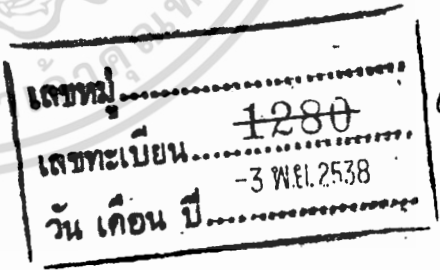
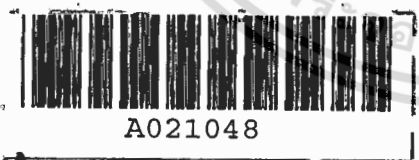


ปริญญาานิพนธ์

ตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็ก
(SENSOR AND DISPLAY OF METAL PIPE CHECKDETECTOR)



นายเกรียงศักดิ์ เหลืองอำพล
นายธนาธิป คำจันทร์
นายปรีดา กาลธียานันท์
นายภาสกร อนันตพันธ์



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรครุศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตรบัณฑิต
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาควิชา วิศวกรรมเครื่องกล

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาานิพนธ์

ชื่อหัวข้อปริญญาานิพนธ์ ตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็ก
SENSOR AND DISPLAY OF METAL PIPE CHECK DETECTOR

ชื่อนักศึกษา

1. นายเกรียงศักดิ์ เหลืองอำพล
2. นายธนาธิป คำจันทร์
3. นายปรีดา กาลธียนันท์
4. นายภาสกร อนันตพันธ์

อาจารย์ผู้ควบคุมปริญญาานิพนธ์

1. อาจารย์กิติพงศ์ มะโน
2. อาจารย์วรวิทย์ สมหา
3. อาจารย์สันติ ตันตระกูล

คณะกรรมการสอบปริญญาานิพนธ์	ลายมือ
1. อาจารย์กิติพงศ์ มะโน	
2. อาจารย์พีระวุฒิ สุวรรณจันทร์	
3. อาจารย์สุชิน อาจหาญ	
4. อาจารย์โกศล ตราชู	
5. อาจารย์ปิยะ จิตรธรรมมาภิรมย์	

วันเดือนปี ที่สอบ วันที่ 24 เดือนธันวาคม พ.ศ.2537 เวลา 13.30 น. ถึง 14.30 น.

สถานที่สอบ ห้อง ค.301 คณะวิศวกรรมศาสตร์



ลงนาม
ภาควิชา วิศวกรรมเครื่องกล

(ผศ.ดร.ธีระพิศ พงษ์พิพัฒน์ ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น

ไม่อาจรับผิดชอบต่อข้อผิดพลาดใดๆ และต้องแจ้งให้เจ้าของเอกสารหรือผู้ที่เกี่ยวข้องรับทราบ

ปริญญานิพนธ์

เรื่อง ตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็ก

SENSOR AND DISPLAY OF METAL PIPE CHECK DETECTOR

ผู้จัดทำ

1. นายเกรียงศักดิ์ เหลืองอำพล รหัสประจำตัว 36031405
2. นายธนาธิป คำจันทร์ รหัสประจำตัว 36031412
3. นายปรีดา กาลธียนันท์ รหัสประจำตัว 36031419
4. นายภาสกร อนันตพันธ์ รหัสประจำตัว 36031423

อาจารย์ที่ปรึกษา

ลงนาม

(อาจารย์กิติพงศ์ มะโน)

ลงนาม

(อาจารย์วรัววิทย์ สมนา)

ลงนาม

(อาจารย์สันติ ต้นตระกูล)

หัวหน้าภาควิชา

ลงนาม

(ผศ.ดร. ชีระพล เทพหัสดิน ณ อยุธยา)

ปีการศึกษา 2537

ปริญญาานิพนธ์

เรื่อง ตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็ก
(SENSOR AND DISPLAY OF METAL PIPE CHECK DETECTOR)

วัตถุประสงค์

1. เพื่อศึกษาถึงวิธีการทดสอบแบบไม่ทำลาย
2. เพื่อศึกษาการใช้ภาษาซีในการเขียนโปรแกรมควบคุมระบบให้เลิกซึ่งยิ่งขึ้น
3. เพื่อศึกษาการใช้สนามแม่เหล็กในการทดสอบท่อเหล็ก
4. เพื่อสร้างชุดต้นแบบตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็ก
5. เพื่อเป็นแนวทางในการพัฒนาเครื่องต้นแบบให้มีประสิทธิภาพที่สมบูรณ์ต่อไป

ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจถึงวิธีการทดสอบแบบไม่ทำลาย
2. ได้ความรู้เกี่ยวกับเทคนิคการเขียนโปรแกรมภาษาซี ตลอดจนวิธีการอินเตอร์เฟสระหว่างเครื่องไมโครคอมพิวเตอร์กับชุดหัวอ่าน โดยผ่านพอร์ต 8255
3. ได้ความรู้เกี่ยวกับการใช้สนามแม่เหล็กในการทดสอบท่อเหล็กและการพันขดลวดในการทำชุดหัวอ่าน(Shoe)
4. ได้ตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็ก ใช้งานได้ 1 เครื่อง
5. ได้แนวทางในการพัฒนาตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็กให้มีประสิทธิภาพสูงต่อไป

ตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวท่อเหล็ก

นายเกรียงศักดิ์ เหลืองอำพล

นายธนาธิป คำจันทร์

นายปรีดา กาลธียนันท์

นายภาสกร อนันตพันธ์

อาจารย์ที่ปรึกษา

นายกิตติพงศ์ มะโน

นายวรวิทย์ สมหา

นายสันติ ตันตระกุล

ปีการศึกษา 2537

บทคัดย่อ

เนื่องจากในปัจจุบันในอุตสาหกรรมต่าง ๆ ที่ใช้ท่อเหล็กเป็นในการขนส่งวัตถุดิบ เช่น อุตสาหกรรมขุดเจาะน้ำมัน หรือก๊าซ เป็นอุตสาหกรรมที่จำเป็นต้องใช้ท่อเหล็ก และจะต้องมีการตรวจสอบก่อนที่จะนำไปใช้งาน หรือนำท่อที่ใช้งานแล้วมาตรวจสอบก่อนที่จะนำไปใช้งานอีกครั้ง ดังนั้นโครงการนี้จึงได้สร้างต้นแบบตัวตรวจจับและแสดงผลของเครื่องตรวจหารอยร้าวท่อเหล็กขึ้นมา โดยใช้วิธีการวัดเส้นแรงแม่เหล็กที่เกิดจากการทำให้ท่อเหล็กมีสภาพเป็นแม่เหล็ก เมื่อมีรอยร้าวหรือจุดบกพร่องเกิดขึ้น ก็จะเกิดเส้นแรงแม่เหล็กรั่วไหลขึ้นที่บริเวณดังกล่าว และวัดเส้นแรงแม่เหล็กรั่วไหลเพื่อนำไปประมวลผลและแสดงผลโดยไมโครคอมพิวเตอร์ ซึ่งจะสามารถบันทึกข้อมูลเก็บไว้ได้ และแสดงผลออกทางจอภาพและเครื่องพิมพ์ได้

SENSOR AND DISPLAY OF METAL PIPE CHECK DETECTOR

MR. KREANGSAK LEANGUMPOL

MR. TANATIP KHAMCHAN

MR. PREEDA KALTIYANAN

MR. PASAKORN ANANTAPAN

ADVISOR

MR. KITTIPONG MAÑO

MR. WORAWIT SOMHA

MR. SANTI TANTAKOOL

1994

IN THE PRESENT, MANY INDUSTRY WILL USE METAL PIPE IN TRANSPORTATION, SUCH AS PETOLEUM. IT IS IMPORTANT TO INSPECT EITHER BEFORE USING METAL PIPE OR REUSING METAL PIPE. THEREFORE, THIS PROJECT WILL INSTRUCT A SENSOR AND DISPLAY OF METAL PIPE CHECK DETECTOR, WHICH MEASURING WILL MAKE METAL PIPE BECOMES A MAGNETIC. THERE IS MAGNETIC FLUX IN THE PLACE OF BROKEN OR ERROR POINT, SO WE CAN MEASURE LEAK MAGNETIC FLUX FOR EXECUTE AND PRESENT BY MICROCOMPUTER, THAT CAN BE RECORDED AND DISPLAY ON MONITOR OR PRINTER.

กิติกรรมประกาศ

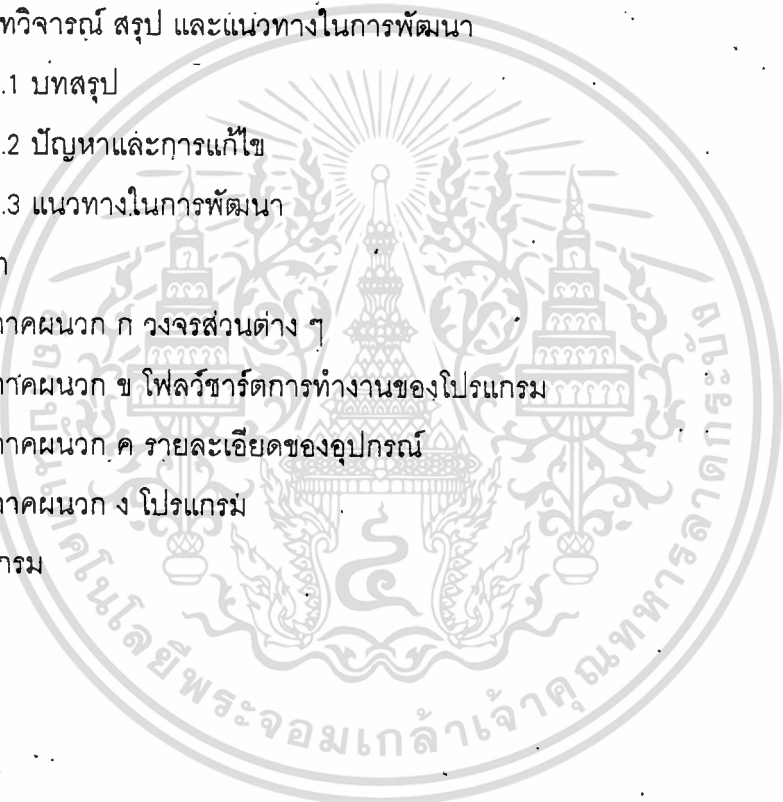
การที่โครงการนี้จะสามารถสำเร็จลุล่วงได้นั้น เนื่องจากการช่วยเหลือในด้านต่าง ๆ จากหลาย ๆ ฝ่ายด้วยกัน ในด้านการให้คำปรึกษานั้น ต้องขอขอบพระคุณ ท่านอาจารย์กิติพงศ์ มะโน , อาจารย์สันติ ตันตระกูล , อาจารย์วรวิทย์ สมหา และอาจารย์ประจำภาควิชา ครุศาสตร์วิศวกรรมทุกท่าน ที่นอกจากให้ความช่วยเหลือในด้านกรให้คำปรึกษาแล้วยังช่วยในการวิเคราะห์ปัญหา และการให้ความสนับสนุนทางด้านเครื่องมือต่างๆ นอกจากนี้ขอขอบพระคุณเพื่อน ๆ และพี่ ๆ ที่คอยให้คำแนะนำในเรื่องต่าง ๆ ไม่ว่าจะเป็นการปฏิบัติตนและการจัดทำปฏิญญานิพนธ์ คุณพ่อ,คุณแม่ที่ให้การสนับสนุนในด้านงบประมาณบางส่วนและอีกหลาย ๆ ท่านที่คอยให้กำลังใจ จนโครงการนี้สามารถสำเร็จลุล่วงไปด้วยดี จึงขอขอบพระคุณมา ณ โอกาสนี้



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	
บทคัดย่อภาษาอังกฤษ	
กิจกรรมประกาศ	
สารบัญ	
สารบัญตาราง	
สารบัญภาพ	
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 การทดสอบรอยบกพร่องของวัสดุ	2
2.1.1 การทดสอบแบบทำลาย	2
2.1.2 การทดสอบแบบไม่ทำลาย	3
2.1.3 วัตถุประสงค์ของการทดสอบแบบไม่ทำลาย	7
2.1.4 ประเภทของการทดสอบโดยไม่ทำลาย	8
เพื่อตรวจหารอยบกพร่องบริเวณผิว	
2.2 การทดสอบโดยใช้วิธีการรั่วไหลของสนามแม่เหล็ก	9
2.3 ความเข้มของสนามแม่เหล็ก	12
2.3.1 การเป็นแม่เหล็กที่ถูกจำกัด	14
บทที่ 3 การออกแบบและการสร้าง	15
3.1 HARDWARE	15
3.1.1 ภาค Amplifier	15
3.1.2 ภาค Multiplex	20
3.1.3 ภาค Sample & Hold	21
3.1.4 ภาค Analog to Digital หรือ A/D	23
3.1.5 รายละเอียดของสัญญาณต่าง ๆ บนสล๊อต	26
3.1.7 การ Interface โดยใช้ 8255	30

3.1.8 การสร้างหัวอ่าน หรือ shoe	31.
3.2 SOFTWARE	37
3.2.1 โปรแกรมหลัก	37
3.2.2 โปรแกรมย่อย	40
บทที่ 4 การทดลองและผลการทดลอง	42
4.1 ขั้นตอนการทดลอง HARDWARE	42
4.2 ขั้นตอนการทดลอง SOFTWARE	43
บทที่ 5 บทวิจารณ์ สรุป และแนวทางในการพัฒนา	48
5.1 บทสรุป	48
5.2 ปัญหาและการแก้ไข	48
5.3 แนวทางในการพัฒนา	49
ภาคผนวก	50
ภาคผนวก ก วงจรส่วนต่าง ๆ	50
ภาคผนวก ข ไฟล์ซอร์สโค้ดการทำงานของโปรแกรม	53
ภาคผนวก ค รายละเอียดของอุปกรณ์	63
ภาคผนวก ง โปรแกรม	78
บรรณานุกรม	162



สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ชนิดของการทดสอบโดยไม่ทำลาย	9
ตารางที่ 3.2x แสดงตารางความจริงของ MC 14081B	21



สารบัญรูปภาพ

รูปภาพ	หน้า
รูปที่ 2.1	3
รูปที่ 2.2	4
รูปที่ 2.3	4
รูปที่ 2.4	5
รูปที่ 2.5	5
รูปที่ 2.6	5
รูปที่ 2.7	6
รูปที่ 2.8	6
รูปที่ 2.9	6
รูปที่ 2.10	7
รูปที่ 2.11	10
รูปที่ 2.12	10
รูปที่ 2.13	11
รูปที่ 2.14ก	12
รูปที่ 2.14ข	12
รูปที่ 3.1	15
รูปที่ 3.2	19
รูปที่ 3.3ก	21
รูปที่ 3.4ก	22
รูปที่ 3.4ข	22
รูปที่ 3.5	22
รูปที่ 3.6ก	23
รูปที่ 3.6ข	24
รูปที่ 3.7	24
รูปที่ 3.8	29

รูปที่ 3.9	แสดงวงจร Decode ที่ใช้ 74138	31
รูปที่ 3.10	ลักษณะของขดลวด	32
รูปที่ 3.11ก	ลักษณะของขดลวดที่ใส่ลงไปใน shoe	32
รูปที่ 3.11ข	ลักษณะของ shoe ที่ต่อสายไฟเรียบร้อยแล้ว	33
รูปที่ 3.12	หัว shoe ขนาดต่าง ๆ	33
รูปที่ 3.13ก	ลักษณะของส่วนที่ขยายสัญญาณของ shoe ทั้ง 8 ชุด	34
รูปที่ 3.13ข	แผงด้านหน้าของส่วนที่ขยายสัญญาณ shoe	35
รูปที่ 3.14	ลักษณะการต่อใช้งานกับเครื่องคอมพิวเตอร์	35
รูปที่ 3.15	ลักษณะของการ์ดอินเตอร์เฟส	36
รูปที่ 3.16	การติดตั้งใช้งาน	37
รูปที่ 3.17	ส่วนประกอบของโปรแกรมหลัก	38
รูปที่ 3.18	ส่วนประกอบของโปรแกรมย่อย	40
รูปที่ 4.1	บล็อกไดอะแกรมการต่อมัลติมิเตอร์วัดสัญญาณ	42
รูปที่ 4.2	แสดงการใช้มัลติมิเตอร์วัดสัญญาณที่เอาท์พุท	42
รูปที่ 4.3	บล็อกไดอะแกรมการทดลองใช้งานกับคอมพิวเตอร์	43
รูปที่ 4.4	ขณะทำการทดลอง	43
รูปที่ 4.5	แสดงลักษณะหน้าจอหลังจากเรียกโปรแกรม	44
รูปที่ 4.6	ลักษณะจอภาพหลังจากกดคีย์ใด ๆ	45
รูปที่ 4.7	ลักษณะจอภาพหลังจากกดคีย์ F5 หรือ Save	46
รูปที่ 4.8	เมนูสำหรับป้อนข้อมูลรายละเอียดของท่อเหล็ก	46
รูปที่ 4.9	กราฟและข้อมูลที่ได้จากการพิมพ์	47

บทที่ 1

บทนำ

ในปัจจุบัน อุตสาหกรรมชุดเจาะน้ำมัน หรืออุตสาหกรรมก๊าซ ได้หันมาใช้การขนส่งน้ำมันหรือก๊าซทางท่อกันมากขึ้นและในการใช้งาน ท่อที่ใช้ส่งจำเป็นที่จะต้องมีการตรวจสอบให้สามารถใช้งานได้คืออยู่เสมอ และเครื่องมือที่ใช้ในการตรวจสอบนั้นมีราคาแพงมาก ต้องนำเข้าจากต่างประเทศ จึงเป็นเหตุผลที่ทำให้ริเริ่มจัดทำโครงการนี้ขึ้น ตัวตรวจจับและแสดงผลของเครื่องตรวจหารอยรั่วท่อเหล็กที่จัดทำขึ้นนี้ จะมีประสิทธิภาพเหมือนกันกับเครื่องที่สั่งเข้าจากต่างประเทศ และมีราคาถูกกว่า ซึ่งจะทำให้สามารถลดค่าใช้จ่ายในการจัดซื้อเครื่องมือตรวจสอบ นอกจากนี้ยังสามารถนำไปประยุกต์ใช้กับอุตสาหกรรมอื่น ๆ ได้อีกด้วย เช่น อุตสาหกรรมการผลิตท่อเหล็ก โดยจะใช้ตรวจสอบคุณภาพของท่อเหล็กก่อนที่จะส่งให้ลูกค้า ซึ่งจะทำให้สามารถลดต้นทุนในการผลิตได้เนื่องจากการทดสอบแบบไม่ทำลาย หลักการของเครื่องนี้จะใช้หลักการของสนามแม่เหล็กในการตรวจสอบ โดยการป้อนสนามแม่เหล็กผ่านท่อและมีหัวอ่าน(shoe)ซึ่งจะมีทั้งหมด 8 หัว อ่านสัญญาณที่ได้ผ่านวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล จากนั้นก็จะส่งสัญญาณไปให้คอมพิวเตอร์แสดงผลซึ่งจะแสดงผลเป็นกราฟและเก็บข้อมูลลงดิสก์ นอกจากนี้ยังสามารถพิมพ์กราฟที่ได้ออกทางเครื่องพิมพ์ได้ ซึ่งมีรายละเอียดเกี่ยวกับวงจรและวิธีการทำงานของเครื่องในแต่ละบทในปฏิญานินพจน์ดังนี้

ในบทที่ 2 ซึ่งเป็นส่วนของทฤษฎีและหลักการต่าง ๆ ที่ใช้ในการออกแบบ เช่น การทำให้ท่อเหล็กเป็นแม่เหล็ก การ INTERFACE กับคอมพิวเตอร์ การแปลงสัญญาณอนาลอกเป็นดิจิตอล เป็นต้น

ในบทที่ 3 จะเป็นการออกแบบวงจรในแต่ละส่วน เช่น ภาค Amplifier ,ภาค Multiplex, ภาค Sample & Hold ,ภาค Analog to Digital ,การ INTERFACE และโปรแกรมการทำงาน

ในบทที่ 4 จะเป็นส่วนของการทดลองวงจรและโปรแกรม

ในบทที่ 5 จะเป็นการสรุปผลการดำเนินงานที่ผ่านมา เช่น ผลการทดลอง ปัญหาที่พบ แนวทางการแก้ปัญหา และแนวทางในการพัฒนาในอนาคตต่อไป

บทที่ 2

ทฤษฎีและหลักการ

2.1 การทดสอบรอยบกพร่องของวัสดุ

ในการทดสอบรอยบกพร่องของวัสดุ หมายถึงการทดสอบเพื่อหาความผิดปกติของวัสดุ นั้น ๆ โดยการทดสอบรอยบกพร่องดังกล่าวสามารถแบ่งออกได้เป็น 2 วิธีด้วยกันคือ การทดสอบแบบทำลาย(Destructive Testing) และการทดสอบแบบไม่ทำลาย(Non Destructive Testing) ดังมีรายละเอียดดังต่อไปนี้

2.1.1 การทดสอบแบบทำลาย (Destructive Testing : DT)

เป็นวิธีการทดสอบที่ทำให้เกิดการแตกหักก่อนแล้วพิจารณาถึงโครงสร้างของวัสดุอีกครั้งโดยมากจะใช้ทดสอบพารามิเตอร์ดังนี้

การทดสอบความทนทานต่อแรงดึง(Tension Test)

การทดสอบความทนทานต่อการดัดโค้ง(Bending Test)

การทดสอบความทนทานต่อแรงกระแทก(Impact Test)

การทดสอบความทนทานต่อความล้า(Fatigue Test)

การทดสอบความแข็ง(Hardness Test)

การทดสอบการรับแรง(Load Test)

การทดสอบความทนทานต่อการแตกหักด้วยแรงดันภายใน(Breaking Test by Internal Pressure)

แต่ว่าการทดสอบโดยการทำลายดังกล่าวนี้ จะทำให้วัสดุ อุปกรณ์ หรือโครงสร้างเกิดการแตกหักเสียหายใช้งานอีกไม่ได้ ยกตัวอย่างเช่น กรณีของผลิตภัณฑ์ของโรงงานผลิตภัณฑ์เหล็กและเหล็กกล้า ซึ่งทำผลิตภัณฑ์รูปร่างและขนาดเป็นแบบเดียวกัน โดยวิธีการผลิตแบบเดียวกัน(หล่อ,อัด,รีด,เชื่อม เป็นต้น)เป็นจำนวนมากจะใช้วิธีสุ่มตัวอย่างจากผลิตภัณฑ์แต่ละชุด(ในวิธีการสุ่มตัวอย่าง จะเรียกชุดของผลิตภัณฑ์ว่า lot)มาทำการทดสอบแบบทำลายและประเมินค่าความแข็งแรงของผลิตภัณฑ์ทั้งจำนวนได้ดังในรูปที่ 2.1 ถึง 2.4

ในทางกลับกัน ตัวอย่างในกรณีของผลิตภัณฑ์จากอุตสาหกรรมหนัก ซึ่งผลิตทีละจำนวนน้อยหรือเพียงชิ้นเดียว ถ้านำมาทดสอบแบบทำลายแล้ว ก็จะไม่เหลือผลิตภัณฑ์ไปส่งมอบ

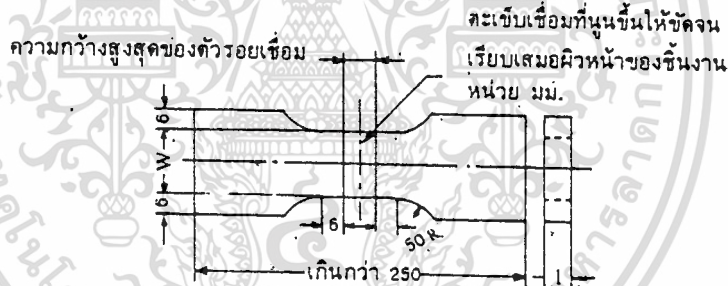
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก่ลูกค้าได้ ดังนั้นในกรณีที่ต้องการใช้ผลิตภัณฑ์ชิ้นนี้ ก็จะนำไปทดสอบแบบทำลายไม่ได้ จึงอาศัยวิธีการทดสอบแบบไม่ทำลายซึ่งจะกล่าวรายละเอียดในหัวข้อถัดไป

2.1.2 การทดสอบแบบไม่ทำลาย(Non Destructive Testing : NDT)

ซึ่งสมาคมวิศวกรรมเครื่องกลของอเมริกา(ASME) จะเรียกว่าการตรวจสอบแบบไม่ทำลาย(Non Destructive Examination : NDE) เนื่องจาก เป็นการทดสอบวัสดุชิ้นงานโดยไม่ทำลายให้เสียหายนั่นเอง กล่าวคือ เป็นการทดสอบชิ้นงานที่เหมือนกับวัสดุ อุปกรณ์ หรือ โครงสร้าง (ต่อไปจะเรียกว่าชิ้นทดสอบ) โดยไม่ก่อให้เกิดรอยขีดข่วน การแยกส่วน หรือการแตกหักภายใน และเกิดการเปลี่ยนแปลงรูปร่างขนาด หรือสมรรถนะ

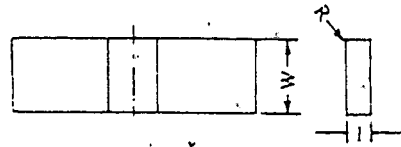
ในการนี้จะใช้ลักษณะสมบัติทางฟิสิกส์ของชิ้นทดสอบ เช่น แสง ความร้อน รั้งสี คลื่นเสียงอัลตราโซนิก(Ultrasonic) ไฟฟ้า หรืออำนาจแม่เหล็กซึ่งจะเปลี่ยนแปลงไปตามความผิดปกติของโครงสร้างภายใน หรือของโครงสร้างภายใน หรือรอยบกพร่องที่มีอยู่ได้



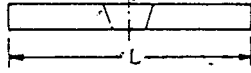
รูปที่ 2.1 ชิ้นงานทดสอบความทนทานต่อแรงดึงของรอยเชื่อมประสาน (JIS Z 3121 - หมายเลข 1)

ในวิธีการทดสอบนั้นไม่จำเป็นต้องใช้อุปกรณ์ทดสอบเลยก็ได้ วิธีที่เร็วที่สุด ประหยัดที่สุด และยังมีประสิทธิภาพมากที่สุดคือการตรวจสอบด้วยสายตาเป็นขั้นเริ่มแรก จากนั้นก็มีการตรวจสอบโดยใช้รั้งสี การตรวจสอบรอยบกพร่องโดยใช้อัลตราโซนิก การตรวจสอบรอยบกพร่องด้วยผงแม่เหล็ก การตรวจสอบรอยบกพร่องโดยใช้สนามแม่เหล็ก (การตรวจสอบรอยบกพร่องด้วยกระแสไฟฟ้า) ซึ่งเป็นการใช้อุปกรณ์ต่าง ๆ เข้ามาช่วยดังรูปที่ 2.5 - 2.8

ที่กล่าวมาข้างต้นว่า การทดสอบแบบไม่ทำลายนั้นใช้แทนการทดสอบแบบทำลายได้ ไม่ได้หมายความว่าเมื่อทำการทดสอบแบบไม่ทำลายได้แล้ว การทดสอบแบบทำลายจะกลายเป็นสิ่งที่ไม่จำเป็นไป การทดสอบแบบไม่ทำลายนั้นเป็นการใช้ลักษณะสมบัติทางฟิสิกส์



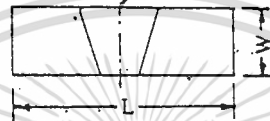
ตะเข็บเชื่อมที่นูนขึ้นให้ชัดเจน
เรียบเสมอผิวหน้าของชิ้นงาน



(ก) ชิ้นงานทดสอบงอด้านหน้าหรือหลัง



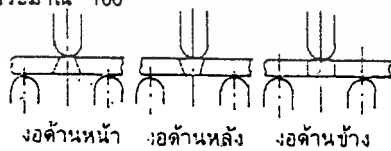
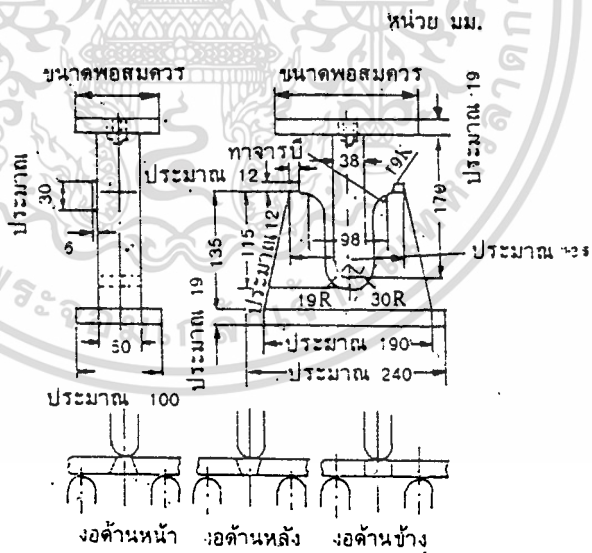
ตะเข็บเชื่อมที่นูนขึ้นให้ชัดเจน
เรียบเสมอผิวหน้าของชิ้นงาน



(ข) ชิ้นงานทดสอบงอด้านข้าง

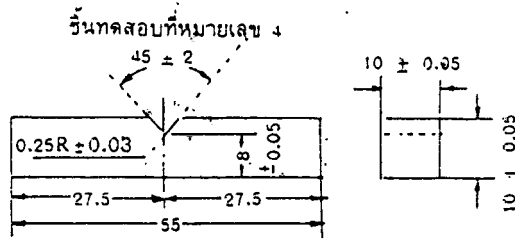
ความหนาของชิ้นงาน 38 มม.

รูปที่ 2.2 ชิ้นงานทดสอบความทนทานการติดตั้งของรอยเชื่อมประสาน



(การงอด้านข้างให้เอียงที่มีจุดบกพร่องมากไว้ข้างล่าง)

รูปที่ 2.3 วิธีการทดสอบความทนทานต่อการติดตั้ง



รูปที่ 2.4 วิธีการทดสอบความทนทานต่อแรงกระแทก V-Charpy

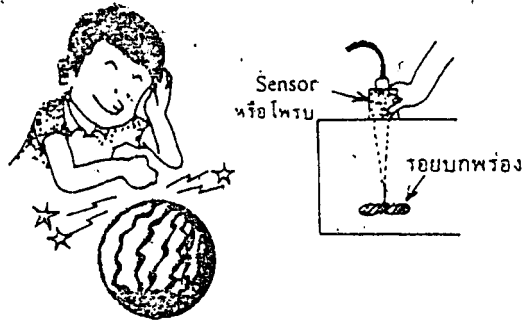


รูปที่ 2.5 การตรวจสอบด้วยสายตา



รูปที่ 2.6 การตรวจสอบด้วยรังสี

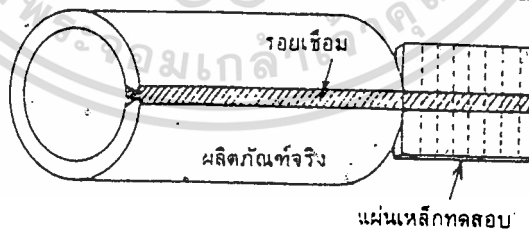
ของชิ้นงานทดสอบในการประเมินระดับของรอยร้าวหรือความแข็งแรง ทว่า ข้อมูลของการทดสอบแบบทำลายก็เป็นส่วนประกอบที่จำเป็นเช่นกัน ตัวอย่างเช่น การเชื่อมแผ่นเหล็กกับเหล็กทดสอบโดยวิธีการเดียวกันกับการเชื่อมโครงสร้างของสิ่งก่อสร้าง แล้วใช้การทดสอบแบบทำลายกับแผ่นเหล็กทดสอบ(ซึ่งเรียกว่า Test Plate Testing) เพื่อเป็นการประเมินความเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก่อนนั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การตรวจสอบด้วยอัลตราโซนิก



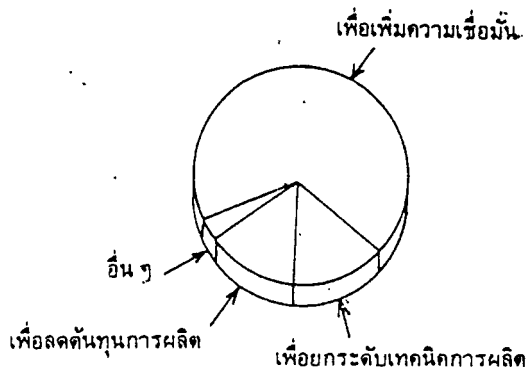
รูปที่ 2.8 การตรวจสอบด้วยผงแม่เหล็ก



รูปที่ 2.9

แข็งแรงของรอยเชื่อม และเป็นการประกันความปลอดภัยของรอยเชื่อม เป็นตัวอย่างหนึ่งของการทดสอบแบบทำลายชิ้นงานทดสอบที่ใช้ประกอบกับการทดสอบแบบไม่ทำลายผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 วัตถุประสงค์ของการทดสอบโดยไม่ทำลาย

2.1.3 วัตถุประสงค์ของการทดสอบแบบไม่ทำลาย

วัตถุประสงค์หลักของการทดสอบแบบไม่ทำลายที่พอจะกล่าวถึงได้ มีดังต่อไปนี้

2.1.3.1 เพื่อเพิ่มความเชื่อมั่นของผลิตภัณฑ์

อุปกรณ์ และโครงสร้างต่าง ๆ นั้น จะต้องมีความปลอดภัยในขณะที่ใช้งาน ถ้าไม่มีรอยบกพร่องอยู่เลยก็จะมี ความแข็งแรงทางกลเท่าเทียมกับวัสดุที่ไม่มีรอยบกพร่อง ซึ่งตามความเป็นจริงแล้วการที่จะให้มีวัสดุที่ไม่มีรอยบกพร่องเลยย่อมเป็นไปได้ ทำนองเดียวกันวิธีการผลิตที่ไม่ก่อให้เกิดรอยบกพร่องเลยก็เป็นไปไม่ได้

ดังนั้น ที่ตัวชิ้นส่วน หรือบริเวณรอยเชื่อมนั้นมีรอยบกพร่องอยู่เท่าใดจะประเมินได้โดยการตรวจสอบอุปกรณ์หรือโครงสร้างจริงที่จะใช้งาน แล้วตัดสินใจว่ารอยบกพร่องที่มีนั้นมีความปลอดภัยเพียงพอสอดคล้องกับวัตถุประสงค์ในการใช้งานหรือไม่ ข้อมูลที่จะใช้ในการตัดสินใจดังกล่าวจะได้จากการทดสอบโดยไม่ทำลายนั่นเอง

โดยการใช้วิธีการทดสอบแบบไม่ทำลายอย่างเหมาะสม จะสามารถตรวจสอบให้แน่ใจถึงความปลอดภัยและเพิ่มความเชื่อมั่นขึ้นได้ ซึ่งก็เป็นวัตถุประสงค์ที่สำคัญประการหนึ่งของการตรวจสอบแบบไม่ทำลาย

2.1.3.2 เพื่อการปรับปรุงเทคนิคการผลิต

ตัวอย่างเช่น เพื่อกำหนดวิธีการเชื่อมที่เหมาะสม จะทำการวางแผนการเชื่อมต่าง ๆ แล้วสร้างแผนทดสอบโดยเชื่อมตามแผนที่วางไว้ จากนั้นก็ทำการทดสอบโดยไม่ทำลายโดยวิธีการต่าง ๆ เริ่มตั้งแต่การทดสอบด้วยรังสีจากผลการทดสอบก็จะนำมาแก้ไขปรับปรุงวิธีการเชื่อม แล้วตัดสินใจเลือกวิธีการเชื่อมที่จะได้รอยเชื่อมที่มีคุณภาพตามต้องการ ซึ่งวิธีการ

เอกสารถ่ายโดยไม่ทำลายก็มีส่วนช่วยในการปรับปรุงเทคนิคในการเชื่อมได้ ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันนี้ การทดสอบแบบไม่ทำลายได้มีส่วนช่วยในการปรับปรุงเทคนิคการผลิตต่าง ๆ เป็นอย่างมากเริ่มจากเทคนิคในการเชื่อมเป็นประการแรก ต่อไปในอนาคตการทดสอบโดยไม่ทำลายก็จะมีบทบาทสำคัญโดยขาดเสียมิได้ในการช่วยพัฒนาปรับปรุงเทคนิคการผลิตต่าง ๆ มากยิ่งขึ้น

2.1.3.3 เพื่อลดต้นทุนการผลิต

ในการทดสอบแบบไม่ทำลายนั้น จะเพิ่มค่าใช้จ่ายในการทดสอบขึ้น และอาจจะคิดว่า ค่าใช้จ่ายในการผลิตจะเพิ่มขึ้น. แต่ถ้าทำการทดสอบโดยไม่ทำลายในระหว่างกระบวนการผลิต อย่างเคร่งครัด. จะทำให้ไม่มีการส่งผ่านผลิตภัณฑ์ที่เป็นของเสียไปยังกระบวนการต่อไปโดยไม่รู้ ล่วงหน้า ลดการสูญเสียในขั้นตอนต่อ ๆ ไปได้ ซึ่งทำให้สามารถลดต้นทุนการผลิตโดยรวมลงไป อุปมาเหมือนกับการเจ็บไข้ได้ป่วย ถ้ารู้อาการล่วงหน้าได้ยิ่งเร็วเท่าใด ก็สามารถ รักษาให้หายได้เร็วขึ้นเท่านั้น ทำนองเดียวกันถ้าได้ประยุกต์การทดสอบโดยไม่ทำลายตั้งแต่ขั้น ต้น ๆ ของกระบวนการผลิต ทำการดึงเอาผลิตภัณฑ์ของเสียออกและซ่อมแซมเพิ่มเติมแล้ว ก็ จะจัดการสูญเสียเปล่าในกระบวนการผลิต ประหยัดกระบวนการ และค่าใช้จ่ายโดยรวมไปได้ ซึ่ง เท่ากับเป็นการลดต้นทุนการผลิต

2.1.4 ประเภทของการทดสอบโดยไม่ทำลายเพื่อตรวจหารอยบกพร่องบริเวณผิว

2.1.4.1 การทดสอบโดยการตรวจพินิจ(Visual Testing)

ตามทฤษฎีแล้วจะใช้สายตาดู ในกรณีที่จะใช้แว่นขยาย, ไม้บรรทัดเว้า(Convex ruler), เกจชนิดพิเศษต่าง ๆ เพื่อวัดขนาดของรอยแตกรอยเหลื่อม(Overlapped) หลุมบ่อ ต่าง ๆ ความสูงของตะเข็บเชื่อม และการเชื่อมกินเนื้องาน (Undercut) เป็นต้น

2.1.4.2 การทดสอบโดยใช้ผงแม่เหล็ก(Magnetic Particle Testing)

สามารถใช้ตรวจสอบหารอยบกพร่องบริเวณผิวหรือภายใต้ผิวได้ แต่ไม่สามารถใช้กับ ชิ้นทดสอบที่มีสภาพความเป็นแม่เหล็กสูง

2.1.4.3 การทดสอบโดยใช้การซึมของของเหลว(Liquid Penetrant Testing)

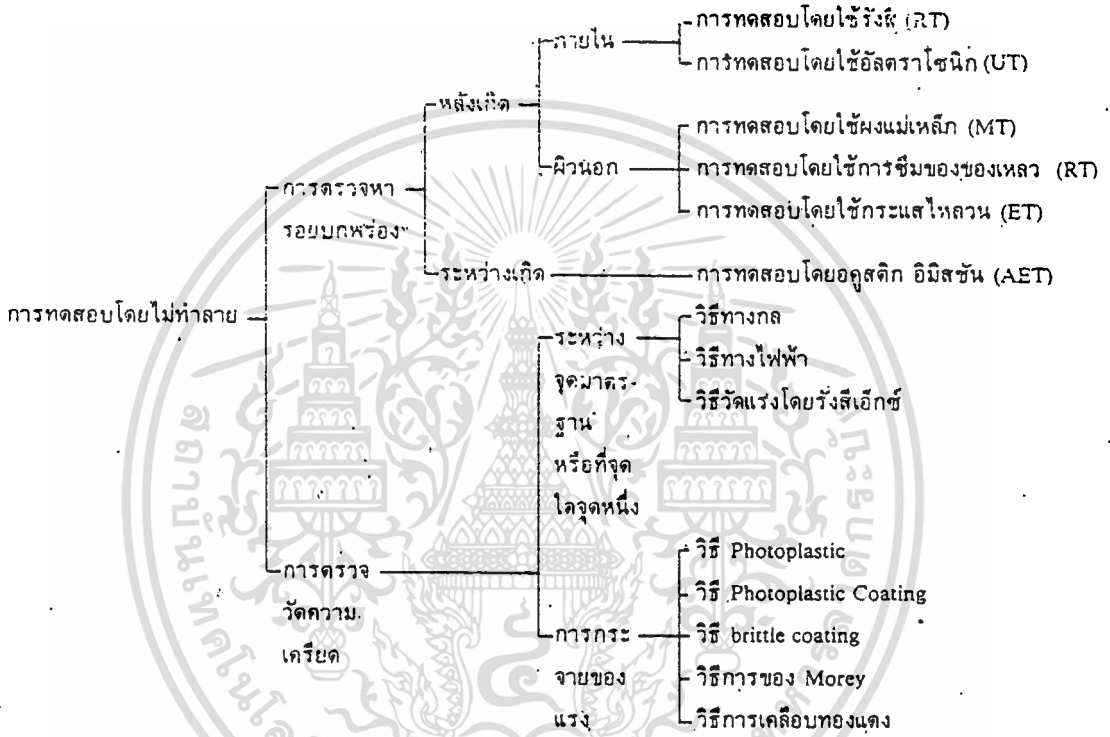
ใช้ตรวจหารอยแตกเฉพาะที่บริเวณผิวพื้นเท่านั้น ใช้กับวัสดุที่เป็นโลหะ และอโลหะ โดยทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.4 การทดสอบโดยกระแสไหลวน(Eddy Current Testing)

