ชุดฝึกพิมพ์สำหรับคนตาบอด

TYPING TRAINING SET FOR THE BLINDS

ชมคิด จารุกิจวิกล์กุล
SOMKID JARUKITWIKAIKOL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2544
ISBN 974-608-344-7

ชุดฝึกพิมพ์สำหรับคนตาบอด

TYPING TRAINING SET FOR THE BLINDS

สมคิด จรัสกิจวิกัยกุล

SOMKID JARUSKITWIKAIKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2544

ISBN 974-648-344-7

TYPING TRAINING SET FOR THE BLINDS

SOMKID JARUSKITWIKAIKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

ISBN 974-648-344-7

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ ชุดฝึกพิมพ์สำหรับคนตาบอด

TYPING TRAINING SET FOR THE BLINDS

ชื่อนักศึกษา นายสมคิด จรัสกิจวิภัยกุล

รหัสประจำตัว 39061039

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.พลผดุง ผดุงกุล

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.ปิติเขต	สู้รักษา	
ผศ.พิชัย	ดูศิริวานิชกร	
ผศ.ขนิษฐา	แซ่ตั้ง	
ผศ.ประภากร	สุวรรณะ	
ผศ.พลผดุง	ผดุงกุล	

วัน/เดือนปี ที่สอบ 21 พฤษภาคม 2544 เวลา 12.00-13.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-402)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.ปัญญัติ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่ 25 **เดือน** พฤษภาคม **พ.ศ.** 2544

หัวข้อวิทยานิพนธ์	ชุดฝึกพิมพ์สำหรับคนตาบอด
นักศึกษา	นายสมดิด จรัสกิจวิกัยกุล
รหัสประจำตัว	39061039
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2544
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.พลผดุง ผดุงกุล

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอชุดฝึกพิมพ์สำหรับคนตาบอด โดยจะประกอบด้วย 2 ส่วนที่สำคัญ คือ ชุดของการ์ดบันทึกเสียงโดยเชื่อมต่อกับคอมพิวเตอร์ทางสล๊อตขยายระบบพร้อมทั้งโปรแกรมที่ใช้ในการบันทึกเสียงที่สามารถปรับปรุงคุณภาพของเสียงที่ต้องการบันทึก ตัดต่อ และยังสามารถบันทึกให้อยู่ในรูปของไฟล์ได้ และส่วนของชุดฝึกพิมพ์ที่ใช้เชื่อมต่อกับคีย์บอร์ดทั่วไปที่ใช้กับเครื่องคอมพิวเตอร์ โดยการใช้งานไม่จำเป็นที่จะต้องเชื่อมต่อกับคอมพิวเตอร์ ซึ่งในส่วนของชุดฝึกพิมพ์นี้จะประกอบด้วยชุดฝึกพิมพ์ทั้งภาษาไทยและภาษาอังกฤษและยังมีแบบฝึกหัดที่ใช้ฝึกฝนการพิมพ์ให้ชำนาญยิ่งขึ้นด้วย พร้อมทั้งสามารถประเมินผลการฝึกพิมพ์และบอกผลการประเมินในรูปของเสียงได้อีกด้วย อีกทั้งผู้ใช้ยังสามารถทำการแก้ไขแบบฝึกหัดในแต่ละส่วนได้ด้วยตนเองโดยโปรแกรมบนคอมพิวเตอร์

Thesis Title	Typing Training Set for the Blinds
Student	Mr.Somkid Jaruskitwikaikul
Student ID	39061039
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2001
Thesis Advisor	Asst.Prof.Polphadung Phadungkul

ABSTRACT

This thesis proposes a Typing Training Set for the blinds, which consists of two main parts. First part is a sound recording card that is connected to a PC slot and a recording program on the PC which can realign, reform the recording sound. Furthermore we can save it as a file on the PC. Another part is a typing training set. It can be used by connecting to the general PC keyboard without using the computer. This typing training set can be used to train both in Thai and English languages, including exercises for typing skill. Moreover, the trainee can be evaluated by the training set and reported out by sound. Additionally, the trainer can edit the exercises by computer program.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี ด้วยคำแนะนำและคำปรึกษาจาก ผศ.พลผดุง ผดุงกุล ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณครูและอาจารย์ทุกท่านตั้งแต่อดีตจนถึงปัจจุบันที่ได้ประสิทธิประสาทวิชาความรู้ต่างๆต่อผู้วิจัยจนกระทั่งได้ศึกษาสำเร็จมาจนถึงปัจจุบันนี้

ขอบขอบคุณเพื่อนๆ พี่ๆ และน้องๆทุกท่านที่ได้ให้คำแนะนำและช่วยเหลือในการให้ยืมอุปกรณ์ต่างๆในการทดลอง รวมถึงการบันทึกเสียงที่ใช้ในวิทยานิพนธ์ และกำลังใจต่อผู้วิจัยด้วยดีเสมอมา

