ปริญญานิพนธ์

ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

AUTOMATIC SPEECH FOR EXAMINATION RESULT REPORTING

นายพงศ์เดช ฉรงค์ฉุทธิ์

นายยุทธนา สรวลสรรค์

นายวารินทร์ ชุมสำโรง

นายสุริยา อุปถัมภ์

A021057

เลขหมู่ .....

เลขทะเบียน 1289 021057

วัน เดือน ปี -3 พ.ย. 2538

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อปริญญานิพนธ์ ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

AUTOMATIC SPEECH FOR EXAMINATION RESULT REPORTING

ชื่อนักศึกษา

1.นายพงศ์เดช ฉรงค์ฤทธิ์

2.นายยุทธนา สรวลสรรค์

3.นายวารินทร์ ชุมสำโรง

4.นายสุริยา อุปถัมภ์

อาจารย์ผู้ควบคุมปริญญานิพนธ์

1.อาจารย์กิติพงค์ มะโน

2.อาจารย์วรวิทย์ สมหา

3.อาจารย์สันติ ตันตระกูล

คณะกรรมการสอบปริญญานิพนธ์	ลายมือ
อาจารย์กิติพงค์ มะโน	
อาจารย์สันติ ตันตระกูล	
อาจารย์สมชาย หมื่นสายญาติ	
อาจารย์โกศล ตราชู	
อาจารย์ปิยะ ธรรมมาจิตภิรมย์	

วัน/เดือน/ปี ที่สอบ วันที่ 25 เดือนธันวาคม พ.ศ.2537 เวลา 18.00 น.ถึง 19.00 น.

สถานที่สอบ ห้อง ค.301 คณะครุศาสตร์อุตสาหกรรม

ลงชื่อ.....

(ผศ.ศิริ ธีระพันธ์ปิยสิริรัตน์ ณ อยุธยา)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

AUTOMATIC SPEECH FOR EXAMINATION RESULT REPORTING

ผู้จัดทำ

1.นายพงศ์เดช ฉรงค์ฤทธิ์

2.นายยุทธนา สรวลสรรค์

3.นายวารินทร์ ชุมสำโรง

4.นายสุริยา อุปถัมภ์

อาจารย์ที่ปรึกษา

ลงนาม .....

(อาจารย์กิติพงศ์ มะโน)

ลงนาม .....

(อาจารย์วรวิทย์ สมหา)

ลงนาม .....

(อาจารย์สันติ ตันตระกูล)

หัวหน้าภาควิชา

ลงนาม.....

(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

AUTOMATIC SPEECH FOR EXAMINATION RESULT REPORTING

จุดประสงค์

1. เพื่อศึกษาเรื่องการประยุกต์ใช้ไมโครคอมพิวเตอร์ ในการควบคุมการบันทึกเสียงเป็นแฟ้มข้อมูล และการรายงานผลเป็นเสียงพูดออกทางโทรศัพท์
2. เพื่อออกแบบวงจรการบันทึกเสียงพูด โดยใช้ไอซี เอทูดี (A/D) และ ดีทูเอ (D/A) เป็นวงจรควบคุมการส่งเสียงพูดออกทางสายโทรศัพท์ และวงจรการรับข้อมูลจากสายโทรศัพท์มาประมวลผล
3. เพื่อสร้างอุปกรณ์และเขียนโปรแกรมของการบันทึกเสียง แล้วเก็บเป็นแฟ้มข้อมูลและสามารถส่งเสียงพูดที่บันทึกไว้ออกทางคู่สายโทรศัพท์ และสามารถที่จะรับข้อมูลจากคู่สายโทรศัพท์มาประมวลผลได้
4. เพื่อเขียนโปรแกรมเก็บข้อมูลผลการสอบของนักศึกษา ในการนำมาเป็นข้อมูลที่จะใช้ส่งผ่านวงจรเสียงออกทางคู่สายโทรศัพท์
5. เพื่อแสดงผลการทำงานของวงจร และโปรแกรมที่ใช้ควบคุมว่าสามารถที่จะทำการบันทึกเสียงลงในหน่วยความจำและสามารถแสดงผลตอบรับเป็นเสียงไปตามสายโทรศัพท์โดยอัตโนมัติ
6. เพื่อใช้ในสถานศึกษาสำหรับการรายงานผลการสอบประจำภาคเรียน โดยที่นักศึกษาโทรศัพท์เข้ามาสอบถามระบบอัตโนมัติ

ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจการทำงานรวมทั้งการออกแบบ การสร้างอุปกรณ์และรู้ถึงวิธีการประยุกต์ใช้งานของเครื่องไมโครคอมพิวเตอร์ร่วมกับโทรศัพท์ การรายงานผลการสอบโดยเป็นระบบที่มีเสียงพูดอัตโนมัติ
2. เป็นประโยชน์ต่อสถานศึกษาที่จะใช้การรายงานผลการสอบแต่ละภาคเรียนโดยไม่ต้องยุ่งยากในการมาติดต่อของนักศึกษา อำนวยความสะดวกแก่เจ้าหน้าที่
3. อำนวยความสะดวกแก่นักศึกษา โดยไม่ต้องเสียเวลาในการเดินทางมายังสถานศึกษา อีกทั้งยังรวดเร็วในการรู้ผลการสอบอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

นาย พงศ์เดช ฌรงค์ฤทธิ์
นาย ยุทธนา สรวลสรรค์
นาย วารินทร์ ชุมสำโรง
นาย สุริยา อุปถัมภ์

อาจารย์ที่ปรึกษา
อาจารย์กิตินงค์ มะโน
อาจารย์วรวิทย์ สมหา
อาจารย์สันติ ตันตระกูล
ปีการศึกษา 2537

บทคัดย่อ

ปริญญานิพนธ์นี้ เป็นการนำเอาระบบคอมพิวเตอร์มาประยุกต์ใช้กับระบบโทรศัพท์สำหรับแจ้งผลการสอบทางโทรศัพท์ในสถาบันการศึกษาต่างๆ ซึ่งข้อมูลถูกออกแบบมาใช้งานกับเครื่องโทรศัพท์ชนิดกดปุ่มโดยระบบจะทำการยกหูและวางหูเองอัตโนมัติ สามารถโต้ตอบกับผู้เรียกพร้อมทั้งทบทวนรหัสของผู้เรียกได้ด้วย ระบบแจ้งผลการสอบนี้สามารถใช้ในการเก็บรหัสประจำตัวและข้อมูลของนักศึกษา พร้อมทั้งเก็บเสียงเป็นแฟ้มข้อมูลในฮาร์ดดิสก์ ซึ่งจะควบคุมการทำงานของระบบด้วยไมโครคอมพิวเตอร์ขนาด 16 บิทขึ้นไป ระบบประกอบด้วยแผ่นวงจรพร้อมอุปกรณ์ที่ต่อใช้งานกับเครื่องไมโครคอมพิวเตอร์ และในส่วนของโปรแกรมควบคุมใช้ภาษาซีในการเขียนโปรแกรม เพื่อง่ายต่อการออกแบบและการพัฒนาระบบให้มีความรวดเร็วและมีประสิทธิภาพสูงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC SPEECH FOR EXAMINATION RESULT REPORTING

MR. PONGDECH NARONGRIT
MR. YUTTHNA SROULSRUNKA
MR. WARIN CHUMSAMRONG
MR. SURIYA UPATHUM

ADVISOR
MR. KITIPONG MANO
MR. WORAVIT SOMHA
MR. SUNTI TUNTRAKOOL
1994

ABSTRACT

THE PRESENTATION OF THIS PROJECT IS AN APPLICATION OF MICROPROCESSOR FOR REPORT THE EXAMINATION RESULT IN THE UNIVERSITY. IT CALLED THROUGH TELEPHONE USED IN THIS SYSTEM BY USED THE PUSHED BUTTON TYPE ONLY. THE SYSTEM IS AN AUTOMATIC OPERATION AND CAN PROVIDE SPEECH RESPONSE BACK TO INCOMING CALL.THE SYSTEM STORED A STUDENT DATA IN THE FILES AND CORESPONDING SOUND IN FILES OF HARDISK. THE SOFTWARE PROGRAM CAN USED WITH ANY 16BITS PERSONAL COMPUTER UPWARES.THIS SYSTEM COMPOSE OF 1 BOX AND SOFTWARE PROGRAM USED THE LANGUAGE THAT EADY TO DESIGN AND DEVELOPE FOR HIGHER SPEED AND MORE EFFICIENCY FOR FUTURE USE.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดีจากการสนับสนุนให้คำแนะนำและการช่วยเหลือจากท่าน อาจารย์ กิตินงค์ มะโน อาจารย์ วรวิทย์ สมหา อาจารย์ สันติ ตันตระกูล ที่เป็นที่ปรึกษา และอาจารย์ทุกท่านในภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม ที่คอยให้คำปรึกษาในการทำปริญญานิพนธ์ คุณดรุณี ชายทอง คุณอภิชาต วัฒนกุล คุณธีรวุฒิ มามุข คุณสุวิทย์ เสวรันต์ ที่เป็นผู้สร้างโครงงานต้นแบบนี้ เพื่อน ๆ ทุกคนที่คอยให้คำปรึกษา และให้คำแนะนำทั้งในด้านความรู้และเป็นกำลังใจตลอดมา ท้ายนี้ขอกราบระลึกถึงพระคุณบิดา มารดา และคณาจารย์ ที่ให้การสนับสนุนในทุกด้านแก่คณะผู้จัดทำปริญญานิพนธ์มาโดยตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
*2.1 ส่วนของโทรศัพท์	2
2.2 ส่วนของไมโครคอมพิวเตอร์	14
2.3 ทฤษฎีและส่วนบันทึกเสียง	29
บทที่ 3 การออกแบบ	43
3.1 ส่วนควบคุมการติดต่อทางโทรศัพท์	44
* 3.2 วงจรถอดรหัส DTMF	47
3.3 ส่วนของการบันทึกเสียง	48
3.4 ส่วนการควบคุม	54
3.5 สัญญาณต่างๆบนสล็อตของ IBM/PC	61
3.6 บัสของแหล่งจ่ายไฟระบบ	62
3.7 การจัดสัญญาณบนสล็อตของ IBM/PC	62
3.8 การใช้งานร่วมกับอุปกรณ์ต่างๆ	62
3.9 ส่วนของการเขียนโปรแกรม	65
บทที่ 4 การทดลองและผลการทดลอง	69
4.1 อุปกรณ์ที่ต้องใช้งาน	69
4.2 การเตรียมแฟ้มข้อมูลสำหรับเก็บรหัสนักศึกษา	69
4.3 การเตรียมแฟ้มข้อมูลเสียง	72
4.4 ขั้นตอนการทำงาน	75
4.5 ผลการทดลอง	76
บทที่ 5 บทวิจารณ์และสรุป	78
5.1.สรุปผลของโครงงาน	78
5.2.ปัญหาในการดำเนินงาน	78
5.3.ข้อแนะนำในการพัฒนา	78
ภาคผนวก ก โปรแกรมการทำงานของระบบ	80
ภาคผนวก ข รายละเอียดในส่วนอุปกรณ์	212
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การจัดแอดเดรสที่ใช้ติดต่อกับอุปกรณ์ภายนอกบน IBM PC	25
ตารางที่ 2.2 แสดงค่ากระแสเอาท์พุทของ Binary-Weighted Resistance D/A Converter	31
ตารางที่ 2.3 แสดง Resolution	35
ตารางที่ 3.1 ตำแหน่งแอดเดรสต่างๆและการนำ 8255 ไปใช้งาน	58
ตารางที่ 3.2 ตำแหน่งของพอร์ต	59
ตารางที่ 3.3 ตัวอย่างการตั้งคอนโทรลเวิ้ลในโหมด 0	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปภาพ		หน้า
รูปที่ 2.1	สัญลักษณ์ของเครื่องส่งและเครื่องรับ	3
รูปที่ 2.2	เครื่องส่งแบบคาร์บอน	3
รูปที่ 2.3	ส่วนประกอบของกระแสไฟสลับและไฟตรง	4
รูปที่ 2.4	การแยกส่วนประกอบของกระแสไฟสลับและไฟตรง	5
รูปที่ 2.5	เครื่องรับ	5
รูปที่ 2.6	กระดิ่งของเครื่องโทรศัพท์	6
รูปที่ 2.7	หน้าปัดของเครื่องโทรศัพท์แบบกดปุ่มและความถี่ที่ใช้	8
รูปที่ 2.8	สัญญาณโทรศัพท์แบบต่างๆ	9
รูปที่ 2.9	โครงสร้างภายในของ MT8870	11
รูปที่ 2.10	การต่อวงจรภาคอินพุท	12
รูปที่ 2.11	วงจรถอดรหัส DTMF	13
รูปที่ 2.12	แผนภูมิเวลาของ MT8870	14
รูปที่ 2.13	บล็อกไดอะแกรมของเครื่องไมโครคอมพิวเตอร์	15
รูปที่ 2.14	รายละเอียดของส่วนตัวเครื่อง	16
รูปที่ 2.15	การทำงานในหน่วยความจำแบบแคช	19
รูปที่ 2.16	การใช้สายเคเบิลเชื่อมต่อเครื่องมือภายนอกเข้ากับคอมพิวเตอร์	21
รูปที่ 2.17	การนำข้อมูลเข้าและส่งข้อมูลออก	21
รูปที่ 2.18	การนำข้อมูลในการเปิดประตูหรือหน้าต่างเข้าสู่คอมพิวเตอร์	22
รูปที่ 2.19	การส่งเสียงเตือนแก่ลำโพงจากเครื่องคอมพิวเตอร์	22
รูปที่ 2.20	การใช้ตัวแปลงสัญญาณเปลี่ยนโมเมนท์การเปิดประตูเป็นสัญญาณทางไฟฟ้าเข้าสู่คอมพิวเตอร์	23
รูปที่ 2.21	สายข้อมูล 8 บิทของ IBM PC	24
รูปที่ 2.22	ระบบเลขฐานสองที่ใช้สื่อสารใน IBM PC	24
รูปที่ 2.23	ระบบ อินพุท/เอาท์พุท สล็อต IBM PC	25
รูปที่ 2.24	ขาสัญญาณต่างๆในสล็อตอินพุท/เอาท์พุท บน IBM PC	26
รูปที่ 2.25	การ์ดที่ใช้เสียบใน อินพุท/เอาท์พุท สล็อต	26
รูปที่ 2.26	รายละเอียดของวงจร อีนาเบิล	28
รูปที่ 2.27	รายละเอียดการต่อวงจรเอาท์พุทแลทชักับ LED	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ	หน้า
รูปที่ 2.28 รายละเอียดของวงจรอินพุทบัฟเฟอร์	30
รูปที่ 2.29 รายละเอียดการต่อสวิทช์ 8 ทางกับวงจรอินพุทบัฟเฟอร์	31
รูปที่ 2.30 D/A Converter with Binary-Weighted Resistance	32
รูปที่ 2.31 Transistor Switch for D/A Converter	34
รูปที่ 2.32 แสดง Stair Case Output Current	35
รูปที่ 2.33 แสดงวงจร R-2R Ladder	37
รูปที่ 2.33 (ต่อ)แสดงวงจร R-2R Ladder	38
รูปที่ 2.34 แสดงถึง Ladder Impedance	39
รูปที่ 2.35 D/A Conversion with R-2R Ladder	40
รูปที่ 2.36 แสดง A/D Conversion with Counter	42
รูปที่ 2.37 แสดง A/D Conversion by Successive Approximasion	44
รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบแจ้งผลสอบทางโทรศัพท์อัตโนมัติ	45
รูปที่ 3.2 สภาวะแรงดันไฟฟ้าของสายโทรศัพท์	46
รูปที่ 3.3 วงจรตรวจสอบสัญญาณกริ่ง	48
รูปที่ 3.4 วงจรถอดรหัส DTMF	50
รูปที่ 3.5 วงจร DAC	52
รูปที่ 3.6 ผังเวลาการทำงานของ 0804	53
รูปที่ 3.7 วงจร ADC	54
รูปที่ 3.8 วงจรบันทึกเสียง	55
รูปที่ 3.9 ขาของไอซีเบอร์ 8255	56
รูปที่ 3.10 บล็อกไดอะแกรมของ 8255	57
รูปที่ 3.11 คอนโทรลเวิร์ดของ 8255 แบบต่างๆ	58
รูปที่ 3.11 (ต่อ)คอนโทรลเวิร์ดของ 8255 แบบต่างๆ	59
รูปที่ 3.12 การตั้งดิปสวิทช์พอร์ต 780H	62
รูปที่ 3.13 การต่อ 8255 เข้ากับอุปกรณ์แบบต่างๆ	65
รูปที่ 3.14 ไอซี 8255 I/O พอร์ต	66
รูปที่ 3.15 โฟลว์ชาร์ทเมนูหลัก	67
รูปที่ 3.16 โฟลว์ชาร์ทเมนูของชุดสัญญาณเสียง	67
รูปที่ 3.17 โฟลว์ชาร์ทแสดงชุดบันทึกข้อมูล	66
รูปที่ 3.18 โฟลว์ชาร์ทการทำงานของโปรมแกรมการค้นหาข้อมูล	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปภาพ	หน้า
รูปที่ 3.19 โฟลว์ชาร์ทการทำงานของโปรแกรม OPERATING	68
รูปที่ 4.1 แสดงเมนูหลัก (MAIN MENU)	69
รูปที่ 4.2 แสดงแบบฟอร์ม DATA BASE FILE	70
รูปที่ 4.3 แสดงแบบฟอร์ม DATA BASE MENU	71
รูปที่ 4.4 แสดงแบบฟอร์ม SOUND MENU	73
รูปที่ 4.5 แสดงแบบฟอร์ม RECORD SOUND	74
รูปที่ 4.6 แสดงแบบฟอร์ม PLAY SOUND	75
รูปที่ 4.7 แสดงแบบฟอร์ม EDIT SOUND	76
รูปที่ 4.8 แสดงแบบฟอร์มชุด OPERATING	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่ 1
บทนำ

ในยุคของโลกาภิวัตรทุกสิ่งทุกอย่างเริ่มที่จะเปลี่ยนแปลงไป มีการพัฒนาแทบทุกด้านยกตัวอย่างเช่น การสื่อสาร การคมนาคม การก่อสร้าง ฯลฯ รวมทั้งระบบคอมพิวเตอร์ที่ต้องการความไวในการปฏิบัติงาน ประเทศไทยเราก็มีการพัฒนาสิ่งต่างๆเหล่านั้นไปพร้อมกับต่างประเทศด้วย สิ่งจำเป็นที่ถือเป็นปัจจัยพื้นฐานของการพัฒนาก็คือ การพัฒนาทรัพยากรมนุษย์หรือบุคคลนั่นเอง ถ้าหากเรามีกำลังคนที่มีคุณภาพแล้วก็สามารถในการสร้างสรรสิ่งที่มีประโยชน์ และมีคุณค่าต่อสังคม และสิ่งที่จะช่วยพัฒนาทรัยากรมนุษย์ของเรานั่นก็คือ การศึกษา

หากมองไปในระบบการศึกษาของบ้านเราในปัจจุบัน ความคล่องตัวในสถานศึกษาต่างๆยังคงมีน้อยอยู่ ระบบการรายงานผลข้อมูลต่างๆยังไม่สะดวกเท่าที่ควร สำหรับสิ่งที่เราสนใจในที่นี้คือการอำนวยความสะดวกทางด้านการรายงานผลการสอบประจำเรียนแบบอัตโนมัติโดยที่นักศึกษาไม่ต้องเสียเวลาเดินทางมาสถานศึกษา สามารถทำได้เพียงแค่โทรศัพท์มายังภาควิชาหรือสถาบันที่ตนสอบ แล้วเราจะมีระบบที่ทำหน้าที่ตอบรับโดยอัตโนมัติซึ่งเราเรียกว่า ระบบแจ้งผลสอบทางโทรศัพท์อัตโนมัติ (Automatic Speech ForExamination Result Reporting)

สำหรับเครื่องมือที่อำนวยความสะดวกดังกล่าว จะทำงานโดยรับรหัสเลขหมายจากการกดรหัสของนักศึกษา เป็นรหัสประจำตัวเข้ามาตรวจสอบค้นหาผลการสอบที่ได้บันทึกผลไว้ก่อนหน้านี้แล้ว แล้วแจ้งผลการสอบออกมาเป็นเกรดเฉลี่ย ถ้าในกรณีที่รหัสที่กดมาระบบค้นหาไม่พบระบบก็จะแจ้งให้นักศึกษาทราบจะเห็นได้ว่าสะดวกมากทีเดียว ถ้าหากสถาบันการศึกษาต่างๆ ได้นำไปใช้นักศึกษาที่อยู่ต่างจังหวัดไม่ต้องเดินทางเข้ามาก็สามารถที่จะรับรู้ผลการสอบได้ และเจ้าหน้าที่ก็จะได้รับความสะดวกเพียงแค่ป้อนข้อมูลและบันทึกผลการเรียนของนักศึกษาครั้งเดียวก็สามารถใช้ได้ทันที และถ้าต้องการใช้ในครั้งต่อไปก็ทำได้โดยการป้อนข้อมูลเก็บในแฟ้มข้อมูลใหม่ก็จะใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ จะประกอบด้วย 2 ส่วนใหญ่ๆที่สำคัญ คือส่วนที่เกี่ยวกับโทรศัพท์ และส่วนไมโครคอมพิวเตอร์

สำหรับการศึกษาจะแยกศึกษาตามส่วนต่างๆที่ได้กล่าวมา และรวมทั้งการเชื่อมต่ออุปกรณ์ภายนอกกับเครื่องไมโครคอมพิวเตอร์ และระบบการบันทึกเสียงพูดทั้งนี้เนื่องมาจากเราจะเชื่อมต่อโทรศัพท์เข้ากับเครื่องคอมพิวเตอร์เมื่อผู้เรียกหมุนเลขหมายเข้ามา ส่วนของระบบจะส่งสัญญาณเป็นเสียงพูดให้ผู้เรียกกดรหัสประจำตัวแล้วระบบจะค้นหาผลแล้วแจ้งไปยังผู้รับเป็นเสียงพูด ซึ่งจะกระทำโดยระบบทั้งสิ้น ซึ่งทฤษฎีก็จะแยกกล่าวเป็นส่วนๆ ดังนี้

2.1.ส่วนของโทรศัพท์

สามารถจะแบ่งได้เป็น 2 ส่วน คือ

2.1.1.ส่วนประกอบและหลักการทำงานของโทรศัพท์

เครื่องโทรศัพท์จะประกอบด้วยส่วนสำคัญๆ ทั้งหมดดังนี้

1.เครื่องส่ง (Transmitter)

2.เครื่องรับ (Receiver)

3.กระดิ่ง (Ringing)

4.ฮุค สวิทช์

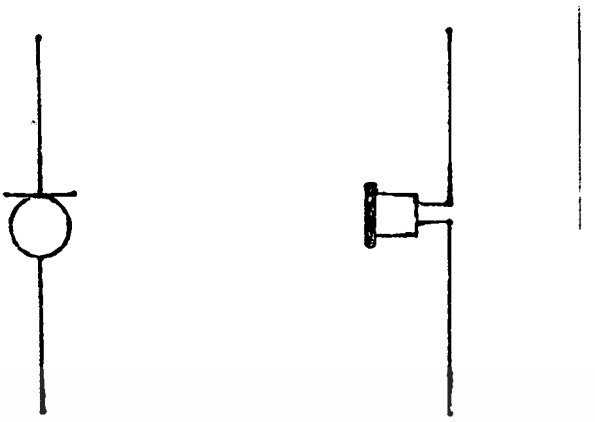
5.หน้าปัดสำหรับกดหรือหมุน (Dial)

เครื่องส่งและเครื่องรับ

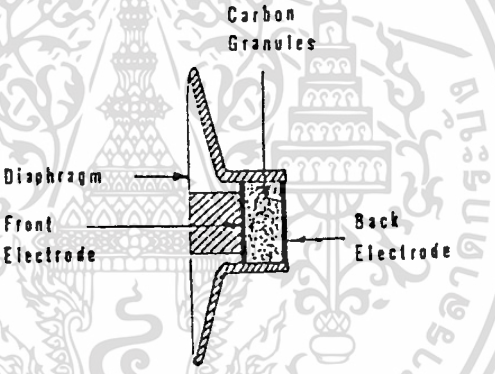
เครื่องส่งและเครื่องรับจะนำมารวมกันเรียกว่า ปากพูดหูฟัง (Handset) โดยใช้เป็นอุปกรณ์แปลงพลังงานโดยที่เครื่องส่งทำหน้าที่เปลี่ยนพลังงานเสียงเป็นพลังงานไฟฟ้าและเครื่องรับทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานเสียง สัญลักษณ์ของเครื่องส่ง และเครื่องรับ แสดงดังรูปที่ 2.1

สำหรับเครื่องส่งที่ใช้กันโดยทั่วๆไปเป็นแบบผงถ่าน เนื่องจากแบบนี้มีประสิทธิภาพและความไวสูง เมื่อเราทำการพูดติดต่อกันก็จะสามารถรับฟังเสียงพูดได้ชัดเจนแม้ว่าตัวส่งจะอยู่ห่างจากปากเราก็ตาม ซึ่งมีส่วนประกอบคือส่วนเล็กของคาร์บอนเรียกว่า ผงถ่าน,แผ่นอิเล็กโทรดแบบคาร์บอน 2 แผ่น และแผ่นไดอะแฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ทำซ้ำ ดัดแปลง หรือนำไปเผยแพร่ต่อ



รูปที่ 2.1 สัญลักษณ์ของเครื่องส่งและเครื่องรับ



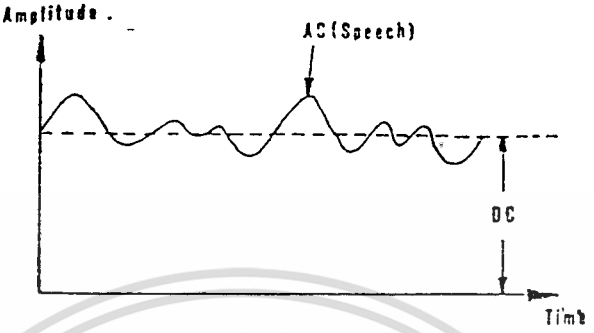
รูปที่ 2.2 เครื่องส่งแบบคาร์บอน

หลักการทำงานคือ เมื่อมีคลื่นเสียงมากระทบที่แผ่นไดอะแฟรมซึ่งทำให้แผ่นไดอะแฟรมสั่น พลังงานเสียงจะเปลี่ยนเป็นพลังงานกล เมื่อแผ่นไดอะแฟรมถูกกดทำให้แผ่นอิเล็กโทรดเคลื่อนที่เข้าผงถ่านจะถูกอัดติดแน่นมาก การอัดตัวของผงถ่านทำให้ความต้านทานระหว่างแผ่นอิเล็กโทรดทั้งสองมีค่าลดลงในทางตรงกันข้ามถ้าแผ่นไดอะแฟรมคลื่นที่ออกจะทำให้อิเล็กโทรดแผ่นหน้าเคลื่อนที่ออกความต้านทานของเครื่องส่งเพิ่มขึ้น เมื่อเราเอาแบตเตอร์รี่ต่อเข้าระหว่างแผ่นอิเล็กโทรดทำให้กระแสไฟตรงไหลผ่าน และเนื่องจากความต้านทานของเครื่องส่ง มีการเปลี่ยนแปลงเมื่อได้รับเสียงแล้ว ดังนั้นกระแสที่ไหลผ่านเครื่องส่งจะเปลี่ยนแปลงพลังงานเสียงก็จะเปลี่ยนเป็นพลังงานไฟฟ้า

ในรูปที่ 2.3 จะเห็นว่ากระแสไฟสลับเกิดจากสัญญาณเสียง จะเกิดซ้อนทับกับกระแสไฟตรงส่วนของไฟสลับจะถูกแยกออกจากไฟตรง โดยใช้หม้อแปลงหรือตัวเก็บประจุเป็นตัวแยกและถูก

เอกสารนี้เป็น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

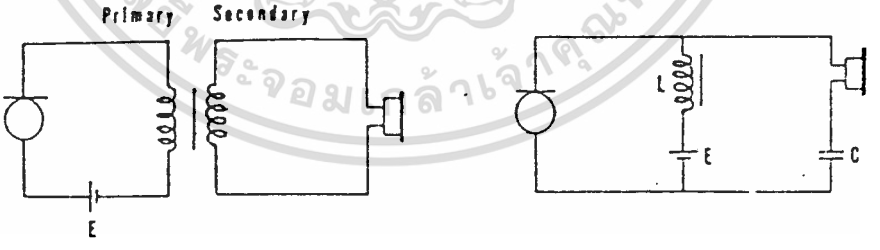
ส่งไปยังเครื่องรับ ดังแสดงในรูปที่ 2.4



รูปที่ 2.3 ส่วนประกอบของกระแสไฟสลับและไฟตรง

รูป 2.4 (ก) กระแสไฟสลับในขดปฐมภูมิจะเหนี่ยวนำให้เกิดกระแสในขดทุติยภูมิ ทำให้มีกระแสไหลผ่านเครื่องรับ ส่วนกระแสไฟตรงจะไหลเฉพาะในขดปฐมภูมิ

รูป 2.4 (ข) กระแสตรงจะไหลผ่านขดลวดที่มีความต้านทานต่ำ และไหลผ่านเครื่องส่งและตัวเก็บประจุ C ได้แต่ไม่ผ่านเข้าขดลวดในขณะที่สัญญาณเสียงที่ได้รับจากโทรศัพท์มีค่าน้อยมาก ดังนั้นควรออกแบบให้มีประสิทธิภาพที่เปลี่ยนพลังงานไฟฟ้าให้เป็นเสียงมากที่สุด



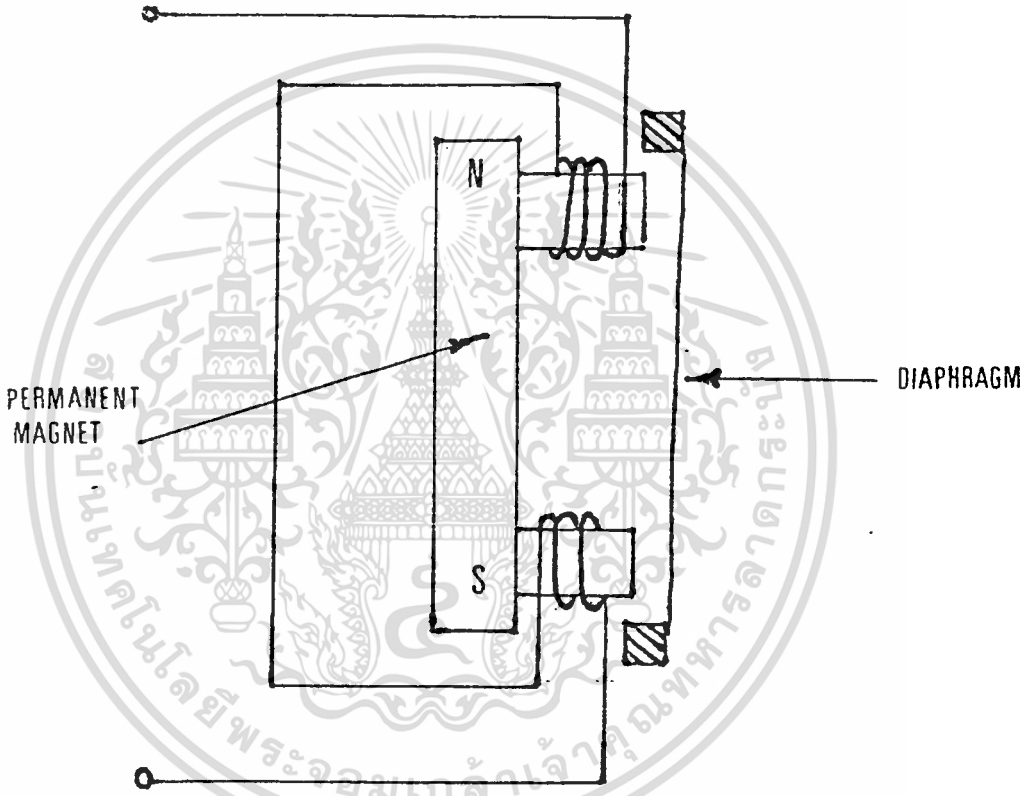
(ก) (ข)

รูปที่ 2.4 การแยกส่วนประกอบของกระแสไฟสลับและไฟตรง

หลักการของเครื่องรับ อธิบายคือ ขดลวดที่พันอยู่ในแกนแม่เหล็กถาวรต่อแบบอนุกรม แต่ขดลวดพันกลับทิศทางแม่เหล็กถาวรจะมีอำนาจดูดแผ่นไดอะแฟรมเข้ามา เมื่อมีไฟกระแสสลับ (Speed current) ไหลผ่านขดลวดทำให้เกิดเส้นแรงแม่เหล็ก ซึ่งอาจมีทิศทางเสริมหรือต้านกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนามแม่เหล็กถาวร แผ่นไดอะแฟรมจะเคลื่อนที่เข้าหรือออกตามขนาดของไฟกระแสสลับ และทำให้เกิดคลื่นเสียงที่มีขนาดและความถี่เท่ากับไฟกระแสสลับที่ไหลเข้ามาในวงจร เสียงที่เกิดขึ้นย่อมมีการสูญเสียบ้าง เนื่องจากการเปลี่ยนรูปพลังงาน ดังนั้นเอาท์พุทจึงน้อยกว่าอินพุท

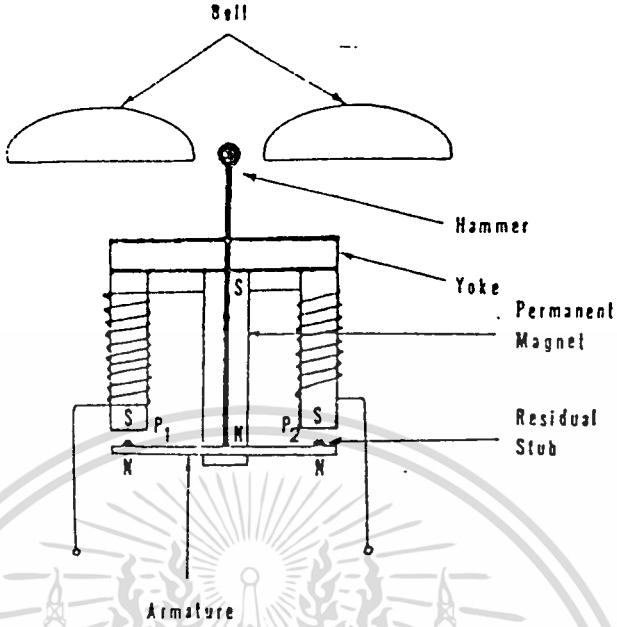


รูปที่ 2.5 เครื่องรับ

กระดิ่งของเครื่องรับโทรศัพท์

เมื่อมีการเรียก กระดิ่งที่เครื่องรับโทรศัพท์ของผู้ถูกเรียกจะดังขึ้น ซึ่งหมายถึงชุมสายโทรศัพท์ได้ทำการส่งไฟฟ้ากระแสสลับของสัญญาณกริ่ง มาป้อนที่กระดิ่งของเครื่องโทรศัพท์โดยใช้กระแสไฟฟ้าของสัญญาณกริ่ง จะมีค่าประมาณ 75-90 โวลท์ 85-25 $H_{\pm}$ ลักษณะของกริ่งดังแสดงในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 กระดิ่งของเครื่องโทรศัพท์

รูปที่ 2.6 กระดิ่งของโทรศัพท์ ประกอบด้วยขดลวด 2 ขด ต่ออนุกรมพันอยู่บนแกนเหล็กติดกับโย้ค มีแม่เหล็กถาวรอยู่ตรงกลางโย้ค ส่วนอาเมเจอร์จะวางอยู่ในลักษณะที่ตรงจุดกึ่งกลางของมันติดอยู่ที่แกนแม่เหล็กถาวร โดยสารที่ติดอยู่ส่วนปลายของอาเมเจอร์ไม่ใช่สารแม่เหล็ก และจะมีเรซิดวลสตับ (Residual Stup) ติดอยู่ตรงข้ามกับขั้ว P_1 และ P_2

หลักการทำงานของเส้นแรงแม่เหล็กจากแม่เหล็กถาวรทำให้เกิดขั้ว S ที่ขั้ว P_1 และP_2 และทำให้เกิดขั้ว n ที่ปลายอาเมเจอร์ในสภาวะปกติจะไม่มีกระแสไหลผ่านขดลวดทำให้ อาเมเจอร์ถูกดูดด้วยแรงเท่ากัน หรืออาจถูกดูดไปที่ปลายข้างใดข้างหนึ่งของขั้ว P_1 และ P_2 ก็ได้ เมื่อมีการเรียกจะมีไฟกระแสสลับไหลผ่านขดลวดทั้งสอง ทำให้เกิดอำนาจแม่เหล็ก N และ S ขึ้นที่ขั้ว P_1 และ P_2 ซึ่งทำให้เกิดอำนาจแม่เหล็กมากที่ปลายขั้ว S (P_2) และที่ปลายขั้วอีกด้าน P_1 หมดอำนาจแม่เหล็กอาเมเจอร์จะถูกดูดมายังขั้วที่มีแรงแม่เหล็กมากกว่า และเนื่องจากเป็นไฟฟ้ากระแสสลับดังนั้นจึงทำให้ อาเมเจอร์ถูกดูดสลับข้างกันตามอำนาจแม่เหล็กก้านตีจะไปตีกระดิ่งทั้งสองสลับข้างกันทำให้กระดิ่งดัง

หน้าปัดของเครื่องโทรศัพท์

หน้าปัดของเครื่องโทรศัพท์ที่ใช้กับชุมสายอัตโนมัติ ปัจจุบันมีอยู่ด้วยกันสองแบบคือแบบหมุน (Rotary Dial) ซึ่งการหมุนทำให้เกิดพัลซ์ ขึ้นเป็นจำนวนเท่ากับหมายเลขที่หมุนและแบบกดปุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งกรรมวิธีของ Dual Tone Multifrequency (DTMF) ในการส่งเลขหมายโทรศัพท์สำหรับระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติจะใช้โทรศัพท์แบบกดปุ่ม เนื่องจากแบบนี้ใช้กันเป็นส่วนใหญ่โดยจะมีข้อดีกว่าแบบหมุน พอจำแนกได้ดังนี้

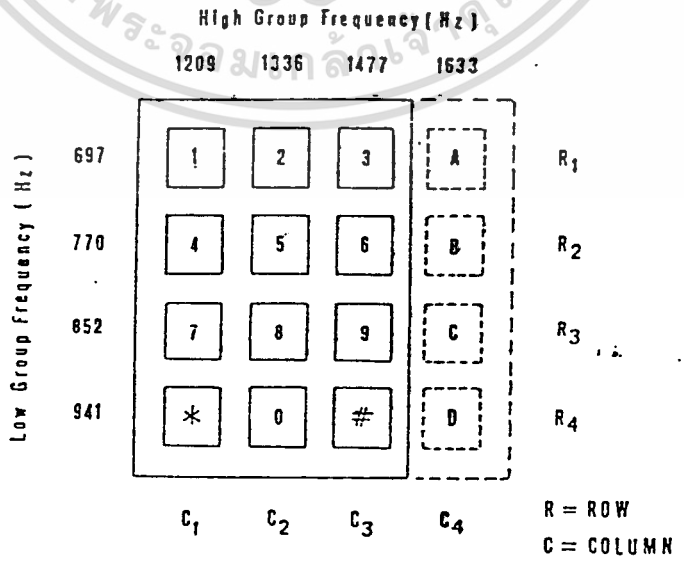
สามารถใช้วงจรทางสารกึ่งตัวนำแทนอุปกรณ์ทางกล ซึ่งทำให้มีความรวดเร็วและแม่นยำในการส่งเลขหมาย

สามารถลดเวลาในการหมุนหมายเลขลงได้ ทำให้ผลเวลาเฉลี่ยที่ใช้โทรศัพท์แต่ละครั้งลดลง (Holding Time) ซึ่งทำให้ชุมสายโทรศัพท์สามารถรับการจราจรได้มากขึ้น

สามารถกดปุ่มเพิ่มขึ้นได้อีก 4 ปุ่ม (หลักที่ 4) เพื่อใช้ในการส่งสัญญาณบริการอื่นๆ

มีความเหมาะสมที่จะใช้กับระบบชุมสายอัตโนมัติ หน้าปัดแบบกดปุ่มใช้กรรมวิธีของ Dual Tone Multifrequency (DTMF) ในการส่งเลขหมายโทรศัพท์นั้นโดยทั่วไปหน้าปัดจะมี 12 ปุ่ม แบ่งเป็น 4 แถว 3 หลักและในเครื่องโทรศัพท์บางแบบอาจมีถึง 16 ปุ่ม โดยเพิ่มหลักที่ 4 ดังแสดงในรูปที่ 2.7 ความถี่ที่ใช้ในแต่ละแถว แต่ละหลัก จึงมีความถี่ต่างกัน ความถี่ทั้ง 4 แถว เรียกว่ากลุ่มความถี่ต่ำ(Low Group Frequency)และความถี่แต่ละ 3 หรือ 4 หลักเรียกว่ากลุ่มความถี่สูง (High Group Frequency) การกดปุ่มที่หมายเลขใดๆจะทำให้วงจรอิเล็กทรอนิกส์ภายในเครื่องโทรศัพท์ผลิตความถี่ออกมา 2 ความถี่ เช่นเมื่อกดหมายเลข 5 ความถี่ที่ผลิตออกมาคือ 770 H_z และ 1336 H_z เป็นต้น

มาตราฐานความถี่ที่ใช้และตำแหน่งของเลขหมายต่างๆ จะถูกจัดให้มีเลขหมายดังแสดงตามรูปที่ 2.7 สำหรับความผิดพลาดที่ยอมให้เกิดขึ้นได้เป็น 1.5% สำหรับการผลิตความถี่ และ 2% สำหรับรับเลขหมาย



รูปที่ 2.7 หน้าปัดของเครื่องโทรศัพท์แบบกดปุ่มและความถี่ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.มาตราฐานสัญญาณโทรศัพท์

สัญญาณโทรศัพท์เป็นสิ่งที่จำเป็นเพื่อให้สัญญาณโทรศัพท์ใช้งานกันได้จึงกำหนดมาตราฐานของสัญญาณโทรศัพท์ขึ้น เพื่อบอกสภาวะการใช้งานของโทรศัพท์สัญญาณต่างๆได้แก่

1.สัญญาณแมวกรน (Dial Tone)

เป็นสัญญาณความถี่ 425 H_z ทำการส่งต่อเนื่องกันไปใช้บอกให้ฝ่ายเรียกเริ่มทำการหมุนหรือกดเลขหมายเพื่อการเรียกออกได้

2.สัญญาณไม่ว่าง (Busy Tone)

เป็นสัญญาณความถี่ 425 H_z ทำการส่ง 0.4 วินาทีและหยุด 0.4 วินาทีสลับกัน เพื่อบอกให้รู้ว่าฝ่ายรับคู่สายไม่ว่างจะต้องทำการวางหูก่อนแล้วรอสัญญาณแมวกรนใหม่

3.สัญญาณเรียกกลับ (Ringing Back Tone)

เป็นสัญญาณความถี่ 425 H_z ทำการส่ง 1 วินาทีและหยุด 4 วินาทีสลับกัน เพื่อบอกให้ฝ่ายเรียกคู่สายได้แล้ว เพียงแต่รอฝ่ายรับมารับสายเท่านั้น

4.สัญญาณกระดิ่ง (Ringing Tone)

เป็นสัญญาณความถี่ประมาณ 25 H_z ทำการส่ง 1 วินาทีและหยุด 4 วินาทีสลับกันเช่นเดียวกับสัญญาณเรียกกลับ แต่จะมีระดับสัญญาณแรงไฟที่สูงกว่า เพื่อบอกให้ฝ่ายรับทราบว่า มีการเรียกเข้ามา เพื่อทำการยกหูแล้วสนทนาติดต่อกันต่อไป

สำหรับการถอดรหัสสัญญาณโทรศัพท์ เราเลือกใช้ไอซีสำเร็จรูปที่ใช้สำหรับการถอดรหัสสัญญาณ DTMF โดยเฉพาะในที่นี้ในส่วนของทฤษฎีเราจะศึกษาไอซี MT8870 เพราะเนื่องจากเป็นเบอร์ที่หาง่าย ใช้ง่ายและเป็นที่นิยม

รายละเอียดของ ไอซีเบอร์ MT8870

1.คุณสมบัติของ MT8870

เป็นตัวรับและถอดรหัสความถี่ DTMF(DTMF Receiver & Decoder)

กินไฟน้อย ใช้ไฟเลี้ยงระดับ TTL

สามารถปรับ การ์ดไทม์ (Guard Time)

สามารถตั้งอัตราขยายภายในตัวไอซีได้

เป็นไอซีคุณภาพสูง

2.การนำ MT8870 ไปใช้งาน

นำไปใช้งานด้านรีโมทคอนโทรล

เครื่องป้องกันโทรศัพท์ทางไกล

ใช้ในงานเกี่ยวกับเครดิตการ์ด

ใช้งานร่วมกับคอมพิวเตอร์

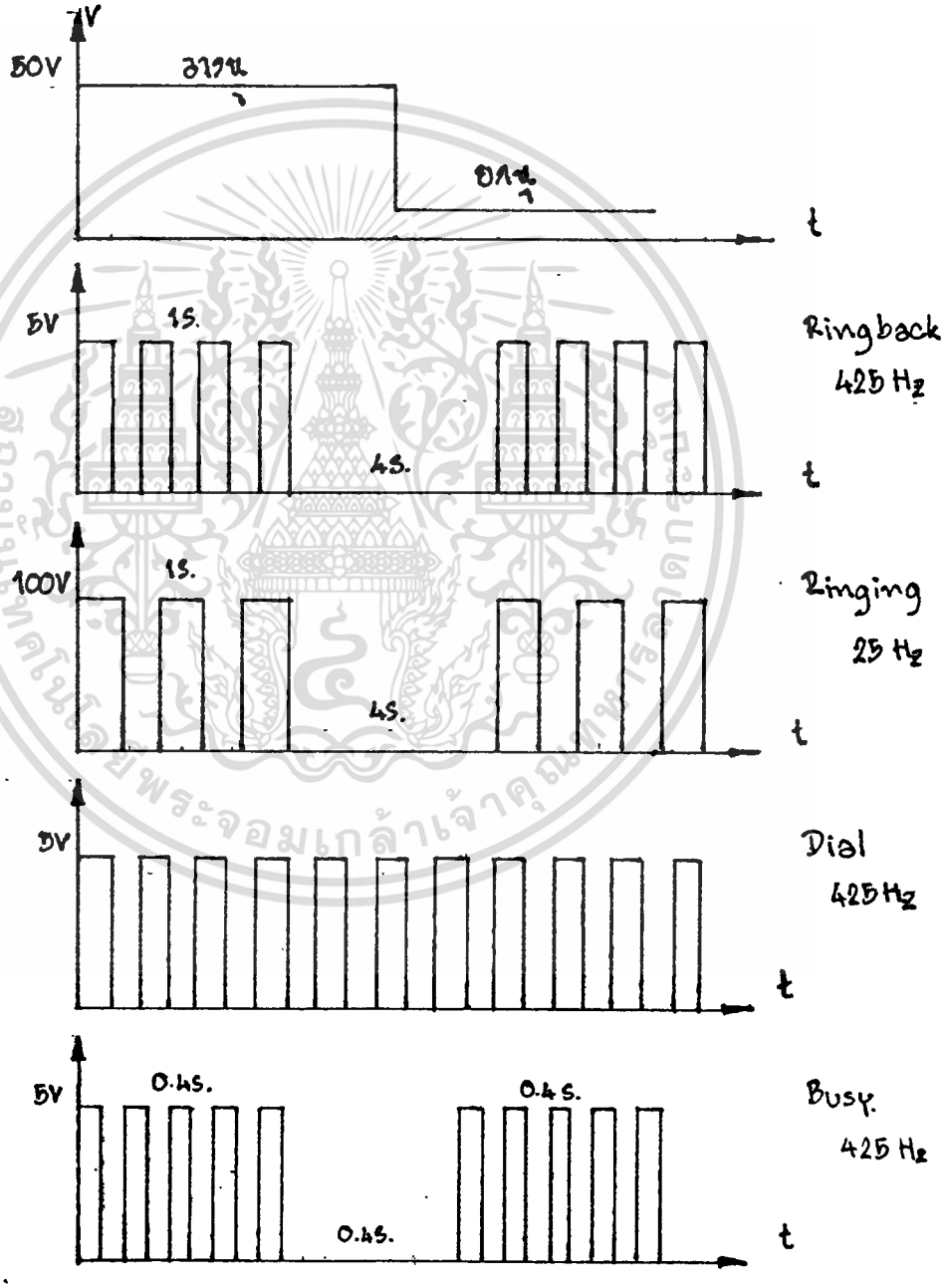
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้กับงานด้านโทรศัพท์โดยทั่วไป

เครื่องกันขโมย

การควบคุมอุปกรณ์ทางโทรศัพท์

ใช้ทำเครื่องสอบถามทางโทรศัพท์



รูปที่ 2.8 สัญญาณโทรศัพท์แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังชั่นทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO^{2}-CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิทช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอลเพื่อตรวจจับ และถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐาน 2 ขนาด 4 บิทและเช็คช่วงเวลาที่สัญญาณเข้ามาส่วนภาคอินพุทเป็นวงจรแลทช์ 3 สถานะภายในโครงสร้างของ MT8870 จะประกอบไปด้วยส่วนสำคัญ 5 ส่วน คือ

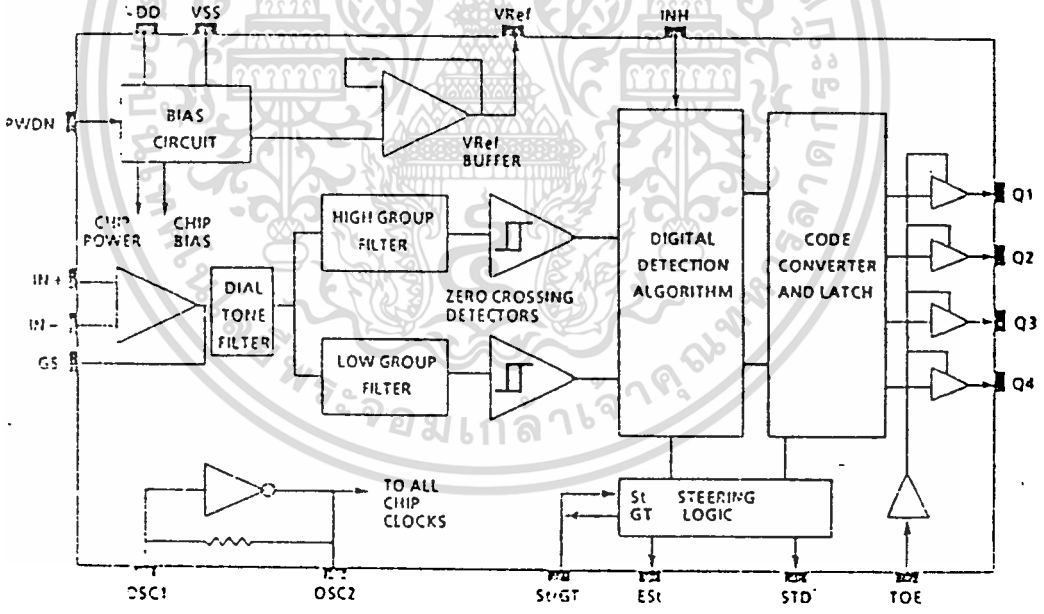
1)ภาคกรองความถี่ (Filter Section)

2)ภาคถอดรหัส (Decoder Section)

3)ภาคตรวจสอบสัญญาณ(Steering Circuit)

4)ภาคขยายสัญญาณความแตกต่าง(Differential Input)

5)ภาคกำเนิดความถี่(Osillator)



รูปที่ 2.9 โครงสร้างภายในของ MT8870

3.1 ภาคกรองสัญญาณความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่คือช่วงความถี่สูงและความถี่ต่ำ โดยใช้วงจรกรองความถี่อันดับ 6 ชนิดสวิทช์คาปาซิเตอร์ (Sixth-Order Switch Capacitor Band Pass Filter)ความถี่ที่แยกได้มี 2 ช่วงคือช่วงความถี่สูงและช่วงความถี่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิตอลและมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตราฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสมเมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่เข้า Est (Early Steering) หรือขา 16 ก็จะแอคทีฟสำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆ นั้น

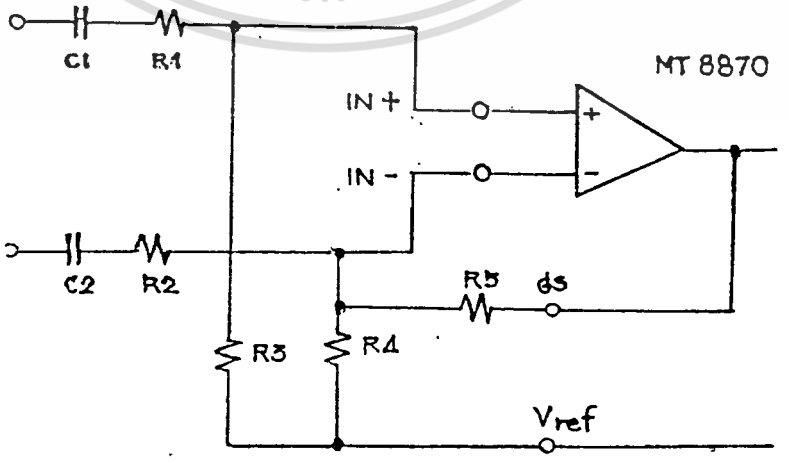
3.3 ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาท์พุทจะมีการตรวจสอบความถี่ช่วงที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตุจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลา พอสมควรมิฉนั้นวงจรนี้จะถือว่าสัญญาณนั้นไม่ถูกต้องส่วนช่วงเวลายาวเท่าใดสามารถได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา Est จะเป็น High นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามาให้ VC สูงขึ้นตัวเก็บประจุ C จะคายประจุทำให้แรงดัน VC สูงขึ้นจนถึงค่าเทรสโฮลด์วงจรถอดรหัสออกเป็นตัวเลขขนาด 4 บิทรายละเอียดในการทำงานดูได้จากแผนภูมิเวลา (Timing Diagram) จะเข้าใจได้ง่าย

สำหรับการ์ดไทม์ นั้นหมายถึงช่วงเวลาของความถี่ที่เข้ามาซึ่งจะต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่เราตั้งไว้ถึงจะยอมรับว่าสัญญาณนั้นถูกต้องหรือพูดได้ว่าเวลาที่เราตั้งไว้โดยRC ก็คือการ์ดไทม์นั่นเองถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่า ก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป

3.4 ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุท MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเพิ่มเข้าไปดังรูป



รูปที่ 2.10 การต่อวงจรภาคอินพุท

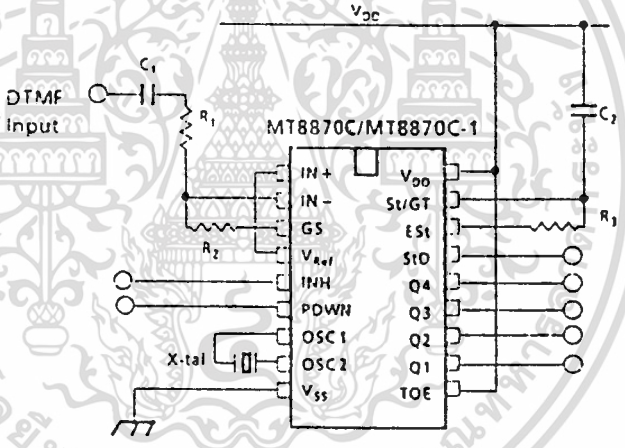
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอินพุท และอิมพีแดนซ์ได้ดังนี้

อัตราขยาย $AV_{diff} = R_1/R_2$

อินพุทอิมพีแดนซ์ $(Zin_{diff} = 2 \quad (R_L)^2 + (1/WC)^2$

สำหรับวงจรการต่อใช้งานก็กำหนดค่าต่างๆมาโดยเฉพาะ การทำงานสำหรับวงจรการถอดรหัส DTMF เมื่อผู้เรียกกดเลขหมาย หลังจากได้รับการตอบรับเรียบร้อยแล้ว สัญญาณความถี่ที่ผสมกันมาจะเข้าสู่วงจรการถอดรหัส DTMF เบอร์ MT8870 เมื่อถอดรหัสแจ้งจะได้เอาท์พุทออกมาเป็น 4 บิทแล้วส่งไปยังส่วนควบคุมต่อไป

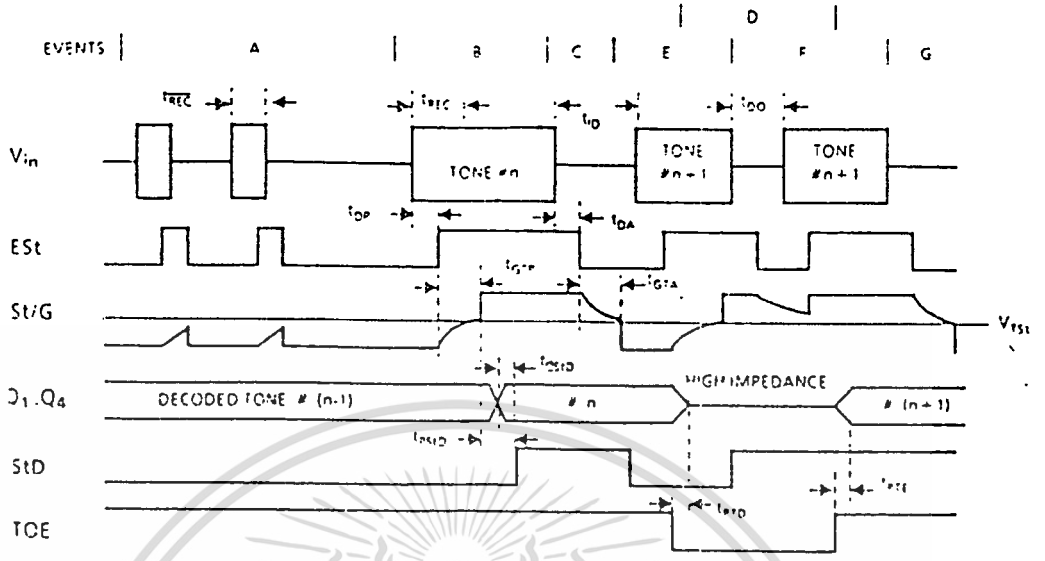


รูปที่ 2.11 วงจรถอดรหัส DTMF

ขั้นตอนการทำงาน

เพื่อให้ทราบถึงรายละเอียดต่างๆ ในการทำงาน และหน้าที่ของขาต่างๆ ของ MT8870 ในการต่อเข้าเป็นระบบเชื่อมโยงกับส่วนอื่นๆ สามารถที่จะอธิบายด้วย แผนภูมิเวลาข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แผนภูมิเวลาของ MT 8870

A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาท์พุทไม่เปลี่ยน

B - ความถี่ # n ถูกตรวจพบ และมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแลทช์ไว้ที่เอาท์พุท

C - ช่วงความถี่ # n ช่วงห่างถูกต้องเอาท์พุทยังคงแลทช์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D - เอาท์พุทเปลี่ยนเป็น High Impedance

E - ความถี่ # n+1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแลทช์ไว้

F - ความถี่ # n+1 หมายถึงช่วงห่างไม่ถูกต้อง เอาท์พุทยังคงแลทช์อยู่

G - จบความถี่ # n+1 ช่วงถูกต้อง เอาท์พุทยังคงแลทช์อยู่ จนถึงความถี่ใหม่ที่ถูกต้องเข้ามา

Vin - สัญญาณความถี่ DTMF ที่เข้ามา

Est - Early Steering Output ใช้แสดงความถี่ที่ถูกต้อง

St/Gt - Steering Input/Guard Time Output สำหรับต่อกับ RC ภายนอก

$Q_1 Q_4$ - เอาท์พุท BCD ขนาด 4 บิท

STD - Delayed Steering Output ใช้แสดงค่าความถี่ที่ได้รับหรือหายไปมีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE - Tone Output Enable (Input) ใช้ควบคุม Q_1 - Q_4 ให้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

High Impedance

T_{REC} - คาบเวลานานที่สุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง

T_{ID} - เวลาสิ้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

T_{DO} - เวลานานที่สุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาที่ถูกต้อง

T_{DP} - เวลาที่ใช้ในการตรวจสอบพบสัญญาณความถี่ DTMF ที่ถูกต้อง

T_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

T_{GTD} - การ์ดไทม์ ของการปรากฏความถี่ DTMF

T_{GTA} - การ์ดไทม์ ของการหายไปของความถี่ DTMF

ภาคกำเนิดความถี่

ในภาคนี้ภายในไอซีจะมีวงจรอยู่ภายใน เพียงแต่ต่อแร่คริสตอลขนาด 3.75 MH_z ก็สามารถใช้งานได้ทันที

2.2.ส่วนของไมโครคอมพิวเตอร์

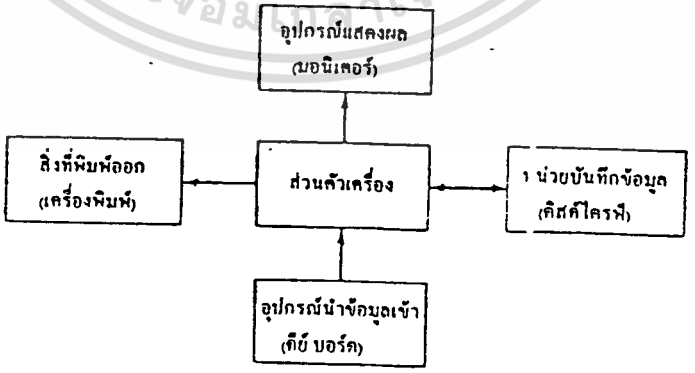
ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติจะใช้คอมพิวเตอร์ขนาด 16 บิทขึ้นไป สำหรับการศึกษาจะแบ่งเป็นส่วนๆเช่นเดียวกันคือ

ส่วนประกอบของไมโครคอมพิวเตอร์

การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับเครื่องไมโครคอมพิวเตอร์เบื้องต้น

ฮาร์ดแวร์อินพุท/เอาท์พุทสำหรับ IBM PC

2.2.1.ส่วนประกอบของไมโครคอมพิวเตอร์



รูปที่ 2.13 บล็อกไดอะแกรมของเครื่องไมโครคอมพิวเตอร์

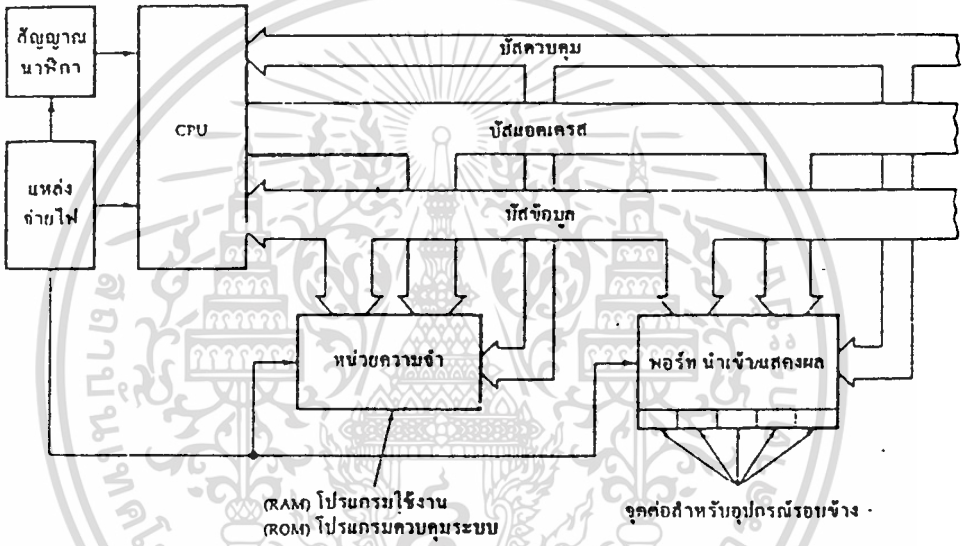
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุด
คณะครุศาสตร์อุตสาหกรรม สจล.
5
พ121 ร 2537

รูปที่ 2.13 สามารถอธิบายได้ดังนี้

ส่วนตัวเครื่อง (Unit System) จะประด้วยหน่วยความจำซีพียู อินพุท/เอาท์พุทพอร์ต รอม (ROM) แรม (RAM) และแหล่งจ่ายไฟบางกรณีอาจรวมถึงแผ่นขับจานแม่เหล็ก (Disk Drive) รวมอยู่ด้วยหรืออาจถือเป็นอุปกรณ์รอบข้างก็ได้

รายละเอียดของส่วนตัวเครื่อง



รูปที่ 2.14 รายละเอียดของส่วนตัวเครื่อง

1.ส่วนประมวลผลกลาง (Central Processing Unit : ซีพียู)

เป็นหน่วยที่สำคัญที่สุดของเครื่องคอมพิวเตอร์ มีหน้าที่ควบคุมการทำงานของหน่วยต่างๆ ของระบบคอมพิวเตอร์ให้ทำงานสอดคล้องกัน โดยทั่วไปซีพียูจะประกอบด้วยส่วนย่อยดังนี้

1.1 หน่วยรีจิสเตอร์ (Register)

เปรียบเสมือนสมองทำหน้าที่เก็บข้อมูลที่ส่งมาจากหน่วยความจำหลัก และนำไปใช้ประมวลผล รีจิสเตอร์แบ่งตามลักษณะการใช้งานได้ 2 แบบ

1.1.1 รีจิสเตอร์แบบเอนกประสงค์ (General Purpose Register)

รีจิสเตอร์ชนิดนี้ทำหน้าที่เก็บค่าต่างๆ ที่นำมาใช้ในการประมวลผลซีพียู ตัวอย่างเช่น แอคคิวมูเลเตอร์ (Accumulator) ที่ใช้เก็บค่าตัวตั้งหรือผลลัพธ์ที่ใช้ในการคำนวณ

1.1.2 รีจิสเตอร์เฉพาะกิจ (Special Purpose Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

~~1289~~ 021057

ทำหน้าที่ติดตามซีพียู เช่น โปรแกรมเคาท์เตอร์ (Program Counter) ซึ่งทำหน้าที่เก็บแอดเดรสขนาด 16 บิท มีหน้าที่สำหรับซีพียู เฟตช์ (Fetch) คำสั่งในหน่วยความจำได้ถูกต้องหรืออินเด็กรีจิสเตอร์(Index Register) จะใช้เป็นฐานการชี้ไปยังบริเวณหน่วยความจำที่ทางผ่านเข้าออกของข้อมูล

1.2 หน่วยความจำและตรรก (Arithmatic Logic Unit : ALU)

เป็นหน่วยประมวลผลข้อมูลด้วยวิธีทางเลขคณิตและทางตรรก เช่น บวก ลบ คูณ หาร เปรียบเทียบ เป็นต้น

1.3 หน่วยควบคุม (Control Unit)

หน่วยนี้ควบคุมการย้ายข้อมูลระหว่างรีจิสเตอร์แต่ละชนิดด้วยกัน และติดต่อกับส่วนภายนอกด้วย นอกจากนี้ยังควบคุมการอ่าน การเขียนข้อมูลจากหน่วยความจำหรืออุปกรณ์รอบข้างด้วย

2.หน่วยความจำ (Memory)

ในคอมพิวเตอร์แบ่งเป็น 2 ส่วนใหญ่ๆ คือ

2.1 หน่วยความจำในซีพียู มี 2 ส่วนคือ

2.1.1 รีจิสเตอร์ (Register) ตามหัวข้อที่ 1.1

2.1.2 หน่วยความจำแบบแคช (Catch Memory)

เป็นหน่วยความจำชั่วคราวที่มีความเร็วในการทำงานสูงมากเชื่อมโยงระหว่างหน่วยประมวลผลกลางและหน่วยความจำหลัก ใช้เก็บข้อมูลหรือคำสั่งการทำงานในขณะนั้นดังรูปที่ 2.15

เหตุที่มีหน่วยความจำแบบแคชเพราะเมื่อซีพียู จะเริ่มทำการแอคซิคิ้วก็จะหาคำสั่งได้จากหน่วยความจำแบบแคช ซึ่งทำงานได้เร็วกว่าเพราะหน่วยความจำแบบแคชมีความเร็วสูงกว่าหน่วยความจำหลัก หน่วยความจำแบบแคชจะมีอยู่ในเครื่องระดับ 80386 ขึ้นไป

2.2 หน่วยความจำนอก ซีพียูแบ่งได้ 2 ส่วนย่อยคือ

2.2.1 หน่วยความจำหลัก (Primary Memory) เป็นที่ซึ่งโปรแกรมหรือข้อมูลจะต้องเข้ามาอยู่ก่อนการประมวลผลซึ่งได้แก่ แรม (RAM) และ รอม (ROM)

2.2.2 หน่วยความจำสำรอง (Secondary Memory) เป็นหน่วยความจำอยู่ภายนอกเครื่องคอมพิวเตอร์ใช้เก็บข้อมูลหรือโปรแกรมได้จำนวนมาก และเก็บไว้ใช้งานได้เป็นเวลานานเช่น เทปแม่เหล็ก หรือ แผ่นจานแม่เหล็ก เป็นต้น

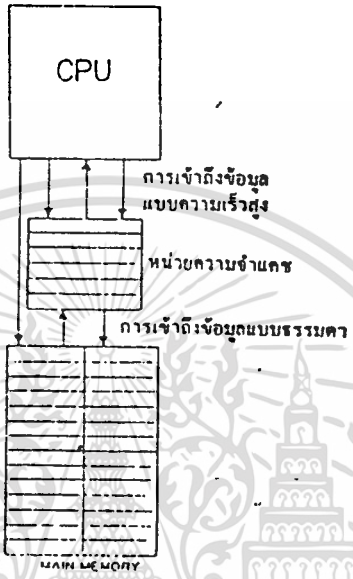
3.บัส (BUS)

บัส หมายถึง เส้นทางในการส่งถ่ายสัญญาณโดยการต่ออุปกรณ์ต่างๆ เช่น หน่วยความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำและอุปกรณ์ภายนอก เป็นต้น

สิ่งที่ต้องพิจารณาเป็นอันดับแรก คือ เรื่องของความจุของบัส ซึ่งเป็นขนาดของจำนวนบิทที่สามารถส่งถ่ายในเวลาเดียวกันได้เท่าใด เช่น บัส 8 บิท และสามารถส่งถ่ายข้อมูล 8 บิท ได้พร้อมๆ กัน



รูปที่ 2.15 การทำงานในหน่วยความจำแบบแคช

ลักษณะการใช้งานของบัสมี 3 แบบ คือ

3.1 สายนำข้อมูล (Data Bus)

เป็นเส้นทางในการส่งถ่ายข้อมูลระหว่างรีจิสเตอร์ และหน่วยความจำซึ่งลักษณะการสื่อสารแบบ 2 ทาง (Bidirectional) โดยทั่วไปไมโครคอมพิวเตอร์จะมีสายนำข้อมูล 8 บิท และ 16 บิท จนพัฒนาขึ้นเป็น 32 บิท ในปัจจุบัน

3.2 สายนำสัญญาณบอกตำแหน่ง (Address Bus)

เป็นทางส่งผ่านตำแหน่งของหน่วยความจำหรืออุปกรณ์ต่างๆโดยระบุผ่านทางแอดเดรสบัสในในการสื่อสารทางเดียว (Unidirectional) โดยส่งจากส่วน ซีพียู

3.3 สายควบคุม (Control Bus)

ทำหน้าที่เป็นทางผ่านของสัญญาณควบคุมจากทั้ง ซีพียู และอุปกรณ์ต่างๆ โดยเป็นการสื่อสารแบบสองทาง (Bidirectional)

4.จุดต่อเข้า/ออก (Input/Output Ports)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือจุดที่อุปกรณ์รอบข้างภายนอกสามารถส่งผลที่ต้องการแสดงให้กับคอมพิวเตอร์ได้อาจเป็นการสื่อสารแบบทางเดียวหรือสองทางก็ได้ แล้วแต่ลักษณะการใช้

5.อุปกรณ์รอบข้าง (Peripheral Device)

หมายถึง อุปกรณ์ต่างๆที่อยู่ภายใต้การควบคุมของซีพียู ซึ่งถือว่าเป็นส่วนหนึ่งของระบบคอมพิวเตอร์ สามารถจำแนกได้ตามลักษณะของอุปกรณ์ได้ดังนี้

5.1 อุปกรณ์นำข้อมูลเข้า (Input Device)

หมายถึงเครื่องมือหรืออุปกรณ์ที่สามารถอ่านข้อมูลและส่งข้อมูลเข้าไปเก็บในเครื่องคอมพิวเตอร์ได้ เพื่อให้ทำการประมวลผลต่อไป

5.1.1 คีย์บอร์ด เป็นอุปกรณ์นำเข้าข้อมูลที่นิยม มี 3 แบบ 84,101 และ102 คีย์

5.1.2 เมาส์ เป็นอุปกรณ์นำเข้าที่ขนาดพอเหมาะกับมือ สามารถเคลื่อนย้ายไปบนพื้นผิวเรียบโดยเชื่อมต่อกับคอมพิวเตอร์จะเป็นตัวเคลื่อนย้ายตัวชี้ตำแหน่ง (Cursor) ในจอ CRT ให้ไปในทิศทางที่ต้องการ

5.2 อุปกรณ์แสดงผล (Output Device)

หน้าที่ของส่วนนี้จะแสดงข้อมูลที่มีอยู่ในหน่วยความจำ หรือผลลัพธ์ที่เครื่องคำนวณคำนวณได้ และแสดงบนจอภาพหรือสิ่งอื่นซึ่งมีหลายชนิด เช่น กระดาษอ่านบันทึกอ่านอุปกรณ์แสดงผลที่สำคัญมีดังนี้

5.2.1 หน่วยจอภาพ หมายถึงจอภาพหรือเห็นเป็นหลอดภาพ

5.2.2 เครื่องพิมพ์เป็นอุปกรณ์รอบข้างอีกชนิดหนึ่งของเครื่องคอมพิวเตอร์ ที่ใช้แสดงผลลัพธ์ที่ได้จากการทำงานของเครื่องคอมพิวเตอร์ โดยการพิมพ์ลงบนกระดาษส่วนพิมพ์ลงในกระดาษนี้เรียกว่า ฮาร์ดดิสก์ กอปปี้

6.หน่วยความจำสำรอง (Secondary Storeage Device)

เป็นสิ่งที่ใช้เก็บข้อมูลเพิ่มจากหน่วยความจำหลักในคอมพิวเตอร์ ตัวอย่างเช่น

1.แผ่นดิสก์ (Floppy Disk)

2.ฮาร์ดดิสก์ (Hard Disk)

3.เทป (Tape)

2.2.2.การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับเครื่องไมโครคอมพิวเตอร์

ในส่วนแรกนั้นได้กล่าวถึงโครงสร้าง ส่วนประกอบของเครื่องไมโครคอมพิวเตอร์ไปแล้ว ในส่วนนี้จะกล่าวถึงการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับเครื่องไมโครคอมพิวเตอร์เบื้องต้น เนื่อง

เอกสารนี้เป็นสิ่งที่ ... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเราจำเป็นต้องต่อเครื่องโทรศัพท์เข้ากับเครื่องไมโครคอมพิวเตอร์ใช้งานร่วมกัน ดังนั้นเราจำเป็นจะต้องรู้หลักการต่อซึ่งตัวอย่างรายละเอียดดังจะกล่าวต่อไปนี้

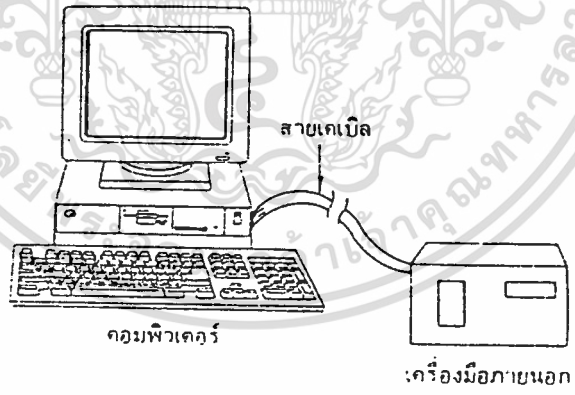
เครื่องคอมพิวเตอร์ที่เราใช้งานกันทั่วไปนอกจากจะใช้ซอฟแวร์แล้ว ยังสามารถใช้งานในการเชื่อมต่อกับเครื่องมือภายนอก (External Instrument) ต่างๆ เพื่อทำการวัดปริมาณทางกายภาพและส่งผลกลับในการควบคุมปริมาณทางกายภาพที่แวดล้อมตัวเรา อาจจะเป็นภายในบ้าน ที่ทำงานหรือห้องทดลอง ปริมาณทางกายภาพเหล่านี้ได้แก่ อุณหภูมิ ความชื้น ความเข้มแสง โมเมนท์การเปิด-ปิดประตูเป็นต้น

ในการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับไมโครคอมพิวเตอร์ได้ใช้สายเคเบิลในการสื่อสารแสดงดังรูปที่ 2.16

คอมพิวเตอร์มีเส้นทางในการเชื่อมต่อกับเครื่องมือภายนอก โดยมีการเรียกข้อมูลเข้าและเมื่อต้องการควบคุมก็ทำการส่งข้อมูลออก ดังนั้นหลักการในการทำงานเชื่อมต่อสามารถกล่าวได้ดังนี้คือ

การเรียกข้อมูลเข้า (Input Data)

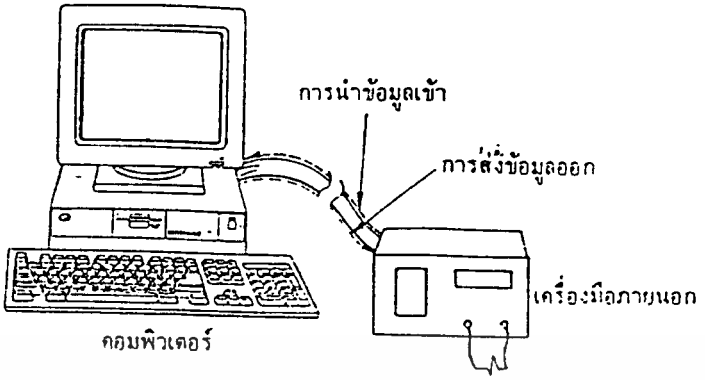
การส่งข้อมูลออก (Output Data) ดังแสดงในรูปที่ 2.17



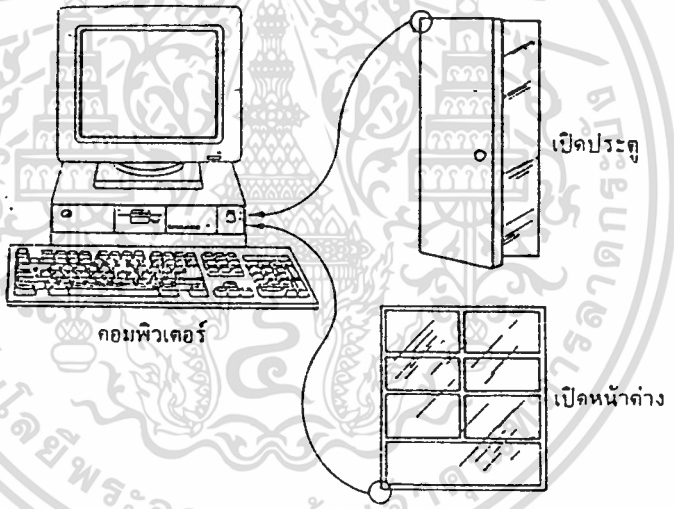
รูปที่ 2.16 การใช้สายเคเบิลเชื่อมต่อเครื่องมือภายนอกเข้ากับคอมพิวเตอร์

สำหรับการเชื่อมต่ออุปกรณ์ควบคุมภายในบ้าน ยกตัวอย่างง่ายๆเช่น เมื่อมีการเปิดประตูหรือหน้าต่าง จะให้คอมพิวเตอร์ส่งสัญญาณเตือนเข้าที่ลำโพง โดยใช้หลักการเชื่อมต่อดังกล่าวเริ่มจากการเรียกข้อมูลการเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การนำข้อมูลเข้าและส่งข้อมูลออก



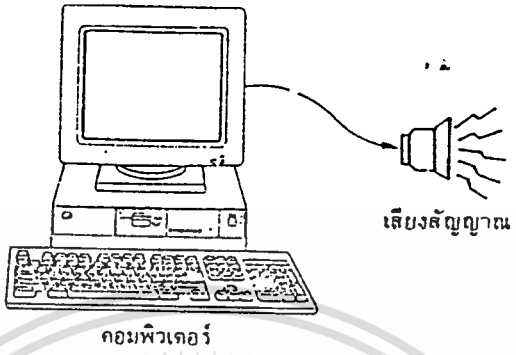
รูปที่ 2.18 การนำข้อมูลในการเปิดประตูหรือหน้าต่างเข้าสู่คอมพิวเตอร์

เริ่มจากการเรียกข้อมูลการเปิดประตูหรือหน้าต่างเข้าสู่คอมพิวเตอร์ เมื่อไม่มีการเปิดประตูก็จะไม่มีการส่งข้อมูลไปที่ลำโพง แต่เมื่อไรมีการเปิดประตู คอมพิวเตอร์ก็จะทำการส่งข้อมูลเสียงเตือนไปที่ลำโพงทันทีแสดงดังรูปที่ 2.18 และ 2.19

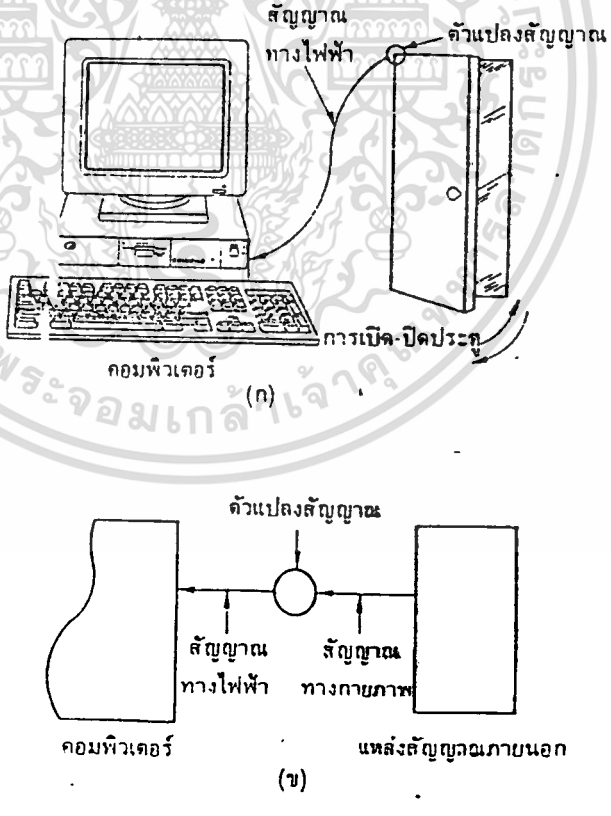
หากพิจารณาต่อไปก็มีคำถามอีกว่า คอมพิวเตอร์นั้นรู้ได้อย่างไรว่ามีการเปิดประตูหรือหน้าต่างซึ่งแน่นอนคอมพิวเตอร์ไม่สามารถไปตรวจสอบการเปิดประตูหรือหน้าต่าง โดยตรงจำเป็นจะต้องใช้ตัวแปลงสัญญาณ (Transducer) ซึ่งมีการเปลี่ยนแปลงโมเมนท์ (Moment) การเปิดประตูหรือหน้าต่างเป็นสัญญาณไฟฟ้าเข้าสู่คอมพิวเตอร์ แสดงดังรูปที่ 2.20 ในการส่งข้อมูลออกที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำโพงก็ต้องส่งข้อมูลออกในรูปสัญญาณไฟฟ้าเช่นเดียวกัน



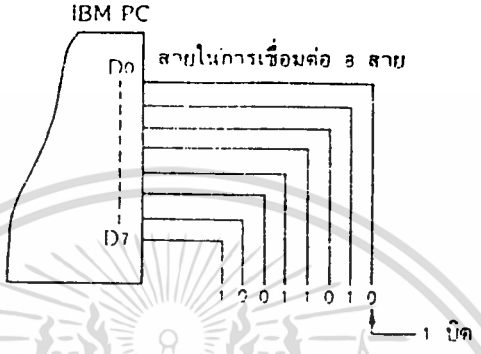
รูปที่ 2.19 การส่งเสียงเตือนแก่ลำโพงจากเครื่องคอมพิวเตอร์



รูปที่ 2.20 การใช้ตัวแปลงสัญญาณเปลี่ยนโมเมนท์การเปิดประตูเป็นสัญญาณทางไฟฟ้าเข้าสู่คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งและรับข้อมูลมีสายในการเชื่อมต่อ 8 สาย (D_0-D_7) แต่ละสายเรียกว่า 1 บิท การสื่อสารจะใช้ระบบเลขฐานสอง (Binary System) คือเป็น "0" และ "1" ซึ่ง "1" ในที่นี้คือแรงดันไฟฟ้า +5 โวลท์ ส่วน "0" คือแรงดันไฟฟ้า 0 โวลท์แสดงดังรูปที่ 2.21 และ 2.22 ตามลำดับ



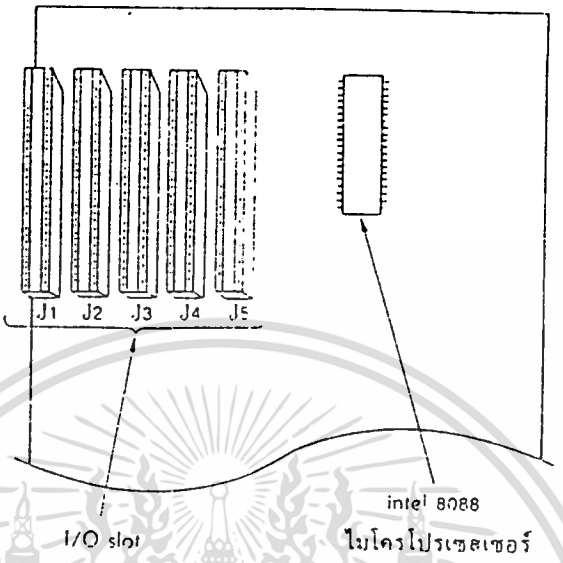
รูปที่ 2.21 สายข้อมูล 8 บิทของ IBM PC

รูปที่ 2.22 ระบบเลขฐานสองที่ใช้สื่อสารใน IBM PC

2.2.3.ฮาร์ดแวร์อินพุท/เอาท์พุท สำหรับ IBM PC

ในการส่งข้อมูลออก (Output Data) และนำข้อมูลเข้า (Input Data) สำหรับ IBM PC ที่เราได้กล่าวถึงในส่วนที่ 2 นั้นจำเป็นต้องมีวงจรอิเล็กทรอนิกส์เชื่อมต่อเข้ากับIBM PC ดังนั้นเราต้องรู้จัก สล็อต (Slot) บน IBM PC ซึ่งมีระบบ อินพุท/เอาท์พุท สล็อตแสดงรายละเอียดดังรูปที่ 2.23 และรูปที่ 2.24 ตามลำดับ

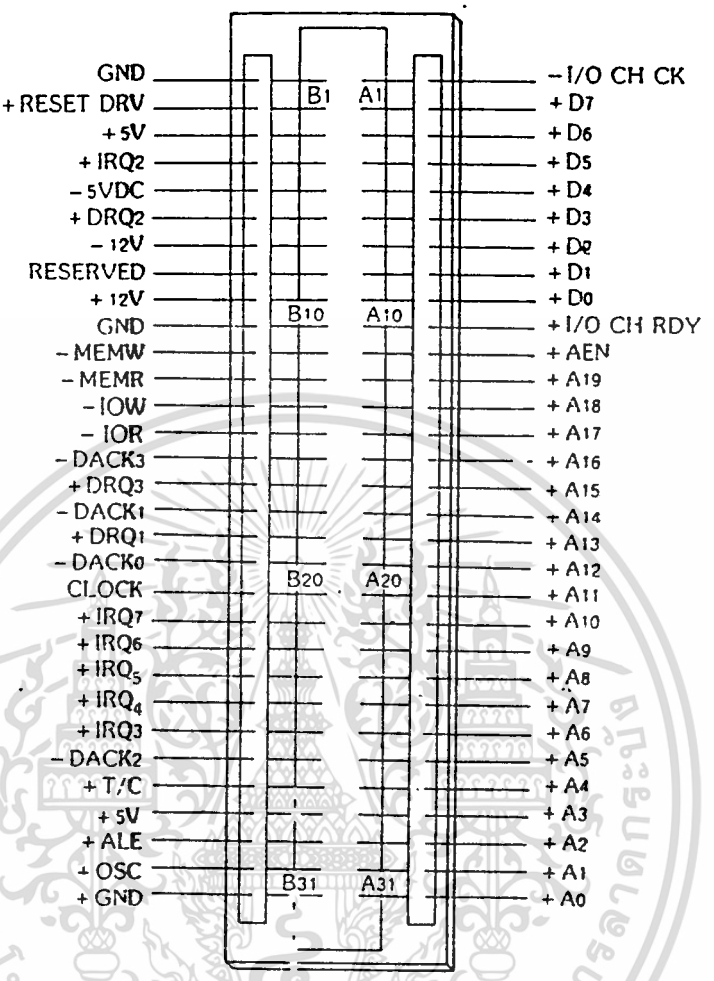
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



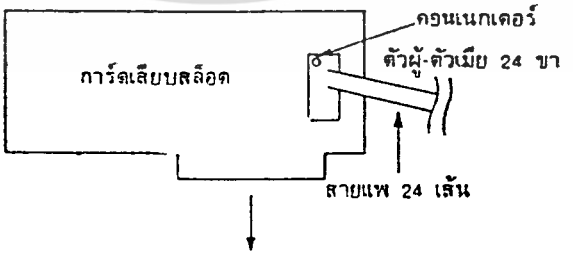
รูปที่ 2.23 ระบบ อินพุท/เอาท์พุท สล็อตบน IBM PC

การส่งข้อมูลออกและนำเข้าจำเป็นต้องมีเส้นทางในที่นี้เราเรียกว่า พอร์ต (Port) ซึ่งมีการจัดแอดเดรสไม่ให้ซ้อนกัน มีแอดเดรสที่แน่นอน ตัวอย่างการจัดสรรแอดเดรสที่ใช้ติดต่อกับอุปกรณ์ภายนอกบน IBM PC แสดงดังตารางที่ 2.1 และ การสร้างการ์ด (Card) เพื่อใช้เชื่อมต่อกับ สล็อตอินพุท/เอาท์พุท แสดงดังรูปที่ 2.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 ขาสัญญาณต่างๆ ในสล็อตอินพุท เอาท์พุท บน IBM PC



รูปที่ 2.25 การ์ดที่ใช้เสียบใน อินพุท/เอาท์พุท สล็อต (PC-Card)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นที่แอดเดรสของอุปกรณ์ I/O										
เลขฐาน 16	9	8	7	6	5	4	3	2	1	0	
00-0F	0	0	0	0	0	Z	A_3	A_2	A_1	A_0	DMA CHIPB237-2
20-21	0	0	0	0	1	Z	Z	Z	Z	A_0	INTERUPT 8259A
40-43	0	0	0	1	0	Z	Z	Z	A_1	A_0	TIMER 8253-5
60-63	0	0	0	1	1	Z	Z	Z	A_1	A_0	PPI 8255-5
80-83	0	0	1	0	0	Z	Z	Z	A_1	A_0	DMA PAGE REGS
AX	0	0	1	0	1						NMI MARK REG
CX	0	0	1	1	0						RESREVE
EX	0	0	1	1	1						RESERVE
3F8-3FF	1	1	1	1	1	1	1	A_2	A_1	A_0	TP RS-232-C CD
3F0-3F7	1	1	1	1	1	1	0	A_2	A_1	A_0	$5^{1/4}$ DRV ADAPTER
2F8-2FF	1	0	1	1	1	1	1	Z	A_1	A_0	RESERVE
378-37F	1	1	0	1	1	1	1	Z	A_1	A_0	PARALLEL PRTR
3D0-3DF	1	1	1	1	0	1	A_3	A_2	A_1	A_0	COLOR/GRAPHIC
278-27F	1	0	0	1	1	1	1	Z	A_1	A_0	RESERVE
200-20F	1	0	0	0	0	0	A_3	A_2	A_1	A_0	GAME I/O ADAPTER
3B0-3BF	1	1	1	0	1	1	A_3	A_2	A_1	A_0	IBM MONOCHROME DISPLAY PARALLEL

ตารางที่ 2.1 การจัดแอดเดรสที่ใช้ติดต่อกับอุปกรณ์ภายนอกบน IBM PC

ฮาร์ดแวร์ในการส่งอินพุท/เอาท์พุทข้อมูล มีรายละเอียดวงจรการเชื่อมต่อดังนี้

1. วงจรอีนาเบิล (Enable Circuit)
2. วงจรเอาท์พุทแลตช์ (Output Latches Circuit)
3. วงจรอินพุทบัฟเฟอร์ (Input Buffer Circuit)

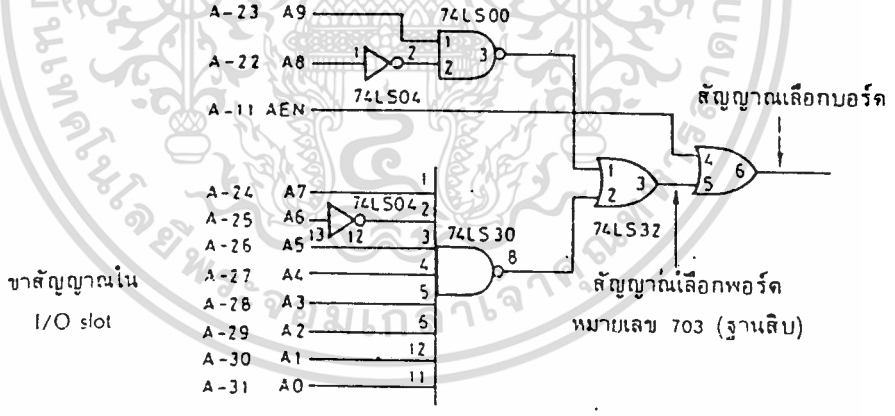
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.วงจรอีนาเบิล (Enable Circuit)

การสื่อสารภายในเครื่องคอมพิวเตอร์มีสายข้อมูล (Data Bus) เพื่อส่งและรับข้อมูลถึงกัน การส่งและรับข้อมูลถ้าไม่มีตำแหน่งการส่งและรับข้อมูลก็ไม่รู้ว่าจะส่งและรับอย่างไร ดังนั้นจึงจำเป็นต้องมีสายแอดเดรสในการกำหนดตำแหน่งภายในคอมพิวเตอร์

กรณีสื่อสารกับอุปกรณ์ภายนอก ก็ต้องมีการถอดรหัสแอดเดรสให้เลขหมายแก่อุปกรณ์ภายนอกตัวอื่น แอดเดรสที่ใช้สื่อสารกับอุปกรณ์ภายนอกสำหรับ IBM PC ดังแสดงรายละเอียดในตารางที่ 2.1 ซึ่งพบว่าแอดเดรสในช่วง 2B0-2BF (เลขฐานสิบหก) ไม่มีการใช้จึงสามารถถอดรหัสแอดเดรสใช้งานการเชื่อมต่อในช่วงดังกล่าว เพื่อใช้ในการส่งข้อมูลออกและนำข้อมูลเข้า

การถอดรหัสแอดเดรสเพื่อการเลือกพอร์ต (Port Select) ได้ใช้สายแอดเดรสที่ A_0 ถึง A_9 มีค่าเท่ากับ 1010111111 (หมายเลข 703_{10}) จึงทำให้สัญญาณการเลือกพอร์ตมีลอจิก "0" แต่การจะส่งหรือรับข้อมูลจากสายข้อมูลได้ต้องมีการต่อกับสาย AEN (Address Enable) ของ IBM PC ด้วยเพราะไม่เช่นนั้นการส่งหรือรับข้อมูลอาจผิดพลาดได้ สัญญาณที่นำมาออร์ (OR) กับ AEN จะใช้ในการอีนาเบิลวงจรส่งหรือรับข้อมูลเกิดขึ้นได้ รายละเอียดแสดงในรูปที่ 2.26



รูปที่ 2.26 รายละเอียดของวงจรอีนาเบิล

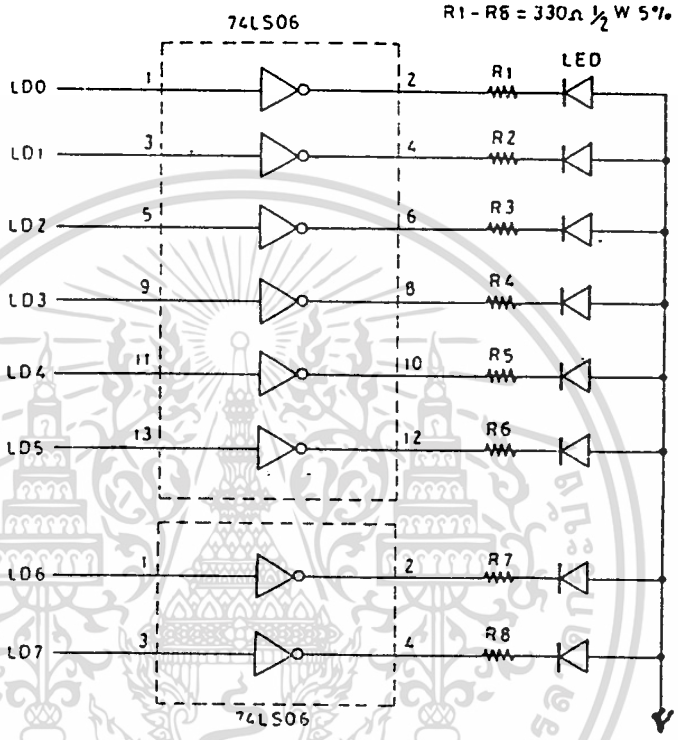
2.วงจรเอาท์พุทแลตช์ (Output Latches Circuit)

การส่งข้อมูลออกใช้ ไอซี #74LS374 เป็นตัวแลตช์ข้อมูล การส่งข้อมูลออกต้องผ่านทางสายข้อมูล D_0-D_7 ต้องทำการอีนาเบิลเพื่อแลตช์ข้อมูล ซึ่งไม่เพียงพอที่จะทำงานได้ต้องมีอีกสายหนึ่งที่มาเกี่ยวข้องคือ -IOW (Input/Output Write) สายสัญญาณนี้มีลอจิก "0" เมื่อมีการใช้สั่งให้ส่งข้อมูลออก ในภาษาเบสิกเพื่อการส่งข้อมูลออก รายละเอียดของวงจรดังแสดงในตาราง

เอกสารนี้เป็น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 2.1

การแสดงผลสามารถใช้ LED 8 ดวง เป็นตัวแสดงผลข้อมูล 8 บิท แสดงการเชื่อมต่อดังรูปที่ 2.27

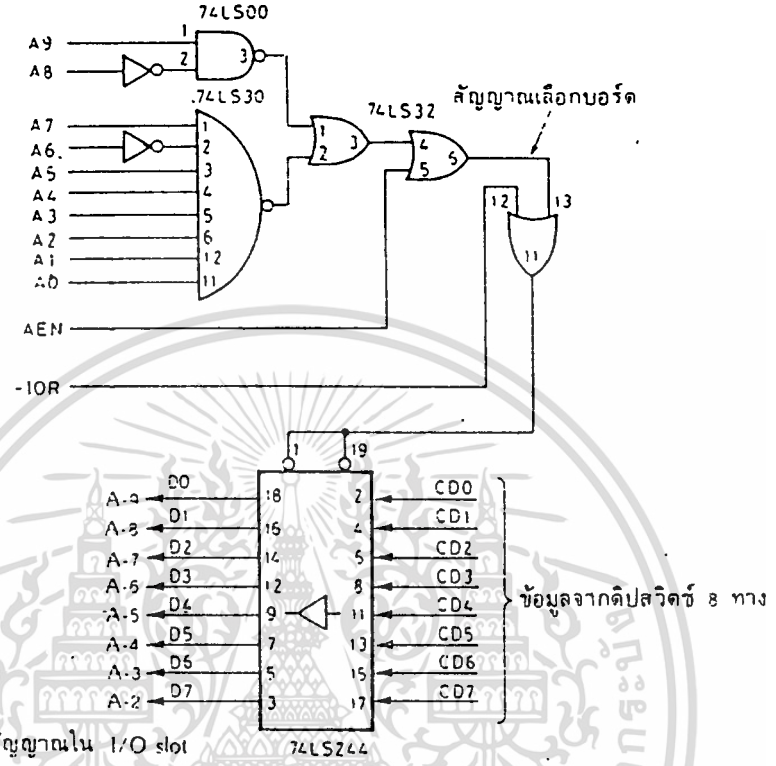


รูปที่ 2.27 รายละเอียดการต่อวงจรเอาท์พุทแลตช์กับ LED 8 ดวง

3.วงจรอินพุทบัฟเฟอร์ (Input Buffer Circuit)

การนำข้อมูลเข้าไอซี #74LS244 เป็นบัฟเฟอร์ของข้อมูลการนำเข้าก็ต้องมีการอีนาเบิลเพื่อนำข้อมูลเข้าทางสายข้อมูลเช่นเดียวกัน อย่างไรก็ตามก็ต้องมีสายอีกเส้นมาเกี่ยวข้องเช่นเดียวกล่าวคือ -IOR (Input/Output Read) สายสัญญาณนี้มีลอจิก "0" เมื่อมีการใช้คำสั่ง A = INP (Address) ในภาษาเบสิกดังนั้นจำเป็นต้องเชื่อมต่อสายสัญญาณนี้เข้ากับวงจรอีนาเบิล รายละเอียดดังแสดงในรูปที่ 2.28

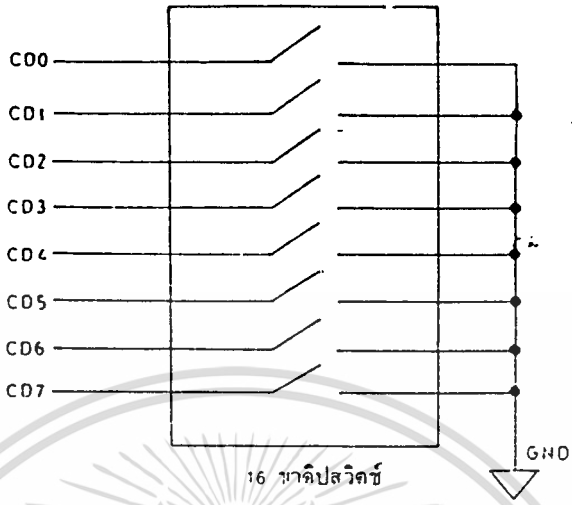
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 รายละเอียดของวงจรอินพุทบัฟเฟอร์

จากรูปที่ 2.28 ที่ตำแหน่ง CD_0-CD_7 ทำการเชื่อมต่อกับดิปสวิทช์ 8 ทาง (16 ขา) เพื่อทำการสับสวิทช์ ปิด-เปิด เป็นข้อมูลเข้าของแต่ละบิท (D_0-D_7) แสดงการต่อดังรูปที่ 2.29 คือเมื่อสวิทช์เปิด (ON) ข้อมูลเข้าสู่ ไอซี #74LS244มีลอจิกเป็น "1" (เพราะเป็นไอซี TTL : Transister Transister Logic) แต่เมื่อสวิทช์ปิด (OFF) ข้อมูลที่เข้าจะเป็นลอจิก "0" ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 รายละเอียดการต่อสวิทช์ 8 ทางกับวงจรอินพุทบัฟเฟอร์

2.3.ทฤษฎีส่วนบันทึกสัญญาณเสียง

ในส่วนนี้เราสามารถแบ่งได้เป็น 2 ส่วน คือ

DIGITAL TO ANALOG CONVERTER (DAC)

ANALOG TO DIGITAL CONVERTER (ADC)