ใช้ตรวจสอบหารอยขีดข่วนที่ผิววัสดุที่เป็นตัวนำไฟฟ้าโดยไม่ต้องสัมผัส และสามารถตรวจสอบได้อย่างรวดเร็วใช้ในการตรวจสอบรอยแตกของแท่งหรือท่อโดยอัตโนมัติ

นอกจากการทดสอบแบบไม่ทำลายเพื่อตรวจหารอยบกพร่องบริเวณผิวแล้วยังมีการทดสอบโดยไม่ทำลายแบบอื่น ๆ อีกและสามารถสรุปเป็นไดอะแกรมได้ดังตารางที่ 2.1

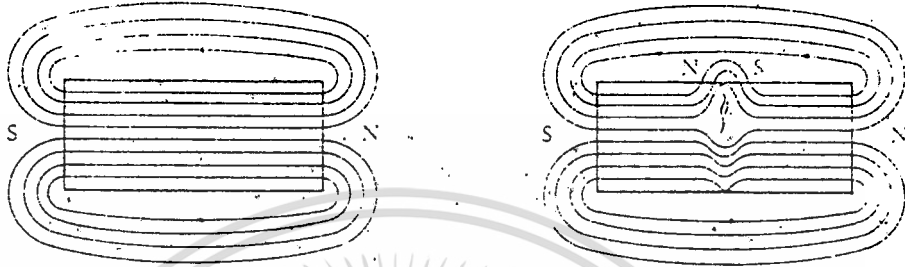


ตารางที่ 2.1 ชนิดของการทดสอบโดยไม่ทำลาย

2.2 การทดสอบโดยใช้วิธีการรั่วไหลของสนามแม่เหล็ก(Flux Leakage)

จะใช้ตรวจสอบหารอยบกพร่องบริเวณผิวหรือภายใต้ผิวได้ คล้ายกับการทดสอบโดยใช้ผงแม่เหล็ก ในการทดสอบนั้นเราจะทำให้ท่อเหล็กที่จะใช้ทดสอบเป็นแม่เหล็กเสียก่อน(วิธีการทำให้ท่อเหล็กเป็นแม่เหล็กจะกล่าวภายหลัง) ซึ่งสนามแม่เหล็กจะเหนี่ยวนำอยู่ภายในท่อเหล็กที่ใช้ทดสอบและจะเกิดเส้นแรงแม่เหล็กขึ้นภายในท่อ ถ้ามีจุดบกพร่อง(รอยร้าว, รอยร้าว)บริเวณผิวของท่อก็จะมีสนามแม่เหล็กรั่วออกมารอบ ๆ บริเวณนั้นเนื่องจากเกิดช่องว่าง ซึ่งมันจะสร้างขั้วแม่เหล็ก(ขั้วเหนือและขั้วใต้)ขึ้นบริเวณนั้นดังรูปที่ 2.11

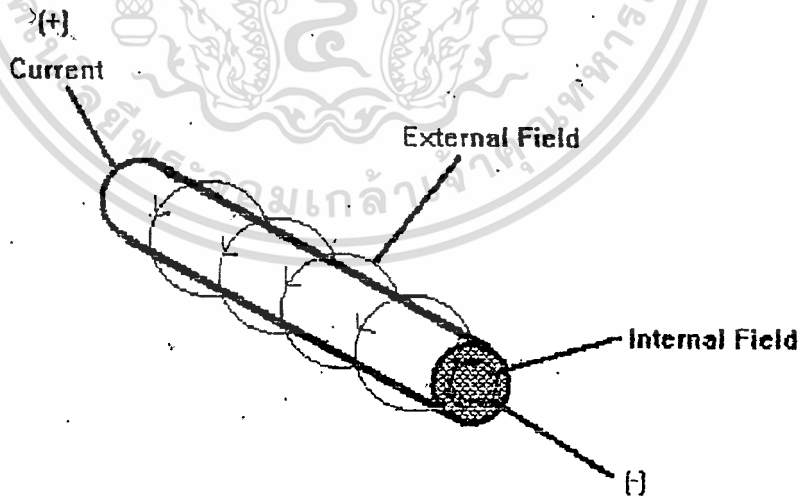
จากหลักการที่ว่าเมื่อมีขดลวดตัดผ่านสนามแม่เหล็กจะทำให้เกิดกระแสไฟฟ้าขึ้น เมื่อเราใช้ขดลวดเคลื่อนที่ผ่านผิวของท่อเหล็กบริเวณที่เกิดแม่เหล็ก ก็จะทำให้มีกระแสไฟฟ้าเกิดขึ้น จึงทำให้เราทราบว่าจะเกิดจุดบกพร่องบริเวณนั้น



รูปที่ 2.11 ลักษณะของเส้นแรงแม่เหล็กและแม่เหล็กที่เกิดขึ้น

2.1.4.5.1 การสร้างท่อให้เป็นแม่เหล็ก

ในการหมุนรอบตัวเองและการเคลื่อนที่รอบนิวเคลียสของ อิเล็กตรอนในอะตอมเป็นสาเหตุทำให้เกิดอำนาจแม่เหล็ก และอะตอมของสารทุกชนิด(ไม่เฉพาะสามแม่เหล็กเท่านั้น) มีอำนาจแม่เหล็กทั้งสิ้น



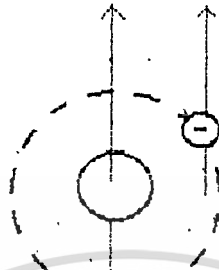
รูปที่ 2.12 การสร้างท่อให้เป็นแม่เหล็ก

อำนาจแม่เหล็กที่เกิดขึ้นจากการหมุนรอบตัวเองและ การหมุนรอบนิวเคลียสของ

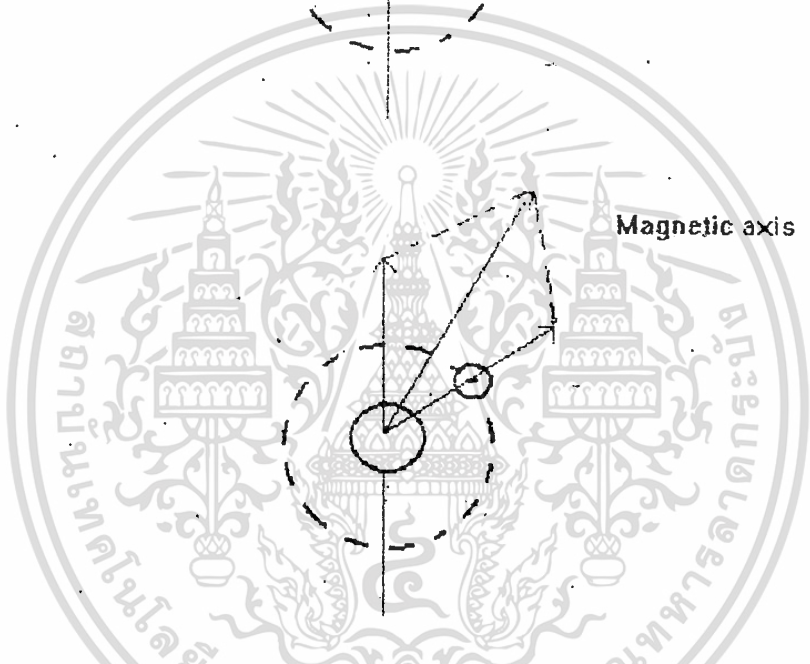
อิเล็กตรอนในอะตอมจะรวมตัวเรียงกันเป็นแกนแม่เหล็ก (magnetic axis) ถ้าหากว่าอำนาจแม่เหล็กที่เกิดที่ตัวอิเล็กตรอนเองและที่นิวเคลียสมีทิศทางไปทางเดียวกันก็สามารถเข้าด้วยกัน

อำนาจแม่เหล็กจาก
การหมุนรอบตัวเอง

อำนาจแม่เหล็กจาก
การหมุนรอบนิวเคลียส



(ก)



(ข)

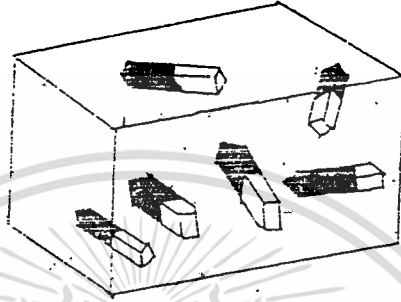
รูปที่ 2.13 ลักษณะการเกิดอำนาจแม่เหล็กจากการหมุนของอิเล็กตรอน

ได้โดยตรง แต่ถ้ามีทิศทางแยกออกจากกันเราต้องหาทิศทางของแกนแม่เหล็กรวม ซึ่งแสดงไว้ในรูปที่ 2.13 ข

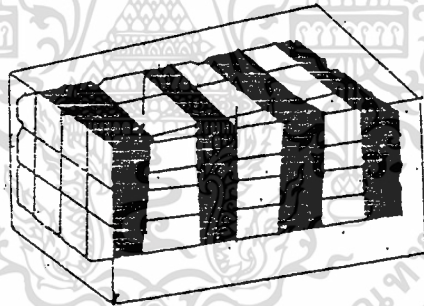
สารแม่เหล็กไม่สามารถแสดงอำนาจแม่เหล็กออกมาได้เพราะว่า เมื่ออะตอมรวมตัวกันเป็นโมเลกุลแต่โมเลกุลเหล่านั้น รวมตัวกันอย่างไม่เป็นระเบียบทำให้ แกนแม่เหล็กต่าง ๆ เหล่านั้นหักล้างกันหมด แล้วจะพบว่าโมเลกุลนั้น ๆ ก็จะมีอำนาจแม่เหล็กปรากฏเหลืออยู่ จากการวิจัยพบว่าภายในแท่งวัตถุ จะมีบริเวณเล็ก ๆ เป็นแท่ง ๆ ซึ่งมีอำนาจแม่เหล็ก แต่ละกลุ่มประกอบด้วยโมเลกุลที่มีอำนาจแม่เหล็กเรียกว่า "แมกเนติกโดเมน" (Magnetic Domain)

โดยปกติแกนแม่เหล็กของแมกเนติกโดเมนแต่ละแห่งจะไม่หันอยู่ในแนวเดียวกันแต่จะมีทิศทางไปคนละทิศคนละทาง จึงเป็นผลทำให้อำนาจแม่เหล็กของแมกเนติกโดเมนแต่ละแห่ง

หักล้างกันเองบ้างหรือ วนเวียนอยู่ในแท่งวัตถุบ้างไม่ปรากฏออกมานอกแท่งแม่เหล็กเมื่อใดที่สามารถทำให้แกนแม่เหล็กหันไปในทิศทางเดียวกันอย่างเป็นระเบียบได้ ก็จะปรากฏอำนาจแม่เหล็กออกมาจากแท่งวัตถุนั้น ซึ่งจะกลายเป็นแม่เหล็กโดยสมบูรณ์



รูปที่ 2.14 ก สารแม่เหล็กที่ไม่มีสภาพเป็นแม่เหล็ก



รูปที่ 2.14 ข สารแม่เหล็กที่มีสภาพเป็นแม่เหล็ก

2.3 ความเข้มของสนามแม่เหล็ก

ผลผลิตที่เป็นที่น่าพอใจนั้นสนามแม่เหล็กในส่วนประกอบจะต้องมีความเข้มพอเพียง การที่ผลที่แสดงจะไม่เปลี่ยนแปลงนั้นความเข้มสนามนี้จะต้องถูกควบคุมภายใต้ข้อจำกัดที่เหมาะสม ปกติ $\pm 25\%$ ตัวแปรซึ่งมีผลต่อความเข้มของสนามคือขนาด, รูปร่าง และวัสดุของส่วนประกอบ และเทคนิคของการทำแม่เหล็ก ซึ่งตัวแปรเหล่านี้มีการเปลี่ยนแปลงอย่างมาก มันยากที่จะตั้งกฎที่แน่นอนสำหรับความเข้มสนามสำหรับทุก ๆ ส่วนที่เป็นไปได้ของโครงสร้าง สำหรับโครงสร้างใหม่มันเป็นการดีที่จะใช้ทดลองกับส่วนที่ไม่ต่อเนื่องซึ่งกำหนดตามความเอกลักษณะนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการที่แท้จริง อย่างไรก็ตามสามารถนำผลการทดลองไปประยุกต์กำหนดระดับของการทำแม่เหล็กทรงกลมและตามแนวได้

ขั้นตอนในการทำแม่เหล็ก ผลของสนามแม่เหล็กที่เหมาะสมกับทั้งพื้นผิวและข้างพื้นผิวที่ไม่ต่อเนื่องกัน แนะนำว่ากระแส FWDC หรือ HW ควรจะใช้ในช่อง 700-900 แอมป์/นิ้ว(หรือ มม.) ในส่วนที่มีเส้นผ่านศูนย์กลาง 5 นิ้วขึ้นไป (125 มม.) 500-700 แอมป์/นิ้ว (หรือ มม.) บนส่วนที่มีเส้นผ่านศูนย์กลางมากกว่า 5 นิ้ว(125 มม.)และมากกว่า 15 นิ้ว (375 มม.) และ 100-300 แอมป์/นิ้ว(หรือ มม.) บนส่วนที่มีเส้นผ่านศูนย์กลางมากกว่า 15 นิ้ว (375 มม.) กระแสที่เปลี่ยนแปลงสนามแม่เหล็กสามารถใช้ประมาณครึ่งหนึ่งของค่าเหล่านี้สำหรับการตรวจพื้นผิวที่ไม่ต่อเนื่อง

ตัวเหนี่ยวนำสนามแม่เหล็กกลาง (Central Conductor Induced Magnetization) วิธีการใช้ตัวนำศูนย์กลาง(Central Conductor) ถูกนำมาใช้ประโยชน์อย่างกว้างขวางในแม่เหล็กที่ใช้ตรวจสอบอนุภาคดังต่อไปนี้:

สนามทรงกลมที่อยู่ภายในเส้นผ่านศูนย์กลางของท่อ ซึ่งไม่สามารถจำลองได้โดยเทคนิคกระแสตรง

การไม่สัมผัสกันจากกรรที่เกิดแม่เหล็กเสมือน ซึ่งตัดส่วนที่เป็น "burning"

การดำเนินการพิสูจน์หลักฐานซึ่งจะได้ประโยชน์นอกเหนือจากการใช้เทคนิคการสัมผัสกันโดยตรงบนส่วนที่เป็นรูปวงแหวน

โดยทั่วไปแล้วมันต้องการให้ตำแหน่งกลางของเซ็นทรัลคอนดักเตอร์ ยอมให้ส่วนรอบทั้งหมดเกิดขบวนการ ณ ช่วงเวลาหนึ่ง ผลของสนามอยู่ที่จุดเดียวกันจะสัมพันธ์กับแกนของส่วนประกอบและเป็นค่าสูงสุดที่เส้นผ่านศูนย์กลางภายใน กับตำแหน่งตรงกลางของเซ็นทรัลคอนดักเตอร์ กระแสที่ทำให้เกิดแม่เหล็กต้องการจะคล้ายกับส่วนที่เป็นแท่งซึ่งมีเส้นผ่านศูนย์กลางภายนอกเหมือนกัน ฉะนั้นก็เช่นเดียวกับบทแนะนำการใช้ประโยชน์ การกำหนดจุดกลางของเซ็นทรัลคอนดักเตอร์มีหลายวิธีที่คิดง่าย ๆ

การปรับค่า offset ที่เซ็นทรัลคอนดักเตอร์ เป็นวิธีที่ใช้บ่อยเมื่อกระแสที่ทำให้เกิดแม่เหล็กเกิดความบกพร่อง อย่างไรก็ตามกรณีของชิ้นงานที่มีเส้นผ่านศูนย์กลางภายนอกใหญ่และวัตถุที่ทำให้ให้สะดวกสำหรับส่วนเล็ก ๆ การเป็นแม่เหล็กกับการปรับค่า offset ของเซ็นทรัลคอนดักเตอร์ จะสังเกตว่าเส้นรอบวงทั้งหมดจะต้องได้รับการตรวจสอบ โดยการนำส่วนต่าง ๆ หลาย ๆ ชิ้นไปวางไว้บนเซ็นทรัลคอนดักเตอร์และสามารถมีประมาณไม่เกิน 10% ของแม่เหล็ก

ที่วางซ้อนอยู่ เป็นระยะทางตามเส้นรอบวงซึ่งมีผลต่อการเป็นสนามแม่เหล็กจะประมาณ 4 เท่าของเส้นผ่านศูนย์กลางของเซ็นทรัลคอนดักเตอร์

2.3.1 การเป็นแม่เหล็กที่ถูกจำกัด

การใช้ Prods กับ การกระตุ้น, การเป็นแม่เหล็กของทรงกลม, ความเข้มของสนามจะมีอัตราส่วนเป็นแอมป์. แต่จะเปลี่ยนแปลงกับ PROD SPACING และความหนาของหน้าตัดส่วนที่กำลังตรวจสอบ จึงแนะนำว่ากระแสที่ทำให้เกิดการเป็นแม่เหล็ก 90-110 แอมป์/นิ้ว (หรือ มม.) ของ PROD SPACING ควรจะใช้กับวัสดุที่มีความหนาดำกว่า 3/4 นิ้ว (19 มม.) และ 100-125 แอมป์/นิ้ว (หรือ มม.) ของ PROD SPACING ควรจะใช้กับวัสดุที่มีความหนามากกว่า 3/4 นิ้ว (หรือ 19 มม.) วงรอบพลังงานที่ยาวนานอาจเป็นสาเหตุที่ทำให้เกิดความร้อนสูงในวงรอบ PROD SPACING ไม่ควรเกิน 8 นิ้ว (200 มม.) ปกติ PROD SPACING ที่น้อยกว่า 3 นิ้ว (75 มม.) ไม่เหมาะสมที่จะใช้เนื่องจากความร้อนของอนุภาครอบ ๆ Prods

การใช้โย็ค ความเข้มของสนามไฟฟ้าของโย็คสามารถกำหนดค่าได้จากการทดลองโดยการวัดกำลังของมัน

การใช้ MANUAL CLAMPS/เทคนิคการใช้แม่เหล็กดูด

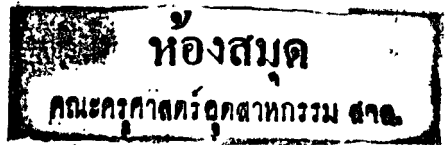
แม่เหล็กในแนวเส้นล่องจรวด ความเข้มของสนามแม่เหล็ก(บนหน้า) ส่วนของเส้นล่องจรวดของการทำให้เกิดสนามแม่เหล็ก ซึ่งผลิตโดยการให้กระแสไหลผ่านคอยล์ที่พันหลาย ๆ รอบในช่วงการหมุนรอบส่วนหรือหน้าตัดของส่วนที่ได้รับการตรวจสอบ การผลิตสนามแม่เหล็กขนานแกนคอยล์นี้จะใช้หน่วยการวัด คือ แอมป์รอบ (NI) (แอมป์คูณด้วยจำนวนรอบที่พันด้วยคอยล์ หรือ เคเบิล) ผลของสนามที่ขยายออกบนข้างใดข้างหนึ่งของคอยล์จะมีระยะทางประมาณเท่ากับรัศมีของคอยล์ที่ใช้ ส่วนด้านยาวควรตรวจสอบหน้าตัดให้มีขนาดไม่เกินความยาวนี้ ซึ่งมี 3 สูตรพื้นฐานในการทำแม่เหล็กตามยาวที่ใช้สำหรับการใช้กับองค์ประกอบ low และ high-fill ของคอยล์

Low-Fill Factor Coils ในกรณีนี้เส้นผ่านศูนย์กลางภายในของคอยล์ที่ให้คงที่ไว้จะใหญ่กว่าขนาดของส่วนประกอบ (น้อยกว่า 10% ของเส้นผ่านศูนย์กลางข้างใน) สำหรับการเป็นแม่เหล็กที่เหมาะสม ดังนั้นส่วนประกอบควรจะอยู่จุดที่ดีภายในคอยล์และปิดด้านในของคอยล์กับ Low Fill Factor นี้ พอที่จะทำให้ความเข้มของสนามเปลี่ยนตำแหน่งไปจากเดิมกับอัตราส่วนของความยาวต่อเส้นผ่านศูนย์กลาง (L/D) ระหว่าง 3 และ 15 ซึ่งคำนวณได้จากสมการดังนี้:

ก) สูตรที่ 1 (สำหรับตำแหน่งของส่วนถึงด้านข้างของคอยล์ (Low Fill Factor Coils):

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$NI = \frac{K}{L/D} (\pm 10\%)$$

เมื่อ

I = กระแสที่ถูกใช้ มีหน่วยเป็น แอมป์

N = จำนวนรอบในคอยล์

L = ส่วนความยาว มีหน่วยเป็น นิ้ว

D = ส่วนเส้นผ่าศูนย์กลาง มีหน่วยเป็น นิ้ว

K = 45000 (ค่าคงที่ที่ได้จากการทดลอง)

NI = กระแสรอบ

ตัวอย่างให้ส่วนยาว 15 นิ้ว เส้นผ่านศูนย์กลางภายนอก 5 นิ้ว อัตราส่วน L/D คือ 15/5 หรือ 3 ต้องการกระแสรอบ (NI = 4500/3) จึงจะเหมาะสมกับความเข้มของสนาม ในส่วนจะเป็น 1500 แอมป์รอบ ถ้าใช้คอยล์หรือเคเบิลพัน 5 รอบ กระแสที่คอยล์ต้องการคือ (I = 1500/5) = 3000 A ($\pm 10\%$)

ข) สูตรที่ 2 (สำหรับส่วนที่ถูกวางตำแหน่งกลางคอยล์) (Low Fill Factor Coils):

$$NI = \frac{K R}{(6L/D)-5} (\pm 10\%)$$

เมื่อ

L = ส่วนยาว มีหน่วยเป็น นิ้ว

D = ส่วนเส้นผ่านศูนย์กลาง มีหน่วยเป็น นิ้ว

R = รัศมีคอยล์ มีหน่วยเป็น นิ้ว

I = กระแสคอยล์ที่ใช้ มีหน่วยเป็น แอมป์

K = 43000 (ค่าคงที่ที่ได้จากการทดลอง)

NI = กระแสรอบ

ตัวอย่าง ให้ส่วนยาว 15 นิ้ว เส้นผ่านศูนย์กลางภายนอก 5 นิ้ว อัตราส่วน L/D คือ 15/5 หรือ 3 ถ้าใช้คอยล์หรือเคเบิลพัน 5 รอบ เส้นผ่านศูนย์กลาง 12 นิ้ว (รัศมี 6 นิ้ว) (i) แอมป์รอบที่ต้องการ คือ

$$NI = \frac{43000 \cdot 6}{(6 \cdot 3) - 5} \text{ หรือ } 19846$$

และ กระแสที่คอยล์ต้องการ คือ :

$$\frac{19846 \text{ หรือ } 3969 \text{ A } (\pm 10\%)}{5}$$

High Fill Factor Coils ในกรณีนี้ เช่น การพันคอยล์หรือเคเบิลจะถูกจัดให้คงที่ เส้นผ่านศูนย์กลางภายนอกของส่วนที่สมบูรณ์หรือใกล้จะสมบูรณ์จะเพิ่มเติมด้วยเส้นผ่านศูนย์กลางภายในของคอยล์ในระหว่างที่หมุนรอบ

ก) สูตรที่ 3 (ส่วนที่ถูกวางตำแหน่งภายใน High Fill Factor Coils) และอัตราส่วน L/D จะมีค่าเท่ากับ หรือ มากกว่า 3

$$NI = \frac{k}{(L/D)+2} (\pm 10\%)$$

เมื่อ

L = กระแสคอยล์ มีหน่วยเป็น แอมป์

N = จำนวนรอบในคอยล์

L = ส่วนยาว มีหน่วยเป็น นิ้ว

D = ส่วนเส้นผ่านศูนย์กลาง มีหน่วยเป็น นิ้ว

K = 35000 แอมป์รอบ, และ

NI = กระแสรอบ

ตัวอย่าง การประยุกต์ใช้สูตรข้างบนสามารถแสดงได้ดังนี้: ให้ส่วนยาว 10 นิ้ว เส้นผ่านศูนย์กลาง 2 นิ้ว อัตราส่วน L/D = 5 และกระแสรอบที่ต้องการ NI = 35000/(5+2) หรือ 5000 ($\pm 10\%$) แอมป์รอบ ถ้าใช้คอยล์หรือเคเบิลพัน 5 รอบ กระแสที่ต้องการ 5000/5 หรือ 1000 A ($\pm 10\%$)

สำหรับ L/D ที่น้อยกว่า 3 ข้อที่เป็นขั้น(เหล็กที่ใช้ทำแม่เหล็กจะมีขนาดประมาณเท่ากับเส้นผ่านศูนย์กลางส่วน) ควรจะใช้การเพิ่มอัตราส่วน L/D หรือ จะใช้วิธีการสร้างแม่เหล็กอีกวิธี เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น การลดกระแส สำหรับ อัตราส่วน L/D มากกว่า .15 ค่า L/D สูงสุด 15 ควรจะใช้สำหรับ
อ้างอิงกับสูตรที่กล่าวมาทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

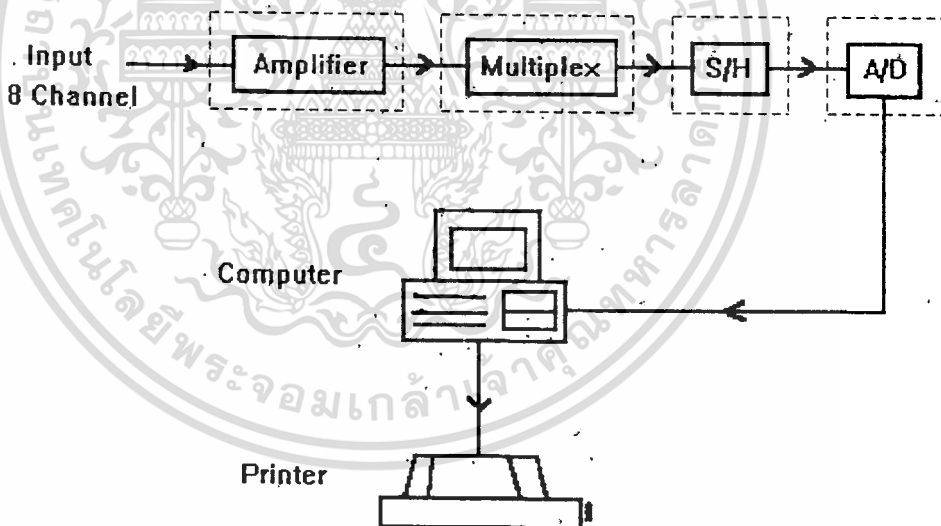
บทที่ 3

การออกแบบและการสร้าง

ในการออกแบบจัดสร้างตัวตรวจจับและแสดงผลของเครื่องตรวจสอบรอยร้าวของท่อเหล็กนี้ ได้แบ่งออกเป็น 2 ส่วนด้วยกันคือ ส่วนที่เป็น HARDWARE และ SOFTWARE

3.1 HARDWARE

ในส่วน จะมีการทำงานดังบล็อกไดอะแกรมดังรูป 3.1 ซึ่งจะมีลักษณะการทำงานคือ สัญญาณอินพุตทั้ง 8 แชนแนล จะถูกขยายสัญญาณโดยภาค Amplifier จากนั้นจะผ่านวงจรมัลติเพล็กซ์ เพื่อเลือกสัญญาณแล้วไปทำการ Sample & Hold เสร็จแล้วก็นำไปแปลงสัญญาณจากอนาลอกเป็นสัญญาณดิจิทัล แล้วจึงส่งสัญญาณไปที่คอมพิวเตอร์เพื่อแสดงผลทางจอภาพหรือพิมพ์ออกทางเครื่องพิมพ์



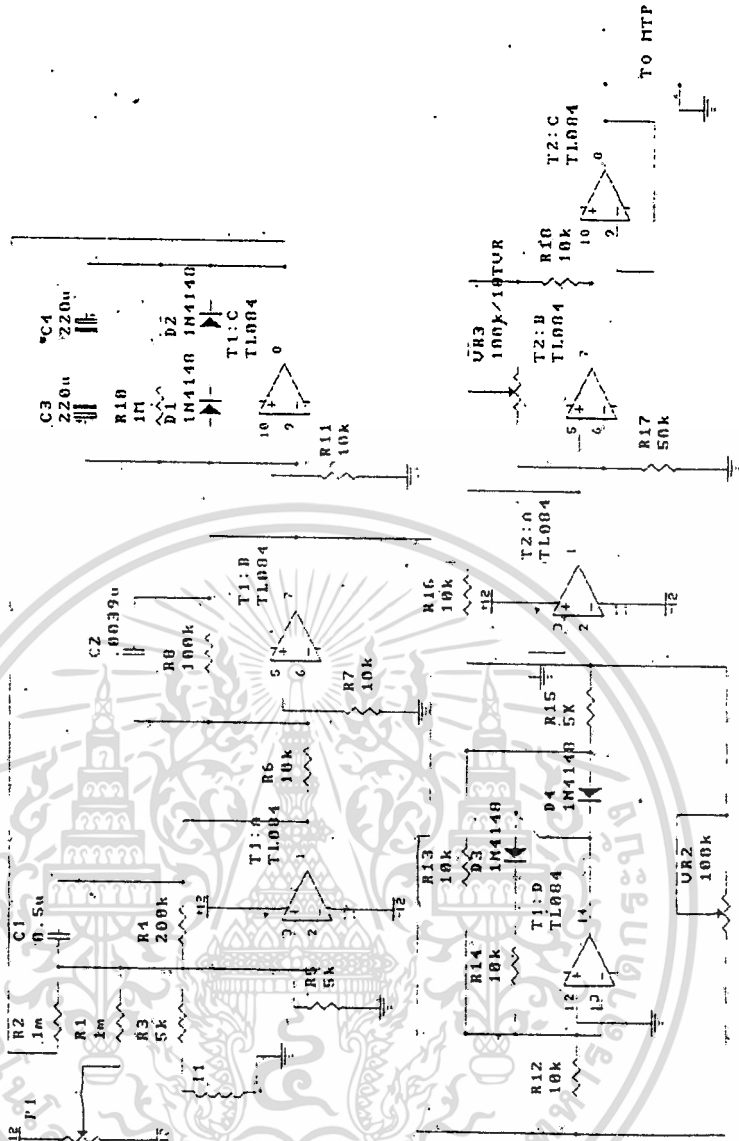
รูป 3.1 บล็อกไดอะแกรมการทำงาน

3.1.1 ภาค Amplifier

จะทำหน้าที่ขยายสัญญาณที่ได้รับมาจากหัว shoe (หัวอ่าน) ในขณะที่นำมาตรวจกับท่อเหล็กโดยถ้าเกิดรอยร้าวบนท่อเหล็ก ก็จะทำให้เกิดการรั่วไหลของแม่เหล็กที่สนามแม่เหล็ก จะทำให้สนามแม่เหล็ก ตัดกับขดลวดเกิดเป็นกระแสซึ่งกระแสที่ได้จะไหลผ่าน R_3 จากนั้นอปแอมป์ตัวแรกซึ่งทำหน้าที่เป็น Summing Integrator ซึ่งจะทำให้การรวมสัญญาณ(แรงดัน)ที่ป้อนกลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจรของภาค Amplifier

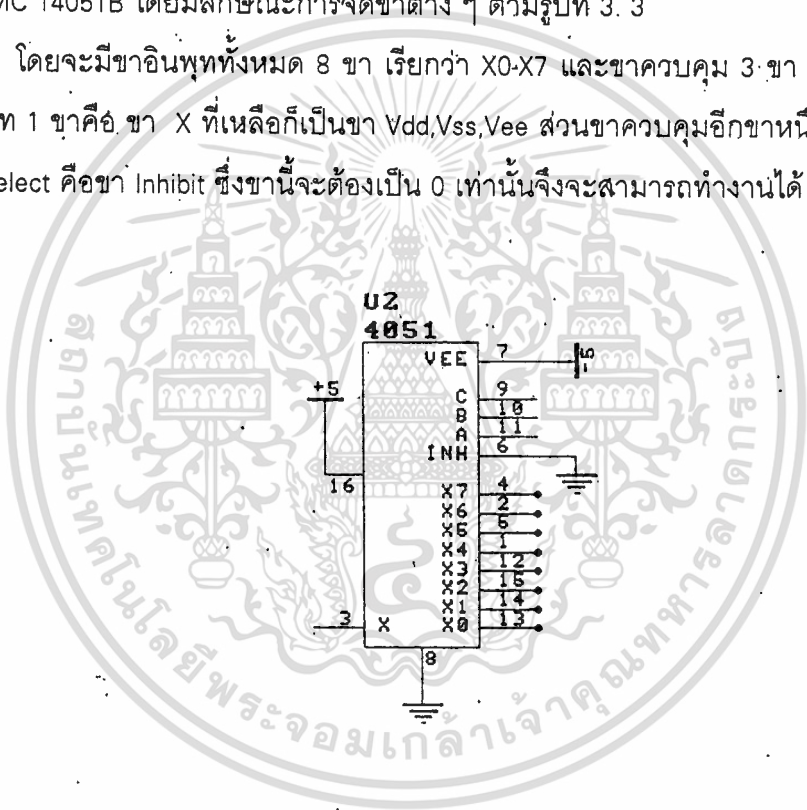
จาก เอาท์พุท หัก ล้างกับแรงดันที่อินพุท แล้วนำไปขยายต่อไปในลักษณะ Integrator. ซึ่งจะได้ คัดค่าที่เอาท์พุท ของออปแอมป์ตัวแรก จากนั้นจะส่งไปยังออปแอมป์ตัวที่สองเพื่อทำการขยาย สัญญาณให้สูงขึ้นจากนั้นจะส่งไปยังวงจร DC Amplifier เพื่อนำระดับแรงดัน DC ป้อนกลับมาจาก อินพุทเพื่อรักษาเกณฑ์ของวงจรให้คงที่ ส่วนเอาท์พุทจะถูกส่งไปอินพุทของวงจรกลับสัญญาณ โดยสัญญาณที่ป้อนเข้ามา 13 ขา Inverting. เมื่อสัญญาณมีศักย์เป็นลบเข้ามา เอาท์พุทจะออก ที่ขา 14 เป็นศักย์บวก กระแสไม่สามารถผ่าน D₄ ได้แต่จะผ่าน D₃ ป้อนกลับมาเป็นอินพุทอีกครั้ง ถ้าสัญญาณอินพุทที่มีเป็นบวกเข้ามาเอาท์พุทจะออกที่ขา 13 ผ่าน Inverting Amp ออกมา เป็นศักย์ลบกระแสไหลผ่านD₄ผ่านR₁₅ไปเป็นอินพุทให้กับ Inverting Amplifier ของออปแอมป์

ตัวต่อไป ซึ่งจะทำการกลับสัญญาณ แล้วออกมาที่ขา 1 ของออปแอมป์ จากนั้นก็ส่งต่อไปยังออปแอมป์ตัวสุดท้ายซึ่งทำหน้าที่เป็นบัฟเฟอร์ เพื่อส่งไปยังภาคมัลติเพล็กซ์ต่อไป ลักษณะของวงจรเป็นดังรูปที่ 3.2

3.1.2 ภาค Multiplex

ในการออกแบบเราต้องการใช้สัญญาณดิจิทัล 3 บิตในการเลือกสัญญาณ เราใช้ไอซีชนิด 3 to 8 Decoder ในวงจรที่จะนำมาใช้นี้เราจะเรียกว่า Analog Multiplexers. 3 to 8 คือไอซีเบอร์ MC 14051B โดยมีลักษณะการจัดขาต่าง ๆ ตามรูปที่ 3. 3

โดยจะมีขาอินพุตทั้งหมด 8 ขา เรียกว่า X0-X7 และขาควบคุม 3 ขา คือ A B C และมีเอาต์พุต 1 ขาคือ ขา X ที่เหลือก็เป็นขา Vdd,Vss,Vee ส่วนขาควบคุมอีกขาหนึ่งจะทำงานเป็น Chip Select คือขา Inhibit ซึ่งขา Inhibit นี้จะต้องเป็น 0 เท่านั้นจึงจะสามารถทำงานได้



(ก)

สถานะอินพุต				เอาต์พุต ที่ต่อกับ X
INHIBIT	A	B	C	
0	0	0	0	X0
0	0	0	1	X1
0	0	1	0	X2
0	0	1	1	X3

0	1	0	0	X4
0	1	0	1	X5
0	1	1	0	X6
0	1	1	1	X7
1	X	X	X	NONE

(ข)

รูปที่ 3.3 ก แสดงลักษณะการจัตขาของ MC 14051B

ข แสดงตารางความจริงของ MC 14051B

การทำงานของมันคือ วงจรภายในจะมีลักษณะคล้ายกับเป็นสวิตช์เลือก 8 ทาง แต่ควบคุมด้วยสัญญาณดิจิทัลสองที่ขา A B C โดยจะทำการต่อระหว่างขาอินพุตคือ X0-X7 ขาใดขาหนึ่งกับขาเอาต์พุต โดยการต่อจะขึ้นอยู่กับค่าดิจิทัลที่เข้ามา เช่นถ้าสมมุติว่าเราให้ A B A = 000 ก็จะเป็นการสั่งให้วงจรทำการต่อขา X0 เข้ากับ X และที่อินพุตอื่น ๆ จะไม่สนใจ แต่ถ้าเป็น 001 ก็จะเป็นการสั่งให้วงจรทำการต่อขา X1 เข้ากับ X ส่วนที่ค่าอื่น ๆ นั้นดูได้จาก ตารางความจริง และสังเกตว่าเราต้องให้ขา inhibit เป็น 0 ด้วยทุกครั้ง

3.1.3 ภาค Sample & Hold

ในโครงการนี้เราจะใช้ไอซี LF-398 ซึ่งเป็นแบบ Monolithic ที่ใช้คุณสมบัติของ BI-FET เทคโนโลยี ซึ่งจะมีความเร็วในการทำงานสูงและค่าความผิดพลาดต่ำ มีราคาถูก ใช้งานได้ง่าย แบนดิทท์กว้าง ลักษณะของ LF 398 แสดงดังรูปที่ 3.4

การทำงานของวงจรมันจะทำหน้าที่ Sampling สัญญาณอินพุตที่เข้ามาทางขา 3 ตามความถี่ Sampling ขนาด 50 KHz ซึ่งความถี่ที่นำมา Sampling จะเป็นสัญญาณสี่เหลี่ยมที่ได้มาจากภาคกำเนิดสัญญาณ Clock โดยความถี่ Sampling จะถูกป้อนเข้าที่ขา 8 ของ LF 398 มีเงื่อนไขว่าถ้าสัญญาณเป็นระดับ Logic 1 LF 398 จะทำการ Sample เอาสัญญาณอินพุตมา และเมื่อสัญญาณที่ขา 8 เปลี่ยนเป็น Logic 0 LF 398 ก็ทำการ Hold สัญญาณเอาไว้ด้วยตัวเก็บประจุ Ch. ซึ่งเราสามารถเลือกค่าได้ตามที่แสดงไว้ ในที่นี้เราใช้ค่า Ch = 100 pf ที่เอาต์พุตของ LF 398 คือที่ขา 5 จะได้ลักษณะดังรูปที่ 3.5

ช่วงเวลา Hold ประมาณ $10 \mu\text{sec}$ ก็ต้องใช้ค่า Ch ประมาณ 100 pf จะเหมาะสมกับช่วงเวลา Hold มากที่สุด

3.1.5 ภาค ANALOG TO DIGITAL หรือ A/D

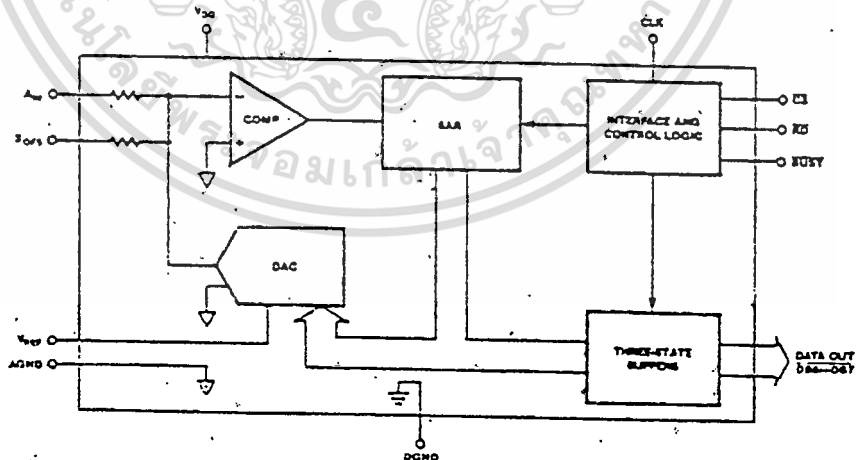
ในที่นี้เราเลือกใช้ไอซีเบอร์ ADC 908 สาเหตุที่เลือกเบอร์นี้เพราะมีจุดเด่นหลายประการ คือ

