เครื่องบันทึกเสียงโทรศัพท์อัตโนมัติ

AUTOMATIC VOICE MAIL BOX

โดย

1. นายกมล โพธิ์งาม รหัส 35.103001
2. นายกิตติ พฤกษ์ธาดาชัย รหัส 35.103003
3. นายสุรวิทย์ วิชช์วัฒนางกูร รหัส 35.103039

วัน เดือน ปี 19 ม.ค. 2539
เลขทะเบียน 034954
เลขเรียกหนังสือ T 39954 ก.4

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องบันทึกเสียงโทรศัพท์อัตโนมัติ

AUTOMATIC VOICE MAIL BOX

โดย

1. นายกมล โพธิ์งาม รหัส 35.103001
2. นายกิตติ พฤกษ์ธาดาชัย รหัส 35.103003
3. นายสุรวิทย์ วิชช์วัฒนางกูร รหัส 35.103039

อาจารย์ที่ปรึกษา

ผศ.สมยศ จุณณะปิยะ

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องบันทึกเสียงโทรศัพท์อัตโนมัติ

ผู้จัดทำ

1. นายกมล โพธิ์งาม 35103001
2. นายกิตติ พฤกษ์ธาดาชัย 35103003
3. นายสุรวิทย์ วิชชัวัฒนางกูร 35103039

----------.อาจารย์ที่ปรึกษา

(ผศ. สมยศ จุณณะปิยะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องบันทึกเสียงโทรศัพท์อัตโนมัติ

AUTOMATIC VOICE MAIL BOX

โดย

1. นายกมล โพธิ์งาม รหัส 35.103001
2. นายกิตติ พฤกษ์ธาดาชัย รหัส 35.103003
3. นายสุรวิทย์ วิชช์วัฒนางกูร รหัส 35.103039

อาจารย์ที่ปรึกษา

ผศ.สมยศ จุณณะปิยะ

บทคัดย่อ.

ในปัจจุบันการสื่อสารมีความสำคัญและจำเป็นในชีวิตประจำวันการสื่อสารทางโทรศัพท์เป็นวิธีการหนึ่งในหลายๆวิธีที่มีความสำคัญและอำนวยความสะดวกแก่ผู้ใช้ในด้านของการทำธุรกิจและการแก้ไขปัญหาจราจรแต่ปัญหาของการใช้โทรศัพท์สามารถจะเกิดขึ้นได้ถ้ามีจำนวนผู้ใช้โทรศัพท์จำนวนมากในเวลาเดียวกัน ทำให้ผู้ใช้โทรศัพท์ไม่สามารถติดต่อสื่อสารกับผู้อื่นได้ อย่างไรก็ตามการสื่อสารด้วยโทรศัพท์กับผู้อื่นจะเป็นไปได้ถึงแม้ว่าผู้รับจะไม่อยู่ถ้าหากว่าเราสามารถฝากข้อความให้แก่ผู้รับได้ โครงงานนี้ได้แบ่งฮาร์ดแวร์ออกเป็น 2 ส่วนคือส่วนแรกทำหน้าที่ติดต่อกับสัญญาณโทรศัพท์ส่วนที่สองทำหน้าที่ติดต่อกับการ์ดเสียง (Sound Card) ซึ่งทำหน้าที่เปลี่ยนสัญญาณเสียงเป็นสัญญาณดิจิตอลแล้วเก็บลงบนฮาร์ดดิสก์โดยใช้โปรแกรมควบคุมการบันทึกเสียงและการโอนสายอัตโนมัติ

ABSTRACT.

Nowadays, Communications are important and necessary is our dialy lives. Telephone Communication a mean of various kind of communications, is very important and it also provide a lot of comforts to the users. No matter in the attach of doing businesses or traffic jam problem solving but problem of using telephone can occur if there is a great amount of using at the same time. It makes users cannot communicate with another. In this situation they feel discomfort. Although communication by telephone with one another is possible even the receivers are out. Because we can leave the message. It consists of two part, the first part interfaces with telephone signal and the second part interfaces with sound card. The sound card transform analog signal to digital data and keep into harddisk. By using program are control record voice , automatic transfer line telephone.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทที่	
1. บทนำ	1
2. ส่วนติดต่อกับคอมพิวเตอร์	2
3. ส่วนถอดรหัสความถี่โทรศัพท์	12
4. ปัญหาที่เกิดขึ้นในการดำเนินงาน	18
5. ขั้นตอนการแก้ปัญหาที่เกิดขึ้นในการดำเนินงาน	19
6. การใช้งานระบบตอบรับโทรศัพท์อัตโนมัติ	20
สรุปผลการดำเนินงาน	26
ข้อเสนอแนะ	27
ภาคผนวก ก. วงจรสมบูรณ์ของระบบตอบรับโทรศัพท์อัตโนมัติ	28
ภาคผนวก ข. โฟลว์ชาร์ตแสดงการทำงานของระบบตอบรับโทรศัพท์อัตโนมัติ	31
ภาคผนวก ค. โปรแกรมควบคุมการทำงานของระบบตอบรับโทรศัพท์อัตโนมัติ	35
ภาคผนวก ง. รายการอุปกรณ์	66
กิตติกรรมประกาศ	69
บรรณานุกรม	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันความเจริญก้าวหน้าทางเทคโนโลยีได้เจริญเติบโตเร็วมาก เป็นผลให้ประเทศได้รับการพัฒนาที่รวดเร็วยิ่งขึ้นจะเห็นได้ว่าการใช้โทรศัพท์ในการติดต่อสื่อสารก็เป็นปัจจัยที่สำคัญต่อความเจริญก้าวหน้าของประเทศ ทั้งยังช่วยลดปัญหาการจราจรได้ดีอีกด้วย แต่สภาพสังคมในปัจจุบัน การติดต่อสื่อสารด้วยโทรศัพท์ยังไม่สามารถใช้ประโยชน์ได้เต็มที่อันเนื่องมาจากการที่เราไม่สามารถติดต่อสื่อสารกับผู้ที่เราต้องการจะพูดด้วย ดังนั้นเพื่อช่วยลดปัญหาที่เกิดขึ้นนี้จึงได้จัดทำโครงงานเพื่อบันทึกเสียงและตอบรับโทรศัพท์อัตโนมัติ ทำให้การติดต่อสื่อสารเป็นไปอย่างต่อเนื่องทั้งยังช่วยลดทราฟฟิคของการใช้สายโทรศัพท์ลงอย่างมากอีกด้วย เพราะว่าเราสามารถติดต่อกับผู้รับได้ตลอดเวลาจึงไม่ต้องรอติดต่อในช่วงเวลาเย็นซึ่งมีทราฟฟิคสูงอีกด้วย ซึ่งโครงงานนี้ยังสามารถนำไปใช้กับชุมสายปลายทางอัตโนมัติ (Private Automatic Branch Exchange) หรือ PABX เพื่อเป็นโอเปอร์เรเตอร์ (operator) เพื่อใช้ในการประกาศข้อความการโอนสาย ถึงแม้ว่าในปัจจุบันเทคโนโลยีได้เจริญก้าวหน้าขึ้นเป็นอย่างมากบริษัทผู้ผลิต PABX ได้เพิ่มเติมความสามารถเหล่านี้ลงในตู้ชุมสายปลายทางอัตโนมัติแต่ระบบเหล่านี้มีราคาที่ค่อนข้างสูง อีกทั้งยังเป็นการเพิ่มเติมทางฮาร์ดแวร์ (HARDWARE) การแก้ไขเพิ่มเติมหน้าที่การทำงานที่เหมาะกับสภาพเศรษฐกิจในประเทศเราทำได้ไม่ค่อยสะดวก ดังนั้นคณะผู้จัดทำและอาจารย์ที่ปรึกษาจึงได้คิดทำโครงงานนี้เพื่อให้สามารถทำงานแทนได้อย่างมีประสิทธิภาพที่สามารถนำมาทดแทนกันได้ โดยได้นำเอาคอมพิวเตอร์ (Computer) เข้ามาช่วยซึ่งในปัจจุบันเราจะพบว่าคอมพิวเตอร์เป็นสิ่งที่จำเป็นต่อการดำเนินชีวิตของมนุษย์เป็นอย่างมากอีกทั้งยังมีราคาค่อนข้างถูก โดยระบบจะประกอบไปด้วยวงจรตรวจสอบสัญญาณเรียกเข้า (RINGING) เมื่อพบสัญญาณเรียกเข้าระบบจะทำการตรวจนับสัญญาณกระดิ่งเมื่อครบตามจำนวนครั้งที่ตั้งไว้โดยซอฟต์แวร์ก็จะทำการตอบรับโทรศัพท์อัตโนมัติจากนั้นจึงส่งข้อความไปยังผู้เรียกเพื่อให้ผู้เรียกกดเลขหมายภายในหรือกดรหัสสำหรับบันทึกเสียงหรือกดรหัสสำหรับสอบถามข้อความเพื่อทำการติดต่อระบบจะทำการถอดรหัสสัญญาณการกดปุ่มโทรศัพท์ (Dual Tone Multi Frequency) หรือ DTMF ที่ส่งมาจากผู้เรียกจากนั้นจึงทำการตรวจสอบรหัสที่ได้รับกับฐานข้อมูลถ้ามีรหัสอยู่ในฐานข้อมูลก็จะทำการโอนสายโทรศัพท์ซึ่งให้วงจรสร้างสัญญาณการกดปุ่มโทรศัพท์เพื่อส่งข้อมูลหมายเลขโทรศัพท์ให้กับชุมสายปลายทางอัตโนมัติโดยที่ระบบการบันทึกเสียงจะใช้ซอฟต์แวร์ในการควบคุมร่วมกับฮาร์ดแวร์ที่สร้างขึ้นและการ์ดเสียง โดยอาศัยคอมพิวเตอร์เป็นอุปกรณ์ควบคุมการทำงานทำให้มีความยืดหยุ่นในการใช้งานสูงทั้งยังทำให้เกิดประโยชน์การใช้งานคอมพิวเตอร์อย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ส่วนติดต่อกับคอมพิวเตอร์

ในการสร้างระบบบันทึกเสียงโทรศัพท์อัตโนมัติวงจรในส่วนติดต่อกับคอมพิวเตอร์เป็นส่วนที่ทำให้ระบบบันทึกเสียงโทรศัพท์ถูกควบคุมการทำงานจากคอมพิวเตอร์ได้ โดยประกอบไปด้วย สล็อต (SLOT) และวงจรในส่วนการถอดรหัสหมายเลขพอร์ต (Decode Port)

รายละเอียดของสัญญาณต่างๆบนสล็อต

OSC (oscillator : ขา B30) เป็นขาเอาต์พุต เป็นสัญญาณนาฬิกาความถี่ 14.31818 MHz

CLK (clock : ขา B20) เป็นขาเอาต์พุต เป็นสัญญาณนาฬิกาความถี่ 4.77 MHz ซึ่งถูกใช้เป็นสัญญาณนาฬิกาของระบบด้วย

RESET DRV (ขา B2) เป็นขาเอาต์พุต ซึ่งจะแอคทีฟ (ลอจิก "1") ในช่วงที่เราเริ่มจ่ายไฟให้แก่ระบบ และจะยังคงแอคทีฟไปจนกว่าระบบต่างๆ ภายในคอมพิวเตอร์พร้อมที่จะทำงานได้ จากนั้นสัญญาณนี้จะเปลี่ยนกลับเป็นลอจิก "0"

A0-A19 (Address Bus : ขา A31-A12) ขาสัญญาณทั้ง 20 ขานี้เป็นเอาต์พุตซึ่งใช้สำหรับกำหนดแอดเดรสของหน่วยความจำ หรืออุปกรณ์อินพุต/เอาต์พุต ที่ซีพียู (Central Processing Unit) หรือ CPU ต้องการติดต่อด้วยโดยที่สัญญาณ A0 จะเป็นบิตที่มีนัยสำคัญต่ำสุด (Least SignificantBit) และ A19 จะเป็นบิตที่มีนัยสำคัญสูงสุด (Most Significant Bit) สำหรับการอ้างแอดเดรสของพอร์ตอินพุต/เอาต์พุตนั้นจะใช้แอดเดรสเพียง 16 เส้นคือ A0-A15 ซึ่งจะทำให้อ้างแอดเดรสของพอร์ตได้ 64k พอร์ตโดยผ่านทางชุดคำสั่ง IN และ OUT ส่วนเส้นแอดเดรส A16 - A19 นั้นจะไม่ถูกนำมาใช้อย่างไรก็ตามภายในคอมพิวเตอร์จะใช้เส้นแอดเดรสในการอ้างแอดเดรสของพอร์ตเพียง 10 เส้นคือ A0 - A9 และค่าแอดเดรสที่ใช้งานจะต้องอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น

D0-D7 (Data Bus : ขา A9 - A2) ขาสัญญาณนี้จะเป็นแบบสองทิศทาง (Bi - Directional) ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ตอินพุต/เอาต์พุต กับคอมพิวเตอร์ โดยบิต D0 จะเป็นบิตที่มีนัยสำคัญต่ำสุด และ บิต D7 จะเป็นบิตที่มีนัยสำคัญสูงสุด

ALE (Address Latch Enable : ขา B28) เป็นขาสัญญาณเอาต์พุตที่สร้างขึ้นเพื่อใช้แสดงการเริ่มต้นของบัสไซเคิล และแสดงให้อุปกรณ์ภายนอกทราบว่าแอดเดรสที่คอมพิวเตอร์ต้องการติดต่อด้วยได้ถูกส่งออกมาบนบัสแอดเดรสแล้วโดยที่สัญญาณ ALE นี้จะเปลี่ยนจากลอจิก "1" เป็น "0" เมื่อค่าแอดเดรสที่ถูกต้อง ถูกส่งออกมาบนบัสข้อมูลแล้ว ดังนั้นขอบขาลงของสัญญาณ ALE นี้จะถูกใช้ในการแลตช์ค่าแอดเดรสจากบัสแอดเดรสข้อมูล AD0-AD7 ทำให้สามารถแยกค่าแอดเดรส (A0-A19) และข้อมูล D0-D7 ออกจากกันได้ อย่างไรก็ตามสัญญาณ ALE จะแอคทีฟเฉพาะในบัสไซเคิลที่สร้างขึ้นโดยซีพียูเท่านั้น โดยจะไม่แอคทีฟในระหว่างขบวนการ DMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{I/O CHECK}}$ (**I/O Channel Check : ขา A1)** ขาสัญญาณนี้เป็นอินพุตที่ใช้ในการแสดงความผิดพลาดเกี่ยวกับพาริตี้ที่เกิดขึ้นในการทำงานของวงจรอินเตอร์เฟส (INTERFACE) หรืออุปกรณ์อินพุตเอาต์พุตเมื่อขาสัญญาณนี้ได้รับลอจิก "0" จะทำให้ซีพียูถูกอินเตอร์รัพต์ (INTERRUPT) แบบนอนมาสค์เอเบิล (Non - Maskable) หรือ NMI อย่างไรก็ตามเราสามารถที่จะกำหนดให้วงจรภายในของคอมพิวเตอร์ทำการขออินเตอร์รัพต์ (เมื่อได้รับสัญญาณ $\overline{\text{I/O CHECK}}$) หรือไม่ก็ได้ โดยการกำหนดลอจิกของบิตของพอร์ตที่ควบคุม การขออินเตอร์รัพต์แบบนอนมาสค์เอเบิล คือบิต D7 ของพอร์ต 00A0H ในกรณีที่บิต D7 ของพอร์ต 00A0H ถูกเซตเป็น "1" ก็จะทำให้วงจรภายนอกขออินเตอร์รัพต์แบบนอนมาสค์เอเบิลได้ แต่ถ้าบิต D7 ของพอร์ต 00A0H ถูกเซตเป็น "0" ก็จะเป็นการดิสเอเบิล (DISABLE) การขออินเตอร์รัพต์แบบนอนมาสค์เอเบิล

I/O CHRDY (I/O Channel Ready : ขา A10) ขาสัญญาณนี้เป็นอินพุตที่ใช้เพิ่มช่วงเวลาในบัสไซเคิลในกรณีที่อุปกรณ์ I/O หรือหน่วยความจำที่เกี่ยวข้องกับขบวนการในบัสไซเคิลที่เกิดขึ้นนั้น ไม่สามารถทำงานทันตามช่วงเวลาปกติของบัสไซเคิลนั้นๆ ได้ (ช่วงเวลาของบัสไซเคิลที่เกี่ยวกับหน่วยความจำใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 4 ลูก หรือ 840 นาโนวินาที ในขณะที่บัสไซเคิลที่เกี่ยวกับอินพุตเอาต์พุต จะใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 5 ลูก หรือ 1.05 ไมโครวินาที) เมื่ออุปกรณ์อินพุต/เอาต์พุตหรือหน่วยความจำต้องการที่จะเพิ่มช่วงเวลาในบัสไซเคิลให้นานขึ้นอีกนั้น จะสามารถทำได้โดยการป้อนลอจิก "0" ให้กับขา I/O CHRDY ในช่วงเวลาที่อินพุต/เอาต์พุต หรือหน่วยความจำที่ถูกกำหนดนั้น ได้รับสัญญาณจากการถอดรหัสแอดเดรส และสัญญาณ $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$ หรือ $\overline{\text{IOW}}$ แอกทีฟ