สุดท้ายนี้ขอขอบพระคุณคุณพ่อและคุณแม่ของผู้วิจัยที่ได้สนับสนุนให้การศึกษาที่ดีและทุนทรัพย์ในการศึกษาตลอดมา

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

สมคิด จรัสกิจวิกัยกุล

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป(ต่อ)

สารบัญรูป(ต่อ)

บทที่ 1
บทนำ

1.1 กล่าวนำ

ในปัจจุบันเทคโนโลยีสมัยใหม่ได้มีบทบาทต่อการดำเนินชีวิตประจำวันมากขึ้น ดังนั้นเราจะหลีกเลี่ยงที่จะเรียนรู้เทคโนโลยีไม่ได้ โดยเฉพาะเทคโนโลยีของคอมพิวเตอร์ที่สามารถใช้ในการศึกษาหาความรู้หรือใช้บริการต่างๆผ่านทางโครงข่ายอินเตอร์เน็ต หรืออาจจะใช้โปรแกรมสำเร็จรูปต่างๆเพื่อที่จะช่วยให้งานเสร็จอย่างรวดเร็วและเป็นระเบียบเรียบร้อยขึ้น ดังนั้นการใช้งานคอมพิวเตอร์จึงเป็นสิ่งที่จำเป็นมากขึ้นทุกวัน โดยแน่นอนว่าการที่จะใช้งานคอมพิวเตอร์สิ่งที่จะขาดมิได้ก็คือคีย์บอร์ดเพื่อใช้พิมพ์สั่งงานคอมพิวเตอร์ ซึ่งสำหรับคนตาปกตินั้นการกดคีย์บอร์ดคงไม่ใช่เรื่องที่ยากอะไรเพราะสามารถมองเห็นได้ว่าตนเองกดคีย์อะไรไปบ้าง แต่สำหรับคนตาบอดแล้วการพิมพ์คีย์บอร์ดนี้คงเป็นเรื่องที่ยากพอสมควร จึงได้คิดสร้างชุดฝึกพิมพ์สำหรับคนตาบอดขึ้นเพื่อที่จะให้คนตาบอดได้ทราบว่าได้พิมพ์คีย์ใด อีกทั้งยังมีแบบฝึกหัดที่ใช้ฝึกฝนทักษะในการพิมพ์สำหรับคนตาบอดและคนตาปกติได้อีกด้วย

ในส่วนที่ 2 เป็นส่วนของการ์ดบันทึกเสียงที่ใช้ MC3418 โดยที่ไอซีที่ใช้ในการบันทึกเสียงโดยเฉพาะในปัจจุบันก็มีมากมายที่สามารถเลือกใช้ได้ตามความเหมาะสม แต่ว่าในบรรดาไอซีเหล่านั้นจะมีส่วนน้อยที่มีชุดพัฒนาการบันทึกเสียงที่สามารถตัดต่อหรือปรับปรุงคุณภาพของเสียงจนเป็นที่พอใจก่อนนำไปใช้จริงได้ ส่วนมากจะเป็นการบันทึกเพียงครั้งเดียวถ้าบันทึกได้ไม่เป็นที่พอใจก็ต้องทำการบันทึกใหม่ทั้งหมด หรือไอซีบางเบอร์อาจมีชุดพัฒนาการบันทึกเสียงแต่ว่ามีราคาที่ค่อนข้างสูงและต้องนำเข้าจากต่างประเทศ จึงได้นำเอาไอซีเบอร์ MC3418 ซึ่งเป็นไอซีที่สามารถหาซื้อได้ง่ายและมีราคาถูกมาใช้ ซึ่งไอซีเบอร์นี้ใช้หลักการ CVSD ในแปลงสัญญาณเสียงเป็นสัญญาณดิจิตอล และนำสัญญาณดิจิตอลนี้ขึ้นไปประมวลผลบนคอมพิวเตอร์ให้ได้สัญญาณตามที่ต้องการก่อนที่จะบันทึกเพื่อที่จะนำไปใช้จริงในภายหลัง

1.2 วัตถุประสงค์ของวิทยานิพนธ์

วิทยานิพนธ์นี้นำเสนองาน 2 ส่วนด้วยกันคือส่วนแรกเป็นชุดฝึกพิมพ์สำหรับคนตาบอดและส่วนที่ 2 คือการ์ดบันทึกเสียงสำหรับ MC3418 สำหรับส่วนของชุดฝึกพิมพ์สำหรับคนตาบอดสร้างขึ้นมาเพื่อจุดประสงค์ให้คนตาบอดมีโอกาสที่จะใช้คอมพิวเตอร์ได้ทัดเทียมกับคนปกติคือมีเสียงมาช่วยบอกคนตาบอดให้รับทราบว่าได้กดคีย์ใดไป และนอกจากนี้ทั้งคนตาบอดและคนปกติยังสามารถใช้ชุดฝึกพิมพ์นี้ในการฝึกฝนทักษะในการพิมพ์ได้อีกด้วย เพราะบนชุดฝึกพิมพ์จะมีแบบฝึกหัดสำหรับใช้ฝึกฝนและทดสอบความสามารถในการพิมพ์ทั้งภาษาไทยและภาษาอังกฤษ