- มีความเที่ยงตรงสูง
- ทำงานได้ทั้งแบบ Unipolar และ Bipolar
- สามารถใช้กับสัญญาณขนาด $+10 \text{ Vp-p}$ ถึง -10 Vp-p แบบ Bipolar และ $0-20 \text{ Vp-p}$

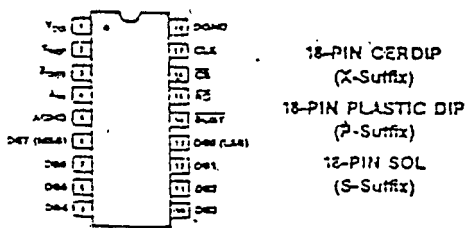
แบบ Unipolar

- ใช้ไฟเลี้ยง $+5 \text{ V}$ ชุดเดียว
- ช่วงเวลาที่ใช้ในการแปลงสัญญาณเพียง $6 \mu\text{sec}$
- มีวงจรผลิต Clock ในตัว
- ราคาถูก

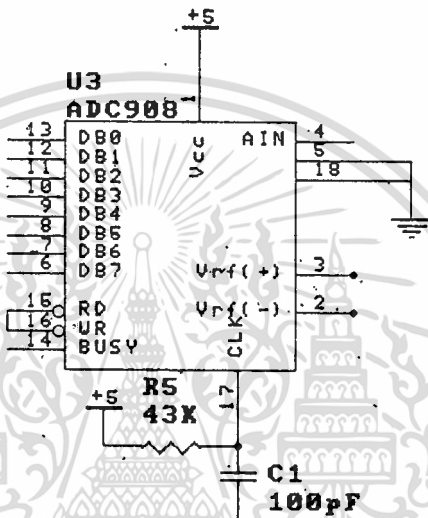
เราจะเห็นได้ว่ามีจุดเด่นอยู่มาก ที่นี้เราลองมาดูการทำงานของมัน ลักษณะภายในแสดงได้ดังในรูปที่ 3.6 ก และรูปที่ 3.6 ข แสดงตำแหน่งการวางขา



รูปที่ 3.6 ก แสดงวงจรภายในของ ADC 908



รูปที่ 3.6 ข. แสดงการจัดขาของ ADC 908



รูปที่ 3.7 แสดงวงจร A/D ที่ใช้ ADC 908

หน้าที่ของแต่ละขา

- ขาที่ 1 VDD เป็นไฟเลี้ยง +5 V
- ขาที่ 2 Vref เป็นขาของแรงดันอ้างอิง โดยทั่วไปจะใช้ไฟเลี้ยง +10 V
- ขาที่ 3 B_{ofs} เป็นขาการเลือกการทำงานที่ต้องการแบบชั่วคราวหรือสองชั่วคราวโดยทั่วไปถ้าต้องการให้ทำงานแบบสองชั่วคราวจะใช้แรงดัน +10 V
- ขาที่ 4 A_{in} เป็นขาป้อนสัญญาณอนาลอกอินพุต
- ขาที่ 6-13 เป็นขา DB0-DB7, คือดิจิตอลเอาต์พุต
- ขาที่ 14 BUSY เป็นขาที่บ่งบอกช่วงเวลาการแปลงสัญญาณ คือถ้าสัญญาณที่ขาที่เปลี่ยนแปลงจาก 1 มาเป็น 0 แสดงว่าเริ่มเปลี่ยนข้อมูล และเมื่อสัญญาณที่ขาที่เปลี่ยนแปลงจาก 0 มาเป็น 1 หมายถึงเสร็จสิ้นการเปลี่ยนข้อมูลและมีข้อมูลพร้อมอยู่บนขาเอาต์พุต

ขาที่ 15 RD เป็นขาที่จะใช้สำหรับสั่งให้เปลี่ยนข้อมูล เมื่อเราต้องการแปลงข้อมูลเป็นดิจิตอลเราจะให้ขานี้เป็น 0 และเมื่อใดที่ขานี้เป็น 1 หมายถึงการเคลียร์วงจรรับภายในหรือเคลียร์ข้อมูลทางเอาต์พุต

ขาที่ 16 CS เป็นขาที่สั่งให้พร้อมในการทำงาน

ขาที่ 17 CLK เป็นขาป้อนสัญญาณ Clock

ขาที่ 18 D_{gnd} เป็นขากกราวด์

เราต้องการให้ ADC ทำงานในแบบ Bipolar ดังนั้นเราต้องปรับขา B_{ofs} ดังรูปที่ 3.7

จากวงจรจะเห็นว่าเราต่อขา CS กับ RD ร่วมกันเพื่อประโยชน์ที่ว่าเราจะให้ ADC 908 พร้อมเมื่อเราต้องการอ่านข้อมูลเท่านั้น และสองขานี้จะต่อกับสัญญาณความถี่เดียวกับความถี่ที่ภาค Sample & Hold เพราะวงจรของเราจะแปลงสัญญาณเฉพาะในช่วงเวลาที่ Hold สัญญาณไว้ส่วนสัญญาณ clock ที่ใช้กับ ADC 908 จะเป็น Clock ที่กำเนิดขึ้นภายในโดยการจัดค่า R,C ที่ขา 17 ส่วนขา BUSY นั้นเป็นขาเอาต์พุตที่เราต้องใช้เพื่อให้ CPU มาตรวจสอบว่ามีข้อมูลพร้อมแล้วหรือไม่ โดยมีเงื่อนไขว่า CPU จะอ่านข้อมูลบน DATA BUS ได้ก็ต่อเมื่อสัญญาณที่ขา BUSY เป็น 1 และสัญญาณที่ขา RD เป็น 0 เท่านั้น

การทำงานของวงจรเมื่อเราต่อวงจรเรียบร้อยแล้วเราต้องปรับให้แรงไฟที่ขา 2 = -10 V และที่ 3 = +10 V เสียก่อนในวงจรของเรานั้น ADC 908 จะเป็นอิสระจาก CPU กล่าวคือมันจะแปลงข้อมูลโดยไม่สนใจว่า CPU จะสั่งมาหรือไม่มันจะเปลี่ยนสัญญาณไปเรื่อย ๆ โดย CPU จะคอยมาตรวจสอบที่ขา BUSY กับขา RD ว่าเป็นไปตามเงื่อนไขหรือไม่ ถ้าเป็นแล้วก็แสดงว่ามีข้อมูลที่ถูกต้องอยู่บน DATA BUS CPU ก็จะทำกรเก็บข้อมูลที่อยู่บน DATA BUS ในขณะนั้นเข้าไป และจะมารอตรวจสอบอีก ที่ขา RD กับ CS ของ ADC 908 ร่วมกันโดยมีสัญญาณที่ได้จากภาคกำเนิดความถี่ ซึ่งในขณะที่สัญญาณนี้เป็น 1 ก็จะเป็นการ Sample สัญญาณอนาล็อกเข้ามา ADC 908 จะไม่ทำงานแต่เมื่อใดที่สัญญาณเป็น 0 คือการ Hold สัญญาณ ADC 908 จะทำการเปลี่ยนสัญญาณอนาล็อกที่ถูก Hold ไว้เป็นสัญญาณดิจิตอล และจะส่งสัญญาณไปบอก CPU เมื่อได้ข้อมูลดิจิตอลแล้ว โดยในการเปลี่ยนแต่ละครั้งจะใช้เวลาเท่ากันคือ 5 μ sec แต่เราใช้สัญญาณ RD ที่มีเวลาเท่ากับ 10 μ sec จะเห็นว่าเรามีเวลาเหลือถึง 4 μ sec ที่จะมีสัญญาณดิจิตอลในขณะนั้นอยู่ซึ่งเพียงพอที่จะใช้เก็บข้อมูลและนำเอาไปแสดงผลได้ทัน

ในการทำงานที่เป็นลักษณะ Bipolar ของ ADC 908 นั้นระดับการควอนไทซ์ จะมีลักษณะที่พิเศษ คือ บิทที่มีนัยสำคัญสูงสุดจะเป็นบิทเครื่องหมาย ถ้าบิทนี้เป็น 1 แสดงว่าข้อมูลอยู่ในช่วงบวกและถ้าบิทนี้เป็น 0 แสดงว่าข้อมูลอยู่ในช่วงลบ ส่วนบิทที่เหลืออีก 7 บิทจะ

เป็นบิตข้อมูลโดยแต่ละ Step ของการควอนไทซ์นั้นจะแตกต่างกัน คือในขณะที่ข้อมูลดิจิทัลห่างกันอยู่ 1 เช่น 0000 0001 กับ 0000 0010 จะมีความแตกต่างของแรงดันตาม คือ 80 mV เมื่อเป็นสัญญาณแบบ Bipolar ก็ย่อมจะต้องมีระดับอ้างอิงของสัญญาณ โดยอนาลอกอินพุทที่เข้ามาจะอ้างอิงที่ระดับ 0 แต่ข้อมูลดิจิทัลที่ได้จะมีระดับอ้างอิงที่ 0111 1111 ถ้าได้ข้อมูลตัวนี้ออกมาแสดงว่าเป็นระดับหรือจุดตัดแกนศูนย์ของสัญญาณอนาลอก

เอาท์พุทของ ADC 908 นั้นเราจะต่อเข้ากับชุด INTERFACE โดยให้ต่อกับ PORT INPUT และเราจะต่อสัญญาณ BUSY และ RD เข้าที่ PORT INPUT อีกตัวหนึ่งเพื่อที่ CPU จะสามารถมาตรวจสอบได้ว่า มีข้อมูลพร้อมบน DATA BUS หรือไม่

3.1.6 รายละเอียดของสัญญาณต่าง ๆ บนสล๊อต

ในการที่จะเชื่อมต่ออุปกรณ์ร่วมกับคอมพิวเตอร์เราจำเป็นต้องติดต่อผ่านสล๊อต เราจึงควรที่จะรู้จักรายละเอียดของสัญญาณต่าง ๆ บนสล๊อต ซึ่งมีรายละเอียดดังต่อไปนี้

(I), (O) และ (I,O) หมายถึง ทิศทางของขาสัญญาณเมื่อเทียบกับเมนบอร์ด

โดยที่ (I) หมายถึง ขาสัญญาณอินพุท

(O) หมายถึง ขาสัญญาณเอาท์พุท

(I,O) หมายถึง ขาสัญญาณที่เป็นได้ทั้งอินพุทและเอาท์พุท

(*I,O) หมายถึง ในช่วงการทำงานปกติจะเป็นขาสัญญาณเอาท์พุท แต่จะเป็น

อินพุทในช่วงที่เกิดขบวนการ DMA

สัญญาณที่ต่ออยู่บนสล๊อตนี้สามารถขับไอซีที่ทีแอลชนิด Low-power ได้สองตัว โดยไม่ทำให้เกิดการโหลด หรือ การเพี้ยนของสัญญาณ ขาสัญญาณต่าง ๆ บนสล๊อตของ AT และ XT สามารถแบ่งออกเป็นกลุ่ม ๆ ได้ดังนี้

3.1.6.1 เพาเวอร์ซัพพลาย

GROUND	ขาสัญญาณนี้ต่ออยู่กับกราวด์ของระบบเรกูเลเตอร์
+5 V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเรเตอร์ +5 V
-5 V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเรเตอร์ -5 V
+12 V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเรเตอร์ +12 V
+12 V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเรเตอร์ -12 V

เอกสารนี้เป็นเอกสาร 3.1.6.2 แอดเดรสบัส และ สัญญาณต่าง ๆ ที่เกี่ยวข้อง นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SA0-SA19 เป็นแอดเดรสบิตที่ 0 ถึง 19 โดยที่ มีนัยสำคัญต่ำที่สุด ขาสัญญานนี้ จะแอกทีฟ เมื่อขาสัญญาน มีสถานะเป็น 1 และจะถูกแลทช์ไว้ตอนขอบขาของ สัญญาน แอดเดรสทั้ง 20 บิตนี้ สามารถอ้างหน่วยความจำได้ถึง 1 MB สำหรับ XT และ สำหรับ AT เมื่อใช้ร่วมกับ LA17-LA23 จะอ้างได้ถึง 15 MB

LA17-LA23 (เฉพาะรุ่น AT) ขาสัญญานนี้จะแอกทีฟ

(*I/O) เมื่อขาสัญญาน BALE มีสถานะเป็นลอจิก 1 แต่จะไม่มีแลทช์ไว้ ตอนขอบขาของสัญญาณ ดังนั้นถ้าอุปกรณ์ ไม่มีการอ้างแอดเดรสเกิน 1 MB ขาสัญญานนี้ ก็ไม่จำเป็นต้องใช้ แต่ถ้ามีการอ้างแอดเดรสเกิน อุปกรณ์ I/O จะต้องทำการแลทช์สัญญาณนี้ โดยใช้ขอบขาของสัญญาณ BALE ร่วมกับขาสัญญาน MEMW และ MEMR

AEN (ADDRESS ENABLE) ขาสัญญานนี้จะแอกทีฟ

(O) เมื่อตัวควบคุม DMA ได้ทำการควบคุมบัสต่าง ๆ ของระบบแล้ว ดังนั้น การอ้างพอร์ตของอุปกรณ์ I/O จะต้องใช้สัญญาณนี้ในการติโคดด้วย เพื่อที่จะไม่ทำให้เกิดการ ติดต่อระหว่างระบบกับอุปกรณ์ I/O ตัวอื่นยกเว้นตัวที่ทำขบวนการ DMA อยู่

BALE (ADDRESS LATCH ENABLE) ขาสัญญานนี้ใช้ใน

(O) การแสดงการเริ่มต้นของขบวนการต่าง ๆ ที่มีการติดต่อกับหน่วยความ จำโดยจะแอกทีฟเมื่อค่าแอดเดรสที่ CPU ต้องการติดต่อดำเนินการเรียบร้อยแล้ว ตาม ปกติขอบขาของสัญญาณนี้จะทำให้เกิดการแลทช์สัญญาณ SA0-SA19 และถ้ามีการอ้าง แอดเดรสเกิน 1 MB ใน AT จะใช้ขอบขาของสัญญาณนี้ในการแลทช์สัญญาณ LA17-LA23 ด้วยเช่นกัน แต่สำหรับในขบวนการ DMA สัญญาณนี้จะมีค่าเป็น 1 ตลอด

SBHE (BUS HIGH ENABLE) เฉพาะรุ่น AT เป็นขา

(*I/O) สัญญาณที่ใช้แสดงว่ามีการรับส่งข้อมูลในบิตที่ SD8-SD15

3.1.6.3 ดาต้าบัส

SD0-SD7 สำหรับรุ่น AT จะมี SD0-SD15 เพิ่มขึ้นมา

(I/O) ด้วยกันคือ ดาต้าบิต 0 ถึง 7 สำหรับรุ่น XT และสำหรับรุ่น AT คือ ดาต้าบิต 0 ถึง 15 โดยที่ SD0 มีนัยสำคัญต่ำที่สุด สำหรับ AT ถ้ามีการติดต่อกับบิตที่ SD8-SD15 สามารถตรวจสอบได้จากขาสัญญาน SBHE

3.1.6.4 สัญญาณอินเตอร์รัพท์

เอกสารนี้เป็นเอกสาร IRQ2-IRQ7 สำหรับ (INTERRUPT REQUEST) สำหรับรุ่น AT จะเป็น IRQ3-7,9-12,14-15 การค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(II) เป็นคำสั่งสัญญาณอินเทอร์รัพต์ CPU สำหรับ AT ลำดับความสำคัญ ของสัญญาณ เป็นดังนี้ คือ 9,10,11,12,14,15,3,4,5,6 และ 7 โดย IRQ9 มีลำดับความสำคัญมากที่สุดและ IRQ7 มีลำดับความสำคัญน้อยที่สุด สำหรับ XT IRQ2 จะมีลำดับความสำคัญมากที่สุด รอง ลงไปคือ IRQ3,4,5,6,7

โดยปกติสัญญาณนี้จะมีสถานะเป็น 0 เสมอ ถ้าต้องการอินเทอร์รัพต์ ให้ส่งพัลส์ที่เป็น ลอจิก 1 ให้กับมันโดยไม่จำเป็นต้องคำนึงถึงคาบเวลาของพัลส์ ทั้งนี้เพราะระบบของ IBM ตัว อินเทอร์รัพต์คอนโทรลเลอร์(8259 Interrupt Controller) จะถูกโปรแกรมให้ทำการตรวจสอบ สัญญาณอินเทอร์รัพต์โดยใช้ขอบขาลงของสัญญาณนี้

I/O CH CK (I/O CHANNEL CHECK) เป็นคำสั่งสัญญาณที่บอกถึงความผิดพลาดใน การรับส่งข้อมูลซึ่งตรวจสอบจากพาริตีบิต ถ้าพาริตีบิตที่อ่านจากหน่วยความจำกับพาริตีที่ สร้างขึ้นจากขบวนการรับส่งข้อมูลมีค่าไม่เท่ากันแสดงว่าเกิดความผิดพลาดในการรับส่งข้อมูล สัญญาณนี้จะทำให้เกิดการอินเทอร์รัพต์ CPU แบบ NMI เพื่อบอกให้ CPU ทราบว่าเกิด Parity Error ขึ้น CPU จะแสดงข้อความบอกความผิดพลาดขึ้นและหยุดการทำงาน (HALT) เพื่อให้ผู้ใช้ ตรวจสอบหาสาเหตุของการผิดพลาด

3.1.6.5 สัญญาณควบคุมต่าง ๆ

MEMR (*I/O) MEMORY READ (สำหรับรุ่น AT คือคำสั่งสัญญาณ SMEMR) ขา สัญญาณนี้จะเป็นตัวบอกให้หน่วยความจำส่งข้อมูลออกมาที่ดาต้าบัส แต่สำหรับ AT สัญญาณ

SMEMR จะแอกทีฟเมื่อเกิดการอ่านข้อมูลที่อยู่ในหน่วยความจำ 1 MB แรกเท่านั้น

MEMR (O) เฉพาะรุ่น AT ขาสัญญาณนี้จะไม่เหมือนกับ MEMR ใน XT มัน จะแอกทีฟในทุก ๆ ขบวนการอ่านที่เกิดขึ้นไม่ว่าจะอยู่ในหน่วยความจำ 1 MB หรือไม่

MEMW (*I/O) MEMORY WRITE สำหรับรุ่น AT คือคำสั่งสัญญาณ SMEMW ขา สัญญาณนี้จะเป็นตัวบอกให้หน่วยความจำเก็บข้อมูลจากดาต้าบัส สำหรับ AT ขา สัญญาณนี้จะแอกทีฟเมื่อเกิดการเก็บข้อมูลจากหน่วยความจำที่อยู่ใน 1 MB แรกเท่านั้น

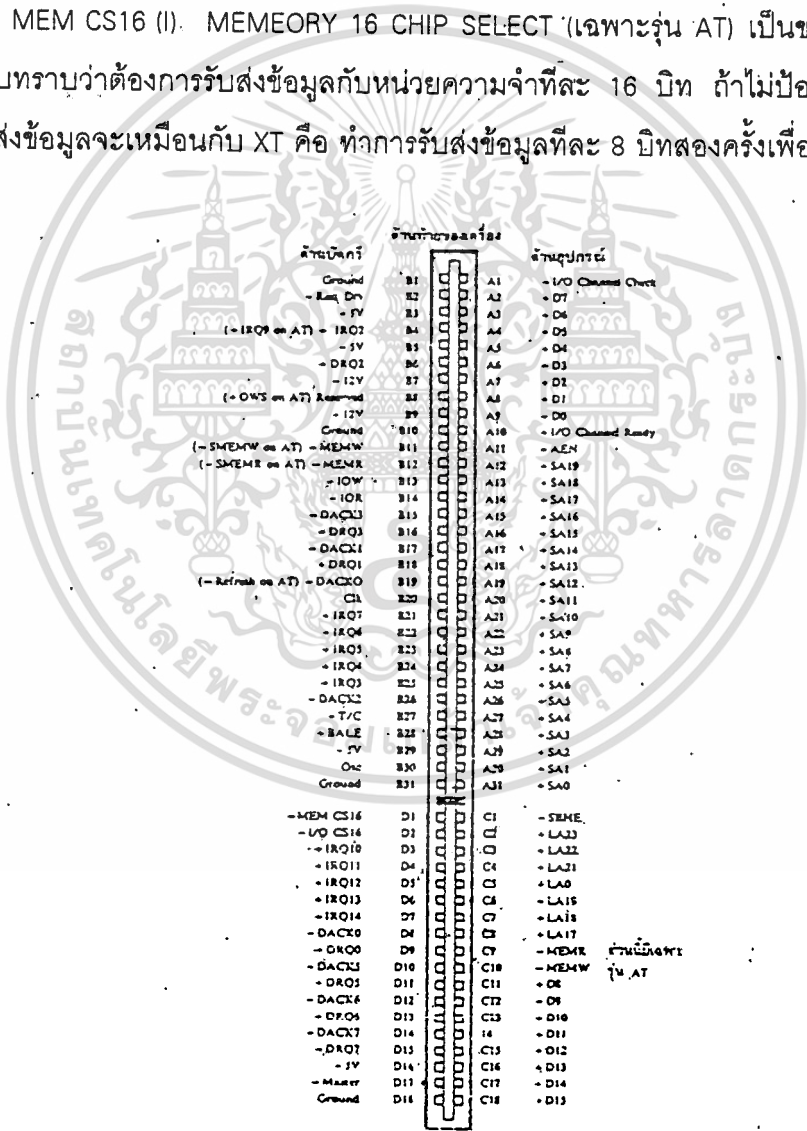
MEMW (O) (เฉพาะรุ่น AT) ขาสัญญาณนี้ไม่ใช่ขาสัญญาณเดียวกับขา MEMW ใน XT โดยขาสัญญาณนี้จะแอกทีฟในทุก ๆ ขบวนการเก็บข้อมูลที่เกิดขึ้นไม่ว่าจะอยู่ในช่วงหน่วย ความจำ 1 MB แรกหรือไม่

IOR (*I/O) I/O READ เป็นขาสัญญาณที่บอกให้อุปกรณ์ภายนอกที่ต่ออยู่ทำการส่งข้อมูลลงมาที่ดาต้าบัส

IOW (*I/O) I/O WRITE เป็นขาสัญญาณที่บอกให้อุปกรณ์ภายนอกที่ต่ออยู่ทำการเก็บข้อมูลจากดาต้าบัสเข้าไป

RESET DRV (O) RESET DRIVER เป็นขาสัญญาณที่แอดที่พตอนช่วงที่เราเริ่มจ่ายไฟให้กับระบบเพื่อใช้ในการรีเซท CPU และอุปกรณ์ต่าง ๆ ในระบบคอมพิวเตอร์ รวมทั้งอุปกรณ์ I/O ที่ต่ออยู่ด้วย

MEM CS16 (I) MEMEORY 16 CHIP SELECT (เฉพาะรุ่น AT) เป็นขาสัญญาณที่บอกให้ระบบทราบว่าต้องการรับส่งข้อมูลกับหน่วยความจำที่ละ 16 บิต ถ้าไม่ป้อนสัญญาณที่ขาขึ้น การรับส่งข้อมูลจะเหมือนกับ XT คือ ทำการรับส่งข้อมูลทีละ 8 บิตสองครั้งเพื่อให้ได้ข้อมูลขนาด 16 บิต



รูปที่ 3.8 แสดงตำแหน่งของสัญญาณต่าง ๆ บนสลีต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

I/O CS16 I/O 16 CHIP-SELECT (เฉพาะรุ่น AT) เป็นขาสัญญาณที่บอกให้ระบบทราบว่าต้องการรับส่งข้อมูลกับอุปกรณ์ I/O ทีละ 16 บิต ถ้าไม่ป้อนสัญญาณนี้ การรับส่งข้อมูลจะทำเหมือนกับ XT คือทีละ 8 บิตสองครั้ง

3.1.6.6 สัญญาณนาฬิกา

CLK (O) SYSTEM CLOCK สำหรับ ขาสัญญาณนี้จะมีค่าประมาณ 4.77 MHz หรืออาจมากกว่าสำหรับรุ่นใหม่ ๆ และสำหรับ AT ขาสัญญาณนี้จะมีค่าประมาณ 6 MHz หรือในรุ่นใหม่ ๆ อาจมีค่าที่สูงกว่า

OSC (O) OSCILLATOR เป็นขาสัญญาณที่มีความถี่สูง คือ 14.318 MHz ค่าของความถี่ของสัญญาณนี้จะคงที่เสมอ และไม่ซิงโครนัสกับสัญญาณอื่น ๆ ในระบบ ดังนั้นจึงไม่ควรนำสัญญาณนี้ไปใช้เป็นสัญญาณ Clock ของอุปกรณ์ I/O ที่ต่ออยู่กับระบบ

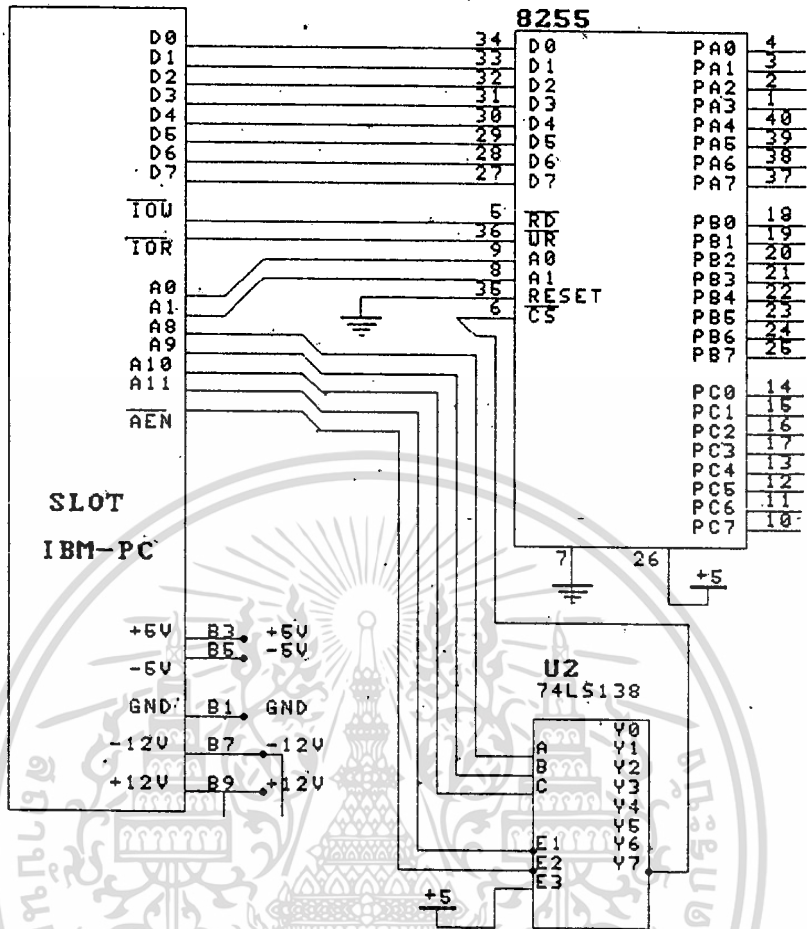
3.1.7 การ INTERFACE โดยใช้ 8255

การที่เราจะนำเอาข้อมูลจากอุปกรณ์ภายนอกเข้าไบนั้น เราจะใช้ 8255 PPI ในวงจรที่เราออกแบบนี้เราออกแบบจะกำหนดหน้าที่การทำงานของแต่ละ PORT ดังนี้ คือ PORT A เป็นเอาต์พุต ,PORT B เป็นอินพุต ,PORT C เป็นตัวควบคุม และในการ DECODE NUMBER PORT นั้นเราใช้ไอซีเบอร์ 74LS 138 ที่สามารถอ้างได้ 4 PORT

ในการออกแบบวงจร DECODER เนื่องจากวงจรของเราเป็นอุปกรณ์ภายนอกที่เป็นแบบ I/O สัญญาณที่นำมาใช้ในวงจร DECODE PORT มีดังนี้

ADDESS BUS	ใช้สำหรับถอดรหัสเฉพาะตำแหน่งที่ CPU เรียกมาที่ 0700H
I/OR	เป็นสัญญาณที่ส่งมาจาก CPU ว่าต้องการอ่านข้อมูลจากดาต้าบัส
I/OW	เป็นสัญญาณที่ส่งมาจาก CPU ว่าต้องการเขียนข้อมูลลงบนดาต้าบัส
AEN	เราจะใช้ ADDRESS BUS จากที่ CPU เรียกมาได้ก็ต่อเมื่อที่ขานี้เป็น 0

A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 X X
 = 0700H-0703H



รูปที่ 3.9 แสดงวงจร DECODE ที่ใช้ 74138

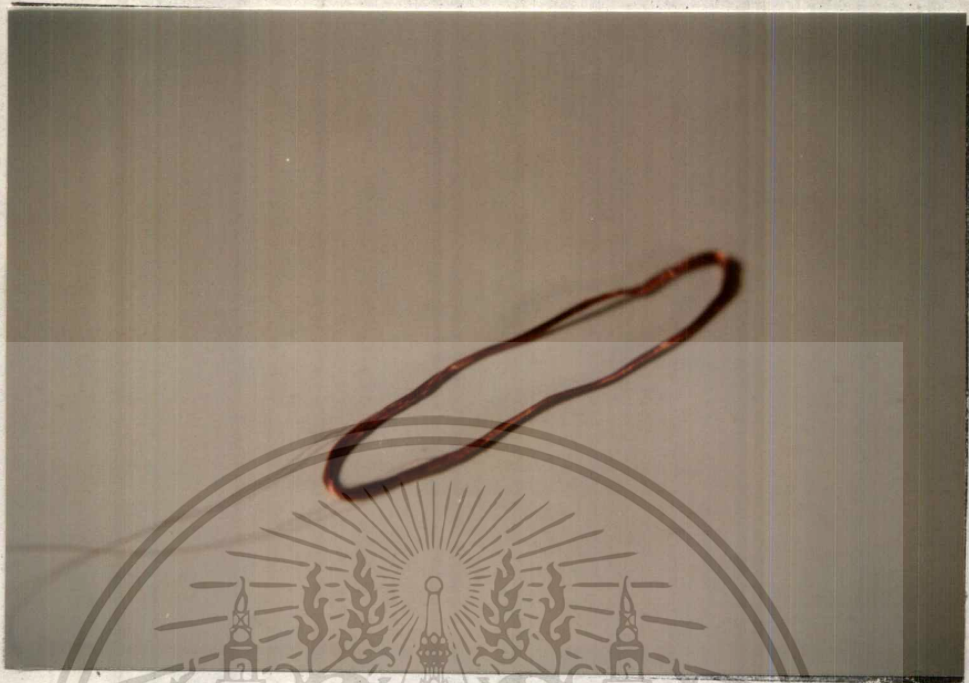
3.1.8 การสร้างหัวอ่าน หรือ Shoe

ในส่วนของหัวอ่านหรือ Shoe ในการสร้างนั้นเราจะใช้ขดลวดเบอร์ 45 พัน ประมาณ 40-50 รอบดังรูป 3.10

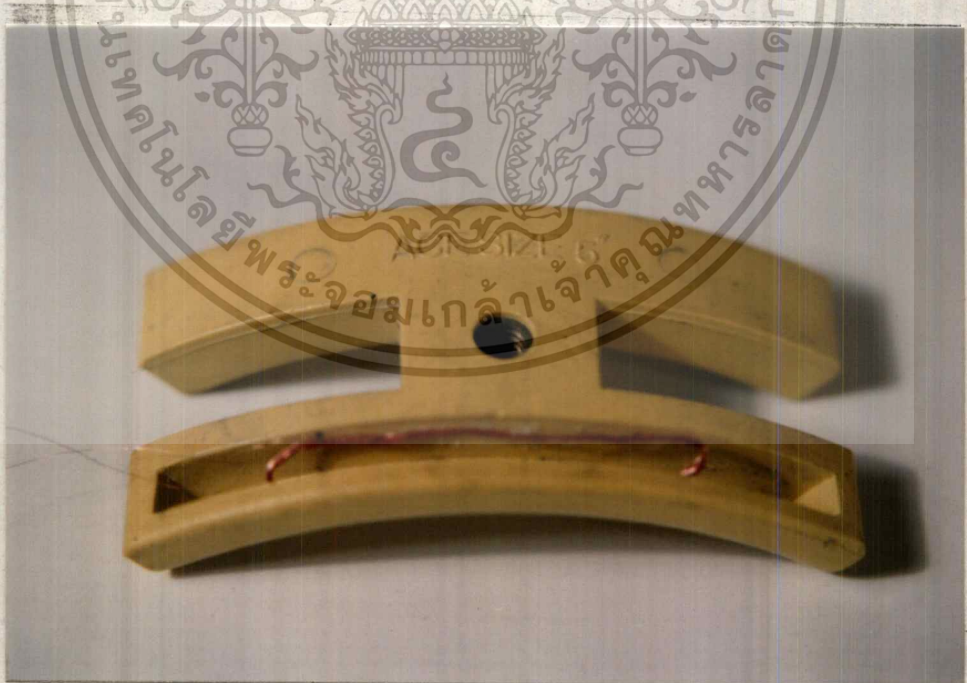
จากนั้นเราก็จะนำขดลวดใส่ลงไปใน Shoe ดังรูป 3.11 ก แล้วต่อสายไฟออกมาภายนอก ดังรูป 3.11 ข ในการต่อสายไฟออกมานั้นเราจะต้องใจเย็น ๆ ค่อย ๆ ต่อเพราะว่าถ้าเราต่อสายไม่ดี หรือบัดกรีไม่ดี จะทำให้ขดลวดไม่ต่อกับสายไฟ ซึ่งจะทำให้ไม่สามารถส่งสัญญาณออกมาได้ ส่วนสายไฟที่ใช้ ไม่ควรนำสายไฟธรรมดามาต่อควรจะใช้สายชีลด์เพราะจะสามารถลดสัญญาณรบกวนจากภายนอกได้

ในการสร้าง shoe นั้นเราจะทำทั้งหมด 8 หัว และจะมีวงจรขยายสัญญาณ(มี 8 ชุด เช่นกัน) ดังรูปที่ 3.13 ก เหตุที่ต้องทำ shoe ถึง 8 ชุดก็คือเราจะนำ shoe ไปประกบรอบ ๆ ท่อ

โลหะ

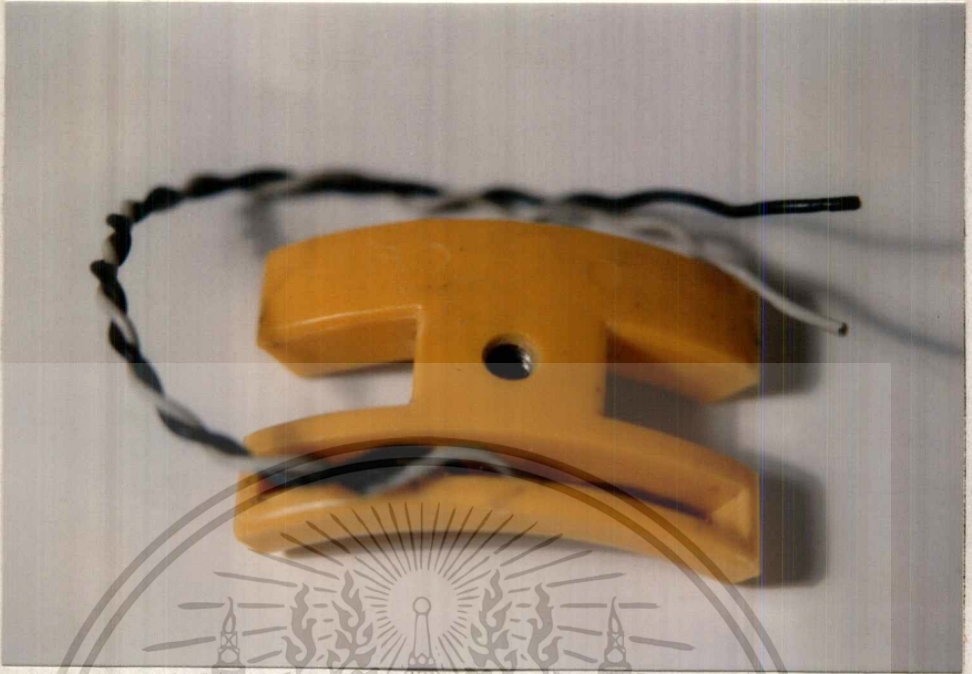


รูปที่ 3.10 ลักษณะของขดลวด



รูปที่ 3.11 ก ลักษณะของขดลวดที่ใส่ลงใน shoe

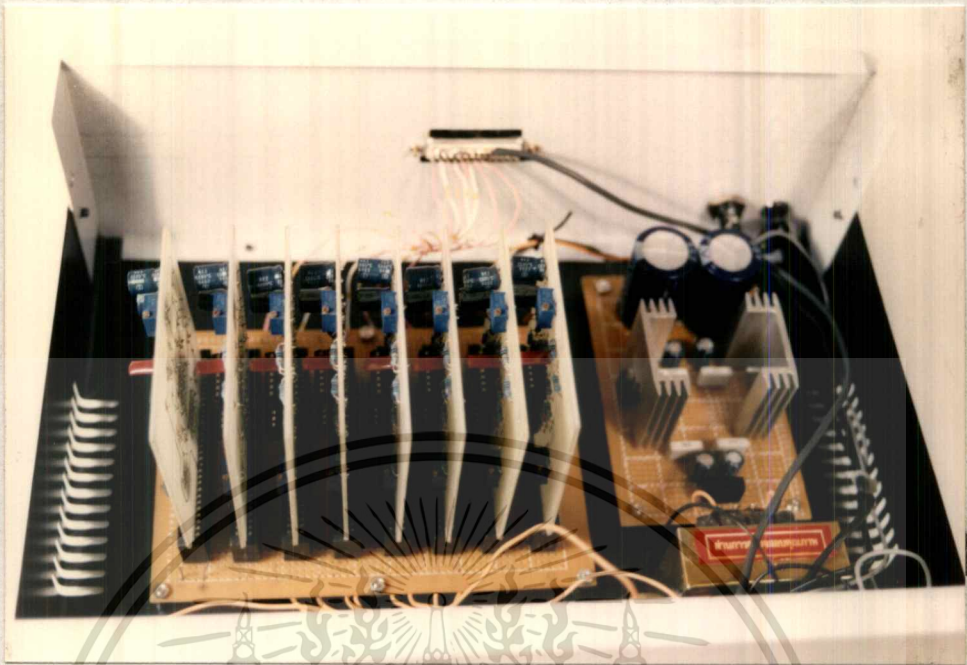
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ข ลักษณะของ shoe ที่ต่อสายไฟเรียบร้อยแล้ว



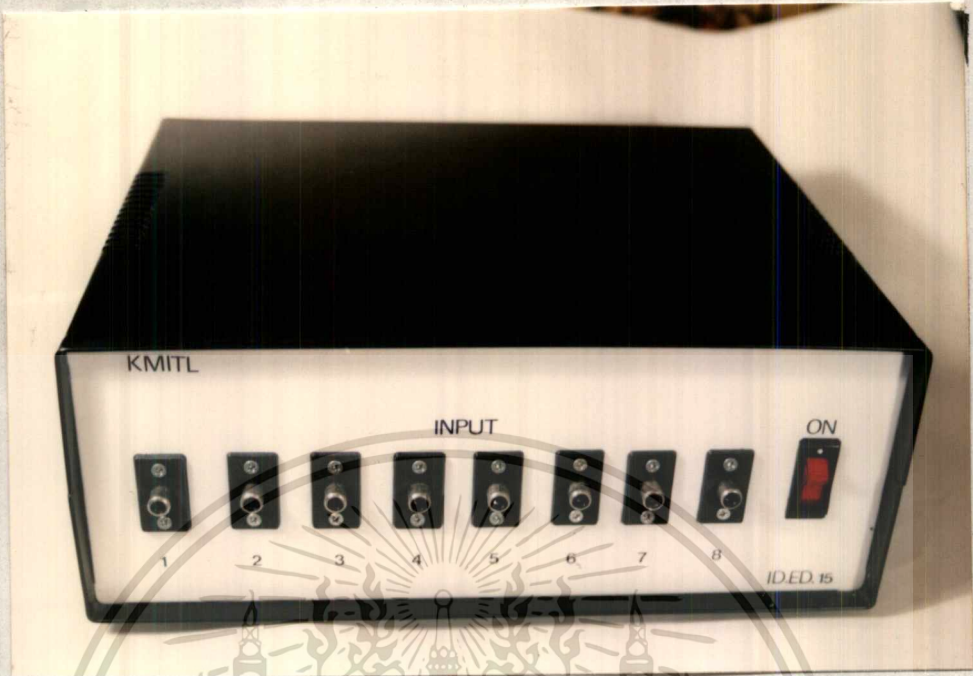
รูปที่ 3.12 หัว shoe ขนาดต่าง ๆ



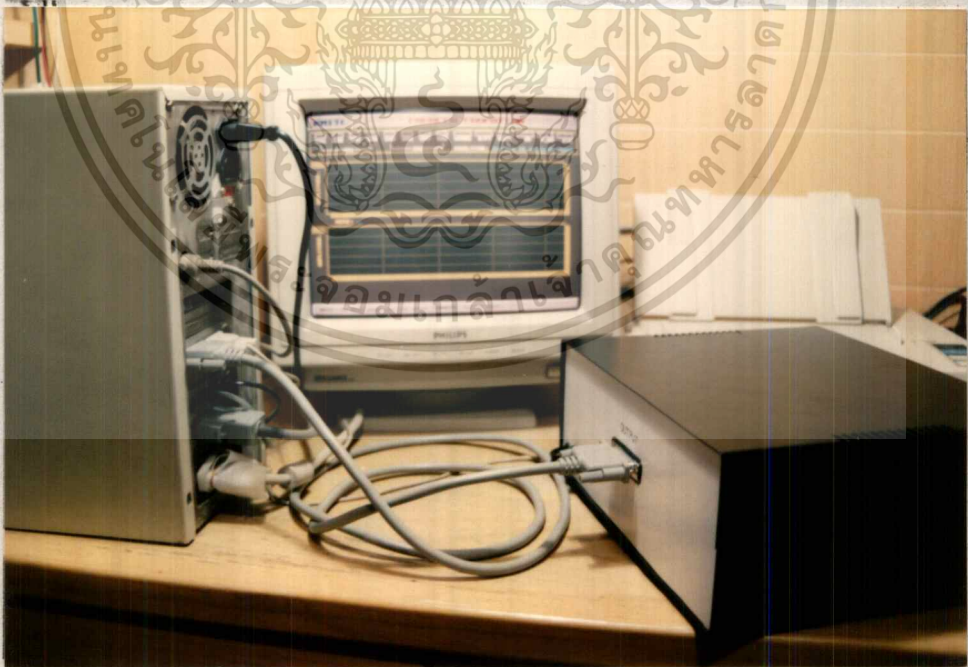
รูปที่ 3.13 ก ลักษณะของส่วนที่ขยายสัญญาณของ Shoe ทั้ง 8 ชุด

