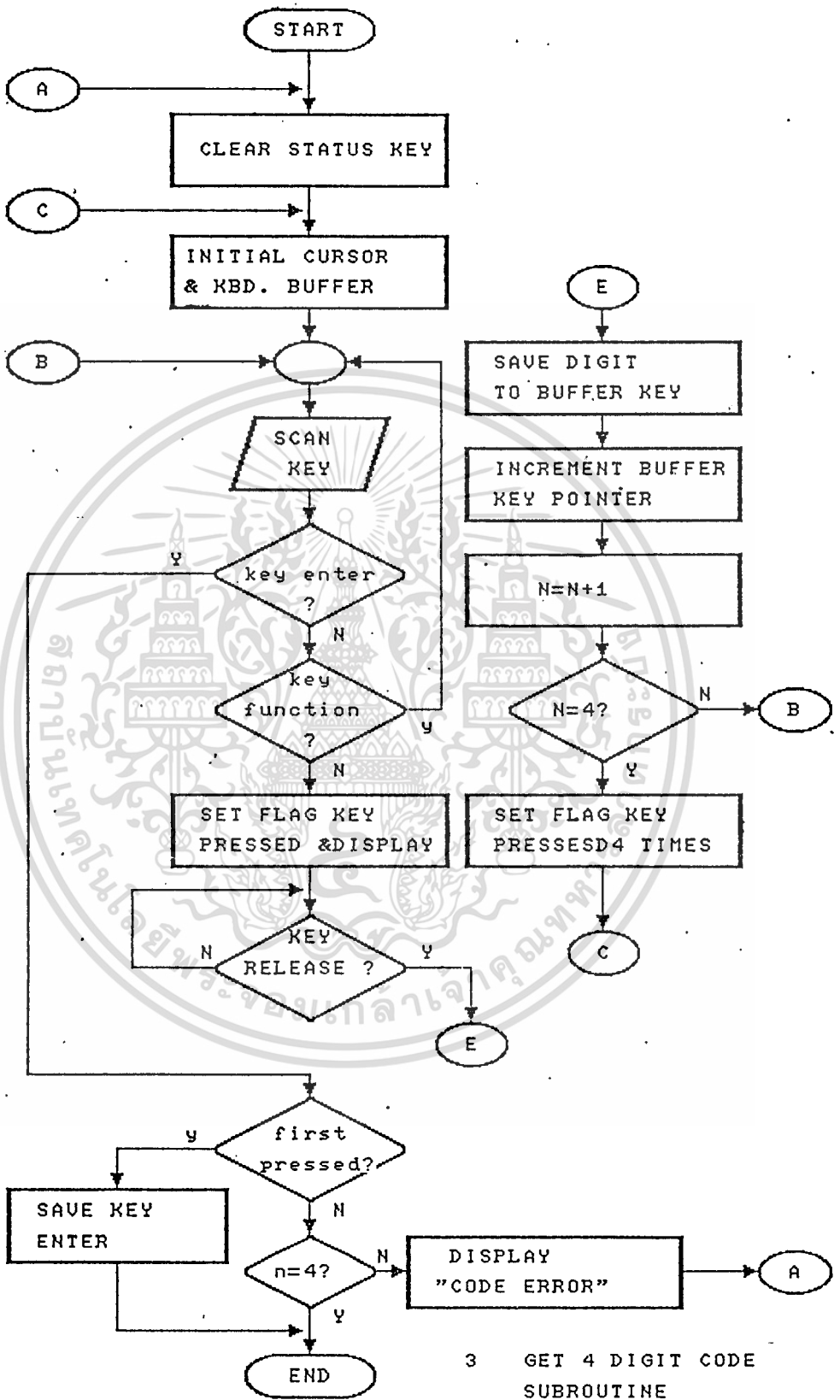
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



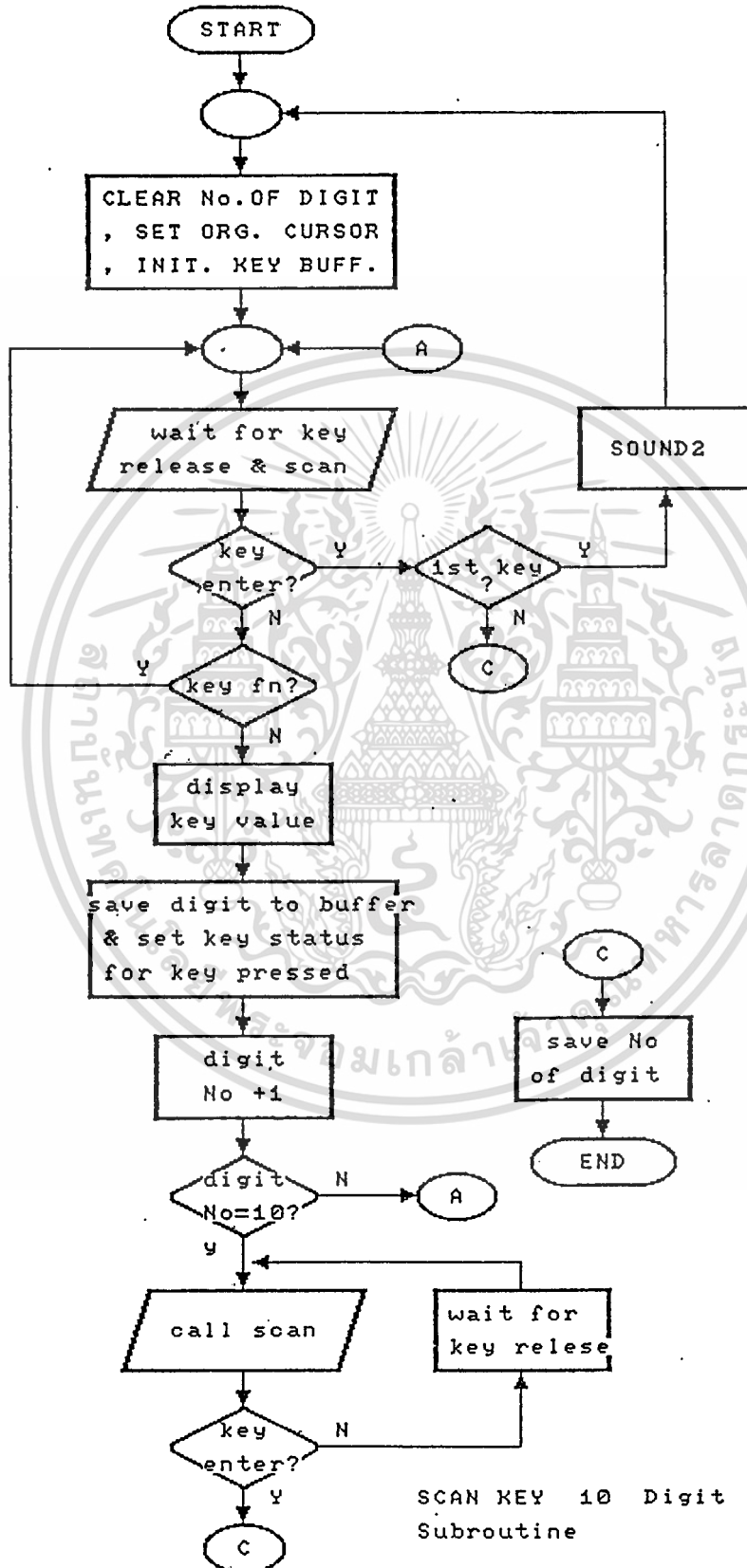
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



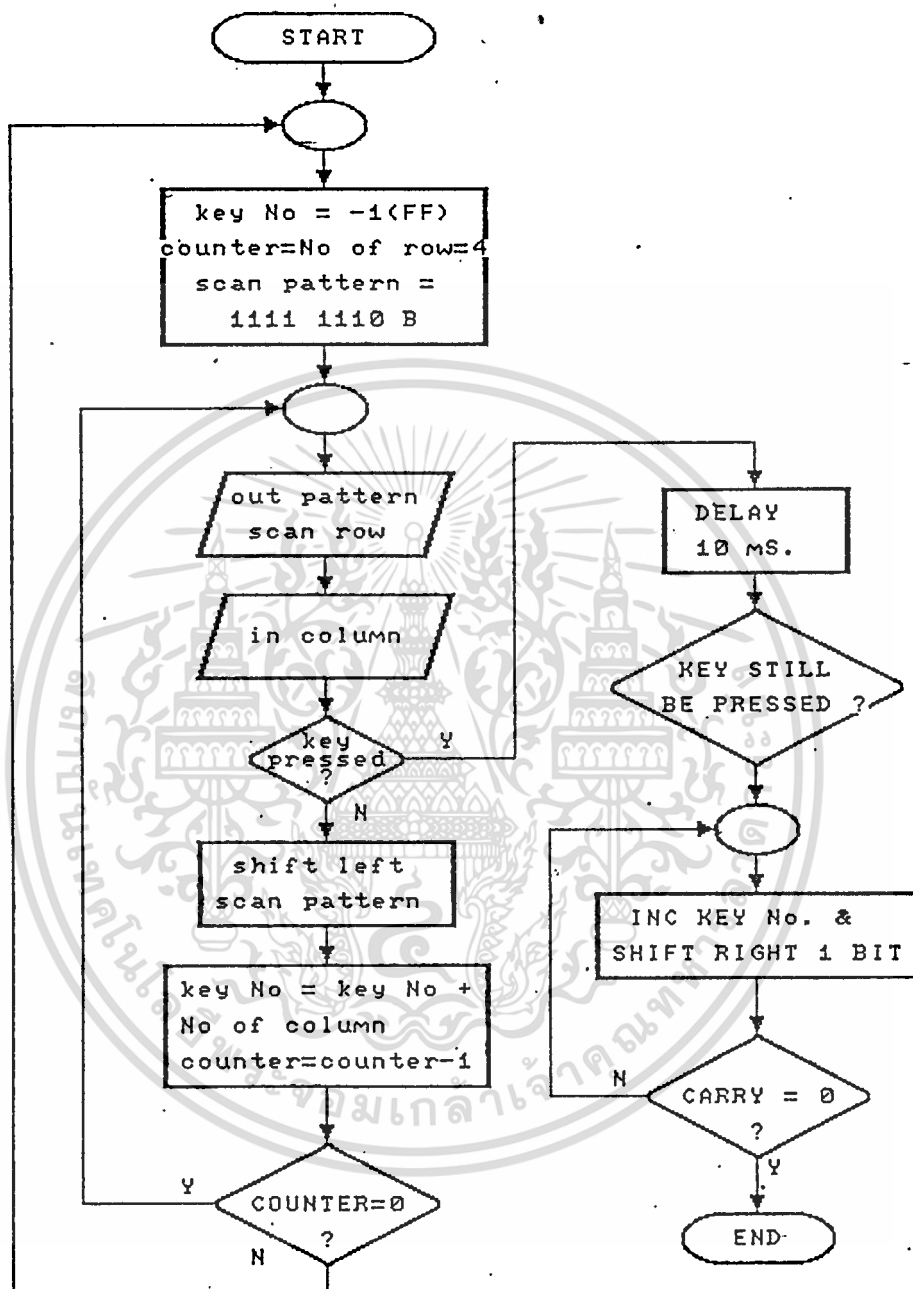
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



SCAN KEY SUBROUTINE

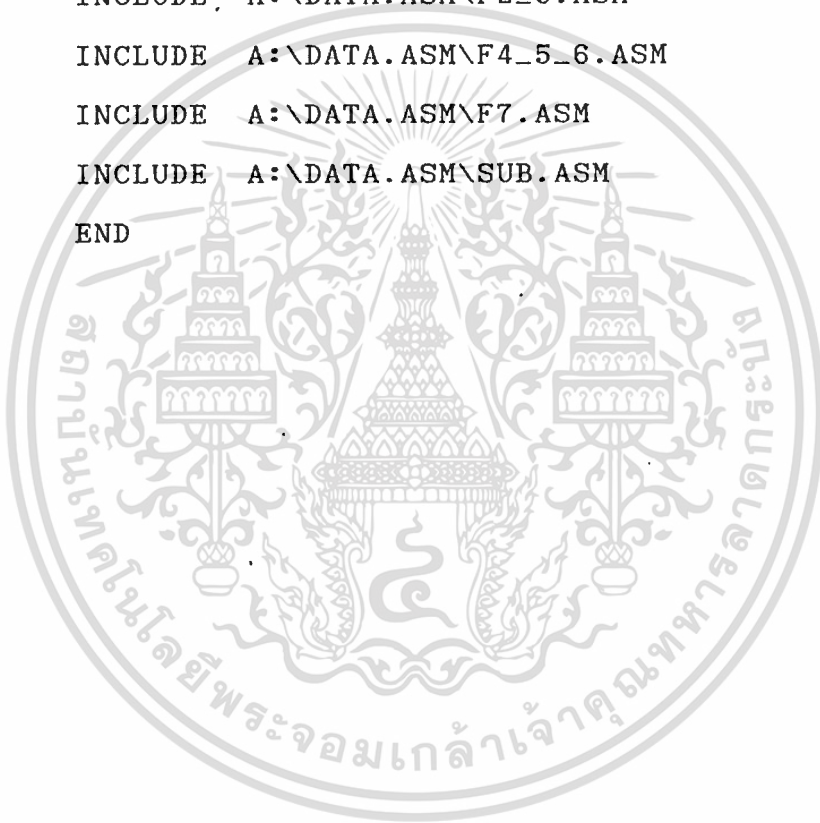
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; +-----+
; | FILES A.ASM ( INCLUDE FILES ) |
; +-----+

INCLUDE A:\DATA.ASM\MAIN.ASM
INCLUDE A:\DATA.ASM\F1.ASM
INCLUDE A:\DATA.ASM\F11.ASM
INCLUDE A:\DATA.ASM\F12.ASM
INCLUDE A:\DATA.ASM\F2_3.ASM
INCLUDE A:\DATA.ASM\F4_5_6.ASM
INCLUDE A:\DATA.ASM\F7.ASM
INCLUDE A:\DATA.ASM\SUB.ASM
END

```



```

;*****
;*      The mail box telephone monitor program .      *
;*                               By KMITL ET 2L        *
;*****

                ORG  0000H

PORTA:          EQU  80H          ;on board CPA
PORTB:          EQU  81H
PORTC:          EQU  82H
CTRP:          EQU  83H
CTRW:          EQU  80H
;
-----
PORTA1:         EQU  0A0H         ;o/p
PORTB1:         EQU  0A1H         ;i/p
PORTC1:         EQU  0A2H         ;U=i/p,L=o/p
CTRP1:         EQU  0A3H
CTRW1:         EQU   8AH
;
-----
PORTA2:         EQU  0C0H         ;o/p
PORTB2:         EQU  0C1H         ;o/p
PORTC2:         EQU  0C2H         ;o/p
CTRP2:         EQU  0C3H
CTRW2:         EQU   80H
;
-----
DTMFP:         EQU  0E0H         ;DTMF port
;
-----
TIME:          EQU  0DFFFH      .
TIME1:         EQU  0564H       ;for delay 10 mS.
TIME2:         EQU  0AC8H
TIME3:         EQU  102CH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TIME6:           EQU  2058H
TIMEF:           EQU  0FFFFH      ;for delay 475 mS.
ADD_ERR:         EQU  4DH
CMD_CUR:         EQU  4EH
STACK:           EQU  3000H
KBUF4:           EQU  2000H
KBUF1:           EQU  2005H
KBUF10:          EQU  2010H
PSWBUF:          EQU  2025H
BCBUF:           EQU  2040H
CLOCK1:          EQU  205FH
CLOCK2:          EQU  2060H
CLOCK3:          EQU  2061H
CLOCK4:          EQU  2062H
D_NUM:           EQU  20FFH
STA_K:           EQU  2105H
SYS1:            EQU  2106H
SYS2:            EQU  2107H
PLBNUM:          EQU  2108H
VCE_CNT:         EQU  2109H      ;count. send 1 st. vce.
TME_REC:         EQU  210AH
CNTR:            EQU  210BH
TAPE_BUFF:       EQU  210FH      ;45 byte:tape sta.,position
                                   ;and rec. time