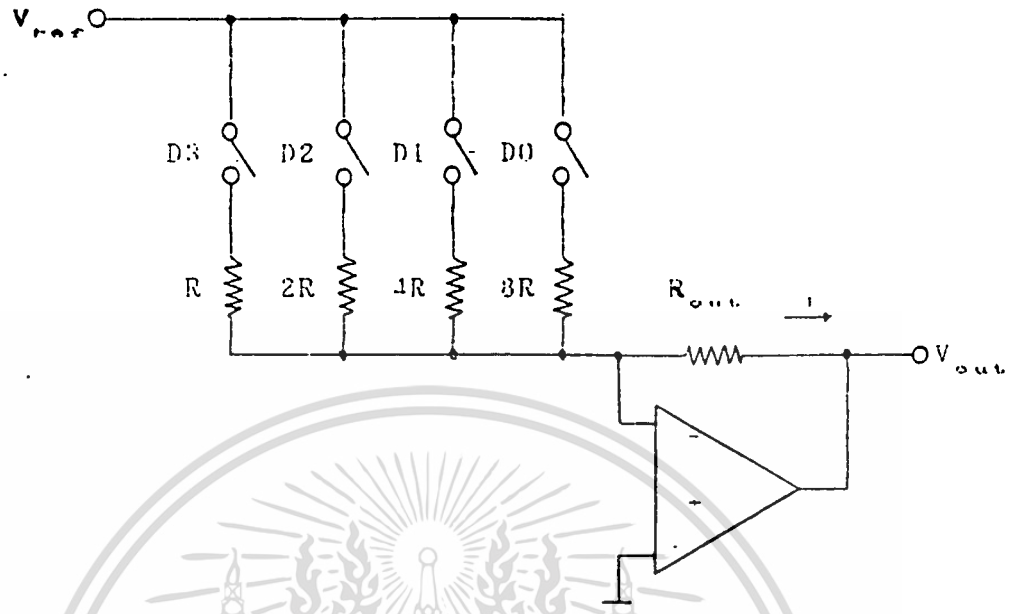
2.3.1.DIGITAL TO ANALOG CONVERTER (DAC)

DIGITAL TO ANALOG CONVERTER เป็นการเปลี่ยนสัญญาณดิจิตอลให้เป็นสัญญาณอนาล็อกซึ่งมีวิธีทำได้หลายวิธี

1.Binary-Weighted Resistance D/A Converter

หลักการของ Digital to Analog Converter แบบ Weighted Resistor D/A นั้น อาศัยหลักการที่ว่ากระแส output เป็นผลรวมของกระแส input โดยการเลือกค่าความต้านทาน input ให้เพิ่มขึ้นตามหลักของเลข binary ซึ่งมีลักษณะของวงจรดังรูปที่ 2.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 D/A Converter with Binary-Weighted Resistance

จากวงจรจะเห็นว่าประกอบด้วย Reference Voltage (V_{Ref}), Weighted Resistive Network, Switch and Summing Element เมื่ออินพุททุกบิทเป็น High หรือเมื่อสวิทช์ทุกตัวถูกปิด จะได้กระแสอินพุทเป็นดังนี้คือ

$$I_3 = V_{Ref}/R$$

$$I_2 = V_{Ref}/2R$$

$$I_1 = V_{Ref}/4R$$

$$I_0 = V_{Ref}/8R$$

เมื่อกระแสเอาท์พุท คือ ผลรวมของกระแสอินพุททั้งหมด

$$I = V_{Ref}/R\ (1+0.5+0.25+0.125) \qquad (1)$$

$$I = 1.875\ VRef/R$$

และถ้าให้ "0" แทนการเปิดสวิทช์ (Open) และ "1" แทนการเปิดสวิทช์ (close) เราจะสามารถเขียนสมการได้ใหม่ดังนี้คือ

$$I = V_{Ref}/R\ (D_3+0.5D_2+0.25D_1+0.125D_0) \qquad (2)$$

และสามารถเขียนสมการใหม่ได้เป็นให้อยู่ในรูปของกำลัง 2 ได้ดังนี้

$$I = V_{Ref}/R\ (D_3+2^{-1}D_2+2^{-2}D_1+2^{-3}D_0) \qquad (3)$$

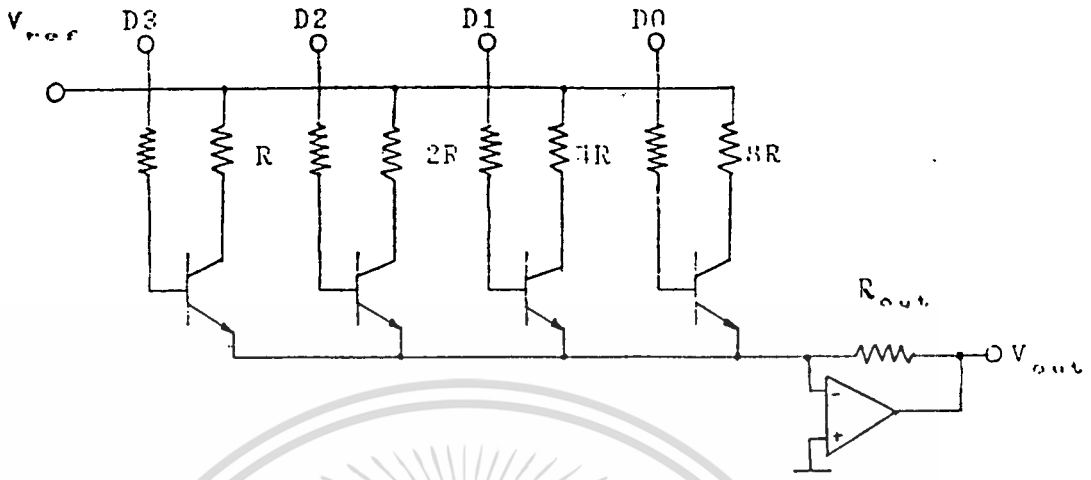
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวมาแล้วว่ากระแสเอาท์พุทคือผลรวมของกระแสอินพุทตามหลักของ Binary Weigth ดังนั้นถ้าเราสมมุติให้ VRef = 5V และ R = 5 Kohm เราก็จะได้กระแสเอาท์พุทที่เปลี่ยนแปลงจาก 0 ถึง 1.875 mA ดังแสดงในตารางที่ 2.2

D_3 D_2 D_1 D_0	Output Current (mA)	Fraction of Maximum
0 0 0 0	0	0
0 0 0 1	0.125	1/15
0 0 1 0	0.25	2/15
0 0 1 1	0.375	3/15
0 1 0 0	0.5	4/15
0 1 0 1	0.625	5/15
0 1 1 0	0.75	6/15
0 1 1 1	0.875	7/15
1 0 0 0	1	8/15
1 0 0 1	1.125	9/15
1 0 1 0	1.25	10/15
1 0 1 1	1.375	11/15
1 1 0 0	1.5	12/15
1 1 0 1	1.625	13/15
1 1 1 0	1.75	14/15
1 1 1 1	1.875	15/15

ตารางที่ 2.2 แสดงค่ากระแสเอาท์พุทของ Binary-Weighted Resistance D/A Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

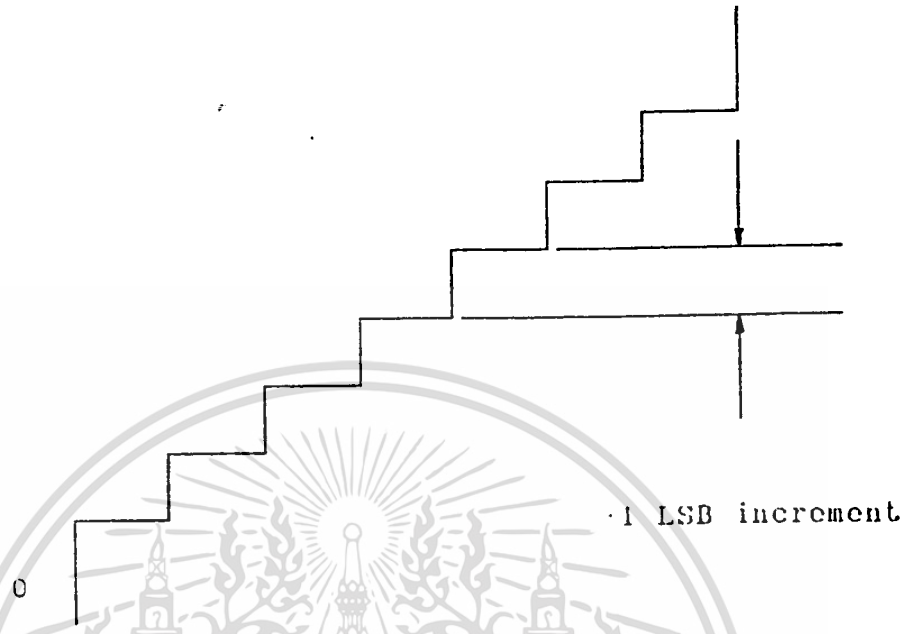


รูปที่ 2.31 Transister Switch For D/A Converter

รูปที่ 2.31 แสดงการใช้ทรานซิสเตอร์เป็นสวิทช์ข้อมูลจาก D_3-D_0 ซึ่งจะไปขับเบสของทรานซิสเตอร์ เมื่อบิทหนึ่งบิทใดเป็น High ก็จะทำให้มีกระแสเบสเพียงพอที่จะใช้ทรานซิสเตอร์ Saturate และเมื่อเป็น Low ก็จะสามารถทำให้ทรานซิสเตอร์ Cut off ได้ ดังนั้นเนื่องจากทรานซิสเตอร์แต่ละตัวสามารถ Saturate หรือ Cut off ได้ จึงทำให้มีลักษณะเหมือนกับการปิดหรือเปิดสวิทช์นั่นเอง

จากวงจรในรูปที่ 2.31 ถ้าเราป้อนอินพุท D_3-D_0 ด้วยเลข binary ค่าตั้งแต่ 0000 ถึง 1111 ก็จะทำให้การเปลี่ยนแปลงของข้อมูลจากดิจิตอลเป็นอนาล็อกมีลักษณะเป็นรูปขั้นบันได หรือที่เรียกว่า Stair case ดังนั้นถ้าความต้านทานทั้งหมดมีค่าถูกต้อง และทรานซิสเตอร์แมทช์กันแล้วขั้นบันไดทุกขั้นก็จะมีขนาดเท่ากัน ดังในรูปที่ 2.32 ซึ่งขั้นบันไดหนึ่งขั้นเราเรียกว่า LSB increment เพราะเกิดจากการเปลี่ยน LSB หนึ่งครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 แสดง Stair Case Output Current

วิธีหนึ่งที่จะวัดคุณภาพของ D/A Converter ก็คือ ค่ารีโซลูชั่น (Resolution) ซึ่งค่ารีโซลูชั่นนี้คือ อัตราส่วนของ LSB increment ต่อค่าเอาท์พุทสูงสุดมีสูตรดังนี้คือ

$$\text{Resolution} = 1/(2^n-1)$$

ตัวอย่างเช่น ถ้าเป็น 4 บิท converter ก็จะมีค่า Resolution เป็น

$$\text{Resolution} = 1/(2^n-1)$$
$$= 1/(2^4-1)$$
$$= 1/15$$

เมื่อ n คือจำนวนบิทของดิจิตอลอินพุท

ดังนั้นค่า Percent Resolution

$$= (1/15)*100$$
$$= 6.67\ \%$$

ถ้าจำนวนบิทของอินพุทมากขึ้นก็จะทำให้ค่า Resolution ดีขึ้น คือ จำนวนขั้นบันไดจะมากขึ้น ซึ่งทำให้ convert ละเอียดขึ้น ดังตัวอย่างในตารางที่ 2.3 ใน D/A Converterค่าความเที่ยงตรงเทียบได้กับการปิดกระแสเอาท์พุทแต่ละตัวซึ่งเป็นค่าทางความคิดในรูปที่ 2.31 ค่า

เอกสารนี้เป็น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเที่ยงตรงขึ้นอยู่กับโวลท์เต็จอ้างอิง ,ค่าความคลาดเคลื่อนของตัวต้านทานและทรานซิสเตอร์ ปกติในการใช้งานจะมีตัวปรับความละเอียด เพื่อตั้งฟูลสเกลของเอาท์พุท

ความเที่ยงตรงสัมพันธ์เทียบได้กับการปิด (close) ของสวิทช์กับระดับ เอาท์พุท ซึ่งเป็นสัดส่วนทางความคิดของฟูลสเกลเอาท์พุท การเปลี่ยนแปลงของสวิทช์ทั้ง 4 บิท จะมีระดับของเอาท์พุททางความคิดเป็นสัดส่วนกับฟูลสเกลดังนี้คือ 0, 1/15, 2/15, 3/15 เป็นต้น เนื่องจากข้อมูลทั้งหลายนี้ เป็นตัวระบุถึงความเที่ยงตรงสัมพันธ์กับความเที่ยงตรงที่แท้จริง ในการพิจารณาความเที่ยงตรงสัมพันธ์ ดังนั้นความเที่ยงตรงสัมพันธ์ จะขึ้นอยู่กับปัจจัยหลักใหญ่ๆ คือค่าความคลาดเคลื่อนของ Weight resistor คือ R, 2R, 4R และ 8R

ข้อเสียของ Weighted Resistance D/A Converterก็คือจะเกิดความคลาดเคลื่อนของค่าความต้านทานที่มากกว่าค่า Percent Resolution เมื่อค่า Resolution คือค่าอัตราส่วนของ LSB increment ต่อเอาท์พุทสูงสุด ตัวอย่างเช่น Resolution ของ 4 convert คือ 1/15 ค่า Percent Resolution = 6.68 % ดังนั้นค่าความต้านทานที่ใช้ต้องคลาดเคลื่อนน้อยกว่า 6.68 % จึงจะทำให้เกิดขั้นบันไดแบบ Monotonic คือขั้นบันไดแต่ละขั้นมีขนาดเท่ากัน

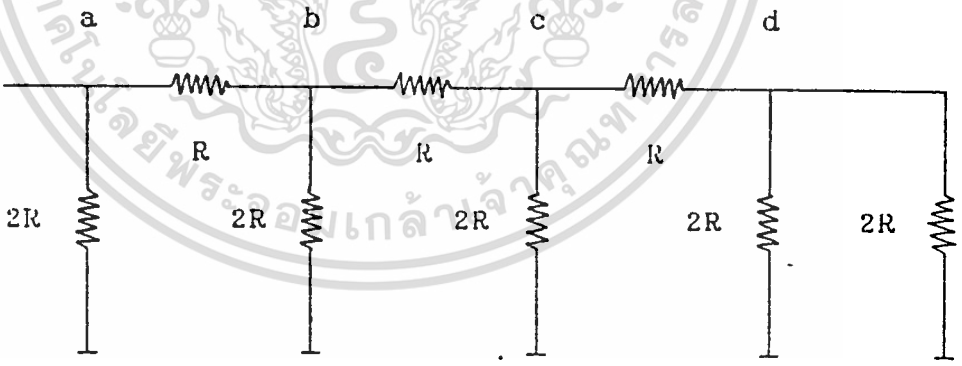
ความยุ่งยากอีกอย่างหนึ่งที่เกิดขึ้นก็คือ เมื่อจำนวนบิทของดิจิตอลอินพุทเพิ่มมากขึ้นก็จะทำให้ย่านของค่าความต้านทานเพิ่มขึ้นเป็น R ,2R ,4R ,8R เท่าของความต้านทานตัวที่น้อยที่สุด และถ้าเป็น 12 บิท ความต้านทานตัวที่มีค่ามากที่สุดจะเป็น 2048 เท่าของตัวที่เล็กที่สุด ดังนั้นเนื่องจากปัญหาความคลาดเคลื่อนของตัวต้านทานและย่านของของความต้านทานที่หาค่ายากจึงทำให้ Weight Resister D/A Converter ไม่เป็นที่นิยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bits	Resolution	Percent
4	1 part in 15	6.67
6	1 part in 63	1.59
8	1 part in 255	0.392
10	1 part in 1,023	0.0978
12	1 part in 4,095	0.0244
14	1 part in 16,383	0.0061
16	1 part in 65,535	0.00153
18	1 part in 262,143	0.000381

ตารางที่ 2.3 แสดง Resolution

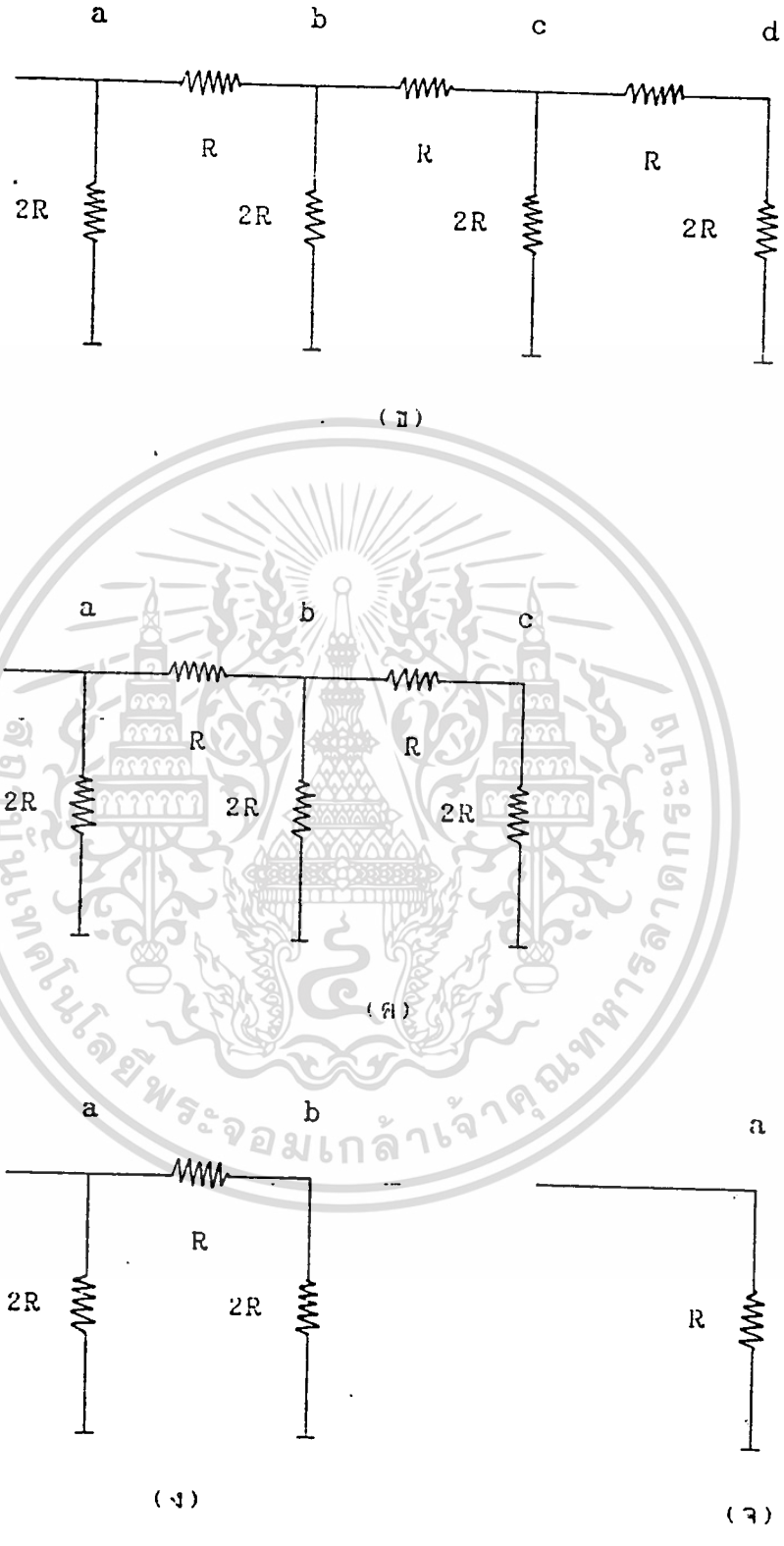
2.Ladder Method D/A Converter



(ก)

รูปที่ 2.33 แสดงวงจร R-2R Ladder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

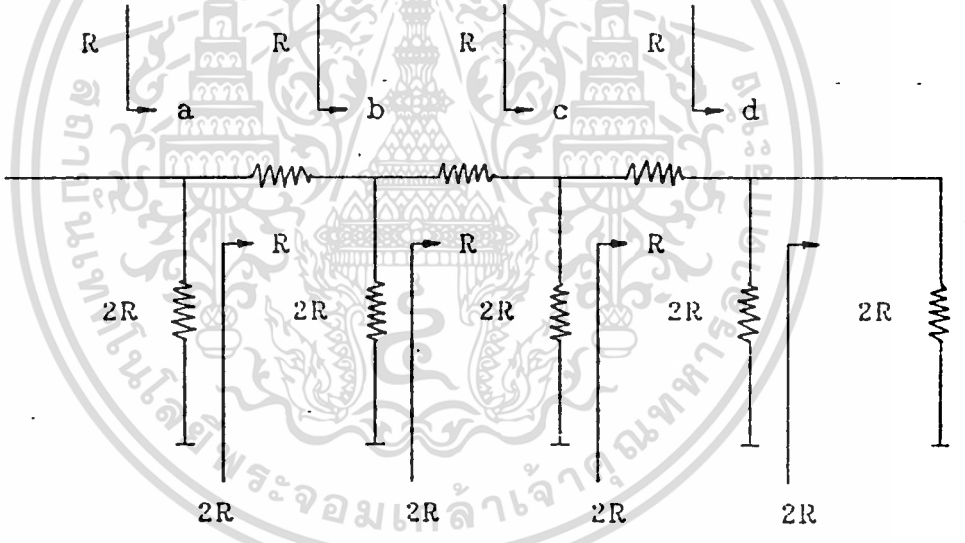


รูปที่ 2.33(ต่อ) แสดงวงจร R-2R Ladder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

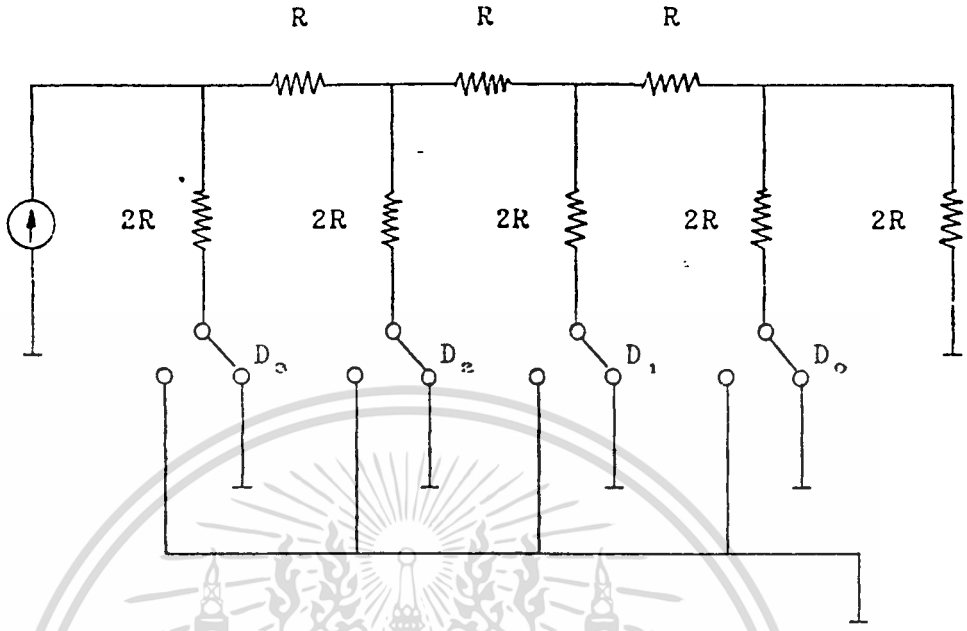
วิธีที่จะแก้ปัญหาของ binary-weighted ก็คือใช้วงจร Ladder แสดงในรูปที่ 2.33 ลักษณะวงจรจะประกอบด้วยวงจร R-2R Ladder ซึ่งใช้ความต้านทานเพียง 2 ค่า จึงตัดปัญหาเรื่องย่านของความต้านทาน เพราะสามารถใช้ความต้านทานที่ทำจากชิพเดียวได้ จึงแก้ปัญหาในเรื่องความคลาดเคลื่อนของตัวต้านทานได้ในวงจร R-2R Ladder นี้ อิมพีแดนซ์ที่จุดต่างๆ ในวงจรจะเริ่มที่ความต้านทานสองตัว คือที่จุด D จากรูปที่ 2.33(ก) จะได้ 2R กับ 2R ขนานกัน เหลือ R ดังในรูปที่ 2.33(ข) และในรูปที่ 2.33(ข) นี้ที่จุด C มี R อนุกรมกับ R รวมเป็น 2R ฉะนั้นที่จุด C จะมี 2R ขนานกับ 2R อีก ดังนั้นวงจรเมื่อยุบหมดแล้วจะได้วงจรดังรูปที่ 2.33(จ)

รูปที่ 2.34 ผลรวมของ ladder อิมพีแดนซ์เมื่อมองเข้าไปด้านซ้ายของจุดใดจุดหนึ่งเราจะเห็น R เสมอ และถ้าดูทางด้านขวาเราก็จะเห็น 2R ปรากฏการณ์ทางอิมพีแดนซ์เช่นนี้เป็นหัวใจในการวิเคราะห์ D/A Converter แบบใหม่



รูปที่ 2.34 แสดงถึง Ladder Impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 D/A Conversion with R-2R Ladder

จากรูปที่ 2.35 แสดงวงจร Ladder ซึ่งปกติ D/A Converter จะมีกระแสอ้างอิงที่กำหนดให้โดยผู้ใช้จากวงจรสมมติให้กระแสอ้างอิงเท่ากับ 2 mA ดังนั้นที่ทางขวาของจุด A เราจะเห็น 2R กระแสอินพุท 2 mA จะถูกแบ่งออกเท่าๆกันที่จุด A ในทำนองเดียวกันที่จุด B มี 2R ขนานกับ 2R กระแสก็จะถูกแบ่งออกเป็นสองส่วนเท่าๆ กันคือ 0.5 mA และกระแสจะถูกแบ่งเช่นนี้เรื่อยไปจึงทำให้เราได้กระแสที่ไหลจากกราวด์บนคือ 1,0.5,0.25 และ 0.125 mA ดังนั้นจะได้กระแสเอาท์พุทเป็น

$$I_{out} = (D_3 + 2^{-1}D_2 + 2^{-2}D_1 + 2^{-3}D_0)I_{ref}/2 \qquad ---(4)$$

ซึ่งกระแสเอาท์พุทของ 4 บิท Ladder จะมีค่าตั้งแต่ 0 ถึง 15/16 I_{ref} และถ้ามีอินพุทมากขึ้นเราก็จะได้กระแสเป็น

$$I_{out} = (D_{n-1} + 2^{-1}D_{n-2} + \ldots + 2^{1-n}D_0)I_{ref}/2 \qquad ---(5)$$

และถ้าพิจารณาในรูปของแรงดันก็จะได้ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_{out} = -V_{Ref} (D_{n-1}2^{-1} + D_{n-2}2^{-2} + \dots +$$

$$D_1 2^{-(n-1)} + D_0 2^{-n}) \qquad ---(6)$$

หรือ

$$V_{out} = -V_{Ref}2^{-n} (D_{n-1}2^{n-1} + D_{n-2}2^{n-2} + \dots +$$

$$D_1 2^{1} + D_0 2^{0}) \qquad ---(7)$$

2.3.2.ANALOG TO DIGITAL CONVERTER (ADC)

1.The Counter Methode of A/D Conversion

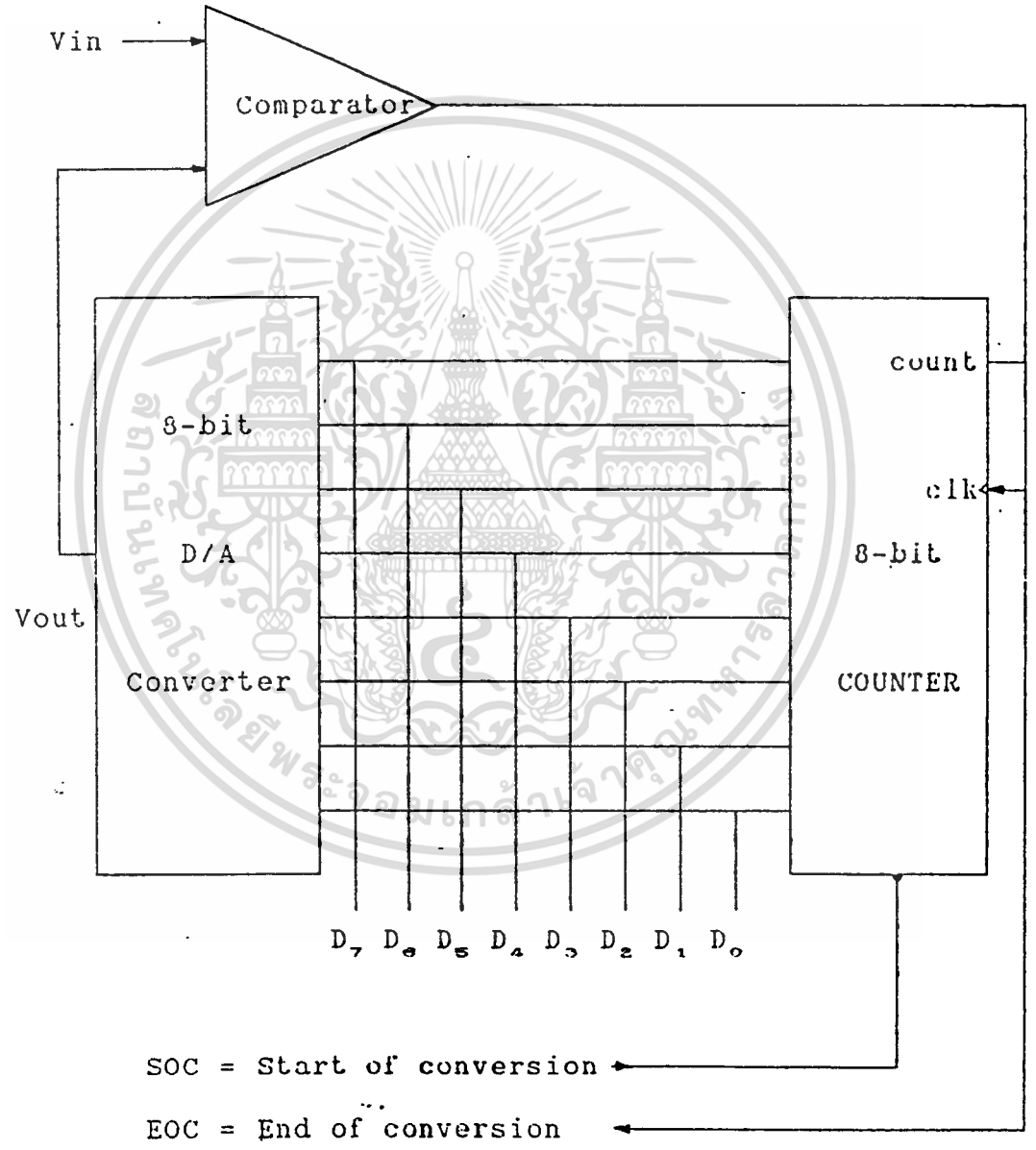
จากรูปที่ 2.36 แสดงวิธีการของ A/D Convert ที่ง่ายแต่ไม่ค่อยนิยมใช้กันเพราะว่ามีความคลาดเคลื่อนมาก จากวงจรจะว่ามี V_{in} เป็นอนาล็อกอินพุทโวลท์เต็จ D_7 ถึง D_0 เป็นดิจิตอลเอาท์พุทซึ่งหลักการของวิธี Counter Methode คือ จะนำเอาค่าที่เคาท์เตอร์นับได้นั้นไปป้อนเข้าวงจร D/A จากนั้นวงจร D/A ก็จะเปลี่ยนค่า binary ที่นับได้นั้นให้เป็นค่าอนาล็อกแล้วจึงนำไปเปรียบเทียบกับค่าโวลท์เต็จอินพุทรับเข้ามา ถ้าเท่ากันวงจรเคาท์เตอร์ก็จะหยุดนับ ซึ่งจะทำให้ได้ค่าอนาล็อกที่ป้อนเข้ามาแต่ยังไม่เท่าวงจรเคาท์เตอร์ก็จะนับเพิ่มขึ้นอีกหนึ่ง แล้วเปลี่ยนให้เป็นอนาล็อก จากนั้นจึงส่งไปเปรียบเทียบกับอนาล็อกใหม่จนกว่าจะเท่ากันจึงหยุดนับและจะได้ค่าดิจิตอลที่มีค่าเท่ากับค่าอนาล็อกที่ป้อนเข้ามา

การทำงานของ A/D Conversion แบบ Counter Methode มีลักษณะดังนี้ คือเริ่มแรกเมื่อ Start Pulse เป็น low ก็จะทำให้เคาท์เตอร์ถูกเคลียร์และเมื่อ Start Pulse เป็น high เคาท์เตอร์ก็พร้อมจะนับ ดังนั้น V_{out} ซึ่งก็คือเอาท์พุทของ D/A ก็จะเป็นศูนย์ด้วย และเมื่อเปรียบเทียบกับอนาล็อกอินพุทแล้ว ถ้าผลการเปรียบเทียบไม่เท่ากันเอาท์พุทของ OP-AMP ก็จะเป็น high และขาก็จะเป็น high และขาก็จะเป็น Count เป็น high ด้วย จึงทำให้เคาท์เตอร์เริ่มนับโดยจะนับขึ้นจากศูนย์ และถูกเปลี่ยนให้เป็นค่าอนาล็อกโดย D/A Converter หลังจากนั้นจะถูกส่งไปเปรียบเทียบกับอนาล็อกอินพุท ตราบใดที่ V_{in} ยังมากกว่า V_{out} เอาท์พุทของ OP-AMP ก็ยังเป็น high และเมื่อ clock ลูกต่อไปเข้ามาเคาท์เตอร์ก็จะนับเพิ่มขึ้นอีกหนึ่งจากจะถูกเปลี่ยนให้เป็นค่าอนาล็อกและเปรียบเทียบกับอนาล็อกอินพุทเป็นเช่นนี้เรื่อยๆ ไปจนกว่าจะได้ค่า V_{out} ที่เท่ากับหรือสูงกว่า V_{in} ก็จะทำให้ขา count กลายเป็น low จึงทำให้เคาท์เตอร์หยุดนับตอนนี้เอาท์พุท D_7-D_0 ก็จะเป็นค่าดิจิตอลที่เท่ากับค่าของอนาล็อกอินพุทและขอบที่ตกลงของขา count ก็จะใช้เป็นสัญญาณ End of Conversion ถึงตอนนี้การเปลี่ยนก็จะสิ้นสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าอนาล็อกอินพุทหรือ V_{in} เปลี่ยนค่าใหม่วงจรภายนอกก็จะต้องส่ง Start Pulse ลูกใหม่เข้ามาที่ขา Start of conversion เพื่อเคลียร์การนับ แล้วจึงเริ่มรอบการเปลี่ยนใหม่อีกที

ข้อเสียที่สำคัญของ A/D แบบ Count Methode ก็คือความเร็วในการเปลี่ยนช้า และเวลาใช้ในการเปลี่ยนค่าอนาล็อกแต่ละค่าให้เป็นค่าดิจิตอลจะไม่เท่ากัน กล่าวคือถ้าค่าอนาล็อกมีค่าน้อยก็จะใช้เวลาในการเปลี่ยนน้อยและถ้าค่าอนาล็อกมีค่ามากก็จะใช้เวลาในการเปลี่ยนมาก



รูปที่2.36 แสดง A/D Conversion with Counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.Successive Approximasion A/D Conversion

A/D Conversion แบบที่นิยมใช้มากที่สุดคือแบบ Successive Approximasion ซึ่งมีหลักการทำงานดังนี้คือเริ่มแรกก่อนที่เอาท์พุทของ D/A Convers ถูกส่งไปเปรียบเทียบกับอนาล็อกอินพุทที่ OP-AMP คอมพาราเตอร์นั้น ความแตกต่างเป็นเท่าไหร่ขึ้นอยู่กับรีจิสเตอร์ SAR จะป้อนข้อมูลดิจิตอลออกมาถ้าไม่เท่ากับค่าอนาล็อกอินพุทของ OP-AMP ก็จะส่งผลไปยัง control เพื่อให้ไป set ค่าในรีจิสเตอร์ SAR ใหม่ แล้วจึงเปลี่ยนให้เป็นอนาล็อกจากนั้นจึงส่งไปเปรียบเทียบกับอนาล็อกอินพุทที่ OP-AMP คอมพาราเตอร์ จนกว่าจะเท่ากันจึงจะหยุด เมื่อการเปลี่ยนแปลงสิ้นสุดลงข้อมูลดิจิตอลที่ได้นั้นจะถูกส่งมาอยู่ที่เอาท์พุทบัฟเฟอร์รีจิสเตอร์

การทำงานของ A/D Conversion แบบ Successive Approxmasion คือ เมื่อสัญญาณ Start of Conversion เป็น low รีจิสเตอร์ SAR จะถูกเคลียร์ให้เป็นศูนย์และเมื่อสัญญาณ Start of conversion เป็น high การเปลี่ยนของ A/D แบบ Successive Approximasion Methode จึงจะเริ่มขึ้นโดยการ set บิท MSB ของรีจิสเตอร์ SAR หรืออีกนัยหนึ่งก็คือ ระหว่างสัญญาณลูกแรกวงจรคอนโทรลโหลด High MSB เข้าไปในรีจิสเตอร์ SAR ซึ่งเอาท์พุทของรีจิสเตอร์ SAR จะมีค่าเป็น 10000000 ดังนั้นค่าของดิจิตอลเอาท์พุท V_{out} จึงกระโดดไปเป็น 128/255 เท่าของค่าฟูลสเกลถ้า V_{out} มากกว่า V_{in} สัญญาณที่ได้จากการเปรียบเทียบจะเป็นลบ วงจรคอนโทรลจะ reset MSB ในทางกลับกันถ้า V_{out} น้อยกว่า V_{in} การเปรียบเทียบจะได้เอาท์พุทเป็นบวก จึงทำให้ MSB ถูก Set สมมติว่า MSB ไม่ถูก Reset รีจิสเตอร์ SAR ก็จะมีข้อมูลเป็น 10000000 และเมื่อสัญญาณนาฬิกาลูกต่อไปเข้ามาก็จะ set บิท D_6 ของรีจิสเตอร์ SAR ดังนั้นดิจิตอลเอาท์พุทจึงมีค่าเป็น 11000000 ตอนนี้ค่าของ V_{out} เป็น 192/255 เท่าของค่าฟูลสเกลถ้า V_{out} มากกว่า V_{in} เอาท์พุทของ OP-AMP จะเป็นลบทำให้ D_6 ถูก Reset แต่ถ้า V_{out} ยังน้อยกว่า V_{in} บิท D_6 ก็จะยังคงถูก Set ต่อไป การทำงานจะเป็นเช่นนี้เรื่อยไป คือบิททั้งหมดจะถูก Set และ Reset เพื่อนำไปทดสอบ ถ้า V_{out} มากกว่า V_{in} บิทนั้นจะถูก Reset ด้วยวิธีนี้ทุกบิทจะถูกเซต,ทดสอบและรีเซทถ้าจำเป็นด้วยการทำงานของวงจรที่เร็ว การเปลี่ยนจะสิ้นสุดหลังสัญญาณนาฬิกาผ่านไป 8 ลูก

เมื่อการเปลี่ยนสิ้นสุดลงวงจรคอนโทรลจะส่งสัญญาณEnd of Conversion ออกไปและขอบที่ตกลงของสัญญาณนี้จะเปิดบัฟเฟอร์รีจิสเตอร์ เพื่อเปิดบัฟเฟอร์รีจิสเตอร์ให้ข้อมูลดิจิตอลสามารถเข้าไปในบัฟเฟอร์รีจิสเตอร์ได้ และค่าดิจิตอลที่ได้นี้ก็จะคงอยู่จนกระทั่งเราเริ่มรอบการเปลี่ยนใหม่

ข้อดีประการสำคัญของ Successive Approximasion คือความเร็วซึ่งจะใช้เวลาเพียง n Clock Pulse เพื่อทำ n-Bit Resolution ของสัญญาณอนาล็อก ซึ่งเป็นสิ่งที่พัฒนาให้เหนือกว่าแบบ Counter Methode

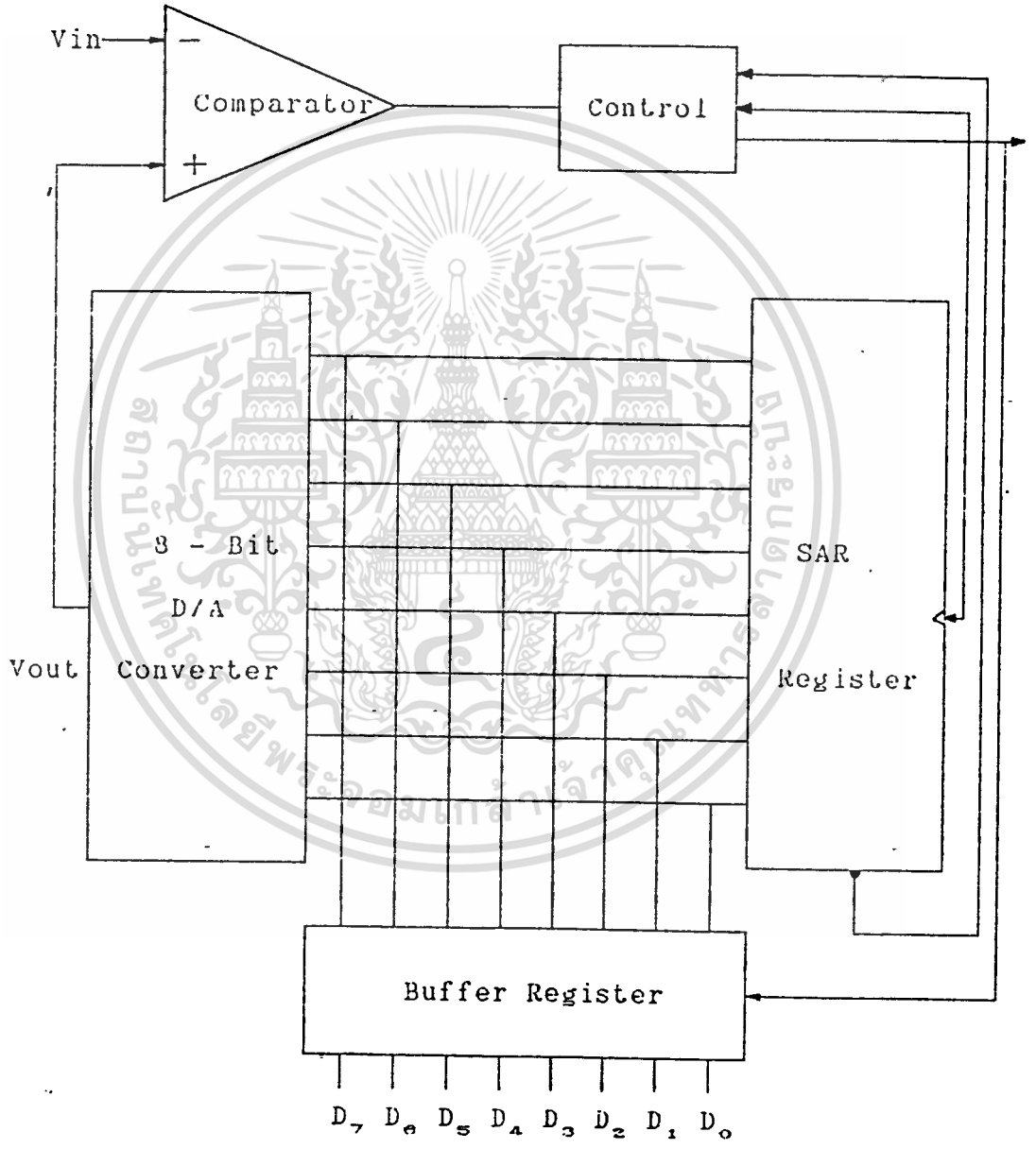
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLK = Clock

EOC = End of conversion

SOC = Start of conversion

SAR = Successive approximasion register



รูปที่ 2.38 แสดง A/D Conversion by Successive Approximasion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

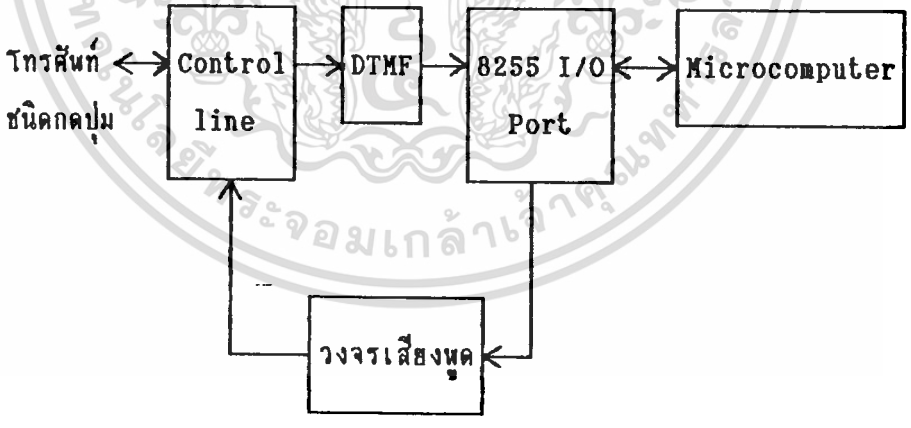
บทที่ 3

การออกแบบ

ในบทนี้จะกล่าวถึงการออกแบบ ซึ่งเป็นการออกเกี่ยวกับระบบของฮาร์ดแวร์และการเขียนโปรแกรมในส่วนฮาร์ดแวร์ของระบบแจ้งผลการสอบคัดเลือกนั้น จะประกอบด้วยส่วนต่างๆ ซึ่งจะแบ่งออกเป็น 4 ส่วนใหญ่ดังนี้คือ

1. ส่วนควบคุมการติดทางโทรศัพท์
2. ถอดรหัส DTMF (DTMF Decoder)
3. ส่วนของการบันทึกสัญญาณเสียง
4. ส่วนของการควบคุม (8255 I/O Port)

ส่วนประกอบต่างๆของระบบสามารถแสดงโดย บล็อกไดอะแกรม ข้างล่างนี้



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

ในระบบของฮาร์ดแวร์ดังกล่าว จะมีส่วนประกอบเพียง 4 ส่วนแรกเท่านั้นโดยจะทำงานสัมพันธ์กันอย่างใกล้ชิด โดยการใช้ซอฟต์แวร์จากไมโครคอมพิวเตอร์ซึ่งจะทำงานในลักษณะของกระบวนการคือ ตรวจจับสัญญาณเรียกเข้ายกหูโทรศัพท์แล้วถ่ายทอดสัญญาณเสียงถอดรหัสประจำตัวสอบ ตอบรับด้วยเสียงพูด จากนั้นก็วางหูโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

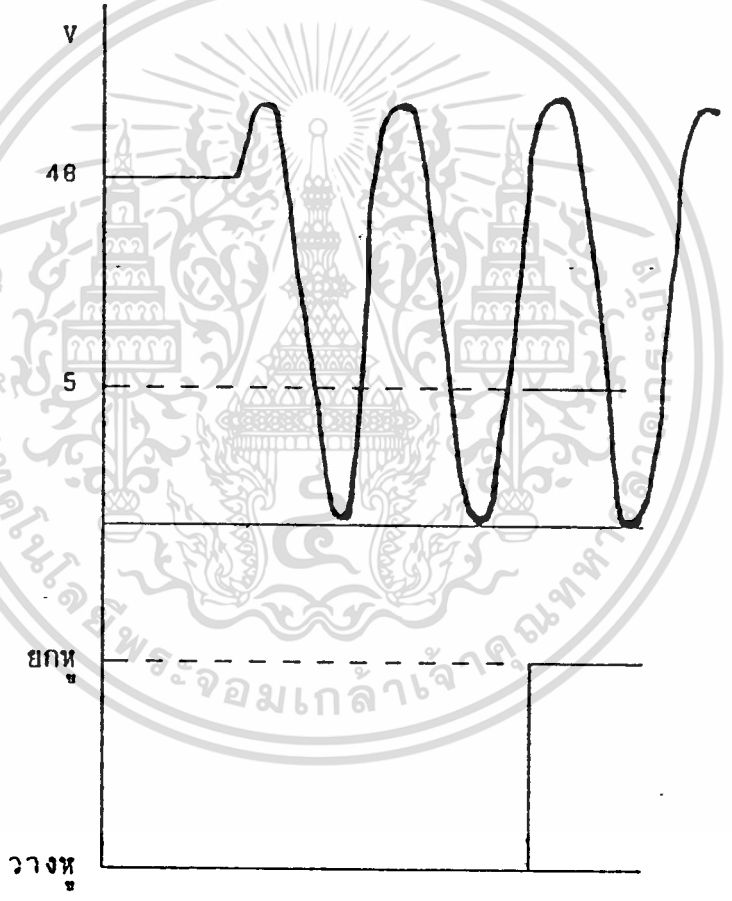
3.1.ส่วนควบคุมการติดต่อทางโทรศัพท์

ในส่วนนี้นับเป็นส่วนแรกและเป็นส่วนสำคัญของการที่สัญญาณจะถูกส่งผ่านเข้ามาในระบบของเครื่อง ในส่วนนี้จะแยกออกเป็น 2 ส่วนคือ

ส่วนการตรวจจับสัญญาณกระดิ่ง

ส่วนการยกหูและวางหู

3.1.1.ส่วนการตรวจจับสัญญาณกริ่ง



รูปที่ 3.2 สภาวะแรงดันไฟฟ้าของสายโทรศัพท์

การยกหูโทรศัพท์ เราอาศัยพื้นฐานทางสัญญาณที่ว่าในขณะสายว่างคู่สายโทรศัพท์จะมีแรงดันประมาณ 48 โวลท์ ซึ่งจ่ายมาจากชุมสายโทรศัพท์ และเมื่อผู้เรียกเรียกเข้ามาทางชุมสายจะส่งสัญญาณกระดิ่งมามีขนาดแรงดันเป็น 100 V_{rms} เป็นเวลา 1 วินาที และหยุดเป็นเวลา 4 วินาทีเป็นจังหวะแบบนี้ซึ่งแรงดันนี้ จะทำให้กระดิ่งภายในเครื่องโทรศัพท์ทำงานและทางชุมสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะรับทราบการยกหูโทรศัพท์จากการที่เรายกหูโทรศัพท์ซึ่งสวิทช์ภายในเครื่องโทรศัพท์ จะทำกี่ต่อคู่สายเข้ากับวงจรภายใน ซึ่งมีความต้านทานทางกระแสตรงต่ำก็จะเกิดการครบวงจรขึ้น ทำให้แรงดันไฟฟ้า 48 โวลท์ ลดลงเหลือ 5-10 โวลท์ เมื่อชุมสายรับรู้แล้วก็จะต่อคู่สายของเรากับผู้เรียกเข้าด้วยกัน และจากหลักการดังกล่าวส่วนของการตรวจสอบเราก็อาศัยช่วงสัญญาณกระดิ่ง กรณีที่มีผู้เรียกเข้ามา)มาทำการดีเท็ค (Detect) แล้วส่งสัญญาณไปยังไมโครคอมพิวเตอร์ว่ามีคนโทรเข้าชิ่งเครื่องไมโครคอมพิวเตอร์ก็จะส่งสัญญาณไปบังคับให้ส่วนควบคุมการยกหู ทำการยกหูเพื่อทำการติดต่อและถ่ายทอดสัญญาณระหว่างวงจรต่างๆ กับคู่สายโทรศัพท์ต่อไป

ในการออกแบบ เราออกแบบให้ตรวจจับสัญญาณกริ่งแทนการเฝ้ารอดอยสัญญาณกริ่งของพนักงาน ซึ่งในกรณีนี้ได้ใช้อุปกรณ์เชื่อมต่อทางแสง (Opto Couple) เพื่อกันการรบกวนจากสัญญาณต่างๆที่สอดแทรกเข้ามาระหว่างกลางทางซึ่งวงจรตรวจจับสัญญาณกริ่งแสดงไว้ในรูปที่ 3.3