เมื่อเราทำการขยายสัญญาณที่ได้มาทั้ง 8 ชุดแล้ว เราก็จะส่งออกไปทางพอร์ตขนาน โดยใช้สายคอมพิวเตอร์ต่อไปยังการ์ดอินเตอร์เฟสที่เสียบอยู่บนสลอตของคอมพิวเตอร์ ดังรูปที่ 3.14 ซึ่งในการ์ดอินเตอร์เฟสที่เราสร้างขึ้นจะมีภาคแปลงสัญญาณอนาลอกที่ได้มาจากภาคขยายเป็นสัญญาณดิจิทัลแล้วนำไปแสดงผล

การทำงานของวงจรทั้งหมดจะเริ่มจาก การที่เรานำสัญญาณอนาลอก ซึ่งมาจากฮาร์ดแวร์เข้าไปที่อินพุทของวงจร ซึ่งจะผ่านมายัง R ปรับค่าเกนของวงจรขยายก่อน จากนั้นก็จะเข้ามายังอินพุททั้ง 8 แชนแนล ของ MC 14051B ซึ่งเป็นไอซีมัลติเพล็กซ์ ซึ่งจะมีสัญญาณดิจิทัล 3 บิตทำการเลือกสัญญาณที่จะออกไปยังเอาต์พุทเพียง 1 เอาต์พุท จากสัญญาณอินพุท 8 แชนแนล และสัญญาณดิจิทัล 3 บิตนี้จะถูกส่งมาจากไอซี 8255 ที่ถูกควบคุมโดย CPU จากนั้นสัญญาณเอาต์พุทจากการมัลติเพล็กซ์จะถูกป้อนมายังอินพุทของไอซี LM 398 ซึ่งเป็น IC Sample and Hold ซึ่งจะ Sampling สัญญาณอินพุทไว้ โดยพัลส์ที่มาจาก การ Sampling จะมีลักษณะดังนี้ คือ ในช่วงที่ขา 8 เป็น "1" จะทำการ Sampling สัญญาณไว้ และในช่วงที่เป็น "0" จะทำการ Hold สัญญาณนี้ไว้ โดยการ Hold จะถูกกำหนดช่วงเวลาโดย C3 แล้วสัญญาณเอาต์พุทจะส่งเข้าไปยัง IC ADC 908 ซึ่งเป็นไอซีแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล สัญญาณนี้จะส่งเข้าวงจร A/D ที่จัดการทำงานเป็นแบบไบโพลาร์ และถูก



รูปที่ 3.13 ข แผงด้านหน้าของส่วนที่ขยายสัญญาณ Shoe



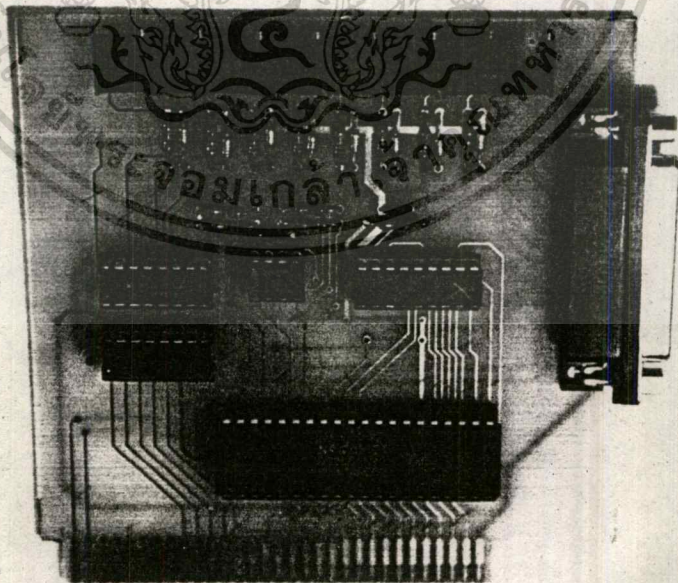
รูปที่ 3.14 ลักษณะการต่อใช้งานกับเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บังคับการทำงานด้วย Sampling Pulse เมื่อจังหวะที่พัลส์ Sample เป็น "1" คือการ Sample ซึ่ง วงจร A/D จะไม่ทำงาน จนกระทั่ง Pulse Sample เป็น 0 คือจังหวะที่ขา RD และ CS ของ A/D แอคทีฟจะเปลี่ยนข้อมูลเป็นดิจิตอลในขณะที่สัญญาณถูก Hold ไว้พอเกิดการเปลี่ยนข้อมูลจะใช้ เวลาประมาณ 16 μsec หลังจาก A/D ได้รับสัญญาณแอกทีฟเข้ามาที่ขา RD และ CS ซึ่ง A/D จะเปลี่ยนสัญญาณที่ BUSY จาก "1" เป็น "0" จนกว่าจะสิ้นสุดขบวนการ คือได้ข้อมูลดิจิตอล ออกมาแล้วขา BUSY จะกลับจาก "0" เป็น "1" ตามเดิมฉะนั้นเราจึงต่อขา RD CS และ BUSY ไว้ ที่ Port C เพื่อให้ซีพียู มาเช็คสภาวะว่าขา BUSY แอคทีฟแล้วหรือยังและขา RD กับ CS เป็น ไปตามเงื่อนไขหรือไม่ เพื่อให้ซีพียูจะได้เฉพาะข้อมูลที่ถูกต้องออกไปทุกครั้ง

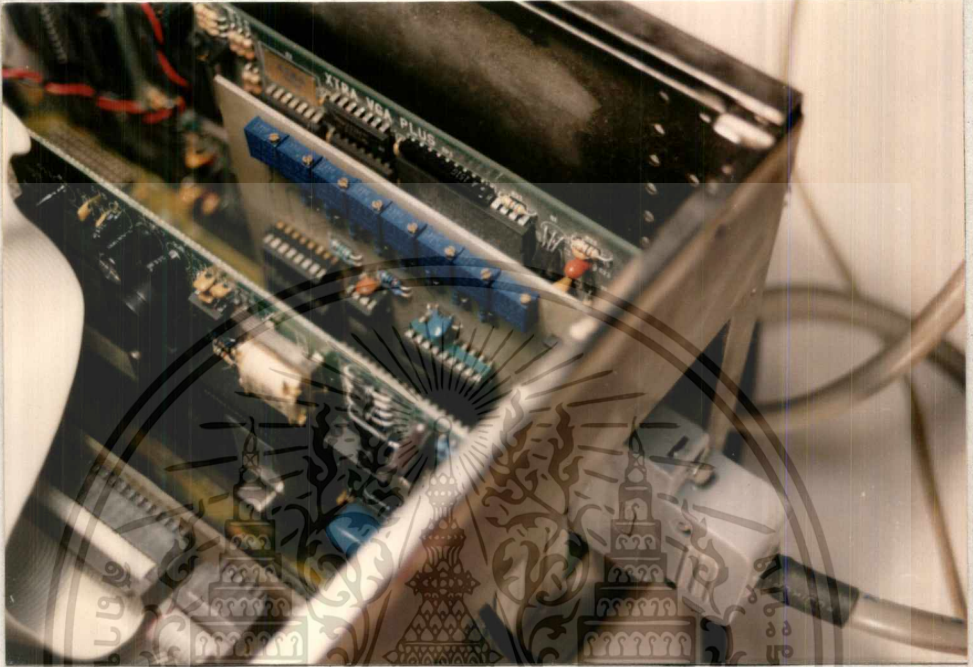
เมื่อซีพียูมาเช็คที่ Port C แล้วตรวจสอบว่าได้จังหวะที่ถูกต้องก็จะทำการอ่านข้อมูล ดิจิตอลจากเอาต์พุตของ A/D ผ่าน Port B เข้าไป เมื่อ A/D ส่งข้อมูลไปให้แล้วมันจะเคลียร์ตัว เองและรอรับสัญญาณ Sample ไปด้วย

จะเห็นว่าการทำงานไม่มีอะไรซับซ้อนแต่ที่ต้อคำนึงคือ ช่วงเวลาที่จะมาเก็บข้อมูลเข้า ไปจะต้องเป็นช่วงเวลาที่ข้อมูลดิจิตอลที่ถูกต้องอยู่เท่านั้น การทำงานจะเป็นแบบ Bipolar คือ บิต DB7 (MSB) ของข้อมูลดิจิตอลจะเป็น Sign Bit หรือบิตที่ใช้บอกค่าของสัญญาณ ถ้าบิตนี้เป็น "1" สัญญาณจะเป็นบวก ถ้าบิตนี้เป็น "0" สัญญาณจะเป็นลบ ส่วนข้อมูลจริงจะอยู่ที่บิต X0-X6



รูปที่ 3.15 ลักษณะของการ์ดอินเตอร์เฟส

ซึ่งเวลาเขียนโปรแกรมแสดงผลจะต้อง เอาเฉพาะ X0-X6 เท่านั้นที่ไปแสดงผลโดยตรวจสอบ
ขั้วจากบิต X7 ทุกครั้ง



รูปที่ 3.16 การติดตั้งใช้งาน

3.2 software

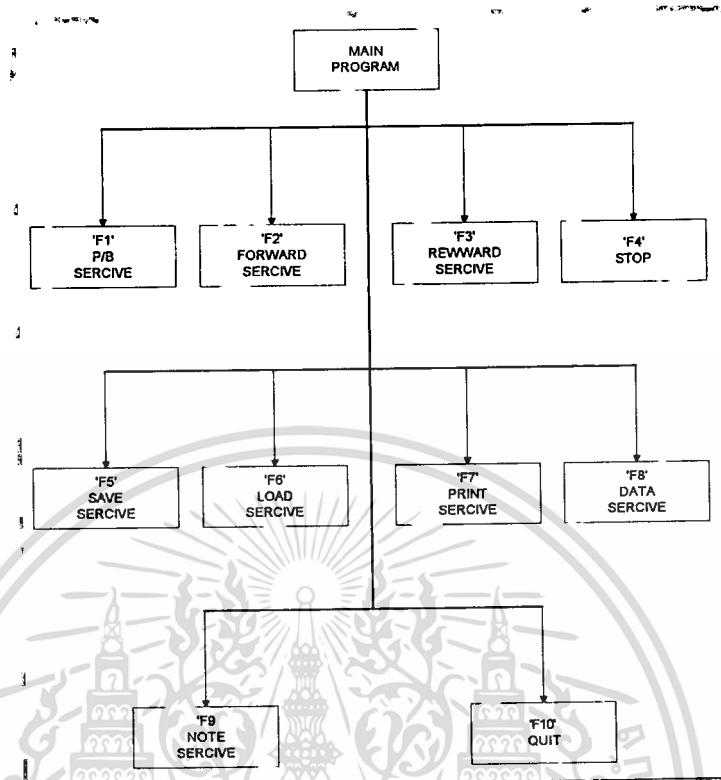
ในส่วนของ SOFTWARE จะแบ่งการทำงานออกเป็นสองส่วนคือ ส่วนที่เป็นโปรแกรมหลักและส่วนที่เป็นโปรแกรมย่อย

3.2.1 โปรแกรมหลัก

ในส่วนของโปรแกรมหลักจะประกอบด้วยส่วนต่าง ๆ ดังรูปที่ 3.17 ซึ่งจะมีการทำงานดังต่อไปนี้

3.2.1.1 SET CONTROL WORD

จะทำหน้าที่ เซ็ทพอร์ต C ของ 8255 ให้ PC0 และ PC4 เป็นขาส่งสัญญาณ RD, WR และ BUSY และเซ็ทพอร์ต A เริ่มรับข้อมูลที่แชนแนล 0



รูปที่ 3.17 ส่วนประกอบของโปรแกรมหลัก

3.2.1.2 DISPLAY MENU & TITLE

จะแสดงเมนู ชื่อของโครงการและผู้ร่วมงาน รวมทั้งบอกขนาดของหน่วยความจำที่เครื่องใช้อยู่ และแสดงหน้าจอของโปรแกรมซึ่งพร้อมที่จะใช้งาน

3.2.1.3 INPUT BIOS KEYS = SCANCODE

จะทำหน้าที่ในการตรวจค่าคีย์บอร์ดโดยใช้ตัวแปร SCANCODE เพื่อจะใช้เลือกเมนู

3.2.1.4 SCANCODE = F1

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F1 เพื่อบอกสถานะว่าเริ่มอ่านจากปลายท่อหรือ อีกด้านของปลายท่อ

3.2.1.5 SCANCODE = F2

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F2 เพื่อบอกสถานะว่าเลือกเข้าเมนูการพล็อตกราฟโดยให้ค่า B = 0 เพื่อบอกการพล็อตแบบ FORWARD READ

3.2.1.6 SCANCODE = F3

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F3 เพื่อบอกสถานะว่าเลือกเข้าเมนูการพล็อตกราฟโดยให้ค่า B = 1 เพื่อบอกการพล็อตแบบ REWARD READ

3.2.1.7 SCANCODE = F4

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F4 เพื่อบอกสถานะว่าหยุดการทำงาน

3.2.1.8 SCANCODE = F5

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F5 เพื่อบอกสถานะว่าเลือกเข้าสู่ขั้นตอนการเก็บข้อมูล

3.2.1.9 SCANCODE = F6

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F6 เพื่อบอกสถานะว่าเลือกเข้าสู่ขั้นตอนการอ่านข้อมูล

3.2.1.10 SCANCODE = F7

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F7 เพื่อบอกสถานะว่าเลือกเข้าสู่ขั้นตอนการพิมพ์ข้อมูล

3.2.1.11 SCANCODE = F8

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F8 เพื่อบอกสถานะว่าเลือกเข้าสู่ขั้นตอนการเขียนรายละเอียดของท่อที่นำมาทดสอบ

3.2.1.12 SCANCODE = F9

จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F9 เพื่อบอกสถานะว่าเลือกเข้าสู่ขั้นตอนการเขียนบันทึกข้อความพิเศษ(NOTE)

3.2.1.13 SCANCODE = F10

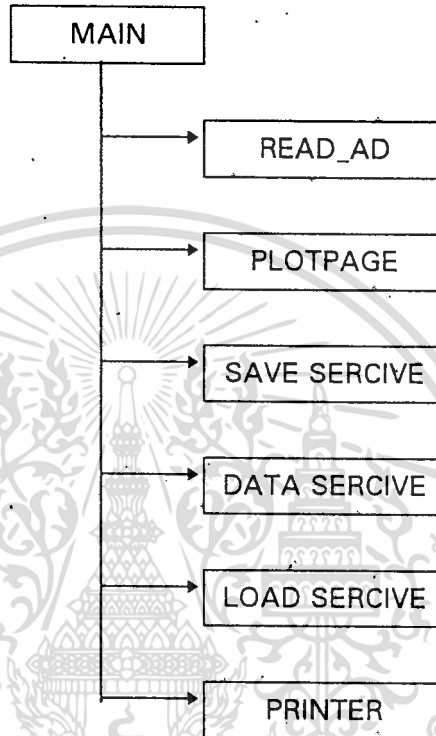
จะใช้ตรวจสอบคีย์บอร์ดว่าเป็น F10 เพื่อบอกสถานะว่าออกจากโปรแกรมใช้งานกลับสู่

DOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โปรแกรมย่อย

ในส่วนของโปรแกรมย่อยจะประกอบด้วยส่วนต่าง ๆ ดังรูปที่ 3.18 ซึ่งจะมีการทำงานดังต่อไปนี้



รูปที่ 3.18 ส่วนประกอบของโปรแกรมย่อย

3.2.2.1 ฟังก์ชัน READ_AD

เป็นฟังก์ชันเลือกแบบของการพล็อตกราฟทางจอภาพ โดยตรวจเช็คค่า B โดยถ้า $B = 0$ จะเป็นแบบ FORWARD READ แต่ถ้า $B = 1$ จะเป็นแบบ REWARD READ จากนั้นจะ CLEAR ค่าของ SH1-SH8 แล้วเข้าสู่ฟังก์ชัน PLOTPAGE เมื่อเสร็จแล้วจะรอรับคีย์ถ้ากดคีย์ ESC และ F4 จะเข้าสู่ MENU2 แล้วทำการตรวจเช็คค่าคีย์ โดยถ้ากด F2 และ F3 จะเข้าสู่ฟังก์ชัน READ_AD แบบ FORWARD และ REWARD แต่ถ้ากด F5 จะเข้าสู่ฟังก์ชัน SAVE หรือ กด F9 จะเข้าสู่ฟังก์ชัน NOTE แต่ถ้ากดคีย์ ESC จะกลับสู่โปรแกรมหลัก

3.2.2.2 ฟังก์ชัน PLOTPAGE

จะแสดงเวลาปัจจุบันทางหน้าจอ และแสดงค่า SH1-SH8 แล้วจะพล็อตกราฟ CH1 และ CH2 แล้วเปลี่ยน PAGE ของหน้าจอโดยใช้หลักการสลับ PAGE แสดง จากนั้นจะรอรับคีย์ F4 จึงจะหยุดการพล็อตแล้วกลับเข้าสู่ MENU2

3.2.2.3 ฟังก์ชัน SAVE SERCIVE

จะแสดง MENU SAVE ที่หน้าจอ จากนั้นจะให้ตั้งชื่อไฟล์และจะเก็บไว้ในตัวแปร GREAD[7] แล้วทำการเปิดไฟล์และเก็บข้อมูลลงดิสก์โดยต้องมีที่ว่างไม่น้อยกว่า 2 KB จากนั้นจึงกลับสู่โปรแกรมหลัก

3.2.2.4 ฟังก์ชัน DATA SERCIVE

จะแสดง MENU ACTIVE และ MENU DATA โดยเริ่มกำหนดให้ POINT = 0 และให้ *MSG เป็นตัวชี้ โดยเก็บที่ GRADE[POINT] และเมื่อป้อน DATA เสร็จ ก็จะรอรับคีย์ ENTER แล้วจะเพิ่มค่า POINT ขึ้น 1 และเมื่อ POINT > 7 ก็จะกลับไปรับคีย์โดยเขียน DATA ที่ GRADE [POINT] ต่อไปจน POINT > 7 จึงจะกลับเข้าสู่ โปรแกรมหลัก

3.2.2.5 ฟังก์ชัน LOAD SERCIVE

จะแสดง MENU LOAD โดยรอรับชื่อไฟล์ที่จะ LOAD จากนั้นจึงค้นหาไฟล์เมื่อพบก็จะเปิดไฟล์และแสดงข้อมูล SH1-SH8 และข้อมูลทั้งหมดแล้วพล็อตกราฟ

3.2.2.6 ฟังก์ชัน PRINTER

จะแสดงข้อมูลของ SH1-SH8 และจะพิมพ์กราฟของ CH1 และ CH2 เมื่อพิมพ์เสร็จแล้วก็จะกลับสู่โปรแกรมหลัก

3.2.2.7 ฟังก์ชัน NOTE

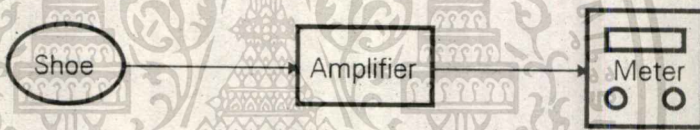
จะแสดง MENU NOTE ออกทางหน้าจอ และรับข้อมูลที่ป้อนเข้ามา และเมื่อได้รับคีย์ ENTER หรือ ESC ก็จะกลับเข้าสู่โปรแกรมหลัก และแสดงข้อมูลที่ป้อนไว้ออกทางหน้าจอ

บทที่ 4

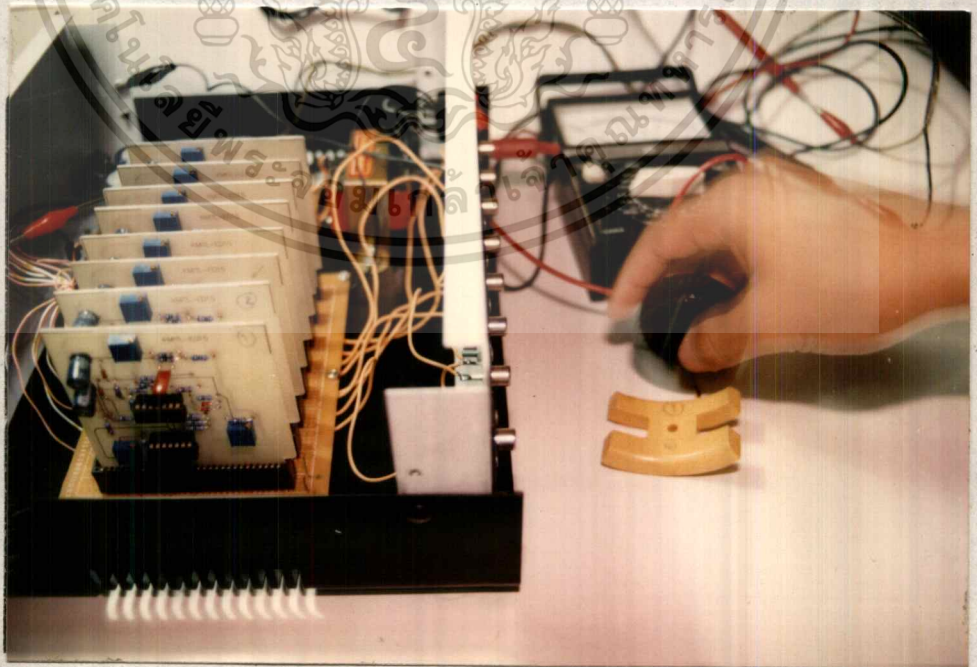
การทดลองและผลการทดลอง

ในการทดลองเราได้ทำการทดลองในส่วนของ HARDWARE และ SOFTWARE โดยในขั้นแรกเราจะทดลองในส่วนของ HARDWARE ก่อนขณะเดียวกันเราก็จะทำการเขียนโปรแกรมควบคู่กันไปด้วย แต่ทางด้าน HARDWARE นั้นจะเสร็จเร็วกว่าทางด้าน SOFTWARE เพราะวงจรที่ใช้ไม่ซับซ้อน ดังนั้นในการทดลองส่วนใหญ่จะเป็นการทดลองทางด้าน SOFTWARE มากกว่า เนื่องจากมีการทำงานที่ซับซ้อนอีกทั้งโปรแกรมก็มีความยาวมาก

4.1 ขั้นตอนการทดลอง HARDWARE



รูปที่ 4.1 บล็อกไดอะแกรมการต่อมัลติมิเตอร์วัดสัญญาณ



รูปที่ 4.2 แสดงการใช้มัลติมิเตอร์วัดสัญญาณที่เอาต์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่โดยไม่เสียค่าใช้จ่าย
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

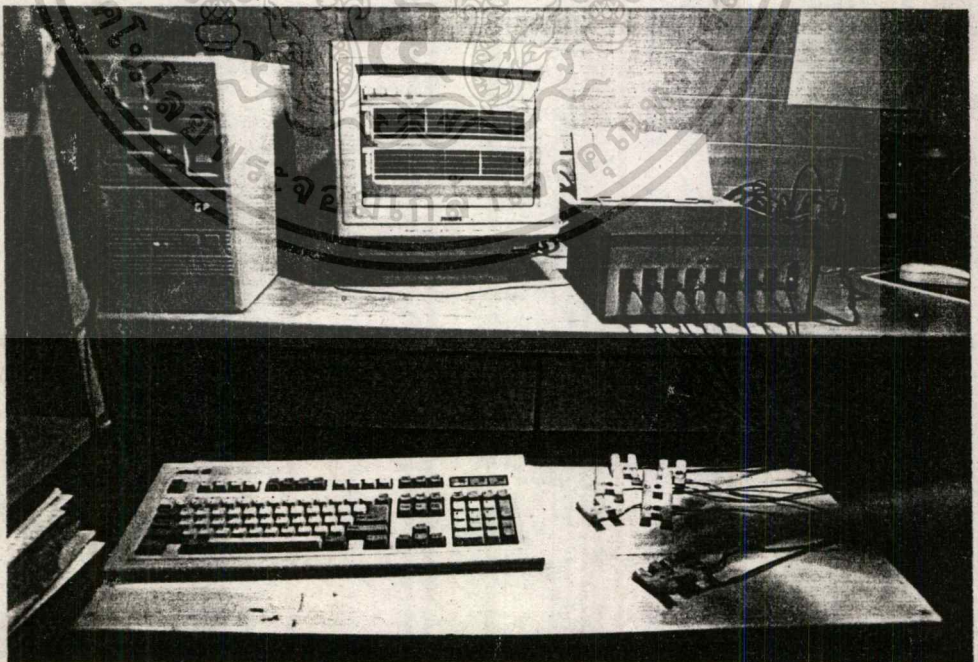
ในการทดลองส่วน HARDWARE นั้นในขั้นแรกเราจะทดลองวัดสัญญาณเอาต์พุตที่ได้จากภาค AMPLIFIER ก่อน โดยการใช้มัลติมิเตอร์วัดแรงดันเอาต์พุตดังรูปที่ 4.1 ซึ่งจะเป็นสัญญาณ PEAK ชั่วขณะ ซึ่งเกิดจากการนำแม่เหล็กมาผ่านหัวตรวจดังรูปที่ 4.2 และจะทำการทดลองทุกแขนง

ซึ่งสัญญาณที่ได้จากการวัดจะส่งเข้าไปยังภาคขยาย จากนั้นก็จะส่งไปยังการดิอินเตอร์เฟสเพื่อแสดงผลต่อไป

4.2 ขั้นตอนการทดลอง SOFTWARE



รูปที่ 4.3 บล็อกไดอะแกรมการทดลองใช้งานกับคอมพิวเตอร์



รูปที่ 4.4 ขณะทำการทดลอง

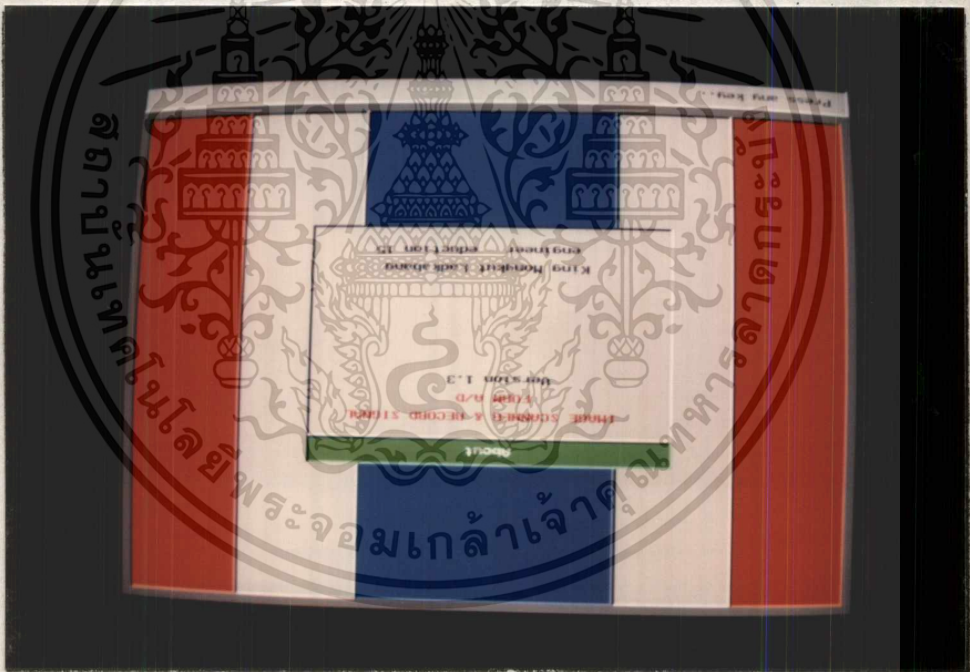
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของโปรแกรมนั้นจะใช้ภาษา C ในการเขียนโปรแกรม เครื่องคอมพิวเตอร์ที่ใช้ทำการทดลองมีลักษณะดังนี้เป็นเครื่องระดับ AT ซึ่งใช้จอมอนิเตอร์สี (Super VGA) และจะใช้พื้นที่เก็บข้อมูลขนาดประมาณ 613 KB

ในการทดลองส่วน SOFTWARE นั้นเราจะนำสัญญาณที่วัดได้จากการทดลอง HARDWARE มาต่อกับการ์ดอินเทอร์เฟซเพื่อจะนำสัญญาณไปแสดงผล ดังรูปที่ 4.3

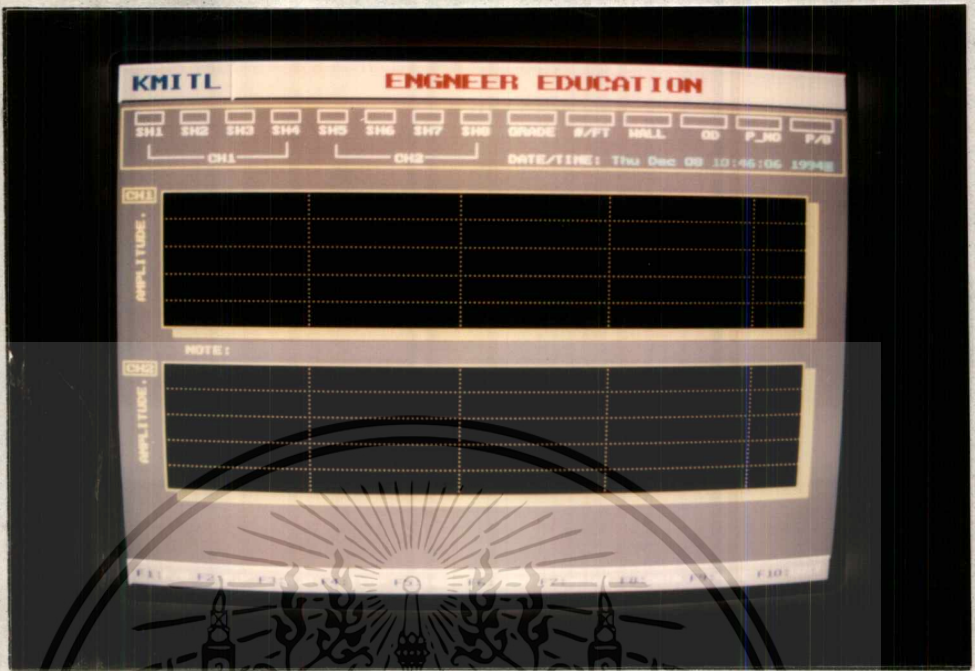
4.2.1 การทดลองใช้งานโปรแกรม

เข้าโปรแกรมโดยการเรียกไฟล์ PLAY.BAT แล้วจอภาพจะปรากฏภาพดังรูปที่ 4.5 จากนั้นให้กดคีย์ใด ๆ 1 ครั้ง ก็จะปรากฏภาพดังรูปที่ 4.6



รูปที่ 4.5 แสดงลักษณะหน้าจอหลังจากเรียกโปรแกรม

จากนั้นให้กดคีย์ใด ๆ อีก 1 ครั้งซึ่งโปรแกรมก็พร้อมที่จะทำงาน ในการใช้งานโปรแกรมเราจะสามารถใช้งานได้ง่าย โดยจะมีปุ่มที่ใช้งานคือปุ่ม F1-F10 ซึ่งจะมีการทำงาน คือ F1 เป็นการเปลี่ยนจาก P-B คือการเลือกการอ่านจากด้านหัวท่อหรือจากปลายท่อ F2 จะเป็นการอ่านข้อมูลจากตัวสแกน มาแสดงผลทางจอภาพ โดยแบ่งการเอกสารนี้เป็นเอกสารที่ส่งวนเวียนสำหรับกรใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปไซ้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

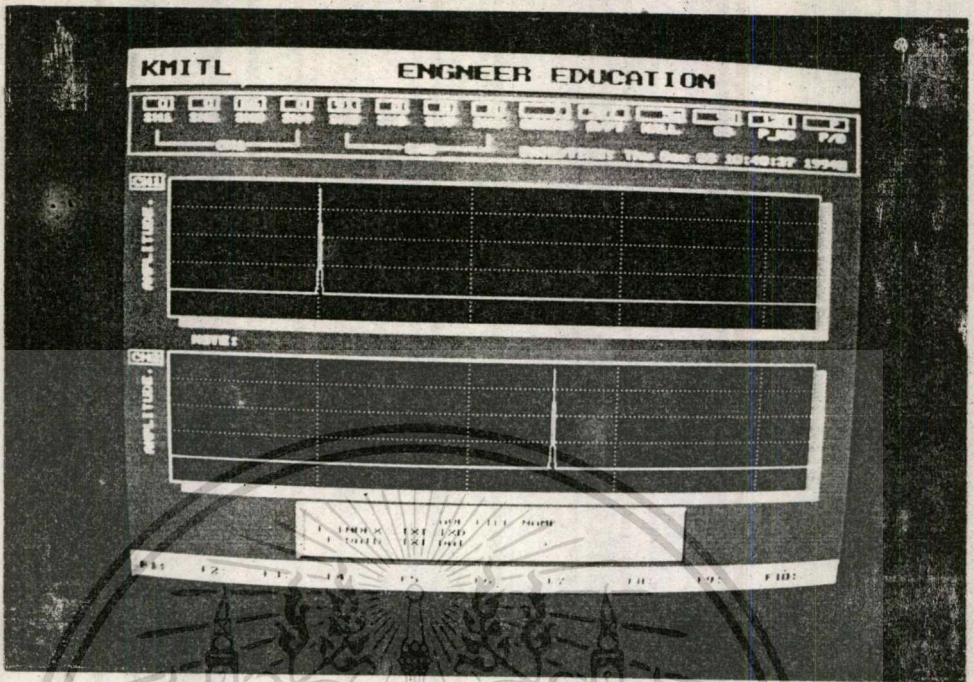


รูปที่ 4.6 ลักษณะจอภาพหลังจากกดคีย์ใด ๆ

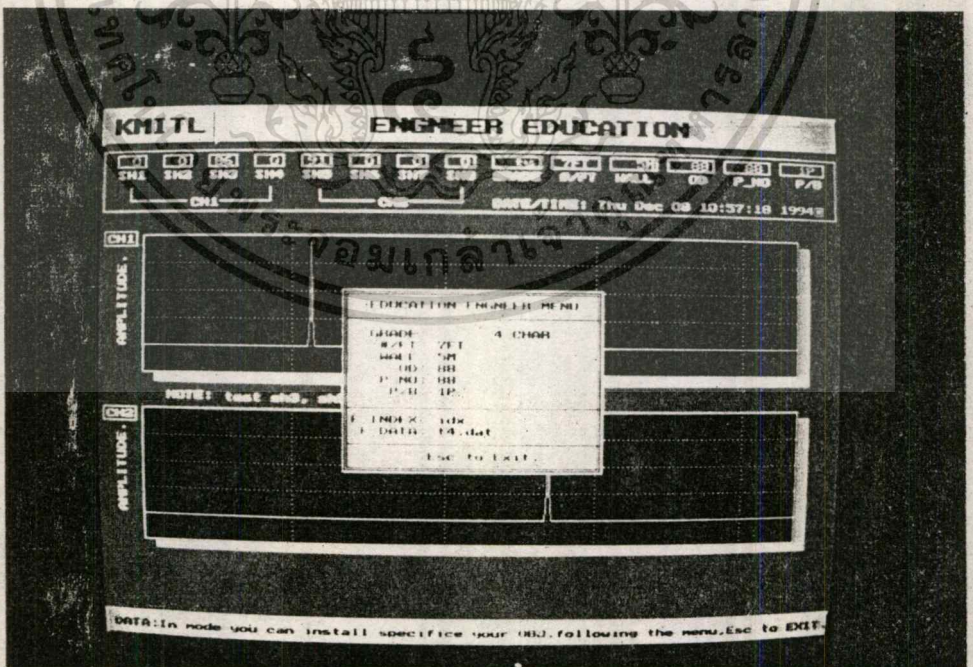
แสดงผล เป็นตัวเลขที่ SH1-SH8 ซึ่ง SH1 แสดงค่าที่อ่านได้จากเซ็นเซอร์ตัวที่ 1 เป็นดังนี้ จนครบตัวที่ 8 และการแสดงกราฟ ซึ่งจะแบ่งแสดงที่ SH1-SH4 ใน CH1 และ SH5-SH8 ใน CH2 F3 การทำงานจะคล้ายกับ F2 แต่จะอ่านจาก SH8 เลื่อนขึ้นมาถึง SH1 F4 ใช้สำหรับหยุดการทำงานของกราฟ (Stop) F5 ใช้สำหรับ SAVE ข้อมูล โดยจะปรากฏภาพดังรูปที่ 4.7

โดยที่ F_INDEX: สำหรับใส่ FILE INDEX และ F_DATA: สำหรับใส่ FILE NAMEจากนั้นก็กด ENTER โปรแกรมจะทำการ Save ข้อมูลลง Disk เมื่อ Saveเรียบร้อยแล้ว ก็จะกลับเข้าสู่เมนูหลักทันที ถ้าหากเกิดข้อผิดพลาดขึ้นก็จะมีข้อความ "WRITE TO DISK ERROR" ให้กดคีย์ใด ๆ 1 ครั้งก็จะกลับเข้าสู่เมนูหลัก F6 จะใช้สำหรับอ่านข้อมูลจากไฟล์ที่บันทึกเอาไว้มาแสดง หรือเพื่อนำไปพิมพ์ออกทางเครื่องพิมพ์ F7 ใช้สำหรับพิมพ์ข้อมูลออกทางเครื่องพิมพ์ดังรูปที่ 4.9 (ต้องเตรียมเครื่องพิมพ์ไว้เรียบร้อยแล้ว) F8 ใช้สำหรับป้อนข้อมูล ตามเมนูดังรูปที่ 4.8 โดยสามารถป้อนได้ข้อมูลละ 4 ตัวอักษร ซึ่งจะใช้ระบุคุณสมบัติทางเทคนิคบางอย่างของท่อที่นำมาตรวจสอบเพื่อเป็นข้อมูลเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่จากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ลักษณะจอภาพหลังจากกดคีย์ F5 หรือ Save



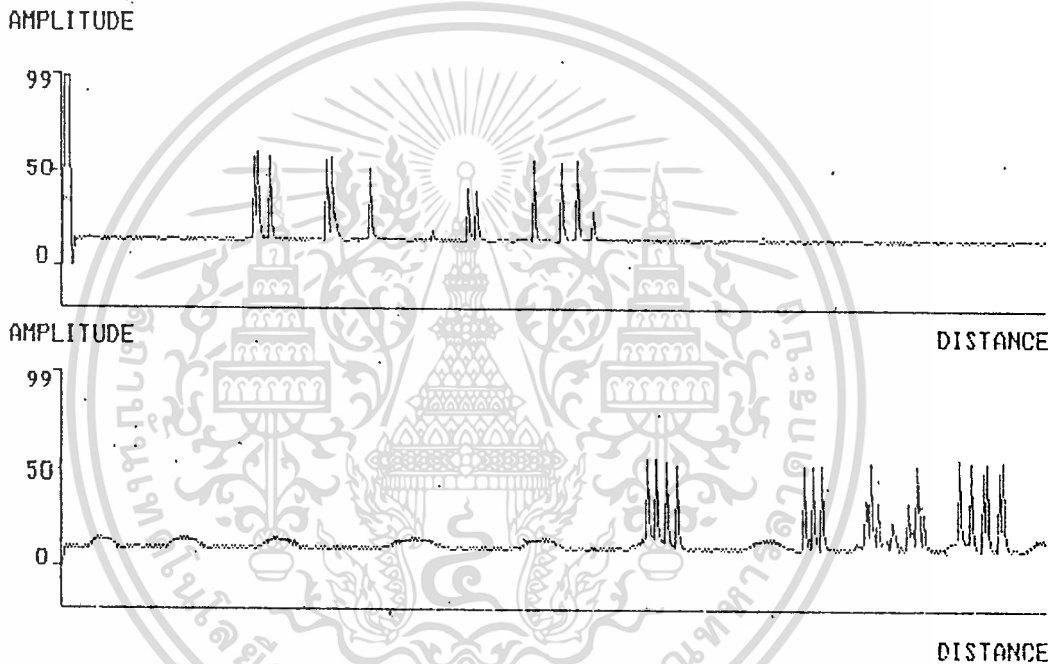
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 4.8 เมนูสำหรับป้อนข้อมูลรายละเอียดข้อต่อหลัก ขอบข่ายงานการคำนวณทางวิศวกรรมไฟฟ้า ไม่สามารถแก้ไขได้ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
=====
```