และแบบฝึกหัดเหล่านี้ยังสามารถแก้ไขได้โดยซอฟต์แวร์ที่เขียนขึ้นบนระบบปฏิบัติการไมโครซอฟต์วินโดว์ได้ด้วยตนเองอีกด้วย ทั้งนี้เพื่อความทันสมัยของชุดแบบฝึกหัดและเพื่อความเหมาะสมกับบุคคลที่จะฝึกพิมพ์

ในส่วนที่ 2 คือการ์ดบันทึกเสียงสำหรับ MC3418 สร้างขึ้นมาเพื่อให้เราสามารถได้ใช้งานไอซีเบอร์ MC3418 ได้อย่างสะดวกมากขึ้น กล่าวคือเราสามารถอ่านข้อมูลเสียงที่ได้บันทึกไว้โดยไอซีเบอร์นี้ขึ้นไปยังซอฟต์แวร์บนคอมพิวเตอร์ไว้ก่อนเพื่อทำการประมวลผลต่างๆ เช่นการตัด-ต่อสัญญาณเสียง หรือปรับปรุงคุณภาพของเสียง และบันทึกเป็นไฟล์ข้อมูลบนคอมพิวเตอร์เก็บไว้ เมื่อต้องการนำไปใช้ก็นำไฟล์ที่ได้บันทึกไว้นี้ไปโปรแกรมลงบน EPROM หรือ EEPROM เพื่อที่จะใช้ไมโครโปรเซสเซอร์หรือวงจรดิจิตอลอื่นๆอ่านข้อมูลเสียงใน EPROM หรือ EEPROM นี้ ไปสู่วงจรดีมอดูเลชันที่ใช้ MC3418 ต่อไป ทั้งนี้เราก็สามารถใช้ไอซีเบอร์ MC3418 นี้ทดแทนการใช้งานไอซีเสียงอื่นๆที่มีราคาที่แพงกว่าได้ และสามารถใช้บันทึกเสียงได้ไม่จำกัดเวลาเหมือนไอซีเสียงอื่นๆเพราะว่าได้ใช้หน่วยความจำที่อยู่ภายนอกดังนั้นเวลาที่บันทึกเสียงได้จึงขึ้นอยู่กับขนาดของหน่วยความจำที่ต่อภายนอกนั่นเอง

1.3 รายละเอียดในวิทยานิพนธ์

ในวิทยานิพนธ์นี้ได้แบ่งออกเป็น 5 บท โดยในแต่ละบทมีรายละเอียดโดยสังเขปดังนี้

บทที่ 1 กล่าวถึงรายละเอียดและวัตถุประสงค์ของวิทยานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีบางส่วนที่เกี่ยวข้องกับวิทยานิพนธ์

บทที่ 3 กล่าวถึงการออกแบบทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์รวมถึงวิธีการใช้งานฮาร์ดแวร์และซอฟต์แวร์ที่ออกแบบไว้

บทที่ 4 ผลการทดสอบระบบที่ออกแบบ

บทที่ 5 สรุป วิจารณ์และข้อเสนอแนะแนวทางในการพัฒนาประสิทธิภาพของระบบ

และในส่วนท้ายของวิทยานิพนธ์ยังมีภาคผนวกต่างๆซึ่งมีรายละเอียดดังนี้

ภาคผนวก ก. วงจรรวมทั้งหมดของชุดฝึกพิมพ์และการ์ดบันทึกเสียง

ภาคผนวก ข. โปรแกรมควบคุมการทำงานของการ์ดบันทึกเสียง

ภาคผนวก ค. โปรแกรมควบคุมการทำงานของชุดฝึกพิมพ์

ภาคผนวก ง. โปรแกรมในส่วนของการแก้ไขแบบฝึกหัดบนชุดฝึกพิมพ์

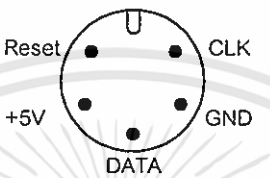
ภาคผนวก จ. การออกแบบวงจรควบคุมของการ์ดบันทึกเสียงที่พัฒนาบน FPGA

ภาคผนวก ฉ. ผลงานที่ได้รับการตีพิมพ์

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 คีย์บอร์ดของ IBM PC

คีย์บอร์ดของ IBM PC จะเชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์โดยผ่านหัวต่อแบบ DIN ดังรูปที่ 2.1