3.1.2.ส่วนการยกหูและการวางหู

ในส่วนนี้การออกแบบให้ทำหน้าที่ยกหู หลังจากที่วงจรตรวจจับสัญญาณกริ่งตรวจพบสัญญาณกริ่ง และทำการวางหูเมื่อสิ้นสุดการสนทนา พร้อมกันนั้นวงจรนี้จะทำหน้าที่รับเอาสัญญาณความถี่ DTMF อีกด้วย

วงจรส่วนนี้จะประกอบด้วยรีเลย์ 2 ตัว ตัวหนึ่งจะทำหน้าที่ตัดต่อเสียงเข้าสู่สายของโทรศัพท์ และอีกตัวหนึ่งทำหน้าที่รับสัญญาณ DTMF รีเลย์ทั้ง 2 ตัวจะถูกขับด้วยทรานซิสเตอร์ซึ่งจะถูกควบคุมโดยส่วนควบคุมอีกทีหนึ่ง

การทำงานของวงจร

ในสภาวะที่ไม่มีสายภายนอกเข้ามา หรือไม่มีสัญญาณกริ่งสภาวะต่างๆ ของสัญญาณต่างๆ จะเป็นดังต่อไปนี้

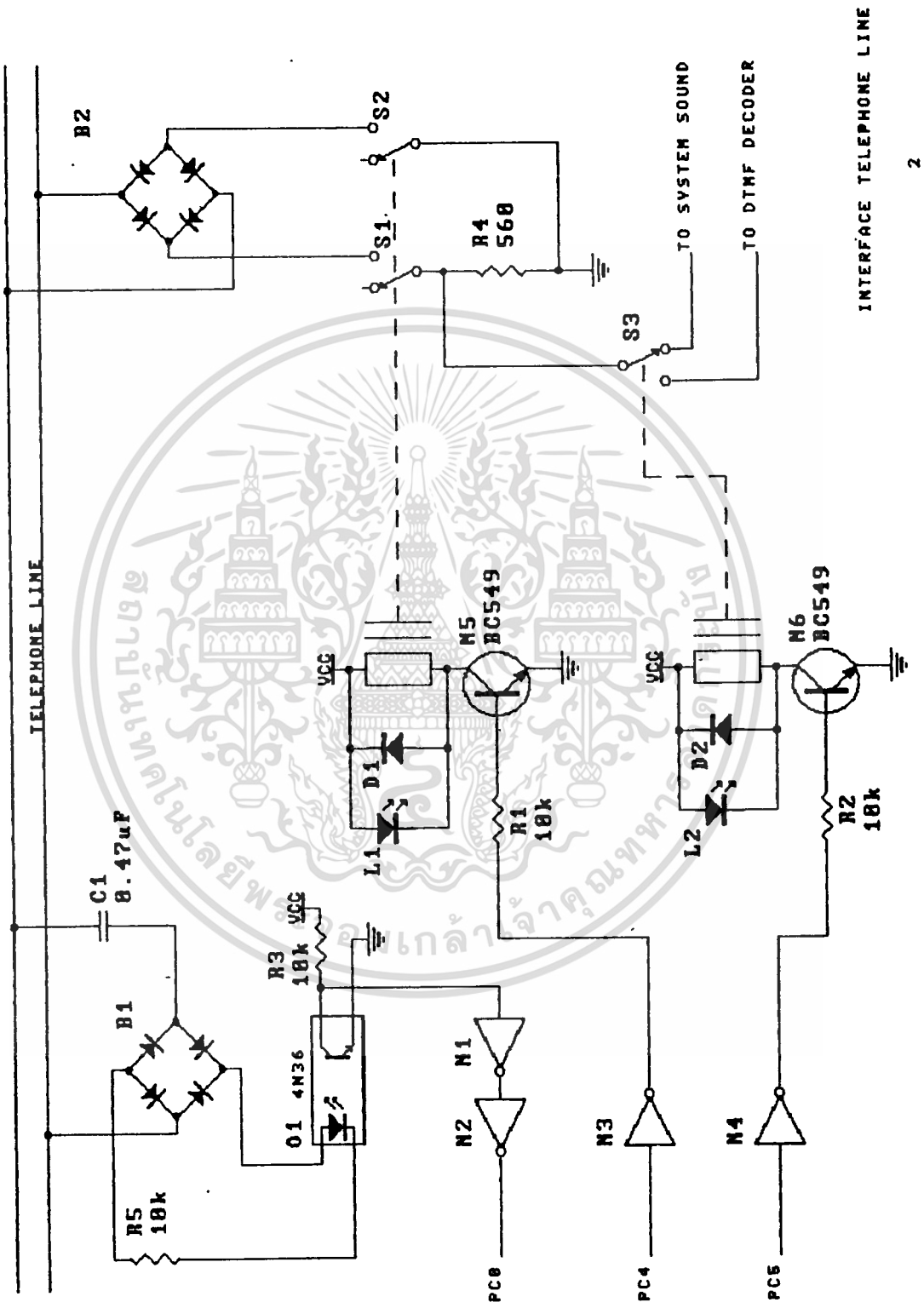
สัญญาณตรวจสอบการยกหูเป็น Logic "0" เพราะ Opto 4N36 off

สัญญาณหน่วงเวลา จะเป็น Logic "0" เพราะไม่มีการยกหู

สัญญาณตรวจสอบสาย ก็จะไม่มีเสียงเข้ามา

ในสภาวะที่มีสัญญาณเสียงกริ่งมาจากองค์การโทรศัพท์สัญญาณจะถ่ายทอดผ่านตัวเก็บประจุ 0.47 uF ผ่านวงจรแปลงไฟเป็นกระแสตรงไปเลี้ยงไดโอดเปล่งแสงภายใน Opto 4N36 โดยมีตัวต้านทาน 10 KOhm จำกัดขนาดกระแสไว้ทำให้โฟโต้ทรานซิสเตอร์นำกระแสสัญญาณ ตรวจสอบการยกหู จึงเป็น Logic "1" เมื่อวงจรไมโครโปรเซสเซอร์อ่านพบสัญญาณตรวจสอบการวางหูนี้ก็จะสั่งให้ทำการรับสายไว้และส่งสัญญาณไปควบคุมวงจรเสียงตอบรับด้วย นั่นก็คือเมื่อมีเสียงกริ่งเข้ามาวงจรจะทำงานตามขั้นตอนดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรตรวจสอบสัญญาณกริ่ง

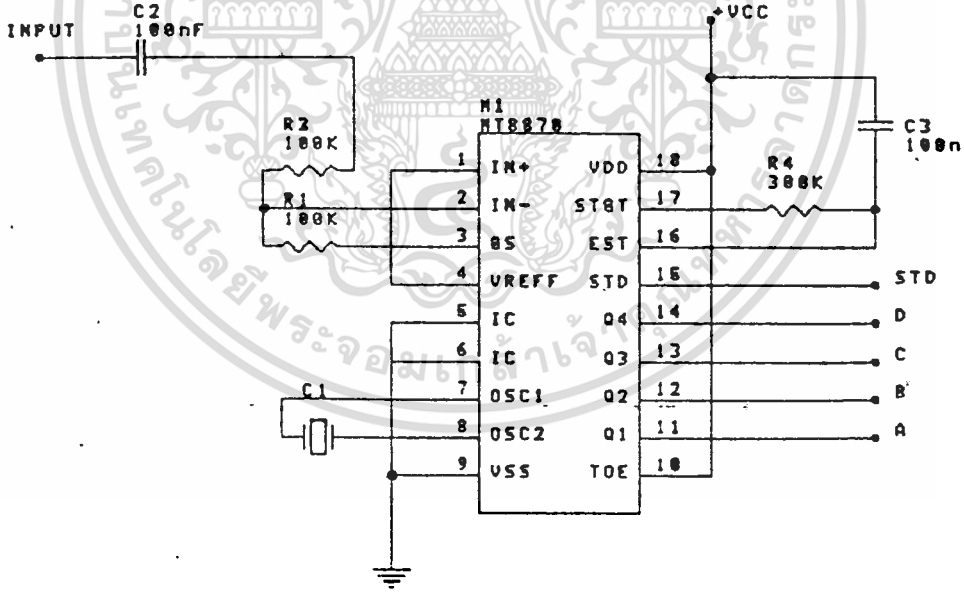
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรตรวจจับสัญญาณกริ่งทำงานที่เอาท์พุทของเครื่องโทรศัพท์เป็น "1"
วงจรไมโครจะทำการรับสายภายนอกไว้และส่งสัญญาณไปวงจรตอบรับ
วงจรไมโครรอรับสัญญาณการกดหมายเลขจากผู้เรียก

3.2.วงจรถอดรหัส DTMF (DTMF Decoder)

ในการออกแบบวงจรการถอดรหัส DTMF เนื่องจากเราเลือกใช้ไอซีสำเร็จรูปที่มีวงจรการต่อใช้งานอยู่แล้ว สำหรับวงจรถอดรหัส DTMF นั้น เมื่อผู้เรียกกดหมายเลขหลังจากได้รับการตอบรับเรียบร้อยแล้ว สัญญาณความถี่สองความถี่ผสมกันมาจะเข้าสู่วงจรถอดรหัส DTMF MT8870 เมื่อถอดรหัสแล้วจะได้เอาท์พุทออกมาเป็น 4 บิท แล้วส่งไปส่วนควบคุมต่อไป ซึ่งรายละเอียดของวงจรแสดงไว้ดังนี้

ในยุคก่อนการออกแบบวงจรถอดรหัสความถี่ของโทรศัพท์ มักใช้ไอซีจำพวกเฟสล็อกลูป ซึ่งสร้างปัญหาสารพัดไม่ว่าเรื่องของความถี่เปลี่ยนแปลง การปรับแต่งวงจร วงจรมีขนาดใหญ่ เพราะต้องใช้ไอซีจำนวนมาก



รูปที่ 3.4 วงจรถอดรหัส DTMF

ขั้นตอนการทำงาน

เมื่อผู้เรียกกดหมายเลขเข้ามา หลังจากได้รับการตอบรับแล้วจะมีสัญญาณสองความถี่(คือความถี่ต่ำและความถี่สูง) ผสมกันมาเข้าสู่วงจร DTMF Decoder เบอร์ MT8870 จะทำหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอดรหัสออกมาเป็นสัญญาณดิจิตอล 4 บิท ตัวคริสตอลที่ต่อในวงจรเพื่อเป็นตัวกำเนิดความถี่อ้างอิงเพื่อใช้ในการถอดรหัส DTMF สำหรับความถี่ที่ผลิตคือ 3.59 MH$_z$ สำหรับ R_1 และ R_2 ต่อไว้เพื่อทำหน้าที่เป็นอัตราการขยายทางด้านเอาท์พุทและเป็นตัวกำหนดอิมพีแดนซ์ โดยที่

อัตราการขยาย $Av = R_1 / R_2$

อินพุทอิมพีแดนซ์ $Z_{in} = 2 (R_1)^2 + (R_2)^2$

การตรวจสอบสัญญาณ ก่อนที่จะมีการถอดรหัสออกที่เอาท์พุท จะมีการตรวจความถี่ช่วงที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยการสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องมีความถี่ออกมาเป็นช่วงเวลาพอสมควรมิฉะนั้นวงจรนี้จะถือว่าสัญญาณไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใด สามารถตั้งได้โดยการใช้ RC ต่อภายนอกในที่นี้คือ R_4 และ C_3 สัญญาณที่ขา Est เป็นตัวตรวจสอบความถี่ที่เข้ามาว่าถูกต้องหรือไม่โดยสัญญาณที่ขา Est จะเป็นไฮใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามาทำให้ Vc สูงขึ้น ตัวเก็บประจุ C_3 จะคายประจุทำให้แรงดัน Vc สูงขึ้นจนถึงค่าเทรดโฮลด์ (Traeshold) วงจรถอดรหัสจึงทำการถอดรหัสออกเป็นตัวเลข 4 บิท

สำหรับการ์ดไทม์นั้นหมายถึงช่วงคาบเวลาของความถี่ที่เข้ามาซึ่งต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่เราตั้งไว้โดยการใช้ R_4 และ C_3 คือถ้ามีสัญญาณความถี่ที่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัส

ถ้าต้องการจะดูทางด้านคุณสมบัติและวงจรภายในจะแสดงไว้ในบทที่ 2 ในส่วนของทฤษฎี

3.3.ส่วนของการบันทึกสัญญาณเสียง

ส่วนนี้ได้แยกออกจากระบบของส่วนต่างๆ ทั้งสี่ซึ่งเป็นขั้นตอนหรือวิธีการในการที่จะเก็บข้อมูลจากสัญญาณเสียงอนาล็อก ให้เป็นข้อมูล ในส่วนนี้จะมีการทำงานแบ่งออกเป็น 2 ส่วนคือ

3.3.1.DIGITAL TO ANALOG CONVERTER (DAC)

การควบคุมอุปกรณ์หลายชนิดด้วยคอมพิวเตอร์ต้องอาศัยวงจรแปลงข้อมูลเป็นอนาล็อกเพื่อให้สามารถสั่งการได้อย่างถูกต้อง วงจรที่ใช้เป็นวงจรที่สร้างสัญญาณอนาล็อกที่แปรตามข้อมูลดิจิตอลที่ได้รับ เรียกว่าอุปกรณ์ DAC (Digital to Analog Converter) อุปกรณ์ประเภทนี้อาจจะมีเอาท์พุทเป็นระดับของแรงดันไฟฟ้า หรืออาจเป็นการเปลี่ยนแปลงของปริมาณการไหลของกระแสไฟฟ้าก็ได้ ซึ่งอุปกรณ์แบบหลังนี้ผู้ใช้ต้องออกแบบวงจรสำหรับแปลงกระแสเป็นแรงดันไฟฟ้าด้วย ซึ่งจะทำให้สามารถเลือกช่วงของแรงดันไฟฟ้าได้ตามต้องการ

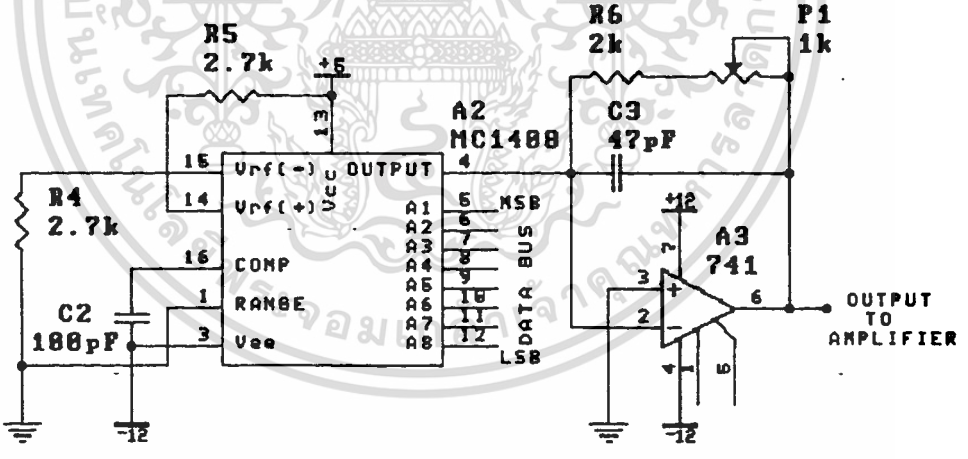
ในอุดมคติอุปกรณ์ DAC ควรจะต้องมีคุณสมบัติ 2 ประการคือ สามารถสร้างระดับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟฟ้าใดๆ ก็ได้ในช่วงที่กำหนดและใช้เวลาในการแปลงสัญญาณเป็นศูนย์ ในทางปฏิบัติอุปกรณ์ดังกล่าวไม่สามารถสร้างระดับของสัญญาณไฟฟ้าแบบอนาล็อกจริงๆ ได้ทุกระดับ แต่จะสามารถสร้างสัญญาณให้ใกล้เคียงมากขึ้นถ้าจำนวนบิทข้อมูลที่ส่งให้มีจำนวนมากขึ้น ดังนั้นอุปกรณ์แปลงสัญญาณขนาด 8 บิทจะสามารถสร้างสัญญาณได้ถูกต้องกว่าอุปกรณ์ขนาด 4 บิท

ความละเอียด (Resolution) ของการแสดงระดับสัญญาณของอุปกรณ์แปลงสัญญาณหมายการเปลี่ยนแปลงระดับของสัญญาณเมื่อมีการเปลี่ยนแปลงค่าทางดิจิตอลทีละหนึ่ง อุปกรณ์ขนาด 8 บิทที่ใช้กับแรงดันไฟฟ้าเต็มพิกัด (Full Scale) 10 โวลท์ จะมีค่าความคลาดเคลื่อนละเอียด 39.1 มิลลิโวลท์ ถ้าเป็น DAC 12 บิทจะมีความละเอียด 2.4 มิลลิโวลท์ เมื่อทราบค่าของแรงดันไฟฟ้าเต็มพิกัดสมมุติเป็น 10 โวลท์ ระดับไฟฟ้าสูงสุดที่สร้างโดย DAC ขนาด 8 บิทจะหาได้จาก 255/256*10 = 9.960375 โวลท์

สำหรับเวลาในการแปลงข้อมูลของ DAC นั้นเรียกว่า Setting Time ซึ่งมีค่าตั้งแต่ 100 นาโนวินาที จนถึง 1.5 ไมโครวินาที ซึ่งผู้เขียนโปรแกรมควรจะต้องระวังไม่ส่งข้อมูลต่อเนื่องกันเร็วเกินกว่าเวลาที่ใช้ในการแปลงสัญญาณนี้มิฉะนั้นอุปกรณ์จะไม่ทำงานได้อย่างถูกต้อง



DIGITAL TO ANALOG WITH MC1408 AND PREAMPLIFIER WITH 741

รูปที่ 3.5 วงจร DAC

3.3.2.ANALOG TO DIGITAL CONVERTER (ADC)

อุปกรณ์ทางไฟฟ้าจำนวนมากทำงานด้วยสัญญาณอนาล็อก ขณะที่ระบบคอมพิวเตอร์เป็นดิจิตอล จึงต้องมีวงจรแปลงสัญญาณจากอนาล็อกเป็นดิจิตอลเพื่อให้ระบบคอมพิวเตอร์สามารถรับรู้ปรากฏการณ์ต่างๆได้อย่างถูกต้องปรากฏการณ์เหล่านี้ได้แก่ การเปลี่ยนแปลงของอุณหภูมิ แรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อากาศ น้ำหนัก ฯลฯ ซึ่งใช้เซนเซอร์แตกต่างกัน เซนเซอร์เหล่านี้จะสร้างสัญญาณไฟฟ้าแบบอนาล็อกซึ่งแปรผันตามค่าต่างๆ ของปรากฏการณ์ที่น่าสนใจ สัญญาณเหล่านี้มักจะมีขนาดน้อยมาก มีหน่วยเป็นไมโครโวลท์หรือมิลลิโวลท์ต้องใช้วงจรปรับปรุงคุณภาพสัญญาณ(Signal Conditioner circuit) ซึ่งขยายขนาดสัญญาณ ลดขนาดของสัญญาณรบกวน ปรับระดับให้ได้ช่วงของสัญญาณตามต้องการ

อุปกรณ์ที่ใช้สำหรับการแปลงสัญญาณอนาล็อกเป็นดิจิตอลที่เรียกสั้นๆ ว่า ADC (Analog to Digital Converter) ซึ่งมีด้วยกันหลายชนิด แบ่งตามหลักการทำงานได้ 4 ชนิด

1.Successive ADC

2.Flash ADC

3.Dual- Slope ADC

4.Delta Mod ADC

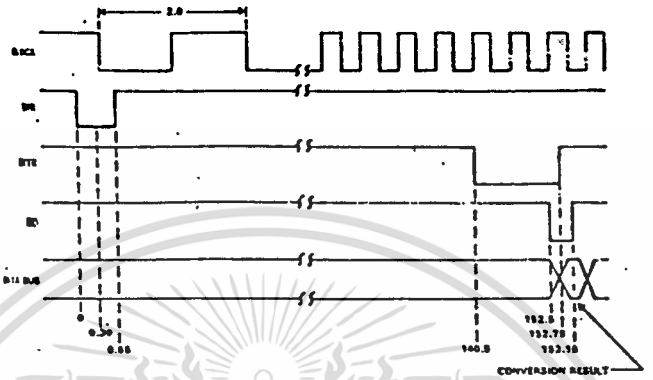
ที่นิยมใช้ทั่วไปได้แก่แบบแรกซึ่งจะนำมาใช้ในโครงงานนี้ แบบที่สองเป็นแบบที่มีราคาสูงแต่มีความเร็วในการทำงานสูงมาก แบบที่สามมีราคาถูกแต่มีความเร็วต่ำมักใช้ในเครื่องมือวัดอย่างง่าย แบบที่สี่นิยมใช้กับอุปกรณ์เก็บข้อมูลที่มีหน่วยความจำจำกัดเนื่องจากใช้ที่เก็บข้อมูลน้อยกว่า แต่ก็มีความคลาดเคลื่อนสูงกว่าด้วย

ADC แบบ Successive Approximation จะใช้สัญญาณนาฬิกาเป็นตัวนับประกอบการทำงานเพื่อจะสร้างระดับสัญญาณไปเปรียบเทียบกับสัญญาณอินพุท ถ้าใช้อุปกรณ์นี้ใช้สัญญาณความถี่สูงก็หมายความว่าเวลาที่จะใช้ในการทำงานเพื่อให้ได้ข้อมูลดิจิตอลออกมา (Conversion Time) ก็จะน้อยด้วย ผู้ออกแบบวงจรสามารถคำนวณได้ว่าเวลานี้ควรมีค่าเท่าใดจากความถี่ของสัญญาณนาฬิกา และข้อมูลจากผังเวลาของอุปกรณ์นี้จะทำให้สามารถคำนวณได้ว่า หลังจากสั่งให้อุปกรณ์ตัวนี้เริ่มทำงาน โดยการส่งศูนย์มาที่ขา WR (หรือขา START) แล้วเป็นเวลาเท่าใดจึงควรจะอ่านข้อมูลจาก ADC (สำหรับ ADC 0804 ถ้าใช้สัญญาณนาฬิกาความถี่ 500 KHz จะใช้เวลาแปลงสัญญาณประมาณ 140 มิลลิวินาที) ในการอ่านข้อมูลนี้ต้องส่งสัญญาณมาให้ ADC ที่ขา RD (หรือขา OE) แล้ว ADC จะส่งข้อมูลออกทางบัสข้อมูล ถ้ายังต้องการข้อมูลตัวต่อไปอีกต้องเริ่มส่งสัญญาณมาเข้าขา WR ใหม่อีกครั้ง

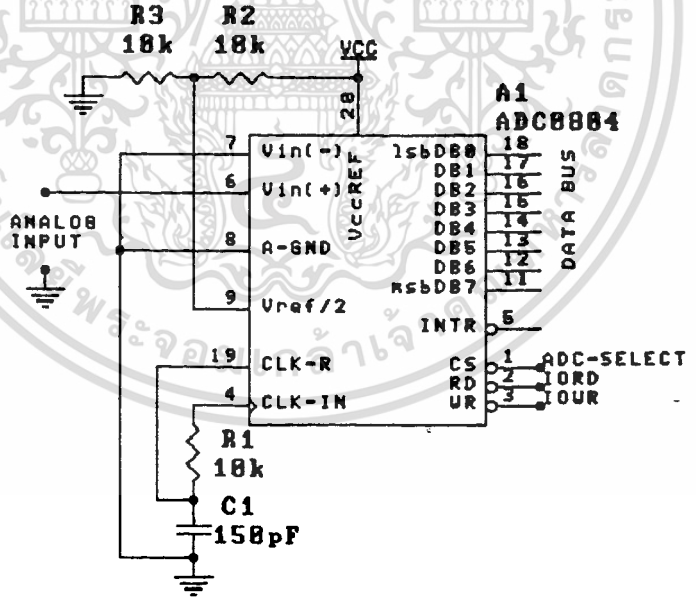
นอกจากนี้ ADC มักจะถูกออกแบบให้ทำงานร่วมกับระบบคอมพิวเตอร์ในแบบอินเตอร์รัพท์ได้อีก โดยมีขาชื่อ EOC (End of Conversion) หรือ INTR ซึ่งจะปรากฏสัญญาณเปลี่ยนแปลงเมื่อการเปลี่ยนแปลงสิ้นสุดลงซึ่งสามารถนำไปกระตุ้นกระบวนการอินเตอร์รัพท์ได้ ดังนั้นผู้ออกแบบระบบอาจจะเขียนโปรแกรมเพื่อการอ่านข้อมูลจาก ADC เป็นโปรแกรมให้การอินเตอร์รัพท์ แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนโปรแกรมหลักสำหรับการทำงานทั่วไป และวนกลับมาตรวจสอบตำแหน่งข้อมูลที่โปรแกรมอินเตอร์รัพท์บันทึกไว้ หรืออาจเขียนโปรแกรมสำหรับวนอ่านสถานะของสัญญาณนี้เพื่อจะได้ทราบว่าแปลงข้อมูลเสร็จแล้วหรือยังก็ได้ เมื่อมีการอ่านข้อมูลหนึ่งครั้งจะมีการรีเซทสัญญาณ INTR ด้วย



รูปที่ 3.6 ผังเวลาการทำงานของ 0804



รูปที่ 3.7 วงจร ADC

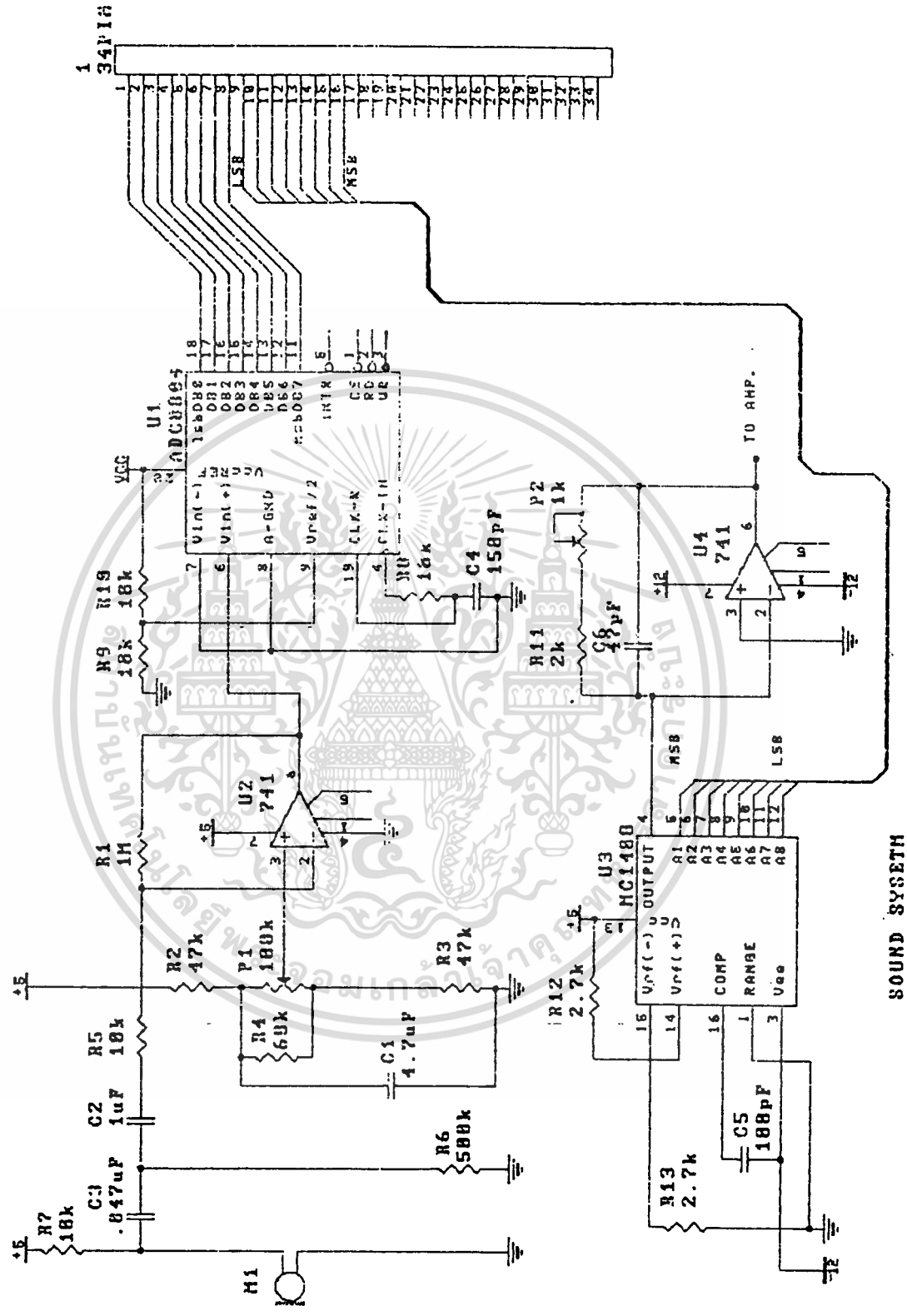
3.3.3. วงจรเสียง

หลักการทำงาน M_1 จะเป็นตัวรับสัญญาณเสียง จากนั้นจะนำมาขยายโดย U_2 เทียบ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ได้มาจากขา 3 จากนั้นสัญญาณเอาท์พุทที่ขา 6 ซึ่งเป็นสัญญาณอนาล็อกจะถูกนำไปแปลงเป็นสัญญาณดิจิตอล โดยใช้ IC U_2 จากนั้นจะถูกนำไปเก็บในเครื่องคอมพิวเตอร์โดยใช้โปรแกรมที่เขียนขึ้น เสียงจะถูกเก็บเป็นแฟ้มข้อมูล เมื่อต้องการที่จะใช้เสียงที่เก็บบันทึกเครื่องคอมพิวเตอร์จะส่งสัญญาณเสียงที่ถูกเก็บในรูปสัญญาณดิจิตอลออกมา จากนั้นจึงนำสัญญาณที่ไปแปลงเป็นสัญญาณอนาล็อกอีกครั้ง โดยใช้ IC U_3 เอาท์พุทที่ได้จะนำไปใช้งานต้องทำการขยายโดย IC U_4 อีกครั้งเพื่อนำไปใช้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรบันทึกเสียง

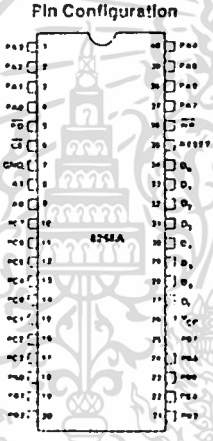
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.ส่วนการควบคุม

เนื่องจากในส่วนนี้การใช้งาน จะใช้ติดต่อควบคุมการทำงานของระบบทั้งหมดทั้งในส่วนของการต่อกับไมโครคอมพิวเตอร์ และส่วนของฮาร์ดแวร์ที่ต่อกับโทรศัพท์ ดังนั้นจึงเลือกใช้ไอซี 8255 เป็นอินพุท/เอาท์พุทพอร์ต เนื่องจากเป็นไอซีเบอร์ที่ใช้ง่ายไม่ยุ่งยากเป็นที่นิยมหาซื้อได้ง่าย และนอกจากนี้ก็ต้องใช้ไอซีถอดรหัสคู่กันในการถอดรหัสสัญญาณซึ่งเลือกใช้ #74LS688 ,74LS139

1.การใช้งานไอซีเบอร์ 8255

ไอซี 8255 นี้จะเป็นไอซี ซึ่งประกอบด้วยพอร์ตใช้งาน 3 พอร์ตและอีก 1พอร์ต ควบคุม ก่อนที่จะใช้ 8255 เราจะส่งข้อมูลไปยังพอร์ตควบคุมก่อนว่าจะให้พอร์ตทั้ง 3 พอร์ต ของ 8255 ที่เหลือนั้นทำหน้าที่อะไรเป็น อินพุท หรือ เอาท์พุทพอร์ต ซึ่งเราจะต้องกำหนดพอร์ตควบคุม ดังรูป

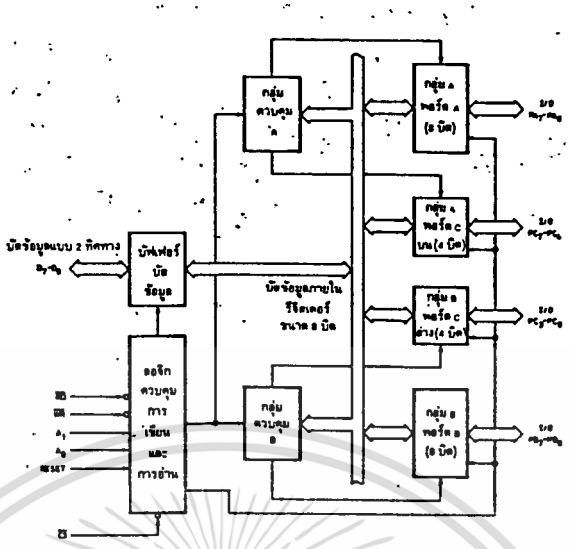


Pin Names

D_7-D_0	Data Bus (Bi-Directional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0, A1	Port Address
PA7-PA0	Port A (BIT)
PB7-PB0	Port B (BIT)
PC7-PC0	Port C (BIT)
V_{CC}	+ 5 Volts
GND	0 Volts

รูปที่ 3.9 ขาของไอซีเบอร์ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 บล็อกไดอะแกรมของ 8255

A_0-A_1 ตำแหน่งอินพุทที่ใช้สำหรับชี้ตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ซีพียูต้องการการติดต่อด้วย

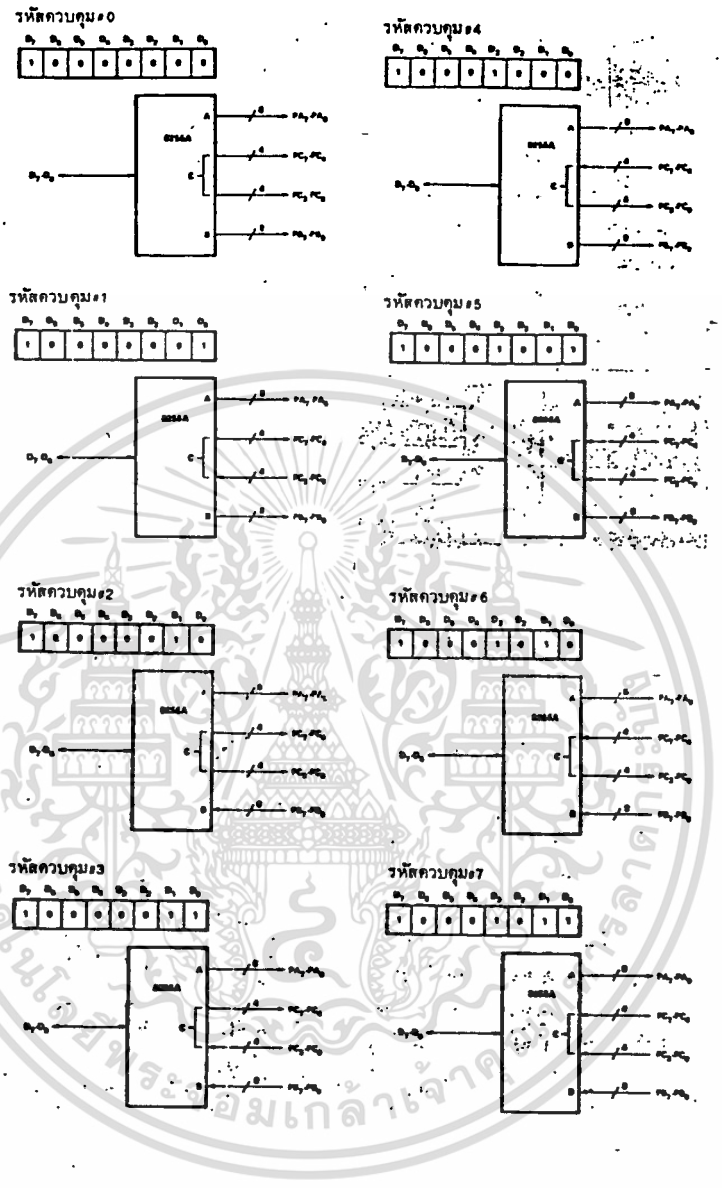
RESET เมื่อขานี้มีค่าลอจิก "1" ไอซี 8255 จะอยู่ในช่วงรีเซตพอร์ตทุกพอร์ตจะอยู่ในโหมดของอินพุทพอร์ต

PA_0-PA_7 พอร์ตของข้อมูลที่ใช้สำหรับต่อกับอุปกรณ์ภายนอก

PB_0-PB_7 พอร์ตของข้อมูลที่ใช้สำหรับต่อกับอุปกรณ์ภายนอก

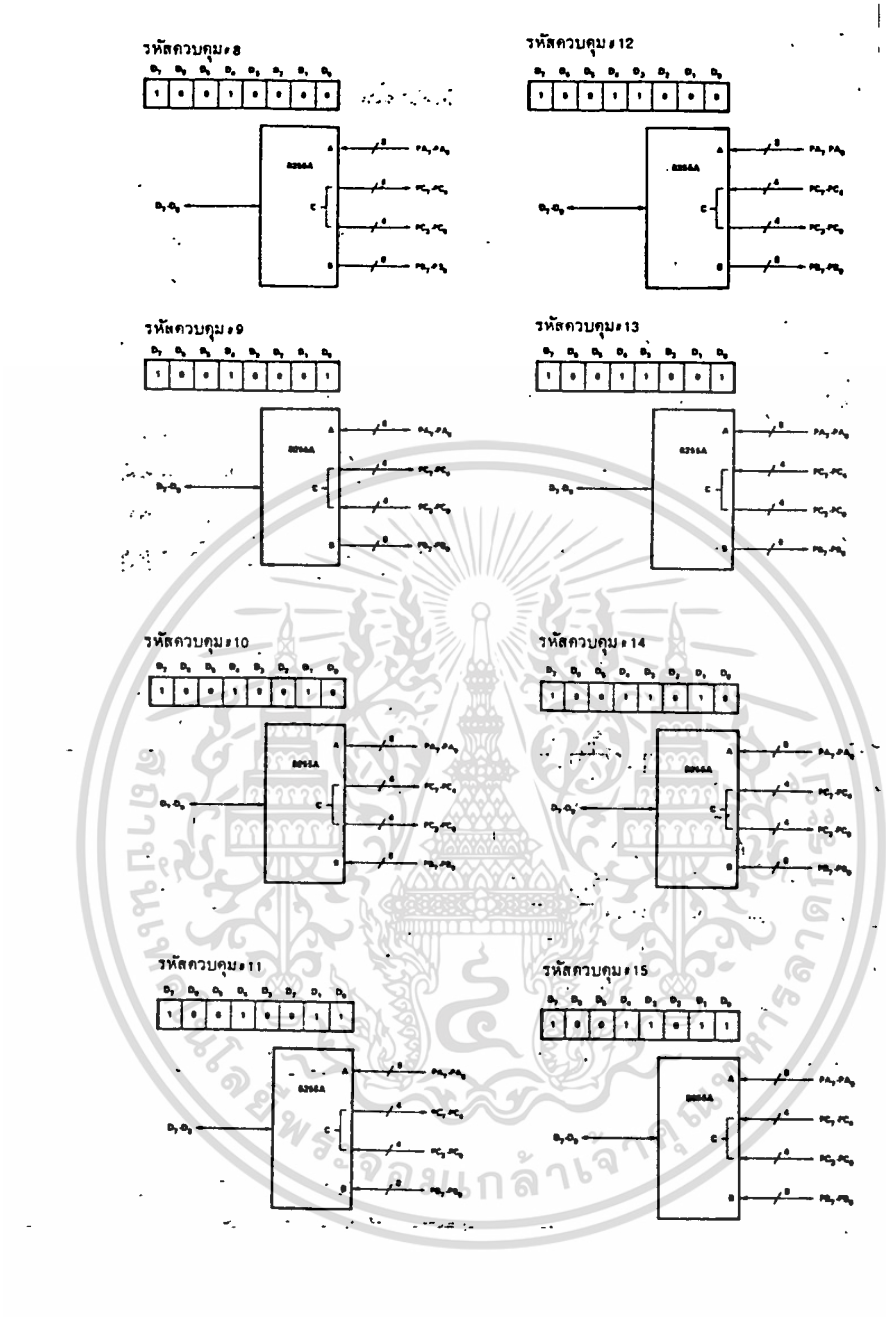
PC_0-PC_7 พอร์ตของข้อมูลที่ใช้สำหรับต่อกับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 คอนโทรลเวิร์ด ของ 8255 แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 (ต่อ) คอนโทรลเวิร์ด ของ 8255 แบบต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hex Range	Usage
000-00F	DMA Chip 8237A-5
020-021	Interrupt 8259A
040-043	Timer 8253-5
060-063	PPI 8255A-5
080-083	DMA Page Registers
0AX*	NMI Mask Register
0CX	Reserved
0EX	Reserved
200-20F	Game Control
210-217	Expansion Unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Communications (Secondary)
300-31F	Prototype Card
320-32F	Fixed Disk
378-37F	Printer
380-38C**	SDLC Communications
380-389**	Binary Synchronous Communications (Secondary)
3A0-3A9	Binary Synchronous Communications (Primary)
3B0-3BF	IBM Monochrome Display/Printer
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3E7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous Communications (Primary)

ตารางที่ 3.1 ตำแหน่งต่างๆ และการนำไปใช้

2.การถอดรหัสพอร์ต

การถอดรหัสพอร์ต 8255 บนการ์ด จะใช้ไอซี # 74LS688 ,74LS139 และดิปสวิทช์ 8 ขา เป็นวงจรถอดรหัส เพื่อให้สามารถปรับตั้งดิปสวิทช์ตงตำแหน่งเบอร์พอร์ตของการ์ดได้ โดยการปรับดิปสวิทช์นั้นจะต้องไม่ไปตรงกับตำแหน่งพอร์ตของเครื่องคอมพิวเตอร์ด้วยดังรูปโดยการ์ด 8255 จะใช้ตำแหน่งพอร์ต 12 พอร์ตต่อ การ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XX0H	PORT A (1)
XX1H	PORT B (1)
XX2H	PORT C (1)
XX3H	PORT CONTROL 1
XX4H	PORT A (2)
XX5H	PORT B (2)
XX6H	PORT C (2)
XX7H	PORT CONTROL 2
XX8H	PORT A (3)
Xx9H	PORT B (3)
XXAH	PORT C (3)
XXBH	PORT CONTROL 3

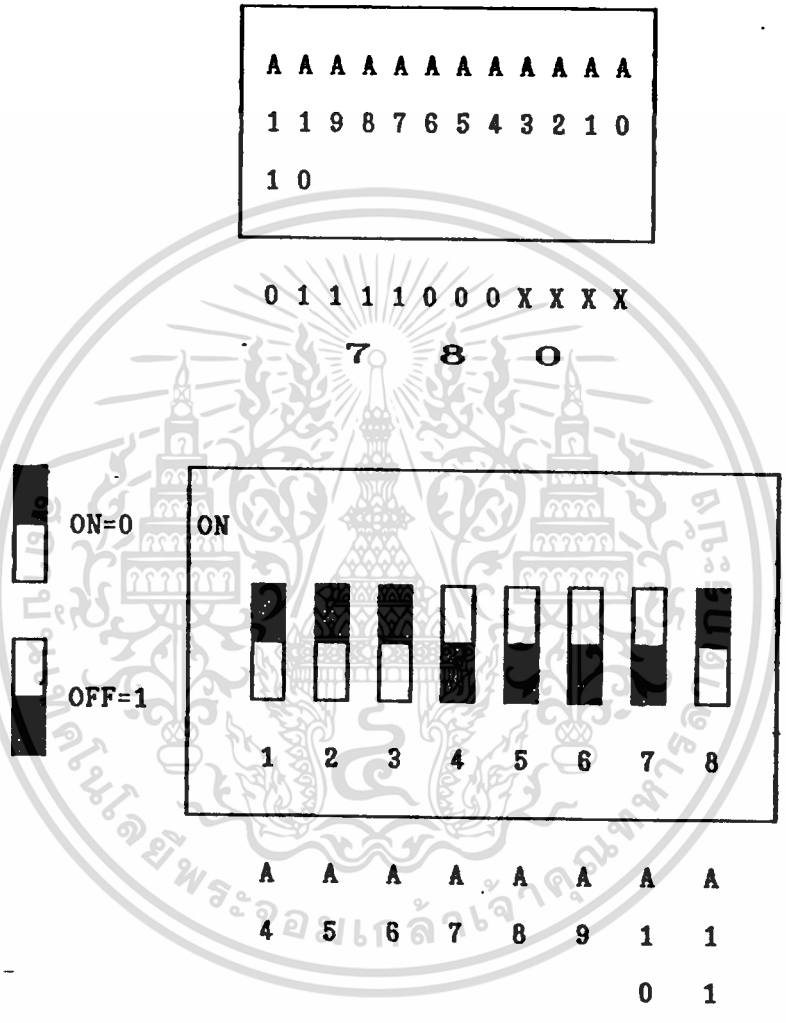
ตารางที่ 3.2 ตำแหน่งของพอร์ต

เราตั้งเบอร์ถอดรหัสพอร์ตได้โดยการปรับดิปสวิทช์ ซึ่งมีค่าเท่ากับค่าตำแหน่งนั้นๆ เช่น เราตั้งตำแหน่ง 780H จะตั้งสวิทช์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตั้งดิปสวิทช์พอร์ต

พอร์ต 780H



รูปที่ 3.12 การตั้งดิปสวิทช์พอร์ต 780H

3.การใช้งาน 8255 ในโหมด 0

การทำงานของ 8255 ในโหมด 0 จะเป็นพอร์ตอินพุทหรือเอาท์พุท แบบธรรมดาเรา สามารถกำหนดให้ 8255 ทำงานในโหมด 0 ได้โดยส่งคอนโทรลเวิร์ดไปยังพอร์ตควบคุม มีค่าต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0

ตารางที่ 3.3 ตัวอย่างการตั้งคอนโทรลเวิร์ดในโหมด 0

จากคำสั่งควบคุมในตารางที่ 3.3 เราสามารถอธิบายความหมายของบิทต่างๆ ได้ดังนี้

D_7 = กำหนดให้ข้อมูลนี้เป็น คอนโทรลเวิร์ด

D_6, D_5 = กำหนดให้พอร์ต A ทำงานในโหมด 0

D_4 = กำหนดให้พอร์ต A เป็นเอาท์พุท

D_3 = กำหนดให้ 4 บิทบนของพอร์ต C เป็นเอาท์พุท

D_2 = กำหนดให้พอร์ต B ทำงานในโหมด 0

D_1 = กำหนดให้พอร์ต B เป็นเอาท์พุท

D_0 = กำหนดให้ 4 บิทล่างของพอร์ต C เป็นเอาท์พุท

3.5.สัญญาณต่างๆ บนสล็อตของ IBM/PC

ภายใน IBM/PC ได้มีการออกแบบ ให้สามารถที่จะเพิ่มเติมวงจรอินเตอร์เฟสเข้าไปในภายหลัง โดยผ่านทางสล็อตที่อยู่บนเมนบอร์ด (Main Board) สำหรับสล็อตบนเมนบอร์ดนี้จะมีจำนวน 5 สล็อต สำหรับ IBM/PC จะมี 8 สล็อตซึ่งแต่ละสล็อตจะมีจำนวนขาทั้งสิ้น 62 ขาแบ่งออกเป็น 2 ข้างๆ ละ 31 ขาส่วนการเรียกตำแหน่งขาของสล็อตเหล่านี้ จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อต โดยขาที่อยู่ทางด้านซ้ายของสล็อต จะเรียกโดยใช้ตัวอักษร "B" นำหน้าเลขตำแหน่งของขา เช่น B16 ก็คือขาทางด้านซ้ายของสล็อตขาที่ 16 (นับจากทางด้านท้ายของเครื่อง)ส่วนขาที่อยู่ทางด้านขวาของสล็อตจะเรียกโดยใช้ตัวอักษร "A" นำหน้าเลขตำแหน่งของขา เช่น A24 ก็คือขาทางด้านขวาของสล็อตขาที่ 24 (นับจากทางด้านท้ายของเครื่อง)

แต่ละขาของสล็อตเหล่านี้ จะเชื่อมต่อกับเส้นสัญญาณต่างๆ บนเมนบอร์ดทำให้การสร้างวงจรอินเตอร์เฟสกับ IBM/PC สามารถทำได้โดยสะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขาของสล็อตเหล่านี้ จะประกอบไปด้วยเส้นสัญญาณของแอดเดรส (Address Bus) ดาต้า (Data Bus) บัสควบคุมสำหรับการเขียน/อ่านข้อมูล จากหน่วยความจำหรือพอร์ต I/O เส้นสัญญาณสำหรับการขออินเตอร์รัมท์ ของวงจรอินเตอร์เฟสเส้นสัญญาณสำหรับขอ DMA สัญญาณฐานเวลา (Timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal) ต่างๆ ที่ใช้ในระบบเส้นสัญญาณ แสดงการรีเฟรชหน่วยความจำ และสัญญาณการตรวจสอบความผิดพลาด (I/O CHECK)

นอกจากเส้นสัญญาณเหล่านี้แล้ว สล็อตบนเมนบอร์ดยังเชื่อมต่อกับแหล่งจ่ายไฟต่างๆ ที่ใช้ในระบบอีกด้วยคือ +5Vdc, -5Vdc, +12Vdc และ -12Vdc

สำหรับรูปของสล็อต แสดงในรูปที่ 2.24 ในบทที่ 2

3.6.บัสของแหล่งจ่ายไฟระบบ

+5Vdc (ขา B3 และ B29) :

ขาทั้งสองนี้ต่อกับแหล่งจ่ายไฟ DC +5V ของระบบ โดยจะมีค่าความเที่ยงตรง 5% (+4.75 ถึง +5.25 Vdc)

+12Vdc (ขา B9) :

ขานี้จะต่อกับแหล่งจ่ายไฟ DC +12V ของระบบ โดยมีค่าความเที่ยงตรง 5% (+11.4 ถึง +12.6 Vdc)

-5Vdc (ขา B5) :

ขานี้จะต่อกับแหล่งจ่ายไฟ -5V ของระบบ โดยมีค่าความเที่ยงตรง 10% (-5.5 ถึง -4.5 Vdc)

-12Vdc (ขา B7) :

ขานี้จะต่อกับแหล่งจ่ายไฟ -12V ของงระบบ โดยมีค่าความเที่ยงตรง 10% (-13.2 ถึง -10.8Vdc)

GND (ขา B1,B10 และ B13) :

ขาทั้งสามนี้จะต่อกับกราวด์ของระบบ

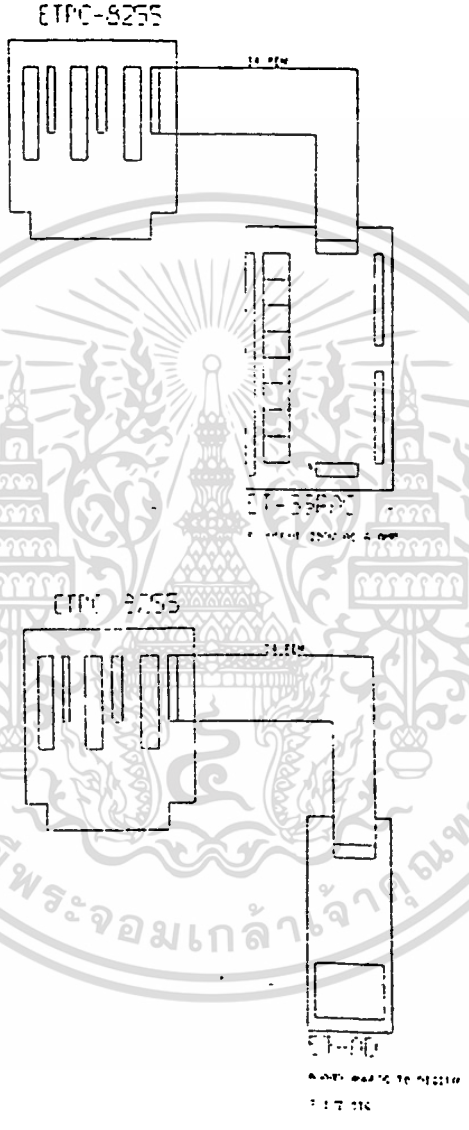
3.7.การจัดสัญญาณบนสล็อตของ IBM PC/XT

สำหรับใน IBM PC/XT นั้นจะมีสล็อต สำหรับเชื่อมต่อกับวงจร ภายนอกได้มากขึ้น คือจะมี 8 สล็อต จากเดิมที่มี 5 สล็อต ส่วนการจัดสัญญาณต่างๆ ทั้ง 8 สล็อต ยังคงเหมือนเดิม เพียงแต่สัญญาณต่างๆ ที่ส่งมายังสล็อตที่ 8 จะถูกต่อผ่านวงจรขับกระแส (Buffer) ก่อนและขา B8 จะถูกใช้งานโดยเป็นขา CRD-SLCTD(Card Selected) ซึ่งขาสัญญาณนี้จะเป็นสัญญาณเอาท์พุท จากวงจรภายนอกที่เสียบอยู่บนสล็อตที่ 8 เพื่อให้วงจรบนเมนบอร์ดรู้ว่าการ์ดที่เสียบอยู่สล็อตนี้ถูกเลือกใช้งานอยู่ ซึ่งจะทำให้ Driver บนเมนบอร์ดทำการอ่านหรือส่งข้อมูลไปยังสล็อตที่ 8