IRQ2-IRQ7 (Interrupt Request 2 Through 7 : ขา B4 และ B25-B21) ขาสัญญาณทั้ง 6 นี้เป็นขาอินพุตที่ใช้สำหรับการขออินเตอร์รัพต์จากซีพียู โดยสัญญาณเหล่านี้จะต่อเข้ากับอุปกรณ์ที่ทำหน้าที่จัดลำดับการอินเตอร์รัพต์บนเมนบอร์ดโดยตรง โดยที่ IRQ2 จะมีลำดับความสำคัญสูงสุด (Highest Priority) และ IRQ7 มีลำดับความสำคัญต่ำสุด ในกรณีที่มีการขออินเตอร์รัพต์เกิดขึ้นคือระดับลอจิกที่ขา IRQ ขาใดขาหนึ่งถูกเปลี่ยนจากลอจิก "0" เป็นลอจิก "1" (ขอบขาขึ้น) อุปกรณ์ที่ทำหน้าที่จัดลำดับของการอินเตอร์รัพต์จะส่งสัญญาณอินเตอร์รัพต์ให้กับซีพียูเพื่อทำการขออินเตอร์รัพต์ สิ่งสำคัญในการขออินเตอร์รัพต์โดยผ่านทาง IRQ2-IRQ7 ก็คืออุปกรณ์ที่ทำฉารขออินเตอร์รัพต์โดยผ่านทาง IRQ ขาใดก็จะต้องรักษาระดับสัญญาณที่ขา IRQ นั้นให้แอกทีฟ (ลอจิก "1") อยู่จนกว่าจะได้รับสัญญาณอินเตอร์รัพต์แอคโนว์เลดจ์ (Interrupt Acknowledge) จากซีพียูเสียก่อน ถ้าไม่เช่นนั้นการขออินเตอร์รัพต์จะถูกยกเลิก และอินเตอร์รัพต์ Level 7 (IRQ 7) ก็จะถูกสร้างขึ้นโดยอัตโนมัติ ไม่ว่าการขออินเตอร์รัพต์ที่ถูกยกเลิกนั้นจะเป็นการขออินเตอร์รัพต์ใน Level หรือขาใด แต่อย่างไรก็ตามสัญญาณอินเตอร์รัพต์แอคโนว์เลดจ์จะไม่ถูกต่อออกมาที่ขาของสล็อต ดังนั้นโปรแกรมที่ทำการตอบสนองต่อการขออินเตอร์รัพต์ (Interrupt Service Routine) จะต้องทำการรีเซต สัญญาณ IRQ เอง โดยใช้คำสั่งในการเอาต์พอร์ตไปยังพอร์ตอินพุตเอาต์พุตที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{IOR}}$ (I/O Read : ขา B14) ขาสัญญาณนี้เป็นเอาต์พุตแอคทีฟที่ลอจิก "0" ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้ เป็นบัสไซเคิลของการอ่านข้อมูลจากพอร์ตอินพุต เอาต์พุต เพื่อให้พอร์ตอินพุตเอาต์พุตที่มีแอดเดรสตรงกันบนบัสแอดเดรส นั้นส่งข้อมูลออกมาบนบัสข้อมูลก่อนขอบขาขึ้นของสัญญาณ $\overline{\text{IOR}}$ ประมาณ 30 นาโนวินาทีเพื่อให้แน่ใจได้ว่าหน่วยประมวลผลกลาง สามารถรับข้อมูลได้ถูกต้อง สำหรับในขบวนการติดต่อกับหน่วยความจำโดยตรง (Direct Memory Access) 8237A-5 DMA-Controller จะทำการสร้างสัญญาณ $\overline{\text{IOR}}$ เอง โดยที่ค่าแอดเดรสที่อยู่บนบัสแอดเดรส จะเป็นค่าแอดเดรสของหน่วยความจำแทนที่จะเป็นแอดเดรสของพอร์ตอินพุต/เอาต์พุต ที่พอร์ตอินพุตเอาต์พุตที่ขา DMA ต้องการจะนำข้อมูลไปเก็บการที่พอร์ตใดจะส่งข้อมูลออกมาบนบัสข้อมูลนั้น จะอาศัยสัญญาณ DACK จาก DMA Controller เป็นตัวกำหนดเช่นกรณีที่สัญญาณ DACK1 แอคทีฟก็จะแสดงว่าพอร์ตอินพุตเอาต์พุตที่ส่งข้อมูลออกมาบนบัสข้อมูลก็คือพอร์ตอินพุตเอาต์พุต ที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1) เป็นต้น

$\overline{\text{IOW}}$ (I/O Write : ขา B13) ขาสัญญาณนี้เป็นเอาต์พุตแอคทีฟที่ลอจิก "0" ซึ่งถูกสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้แสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ตอินพุต เอาต์พุตที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้น รับข้อมูลที่อยู่บนบัสไปเก็บไว้อย่างไรก็ตาม เนื่องจากในช่วงเวลาที่สัญญาณ $\overline{\text{IOW}}$ นี้แอคทีฟ (ลอจิก "0") นั้นข้อมูลบนบัสข้อมูลอาจจะยังไม่สมบูรณ์ ดังนั้นในการออกแบบจึงควรใช้ขอบขาขึ้นของสัญญาณ $\overline{\text{IOW}}$ แทนขอบขาลงในการทำให้พอร์ตอินพุตเอาต์พุตที่เกี่ยวข้องรับข้อมูลไปเก็บไว้เพื่อให้ข้อมูลบนบัสข้อมูลนั้นสมบูรณ์เสียก่อน สำหรับ ในขบวนการ DMA นั้น DMA Controller จะทำการสร้างสัญญาณ $\overline{\text{IOW}}$ เอง โดยที่ค่าแอดเดรสของหน่วยความจำที่พอร์ตอินพุตเอาต์พุตที่ขอ DMA ต้องการจะอ่านข้อมูล

$\overline{\text{MEMW}}$ (Memory Write : ขา B11) ขานี้เป็นเอาต์พุตแอคทีฟที่ลอจิก "0" ซึ่ง 8288 Bus Controller สร้างขึ้นในระหว่างบัสไซเคิลในการเขียนข้อมูลลงในหน่วยความจำของซีพียู สัญญาณ $\overline{\text{MEMW}}$ นี้จะถูกส่งออกมาเพื่อให้หน่วยความจำที่แอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้น ทำการรับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้ โดยทั่วไปหน่วยความจำจะรับข้อมูลในช่วงขอบขาขึ้นของสัญญาณ $\overline{\text{MEMW}}$ สำหรับในระหว่างขบวนการ DMA นั้น 8237A-5 DMA-Controller จะทำการควบคุมบัสต่างๆ ของระบบแทนและสัญญาณ $\overline{\text{MEMW}}$ จะถูกใช้ในบัสไซเคิลของการเขียนข้อมูลลงในหน่วยความจำ ข้อมูลถูกส่งจากหน่วยอุปกรณ์อินพุตเอาต์พุตไปให้กับหน่วยความจำ

MEMR (Memory Read : ขา B12) ขานี้เป็นเอาต์พุตจาก 8288 สัญญาณนี้จะแอคทีฟ (ลอจิก "0") ระหว่างบัสไซเคิลของการอ่านข้อมูลจากหน่วยความจำของซีพียูเพื่อให้หน่วยความจำที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้น ทำการส่งข้อมูลออกมาบนบัสข้อมูล โดยที่หน่วยความจำนั้นจะต้องส่งข้อมูลออกมาในช่วงเวลา 30 นาโนวินาที ก่อนที่สัญญาณ $\overline{\text{MEMW}}$ จะกลับเป็นลอจิก "1" ทั้งนี้ก็เพื่อให้ซีพียูรับข้อมูลได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRQ1-DRQ3 (DMA Request 1-3 : ขา B18 , B6 และขา B16) ขาสัญญาณทั้งสามนี้เป็นสัญญาณอินพุตแอคทีฟที่ลอจิก "1" ซึ่งอุปกรณ์ภายนอกสามารถใช้ในการขอ DMA จากระบบ โดยการป้อนระดับสัญญาณลอจิก "1" ให้กับขา DRQ ขาใดขาหนึ่ง (ขา DRQ ทั้งสามนี้จะต่อเข้ากับ DRQ1-DRQ3 ของ 8237A-5)

$\overline{\text{DACK0}}$ - $\overline{\text{DACK3}}$ (DMA Acknowledge 0 - 3 : ขา B19 , B17 , B26 และ B15) สัญญาณทั้ง 4 นี้เป็นเอาต์พุตแอคทีฟที่ลอจิก "0" ซึ่ง 8237A-5 สร้างขึ้นเพื่อแสดงให้วงจรภายนอกที่ขอ DMA ทราบว่าการขอ DMA นั้นได้รับการตอบสนองแล้ว และ 8237A-5 จะเข้าสู่ขบวนการ DMA เพื่อให้การส่งผ่านข้อมูลระหว่างอุปกรณ์อินพุตเอาต์พุตที่ขอ DMA ทราบว่าการขอ DMA นั้นได้รับการตอบสนองแล้ว และ 8237A-5 จะเข้าสู่ขบวนการ DMA เพื่อให้การส่งผ่านข้อมูลระหว่างอุปกรณ์อินพุตเอาต์พุตที่ขอ DMA กับหน่วยความจำเกิดขึ้นได้โดยตรง (คือไม่ต้องผ่านซีพียู) โดยสัญญาณ DACK นี้จะแอคทีฟในแชนแนลใดก็ขึ้นอยู่กับว่าขบวนการ DMA ที่จะเกิดขึ้นนั้นเป็นการตอบสนองต่อการขอ DMA ในแชนแนลใด เช่นถ้าขบวนการ DMA ที่จะเกิดขึ้นนั้นเป็นการตอบสนองต่อการขอ DMA ให้กับแชนแนลที่ 2 (DRQ2) สัญญาณ $\overline{\text{DACK2}}$ ก็จะแอคทีฟ

AEN (Address Enable : ขา A11) สัญญาณนี้เป็นเอาต์พุตที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอคทีฟ (ลอจิก "1") นั้นเป็นบัสไซเคิลของขบวนการ DMA

T/C (Terminal Count : ขา B27) สัญญาณนี้ถูกสร้างขึ้นจากการนำเอาสัญญาณเอาต์พุตที่ขา EOP ของ 8237A-5 มากลับลอจิก (โดยใช้เกต Inverter) ทำให้สัญญาณ T/C นี้แอคทีฟที่ลอจิก "1"

บัสของแหล่งจ่ายไฟของระบบ

+ 5 Vdc (ขา B3 และ B29) ขาทั้งสองนี้ต่อกับแหล่งจ่ายไฟ DC +5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) + 5 เปอร์เซนต์ คืออยู่ช่วง +4.75 ถึง +5.25 Vdc

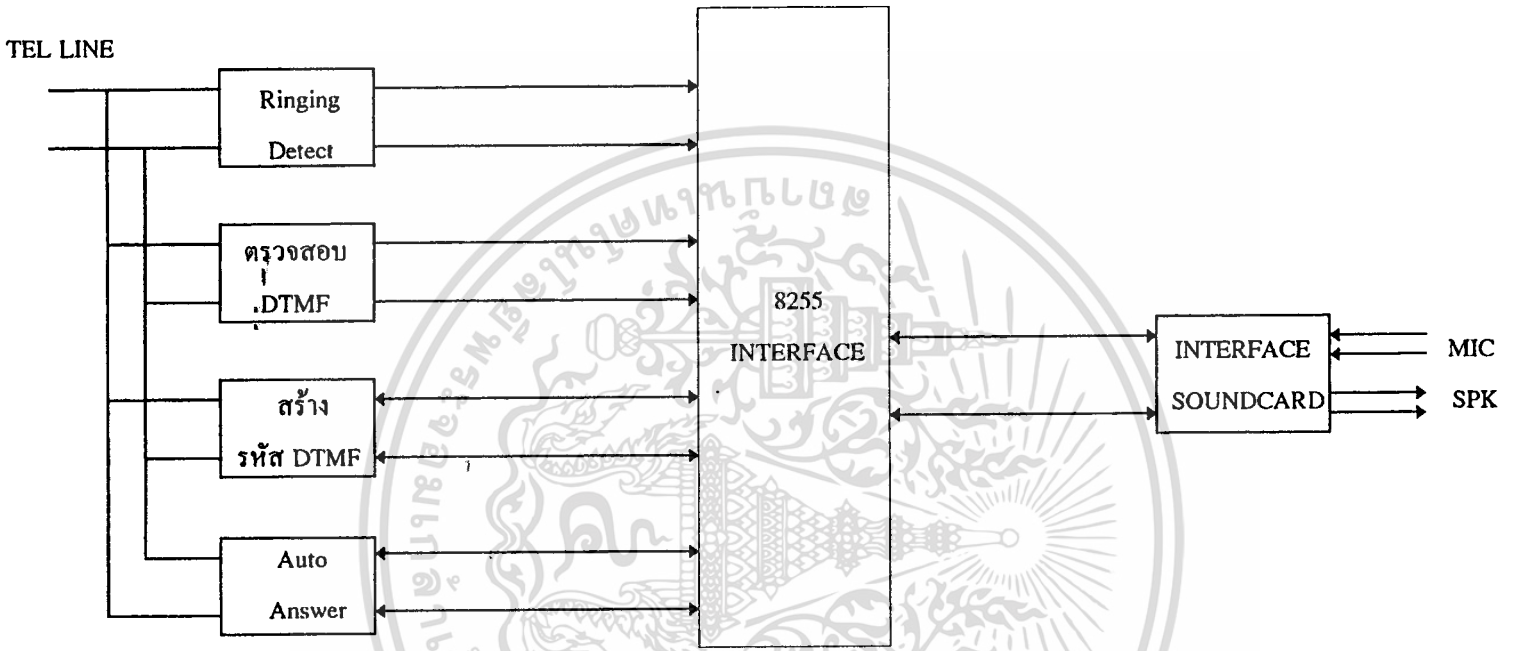
+ 12 Vdc (ขา B9) ขานี้ต่อกับแหล่งจ่ายไฟ DC +12V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) +5 เปอร์เซนต์ คืออยู่ในช่วง +11.4 ถึง +12.6 Vdc

- 5 Vdc (ขา B5) ขานี้ต่อกับแหล่งจ่ายไฟ DC -5V ของระบบ โดยมีค่าความเที่ยงตรง (Regulated) +10 เปอร์เซนต์ คืออยู่ในช่วง -5 ถึง -4.5 Vdc

- 12 Vdc (ขา B7) ขานี้จะต่อกับแหล่งจ่ายไฟ DC -12V ของระบบ โดยมีค่าความเที่ยงตรง +10 เปอร์เซนต์ คืออยู่ในช่วง -13.2 ถึง -10.8 Vdc

GND (ขา B1 , B10 และ B31) ขาทั้งสามนี้จะต่อเข้ากับกราวนด์ (Ground) ของระบบ

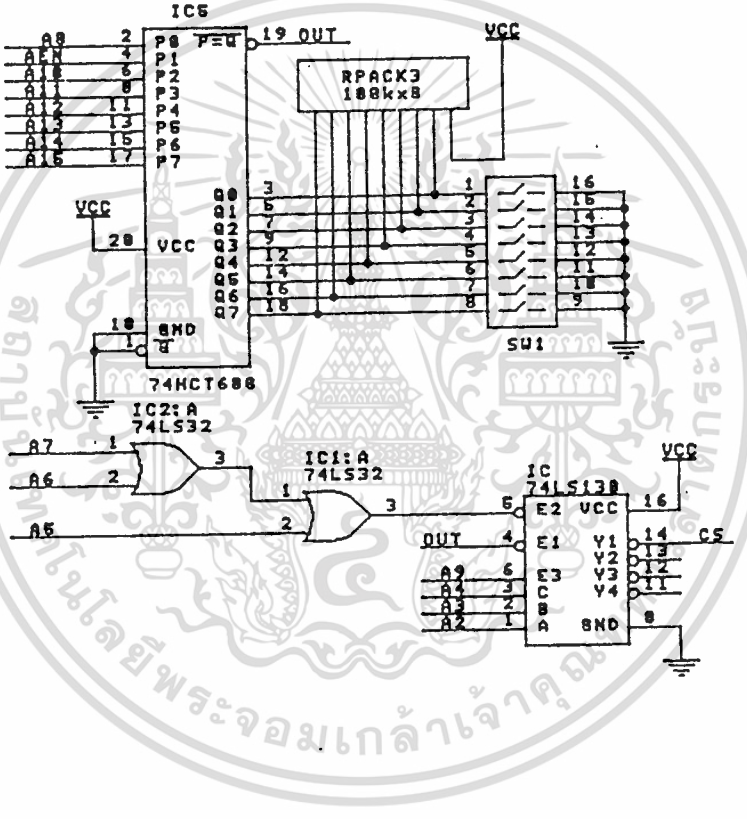
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงบล็อคไดอะแกรมของระบบบันทึกเสียงอัตโนมัติ

การทำงานของพอร์ตที่สร้างขึ้น (USER PORT)

การทำงานของพอร์ตที่สร้างขึ้น แบ่งออกเป็นส่วนถอดรหัสหมายเลขพอร์ต และส่วนอินพุต เอาต์พุต ที่ทำการติดต่อกับฮาร์ดแวร์ที่สร้างขึ้น ซึ่งในส่วนของวงจรถอดรหัสหมายเลขพอร์ต ได้ใช้ไอซี เบอร์ 74LS688 ซึ่งจะใช้สัญญาณจากแอดเดรสบัสที่นำมาถอดรหัสเข้ามาเปรียบเทียบกับสัญญาณที่ไอซี 74LS688 ถ้าข้อมูลตรงกันเอาต์พุตจะเท่ากับ "0" ไปอีนาเบิ้ล ไอซี 74LS138 โดยการทำงานจะอาศัย ดิฟสวิทช์ (DIP SWITCH) เพื่อเลือกหมายเลขพอร์ตที่ต้องการ แสดงวงจรดังรูปที่ 1.