รูปที่ 2.1 หัวต่อแบบ DIN ของคีย์บอร์ด

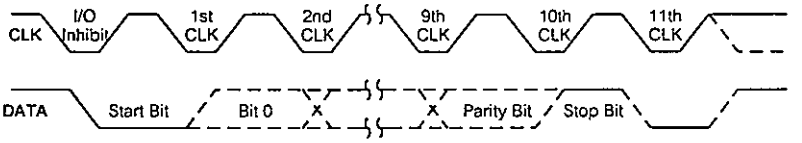
คีย์บอร์ดและคอมพิวเตอร์จะติดต่อสื่อสารกันผ่านทางสายสัญญาณ Clock และ Data โดยต้นทางของแต่ละสายสัญญาณจะเป็นอุปกรณ์แบบ Open collector เพื่อที่จะให้ทั้งคีย์บอร์ดและคอมพิวเตอร์สามารถทำให้สัญญาณเป็น low ได้ เมื่อไม่มีการส่งข้อมูลใดๆสัญญาณ Clock จะเป็น high ส่วนสัญญาณ Data จะถูกดึงเป็น high โดยคีย์บอร์ด สัญญาณ low จะมีค่าแรงดันอยู่ระหว่าง 0 ถึง 0.7V ส่วนสัญญาณ high จะมีค่าแรงดันไม่ต่ำกว่า +2.4V แต่ไม่เกิน +5.5V

ข้อมูลที่จะรับ-ส่งกันระหว่างคีย์บอร์ดและคอมพิวเตอร์ประกอบด้วยข้อมูล 11 บิตส่งแบบอนุกรมผ่านทางสายสัญญาณ Data ซึ่งแต่ละบิตมีความหมายดังตารางที่ 2.1

ตารางที่2.1 ความหมายของสัญญาณบิตต่างๆของคีย์บอร์ด

บิต	ความหมาย
11	Stop bit (always '1')
10	Parity bit (odd parity)
9	Data bit 7 (most significant bit)
8	Data bit 6
7	Data bit 5
6	Data bit 4
5	Data bit 3
4	Data bit 2
3	Data bit 1
2	Data bit 0 (least significant bit)
1	Start bit (always '0')

2.1.1 การส่งข้อมูลจากคอมพิวเตอร์ไปคีย์บอร์ด



รูปที่ 2.2 การส่งสัญญาณจากคอมพิวเตอร์ไปยังคีย์บอร์ด

เมื่อคอมพิวเตอร์พร้อมที่จะส่งข้อมูลไปยังคีย์บอร์ดคอมพิวเตอร์จะทำการตรวจสอบว่าคีย์บอร์ดทำการส่งข้อมูลอยู่หรือไม่ ถ้าคีย์บอร์ดทำการส่งข้อมูลอยู่แต่ว่ายังไม่ถึง Clock ที่ 10 คอมพิวเตอร์สามารถหยุดการส่งข้อมูลได้โดยการทำให้สัญญาณ Clock เป็น low แต่ถ้าคีย์บอร์ดส่งข้อมูลเลยบิตที่ 10 แล้วคอมพิวเตอร์ก็จะทำการรับข้อมูลจนครบ

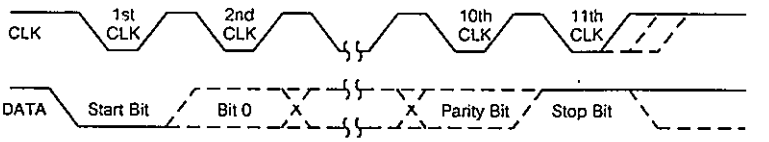
ถ้าคีย์บอร์ดไม่ได้กำลังส่งข้อมูลอยู่ หรือส่งอยู่แต่คอมพิวเตอร์เลือกที่จะหยุดการส่งข้อมูลจากคีย์บอร์ด คอมพิวเตอร์ก็จะทำการทำให้ Clock เป็น low มากกว่า 60μs เพื่อเตรียมที่จะส่งข้อมูล เมื่อระบบพร้อมที่จะส่ง start bit (สัญญาณ data เป็น low) คอมพิวเตอร์จะทำการปล่อยให้สัญญาณ Clock เป็น high ได้