```
DATA Printer
```

```
=====
```

```
CH1
sh1= 7 ,sh2= 9 ,sh3= 11 ,sh4= 12
CH2
sh5= 8 ,sh6= 7 ,sh7= 99 ,sh8= 99
GRADE= type ,#FT= 7FT ,WALL= 5M ,OD= 99 ,P_NO= 01 ,P/R= 01P
NOTE:test printer by FD kmitl
```



รูปที่ 4.9 กราฟและข้อมูลที่ได้จากการพิมพ์

F9 ใช้สำหรับการเขียนข้อความบางอย่าง (ซึ่งจะปรากฏทางหน้าจอภาพโดยตลอด) เมื่อบันทึกข้อความสิ้นสุดแล้วให้กด Enter ส่วน F10 จะใช้เมื่อต้องการออกจากโปรแกรม เพื่อกลับไปสู่ DOS

บทที่ 5

บทวิจารณ์ สรุป และแนวทางในการพัฒนา

5.1 บทสรุป

ตัวตรวจจับและแสดงผลในโครงงานนี้ สร้างขึ้นเพื่อพัฒนาและประกอบเป็นเครื่องตรวจสอบรอยร้าวท่อเหล็ก ในการสร้างนี้คณะผู้จัดทำได้วางขอบเขตของโครงงานไว้ดังนี้

-สามารถรับสัญญาณจากหัวอ่านมาแสดงผลบนเครื่องไมโครคอมพิวเตอร์ได้ โดยการรับสัญญาณจากไอซีมัลติเพล็กซ์ MC 14051B

-สามารถบันทึกสัญญาณที่แสดงผลที่อยู่บนหน้าจอของเครื่องไมโครคอมพิวเตอร์ลงในดิสก์

-สามารถอ่านข้อมูลที่บันทึกไว้แล้วมาแสดงผลได้ และสามารถพิมพ์ออกทางเครื่องพิมพ์ได้

-เมื่อสัญญาณปรากฏอยู่ที่หน้าจอของเครื่องไมโครคอมพิวเตอร์สามารถ Hold สัญญาณที่อ่านเข้ามาไว้ได้ เพื่อทำการพิมพ์ออกทางเครื่องพิมพ์

5.2 ปัญหาและการแก้ไข

5.2.1 ปัญหา

-ในการออกแบบวงจรภาคขยายในโครงงานนี้เราใช้ไอซี TL084 ซึ่งมีออปแอมป์อยู่ภายใน 4 ตัวซึ่งการใช้ไอซีในลักษณะนี้ทำให้เกิดความร้อนภายในตัวไอซีมากทำให้วงจรขาดเสถียรภาพ

-การเชื่อมต่อ Coil shoe กับสายนำสัญญาณควรมีความระมัดระวังในการต่อ เพราะถ้าต่อไม่ดีอาจทำให้ Coil shoe กับสายนำสัญญาณไม่สามารถต่อกันได้ ทำให้รับสัญญาณเข้ามาไม่ได้

-การเลือกอุปกรณ์ที่จะนำมาใช้บางค่าหายากหรือไม่มีขายในท้องตลาดเช่น R และ C ที่เป็นตัวสร้างสัญญาณนาฬิกาให้กับ ADC 908 ในการ์ดอินเตอร์เฟส

-ในส่วนของ Software บางเมนูยังไม่สมบูรณ์เท่าที่ควร เช่น ในการใช้คีย์ Backspace เป็นต้น

-เกิดการรบกวนจากสัญญาณภายในเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การแก้ไข

-ในการออกแบบวงจรครั้งต่อไป ควรออกแบบวงจรโดยใช้ไอซีที่มีอุปแอมป์ภายในตัวเดียว(ในการใช้งานจริงทางอุตสาหกรรมเราใช้ไอซี OP07)

-ควรเชื่อมต่อ Coil shoe กับสายนำสัญญาณอย่างระมัดระวังหรือออกแบบหัวตรวจให้มีความเหมาะสมและแข็งแรงมากกว่านี้

-ในการออกแบบวงจรนั้นถ้าสูตรใดที่ต้องคำนวณค่า C เราต้องเลือกค่า C ที่เหมาะสมขึ้นมาก่อนและเป็นค่าที่มีขายในท้องตลาดแล้วจึงคำนวณหาค่า R ถ้าค่าที่คำนวณหาไม่ได้ก็ใช้ค่าที่ใกล้เคียง

-ในส่วนของ Software สามารถแก้ไขได้ถ้ามีเวลาพอ

-ในเครื่องคอมพิวเตอร์มีการออสซิลเลทของอุปกรณ์บางตัวซึ่งจะมารบกวนสัญญาณนอกทางอินพุทของเราได้ ดังนั้นที่จุดต่อสัญญาณอินพุทเข้ามาที่วงจรควรใช้สายชิลด์เพื่อป้องกันสัญญาณรบกวน

5.3 แนวทางในการพัฒนา

-พัฒนาให้สามารถเคลื่อนที่หัวอ่านได้โดยอัตโนมัติ เนื่องจากในปัจจุบันยังใช้มือในการเคลื่อนหัวอ่านไปตามท่อ โดยจะใช้ Stepping Motor เป็นตัวขับเคลื่อนหัวอ่านให้หมุนไปตามท่อ ซึ่งจะควบคุมด้วยคอมพิวเตอร์

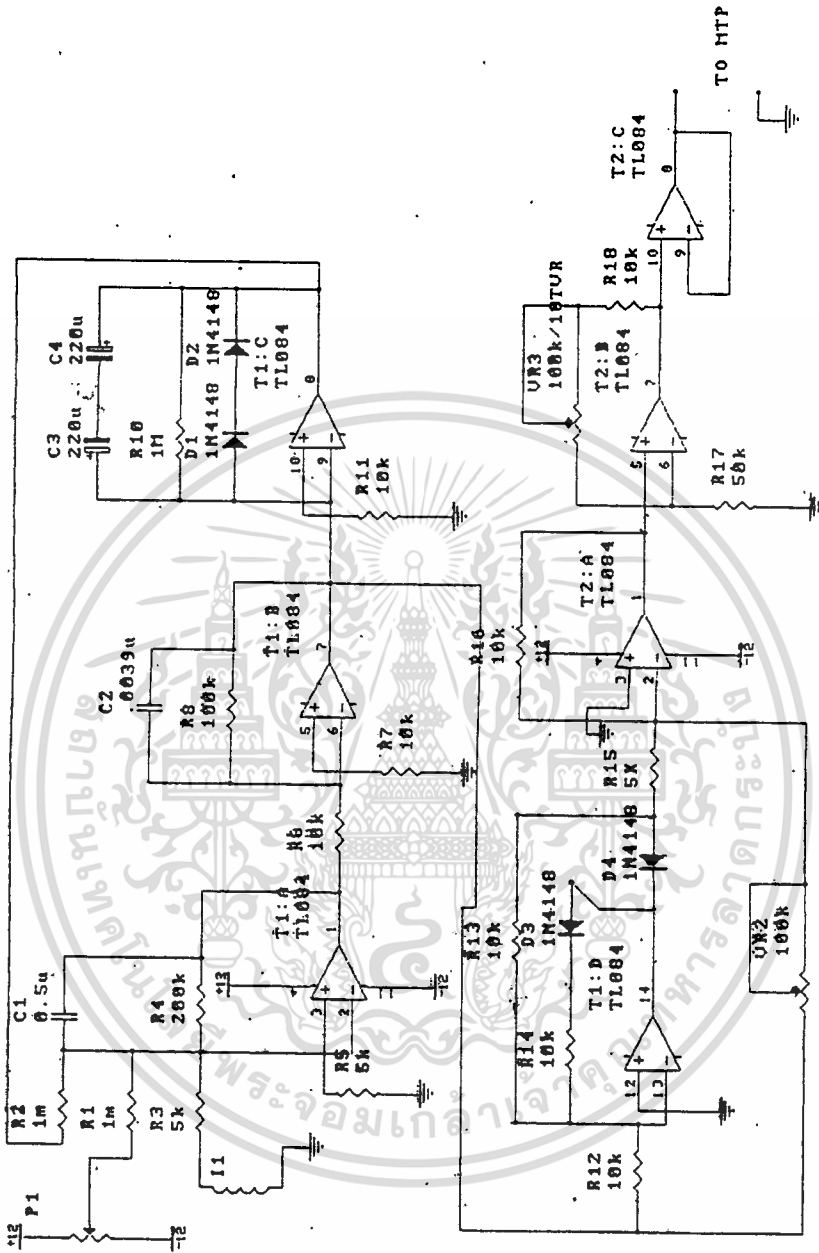
-พัฒนารูปร่างและขนาดเครื่องให้มีขนาดและรูปร่างที่สวยงามและกระทัดรัดมีความคล่องตัวในการใช้งาน เนื่องจากในปัจจุบันยังมีความยุ่งยากในการติดตั้งใช้งาน.

-พัฒนาในส่วนของระบบป้องกันการผิดพลาด เพื่อให้สัญญาณที่ได้รับเข้ามามีเสถียรภาพดีไม่เปลี่ยนแปลงและไม่มีสัญญาณรบกวนเพื่อความแน่นอนในการตรวจสอบ

-พัฒนาทางด้าน SOFTWARE ให้มีฟังก์ชันการใช้งานมากขึ้นและมีประสิทธิภาพมากขึ้นและเหมาะสมกับการพัฒนาด้าน HARDWARE



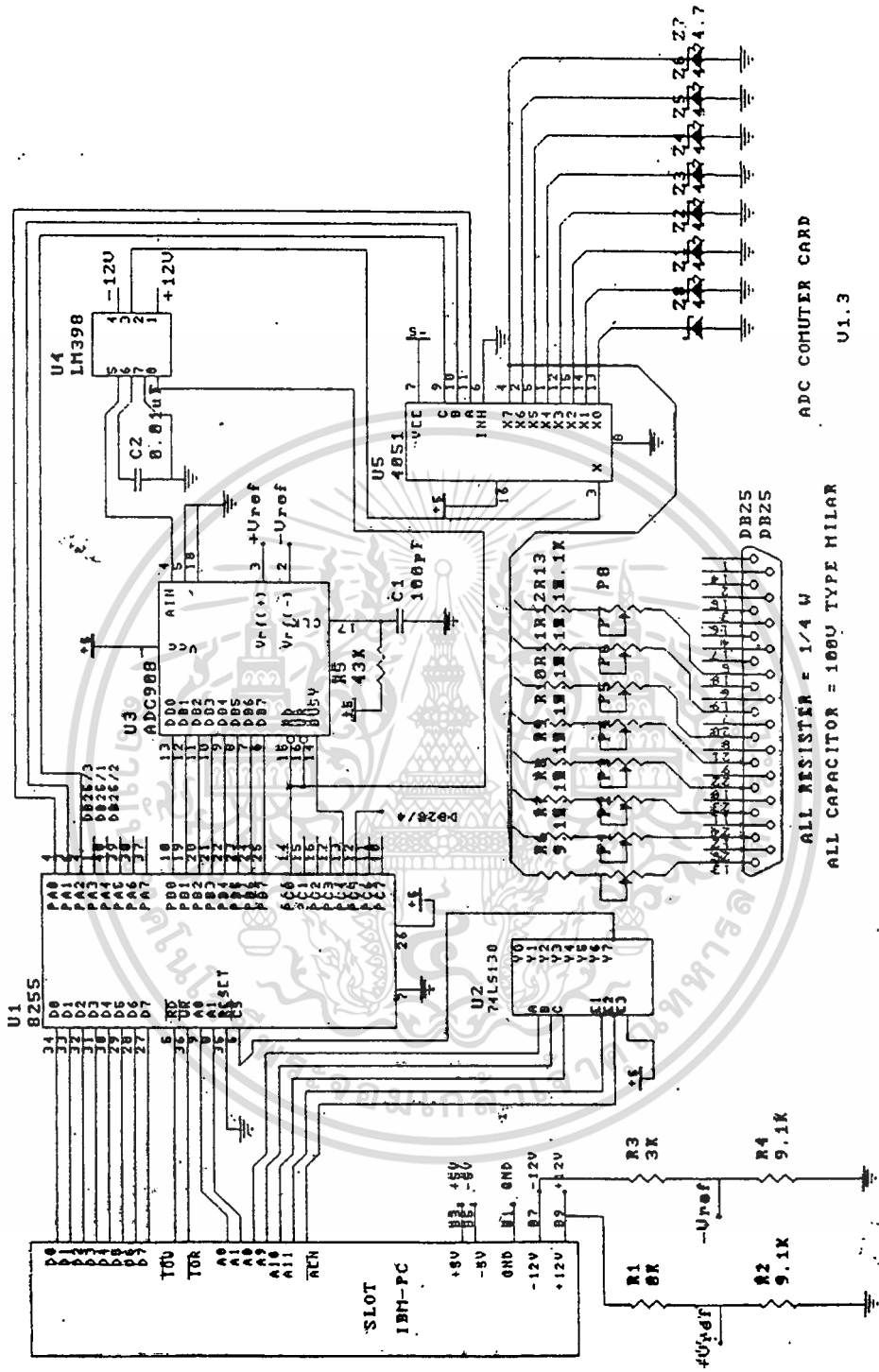
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



SENCER UL.1 (ED 15)

3631405

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



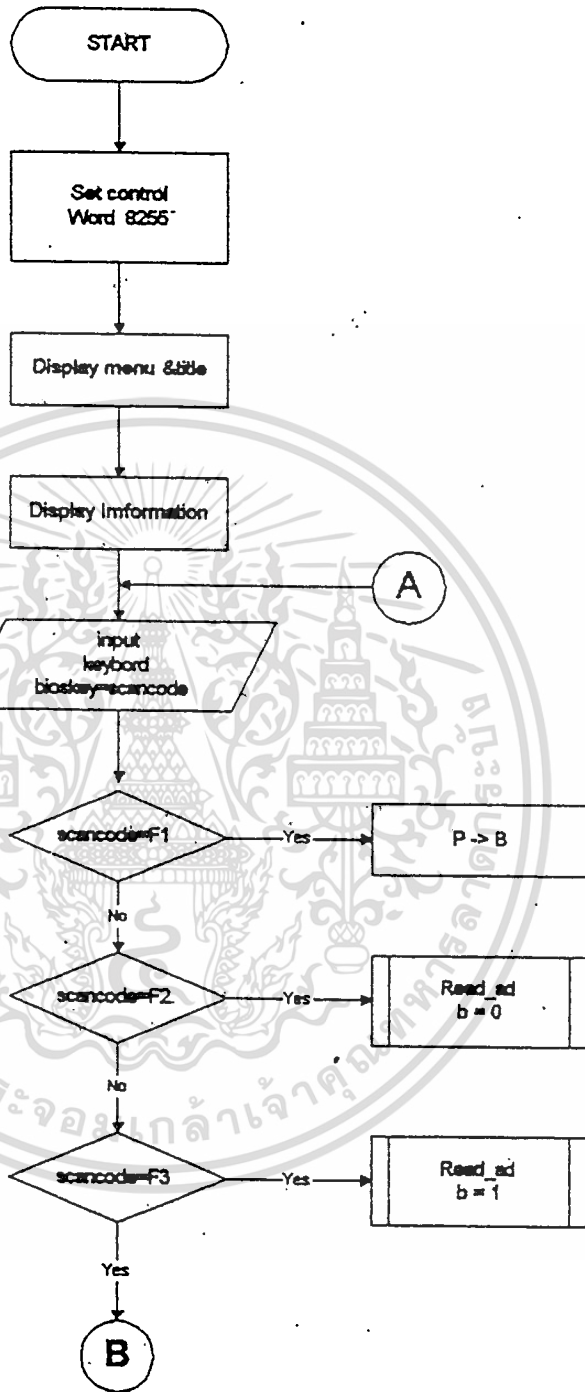
ADC COMPUTER CARD
U1.3

ALL RESISTOR = 1/4 W
ALL CAPACITOR = 100V TYPE MILAR

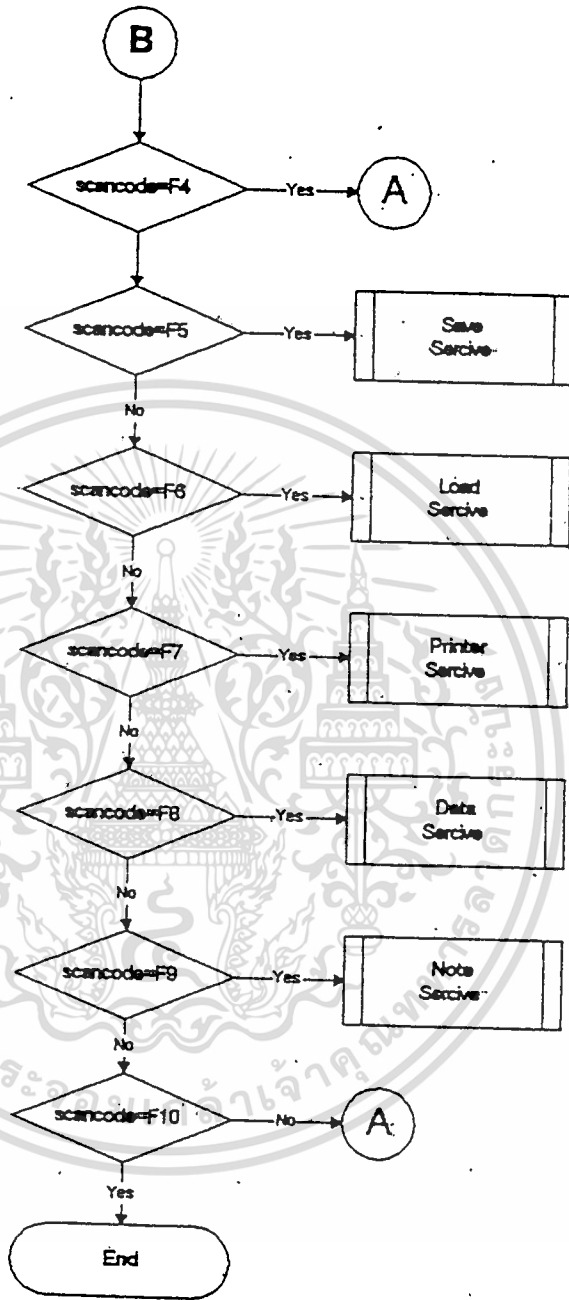
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



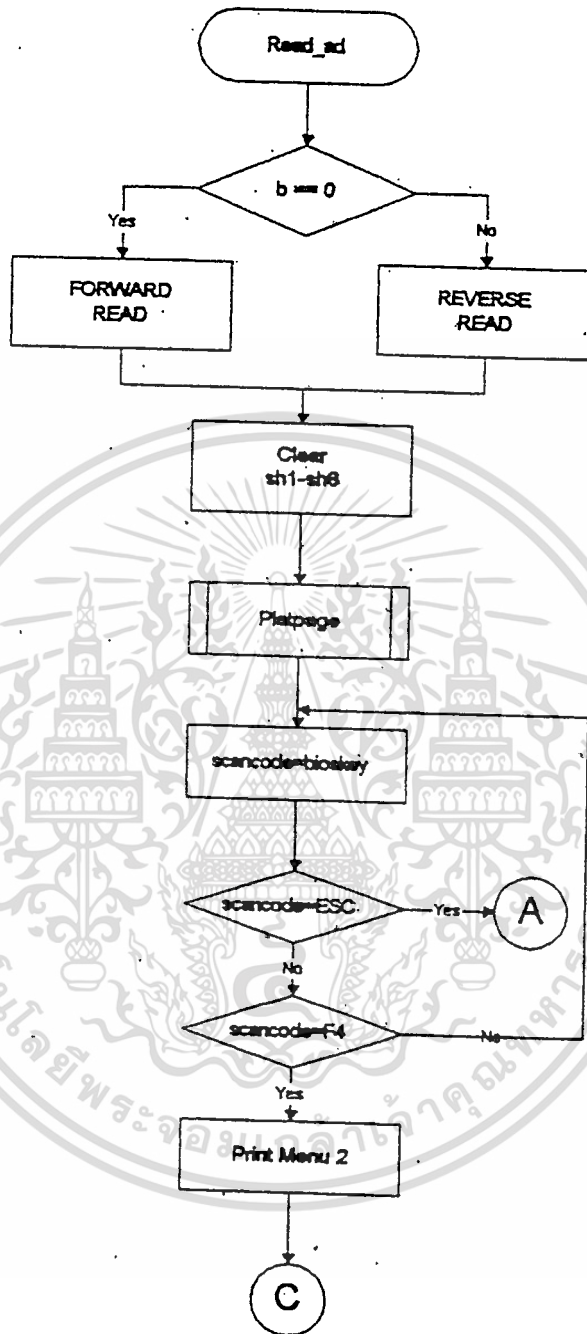
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



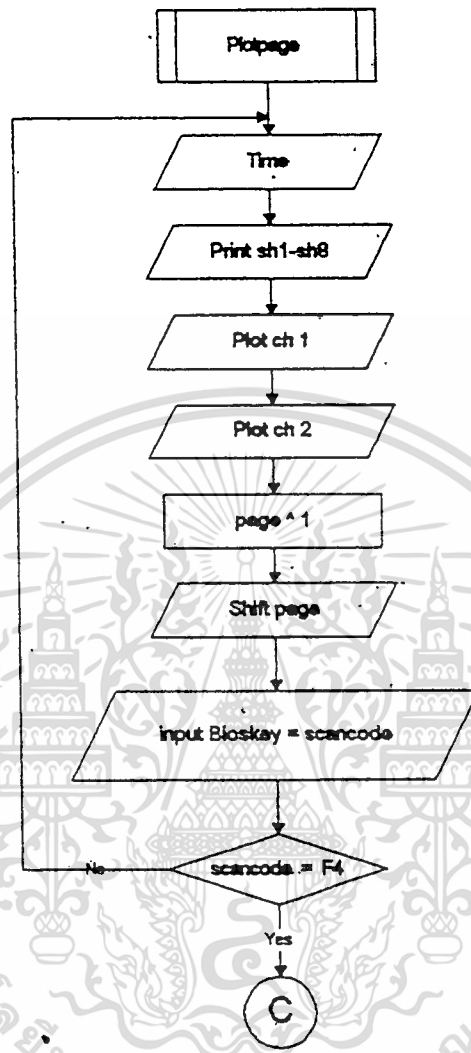
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



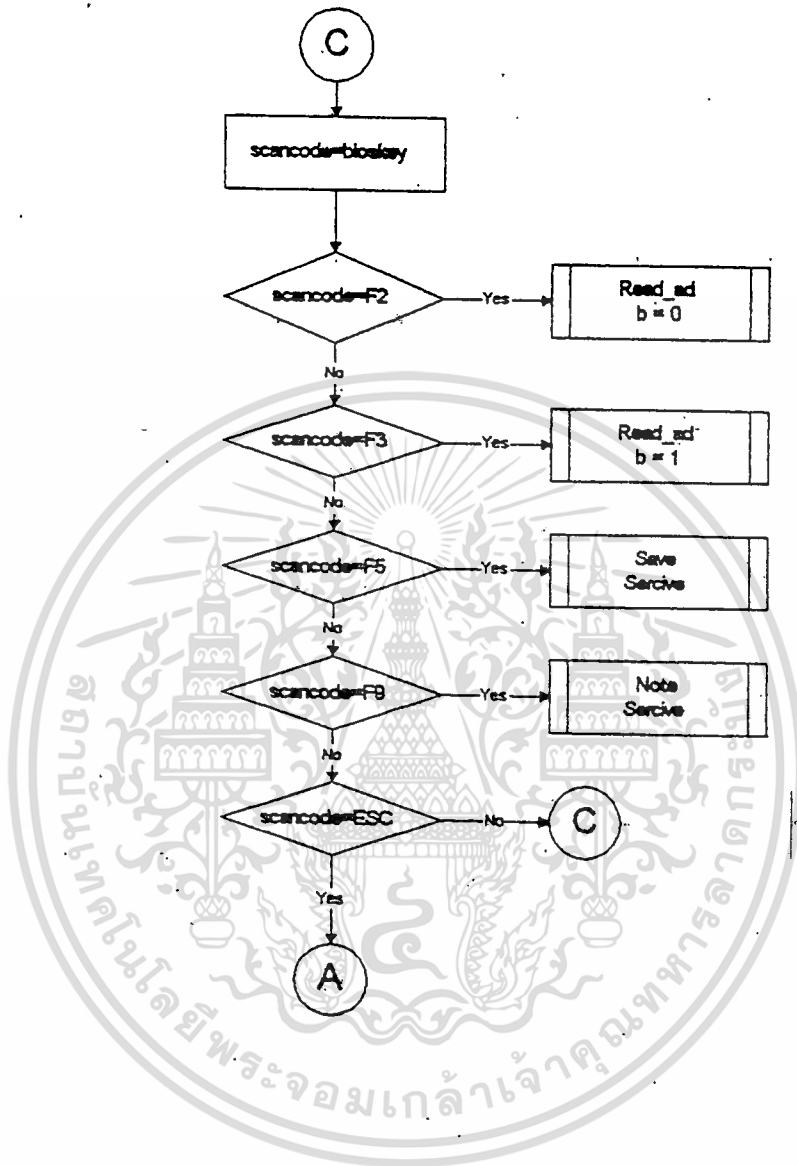
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



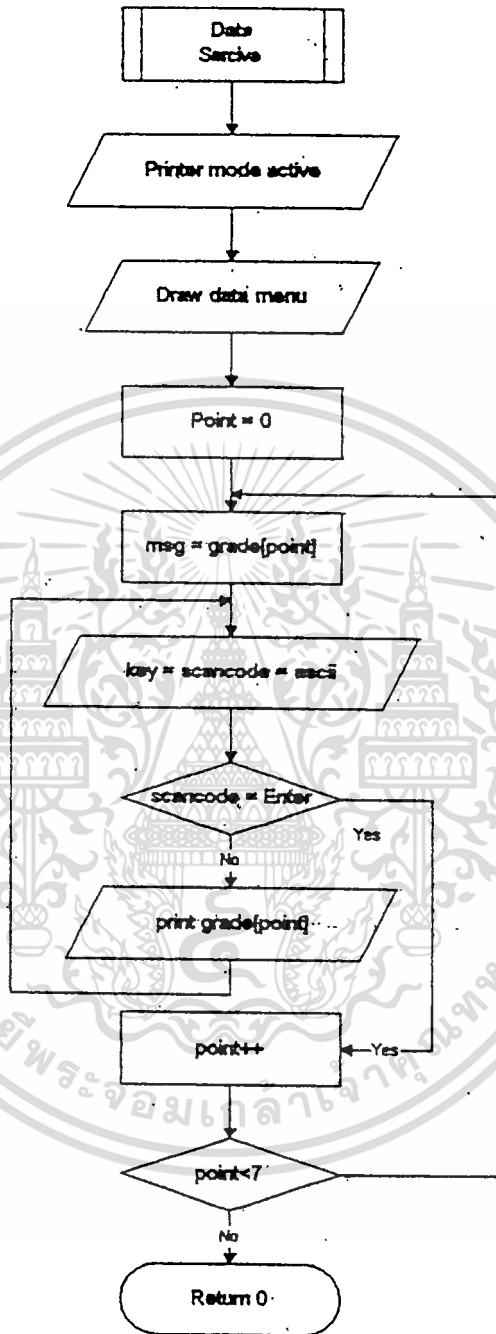
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



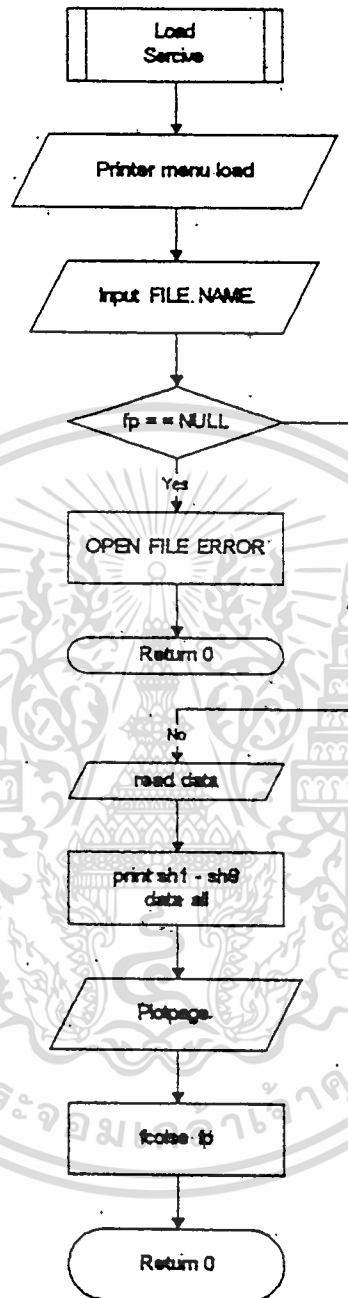
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



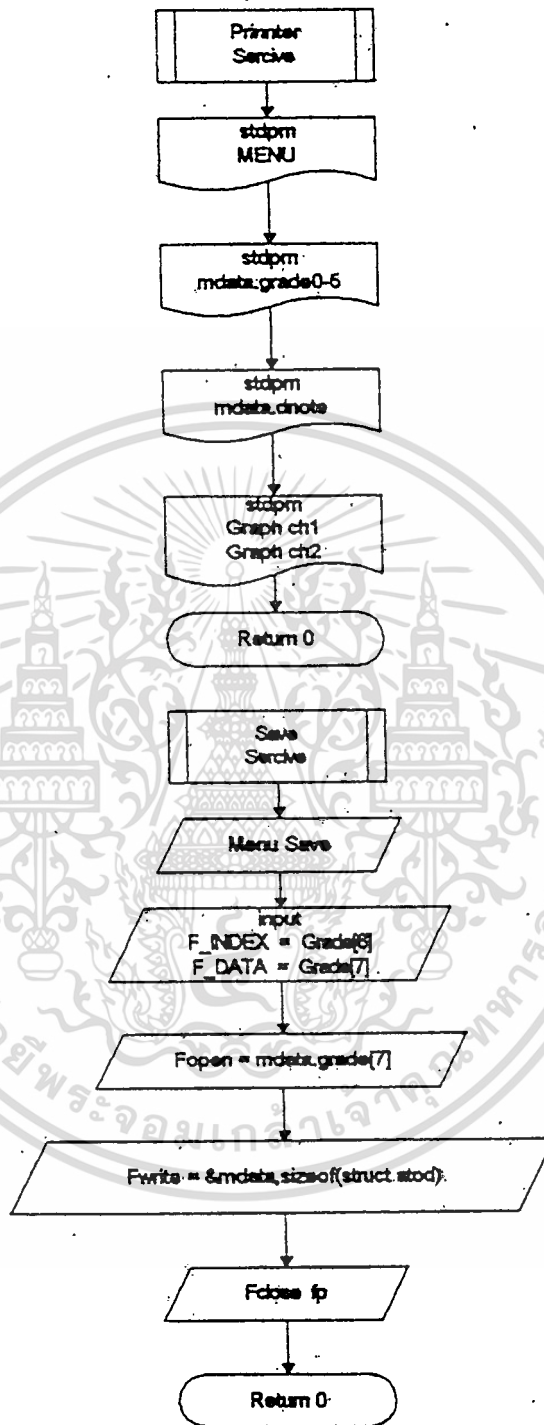
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



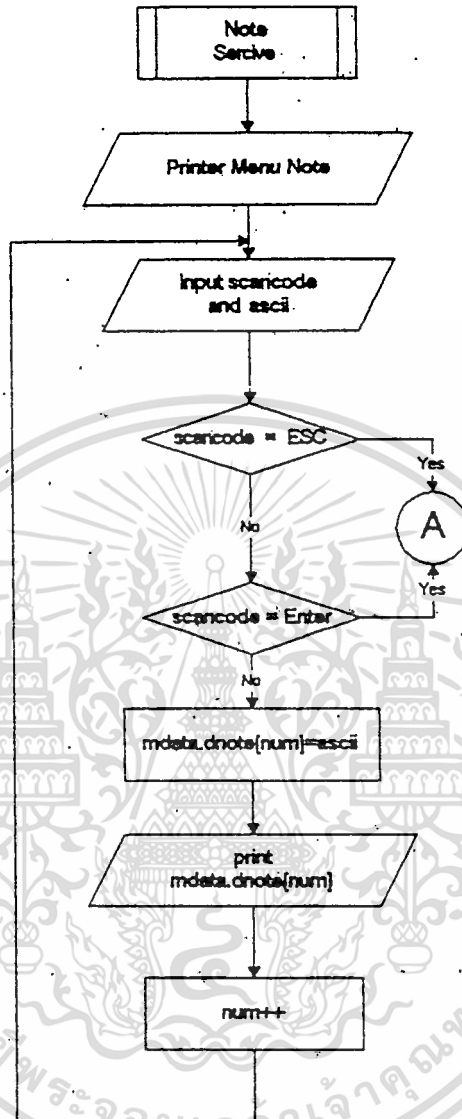
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A/D CONVERTER SELECTION GUIDE

8-Bit Resolution

Product	Type	Conversion Time (μ s)	Linearity Error	Supply Voltage	Reference Voltage	Input Voltage Range	Digital Output Coding
PM0820	CMOS Half-Flash	1.5	$\pm 1/2$ LSB	+5	+5	+5	Binary
ADC908	CMOS SAR	6	$\pm 1/2$ LSB	+5	-10	+10 ± 10	Binary Offset Binary
PM7574	CMOS SAR	15	$\pm 1/2$ LSB	+5	-10	+10 ± 10	Binary Offset Binary

10-Bit Resolution

Product	Type	Conversion Time (μ s)	Linearity Error	Supply Voltage	Reference Voltage	Input Voltage Range	Digital Output Coding
ADC910	Bipolar SAR	6	$\pm 1/2$ LSB	-5	2.5Int	-5, -10 $\pm 5, \pm 10$	Binary Offset Binary

12-Bit Resolution

Product	Type	Conversion Time (μ s)	Linearity Error	Supply Voltage	Reference Voltage	Input Voltage Range	Digital Output Coding
ADC922	Bipolar SAR	3	$\pm 1/2$ LSB	+5, ± 15	10Int	+10, -20 $\pm 5, \pm 10$	Binary Offset Binary
PM7572	CMOS SAR	5, 12	$\pm 1/2$ LSB	+5, -12	-5.2Int	-5	Binary
ADC912	CMOS SAR	12	$\pm 1/2$ LSB	+5, -12	-5	-10	Binary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADC-908

CMOS MICROPROCESSOR-COMPATIBLE
FAST 8-BIT A/D CONVERTER

Precision Monolithics Inc.

FEATURES

- 8-Bit Resolution and Accuracy
- No Missing Codes over Full Temperature Range
- 5µs Conversion Time
- Flexible µP Interface
- 2.5mA Maximum Standby Current
- Replaces AD7574 with Improved Speed
- Available in Die Form

ORDERING INFORMATION

PACKAGE: 18-PIN DIP AND SO				
INL (LSB)	ONL (LSB)	MILITARY TEMPERATURE -55°C TO +125°C	EXTENDED INDUSTRIAL TEMPERATURE -40°C TO +85°C	COMMERCIAL TEMPERATURE 0°C TO +70°C
±1/2	±3/4	ADC908AX	ADC908EX	ADC908GP
±3/4	±7/8	ADC908BX	ADC908FX	—
±3/4	±7/8	—	ADC908FP	—
±3/4	±7/8	—	ADC908FS	—

- For devices processed in total compliance to MIL-STD-883, add 883 after part number. Consult factory for 883 data sheet.
- Burn-in is available on commercial and industrial temperature range parts in CerDIP, plastic DIP and TO-Can packages. For ordering information see PMI's Data Book, Section 2.

GENERAL DESCRIPTION

The ADC-908 is a monolithic CMOS successive-approximation analog-to-digital converter. When used with a 1.35MHz clock, a conversion time of 5µs is achieved, with full accuracy over the operating temperature range.

The ADC-908 outputs use 3-state logic, allowing direct connection to the data bus or system input port. Active-LOW chip select (\overline{CS}) and read/write (\overline{RD}) inputs are used to control all

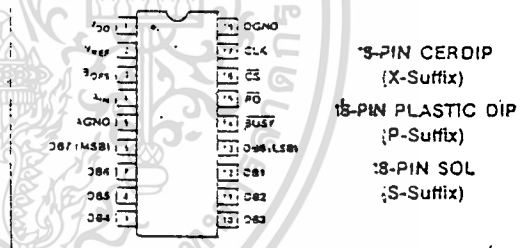
operations. This input structure permits the ADC-908 to be used as a memory-mapped input device. Depending on the control timing waveforms, the ADC-908 is interfaced like static RAM, ROM, or slow memory.

The low power consumption of the ADC-908 is derived from a single +5V supply. A negative reference voltage must also be supplied. Optimum accuracy is achieved when the reference is at -10.00V with a low output resistance. For a low-cost precision -10V/-10.24V reference, ask your PMI sales representative about the REF-08.

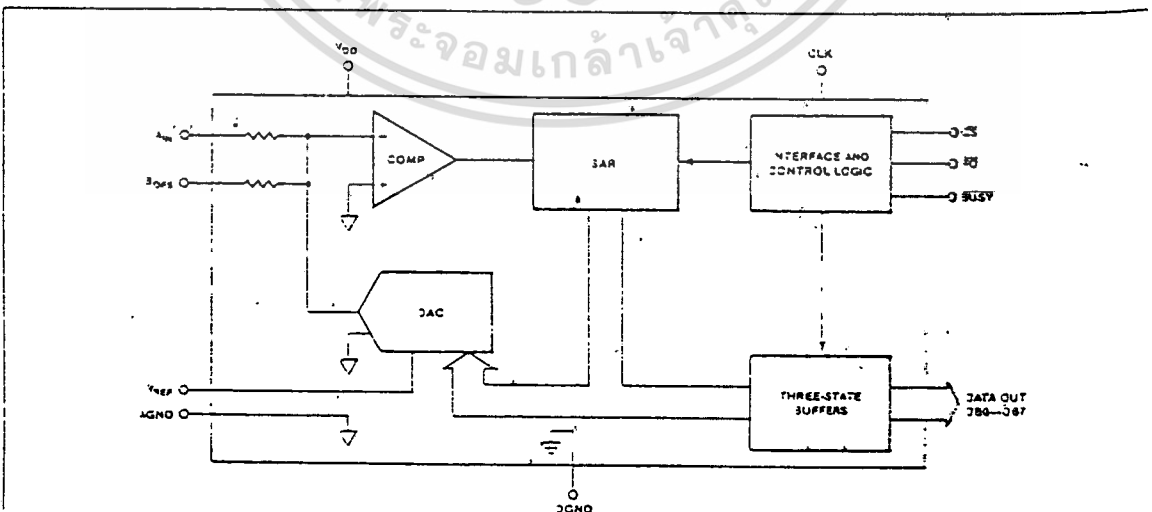
With its on-board comparator, interface logic, optional internal clock, and -5V operation, the ADC-908 is the ideal low-cost solution for microprocessor-based 8-bit A/D systems.

PMI's ADC-908 is pin-and-function compatible with the PM-7574, but offers faster conversion time and faster microprocessor bus interface timing. Conversion time has been reduced by 60% and most key timing specifications, including data access time, START command propagation delay (twpp), and reset time, have been improved.

PIN CONNECTIONS



FUNCTIONAL DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADC-908 CMOS MICROPROCESSOR-COMPATIBLE FAST 8-BIT A/D CONVERTER

ABSOLUTE MAXIMUM RATINGS ($T_A = -25^\circ\text{C}$, unless otherwise noted)

V_{DD} to AGND	0V, +7.0V
V_{DD} to DGND	0V, -7.0V
AGND to DGND	-0.3V, V_{DD}
\overline{CS} , \overline{RD} to DGND	-0.3V, $V_{DD} + .3V$
\overline{DB}_3 , \overline{DB}_2 to DGND	-0.3V, V_{DD}
CLK, \overline{BUSY} to DGND	-0.3V, V_{DD}
\overline{B}_{DFF3} , \overline{A}_{IN}	$\pm 20V$
V_{REF}	0V, -20V

Operating Temperature Range

ADC-908AX, BX	-55°C to $+125^\circ\text{C}$
ADC-908EX, FX, FP, FS	-40°C to $+85^\circ\text{C}$
ADC-908GP	0°C to $+70^\circ\text{C}$

Storage Temperature	-65°C to $+150^\circ\text{C}$
Lead Temperature (Soldering, 10 sec)	$+300^\circ\text{C}$

PACKAGE TYPE	θ_{JA} (Note 2)	θ_{JC}	UNITS
18-Pin Hermetic DIP (X)	79	11	$^\circ\text{C/W}$
18-Pin Plastic DIP (P)	70	30	$^\circ\text{C/W}$
18-Pin SOL (S)	38	25	$^\circ\text{C/W}$

NOTES:

- Digital pins are zener protected. However, proper ESD handling precautions are recommended.
- θ_{JA} is specified for worst case mounting conditions, i.e., θ_{JA} is specified for device in socket for TO, CerDIP, P-DIP, and LCC packages; θ_{JA} is specified for device soldered to printed circuit board for SO and PLCC packages.

ELECTRICAL CHARACTERISTICS at $V_{DD} = -5V$, $V_{REF} = -10V$, Unipolar Configuration, $R_{CLK} = 43k\Omega$, $C_{CLK} = 100pF$; $-40^\circ\text{C} \leq T_A \leq -85^\circ\text{C}$ for ADC-908EF, $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ for ADC-908G, $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$ for ADC-908A/B, unless otherwise noted.

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS
			MIN	TYP	MAX	
ACCURACY						
Resolution	N		8	-	-	Bits
Integral Nonlinearity	INL	A/E/G Grades B/F Grades	-1.2 -3/4	-	+1/2 +3/4	LSB
Differential Nonlinearity	DNL	A/E/G Grades B/F Grades	-3/4 -1.5	-	+3/4 +1.8	LSB
Gain Error	$\pm 1/2 \epsilon$	A/E/G Grades B/F Grades $T_A = -25^\circ\text{C}$ $T_A = \text{Full Temp Range}$ $T_A = -25^\circ\text{C}$ $T_A = \text{Full Temp Range}$	-3 -1.5 -3 -3.5	-	+3 +1.5 +3 +6.5	LSB
Offset Error	$V_{25\epsilon}$	A/E/G Grades B/F Grades $T_A = -25^\circ\text{C}$ $T_A = \text{Full Temp Range}$ $T_A = -25^\circ\text{C}$ $T_A = \text{Full Temp Range}$	-30 -30 -20 -30	-	+30 +30 +60 +40	mV
ANALOG INPUTS						
Resistance Mismatch \overline{B}_{DFF3} to A_N	$R_{\Delta B}$		-1	-	+1	%
Input Resistance at V_{REF} (Note 1)	R_{REF}		5	-	15	k Ω
Input Resistance at \overline{B}_{DFF3} , A_N	R_{DFF3} R_{AIN}		10	-	30	k Ω
Reference Voltage Range	V_{REF}	Specified Conversion Accuracy	-	-10	-	V
Reference Voltage Range	V_{REF}	Degraded Conversion Accuracy	-5	-	+15	V
Reference Current Note 3)	I_{REF}	Conversion Complete Prior to Reset	-	-	2.4	mA
Nominal Analog Input Range						
Unipolar Mode	V_{UN}		-	0 to $+V_{REF2}$	-	V
Bipolar Mode	V_{B}		-	$-V_{REF2}$ to $+V_{REF2}$	-	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ELECTRICAL CHARACTERISTICS at $V_{DD} = -5V$, $V_{REF} = -10V$, Unipolar Configuration, $R_{CLK} = 43k\Omega$, $C_{CLK} = 100pF$; $-40^\circ C \leq T_A \leq +85^\circ C$ for ADC-908E/F, $0^\circ C \leq T_A \leq +70^\circ C$ for ADC-908G, $-55^\circ C \leq T_A \leq +125^\circ C$ for ADC-908A/B, unless otherwise noted.
Continued

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS
			MIN	TYP	MAX	
LOGIC INPUTS						
Input HIGH Voltage \overline{RD} , \overline{CS} Inputs	V_{IH}		2.4	—	—	V
Input LOW Voltage \overline{RD} , \overline{CS} Inputs	V_{IL}		—	—	0.8	V
Input Current \overline{RD} , \overline{CS} Inputs	I_{IN}	$T_A = -25^\circ C$ $T_A = \text{Full Temp Range}$	—	—	1 10	μA
Input Capacitance \overline{RD} , \overline{CS} Inputs (Note 6)	C_{IN}		—	—	5	pF
Input HIGH Voltage, Clock Input	V_{IH}		2.4	—	—	V
Input LOW Voltage, Clock Input	V_{IL}		—	—	3.8	V
Input HIGH Current, Clock Input	I_{IH}		—	—	2	μA
Input LOW Current, Clock Input	I_{IL}	$T_A = -25^\circ C$ $T_A = \text{Full Temp Range}$	—	—	1 10	μA
LOGIC OUTPUTS						
Output HIGH Voltage \overline{BUSY} , $\overline{DB0-7}$	V_{OH}	SOURCE = 40 μA	2.0	—	—	V
Output LOW Voltage \overline{BUSY} , $\overline{DB0-7}$	V_{OL}	SINK = 5mA	—	—	1.4	V
Floating Leakage Current, $\overline{DB0-7}$	I_{LKG}	$T_A = -25^\circ C$ $T_A = \text{Full Temp Range}$	—	—	10	μA
Floating State Output Capacitance	C_{OZ}	Note 6)	—	—	7	pF
POWER REQUIREMENTS						
Standby Current	I_{DD}	$V_{DD} = -4.75V \text{ to } -5.25V$	—	—	2.5	μA
DIGITAL INTERFACE TIMING						
\overline{CS} Minimum Pulse Width (Note 5)	t_{CS}	$T_A = -25^\circ C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$	50 50 30	—	—	ns
\overline{RD} to \overline{CS} Setup Time (Note 6)	t_{wSCS}		0	—	—	ns
\overline{CS} to \overline{BUSY} Propagation Delay (Note 5)	t_{CSPO}	\overline{BUSY} Load = 200pF $T_A = -25^\circ C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$ \overline{BUSY} Load = 100pF $T_A = -25^\circ C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$	— — — — — — —	—	20 100 50 50 20 100	ns
\overline{BUSY} to \overline{RD} Setup Time (Notes 2, 3)	t_{SSA}		0	—	—	ns
\overline{BUSY} to \overline{CS} Setup Time (Note 5)	t_{SSCS}		0	—	—	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



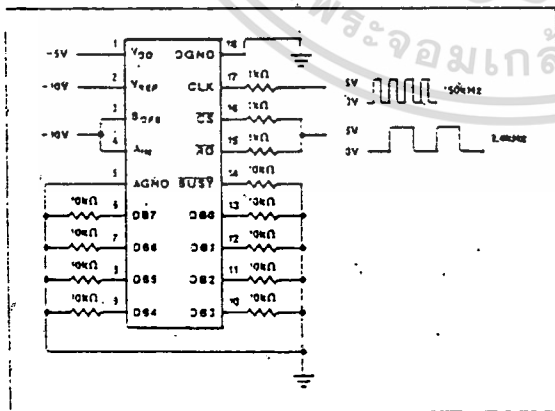
ELECTRICAL CHARACTERISTICS at $V_{DD} = +5V$, $V_{REF} = -1V$, Unipolar Configuration, $R_{CLK} = 43k\Omega$, $C_{CLK} = 100pF$; $-40^\circ C \leq T_A \leq +85^\circ C$ for ADC-908E/F, $0^\circ C \leq T_A \leq +70^\circ C$ for ADC-908G, $-55^\circ C \leq T_A \leq +125^\circ C$ for ADC-908A/B, unless otherwise noted.
Continued

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS	
			MIN	TYP	MAX		
Data Access Time (Note 6)	t_{RAD}	$C_L = 20pF$					
		$T_A = -25^\circ C$	—	—	140		
		$T_A = T_{MIN}$	—	—	100		
		$T_A = T_{MAX}$	—	—	200		
		$C_L = 100pF$					ns
		$T_A = -25^\circ C$	—	—	170		
Data Hold Time (Notes 3, 5)	t_{AH0}	$T_A = -25^\circ C$ (Note 3)	30	—	100		
		$T_A = T_{MIN}$	20	—	70	ns	
		$T_A = T_{MAX}$	40	—	140		
\overline{CS} to \overline{RD} Hold Time (Note 5)	t_{HCS}	$T_A = -25^\circ C$	—	—	200		
		$T_A = T_{MIN}$	—	—	20	ns	
		$T_A = T_{MAX}$	—	—	250		
Reset Time Requirement (Note 6)	t_{RESET}	$T_A = -25^\circ C$	450	—	—	ns	
		$T_A = \text{Full Temp. Range}$	500	—	—		
Conversion Time (Note 4)	$t_{CONVERT}$	Static RAM Mode					
		External Clock			5	ns	
		$f = 1.35MHz$					
		ROM Mode			7		
\overline{RD} HIGH to \overline{BUSY} Propagation Delay, ROM Mode (Notes 4, 5, 6)	t_{WSP0}	$C_L = 20pF$					
		$T_A = -25^\circ C$	—	—	300	ns	
		$T_A = T_{MIN}$	—	—	400		
		$T_A = T_{MAX}$	—	—	300		

NOTES:

- For optimum gain accuracy over the full temperature range, the source resistance at pin 2 should be kept low.
- In ROM mode, \overline{RD} can go LOW prior to $\overline{BUSY} = \text{HIGH}$, but must not return HIGH until $\overline{BUSY} = \text{HIGH}$.
- Output loading: 10pF, A 3k Ω pullup resistor to +5V is used for V_{OH} in high-Z, or V_{OH} to high-Z, a 3k Ω pulldown to GND is used. Measured to 0.5V output change.
- When using the ADC-908 internal oscillator, actual conversion time depends on clock resistor and capacitor as well as temperature.
- ROM interface mode conversion times are typically 1.5x longer than conversion times for other modes, but the ROM interface mode includes an automatic reset in the conversion time.
- Guaranteed but not tested.

BURN-IN CIRCUIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERAL CIRCUIT INFORMATION

The ADC-908 is an 8-bit analog-to-digital converter which uses a successive approximation technique to convert an unknown analog input into a digital code output. The control logic inputs allow easy interface to most microprocessors while three-state outputs allow direct connection to the data bus. Most applications require only passive RC clock components, a $-10V$ reference, and a $+5V$ power supply. The RC-timed internal clock may be used, or an external clock. may be applied to the ADC to maximize performance.

When a Start Conversion command is applied to the \overline{CS} or \overline{RD} inputs (see Operating Descriptions for details), \overline{BUSY} goes LOW indicating a conversion in progress. \overline{BUSY} may be used as an interrupt to halt the controlling microprocessor during conversion or may be polled to prevent premature data reads.

Starting with the most significant bit (MSB), each successive bit in the DAC is turned on (see Figure 1). The comparator then decides if the DAC output is less than or greater than the signal being converted, and that bit is latched on or off, respectively, before proceeding to the next lower bit and repeating the cycle. When all eight bits have been tested, \overline{BUSY} goes HIGH, signaling a completed conversion.

Under control of the \overline{RD} input, the three-state data outputs (D_0 - D_7) change from high-impedance to presenting the new conversion results to the data bus. Following the data read, \overline{RD} returns HIGH resetting the SAR to 1000 0000 and preparing the ADC for its next conversion.

PIN FUNCTIONS

NOTE: For greater detail on digital input functions, consult Truth Tables and Timing Diagrams.

- Pin 1. V_{DD} Power Supply input. $-5V$.
 Pin 2. V_{REF} Voltage Reference input, nominal $-10V$.
 Pin 3. B_{OFS} Bipolar Offset input. $+10V$ input for bipolar mode operation, tie to V_{IN} for unidolar mode operation.
 Pin 4. A_{IN} Analog Input, $0V$ to $+10V$ in unipolar mode, $-10V$ to $+10V$ in bipolar mode.

- Pin 14. \overline{BUSY} Conversion status output. \overline{BUSY} indicates conversion in progress by going LOW at start of conversion and returning HIGH at end of conversion. May be used to interrupt controlling microprocessor or to gate control inputs.
 Pin 15. \overline{RD} READ input. Used to read data (on falling edge) and to reset converter (on rising edge).
 Pin 16. \overline{CS} Chip Select input. Asserted to allow ADC operation. Starts conversion when converter is in reset condition. Note: Holding \overline{CS} HIGH will not prevent a rising edge on \overline{RD} from resetting the converter.
 Pin 17. CLK External clock input/internal clock RC timing input.

APPLICATIONS INFORMATION

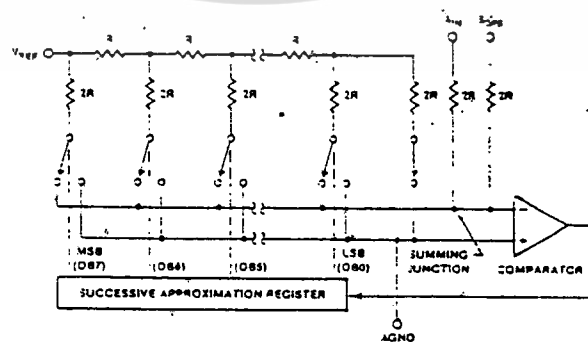
The ADC-908 may be interfaced as if it were a static RAM, a ROM, or a slow-memory device. Each of these interface modes has its own timing and software requirements as described below. These requirements must be rigidly met, as improper timing may cause the ADC-908 to change modes.

HOW TO CHOOSE AN OPERATING MODE