3.8.การใช้งานร่วมกับอุปกรณ์ต่างๆ

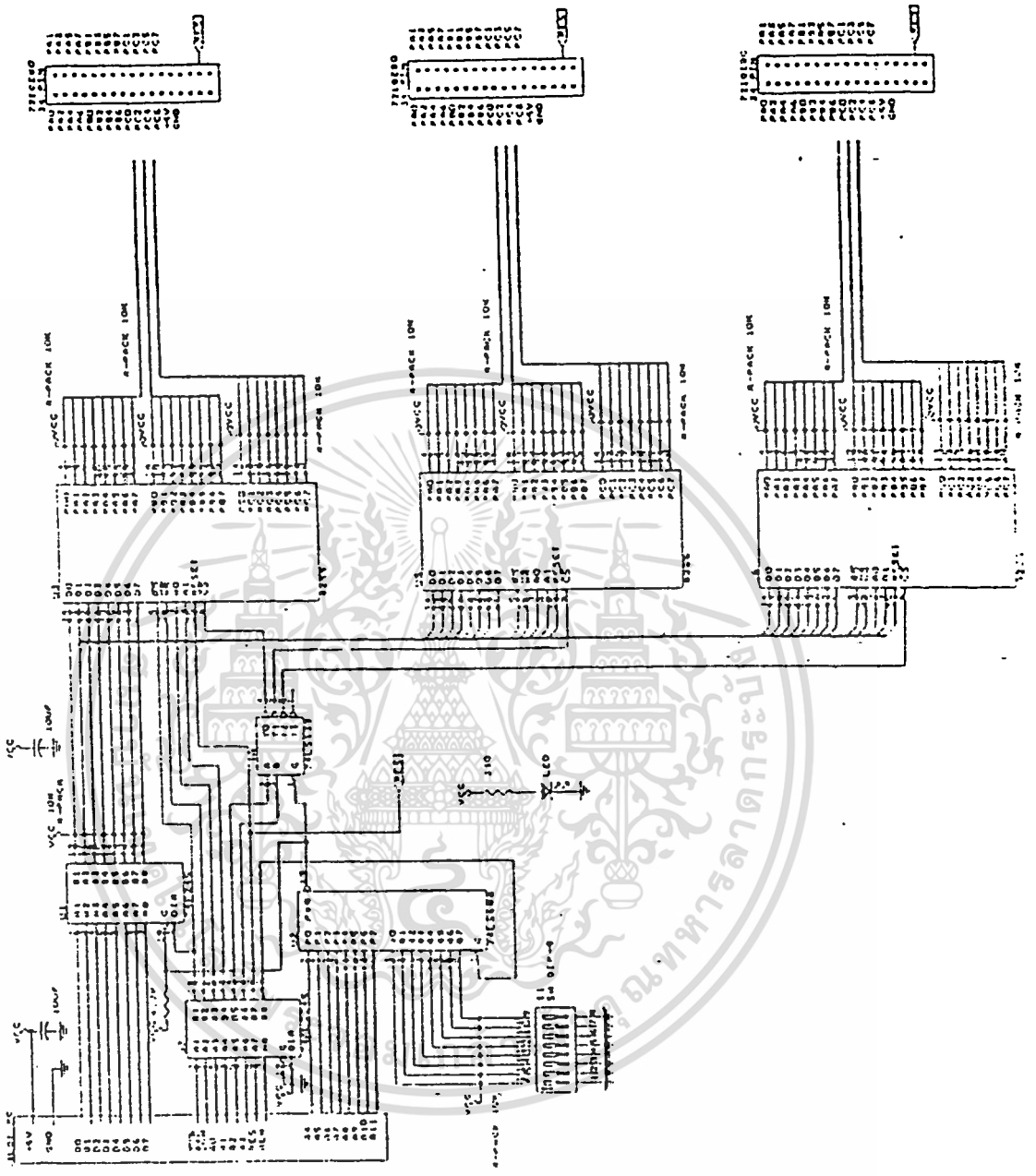
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถต่อ ตัวต่อ 34 ขาของการ์ด 8255 ไปยังบอร์ดต่างๆ ได้เช่น ต่อใช้งานกับ ET-AD ซึ่งเป็นบอร์ดควบคุมไฟ AC 220V ได้ 8 เอาท์พุทดังรูป หรือจะต่อรับกับ ET-AD ซึ่งเป็นบอร์ดวัดแรงดัน,ความต้านทาน หรืออุณหภูมิ ได้ด้วยดังรูป



รูปที่ 3.13 การต่อ 8255 เข้ากับอุปกรณ์แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ไอซี 8255 I/O พอร์ต

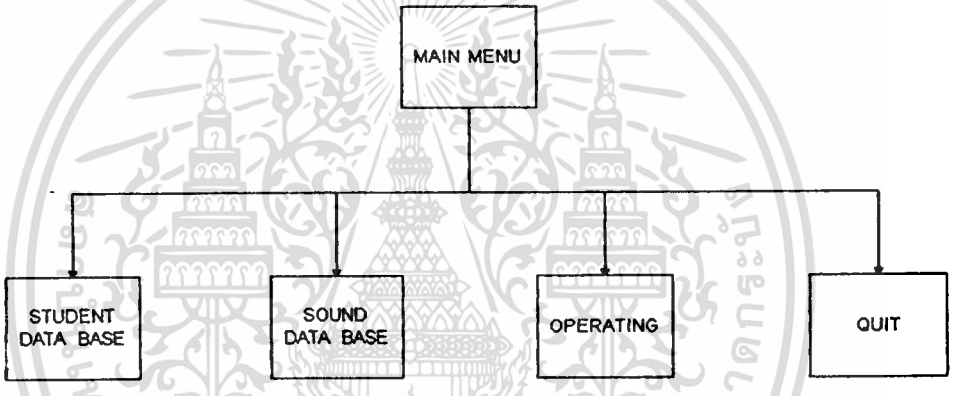
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9.ส่วนของการเขียนโปรแกรมระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

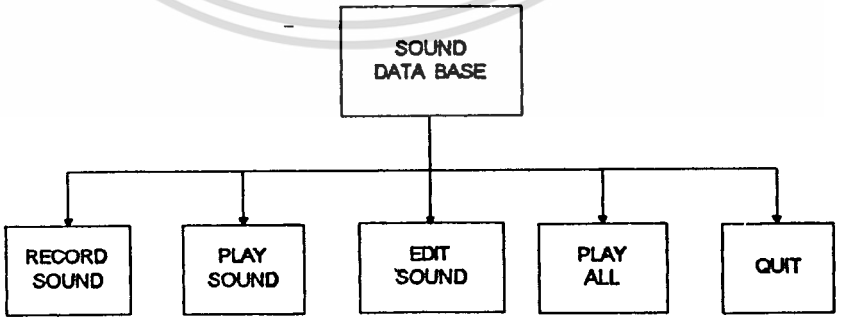
เมื่อสิ้นสุดการออกแบบฮาร์ดแวร์แล้ว ในส่วนต่อไปจะกล่าวถึงทางด้านซอฟแวร์หรือส่วนของโปรแกรมของระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

ขั้นตอนทางด้านโปรแกรม แรกสุดต้องเข้าใจการทำงานของระบบก่อน หลังจากนั้นก็นำขั้นตอนต่างๆ มาเขียนโฟลว์ชาร์ทการทำงาน แล้วจึงเขียนโปรแกรมตามโฟลว์ชาร์ทให้สัมพันธ์กันในแต่ละส่วน

โดยที่เราจะแบ่งเป็นโปรมแกรมที่ใช้ในการควบคุมวงจรสัญญาณเสียง และโปรแกรมที่ใช้ในการเปิดแฟ้มข้อมูลที่เขียนขึ้นใหม่

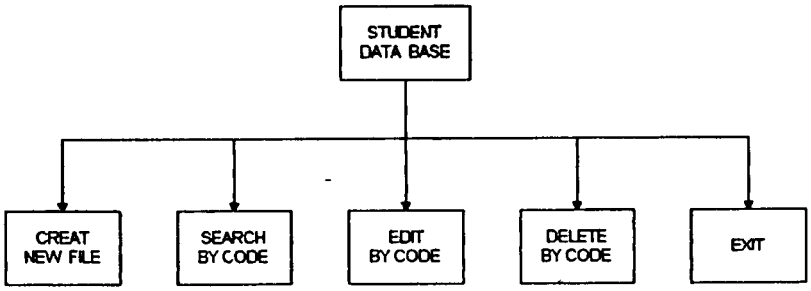


รูปที่ 3.15 โฟลว์ชาร์เมนูหลัก



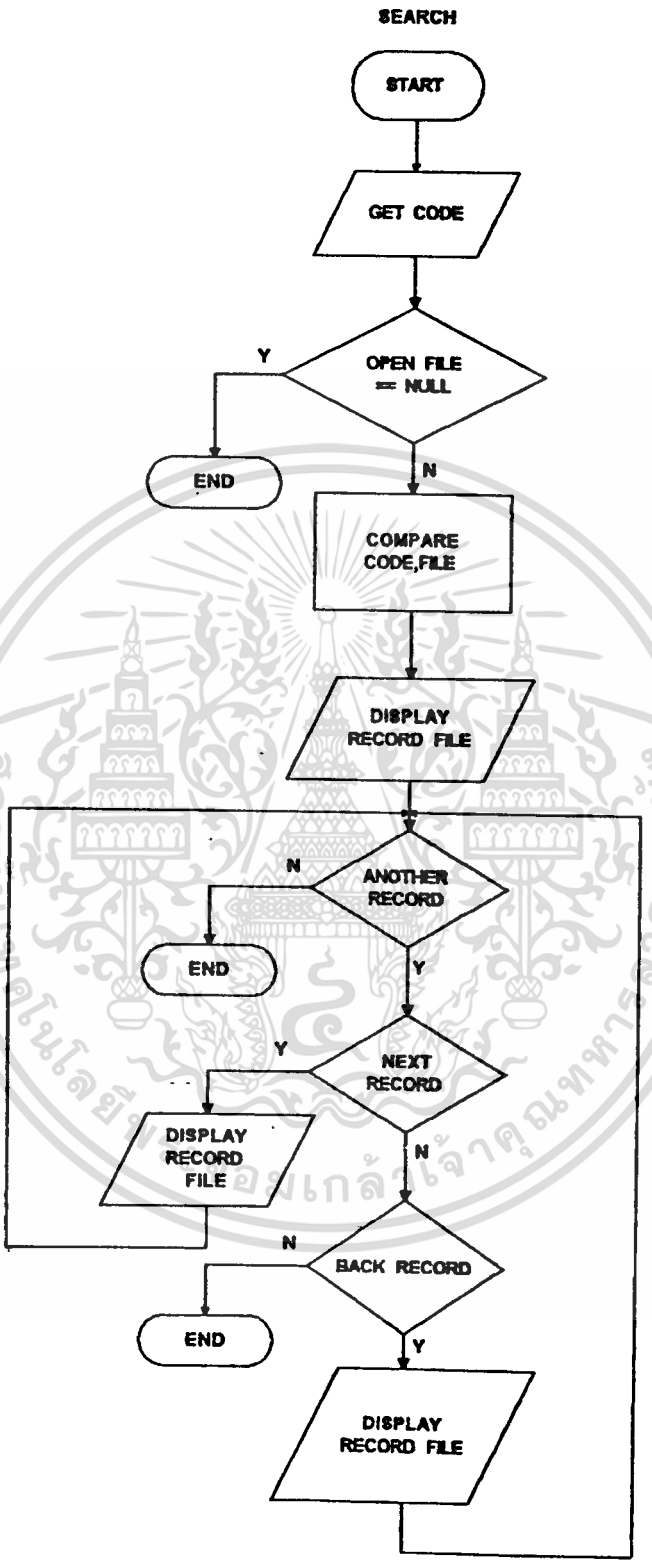
รูปที่ 3.16 โฟลว์ชาร์ทเมนูของชุดสัญญาณเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



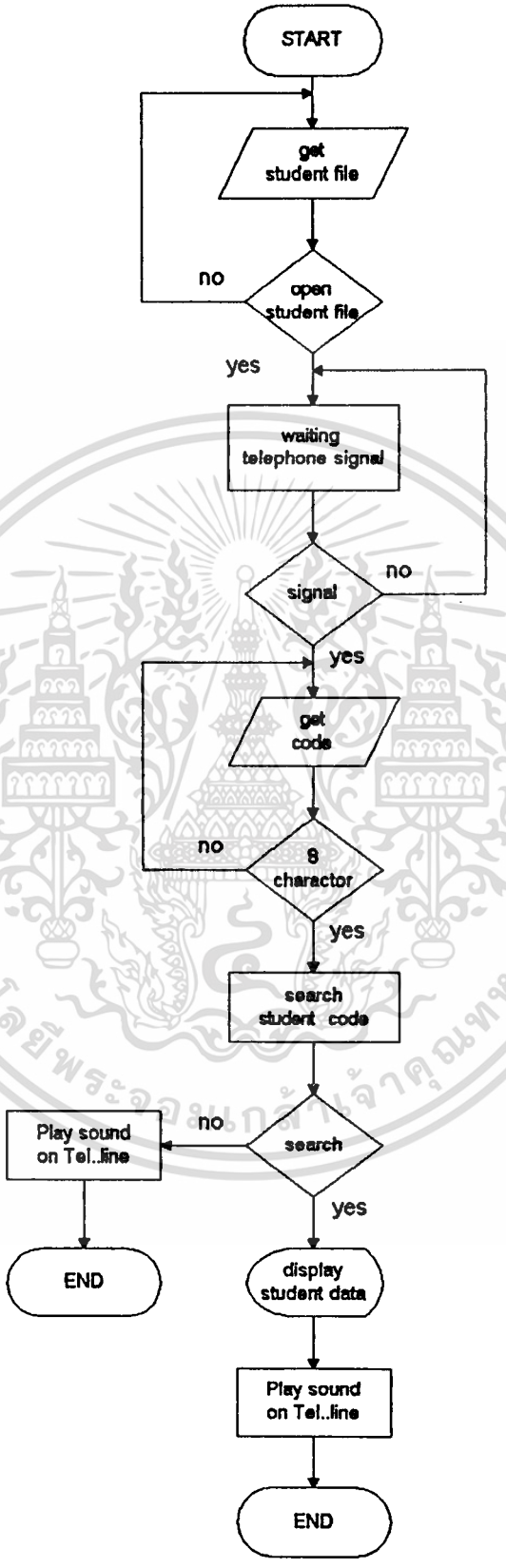
รูปที่ 3.17 โฟลว์ชาร์ทแสดงชุดบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 โฟลว์ชาร์ทการทำงานของโปรมแกรมการค้นหาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 โฟลว์ชาร์ทการทำงานของโปรแกรม OPERATING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะเป็นการทดลองในการใช้งาน ระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติสิ่งที่ต้องเตรียมมีดังนี้

4.1.อุปกรณ์ที่ต้องใช้งาน

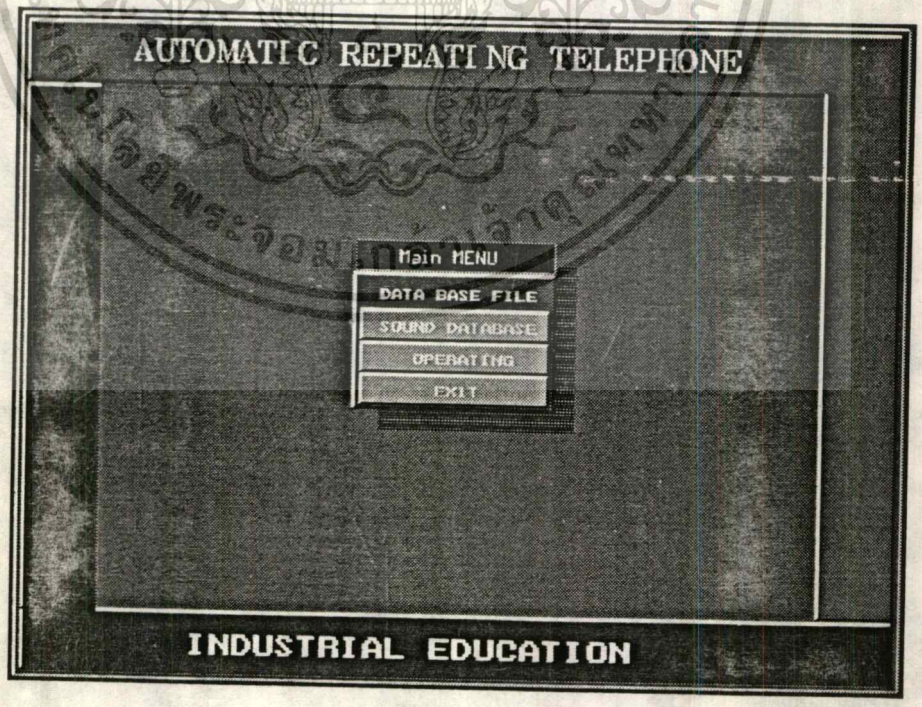
1.เครื่องไมโครคอมพิวเตอร์ขนาด 32 บิทขึ้นไป

2.ส่วนแสดงผลเป็นจอแบบ SVGA

3.โปรแกรม PROJ.EXE, FSOUND.DAT, SNDDTB.H, OPRT.H ควรเก็บไว้ใน Hardisk หรือ Directory เดียวกัน

4.2.การเตรียมแฟ้มข้อมูลสำหรับเก็บรหัสนักศึกษา

1.เรียกแฟ้มข้อมูลชื่อ PROJ.EXE จะปรากฏภาพเมนูหลัก (MAIN MENU) ดังรูปที่ 4.1 ในเมนูหลักจะประกอบด้วยแฟ้มข้อมูลอีก 4 ชุด คือ



รูปที่ 4.1 แสดงเมนูหลัก (MAIN MENU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

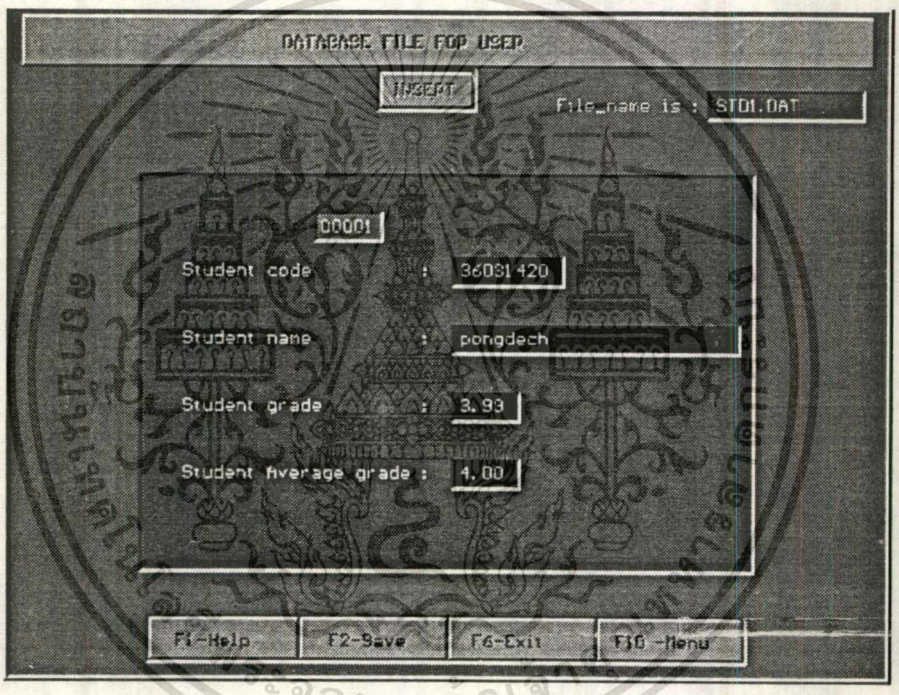
DATA BASE FILE

SOUND DATA BASE

OPERATING

EXIT

2.เมื่อเราเลือกที่ DATA BASE FILE จะเป็นการให้บันทึกข้อมูลนักศึกษา ดังรูปที่ 4.2 โดยใส่ชื่อแฟ้มข้อมูลที่เราจะจัดเก็บข้อมูลของนักศึกษา ลงไป



รูปที่ 4.2 แสดงแบบฟอร์ม DATA BASE FILE

3.ใส่ชื่อแฟ้มข้อมูลลงไปแล้ว ก็จะปรากฏ Cursor ให้เราเริ่มบันทึกข้อมูลของนักศึกษา ลงไป โดยมีข้อมูลที่จะบันทึกมีดังนี้

CODE : รหัสประจำตัวนักศึกษา

NAME : ชื่อ นามสกุลของนักศึกษา

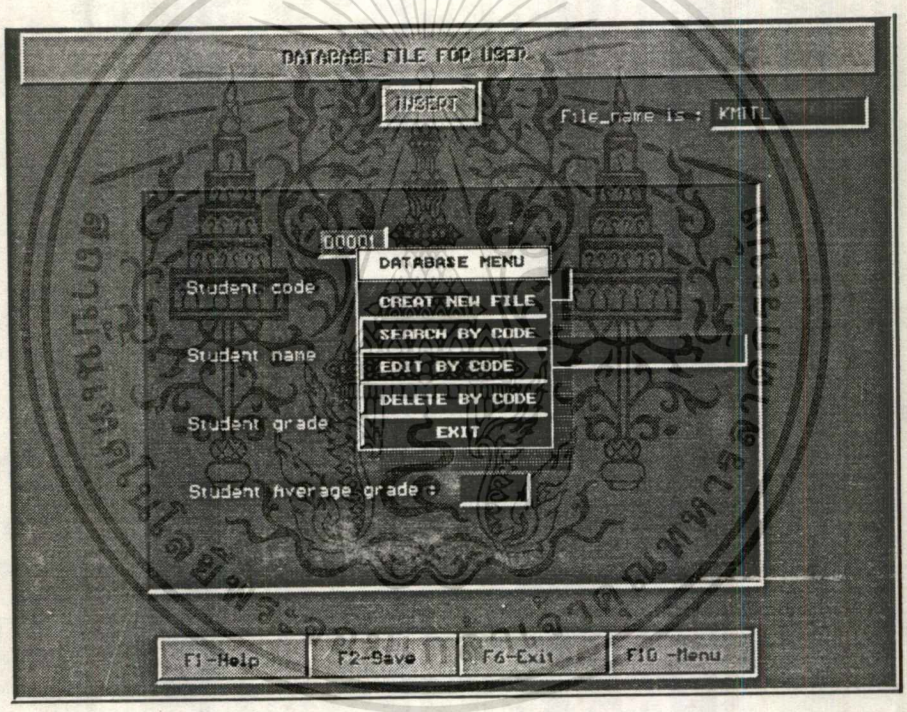
GRADE : เกรดเฉลี่ยประจำเทอม

AV GRADE : เกรดเฉลี่ยสะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.เมื่อป้อนข้อมูลของนักศึกษาแต่ละคนแล้ว ให้เก็บบันทึกข้อมูล โดยการกดคีย์ F_2 เพื่อเก็บลง Disk และเมื่อบันทึกเสร็จแล้ว ก็จะเป็นการบันทึกข้อมูลของนักศึกษาคนต่อไป

5.เมื่อเกิดการผิดพลาดของข้อมูลที่บันทึกลง Disk และต้องการแก้ไขหรือเพื่อการค้นหาและต้กงการเปิดแฟ้มข้อมูลใหม่ ก็ให้กดคีย์ F_{10} จะปรากฏภาพเมนูชุด DATA BASE MENU ขึ้นมาดังรูปที่ 4.3 ซึ่งใช้สำหรับจัดการเกี่ยวกับข้อมูลของนักศึกษาที่เราจะบันทึกลงใน Disk เพื่อที่จะใช้เป็น ข้อมูลในการ บอกเกรดของนักศึกษาผ่านคู่สายโทรศัพท์ ซึ่งใน Menu DATA BASE FILE จะประกอบด้วยแฟ้มข้อมูลอีก 5 ชุด คือ



รูปที่ 4.3 แสดงแบบฟอร์ม DATA BASE MENU

5.1.CREAT NEW FILE

ใช้สำหรับเปิดไฟล์ใหม่ เพื่อเก็บข้อมูลนักศึกษา

5.2.SEARCH BY CODE

ใช้สำหรับค้นหาข้อมูลนักศึกษา โดยใช้รหัสประจำตัวนักศึกษาเป็นสื่อค้นหาตำแหน่งข้อมูลของนักศึกษา

5.3.EDIT BY CODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการค้นหาข้อมูลจากรหัสประจำตัวนักศึกษาออกมาแก้ไข โดยหลังจากแก้ไขจะต้องเก็บบันทึกข้อมูลทุกครั้ง

5.4.DELETE BY CODE

เพื่อเป็นการลบข้อมูลของนักศึกษา โดยป้อนรหัสประจำตัวของนักศึกษาที่ต้องการลบลงไป

5.5.EXIT

จะใช้ในการกลับสู่ MAIN MENU

6.เมื่อบันทึกข้อมูลนักศึกษาครบแล้ว ต้องการจะออกจากจากชุด DATA BASE FILE ทำได้โดยการกดคีย์ F_{10} จะปรากฏบล็อกขึ้นมาถามว่าจะออกจากโปรแกรมหรือไม่ ให้ Y เพื่อตอบว่า Yes ก็จะออกจากชุด DATA BASE กลับไปยังเมนูหลัก (MAIN MENU) ก็เป็นอันเสร็จสิ้นการเตรียมข้อมูลเพื่อเก็บข้อมูลนักศึกษา

4.3.การเตรียมแฟ้มข้อมูลเสียง

1.เปิดแฟ้มข้อมูลชื่อ PROJ.EXE จะปรากฏโปรแกรมหลัก(MIAN MENU) แล้วเลือกไปที่ SOUND DATABASE จะปรากฏเมนูของเสียง (SOUND MENU) ดังรูปที่ 4.4 ซึ่งจะประกอบด้วยแฟ้มข้อมูลย่อย 5 ชุด

1.1.ชุดบันทึกเสียง (RECORD SOUND)

โปรแกรมชุดนี้ใช้เพื่อบันทึกเสียง โดยใช้ร่วมกับวงจรชุดเสียงซึ่งจะบันทึกโดยบันทึกผ่านไมโครโฟนของวงจรชุดเสียงไปเก็บไว้ใน MEMORY

1.2.ชุดแก้ไขเสียง (EDIT SOUND)

โปรแกรมชุดนี้จะแสดงชื่อแฟ้มข้อมูลของเสียงซึ่งใช้งานอยู่ในปัจจุบัน และสามารถแก้ไขเป็นแฟ้มข้อมูลของเสียงที่เราต้องการใช้งานได้

1.3.ชุดเล่นเสียง (PLAY SOUND)

โปรแกรมชุดนี้จะใช้เมื่อต้องการทดสอบเสียงที่ได้จัดเก็บเอาไว้ออกมาฟัง โดยการป้อนชื่อแฟ้มข้อมูลของเสียงที่เราต้องฟังลงไป

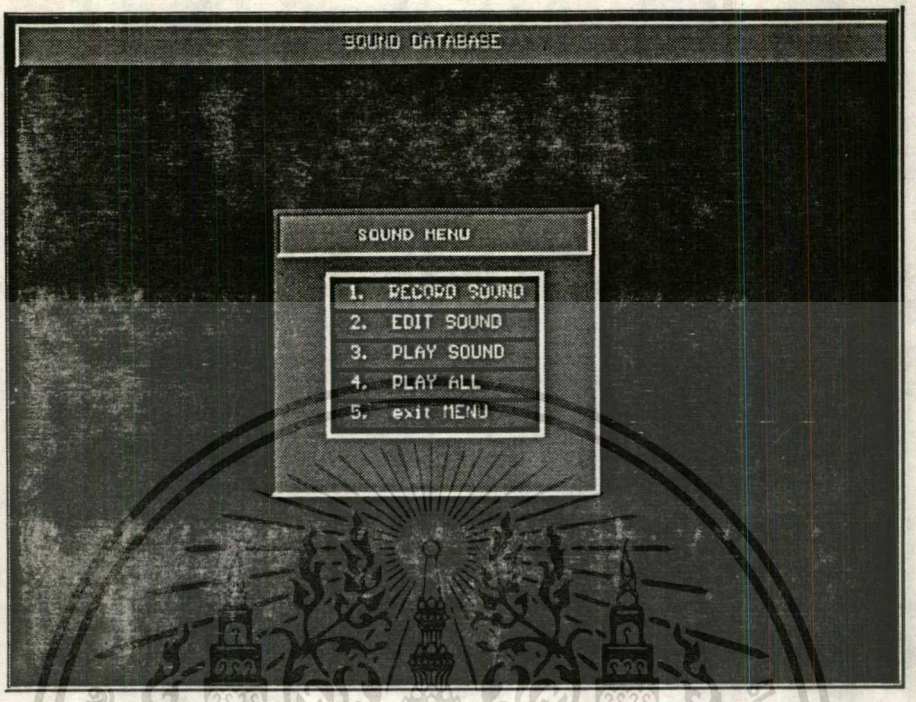
1.4.ชุดเล่นเสียงทั้งหมด (PLAY ALL)

โปรแกรมชุดนี้ใช้ในการเล่นเสียงทั้งหมดที่ใช้ในการบอกเกรดออกมาฟังโดยจะเรียงตามลำดับแฟ้มข้อมูลของเสียงที่บันทึกใน ชุดแก้ไขเสียง (EDIT SOUND)

1.5.ชุดเลิกการทำงาน (EXIT MENU)

โปรแกรมนี้ใช้เมื่อต้องการกลับไปยัง โปรแกรมหลัก (MAIN MENU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงแบบฟอร์ม SOUND MENU

2.เลือกไปที่ RECORD SOUND จะปรากฏดังรูปที่ 4.5 ซึ่งจะเป็นชุดที่ใช้ในการบันทึกเสียง และจะมีเคอร์เซอร์ชี้นมาให้ใส่ชื่อแฟ้มข้อมูลที่ต้องการจะบันทึกเสียง

3.เมื่อป้อนชื่อแฟ้มข้อมูลแล้ว ทำการบันทึกเสียงผ่านไมโครโฟนของกล่องวงจรชุดเสียง เมื่อสิ้นสุดประโยคที่ต้องการจะใช้งานให้กดปุ่มใดปุ่มหนึ่งเพื่อหยุดการบันทึก โปรแกรมจะถามว่าจะบันทึกแฟ้มข้อมูลของเสียงอีกหรือไม่ ถ้าต้องการให้กดปุ่ม Y คือ YES โปรแกรมก็จะชี้นให้ใส่ชื่อแฟ้มข้อมูลที่จะบันทึกใหม่ และถ้าไม่ต้องการให้กด N หรือ NO จะกลับไปที่ เมนูเสียง (SOUND MENU)

4.เมื่อกลับมาที่ SOUND MENU สามารถทำการทดสอบเสียงที่เก็บไว้เป็นแฟ้มข้อมูลของเสียงโดยเลือกไปที่ PLAY SOUND จะปรากฏดังรูปที่4.6 แล้วจึงป้อนชื่อข้อมูลของเสียงที่ต้องการทดสอบเสียง ถ้าไม่พอใจในคุณภาพเสียง ต้องกลับไปบันทึกเสียงใหม่ที่ชุดของการ บันทึกเสียง (RECORD SOUND)

5.เมื่อทำการทดสอบเสียงทุกเสียงที่จะใช้เสร็จแล้วก็นำชื่อของเสียงเหล่านั้นมากำหนดในชุดแก้ไขเสียง(EDIT SOUND) โดยการกลับมาที่ SOUND MENU แล้วเลือก EDIT SOUND จะปรากฏดังรูปที่ 4.7 โดย

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

file sound ZERO ถึง file sound NINE จะต้องเป็นชื่อของแฟ้มข้อมูลของเสียง "ศูนย์ ถึง เก้า" ("0 ถึง 9")

file sound POINT คือเสียงของ "จุด"

file sound Title_1 คือเสียงของคำแนะนำให้ผู้โทร กดรหัสประจำตัวนักศึกษาที่ต้องการสอบถามเกรด

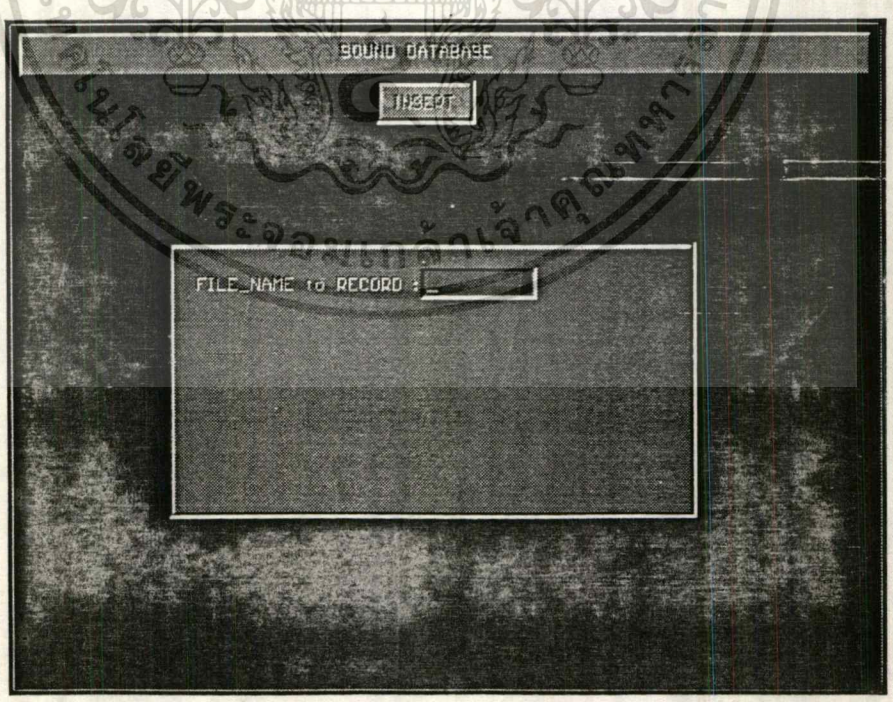
file sound Title_2 คือเสียงของการทวนรหัสที่กดเข้ามา เช่น "รหัสที่ท่านป้อนเข้ามาคือ"

file sound Title_3 คือเสียงแนะนำก่อนการบอกเกรด เช่น "เกรดที่ได้คือ"

file sound End_1 คือเสียงที่ใช้บอกเมื่อมีการ ไม่พบข้อมูลตามรหัสที่ป้อนเข้ามา เช่น " ไม่พบข้อมูลตามรหัสที่ท่านป้อนเข้ามา "

file sound End_2 และ file sound End_3 คือเสียงที่ใช้บอกถึงการสิ้นสุดการสอบถามเกรด เช่น " ขอบคุณที่ใช้บริการ "

เมื่อป้อนชื่อแฟ้มข้อมูลของเสียงครบทุกเสียงแล้ว จะต้องทำการ SAVE ข้อมูลด้วยการกด ปุ่ม F_2 และถ้าต้องการออกจาก EDIT SOUND เพื่อกลับไปยังเมนูเสียง (SOUND MENU) ก็ทำได้ด้วยการกด ESC

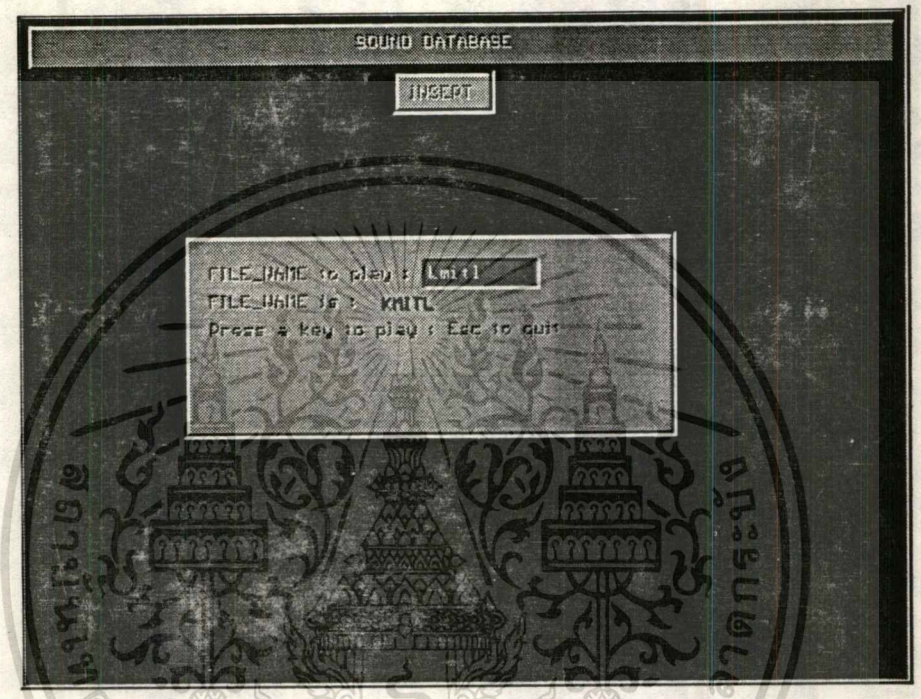


รูปที่ 4.5 แสดงแบบฟอร์ม RECORD SOUND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.สามารถทดสอบเสียงทั้งหมดด้วยการกลับไปที่ SOUND MENU แล้วเลือก PLAY ALL ก็จะทำการเล่นเสียงทั้งหมด ตั้งแต่แฟ้มข้อมูลของเสียงแรกจนถึงเสียงสุดท้าย

7.เมื่อเสร็จสิ้นการเตรียมแฟ้มข้อมูลของเสียงแล้ว และต้องการออกจากเมนูเสียงหรือ SOUND MENU ให้เลือกที่ EXIT MENU



รูปที่ 4.6 แสดงแบบฟอร์ม PLAY SOUND

4.4.ขั้นตอนการทำงาน

1.เข้าไปยังโปรแกรมหลัก (MAIN MENU) โดยการเรียกโปรแกรม PROJ.EXE

2.จัดเตรียมแฟ้มข้อมูลของนักศึกษาที่ต้องการจะใช้งาน

3.จัดเตรียมแฟ้มข้อมูลของเสียงที่จะใช้แจ้งเกรดประจำภาคเรียน จากนั้นกลับไปยังโปรแกรมหลัก (MAIN MENU) แล้วเลือก OPERATING จะปรากฏดังรูปที่ 4.8

4.ป้อนชื่อแฟ้มข้อมูลของนักศึกษาที่ต้องการจะใช้งานใน Student File โปรแกรมจะทำการรอสัญญาณโทรศัพท์ที่ผู้สอบถามจะเรียกเข้ามา ซึ่งโปรแกรมจะแสดง WAITING FOR TEL..RING

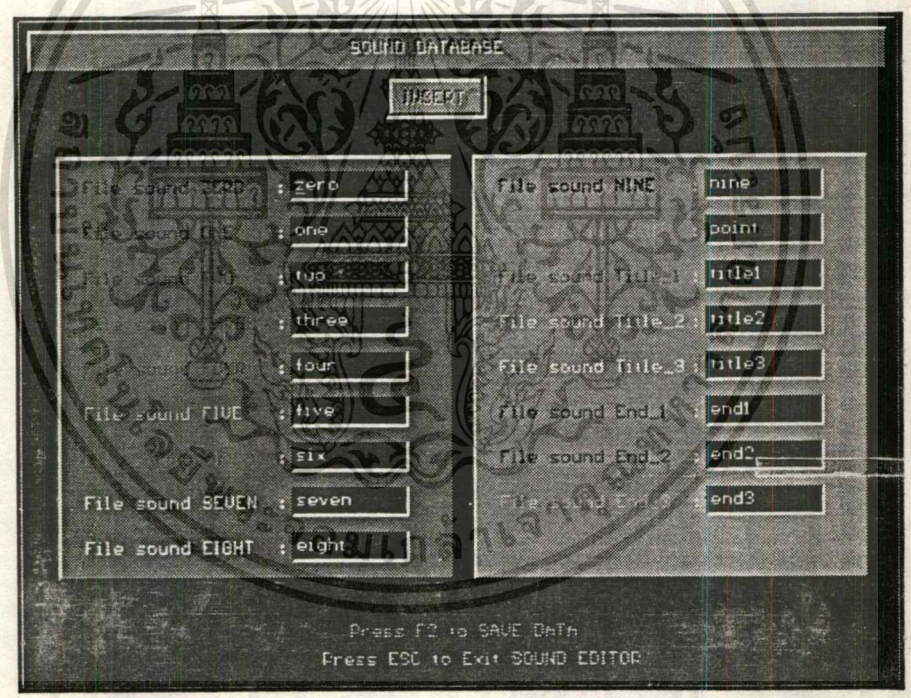
5.เมื่อมีสัญญาณ ring เข้ามา เครื่องจะทำการยกหู และส่งเสียง ตามแฟ้มข้อมูลของ file sound Title_1 จากนั้นเครื่องจะรอรับรหัสประจำตัวนักศึกษา โดยแสดง WAITING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FOR TEL..CODE ที่หน้าจอ

6.เมื่อผู้เรียกกดรหัสประจำตัวนักศึกษาเข้ามาครบ 8 ตัว โปรแกรมจะทำการทวนรหัสให้อีกครั้ง จากนั้นโปรแกรมจะทำการเปิดแฟ้มข้อมูลที่ใส่ไว้ใน student file : เพื่อค้นหารหัสประจำตัวนักศึกษาเปรียบเทียบกับรหัสที่รับเข้ามาทางโทรศัพท์ ถ้าพบรหัสดังกล่าว โปรแกรมจะแสดงข้อมูลของนักศึกษาตามรหัสนั้นออกมา และจะบอกเกรดออกมา เช่น " คุณได้เกรด สาม จุด เก้า เก้า " ถ้าหากค้นหารหัสไม่พบ โปรแกรมคะบอกว่า " ไม่มีรหัสนี้ในแฟ้มข้อมูล " หรือแฟ้มข้อมูลของเสียงที่บันทึกไว้ใน file sound End_1

7.ในขั้นตอนสุดท้ายก่อนจบโปรแกรมจะพูดว่า " ขอบคุณครับ " หรือคำพูดที่ต้องการแจ้งให้ทราบอื่นๆ ตามที่บันทึกไว้ใน file sound End_2 และ End_3 แล้วโปรแกรมจะทำการวางหูโทรศัพท์ เพื่อรอรับโทรเข้ามาอีกครั้ง



รูปที่ 4.7 แสดงแบบฟอร์ม EDIT SOUND

4.5.ผลการทดลอง

จากขั้นตอนการทำงานตามที่กล่าวมาแล้วข้างต้น เมื่อรอรับสัญญาณที่โทรเข้ามา ระบบจะทำงานตามขั้นตอนดังนี้

1.เมื่อผู้เรียกทำการโทรเข้ามาระบบจะทำการยกหูอัตโนมัติ โดยที่หน้าจอคอมพิวเตอร์

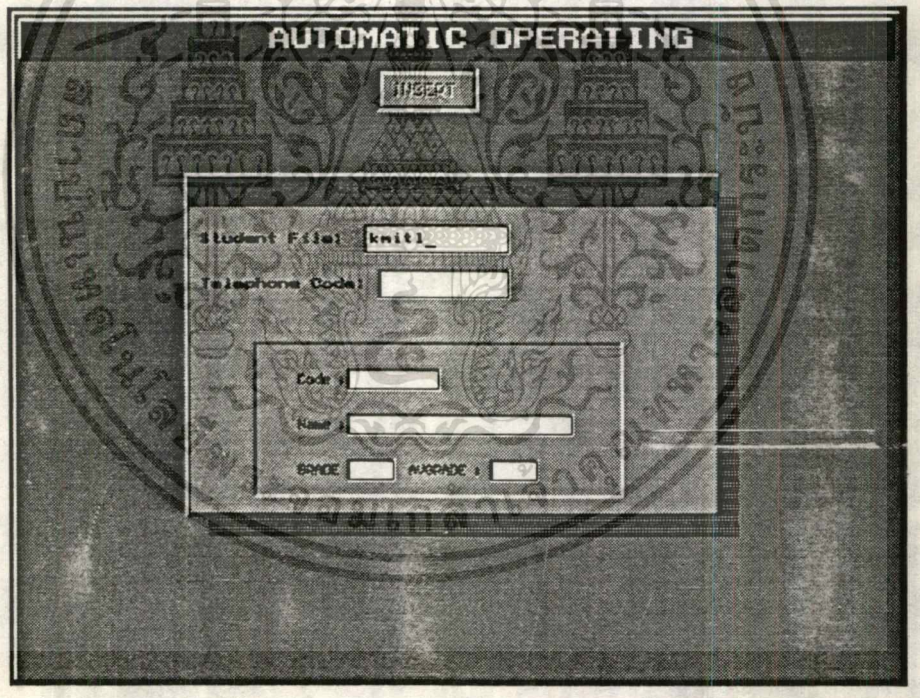
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะแสดง WAIT FOR TEL..CODE เพื่อรอรับรหัส แต่ยังไม่สามารถส่งเสียงรายงานไปยังผู้เรียกให้กดรหัสประจำตัวได้

2.เมื่อระบบได้รับหมายเลขรหัสครบ 8 ตัว ในส่วนของหน้าจอจะแสดงหมายเลขที่ผู้เรียกป้อนเข้ามา แต่ยังไม่สามารถส่งเสียงทวนรหัสได้

3.หลังจากนั้นระบบจะทำการค้นหาข้อมูล ที่หน้าจอจะแสดง "SEARCHING CODE" แสดงว่ากำลังค้นหาข้อมูลอยู่เมื่อค้นพบหน้าจอจะแสดง "CODE OK...!" และแสดงผลข้อมูลของนักศึกษาออกที่หน้าจอ แต่ถ้าค้นหาไม่พบที่หน้าจอจะแสดง "NO SUCH CODE IN FILE" แต่ยังไม่สามารถส่งเสียงบอกเกรดหรือไม่พบข้อมูล ออกสายโทรศัพท์ได้

4.หลังจากค้นหาข้อมูลและแสดงผลเสร็จแล้วโปรแกรมจะทำการวางหูโทรศัพท์เพื่อรอรับสัญญาณโทรศัพท์เข้ามาใหม่ โดยที่หน้าจอจะแสดง "WAITING FOR TEL RING"



รูปที่ 4.8 แสดงแบบฟอร์มชุด OPERATING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1.สรุปผลของโครงงาน

โครงงานระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ ที่ได้สร้างขึ้นนี้ผลของโครงงานนี้สามารถบรรลุวัตถุประสงค์ที่ได้ตั้งไว้กล่าวคือ

5.1.1.ได้รู้และเข้าใจการประยุกต์ใช้งานเครื่องไมโครคอมพิวเตอร์ใช้ในการควบคุมการบันทึกเสียงเก็บเป็นไฟล์ข้อมูลและการรายงานผลเป็นเสียงพูด

5.1.2.สามารถออกแบบเครื่องต้นแบบของระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติที่สามารถนำไปใช้งานได้

5.1.3.สามารถเขียนโปรแกรมฐานข้อมูลเพื่อเก็บข้อมูลของนักศึกษาและสามารถค้นหาข้อมูลได้

5.1.4.สามารถสร้างเครื่องต้นแบบของระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติและเขียนโปรแกรมควบคุมการทำงานของระบบ โดยสามารถติดต่อกับโทรศัพท์ได้

5.2.ปัญหาในการดำเนินงาน

ปัญหาในการดำเนินการของระบบนั้น พบว่าปัญหาจะพบอยู่ในส่วนของโปรแกรม

5.2.1.เสียงพูดที่ออกจากเครื่องไม่มีความชัดเจน เนื่องจากความเร็วในการทำงานของเครื่องคอมพิวเตอร์ช้ากว่าการทำงานของชุดสัญญาณเสียง

5.2.2.การสุ่มสัญญาณเสียงที่จะเก็บเป็นแฟ้มข้อมูลเกิดการผิดเพี้ยน สืบเนื่องจากต้องรอเครื่องคอมพิวเตอร์ทำงานอื่นให้เสร็จก่อนถึงจะมารับแฟ้มข้อมูล

5.2.3.โปรแกรมที่ใช้ควบคุมแต่ละระบบเมื่อนำมาใช้ร่วมกัน จะเกิดการผิดพลาดในการทำงานบางส่วน

5.3.ข้อแนะนำในการพัฒนา

5.3.1.ควรใช้เครื่องคอมพิวเตอร์ที่มีความเร็วสูง ขนาด 32 บิทขึ้นไป

5.3.2.ในอนาคตควรสร้างเครื่องต้นแบบของระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติทำให้อยู่ในรูปของการ์ดที่เสียบบนสล็อตเครื่องคอมพิวเตอร์

5.3.3.โปรแกรมที่ใช้ควบคุมยังสามารถพัฒนาให้มีประสิทธิภาพสูงยิ่งขึ้น เพื่อให้ผู้ควบคุมใช้งานได้สะดวกยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมการทำงานของระบบแจ้งผลการสอบทางโทรศัพท์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************************************************************/
/*       This is Main Program for Main Menu and Data Base            */
/***************************   Main   ********************************/
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<graphics.h>
#include<alloc.h>
#include<dos.h>
#include<stdlib.h>
#define UP 72
#define DOWN 80
#define ESC 27
#define ENTER 13
#define BACK_SPC 8
#define DEL 83
#define LEFT 75
#define RIGHT 77
#define INSERT 82
#define SPCBAR 32
#define HOME 71
#define END 79
#define PgUp 73
#define PgDn 81
#define F1 59
#define F2 60
#define F6 64
#define F10 68
#define NO_CHOICE 6
#define NO_MENU 2
#define No_sound  17
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Image_screen(void);
void Insert_Mode(void);
void Display_Dataget_word(int ll, char *buffer,int bkgnd_color);
void Display_Data(int ll, char *buffer,int bkgnd_color);
void Move_Data_buffer(char *source, char *dest, int buffer_lenght);
void Insert_Mode(void);
void Transfer_data(char *data,int Num,int lenght);
int Read_Key();

/*-------------- Function Sound_Menu ----------------*/

void menu_assignment(void);
void display_menu(int menu_no,int color,int text_color,int mode);
void select_menu(int menu_no,int choice_no,int color);
void unselect_menu(int menu_no,int choice_no,int color);
void inverse(int menu_no,int choice_no,int color);
void normal(int menu_no,int choice_no);

/*-------------- Function Graphics ------------------*/

void Initialize_Graphics_Mode(void);
void draw_fill_rectangle(int x1,int y1,int x2,int y2,int color);
void draw_rectangle(int lt,int tp,int rt,int bt,int color);
void erase_fill_rectangle(int lt,int tp,int rt,int bt,int color);
void fill_bar(int x1,int y1,int x2,int y2,int color,int speed);
void up_down_bar(int x,int y,int x1,int y1,int mode);
void up_down_bar_color(int x,int y,int x1,int y1,
int color1,int color2);
void ud_fill_bar(int x1,int y1,int x2,int y2,int color,
int speed,int mode);
void shardow(int x,int y,int x1,int y1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void shardow2(int x,int y,int x1,int y1,int color);
void erase_fill_bar(int x1,int y1,int x2,int y2,int color);
void set_fill_pattern(int pattern,int color);
void set_color(int color);
void out_text_xy(int x,int y,char text[],int color);
unsigned int image_size(int x1,int y1,int x2,int y2);
int get_max_x(void);
int get_max_y(void);

/*------------------ Cursor --------------------*/

void Draw_Cursor_Arrow(int x,int y);