คีย์บอร์ดจะทำการตรวจสอบสถานะของ Clock เป็นระยะๆไม่เกิน 10ms ถ้าพบว่าคอมพิวเตอร์จะส่งข้อมูลคีย์บอร์ดจะทำการรับข้อมูลทั้ง 11 บิต โดยหลังจากบิตที่ 10 คีย์บอร์ดจะทำการตรวจสอบว่าสัญญาณ Data เป็น high หรือไม่ ถ้า สัญญาณ Data เป็น high คีย์บอร์ดจะทำให้เป็น low และนับเพิ่มอีก 1 บิตเป็นการบอกว่าคีย์บอร์ดได้รับข้อมูลครบแล้ว เมื่อคอมพิวเตอร์ทราบว่าคีย์บอร์ดรับข้อมูลเสร็จเรียบร้อยแล้วคอมพิวเตอร์ก็จะเข้าสู่สภาวะเตรียมพร้อม พร้อมที่จะรับส่งข้อมูลต่อไป

แต่ถ้าคีย์บอร์ดพบว่าสัญญาณ Data เป็น low หลังจากบิตที่ 10 แสดงว่าเกิดความผิดพลาดเกิดขึ้น คีย์บอร์ดก็จะทำการรอจนกระทั่งสัญญาณ Data เป็น high หลังจากนั้นคีย์บอร์ดก็จะทำให้สัญญาณ Data เป็น low เพื่อส่งคำสั่งบอกให้คอมพิวเตอร์ส่งข้อมูลนั้นใหม่อีกครั้ง

ในการส่งคำสั่งหรือข้อมูลไปยังคีย์บอร์ด คอมพิวเตอร์ต้องการการตอบสนองจากคีย์บอร์ดก่อนที่คอมพิวเตอร์จะส่งข้อมูลต่อไปไปยังคีย์บอร์ดได้ โดยคีย์บอร์ดจะทำการตอบสนองภายในเวลา 20ms (นอกจากว่าคอมพิวเตอร์ป้องกันมิให้คีย์บอร์ดส่งข้อมูลคือให้ Clockเป็น low) ถ้าเกิดว่าการตอบสนองของคีย์บอร์ดไม่ถูกต้องหรือเกิดมีความผิดพลาดของ parity ขึ้นคอมพิวเตอร์ก็จะทำการส่งข้อมูลไปใหม่อีกครั้ง

2.1.2 การส่งข้อมูลจากคีย์บอร์ดไปคอมพิวเตอร์



รูปที่ 2.3 การส่งสัญญาณจากคีย์บอร์ดไปยังคอมพิวเตอร์

เมื่อคีย์บอร์ดพร้อมที่จะส่งข้อมูลคีย์บอร์ดจะทำการตรวจสอบสัญญาณ Clock และ Data ถ้าสัญญาณ Clock เป็น low ข้อมูลจะถูกเก็บไว้ในคีย์บอร์ดบัฟเฟอร์ก่อน(สัญญาณ Clock เป็น low แสดงว่าคอมพิวเตอร์ยังไม่พร้อมที่จะรับข้อมูล) ถ้าสัญญาณ Clock เป็น high แต่สัญญาณ Data เป็น low (หมายถึงคอมพิวเตอร์จะส่งข้อมูล) ข้อมูลจะถูกเก็บไว้ในคีย์บอร์ดบัฟเฟอร์ก่อน จากนั้นคีย์บอร์ดก็จะทำการรับข้อมูลจากคอมพิวเตอร์

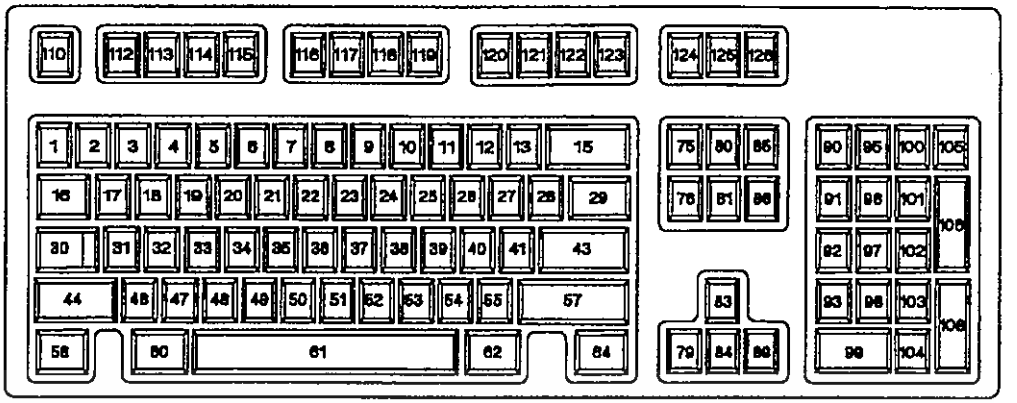
ถ้าสัญญาณ Clock และ Data เป็น high ทั้งคู่คีย์บอร์ดก็จะทำการส่งข้อมูลทั้ง 11 บิตคือ start bit ,8 bits data ,parity bit และ stop bit โดยที่ข้อมูลแต่ละบิตจะถูกต้องก่อน Clock ที่ขอบขาลงจนเลย Clock ที่ขอบขาขึ้น ในระหว่างการส่งข้อมูลแต่ละบิตคีย์บอร์ดจะทำการตรวจสอบสัญญาณ Clock ให้เป็น high อย่างน้อย 60ms ถึงจะทำการส่งข้อมูลบิตถัดไปแต่ถ้าสัญญาณ Clock ถูกทำให้เป็น low โดยคอมพิวเตอร์ในระหว่างการส่งข้อมูล สภาวะนี้จะเรียกว่ามีการแย่งใช้สายสัญญาณเกิดขึ้น(Line contention) คีย์บอร์ดจะหยุดการส่งข้อมูล ถ้าเกิดการแย่งใช้สายสัญญาณขึ้นก่อนขอบขาขึ้นของ Clock ที่ 10 คีย์บอร์ดจะทำการปล่อยให้สัญญาณ Clock และ Data เป็น high (คีย์บอร์ดหยุดส่งข้อมูล) แต่ถ้าเกิดการแย่งใช้สายสัญญาณหลังจาก Clock ที่ 10 ไปแล้วจะถือว่าคีย์บอร์ดได้ส่งข้อมูลนั้นไปครบเรียบร้อยแล้ว หลังจากพ้นสภาวะการแย่งใช้สายสัญญาณแล้วคอมพิวเตอร์อาจจะส่งคำสั่งให้คีย์บอร์ดส่งข้อมูลนั้นมาใหม่หรือไม่ก็ได้