PLB_BYTE1:       EQU  2140H      ;
TOGGLE:          EQU  2141H
CLOCK5:          EQU  2142H
CLOCK6:          EQU  2143H
CLOCK7:          EQU  2144H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLOCK8:	EQU	2145H	
CLOCK9:	EQU	2146H	
CLOCKA:	EQU	2147H	
CLOCKB:	EQU	2148H	
CLOCKC:	EQU	2149H	
CLOCKD:	EQU	214AH	
CLOCKE:	EQU	214BH	
CLOCKF:	EQU	214CH	
CLOCKG:	EQU	214DH	
CLOCKH:	EQU	214EH	
CLOCKI:	EQU	214FH	
CLOCKJ:	EQU	2150H	
B_STA2:	EQU	2151H	;20 bytes of rec. sta. disp.
TTKBUFF1:	EQU	216AH	;touch tone key buff. 4 byte
PLAYBYTE2:	EQU	2170H	
PLB_BYTE2:	EQU	2175H	
TP_FLAG:	EQU	217AH	;15 bytes of tape location
TP_HEAD:	EQU	2190H	;real position of head
TEM_HEAD:	EQU	2195H	
DIS_MOV:	EQU	219AH	
TP_FLG_B:	EQU	219EH	
DIR_FLG:	EQU	21A5H	
POINT_T:	EQU	21A7H	
DIS_TMP:	EQU	21AAH	
DTMF_TMP:	EQU	21ACH	
TEM_PNT:	EQU	21AFH	
Q_ANS:	EQU	21B5H	
PLB_B2:	EQU	21B7H	
JUST_RD:	EQU	21BAH	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

NOP
;
; -----
; * INITIAL LCD *
; -----
INIT_LCD:      LD      A,0          ;
               OUT     (PORTC2),A    ;
               LD      A,38H        ;function set 38h
                                   ;D=1 8 bit,N=1 1/16 duty
               OUT     (PORTB2),A    ;
               CALL    EPULSE
               CALL    DELAY        ;delay > 4.1 ms
               LD      A,0FH        ;disp on/off control D=1 off
                                   ;c=1 cursor on,B=1 blink
               OUT     (PORTB2),A
               CALL    EPULSE
               LD      A,06H        ;entry mode set
                                   ;I/D=1 increment ,S=0 right
               OUT     (PORTB2),A
               CALL    EPULSE
               LD      A,01H
               OUT     (PORTB2),A   ;clear all display
               CALL    EPULSE
               CALL    DELAY
;
; -----
; * CHECK COLD BOOT *
; -----
LD      A,(SYS1)
CP      10101010B
JR      NZ,CDB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A, (SYS2)
CP      01010101B
JP      Z, RST_WBOOT
CDB:    NOP
LD      A, 01010101B
LD      (SYS2), A
LD      A, 10101010B      ;COLD BOOT FLAG
LD      (SYS1), A
;
;
;
NONMI:  LD      IX, PSWBUF      ;
LD      A, 35H              ;
JR      NONMI              ;avoid NMI
JP      NMI
LD      (IX+0), A          ;
LD      (IX+1), A          ;set pass word default
LD      (IX+2), A          ;
LD      (IX+3), A          ;
LD      A, 29H
LD      (IX+4), A
LD      A, 20H
LD      (IX+5), A
;
;
;
;
LD      HL, BCBUF
LD      A, 31H              ;start block No.1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                LD      C,05H          ;No. of bolck
ST:                               NOP
                                LD      B,04H
                                PUSH   BC
RELOOP:                          LD      (HL),A
                                INC     HL
                                DJNZ   RELOOP
                                POP     BC
                                INC     A
                                DEC     C
                                JR      NZ,ST
;
;
;
                                * INIT. T6668 *
-----
                                LD      A,0001000B      ;initial
                                OUT     (PORTA1),A
                                CALL    DELAY60
                                RES     4,A
                                OUT     (PORTA1),A      ;RES
                                CALL    DELAY60
                                SET     4,A
                                OUT     (PORTA1),A
                                CALL    DELAY60
;
;
;
                                * set initialization of status *
                                * ,position and rec. time of tape. *
-----
                                LD      BC,45D          ;lenght
                                LD      HL,TAPE_TBL      ;source

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      DE,TAPE_BUFF      ;distination
LDIR
;
;
;
-----
* set initialize of TP_FLG table=0 *
-----
LD      IX,TP_FLAG
LD      A,00H
LD      B,0FH
INIT_CLR: LD      (IX+0),A
          INC      IX
          DJNZ     INIT_CLR
;
;
;
-----
* set table disp. title of rec. block sta.*
-----
LD      IX,B_STA2
LD      A,30H
LD      B,14H
RE_SET: LD      (IX+0),A
          INC      IX
          DJNZ     RE_SET
LD      IX,B_STA2
LD      A,20H
LD      (IX+3),A
LD      (IX+7),A
LD      (IX+0BH),A
LD      (IX+0FH),A
LD      (IX+13H),A
;
-----
CALL    DISP_CDB          ;cold boot title diap.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; -----
RST_WBOOT:      NOP ;start of warm boot
; -----

LD      A,00H
OUT     (PORTA2),A      ;init. tape
LD      A,00010000B
OUT     (PORTA1),A      ;init SW T6668
CALL    DISP_WB        ;disp warm boot&get
; key fn.

GET_K:          CALL    K_RELEASE
; -----
; -----
; -----
CHKF:          LD      A,(KBUFF1)
CP      31H
JP      Z,FUNCT1
CP      32H
JP      Z,FUNCT2
CP      33H
JP      Z,FUNCT3
CP      34H
JP      Z,FUNCT4
CP      35H
JP      Z,FUNCT5
CP      36H
JP      Z,FUNCT6
CP      37H
JP      Z,FUNCT7
CP      38H              ;DEBUG

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JP      Z,FUNCT8
JP      RST_WBOOT      ; restart
;
-----
TAPE_TBL:
DB      10H,20H,00H,20H,20H,00H,30H,20H,00H
DB      10H,20H,00H,20H,20H,00H,30H,20H,00H
DB      10H,20H,00H,20H,20H,00H,30H,20H,00H
DB      10H,20H,00H,20H,20H,00H,30H,20H,00H
DB      10H,20H,00H,20H,20H,00H,30H,20H,00H
;
* END OF MAIN PROGRAM *
END

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          *****
;          * FUNCTION 1 run routine *
;          *****
FUNCT1:   CALL  CLRDISP
          LD   A,00
          CALL GOTO
          LD   HL,F1D1L1
          CALL WR_LINE      ;disp. Funct.1 system RUN
          CALL DELAYF
          CALL DELAY60
          CALL CLRDISP
          CALL DELAY60
          ;
          LD   A,00
          CALL GOTO
          LD   HL,F1D6L1      ;disp. quick answer or not?
          CALL WR_LINE
          LD   A,40H
          CALL GOTO
          LD   HL,F1D1L2      ;disp. press 1=YES/0=NO
          CALL WR_LINE
CHK_AGN:  CALL  K_RELEASE
          CALL  SCAN
          LD   A,(KBUFF1)
          CP   30H
          JR   NZ,CHK_K_1
          LD   A,00
          LD   (Q_ANS),A
          JR   N_QANS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHK_K_1:      CP      31H
              JR      NZ,CHK_AGN
              LD      A,11H
              LD      (Q_ANS),A      ;set quick answer flag
;
;
N_QANS:      CALL    CLRDISP
              CALL    DELAYF
              CALL    DELAY60
RE_ADJ:      LD      A,00H
              CALL    GOTO
              LD      HL,F1D2L1
              CALL    WR_LINE      ;disp. tape adj. or not ?
              CALL    DELAY60
              LD      A,40H
              CALL    GOTO
              LD      HL,F1D1L2      ;disp. press 1=YES/0=NO
              CALL    WR_LINE
              CALL    K_RELEASE
              CALL    SCAN
              LD      A,(KBUF1)
              CP      31H
              JR      Z,ADJ
              CP      30H
              JP      Z,NO_ADJ
              JR      RE_ADJ
ADJ:         CALL    ADJ_FREAEA
;
;
;          * blink title *
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NO_ADJ:          LD      B,02H
                  CALL   SOUND2
                  CALL   SOUND2
BLKRUN:          CALL   CLRDISP
                  CALL   DELAY60
                  CALL   DELAY60
                  LD      A,00H
                  CALL   GOTO
                  LD      HL,F1D3L1
                  CALL   WR_LINE      ;disp. READY IN RUN STATE
                  CALL   SOUND2
                  CALL   DELAYF
                  DJNZ   BLKRUN
;
;
;
TEL_SCAN:        IN      A,(PORTC2)
                  RES     6,A
                  OUT    (PORTC2),A      ;motor off
                  IN      A,(PORTA2)
                  RES     0,A
                  OUT    (PORTA2),A      ;clr. nmi CK
                  LD      IY,B_STA2
                  LD      IX,TAPE_BUFF
                  LD      B,05H          ;count block
RST_SETB:        LD      A,(IX+0)
                  BIT     1,A
                  JR     Z,ZE_RO
                  LD      A,31H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      (IY+0),A      ;new rec sta. of block
JR      NEXT_B
ZE_RO:  LD      A,30H
LD      (IY+0),A      ;
NEXT_B: INC      IX
INC      IX
INC      IX
INC      IY
LD      A,(IX+0)
BIT     1,A
JR      Z,ZE_RO1
LD      A,31H
LD      (IY+0),A      ;new rec sta. of block
JR      NEXT_B1
ZE_RO1: LD      A,30H
LD      (IY+0),A      ;
NEXT_B1: INC      IX
INC      IX
INC      IX
INC      IY
LD      A,(IX+0)
BIT     1,A
JR      Z,ZE_RO2
LD      A,31H
LD      (IY+0),A      ;new rec sta. of block
JR      NEXT_B2
ZE_RO2: LD      A,30H
LD      (IY+0),A      ;
NEXT_B2: INC      IX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC    IX
INC    IX
INC    IY
INC    IY
DJNZ   RST_SETB
;
-----
LD     A,00
LD     (CLOCK2),A
LD     (TOGGLE),A
KEYIN2:
LD     C,PORTC1
LD     L,OFFH
LD     B,4
LD     D,03H
LD     E,11111110B
FROW2:
LD     H,01110000B
OUT    (C),E
IN     A,(PORTC1)
AND    H
CP     H
JP     NZ,FCOL2
LD     A,L
ADD    A,D
LD     L,A
RLC    E
DJNZ   FROW2
;
-----
IN     A,(PORTA2)
SET    0,A
OUT    (PORTA2),A    ;NMI on

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        IN      A,(PORTB1)
        BIT    4,A          ;ring sta. ACTV 'H'
        JR     NZ,GOTO_RNG
        CALL   DELAY10
        IN      A,(PORTB1)
        BIT    4,A
        JR     Z,SWAP_CHK
GOTO_RNG:  JP     RING_IN
SWAP_CHK:  NOP
          LD     A,(CLOCK2)
          SUB    2D
          JP     NC,CHG_PATRN
          CALL   DELAY
          LD     A,(CLOCK2)
          SUB    2D
          JR     NC,CHG_PATRN
          IN     A,(PORTA2)
          RES    0,A
          OUT   (PORTA2),A  ;nmi CK OFF
          JP     KEYIN2
;
          -----
CHG_PATRN: IN     A,(PORTA2)
          RES    0,A
          OUT   (PORTA2),A  ; NMI off
          LD     A,(TOGGLE)
          BIT    0,A
          JR     Z,B_STA
          CALL   CLRDISP
          LD     A,00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL GOTO
LD HL,F1D3L1 ;disp MAIL BOX TELEPPHONE
CALL WR_LINE
LD A,00
LD (TOGGLE),A
LD (CLOCK2),A
;
;
;
IN A,(PORTA2)
SET 0,A
OUT (PORTA2),A ;nmi CK ON
JP KEYIN2
B_STA:
CALL CLRDISP
LD A,00H
CALL GOTO
LD HL,B_STA1
CALL WR_LINE ;disp B1-B2---B5
LD A,40H
CALL GOTO
LD HL,B_STA2 ;disp subblock sta.
CALL WR_LINE
LD A,01H
LD (TOGGLE),A
LD A,00
LD (CLOCK2),A
JP KEYIN2
;
FCOL2:
CALL DELAY10
IN A,(PORTC1)
AND H
CP H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JP      Z,KEYIN2
        SRL    A
        SRL    A
        SRL    A
        SRL    A
        OR     A
FIND3:  INC    L
        RR     A
        JP     C,FIND3
        LD     A,L
        LD     HL,ASCIIT
        LD     BC,00H
        LD     C,A
        ADD   HL,BC
        LD     A,(HL)
        LD     (KBUFF1),A
        IN    A,(PORTA2)
        RES   0,A
        OUT   (PORTA2),A ;clr. nmi CK
        CALL  SOUND2 ;
        CALL  K_RELEASE
        JP    RST_WBOOT
;
;          -----
;          * Adjust to free area *
;          * Subroutine          *
;          -----
ADJ_FREAEA:  PUSH  AF
             PUSH  BC
             PUSH  DE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH  HL
IN    A,(PORTC2)
SET   6,A           ;motor run
OUT   (PORTC2),A
CALL  DELAY60
CALL  CLRDISP
CALL  DELAY60
LD    A,00
CALL  GOTO
LD    HL,F1D3L2     ;disp. Adjusting in execute
CALL  WR_LINE
;
CALL  BACK_ORG
CALL  DELAYF
CALL  FRE_AREA     ;O/P= reg A free area
CALL  MOV_TOAEA
;
CALL  CLRDISP     ;clear all disp.
CALL  DELAYF
POP   HL
POP   DE
POP   BC
POP   AF
RET
;
-----
F1D6L1:  DB    'Quick Answer or not?'
        END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; -----
; * NOW! we are start from ring was comming *
; -----
RING_IN:    CALL    DELAYF
            IN      A,(PORTB1)
            BIT     4,A           ;RING active "H"
            JP     NZ,RING_OK
            CALL    DELAY10
            IN      A,(PORTB1)
            BIT     4,A
            JP     Z,TEL_SCAN
RING_OK:    CALL    CLRDISP
            LD      A,00
            CALL    GOTO
            LD      HL,F1D4L1     ; ring active !
            CALL    WR_LINE
; -----
            LD      A,(Q_ANS)
            CP     11H
            JR     NZ,NML_ANS
            IN      A,(PORTA2)
            RES    0,A
            OUT    (PORTA2),A     ;clr. NMI ck
            LD      A,00
            LD      (CLOCK1),A
            CALL    DELAY10
            SET    0,A
            OUT    (PORTA2),A     ;en. NMI ck
TILL_7:    IN      A,(PORTB1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        BIT    4,A                ;RING active "H"
        JP     NZ,CHK_TME
        CALL  DELAY60.
        IN    A,(PORTB1)
        BIT    4,A
        JP     NZ,CHK_TME
        IN    A,(PORTB1)
        BIT    7,A                ;hang up active "L"
        JR     Z,HANG_UP
CHK_TME: LD    A,(CLOCK1)
        SUB   14D                ;wait hang up 7 sec
        JR     C,TILL_7
        JR     AUTO_OFHK
HANG_UP: IN    A,(PORTA2)
        RES   0,A
        OUT  (PORTA2),A        ;clr. NMI ck
        JP   TEL_SCAN        ;no run when hand set hang up
;
NML_ANS: IN    A,(PORTA2)
        RES   0,A
        OUT  (PORTA2),A        ;clr. NMI ck
        LD   A,00
        LD   (CLOCK1),A
        CALL DELAY10
        SET  0,A
        OUT  (PORTA2),A        ;en. NMI ck
TILL_25: IN    A,(PORTB1)
        BIT    4,A                ;RING active "H"
        JP     NZ,CHK_TME1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    DELAY60
IN      A,(PORTB1)
BIT     4,A
JP      NZ,CHK_TME1
IN      A,(PORTB1)
BIT     7,A           ;hang up active "L"
JR      Z,HANG_UP1
CHK_TME1:
LD      A,(CLOCK1)
SUB     50D           ;wait hang up 25 sec
JR      C,TILL_25
JR      AUTO_OFHK
HANG_UP1:
IN      A,(PORTA2)
RES     0,A
OUT     (PORTA2),A   ;clr. NMI ck
JP      TEL_SCAN     ;JMP when hang up (off hook)
;
AUTO_OFHK:
IN      A,(PORTC2)
SET     6,A
OUT     (PORTC2),A   ;motor run
IN      A,(PORTC2)
SET     4,A
OUT     (PORTC2),A   ;off hook
LD      A,40H
CALL    GOTO
LD      HL,F1D2L2     ;automatic off hook
CALL    WR_LINE
;
;
;
Aautomatic tel. line
record