/*-------------- DATABASE FILE -----------------*/

void DTB_wordprocessor(int No);
void get_word(int max,int X,int Y,int bkgnd_color);
void STUDENT_DTB(void);
void DTB_MENU(void);
void image_openfile(void);
void image_database(void);
void Image_blkgnd(void);
void Edit_DTB(void);
void sort(char Text[]);
void del_code(void);
void search(char text[]);
void Save_New_Edit(void);
void Save_Data(char Text[]);
void _Quit(void);
void copy_data(char *copy,int lp,int max);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Erase_Old_Rec(void);
void Erase_DTB(void);
void Erase_Buffer(char *buf,int le);
void Compare_REC(int lg);
void Cnt_Chck_Rec(void);
int compare(char text1[],char text2[]);

/*--------------- MAIN_MENU --------------------*/

void MAIN_MENU(void);
void Screen_main_menu();
void BckgndMAIN(void);

/*-------- Control MAIN_MENU & DTB_MENU -------*/

void select_KEY(int menu_no,int Old_No,int color,int Ocolor,
int textcolor);
void unclick_bar(int menu,int No_mn,int color,int textcolor);
void click_bar(int menu,int No_mn,int color,int textcolor);

/*---------------   VALUE   ------------------*/

void *cursor;
int new_X,new_Y,n_X,n_Y;
int Mode_Insert;
int s,MAX_LONG,Old_s,Seek_Save;
int count;
int step_X = 8,step_Y = 16;
char file_name[13];
char Code[9];
char Cnt_Record[6];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char *Charactor_Main =  "AUTOMATIC REPEATING TELEPHONE";
char *ALLbuf="                                          ";

/*------------  Record of SOUND_MENU --------------*/

  typedef struct heading {
char *choice;
   };
  typedef struct menu_struct {
  int frame[4];
  int row[NO_CHOICE];
  int col;
  struct heading item[NO_CHOICE];
  int last_choice;
   };

  struct menu_struct menus[NO_MENU];

 /*------------ Record of student_DTB --------------*/

   struct Std_Rec {
  char code[9];
  char Std_name[25];
  char grade[5];
  char AVgrade[5];
  }rec1,rec2;

  /*---------------- FUNCTION MAIN -------------------*/

  main()
     {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Initialize_Graphics_Mode();
menu_assignment();
BckgndMAIN();
select_KEY(1,0,8,7,14);
closegraph();
   }

/*------------ Initial Graphics_Mode --------------*/


void Initialize_Graphics_Mode(void)
 {
   int gd=DETECT,gm,errorcode;
   initgraph(&gd,&gm,"");

   errorcode=graphresult();

   if(errorcode != grOk)
{
 printf("graphics error: %s \n",grapherrormsg(errorcode));
 printf("press any key to halt:");
 getch();
 exit(1);

}
   }

/*----------------- Draw Block ---------------------*/

 void draw_rectangle(int lt,int tp,int rt,int bt,int color)
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  setcolor(color);
  rectangle(lt,tp,rt,bt);
  }
/*------------------ Del Block ----------------------*/

 void erase_fill_rectangle(int lt,int tp,int rt,int bt,int color)
  {
      set_fill_pattern(1,color);
      bar(lt,tp,rt,bt);
  }

/*----------------- Animate block --------------------*/

 void fill_bar(int x1,int y1,int x2,int y2,int color,int speed)
   {
    int i,j,new_x,new_y;

    new_x = (x1+x2)/2;
    new_y = (y1+y2)/2;
    i=new_x;j=new_y;
do{setfillstyle(1,color);
if(i>x1)
i -= speed;
if(j>y1)
j -= speed;
if(new_x<x2)
new_x += speed;
if(new_y<y2)
new_y += speed;
if(i<x1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(j<y1)
j += 1;
if(new_x>x2)
new_x -= 1;
if(new_y>y2)
new_y -= 1;
if(i>=x1 && j>=y1 && new_x<=x2 && new_y<=y2)
bar(i,j,new_x,new_y);
  }while((i!=x1 || j!=y1 || new_x!=x2 || new_y!=y2));
    }
/*--------------- Animate ERASE Block -----------------*/

 void erase_fill_bar(int x1,int y1,int x2,int y2,int color)
  {
int x,y;
x= (x1+x2)/2;
y= (y1+y2)/2;
setcolor(color);
do{rectangle(x1,y1,x2,y2);
    x1++; y1++; x2--; y2--;
  }while(x2 >= x && y2 >= y);
  }

/*--------------- Shardow of BLOCK -----------------*/

 void shardow(int x,int y,int x1,int y1)
  {
   int a,b,c,d,e;

   a=x1+1; b=y+15; c=x1+14; d=y1+14;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(e=0;e<2;e++) {
      while(b != d) {
while(a != c) {
      putpixel(a,b,0);
      if(a<c) a+=1;
      else a--;
     }putpixel(a,b,0);
      if(b<d) b+=2;
      else b--;
      if(e==0) a=x1+1;
      else a=x+15;
    } a=x+15;b=y1+1;c=x1;d=y1+14;
}
   }

/*---------------- Shardow of Block --------------------*/

void shardow2(int x,int y,int x1,int y1,int color)
{
 int cx=x,cy=y;
  while(cx!=x1) {
      putpixel(cx,cy,color);
      if(cx<x1) cx+=2;
      else cx--;
} putpixel(cx,cy,color);cx=x;
  while(cy != y1){
 putpixel(cx,cy,color);
 if(cy<y1) cy+=2;
 if(cy>y1) cy--;
      while(cx!=x1) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(cx<x1) cx+=2;
        if(cx>x1) cx--;
  ) putpixel(cx,cy,color);cx=x;
    }
      }

/*---------------- UP_DOWN ANIMATE BLOCK ------------------*/

void  ud_fill_bar(int x1,int y1,int x2,int y2,
int color,int speed,int mode)
 {
    int i,j,new_x,new_y,m,n;
    new_x = (x1+x2)/2;
    new_y = (y1+y2)/2;
    i=new_x;j=new_y;
do{setfillstyle(1,color);
if(j>x1)
i -= speed;
if(j>y1)
j -= speed;
if(new_x<x2)
new_x += speed;
if(new_y<y2)
new_y += speed;
if(i<x1)
i += 1;
if(j<y1)
j += 1;
if(new_x>x2)
new_x -= 1;
if(new_y>y2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
new_y -= 1;
if(i)=x1 && j)=y1 && new_x(=x2 && new_y(=y2)
bar(i,j,new_x,new_y);
  }while((i!=x1 || j!=y1 || new_x!=x2 || new_y!=y2));
    if(mode == 1)
setcolor(15);
    else setcolor(0);
   for(n=0;n(2;n++) {
line(x1,y1+n,x2,y1+n);
line(x1+n,y1,x1+n,y2);
     }
    if(mode == 1)
setcolor(0);
    else setcolor(15);

   for(m=0;m(2;m++) {
line(x1+m,y2-m,x2-m,y2-m);
line(x2-m,y1+m,x2-m,y2-m);
    }
}

/*-------------- UP_DOWN BLOCK two COLOR -----------------*/

 void up_down_bar(int x,int y,int x1,int y1,int mode)
  {  int n;
 for(n=1;n(=2;n++) {
if(mode==0)
   setcolor(0);
if(mode==1)
   setcolor(15);
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
line(x-n,y-n,x-n,y1+n);
if(mode==0);
   setcolor(15);
if(mode==1)
   setcolor(0);
line(x1+n,y-n,x1+n,y1+n);
line(x-n,y1+n,x1+n,y1+n);
   }
     }
/*---------------- UP_DOWN BLOCK 16 COLOR ----------------*/
 void up_down_bar_color(int x,int y,int x1,int y1,
 int color1,int color2)
  {  int n;
 for(n=1;n<=2;n++) {
   setcolor(color1);
line(x-n,y-n,x1,y-n);
line(x-n,y-n,x-n,y1+n);
   setcolor(color2);
line(x1+n,y-n,x1+n,y1+n);
line(x-n,y1+n,x1+n,y1+n);
   }
     }

/*---------------- SET STYLE & COLOR TO FILL ---------------*/

 void set_fill_pattern(int pattern,int color)
  {
      setfillstyle(pattern,color);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*---------------- Display Text color XY ------------------*/

 void out_text_xy(int x,int y,char text[],int color)
  {
       set_color(color);
       settextstyle(2,0,5);
       outtextxy(x,y,text);
  }

/*--------------------- SET COLOR -------------------------*/

 void set_color(int color)
  {
       setcolor(color);
  }

/*--------------- MAXIMUM OF Coordinate X ----------------*/

 int get_max_x(void)
  {
       return(getmaxx());
  }

/*-------------- MAXIMUM OF Coordinate Y ----------------*/

 int get_max_y(void)
  {
       return(getmaxy());
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*------------------- COLOR BLOCK -------------------------*/

void draw_fill_rectangle(int x1,int y1,int x2,int y2,int color)
{
 setfillstyle(1,color);
 bar(x1,y1,x2,y2);
 }

/*----------------- RETURN SIZE OF PIXEL ------------------*/

 unsigned int image_size(int x1,int y1,int x2,int y2)
   {
      return (imagesize(x1,y1,x2,y2));
   }

/*--------------------- DRAW BLOCK -------------------------*/

 void Draw_Rectangle(int lt,int tp,int rt,int bt)
   {
   rectangle(lt,tp,rt,bt);
   }

/*------------------ DISPLAY CURSOR ----------------------*/

 void Draw_Cursor_Arrow(int x,int y)
  {
   int i,j;
   unsigned int size;

    size = image_size(x,y,x+step_X-1,y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cursor = malloc(size);
getimage(x,y,x+step_X-1,y,cursor);

    while(!kbhit())
     {
setcolor(14);
line(x,y,x+step_X-1,y);
delay(80);
setcolor(0);
line(x,y,x+step_X-1,y);
delay(80);
    }
       putimage(x,y,cursor,0);
free(cursor);
     }
/*-------------------------------------------------------------*/

void get_word(int max,int X,int Y,int bkgnd_color)
 {
   int lenght;
   char *buffer = "                    ";

   n_X=X;
   n_Y=Y;
   MAX_LONG = max;
   lenght = 0;
   Mode_Insert =! INSERT;
   Insert_Mode();
   Draw_Cursor_Arrow(n_X,n_Y);
   do{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        s = Read_key();
    if(s==0) {
       s=getch();

    switch(s) {

       case HOME : while(lenght)
    {
    lenght--;
    buffer--;
    n_X-=step_X;
    }
    Draw_Cursor_Arrow(n_X,n_Y);
    break;

       case END  : while(lenght < MAX_LONG )
    {
       buffer++;
       lenght++;
       n_X+=step_X;
    }
    while(*buffer == ' ' && lenght > 0)
    {
       buffer--;
       lenght--;
       n_X-=step_X;
    }
    if(lenght+1 != MAX_LONG && lenght != 0 )
      {
       n_X+=step_X;
       buffer++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lenght++;
}
    Draw_Cursor_Arrow(n_X,n_Y);
    break;


    case LEFT :
        if(lenght >= 1) {
 lenght--;
 buffer--;
 n_X-=step_X;
 }
 Draw_Cursor_Arrow(n_X,n_Y);
 break;

    case RIGHT : if(lenght < MAX_LONG-1) {
    lenght++;
    buffer++;
    n_X+=step_X;
    }
    Draw_Cursor_Arrow(n_X,n_Y);
    break;

    case INSERT: Insert_Mode();
Draw_Cursor_Arrow(n_X,n_Y);
break;

 case DEL : Move_Data_buffer(buffer+1,buffer,MAX_LONG-lenght+1);
       *(buffer + MAX_LONG-lenght-1)=' ';
       Display_Dataget_word(lenght, buffer,bkgnd_color);
       Draw_Cursor_Arrow(n_X,n_Y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

default    : Draw_Cursor_Arrow(n_X,n_Y);break;

}

}

     else {

switch(s) {

case SPCBAR  : Draw_Cursor_Arrow(n_X,n_Y);
  break;

case BACK_SPC:if(lenght > 0) {
 lenght--;
 Move_Data_buffer(buffer,buffer-1,MAX_LONG-lenght);
 *(buffer + MAX_LONG-lenght-1)=' ';
 buffer--;
 n_X-=step_X;
 Display_Dataget_word(lenght, buffer,bkgnd_color);
   }
 Draw_Cursor_Arrow(n_X,n_Y);
 break;

case ESC   :   _Quit();
Draw_Cursor_Arrow(n_X,n_Y);
break;

case ENTER :  copy_data(buffer,lenght,MAX_LONG);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

default    : Draw_Cursor_Arrow(n_X,n_Y);break;

}

}

     else {

switch(s) {

case SPCBAR  : Draw_Cursor_Arrow(n_X,n_Y);
  break;

case BACK_SPC:if(lenght > 0) {
 lenght--;
 Move_Data_buffer(buffer,buffer-1,MAX_LONG-lenght);
 *(buffer + MAX_LONG-lenght-1)=' ';
 buffer--;
 n_X-=step_X;
 Display_Dataget_word(lenght, buffer,bkgnd_color);
   }
 Draw_Cursor_Arrow(n_X,n_Y);
 break;

case ESC   :   _Quit();
Draw_Cursor_Arrow(n_X,n_Y);
break;

case ENTER :  copy_data(buffer,lenght,MAX_LONG);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Move_Data_buffer(char *source, char *dest, int buffer_lenght)
  {

   if( dest > source)
while( buffer_lenght-- )
   *(dest + buffer_lenght) = *(source + buffer_lenght);
   else
 while( buffer_lenght-- )
     *dest++ = *source++;
  }

/*----------------- DISPLAY INSERT ----------------------*/

void Insert_Mode(void)
  {
     if(Mode_Insert==INSERT) {
Mode_Insert =! INSERT;
draw_fill_rectangle(34*step_X-2,3*step_Y-2,42*step_X+2,
4*step_Y+7,7);
out_text_xy(35*step_X,4*step_Y-12,"NORMAL",0);
out_text_xy(35*step_X+1,4*step_Y-11,"NORMAL",14);
  }
     else {
 Mode_Insert = INSERT;
 draw_fill_rectangle(34*step_X,3*step_Y,42*step_X,4*step_Y+5,7);
 up_down_bar(34*step_X-2,3*step_Y-2,42*step_X+2,4*step_Y+5+2,0);
 up_down_bar(34*step_X,3*step_Y,42*step_X,4*step_Y+5,1);
 out_text_xy(35*step_X,4*step_Y-12,"INSERT",0);
 out_text_xy(35*step_X+1,4*step_Y-11,"INSERT",14);
     }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*------------- DISPLAY TEXT FORM BUFFER ----------------*/

 void Display_Dataget_word(int ll, char *buffer,int bkgnd_color)
  {
   char ch,bh;
   int x=n_X,y=n_Y+3;

    while(ll < MAX_LONG) {
if(*(buffer)==' ') {
   setcolor(bkgnd_color);
   settextstyle(0,0,2);
   outtextxy(x-2,y-step_Y-1," "); }
else {
   setcolor(bkgnd_color);
   settextstyle(0,0,2);
   outtextxy(x-2,y-step_Y-1," ");
   out_text_xy(x,y-step_Y,buffer,14); }

if(ll==MAX_LONG)
   break;
else {
    ll++;
    buffer++;
    x+=step_X;
}
   }
  }

/*--------------- COPY DATA FROM BUFFER ------------------*/

 void copy_data(char *copy,int lp,int max)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    int i=0;
    while(lp--)
 {
  copy--;
 }
    while(lp++ < MAX_LONG-1)
{
   if(max==12) file_name[i]=toupper(*copy);
   if(max==8)  Code[i]=toupper(*copy);
   else ;
   copy++;
   i++;
}
  }

/*-------------------- Erase_Buffer ----------------------*/

 void Erase_Buffer(char *buf,int le)
 {
  while(le > 0)
  {
   buf--;
   le--;
  }
  while(le < MAX_LONG)
  {
   *buf= ' ';
   buf++;
   le++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while(le > 0) {
buf--;
le--;
}
  }

/*---------------- Std_DTBase --------------*/

 void STUDENT_DTB(void)
   {    Image_blkgnd();
image_database();
image_openfile();
image_key_ctrl();
DTB_wordprocessor(0);
   }

/*-------------- BACKGROUND SCREEN OF DATABASE ------------*/

 void Image_blkgnd(void)
{
   int x,i;

   draw_fill_rectangle(0,0,getmaxx(),getmaxy(),3);
   up_down_bar(2,2,getmaxx()-2,getmaxy()-2,1);
   up_down_bar(4,4,getmaxx()-4,getmaxy()-4,0);

   fill_bar(100,120,getmaxx()-100,getmaxy()-80,8,3);
   up_down_bar(100,120,getmaxx()-100,getmaxy()-80,0);
   fill_bar(100,120,getmaxx()-100,getmaxy()-80,7,3);
   fill_bar(100,120,getmaxx()-100,getmaxy()-80,9,3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fill_bar(15,8,getmaxx()-15,40,7,4);
    up_down_bar(15,8,getmaxx()-15,40,1);
    out_text_xy(200,18,"DATABASE FILE FOR USER",0);
    out_text_xy(201,18,"DATABASE FILE FOR USER",0);
    out_text_xy(202,19,"DATABASE FILE FOR USER",14);

    out_text_xy(getmaxx()-241,61,"File_name is :",8);
    out_text_xy(getmaxx()-240,62,"File_name is :",14);
    draw_fill_rectangle(getmaxx()-130,60,getmaxx()-22,77,8);
    up_down_bar(getmaxx()-130,60,getmaxx()-22,77,0);
}

/*------------------ Erase DATABASE --------------------*/

 void Erase_DTB(void)
  {
  erase_fill_bar(100,120,getmaxx()-100,getmaxy()-80,15);
  fill_bar(250,270,getmaxx()-250,getmaxy()-230,7,3);
  fill_bar(100,120,getmaxx()-100,getmaxy()-80,8,3);
  fill_bar(100,120,getmaxx()-100,getmaxy()-80,9,3);
  }

/*--------------- BLOCK OF FILENAME ---------------------*/

 void image_openfile(void)
 {
  void *blk;
  unsigned int size;

  size = image_size(160,200,465,265);
  blk = malloc(size);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   getimage(160,200,465,265,blk);

   fill_bar(160,200,450,250,1,1);
   draw_rectangle(160,200,450,250,14);
   shardow(160,200,450,250);
   out_text_xy(170,215,"Filename to created :",14);
   draw_fill_rectangle(330,211,438,232,12);
   draw_rectangle(330,211,438,232,14);
   get_word(12,335,228,12);
   putimage(160,200,blk,0);
   free(blk);
   draw_fill_rectangle(getmaxx()-130,60,getmaxx()-22,77,8);
   out_text_xy(getmaxx()-125,62,file_name,14);
   }

/*----------------- SCREEN DATABASE --------------------*/

 void image_database(void)
{
   out_text_xy(130,180-2*step_Y,"record.No :",4);
   out_text_xy(130,180,"Student code         :",14);
   out_text_xy(130,180+3*step_Y,"Student name          :",14);
   out_text_xy(130,180+6*step_Y,"Student grade         :",14);
   out_text_xy(130,180+9*step_Y,"Student Average grade :",14);
   draw_fill_rectangle(225,179-2*step_Y,270,196-2*step_Y,12);
   up_down_bar(225,179-2*step_Y,270,196-2*step_Y,0);
   draw_fill_rectangle(325,179,402,196,1);
   up_down_bar(325,179,402,196,0);
   draw_fill_rectangle(325,179+3*step_Y,529,196+3*step_Y,1);
   up_down_bar(325,179+3*step_Y,529,196+3*step_Y,0);
   draw_fill_rectangle(325,179+6*step_Y,370,196+6*step_Y,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    up_down_bar(325,179+6*step_Y,370,196+6*step_Y,0);
    draw_fill_rectangle(325,179+9*step_Y,370,196+9*step_Y,1);
    up_down_bar(325,179+9*step_Y,370,196+9*step_Y,0);
}

/*---------------------------------------*/

void DTB_wordprocessor(int No)
 {
  char *buffer;
  int i,s_Y=16,lenght;

  buffer = ALLbuf;
  No=0;
  MAX_LONG = 8;
  lenght   = 0;
  new_X=330;new_Y=177+s_Y;
  Text_key();
  Cnt_Chck_Rec();
  Draw_Cursor_Arrow(new_X,new_Y);

  do {
     s = Read_key();
if(s == 0) {
 s =  getch();

    switch(s) {
case F1 : Key_F1();
  Draw_Cursor_Arrow(new_X,new_Y);
  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  case F2 : Key_F2();
while(lenght > 0) {
     buffer--;
      lenght--;
}
     i=0;
    while(lenght < MAX_LONG) {
  if(No==0)  rec1.code[i] = *buffer;
  if(No==1)  rec1.Std_name[i] = *buffer;
  if(No==2)  rec1.grade[i] = *buffer;
  if(No==3)  rec1.AVgrade[i]= *buffer;
  i++;
  buffer++;
  lenght++;
}
    while(lenght > 0) {
  buffer--;
  lenght--;
    }
    Erase_Buffer(buffer,lenght);
    lenght = 0;

  if(Old_s==F10) {
                          Save_New_Edit();
  }
  else  Save_Data(file_name);
    Erase_DTB();
    image_database();
    Cnt_Chck_Rec();
    Erase_Buffer(buffer,lenght);
    new_X=330;new_Y=177+s_Y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MAX_LONG = 8;
    lenght = 0;
    Erase_Old_Rec();
    No = 0;
    Old_s = 0;
    Draw_Cursor_Arrow(new_X,new_Y);
    break;

case F6 : Key_F6();
    _Quit();
     Draw_Cursor_Arrow(new_X,new_Y);
     break;

case F10 : Key_F10();
     Erase_Buffer(buffer,lenght);
     select_KEY(0,0,8,4,14);
     image_key_ctrl();
     Text_key();
      new_X=330;new_Y=177+s_Y;
      MAX_LONG = 8;
      lenght = 0;
      No = 0;
     Draw_Cursor_Arrow(new_X,new_Y);
     break;

case UP  : while(lenght > 0) {
      buffer--;
      lenght--;
}
i=0;
     while(lenght < MAX_LONG) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(No==0)  rec1.code[i] = *buffer;
if(No==1)  rec1.Std_name[i] = *buffer;
if(No==2)  rec1.grade[i] = *buffer;
if(No==3)  rec1.AVgrade[i]= *buffer;
i++;
buffer++;
lenght++;
}
     while(lenght > 0) {
     buffer--;
     lenght--;
    }
     Erase_Buffer(buffer,lenght);
     lenght=0;
if(No==0) {
     No = 3;
     new_Y += 9*s_Y;
     MAX_LONG = 4;
  }
else { if(No==1) MAX_LONG = 8;
if(No==2) MAX_LONG = 24;
if(No==3) MAX_LONG = 4;
new_Y-=3*s_Y;
No--;
     }
     new_X=330;
     Transfer_data(buffer,No,lenght);
     Draw_Cursor_Arrow(new_X,new_Y);
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     case DOWN :  while(lenght > 0) {
```

```
buffer--;
lenght--;
  }
     i=0;
while(lenght < MAX_LONG) {
   if(No==0)  rec1.code[i] = *buffer;
   if(No==1)  rec1.Std_name[i] = *buffer;
   if(No==2)  rec1.grade[i] = *buffer;
   if(No==3)  rec1.AVgrade[i]= *buffer;
   i++;
   buffer++;
   lenght++;
}
while(lenght > 0) {
   buffer--;
   lenght--;
    }
   Erase_Buffer(buffer,lenght);
   lenght=0;
if(No==3) {
   No = 0;
   MAX_LONG = 8;
   new_Y -= 9*s_Y;
   }
else { if(No==0) MAX_LONG = 24;
if(No==1) MAX_LONG = 4;
if(No==2) MAX_LONG = 4;
      new_Y+=3*s_Y;
                              No++;
      }
      new_X=330;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Transfer_data(buffer,No,lenght);
        Draw_Cursor_Arrow(new_X,new_Y);
    break;

      case HOME : while(lenght)
    {
    lenght--;
    buffer--;
    new_X-=step_X;
    }
    Draw_Cursor_Arrow(new_X,new_Y);
    break;

      case END  : while(lenght < MAX_LONG)
    {
      buffer++;
      lenght++;
      new_X+=step_X;
    }
    while(*buffer == ' ' && lenght > 0)
    {
      buffer--;
      lenght--;
      new_X-=step_X;
    }
    if(lenght+1 != MAX_LONG )
      {
      new_X+=step_X;
      buffer++;
      lenght++;
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Draw_Cursor_Arrow(new_X,new_Y);
   break;

   case LEFT :
      if(lenght >= 1) {
 lenght--;
 buffer--;
 new_X-=step_X;
 }
 Draw_Cursor_Arrow(new_X,new_Y);
 break;

   case RIGHT : if(lenght < MAX_LONG-1) {
   lenght++;
   buffer++;
   new_X+=step_X;
   }
   Draw_Cursor_Arrow(new_X,new_Y);
   break;

   case INSERT: Insert_Mode();
Draw_Cursor_Arrow(new_X,new_Y);
break;

   case DEL : Move_Data_buffer(buffer+1, buffer, MAX_LONG-lenght+1);
      *(buffer + MAX_LONG-lenght-1/*(strlen(buffer)-1)*/)=' ';
      Display_Data(lenght, buffer,1);
                     Draw_Cursor_Arrow(new_X,new_Y);
      break;

   default : Draw_Cursor_Arrow(new_X,new_Y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

}
   }
     else  {

     switch(s) {

    case ESC : _Quit();
   s=0;
   Draw_Cursor_Arrow(new_X,new_Y);
   break;

    case BACK_SPC: if(lenght > 0) {
     lenght--;
     Move_Data_buffer(buffer, buffer-1, MAX_LONG-lenght);
     *(buffer +MAX_LONG-lenght-1/*(strlen(buffer)-1)*/)=' ';
     buffer--;
     new_X-=step_X;
     Display_Data(lenght, buffer,1);
}
     Draw_Cursor_Arrow(new_X,new_Y);
     break;


   case ENTER : while(lenght > 0)  {
      buffer--;
      lenght--;
}
     i=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(No==0)  rec1.code[i] = *buffer;
    if(No==1)  rec1.Std_name[i] = *buffer;
    if(No==2)  rec1.grade[i] = *buffer;
    if(No==3)  rec1.AVgrade[i]= *buffer;
    i++;
    buffer++;
    lenght++;
}
while(lenght > 0) {
    buffer--;
    lenght--;
}
    Erase_Buffer(buffer,lenght);
    lenght=0;
if(No==3) {
        No = 0;
        MAX_LONG = 8;
        new_Y -= 9*s_Y;
}
else { if(No==0) MAX_LONG = 24;
        if(No==1) MAX_LONG = 4;
        if(No==2) MAX_LONG = 4;
        new_Y+=3*s_Y;
        No++;
}
   new_X = 330;
   Transfer_data(buffer,No,lenght);
   Draw_Cursor_Arrow(new_X,new_Y);
   break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if(Mode_Insert != INSERT)
*buffer = s;
      else {
Move_Data_buffer(buffer,buffer+1,MAX_LONG-lenght-1);
*buffer = s; }
Display_Data(lenght, buffer,1);

      if(lenght < MAX_LONG-1) {
 lenght++;
 buffer++;
 new_X+=step_X;
}
 Draw_Cursor_Arrow(new_X,new_Y);
}
}
} while(s != ESC);
            Erase_Buffer(buffer,lenght);
}

/*------ display text DTB ---------*/

 void Display_Data(int ll, char *buffer,int bkgnd_color)
  {
   char ch,bh;
   int x=new_X,y=new_Y+3;

    while(ll < MAX_LONG) {
if(*(buffer)==' ') {
    draw_fill_rectangle(x,y-step_Y,x+step_X-1,y,bkgnd_color);
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    draw_fill_rectangle(x-2,y-step_Y,x+step_X-1,y,bkgnd_color);
    out_text_xy(x,y-step_Y,buffer,14); }

if(ll==MAX_LONG)
   break;
else {
    ll++;
    buffer++;
    x+=step_X;
}
   }
  }

/*------ Erase old student Record ------*/

 void Erase_Old_Rec(void)
  {
   int i=0,j,m;

   while(i < 4)
   {
    if(i==0) m=8;
    if(i==1) m=4;
    if(i==2) m=4;
    if(i==3) m=24;
for(j=0;j<m;j++) {
    if(i==0) rec1.code[j]    = ' ';
    if(i==1) rec1.AVgrade[j] = ' ';
    if(i==2) rec1.grade[j]   = ' ';
    if(i==3) rec1.Std_name[j]   = ' ';
                                                    } else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
}

/*-------------------- CTRL HOT KET ----------------------*/

 image_key_ctrl()
   {
   fill_bar(105,getmaxy()-45,getmaxx()-105,getmaxy()-15,7,5);
   up_down_bar(105,getmaxy()-45,getmaxx()-105,getmaxy()-15,0);
   up_down_bar(107,getmaxy()-43,getmaxx()-107,getmaxy()-17,1);
   up_down_bar(107,getmaxy()-43,210,getmaxy()-17,1);
   up_down_bar(215,getmaxy()-43,318,getmaxy()-17,1);
   up_down_bar(323,getmaxy()-43,426,getmaxy()-17,1);
   up_down_bar(431,getmaxy()-43,532,getmaxy()-17,1);
   }

 Text_key()
  {
out_text_xy(125,getmaxy()-35,"F1     F2      F6        F10",0);
out_text_xy(126,getmaxy()-34,"F1     F2      F6        F10",10);
out_text_xy(125,getmaxy()-35,"  -Help -Save -Exit -Menu",4);

  }

 Text_key_search()
{
  out_text_xy(125,getmaxy()-35,"  Pg_Dn  Pg_Up  ESC   ",0);
  out_text_xy(126,getmaxy()-34,"  Pg_Dn  Pg_Up  ESC   ",10);
  out_text_xy(125,getmaxy()-35,"                -Exit ",4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
Key_F1()
 {
   draw_fill_rectangle(105,getmaxy()-45,212,getmaxy()-15,7);
   Text_key();
   delay(400);
   up_down_bar(107,getmaxy()-43,210,getmaxy()-17,1);
   Text_key();
  }

Key_F2()
  {
   draw_fill_rectangle(213,getmaxy()-45,320,getmaxy()-15,7);
   Text_key();
   delay(400);
   up_down_bar(215,getmaxy()-43,318,getmaxy()-17,1);
   Text_key();
  }

Key_F6()
  {
   draw_fill_rectangle(321,getmaxy()-45,428,getmaxy()-15,7);
   Text_key();
   delay(400);
   up_down_bar(323,getmaxy()-43,426,getmaxy()-17,1);
   Text_key();
  }