The static-RAM interface mode offers advantages in a tightly controlled hardware and software environment, where the relationship between WRITE and READ instruction pairs is certain. As long as minimum timing is satisfied, converted data may be read at any convenient time after conversion. The use of separate commands to start a conversion, and then read the results, is conceptually easy. However, if the software is subject to uncontrolled modifications, then the paired relationship between WRITE and READ instructions may be lost. Resulting software bugs may result in converted data of unknown age, or altogether invalid data being read.

By contrast, the ROM mode may be more resistant to software bugs. As long as minimum timing is satisfied, each READ instruction obtains new, valid data. However, since the data

FIGURE 1: D/A Converter Used in ADC-908





output at any previous READ instruction is obtained from a conversion performed just after the previous READ instruction, data may be out-of-date. To be sure of obtaining up-to-date data, READ instructions may be coded in pairs (with some NOPs between them); use only the data from the second READ in each pair. The first READ starts the conversion, acting as a substitute for the static-RAM mode WRITE command; the second READ gets the results. The advantage of the ROM mode is the use of a single command, rather than the alternating READ-WRITE required by static-RAM mode.

The slow-memory mode is the simplest mode of all. It is the method of choice where compact coding is essential, or where software bugs are a hazard. In this mode, a single READ instruction will initiate a data conversion, interrupt the microprocessor until completion (WAIT states are introduced), then read the results. If the system throughout tolerates WAIT states, and the hardware is correct, then the slow-memory mode is virtually immune to subsequent software modifications.

OPERATING DESCRIPTION: STATIC-RAM MODE

In this mode, input \overline{CS} is derived from the ADC-908 address decoder, and input \overline{RD} is derived from an active-LOW memory READ signal. (See Figure 2.)

To start a conversion, execute a memory WRITE to the ADC-908. The completed conversion data is obtained by executing a memory READ to the ADC-908. During conversion, output \overline{BUSY} will be LOW. Do not attempt to read data until \overline{BUSY} returns HIGH. The required minimum time between WRITE and READ is usually obtained by including one or more NOP or other program instructions. The use of branch or conditional commands between the WRITE and READ instructions is not recommended due to the possibility of software bugs.

It is important that the WRITE and READ commands be alternately executed. A WRITE instruction has no effect unless the results of the previous WRITE have already been read. Once data has been read, the ADC-908 is internally reset. In other words, two or more READ operations cannot be used in

succession, since only the first READ will produce valid data. A new conversion must be started using WRITE, and the conversion must be completed, before a new READ will produce valid data.

TABLE 1: Truth Table, Static RAM Mode

INPUTS		OUTPUTS		ADC-908 OPERATION
\overline{CS}	\overline{RD}	\overline{BUSY}	D87-D80	
L	H	H	HIGH-Z	Start Convert (Write Cycle)
L	\downarrow	H	HIGH-Z to DATA	Read Data (Read Cycle)
L	\uparrow	H	DATA to HIGH-Z	Reset Converter
H	X (Note 1)	X	HIGH-Z	No Effect (Not Selected)
L	H	L	HIGH-Z	No Effect (Converter Busy)
L	\downarrow	L	HIGH-Z	No Effect (Converter Busy)
L	\uparrow (Note 1)	L	HIGH-Z	Conversion Error Not Allowed

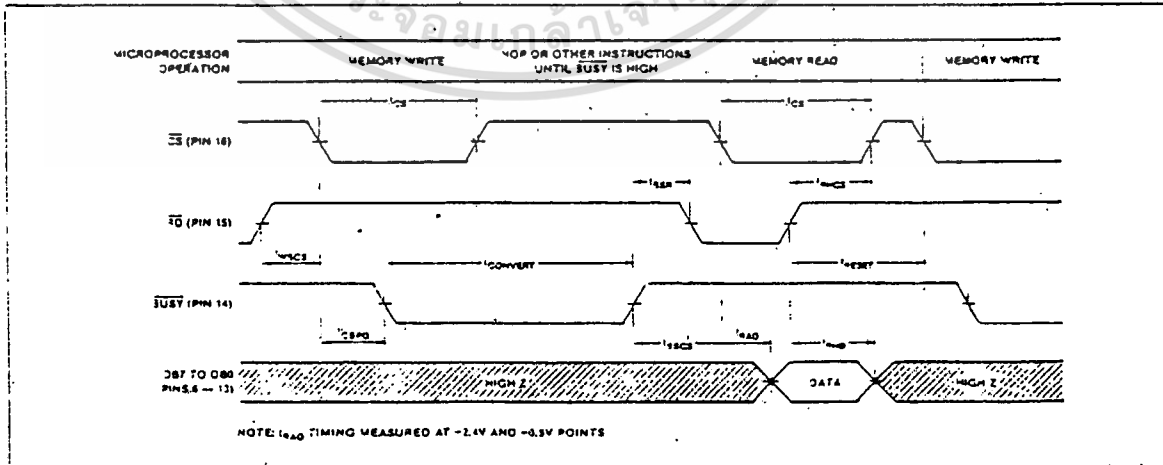
NOTE 1: if \overline{RD} goes LOW to HIGH, the A/D is internally reset, regardless of the states of \overline{CS} or \overline{BUSY}

OPERATING DESCRIPTION: ROM MODE

In ROM mode, input \overline{CS} is tied LOW, and input \overline{RD} is derived from the ADC-908 address decoder. To satisfy timing, it is recommended that the decoder be enabled by a system MEMRD (8080), YMA (8800), or similar strobe. (See Figure 3.)

In ROM mode, data is read by executing a READ instruction to the ADC-908 address. At the conclusion of the READ instruction, the ADC-908 automatically resets itself and then proceeds to perform a new data conversion. Output \overline{BUSY} is LOW during conversion. A new READ instruction to the ADC-908 must not be executed until \overline{BUSY} returns HIGH.

FIGURE 2: Static RAM Mode Timing Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



This requirement may be met by inserting NOP or other program instructions between consecutive READ operations. Conditional or branch instructions may be used, but keep in mind that data may become out-of-date if excessive time elapses between consecutive READ instructions.

TABLE 2: Truth Table, ROM Mode.

INPUTS		OUTPUTS		ADC-908 OPERATION
CS	RD	BUSY	D87-D80	
L	L	H	HIGH-Z to DATA	Read Data
L	L	L	DATA to HIGH-Z	Reset and Start New Conversion
L	L	L	HIGH-Z	No Effect (Converter Busy)
L	(NOTE 1)	L	HIGH-Z	Conversion Error Not Allowed

NOTE 1: If RD goes LOW to HIGH, the ADC is internally reset, regardless of the states of CS or BUSY.

OPERATING DESCRIPTION: SLOW-MEMORY MODE

The slow-memory mode is intended for systems in which the ADC-908 BUSY output is used as an interrupt to force the

microprocessor into WAIT states during data conversion.

In slow-memory mode, inputs CS and RD are tied together. The common RD and CS signal is derived from the ADC-908 address decoder. To satisfy the timing requirements, it is advisable to latch the address using ALE (8085) or SYNC (8080). For 8080 or 8085-based systems, connect the microprocessor READY input to the ADC-908 BUSY output. (See Figure 4.)

TABLE 3: Truth Table, Slow-Memory Mode

INPUTS		OUTPUTS		ADC-908 OPERATION
CS & RD	BUSY	D87-D80		
H	H	HIGH-Z		No Effect (Not Selected)
L	L	HIGH-Z		Start Conversion
L	L	HIGH-Z		Conversion in Progress. μ P in WAIT State
L	L	HIGH-Z to DATA		Conversion Complete. Read Data
L	L	DATA to HIGH-Z		Reset and Deselect Converter

FIGURE 3: ROM Mode Timing Diagram (CS Held LOW)

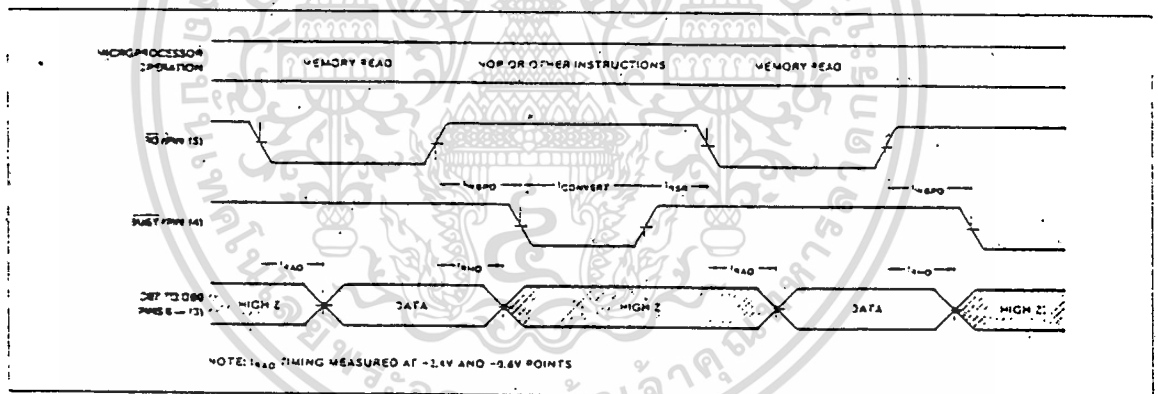
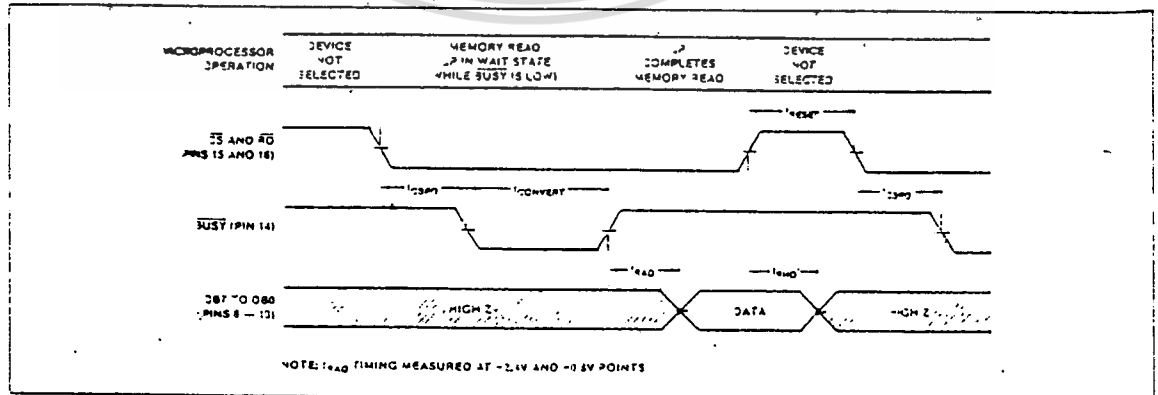


FIGURE 4: Slow-Memory Mode Timing Diagram (CS and RD Tied Together)





Do not execute a WRITE instruction at the ADC-908 address when in slow-memory mode, since bus conflicts will arise. In some architectures, an accidental WRITE instruction may be locked out in hardware, by proper strobing of the ADC-908 address decoder.

INITIALIZATION

In all operating modes, the ADC-908 is initialized by executing a READ instruction to the ADC-908 address. The data obtained should be ignored.

CLOCK OSCILLATOR

The ADC-908 may be used with its internal asynchronous clock oscillator. An external resistor and capacitor are required. Typical values are $R = 43k\Omega$ and $C = 100pF$, for conversion times in the $6\mu s$ range. For applications in which the fastest conversion times are required, an external clock is recommended. The external clock must be gated by the use of a 74125-type three-state buffer, with an output pullup resistor. Optimum conversion accuracy is obtained when CS goes LOW on a positive clock edge. The maximum external clock frequency is 1.25MHz (See Figure 5 and 6.)

REFERENCE VOLTAGE

A negative reference voltage must be applied to the ADC-908 V_{REF} input. Optimum full-scale accuracy is obtained using $-10.00V$, although V_{REF} may be $-5.00V$, $-10.24V$, or other voltages within its specified range.

Over the full temperature range, optimum gain accuracy is obtained when the input to the V_{REF} pin is from a low-impedance source. A resistor or trimmer may be used in series with the V_{REF} pin, but this trim technique is not as accurate as a low-impedance source. (See Figure 7.)

For a cost-effective $-10.00V$ or $-10.24V$ reference with excellent accuracy and low temperature coefficient, ask for PMI's REF-08. Consult your sales representative for availability.

FIGURE 5: Using the Internal Clock Oscillator

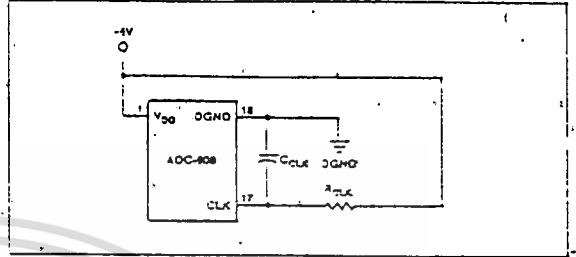


FIGURE 6: Using an External Clock

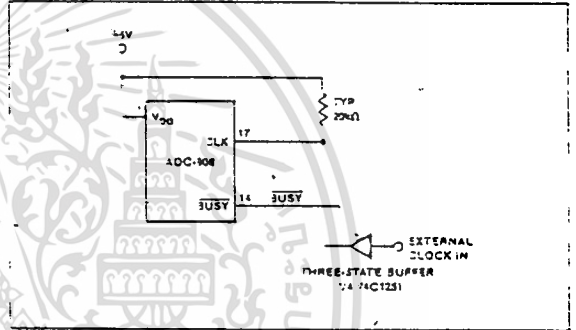
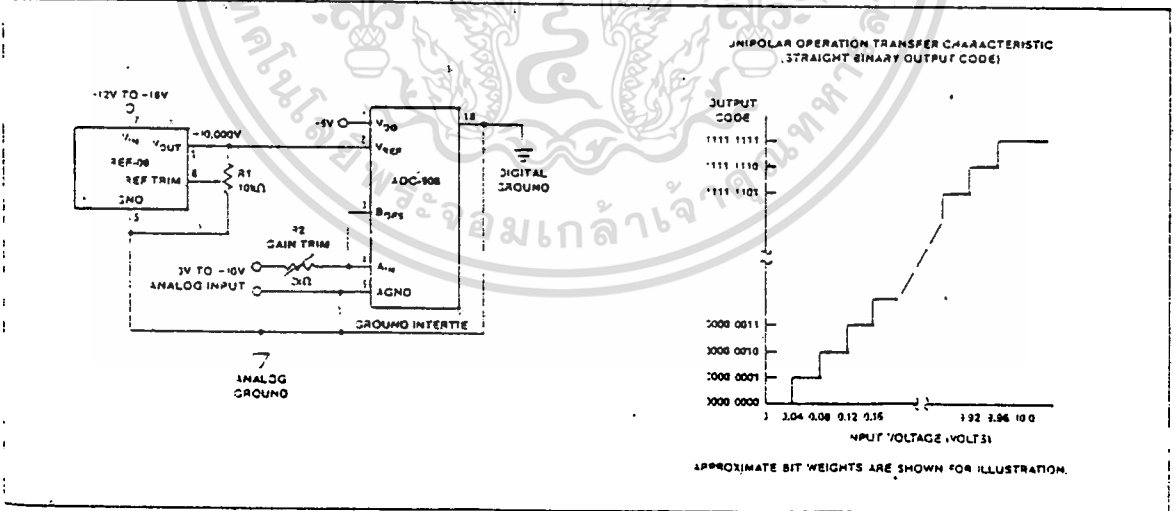


FIGURE 7: Unipolar Operation





Gain Adjustment

- 1) Apply +9.922V across the analog input.
- 2) While performing continuous conversions, adjust R3 until DB7 to DB1 are HIGH and DB0 (LSB) flickers.

DIGITAL CONSIDERATIONS

Control Timing—Fresh data from a recent conversion must be read before beginning a new conversion. Following the data READ, as RD goes HIGH, it resets the SAR and clears the data from the previous conversion.

The timing restrictions detailed in the interface timing diagrams must be observed to prevent the ADC-908 from changing interface modes. For example, if CS is held LOW too long while in RAM mode, the converter will change to ROM mode and initiate a new conversion.

Logic Deglitching—Unrelated activity on the address bus may cause unexpected glitch inputs to the ADC. The glitches may cause unwanted READs, resets, or conversions. In ROM or RAM modes, these may be avoided by gating the address decode logic with RD or WR (8080) or VMA (6800). In slow-memory mode, ALE (8085) or SYNC (8080) may be used to latch the address.

Initialization—Following power-up, the SAR is in an unknown state. Executing a memory READ (disregard the data) will reset the ADC.

ANALOG CONSIDERATIONS

Analog Input Impedances—Low impedance sources must be used to drive the VREF, AIN, and BOPF inputs. Excessive source

impedances may cause errors due to the loading effects of the inputs' finite impedances.

Ground Management—AGND and DGND pins should be connected at or near the ADC to minimize noise effects. If the two grounds cannot be connected near the ADC, the grounds should be clamped with back-to-back Schottky diodes between the AGND and DGND pins.

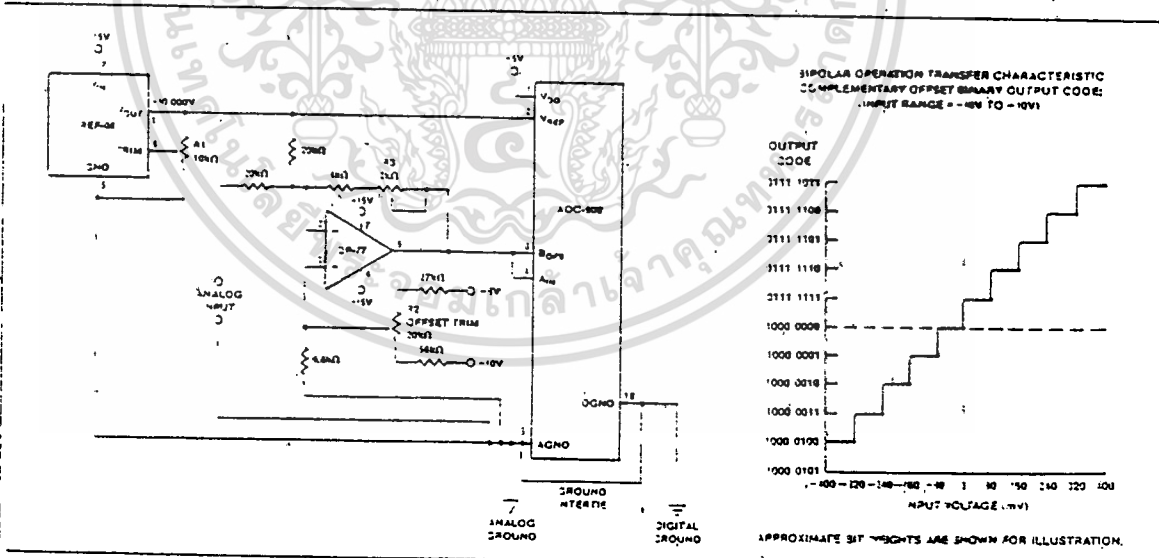
Offset Correction—Conversion offset errors may be corrected by counter-offsetting the buffer amplifier driving AIN. This offset correction may be accomplished by applying a correction current to the buffer's summing junction or by tapping a voltage divider sitting between VDD and VREF, and applying this tap voltage to the noninverting input of the buffer.

Ratiometric Operation—The R-2R type DAC in the ADC-908 permits ratiometric operation of the ADC. Performance degradation may, however, occur as VREF varies from -10.000V. This decrease in performance is due to comparator limitations including offset-voltage, gain, and input noise.

The ADC-908 uses the reference as a power supply for the comparator to increase speed and accuracy. Reference voltages of a magnitude less than -9V must be avoided for accurate comparator operation. For best accuracy, the use of a 0.1µF bypass capacitor from VREF (Pin 2) to AGND is recommended.

Power Supply Bypassing—For best accuracy, VDD (Pin 1) should be bypassed to AGND with a 0.1µF capacitor.

FIGURE 9: Complementary Offset Bipolar Operation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MONOLITHIC SAMPLE AND HOLD CIRCUITS

LF198/LF298/LF398

DESCRIPTION

The Signetics LF198/LF298/LF398 are monolithic sample and hold circuits which utilize high-voltage ion implant JFET technology to obtain ultra-high DC accuracy with fast acquisition of signal and low droop rate. Operating as a unity gain buffer, DC gain accuracy is 0.002% typical and acquisition time is as low as 6 μ s to 0.01%. A bipolar input stage is used to achieve low offset voltage and wide bandwidth. Input offset adjust is accomplished with a single pin and does not degrade input offset drift. The wide bandwidth allows the LF198 to be included inside the feedback loop of 1MHz op amps without having stability problems. Input impedance of $10^{10}\Omega$ allows high source impedances to be used without degrading accuracy.

P-channel junction FET's are combined with bipolar devices in the output amplifier to give droop rates as low as 5mV/min with a 1 μ F hold capacitor. The JFET's have much lower noise than MOS devices used in previous designs and do not exhibit high temperature instabilities. The overall design guarantees no load-through from input to output in the hold mode even for input signals equal to the supply voltages.

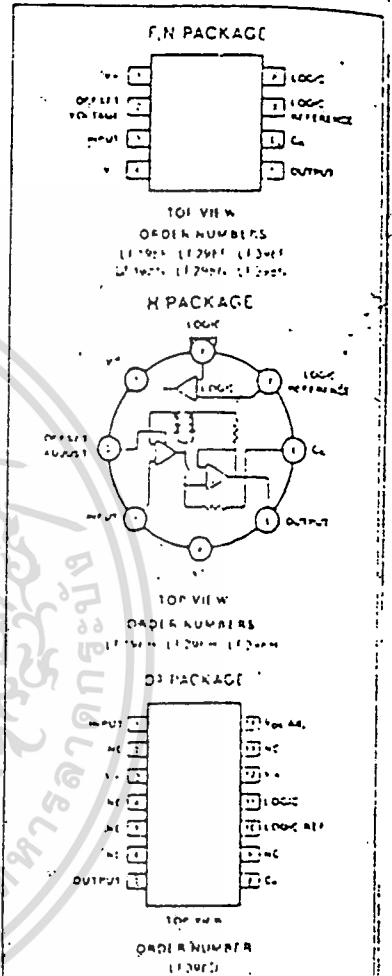
Logic inputs are fully differential with low input current, allowing direct connection to TTL, PMOS, and CMOS. Differential threshold is 1.4V. The LF198/LF298/LF398 will operate from $\pm 5V$ to $\pm 18V$ supplies. They are available in an 8-lead TO5 package, or an 8-pin plastic DIP.

FEATURES

- Operates from $\pm 5V$ to $\pm 18V$ supplies
- Less than 10 μ s acquisition time
- TTL, PMOS, CMOS compatible logic input
- 0.5mV typical hold step at $C_H = 0.01\mu F$
- Low input offset
- 0.002% gain accuracy
- Low output noise in hold mode
- Input characteristics do not change during hold mode
- High supply rejection ratio in sample or hold
- Wide bandwidth

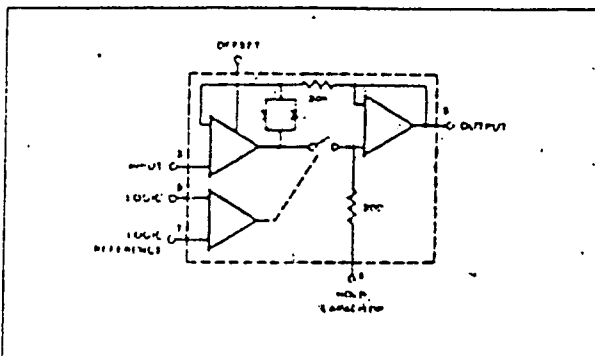
APPLICATIONS

- The LF198/LF298/LF398 are ideally suited for a wide variety of sample and hold applications including data acquisition, analog-to-digital conversion, synchronous demodulation, and automatic test setup.

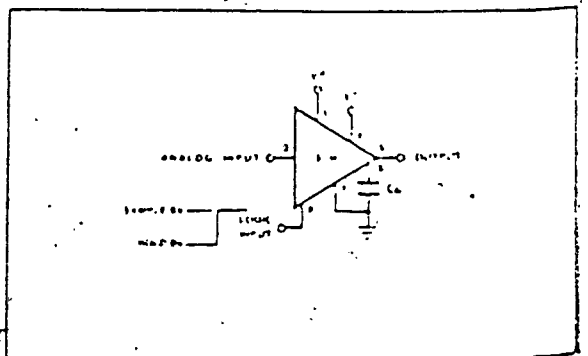


- NOTES
1. SOI - Released in Large SO package only
 2. SOI and non-standard pinout
 3. SOI and non-standard pinouts

FUNCTIONAL DIAGRAM



TYPICAL APPLICATIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการนี้ใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MONOLITHIC SAMPLE AND HOLD CIRCUITS

LF198/LF298/LF398

ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Supply voltage	± 18	V
Power dissipation (package limitation) ¹	500	mW
Operating ambient temperature range		
LF198	-55 to +125	°C
LF298	-25 to +85	°C
LF398	0 to +70	°C
Storage temperature range	-65 to +150	°C
Input voltage	Equal to supply voltage	
Logic to logic reference differential voltage ²	+7, -30	V
Output short circuit duration	Indefinite	
Hold capacitor short circuit duration	10	sec
Lead temperature (soldering, 10sec)	300	°C

DC ELECTRICAL CHARACTERISTICS

Unless otherwise specified, the following conditions apply. Unit is in "sample" mode. $V_S = \pm 15V$, $T_1 = 25^\circ C$, $-11.5V \leq V_{IN} \leq +11.5V$, $C_H = 0.01\mu F$, and $R_L = 10k\Omega$. Logic reference voltage = 0V and logic voltage = 2.5V.

PARAMETER	TEST CONDITIONS	LF198/LF298			LF398			UNIT
		Min	Typ	Max	Min	Typ	Max	
Input offset voltage ³	$T_1 = 25^\circ C$	1	3	5	2	7	10	mV
Input bias current ⁴	$T_1 = 25^\circ C$ Full temperature range	5	25	75	10	50	100	nA
Input impedance	$T_1 = 25^\circ C$	10^{10}			10^{10}			Ω
Gain error	$T_1 = 25^\circ C$, $R_L = 10k$ Full temperature range	0.002	0.005	0.01	0.004	0.01	0.02	%
Feedthrough attenuation ratio at 1kHz	$T_1 = 25^\circ C$, $C_H = 0.01\mu F$	60	60	60	60	60	60	dB
Output impedance	$T_1 = 25^\circ C$, "HOLD" mode Full temperature range	0.5	2	4	0.5	4	6	Ω
"HOLD" step ⁵	$T_1 = 25^\circ C$, $C_H = 0.01\mu F$, $V_{OUT} = 0$	0.5	2.0	4	1.0	2.5	4	mV
Supply current ⁶	$T_1 = 25^\circ C$	4.5	5.5	6	4.5	6.5	7	mA
Logic and logic reference input current	$T_1 = 25^\circ C$	2	10	15	2	10	15	μA
Leakage current into hold capacitor ⁷	$T_1 = 25^\circ C$, Hold mode	30	100	150	30	200	300	nA
Acquisition time to 0.1%	$\Delta V_{OUT} = 10V$, $C_H = 100pF$ $C_H = 0.01\mu F$	4	20	30	4	20	30	μs
Hold capacitor charging current	$V_{IN} - V_{OUT} = 2V$	5	10	15	5	10	15	mA
Supply voltage rejection ratio	$V_{OUT} = 0$	60	110	150	60	110	150	dB
Differential logic threshold	$T_1 = 25^\circ C$	0.6	1.4	2.4	0.6	1.4	2.4	V

NOTES

1 The maximum junction temperature of the LF398 is 150°C. When operating at elevated ambient temperature, the TO-8 and plastic DIP packages must be derated based on a thermal resistance (θ_{JA}) of 160°C/W.

2 Although the differential voltage may not exceed the limits given, the common mode voltage on the logic pins may be equal to the supply voltage without causing damage to the device. For proper logic operation, however, one of the logic pins must always be at least 2V below the positive supply and 3V above the negative supply.

3 Unless otherwise specified, the following conditions apply. Unit is in "sample" mode. $V_S = \pm 15V$, $T_1 = 25^\circ C$, $-11.5V \leq V_{IN} \leq +11.5V$, $C_H = 0.01\mu F$, and $R_L = 10k\Omega$. Logic reference voltage = 0V and logic voltage = 2.5V.

4 Hold step is sensitive to stray capacitance loading between input logic signals and the hold capacitor. For 100pF or more, add three or an additional 0.1mV step with 1V logic swing and a 0.01 μF hold capacitor. Maximum of the hold step is inversely proportional to hold capacitor value.

5 Leakage current is measured at a junction temperature of 75°C. The effects of junction temperature are due to power dissipation and generated ambient conduction, not to the 25°C value. For each 11°C increase in junction temperature, leakage is guaranteed over full input signal range.

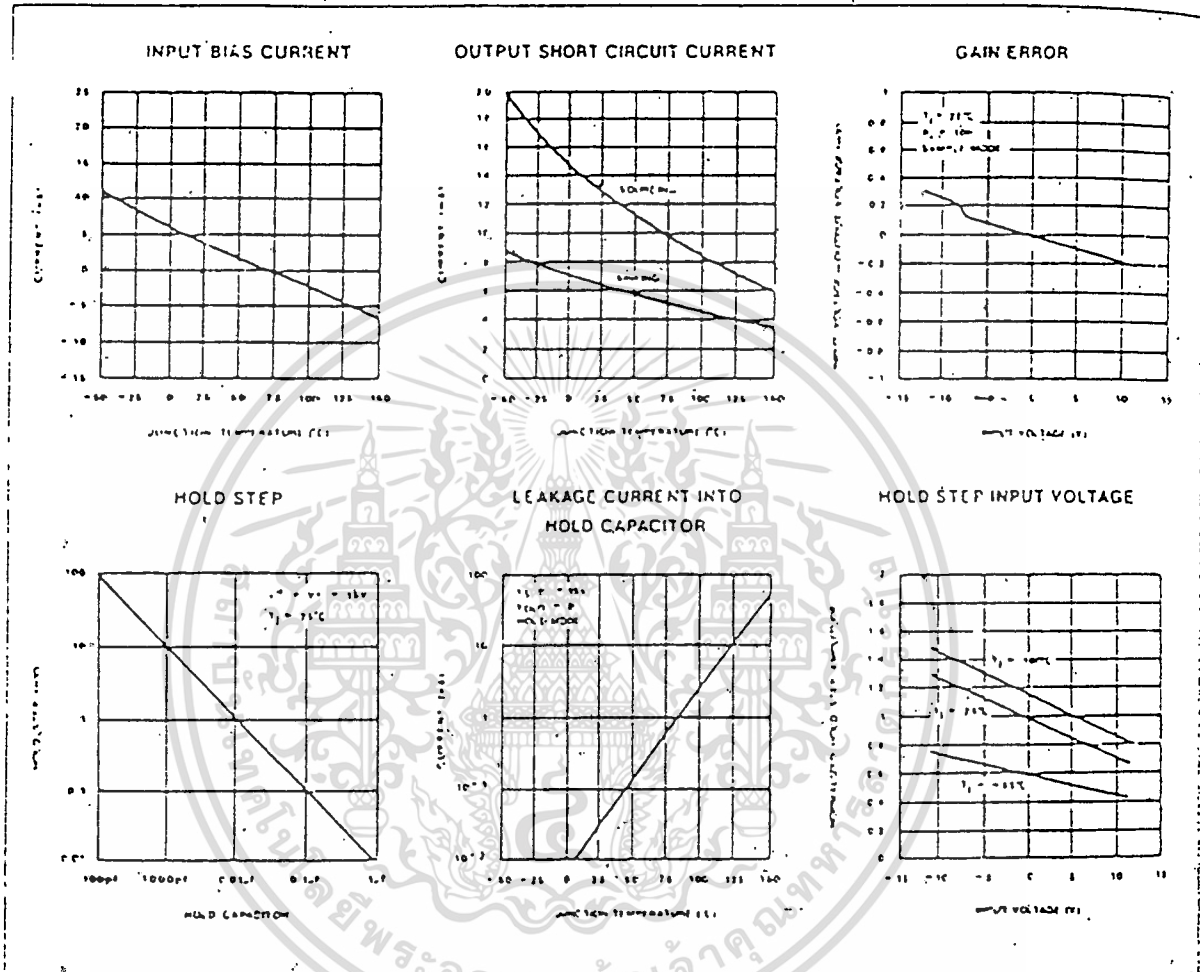
6 The parameters guaranteed over a supply voltage of 2.5 to 15V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

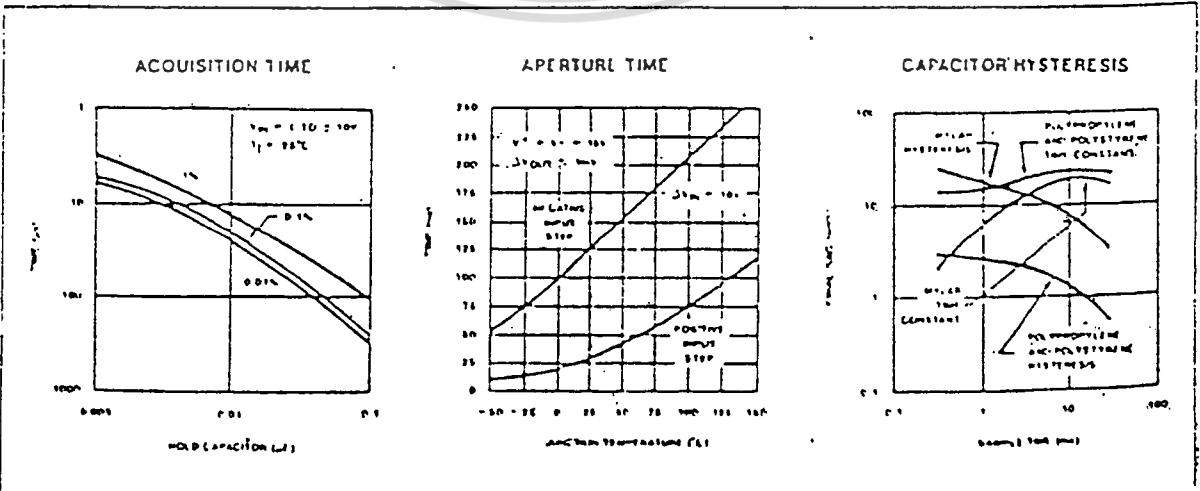
MONOLITHIC SAMPLE AND HOLD CIRCUITS

LF198/LF298/LF398

TYPICAL DC PERFORMANCE CHARACTERISTICS



TYPICAL AC PERFORMANCE CHARACTERISTICS



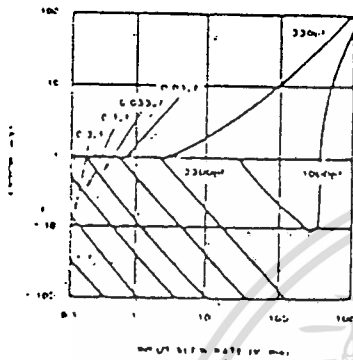
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MONOLITHIC SAMPLE AND HOLD CIRCUITS

LF198/LF298/LF398

TYPICAL AC PERFORMANCE CHARACTERISTICS (cont'd)

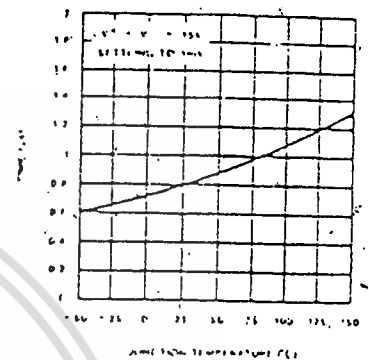
DYNAMIC SAMPLING ERROR



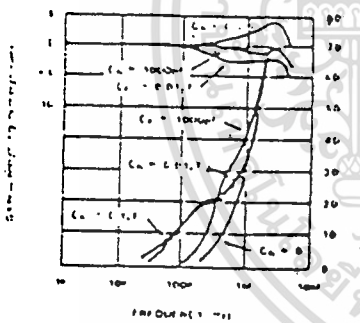
OUTPUT DROOP RATE



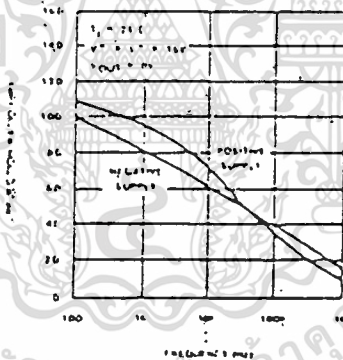
'HOLD' SETTLING TIME



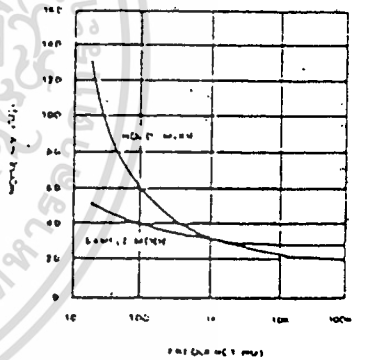
PHASE AND GAIN (INPUT TO OUTPUT, SMALL SIGNAL)



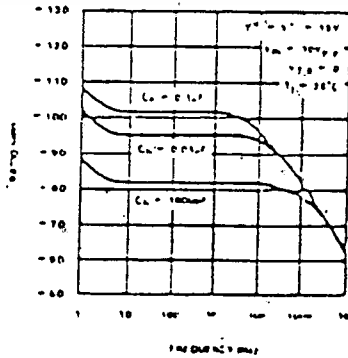
POWER SUPPLY REJECTION



OUTPUT NOISE



FEEDTHROUGH REJECTION RATIO (HOLD MODE)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <io.h>
#include <dos.h>
#include <math.h>
#include <bios.h>
#include <time.h>
#include <fcntl.h>
#include <alloc.h>
#include <conio.h>
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <graphics.h>
/*scanner.c*/
void data();
void kbenter(int x,int y,int charecter,char *msg,int kbb);
void kbdata(int x,int charrecter,char *msg1);
void loaddata(void);
int ldscr();
void menusave();
void note();
void printer();
void plotpage(int bb);
int read_ad(int b);
void restore(void);
void savedata(void);
void setint(void);
void Image(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void XMSmenu();
void Ramdom();
void interrupt int8(void);
void interrupt (*oldfunc)();
void chow_sh();