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; -----
TELREC:      LD      A,00
             LD      (VCE_CNT),A      ;clr non strobe counter

SND_VCE1:   IN      A,(PORTA2)
             RES     0,A
             OUT     (PORTA2),A      ;disabe NMI ck
             LD      A,00H
             LD      (CLOCK3),A      ;clr non stb. time
             LD      A,00
             LD      (PLBNUM),A      ;voice block
             CALL    VCE_TTK          ;Send voice#0 to tel.line
                                     ;what do you want to do
                                     ;REC or PLB
; -----
TTK_UNPRSS: IN      A,(PORTA2)
             SET     0,A
             OUT     (PORTA2),A      ;enable NMI ck.
; -----
CHKSTRB:   IN      A,(PORTB1)
             BIT     5,A
             JR      NZ,FST_MEET
             CALL    DELAY10
             IN      A,(PORTA2)
             BIT     5,A
             JR      Z,ONSTB          ;strobe active "1"

FST_MEET:  IN      A,(DTMFP)          ;get DTMF value
             PUSH    AF

STB_RELS:  IN      A,(PORTB1)          ;
             BIT     5,A

```

```

JR      NZ,STB_RELS
;
-----
IN      A,(PORTA2)
RES     0,A
OUT     (PORTA2),A      ;disable NMI ck
LD      A,00
LD      (CLOCK3),A     ;clear count time #3
POP     AF
AND     0FH             ;clr 'H' bit
CP      0BH             ;for '*'
JR      Z,STAR         ;
CP      0CH             ;for '#'
JR      NZ,TELREC      ;
CALL    DOSHARP        ;do case of "#" <PLB>
JR      ENDF1
STAR:   CALL    DOSTAR  ;do case of "*" <REC>
JR      ENDF1
;
-----
NONSTB: LD      A,(CLOCK3) ;
SUB     20D             ;is stb loss=10sec. ?
JR      C,CHKSTRB     ;wait if stb. loss<10sec
;
-----
LD      HL,VCE_CNT
INC     (HL)
LD      A,03
CP      (HL)
JR      NZ,SND_VCE1   ;we send introduce voice
;for 3 times
IN      A,(PORTA2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RES    0,A
OUT    (PORTA2),A    ;NMI off
;
-----
ENDF1:  IN    A,(PORTC2)
RES    4,A
OUT    (PORTC2),A    ;on hook
JP     TEL_SCAN
;
+-----+
;      |   END OF FUNCTION 1   |
;      +-----+
;-----
;
+-----+
;      | perform case of key "*" |
;      | RECORD subroutine      |
;      +-----+
DOSTAR: PUSH  AF          ;record
        PUSH  BC
        PUSH  DE
        PUSH  HL
        LD   A,00
        LD   (VCE_CNT),A    ;clr strobe counter
NO_STBLOP: IN    A,(PORTA2)
RES    0,A
OUT    (PORTA2),A    ;disable NMI ck.
LD     A,00
LD     (CLOCK3),A    ;clr non stb. time
SND_VCE: LD     A,01H
        LD     (PLBNUM),A
        CALL  VCE_TTK    ;Send voice#1 to tel. line

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;block No. require,"*" if
;don't know

;
-----
IN    A,(PORTA2)
SET   0,A
OUT   (PORTA2),A    ;en. NMI
;
-----
CHK_STRB:    IN    A,(PORTB1)
             BIT   5,A
             JR   NZ,FSTMEET
             CALL  DELAY10
             IN   A,(PORTB1)
             BIT   5,A
             JR   Z, NON_STB    ;strobe active ("1") ?
FSTMEET:    IN   A,(DTMFP)
             PUSH  AF
UN_DTMF:    IN   A,(PORTB1)
             BIT   5,A
             JR   NZ,UN_DTMF    ;wait till key release
;
-----
IN    A,(PORTA2)
RES   0,A    ;disable NMI ck.
OUT   (PORTA2),A
LD    A,00
LD    (CLOCK3),A    ;clear count time #3
POP   AF
AND   0FH    ;clr 'H' bit
CP    0BH
JR   NZ,NO_HPRB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          NOP          ;high priority block REC '*'
          LD    A,01     ;select block#1
          JP    REC_PROCD
;
NON_STB:  LD    A,(CLOCK3) ;clock3 =time of loss stb.
          SUB   20D      ;wait 10sec
          JR    C,CHK_STRB ;wait if stb loss<10 sec
          LD    HL,VCE_CNT
          INC   (HL)
          LD    A,03
          CP    (HL)
          JP    NZ,NO_STBLOP ;wait till stb. loss
                                   ;10 sec. 3 times
          IN   A,(PORTA2)
          RES   0,A
          OUT  (PORTA2),A ;NMI CK off
          JP    FINALSTP
;
NO_HPRB:  LD    (TTKBUFF1),A
          SUB   06H
          JR    NC,NO_STBLOP ;accept key 1-5
;
;          +-----+
;          | record procedure for any block |
;          +-----+
;
REC_PROCD: LD    A,(TTKBUFF1)
          LD    (TP_FLG_B),A ;tempolary tape flag buffer
;
;          PUSH  AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          PUSH   DE
;          LD     A,10H
;          CALL  GOTO
;          LD     A,(TP_FLG_B)
;          ADD   A,30H
;          LD     D,A
;          CALL  WRBYTE
;          POP   DE
;          POP   AF
;          -----
;          CALL  CHK_STA_PSI    ;check sta.&position
;          CP    77H           ;is it full?
;          JP    Z,T_FULL1
;          PUSH  HL            ;
;          POP   IX           ;STA. byte of avilable
;          -----
;          IN    A,(PORTC2)
;          SET   6,A
;          OUT   (PORTC2),A    ;motor run
;          LD    A,04H
;          LD    (PLBNUM),A
;          CALL  SENDVOC       ;send voice#4 start REC
;          IN    A,(PORTA2)
;          SET   3,A           ;bit play
;          OUT   (PORTA2),A
;          CALL  DELAYF        ;avoid begin area
;          -----
;          LD    A,00
;          LD    (CLOCK1),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      (CLOCK3),A      ;clear time count
IN      A,(PORTA2)
OR      01000001B      ;en. NMI ck.& en. REC
OUT     (PORTA2),A      ;start record
LOOP_REC1:
LD      A,00
LD      (CLOCK2),A
;
-----
IN      A,(PORTB1)
BIT     5,A
JR      NZ,DTMF_GT
CALL    DELAY10
IN      A,(PORTB1)
BIT     5,A
JR      NZ,DTMF_GT
;
-----
IN      A,(PORTB1)
BIT     3,A      ;check voice loss
JR      NZ,CHK_2M1 ;voice active "1"
W_L10S:
LD      A,(CLOCK2) ;time of voice loss
SUB     20D      ;voice loss 10 sec?
JR      NC,REC_SUCS
IN      A,(PORTB1)
BIT     5,A
JR      NZ,DTMF_GT
CALL    DELAY10
IN      A,(PORTB1)
BIT     5,A
JR      Z,NO_MLOSS
DTMF_GT:
IN      A,(DTMFP)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND    0FH
CP     0BH
JR     NZ,NO_MLOSS
JR     END_F
REC_SUCS:
LD     A,(CLOCK1)
SUB    20D           ;subtrack by 10 Sec.
                   ;compensate waiting voice
                   ;loss
JR     NC,SAVE_TME ;allow save time for
                   ;REC time>10 sec.
LD     A,10D       ;set rec. time 5 sec if
                   ;REC time < 10 sec.
SAVE_TME:
LD     (CLOCK1),A  ;save real record time
JR     END_F
NO_MLOSS:
IN     A,(PORTB1)
BIT    3,A
JR     Z,W_L10S
CALL   DELAY10    ;debounce
IN     A,(PORTB1)
BIT    3,A
JR     Z,W_L10S
JR     LOOP_REC1
CHK_2M1:
LD     A,(CLOCK1)
SUB    200D        ;avilable REC 100 SEC
JR     C,LOOP_REC1
;
-----
END_F:
IN     A,(PORTA2)
AND    10111110B
OUT    (PORTA2),A  ;stop REC,disable NMI ck

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,05H
LD      (PLBNUM),A
CALL    SENDVOC      ;send voice#5 END REC
;
IN      A,(PORTC2)
;
RES     4,A
;
OUT     (PORTC2),A   ;on hook
CALL    DELAYF
LD      A,(IX+0)
SET     0,A          ;rec status full
SET     1,A          ;set don't read yet
LD      (IX+0),A    ;set new sta.
                        ;(subblock =4 bit MSB)
AND     0F0H        ;clr. 4 bits LSB.
LD      HL,TP_FLG_B
LD      B,(HL)      ;get block No. 4 bit LSB
OR      B
SET     7,A          ;sta. don't read yet
LD      HL,TP_FLAG  ;first of tape flag table
LD      DE,(TP_HEAD);real position of head
DEC     E
LD      D,00
ADD     HL,DE
LD      (HL),A      ;new sta. of TP_FLAG
LD      A,(CLOCK1)
LD      (IX+2),A    ;save time REC.
INC     E
LD      (IX+1),E    ;save position of data
;
-----
IN      A,(PORTA2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RES      3,A
OUT      (PORTA2),A      ;bit play stop
CALL     DELAYF
CALL     DELAYF
;
-----
;
move head to free area
-----
;
CALL     FRE_AREA      ;find out empty area
;
;
;
CALL     MOV_TOAEA     ;tape move to empty area
;
;
;
CALL     ADJ_FREAEA    ;i/p reg.A=Distination
JR       FINALSTP
;
-----
T_FULL1: LD      A,02H
LD      (PLBNUM),A
CALL     SENDVOC      ;send voice#2 tape full
CALL     DELAYF
FINALSTP: IN      A,(PORTC2)
AND     10111111B
OUT     (PORTC2),A    ;motor off
;
-----
;
end of record proceduer
-----
;
END_STAR: POP     HL
POP     DE
POP     BC
POP     AF
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          +-----+
;          | perform case of key "#" |
;          | PLAY BACK  subroutine  |
;          +-----+
DOSHARP:   PUSH  AF
          PUSH  BC
          PUSH  DE
          PUSH  HL
          LD   A,06
          LD   (PLBNUM),A
          CALL SENDVOC      ;vce# 6 'block number req'
;          -----
          LD   B,04H      ;count NUM of digit
          LD   HL,TTKBUFF1
OK_K:     LD   A,00
          LD   (CLOCK1),A
          IN   A,(PORTA2)
          SET  0,A
          OUT  (PORTA2),A      ;en. NMI ck
STRT_GET: IN   A,(PORTB1)
          BIT  5,A      ;chk STB.
          JR   NZ,MEET_FST
          CALL DELAY10
          IN   A,(PORTB1)
          BIT  5,A
          JR   Z,NO_STB1
MEET_FST: IN   A,(DTMFP)      ;get DTMF data
          AND  0FH      ;DTMF only 4 bits low
          CP   0AH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JR    NZ,NO_RECNF    ;reconfiguration DTMF
                .                    ;value from "0AH"to "00"
                LD    A,00
NO_RECNF:      ADD    A,30H          ;BIN to ASCII
                LD    (HL),A
                INC   HL             ;update pointer
REQ_L:         IN    A,(PORTB1)     ;
                BIT   5,A
                JR    NZ,REQ_L      ;key release
                IN    A,(PORTA2)
                RES   0,A
                OUT   (PORTA2),A    ;NMI CK off
                DJNZ OK_K
                JR    END_TTK
;
NO_STB1:      LD    A,(CLOCK1)
                SUB   20D           ;allow lose 10 sec
                JR    C,STRT_GET
                JP   TTK_ERR1      ;lose key error
;
END_TTK:     LD    B,01
                LD    IX,TTKBUFF1   ;buffer of touch tone key
                LD    IY,BCBUFF
ALL_CPR:     LD    A,(IX+0)
                LD    C,(IY+0)
                CP    C
                JR    NZ,MISS1
                LD    A,(IX+1)
                LD    C,(IY+1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP      C
JR      NZ,MISS1
LD      A,(IX+2)
LD      C,(IY+2)
CP      C
JR      NZ,MISS1
LD      A,(IX+3)
LD      C,(IY+3)
CP      C
JR      NZ,MISS1
JR      ALL_PASS
MISS1:  INC      IY
        INC      IY
        INC      IY
        INC      IY
        INC      B
        LD      A,06          ;chk. 5 block
        CP      B
        JR      NZ,ALL_CPR
        JP      TTK_ERR1     ;DON'T meet at all
;
ALL_PASS: LD      A,B          ;block number (1-5)
          LD      (PLB_BYTE1),A ;save block No. 4 bits LSB.
;
          CALL    REC_STA     ;check data in TAPE_BUFF
                                ; (A) =offset value
          CP      77H        ;o/p sign when empty
          JR      Z,NO_PLBK
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          check data in TP_FLAG
;          -----
LD        C,B
LD        B,0FH
LD        IY,TP_FLAG
CHKTPF_LP: LD        A,(IY+0)
AND       00001111B
CP        C
JR        Z,GET1_SBK
INC       IY
DJNZ     CHKTPF_LP
NO_PLBK:  LD        A,09H
LD        (PLBNUM),A
CALL     SENDVOC          ;#9 'no data in block'
JP       FINAL_PLB
;          -----
GET1_SBK: LD        (POINT_T),HL      ;save location of block
;          -----
ST_GET:   LD        A,07H
LD        (PLBNUM),A
CALL     SENDVOC          ;voice#7 subblock req.
;          -----
NRDY_GET: LD        A,00
LD        (CLOCK1),A
IN       A,(PORTA2)
SET      0,A
OUT     (PORTA2),A      ;en. NMI ck
STRT_GET1: IN       A,(PORTB1)
BIT      5,A            ;chk STB. DTMF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR      NZ,GETV_DTMF

CALL    DELAY10

IN      A,(PORTA2)

BIT     5,A

JR      Z,NO_STB2

;
-----
GETV_DTMF:  IN      A,(DTMFP)          ;get DTMF data
            AND     0FH
            LD      (TTKBUFF1),A      ;save to buffer