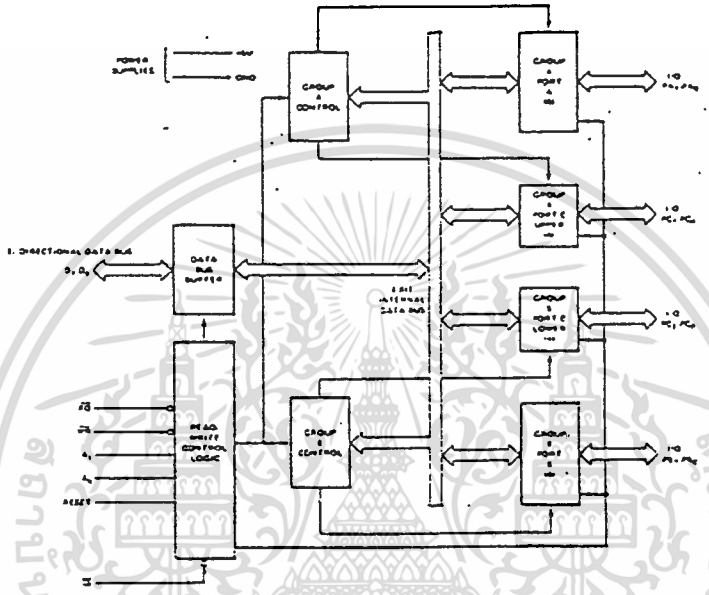


รูปที่ 1 วงจรแสดงการถอดรหัสหมายเลขพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดเกี่ยวกับ 8255

ไอซีเบอร์ 8255 เป็นอุปกรณ์ที่ใช้สำหรับเป็นพอร์ตอินพุต/เอาต์พุต โดยมีขาอินพุต/เอาต์พุตทั้งหมด 24 ขาซึ่งแบ่งได้เป็นขาของพอร์ต A พอร์ต B และพอร์ต C โดยที่แต่ละพอร์ตจะมี 8 ขา โดยพอร์ต C แยกได้เป็น 2 ส่วนคือพอร์ต C ล่างหรือตั้งแต่ PC0-PC3 มีจำนวน 4 บิต และพอร์ตบนหรือตั้งแต่ PC4-PC7 ที่พิเศษคือพอร์ตทุกพอร์ตเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุต



รูปที่ 2 โครงสร้างภายในของไอซี 8255

หน้าที่การทำงานของ 8255 (8255 Basic Functional Description)

1. บัฟเฟอร์บัสข้อมูล (Data Bus Buffer) DO-D7:บัสข้อมูลสองทิศทางเชื่อมโยงกับบัสข้อมูลของซีพียู ข้อมูลถูกส่งออกหรือเข้าตามคำสั่งเอาท์พุตหรืออินพุตผ่านบัฟเฟอร์บัสนี้

2. ลอจิกสัญญาณอ่าน/เขียน (Read / Write Control Logic) มีหน้าที่จัดการ รับส่งข้อมูลโดยส่งสัญญาณควบคุมแสดงสถานะการทำงานและควบคุมกลุ่มของพอร์ต A และพอร์ต B

3. สัญญาณเลือก CS (Chip Select) เป็นขาอินพุตซึ่งจะทำงานที่ลอจิก "LOW" เป็นสัญญาณเลือก 8255 ให้สามารถติดต่อกับซีพียูได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สัญญาณอ่าน RD (Read) เป็นขาอินพุตจะทำงานที่ลอจิก "LOW" เพื่อที่ทำให้ 8255 สามารถ ส่งข้อมูล (Status Information) ได้ทำงานเป็นอินพุตพอร์ต

5. สัญญาณเขียน WR (Write) เป็นขาอินพุตซึ่งจะทำงานที่ลอจิก "LOW" เพื่อที่ทำให้ 8255 สามารถส่งข้อมูล (Status Information) ได้ ทำงานเป็นอินพุตพอร์ต

6. ขา A0-A1 (Port Select 0 and Port Select 1) เป็นขาอินพุตซึ่งจะรับสัญญาณจากบัสแอดเดรส A0 และ A1 เพื่อควบคุมหรือเลือกพอร์ตได้ทั้ง 4 พอร์ต ภายใน 8255 ให้ทำงานตามสภาวะการอ่านหรือเขียน ซึ่งซีพียูส่งมา

7. ขา RESET (RESET) ขาอินพุตทำงานที่ลอจิก "HIGH" จะทำการเคลียร์ (clear) รีจิสเตอร์ (Register) ทั้งหมดภายในรีจิสเตอร์ควบคุม (Control Register) และพอร์ต A,B,C ให้ทำงานเป็นโหมด 0 อินพุต

8. การควบคุมพอร์ตกลุ่ม A และ B (Group A and Group B Controls) หน้าที่ของแต่ละกลุ่ม จะถูกโปรแกรมโดยซอฟต์แวร์จากซีพียู ซึ่งจะทำการส่งรหัสควบคุมมายังพอร์ตควบคุมของ 8255 โดยรหัสควบคุมที่จะส่งมาจะเป็นการสั่งให้ 8255 เลือกโหมดการทำงาน เซ็ทบิทหรือรีเซ็ทบิทในโหมดนั้นตามที่ซีพียูส่งออกมาซึ่งแต่ละกลุ่มที่ถูกควบคุมพอร์ตA,B จะรับคำสั่งจากลอจิกควบคุมสัญญาณอ่าน/เขียน (Read Write Control Logic) และรับรหัสควบคุมจากบัสข้อมูลภายใน และส่งสัญญาณ ไปจัดการควบคุมพอร์ตให้เป็นไปตามรหัสควบคุมที่ได้รับ

8.1 กลุ่มควบคุม A (Control Group A) จะควบคุมพอร์ต A และพอร์ต C ส่วนบน (PC4 - PC7)

8.2 กลุ่มควบคุม B (Control Group B) จะควบคุมพอร์ต B และ พอร์ต C ส่วนล่าง (PC0 - PC3)

คุณสมบัติของพอร์ต

1. พอร์ต A สามารถทำงานเป็น พอร์ตเอาต์พุตขนาด 8 บิท แบบมีบัฟเฟอร์และแลตช์ข้อมูลได้ (Buffer / Latch) หรือเป็นพอร์ตอินพุตขนาด 8 บิท แบบเก็บข้อมูลค้างไว้ได้

2. พอร์ต B สามารถทำงานเป็นพอร์ตอินพุต/เอาต์พุตขนาด 8 บิท มีบัฟเฟอร์และแลตช์ข้อมูลได้ หรือเป็นพอร์ตอินพุตขนาด 8 บิต

3. พอร์ต C สามารถทำงานเป็นพอร์ตเอาท์พุตขนาด 8 บิทแบบมีบัฟเฟอร์และแลตช์ข้อมูลได้ หรือเป็นพอร์ตอินพุทขนาด 8 บิท ไม่ค้างข้อมูลสำหรับการอินพุต และที่พอร์ตนี้ยังแบ่งออกเป็น 2 ส่วน ส่วนละ 4 บิท ตามรหัสควบคุมเลือกโหมดการทำงานโดยแต่ละส่วนของ 4 บิท ตามรหัสควบคุมเลือกโหมดการทำงานซึ่งสามารถใช้สำหรับเป็นสัญญาณควบคุมหรือเป็นสัญญาณอินพุตแสดงสภาวะ (Status Signal Input) ร่วมกับพอร์ต A หรือ พอร์ต B ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034954

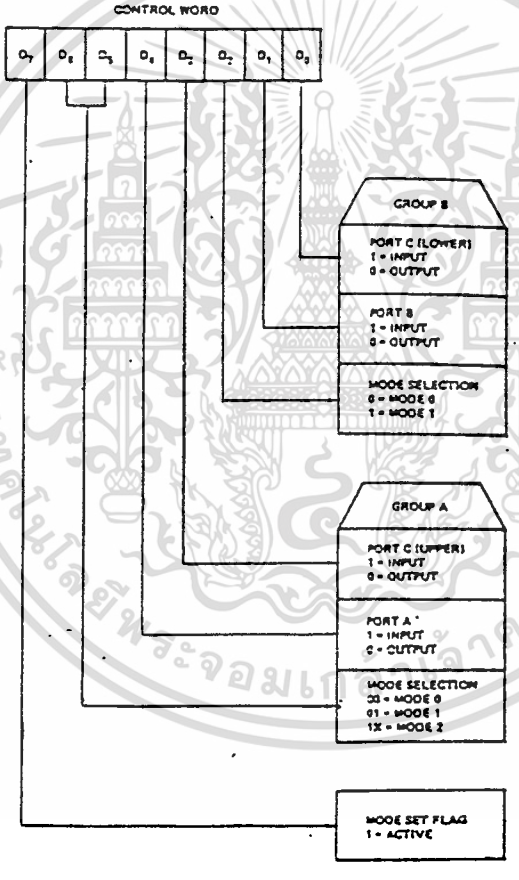
การเลือกโหมดของ 8255 (8255 Mode Select)

8255 สามารถเลือกการทำงานได้โดยซอฟต์แวร์ได้ 3 โหมดคือ

1. โหมด 0 (Mode 0) การทำงานเป็นพอร์ตอินพุต/เอาต์พุตแบบพื้นฐาน (Basic Input/Output)

2. โหมด 1 (Mode 1) การทำงานเป็นพอร์ตอินพุต/เอาต์พุตแบบสโตร์บ (Strobed Input/Output)

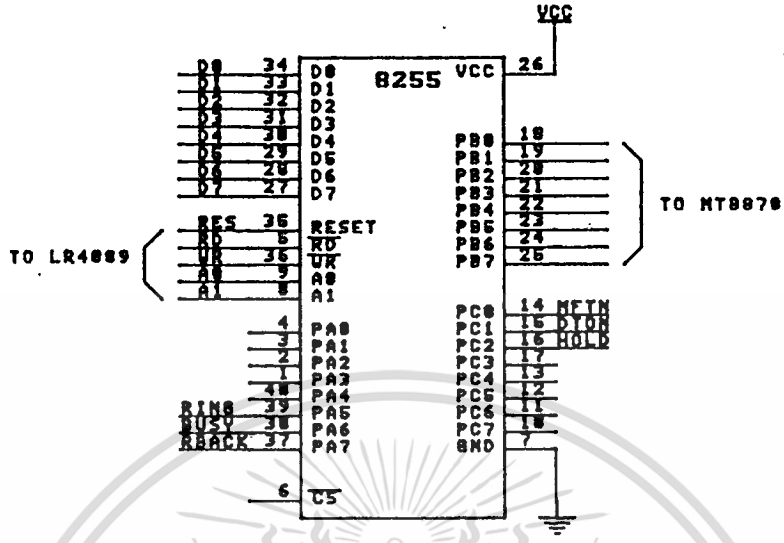
3. โหมด 2 (Mode 2) การทำงานเป็นแบบบัสสองทิศทาง (Bi Direction Bus)



รูปที่ 3 รหัสควบคุมและหน้าที่ของรหัสควบคุมแต่ละบิทของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่การทำงานของ 8255



รูปที่ 4 แสดงการใช้งาน 8255

การทำงานของพอร์ต A

PA0-PA4 เพื่อรับข้อมูลรหัส DTMF ที่ได้รับการแปลงมาจาก IC MT8870

PA5 ใช้สำหรับตรวจจับสัญญาณกระดิ่ง

การทำงานของพอร์ต B

PB0-PB7 เป็นเอาท์พุตพอร์ตเพื่อส่งข้อมูลให้กับ IC LR4089 เพื่อสร้างรหัส DTMF

การทำงานของพอร์ต C

PC0 เป็นสัญญาณที่ส่งไปควบคุมอนาล็อกสวิทช์เพื่อให้สัญญาณ DTMF ออกสู่สายภายนอก

PC1 เป็นสัญญาณที่ส่งไปควบคุมอนาล็อกสวิทช์เพื่อให้สัญญาณ DTMF จากภายนอกเข้าสู่ IC MT8870 เพื่อทำการถอดรหัสต่อไป

PC2 เป็นสัญญาณที่ส่งออกไปบังคับให้รีเลย์ทำงานเพื่อให้เกิดสภาวะการทำงานแบบยกหู

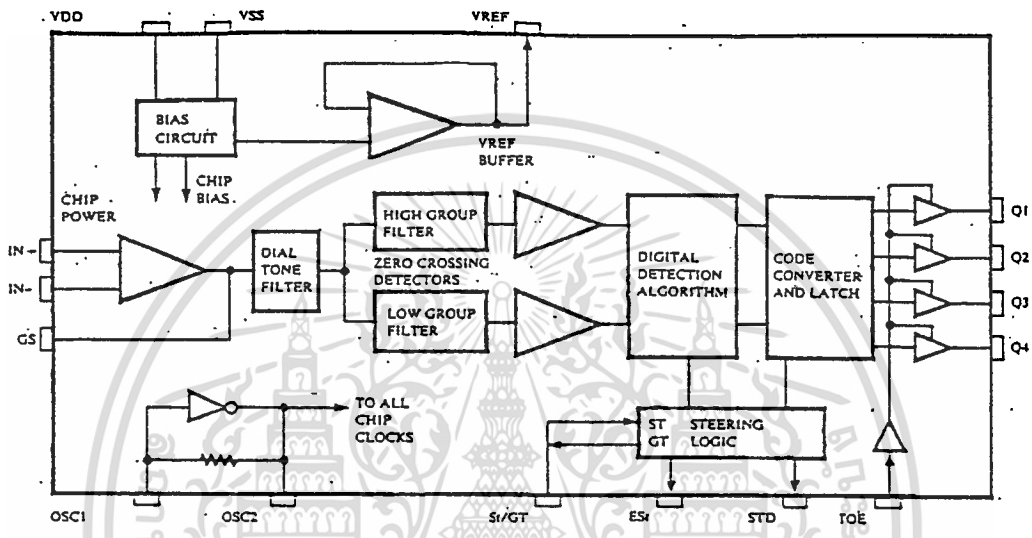
PC3 เป็นสัญญาณที่ส่งออกไปเพื่อให้สัญญาณเสียงจากภายนอกเข้าสู่ช่องสัญญาณไมโครโฟน

PC4 เป็นสัญญาณที่ส่งออกไปเพื่อให้ช่องสัญญาณจากลำโพงออกสู่สายภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ส่วนถอดรหัสความถี่โทรศัพท์

การถอดรหัสความถี่โทรศัพท์หมายถึงการแปลงสัญญาณความถี่ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่ม (ชนิด TONE หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิตอลซึ่งไอซี MT8870 ใช้แปลงความถี่โทรศัพท์ให้เป็นเลขฐานสองขนาด 4 บิต



รูปที่ 5 แสดงโครงสร้างภายในของ MT 8870

คุณสมบัติของ MT8870

- เป็นตัวรับและถอดรหัสความถี่
- กินไฟน้อยใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดไทม์ได้

การนำ MT8870 ไปใช้งาน

- นำไปใช้งานด้านรีโมตคอนโทรล
- เครื่องป้องกันโทรศัพท์ทางไกล
- ใช้ในงานเกี่ยวกับเครดิตการ์ด
- ใช้งานร่วมกับคอมพิวเตอร์
- ใช้ในเครื่องชุมสายขนาดย่อย
- การควบคุมอุปกรณ์ทางโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังก์ชั่นทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยีแบบ ISO^2-CMOS ในส่วนของวงจรที่ใช้กรองความถี่จะใช้เทคนิคของสวิตซ์คาปาซิเตอร์ฟิลเตอร์สำหรับกรองความถี่สูงและต่ำส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอล เพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิต และเช็คช่วงเวลาที่สัญญาณเข้ามา ส่วนภาคอินพุตเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอก เอาต์พุตเป็นวงจรแลตช์ 3 สถานะ

F_{LOW}	F_{HIGH}	NO	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1209	0	H	1	0	1	0
941	1336	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

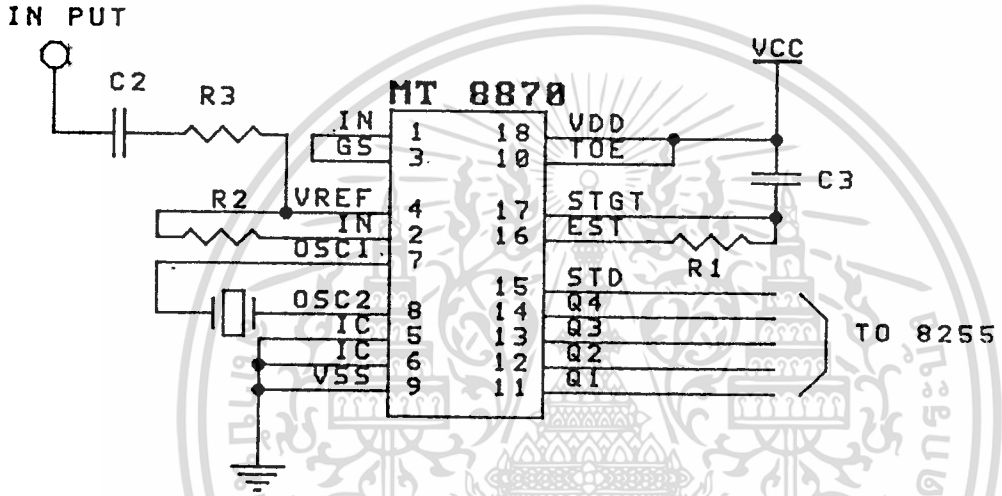
รูปที่ 6 แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่นการทำงานภายใน MT8870

ภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

- ภาคกรองความถี่ (Filter Section)
- ภาคถอดรหัส (Decoder Section)
- ภาคตรวจสอบสัญญาณ (Sterring Circuit)
- ภาคขยายสัญญาณความแตกต่าง (Differential Circuit)
- ภาคกำเนิดความถี่ (Oscillator Circuit)



รูปที่ 7 แสดงวงจรใช้งานเบื้องต้นของ MT8870

ภาคกรองสัญญาณความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่คือช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้กรองแถบความถี่อันดับ 6 ชนิดสวิทซ์คาปาซิเตอร์ (six - order switched capacitor band pass filter) ซึ่งความถี่ที่ แยกได้มี 2 ช่วงคือช่วงความถี่สูงและช่วงความถี่ต่ำ

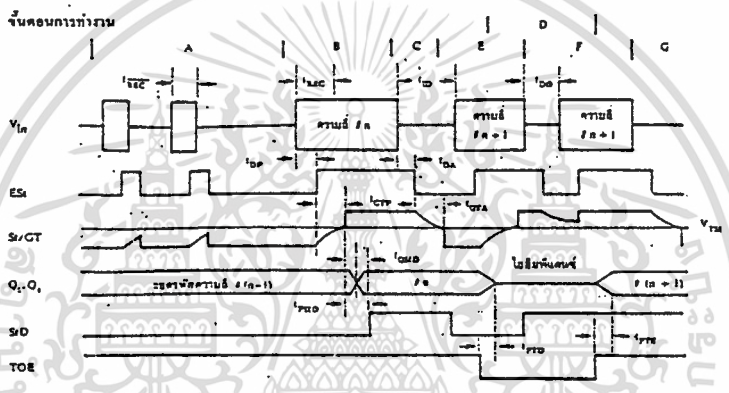
ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิตอล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่เพื่อป้องกันความถี่อื่นเข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา EST ก็จะแอคทีฟ สำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆนั้น แสดงในรูปที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุตจะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา EST จะเป็น"High"นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 6 เมื่อขา EST เป็น"high"ทำให้ VC สูงขึ้นตัวเก็บประจุ C จะคายประจุทำให้แรงดัน VC สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัส จึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงานขอให้ดูจากแผนภูมิเวลาหรือไทมิ่งไดอะแกรม (timing diagram) ในรูปที่ 8 จะเข้าใจได้ง่ายกว่า



รูปที่ 8 แสดงแผนภูมิเวลาของ MT8870

อธิบายขั้นตอนการทำงาน

A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาต์พุตไม่เปลี่ยน

B - ความถี่ # n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแลตช์ไว้ที่เอาต์พุต

C - จบความถี่ # n ช่วงห่างถูกต้อง เอาต์พุตยังคงแลตช์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D - เอาต์พุตเปลี่ยนเป็นไฮอิมพีแดนซ์

E - ความถี่ # n+1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแลตช์ไว้

F - ความถี่ # n+1 หายไป ช่วงห่างไม่ถูกต้อง เอาต์พุตยังคงแลตช์อยู่

G - จบความถี่ # n+1 ช่วงห่างถูกต้อง เอาต์พุตยังแลตช์อยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

Vin - สัญญาณความถี่ DTMF ที่เข้ามา

Est - Early Sterring output ใช้แสดงความถี่ที่ถูกต้อง

St/Gt - Sterring input / Guard Time output สำหรับต่อ RC ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Q1-Q4 - เอาต์พุต BCD ขนาด 4 บิต

Std - Delayed Sterring output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด

TOE - Tone Output Enable (input) ใช้ควบคุม Q1-Q4 ให้เป็นไฮอิมพีแดนซ์

T_{REC} - คาบเวลานานสุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง

t_{REC} - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

tID - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

tDO - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

tDP - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง

tDA - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

tGTP - การ์ดไทม์ของการปรากฏความถี่ DTMF

tGTA - การ์ดไทม์ของการหายไปของความถี่ DTMF

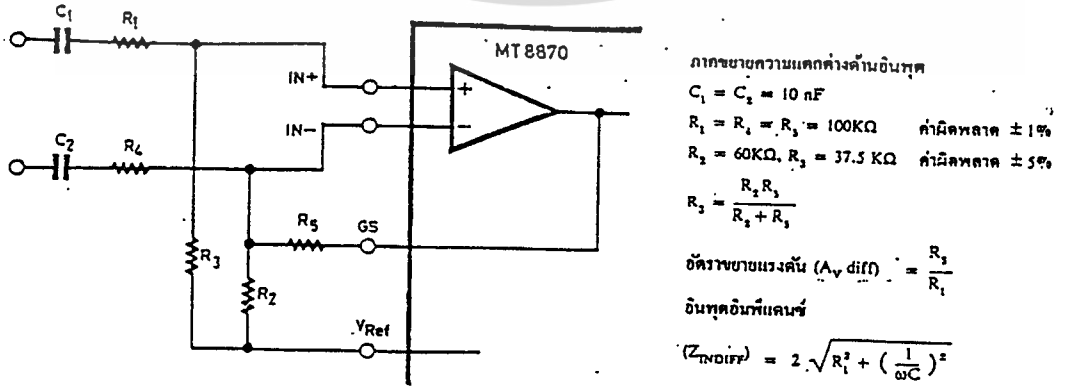
สำหรับคำว่าการ์ดไทม์นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับหรือมากกว่าที่เราตั้งไว้ จึงจะได้รับการยอมรับว่าสัญญาณความถี่นั้นถูกต้องหรือพูดได้ว่าเวลาที่เราตั้งไว้โดย RC ก็คือการ์ดไทม์นั้นเอง เมื่อสัญญาณความถี่เข้ามานานเท่าหรือมากกว่าเวลาที่ตั้งไว้จึงจะสามาร่ถแปลงเป็นตัวเลขได้ถ้าสัญญาณความถี่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป

ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุตของ MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเพิ่มเข้าไป รูปที่ 9 แสดงการต่อวงจรภายนอกเข้ากับอินพุตซึ่งสามารถคำนวณอัตราการขยายความแตกต่างของอินพุตและอิมพีแดนซ์ได้ดังนี้

$$\text{อัตราขยาย } (A_v\text{diff}) = R5/R1$$

$$\text{อินพุตอิมพีแดนซ์ } (Z_{in}\text{diff}) = 2\sqrt{R1^2 + (1/wc)^2}$$

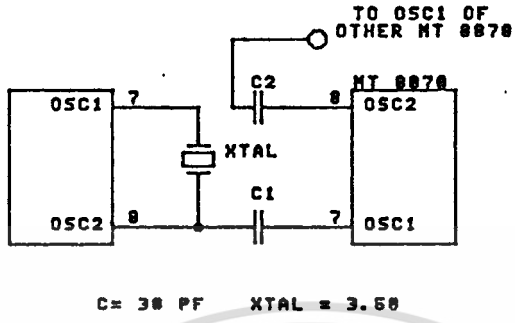


รูปที่ 9 แสดงการต่อวงจรภาคอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคกำเนิดความถี่

ในภาคนี้ภายในไอซีจะมีวงจรเวลาอยู่ภายใน เพียงแต่ต่อแร่คริสตอลขนาด 3.58 MHz ก็สามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่แสดงในรูปที่ 10



รูปที่ 10 แสดงการต่อวงจรผลิตความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ปัญหาที่เกิดขึ้นในการดำเนินงาน

- ได้ทำการทดลองต่อบอร์ดอเนกประสงค์ซึ่งวงจรในส่วนอินเตอร์เฟซปากพูดหูฟังสามารถทำงานได้โดยใช้อุปกรณ์เซมิคอนดัคเตอร์ IC TA31032P ซึ่งเป็นไอซี ที่ใช้ในการอินเตอร์เฟซปากพูดหูฟัง แต่เมื่อนำอุปกรณ์มาประกอบลงบนยูนิเวอร์แซลบอร์ด (UNIVERSAL BOARD) สำหรับพีซี แล้วการทำงานของวงจรเกิดการออสซิลเลทขึ้นทำให้มีเสียงสัญญาณรบกวนออกสู่ภายนอก
- การออกแบบส่วนวงจรตรวจจับสัญญาณกระดิ่งในครั้งแรกได้ใช้ฮาร์ดแวร์เพื่อนับสัญญาณกระดิ่งแล้วปรากฏว่าจำนวนพัลซ์ที่ออกมาจากการทำงานของวงจรตรวจจับสัญญาณกระดิ่งมีมากกว่าหนึ่งลูกต่อสัญญาณกระดิ่งเพียงหนึ่งครั้งจึงไม่สามารถทำให้วงจรนับทำงานได้ปกติ จึงได้ต่อเพิ่มวงจร RC และวงจรชมิตต์ทริกเกอร์ (Schmitt trigger) เข้าไปทำให้สัญญาณพัลซ์ที่ได้สามารถนำมาเข้าวงจรนับได้
- ในการทดลองครั้งแรกได้ใช้โทรศัพท์ประเภทไร้สาย ซึ่งจะผลิตความถี่เข้าไปกวนในวงจรมากทำให้สัญญาณต่างๆที่วัดได้จากออสซิลโลสโคปผิดเพี้ยนไปอย่างมาก
- ปัญหาที่เกิดจากการที่การ์ดเสียงได้ส่งสัญญาณรบกวนออกมาสู่ภายนอกตลอดเวลา โดยเป็นรูปของสัญญาณเสียงขนาดเล็ก ซึ่งทำให้เกิดปัญหาในการติดต่อสื่อสาร
- ปัญหาที่เกิดจากการนำซอฟต์แวร์ที่ใช้ในครั้งแรกคณะผู้จัดทำได้คิดว่าจะใช้โปรแกรมวิชวลเบสิก (VISUAL BASIC) มาทำการเขียนโปรแกรม แต่เนื่องจากโปรแกรมวิชวลเบสิกไม่สามารถทำการติดต่อกับพอร์ตได้ จึงนำเอาโปรแกรมวิชวลเบสิกมาทำฐานข้อมูล ส่วนซอฟต์แวร์ที่ใช้จัดการติดต่อกับพอร์ตได้ใช้ภาษาซี ในการควบคุมฮาร์ดแวร์แทน
- ได้ทำการทดลองเก็บข้อมูลเสียงลงฮาร์ดดิสก์ซึ่งใช้เวลาในการบันทึกข้อมูล 1 นาที ใช้จำนวนหน่วยความจำบนฮาร์ดดิสก์ประมาณ 1 เมกะไบต์ (Mbytes) โดยเป็นฟอร์แมตของสัญญาณเสียงคือ WAVE FILE นามสกุลของไฟล์คือ *.WAV
- ปัญหาที่เกิดจากการวางโครงสร้างของซอฟต์แวร์ในครั้งแรกคณะผู้จัดทำได้คิดรูปแบบในการเก็บไฟล์อยู่ในรูปคิว (QUEUE) แต่มีปัญหาในด้านการจัดการทางด้านซอฟต์แวร์จึงได้ใช้โครงสร้างของไฟล์แทน โดยมีการจัดเก็บข้อมูลเป็นไฟล์ หนึ่งไฟล์ต่อข้อมูลเสียงครึ่งนาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ขั้นตอนการแก้ปัญหาที่เกิดขึ้นในการดำเนินงาน

- ทำการเปลี่ยนจากอุปกรณ์เซมิคอนดัคเตอร์ไปเป็นอุปกรณ์พาสซีฟโดยใช้ไฮบริดทรานส์ฟอร์เมอร์ (Hybrid Transformer) ทำให้ไม่เกิดการออสซิลเลท
- เพิ่มวงจร RC เข้าไปและใช้วงจรชมิตต์ทริกเกอร์ทำให้สัญญาณพัลซ์ที่ได้สามารถนำมาเข้าวงจรนับจึงสามารถทำให้ซอฟต์แวร์สามารถอ่านค่าได้
- ทำการเพิ่มอนาล็อกสวิทช์เพื่อตัดสัญญาณเสียงที่รบกวนออกไปสู่ภายนอกทำให้สัญญาณเสียงที่การ์ดเสียงผลิตออกมาไม่ไปรบกวนวงจรภายนอก
- ได้ทำการทดลองเก็บข้อมูลเสียงลงฮาร์ดดิสก์ซึ่งใช้เวลาในการบันทึกข้อมูล 1 นาที ใช้จำนวนหน่วยความจำบนฮาร์ดดิสก์ประมาณ 500 kbytes โดยเป็นฟอร์แมตของสัญญาณเสียงคือ นามสกุลของไฟล์คือ *.VOC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6.

การใช้งานระบบตอบรับโทรศัพท์อัตโนมัติ

1. การติดตั้งฮาร์ดแวร์เป็นการติดตั้งการ์ดระบบตอบรับโทรศัพท์อัตโนมัติและการ์ดเสียงภายในไมโครคอมพิวเตอร์ โดยทำตามขั้นตอนดังนี้

1.1 ติดตั้งการ์ดระบบตอบรับโทรศัพท์อัตโนมัติในเครื่องไมโครคอมพิวเตอร์ และต่อสายโทรศัพท์เข้ากับการ์ดเสียง และช่องสัญญาณไมโครโฟนและลำโพงเข้ากับการ์ดเสียง

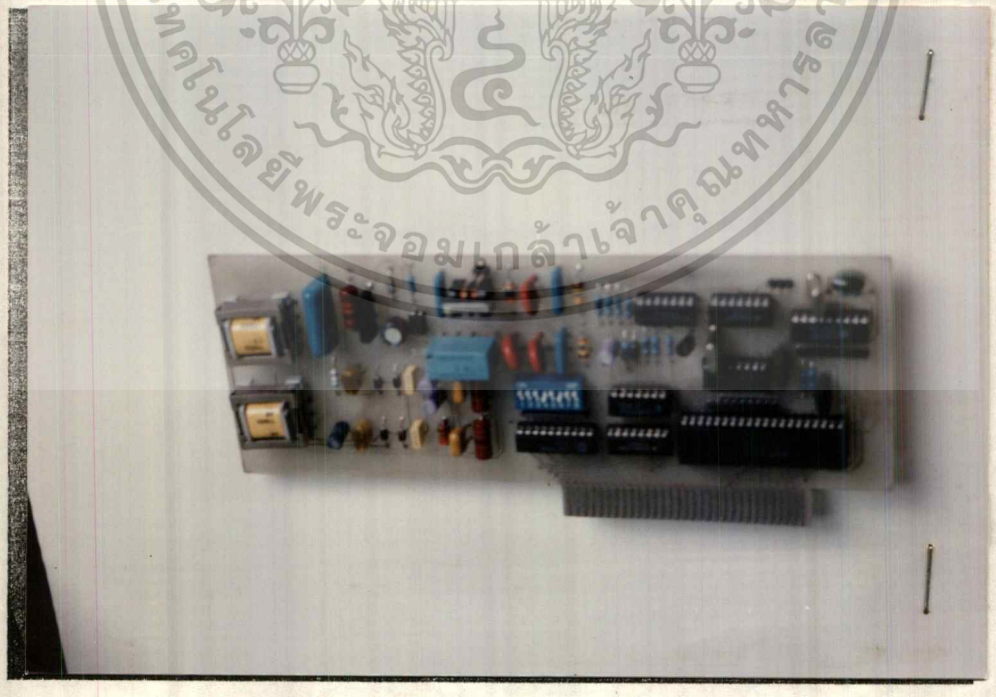
1.2 การติดตั้งการ์ดเสียง ควรติดตั้งห่างจากแอดเดรสของการ์ดระบบตอบรับโทรศัพท์อัตโนมัติซึ่งอยู่ที่แอดเดรส 300H-303H ส่วนการติดตั้งอินเตอร์รัพต์และดีเอ็มเอ สามารถศึกษาได้จากคู่มือของการ์ดเสียง

2. การติดตั้งซอฟต์แวร์สำหรับระบบตอบรับโทรศัพท์อัตโนมัติแบ่งออกได้ 2 ส่วนดังนี้

2.1 การติดตั้งซอฟต์แวร์จัดการฐานข้อมูล ซึ่งเป็นข้อมูลของเลขหมายโทรศัพท์และรหัสผ่านของผู้ใช้ โดยจัดทำขึ้นด้วยวิชวลเบสิกและทำงานภายใต้วินโดวส์ (Windows)

2.2 การติดตั้งซอฟต์แวร์ที่ใช้ควบคุมการบันทึกสัญญาณเสียงและการตอบสนองสัญญาณโทรศัพท์ โดยจัดทำขึ้นด้วยโปรแกรมภาษาซี ซึ่งทำงานภายใต้ระบบปฏิบัติการดอส (DOS)

3. การเริ่มการทำงานของระบบแบ่งออกเป็นการติดตั้งหมายเลขผู้ใช้ และรหัสผ่านโดยผ่านโปรแกรมที่จัดทำขึ้นด้วยวิชวลเบสิก จากนั้นจึงทำการสั่งให้โปรแกรมที่เขียนขึ้นด้วยภาษาซีทำงานเพื่อทำการตรวจจับสัญญาณโทรศัพท์



รูปที่ 11 การ์ดระบบตอบรับโทรศัพท์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของระบบ

1. ระบบจะทำการตรวจจับสัญญาณกระดิ่งเมื่อพบสัญญาณกระดิ่งจะเริ่มทำการจับเวลา ถ้าอยู่ภายในเวลาที่กำหนดมีสัญญาณกระดิ่งครบ 2 ครั้งจะส่งสัญญาณเสียงออกไป ถ้ามีสัญญาณกระดิ่งไม่ครบตามจำนวนที่ตั้งไว้และตามเวลาที่กำหนด ระบบจะเริ่มทำการตรวจจับสัญญาณกระดิ่งใหม่

Operator Automatic

Signal Ring : OFF

รูปที่ 12 แสดงการทำงานเริ่มต้นของระบบ

2. เมื่อนับสัญญาณกระดิ่งได้ตามจำนวนครั้งที่กำหนด ซึ่งอยู่ภายในเวลาที่กำหนดระบบจะส่งสัญญาณเสียงออกไป เพื่อแจ้งข่าวสารในการใช้งานระบบ

Operator Automatic

Signal Ring : ON 1 circle

ring

OK.