```

```

struct atod(                                     /* use save data all */
    char   grade0[5];
    char   grade1[5];
    char   grade2[5];
    char   grade3[5];
    char   grade4[5];
    char   grade5[5];
    char   grade6[12];
    int    sh[8];
    char   dnote[20];
    unsigned char memseg1[560];
    unsigned char memseg2[560];
)mdata;

unsigned char a,vdata[2400];mux[2400];
unsigned char index_y1,index_y2; /* use by input data */;
unsigned int  index=0,saveflag; /* use by save/res screen */
unsigned int  x,y,x1,y1,page=0; /* use by plot graph & toggle page*/
char  ascii;                                     /* use by readkbd */
char  kload[]=" ";
char  sh[]={0,0,0,0,0,0,0,0};
char  *grade[]=(" " " " " " " " " " P","TXT.IDX " "TXT.DAT " ");
char  kbbuff[40],kbbuff1[20],kbbuff2[20],fname[20],dnote[20];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int    *image1,kb_en,kb_data,point,endline,kbdy,kbesc;
int    msh[2][8],maxx=639,maxy=349,midx=319,midy=174;
int    count,idx=0;
int    m,scancode,i;

/*****
**/
/*          MAIN PROGRAM          */
/*****
**/
void main ()
{
    char    *msg;
    int     x,y;
    char    flag=0;

/* Set control word 8255 */
    outportb(0x703,0x8a);
    outportb(0x700,0);

/* Display menu & tittle */
    Initialize();
    title();
    playt();
    XMSmenu(midx-160,midy-112);
    getch();
    icon();

/* Display Information */
    x=338,y=35;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(11);

printf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],grade[3],
grade[4],grade[5]);

/* Loop for read keyboard */
for (;;)
{
    if (flag==0)
    {
        setcolor(0);
        board(0,maxy-24,maxx,maxy,7);
        setcolor(1);
        outtextxy(0,maxy-15," F1:   F2:   F3:   F4:   F5:   F6:   F7:   F8:
F9:   F10: ");
        setcolor(15);
        outtextxy(0,maxy-15," P/B   F.F   R.W   STOP   SAVE   LOAD   PRINT
DATA   NOTE   QUIT ");
        setcolor(1);
        flag=1;
    }
    readkb();

    if(scancode == 0x3b00){ /* Help service F1 */
        x=338,y=35;
        setcolor(11);

        printf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],
grade[3],grade[4],grade[5]);

        setcolor(0);
        msg=grade[5];
        msg+=4;

        if (*msg==0x42)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        flag=0x50;
    if (*msg==0x62)
        flag=0x50;
    if (*msg==0x50)
        flag=0x42;
    if (*msg==0x70)
        flag=0x42;
    *msg=flag;
    x=338,y=33;
    setcolor(11);
    gprintf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],
grade[3],grade[4],grade[5]);
    setcolor(1);
    flag=0;
}
if(scancode == 0x3c00){ /* F.F sercive F2 */
    read_ad(0);
    flag=0;
}
if(scancode == 0x3d00){ /* R.W sercive F3 */
    read_ad(1);
    flag=0;
}
if(scancode == 0x3f00){ /* Save sercive F5 */
    menusave();
    flag=0;
}
if(scancode == 0x4000){ /* Load sercive F6 */
    loaddata();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        flag=0;
    }

    if(scancode == 0x4100){                /* Print servive F7 */
        printer();
        flag=0;
    }

    if(scancode == 0x4200){                /* Data servive F8 */
        data();
        flag=0;
    }

    if(scancode == 0x4300){                /* Exit servive F9 */
        note();
        flag=0;
    }

    if(scancode == 0x4400){                /* Exit servive F10 */
        closegraph();
        free(image1);
        exit(0);
    }

    else{

        } /* End readkbd loop */

    } /* End function MAIN */

    /*****/
    /* Read data from a/d */
    /*****/

    int read_ad(int b)
    {
        unsigned char a,z;

        int m1,m2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Clear allocate memory */ /* clrmem(); */
index=0;
saveflag=0; /* srscr(); */
/* Set loop FF. & RW. */
for (;;)
{
/* Print active mode.*/
board(0,maxy-24,maxx,maxy,7); /* Print active mode */
setcolor(0);
if (b==0){
outtextxy(0,maxy-15," FORWARD READ: Read data from A/D &
Plotgraph on screen. STOP.");
setcolor(4);
outtextxy(0,maxy-15,"
");
}
else {
outtextxy(0,maxy-15," REVERSE READ: Read data from A/D & Plotgraph
on screen. F4:STOP.");
setcolor(4);
outtextxy(0,maxy-15,"
");
}

setcolor(1);
/* Start motor */
if (b==0)
{
/* Start motor F.F */
a=inportb(0x700);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ทำกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        a=a&0x07;
        a=a|0x08;
        .outportb(0x700,a);
    }
else
{
        a=inportb(0x700);          /* Start motor R.W */
        a=a&0x07;
        a=a|0x20;
        outportb(0x700,a);
    }
/* Clear SH1-SH8 */
z=0;
for (a=0 ; a < 8 ; a++)
    mdata.sh[a]=z;
for(m1=0;m1<2;m1++)
for(m2=0;m2<8;m2++)
    msh[m1][m2]=0;
/* Start read A/D */
plotpage(b);
/* Esc to exit */
if (scancode==0x011b)
    break;
/* Read key loop */
for (;;)
{
    readkbd();
/* Esc key press */
if (scancode==0x011b)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
/* F4 stop */
        if (scancode==0x3e00)
            break;
/* F2 key press */
        if (scancode==0x3c00)
        {
            b=0;
            break;
        }
/* F3 key press */
        if (scancode==0x3d00)
        {
            b=1;
            break;
        }
/* F5 key press */
        if (scancode==0x3f00)
        {
            menu_save();
            continue;
        }
/* F9 key press */
        if (scancode==0x4300)
        {
            note();
        }
/* Print main menu */

```

```
board(0,maxy-24,maxx,maxy,7);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(0);

outtextxy(0,maxy-15,"READ MODE: P/B F.F R.W SAVE NOTE to
EXIT.");

setcolor(4);

outtextxy(0,maxy-15," F1 F2 F3 F5 F9 Esc ");

setcolor(1);

        continue;

    }

    } /* End key loop */

    if (scancode==0x011b)
        break;

} /* End loop FF. & RW. */

}

/******
/*      Plot grap 2 page      */
/******

void plotpage(int bb)
{
    int x,y,d,mc,z,m2,m1;
    char a;

    /* Set interrupt vector */
    setint();

    /* Loop display on screen */
    for (;;)
    {

    /* Print box */

        setfillstyle(1,8);
        bar(7,33,330,43);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(15);
    d=15;
for (mc=8;mc > 0;mc--)
{
    rectangle(d,33,d+24,43);
    d+=40;
}

/* Display 640 pixel 2 chanel page 1 */
if(page==1)
{
    icon();
    x=20,y=35;
    setcolor(11);
    fprintf(x,y,"%2d %2d %2d %2d %2d %2d %2d %2d",mdata.sh[0],
mdata.sh[1],mdata.sh[2],
    mdata.sh[3],mdata.sh[4],mdata.sh[5],mdata.sh[6],mdata.sh[7]);
    x=338,y=35;
    setcolor(11);
    fprintf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],grade
[3],grade[4],grade[5]);
    outtextxy(110,200,mdata.dnote);
    Time();
    board(0,maxy-24,maxx,maxy,7);    /* Print active mode */
    setcolor(0);
    if (bb==0){
        outtextxy(0,maxy-15," FORWARD READ: Read data from A/D &
Plotgraph on screen.  STOP.");
        setcolor(4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        outtextxy(0,maxy-15,"
");
    }
    else{
        outtextxy(0,maxy-15," REVERSE READ: Read data from A/D & Plotgraph
on screen. F4:STOP.");
        setcolor(4);
        outtextxy(0,maxy-15,"
");
    }
    setcolor(1);
} /* if */
page^=1;
setactivepage (page);

setcolor(0);
if (bb==0){
    outtextxy(0,maxy-15," FORWARD READ: Read data from A/D &
Plotgraph on screen. STOP.");
    setcolor(4);
    outtextxy(0,maxy-15,"
");
}
else {
    outtextxy(0,maxy-15," REVERSE READ: Read data from A/D & Plotgraph
on screen. F4:STOP.");
    setcolor(4);

```

```

        outtextxy(0,maxy-15,"
";
    }
    setcolor(1);

    page^=1;
    setactivepage (page);

/* Clear SH1-SH8 */
    z=0;
    for ( a=0 ; a < 8 ; a++)
        mdata.sh[a]=z;
    for(m1=0;m1<2;m1++)
        for(m2=0;m2<8;m2++)
            msh[m1][m2]=0;

    chow_sh();

    setfillstyle(1,8);
    bar(7,33,330,43);
    setcolor(15);
        d=15;
    for (mc=8;mc > 0;mc--)
    {
        rectangle(d,33,d+24,43);
        d+=40;
    }

        x=20,y=35;
    setcolor(11);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gprintf(x,y,"%2d %2d %2d %2d %2d %2d %2d %2d",mdata.sh[0],
mdata.sh[1],mdata.sh[2],
mdata.sh[3],mdata.sh[4],mdata.sh[5],mdata.sh[6],mdata.sh[7]);
x=338,y=35;
setcolor(11);
gprintf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],grade[3],
grade[4],grade[5]);
outtextxy(110,200,mdata.dnote);

x=40,y=161;
Gimage(89); setcolor(11);
plot(x,y);

x=40,y=283;
Gimage(211); setcolor(11);
plott(x,y);

setvisualpage (page);
page^=1;
setactivepage (page);

```

```

/* If key press -> break */

```

```

readkbd();

```

```

if (scancode==0x3e00) /* F4 Stop */

```

```

{

```

```

/* Stop motor */

```

```

outportb(0x700,0x10);

```

```

outportb(0x700,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        restore();
        break;
    }
/* If key press -> break */
    if (scaancode==0x011b)        /* Esc */
    {
/* Stop motor */
        outportb(0x700,0x10);
        outportb(0x700,0);
        restore();
        break;
    }
} /* End loop display on screen */
/* Print main menu */

setvisualpage (page);

board(0,maxy-24,maxx,maxy,7);        /* Print active mode */
setcolor(0);
outtextxy(0,maxy-15," READ MODE:  P/B   F.F   R.W   SAVE   NOTE   to
EXIT:");

setcolor(4);

outtextxy(0,maxy-15,"          F1:   F2:   F3:   F5:   F9:   Esc   ");

setcolor(1);

}

/*****/

/*      chow data int to sh      */

/*****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void chow_sh()
{
    int ssh,sum,sum1,m1,m2,n=1,b,sidx,count;

    sidx=idx;
    if(sidx<560)
    {
        b=0;
        for(count=0;count<sidx;count++)
        {
            m2=mux[count];
            m1=msh[n-1][b];
            if(m1>m2){
                msh[n][b]=m1;
                mdata.sh[b]=m1;
            }
            if(m2>m1){
                msh[n][b]=m2;
                mdata.sh[b]=m2;
            }
            msh[n-1][b]=mdata.sh[b];
            if(mdata.sh[b]>100)
                mdata.sh[b]=99;
            if(msh[n-1][b]>100)
                msh[n-1][b]=99;
            if(msh[n][b]>100)
                msh[n][b]=99;
            if(m2>100)
                m2=99;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        b++;
        if(b > 7)
            b=0;
    } /* for */
} /* if */
else
{
    ssh=sidx-559;
    sum=ssh/8;
    sum1=sum*8;
    b=ssh-sum1;
    for(count=0;count<maxx;count++)
    {
        m2=mux[sidx-maxx+count]; /* input data */
        m1=msh[n-1][b];
        if(m1>m2){
            msh[n][b]=m1;
            mdata.sh[b]=m1;
        }
        if(m2>m1){
            msh[n][b]=m2;
            mdata.sh[b]=m2;
        }
        msh[n-1][b]=mdata.sh[b];
        if(mdata.sh[b]>100)
            mdata.sh[b]=99;
        if(msh[n-1][b]>100)
            msh[n-1][b]=99;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(msh[n][b]>100)
        msh[n][b]=99;
    if(m2>100)
        m2=99;
    b++;
    if(b > 7)
        b=0;
} /* for */
} /* else */
}
/*****
/* Plot graph from buffer memory to monitor display CH1 */
*****/
int plot(int x,int y)
{
    int x1=40,y1,a,psh,pidx,psum,psum1,p,p2,aa=0,tmem=0;

    index=idx;
    pidx=idx;
    moveto(x,y);

    if (index<4480)
    {
        for(count=0;count<index;count++)
        {
            psum=count/8;
            psum1=psum*8;
            p=count-psum1;

            if(p<4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    for(p2=0;p2<4;p2++)
    {
        a=vdata[count]; /* input data */
        aa+=a;
        count++;
    }

    aa=aa/4;
    mdata.memseg1[tmem]=aa;
    tmem++;
    y1=168-aa;
    if(y1 > 185) { y1 = 185; }
    if(y1 < 90) { y1 = 90; }
    lineto(x1,y1);
    x1++;
    if(x1==600)
        x1=39;
    }
} /* for */
}

else
{
    for(count=0;count<4480;count++)
    {
        psh=pidx-4479;
        psum=psh/8;
        psum1=psum*8;
        p=psh-psum1;

        if(p<4)

```

```

{
for(p2=0;p2<4;p2++)
{
a=vdata[index+count-4479]; /* input data */
aa+=a;
count++; pidx++;
}

aa=aa/4;
mdata.memseg1[tmem]=aa;
tmem++;
y1=168-aa;
if(y1 > 185) { y1 = 185; }
if(y1 < 90) { y1 = 90; }
lineto(x1,y1);
x1++;
if(x1==600){
x1=39; y1=168; moveto(39,168);
}
}
pidx++;
}/* for */
}

y=y1;
}

/*****
/* Plot graph from buffer memory to monitor display CH2 */
*****/

int plott(int x,int y)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int x2=40,y2,a,psh,pidx,psum,psum1,p,pp,aa=0,tmem=0;
```

```
index=idx;
```

```
pidx=idx;
```

```
moveto(x,y);
```

```
if (index<4480)
```

```
{
```

```
for(count=0;count<index;count++)
```

```
{
```

```
psum=count/8;
```

```
psum1=psum*8;
```

```
p=count-psum1;
```

```
if(p>=4)
```

```
{
```

```
for(pp=0;pp<4;pp++)
```

```
{
```

```
a=vdata[count]; /* input data */
```

```
aa+=a;
```

```
count++;
```

```
}
```

```
aa=aa/4;
```

```
mdata.memseg2[tmem]=aa;
```

```
tmem++;
```

```
y2=290-aa;
```

```
if(y2 > 307) { y2 = 307; }
```

```
if(y2 < 212) { y2 = 212; }
```

```
lineto(x2,y2);
```

```
x2++;
```

```
if(x2==600)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        x2=39;
    }
} /* for */
}
else
{
    for(count=0;count<4480;count++)
    {
        psh=pidx-1119;
        psum=psh/8;
        psum1=psum*8;
        p=psh-psum1;
        if(p>=4)
        {
            for(pp=0;pp<4;pp++)
            {
                a=vdata[index+count-4479]; /* input data */
                aa+=a;
                count++; pidx++;
            }
            aa=aa/4;
            mdata.memseg2[tmem]=aa;
            tmem++;
            y2=290-aa;
            if(y2 > 307) { y2 = 307; }
            if(y2 < 212) { y2 = 212; }
            lineto(x2,y2);
            x2++;
            if(x2==600){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        x2=39; moveto(39,290);
    }
}
pidx++;
} /* for */
}
y=y2;
}

/******/
/* Time / Day */
/******/
Time()
{
    int x,y;
    struct tm *newtime;
    time_t aclock;
    setcolor(7);
        x=425,y=70;
/* Time */
    setcolor(8);
    gprintf(x,y,""); /* edit putimage */
    setcolor(11);
    time(&aclock);
    newtime=localtime(&aclock);
    gprintf(x,y,"%s",asctime(newtime));
}

/******/
/* Read key */
/******/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

readkbd()
{
/* Scancode => Key */
    Time();
    scancode = bioskey(1);
    if(scancode != 0) bioskey(0);
    return(scancode);
}

/*****
/*      XMS install      */
*****/
void XMSmenu(int x,int y)
{
    board(x,y+24,x+311,y+212,7);
    setcolor(4);
    gprintf(x+13,y+(12*3),"Initial XMS");
    if(!XMSinstalled())
    {
        gprintf(x+13,y+(12*4),"This program use Extended memory and HIMEM.SYS");
        exit(1);
    }
    gprintf(x+12,y+(13*4),"Initial passed..");
    gprintf(x+12,y+(13*5),"Conventional memory left %1u bytes",farcoreleft());
    gprintf(x+12,y+(13*6),"Extended memory left %d Kbytes",XMSleftall());
    gprintf(x+12,y+(13*7),"Allocate memory");
    getch();
}

/*****
/*      Note service routine      */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
void note(void)
{
    int    num,x,y;
    char   a;

/* Display comment */
    board(0,maxy-24,maxx,maxy,7);    /* Print active mode */
    setcolor(0);
    outtextxy(0,maxy-15,"    In this mode you can enter COMMENT.    to EXIT.");
    setcolor(4);
    outtextxy(0,maxy-15,"    Note:                                ESC    ");
    setcolor(11);
    x=110,y=200;
    setcolor(8);
    outtextxy(110,200,"    ");
    setcolor(11);
    outtextxy(x,y,"_");
    num=0,a=0;
    mdata.dnote[num+1]=a;

/* Loop for readkbd */
    for (;;)
    {
        scancode=ascii=getch();

        if(scancode == 27)    /* ESC */
        {
            setcolor(8);

                o u t t e x t x y ( 1 1 0 , 2 0 0 , "    ");

            break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    if(scancode == 34)    /* " */
    {
        break;
    }

    if(scancode == 13)    /* Enter */
    {
        break;           /* continue; */
    }
else
{
    if (num==70)
        continue;
    mdata.dnote[num]=ascii;
    num++;
    a=0x5f;
    mdata.dnote[num]=a;
    a=0;
    mdata.dnote[num+1]=a;
    setcolor(8);
    outtextxy(110,200,"
");
    setcolor(11);
    outtextxy(110,200,mdata.dnote);
    x+=8;
}
}

/* End of loop readkbd */
a=0;
mdata.dnote[num]=a;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outtextxy(110,200,mdata.dnote);
    setcolor(8);
    outtextxy(110,200,"                               ");
    setcolor(11);
    outtextxy(110,200,mdata.dnote);
}
/*****/
/*  Data menu  */
/*****/
void data(void)
{
    char *msg;
    int a,b,mç,m;
/* Delete box */
    x=338,y=35,kb_data=0;
    setcolor(11);
    gprintf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],grade[3],
grade[4],grade[5]);
    setcolor(1);
/* Print mode active */
    board(0,maxy-24,maxx,maxy,7); /* Print active mode */
    setcolor(4);
    outtextxy(0,maxy-15," DATA:                               OBJ.           Esc
");
    setcolor(0);
    outtextxy(0,maxy-15," In mode you can install specifice your following the
menu, to EXIT.");
    setcolor(11);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Image for MENU */
a=(maxx/2),b=(maxy/2);
/* Draw data menu by put image */
image1=malloc(imagesize(209,94,429,254));
getimage(209,94,429,254,image1);
setcolor(7);
bar(a-110,b-80,a+110,b+80);
setcolor(0);
rectangle(a-110+2,b-80+2,a+110-2,b+80-2);
rectangle(a-110+4,b-80+4,a+110-4,b+80-4);
rectangle(a-110+4,b-80+25,a+110-4,b+80-25);
line(a-110+4,b+80-57,a+110-4,b+80-57);
x=a-110,y=b-70; /* 209,169 */
setcolor(4);
gprintf(x,y," EDUCATION ENGINEER MENU");
gprintf(x,y+25," GRADE:");
gprintf(x,y+35," #/FT:");
gprintf(x,y+45," WALL:");
gprintf(x,y+55," OD:");
gprintf(x,y+65," P_NO:");
gprintf(x,y+75," P/B:");
gprintf(x,y+100," F_INDEX:");
gprintf(x,y+110," F_DATA:");
gprintf(x,y+132," Esc to Exit.");
setcolor(0);
gprintf(x,y+25," %s 4 CHAR",grade[0]);
gprintf(x,y+35," %s",grade[1]);
gprintf(x,y+45," %s",grade[2]);
gprintf(x,y+55," %s",grade[3]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gprintf(x,y+65,"      %s",grade[4]);
gprintf(x,y+75,"      %s",grade[5]);
gprintf(x,y+100,"     %s",grade[6]);
gprintf(x,y+110,"     %s",grade[7]);
/* Set start address */
x=a-110+10,y=b-70+15;
kbdy=y;
point=0; endl=0;
/* Read keyboard */
do{
    msg=grade[point];
    if (kbdy < 193)
        kbddata(x,6,msg);
    if(point==1)
    {
        for(m=0;m<4;m++)
            mdata.grade0[m]=*msg++;
    }
    if(point==2)
    {
        for(m=0;m<4;m++)
            mdata.grade1[m]=*msg++;
    }
    if(point==3)
    {
        for(m=0;m<4;m++)
            mdata.grade2[m]=*msg++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(point==4)
{
for(m=0;m<4;m++)
mdata.grade3[m]=*msg++;
}
if(point==5)
{
for(m=0;m<4;m++)
mdata.grade4[m]=*msg++;
}
if(point==6)
{
for(m=0;m<4;m++)
mdata.grade5[m]=*msg++;
}
if(kbdy>193)
kbenter(x,kbdy,16,msg,1); /* EDIT */
if(point==7)
{
for(m=0;m<12;m++)
mdata.grade6[m]=*msg++;
}

}while( kb_data==0);
/* Display box */
setfillstyle(1,8);
bar(325,33,maxx-5,43);
setcolor(15);
a=335;

```

```
for(mc=6;mc > 0;mc--).
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    rectangle(a,33,a+40,43);
        a+=50;
}
x=338,y=35;
setcolor(11);
gprintf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],grade[3],
grade[4],grade[5]);
kb_data=0;

page^=1;
setactivepage (page);

setfillstyle(1,8);
bar(325,33,maxx-5,43);
setcolor(15);
a=335;
for (mc=6;mc > 0;mc--)
{
    rectangle(a,33;a+40,43);
        a+=50;
}
x=338,y=35;
setcolor(11);
gprintf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],grade[3],
grade[4],grade[5]);

page^=1;
setactivepage (page);

```

```

    putimage(209,94,image1,0);
}
/******
/*      Kbdata one line      */
/******

void kbdata(int x,int character,char *msg)
{
    int    kbd2,kbindex2;
    char   flag2=0;

/* Clear key buffer */
    flag2=0;
    setcolor(1);
    ascii=0;
    for(kbindex2=0;kbindex2 < 20;kbindex2++)
        kbuff2[kbindex2]=ascii;
/* Display on x,y position */
    setcolor(7);
    outtextxy(x+70,kbdy+10,"");
    setcolor(0);
    outtextxy(x+70,kbdy+10,"_");
    kbindex2=0;

/* Wait for key press */
    for (;;)
    {
        scancode=ascii=getch();
        if (scancode==27)          /* key ESC */
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        kb_data +=1;
        break;
    }

    /* Down key press , key ENTER */
    if (scancode==13)
    {
        point++;
        kbdy+=10;
        break;
    }
    if (ascii!=0)
    {
        /* Back space key press */
        if (scancode==32)
        {
            if (kbindex2!=0)
            {
                kbindex2--;
                ascii=0x5f;
                kbuff2[kbindex2]=ascii;
                ascii=0x20;
                kbuff2[kbindex2+1]=ascii;
                setcolor(7);
                outtextxy(x+70,kbdy+10,"  ");
                setcolor(0);
                outtextxy(x+70,kbdy+10,kbuff2);
                flag2=1;
                saveflag=0;
            }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else
{
/* Other key press */
if (kbindex2!=charecter)
{
kbuff2[kbindex2]=ascii;
ascii=0x5f;
kbuff2[kbindex2+1]=ascii;
kbindex2++;
setcolor(7);
outtextxy(x+70,kbdy+10,"");
setcolor(0);
outtextxy(x+70,kbdy+10,kbuff2);
flag2=1;
saveflag=0;
}
if(kbindex2 == (charecter-1))
{
endline = 1;
if (kbdy!=244)
{
point++;
kbdy+=10;
break;
}
break;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
/* Save key buffer to input buffer */
if (flag2!=0)
{
    kbd2=0;
    kbuff2[kbindex2]=kbd2;
    for (kbd2=0;kbd2 < charecter;kbd2++,msg++)
        *msg=kbuff2[kbd2];
    msg=charecter;
    setcolor(7);
    outtextxy(x+70,kbdy," ");
    setcolor(0);
    outtextxy(x+70,kbdy,msg);
}
if(point==6)
    kbdy+=15;
} /* End loop function */
/*****/
/* Menu save data file */
/*****/
void menusave()
{
    int x,y,a,b;
    char *msg;

    index=idx;
    /* Put menu to screen */
    image1=malloc(imagesize(150,maxy-80,maxx-150,maxy-30));

```

```

getimage(150,maxy-80,maxx-150,maxy-30,image1);
a=150+10,b=maxy-80+10;
board(150,maxy-80,maxx-150,maxy-30,7);
setcolor(4);
rectangle(4+150,maxy-80+4,maxx-150-4,maxy-30-4);
rectangle(6+150,maxy-80+6,maxx-150-6,maxy-30-6);
outtextxy(a,b,"          SAVE FILE NAME");
outtextxy(a,b+10," F_INDEX:");
outtextxy(a,b+20," F_DATD:");
x=240,y=279;
point=6;
setcolor(0);
gprintf(x,y+10,"%s->",grade[6]);
gprintf(x,y+20,"%s->",grade[7]);
x=240,y=279;
kb_en=0;
kb_data=0;
kbesc=0;
/* Read keyboard */
for (;;)
{
    readkbd();
    msg=grade[point];
    if(kb_en == 0)
        kbenter(x,y,19,msg,1);
    if(point==7)
    {
        for(m=0;m<12;m++)
            mdata.grade6[m]=*msg++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    /* Esc key press -> break */
    if(kbesc==1 && kb_en==1)
        break;

    /* Enter key press */
    if (kb_data==1 && kb_en == 1)
    {
        savedata();
        break;
    }
} /* End key loop */

/* Restore screen */
grade[6]="TXT.IXD
grade[7]="TXT.DAT";
putimage(150,maxy-80,image1,0);
free(image1);
}

/*****
/* Recive key to kbuffer */
*****/

void kbenter(int x,int y,int charecter,char *msg,int kbb)
{
    int kbd1,kbindex,kb;
    char flag=0;

    /* Clear key buffer */
    flag=0;
    setcolor(1);
    ascii=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(kbindex=0;kbindex < 20;kbindex++)
    kbuff[kbindex]=ascii;
/* Display on x,y position */
    setcolor(7);
    outtextxy(x+70,y+10,"");
    setcolor(0);
    outtextxy(x+70,y+10,"_");
    kbindex=0;
/* Wait for key press */
for (;;)
{
    scancode=ascii=getch();
    if (scancode==84)
        break;
    if (scancode==27){ /* esc */
        kbesc=1; kb_data=1;
        goto Exit;
    }
    if (scancode==34)
        break;
    if (scancode==13) /* enter */
    {
        y+=10; kb=1; point++;
        break;
    }
    if (ascii!=0)
    {
        /* Back space key press */

```

```

        if (scancode==32)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    flag=1;
    saveflag=0;
}
}
}

/* Save key buffer to input buffer */
if (flag!=0)
{
    kbd1=0;
    kbbuff[kbindex]=kbd1;
    for (kbd1=0;kbd1 < character;kbd1++,msg++)
        *msg=kbbuff[kbd1];
    msg-=character;
    setcolor(7);
    outtextxy(x+70,y,"");
    setcolor(0);
    outtextxy(x+70,y,msg);
}

/***** F_DATA *****/
if(kbb == 1)
{
    msg=grade[point];
    flag=0;
    setcolor(1);
    ascii=0;
    for(kbindex=0;kbindex < 20;kbindex++)
        kbbuff1[kbindex]=ascii;

    /* Display on x,y position */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setcolor(7);
        outtextxy(x+70,y+10,"");
        setcolor(0);
        outtextxy(x+70,y+10,"_");
        kbindex=0;
/* Wait for key press */
for (;;)
{
    scancode=ascii=getch();
    if (scancode==84)
        break;
    if (scancode==27)
    {
        kbesc=1; kb_data=1;
        goto Exit;
    }
    if (scancode==34)
        break;
    if (scancode==13 && kb == 0)
    {
        y+=10;
        break;
    }
    if (scancode==13)
        break;
    if (ascii!=Q)
    {
/* Back space key press */
        if (scancode==32)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if (kbindex!=0)
    {
        kbindex--;
        ascii=0x5f;
        kbuff1[kbindex]=ascii;
        ascii=0x20;
        kbuff1[kbindex+1]=ascii;
        setcolor(7);
        outtextxy(x+70,y+10,"");
        setcolor(0);
        outtextxy(x+70,y+10,kbuff1);
        flag=1;
        saveflag=0;
    }
}
else
{
    /* Other key press */
    if (kbindex!=charecter)
    {
        kbuff1[kbindex]=ascii;
        ascii=0x5f;
        kbuff1[kbindex+1]=ascii;
        kbindex++;
        setcolor(7);
        outtextxy(x+70,y+10,"");
        setcolor(0);
        outtextxy(x+70,y+10,kbuff1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        flag=1;
        saveflag=0;
    }
}
}

/* Save key buffer to input buffer */
if (flag!=0)
{
    kbd1=0;
    kbbuff1[kbindex]=kbd1;
    for (kbd1=0;kbd1 < character;kbd1++,msg++)
        *msg=kbbuff1[kbd1];
    msg=character;
    setcolor(7);
    outtextxy(x+70,y+10,"");
    setcolor(0);
    outtextxy(x+70,y+10,msg);
} /* end for */
} /* end if-kbb */

if(point==7)
    kb_data=1;

Exit:
    kb_en=1;
},
/*****/
/* Save data from buffer to hard disk */
/*****/

void savedata(void) /* MUST EDIT */

```

```

{
    FILE *fp;
    int x,y;

    /* Save index data */
    if((fp=fopen(grade[7],"wb")) != NULL)
    {
        fwrite(&mdata,sizeof(struct atod),1,fp);
        fclose(fp);
    } /* end loop save file */
    /* Set page display & save it to buffer */
    if(fp==NULL)
    {
        putimage(150,maxy-80,image1,0);
        board(180,maxy-80,maxx-180,maxy-30,7);
        setcolor(0);
        rectangle(4+180,maxy-80+4,maxx-180-4,maxy-30-4);
        rectangle(6+180,maxy-80+6,maxx-180-6,maxy-30-6);
        setcolor(4);
        x=240,y=279;
        outtextxy(x-5,y+5," WRITE TO DISK ERROR.");
        outtextxy(x-5,y+15," ANY KEY TO CONTINUE. ");
        getch();
        putimage(150,maxy-80,image1,0);
        setcolor(1);
        bioskey(0);
    } /* end loop open file error */
}