REQ_L1:    IN      A,(PORTB1)        ;
            BIT     5,A
            JR      NZ,REQ_L1        ;key release
            LD      A,(TTKBUFF1)
            SUB     04H               ;accept 1-3
            JR      NC,ST_GET        ;jmp if wrong subblock
            IN      A,(PORTA2)
            RES     0,A
            OUT     (PORTA2),A       ;disable NMI ck
            JR      BEGIN_PLB

;
-----
NO_STB2:   LD      A,(CLOCK1)
            SUB     20D               ;allow lose 10 sec
            JR      C,STRT_GET1
            JP      TTK_ERR1        ;lose key error

;
-----
BEGIN_PLB: LD      A,(TTKBUFF1)      ;subblock#n from DTMF
            ;bin. value
            PUSH    AF

;
-----

```

```

;          include subblock&block
;          and save to PLB_BYTE1
;          -----

SLA      A
SLA      A
SLA      A          ;shift subblock to
SLA      A          ;4 bits MSB
LD       B,A
LD       A,(PLB_BYTE1)
OR       B
LD       (PLB_B2),A
;          -----
POP      AF
PUSH    AF
SLA     A
SLA     A
AND     00001100B
PUSH    HL
LD      HL,(POINT_T) ;first byte of block
OR      (HL)
LD      (JUST_RD),A ;keep subblock that
;          ;just read to 1 st of
;          ;block (bit 2,3)

POP      HL
POP      AF
;          -----
DEC     A
LD      C,A
SLA     A          ;subblock*2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD    A,C           ;subblock*3
LD     HL,00
LD     L,A
LD     DE,(POINT_T) ;get location of block
                        ;(from PUSH HL)
ADD    HL,DE         ;location byte of BLOCK:sub
LD     A,(HL)        ;sta. byte of subblock
BIT    0,A
JR     Z,DS_PLB1     ;jmp. to don't PLB.if data
                        ;recorded empty.
;
; -----
; chk. data still be in
; tape or not
; -----
;
PUSH   HL
POP    IX
LD     B,0FH
LD     C,01H
LD     HL,TP_FLAG
LD     A,(PLB_B2)
RES    7,A

RE_CMPR1:
LD     B,(HL)
RES    7,B
CP     B
JR     Z,EN_PLB1
INC    HL
INC    C
DJNZ  RE_CMPR1

DS_PLB1:
LD     A,09H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      (PLBNUM),A
CALL    SENDVOC      ;voice#9 no data in block
JP      ST_GET
;
-----
EN_PLB1:
LD      A,C
LD      (DIS_TMP),A  ;save value of distination
PUSH    HL.          ;save position of TP_FLAG
                        ;(distination for mov. tape)
                        ;that must be clear read sta
                        ;after PLB.
;
-----
LD      A,(TP_HEAD)  ;DEBUG
LD      C,A          ;"
LD      A,(DIS_TMP)  ;"
CP      C            ;"
JR      Z,NO_BCKORG  ;"
LD      A,03
LD      (PLBNUM),A
CALL    SENDVOC      ;vce#3 please wait
IN      A,(PORTC2)
SET     7,A
OUT     (PORTC2),A   ;waiting music on
;
-----
CALL    BACK_ORG     ;park tape at ORG befor move
                        ;to data area
LD      A,10101010B
LD      (COND_MOV),A ;PLB flag
CALL    DELAY60      ;DEBUG
LD      A,(DIS_TMP) ;distination

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    MOV_TOAEA      ;mov tape to data area
NO_BCKORG:
IN      A,(PORTC2)
SET     6,A
OUT     (PORTC2),A    ;motor on
IN      A,(PORTC2)
RES     7,A
OUT     (PORTC2),A    ;waiting music off
;
;
;
start PLB now!
-----
IN      A,(PORTA2)
RES     1,A
OUT     (PORTA2),A    ;CHG. RY5 to NC. for
;send voce to Tel. line
CALL    DELAYF
IN      A,(PORTA2)
SET     3,A          ;bit play
OUT     (PORTA2),A
CALL    DELAYF      ;avoid org. area
CALL    DELAYF
CALL    DELAYF
SET     4,A          ;en RY6
OUT     (PORTA2),A
;
-----
LD      A,00
LD      (CLOCK1),A   ;clear time count
IN      A,(PORTA2)
SET     0,A          ;en. NMI ck
OUT     (PORTA2),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IN_PLB2:      LD      A,(CLOCK1)
              LD      B,(IX+2)
              SUB     B
              JR      C,IN_PLB2
;
              -----
              IN      A,(PORTA2)
              RES     4,A
              OUT     (PORTA2),A      ;enable digital voice
              CALL    DELAY10
              LD      A,05H
              LD      (PLBNUM),A
              CALL    SENDVOC      ;vce#5 plb. time finish
              IN      A,(PORTA2)
              RES     4,A      ;RY6 off
              OUT     (PORTA2),A
              CALL    DELAYF
;
;
;
              IN      A,(PORTC2)
              RES     4,A
              OUT     (PORTC2),A      ;on hook
              IN      A,(PORTA2)
              AND     11110110B
              OUT     (PORTA2),A      ;bit play,NMI off
              LD      A,(IX+0)
              RES     1,A      ;set ready read
              LD      (IX+0),A
              POP     HL
              LD      A,(HL)
              RES     7,A
              LD      (HL),A      ;res. bit ready read

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;in TP_FLAG table

LD      A,(JUST_RD)
LD      (POINT_T),A
;
;-----
;
;      move head to free area
;-----
;
;      CALL      FRE_AREA      ;find out empty area
;
;                                ;o/p reg.A=No.of area
;
;      CALL      MOV_TOAEA     ;tape move to empty area
;
;                                ;i/p reg.A=Distination
;
;      CALL      ADJ_FREAEA
;
;      JR        FINAL_PLB
;-----
TTK_ERR1: LD      A,08H
          LD      (PLBNUM),A
          CALL    SENDVOC      ;VCE#8 pressing code error
;-----
FINAL_PLB: IN      A,(PORTA2)
          RES     0,A
          OUT    (PORTA2),A    ;OFF NMI
          IN      A,(PORTC2)
          AND    10111111B    ;motor off
          OUT    (PORTC2),A
          POP    HL
          POP    DE
          POP    BC
          POP    AF
          RET
;-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

; -----
; * move tape to any area *
; * I/P reg A = distination *
; -----
;
MOV_TOAEA:      PUSH   BC
                 PUSH   HL
                 LD     (TEM_HEAD),A    ;save distination
                                           ;to temporary buffer
; -----
LD     C,A
LD     HL,TP_HEAD
LD     B,(HL)    ;head position
CP     B
JR     NZ,WORK
LD     A,(COND_MOV)
CP     10101010B ;check condition of moving
                                           ;move for PLB or prepair REC
JP     Z,PASS_MOV ;no moving if PLB
LD     A,01H    ;case of HP=DIST. we assign
                                           ;the value of sync for REV=1
LD     (DIS_MOV),A
JR     H_EQU_DST
; -----
; CALL   SOUND1
; CALL   DELAYF
; CALL   DELAYF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          CALL    SOUND1
;          JP      PASS_MOV
;          -----
WORK:      SUB     B          ;dis.-head
          JR      C,HP_M_DP
                                     ;distination>head position
                                     ;must be forward
          LD     (DIS_MOV),A    ;save new dist. for mov
          LD     A,00
          LD     (DIR_FLG),A    ;forward flag "00"
          JR      FWDD
;          -----
;HP_M_DP:  NOP             ;head position>distination
                                     ;must be reverse
HP_M_DP:   LD     A,(TP_HEAD)
          SUB     C          ;head-dist.
          INC    A          ;compensate 1 sync
          LD     (DIS_MOV),A    ;save dist. for move
;          -----
H_EQU_DST: LD     A,OFFH
          LD     (DIR_FLG),A    ;reverse flag "OFF"
          IN     A,(PORTA2)
          SET    3,A          ;set bit play
          RES    5,A          ;clear bit lock(en.REW)
          OUT   (PORTA2),A
          CALL   DELAYF
          SET    2,A          ;set bit speed
          OUT   (PORTA2),A    ;tape reverse
          CALL   DELAY10      ;DEBUG

```

```

JR      RUN_FND
;
-----
FWDD:   IN      A, (PORTA2)
        SET     5,A           ;FWD unlocked
        OUT     (PORTA2),A
        CALL    DELAYF
        SET     3,A           ;set bit play
        OUT     (PORTA2),A
        CALL    DELAYF
        SET     2,A           ;bit speed
        OUT     (PORTA2),A   ;tape forward
        CALL    DELAY10
;
-----
RUN_FND: LD     B,00
REP_FND: IN     A, (PORTB1)   ;get sync pulse of block
        BIT     1,A
        JR     Z,REP_FND     ;wait for sync active "1"
IN_SYNC: IN     A, (PORTB1)   ;wait for end of sync pulse
        BIT     1,A
        JR     NZ,IN_SYNC
;
-----
        CALL    SOUND2       ;DEBUG
;
-----
        INC     B
        LD     A, (DIS_MOV)   ;position value
        CP     B
        JR     NZ,REP_FND
;
-----
        CALL    DELAY60      ;DEBUG away from sync area

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL DELAY60
CALL DELAY60
CALL DELAY60
IN A, (PORTA2)
RES 2,A
OUT (PORTA2),A ;clr bit speed
CALL DELAY10
AND 11010111B
OUT (PORTA2),A ;clr. bit lock,play
;
-----
LD A, (DIR_FLG)
CP 00H
JR Z,NO_READJ
CALL DELAY60
CALL DELAYF
;
;
;
;
;
;
IN A, (PORTA2)
SET 5,A ;FWD unlocked
OUT (PORTA2),A
CALL DELAYF
SET 3,A ;set bit play
OUT (PORTA2),A
CALL DELAYF
SET 2,A ;bit speed
OUT (PORTA2),A ;tape forward
CALL DELAY10

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

W_SNCUP:      IN      A, (PORTB1)
               BIT     1,A
               JR     Z,W_SNCUP

W_SNCDWN:     IN      A,(PORTB1)
               BIT     1,A
               JR     NZ,W_SNCDWN

               CALL    SOUND2          ;DEBUG
               CALL    DELAY60         ;away from sync
               CALL    DELAY60
               CALL    DELAY60
               CALL    DELAY60
               IN      A, (PORTA2)
               RES     2,A
               OUT     (PORTA2),A      ;clr. bit speed
               CALL    DELAYF
;
NO_READJ:     IN      A, (PORTA2)
               RES     3,A
               OUT     (PORTA2),A      ;clr bit play
               CALL    DELAYF
               RES     5,A             ;clr bit lock
               OUT     (PORTA2),A
               CALL    DELAYF

PASS_MOV:     LD      A,00001111B
               LD      (COND_MOV),A    ;clear MOV PLB flag
               IN      A, (PORTC2)
               RES     6,A             ;motor off
               OUT     (PORTC2),A
               LD      A, (TEM_HEAD)

```

```

LD      (TP_HEAD),A      ;save new position of head
POP     HL
POP     BC
RET

;
; -----
; * find position of tape *
; * I/P: IX =point to sta. byte *
; * bit#2 of reg A=direction *
; * 0=FWD/1=REV subroutine *
; -----
;FIND_TAPE:
PUSH   AF
PUSH   BC
PUSH   DE
BIT    2,A               ;check direction
JR     Z,FWD
IN     A,(PORTA2)
SET    3,A               ;set bit play
RES    5,A               ;clr bit lock fwd
OUT    (PORTA2),A
CALL   DELAY10
SET    2,A               ;set bit speed
OUT    (PORTA2),A       ;tape reverse
JR     RUN_FIND
;;
;FWD:
IN     A,(PORTA2)
SET    5,A               ;set bit lock fwd
OUT    (PORTA2),A
CALL   DELAYF
SET    3,A               ;set bit play

```

```

;          SET    2,A          ;bit speed
;          OUT    (PORTA2),A    ;tape forward
;
;-----
;RUN_FIND: LD    B,00
;REP_FIND: IN    A,(PORTB1)    ;get sync pulse of block
;          BIT    1,A
;          JR    Z,REP_FIND    ;wait for sync active "1"
;IN_SYN:   IN    A,(PORTB1)    ;wait for end of sync pulse
;          BIT    1,A
;          JR    NZ,IN_SYN
;          INC   B
;          LD    A,(IX+1)      ;position value
;          CP    B
;          JR    NZ,REP_FIND
;          IN    A,(PORTA2)
;          AND   11010011B    ;clr bit play,speed,lock
;          OUT   (PORTA2),A
;          POP   HL
;          POP   DE
;          POP   BC
;          POP   AF
;          RET
;
;-----
;          * find subblock sta. & position *
;          * Subroutine . *
;          * i/p reg A =block No.[01-05] *
;          * o/p reg A =77H if block full *
;          * o/p [HL] =Point to empty subblock *
;-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

TEST2_STA:      LD      A, (HL)
                BIT      1, A
                JR      NZ, CHK_NEXT
;
;
;
                -----
                check just read area ?
                -----
                LD      DE, (TEM_PNT) ;get last read sub.
                LD      A, (DE)
                SLA     A
                SLA     A
                AND     00110000B
                LD      C, A
                LD      A, (HL)
                AND     00110000B
                CP      C
                JR      NZ, MEET_PNT
                LD      (AVI_SUB), HL ;the last just read one was
;the last one that can be
;recorded so must be used.
                LD      A, 01010101B ;set ready sign
                LD      (RD_SIGN), A
CHK_NEXT:      INC     HL
                INC     HL
                INC     HL
                DJNZ   TEST2_STA
                LD      A, (RD_SIGN)
                CP      01010101B
                JR      NZ, ALL_FULL
                LD      HL, (AVI_SUB)

```

```

;          JR      MEET_PNT
ALL_FULL: LD      A,77H          ;block full sta.
MEET_PNT: POP     DE
          POP     BC          ;now we get 1 st. byte
;                                     ;of subblock that can REC.
;                                     ;and keep in HL

```

```
RET
```

```

;
; -----
; * send voice Instruction *
; * Tel. line Subroutine *
; -----
SENDVOC:  PUSH    AF
          PUSH    BC
          PUSH    DE
          PUSH    HL
;
; -----
;          set channel for PLB
; -----
          LD      A,00010000B
          OUT     (PORTA1),A    ;init. T6668 SW
          CALL    DELAY60
          IN      A,(PORTA1)
          RES     4,A
          OUT     (PORTA1),A
          CALL    DELAY60
          IN      A,(PORTA1)
          SET     4,A
          OUT     (PORTA1),A
          CALL    DELAY60

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;

```

-----
IN      A, (PORTA2)
RES     1,A
OUT     (PORTA2),A   ;CHG. RY5 to NC.
                                ;send voice to Tel. line