เมื่อมีการกดคีย์ใดๆ(ยกเว้นคีย์ Pause) จะมีการส่ง Make/Break code ออกมาจากคีย์บอร์ด โดยที่ Make code จะส่งออกมาเมื่อมีการกดคีย์ส่วน Break code จะส่งออกมาเมื่อมีการปล่อยคีย์ และคีย์ทุกคีย์(ยกเว้นคีย์ Pause)จะมีการพิมพ์แบบอัตโนมัติคือเมื่อมีการกดคีย์ค้างไว้คีย์บอร์ดจะส่ง Make code ครั้งแรกออกมาหลังจากนั้นอีก 50ms$\pm$20% ถ้ายังไม่มีการปล่อยคีย์นั้นคีย์บอร์ดก็จะทำการส่ง Make code ของคีย์นั้นออกมาอีกในอัตรา 10.9 ครั้งต่อวินาทีจนกระทั่งปล่อยคีย์นั้น ถ้ามีการกดคีย์มากกว่า 1 คีย์พร้อมกันค้างไว้จะมีเฉพาะคีย์สุดท้ายเท่านั้นที่จะมีการพิมพ์แบบอัตโนมัติและจะหยุดเมื่อคีย์สุดท้ายนั้นถูกปล่อยถึงแม้ว่าคีย์อื่นยังถูกกดอยู่ก็ตาม ถ้ามีการกดคีย์ค้างไว้ขณะที่คีย์บอร์ดยังไม่สามารถส่งข้อมูลได้ Make code ครั้งแรกเท่านั้นที่จะเก็บไว้ในคีย์บอร์ดบัฟเฟอร์เพื่อป้องกันมิให้คีย์บอร์ดบัฟเฟอร์เต็มอันเนื่องมาจากผลของการพิมพ์แบบอัตโนมัติ

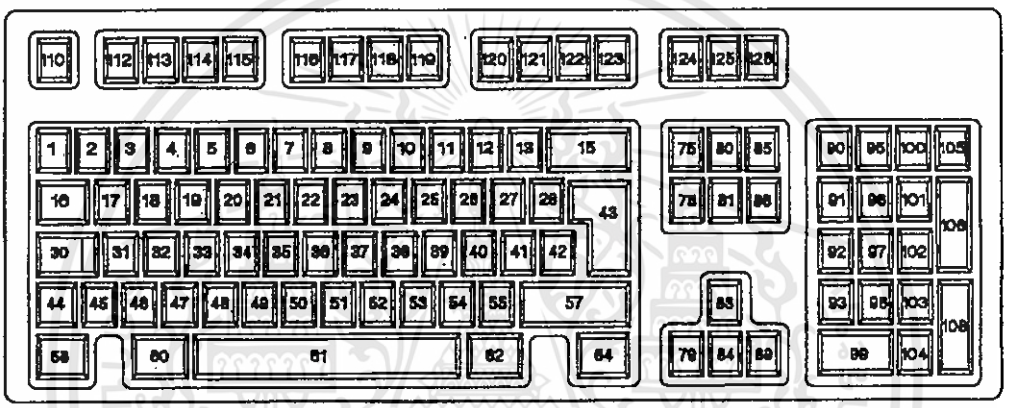
ตารางที่ 2.2 ค่า Make code และ Break code ของคีย์บอร์ด