รูปที่ 13 แสดงการตอบรับโทรศัพท์อัตโนมัติ

3. รอรับสัญญาณ DTMF ภายในเวลาที่กำหนดเมื่อได้รับสัญญาณ DTMF ตัวแรกจะทำการตรวจสอบสถานะว่าอยู่ในการทำงานลักษณะใดซึ่งแบ่งออกได้ 4 รูปแบบคือ

3.1 การติดต่อกับโอเปอเรเตอร์

3.2 การติดต่อกับเลขหมายภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การฝากข้อความให้กับเลขหมายภายในที่ต้องการ

3.4 การสอบถามข้อมูลตามเลขหมายที่ต้องการ

4. ถ้าพบสัญญาณการกดปุ่มตัวแรกเป็นเลขศูนย์ หมายถึงระบบจะต้องทำการโอนสายไปยังโอเปอเรเตอร์ ซึ่งเลขหมายของโอเปอเรเตอร์ถูกกำหนดไว้ภายในโปรแกรม

welcome.voc

Transfer line to Operator

wait.voc

รูปที่ 14 แสดงการโอนสายไปยังโอเปอเรเตอร์

5. ถ้าพบสัญญาณกดปุ่ม ตัวแรกเป็นเลขหมายธรรมดาที่ไม่ใช่เลขศูนย์ จะทำการรอรับเลขหมายจนครบ 3 ตัว ซึ่งจะทำการตรวจสอบเวลาไปด้วยถ้าการกดรหัสไม่ได้อยู่ภายในเวลาที่กำหนดไว้ระบบจะส่งข้อความแสดงความผิดพลาดแล้วระบบจะเริ่มทำขั้นตอนที่ 2 ใหม่และถ้าการกดรหัสอยู่ภายในเวลาที่กำหนดครบ 3 เลขหมายจะทำการตรวจสอบเลขหมายภายในแฟ้มข้อมูล ถ้าไม่พบเลขหมายในแฟ้มข้อมูลจะแจ้งข่าวสารการผิดพลาดและเริ่มการทำงานในขั้นตอนที่ 2 และถ้าพบเลขหมายในแฟ้มข้อมูลระบบจะทำการส่งข้อความเสียงแล้วโอนสายและเริ่มต้นทำงานตามขั้นตอนที่ 1

welcome.voc

Transfer line to Number 101

wait.voc

รูปที่ 15 แสดงการโอนสายภายใน

6. ถ้าพบว่าสัญญาณการกดปุ่มตัวแรกเป็นรหัสดอกจันทร์ (*) ระบบจะทำการส่งข้อความเสียงไปยังปลายทางเพื่อแจ้งการกดเลขหมายภายในหลังจากแจ้งข้อความแล้วจะเริ่มจับเวลาในการตรวจสอบสัญญาณการกดปุ่ม ถ้าหากเลขหมายภายใน 3 ตัวไม่ถูกป้อนภายในเวลาที่กำหนดระบบจะส่งข้อความเสียงแจ้งรายงานการผิดพลาดแล้วเริ่มทำงานในขั้นตอนที่ 2 ใหม่ และถ้าเลขหมายภายในถูกป้อนภายในเวลาที่กำหนดระบบจะทำการค้นหาภายในฐานข้อมูลเมื่อไม่พบว่าอยู่ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งข้อความการผิดพลาดแจ้งไปยังปลายทางแล้วเริ่มทำงานในขั้นตอนที่ 2 ใหม่ แต่ถ้าพบว่าเลขหมายอยู่ในฐานข้อมูล ระบบจะส่งข้อความเสียงแจ้งให้ปลายทางทราบว่าระบบจะเริ่มต้นบันทึกเสียงหลังจากสิ้นสุดเสียงสัญญาณ ระบบจะทำการบันทึกเสียงภายในเวลาครึ่งนาที เมื่อครบครึ่งนาทีระบบจะแจ้งข้อความขอบคุณแล้วเริ่มทำขั้นตอนที่ 2 ใหม่

welcome.voc

File Lastfile : 10102.voc

voice record ended.

รูปที่ 16 แสดงการบันทึกสัญญาณเสียง

7. ถ้าพบว่าเลขหมายที่กดเป็นเครื่องหมายสี่เหลี่ยม (#) ระบบจะทำการส่งข้อความเสียงไปสู่สายภายนอกเพื่อรอรับสัญญาณการกดปุ่ม ซึ่งเป็นเลขหมายภายใน จากภายนอกและทำการจับเวลาในการป้อนรหัส ถ้าการป้อนรหัสไม่ได้อยู่ภายในเวลาที่กำหนด ระบบจะทำการส่งข้อความเสียงเพื่อแจ้งเตือนเวลาในการป้อนรหัส แล้วระบบจะเริ่มทำขั้นตอนที่ 2 ใหม่ แต่ถ้าการป้อนรหัสอยู่ภายในเวลาที่กำหนดและอยู่ภายในเวลาที่ตั้งไว้ระบบจะทำการนำเอาสัญญาณที่ป้อนมาเข้าไปตรวจสอบกับแฟ้มข้อมูลถ้าหากไม่มีหมายเลขอยู่ภายในฐานข้อมูลระบบจะทำการส่งข้อความเสียงแจ้งข้อผิดพลาดและจะเริ่มเข้าสู่ขั้นตอนการทำงานในขั้นที่ 2 แต่ถ้าหากมีเลขหมายอยู่ภายในฐานข้อมูล ระบบจะทำการส่งข้อความเสียงแจ้งการป้อนรหัสผ่านและรอรับรหัสผ่านภายในเวลาที่กำหนดเมื่อมีการป้อนรหัสผ่านภายในเวลาที่กำหนดและจะทำการตรวจสอบรหัสผ่านกับเลขหมายภายในถ้าหากรหัสผ่านที่ป้อนตรงกับแฟ้มข้อมูลระบบจะทำการส่งข้อความเสียงที่เก็บอยู่ในฮาร์ดดิสก์ ที่อยู่ในพาธ (path) ที่ทำการพิจารณา จากเลขหมายที่ป้อนครั้งแรกไปจนกว่าจะหมดข้อความเสียงและ เมื่อทำการส่งข้อความเสียงไปหนึ่งข้อความแล้วระบบจะสอบถามว่าจะลบข้อมูลนั้นทิ้งหรือไม่และรอการกดคีย์ใดๆ เมื่อมีการกดคีย์ใดๆ ข้อมูลเสียงที่ผ่านมาจะถูกลบทันทีและจะทำการจัดเรียงโครงสร้างของไฟล์ใหม่แต่ถ้าไม่มีการกดคีย์ใดๆข้อมูลเสียงซึ่งเป็นข้อความที่ต่อมาก็จะถูกส่งออกไปเรื่อยๆโดยทำเช่นนี้ไปจนกว่าจะหมดข้อมูลที่อยู่ในพาธนั้นและเมื่อหมดข้อความแล้วระบบจะเริ่มทำขั้นตอนที่ 1 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

welcome.voc

Password.voc

Not able to delete Dummy.voc

File name : 10100.voc

File name is deleted : 10100.voc

File name : 10100.voc

File name : 10101.voc

File name is deleted : 10101.voc

File name : 10101.voc

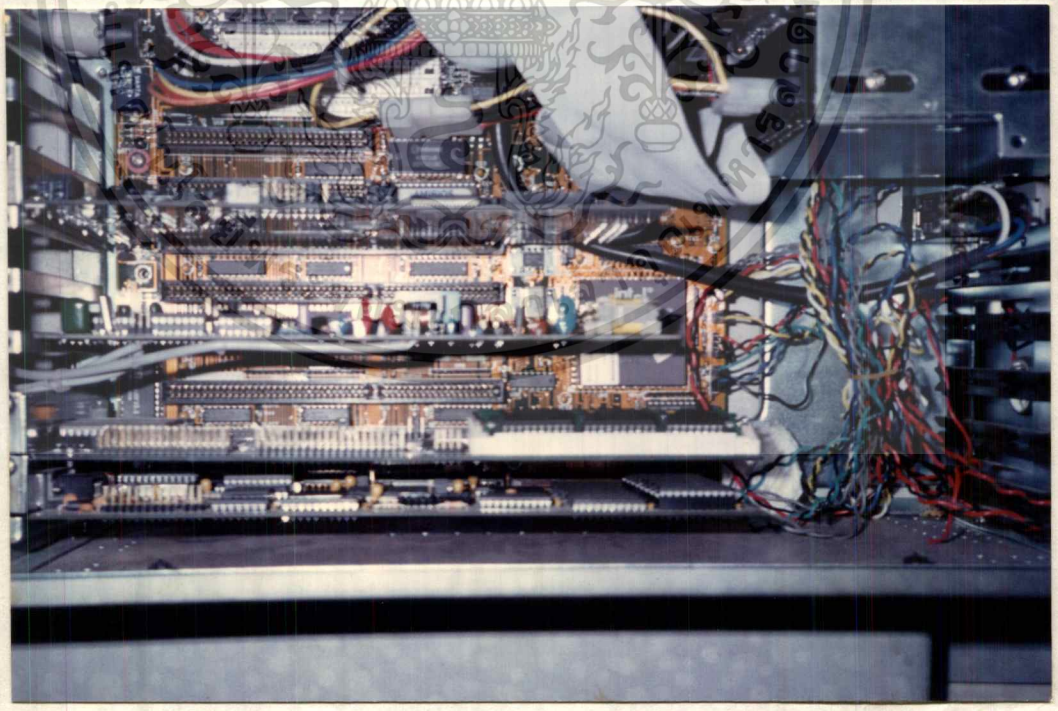
File name : 10102.voc

File name : 10103.voc

File name is deleted : 10103.voc

File name : 10103.voc

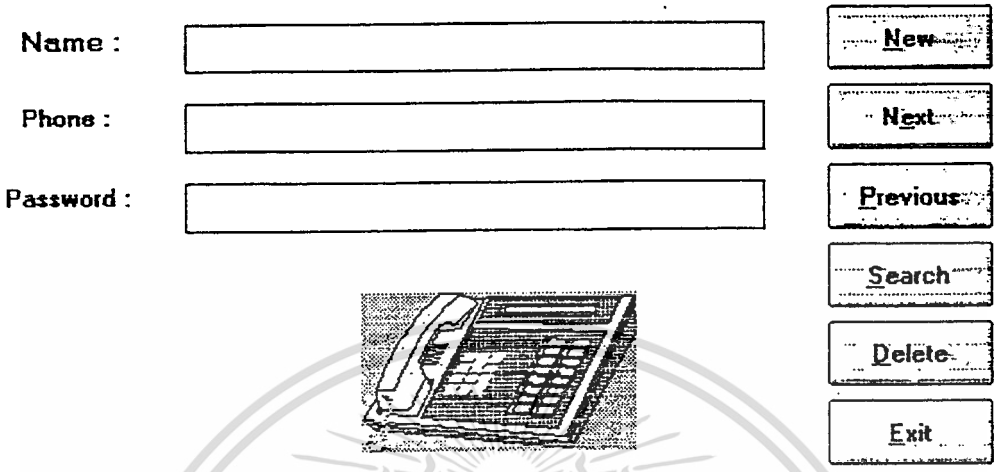
รูปที่ 17 แสดงการสอบถามข้อความ



รูปที่ 18 แสดงการติดตั้งการ์ดระบบตอบรับโทรศัพท์อัตโนมัติ

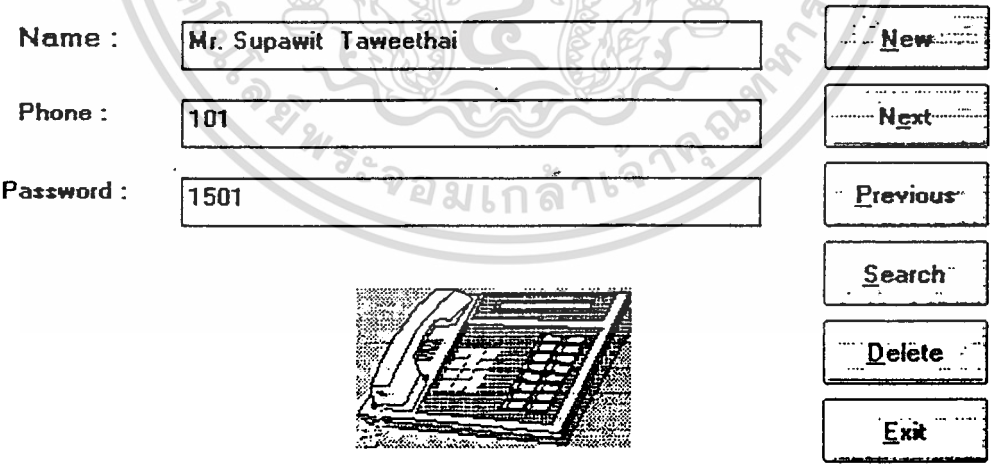
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. แสดงการเริ่มต้นการทำงานของโปรแกรมวิชวลเบสิกในการสร้างแฟ้มข้อมูล phone.dat ซึ่งประกอบไปด้วยข้อมูลคือ ชื่อ-นามสกุล เจ้าของเลขหมาย ,เลขหมาย และ รหัสผ่าน



รูปที่ 19 แสดงการเริ่มต้นการทำงานของโปรแกรมสร้างแฟ้มข้อมูล

9. การป้อนข้อมูลลงบนโปรแกรมที่สร้างขึ้นโดยโปรแกรมจะเก็บข้อมูลไว้ในไฟล์ phone.dat เพื่อให้โปรแกรมที่เขียนขึ้นด้วยภาษาซีอ่านข้อมูลนำไปใช้ในการตรวจสอบเลขหมายและรหัสผ่าน



รูปที่ 20 แสดงการป้อนข้อมูลเพื่อสร้างแฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการดำเนินงาน

ในโครงงานชิ้นนี้ได้แบ่งการดำเนินงานเป็น 2 ชนิดคือฮาร์ดแวร์ และ ซอฟต์แวร์

ผลการดำเนินงานทางด้านฮาร์ดแวร์

- จัดทำวงจร (circuit)
- จัดทำแผงวงจรแบบเดินสาย (wirewrap) เป็นต้นแบบ
- จัดทำเป็นแผ่นปรินท์ที่ลงอุปกรณ์และทำการทดสอบการทำงานทุกอย่างทำงานถูกต้อง

ผลการดำเนินงานด้านซอฟต์แวร์

- จัดทำฐานข้อมูลที่สร้างขึ้นด้วยโปรแกรมวิชวลเบสิก
- จัดทำโปรแกรมในการควบคุมฮาร์ดแวร์เพื่อบันทึกเสียงและเล่นข้อความเสียงรวมทั้งการจัดการสัญญาณโทรศัพท์ และการจัดการโครงสร้างของไฟล์เก็บเป็นแฟ้มข้อมูลเสียงโดยใช้โปรแกรมภาษาซีในการควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