```

;

```

-----
LD      A, (PLBNUM)
AND     0FH
SET     4,A
OUT     (PORTA1),A   ;set channel
CALL    DELAY60
SET     7,A
OUT     (PORTA1),A   ;start PLB
CALL    DELAY60
IN      A, (PORTA1)
RES     7,A
OUT     (PORTA1),A
SNDVLOP:
IN      A, (PORTB1)
BIT     0,A
JR      Z, SNDVLOP   ;T6668 bit status
IN      A, (PORTA2)
SET     1,A
OUT     (PORTA2),A   ;CHG. RY5 to NO. for
                                ;brake voice to Tel. line

POP     HL
POP     DE
POP     BC
POP     AF

RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; -----
; * send voice & check key *
; * DTMF (no get KEY value) *
; * o/p reg A =77 for key press *
; -----
VCE_TTK:      PUSH  BC
               PUSH  DE
               PUSH  HL
;
;             -----
;             set channel for PLB
;             -----
LD      A,00010000B
OUT     (PORTA1),A      ;init. T6668 SW
CALL    DELAY60
IN      A,(PORTA1)
RES     4,A
OUT     (PORTA1),A
CALL    DELAY60
IN      A,(PORTA1)
SET     4,A
OUT     (PORTA1),A
CALL    DELAY60
;
; -----
IN      A,(PORTA2)
RES     1,A
OUT     (PORTA2),A      ;CHG. RY5 to NC. for
;send voice to Tel. line
;
; -----
LD      A,(PLBNUM)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND    OFH
SET    4,A
OUT    (PORTA1),A    ;set channel
CALL   DELAY60
SET    7,A
OUT    (PORTA1),A    ;start PLB
CALL   DELAY60
IN     A,(PORTA1)
RES    7,A
OUT    (PORTA1),A
-----
;
VCE_LP:
IN     A,(PORTB1)
BIT    5,A
JR     Z,NO_TTK
IN     A,(PORTA1)
SET    6,A
OUT    (PORTA1),A
CALL   DELAY60
RES    6,A
OUT    (PORTA1),A    ;pulse for stop send voice
JR     TTK_PRSS

NO_TTK:
IN     A,(PORTB1)
BIT    0,A
JR     Z,VCE_LP      ;T6668 bit status

TTK_PRSS:
IN     A,(PORTA2)
SET    1,A
OUT    (PORTA2),A    ;CHG. RY5 to NO. for
                        ;Brake voice to Tel. line

POP    HL

```

```

POP     DE
POP     BC
RET

; -----
; * move tape to ORG. *
; -----
BACK_ORG:  PUSH  AF
           PUSH  BC
           PUSH  DE
           PUSH  HL
           LD   A,00
           LD   (CLOCK1),A
           LD   (CLOCK2),A
           IN   A,(PORTC2)
           SET  6,A
           OUT  (PORTC2),A      ;motor run
           IN   A,(PORTA2)
           SET  2,A
           OUT  (PORTA2),A      ;tape REV.
           CALL DELAYF
           IN   A,(PORTA2)
           SET  0,A
           OUT  (PORTA2),A      ;NMI on
GET_PULSE: IN   A,(PORTB1)      ;get pulse of tape move
           BIT  2,A
           JR   NZ,TSK_H
           LD   A,00
           LD   (CLOCK1),A      ;clear count of pulse Z
           LD   A,(CLOCK2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SUB      2D
JR      NC,TP_END
JR      GET_PULSE
TSK_H:  LD      A,00
LD      (CLOCK2),A
LD      A,(CLOCK1)
SUB      2D
JR      C,GET_PULSE
TP_END  IN      A,(PORTA2)
RES     0,A
OUT     (PORTA2),A      ;NMI off
IN      A,(PORTA2)
RES     2,A
OUT     (PORTA2),A      ;rev stoped
LD      A,00
LD      (TP_HEAD),A      ;real position of head=0
CALL    DELAYF
;
POP     HL
POP     DE
POP     BC
POP     AF
RET
;
F1D1L1: DB      'Funct.1 system RUN '
F1D2L1: DB      ' tape adj. or not? '
F1D3L1: DB      '<MAIL BOX TELEPHONE>'
F1D1L2: DB      ' press 1=YES/0=NO '
F1D4L1: DB      ' ! RING ACTIVE ! '

```

F1D2L2: DB 'automatic off hook !'
 F1D3L2: DB 'Adjust to free area '
 B_STA1: DB '-B1--B2--B3--B4--B5 '
 END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; * FUNCTION 2 program block code *
; * & change new pass word *
; *****
FUNCT2: LD A,00 ;start funct 2
CALL GOTO
LD HL,TABSIX1 ;disp. func2 program code
CALL WR_LINE
CALL CHKPSW
CALL SOUND2 ;this point pass word OK!
LD A,00H
CALL GOTO
LD HL,PSWOK1
CALL WR_LINE ;disp. pass word ok!
LD A,0EH
CALL GOTO
LD HL,PSWBUFF
CALL WR_LINE ;disp. old pass word
CALL NEWPSW ;set new pass word routine
CALL NEWBC ;set new block code
JP TEL_SCAN ;end of funct. 2
; *****
; end of main function2
; *****
; -----
; * GET NEW BLOCK CODE ROUTINE *
; -----
NEWBC: PUSH AF
PUSH BC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH    DE
PUSH    HL
PUSH    IX
PUSH    IY

LD      IX,TBC1      ;block code# n. code(XXXX)
LD      IY,TBC2      ;fill new code:
LD      DE,BCBUFF    ;block code buffer 20 byte
LD      B,05H        ;count No. of block code

NEW_BC:
LD      A,00
CALL    GOTO
PUSH    IX
POP     HL
CALL    WR_LINE      ;disp. new block code line1
LD      A,0EH
CALL    GOTO          ;set disp. old block code
PUSH    DE
POP     HL           ;get block code buff table
CALL    WR_LINE      ;disp. block code:XXXX
LD      A,12H
CALL    GOTO
LD      D,29H        ;
CALL    WRBYTE       ;
LD      D,20H        ;mark end line ") "
CALL    WRBYTE       ;
LD      A,40H
CALL    GOTO
PUSH    IY
POP     HL
CALL    WR_LINE      ;disp. fill new code

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    GET_4DG      ;set new code
LD      A, (KBUF4)
CP      3EH          ;
JR      Z,NOSVE     ;ENTER key for don't change
;
;
-----
* move block code from key buff. *
* to block code buff.           *
* DE=distination,HL=source,BC=long *
-----
PUSH    BC
PUSH    DE
LD      BC,0004H    ;
LD      HL,KBUF4    ;
LDIR
POP     DE
POP     BC
NOSVE:  INC     DE
        INC     DE
        INC     DE
        INC     DE      ;up date block code buff
;
-----
inc table for disp. 20 byte
-----
PUSH    DE
PUSH    IX
POP     HL
LD      DE,14H
ADD     HL,DE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH  HL
POP   IX
PUSH  IY
POP   HL
ADD   HL,DE
PUSH  HL
POP   IY
POP   DE
DJNZ  NEW_BC
;
CALL  SOUND2
LD    A,00H
CALL  GOTO
LD    HL,CLRL
CALL  WR_LINE      ;clear all display
LD    A,40H
CALL  GOTO
LD    HL,CLRL
CALL  WR_LINE
CALL  DELAYF
;
LD    A,00H
CALL  GOTO
LD    HL,ENDF2
CALL  WR_LINE      ;display end of function2
CALL  DELAYF
CALL  DELAYF
CALL  DELAYF
POP   IY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP      IX
POP      HL
POP      DE
POP      BC
POP      AF
RET

TBC1:    DB      'Block #1 CODE(      '
          DB      'Block #2 CODE(      '
          DB      'Block #3 CODE(      '
          DB      'Block #4 CODE(      '
          DB      'Block #5 CODE(      '
TBC2:    DB      'fill new code:      '
          DB      'fill new code:      '
          DB      'fill new code:      '
          DB      'fill new code:      '
          DB      'fill new code:      '
ENLINE:  DB      ') '
ENDF2:   DB      'Function#2 complete.'
TPSWERR: DB      ' PASSWORD ERROR ! '
PSWOK2:  DB      'new password :      '
PSWOK1:  DB      'PASSWORD OK! (      '
TABSIX1: DB      'Funct.2 Program code'
PSWREQ:  DB      'password is :      '
;
;
;
FUNCT3:  LD      A,00
          CALL   GOTO
          LD      HL,F3D1L1      ;function3 instruction

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL WR_LINE
LD A,40H
CALL GOTO
LD HL,F3D1L2 ;voice recording
CALL WR_LINE
CALL DELAYF
CALL DELAYF
;
-----
CALL CHKPSW ;check pass word
CALL DELAYF
CALL SOUND2
CALL DELAYF
LD A,00
CALL GOTO
LD HL,F3D2L1 ;digital record #
CALL WR_LINE
LD B,0AH ;count for rec. No.
LD D,30H ;start rec. No. 0
REC_N: LD A,11H ;cursor disp. rec.No.
CALL GOTO
CALL WRBYTE
LD A,40H
CALL GOTO
LD HL,F3D2L2 ;disp start/stop rec.
CALL WR_LINE
WAITST: CALL K_RELEASE
CALL SCAN
LD A,(Kbuff1)
CP 3EH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR      NZ, WAITST
CALL    DREC                      ;perform recording
INC     D
DJNZ    REC_N
JP      RST_WBOOT
;
;      *****
;      END OF FUNCTION3
;      *****
F3D1L1: DB      'Funct.3 Instruction '
F3D1L2: DB      'voice recording '
F3D2L1: DB      'Record to CH. # '
F3D2L2: DB      'ENTER for start/stop'
F3D3L2: DB      'to be recording '
F3D4L2: DB      'record stoped '
F3D5L2: DB      'MEMORY FULL!'
;
;      * PERFORM RECORDING ROUTINE *
;
DREC:   PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL

        LD      A, 00010000B
        OUT    (PORTA1), A          ;initial SW. T6668
        LD      A, 0FH
        AND    D                    ;D=ASCII channel
        SET    4, A                 ; no reset
        OUT    (PORTA1), A          ;set channel
        CALL   DELAY60

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
;
;
-----
start record
-----

SET 5,A ;REC

OUT (PORTA1),A

CALL DELAY60

SET 7,A

OUT (PORTA1),A ;start REC

CALL DELAY60

-----
;
RE_REC:
LD A,40H
CALL GOTO
LD HL,F3D3L2 ;disp. to be recording
CALL WR_LINE
CALL K_RELEASE
CALL SCAN1
LD A,(KBUF1)
CP 3EH ;ENTER for stop REC
JR Z,ENDREC
CP 20H ; REC status off
JR Z,NOREC ;
JR RE_REC
JR ENDREC

NOREC:
LD A,40H
CALL GOTO
LD HL,F3D5L2 ;disp. memory full
CALL WR_LINE
CALL SOUND2
CALL DELAYF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    DELAYF
CALL    DELAYF
JR      PASS1
ENDREC: IN    A, (PORTA1)
        SET   6, A
        OUT   (PORTA1), A      ;Stop REC
        CALL  DELAY60
        RES   6, A
        OUT   (PORTA1), A
PASS1:  POP   HL
        POP   DE
        POP   BC
        POP   AF
        RET
        END

```



```

; *****
; * FUNCTION 4 digital record play back *
; *****
FUNCT4: LD A,00
        CALL GOTO
INPLB: LD HL,F4D1L1 ;disp. Funct.4 play back
        CALL WR_LINE
GETNUM: LD A,40H
        CALL GOTO
        LD HL,F4D1L2 ;disp. voice instruction.
        CALL WR_LINE
        LD A,53H
        CALL GOTO ;set cursor for wait key
        CALL K_RELEASE
        CALL SCAN
W_RET: LD A,(KBUFF1)
        LD (PLBNUM),A ;save play back No.
        LD D,A
        CALL WRBYTE ;disp. key pressed
        LD B,A
        LD A,39H ;high of channel
        SUB B
        JR C,GETNUM
        CALL K_RELEASE ;
        LD A,53H
        CALL GOTO
        CALL SCAN
        LD A,(KBUFF1)
        CP 3EH ;wait ENTER key

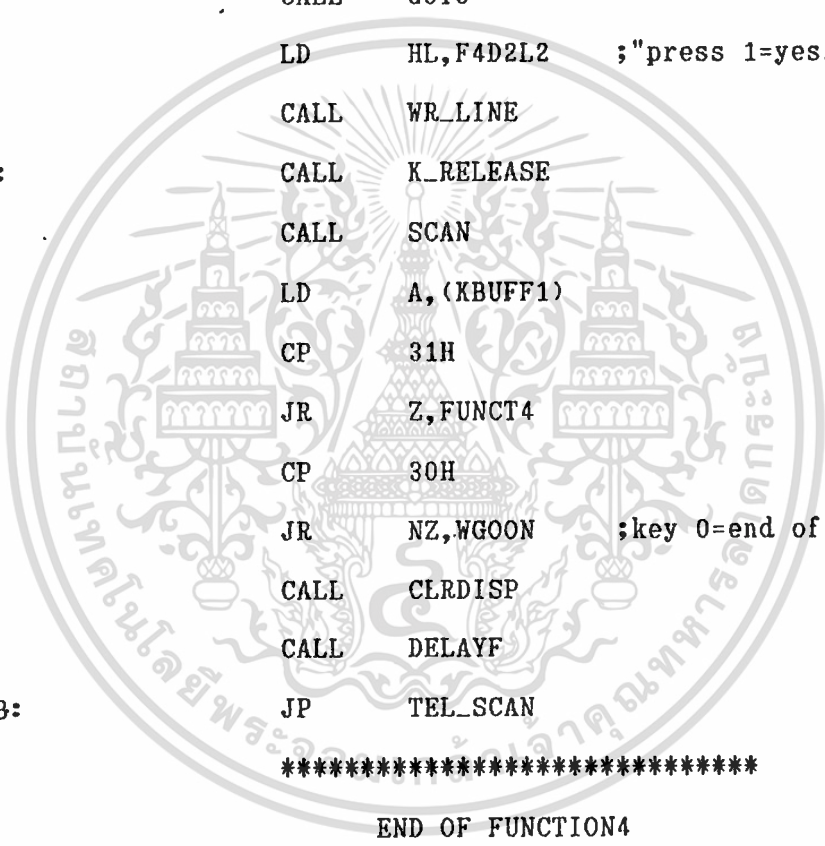
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                JR      NZ,W_RET
RUN_PLB:                       CALL    DPLB          ;digital voice PLB
                                LD      A,00
                                CALL    GOTO
                                LD      HL,F4D2L1      ;"do you want to go on"
                                CALL    WR_LINE
                                LD      A,40H
                                CALL    GOTO
                                LD      HL,F4D2L2      ;"press 1=yes/0=no"
                                CALL    WR_LINE
WGOON:                          CALL    K_RELEASE
                                CALL    SCAN
                                LD      A,(KBUFF1)
                                CP      31H
                                JR      Z,FUNCT4
                                CP      30H
                                JR      NZ,WGOON       ;key 0=end of play back
                                CALL    CLRDISP
                                CALL    DELAYF
ENDPLB:                         JP      TEL_SCAN
;                               *****
;                               END OF FUNCTION4
;                               *****
F4D1L1:                         DB      'Funct.4 Playback '
F4D1L2:                         DB      'instruction voice#:'
F4D2L1:                         DB      'Do you want to go on'
F4D2L2:                         DB      'Press 1=YES/0=NO '
;
;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          * perform instruction voice *
;          * playback                    *
;          -----
DPLB:      PUSH    AF

           PUSH    BC

           PUSH    DE

           PUSH    HL

;          -----
;          set channel for PLB
;          -----
LD        A,00010000B
OUT       (PORTA1),A      ;init. T6668 SW
CALL     DELAY60
IN       A,(PORTA1)
RES      4,A
OUT      (PORTA1),A
CALL     DELAY60
IN       A,(PORTA1)
SET      4,A
OUT      (PORTA1),A
CALL     DELAY60
LD        A,(PLBNUM)