Key_F10()
  {
   draw_fill_rectangle(429,getmaxy()-45,534,getmaxy()-15,7);
```

เอกสารนี้เป็นเอกสาร... ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Text_key();
    delay(400);
    up_down_bar(431,getmaxy()-43,532,getmaxy()-17,1);
    Text_key();
}

Key_PGD()
  {
    draw_fill_rectangle(105,getmaxy()-45,212,getmaxy()-15,7);
    Text_key_search();
    delay(400);
    up_down_bar(107,getmaxy()-43,210,getmaxy()-17,1);
    Text_key_search();
  }

Key_PGU()
  {
    draw_fill_rectangle(213,getmaxy()-45,320,getmaxy()-15,7);
    Text_key_search();
    delay(400);
    up_down_bar(215,getmaxy()-43,318,getmaxy()-17,1);
    Text_key_search();
  }

Key_ESC()
{
    draw_fill_rectangle(321,getmaxy()-45,428,getmaxy()-15,7);
    Text_key_search();
    delay(400);
    up_down_bar(323,getmaxy()-43,426,getmaxy()-17,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
/*--------------- EXIT TO MAIN_MENU --------------------*/

 void _Quit(void)
  {
   void *Exbk;
   char ch;
   unsigned int size;

  size = image_size(150,180,445,265);
  Exbk = malloc(size);
  getimage(150,180,445,265,Exbk);

  fill_bar(150,180,430,250,8,3);
  draw_rectangle(150,180,430,250,14);
  shardow(150,180,430,250);
  out_text_xy(160,190,"Do you want to Exit...[Y/N].?",14);
  draw_fill_rectangle(200,210,240,235,7);
  draw_fill_rectangle(260,210,300,235,7);
  draw_fill_rectangle(320,210,370,235,7);

  up_down_bar(260,210,300,235,1);
  up_down_bar(200,210,240,235,1);
  up_down_bar(320,210,370,235,1);

  draw_rectangle(198,208,242,237,0);
  draw_rectangle(258,208,302,237,0);
  draw_rectangle(318,208,372,237,0);
  out_text_xy(205,215,"Yes     No     Cancel",9);
  out_text_xy(205,217,"_       _      _",14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ch = toupper(getch());
if(ch == 'Y')
  {
   draw_fill_rectangle(199,209,241,236,7);
   out_text_xy(205,215,"Yes",0);
   out_text_xy(205,217,"_",14);
   delay(400);
   up_down_bar(200,210,240,235,1);
   draw_rectangle(198,208,242,237,0);
   delay(300);
   putimage(150,180,Exbk,0);
   free(Exbk);
   BckgndMAIN();
   select_KEY(1,0,8,7,14);
   }
if(ch == 'N')
   {
    draw_fill_rectangle(259,209,301,236,7);
    out_text_xy(205,215,"        No",0);
    out_text_xy(205,217,"        _",14);
    delay(400);
    up_down_bar(260,210,300,235,1);
    draw_rectangle(258,208,302,237,0);
    delay(300);
    }
 if(ch == 'C')
   {
    draw_fill_rectangle(319,209,371,236,7);
    out_text_xy(205,215,"                Cancel",0);
    out_text_xy(205,217,"                _",14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   up_down_bar(320,210,370,235,1);
   draw_rectangle(318,208,372,237,0);
   delay(300);
   }
     } while(ch != 'Y' && ch != 'N' && ch != 'C');
  putimage(150,180,Exbk,0);
  free(Exbk);
     }

/*--------------- SAVE CREATED DATA ---------------------*/

  void Save_Data(char Text[])
  {
   FILE *fp;
   unsigned int size;
   char ch=' ';
   void *blk;

    size = image_size(190,250,465,305);
    blk  = malloc(size);
    getimage(190,250,465,305,blk);

    fill_bar(190,250,450,290,12,1);
    shardow(190,250,450,290);
    draw_rectangle(190,250,450,290,14);
    draw_rectangle(192,252,448,288,14);
    out_text_xy(200,265,"Do you want SAVE DATA [Y/N]..?",14);

      while(ch != 'Y' && ch != 'N')
ch = toupper(getch());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (ch == 'Y') {

   if((fp=fopen(Text,"a"))==NULL)  {
out_text_xy(200,265,"Do you want SAVE DATA [Y/N]..7",12);
out_text_xy(220,265,"Couldn't Save :",14);
out_text_xy(340,265,Text,15);
getch();

 }
   else {
 fwrite(&rec1,sizeof rec1,1,fp);
 if(ferror(fp)) {
    out_text_xy(230,265,"Error in writting",14);
    getch();
} fclose(fp);
    } putimage(190,250,blk,0);
      sort(file_name);
    }
    else  putimage(190,250,blk,0);
  free(blk);
  }


/*----------- SEARCH DATA by Code ----------*/

 void search(char text[])
    {
     FILE *fp;
     void *block;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    unsigned int size;
    int c,No_rec,Od_rec,Mov,lg;

    size = image_size(160,200,465,265);
    block = malloc(size);
    getimage(160,200,465,265,block);

    fill_bar(160,200,450,250,1,1);
    draw_rectangle(160,200,450,250,14);
    shardow(160,200,450,250);
    out_text_xy(180,215,"Code for searching :",14);
    draw_fill_rectangle(355,211,430,232,12);
    draw_rectangle(355,211,430,232,14);
     get_word(8,360,228,12);

   if((fp = fopen(file_name,"rb"))==NULL) {
out_text_xy(180,215,"Code for searching :",1);
draw_fill_rectangle(355,211,430,232,1);
out_text_xy(230,215,"Error in READING file",14);
getch();
        putimage(160,200,block,0);
  }

    else { fseek(fp,0,2);
   No_rec = ftell(fp)/sizeof rec1;
   rewind(fp);

      while(fread(&rec1,sizeof rec1,1,fp)==1)
{     c=compare(text,rec1.code);
     if(c==1) { putimage(160,200,block,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
image_database();
count=ftell(fp)/sizeof rec1;
break;
      }
     if(ferror(fp)) {
out_text_xy(180,215,"Code for searching :",1);
draw_fill_rectangle(355,211,430,232,1);
out_text_xy(230,215,"Error in READING file",14);
getch();
putimage(160,200,block,0);
     }
 }

      if(c!=1){
  out_text_xy(180,215,"Code for searching :",1);
  draw_fill_rectangle(355,211,430,232,1);
  out_text_xy(230,215,"Error in READING file",1);
  out_text_xy(220,215,"No such code in this file",14);
   getch();
              putimage(160,200,block,0);
   }
     else {
   ltoa(count,Cnt_Record,10);
   lg=strlen(Cnt_Record);
   Compare_REC(lg);
   out_text_xy(327,180,rec1.code,14);
   out_text_xy(327,180+3*step_Y,rec1.Std_name,14);
   out_text_xy(327,180+6*step_Y,rec1.grade,14);
   out_text_xy(327,180+9*step_Y,rec1.AVgrade,14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Text_key_search();
  draw_fill_rectangle(431,getmaxy()-47,536,getmaxy()-13,3);
do {
   s=Read_key();
     if(s==0) {
s = getch();

      switch(s) {
    case PgUp :  Key_PGU();
 if(count>1) {
    Od_rec=ftell(fp)/sizeof rec1;
    Mov = Od_rec-2;
    fseek(fp,Mov*sizeof rec1,0);
    fread(&rec1,sizeof rec1,1,fp);
    count--;
      }
 break;

    case PgDn :  Key_PGD();
 if(count<No_rec) {
    fread(&rec1,sizeof rec1,1,fp);
    count++;
     }
 break;

}
       Erase_DTB();
       image_database();
       ltoa(count,Cnt_Record,10);
       lg=strlen(Cnt_Record);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        out_text_xy(327,180,rec1.code,14);
        out_text_xy(327,180+3*step_Y,rec1.Std_name,14);
        out_text_xy(327,180+6*step_Y,rec1.grade,14);
        out_text_xy(327,180+9*step_Y,rec1.AVgrade,14);

    }
       else {
     switch(s) {

     case ESC  :  Key_ESC();
  break;
        }
      }
     }while(s != ESC);
        }
  fclose(fp); }
  Erase_DTB();
  image_database();
  free(block);
  Erase_Old_Rec();
  s=0;
      }
  /*------------------ SORT BY CODE ----------------------*/

  void sort(char Text[])
     {
      FILE *fp;
      void *rct;
      unsigned int size;
      int i,j,n_record;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    size = image_size(190,250,465,305);
    rct  = malloc(size);
    getimage(190,250,465,305,rct);

    fill_bar(190,250,450,290,5,1);
    shardow(190,250,450,290);
    draw_rectangle(190,250,450,290,14);
    draw_rectangle(192,252,448,288,14);

     if((fp=fopen(Text,"r+b"))==NULL) {
out_text_xy(210,265,"Couldn't open file to SORT",14);
getch();
}
     else {
          out_text_xy(230,265,"DATA HAVE BEEN SORTED",14);
  fseek(fp,0,2);
  n_record = ftell(fp) / sizeof rec1;
  rewind(fp);

  for(i=1;i <= n_record-1;++i) {
     fread(&rec1,sizeof rec1,1,fp);
  for(j=1;j <= n_record-1;++j) {
     fread(&rec2,sizeof rec2,1,fp);
  if(strcmp(rec1.code,rec2.code) >0 ) {
     fseek(fp,ftell(fp)-2*sizeof rec1,0);
     fwrite(&rec2,sizeof rec2,1,fp);
     fwrite(&rec1,sizeof rec1,1,fp);
  if(ferror(fp)){
     out_text_xy(230,265,"DATA HAVE BEEN SORTED",5);
     out_text_xy(210,265,"File error in sortting",14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        fseek(fp,ftell(fp),0);
    }
        else
        rec1 = rec2;
    }
    rewind(fp);
      }
    }
    fclose(fp);
    delay(500);
            printf("\007");
    putimage(190,250,rct,0);
    free(rct);
        }

/*---------------------- DEL BY CODE ----------------------------*/

    void del_code(void)
        {
         FILE *fp;
         void *block;
         long count;
         unsigned int size;
         int c,lg;
         int n_record,i,record;
         char ch;
         struct Std_Rec *pt;

        size = image_size(160,200,465,265);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    getimage(160,200,465,265,block);

    if((fp = fopen(file_name,"r+b"))== NULL) {
        fill_bar(160,200,450,250,1,8);
 draw_rectangle(160,200,450,250,14);
 shardow(160,200,450,250);
 out_text_xy(230,215,"Error in READING file",14);
getch();
 free(block);
 putimage(160,200,block,0);
        Erase_DTB();
 image_database();
 BckgndMAIN();
 select_KEY(1,0,8,7,14);
  }

     else  {
    fill_bar(160,200,450,250,1,8);
    draw_rectangle(160,200,450,250,14);
    shardow(160,200,450,250);
    out_text_xy(180,215,"Code for Deletting :",14);
    draw_fill_rectangle(355,211,430,232,12);
    draw_rectangle(355,211,430,232,14);
 get_word(8,360,228,12);
    putimage(160,200,block,0);
    while(fread(&rec1,sizeof rec1,1,fp)==1)
{  count=ftell(fp)/sizeof rec1;
   c=compare(Code,rec1.code);

   if(c==1) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      image_database();
      break;
      }
    if(ferror(fp)) {
       fill_bar(160,200,450,250,1,8);
       draw_rectangle(160,200,450,250,14);
       shardow(160,200,450,250);
       out_text_xy(230,215,"Error in Reading",14);
       getch();
       putimage(160,200,block,0);
       fclose(fp);
       free(block);
       exit(1);
}
}
   if(c!=1) {
fill_bar(160,200,450,250,1,8);
draw_rectangle(160,200,450,250,14);
shardow(160,200,450,250);
out_text_xy(220,215,"No such code in file",14);
getch();
                putimage(160,200,block,0);

      }
    else {
record=ftell(fp)/sizeof rec1;
ltoa(count,Cnt_Record,10);
lg=strlen(Cnt_Record);
Compare_REC(lg);
out_text_xy(327,180,rec1.code,14);
out_text_xy(327,180+3*step_Y,rec1.Std_name,14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out_text_xy(327,180+6*step_Y,rec1.grade,14);
out_text_xy(327,180+9*step_Y,rec1.AVgrade,14);

  while(!kbhit()) {
out_text_xy(150,180+11*step_Y,
"HIT a key delete this RECORD <ESC to QUIT>",10);
delay(100);
out_text_xy(150,180+11*step_Y,"
HIT a key delete this RECORD <ESC to QUIT>",9);
delay(100);
  }
  s=Read_key();
    if(s==ESC) {
free(block);
fclose(fp);
            BckgndMAIN();
select_KEY(1,0,8,7,14);

}
  fseek(fp,0,2);
  n_record=ftell(fp)/sizeof rec1;

      if((pt = malloc(n_record*sizeof rec1))==NULL) {
        fill_bar(160,200,450,250,1,8);
 draw_rectangle(160,200,450,250,14);
 shardow(160,200,450,250);
 out_text_xy(230,215,"Couldn't DELETE",14);
 getch();
        putimage(160,200,block,0);
        free(pt);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
exit(0);
}
 rewind(fp);
  for(i=1;i <= n_record-1;++i) {

      if(ftell(fp) == (long)(record-1)*sizeof rec1)
      fseek(fp,sizeof rec1,1);
      fread(pt,sizeof rec1,1,fp);
      if(ferror(fp)){
 fill_bar(160,200,450,250,1,8);
 draw_rectangle(160,200,450,250,14);
 shardow(160,200,450,250);
 out_text_xy(230,215,"Error in reading",14);
 getch();
                putimage(160,200,block,0);
                free(pt);
 free(block);
 exit(0);
      }
++pt;
      }
 if((fp=fopen(file_name,"w+b"))==NULL) {
            fill_bar(160,200,450,250,1,8);
    draw_rectangle(160,200,450,250,14);
    shardow(160,200,450,250);
    out_text_xy(230,215,"Error in reading",14);
     getch();
     putimage(160,200,block,0);
     free(pt);
     free(block);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      select_KEY(1,0,8,7,14);
  }
  for(i=1;i<=n_record-1;++i)
  --pt;
  for(i=1;i<=n_record-1;++i) {
     fwrite(pt,sizeof rec1,1,fp);
       if(ferror(fp)){
  fill_bar(160,200,450,250,1,8);
  draw_rectangle(160,200,450,250,14);
  shardow(160,200,450,250);
  out_text_xy(230,215,"Error in writing",14);
  getch();
                  putimage(160,200,block,0);
                  free(pt);
  free(block);
  exit(0);
  }
  ++pt;
        }
  fill_bar(160,200,450,250,1,8);
  draw_rectangle(160,200,450,250,14);
  shardow(160,200,450,250);
  out_text_xy(230,215,"delete complete",14);
  delay(1000);
  putimage(160,200,block,0);
   }
  rewind(fp);
  fclose(fp);
  }
  if(c==1) free(pt);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
free(block);
Erase_Old_Rec();
                Erase_DTB();
image_database();
s=0;

    }

/*------------ Load data from code for Edit ------------*/

 void Edit_DTB(void)
  {
   FILE *fp;
   void *block;
   char *buffer;
   unsigned int size;
   int c,lg;
   long count;

    buffer = ALLbuf;
    size = image_size(160,200,465,265);
    block = malloc(size);
    getimage(160,200,465,265,block);

    Erase_DTB();
    image_database();
 fill_bar(160,200,450,250,1,1);
 draw_rectangle(160,200,450,250,14);
 shardow(160,200,450,250);
 out_text_xy(180,215,"Code for Editting :",14);
 draw_fill_rectangle(355,211,430,232,12);
 draw_rectangle(355,211,430,232,14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(s!=ESC)  {
     if((fp = fopen(file_name,"r+b"))==NULL) {
out_text_xy(180,215,"Code for Editting :",1);
draw_fill_rectangle(355,211,430,232,1);
out_text_xy(230,215,"Error in READING file",14);
getch();
}
else { rewind(fp);

      while(fread(&rec1,sizeof rec1,1,fp)==1)
 {    Seek_Save=ftell(fp)/sizeof rec1;
      c=compare(Code,rec1.code);
      if(c==1) { putimage(160,200,block,0);
break;
      }
      if(ferror(fp)) {
out_text_xy(180,215,"Code for Editting :",1);
draw_fill_rectangle(355,211,430,232,1);
out_text_xy(230,215,"Error in READING file",14);
getch();
      }
}  fclose(fp);

      if(c==1) {
      count = Seek_Save;
      ltoa(count,Cnt_Record,10);
      lg=strlen(Cnt_Record);
      Compare_REC(lg);
  out_text_xy(330,180,rec1.code,14);
  out_text_xy(330,180+3*step_Y,rec1.Std_name,14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    out_text_xy(330,180+9*step_Y,rec1.AVgrade,14);
    Transfer_data(buffer,0,0);
    Old_s=F10;
    }
        else {
    out_text_xy(180,215,"Code for Editting :",1);
    draw_fill_rectangle(355,211,430,232,1);
    out_text_xy(230,215,"Error in READING file",1);
    out_text_xy(220,215,"No such code in this file",14);
        getch();
        putimage(160,200,block,0);
        Erase_Old_Rec();
        Cnt_Chck_Rec();
     }
}
        }
        else putimage(160,200,block,0);
     free(block);

       }

/*------Save new_data from editting-------*/

 void Save_New_Edit(void)
  {
   FILE *fp;
   void *blk;
   unsigned int size;
   int c;
   char ch=' ';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    size = image_size(190,250,465,305);
    blk  = malloc(size);
    getimage(190,250,465,305,blk);

    fill_bar(190,250,450,290,12,1);
    shardow(190,250,450,290);
    draw_rectangle(190,250,450,290,14);
    draw_rectangle(192,252,448,288,14);
    out_text_xy(200,265,"You want SAVE NEW DATA [Y/N]..?",14);

     while(ch != 'Y' && ch != 'N')
   ch = toupper(getch());

 if(ch == 'Y') {
   if((fp = fopen(file_name,"r+b"))==NULL) {
out_text_xy(200,265,"You want SAVE NEW DATA [Y/N]..?",12);
out_text_xy(220,265,"Couldn't Save :",14);
out_text_xy(340,265,file_name,15);
getch();putimage(190,250,blk,0);
}
   else {
      fseek(fp,(Seek_Save-1)*sizeof rec1,0);
      fwrite(&rec1,sizeof rec1,1,fp);
if(ferror(fp)) {
out_text_xy(200,265,"You want SAVE NEW DATA [Y/N]..?",12);
out_text_xy(220,265,"Couldn't Save :",14);
out_text_xy(340,265,file_name,15);
getch();
  }
       putimage(190,250,blk,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
      else putimage(190,250,blk,0);
 fclose(fp);
 free(blk);
     }

/*------------------- String Compare ---------------------*/

  int compare(char text1[],char text2[])
  {
    int i,c=0;
    for(i=0;i<8;i++)
{
if(toupper(text1[i])==toupper(text2[i])) c++;
else c=0;
}
if(c==8) c=1;
else c=0;
return(c);
}

/*------------------ Count & check => rec ----------------*/

 void Cnt_Chck_Rec(void)
   {
    FILE *fp;
    void *BB;
    unsigned int size;

    long count_rec;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while((fp=fopen(file_name,"a"))==NULL) {
  size = image_size(200,150,415,195);
  BB = malloc(size);
  getimage(200,150,415,195,BB);
  fill_bar(200,150,400,180,8,1);
  draw_rectangle(200,150,400,180,14);
  shardow(200,150,400,180);
  out_text_xy(210,160,"Couldn't Created Filename",14);
  getch();
  putimage(200,150,BB,0);
  free(BB);
  image_openfile();
 }
      fseek(fp,0,2);
  count_rec=ftell(fp)/sizeof rec1;
  fclose(fp);
  count_rec++;
  ltoa(count_rec,Cnt_Record,10);
 lg=strlen(Cnt_Record);
 Compare_REC(lg);
}

/*--------------- Compare Number of Record --------------*/

 void Compare_REC(int lg)
    { int i=0;

      if(lg > 0 && lg < 6) {
 if(lg==1) {
    out_text_xy(228,181-2*step_Y,"0000",14);
    i=4*step_X;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
 if(lg==2) {
     out_text_xy(228,181-2*step_Y,"000",14);
     i=3*step_X;
    }
  if(lg==3) {
     out_text_xy(228,181-2*step_Y,"00",14);
     i=2*step_X;
    }
  if(lg==4) {
     out_text_xy(228,181-2*step_Y,"0",14);
     i=step_X;
    }
     out_text_xy(228+i,181-2*step_Y,Cnt_Record,14);
    }
      else  out_text_xy(228,181-2*step_Y,"FULL",14);
       }


/*------------- Copy std_record to buffer -----------------*/

  void Transfer_data(char *data,int Num,int lenght)
    { int i=0;

      while(lenght < MAX_LONG)  {
      if(Num==0) *data = rec1.code[i];
      if(Num==1) *data = rec1.Std_name[i];
      if(Num==2) *data = rec1.grade[i];
      if(Num==3) *data = rec1.AVgrade[i];
       data++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      lenght++;
   }
    while(lenght > 0)  {
  data--;
  lenght--;
  }
   }
/*----------------- record of menu ------------------------*/

  void menu_assignment(void)

   {

/* assign menu no. 0 menu File */
menus[0].frame[0] = 250;
menus[0].frame[1] = 160;
menus[0].frame[2] = getmaxx()-250;
menus[0].frame[3] = getmaxy()-177;

menus[0].row[0] = 195;
menus[0].row[1] = 218;
menus[0].row[2] = 241;
menus[0].row[3] = 264;
menus[0].row[4] = 287;

menus[0].col = 267;

menus[0].item[0].choice = "CREAT NEW FILE   ";
menus[0].item[1].choice = "SEARCH BY CODE ";
menus[0].item[2].choice = "EDIT BY CODE";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
menus[0].item[4].choice = "     EXIT";

menus[0].last_choice = 4;

/* assign menu no. 1menu File */
menus[1].frame[0] = 250;
menus[1].frame[1] = 160;
menus[1].frame[2] = getmaxx()-250;
menus[1].frame[3] = getmaxy()-201;

menus[1].row[0] = 195;
menus[1].row[1] = 218;
menus[1].row[2] = 241;
menus[1].row[3] = 264;
menus[1].row[4] = 287;

menus[1].col = 267;

menus[1].item[0].choice = "DATA BASE FILE";
menus[1].item[1].choice = "SOUND DATABASE";
menus[1].item[2].choice = "  OPERATING";
menus[1].item[3].choice = "     EXIT";
menus[1].item[4].choice = "";

menus[1].last_choice = 3;
}

/*--------------------- Database menu ----------------------*/

  void DTB_MENU(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fill_bar(250,160,getmaxx()-250,getmaxy()-177,0,1);
draw_fill_rectangle(252,186,getmaxx()-252,getmaxy()-179,4);
    draw_rectangle(250,160,getmaxx()-250,getmaxy()-177,14);
    draw_fill_rectangle(252,162,getmaxx()-252,181,15);
    draw_rectangle(251,161,getmaxx()-251,182,0);
    setcolor(14);
    line(250,183,getmaxx()-250,183);
    up_down_bar(254,188,getmaxx()-254,206,1);
    up_down_bar(254,211,getmaxx()-254,229,1);
    up_down_bar(254,234,getmaxx()-254,252,1);
    up_down_bar(254,257,getmaxx()-254,275,1);
    up_down_bar(254,280,getmaxx()-254,298,1);
    setcolor(8);
    settextstyle(0,0,1);
    outtextxy(267,167,"DATABASE MENU");
    setcolor(13);
    outtextxy(267,168,"DATABASE MENU");
    setcolor(14);
    settextstyle(0,0,1);
    outtextxy(267,195,"CREAT NEW FILE");
    outtextxy(267,218,"SEARCH BY CODE");
    outtextxy(267,241,"EDIT BY CODE");
    outtextxy(267,264,"DELETE BY CODE");
    outtextxy(267,287,"     EXIT");
    shardow(250,160,getmaxx()-250,getmaxy()-177);
}
/*---------------- Select key of main & DTB ----------------*/
  void select_KEY(int menu_no,int Old_No,int color,
  int Ocolor,int textcolor)
   {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    void *menu;
    unsigned int size;
    int new_No;

    size=image_size(menus[menu_no].frame[0]-5,
    menus[menu_no].frame[1]-5,
    menus[menu_no].frame[2]+20,menus[menu_no].frame[3]+20);

    menu = malloc(size);
    getimage(menus[menu_no].frame[0]-5,
    menus[menu_no].frame[1]-5,
    menus[menu_no].frame[2]+20,
    menus[menu_no].frame[3]+20,menu);

    click_bar(menu_no,Old_No,color,textcolor);
    if(menu_no==0) DTB_MENU();
    else MAIN_MENU();
    click_bar(menu_no,Old_No,8,14);
   do{
s=Read_key();
 if(s==0) {
    s=getch();
    switch(s) {

      case UP:if(Old_No==0) new_No=menus[menu_no].last_choice;
   else new_No = Old_No-1;
   unclick_bar(menu_no,Old_No,0color,textcolor);
   click_bar(menu_no,new_No,color,textcolor);
   Old_No=new_No;
    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case DOWN :  if(Old_No==menus[menu_no].last_choice)
        new_No = 0;
     else new_No = Old_No+1;
     unclick_bar(menu_no,Old_No,Ocolor,textcolor);
     click_bar(menu_no,new_No,color,textcolor);
     Old_No=new_No;
      break;
}
      }
else {
 switch(s) {
    case ENTER :
putimage(menus[menu_no].frame[0]-5,
menus[menu_no].frame[1]-5,menu,0);
free(menu);
     switch(menu_no) {
   case 0: switch(Old_No) {
        case 0: image_openfile();
        Erase_DTB();
        image_database();
        Cnt_Chck_Rec();
        break;

        case 1: search(Code);
        Cnt_Chck_Rec();
        break;

        case 2: Edit_DTB();
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Cnt_Chck_Rec();
        break;

        case 4:  BckgndMAIN();
select_KEY(1,0,8,7,14);
break;
}
s=ENTER;
break;
  case 1  : switch(Old_No) {
        case 0:  STUDENT_DTB();
break;

        case 1:  SOUND_Ctrl();
break;

        case 2:  Auto_Operate();
break;

        case 3:  closegraph();
exit(0);
break;
        }  break;
     }break;
}
     }

}while(s!=ENTER);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*---------------------- Key_Select ----------------------*/

   void click_bar(int menu,int No_mn,int color,int textcolor)
    {
     draw_fill_rectangle(menus[menu].col-15,
menus[menu].row[No_mn]-9,
 getmaxx()-252,menus[menu].row[No_mn]+13,color);
     setcolor(textcolor);
     settextstyle(0,0,1);
     outtextxy(menus[menu].col,menus[menu].row[No_mn],
menus[menu].item[No_mn].choice);
     }
/*---------------------- Unselect_Key ----------------------*/

   void unclick_bar(int menu,int No_mn,int color,int textcolor)
     {
     draw_fill_rectangle(menus[menu].col-15,
 menus[menu].row[No_mn]-9,
 getmaxx()-252,menus[menu].row[No_mn]+13,color);
     up_down_bar(menus[menu].col-13,menus[menu].row[No_mn]-7,
 getmaxx()-254,menus[menu].row[No_mn]+11,1);
     setcolor(textcolor);
     settextstyle(0,0,1);
     outtextxy(menus[menu].col,menus[menu].row[No_mn],
menus[menu].item[No_mn].choice);
}
/*--------------- Screen of Main_Menu ----------------*/

 void BckgndMAIN(void)
 { int i,j,k,l,m,n;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  char *Ch_Main;

  fill_bar(5,5,getmaxx()-5,getmaxy()-5,8,8);
  up_down_bar_color(5,5,getmaxx()-5,getmaxy()-5,15,7);

  fill_bar(10,10,getmaxx()-10,50,2,6);
  up_down_bar_color(10,10,getmaxx()-10,50,15,0);

  fill_bar(10,getmaxy()-50,getmaxx()-10,getmaxy()-10,1,6);
  up_down_bar_color(10,getmaxy()-50,getmaxx()-10,getmaxy()-10,15,0);

  up_down_bar_color(10,56,60,getmaxy()-53,15,0);
  fill_bar(10,56,60,getmaxy()-49,1,6);

  up_down_bar_color(getmaxx()-60,53,getmaxx()-10,
  getmaxy()-56,15,0);
  fill_bar(getmaxx()-60,48,getmaxx()-10,getmaxy()-56,2,6);
  setcolor(14);
  settextstyle(0,0,2);
  outtextxy(130,getmaxy()-35,"INDUSTRIAL EDUCATION");

for(j=0;j<2;j++) {
    i=68;
   setcolor(0);
   settextstyle(1,0,3);
   Ch_Main= Charactor_Main;
   while(*(Ch_Main) != '\0'){
*Pnt=*Ch_Main;
outtextxy(i+=15,15,Pnt);
delay(10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
  i=69;
  setcolor(7);
  Ch_Main= Charactor_Main;

  while(*(Ch_Main) != '\0'){
*Pnt=*Ch_Main;
outtextxy(i+=15,15,Pnt);
delay(10);
Ch_Main++;
}
  i=70;
  setcolor(14);
  Ch_Main=Charactor_Main;

  while(*(Ch_Main) != '\0'){
*Pnt=*Ch_Main;
outtextxy(i+=15,15,Pnt);
delay(10);
Ch_Main++;
}
}
}
/*------------ Main_Menu ------------------*/

 void MAIN_MENU(void)
 { int i;
  fill_bar(menus[1].frame[0]-1,menus[1].frame[1]+25,
   menus[1].frame[2],menus[1].frame[3],8,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      menus[1].frame[2],menus[1].frame[3],15,0);
    shardow(menus[1].frame[0]+2,menus[1].frame[1]+2,
    menus[1].frame[2]+2,menus[1].frame[3]+2);

    draw_fill_rectangle(menus[1].frame[0]-2,menus[1].frame[1],
        menus[1].frame[2]+1,menus[1].frame[1]+21,1);
    draw_rectangle(menus[1].frame[0]-2,menus[1].frame[1],
   menus[1].frame[2]+1,menus[1].frame[1]+21,12);
    settextstyle(0,0,2);
    out_text_xy(menus[1].frame[0]+30,menus[1].frame[1]+5,
  "Main MENU",8),
    out_text_xy(menus[1].frame[0]+31,menus[1].frame[1]+5,
  "Main MENU",14);
    for(i=0;i<4;i++) unclick_bar(1,i,7,14);
    }


  /*-------------------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************/
/*** This subprogram used for automatic search and *******/
/******* tell student grade on telephone line. ***************/
/*****************************************************************/
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<graphics.h>
#include<alloc.h>
#include<dos.h>
#include<stdlib.h>
#include<time.h>
#include<bios.h>
#define UP 72
#define DOWN 80
#define ESC 27
#define ENTER 13
#define BACK_SPC 8
#define DEL 83
#define LEFT 75
#define RIGHT 77
#define INSERT 82
#define SPCBAR 32
#define HOME 71
#define END 79
#define PgUp 73
#define PgDn 81
#define F1 59
#define F2 60
#define F6 64
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define F10 68
extern step_Y,step_X,Mode_Insert,n_X,n_Y,s,MAX_LONG;
extern struct Std_Rec {
      char code[9];
      char Std_name[25];
      char grade[5];
      char AVgrade[5];
}rec1,rec2;
typedef struct file_struct {
char sound_filename[10];
};
extern struct file_struct header_sound[17];
void Oprt_get_word(int max,int X,int Y,int bkgnd_color);
void Auto_Operate(void);
void BckGnd_Scr(void);
void Bckgnd_1(void);
void Erase_display(void);
void Cursor_Arrow(int x,int y);
void OprtMove_Data_buffer(char *source,char *dest,int buffer_lenght);
void Out_Data(int ll, char *buffer,int bkgnd_color);
void copy_Fname(char *copy,int lp);
void ErasePointer(char *buf,int le);
void Chck_file(void);
void Compare_tel(void);
void Read_Sndfilename(void);
void Waiting_Tel(void);
void sound_play(char play_name[]);
void Play_grade(void);
void Play_CODE(void);
void Play_Memo(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************/
void *Cursor;
char Fname[13];
int data_code;
char TelCode[9];
char *BFF = "                ";

/*--------- OPERATING AUTOMAIC ----------*/

void Auto_Operate(void)

{  Bckgnd_1();
 BckGnd_Scr();
 Oprt_get_word(12,262,169,15);
 Read_Sndfilename();
 Waiting_Tel();
 BckgndMAIN();
 select_KEY(1,0,8,7,14);
}

/*--------- BackGround Screen & database ------------*/

void BckGnd_Scr(void)
{
 fill_bar(130,120,getmaxx()-130,getmaxy()-121,7,2);
 up_down_bar_color(130,120,getmaxx()-130,getmaxy()-121,15,8);
 draw_rectangle(130,120,getmaxx()-130,getmaxy()-121,0);
 shardow(130,120,getmaxx()-130,getmaxy()-121);
 draw_fill_rectangle(130,120,getmaxx()-130,140,2);
 draw_rectangle(130,120,getmaxx()-130,140,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(0);
settextstyle(0,0,1);
outtextxy(230,125,"Automatic Searching");
setcolor(0);
settextstyle(0,0,1);
outtextxy(140,160,"Student File:");
outtextxy(140,160+2*step_Y,"Telephone Code:");

up_down_bar_color(180,getmaxy()-240,440,getmaxy()-135,8,15);
setcolor(0);
settextstyle(2,0,0);
outtextxy(210,162+6*step_Y,"Code :");
outtextxy(210,162+8*step_Y,"Name :");
outtextxy(210,162+10*step_Y,"GRADE :");
outtextxy(290,162+10*step_Y,"AVGRADE :");
draw_fill_rectangle(260,156,360,172,15);
up_down_bar_color(260,156,360,172,8,15);
draw_rectangle(259,155,361,173,8);

draw_fill_rectangle(270,156+2*step_Y,360,172+2*step_Y,15);
up_down_bar_color(270,156+2*step_Y,360,172+2*step_Y,8,15);
draw_rectangle(269,155+2*step_Y,361,173+2*step_Y,8);

draw_fill_rectangle(247,258,310,269,15);
up_down_bar_color(247,258,310,269,8,15);
draw_rectangle(246,257,311,270,0);

draw_fill_rectangle(247,258+2*step_Y,405,269+2*step_Y,15);
up_down_bar_color(247,258+2*step_Y,405,269+2*step_Y,8,15);
draw_rectangle(246,257+2*step_Y,406,270+2*step_Y,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 draw_fill_rectangle(350,258+4*step_Y,380,269+4*step_Y,15);
 up_down_bar_color(350,258+4*step_Y,380,269+4*step_Y,8,15);
 draw_rectangle(349,257+4*step_Y,381,270+4*step_Y,0);
 draw_fill_rectangle(247,258+4*step_Y,277,269+4*step_Y,15);
 up_down_bar_color(247,258+4*step_Y,277,269+4*step_Y,8,15);
 draw_rectangle(246,257+4*step_Y,278,270+4*step_Y,0); }

/*-------------- BackGround Screen -----------------*/

 void Bckgnd_1(void)
{
  draw_fill_rectangle(0,0,getmaxx(),getmaxy(),0);
  erase_fill_bar(5,5,getmaxx()-5,getmaxy()-5,1);
  fill_bar(5,5,getmaxx()-5,getmaxy()-5,3,8);
  fill_bar(5,5,getmaxx()-5,getmaxy()-5,15,8);
  fill_bar(5,5,getmaxx()-5,getmaxy()-5,3,8);
  up_down_bar_color(5,5,getmaxx()-5,getmaxy()-5,15,8);
  fill_bar(10,10,getmaxx()-10,35,8,2);
  up_down_bar_color(10,10,getmaxx()-10,35,15,0);
  setcolor(14);
  settextstyle(0,0,2);
  outtextxy(190,15,"AUTOMATIC OPERATING");
  }

/*--------------- Clear text ----------------------*/

 void Erase_display(void)
 {
  draw_fill_rectangle(247,258+4*step_Y,277,269+4*step_Y,15);
  draw_fill_rectangle(350,258+4*step_Y,380,269+4*step_Y,15);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  draw_fill_rectangle(247,258+2*step_Y,405,269+2*step_Y,15);
  draw_fill_rectangle(247,258,310,269,15);
  draw_fill_rectangle(270,156+2*step_Y,360,172+2*step_Y,15);
 }

/*--------------- DisPlay_CurSor ------------------*/

 void Cursor_Arrow(int x,int y)
  {
   int i,j;
   unsigned int size;

    size = image_size(x,y,x+step_X-1,y);
    Cursor = malloc(size);
    getimage(x,y,x+step_X-1,y,Cursor);
   while(!kbhit())
     {
setcolor(0);
line(x,y,x+step_X-1,y);
delay(100);
setcolor(15);
line(x,y,x+step_X-1,y);
delay(100);
    }
putimage(x,y,Cursor,0);
free(Cursor);
     }
/*-------------- Operate word processor -------------*/

 void Oprt_get_word(int max,int X,int Y,int bkgnd_color)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  int lenght;
  char *buffer;
  buffer = BFF;
  n_X=X;
  n_Y=Y;
  MAX_LONG = max;
  lenght = 0;
  Mode_Insert =! INSERT;
  Insert_Mode();
  Draw_Cursor_Arrow(n_X,n_Y);
  do{
     s = Read_key();
if(s==0) {
  s=getch();

switch(s) {
  case HOME : while(lenght)
{
lenght--;
buffer--;
n_X-=step_X;
}
Cursor_Arrow(n_X,n_Y);
break;
  case END  : while(lenght < MAX_LONG )
{
  buffer++;
  lenght++;
  n_X+=step_X; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while(*buffer == ' ' && lenght > 0)
{
   buffer--;
   lenght--;
   n_X-=step_X;
}
if(lenght+1 != MAX_LONG && lenght != 0 )
  {
   n_X+=step_X;
   buffer++;
   lenght++;
  }
   Cursor_Arrow(n_X,n_Y);
   break;
   case LEFT : if(lenght >= 1) {
 lenght--;
 buffer--;
 n_X-=step_X;
 }
 Cursor_Arrow(n_X,n_Y);
 break;
   case RIGHT : if(lenght < MAX_LONG-1) {
   lenght++;
   buffer++;
   n_X+=step_X;
   }
Cursor_Arrow(n_X,n_Y);
   break;
   case INSERT: Insert_Mode();
Cursor_Arrow(n_X,n_Y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
break;
 case DEL : OprtMove_Data_buffer(buffer+1,buffer,MAX_LONG-lenght+1);
      *(buffer + MAX_LONG-lenght-1)=' ';
      Out_Data(lenght, buffer,bkgnd_color);
      Cursor_Arrow(n_X,n_Y);
      break;
   default   : Cursor_Arrow(n_X,n_Y);break; } }
else {
switch(s) {

  case SPCBAR  : Cursor_Arrow(n_X,n_Y);
  break;
  case BACK_SPC:if(lenght > 0) {
 lenght--;
 OprtMove_Data_buffer(buffer,buffer-1,MAX_LONG-lenght);
 *(buffer + MAX_LONG-lenght-1)=' ';
 buffer--;
 n_X-=step_X;
 Out_Data(lenght, buffer,bkgnd_color);
   }
 Cursor_Arrow(n_X,n_Y);
 break;

  case ESC : _Quit();
     Cursor_Arrow(n_X,n_Y);
     break;
  case ENTER :  copy_Fname(buffer,lenght);
 while(lenght > 0)  {
lenght--;
buffer--;      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 ErasePointer(buffer,lenght);
 Chck_file();
 n_X=X;n_Y=Y;
 if(s==0) Cursor_Arrow(n_X,n_Y);
 else ;
 break;
  default :  if(Mode_Insert != INSERT)
*buffer = s;
      else {
   OprtMove_Data_buffer(buffer,buffer+1,MAX_LONG-lenght-1);
*buffer = s; }
Out_Data(lenght, buffer,bkgnd_color);

if(lenght < MAX_LONG-1) {
 lenght++;
 buffer++;
 n_X+=step_X;
 }
 Cursor_Arrow(n_X,n_Y);
  }
     }
   }while( s != ENTER);
    ErasePointer(buffer,lenght);
   }


/*------------------ SHIFT BUFFER --------------------*/

 void OprtMove_Data_buffer(char *source,char *dest,int buffer_lenght)
    {    if( dest > source)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while( buffer_lenght-- )
   *(dest + buffer_lenght) = *(source + buffer_lenght);
   else
 while( buffer_lenght-- )
     *dest++ = *source++;
    }


/*--------------- DISPLAY BUFFER TEXT -------------------*/

 void Out_Data(int ll, char *buffer,int bkgnd_color)
  {
   int x=n_X,y=n_Y+3;
      setcolor(1);
      settextstyle(0,0,1);
    while(ll < MAX_LONG) {
if(*(buffer)==' ') {
   draw_fill_rectangle(x,y-step_Y,x+step_X-1,y,bkgnd_color);
   }
else {
   draw_fill_rectangle(x,y-step_Y,x+step_X-1,y,bkgnd_color);
   outtextxy(x,y-12,buffer); }

if(ll==MAX_LONG)
   break;
else {
    ll++;
    buffer++;
    x+=step_X;
}    } }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*---------------- COPY BUFFER TO Fname ------------------*/

 void copy_Fname(char *copy,int lp)
  {
    int i=0;
    while(lp--)
{
  copy--;
}
    while(lp++ < MAX_LONG-1)
{
   Fname[i]=toupper(*copy);
   copy++;
   i++;
}
  }
/*-------------------- Erase_Buffer ----------------------*/
 void ErasePointer(char *buf,int le)
 {
  while(le > 0)
  {
   buf--;
   le--;
  }
  while(le < MAX_LONG)
  {
   *buf= ' ';
   buf++;
   le++;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  while(le > 0) {
buf--;
le--;
}   }
/*-------------- CHECK FILENAME TO OPEN ----------------*/

 void Chck_file(void)
  {
   FILE *fp;
   void *Block;
   unsigned int size;
   size = image_size(150,200,415,255);
   Block = malloc(size);
   getimage(150,200,415,255,Block);
   if((fp=fopen(Fname,"rb"))==NULL)
      {
fill_bar(150,200,400,236,8,6);
draw_rectangle(150,200,400,236,14);
shardow(150,200,400,236);
out_text_xy(170,210,"FILE Not FOUND:",15);
out_text_xy(290,210,Fname,14);
getch();
putimage(150,200,Block,0);
free(Block);
s=0;
}    else { fclose(fp);
  free(Block);
  Waiting_Tel();
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************/
/*          COMPARE TEL.CODE Vs STUDENT CODE                 */
/*   AND SEND SOUND GRADE ,SOUND CODE ON TEL. LINE           */
/*************************************************************/

void Compare_tel(void)
{
 FILE *fp;
 void *bk;
 unsigned int size;
 int c;

 size = image_size(150,200,415,255);


 bk = malloc(size);
 getimage(150,200,415,255,bk);

 fill_bar(150,200,400,236,8,6);
 draw_rectangle(150,200,400,236,14);
 shardow(150,200,400,236);
 out_text_xy(200,210,"Searching Code",14);
 delay(2500);

 fp= fopen(Fname,"r");
 while(fread(&rec1,sizeof rec1,1,fp)==1)
       { c = compare(TelCode,rec1.code);
         if(c==1) { putimage(150,200,bk,0);
                    break;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if(ferror(fp))
   {
     break;
   }
} /* while */
  fclose(fp);

if(c!=1) {
          fill_bar(150,200,400,236,8,6);
          draw_rectangle(150,200,400,236,14);
          shardow(150,200,400,236);
          out_text_xy(200,210,"No such CODE in file",14);
          delay(2000);
          putimage(150,200,bk,0);
         }
else  {
          fill_bar(200,200,350,236,8,6);
          draw_rectangle(200,200,350,236,14);
          shardow(200,200,350,236);
          out_text_xy(240,210,"CODE OK...!",14);
          delay(1000);
          putimage(150,200,bk,0);
          setcolor(0);
          Erase_display();
          settextstyle(0,0,1);
          outtextxy(272,160+2*step_Y,TelCode);
          settextstyle(2,0,4);
          outtextxy(249,258,rec1.code);
          outtextxy(249,258+2*step_Y,rec1.Std_name);
          outtextxy(249,258+4*step_Y,rec1.grade);
          outtextxy(352,258+4*step_Y,rec1.AVgrade);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Play_CODE();
                Play_grade();
                Play_Memo();
            }
                free(bk);
                Read_Sndfilename();
                Waiting_Tel();
        }

/*-------------- Check file 'FSOUND.DAT' ----------------*/

 void Read_Sndfilename(void)
  {
   FILE *fp;
   int i;

   if((fp=fopen("FSOUND.DAT","r"))==NULL)
      {
       fill_bar(140,200,430,240,15,1);
       draw_rectangle(140,200,430,240,14);
       out_text_xy(160,215,"This Program need FSOUND.DAT",0);
       getch();
       BckgndMAIN();
       select_KEY(1,0,8,7,14);
      }
    while(fread(&header_sound,sizeof header_sound,1,fp)==1);
         fclose(fp);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-----------------Wait Telephone ring--------------------*/

void Waiting_Tel(void)
 {
  void *bk;
  int check_ring,check_code,st;
  time_t start,end;
  char *ch=" ";
  unsigned int size;
  size = image_size(150,200,415,255);
  bk = malloc(size);
  getimage(150,200,415,255,bk);

  outportb(0x307,0x93);        /* control word 8255 #2 */
                               /*   Port A  <---       */
                               /*           B  <---    */
                               /*           Cl <---    */
                               /*           Cu --->    */

  outportb(0x306,0xff);        /* control port Cu "Hook off" */

      fill_bar(150,200,400,236,8,6);
      draw_rectangle(150,200,400,236,14);
      shardow(150,200,400,236);
      out_text_xy(200,210,"WAIT FOR TEL..RING",14);

 do
   { check_ring=inportb(0x306);
     check_ring=check_ring&1;
   } while(check_ring!=0);
          putimage(150,200,bk,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fill_bar(150,200,400,236,8,6);
    draw_rectangle(150,200,400,236,14);
    shardow(150,200,400,236);
    out_text_xy(200,210,"WAIT FOR TEL..CODE",14);

    outportb(0x306,0xef);        /* "Hook on" */
    delay(1);

    sound_play(header_sound[11].sound_filename);

    outportb(0x306,0xcf);       /* control port get DTMF */
    for(st=0;st<=7;st++)
      {  /* start=time('\0');    check start time */
 do
   { check_code=inportb(0x304);
     check_code=check_code&1;

/* end=time('\0');              check end time
 if(difftime(end,start)==10)    check not more 10 second
   {  outportb(0x306,0xFF);     "Hook off"
   }                            goto check ring
                                                  */
   } while(check_code!=1);

     data_code=inportb(0x305);        /* get code */
     data_code=data_code & 15;
  if(data_code==10)
     data_code=0;
     itoa(data_code,ch,10);
     TelCode[st]=*ch;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     do{ check_code=inportb(772);
         check_code=check_code&1;
        }while(check_code!=0);
           }
         putimage(150,200,bk,0);
         free(bk);
         Compare_tel();
     }

/*------------ Play sound filename ------------------*/

void sound_play(char play_name[] )
{
  FILE *f1;
  void *bk;
  unsigned char a;
  unsigned int size;

      outportb(0x303,0x99);

  if ((f1=fopen(play_name,"r"))!=NULL)
    { while((a=getc(f1))!= 0)
          { outportb(0x301,a);
            if(bioskey(1)!=0)
              { bioskey(0);
                break;
               }
           }
            fclose(f1);   }
    else {
       size = image_size(150,200,415,255);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        bk = malloc(size);
        getimage(150,200,415,255,bk);

        fill_bar(150,200,400,236,8,6);
        draw_rectangle(150,200,400,236,14);
        shardow(150,200,400,236);
        out_text_xy(200,210,"Couldn't Open file",14);
        delay(2500);


        putimage(150,200,bk,0);
        free(bk);
    }
        outportb(0x301,0x0);
    }

/*------------- SEND SOUND GRADE ON TEL. LINE -------------*/

 void Play_grade(void)
 {
   int i;
   sound_play(header_sound[0].sound_filename);
 for(i=0;i<4;i++) {
     switch(rec1.grade[i]) {

     case '0' :  sound_play(header_sound[0].sound_filename);
                 break;

     case '1' :  sound_play(header_sound[1].sound_filename);
                 break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case '2' :  sound_play(header_sound[2].sound_filename);
            break;

case '3' :  sound_play(header_sound[3].sound_filename);
            break;

case '4' :  sound_play(header_sound[4].sound_filename);
            break;

case '5' :  sound_play(header_sound[5].sound_filename);
            break;

case '6' :  sound_play(header_sound[6].sound_filename);
            break;

case '7' :  sound_play(header_sound[7].sound_filename);
            break;

case '8' :  sound_play(header_sound[8].sound_filename);
            break;

case '9' :  sound_play(header_sound[9].sound_filename);
            break;

case '.' :  sound_play(header_sound[10].sound_filename);
            break;

default :   break;
}
 }    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*------------ SEND SOUND CODE ON TEL.. LINE ----------------*/

 void Play_CODE(void)
 {
   int i;
   sound_play(header_sound[14].sound_filename);
   for(i=0;i<8;i++) {
       switch(TelCode[i]) {
     case '0' :  sound_play(header_sound[0].sound_filename);
                 break;

     case '1' :  sound_play(header_sound[1].sound_filename);
                 break;

     case '2' :  sound_play(header_sound[2].sound_filename);
                 break;

     case '3' :  sound_play(header_sound[3].sound_filename);
                 break;

     case '4' :  sound_play(header_sound[4].sound_filename);
                 break;

     case '5' :  sound_play(header_sound[5].sound_filename);
                 break;

     case '6' :  sound_play(header_sound[6].sound_filename);
                 break;

     case '7' :  sound_play(header_sound[7].sound_filename);
                 break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      case '8' :  sound_play(header_sound[8].sound_filename);
                  break;

      case '9' :  sound_play(header_sound[9].sound_filename);
                  break;

      case '.' :  sound_play(header_sound[10].sound_filename);
                  break;

      default  :  break;
    }
      }
}

/*-------------------- play memo ------------------------*/

 void Play_Memo(void)
  {
   sound_play(header_sound[15].sound_filename);
   sound_play(header_sound[16].sound_filename);
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************************************************************/
/*      This Subprogram used for Record sound to memory ,            */
/*      play sound which opened in memory and edit sound.            */
/*********************************************************************/

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<graphics.h>
#include<alloc.h>
#include<dos.h>
#include<stdlib.h>
#include<bios.h>

#define SHAPE_TABLE_X 7
#define SHAPE_TABLE_Y 12
#define coor_X 30*step_X
#define coor_Y 15*step_Y
#define UP 72
#define DOWN 80
#define ESC 27
#define ENTER 13
#define BACK_SPC 8
#define DEL 83
#define LEFT 75
#define RIGHT 77
#define INSERT 82
#define SPCBAR 32
#define HOME 71
#define END 79
#define F2 60
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define No_sound  17
#define NO_CHOICE 6
#define NO_MENU 1
#define SOUND_ZERO    header_sound[0].sound_filename
#define SOUND_ONE     header_sound[1].sound_filename
#define SOUND_TWO     header_sound[2].sound_filename
#define SOUND_THREE   header_sound[3].sound_filename
#define SOUND_FOUR    header_sound[4].sound_filename
#define SOUND_FIVE    header_sound[5].sound_filename
#define SOUND_SIX     header_sound[6].sound_filename
#define SOUND_SEVEN   header_sound[7].sound_filename
#define SOUND_EIGHT   header_sound[8].sound_filename
#define SOUND_NINE    header_sound[9].sound_filename
#define SOUND_POINT   header_sound[10].sound_filename
#define SOUND_TITLE1 header_sound[11].sound_filename
#define SOUND_TITLE2 header_sound[12].sound_filename
#define SOUND_TITLE3 header_sound[13].sound_filename
#define SOUND_END1    header_sound[14].sound_filename
#define SOUND_END2    header_sound[15].sound_filename
#define SOUND_END3    header_sound[16].sound_filename

extern step_X,step_Y,new_X,new_Y,MAX_LONG,s;
extern Mode_Insert;

/*--------------- sound menu ----------------*/

void display_menu(int menu_no,int color,int text_color,int mode);
void select_menu(int menu_no,int choice_no,int color);
void unselect_menu(int menu_no,int choice_no,int color);
void inverse(int menu_no,int choice_no,int color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void normal(int menu_no,int choice_no);
void image_edit_sound(void);
void image_record_sound(void);
void play_sound(void);
void Snd_menu_assignment(void);
void image_normal(int menu_no)

/*---------- sound database ----------------*/

void Transfer_Snd_data(char *data,int Num,int lenght);
void Snd_Move_Data_buffer(char *source, char *dest, int buffer_lenght);
void Snd_get_word(char *buff,int X,int Y,int bkgnd_color);
void Edit_Sound_word(char *buff,int No);
void bckgd(void);
void Read_file(void);
void Snd_Save_Data(void);
void SOUND_Ctrl(void);
void Snd_copy_data(char *copy,int lp);
void sound_playname(char play_name[] );
void all_sound_play(char play_name[] );
void Display_Sound_All(void);
void Image(void);


char Sndfile_name[10];
char *B_filename= "         ";
char *S_name =    "         ";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*------------  Record of SOUND_MENU --------------*/

 typedef struct heading {
         char *choice;
      };

 typedef struct menu_struct {
 int frame[4];
 int row[NO_CHOICE];
 int colum;
 struct heading item[NO_CHOICE];
 int last_choice;
  };

 struct menu_struct mnus[NO_MENU];

/*------------ Record of SOUND_FILENAME --------------*/

 typedef struct file_struct {
            char sound_filename[10];
  };

 struct file_struct header_sound[17];

/*-----------------------------------------------*/

 int start = 230,
     one_choice_width = 23;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------- SOUND DATABASE -----------------*/

void SOUND_Ctrl(void)
   {
     Image();
     Snd_menu_assignment();
     select_menu(0,1,0);
   }

/*------------- SOUND RECORD ---------------*/

void Snd_menu_assignment(void)
   {
   int one_part;
   one_part = 150;

/*  assign menu no. 0 menu About */
   mnus[0].frame[0] = start;
   mnus[0].frame[1] = 160+one_choice_width+4;
   mnus[0].frame[2] = start+one_part;
   mnus[0].frame[3] = 160+one_choice_width+one_choice_width*5;

   mnus[0].row[0] = 160;
   mnus[0].row[1] = 192;
   mnus[0].row[2] = 213;
   mnus[0].row[3] = 234;
   mnus[0].row[4] = 255;
   mnus[0].row[5] = 276;

   mnus[0].colum = start;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    mnus[0].item[0].choice = "  SOUND MENU    ";
    mnus[0].item[1].choice = "  1.  RECORD SOUND";
    mnus[0].item[2].choice = "  2.  EDIT SOUND  ";
    mnus[0].item[3].choice = "  3.  PLAY SOUND  ";
    mnus[0].item[4].choice = "  4.  PLAY ALL    ";
    mnus[0].item[5].choice = "  5.  exit MENU   ";

    mnus[0].last_choice = 5;
      }

 /*--------------- SOUND SCREEN ---------------*/

void bckgd(void)
 {
  fill_bar(190,140,getmaxx()-220,getmaxy()-140,3,3);
  up_down_bar_color(190,140,getmaxx()-220,getmaxy()-140,0,15);
  fill_bar(195,145,getmaxx()-225,170,13,3);
  up_down_bar_color(195,145,getmaxx()-225,170,8,15);
  settextstyle(0,0,1);
  setcolor(1);
  outtextxy(250,155,"SOUND MENU");
  setcolor(14);
  outtextxy(251,155,"SOUND MENU");
  }

/*------------ SOUND Wordprocessor ----------------*/

void Snd_get_word(char *buff,int X,int Y,int bkgnd_color)
 {
  int lenght;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char *buffer;

new_X=X;
new_Y=Y;
buffer=buff;
MAX_LONG = strlen(buffer);
lenght = 0;
Mode_Insert =! INSERT;
Insert_Mode();
Draw_Cursor_Arrow(new_X,new_Y);
do{
   s = Read_key();
  if(s==0) {
     s=getch();

     switch(s) {

case HOME : while(lenght)
                  {
                  lenght--;
                  buffer--;
                  new_X-=step_X;
                   }
                  Draw_Cursor_Arrow(new_X,new_Y);
                  break;

case END  : while(*(buffer+1) != '\0')
                  {
                  buffer++;
                  lenght++;
                  new_X+=step_X; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                while(*buffer == ' ' && lenght > 0)
                    {
                    buffer--;
                    lenght--;