```

```

/*****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Load data to display on screen */
/*****/
void loaddata(void)
{
    FILE *fp;
    int x,y,i;
    char *lmsg;

    board(0,maxy-24,maxx,maxy,7); /* Print active mode */
    setcolor(4);
    outtextxy(0,maxy-15," LOAD: F_INDEX F_DATA Esc
");
    setcolor(0);
    outtextxy(0,maxy-15," In mode you can enter to search & display
to EXIT.");
    /* Clear memory befor load */
    image1=malloc(imagesize(150,maxy-80,maxx-150,maxy-30));
    getimage(150,maxy-80,maxx-150,maxy-30,image1);

    board(150,maxy-80,maxx-150,maxy-30,7);
    setcolor(0);
    rectangle(4+150,maxy-80+4,maxx-150-4,maxy-30-4);
    rectangle(6+150,maxy-80+6,maxx-150-6,maxy-30-6);
    setcolor(4);
    x=166,y=279;
    outtextxy(x,y," LOAD INDEX FILENAME.");
    outtextxy(x,y+10,"F_DATA.");
    setcolor(0);
    point=7;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

kb_en=0;
lmsg=kload;
if(kb_en == 0)
    kbenter(x,y,19,lmsg,0);
/* Load index pointer */
if((fp=fopen(kload,"rb")) == NULL)
{
    putimage(150,maxy-80,image1,0);
    board(180,maxy-80,maxx-180,maxy-30,7);
    setcolor(0);
    rectangle(4+180,maxy-80+4,maxx-180-4,maxy-30-4);
    rectangle(6+180,maxy-80+6,maxx-180-6,maxy-30-6);
    setcolor(4);
    x=188,y=maxy-80+10;
    outtextxy(x-5,y+5," OPEN FILE TO DISK ERROR.");
    outtextxy(x-5,y+15," ANY KEY TO CONTINUE. ");
    setcolor(1);
    getch();
    goto LABEL;
}
/* If Esc key press to break */
if (scancode==27)
{
    setfillstyle(1,8);
    bar(150,maxy-80,maxx-150,maxy-30);
    board(0,maxy-24;maxx,maxy,7);
    setcolor(1);
    outtextxy(0,maxy-15," F1: F2: F3: F4: F5: F6: F7: F8: F9:
F10: ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(15);
outtextxy(0,maxy-15," P/B F.F R.W STOP SAVE LOAD PRINT
DATA NOTE QUIT ");
setcolor(1);
/* break; */
} /* end load key esc */
/* If enter key press to load from disk & display */
if(fp!=NULL)
{
rewind(fp);

if(fread(&mdata,sizeof(struct atod),1,fp));

putimage(150,maxy-80,image1,0);
board(150,maxy-80,maxx-150,maxy-30,7);
setcolor(0);
rectangle(4+150,maxy-80+4,maxx-150-4,maxy-30-4);
rectangle(6+150,maxy-80+6,maxx-150-6,maxy-30-6);
setcolor(11);
x=198,y=maxy-70;
outtextxy(x-5,y+5," OPEN FILE TO DISK OK.");
outtextxy(x-5,y+15," ANY KEY TO CONTINUE. ");
setcolor(1);

putimage(150,maxy-80,image1,0);

ldscr();

goto LABEL1;

/* Fill data to display screen */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

TABLE:

```
putimage(150,maxy-80,image1,0);
free(image1);
```

TABLE1:

```
fclose(fp);
board(0,maxy-24,maxx,maxy,7);
setcolor(1);
outtextxy(0,maxy-15," F1: F2: F3: F4: F5: F6: F7: F8: F9:
F10: ");
setcolor(15);
outtextxy(0,maxy-15," P/B F.F R.W STOP SAVE LOAD PRINT
DATA NOTE QUIT ");
setcolor(1);
for(i=0;i<20;i++)
    kload[i]=0;

} /* Out of function */
/*****
/* Load data from disk to memory */
*****/
int ldscr()
{
    int pointr,x,y,mc,d;
    char *gr;

    /* pointc=pointc*36; */
    index=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Load note */
    setcolor(8);

    outtextxy(110,200,"");

    setcolor(11);

    outtextxy(110,200,mdata.dnote);

/* Display grade & box menu */

    gr=grade[0];
    for (pointr = 0 ; pointr < 5 ; pointr++)
        *gr++=mdata.grade0[pointr];
    gr=grade[1];
    for (pointr = 0 ; pointr < 5 ; pointr++)
        *gr++=mdata.grade1[pointr];
    gr=grade[2];
    for (pointr = 0 ; pointr < 5 ; pointr++)
        *gr++=mdata.grade2[pointr];
    gr=grade[3];
    for (pointr = 0 ; pointr < 5 ; pointr++)
        *gr++=mdata.grade3[pointr];
    gr=grade[4];
    for (pointr = 0 ; pointr < 5 ; pointr++)
        *gr++=mdata.grade4[pointr];
    gr=grade[5];
    for (pointr = 0 ; pointr < 5 ; pointr++)
        *gr++=mdata.grade5[pointr];
    gr=grade[6];
    for (pointr = 0 ; pointr < 12 ; pointr++)
        *gr++=mdata.grade6[pointr];

```

```
board(0,maxy-24,maxx,maxy,7);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(4);

outtextxy(0,maxy-15," Esc      Enter");

setcolor(0);

outtextxy(0,maxy-15," to EXIT.    to continue.");

setfillstyle(1,8);

bar(7,33,maxx-5,43);

setcolor(15);

d=15;

for (mc=8;mc > 0;mc-)
{
rectangle(d,33,d+24,43);
d+=40;
}

d=335;

for (mc=6;mc > 0;mc-)
{
rectangle(d,33,d+40,43);
d+=50;
}

x=338,y=35;

setcolor(11);

printf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],
grade[3],grade[4],grade[5]);

x=20,y=35;

printf(x,y,"%2d %2d %2d %2d %2d %2d %2d %2d",mdata.sh[0],
mdata.sh[1],mdata.sh[2],

mdata.sh[3],mdata.sh[4],mdata.sh[5],mdata.sh[6],mdata.sh[7]);

/* Display signal */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x=40,y=168; Gimage(89); setcolor(11);
lplot(x,y);
x=40,y=290; Gimage(211); setcolor(11);
lplott(x,y);
/* Print mode active */
board(0,maxy-24,maxx,maxy,7);
.setcolor(4);
outtextxy(0,maxy-15," Esc      Enter ");
setcolor(0);
outtextxy(0,maxy-15," to EXIT. to contine. ");
/* Display page^1 */
page^=1;
setactivepage (page);
board(0,maxy-24,maxx,maxy,7);
.setcolor(4);
outtextxy(0,maxy-15," Esc      Enter");
setcolor(0);
outtextxy(0,maxy-15," to EXIT. to contine. ");

setfillstyle(1,8);
bar(7,33,maxx-5,43);
setcolor(15);
d=15;
for (mc=8;mc > 0;mc-)
{
rectangle(d,33,d+24,43);

```

```
d+=40;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    d=335;
    for (mc=6;mc > 0;mc--)
    {
        rectangle(d,33,d+40,43);
        d+=50;
    }

    x=338,y=35;
    setcolor(11);
    gprintf(x,y,"%4s %4s %5s %4s %4s %4s",grade[0],grade[1],grade[2],
        grade[3],grade[4],grade[5]);
    x=20,y=35;
    gprintf(x,y,"%2d %2d %2d %2d %2d %2d %2d %2d",mdata.sh[0],
mdata.sh[1],mdata.sh[2],
mdata.sh[3],mdata.sh[4],mdata.sh[5],mdata.sh[6],mdata.sh[7]);
/* Display signal */
    x=40,y=168; Gimage(89); setcolor(11);
    lplot(x,y);
    x=40,y=290; Gimage(211); setcolor(11);
    lplot(x,y);
/* Print mode active */
    board(0,maxy-24,maxx,maxy,7);
    setcolor(4);
    outtextxy(0,maxy-15," Esc      Enter ");
    setcolor(0);
    outtextxy(0,maxy-15," to EXIT.    to continue. ");

    page^=1;
    setactivepage (page);

```

```

}
/*****/
/*      Plot data to load      */
/*****/
lplot(int x,int y)
{
    int x1=40,y1;

    for(i=0 ;i < 560;i++)
    {
        index_y1=mdata.memseg1[i];
        y1=(168-index_y1);
        x1++;
        if(y1 > 185)
            y1 =185;
        if(y1 < 90)
            y1 =90;
        line(x,y,x1,y1);
        y=y1;
        x=x1;
    }/* for */
}

lplott(int x,int y)
{
    int x1=40,y1;

    for(i=0 ;i < 560;i++)
    {
        index_y1=mdata.memseg2[i];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y1=(290-index_y1);
x1++;
if(y1 > 307)
    y1 =307;
if(y1 < 211)
    y1 =212;
line(x,y,x1,y1);
y=y1;
x=x1;
} /* for */
}
/*****/
/* Set interrupt vector */
/*****/
void setint(void)
{
    oldfunc=getvect(8);
    setvect(8,int8);
    /* Set timer to fast trick */
    outportb(0x43,0x36);
    outportb(0x40,0x0);
    outportb(0x40,0x40);
}
/*****/
/* Restore interrupt vector */
/*****/
void restore(void)
{
    setvect(8,oldfunc);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Set timer to normal trick */
outportb(0x43,0x36);
outportb(0x40,0);
outportb(0x40,0);
outportb(0x700,0); /* reset Muxx */
}

/*****/
/* Interrupt service routine to read data from A/D */
/*****/
void interrupt int8(void)
{
    unsigned char a;
    static int oldcount=0;
    char m;
    for(;;)
    {
        outportb(0x702,0);
        a=inportb(0x702);
        a&=0x10;
        if (a!=0)
            break;
    }
    vdata[idx]=inportb(0x701);
    mux[idx]=vdata[idx];
    /* Multiplex chanel */
    m=inportb(0x700);
    m=m&7;
    if(m==7)

```

```

{
    m=inportb(0x700);
    m=m&0xf8;
    outportb(0x700,m);
}
else
{
    m=inportb(0x700);
    m++;
    outportb(0x700,m);
}
outportb(0x702,0x0f);
idx++;
if (idx==2400)
    idx=0;

if (oldcount==3)
{ (*oldfunc());
  oldcount=0; }
else
{ oldcount++;
  outportb(0x20,0x20); }
}
/******
/*          Printer          */
/******

void printer()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
char RET=0x0d,
    LF=0x0a;

char m;

int i,j,k,Msb,Lsb,MaxX=getmaxx(),MaxY=getmaxy(),x,y;

setactivepage (page);
page^=1;
setactivepage (page);

cleardevice();

fprintf(stdprn,"===== %
c%c",RET,LF);
fprintf(stdprn,"%c%c DATA Printer %c%c%c%c",RET,LF,RET,LF,RET,LF);
fprintf(stdprn,"===== %
c%c%c%c",RET,LF,RET,LF);
fprintf(stdprn,"CH1 %c%c",RET,LF);
fprintf(stdprn,"sh1= %2d ,sh2= %2d ,sh3= %2d ,sh4= %2d %c%c",mdata.sh[0],
    mdata.sh[1],mdata.sh[2],mdata.sh[3],RET,LF);
fprintf(stdprn,"CH2 %c%c",RET,LF);
fprintf(stdprn,"sh5= %2d ,sh6= %2d ,sh7= %2d ,sh8= %2d %c%c",mdata.sh[4],
    mdata.sh[5],mdata.sh[6],mdata.sh[7],RET,LF);
fprintf(stdprn,"GRADE= %4s ,#FT= %4s ,WALL= %5s ,OD %4s ,P_NO %4s ,P/B= %4s
%c%c%c%c",grade[0],
    grade[1],grade[2],grade[3],grade[4],grade[5],RET,LF,RET,LF);
fprintf(stdprn,"NOTE: ");
fprintf(stdprn,"%20s %c%c",mdata.dnote,RET,LF);

setcolor(15);

.outtextxy(10,65,"AMPLITUDE");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outtextxy(maxx-100,193,"LONG");
    outtextxy(25,161,"0");
    outtextxy(20,89,"99");
    outtextxy(20,125,"50");
    line(35,168,39,168);
    line(35,129,39,129);
    line(35,89,39,89);
    line(39,89,39,89+96);
    line(39,89+96,maxx-39,89+96);
    outtextxy(10,193,"AMPLITUDE");
    outtextxy(maxx-100,320,"LONG");
    outtextxy(25,284,"0");
    outtextxy(20,212,"99");
    outtextxy(20,249,"50");
    line(35,290,39,290);
    line(35,251,39,251);
    line(35,211,39,211);
    line(39,211,39,211+96);
    line(39,211+96,maxx-39,211+96);
    x=40,y=168;    setcolor(11);
    lplot(x,y);
    x=40,y=290;    setcolor(11);
    lplott(x,y);

setviewport(0,0,maxx,maxy,0);
setfillstyle(1,BLACK);
fprintf(stdprn,"\x1B\x30");
Lsb = MaxX & 0x00FF;
Msb = MaxX >> 8;

for(j=0; j<=MaxY/8; j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    fprintf(stdprn, "\x1B*%c%c%c", 6, Lsb, Msb);
    for(i=0; i<=MaxX; i++)
    {
        m = 0;
        for(k=0; k<8; k++)
        {
            m<=1;
            if(getpixel(i,j*8+k)) m++;
        }
        fprintf(stdprn, "%c", m);
    }
    fprintf(stdprn, "\x0D\x0A");
}
fprintf(stdprn, "\f");
page^=1;
setactivepage (page);
}

```

/*scanner.prj*/

scanner.c

scanxms.h

scanply.h



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#if !defined( __PLAY_C)
#define __PLAY_C
#define _max 92
#include <dos.h>
#include <bios.h>
#include <time.h>
/*scanply.h*/
enum note{ _C0,_CS0,_D0,_DS0,_E0,_F0,_G0,_GS0,_A0,_AS0,_B0,
           _C1,_CS1,_D1,_DS1,_E1,_F1,_FS1,_G1,_GS1,_A1,_AS1,_B1,
           _C2,_CS2,_D2,_DS2,_E2,_F2,_FS2,_G2,_GS2,_A2,_AS2,_B2,
           _C3,_CS3,_D3,_DS3,_E3,_F3,_FS3,_G3,_GS3,_A3,_AS3,_B3,
           _C4,_CS4,_D4,_DS4,_E4,_F4,_FS4,_G4,_GS4,_A4,_AS4,_B4,
           _C5,_CS5,_D5,_DS5,_E5,_F5,_FS5,_G5,_GS5,_A5,_AS5,_B5,
           _C6,_CS6,_D6,_DS6,_E6,_F6,_FS6,_G6,_GS6,_A6,_AS6,_B6};

int note_frq[84]=
{ 65, 69, 73, 78, 82, 87, 93, 98,104,110,116,123,
  131,139,147,156,165,175,185,196,208,220,233,247,
  262,227,294,311,330,349,370,392,415,440,466,494,
  523,554,587,622,659,698,740,784,831,880,932,988,
  1047,1109,1175,1245,1319,1397,1480,1568,1661,1760,1865,1976,
  2093,2217,2349,2489,2637,2794,2960,3136,3322,3520,3729,3951,
  4186,4435,4699,4975,5274,5587,5919,6271,6645,7040,7458,7902};

int s[_max]={_C2,_C2,_E2,_G2,_G2,_G2,_A2,_G2,_F2,_A2,_G2,_E2,
             _E2,_E2,_F2,_E2,_D2,_D2,_D2,_D2,_E2,_D2,_C2,_C2,
             _C2,_C2,_E2,_G2,_G2,_G2,_A2,_G2,_F2,_A2,_G2,_E2,
             _G2,_A2,_B2,_G2,_D3,_A2,_A2,_B2,_A2,_G2,
             _G2,_A2,_G2,_F2,_D2,_F2,_A2,_G2,_E2,_C2,

```

```

_C2,_C2,_E2,_D2,_D2,_A2,_B2,_A2,_G2,
_C2,_C2,_E2,_G2,_G2,_G2,_A2,_B2,_C3,_D3,_E3,
_D3,_C3,_A2,_G2,_A2,_B2,_C3,_E3,_C3,_D3,_C3,_C3,_C3,_C3);

```

```

int d[_max]={3,7,3,10,7,3,7,3,7,3,20,7,
             3,7,3,7,3,12,3,7,3,7,3,17,
             3,7,5,10,7,3,7,3,7,3,20,10,
             7,3,7,3,17,3,5,5,20,10,
             7,3,7,2,20,7,3,7,3,17,
             3,7,5,10,7,3,10,7,3,27,
             3,7,3,10,7,3,5,5,5,27,
             3,7,3,7,3,7,3,7,3,10,7,3,5};

```

```
int plays(int snd[],int dly[],int count);
```

```
int playdelay2(int tenth);
```

```
void playt(void)
```

```
{
```

```
    plays(s,d,_max);
```

```
}
```

```
int plays(int snd[],int dly[],int count)
```

```
{
```

```
    int cnt;
```

```
    for(cnt=0;cnt< count;cnt++)
```

```
    {
```

```
        sound(note_frq[snd[cnt]]);
```

```
        playdelay2(dly[cnt]);
```

```
        if(bioskey(1))
```

```
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        nosound();
        return 0;
    }
}

nosound();
return 1;
}

```

```

int playdelay2(int tenth)
{
    clock_t tck;
    int c;

    tck=clock();
    for(c=0;c<tenth-1;c++)
    {
        while(tck == clock());
        tck = clock();
    }
    nosound();
    while(tck == clock());
    return 1;
}

#endif

```

```

#include <dos.h>
#include <graphics.h>
#include <stdio.h>
/*scanxms.h*/
struct XMSblock
{
    long nbytes;
    int shandle;
    long soffset;
    int dhandle;
    long doffset;
};

struct XMSblock XMS_dat;

void far ((*XMSentry))(void);
void icon(void);
int dfill (int a,int b,int c,int d);
void anykey_bar(void);
void puz_credit(int x,int y);
void board(int x1,int y1,int x2,int y2,int color);
void pull(int x1,int y1,int x2,int y2,int color);
void push(int x1,int y1,int x2,int y2,int color);
void title(void);
void Initialize(void);
void gprintf(int x,int y,char *fmt, ...);
void Gimage(int xx1);

```

```
int XMS_ERR;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char *XMS_E80="Funvtion no implenented\n0";
char *XMS_E81="VDISK was delected\n0";
char *XMS_E82="An A20 error occured\n0";
char *XMS_E8E="A genral driver error\n0";
char *XMS_E8F="Unrecoverable driver error\n0";
char *XMS_E90="HMA does mot exist\n0";
char *XMS_E91="HMA is alrady in use\n0";
char *XMS_E92="DX is less than the /HMAMIN = parameter\n0";
char *XMS_E93="HMA is not allocated\n0";
char *XMS_EA0="All extened monory is allocatted\n0";
char *XMS_EA1="All available extendes memory handles are alloceted\n0";
char *XMS_EA2="Invalid handle\n0";
char *XMS_EA3="Source handle is invalid\n0";
char *XMS_EA4="Source effect is invalid\n0";
char *XMS_EA5="Destination handle is invalid\n0";
char *XMS_EA6="Destination effect is invalid\n0";
char *XMS_EA7="Length is invalid\n0";
char *XMS_EA8="Move has an invalid overlap\n0";
char *XMS_EA9="Parity error occurred\n0";
char *XMS_EAA="Block is not locked\n0";
char *XMS_EAB="Block is not-locked\n0";
char *XMS_EAC="Block lock count overflowed\n0";
char *XMS_EAD="Lock failed\n0";
char *XMS_EB0="Only a smaller UMB is available\n0";
char *XMS_EB1="No UMB's are available\n0";
char *XMS_EB2="UMB segment number is invalid\n0";
char *XMS_E00="No error occured\n0";
char *XMS_EFF="Invalid error code,error code is not known\n0";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int Maxx,Maxy,Midx,Midy;
```

```
char far *XMSerr(void);
```

```
int XMSinstalled(void);
```

```
int XMSleftcon(void);
```

```
int XMSleftall(void);
```

```
int XMSalloc(int Kbyte);
```

```
unsigned long XMSlock(int handle);
```

```
int XMSunlock(int handle);
```

```
int XMSfree(int handle);
```

```
int XMSrealloc(int handle,int newsize);
```

```
int XMStransfer(struct XMSblock *XM);
```

```
int XMSread(void far *str,int handle,long n);
```

```
int XMSwrite(void far *str,int handle,long n);
```

```
int XMSreadoff(void far *str,int handle,long n,long offset);
```

```
int XMSwriteoff(void far *str,int handle,long n,long offset);
```

```
char far *XMSerr(void)
```

```
{
```

```
    switch(XMS_ERR)
```

```
    {
```

```
        case 0x00: return XMS_E00;
```

```
        case 0x80: return XMS_E80;
```

```
        case 0x81: return XMS_E81;
```

```
        case 0x82: return XMS_E82;
```

```
        case 0x8e: return XMS_E8E;
```

```
        case 0x8f: return XMS_E8F;
```

```
        case 0x90: return XMS_E90;
```

```
        case 0x91: return XMS_E91;
```

```

case 0x92: return XMS_E92;
case 0x93: return XMS_E93;
case 0xa0: return XMS_EA0;
case 0xa1: return XMS_EA1;
case 0xa2: return XMS_EA2;
case 0xa3: return XMS_EA3;
case 0xa4: return XMS_EA4;
case 0xa5: return XMS_EA5;
case 0xa6: return XMS_EA6;
case 0xa7: return XMS_EA7;
case 0xa8: return XMS_EA8;
case 0xa9: return XMS_EA9;
case 0xaa: return XMS_EAA;
case 0xab: return XMS_EAB;
case 0xac: return XMS_EAC;
case 0xad: return XMS_EAD;
case 0xb0: return XMS_EB0;
case 0xb1: return XMS_EB1;
case 0xb2: return XMS_EB2;
}
return XMS_EFF;
}

```

```
int XMSinstalled(void)
```

```
{
```

```
void *int2f;
```

```
int2f=(void *)getvect(0x2f);
```

```
if(((unsigned long)int2f) == 0x00000000UL || (*(char far *)int2f)=='\xcf')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return 0;

_AX=0x4300;
geninterrupt(0x2f);
if(_AL!=0x80)
return 0;
else
{
_AX=0x4310;
geninterrupt(0x2f);
XMSentry=(void far(*)())MK_FP(_ES,_BX);
}
XMS_ERR=0;
return 1;
}

int XMSleftcon(void)
{
register int temp;
_AH=0x08;
(*XMSentry)();
temp=_AX;
XMS_ERR=_BL;
return temp;
}

int XMSleftall(void)
{

```

```

_AH=0x08;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(*XMSentry)();
XMS_ERR=_BL;
return _DX;
}

```

```
int XMSalloc(int Kbyte)
```

```

{
    _DX=Kbyte;
    _AH=0x09;
    (*XMSentry)();
    if(_AX==0)
    {
        XMS_ERR=_BL;
        return 0;
    }
    XMS_ERR=0;
    return _DX;
}

```

```
unsigned long XMSlock(int handle)
```

```

{
    _DX=handle;
    _AH=0x0c;
    (*XMSentry)();
    if(_AH==0)
    {
        XMS_ERR=_BL;
        return 0UL;
    }
}

```

```

return(unsigned long)MK_FP(_DX,_BX);
}

```

```

int XMSunlock(int handle)

```

```

{
    register int temp;

```

```

    _DX=handle;

```

```

    _AH=0x0d;

```

```

    (*XMSentry)();

```

```

    temp=_AX;

```

```

    XMS_ERR=_BL;

```

```

    return temp;

```

```

}

```

```

int XMSfree(int handle)

```

```

{

```

```

    register int temp;

```

```

    _DX=handle;

```

```

    _AH=0x0a;

```

```

    (*XMSentry)();

```

```

    temp=_AX;

```

```

    XMS_ERR=_BL;

```

```

    return temp;

```

```

}

```

```

int XMSrealloc(int handle,int newsize)

```

```

{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

register int temp;
_BX=newsize;
_DX=handle;
_AH=0x0f;
(*XMSentry)();
temp=_AX;
XMS_ERR=_BL;
return temp;
}

```

```
int XMStansfer(struct XMSblock *XM)
```

```

{
register int temp;

_DS=FP_SEG(XM);
_SI=FP_OFF(XM);
_AH=0x0b;
(*XMSentry)();
temp=_AX;
XMS_ERR=_BL;
return temp;
}

```

```
int XMSread(void far *str,int handle,long n)
```

```

{
XMS_dat.nbytes=(unsigned long)((n>>1)<<1);
XMS_dat.shandle=handle;
XMS_dat.soffset=(unsigned long)0;
XMS_dat.dhandle=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XMS_dat.doffset=(unsigned long)str;
return XMStransfer(&XMS_dat);
}

```

```

int XMSwrite(void far *str,int handle,long n)
{
XMS_dat.nbytes=(unsigned long)((n>>1)<<1);
XMS_dat.shandle=0;
XMS_dat.soffset=(unsigned long)str;
XMS_dat.dhandle=handle;
XMS_dat.doffset=(unsigned long)0;
return XMStransfer(&XMS_dat);
}

```

```

int XMSreadoff(void far *str,int handle, long n, long offset)
{
XMS_dat.nbytes=(unsigned long)((n>>1)<<1);
XMS_dat.shandle=handle;
XMS_dat.soffset=(unsigned long)offset;
XMS_dat.dhandle=0;
XMS_dat.doffset=(unsigned long)str;
return XMStransfer(&XMS_dat);
}

```

```

int XMSwriteoff(void far *str,int handle, long n, long offset)
{
XMS_dat.nbytes=(unsigned long)((n>>1)<<1);
XMS_dat.shandle=0;
XMS_dat.soffset=(unsigned long)str;

```

```

XMS_dat.dhandle=handle;
XMS_dat.doffset=(unsigned long)offset;
return XMStransfer(&XMS_dat);
}

```

```

/*****

```

```

/*          Title          */

```

```

*****/

```

```

void title(void)

```

```

{

```

```

    int m;

```

```

    setfillstyle(1,8); bar(0,0,Maxx,Maxy-26);

```

```

    for(m=0;m<Midy-16;m+=8) {

```

```

        board(8,Midy-m-13,106,Midy+m-13,4);

```

```

        board(107,Midy-m-13,207,Midy+m-13,7);

```

```

        board(208,Midy-m-13,Maxx-214,Midy+m-13,1);

```

```

        board(Maxx-213,Midy-m-13,Maxx-107,Midy+m-13,7);

```

```

        board(Maxx-106,Midy-m-13,Maxx-8,Midy+m-13,4);

```

```

    } m = 0;

```

```

    board(8,8,106,Maxy-33,4);

```

```

    board(107,8,207,Maxy-33,7);

```

```

    board(208,8,Maxx-214,Maxy-33,1);

```

```

    board(Maxx-213,8,Maxx-107,Maxy-33,7);

```

```

    board(Maxx-106,8,Maxx-8,Maxy-33,4);

```

```

    board(0,Maxy-24,Maxx,Maxy,7);

```

```

    puz_credit(Midx-160,Midy-112);

```

```

}

```

```

void Initialize(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    int xasp,yasp ;
    int  GraphDriver=3;          /* The Graphics device driver      */
    int  GraphMode=1;           /* The Graphics mode value         */
    int  ErrorCode;             /* Reports any graphics errors     */
    struct palettetype palette; /* Used to read palette info       */

    /* GraphDriver = DETECT; */ /* Request auto-detection          */

    initgraph( &GraphDriver, &GraphMode, "" );
    ErrorCode = graphresult(); /* Read result of initialization */
    if( ErrorCode != grOk )
    {
        /* Error occurred during init */
        printf(" Graphics System Error: %s\n", grapherrormsg( ErrorCode ) );
        exit( 1 );
    }
    getpalette( &palette ); /* Read the palette from board */
    getaspectratio( &xasp, &yasp ); /* read the hardware aspect */
    Maxx = getmaxx();
    Maxy = getmaxy();
    Midx = Maxx/2;
    Midy = Maxy/2;
}

void push(int x1,int y1,int x2,int y2,int color)
{
    setcolor(0);
    line(x1,y1,x2,y1); line(x1,y1+1,x2-1,y1+1);
    line(x1,y1,x1,y2); line(x1+1,y1,x1+1,y2);
    setcolor(color);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(x1+1,y2,x2,y2); line(x1+2,y2-1,x2,y2-1);
line(x2,y1,x2,y2); line(x2-1,y1+1,x2-1,y2);
}

```

```
void pull(int x1,int y1,int x2,int y2,int color)
```

```

{
setcolor(color);
line(x1,y1,x2,y1); line(x1,y1+1,x2-1,y1+1);
line(x1,y1,x1,y2); line(x1+1,y1,x1+1,y2-1);
setcolor(0);
line(x1+2,y2,x2,y2); line(x1+3,y2-1,x2,y2-1);
line(x2,y1,x2,y2); line(x2-1,y1+1,x2-1,y2);
setcolor(15);
}

```

```
void board(int x1,int y1,int x2,int y2,int color)
```

```

{
setfillstyle(1,color);
bar(x1+2,y1+2,x2-2,y2-2);
pull(x1,y1,x2,y2,color+8);
}

```

```
void puz_credit(int x,int y)
```

```

{
int m; ..

```

```
char name_str[9][40] = {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

" IMAGE SCANNER & RECORD SIGNAL",
" FORM A/D ",
" Version 1.3 ",
" by ",
" MR. KRIENGSAK LUANGUMPOL ",
" MR. PASKON ANUNTAPAN ",
" MR. PREEDA KALTIYANANT ",
" MR. TANATIP KHAMCHAN ",
" ",
" ",
),
rit_str[2][40] = {
" King Mongkut Ladkabang ",
" engineer - education 15 ",
};
board(x,y,x+311,y+23,2);
setcolor(15);
gprintf(x+136,y+9,"About");
board(x,y+24,x+311,y+212,7);
setcolor(0);
for(m=0;m<9;m++) {
if(m<3) setcolor(4);
if(m>3) setcolor(1);
if(m==8)setcolor(0);
gprintf(x+8,y+m*16+40,"%s",name_str(m));
}
for(m=0;m<2;m++)
gprintf(x+8,y+m*16+171,"%s",rit_str[m]);

```

anykey_bar();

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void anykey_bar(void)

{
    board(0,Maxy-24,Maxx,Maxy,7);
    setcolor(128);
    outtextxy(16,Maxy-16,"Press any key..");
}

/*****
/*      draw & fill pattern      */
/*      a=left b=top c=right d=bottom e=count      */
/*****
int dfill (int a,int b,int c,int d)
{
    do
    {
        setlinestyle(USERBIT_LINE,0x5555,NORM_WIDTH);
        line(a,b,c,b);
        b++;
        setlinestyle(USERBIT_LINE,0xaaaa,NORM_WIDTH);
        line(a,b,c,b);
        b++;
    } while (b<=d);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void gprintf(int x,int y,char *fmt, ...)
{
    va_list ap;
    char text[128];
    va_start(ap,fmt);
    vsprintf(text,fmt,ap);
    outtextxy(x,y,text);
    va_end(ap);
}

/*****
/*      Draw main menu      */
*****/

void icon(void)
{
    int mc,a,b,c,d;
    unsigned int x,y;
    struct date now1;

    cleardevice();

    setfillstyle(1,8); bar(0,0,Maxx,Maxy-26);
    board(0,0,100,23,7);
    board(100,0,getmaxx(),23,7);

    settxtstyle(DEFAULT_FONT,HORIZ_DIR,2);
    setcolor(11); outtextxy(14,5,"KMITL");
    setcolor(0); outtextxy(12,5,"KMITL");
    setcolor(1); outtextxy(13,5,"KMITL");

    settxtstyle(DEFAULT_FONT,HORIZ_DIR,2);
    setcolor(12); outtextxy(144,5," ENGINEER EDUCATION VER.1");

```

```

setcolor(0); outtextxy(142,5," ENGINEER EDUCATION VER 1");
setcolor(4); outtextxy(143,5," ENGINEER EDUCATION VER 1");

setcolor(11);
settextstyle(0,0,0);

setcolor(15);
settextstyle(SMALL_FONT,HORIZ_DIR,2);
outtextxy(0,47," SH1 SH2 SH3 SH4 SH5 SH6 SH7 SH8 GRADE #/FT WALL
OD P_NO P/B");
outtextxy(0,71," CH1 CH2 DATE/TIME:");

moveto(27,62);
lineto(27,73),lineto(75,73);
moveto(105,73);
lineto(148,73),lineto(148,61);
moveto(190,63);
lineto(190,73),lineto(235,73);
moveto(265,73);
lineto(308,73),lineto(308,61);

a=15;
for (mc=8;mc > 0;mc--)
{
rectangle(a,33,a+24,43);
a+=40;
}

a=335;
for (mc=6;mc > 0;mc--)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rectangle(a,33,a+40,43);
    a+=50;
}
/*
    setcolor(0);
    board(0,maxy-24,maxx,maxy;7);
    setcolor(1);
    settxtstyle(SMALL_FONT,HORIZ_DIR,2);
    outtextxy(0,maxy-15," F1:  F2:  F3:  F4:  F5:  F6:  F7:  F8:  F9:
F10: ");
    setcolor(15);
    outtextxy(0,maxy-15," P/B  F.F  R.W  STOP  SAVE  LOAD  PRINT
DATA  NOTE  QUIT ");
*/
    setcolor(14);
    rectangle(3,29,Maxx-3,82);
    /***draw box1 display***/
    setfillstyle(1,0);
    bar(39,89,Maxx-39,89+96);
    rectangle(39,89,Maxx-39,89+96);
        a=Maxx-37,b=89+6,c=Maxx-29,d=89+6+96;
    dfill(a,b,c,d);
        a=49,b=89+96,c=Maxx-38,d=89+6+96;
    dfill(a,b,c,d);
    moveto(Maxx-37,89+6);
    lineto(Maxx-29,89+6);
    lineto(Maxx-29,89+7+96);
    lineto(49,89+7+96);
    lineto(49,89+96);

```

```

/****draw box2 display****/
setfillstyle(1,0);
bar(39,211,Maxx-39,211+96);
rectangle(39,211,Maxx-39,211+96);
a=Maxx-37,b=211+6,c=Maxx-29,d=211+6+96;
dfill(a,b,c,d);
a=49,b=211+96,c=Maxx-38,d=211+96+6;
dfill(a,b,c,d);
moveto(Maxx-37,211+6);
lineto(Maxx-29,211+6);
lineto(Maxx-29,211+7+96);
lineto(49,211+7+96);
lineto(49,211+96);
/*****print maghetude/time*****/
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(8,91,"CH1");
outtextxy(8,215,"CH2");
outtextxy(60,200,"NOTE:");
settextstyle(SMALL_FONT,HORIZ_DIR,4);
rectangle(5,89,33,12+89);
rectangle(5,213,33,213+12);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtextxy(25,105,"AMPLITUDE.");
outtextxy(25,235,"AMPLITUDE.");
/*****draw grid*****/
setlinestyle(USERBIT_LINE,0x111,NORM_WIDTH);
setcolor(15);
mc=128+39;
for (a=5;a > 0;a--)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    line(mc,89,mc,89+96);
    mc+=128;
}

mc=128+39;
for (a=5;a > 0;a--)
{
    line(mc,211,mc,211+96);
    mc+=128;
}

mc=89+24;
for (a=4;a > 0;a--)
{
    line(39,mc,Maxx-40,mc);
    mc+=24;
}

mc=211+24;
for (a=4;a > 0;a--)
{
    line(39,mc,Maxx-40,mc);
    mc+=24;
}

setlinestyle(SOLID_LINE,0,NORM_WIDTH);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

void Gimage(int xx1)
{
    int mc,a;

    setfillstyle(1,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(14);
bar(39,xx1,Maxx-39,xx1+96);
rectangle(39,xx1,Maxx-39,xx1+96);
setlinestyle(USERBIT_LINE,0x1111,NORM_WIDTH);
setcolor(15);
    mc=128+39;
for (a=5;a > 0;a--)
{
    line(mc,xx1,mc,xx1+96);
    mc+=128;
}
    mc=xx1+24;
for (a=4;a > 0;a--)
{
    line(39,mc,Maxx-40,mc);
    mc+=24;
}
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

```

บรรณานุกรม

ดุสิต เครื่องงาม, โซลิตเสตทฟิสิกส์, กรุงเทพมหานคร : เอช-เอน, 2535.

ธานินทร์ ถาวรศาสนวงศ์ และทินกร ดูก, การอินเทอร์เฟส IBM PC, กรุงเทพมหานคร :
ฟิสิกส์เซนเตอร์, 2530.

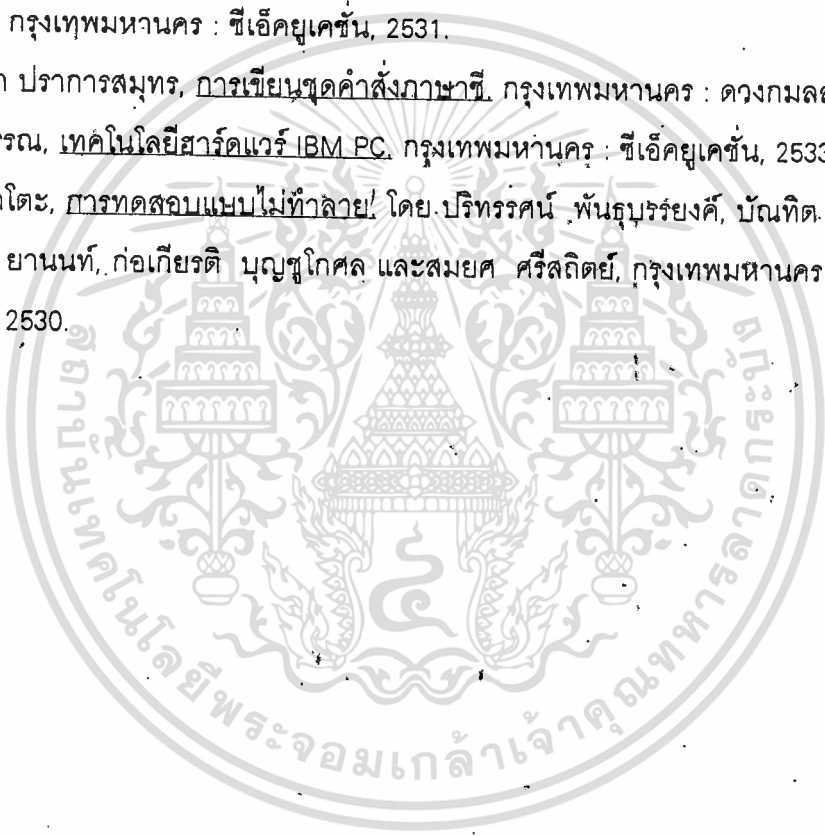
นรินทร์ เนาวประทีป, การใช้งาน Z80, กรุงเทพมหานคร : ฟิสิกส์เซนเตอร์, ม.ป.ป.

บุญเลิศ เขียมทัศนากา, ยืน ภูววรรณ และสมนึก ศิริโต, โปรแกรมคอมพิวเตอร์ภาษาซี,
กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2531.

มณฑนา ปราการสมุทร, การเขียนชุดคำสั่งภาษาซี, กรุงเทพมหานคร : ดวงกมลสมัย, 2530.

ยืน ภูววรรณ, เทคโนโลยีฮาร์ดแวร์ IBM PC, กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2533.

ฟูจิอิ ซาโตะ, การทดสอบแบบไม่ทำลาย โดย.ปริทรรคนันท์ พันธบุรุษย์, บัณฑิต. ภาควิชา
ยานยนต์, ก่อเกียรติ บุญชูโกศล และสมยศ ศรีสถิตย์, กรุงเทพมหานคร : เอเชียเพรส,
2530.



บรรณานุกรม

- ดุสิต เครื่องงาม, โซลิตเสตทฟิลิกส์, กรุงเทพมหานคร : เอช-เอน, 2535.
- ธานีฤทธิ์ ถาวรศาสนวงศ์ และทินกร ดึก, การอินเตอร์เฟส IBM PC, กรุงเทพมหานคร : ฟิลิกส์เซนเตอร์, 2530.
- นรินทร์ เนาวประทีป, การใช้งาน Z80, กรุงเทพมหานคร : ฟิลิกส์เซนเตอร์, ม.ป.ป.
- บุญเลิศ เขียวมัทศนา, ยืน ภู่วรรณ และสมนึก ศิริโต, โปรแกรมคอมพิวเตอร์ภาษาซี, กรุงเทพมหานคร : ซีเอ็ดดูเคชั่น, 2531.
- มัทธนา ปราการสมุทร, การเขียนชุดคำสั่งภาษาซี, กรุงเทพมหานคร : ดวงกมลสมัย, 2530.
- ยืน ภู่วรรณ, เทคโนโลยีฮาร์ดแวร์ IBM PC, กรุงเทพมหานคร : ซีเอ็ดดูเคชั่น, 2533.
- ฟูจิอิ ซาโตะ, การทดสอบแบบไม่ทำลาย, โดย บริษัท สันติบรรยงค์, บัณฑิต โรจน์อารยานนท์, ก่อเกียรติ บุญชูโกศล และสมยศ ศรีสถิตย์, กรุงเทพมหานคร : เอเชียเพรส, 2530.
- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Printice-Hall, inc., 1983.
- Donna Mosich, Namir Shammas and Bryan Flamig, Advanced Turbo C Programmer's Guide, John Wiley & Sons, Inc., 1988.
- Robert C. Hutchison and Strven B. Just, Programming Using The C Language, Mc Graw-Hill, Inc., 1988.
- Stephen G. Kochan, Programming In c, Berkeley, California : Hayden Book Company, 1983.
- Jack Purdum, C Programming, Que Microcomputer Products, 1983.