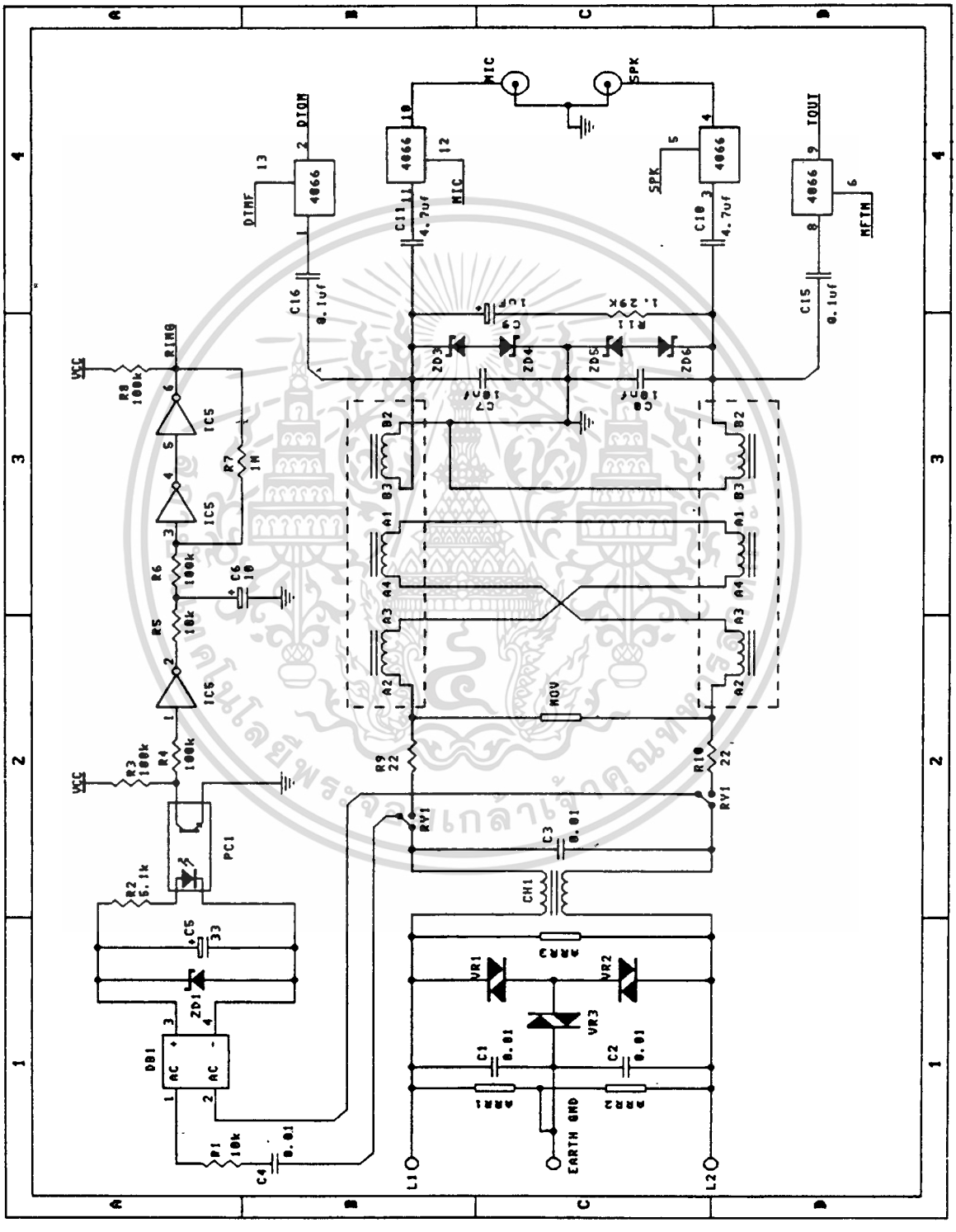
ข้อเสนอแนะ

1. ควรเพิ่มความสามารถในการรับคู่สายโทรศัพท์เพื่อการบันทึกเสียงได้พร้อมกันหลายคู่สาย โดยอาจใช้การมัลติเพล็กซ์สัญญาณ เพื่อทำการบันทึกเสียง
2. ควรเพิ่มวงจรในการตรวจสอบสัญญาณ BUSY TONE และ RINGBACK TONE เพื่อให้สามารถนำไปใช้กับตู้สาขาอัตโนมัติของบริษัทต่างๆ ได้แพร่หลาย
3. ทำการสร้างวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล และดิจิตอลเป็นอนาล็อก รวมอยู่ในการ์ดเดียวกัน เพื่อลดราคาของการสร้างวงจรและลดความยุ่งยากในเขียนซอฟท์แวร์
4. ควรใช้ระบบปฏิบัติการบนวินโดวส์เพื่อให้สามารถทำงานได้พร้อมๆ กันหลายงาน ทำให้การใช้งานคอมพิวเตอร์ได้เกิดประโยชน์สูงสุด
5. การทำงานของซอฟต์แวร์ ควรเพิ่มฟังก์ชั่นการทำงานในการโอนข้อมูลเสียงให้กับโทรศัพท์ภายในเลขหมายอื่นได้ กรณีที่ข้อมูลที่ฝากไว้ไม่ใช่ของผู้รับ
6. การทำงานของซอฟต์แวร์ ควรเพิ่มฟังก์ชั่นการทำงานให้เจ้าของเลขหมายภายในสามารถที่จะฝากข่าวสารให้กับผู้เรียกที่เรียกเข้ามาได้
7. การทำงานของซอฟต์แวร์ ควรเพิ่มฟังก์ชั่นการทำงานให้สามารถส่งข้อมูลเสียงที่ถูกฝากไว้ผ่านทางโทรศัพท์ไปยังผู้รับที่อยู่ที่อื่นได้

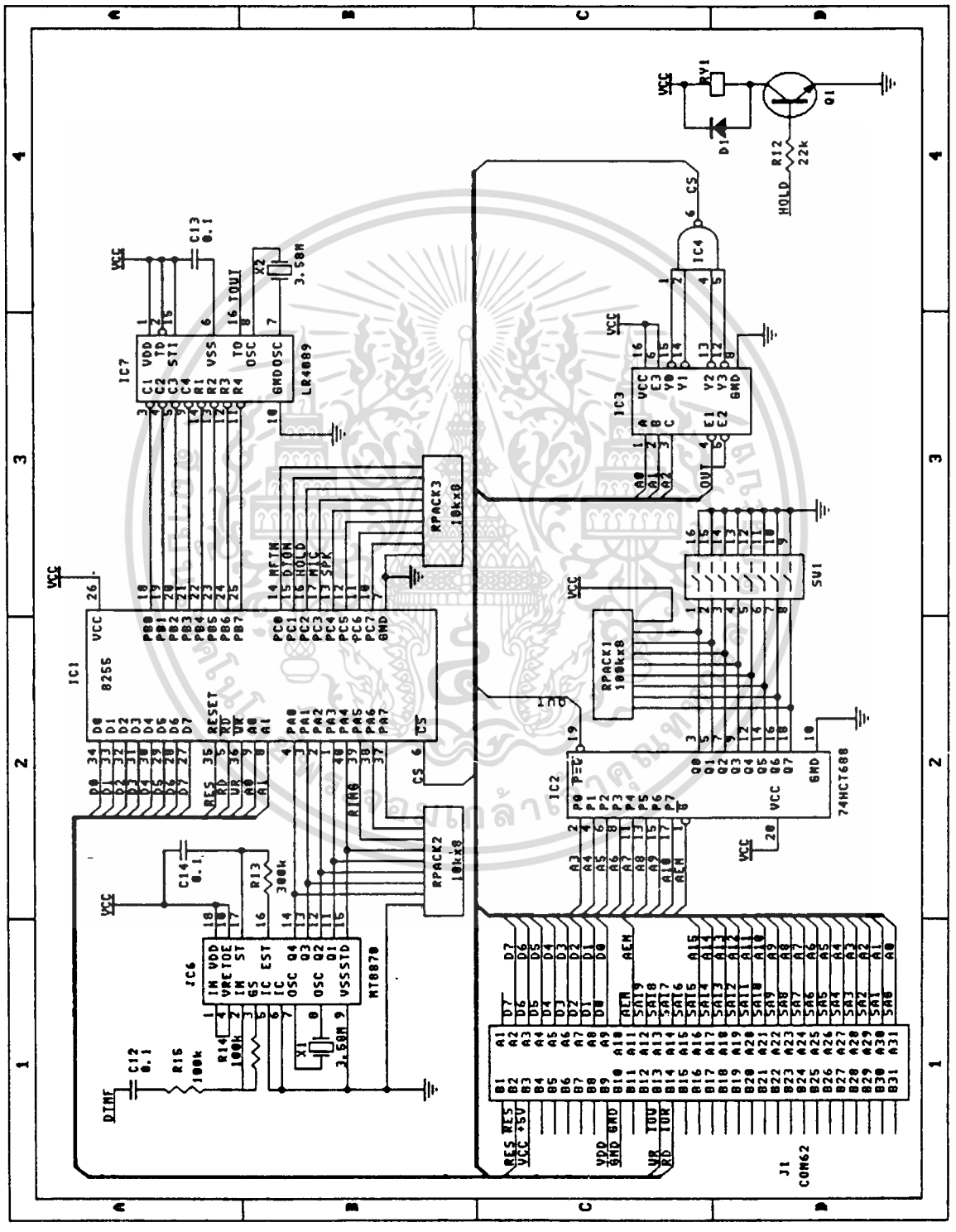
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

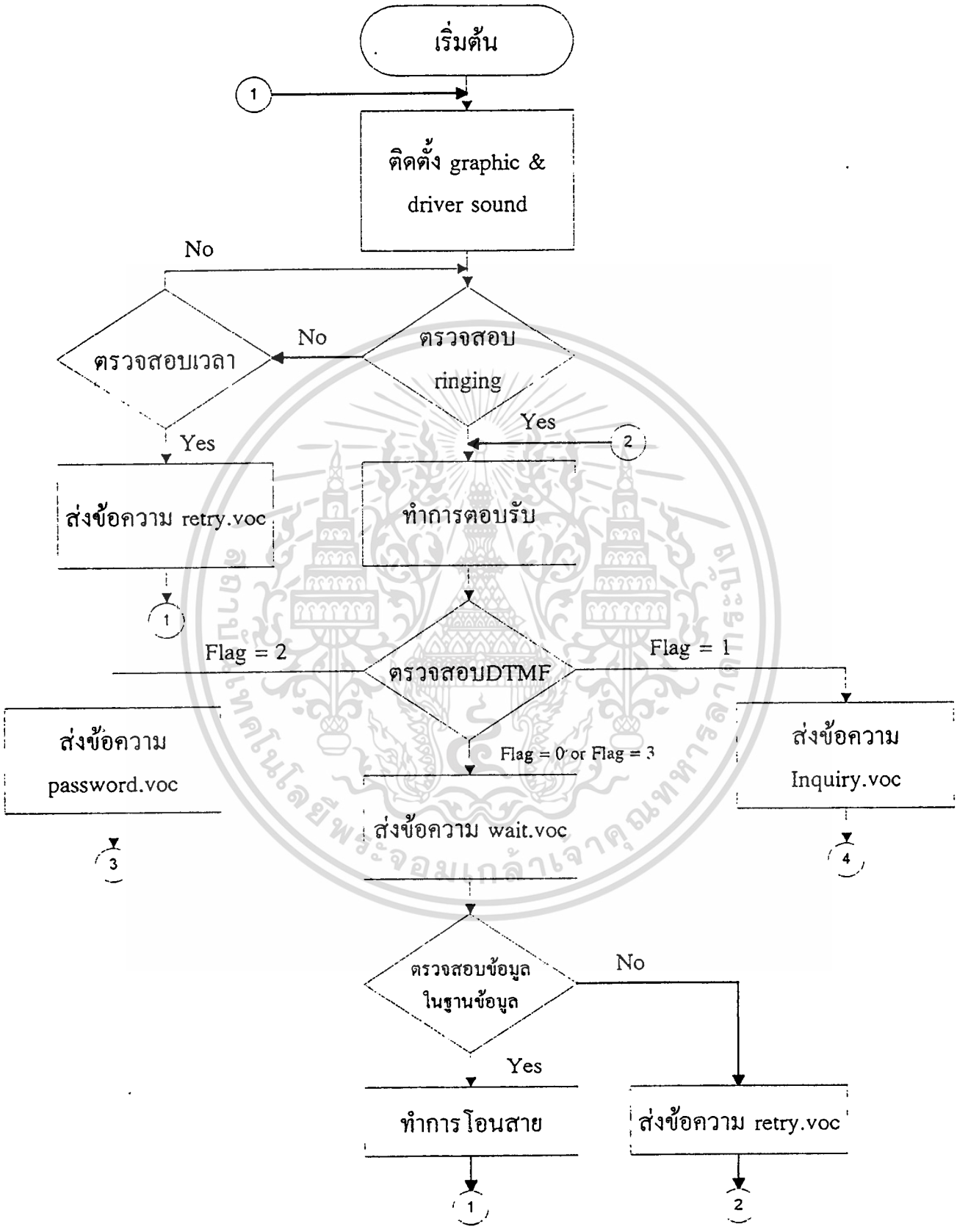


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

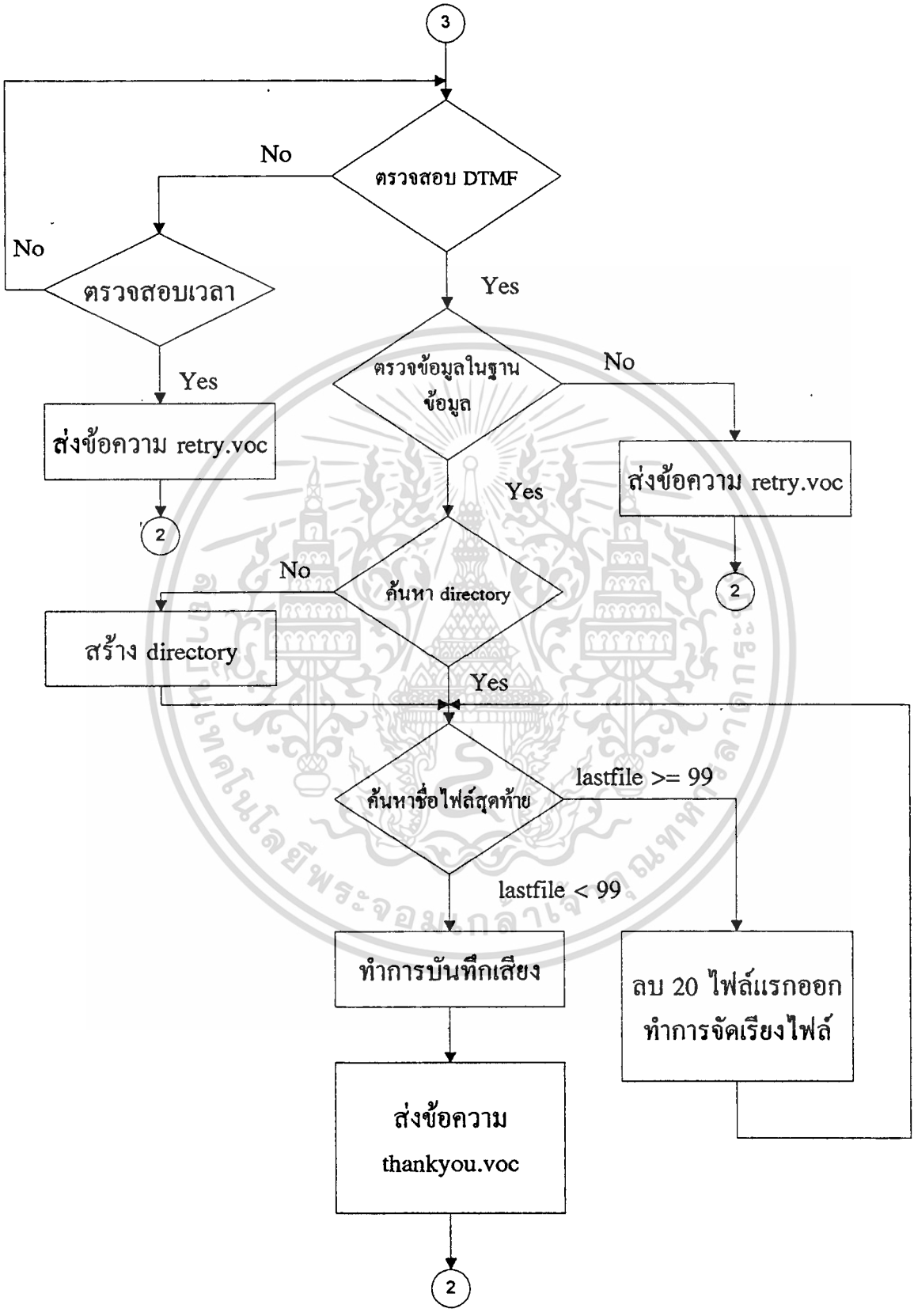
ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