                    new_X-=step_X;
                    }
               if(*(buffer+1) != '\0' && lenght != 0 )
                    {
                    new_X+=step_X;
                    buffer++;
                    lenght++;
                    }
                    Draw_Cursor_Arrow(new_X,new_Y);
                    break;

case LEFT : if(lenght >= 1) {
                   lenght--;
                  buffer--;
                  new_X-=step_X;
                   } Draw_Cursor_Arrow(new_X,new_Y);
                    break;


case RIGHT : if(lenght < MAX_LONG-1) {
                    lenght++;
                    buffer++;
                    new_X+=step_X;
                  } Draw_Cursor_Arrow(new_X,new_Y);
                    break;
case INSERT: Insert_Mode();
               Draw_Cursor_Arrow(new_X,new_Y);
               break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case DEL   : Snd_Move_Data_buffer(buffer+1, buffer,
                                  strlen(buffer+1));
             *(buffer + (strlen(buffer)-1))=' ';
             Display_Data(lenght, buffer,bkgnd_color);
             Draw_Cursor_Arrow(new_X,new_Y);
             break;

default    : Draw_Cursor_Arrow(new_X,new_Y);
             break;
             }
  }

 else {
    switch(s) {

case SPCBAR  : Draw_Cursor_Arrow(new_X,new_Y);
               break;

case BACK_SPC: if(lenght > 0) {
                  lenght--;
                  Snd_Move_Data_buffer(buffer, buffer-1,
                                       strlen(buffer));
                  *(buffer +(strlen(buffer)-1))=' ';
                  buffer--;
                  new_X-=step_X;
                  Display_Data(lenght,buffer,bkgnd_color);
                           }
                  Draw_Cursor_Arrow(new_X,new_Y);
                  break;

case ESC :    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case ENTER :  Snd_copy_data(buffer,lenght);
                  new_X = coor_X;
                  new_Y += step_Y;
                  break;
    default    :  if(Mode_Insert != INSERT)
                     *buffer = s;
                  else {
                     Snd_Move_Data_buffer(buffer,buffer+1,
                                          strlen(buffer)-1);
                     *buffer = s; }
                  Display_Data(lenght, buffer,bkgnd_color);

                  if(lenght < MAX_LONG-1) {
                     lenght++;
                     buffer++;
                     new_X+=step_X;
                      }
                  Draw_Cursor_Arrow(new_X,new_Y);
                      }
               }
   }while( s != ESC && s != ENTER);
          Erase_Buffer(buffer,lenght);
    }

 /*------------------ SHIFT BUFFER----------------------*/

void Snd_Move_Data_buffer(char *source,char *dest,int buffer_lenght)
 {
   if( dest > source)
       while( buffer_lenght-- )
              *(dest + buffer_lenght) = *(source + buffer_lenght);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else  while( buffer_lenght-- )
                *dest++ = *source++;
    }

/*------------- COPY BUFFER TO Sound_filename -----------*/

 void Snd_copy_data(char *copy,int lp)
  {
   int i=0;
      while(lp--)
          {
           copy--;
          }
      while(lp++ < MAX_LONG)
          {
           Sndfile_name[i]=toupper(*copy);
           copy++;
           i++;
          }
   }

/*------------ Screen sound_database --------------*/

 void Image(void)
    { fill_bar(4,4,getmaxx()-4,getmaxy()-4,9,7);
      fill_bar(4,4,getmaxx()-4,getmaxy()-4,1,7);
      ud_fill_bar(10,10,getmaxx()-10,40,13,4,1);
      out_text_xy((getmaxx()/3)+20,10+5," SOUND DATABASE",0);
      out_text_xy((getmaxx()/3)+20+2,10+2+5," SOUND DATABASE",14);
      bckgd();
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*----------------- display database menu ------------------*/

 void display_menu(int menu_no,int color,int text_color,int mode)
   {
   int i;

   if(mode==1)
      fill_bar(mnus[menu_no].frame[0],mnus[menu_no].frame[1],
      mnus[menu_no].frame[2],mnus[menu_no].frame[3],color,1);
   if(mode==0)
      draw_fill_rectangle(mnus[menu_no].frame[0],
                  mnus[menu_no].frame[1],mnus[menu_no].frame[2],
                  mnus[menu_no].frame[3],color);
      draw_rectangle(mnus[menu_no].frame[0]-1,
                  mnus[menu_no].frame[1]-1,
                  mnus[menu_no].frame[2]+1,
                  mnus[menu_no].frame[3]+1,text_color);
      draw_rectangle(mnus[menu_no].frame[0]-2,
                  mnus[menu_no].frame[1]-2,
                  mnus[menu_no].frame[2]+2,
                  mnus[menu_no].frame[3]+2,text_color);
      draw_rectangle(mnus[menu_no].frame[0],
                  mnus[menu_no].frame[1],
                  mnus[menu_no].frame[2],
                  mnus[menu_no].frame[3],text_color);

      for(i=1; i<NO_CHOICE; i++)
          out_text_xy(mnus[menu_no].colum,mnus[menu_no].row[i],
                  mnus[menu_no].item[i].choice,text_color);
         }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------*/

  void image_normal(int menu_no)
   { int j;
     for(j=1;j<NO_CHOICE;j++) {
         erase_fill_rectangle(mnus[menu_no].colum+4,
         mnus[menu_no].row[j],mnus[menu_no].colum+146,
         mnus[menu_no].row[j]+17,9);
         out_text_xy(mnus[menu_no].colum,mnus[menu_no].row[j],
                     mnus[menu_no].item[j].choice,14);
                              }
   }

/*---------------- select menu ----------------------*/

  void select_menu(int menu_no,int choice_no,int color)
   {
    int new_menu_no,
        new_choice_no;

   /*    default menu     */
   display_menu(menu_no,color,14,1);
   image_normal(0);
   inverse(menu_no,choice_no,12);

   for(;;) {
      s = Read_key();
        if(s==0) {
           s=getch();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
switch (s) {

case UP    : if(choice_no == 1)
                new_choice_no
                = mnus[menu_no].last_choice;
             else new_choice_no = choice_no-1;
                normal(menu_no,choice_no);
                inverse(menu_no,new_choice_no,12);
                choice_no = new_choice_no;
                break;

case DOWN : if(choice_no
             == mnus[menu_no].last_choice)
              new_choice_no = 1;
            else new_choice_no = choice_no+1;
             normal(menu_no,choice_no);
             inverse(menu_no,new_choice_no,12);
             choice_no = new_choice_no;
             break;
        }
    }

else {
     switch(s) {

case ENTER : erase_fill_bar(188,138,getmaxx()-218,
             getmaxy()-138,1);
             switch(menu_no) {
               case 0 : switch(choice_no) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    case 1 : image_record_sound();
                             break;

                    case 2 : image_edit_sound();
                             break;

                    case 3 : play_sound();
                             break;

                    case 4 : Display_Sound_All();
                             break;

                    case 5 : BckgndMAIN();
                             select_KEY(1,0,8,7,14);
                             break;
                            };
                             break;
                             };
case ESC : _Quit();
           break;
        }
     }
}
     }

/*-------------------------------------------------------*/

void inverse(int menu_no,int choice_no,int color)
  {
   if(choice_no == 0) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        draw_fill_rectangle(mnus[menu_no].colum+4,
        mnus[menu_no].row[choice_no],
        mnus[menu_no].colum+146,
        mnus[menu_no].row[choice_no]+17,color);
        }
      else {
        draw_fill_rectangle(mnus[menu_no].colum+4,
        mnus[menu_no].row[choice_no],
        mnus[menu_no].colum+146,
        mnus[menu_no].row[choice_no]+17,color);
          }
   out_text_xy(mnus[menu_no].colum,mnus[menu_no].row[choice_no],
            mnus[menu_no].item[choice_no].choice,15);
        }

/*-------------------------------------------------------------*/

  void normal(int menu_no,int choice_no)
    {
     erase_fill_rectangle(mnus[menu_no].colum+4,
                          mnus[menu_no].row[choice_no],
                          mnus[menu_no].colum+146,
                          mnus[menu_no].row[choice_no]+17,9);
    out_text_xy(mnus[menu_no].colum,mnus[menu_no].row[choice_no],
              mnus[menu_no].item[choice_no].choice,14);
     }

/*-------------------------------------------------------------*/

  void unselect_menu(int menu_no,int choice_no,int color)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    normal(menu_no,choice_no);
    display_menu(menu_no,color,color,0);
   }

/*----------------- SCREEN RECORD SOUND ----------------------*/

  void image_record_sound(void)
    {
     char ch;
     int n;

     ud_fill_bar(120,160,500,350,3,3,1);
     up_down_bar(120,160,500,350,0);
  do {
     out_text_xy(130,180," FILE_NAME to RECORD :",0);
     out_text_xy(130+1,180+1," FILE_NAME to RECORD :",14);
     draw_fill_rectangle(302,178,380,195,9);
     up_down_bar(302,178,382,195,0);
     Snd_get_word(B_filename,304,192,9);
     out_text_xy(130,205," FILE_NAME is : ",0);
     out_text_xy(131,206," FILE_NAME is : ",7);

  if(s != ESC) {
    out_text_xy(260,205,Sndfile_name,0);
    out_text_xy(261,206,Sndfile_name,10);
    out_text_xy(130,225," Press a key to record : Esc to quit",0);
    out_text_xy(131,226," Press a key to record : Esc to quit",15);

     n=Read_key();

     if(n==ESC) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        erase_fill_bar(118,158,502,352,1);
        draw_fill_rectangle(34*step_X-4,3*step_Y-4,42*step_X+4,
                            4*step_Y+9,1);
        select_menu(0,1,0);
    }

        draw_fill_rectangle(250,260,340,280,12);
        up_down_bar(250,260,340,280,0);
        out_text_xy(255,265," Recording ",14);
        out_text_xy(210,300,"Press a key to end recording",10);
        Recsound(Sndfile_name);
        draw_fill_rectangle(248,258,342,282,3);
        out_text_xy(210,300,"Press a key to end recording",3);
    }
        out_text_xy(130,275," RECORD another File_name <Y/N> ?",0);
        out_text_xy(131,276," RECORD another File_name <Y/N> ?",15);

ch = ' ';
  while(ch != 'Y' && ch != 'N')
        ch = toupper(getch());

  if( ch == 'Y') {
        out_text_xy(410,275,"YES",14);
        delay(50);
        draw_fill_rectangle(122,162,498,298,3);
                }
} while(ch != 'N');
        out_text_xy(410,275,"NO",14);
        delay(50);
        erase_fill_bar(118,158,502,352,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            draw_fill_rectangle(34*step_X-4,3*step_Y-4,42*step_X+4,
                                4*step_Y+9,1);
            bckgd();
            select_menu(0,1,0);
            }

/*---------------- SCREEN EDIT SOUND -------------------*/

 void image_edit_sound(void)
  {
   char ch;
   int n,y=129,old_y=113,i,j,yy=115;

   ud_fill_bar(30,100,315,400,3,5,1);
   ud_fill_bar(330,100,615,400,7,5,1);
   out_text_xy(50,yy,"File sound ZERO",0);
   out_text_xy(190,yy,":",14);
   out_text_xy(50,yy+=2*step_Y,"File sound ONE",4);
   out_text_xy(190,yy,":",14);
   out_text_xy(50,yy+=2*step_Y,"File sound TWO",5);
   out_text_xy(190,yy,":",14);
   out_text_xy(50,yy+=2*step_Y,"File sound THREE",6);
   out_text_xy(190,yy,":",14);
   out_text_xy(50,yy+=2*step_Y,"File sound FOUR",8);
   out_text_xy(190,yy,":",14);
   out_text_xy(50,yy+=2*step_Y,"File sound FIVE",10);
   out_text_xy(190,yy,":",14);
   out_text_xy(50,yy+=2*step_Y,"File sound SIX",12);
   out_text_xy(190,yy,":",14);
   out_text_xy(50,yy+=2*step_Y,"File sound SEVEN",14);
   out_text_xy(190,yy,":",14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out_text_xy(50,yy+=2*step_Y,"File sound EIGHT",15);
out_text_xy(190,yy,":",14);

yy=115;

out_text_xy(350,yy,"File sound NINE",0);
out_text_xy(490,yy,":",14);
out_text_xy(350,yy+=2*step_Y,"File sound POINT",10);
out_text_xy(490,yy,":",14);
out_text_xy(350,yy+=2*step_Y,"File sound Title_1",12);
out_text_xy(490,yy,":",14);
out_text_xy(350,yy+=2*step_Y,"File sound Title_2",14);
out_text_xy(490,yy,":",14);
out_text_xy(350,yy+=2*step_Y,"File sound Title_3",15);
out_text_xy(490,yy,":",14);
out_text_xy(350,yy+=2*step_Y,"File sound End_1",6);
out_text_xy(490,yy,":",14);
out_text_xy(350,yy+=2*step_Y,"File sound End_2",9);
out_text_xy(490,yy,":",14);
out_text_xy(350,yy+=2*step_Y,"File sound End_3",11);
out_text_xy(490,yy,":",14);

draw_fill_rectangle(200,old_y,280,y,1);
up_down_bar(200,old_y,280,y,0);

  for(i=0;i<=7;i++)   {
      draw_fill_rectangle(200,y+=step_Y,280,y+=step_Y,1);
      up_down_bar(200,y-step_Y,280,y,0);
      }

  y=129;old_y=113;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   draw_fill_rectangle(500,old_y,580,y,1);
   up_down_bar(500,old_y,580,y,0);
       for(j=0;j<=6;j++) {
           draw_fill_rectangle(500,y+=step_Y,580,y+=step_Y,1);
           up_down_bar(500,y-step_Y,580,y,0);
          }
           out_text_xy(240,430,"Press F2 to SAVE DATA",12);
           out_text_xy(220,450,"Press ESC to Exit SOUND EDITOR",13);
           Edit_Sound_word(S_name,0);
         }

/*------------------- Play Sound ---------------------*/

 void play_sound(void)
  {
   char ch;
   int n;

   ud_fill_bar(120,160,470,300,7,3,1);
   up_down_bar(120,160,470,300,0);
do {
   out_text_xy(129,179," FILE_NAME to play :",0);
   out_text_xy(130,180," FILE_NAME to play :",14);
   draw_fill_rectangle(290,178,370,195,12);
   up_down_bar(290,178,370,195,0);
   Snd_get_word(B_filename,293,192,12);
   out_text_xy(129,199," FILE_NAME is : ",0);
   out_text_xy(130,200," FILE_NAME is : ",14);

if (s != ESC) {
    out_text_xy(259,201,Sndfile_name,0);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    out_text_xy(260,202,Sndfile_name,2);
    out_text_xy(129,219," Press a key to play : Esc to quit",0);
    out_text_xy(130,220," Press a key to play : Esc to quit",11);
      n=Read_key();

if(n==ESC) {
    erase_fill_bar(118,158,472,302,1);
    draw_fill_rectangle(34*step_X-4,3*step_Y-4,
                        42*step_X+4,4*step_Y+9,1);
    select_menu(0,1,0);
          }

    ud_fill_bar(250,250,340,270,12,1,0);
    out_text_xy(254,254," Playing ",0);
    out_text_xy(255,255," Playing ",14); }
    sound_playname(Sndfile_name);
    draw_fill_rectangle(250,250,340,270,7);
    out_text_xy(129,269," Play another File_name (Y/N) ?",0);
    out_text_xy(130,270," Play another File_name (Y/N) ?",15);
    ch = ' ';

while(ch != 'Y' && ch != 'N')
      ch = toupper(getch());

  if( ch == 'Y') {
      out_text_xy(410,275,"YES",14);
      delay(50);
      draw_fill_rectangle(122,162,468,298,7);
     }
  } while(ch != 'N');
    out_text_xy(410,275,"NO",14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    delay(50);
    erase_fill_bar(118,158,472,302,1);
    draw_fill_rectangle(34*step_X-4,3*step_Y-4,
                        42*step_X+4,4*step_Y+9,1);
    bckgd();
    select_menu(0,1,0);
 }

/*---------------- Record sound ------------------*/

void Recsound(char text_file[])
 {
  FILE *fp;
  unsigned char result;

  outportb(0x303,0x99);

  if((fp=fopen(text_file,"w"))!=NULL)
      {
do
   { outportb(0x301,0x0);
     result=inportb(0x300);
     putc(result,fp);
   } while(bioskey(1)==0);
   }
  else  out_text_xy(140,290,"couldn't open file",12);
        putc(0x0,fp);
        fclose(fp);
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*---------------- Play sound filename --------------------*/

void sound_playname(char play_name[] )
 {
  unsigned char a;
  FILE *f1;

     outportb(0x303,0x99);
 if ((f1=fopen(play_name,"r"))!=NULL)
   { while((a=getc(f1))!= 0)
     { outportb(0x301,a);
       if(bioskey(1)!=0)
         { bioskey(0);
          break;
     }
   }
    fclose(f1);
   }
 else {
       out_text_xy(239,237,"Couldn't open file",0);
       out_text_xy(240,238,"Couldn't open file",13); }
       outportb(0x301,0x0);
      }

/*---------------- Edit_Sound_Database --------------------*/

void Edit_Sound_word(char *buff,int No)
 {
  int lenght,i,s_Y=16;
  char *buffer;
  buffer=buff;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MAX_LONG = strlen(buffer);
lenght = 0;
Read_file();
Transfer_Snd_data(buffer,No,lenght);
new_X= 202;new_Y=110+s_Y;
Mode_Insert =! INSERT;
Insert_Mode();
Draw_Cursor_Arrow(new_X,new_Y);
do {
    s = Read_key();
      if(s==0) {
        s=getch();

       switch(s) {

      case F2   :  while(lenght > 0) {
                   buffer--;
                   lenght--;
                    }
                   i=0;
                   while(lenght < MAX_LONG) {
                   header_sound[No].sound_filename[i] =  *buffer;
                         i++;
                         buffer++;
                         lenght++;
                   }
                   while(lenght > 0) {
                         buffer--;
                         lenght--;
                         }
                   Erase_Buffer(buffer,lenght);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Snd_Save_Data();
                    new_X=202;
                    Transfer_Snd_data(buffer,No,lenght);
                    Draw_Cursor_Arrow(new_X,new_Y);
                    break;

case UP    :  while(lenght > 0) {
                     buffer--;
                     lenght--;
                    }
                     i=0;
              while(lenght < MAX_LONG) {
                     header_sound[No].sound_filename[i] = *buffer;
                     i++;
                     buffer++;
                     lenght++;
                     }
              while(lenght > 0) {
                     buffer--;
                     lenght--;
                     }
                     Erase_Buffer(buffer,lenght);
                     lenght=0;
              if(No==0) {
                     No = 16;
                     new_Y += 14*s_Y;
                          }
              else {
              if(No==9) new_Y += 16*step_Y;
              else new_Y-=2*s_Y;
                    No--; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(No>8 && No<=16) new_X=502;
            else new_X=202;
            Transfer_Snd_data(buffer,No,lenght);
            Draw_Cursor_Arrow(new_X,new_Y);
            break;

case DOWN :  while(lenght > 0) {
                  buffer--;
                  lenght--;
                 }
                i=0;
            while(lenght < MAX_LONG) {
                  header_sound[No].sound_filename[i] = *buffer;
                  i++;
                  buffer++;
                  lenght++;
                 }
            while(lenght > 0) {
                  buffer--;
                  lenght--;
                 }
                  Erase_Buffer(buffer,lenght);
                  lenght=0;
          if(No==16) {
                  No = 0;
                  new_Y = 110+s_Y;
                   }
          else {
          if(No==8) new_Y = 110+s_Y;
          else new_Y+=2*s_Y;
          No++; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if(No>8 && No<=16) new_X=502;
                else  new_X=202;
                Transfer_Snd_data(buffer,No,lenght);
                Draw_Cursor_Arrow(new_X,new_Y);
                break;

   case HOME : while(lenght)
                     {
                      lenght--;
                      buffer--;
                      new_X-=step_X;
                     }
                      Draw_Cursor_Arrow(new_X,new_Y);
                      break;
   case END  : while(*(buffer+1) != '\0')
                      {
                      buffer++;
                      lenght++;
                      new_X+=step_X;
                      }
               while(*buffer == ' ' && lenght > 0)
                      {
                      buffer--;
                      lenght--;
                      new_X-=step_X;
                      }
              if(*(buffer+1) != '\0' && lenght != 0 )
                 {
                  new_X+=step_X;
                  buffer++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lenght++;
               }
                Draw_Cursor_Arrow(new_X,new_Y);
                break;

case LEFT : if(lenght >= 1) {
                lenght--;
                buffer--;
                new_X-=step_X;
                }
                Draw_Cursor_Arrow(new_X,new_Y);
                break;

case RIGHT : if(lenght < MAX_LONG-1) {
                lenght++;
                buffer++;
                new_X+=step_X;
                }
                Draw_Cursor_Arrow(new_X,new_Y);
                break;

case INSERT: Insert_Mode();
             Draw_Cursor_Arrow(new_X,new_Y);
             break;

case DEL   : Snd_Move_Data_buffer(buffer+1,
                                  buffer,strlen(buffer+1));
             *(buffer + (strlen(buffer)-1))=' ';
             Display_Data(lenght, buffer,1);
             Draw_Cursor_Arrow(new_X,new_Y);
             break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
default   :    Draw_Cursor_Arrow(new_X,new_Y);
               break;
           }
        }

 else {

  switch(s) {

case SPCBAR :   Draw_Cursor_Arrow(new_X,new_Y);
               break;

case BACK_SPC: if(lenght > 0) {
                lenght--;
                Snd_Move_Data_buffer(buffer,
                buffer-1,strlen(buffer));
                *(buffer +(strlen(buffer)-1))=' ';
                buffer--;
                new_X-=step_X;
                Display_Data(lenght, buffer,1);
                }
                Draw_Cursor_Arrow(new_X,new_Y);
                break;

case ESC :     break;

case ENTER : while(lenght > 0)  {
                 buffer--;
                 lenght--;
                 }
                 i=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            while(lenght < MAX_LONG) {
                  header_sound[No].sound_filename[i] =  *buffer;
                  i++;
                  buffer++;
                  lenght++;
                  }
            while(lenght > 0) {
                  buffer--;
                  lenght--;
                  }
                  Erase_Buffer(buffer,lenght);
                  lenght=0;
           if(No==16) s=ESC;
           else {
                if(No==8) new_Y = 110+s_Y;
                else new_Y+=2*s_Y;
                No++;
                }
           if(No>8 && No<=16) new_X = 502;
           else new_X = 202;
           Transfer_Snd_data(buffer,No,lenght);
           Draw_Cursor_Arrow(new_X,new_Y);
           break;

default : if(Mode_Insert != INSERT)
            *buffer = s;
          else {
            Snd_Move_Data_buffer(buffer,buffer+1,
                                 strlen(buffer)-1);
            *buffer = s; }
            Display_Data(lenght, buffer,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(lenght < MAX_LONG-1) {
               lenght++;
               buffer++;
               new_X+=step_X;
               }
               Draw_Cursor_Arrow(new_X,new_Y);
               }
          }
 }while( s != ESC );
         Erase_Buffer(buffer,lenght);
         erase_fill_bar(30,100,615,400,1);
         draw_fill_rectangle(34*step_X-4,3*step_Y-4,
                             42*step_X+4,4*step_Y+9,1);
         out_text_xy(240,430,"Press F2 to SAVE DATA",1);
         out_text_xy(220,450,"Press ESC to Exit SOUND EDITOR",1);
         bckgd();
         select_menu(0,1,0);
 }

/*-------------- SAVE DATA FROM EDIT_SOUND ----------------*/

void Snd_Save_Data(void)
 {
  FILE *fp;
  unsigned int size;
  char ch=' ';
  void *blk;

  size = image_size(190,250,465,305);
  blk  = malloc(size);
  getimage(190,250,465,305,blk);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fill_bar(190,250,450,290,12,1);
shardow(190,250,450,290);
draw_rectangle(190,250,450,290,14);
draw_rectangle(192,252,448,288,14);
out_text_xy(200,265,"Do you want SAVE DATA [Y/N]..?",14);

  while(ch != 'Y' && ch != 'N')
        ch = toupper(getch());
        putimage(190,250,blk,0);

  if (ch == 'Y') {

if((fp=fopen("FSOUND.dat","w"))==NULL)  {
    out_text_xy(200,265,"Do you want SAVE DATA [Y/N]..?",12);
    out_text_xy(220,265,"Program need FSOUND.dat",14);
    getch();
   }
else {
    fwrite(&header_sound,sizeof header_sound,1,fp);
    if(ferror(fp)) {
       out_text_xy(230,265,"Error in writting",14);
       getch();
     } fclose(fp);
 }
       }
 else ;
     free(blk);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*---------------- Check file 'FSOUND.DAT' ------------------*/

void Read_file(void)
  {
  FILE *fp;
  void *bk;
  int i,j,y=113;
  unsigned int size;

 if((fp=fopen("FSOUND.dat","r"))==NULL) {
    size = image_size(190,250,465,305);
    bk   = malloc(size);
    getimage(190,250,465,305,bk);

    shardow(205,265,465,305,0);
    fill_bar(190,250,450,290,12,1);
    draw_rectangle(190,250,450,290,14);
    draw_rectangle(192,252,448,288,14);
    out_text_xy(200,265,"This Program need 'FSOUND.DAT'",14);
    getch();
    putimage(190,250,bk,0);
    free(bk);
    delay(1000);
    BckgndMAIN();
    select_KEY(1,0,8,7,14);
     }
   while(fread(&header_sound,sizeof header_sound,1,fp)==1) {
        fclose(fp);
   for(i=0;i<=8;i++) {
       out_text_xy(202,y,header_sound[i].sound_filename,14);
       y+=2*step_Y; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    y=113;
    for(j=9;j(=16;j++) {
        out_text_xy(502,y,header_sound[j].sound_filename,14);
        y+=2*step_Y;
       }
        }
    }

/*---------------- copy data to buffer -------------------*/

 void Transfer_Snd_data(char *data,int Num,int lenght)
  { int i=0;

    while(lenght ( MAX_LONG)  {
          *data = header_sound[Num].sound_filename[i];
          data++;
          i++;
          lenght++;
         }
    while(lenght ) 0) {
          data--;
          lenght--;
         }
    }

/*------------ DISPLAY SOUND FROM 'FSOUND.DAT" -----------*/

  void Display_Sound_All(void)
   {
    FILE *fp;
    int i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if((fp=fopen("FSOUND.DAT","r"))==NULL) {
    draw_fill_rectangle(190,240,460,300,12);
    up_down_bar(190,240,460,300,0);
    out_text_xy(200,260,"This Program need FSOUND.DAT",14);
    out_text_xy(210,275,"Can not Play SOUND",14);
    delay(1000);
    BckgndMAIN();
    select_KEY(1,0,8,7,14);
    }

while(fread(&header_sound,sizeof header_sound,1,fp)==1) {
      fclose(fp);
      draw_fill_rectangle(190,240,460,290,12);
      up_down_bar(190,240,460,290,0);
      out_text_xy(199,259,"PLAYING Sound Filename :",0);
      out_text_xy(200,260,"PLAYING Sound Filename :",14);

 for(i=0;i<=16;i++) {
      out_text_xy(390,260,header_sound[i].sound_filename,14);
      all_sound_play(header_sound[i].sound_filename);
      out_text_xy(390,260,header_sound[i].sound_filename,12);
     }
      }
      erase_fill_bar(188,238,462,302,1);
      bckgd();
      select_menu(0,1,0);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*--------------- Play sound all ------------------*/

 void all_sound_play(char play_name[] )
  {
   int a;
   FILE *f1;

      outportb(771,152);
   if ((f1=fopen(play_name,"r"))!=NULL)
      { while((a=getc(f1))!= 0)
            { outportb(769,a);
              if(kbhit())
                 a=0;
       }
       fclose(f1);
       }
   else {
        out_text_xy(239,245,"Couldn't open this file",0);
        out_text_xy(240,246,"Couldn't open this file",13);
        delay(1000);
        out_text_xy(239,245,"Couldn't open this file",12);
        out_text_xy(240,246,"Couldn't open this file",12);
        }
        outportb(769,0);
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รายละเอียดในส่วนอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

MITEL

ISO²-CMOS MT8870C/MT8870C-1 Integrated DTMF Receiver

9161-302-094-NA ISSUE 1 November 1987

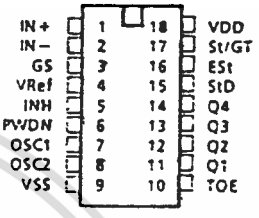
Features

- Complete DTMF Receiver
- Low Power Consumption
- Internal Gain Setting Amplifier
- Adjustable Guard Time
- Central Office Quality
- Power-down Mode
- Inhibit Mode

Applications

- Receiver System for British Telecom (BT) or CEPT Spec (MT8870C-1)
- Paging Systems
- Repeater Systems/Mobile Radio
- Credit Card Systems
- Remote Control
- Personal Computers
- Telephone Answering Machine

Pin Connections

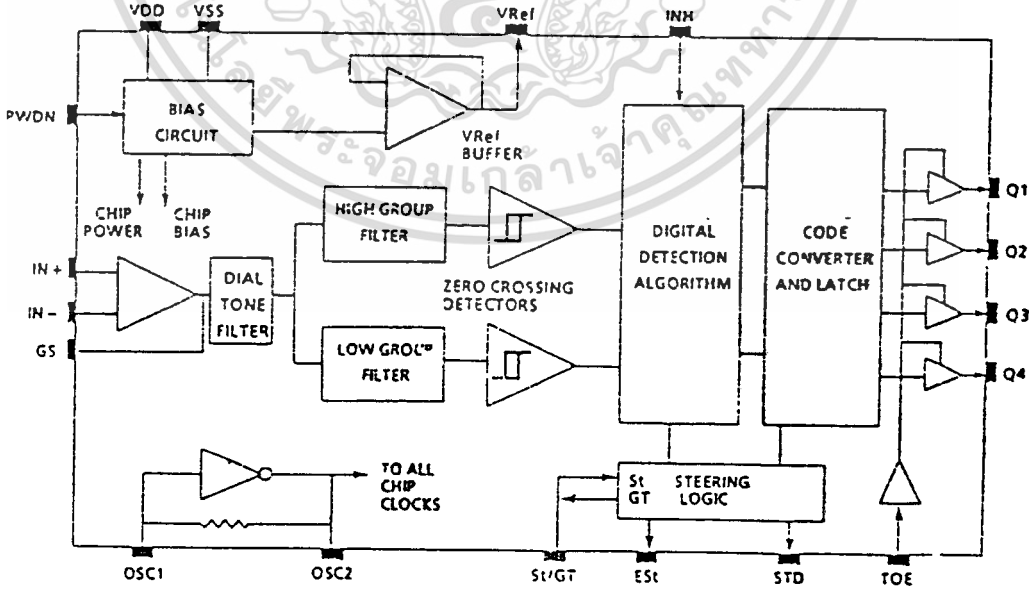


Ordering Information

MT8870CE/MT8870CE-1	Plastic DIP
MT8870CC/MT8870CC-1	Cerdip
MT8870CS/MT8870CS-1	SOIC
−40°C to +85°C	

Description

The MT8870C/MT8870C-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions, fabricated in Mitel's double poly ISO²-CMOS technology. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone-pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

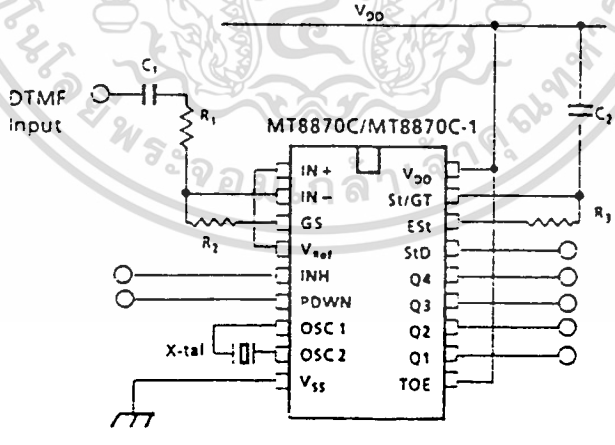


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics - $V_{DD} = 5.0V \pm 5\%$, $V_{SS} = 0V$, $-40°C \le To \le +85°C$, using Test Circuit shown in Figure 2.

		Characteristics	Sym	Min	Typ‡	Max	Units	Conditions
1	TIMING	Tone present detect time	t_{DP}	5	11	14	ms	Note 12
2		Tone absent detect time	t_{DA}	0.5	4	8.5	ms	Note 12
3		Tone duration accept	t_{REC}			40	ms	User adjustable
4		Tone duration reject	$\overline{t_{REC}}$	20			ms	User adjustable
5		Interdigit pause accept	t_{ID}			40	ms	User adjustable
6		Interdigit pause reject	t_{DO}	20			ms	User adjustable
7	OUTPUTS	Propagation delay (St to Q)	t_{PQ}		8	11	μs	TOE = V_{DD}
8		Propagation delay (St to StD)	t_{PStD}		12	16	μs	TOE = V_{DD}
9		Output data set up (Q to StD)	t_{QStD}		3.4		μs	TOE = V_{DD}
10		Propagation delay (TOE to Q ENABLE)	t_{PTE}		50		ns	load of 10 kΩ, 50 pF
11		Propagation delay (TOE to Q DISABLE)	t_{PTD}		300		ns	load of 10 kΩ, 50 pF
12	CLOCK	Crystal /clock frequency	f_C	3.5759	3.5795	3.5831	MHz	
13		Clock input rise time	t_{LHCL}			110	ns	Ext. clock
14		Clock input fall time	t_{HLCL}			110	ns	Ext. clock
15		Clock input duty cycle	DC_{CL}	40	50	60	%	Ext. clock
16		Capacitive load (OSC2)	C_{LO}			30	pF	

‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.



NOTES:
$R_1, R_2 = 100\ k\Omega \pm 1\%$
$R_3 = 300\ k\Omega \pm 1\%$
$C_1, C_2 = 100\ nF \pm 5\%$
X-tal = 3.579545 MHz ±0.1 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

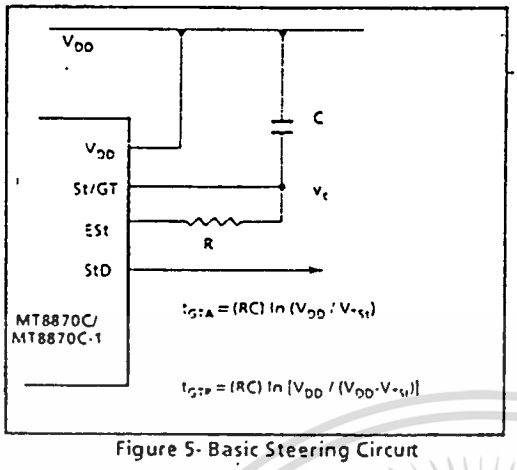


Figure 5- Basic Steering Circuit

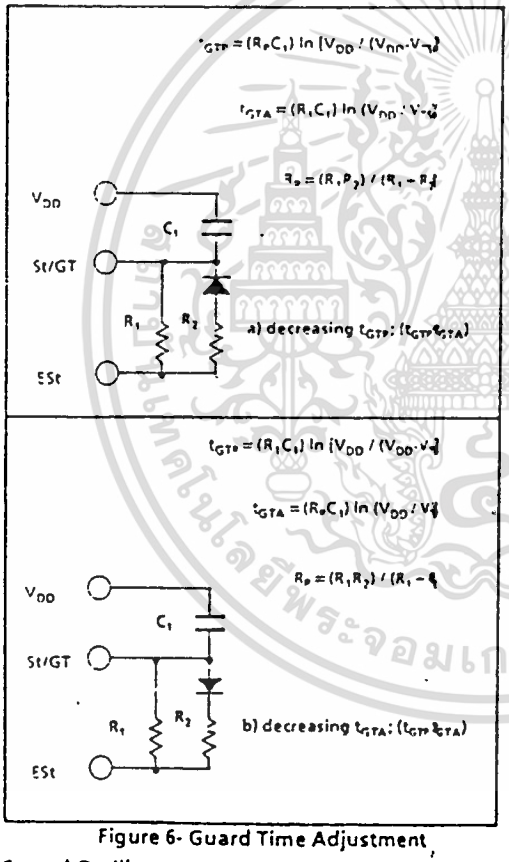


Figure 6- Guard Time Adjustment

Crystal Oscillator

The internal clock circuit is completed with the addition of an external 3.579545 MHz crystal and is normally connected as shown in Figure 7(Single Ended Input Configuration). However, it is possible

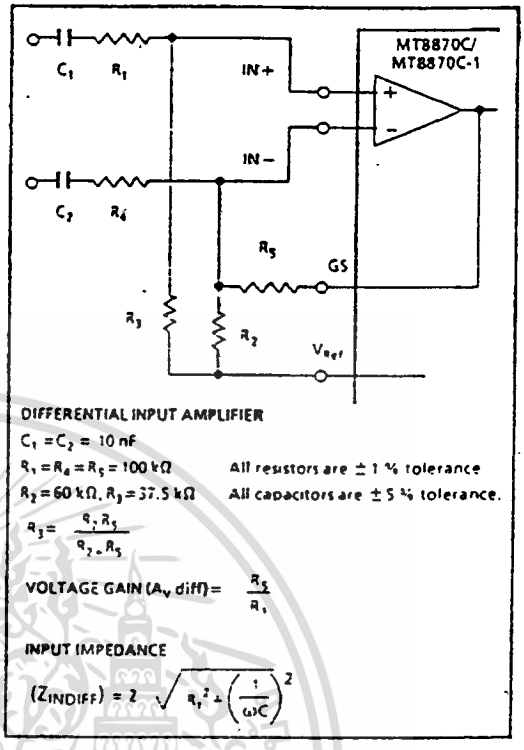
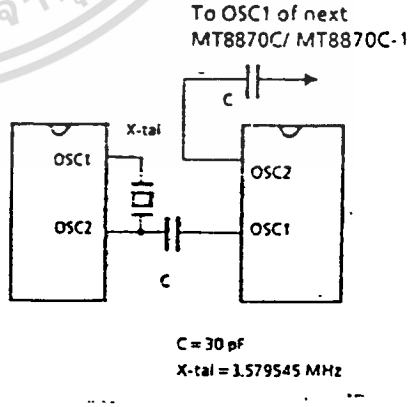


Figure 7- Differential Input Configuration

employing only a single oscillator crystal. The oscillator output of the first device in the chain is coupled through a 30 pF capacitor to the oscillator input (OSC1) of the next device. Subsequent devices are connected in a similar fashion. Refer to Figure 8 for details. The problems associated with unbalanced loading are not a concern with the arrangement shown, ie; precision balancing capacitors are not required.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION

RECEIVER SYSTEM FOR BRITISH TELECOM SPEC POR 1151

The circuit shown in Fig. 10 illustrates the use of MT8870C-1 device in a typical receiver system. BT Spec defines the input signals less than -34 dBm as the non-operate level. This condition can be attained by choosing a suitable values of R_1 and R_2 to provide 3 dB attenuation, such that -34 dBm input signal will correspond to -37 dBm at the gain setting pin GS of MT8870C-1. As shown in the diagram, the component values of R_3 and C_2 are the guard time requirements when the total component tolerance is 6%. For better performance, it is recommended to use the non-symmetric guard time circuit in Fig. 9.

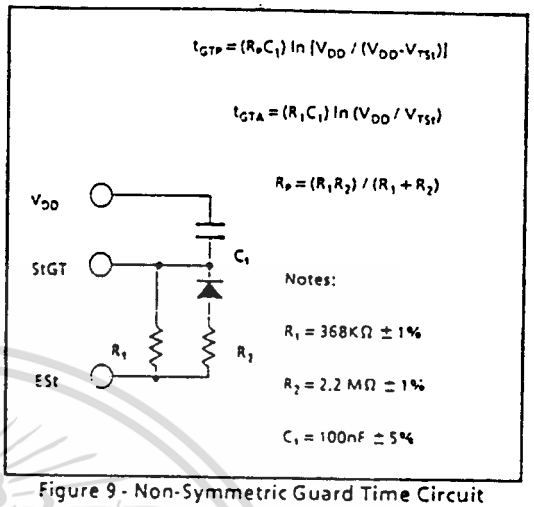


Figure 9 - Non-Symmetric Guard Time Circuit

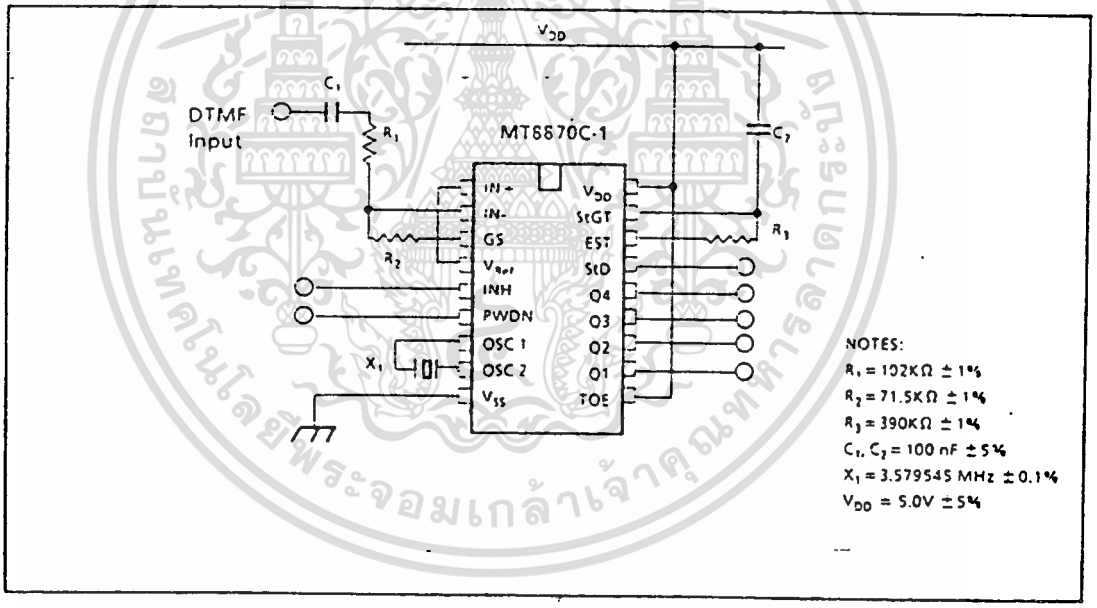
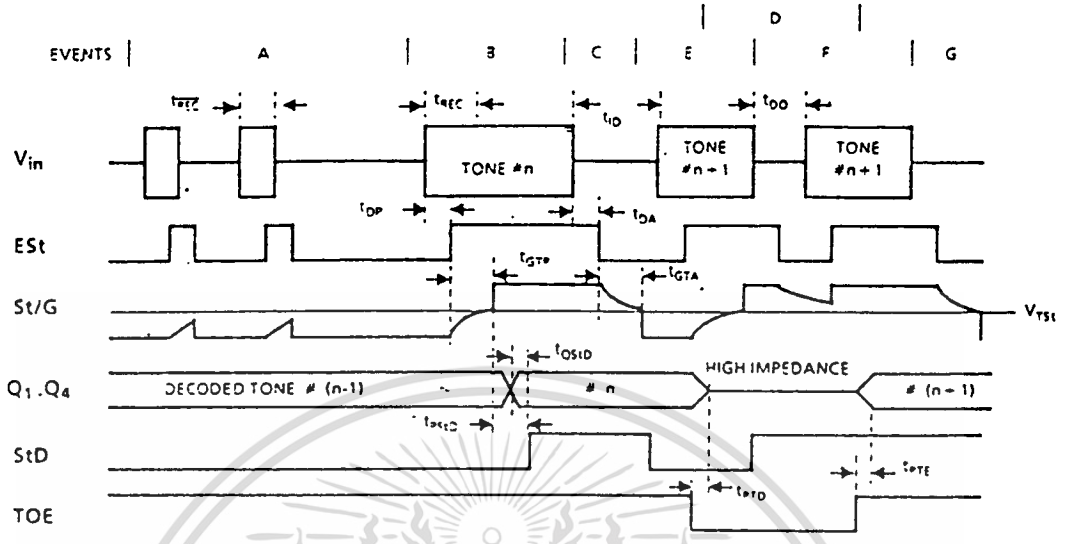


Figure 10 - Single-Ended Input Configuration for BT or CEPT Spec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



EXPLANATION OF EVENTS

A) TONE BURSTS DETECTED, TONE DURATION INVALID, OUTPUTS NOT UPDATED
B) TONE #n DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS
C) END OF TONE #n DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.
D) OUTPUTS SWITCHED TO HIGH IMPEDANCE STATE.
E) TONE #n+1 DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS (CURRENTLY HIGH IMPEDANCE)
F) ACCEPTABLE DROPOUT OF TONE #n + 1, TONE ABSENT DURATION INVALID, OUTPUTS REMAIN LATCHED
G) END OF TONE #n + 1 DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.

EXPLANATION OF SYMBOLS

V_{in}	DTMF COMPOSITE INPUT SIGNAL.
ESt	EARLY STEERING OUTPUT. INDICATES DETECTION OF VALID TONE FREQUENCIES.
St/GT	STEERING INPUT/GUARD TIME OUTPUT. DRIVES EXTERNAL RC TIMING CIRCUIT.
Q_1-Q_4	4-BIT DECODED TONE OUTPUT.
StD	DELAYED STEERING OUTPUT. INDICATES THAT VALID FREQUENCIES HAVE BEEN PRESENT/ABSENT FOR THE REQUIRED GUARD TIME THUS CONSTITUTING A VALID SIGNAL.
TOE	TONE OUTPUT ENABLE (INPUT). A LOW LEVEL SHIFTS Q_1-Q_4 TO ITS HIGH IMPEDANCE STATE.
t_{REZ}	MAXIMUM DTMF SIGNAL DURATION NOT DETECTED AS VALID.
t_{REC}	MINIMUM DTMF SIGNAL DURATION REQUIRED FOR VALID RECOGNITION.
t_{ID}	MINIMUM TIME BETWEEN VALID DTMF SIGNALS.
t_{DO}	MAXIMUM ALLOWABLE DROP OUT DURING VALID DTMF SIGNAL
t_{DP}	TIME TO DETECT THE PRESENCE OF VALID DTMF SIGNALS.
t_{DA}	TIME TO DETECT THE ABSENCE OF VALID DTMF SIGNALS.
t_{GTP}	GUARD TIME, TONE PRESENT.
t_{GTA}	GUARD TIME, TONE ABSENT.

Figure 3- Timing Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digit	TOE	INH	EST	Q_4	Q_3	Q_2	Q_1
ANY	L	X	-	Z	Z	Z	Z
1	H	L	H	0	0	0	1
2	H	L	H	0	0	1	0
3	H	L	H	0	0	1	1
4	H	L	H	0	1	0	0
5	H	L	H	0	1	0	1
6	H	L	H	0	1	1	0
7	H	L	H	0	1	1	1
8	H	L	H	1	0	0	0
9	H	L	H	1	0	0	1
0	H	L	H	1	0	1	0
*	H	L	H	1	0	1	1
#	H	L	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
1	H	H	H	0	0	0	1
2	H	H	H	0	0	1	0
3	H	H	H	0	0	1	1
4	H	H	H	0	1	0	0
5	H	H	H	0	1	0	1
6	H	H	H	0	1	1	0
7	H	H	H	0	1	1	1
8	H	H	H	1	0	0	0
9	H	H	H	1	0	0	1
0	H	H	H	1	0	1	0
*	H	H	H	1	0	1	1
#	H	H	H	1	1	0	0
A	H	H	L	undetected, the output			
B	H	H	L	code will remain the same			
C	H	H	L	as the previous detected			
D	H	H	L	code			

L = LOGIC LOW, H = LOGIC HIGH, Z = HIGH IMPEDANCE

Table 1 - Functional Decode Table

the interdigit pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (dropout) too short to be considered a valid pause. This facility, together with the capability of selecting the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

Guard Time Adjustment

In many situations not requiring selection of tone duration and interdigital pause, the simple steering circuit shown in Figure 5 is applicable. Component values are chosen according to the formula:

$$t_{REC} = t_{DP} + t_{GTP}$$
$$t_{ID} = t_{DA} + t_{GTA}$$

The value of t_{DP} is a device parameter (see Figure 3) and t_{REC} is the minimum signal duration to be recognized by the receiver. A value for C of 0.1 μF is recommended for most applications, leaving R to be selected by the designer.

Different steering arrangements may be used to select independently the guard times for tone present (t_{GTP}) and tone absent (t_{GTA}). This may be necessary to meet system specifications which place both accept and reject limits on both tone duration and interdigital pause. Guard time adjustment also allows the designer to tailor system parameters such as talk off and noise immunity. Increasing t_{REC} improves talk-off performance since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. Alternatively, a relatively short t_{REC} with a long t_{DO} would be appropriate for extremely noisy environments where fast acquisition time and immunity to tone drop-outs are required. Design information for guard time adjustment is shown in Figure 6.

Power-down and Inhibit Mode

A logic high applied to pin 6 (PWDN) will power down the device to minimize the power consumption in a standby mode. It stops the oscillator and the functions of the filters.

Inhibit mode is enabled by a logic high input to the pin 5 (INH). It inhibits the detection of tones representing characters A,B,C and D. The output code will remain the same as the previous detected code (see Table 1).

Differential Input Configuration

The input arrangement of the MT8870C/MT8870C-1 provides a differential-input operational amplifier as well as a bias source(V_{Ref}) which is used to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the op-amp output (GS) for adjustment of gain. In a single-ended configuration, the input pins are connected as shown in Figure 2 with the op-amp connected for unity gain and V_{Ref} biasing the input at $\frac{1}{2}V_{DD}$. Figure 7 shows the differential configuration, which permits the adjustment of gain with the feedback resistor R_5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

Pin #	Name	Description
1	IN+	Non-Inverting Op-Amp (Input).
2	IN−	Inverting Op-Amp (Input).
3	GS	Gain Select. Gives access to output of front end differential amplifier for connection of feedback resistor.
4	V_{Ref}	Reference Voltage (Output). Nominally $V_{DD}/2$ is used to bias inputs at mid-rail (see Fig.2).
5	INH	Inhibit (Input). Logic high inhibits the detection of tones representing characters A, B, C and D. This pin input is internally pulled down.
6	PWDN	Power Down (Input). Active high. Powers down the device and inhibits the oscillator. This pin input is internally pulled down.
7	OSC1	Clock (Input).
8	OSC2	Clock (Output). A 3.579545 MHz crystal connected between pins OSC1 and OSC2 completes the internal oscillator circuit.
9	V_{SS}	Negative Power Supply (Input).
10	TOE	Three State Output Enable (Input). Logic high enables the outputs Q1-Q4. This pin is pulled up internally.
11-14	Q1-Q4	Three State Data (Output). When enabled by TOE, provide the code corresponding to the last valid tone-pair received (see Table 1). When TOE is logic low, the data outputs are high impedance.
15	StD	Delayed Steering (Output). Presents a logic high when a received tone-pair has been registered and the output latch updated; returns to logic low when the voltage on St/GT falls below V_{TSt}.
16	ESt	Early Steering (Output). Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause ESt to return to a logic low.
17	St/GT	Steering Input/Guard time (Output) Bidirectional. A voltage greater than V_{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{TSt} frees the device to accept a new tone pair. The GT output acts to reset the external steering time-constant; its state is a function of ESt and the voltage on St.
18	V_{DD}	Positive power supply (Input).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

	Parameter	Symbol	Min	Max	Units
1	DC Power Supply Voltage	V_{DD}		6	V
2	Voltage on any pin	V_I	$V_{SS}-0.3$	$V_{DD}+0.3$	V
3	Current at any pin (other than supply)	I_I		10	mA
4	Storage temperature	T_{STG}	−65	+150	°C
5	Package power dissipation	P_D		1000	mW

* Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.
Derate above 75 °C at 16 mW / °C. All leads soldered to board

Recommended Operating Conditions - Voltages are with respect to ground (V_{SS}) unless otherwise stated.

	Parameter	Sym	Min	Typ†	Max	Units	Test Conditions
1	DC Power Supply Voltage	V_{DD}	4.75	5.0	5.25	V	
2	Operating Temperature	T_O	−40		+85	°C	
3	Crystal/Clock Frequency	fc		3.579545		MHz	
4	Crystal/Clock Freq Tolerance	Δfc		±0.1		%	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing

DC Electrical Characteristics - V_{DD} = 5.0V ± 5%, V_{SS} = 0V, −40°C ≤ T_O ≤ +85°C, unless otherwise stated

		Characteristics	Sym	Min	Typ†	Max	Units	Test Conditions
1	SUPPLY	Standby supply current	I_{DDO}			100	μA	PWDN = V_{DD}
2		Operating supply current	I_{DD}		3.0	9.0	mA	
3		Power consumption	P_O		15	50	mW	fc = 3.579545 MHz
4	INPUTS	High level input	V_{IH}	3.5			V	V_{DD} = 5.0V
5		Low level input voltage	V_{IL}			1.5	V	V_{DD} = 5.0V
6		Input leakage current	I_{IH}/I_{IL}		0.1		μA	V_{IN} = V_{SS} or V_{DD}
7		Pull up (source) current	I_{SO}		7.5	20	μA	TOE (pin 10) = 0, V_{DD} = 5.0V
8		Pull down (sink) current	I_{SI}		15	45	μA	INH = 5.0V, PWDN = 5.0V, V_{DD} = 5.0V
9		Input impedance (IN+, IN−)	R_{IN}		10		MΩ	@ 1 kHz
10		Steering threshold voltage	V_{TSt}	2.2		2.5	V	V_{DD} = 5.0V
11	OUTPUTS	Low level output voltage	V_{OL}			V_{SS}+0.03	V	No load
12		High level output voltage	V_{OH}	V_{DD}-0.03			V	No load
13		Output low (sink) current	I_{OL}	1.0	2.5		mA	V_{OUT} = 0.4 V
14		Output high (source) current	I_{OH}	0.4	0.8		mA	V_{OUT} = 4.6 V
15		V_{Ref} output voltage	V_{Ref}	2.4		2.7	V	No load, V_{DD} = 5.0V
16		V_{Ref} output resistance	R_{OR}		10		kΩ	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operating Characteristics - V_{DD} = 5.0V ± 5%, V_{SS} = 0V, -40°C ≤ T_O ≤ +85°C, unless otherwise stated.

Gain Setting Amplifier

	Characteristics	Sym	Min	Typ¹	Max	Units	Test Conditions
1	Input leakage current	I_{IN}			100	nA	$V_{SS} \le V_{IN} \le V_{DD}$
2	Input resistance	R_{IN}	10			MΩ	
3	Input offset voltage	V_{OS}			25	mV	
4	Power supply rejection	PSRR	50			dB	1 kHz
5	Common mode rejection	CMRR	40			dB	−3.0 V ≤ V_{IN} ≤ 3.0 V
6	DC open loop voltage gain	A_{VOL}	32			dB	
7	Unity gain bandwidth	f_C	0.30			MHz	
8	Output voltage swing	V_O	4.0			V_{pp}	Load ≥ 100 kΩ to V_{SS}
9	Maximum capacitive load (GS)	C_L			100	pF	
10	Maximum resistive load (GS)	R_L			50	kΩ	
11	Common mode range	V_{CM}	2.5			V_{pp}	No Load

MT8870C AC Electrical Characteristics - V_{DD} = 5.0V ± 5%, V_{SS} = 0V, -40°C ≤ T_O ≤ +85°C, using Test Circuit shown in Figure 2.

	Characteristics	Sym	Min	Typ¹	Max	Units	Notes*
1	Valid input signal levels (each tone of composite signal)		-29			dBm	1,2,3,5,6,9
			-27.5			mV_{RMS}	1,2,3,5,6,9
					-1	dBm	1,2,3,5,6,9
					369	mV_{RMS}	1,2,3,5,6,9
2	Negative twist accept				6	dB	2, 3, 6, 9
3	Positive twist accept				5	dB	2, 3, 6, 9
4	Frequency deviation accept		± 1.5% ± 2 Hz				2, 3, 5, 9
5	Frequency deviation reject		± 3.5%				2, 3, 5, 9
6	Third tone tolerance			-16		dB	2, 3, 4, 5, 9, 10
7	Noise tolerance			-12		dB	2, 3, 4, 5, 7, 9, 10
8	Dial tone tolerance			-22		dB	2, 3, 4, 5, 8, 9, 11

¹Typical figures are at 25 °C and are for design aid only: not guaranteed and not subject to production testing.

*NOTES

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load
2. Digit sequence consists of all DTMF tones.
3. Tone duration = 40 ms, tone pause = 40 ms.
4. Signal condition consists of nominal DTMF frequencies.
5. Both tones in composite signal have an equal amplitude
6. Tone pair is deviated by ± 1.5 % ± 2 Hz.
7. Bandwidth limited (3 kHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %.
9. For an error rate of better than 1 in 10,000.
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. For guard time calculation purposes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MT8870C-1 AC Electrical Characteristics* - V_{DD} = 5.0V ± 5%, V_{SS} = 0V, -40°C ≤ T_O ≤ +85°C, using Test Circuit shown in Figure 2.

	Characteristics	Sym	Min	Typ†	Max	Units	Notes*
1	Valid input signal levels (each tone of composite signal)		-31		+1	dBm	V_{DD} = 5.0V 1,2,3,5,6,9
			21.8		869	mV_{RMS}	
2	Input Signal Level Reject		-37			dBm	V_{DD} = 5.0V 1,2,3,5,6,9
			10.9			mV_{RMS}	
3	Negative twist accept				6	dB	2, 3, 6, 9
4	Positive twist accept				6	dB	2, 3, 6, 9
5	Frequency deviation accept		±1.5% ± 2 Hz				2, 3, 5, 9
6	Frequency deviation reject		±3.5%				2, 3, 5, 9
7	Third tone tolerance		18.5			dB	2, 3, 4, 5, 9, 13
8	Noise tolerance			-12		dB	2, 3, 4, 5, 7, 9, 10
9	Dial tone tolerance			+22		dB	2, 3, 4, 5, 8, 9, 11

†Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

*NOTES

1 dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2 Digit sequence consists of all DTMF tones.
3 Tone duration = 40 ms, tone pause = 40 ms.
4 Signal condition consists of nominal DTMF frequencies.
5 Both tones in composite signal have an equal amplitude.
6 Tone pair is deviated by ± 1.5% ± 2 Hz.
7 Bandwidth limited (3 kHz) Gaussian noise.
8 The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2%.
9. For an error rate of better than 1 in 10,000.
10 Referenced to lowest level frequency component in DTMF signal.
11 Referenced to the minimum valid accept level.
12 For guard time calculation purposes.
13 Referenced to Fig. 10 input DTMF tone level at -25dBm - 28dBm at GS Pin) interference frequency range between 480-3400Hz.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

The MT8870C/MT8870C-1 monolithic DTMF receiver offers small size, low power consumption and high performance. Its architecture consists of a bandsplit filter section, which separates the high and low group tones, followed by a digital counting section which verifies the frequency and duration of the received tones before passing the corresponding code to the output bus.

Filter Section

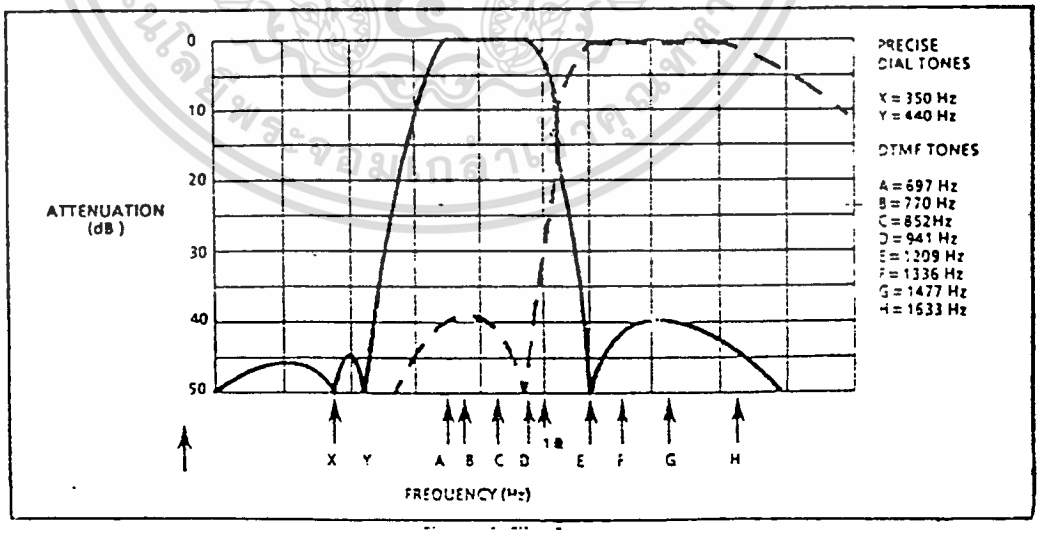
Separation of the low-group and high group tones is achieved by applying the DTMF signal to the inputs of two sixth-order switched capacitor bandpass filters, the bandwidths of which correspond to the low and high group frequencies. The filter section also incorporates notches at 350 and 440 Hz for exceptional dial tone rejection (see Figure 4). Each filter output is followed by a single order switched capacitor filter section which smooths the signals prior to limiting. Limiting is performed by high-gain comparators which are provided with hysteresis to prevent detection of unwanted low-level signals. The outputs of the comparators provide full rail logic swings at the frequencies of the incoming DTMF signals.

Decoder Section

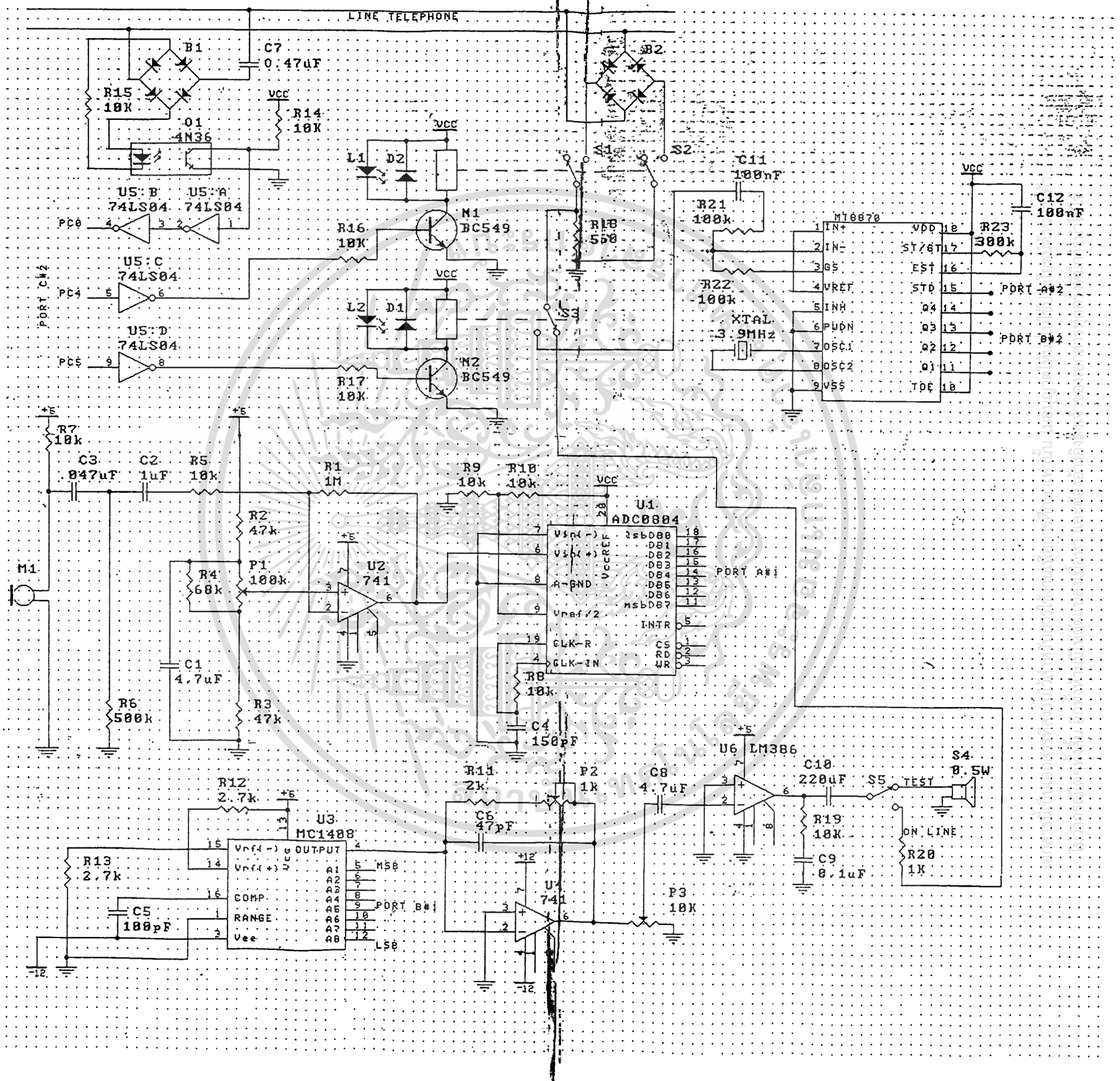
Following the filter section is a decoder employing digital counting techniques to determine the frequencies of the incoming tones and to verify that they correspond to standard DTMF frequencies. A complex averaging algorithm protects against tone simulation by extraneous signals such as voice while providing tolerance to small frequency deviations and variations. This averaging algorithm has been developed to ensure an optimum combination of immunity to talk-off and tolerance to the presence of interfering frequencies (third tones) and noise. When the detector recognizes the presence of two valid tones (this is referred to as the "signal condition" in some industry specifications) the "Early Steering" (ESt) output will go to an active state. Any subsequent loss of signal condition will cause ESt to assume an inactive state (see "Steering Circuit").

Steering Circuit

Before registration of a decoded tone pair, the receiver checks for a valid signal duration (referred to as character recognition condition). This check is performed by an external RC time constant driven by R_T. A logic high on ESt causes v_c (see Figure 5) to rise as the capacitor discharges. Provided signal condition is maintained (ESt remains high) for the validation period (t_{GTP}), v_c reaches the threshold (V_{TSt}) of the steering logic to register the tone pair, latching its corresponding 4-bit code (see Table 1) to the output latch. At this point the GT output is activated and drives v_c to V_{DD}. GT continues to drive high as long as ESt remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (StD) goes high, signalling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the three state control input (TOE) to a logic high. The steering circuit works in reverse to validate



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


LINE TELEPHONE
B1
C7
0.47uF
R15
10K
VCC
R14
10K
O1
4N36
U5:B
74LS04
U5:A
74LS04
PC0
U5:C
74LS04
PC4
PORT C#2
U5:D
74LS04
PC5
L1
D2
L2
D1
M1
BC549
N2
BC549
R16
10K
R17
10K
B2
S1
S2
S3
R18
5K8
C11
100nF
R21
100k
R22
100k
XTAL
3.9MHz
MT8870
IN+
IN-
GS
VREF
INH
PWDN
OSC1
OSC2
VSS
VDD
ST/GT
ESТ
STD
Q4
Q3
Q2
Q1
TOE
R23
300k
C12
100nF
PORT A#2
PORT B#2
R7
10k
C3
047uF
C2
1uF
R5
10k
R1
1M
R9
10k
R10
10k
U1
ADC0804
Vin(-)
Vin(+)
A-GND
Vref/2
CLK-R
CLK-IN
VCC/REF
lsbDB0
DB1
DB2
DB3
DB4
DB5
DB6
msbDB7
INTR
CS
RD
WR
PORT A#1
R2
47k
R4
68k
P1
100k
U2
741
M1
C1
4,7uF
R6
500k
R3
47k
R8
10k
C4
150pF
R12
2,7k
U3
MC1408
Vref(-)
Vref(+)
Vcc
OUTPUT
COMP
RANGE
Vee
A1
A2
A3
A4
A5
A6
A7
A8
MSB
LSB
PORT B#1
R13
2,7k
C5
100pF
R11
2k
P2
1k
C6
47pF
U4
741
P3
10K
C8
4,7uF
U6
LM386
C10
220uF
S5
TEST
S4
0,5W
R19
10K
C9
0,1uF
ON LINE
R20
1K

บรรณานุกรม

จิตติ หนูแก้ว."เทคนิคการเชื่อมต่อ IBM PC กับอุปกรณ์ภายนอกเพื่อประยุกต์ใช้งานต่างๆ"กรุงเทพมหานคร :หจก.เอช-เอน การพิมพ์,2536

ชูชัย ธนะสารตั้งเจริญ และคณะ."การใช้งาน Z-80"กรุงเทพมหานคร :สำนักพิมพ์ฟิสิกส์เซ็นเตอร์,284

บัณฑิต จามรภูมิ. "ฮาร์ดแวร์ไมโครคอมพิวเตอร์ 8088,80286,80386"กรุงเทพมหานคร :หจก.เอช-เอน การพิมพ์,2535

มัณฑนา ปราการสมุทร. "การเขียนชุดคำสั่งภาษา C "กรุงเทพมหานคร :สำนักพิมพ์ไฮเทคพริ้นติ้ง,2534,311

มนตรี พจนารถลาวัณย์."การเขียนโปรแกรมคอมพิวเตอร์ด้วย เทอร์โบซี" กรุงเทพมหานคร :สำนักพิมพ์ซีเอ็ดยูเคชั่น,346

วิทยา เรืองพรวิสุทธิ์."คู่มือโปรแกรมภาษา C สำหรับผู้เริ่มต้น" กรุงเทพมหานคร :เอช-เอน การพิมพ์,333

วิทยา วัชระวิทยากุล. "ภาษาและการโปรแกรม C " กรุงเทพมหานคร :สำนักพิมพ์ซีเอ็ดยูเคชั่น

วิสันท์ อาชาเดโชพล."คอมพิวเตอร์ขนาดเล็กและการประยุกต์ใช้งาน" :สำนักพิมพ์-ฟิสิกส์เซ็นเตอร์,หน้า(1-10)

คู่มือไอซีไมโครโปรเซสเซอร์ และไอซีที่เกี่ยวข้อง บริษัทซีเอ็ดยูเคชั่นจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า