Key	Make code (hex)	Break Code (hex)	Key	Make code (hex)	Break Code (hex)	Key	Make code (hex)	Break Code (hex)
1	0E	F0 0E	37	3B	F0 3B	102	74	F0 74
2	16	F0 16	38	42	F0 42	103	7A	F0 7A
3	1E	F0 1E	39	4B	F0 4B	104	71	F0 71
4	26	F0 26	40	4C	F0 4C	105	7B	F0 7B
5	25	F0 25	41	52	F0 52	106	79	F0 79
6	2E	F0 2E	**42	5D	F0 5D	108	E0 5A	E0 F0 5A
7	36	F0 36	43	5A	F0 5A	110	76	F0 76
8	3D	F0 3D	44	12	F0 12	112	05	F0 05
9	3E	F0 3E	**45	61	F0 61	113	06	F0 06
10	46	F0 46	46	1A	F0 1A	114	04	F0 04
11	45	F0 45	47	22	F0 22	115	0C	F0 0C
12	4E	F0 4E	48	21	F0 21	116	03	F0 03
13	55	F0 55	49	2A	F0 2A	117	0B	F0 0B
15	66	F0 66	50	32	F0 32	118	83	F0 83
16	0D	F0 0D	51	31	F0 31	119	0A	F0 0A
17	15	F0 15	52	3A	F0 3A	120	01	F0 01
18	1D	F0 1D	53	41	F0 41	121	09	F0 09
19	24	F0 24	54	49	F0 49	122	78	F0 78
20	2D	F0 2D	55	4A	F0 4A	123	07	F0 07
21	2C	F0 2C	57	59	F0 59	125	7E	F0 7E
22	35	F0 35	58	14	F0 14	75	E0 70	E0 F0 70
23	3C	F0 3C	60	11	F0 11	76	E0 71	E0 F0 71
24	43	F0 43	61	29	F0 29	79	E0 6B	E0 F0 6B
25	44	F0 44	62	E0 11	E0 F0 11	80	E0 6C	E0 F0 6C
26	4D	F0 4D	64	E0 14	E0 F0 14	81	E0 69	E0 F0 69
27	54	F0 54	90	77	F0 77	83	E0 75	E0 F0 75
28	5B	F0 5B	91	6C	F0 6C	84	E0 72	E0 F0 72
*29	5D	F0 5D	92	6B	F0 6B	85	E0 7D	E0 F0 7D
30	58	F0 58	93	69	F0 69	86	E0 7A	E0 F0 7A
31	1C	F0 1C	96	75	F0 75	89	E0 74	E0 F0 74
32	1B	F0 1B	97	73	F0 73	95	E0 4A	E0 F0 4A
33	23	F0 23	98	72	F0 72	124	E0 12 E0 7C	E0 F0 7C E0 F0 12
34	2B	F0 2B	99	70	F0 70	126	E1 14 77 E1 F0 14 F0 77	
35	34	F0 34	100	7C	F0 7C			
36	33	F0 33	101	7D	F0 7D			

* 101 KEY ONLY ** 102 KEY ONLY



(ก) คีย์บอร์ดแบบ 101 คีย์

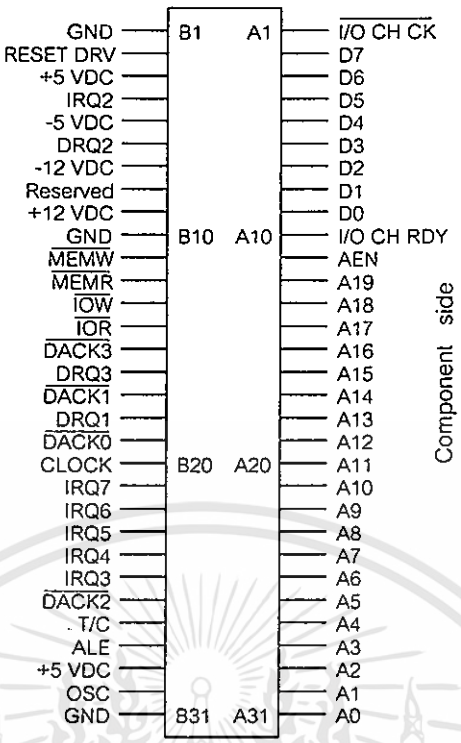


(ข) คีย์บอร์ดแบบ 102 คีย์

รูปที่ 2.4 ตำแหน่งของคีย์ของคีย์บอร์ดแบบ 101 คีย์และแบบ 102คีย์

2.2 สล็อตขยายระบบของ IBM PC

IBM PC ได้ออกแบบมาให้ผู้ใช้สามารถออกแบบการ์ดต้นแบบ(Prototype Card)ขึ้นใช้เองได้ โดยเชื่อมต่อเข้ากับระบบบัสของ IBM PC ผ่านทางสล็อตขยายระบบ โดยสล็อตขยายระบบนี้นอกจากจะมีช่องสัญญาณที่เชื่อมต่อกับระบบบัสของไมโครโปรเซสเซอร์แล้วยังมีสัญญาณอื่นๆ เพิ่มเติมอีกสำหรับกระบวนการทำ DMA กระบวนการขัดจังหวะ(Interrupt) และอื่นๆ โดยสล็อตขยายระบบมีสัญญาณต่างๆดังรูปที่ 2.5