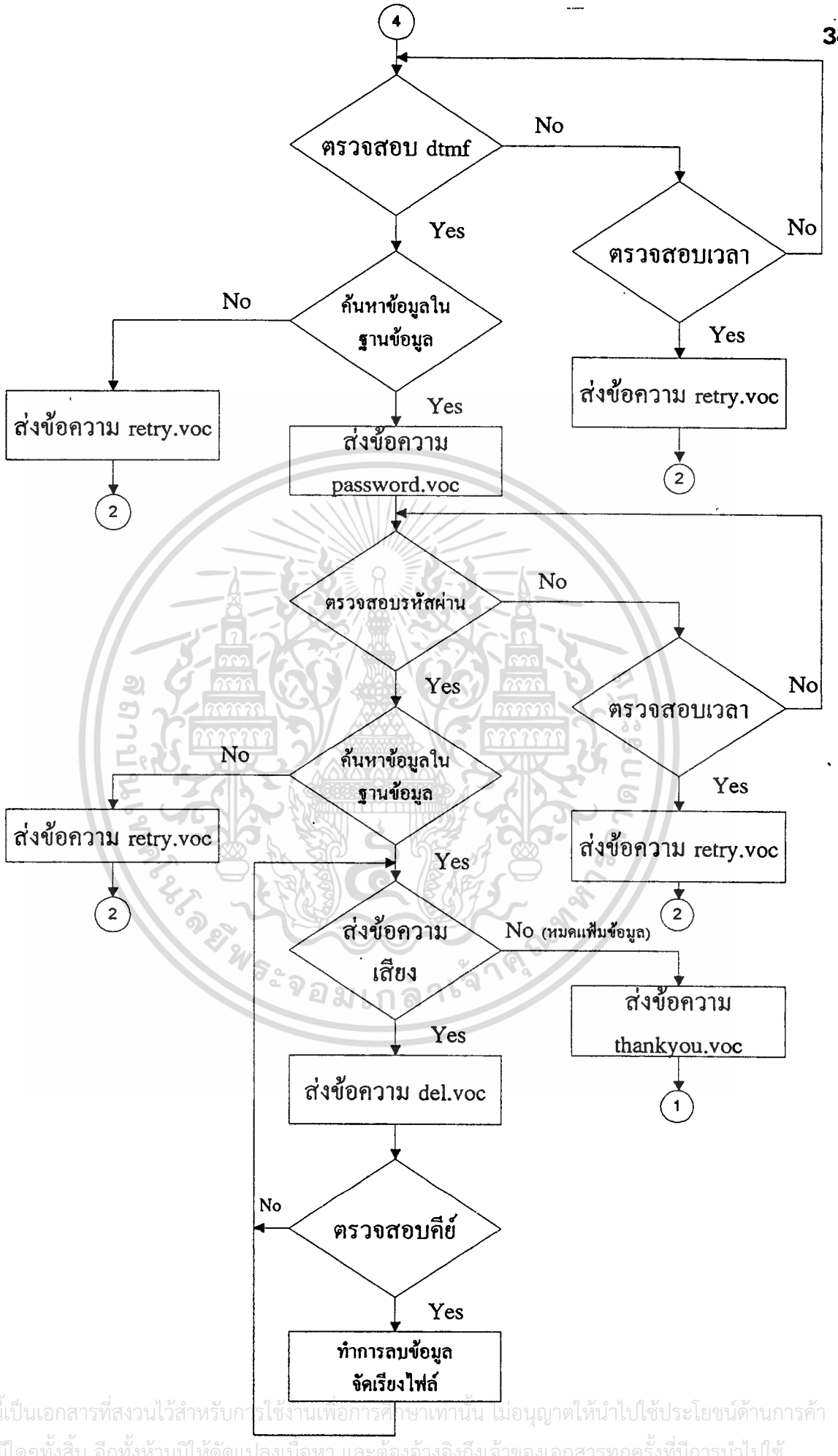
FLOWCHART



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม VISUAL BASIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PHONE.FRM - 1

```
Option Explicit
Dim Person As PersonInfo
Dim FileNum As Integer
Dim RecordLen As Long
Dim CurrentRecord As Long
Dim LastRecord As Long



Sub SaveCurrentRecord ()
    Person.Name = txtName.Text
    Person.phone = txtPhone.Text
    Person.Password = txtPassword.Text
  Put #FileNum, CurrentRecord, Person
End Sub

Sub ShowCurrentRecord ()
Get #FileNum, CurrentRecord, Person
    txtName.Text = Trim(Person.Name)
    txtPhone.Text = Trim(Person.phone)
    txtPassword.Text = Trim(Person.Password)
    phone.Caption = "Record " + Str(CurrentRecord) + "/" + Str(LastRecord)
End Sub

Sub cmdDelete_Click ()
    Dim DirResult
    Dim TmpFileNum
    Dim TmpPerson As PersonInfo
    Dim RecNum As Long
    Dim TmpRecNum As Long
    If MsgBox("Delete this record?", 4) <> 6 Then
       txtName.SetFocus
       Exit Sub
    End If
    If Dir("PHONE.TMP") = "PHONE.TMP" Then
       Kill "PHONE.TMP"
    End If
    TmpFileNum = FreeFile
    Open "PHONE.TMP" For Random As TmpFileNum Len = RecordLen
    RecNum = 1
    TmpRecNum = 1
    Do While RecNum < LastRecord + 1
       If RecNum <> CurrentRecord Then
          Get #FileNum, RecNum, TmpPerson
          Put #TmpFileNum, TmpRecNum, TmpPerson
          TmpRecNum = TmpRecNum + 1
       End If
    RecNum = RecNum + 1
    Loop
    Close FileNum
    Kill "PHONE.DAT"
    Close TmpFileNum
    Name "PHONE.TMP" As "PHONE.DAT"
    FileNum = FreeFile
    Open "PHONE.DAT" For Random As FileNum Len = RecordLen
    LastRecord = LastRecord - 1
    If LastRecord = 0 Then LastRecord = 1
    If CurrentRecord > LastRecord Then
       CurrentRecord = LastRecord
```

เอกสารนี้เป็นเอกสารสงวนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
    ShowCurrentRecord
    txtName.SetFocus
End Sub

Sub cmdExit_Click ()
    SaveCurrentRecord
    End
End Sub
```

PHONE.FRM - 2

```
Sub cmdNew_Click ()
    SaveCurrentRecord
    LastRecord = LastRecord + 1
    Person.Name = " "
    Person.Password = " "
    Person.phone = " "
    Put #FileNum, LastRecord, Person
    CurrentRecord = LastRecord
    ShowCurrentRecord
    txtName.SetFocus
End Sub


Sub cmdNext_Click ()
    If CurrentRecord = LastRecord Then
       Beep
       MsgBox "Begining of file encountered!", 48
    Else
       SaveCurrentRecord
       CurrentRecord = CurrentRecord + 1
       ShowCurrentRecord
    End If
    txtName.SetFocus
End Sub


Sub cmdPrevious_Click ()
    If CurrentRecord = 1 Then
       Beep
       MsgBox "Begining of file encountered!", 48
    Else
       SaveCurrentRecord
       CurrentRecord = CurrentRecord - 1
       ShowCurrentRecord
    End If
    txtName.SetFocus
End Sub


Sub cmdSearch_Click ()
    Dim PhoneToSearch As String
    Dim found As Integer
    Dim RecNum As Long
    Dim TmpPerson As PersonInfo
    PhoneToSearch = InputBox("Enter Phone Number to search :", "Search")
    If PhoneToSearch = "" Then
       txtPhone.SetFocus
       Exit Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
    PhoneToSearch = UCase(PhoneToSearch)
    found = False
    For RecNum = 1 To LastRecord
        Get #FileNum, RecNum, TmpPerson
            If PhoneToSearch = UCase(Trim(TmpPerson.phone)) Then
               found = True
               Exit For
            End If
    Next
    If found = True Then
       SaveCurrentRecord
       CurrentRecord = RecNum
       ShowCurrentRecord
    Else
       MsgBox " Phone Number : " + PhoneToSearch + " not found! "
    End If
PHONE.FRM - 3

    txtName.SetFocus
End Sub

Sub Form_Load ()
    RecordLen = Len(Person)
    FileNum = FreeFile
    Open "PHONE.DAT" For Random As FileNum Len = RecordLen
    CurrentRecord = 1
    LastRecord = FileLen("PHONE.DAT") / RecordLen
    If LastRecord = 0 Then
       LastRecord = 1
    End If
    ShowCurrentRecord
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

โปรแกรมภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
#include <dos.h>
#include <bios.h>
#include <graphics.h>
#include <math.h>
#include <time.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <fcntl.h>
#include <dir.h>
#include <errno.h>
#include <process.h>
#include <conio.h>
#include <sbc.h>
#include <sbcvoice.h>
#include "loaddrv.c"
#define  DEL          1
#define  Ring_Chk     1
#define  Code         3  /* real is 3 */
#define  Code_P       4  /* old is 3 */
#define  background   7
#define  PLAY         11
#define  REC          12
#define  Pass         1
#define  FileNamax    8
#define  PassWordMax        6
#define CF 1  /* carry flag */
void Initial_graph(void);
void PrintEg(int x,int y,int color,char *ch,int size);
void Check_Ring(void);
void Check_Dtmf(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Gen_Dtmf(void);
void Time_Delay(int sec);
void Voice_file(char *Message,int file);
void PlayFile(char *szFilename);
void RecordFile(char *szFilename);
void Scan_Dtmf(void);
void ShowError(void);
void Scan_Password(void);
void Play_voice_file(void);
int  Check_Password(char * UID, char * passwd);
void add_series(char *C_Dtmf);
int  Check_Phone(char *PHONE);
void Sort_File(char *Directory);
void Search_lastfile(void);
int Change_Dir(char *Directory);
int delete_file(char far *filename);
int Convert_Dtmf(int dtmf);
void Initial_voice();
long int Check_size(void);
struct ffblk blk;
  struct code {
         char f_name[20];
  }c_name[100];
char *moo;
int size_name,lek,lastcount;
char buffer[80],filename[10],lastfile[10];
float start,end,rel;
unsigned char data;
unsigned char DTMF;
char Name[100];
char *sigl;
char root[10];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int  Key;
int  ring,wit;
int  Flag,CodePass,PhonePass;
int  Number[Code];
int  Password[Code_P];
int  Operator[Code] = {1,0,2};
extern  char far * near ctvdsk_drv;

main()
  {
        int wait_st,wait_en,wait_rel,yot,count;
        int i,Code_Dtmf=0,Code_Pass=0,chkpath;
        char C_Dtmf[FileNamax], C_Pass[PassWordMax];
        char Origin[10];
        clrscr();
        Initial_voice();
loop:   Initial_graph();
        setfillstyle(1,background);
    bar(0,0,getmaxx(),getmaxy());
    PrintEg(getmaxx()/2,50,13,"Operator Automatic",4);
    outportb(0x0303,0x0090);
    outportb(0x0302,0x0000);
    PrintEg(100,100,13,"Signal Ring :",3);
lp:  while(!kbhit())
       {
    data=(inportb(0x0300)&0x020);
        if(data==0x0000)
          PrintEg(200,100,13,"OFF",3);
        ring=0;
        if(data==0x0020)
         {
           setfillstyle(1,background);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            bar(170,80,230,120);      ring=1;
            Check_Ring();
          }
        if(ring!=Ring_Chk) goto lp;
        else goto Answer;
      } Key=bioskey(0);
            if(Key==0x011B) goto quit;
Answer: outportb(0x0302,0x0008);
            Time_Delay(0.5);
            closegraph();
            clrscr();
              wait_rel=0;
              wait_st = clock();
            do
            {
             printf("Welcome.voc \n");
             Voice_file("WELCOME.VOC",PLAY);
             Check_Dtmf();
             wait_en = clock();
             wait_rel= (wait_en-wait_st)/CLK_TCK;
              if(wait_rel>40) goto loop;
            }
           while(wit==0);
          if( Flag==0 || Flag==3)
          {
          Code_Dtmf=0;
            for(i=0;i<Code;i++)  /* Convert DTMF --> Integer */
            Code_Dtmf=Number[i]*pow(10,Code-i-1)+Code_Dtmf;
            if(Flag==0)
              printf("Transfer line to Operator \n");
            else if(Flag==3)
              printf("Transfer line to Number %d \n",Code_Dtmf);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gcvt((float)Code_Dtmf,Code,C_Dtmf);  /* Convert to Ascii */
PhonePass = Check_Phone(C_Dtmf);
if(PhonePass == Pass) {
   printf("Wait.voc \n");
   Voice_file("WAIT.VOC",PLAY);
}
if(PhonePass != Pass) {
   printf("Retry.voc \n");
   Voice_file("RETRY.VOC",PLAY);
   goto Answer;
}
outportb(0x0302,0x0000);
Time_Delay(1);
Gen_Dtmf();
closegraph();
goto loop;
} else
if(Flag==1)  /*____________Inquiry____________*/
{
Check_Dtmf();
Code_Dtmf=0;
for(i=0;i<Code;i++)  /* Convert DTMF --> Integer */
   Code_Dtmf=Number[i]*pow(10,Code-i-1)+Code_Dtmf;
gcvt((float)Code_Dtmf,Code,C_Dtmf);  /* Convert to Ascii */
PhonePass = Check_Phone(C_Dtmf);
if(PhonePass == Pass) {
   Voice_file("WAIT.VOC",PLAY);
}
if(PhonePass != Pass) {
   Voice_file("RETRY.VOC",PLAY);
   goto Answer;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*__________ C_Dtmf is UID _____________ */
    printf("Password.voc \n");
    count=0;
    wit=0;
   do {
     Voice_file("PASSWORD.VOC",PLAY);  /* Please press password */
     Scan_Password();  /* Check Password in file.dat and return value */
     count++;
   } while (wit==0 && count < 2);
   if(wit==0 && count ==2){
     Voice_file("WAIT.VOC",PLAY);
     goto loop; }
  Code_Pass = 0;
  for(i=0; i < Code_P;i++) {
       Code_Pass = Password[i] * pow(10,Code_P-i-1) + Code_Pass;
  }
  C_Pass[0] = '\0';
  itoa (Code_Pass, C_Pass, 10);
/* ________ C_Pass is Password _____________ */
  CodePass = Check_Password(C_Dtmf , C_Pass);
  /* Open file.dat for check Password */
   if(CodePass==Pass)
{
     strcpy(root,"C:\\TC\\");
     strcat(root,C_Dtmf);
     strcpy(filename,"DUMMY.VOC");
  Sort_File(root);
   for (i=0;i<lastcount;i++)
     {
       Voice_file(c_name[i].f_name,PLAY);
       Change_Dir("C:\\tc");
       printf("File name : %s \n",c_name[i].f_name);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Voice_file("DEL_FILE.VOC",PLAY);
                Change_Dir(root);
                   yot = CHK_DEL();
                   if(yot==DEL)
                   {  strcpy(filename,c_name[i].f_name);
                    printf("File name is deleted: %s \n",c_name[i].f_name);
                      Sort_File(root);  i--;
                    }
              }
 } else {
              printf("CodeFail.voc \n");
              Voice_file("CODEFAIL.VOC",PLAY);
              goto Answer; }

              /* check empty file */
              if (lastcount==0) {
                Change_Dir("C:\\tc");
                Voice_file("EMPTY.VOC",PLAY);
              }
              Change_Dir("C:\\tc");
              goto loop;
}             /* end of flag = 1 */

/*_____________________________________________________________________*/
      if(Flag==2) /*_____________Record Voice Format_____________*/
        {
            Check_Dtmf();
                Voice_file("WAIT.VOC",PLAY); /* real is file Wrecord.voc */
                Code_Dtmf=0;
                for(i=0;i<=Code;i++)
                      Code_Dtmf=Number[i]*pow(10,Code-i-1)+Code_Dtmf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            gcvt((float)Code_Dtmf,Code,C_Dtmf); /* Convert Code to Ascii */
        PhonePass = Check_Phone(C_Dtmf);
        if(PhonePass == Pass) {
            strcpy(root,"C:\\TC\\");
            strcat(root,C_Dtmf);
            chkpath=chdir(root);
            if (chkpath != 0) {
                chdir("C:\\TC");
                mkdir(C_Dtmf);
            }
         }
        if(PhonePass != Pass) {
            Voice_file("RETRY.VOC",PLAY);
            goto Answer;
        }
                /* Gen Tone */
                outportb(0x0302,0x000B);
                outportb(0x0301,0x0DB);
                Time_Delay(1);
                outportb(0x0301,0x0FF);
                strcpy(root,"C:\\TC\\");
                strcat(root,C_Dtmf);
                Change_Dir(root);
                Search_lastfile();
                if (lastcount == 0) {
                   strcpy(Origin,C_Dtmf);
                   strcat(Origin,"00.VOC");
                   strcpy(lastfile,Origin);
                  }
                printf("File Lastfile : %s\n",lastfile);
                start=clock(); /* For Check is Record in 30 sec */
                Voice_file(lastfile,REC);  /* RECORD SOUND INTO HARDDISK*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Change_Dir("C:\\TC");
                    Voice_file("THANKYOU.VOC",PLAY); /* Final in Record */
        closegraph();
        goto Answer;  /* real goto Answer then repeat */
            } /* end of flag = 2 */
quit:   outportb(0x302,0x0000);
    closegraph();
}

void Initial_graph()
{  int GraphDriver  = DETECT;
   int ErrorCode =0;
   int GraphMode;
   initgraph(&GraphDriver,&GraphMode,"");
   ErrorCode = graphresult();
   if (ErrorCode != grOk) exit(1);
}