;          -----

AND      0FH
SET      4,A
OUT      (PORTA1),A      ;set channel

;          -----

CALL     DELAY60
SET      7,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUT    (PORTA1),A      ;start PLB
CALL   DELAY60
IN     A,(PORTA1)
RES    7,A
OUT    (PORTA1),A
;
-----
IN     A,(PORTA2)
SET    1,A
OUT    (PORTA2),A      ;on speaker
LD     A,00
CALL   GOTO
LD     HL,F4D3L1
CALL   WR_LINE        ;disp. OK!NOW! play back
PLBLOOP: IN  A,(PORTB1)
BIT    0,A
JR     Z,PLBLOOP      ;T6668 bit status
;
-----
CALL   CLRDISP
CALL   DELAYF         ;clear all display
;
-----
IN     A,(PORTA2)
RES    1,A
OUT    (PORTA2),A      ;off speaker
LD     A,00
CALL   GOTO
LD     HL,F4D4L1      ;Send voice seccessed
CALL   WR_LINE
CALL   DELAYF
CALL   DELAYF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    DELAYF
POP     HL
POP     DE
POP     BC
POP     AF
RET
F4D3L1: DB    'OK! now in Playback '
F4D4L1: DB    'Send voice succeeded'
;
;
;
;
FUNCT5: CALL   CLRDISP
        LD    A,00
        CALL  GOTO
        LD    HL,F5D1L1    ;disp. Funct.5 manual rec
        CALL  WR_LINE
        CALL  DELAYF
ERR_KEY: CALL  SOUND2
        LD    A,40H
        CALL  GOTO
        LD    HL,F5D1L2    ;disp. Enter block No(1-5)
        CALL  WR_LINE
        LD    A,53H
        CALL  GOTO
        CALL  SCAN
        LD    A,(KBUFF1)
        CALL  WRBYTE
        CP    30H        ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR      Z,ERR_KEY      ;avoid key "0"
SUB     36H             ;block No. must be 1-5
JR      NC,ERR_KEY     ;
LD      A,(KBUFF1)
AND     0FH            ;ASCII to BIN
LD      (TP_FLG_B),A   ;tempolary tape flag buffer
CALL    CHK_STA_PSI    ;check sta.&position
CP      77H            ;is it full?
JP      Z,T_FULL
PUSH    HL              ;
POP     IX              ;STA. byte of avilable
;-----
IN      A,(PORTC2)
SET     6,A
OUT     (PORTC2),A     ;motor run
LD      A,40H
CALL    GOTO
LD      HL,CLRL
CALL    WR_LINE        ;clear disp.
CALL    DELAY60
LD      A,40H
CALL    GOTO
LD      HL,F5D5L2      ;press ENTER to start
CALL    WR_LINE
W_REC:  CALL    K_RELEASE
        CALL    SCAN
        LD      A,(KBUFF1)
        CP      3EH
        JR      NZ,W_REC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IN      A,(PORTA2)
SET     3,A          ;bit play
OUT     (PORTA2),A
CALL    DELAYF      ;avoid begin area
LD      A,40H
CALL    GOTO
LD      HL,F5D4L2   ;disp."rec. in execution"
CALL    WR_LINE
IN      A,(PORTC2)
SET     4,A          ;off hook
RES     7,A          ;DEBUG music distb.
OUT     (PORTC2),A ;
;
LD      A,00
LD      (CLOCK1),A
LD      (CLOCK3),A ;clear time count
IN      A,(PORTA2)
OR      01000001B   ;en. NMI ck.& en. REC
OUT     (PORTA2),A ;DO record
LOOP_REC:
LD      A,00
LD      (CLOCK2),A
;
-----
IN      A,(PORTB1)
BIT     5,A
JR      Z,NO_BRK
IN      A,(DTMF)
AND     0FH
CP      0BH          ;check key '*' for
                        ;stop REC.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR      Z,END_F5
;
-----
NO_BRK: IN      A,(PORTB1)
        BIT     3,A           ;check voice loss
        JR     NZ,CHK_2M     ;voice active "1"
W_TEN:  LD     A,(CLOCK2)    ;time of voice loss
        SUB    20D          ;voice loss 10 sec?
        JR     C,UN_LOSS
        LD     A,(CLOCK1)
        SUB    20D
        LD     (CLOCK1),A
        JR     END_F5
UN_LOSS: IN     A,(PORTB1)
        BIT     3,A
        JR     Z,W_TEN
        CALL   DELAY10      ;debounce
        IN     A,(PORTB1)  ;
        BIT     3,A
        JR     Z,W_TEN
        JR     LOOP_REC
CHK_2M: LD     A,(CLOCK1)
        SUB    200D         ;available REC 100 SEC
        JR     C,LOOP_REC
;
-----
END_F5: LD     A,(IX+0)
        SET    0,A         ;rec status full
        SET    1,A         ;set don't read yet
        LD     (IX+0),A    ;set new sta.
                                ;(subblock =4 bit MSB)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND    OFOH           ;clr. 4 bits LSB.
LD     HL,TP_FLG_B
LD     B,(HL)         ;get block No. 4 bit LSB
OR     B
SET    7,A            ;sta. don't read yet
LD     HL,TP_FLAG     ;first of tape flag table
LD     DE,(TP_HEAD)   ;real position of head
DEC    E              ;TP_HEAD start count
                          ;from 1 so must be reduce
                          ;by one
LD     D,00
ADD    HL,DE
LD     (HL),A         ; new sta. of TP_FLAG
LD     A,(CLOCK1)
LD     (IX+2),A       ;save time REC.
INC    E              ;TP_HEAD count. corrected
LD     (IX+1),E       ;save position of data
;
;
IN     A,(PORTA2)
AND    10110110B
OUT    (PORTA2),A     ;disable NMI ck.&stop REC
CALL   DELAYF
CALL   DELAYF
;
;
;
;
;
CALL   FRE_AREA       ;find out empty area
;
;
;
CALL   MOV_TOAEA      ;tape move to empty area

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;                                     ;i/p reg.A=Distination
CALL ADJ_FREAEA
JR FINALPSS
T_FULL: CALL CLRDISP
CALL DELAYF
LD A,00H
CALL GOTO
LD HL,F5D3L2 ;sorry ! Block full
CALL WR_LINE
CALL SOUND2
CALL DELAY60
CALL SOUND2
CALL DELAYF
CALL DELAYF
FINALPSS: IN A,(PORTC2)
AND 10101111B ;motor off, on hook
OUT (PORTC2),A
CALL DELAYF
JP TEL_SCAN
; *****
; * END OF FUNCT. 5 *
; *****
F5D1L1: DB 'Funct5 manual record'
F5D1L2: DB 'Enter bock No.(1-5) '
F5D2L2: DB 'Please wait ! '
F5D3L2: DB 'sorry ! Block Full '
F5D4L2: DB 'record in execution '
F5D5L2 DB 'press ENTER to start'
; *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          * FUNCTION 6 tape play back      *
;          * to speaker                      *
;          *****
FUNCT6:    CALL    CLRDISP

           LD      A,00

           CALL    GOTO

           LD      HL,F6D1L1      ;" Funct.6 manual PLB."

           CALL    WR_LINE

           CALL    DELAYF

           CALL    SOUND2

           LD      A,40H

           CALL    GOTO

           LD      HL,F6D1L2      ;disp. block code 7:

           CALL    WR_LINE

           LD      A,4DH

           CALL    GOTO

           CALL    PASWD      ;GET block code

           LD      A,(D_NUM)

           CP      04H

           JR      Z,NUM_OK

MISS_CODE: CALL    CLRDISP

           CALL    DELAY60

           CALL    SOUND2

           LD      A,00

           CALL    GOTO

           LD      HL,F6D3L1      ;BLOCK CODE ERROR !

           CALL    WR_LINE

           CALL    DELAY10

           CALL    SOUND2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    DELAY60
CALL    DELAYF
CALL    DELAYF
JP      STOP_PLB

;      +-----+
;      check 4 digit block code
;      -----

NUM_OK: LD    B,01          ;start count 5 block cmp.
        LD    IX,KBUFF10
        LD    IY,BCBUFF
BLOCK_CPR: LD    A,(IX+0)
          LD    C,(IY+0)
          CP    C
          JR    NZ,MISS
          LD    A,(IX+1)
          LD    C,(IY+1)
          CP    C
          JR    NZ,MISS
          LD    A,(IX+2)
          LD    C,(IY+2)
          CP    C
          JR    NZ,MISS
          LD    A,(IX+3)
          LD    C,(IY+3)
          CP    C
          JR    NZ,MISS
          JR    CPR_PASS
MISS:   INC    IY
        INC    IY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        INC     IY
        INC     IY
        INC     B
        LD      A,06H
        CP      B
        JR      NZ,BLOCK_CPR
        JR      MISS_CODE
CPR_PASS:  LD      A,B           ;block number (1-5)
          LD      (PLB_BYTE1),A ;save block No. 4 bits LSB.
;
          -----
          CALL    REC_STA       ;check rec. status
          ;o/p HL=first byte of block
          ;o/p A =77 if empty at all
          CP      77H
          JR      Z,NOT_FOUND
          LD      C,B
          LD      B,0FH
          LD      IY,TP_FLAG
LOOP_TPF:  LD      A,(IY+0)
          AND     00001111B
          CP      C
          JR      Z,FOUND_DATA
          INC     IY
          DJNZ   LOOP_TPF
NOT_FOUND: CALL    CLRDISP
          CALL    DELAYF
          CALL    SOUND2
          LD      A,00
          CALL    GOTO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      HL,F6D2L1      ;SORRY! block empty
CALL    WR_LINE
CALL    DELAYF
CALL    DELAYF
CALL    DELAYF
CALL    DELAYF
CALL    DELAYF
CALL    SOUND2
JP      STOP_PLB
;
-----
FOUND_DATA:      LD      (POINT_T),HL      ;save location of block
;
-----
GETSUB_BLK:      LD      A,40H
CALL     GOTO
LD      HL,F6D3L2      ;subblock No.<1-3>:
CALL     WR_LINE
LD      A,52H
CALL     GOTO
W_SUB_BLK:      CALL     K_RELEASE
CALL     SCAN
LD      A,(Kbuff1)
CP      30H
JP      Z,STOP_PLB
SUB     34H
JR      NC,W_SUB_BLK
LD      A,(Kbuff1)
AND     0FH      ;subblock No.
PUSH    AF
;
-----
;      include subblock&block

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          and save to PLB_BYTE1
;          -----

SLA      A
SLA      A
SLA      A          ;shift subblock to
SLA      A          ;4 bits MSB
LD       B,A
LD       A,(PLB_BYTE1)
OR       B
LD       (PLB_B2),A
;          -----
POP      AF
PUSH     AF
SLA      A
SLA      A
AND      00001100B
PUSH     HL
LD       HL,(POINT_T) ;first byte of block
OR       (HL)
LD       (JUST_RD),A ;keep subblock that
;just read to 1 st of
;block (bit 2,3)

POP      HL
* POP    AF
;          -----

DEC      A
LD       C,A
SLA      A          ;subblock*2
ADD      A,C        ;subblock*3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      HL,00
LD      L,A
LD      DE,(POINT_T) ;get location of block
ADD     HL,DE        ;location byte of BLOCK:sub
LD      A,(HL)      ;sta. byte of subblock
BIT     0,A
JR      Z,DS_PLB    ;jmp. to don't PLB.if data
                        ;recorded empty.
;
;-----
;      chk. data still be in
;      tape or not
;-----
;
PUSH    HL
POP     IX
LD      B,0FH
LD      C,01H        ;count position of head
                        ;used for dist. mov
LD      HL,TP_FLAG
LD      A,(PLB_B2)
RES     7,A
RE_CMPR: LD      B,(HL)
RES     7,B
CP      B
JR      Z,EN_PLB
INC     HL
INC     C
DJNZ   RE_CMPR
LD      A,40H
CALL   GOTO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      HL,CLR_L
CALL   WR_LINE
CALL   DELAYF
CALL   SOUND2
LD      A,40H
CALL   GOTO
LD      HL,F6D7L2      ;"data already clear"
CALL   WR_LINE
CALL   DELAYF
CALL   SOUND2
CALL   DELAYF
CALL   DELAYF
CALL   DELAYF
JP      GETSUB_BLK
-----
;
;
;
;
;
DS_PLB:
LD      A,40H
CALL   GOTO
LD      HL,CLR_L
CALL   WR_LINE
CALL   DELAYF
CALL   SOUND2
LD      A,40H
CALL   GOTO
LD      HL,F6D6L2      ;this subblock empty
CALL   WR_LINE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    DELAYF
CALL    SOUND2
CALL    DELAYF
CALL    DELAYF
CALL    DELAYF
CALL    DELAYF
JP      GETSUB_BLK
;
-----
EN_PLB: LD    A,C
LD      (DIS_TMP),A ;value of distination
PUSH   HL           ;save position of TP_FLAG
;           ;(distination for mov. tape)
;           ;that must be clear read sta
;           ;after PLB.
;
;
;
IN      A,(PORTC2)
SET     6,A
OUT     (PORTC2),A ;motor run
LD      A,40H
CALL    GOTO
LD      HL,CLRL
CALL    WR_LINE    ;clear disp.
CALL    DELAY60
LD      A,40H
CALL    GOTO
LD      HL,F5D2L2 ;disp. please wait
CALL    WR_LINE
;
-----
LD      A,(TP_HEAD)
LD      C,A
LD      A,(DIS_TMP)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP      C
JR      Z,NO_SETORG
CALL    BACK_ORG      ;move tape to ORG befor
                                ;move to data area
;      -----
LD      A,10101010B
LD      (COND_MOV),A  ;PLB flag
LD      A,(DIS_TMP)
CALL    MOV_TOAEA     ;mov tape to data area
NO_SETORG:
IN      A,(PORTC2)
SET     6,A
OUT     (PORTC2),A    ;motor run
;      -----
LD      A,40H
CALL    GOTO
LD      HL,F5D5L2     ;press ENTER to start
CALL    WR_LINE
W_PLB1:
CALL    SCAN
LD      A,(KBUFF1)
CP      3EH
JR      NZ,W_PLB1
;      -----
IN      A,(PORTA2)
SET     3,A          ;bit play
OUT     (PORTA2),A
CALL    DELAYF       ;avoid org. area
OR      00010010B    ;en RY6,RY5 (amp1&2)
OUT     (PORTA2),A  ;play back enabled
LD      A,40H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    GOTO
LD      HL,F6D4L2      ;PLB. in execution
CALL    WR_LINE
;
-----
LD      A,00
LD      (CLOCK1),A    ;clear time count
IN      A,(PORTA2)
RES     0,A           ;clr. NMI ck
OUT     (PORTA2),A
CALL    DELAY
SET     0,A          ;en. NMI ck
OUT     (PORTA2),A
IN_PLB1: LD      A,(CLOCK1)
LD      B,(IX+2)
SUB     B
JR      C,IN_PLB1
IN      A,(PORTA2)
CALL    DELAYF
AND     11101100B
OUT     (PORTA2),A    ;RY6,RY5,NMI off
CALL    DELAYF
CALL    DELAYF
RES     3,A          ;stop play
OUT     (PORTA2),A
LD      A,(IX+0)
RES     1,A          ;set ready read
LD      (IX+0),A
POP     HL
LD      A,(HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RES      7,A
LD       (HL),A          ;res. bit ready read
                                ;in TP_FLAG table

LD       A,(JUST_RD)
LD       (POINT_T),A

;
;
;-----
;      move head to free area
;-----
;      CALL   FRE_AREA      ;find out empty area
;                                ;o/p reg.A=No.of area
;      CALL   MOV_TOAEA     ;tape move to empty area
;                                ;i/p reg.A=Distination
;      CALL   ADJ_FREAEA
STOP_PLB: JP   TEL_SCAN
;-----
;      * FIND BLOCK which be recodeed      *
;      * Subroutine .                      *
;      * i/p reg A =block No.[01-05]      *
;      * o/p reg A =77H if all subblock EMTY *
;      * o/p HL   =first location of block *
;-----
REC_STA: PUSH  BC
LD       C,A            ;save block No.(BKN)
SLA     A              ; PSI=BKN*2
SLA     A              ; PSI=BKN*4
SLA     A              ; PSI=BKN*8
ADD     A,C            ; PSI=BKN*9
SUB     09H            ; PSI=(BKN*9)-9