void PrintEg(int x,int y,int color,char *ch,int size)
{  int a,b,c,d;
   if(size>4)size=4;
   for(b=1;b<=3;b++)
   {  if(b==1){a=1;c=1;}
      if(b==2){a=15;c=3;}
      if(b==3){a=color;c=2;}
      setcolor(a);
   settextjustify(1,1);
   settextstyle(1,0,size);
   outtextxy(x-c,y-c,ch);
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Check_Ring()
{
    double signal[4];
        do
          {
        PrintEg(200,100,13,"ON",3);
              start=clock();
           do
              {
                end=clock();
                  rel=(end-start)/CLK_TCK;
                data=(inportb(0x0300)&0x20);
              }   while(data==0x0020);
              signal[0]=rel;
                start=clock();
          do    {
                end=clock();
                 rel=(end-start)/CLK_TCK;
                data=(inportb(0x0300)&0x020);
              }   while(data==0x0000);
               signal[2]=rel;
              if(signal[1]>=(signal[0]-0.2)&&signal[3]>=(signal[2]-0.2))
                      ring++;   else;
               signal[1]=signal[0]; signal[3]=signal[2];
              if(ring>0)
               {
         gcvt(ring,3,sigl);
         setfillstyle(1,background);
         bar(280,80,330,120);
         PrintEg(300,100,13,sigl,3);
         PrintEg(400,100,13,"circle",3);
               }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }    while(ring<Ring_Chk);
        PrintEg(getmaxx()/2,300,13,"ring",4);
        PrintEg(getmaxx()/2,400,13,"OK",4);
}
void Check_Dtmf()
{
  int i=0;
  outportb(0x302,0x000B);
  do
    {       start=clock();
            do
             {
               DTMF=(inportb(0x0300)&0x0010);
            end =clock();  rel=(end-start)/CLK_TCK;
               if(rel>10) {
                     Voice_file("TIMER.VOC",PLAY); /* WARNING IN TIME */
                       wit=0; i=3;  goto wit;    /* wit=0 COME BACK */
                     }   wit=1;     /* wit=1 pass DTMF */
             }
            while(DTMF==0x0000);
            do
             {
                  DTMF=(inportb(0x0300)&0x0010);
             }
            while(DTMF==0x0010);
            do
             {
               DTMF = (inportb(0x0300)&0x000F);
               DTMF = Convert_Dtmf(DTMF);
             }
            while(DTMF>0x000C);
            if(DTMF==0x000A)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DTMF=0;
wit:        if(DTMF!=0x000C&&DTMF!=0x000B)
            {
              Number[i]=(int)DTMF;
              if (Number[0]==0 && Number[0] !=12 && Number[0] !=11)
                  {
                    for(i=0;i<3;i++)
                      Number[i] = Operator[i];
                    Flag=0;
                  }
              else {
                   Flag=3; }
              i++;
            }
          else  /* printf(":  %d  :",(int)DTMF); */
          if(DTMF==0x000C)
            {
             Number[0]=(int)DTMF;
             Voice_file("INQUIRY.VOC",PLAY);
             i=3;
             Flag=1;
            }
          if(DTMF==0x000B)
            {
             Number[0]=(int)DTMF;
                  Voice_file("RECORD.VOC",PLAY);
                  i=3;
            Flag=2;
            }
        }  while(i<Code);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Gen_Dtmf()
{   int i,Tone;
     outportb(0x0302,0x000B);
   for(i=0;i<Code;i++)
  {  Time_Delay(1);
     switch(Number[i])
          {
            case 1: Tone=0x0EE;     break;
            case 2: Tone=0x0ED;     break;
            case 3: Tone=0x0EB;     break;
            case 4: Tone=0x0DE;     break;
            case 5: Tone=0x0DD;     break;
            case 6: Tone=0x0DB;     break;
            case 7: Tone=0x0BE;     break;
            case 8: Tone=0x0BD;     break;
            case 9: Tone=0x0BB;     break;
            case 0: Tone=0x07D;     break;
            default: break;
          }
    outportb(0x0301,Tone);
    Time_Delay(0.5);
    outportb(0x0301,0x00FF);
  }
    outportb(0x0301,0x00FF);
}

void Scan_Password()
{  int i=0;
     outportb(0x302,0x000B);
   do
     {
       start=clock();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            do
             {
               DTMF=(inportb(0x0300)&0x0010);
            end =clock();  rel=(end-start)/CLK_TCK;
                if(rel>10) {
                      Voice_file("TIMER.VOC",PLAY);  /* WARNING IN TIME */
                      wit=0;
                      i=3;
                      goto wit;
                 } wit=1;
             }
            while(DTMF==0x0000);
            do
             {
                   DTMF=(inportb(0x0300)&0x0010);
             }
            while(DTMF==0x0010);
            do
             {
               DTMF = (inportb(0x0300)&0x000F);
               DTMF = Convert_Dtmf(DTMF);
             }
               while(DTMF>0x000A);
wit:        if(DTMF==0x000A) DTMF=0;
           Password[i]=(int)DTMF;
               i++;
       } while(i<Code_P);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Time_Delay(int sec)
{ start=clock();
   do
   {    end=clock();
            rel=(end-start)/CLK_TCK;
    }   while(rel<=sec);
}

void Initial_voice()
{
     if ( ! GetEnvSetting() )
      {
         if (sbc_check_card() & 4)
            {
              if (sbc_test_int())
                  {
                   if (sbc_test_dma() >= 0)
                    {

                   if ((ctvdsk_drv = LoadDriver("CTVDSK.DRV")) != 0 ) {
                     if ( !ctvd_init(16) )
                           ;
                     else {
                          ShowError();
                          closegraph();
                          exit(1);
                     }
                  }
                  else {
                     ShowError();
                     closegraph();
                     exit(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
            }
            else {
                printf("Error on DMA channel.\n");
                exit (1); }
        }
        else {
            printf("Error on interrupt.\n");
            exit(1);
        }
    }
    else {
        printf("Sound Blaster Card not found or wrong I/O settings.\n") ;
        exit(1);
    }
}
else {
    printf("BLASTER environment not set or incomplete or invalid.\n");
    exit(1);
}
}

void Voice_file(char *Message,int file)
{
    outportb(0x0302,0x001F);
        if(file==PLAY)
            PlayFile(Message);
        else if(file==REC)
            RecordFile(Message);
        else
            ctvd_terminate() ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void PlayFile(char *szFilename)
{
    int     Handle ;

    /* Open and play the voice file */
    if'( (Handle=_open(szFilename, O_RDONLY)) != -1 )
    {
        ctvd_speaker(1) ;
        if ( ctvd_output(Handle) == NO_ERROR )
        {
            Scan_Dtmf() ;
        }
        else
                ShowError () ;
        _close (Handle) ;
    }
    else
        printf ( "Open %s file error\n", szFilename ) ;
}

void RecordFile(char *szFilename)
{
    int     Handle ;
    /* create and record a voice file */
    if ( (Handle=_creat(szFilename, 0)) != -1 )
    {
        ctvd_speaker(0) ;
        if ( ctvd_input(Handle, 8000) == NO_ERROR )
        {
            Scan_Dtmf() ;
           if ( ctvd_drv_error() )
              ShowError() ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            else
                printf ("Voice record ended.\n") ;
        }
        else
            ShowError () ;
        _close (Handle) ;
    }
    else
        printf ("Create %s file error\n", szFilename ) ;
}

#pragma lp_opt(off)      /* turn off loop optimiaztion */

void Scan_Dtmf()
{
        while ( ct_voice_status )
        {  end=clock();
           rel= (end-start)/CLK_TCK;
           if((inportb(0x0300)&0x010)==0x0010||rel==25) /* time for record */
              ctvd_stop();
        }
}
#pragma lp_opt()

void ShowError()
{
    int     Err ;

    /* Show the driver error and the DOS extended error code */
    Err = ctvd_drv_error() ;
    printf("Driver error = %2d\n", Err) ;
    Err = ctvd_ext_error();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if ( Err != 0 )
        printf ("DOS error = %2d\n", Err) ;
}

int  Check_Phone(char * PHONE)
 {
  struct data {
        char name[30];
        char phone[4];
        char passwrd[5];
    }var;
    FILE *fp;
    char s_code[4];
    if((fp=fopen("PHONE.DAT","rb")) == NULL ) {
        printf(" Error in open file \n");
        exit(1); }
      strcpy(s_code,PHONE);
      while((fread(&var,sizeof(var),1,fp)== 1)) {
            var.phone[3] = '\0';
            if (!strcmp(s_code,var.phone))
              {  CodePass=1;
                 break;  }
            else CodePass=0;
            if(ferror(fp)) {
                  printf("Error in file \n");
                  exit(1);
            }
      }
      rewind(fp);
    fclose(fp);
    return( CodePass);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int Check_Password(char * UID, char * passwd)
{
 struct data {
        char name[30];
        char phone[4];
        char passwrd[5];
   }var;
   FILE *fp;
   char s_code[4],s_pass[5],ch;
   if((fp=fopen("PHONE.DAT","rb")) == NULL ) {
        printf(" Error in open file \n");
        exit(1);
    }
      strcpy(s_code,UID);
      strcpy(s_pass,passwd);
      while((fread(&var,sizeof(var),1,fp)== 1)) {
            var.phone[3] = '\0';
            if (!strcmp(s_code,var.phone))
                break;
            if(ferror(fp)) {
                printf("Error in file \n");
                exit(1);
            }
      }
      var.passwrd[4] = '\0';
      var.phone[3] = '\0';
      var.name[29] = '\0';
      if(!strcmp(s_pass,var.passwrd))
        CodePass=1;
       else
        CodePass=0;
      rewind(fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fclose(fp);
    return CodePass;
  }

void Sort_File(char *Directory)
{
 int done,i,j,err;
    Change_Dir(Directory);
     done = findfirst("*.*",&blk,0);
     lastcount=0;
     while (!done) {
           lastcount++;
      done=findnext(&blk);
     }
   done = findfirst("*.*",&blk,0);
   while (!done) {
     for(i=0;i<lastcount;i++)
        {  strcpy(c_name[i].f_name,blk.ff_name);
             done=findnext(&blk);   }
   }
            /* Sort File Name */
        for(size_name=10;size_name>=0;size_name--)
        for(j=0;j<lastcount;j++)
         for(i=0;i<lastcount;i++)
while(c_name[i].f_name[size_name]>c_name[i+1].f_name[size_name]&&i!=(lastcount-1))
        { strcpy(moo,c_name[i].f_name);
          strcpy(c_name[i].f_name,c_name[i+1].f_name);
          strcpy(c_name[i+1].f_name,moo);
        }
          err = delete_file(filename);
          if (err) {
             printf("Not Able to delete %s \n",filename);  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        strcpy(moo,filename);
      for(i=0;i<lastcount;i++)
          {  err=strcmp(moo,c_name[i].f_name);
               if(!err) lek=i;
          }
        for(i=lek;i<lastcount;i++)
       rename(c_name[i+1].f_name,c_name[i].f_name);
      done = findfirst("*.*",&blk,0);
      lastcount=0;
      while (!done) {
            lastcount++;
       done=findnext(&blk);
     }
  }

void Search_lastfile()
{   int done,i,j,err;
    int n01,n10;
    done = findfirst("*.*",&blk,0);
    lastcount = 0;
    while (!done)     {
      lastcount++;
      done=findnext(&blk);
    }
    done = findfirst("*.*",&blk,0);
    while (!done) {
      for(i=0;i<lastcount;i++)
        {  strcpy(c_name[i].f_name,blk.ff_name);
             done=findnext(&blk);   }
    }
            /* Sort File Name */
        for(size_name=10;size_name>=0;size_name--)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            for(j=0;j<lastcount;j++)
             for(i=0;i<lastcount;i++)
while(c_name[i].f_name[size_name]>c_name[i+1].f_name[size_name]&&i!=(lastcount-1))
            { strcpy(moo,c_name[i].f_name);
              strcpy(c_name[i].f_name,c_name[i+1].f_name);
              strcpy(c_name[i+1].f_name,moo);
            } /* lastcount > 99 */
           if(lastcount>99)
    { for(i=0;i<20;i++)
      delete_file(c_name[i].f_name);
     for(i=0;i<lastcount-20;i++)
       rename(c_name[i+20].f_name,c_name[i].f_name);
          lastcount = lastcount-20;   }
    n01=lastcount % 10;   n10=(lastcount-n01)/10;
           c_name[lastcount-1].f_name[4] = 0x30+n01;
       ;    if(lastcount>9)
           c_name[lastcount-1].f_name[3] = 0x30+n10;
   strcpy(lastfile,c_name[lastcount-1].f_name);
}

int Change_Dir(char *Directory)
    {
      int test;
      test = bdosptr(0x3B,Directory,0);
      return(test);
    }

/* _______ Function Delete file _______ */
int delete_file(char far *filename)
{
  union REGS regs;
  struct SREGS sregs;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  int ret;
  regs.h.ah = 0x41;
  regs.x.dx = FP_OFF(filename);
  sregs.ds = FP_SEG(filename);
  ret = intdosx(&regs,&regs,&sregs);
  return(regs.x.cflag ? ret :0);
}

long int Check_size()
{
  struct dfree disk;
  long int dsize;
  getdfree(0,&disk);
  dsize = (long)disk.df_avail* (long)disk.df_bsec * (long)disk.df_sclus;
  return(dsize);
}

int CHK_DEL()
{
  int i=0;
  outportb(0x302,0x000B);
          start=clock();
         do
          {
            DTMF=(inportb(0x0300)&0x0010);
           end =clock();  rel=(end-start)/CLK_TCK;
             if(rel>4) { i=0; goto YOT; }
          }
         while(DTMF==0x0000);
         do
          {
              DTMF=(inportb(0x0300)&0x0010);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            while(DTMF==0x0010);  i=1;
    YOT:  return(i);
}

int Convert_Dtmf(int dtmf)  /* DTMF = Convert_Dtmf(DTMF)  */
 { int conv;
  switch(dtmf)
   { case 0x08: conv=0x01;  break;
     case 0x04: conv=0x02;  break;
     case 0x0C: conv=0x03;  break;
     case 0x02: conv=0x04;  break;
     case 0x0A: conv=0x05;  break;
     case 0x06: conv=0x06;  break;
     case 0x0E: conv=0x07;  break;
     case 0x01: conv=0x08;  break;
     case 0x09: conv=0x09;  break;
     case 0x05: conv=0x0A;  break;
     case 0x0D: conv=0x0B;  break;
     case 0x03: conv=0x0C;  break;
   }
     return(conv);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์

ตัวต้านทาน 1/4 วัตต์ 1%

R2	5.1 k	1 ตัว
R5	10 k	1 ตัว
R3 , R4 , R6 , R8 , R14 , R15	100 k	6 ตัว
R7	1 M	1 ตัว
R11	1.29 k	1 ตัว
R12	22 k	1 ตัว
R13	300 k	1 ตัว

ตัวต้านทาน 1 วัตต์ 5%

R1	10 k	1 ตัว
R9 , R10	22	2 ตัว
RPACK 1	100 k x 8	1 ตัว
RPACK 2 , RPACK 3	10 k x 8	2 ตัว

ตัวเก็บประจุ

C1 , C2 , C3 , C7 , C8	10 nF	5 ตัว
C4 , C9	1 uF	2 ตัว
C5	33 uF	1 ตัว
C6	10 uF	1 ตัว
C10 , C11	4.7 uF	2 ตัว
C12 , C13 , C14 , C15 , C16	1 uF	5 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์

อุปกรณ์เซมิคอนดัคเตอร์

IC 1 8255	1 ตัว
IC 2 74HCT688	1 ตัว
IC 3 74LS138	1 ตัว
IC 4 74LS21	1 ตัว
IC 5 74HCT04	1 ตัว
IC 6 MT8870	1 ตัว
IC 7 LR4089	1 ตัว
Q 1 D789	1 ตัว

อุปกรณ์อื่น ๆ

ARRESTOR ARR1 , ARR2 , ARR 3	3 ตัว
VARISTOR VR1,VR2 100V	2 ตัว
VARISTOR VR3 250 V	2 ตัว
COIL FILTER	1 ตัว
OUTPUT TRANSFORMER	2 ตัว
RELAY 2 CONTACT 5V	1 ตัว
DIODE 1N4148	1 ตัว
DIP SWITCH 8 SW	1 ตัว
X-TAL 3.579 MHz	2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำขอขอบพระคุณต่อบิดา มารดาของคณะผู้จัดทำที่ได้ให้กำลังใจและทุนทรัพย์ในการจัดทำโครงงานและอาจารย์สมยศ จุณณะปิยะ ที่ได้กรุณาให้คำแนะนำเป็นที่ปรึกษาในการทำโครงงาน รวมทั้งคุณมรกต เธียรมนตรี เจ้าหน้าที่องค์การโทรศัพท์แห่งประเทศไทย ที่ได้กรุณาให้คำปรึกษาทางด้านเทคนิคและเพื่อนของคณะผู้จัดทำคุณพิสุทธิ์ ทัศนอนันชัย และคุณสนิท แสงเหลา ที่ได้ให้คำแนะนำปรึกษาในการจัดทำซอฟต์แวร์เพื่อควบคุมการทำงานของโปรแกรมทำให้โครงงานสำเร็จลุล่วงด้วยดี ทางคณะผู้จัดทำจึงใคร่ขอขอบพระคุณต่อคณะบุคคลทุกท่านที่ได้กล่าวนาม และมิได้กล่าวนามไว้ ณ.ที่นี้ด้วย

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ปราโมทย์ วาดเขียน และ วิวัฒน์ กิรานนท์ "พื้นฐานการสื่อสารข้อมูล" กรุงเทพฯ : สำนักพิมพ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2536

[2] ขืน ภู่วรวรรณ "ทฤษฎีและการประยุกต์ไมโครโปรเซสเซอร์ Z-80" กรุงเทพฯ : บริษัท ซีเอ็ดยูเคชั่น จำกัด , 2533

[3] สานนท์ แก้วอบเชย. "สล็อตไอบีเอ็ม พีซี กับการต่อออก" เซมิคอนดัคเตอร์ อิเลคทรอนิกส์ 87 , (2531) : 186-196

[4] มนตรี พจนารถลาวัณย์. "การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี" กรุงเทพฯ : บริษัท ซีเอ็ดยูเคชั่น จำกัด

[5] "IC น่าสน"เซมิคอนดัคเตอร์อิเล็คทรอนิกส์ 88, (2531) : 210-214

[6] Wang Wallace "VISUAL BASCI 3 FOR DUMMIES" : IDG Books WorldWide,1994.

[7] Sound Blaster User Reference Manual , Creative labs,1991

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้