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      HL,TAPE_BUFF
LD      DE,00
LD      E,A
ADD     HL,DE
.PUSH   HL                      ;first byte of beyon block
LD      B,03                    ;No. of subblock =3
TST_STA:
LD      A,(HL)
BIT     0,A
JR      NZ,FULL
INC     HL                      ;byte position
INC     HL                      ;byte rec. time
INC     HL                      ;next subblock
DJNZ   TST_STA
LD      A,77H                  ;ALL subblock EMTY
                                ;can't playback
FULL:   POP     HL              ;getback first location of
                                ;beyon block again
POP     BC
RET
;
;
;
*****
*   END OF FUNCT. 6   *
*****
F6D1L1: DB      'Funct6.tape playback'
F6D2L1: DB      'SORRY! block empty '
F6D3L1: DB      'BLOCK CODE ERROR ! '
F6D1L2: DB      'block code ?:'
F6D2L2: DB      'Please wait ! '
F6D3L2: DB      'subblock No.[1-3]: '
F6D4L2: DB      ' PLB. in execution '

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

F6D5L2          DB .   'press ENTER to start'
F6D6L2:         DB     'this subblock empty '
F6D7L2:         DB     'data already clear! '
CLR_L:          DB     '                   '
                END

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; * Function7 tape format routine *
; *****
FUNCT7:      CALL  CLRDISP
              LD   A,00
              CALL GOTO
              LD   HL,F7D1L1      ;Funct.7 tape format
              CALL WR_LINE
              IN   A,(PORTC2)
              SET  6,A
              OUT  (PORTC2),A      ;motor tape on
              CALL DELAY60
              CALL CHKPSW          ;check pass word
              LD   A,40H
              CALL GOTO
              LD   HL,CLRL
              CALL WR_LINE
              CALL DELAYF
W_CST:      LD   A,40H
              CALL GOTO
              LD   HL,F7D2L2      ;"press ENTER to start"
              CALL WR_LINE
W_KEY:      CALL K_RELEASE
              CALL SCAN
              LD   A,(KBUF1)
              CP   3EH
              JR   NZ,W_KEY
GO_ONF:     LD   A,40H
              CALL GOTO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD . HL,CLRL
CALL WR_LINE
CALL DELAYF
CALL SOUND2
LD A,40H
CALL GOTO
LD HL,F7D3L2 ;"format in execution"
CALL WR_LINE
SET_ORGF:
NOP
CALL BACK_ORG ;rev tape to org.
CALL DELAYF
SET 5,A ;bit fwd lock
OUT (PORTA2),A ;fwd unlock
CALL DELAYF
CALL DELAYF
SET 2,A ;bit speed
OUT (PORTA2),A ;tape fwd
;
;
CALL DELAYF ;DEBUG
CALL DELAYF ;avoide org. of tape
-----
RES 2,A
RES 5,A
OUT (PORTA2),A ;fwd stoped
CALL DELAYF
IN A,(PORTA2)
SET 3,A
OUT (PORTA2),A ;pickup head
;
-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,40H
CALL   GOTO
LD      HL,F7D8L2      ;disp. block#
CALL   WR_LINE
;
-----
LD      D,10H          ;count No. sync of block
IN      A,(PORTA2)
SET     6,A
OUT     (PORTA2),A    ;rec on
CALL   DELAYF
CALL   DELAYF        ;pre area 1 Sec
SYNC_LOOP:
PUSH   DE
LD      A,4AH
CALL   GOTO
LD      B,0AH
LD      D,20H
LP_CLR:
CALL   WRBYTE
DJNZ   LP_CLR        ;clear '>'
POP    DE
PUSH   DE
LD      A,47H        ;CSOR block #N
CALL   GOTO
LD      HL,NUM_D2
LD      E,D
LD      D,0
ADD    HL,DE
LD      D,(HL)
CALL   WRBYTE        ;
LD      A,4AH        ;CSOR for '>'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL GOTO      ;
POP DE
-----
IN  A,(PORTA2)
RES 0,A
OUT (PORTA2),A ;clr NMI CK
LD  A,00
LD  (CLOCK4),A
LD  (CLOCK3),A
LD  (CLOCK2),A
LD  (CLOCK1),A
IN  A,(PORTC2)
SET 5,A ;bit of sync carrier
SET 3,A ;buzzer on
OUT (PORTC2),A
CALL DELAYF ;delay 1/2 Sec (sync time)
RES 5,A ;sync carrier off
RES 3,A ;buzzer off
OUT (PORTC2),A
IN  A,(PORTA2)
SET 0,A
OUT (PORTA2),A ;NMI ck on
LD  E,0AH ;count No. OF ''
VCE_AREA:
LD  A,(CLOCK4)
CP  00H
JR  Z,W_TME
PUSH DE
LD  A,00
LD  (CLOCK4),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      D,'>'
CALL   WRBYTE
POP    DE
DEC    E
JR     NZ,W_TME
;
-----
LD      E,0AH      ;count No. '>'
PUSH   DE
LD      A,4AH
CALL   GOTO
LD      B,0AH
LD      D,20H
LOOP_CLR: CALL WRBYTE
        DJNZ LOOP_CLR ;clear '>'
        POP  DE
        LD   A,4AH    ;set cursor for '>'again
        CALL GOTO
;
W_TME: LD   A,(CLOCK1)
        SUB  240D    ;Area for REC data= 2 Min
        JR   C,VCE_AREA ;
        DEC  D      ;
        JP   NZ,SYNC_LOOP
;
-----
IN      A,(PORTA2)
RES     6,A        ;stop rec
RES     3,A        ;stop play
RES     0,A        ;clr NMI CK
OUT     (PORTA2),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD    A,00
LD    (CLOCK3),A
CALL  DELAYF
```

;

```
-----
LD    D,00           ;count No. of sync
LD    A,40H
CALL  GOTO
LD    HL,F7D4L2     ;disp. format in verify
```

```
CALL  WR_LINE
IN    A,(PORTA2)
SET   3,A           ;play
RES   5,A           ;clr lock fwd.
```

```
OUT   (PORTA2),A
CALL  DELAYF
SET   2,A           ; bit speed
```

```
OUT   (PORTA2),A
LD    A,00
LD    (CLOCK1),A
LD    (CLOCK2),A
```

```
IN    A,(PORTA2)
SET   0,A
OUT   (PORTA2),A   ;NMI on
```

```
CALL  DELAYF
```

;

SEARCH:

```
-----
NOP
CALL  DELAY60
IN    A,(PORTB1)
BIT   1,A           ;is it sync
JR    NZ,SYNC_INC
```

```

; -----
; * check is it end of tape *
; -----
GET_P:      IN      A,(PORTB1)      ;get pulse of tape move
           BIT      2,A
           JR      NZ,DO_H
           LD      A,00
           LD      (CLOCK1),A      ;clear count of pulse Z
           LD      A,(CLOCK2)
           SUB     14D
           JR      NC,TPE_END
           JR      SEARCH
DO_H:      LD      A,00
           LD      (CLOCK2),A
           LD      A,(CLOCK1)
           SUB     14D
           JR      C,SEARCH
TPE_END:   IN      A,(PORTA2)
           RES     0,A
           OUT     (PORTA2),A      ;NMI off
; -----
           LD      A,40H
           CALL    GOTO
           LD      HL,F7D5L2      ;"format not complete"
           CALL    WR_LINE
           CALL    SOUND2
           CALL    DELAYF
           CALL    DELAYF
           CALL    DELAYF

```

```

CALL    DELAYF
LD      A,40H
CALL    GOTO
LD      HL,CLRL
CALL    WR_LINE
CALL    DELAY
LD      A,40H
CALL    GOTO
LD      HL,F7D6L2    ;disp tape too short
CALL    WR_LINE
IN      A,(PORTA2)
RES     2,A
RES     3,A
OUT     (PORTA2),A
CALL    SCAN
JR      FMT_END
SYNC_INC: INC     D
LD      A,10H
CP      D
JR      Z,FMT_CPLT
BOUNCE: IN      A,(PORTB1)
BIT     1,A
JR      NZ,BOUNCE
JR      SEARCH
FMT_CPLT: IN     A,(PORTA2)
RES     0,A
OUT     (PORTA2),A    ;NMI off
LD      A,40H
CALL    GOTO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      HL,CLRL
CALL   WR_LINE
CALL   DELAYF
CALL   SOUND2
CALL   DELAYF
LD      A,40H
CALL   GOTO
LD      HL,F7D7L2      ;disp. fotmat complete
CALL   WR_LINE
CALL   DELAYF
CALL   DELAYF
CALL   DELAYF
CALL   DELAYF
CALL   DELAYF
CALL   DELAYF
CALL   ADJ_FREAEA
;
; CALL   FRE_AREA      ;find out empty area
;
;                                ;o/p reg.A=No.of area
; CALL   MOV_TOAEA     ;tape move to empty area
;
;                                ;i/p reg.A=Distination
JP      TEL_SCAN
F7D1L1: DB      'Funct.7 tape format '
F7D1L2: DB      'CASSETTE NOT FOUND !'
F7D2L2: DB      'press ENTER to start'
F7D3L2: DB      'format in execution '
F7D4L2: DB      ' format in verify '
F7D5L2: DB      'format not complete '
F7D6L2: DB      ' tape too short ! '
F7D7L2: DB      'tape format complete'
F7D8L2: DB      'BLOCK # : '
NUM_D1: DB      '0123456789ABCDEF0123456789ABCDEF0'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NUM_D2:      DB      '10FEDCBA987654321'.
;
;           *****
;           END OF FUNCTION 7
;           *****
;           *****
;           * START OF FUNCTION 8 *
;           * check sta. of head posi.*
;           * in TP_BUFF *
;           *****
FUNCT8:      CALL    CLRDISP
;
;           LD      A,00
;           CALL    GOTO
;           LD      HL,F8L1      ;'HB1 HB2 HB3 HB4 HB5 '
;           CALL    WR_LINE
;           LD      A,40H
;           CALL    GOTO
;           LD      HL,CLRL      ;clear line2
;           CALL    WR_LINE
;           LD      B,05H
;           LD      C,40H
;           LD      IX,TAPE_BUFF
SHO_HP:      LD      A,C
;
;           CALL    GOTO
;           LD      HL,NUM_D1
;           LD      A,(IX+1)
;           LD      E,A
;           LD      D,0
;           ADD     HL,DE
;           LD      D,(HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL WRBYTE ;
```

```
;
```

```
-----
```

```
INC C
```

```
LD A,C
```

```
CALL GOTO
```

```
LD HL,NUM_D1
```

```
LD A,(IX+4)
```

```
LD E,A
```

```
LD D,0
```

```
ADD HL,DE
```

```
LD D,(HL)
```

```
CALL WRBYTE ;
```

```
;
```

```
-----
```

```
INC C
```

```
LD A,C
```

```
CALL GOTO
```

```
LD HL,NUM_D1
```

```
LD A,(IX+7)
```

```
LD E,A
```

```
LD D,0
```

```
ADD HL,DE
```

```
LD D,(HL)
```

```
CALL WRBYTE ;
```

```
;
```

```
-----
```

```
INC C
```

```
INC C
```

```
LD DE,00H
```

```
LD E,09H
```

```
PUSH IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP    HL
ADD    HL,DE
PUSH   HL
POP    IX
DJNZ   SHQ_HP
CALL   SCAN
CALL   K_RELEASE

```

;

```

CALL   CLRDISP
LD     A,0AH
CALL   GOTO
LD     HL,NUM_D1
LD     A,(TP_HEAD)
LD     E,A
LD     D,0
ADD    HL,DE
LD     D,(HL)
CALL   WRBYTE ;
CALL   SCAN
CALL   K_RELEASE
JP     TEL_SCAN

```

```

F8L1:  DB    'HB1 HB2 HB3 HB4 HB5 '
        END

```

```

; -----
; * CHECK PASS WORD ROUTINE *
; -----
CHKPSW:      PUSH    AF
              PUSH    BC
              PUSH    DE
              PUSH    HL
              PUSH    IX
              PUSH    IY
CHKPSW1:     LD      A,40H
              CALL   GOTO
              LD      HL,PSWREQ
              CALL   WR_LINE
              LD      A,4DH          ;command line cursor
              CALL   GOTO
              CALL   PASWD         ;get pass word
              LD      A,(D_NUM)
              CP      04H
              JR      NZ,PSWERR
              LD      B,04H        ;count 4 digit PSW.
              LD      IX,PSWBUFF
              LD      IY,KBUFF10
RECHK:       LD      C,(IX+0)
              LD      A,(IY+0)
              CP      C
              JR      NZ,PSWERR
              INC     IX
              INC     IY
              DJNZ   RECHK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JR      PSWOK
PSWERR:        LD      A,40H
                CALL   GOTO          ;
                LD      HL,TAB_CLR
                CALL   WR_LINE       ;clear line 2
                CALL   SOUND2       ;
                CALL   DELAY60
                CALL   SOUND2
                LD      A,40H
                CALL   GOTO
                LD      HL,TPSWERR
                CALL   WR_LINE       ;disp. pass word error!
                CALL   DELAYF
                CALL   DELAYF
                JR      CHKPSW1     ;get pass word again
PSWOK:        POP     IY
                POP     IX
                POP     HL
                POP     DE
                POP     BC
                POP     AF
                RET
;
;
;
NEWPSW:       PUSH   AF
                PUSH   BC
                PUSH   DE
                PUSH   HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,40H
CALL   GOTO
LD      HL,PSWOK2      ;disp. enter new pass word
CALL   WR_LINE
CALL   GET_4DG        ;get new pass word 4 digit
LD      A,(KBUF4)
CP      3EH
JR      Z,NON
;
;
;      -----
;      move key buff. data block
;      to pass word buff.
;      -----
LD      BC,0004H      ;count
LD      HL,KBUF4      ;Source
LD      DE,PSWBUF     ;distination
LDIR
NON:    POP      HL
        POP      DE
        POP      BC
        POP      AF
        RET
;
;
;      -----
;      * GET PASS WORD ROUTINE *
;      -----
PASWD:  PUSH     AF
        PUSH     BC
        PUSH     DE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH HL
PUSH IX
PUSH IY
STPSW: LD A,00H
LD (STA_K),A ;clear key status
LD C,00H ;count digit
RSTPSW: LD A,4DH ;csor
CALL GOTO
LD HL,KBUFF10 ;
REKEY: CALL K_RELEASE
CALL SCAN
LD A,(KBUFF1)
CP 3EH ;check enter key
JR NZ,KFUNC ;jump for no enter key
LD A,(STA_K)
BIT 0,A
JR NZ,ENDPSWD ;end routine when ENTER after
CALL SOUND2
JR RSTPSW ;get key again if no ENTER
KFUNC: LD A,(KBUFF1)
CP 3CH
JR NZ,KNUM ;not perform for funct. key
CALL SOUND2
CALL DELAYF
CALL SOUND2
JR REKEY
KNUM: LD A,(STA_K)
SET 0,A
LD (STA_K),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,(KBUF1)
LD      D,A
CALL    WRBYTE      ;disp. key value
LD      (HL),A
INC     HL
INC     C
LD      A,08H
CP      C
JR      NZ,REKEY
W_ENTR: CALL    K_RELEASE
        CALL    SOUND2
        CALL    SOUND2
        CALL    SCAN
        LD      A,(KBUF1)
        CP      3EH
        JR      NZ,W_ENTR
ENDPSWD: LD      A,C
        LD      (D_NUM),A
        CALL    SOUND2
        POP     IY
        POP     IX
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        RET
;
;      * GET 4 KEY VALUE *
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GET_4DG:      PUSH  AF
              PUSH  BC
              PUSH  DE
              PUSH  HL
              PUSH  IX
              PUSH  IY
              CALL  K_RELEASE

START:        LD    A,00H
              LD    (STA_K),A      ;clear key status

BEGIN:        LD    C,00H          ;count DIGIT
              LD    A,CMD_CUR
              CALL  GOTO
              LD    HL,KBUFF4
KIN:          CALL  SCAN
              LD    A,(KBUFF1)
              CP    3EH            ;check ENTER key pressed
              JR    Z,ENTER        ;JUMP when enter key
              CP    3CH
              JR    Z,KIN          ;Don't perform when be Fn.key
              LD    D,A
              CALL  WRBYTE         ;disp. key value
              LD    A,(STA_K)
              SET   0,A
              LD    (STA_K),A
              CALL  K_RELEASE      ;wait for release key
              LD    A,(KBUFF1)
              LD    (HL),A
              INC   HL
              INC   C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,04H
CP      C
JR      NZ,KIN
LD      A,(STA_K)
SET     1,A           ;key to be pressed 4 times
LD      (STA_K),A
JR      BEGIN
ENTER:  LD      A,(STA_K)
        BIT     0,A
        JR      Z,SAVE
        BIT     1,A
        JR      NZ,END_GK
        CALL    SOUND2
        CALL    DISP_ERR
        CALL    DELAY60
        CALL    DELAY60
        JR      START
SAVE:   LD      A,3EH
        LD      (KBUF4),A   ;when first key =ENTER
END_GK: POP     IY
        POP     IX
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        RET
;
;      *  INITIAL LCD  *
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INITLCD:      LD      A,0
              OUT     (PORTC2),A      ;
              LD      A,38H          ;function set 38h
                                              ;D=1 8 bit,N=1 1/16 duty,F=0
              OUT     (PORTB2),A      ;
              CALL    EPULSE
              CALL    DELAY          ;delay > 4.1 ms
              ;
              LD      A,0FH          ;disp on/off control D-1 off
                                              ;c=1 cursor on,B=1 blink
              OUT     (PORTB2),A
              CALL    EPULSE
              LD      A,06H          ;entry mode set
                                              ;I/D=1 increment ,S=0 right
              OUT     (PORTB2),A
              CALL    EPULSE
              LD      A,01H
              OUT     (PORTB2),A
              CALL    EPULSE
              CALL    DELAY
              RET

```

```

;
; -----
; * enable pulse *
; -----

```

```

EPULSE:      PUSH    AF
              PUSH    BC
              IN      A,(PORTC2)
              SET     2,A
              OUT     (PORTC2),A

```

```

LD      B,00H
EP1:   DJNZ  EP1
      IN   A,(PORTC2)
      RES  2,A
      OUT  (PORTC2),A
      POP  BC
      POP  AF
      RET
;
;
;
;
GOTO:  PUSH  AF
      PUSH  BC
      SET  7,A
      OUT  (PORTB2),A
      IN   A,(PORTC2)
      AND  11111000B
      OUT  (PORTC2),A
      CALL EPULSE
      POP  BC
      POP  AF
      RET
;
;
;
;
WRBYTE: PUSH  AF
      PUSH  DE

```

```

        IN    A, (PORTC2)
        RES   1, A
        RES   2, A
        SET   0, A           ;data write
;
        LD    A, 01H        ;data write
        OUT   (PORTC2), A
        LD    A, D          ;data byte
        OUT   (PORTB2), A
        CALL  EPULSE
        POP   DE
        POP   AF
        RET
SOUND1:
        PUSH  AF
        IN    A, (PORTC2)
        SET   3, A
        OUT   (PORTC2), A
        CALL  DELAY10
        RES   3, A
        OUT   (PORTC2), A
        POP   AF
        RET
SOUND2:
        PUSH  AF
        IN    A, (PORTC2)
        SET   3, A
        OUT   (PORTC2), A
        CALL  DELAY60
        RES   3, A
        OUT   (PORTC2), A
        POP   AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
DISP_ERR:  PUSH  AF
           PUSH  BC
           PUSH  DE
           PUSH  HL
           LD    HL, TB_ERR
           LD    A, ADD_ERR      ;defind start CURSOR
           CALL  GOTO           ;set ADDR subroutine
           LD    B, 0AH         ;NO. OF CHAR.
AAA:       LD    D, (HL)
           PUSH  BC
           CALL  WRBYTE
           POP   BC
           INC  HL
           DJNZ  AAA
           CALL  DELAYF
           CALL  DELAYF
           LD    HL, CLR_ERR
           LD    A, ADD_ERR
           CALL  GOTO
           LD    B, 0AH
AAA1:     LD    D, (HL)
           PUSH  BC
           CALL  WRBYTE
           POP   BC
           INC  HL
           DJNZ  AAA1
           POP   HL
           POP   DE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP      BC
POP      AF
RET

;
;
;
SCAN:    PUSH  AF
          PUSH  BC
          PUSH  DE
          PUSH  HL
KEYIN:   LD     C,PORTC1      ;port ROW
          LD     L,OFFH      ;key No. (-1)
          LD     B,4         ;No. of ROW
          LD     D,03H       ;No. of COLUMN
          LD     E,0FEH      ;pattern ROW "1111110B"
          LD     H,01110000B ;unpress key status
FROW:   OUT    (C),E
          IN     A,(PORTC1)  ;get key
          AND    H
          CP     H
          JP     NZ,FCOL
          LD     A,L
          ADD    A,D
          LD     L,A
          RLC    E
          DJNZ  FROW
          JP     KEYIN
FCOL:   CALL   DELAY10
          IN     A,(PORTC1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND    H
CP     H
JP     Z,KEYIN
SRL    A
SRL    A
SRL    A
SRL    A
OR     A
FIND1: INC  L
RR     A
JP     C,FIND1
LD     A,L
LD     HL,ASCIIT
LD     BC,00H
LD     C,A
ADD    HL,BC
LD     A,(HL)
LD     (KBUFF1),A
CALL  SOUND1
POP    HL
POP    DE
POP    BC
POP    AF
RET
;
; * SCAN KEY & CHK REC. STATUS *
;
SCAN1: PUSH AF
        PUSH BC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH    DE
PUSH    HL
KEYIN1: LD    C,PORTC1
        LD    L,OFFH
        LD    B,4
        LD    D,03H
        LD    E,0FEH
        LD    H,01110000B
FROW1:  OUT   (C),E
        IN   A,(PORTC1)
        AND  H
        CP   H
        JP   NZ,FCOL1
        LD   A,L
        ADD  A,D
        LD   L,A
        RLC
        DJNZ FROW1
        IN   A,(PORTC2)
        BIT  4,A ;rec status EN/DS
        JR   NZ,MEMF
        JR   KEYIN1
FCOL1:  CALL  DELAY10
        IN   A,(PORTC1)
        AND  H
        CP   H
        JP   Z,KEYIN1
        SRL  A
        SRL  A

```

```

SRL    A
SRL    A
OR     A
FIND2: INC    L
RR     A
JP     C,FIND2
LD     A,L
LD     HL,ASCIIT
LD     BC,00H
LD     C,A
ADD    HL,BC
LD     A,(HL)
LD     (KBUFF1),A
CALL   SOUND1
JR     ENDSCAN1
MEMF:  LD     A,20H
LD     (KBUFF1),A
ENDSCAN1: POP   HL
POP    DE
POP    BC
POP    AF
RET

;
;      * DELAY SUBROUTINE *
;
DELAY: LD     B,0H
DEL1:  NOP
NOP
DJNZ  DEL1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
;
;
;
DELAY10:    PUSH    AF
            PUSH    DE
            LD     DE,TIME1
LOOP2:      DEC     DE
            LD     A,D
            OR     E
            JR    NZ,LOOP2
            POP    DE
            POP    AF
            RET
;
;DELAY20:   PUSH    AF
;
;           PUSH    DE
;
;           LD     DE,TIME2
;LOOP1:     DEC     DE
;
;           LD     A,D
;
;           OR     E
;
;           JR    NZ,LOOP1 .
;
;           POP    DE
;
;           POP    AF
;
;           RET
;
;
;DELAY30:   PUSH    AF
;
;           PUSH    DE
;
;           LD     DE,TIME3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;LOOPP:          DEC    DE
;
;               LD     A,D
;
;               OR     E
;
;               JR     NZ,LOOPP
;
;               POP    DE
;
;               POP    AF
;
;               RET
;
-----

```

```

DELAY60:        PUSH    AF
                PUSH    DE
                LD     DE,TIME6
LOOPPP:         DEC    DE
                LD     A,D
                OR     E
                JR     NZ,LOOPPP
                POP    DE
                POP    AF
                RET
;
-----

```

```

DELAYF:         PUSH    AF
                PUSH    DE
                LD     DE,TIMEF
LOPPP:          DEC    DE
                LD     A,D
                OR     E
                JR     NZ,LOPPP
                POP    DE
                POP    AF
                RET

```

```

; -----
; * CHECK KEY RELEASE *
; -----
K_RELEASE:    PUSH    AF

              PUSH    DE

UN_PRESS:    LD      A,00H

              OUT     (PORTC1),A      ;out "0000" to ROW
              IN      A,(PORTC1)     ;get COLUMN
              LD      D,70H          ;masking
              AND     D
              CP      D
              JR      NZ,UN_PRESS    ;
              POP     DE
              POP     AF
              RET

TB_ERR:      DB      'ERROR!'
CLR_ERR:     DB      ' '
ASCIIIT:    DB      '123456'
            DB      '7890<>'

; -----
; * display cold boot routine *
; -----
DISP_CDB:    PUSH    AF

              PUSH    BC
              PUSH    DE
              PUSH    HL
              PUSH    IX
              PUSH    IY
              LD      A,0            ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUT    (PORTC2),A    ;
LD     A,00001101B   ;disp on/off control
OUT    (PORTB2),A    ;D=1 off,c=0 cursor off,
                        ;B=1 blink

CALL   EPULSE
CALL   DELAY
LD     A,01H         ;clear all display
OUT    (PORTB2),A
CALL   EPULSE
CALL   DELAY
;
LD     IX,TAB_CDB1    ;disp. cold boot line1
LD     IY,TAB_CDB2+42H ;disp. cold boot line2
LD     B,43H
LD     A,0
SHIFT: CALL   GOTO
      PUSH   IX      ;
      POP    HL      ; IX > HL
      CALL   WR_LINE ;disp line1
      LD     A,40H
      CALL   GOTO
      PUSH   IY      ;
      POP    HL      ; IY > HL
      CALL   WR_LINE ;disp line2
      INC    IX
      DEC    IY
      LD     A,01H
      OUT    (PORTB2),A
      CALL   EPULSE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL DELAY60
CALL DELAY60
CALL SOUND2
DJNZ SHIFT ;
CALL DELAY60
CALL DELAY60
CALL DELAY60
CALL DELAY60
CALL DELAY60
CALL DELAY60
CALL DELAY60
;
;
;
BLINK:
LD B,03H
LD A,40H
CALL GOTO
LD HL,TAB_CLR ;clear line2
CALL WR_LINE
CALL SOUND2
CALL DELAY60
CALL DELAY60
LD A,40H
CALL GOTO
LD HL,TAB_READY
CALL WR_LINE ;disp ready line2
CALL DELAY60
CALL DELAY60
CALL DELAY60
CALL DELAY60

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    DELAY60
CALL    DELAY60
DJNZ    BLINK
;
CALL    SCAN
;
CALL    K_RELEASE
CALL    DELAYF
POP     IY
POP     IX
POP     HL
POP     DE
POP     BC
POP     AF
RET
TAB_CDB1: DB ' , '
          DB 'MULTI-CHANNEL RECORDER '
          DB 'FROM A TELEPHONE LINE '
          DB '(MAIL BOX TELEPHONE) '
          DB ' , '
TAB_CDB2: DB ' : System ready : '
          DB ' , '
          DB ' , '
          DB ' , '
          DB ' , '
TAB_CLR:  DB ' , '
TAB_READY: DB ' : System ready : '
TAB_WB1:  DB ' : System ready : '
TAB_WB2:  DB ': Function request : '
TAB_WB3:  DB '[OK! Select <1-7>]'
CLRL:    DB ' , '
;
-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          *   DISPLAY WORM BOOT   *
;          *   & GET KEY FUNCTION  *
;          -----
DISP_WB:   PUSH   AF
          PUSH   BC
          PUSH   DE
          PUSH   HL
          LD     A,01H
          OUT    (PORTB2),A    ;clear all display
          CALL   EPULSE
          CALL   DELAY
          LD     A,00H
          CALL   GOTO
          LD     HL,TAB_WB1
          CALL   WR_LINE      ;disp. warm boot title L1
WAIT_Fn:   LD     A,40H
          CALL   GOTO
          LD     HL,TAB_WB2
          CALL   WR_LINE      ;disp. warm boot title L2
          CALL   SCAN
          LD     A,(KBUFF1)
          CP     3CH
          JP     Z,FUNCT      ;waiting for Fn. key
          LD     A,40H
          CALL   GOTO
          LD     HL,CLRL
          CALL   WR_LINE      ;clear disp. line2
          CALL   SOUND2
          CALL   DELAY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL K_RELEASE
.JR WAIT_Fn
FUNCT: LD A,0
OUT (PORTC2),A
LD A,38H ;function set 38h
OUT (PORTB2),A ;D=1 8 bit,N=1 1/16
; duty,F=0
CALL EPULSE
CALL DELAY ;delay > 4.1 ms
LD A,00001111B ;disp on/off control
OUT (PORTB2),A ;D=1 off,c=0 cursor off,
; B=1 blink
CALL EPULSE
CALL DELAY
WAIT1_6: CALL SOUND2
LD A,40H
CALL GOTO
LD HL,TAB_WB3
CALL WR_LINE
LD A,52H
CALL GOTO
CALL EPULSE ;set command cursor
CALL K_RELEASE
CALL SCAN
LD A,(KBUF1)
LD D,A
LD A,38H ;funct. 1 to 7 (8 for DEBUG)
SUB D

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JR      C, WAIT1_6
        LD      A, (KBUF1)
        CP      30H
        JR      Z, WAIT1_6
        LD      A, 52H          ;set command cursor
        CALL   GOTO
        LD      A, (KBUF1)
        LD      D, A
        CALL   WRBYTE         ;disp. key value
        CALL   DELAY60
        CALL   K_RELEASE
        POP    HL
        POP    DE
        POP    BC
        POP    AF
        RET
;
;
;
NMI:
        PUSH   AF
        PUSH   BC
        PUSH   DE
        PUSH   HL
        -----
        IN     A, (PORTA2)
        RES    0, A
        OUT    (PORTA2), A      ;clear NMI clock
        LD     A, (CLOCK4)
        SET    0, A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      (CLOCK4),A
LD      HL,CLOCK1
INC     (HL)          ;INC clock1
INC     HL
INC     (HL)          ;INC clock2
INC     HL
INC     (HL)          ;INC clock3
;
-----
IN      A,(PORTA2)
SET     0,A
OUT     (PORTA2),A    ;enable NMI clock
POP     HL
POP     DE
POP     BC
POP     AF
RETN
;
;
;
-----
* CLEAR ALL DISPLAY SUBROUTINE *
-----
CLRDISP:
PUSH    AF
PUSH    HL
LD      A,00H
CALL    GOTO
LD      HL,CLRL
CALL    WR_LINE
LD      A,40H
CALL    GOTO
LD      HL,CLRL
CALL    WR_LINE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        POP    HL
        POP    AF
        RET

;
;
;      *   WRITE  1 LINE   *
;      *   HL = START OF CHAR. *
;
;
WR_LINE:    PUSH    BC
            PUSH    DE
            PUSH    HL
            LD     B, 14H
RE_WR:     LD     D, (HL)
            PUSH    BC
            CALL   WRBYTE
            POP     BC
            INC    HL
            DJNZ   RE_WR
            POP     HL
            POP     DE
            POP     BC
            RET
        -END

```