รูปที่ 2.5 สล๊อตขยายระบบของ IBM PC

สัญญาณต่างๆในช่องสล๊อตขยายระบบมีหน้าที่ดังนี้

A0 – A9 เป็นสัญญาณแอดเดรสของหน่วยความจำและอุปกรณ์อินพุต/เอาต์พุต ซึ่งมาจากไมโครโปรเซสเซอร์ หรือมาจาก DMA controller

D0 – D7 เป็นสัญญาณข้อมูล 8 บิตแบบ 2 ทิศทาง ในช่วงของการเขียนข้อมูล สัญญาณข้อมูลจากไมโครโปรเซสเซอร์จะถูกวางลงบนบัสก่อนขอบขาขึ้นของสัญญาณ $\overline{IOW}$ และ $\overline{MEMW}$ ส่วนในช่วงของการอ่านข้อมูลอุปกรณ์อินพุต/เอาต์พุตหรือหน่วยความจำต้องวางข้อมูลลงบนบัสก่อนขอบขาขึ้นของสัญญาณ $\overline{IOR}$ และ $\overline{MEMR}$ เพื่อที่สัญญาณข้อมูลจะได้อ่านเข้าไปในตัวไมโครโปรเซสเซอร์ได้

$\overline{IOR}$ เป็นสัญญาณการอ่านข้อมูลจากอุปกรณ์อินพุต/เอาต์พุต

$\overline{IOW}$ เป็นสัญญาณการเขียนข้อมูลไปยังอุปกรณ์อินพุต/เอาต์พุต

$\overline{MEMR}$ เป็นสัญญาณการอ่านข้อมูลจากหน่วยความจำ

$\overline{MEMW}$ เป็นสัญญาณการเขียนข้อมูลไปยังหน่วยความจำ

ALE เป็นสัญญาณแสดงวงรอบการทำงานของบัส (bus cycle) ใช้ในการแยกสัญญาณแอดเดรสจากสัญญาณข้อมูลจากไมโครโปรเซสเซอร์

AEN เป็นสัญญาณที่มาจาก DMA controller เพื่อบอกว่ากำลังอยู่ในระหว่างกระบวนการทำ DMA

OSC	เป็นสัญญาณนาฬิกาที่มีความถี่ 14.31818 MHz มีดิวตี้ไซเคิลเป็น 50 เปอร์เซ็นต์
CLOCK	เป็นสัญญาณนาฬิกาที่มีความถี่เป็น 1 ใน 3 ของ OSC คือ 4.77 MHz มีดิวตี้ไซเคิลเป็น 33 เปอร์เซ็นต์
IRQ2 – IRQ7	เป็นสัญญาณร้องขอการขัดจังหวะ โดยสัญญาณ IRQ2 มีลำดับความสำคัญสูงสุด และ IRQ7 มีความสำคัญต่ำสุด
I/O CH RDY	เป็นสัญญาณอินพุตสำหรับสร้างสภาวะการรอ (Wait state) เพื่อขยายความยาวของวงรอบบัสออกไป ใช้สำหรับเชื่อมต่อบัสเข้ากับหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุตที่ทำงานช้ากว่าระบบ
$\overline{\text{I/O CH CK}}$	เป็นสัญญาณที่แจ้งให้ไมโครโปรเซสเซอร์ทราบว่ามีการตรวจสอบพาริตี้ (parity) ที่ผิดพลาดในหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุต
RESET DRV	เป็นสัญญาณที่ใช้รีเซ็ตระบบ หรือใช้เริ่มต้นระบบใหม่เมื่อมีการเปิดระบบขึ้นหรือแรงดันไฟเลี้ยงตกลงมากจนไม่สามารถทำงานต่อได้
DRQ1 – DRQ3	เป็นสัญญาณร้องขอกระบวนการทำ DMA โดยที่สัญญาณ DRQ ต้องคงค้างที่สภาวะ 1 จนกระทั่งสัญญาณ $\overline{\text{DACK}}$ ที่สอดคล้องกันกลายเป็นสภาวะ 0
$\overline{\text{DACK0}}$ - $\overline{\text{DACK3}}$	เป็นสัญญาณตอบรับการร้องขอกระบวนการทำ DMA
T/C	เป็นสัญญาณบอกให้ทราบว่ากระบวนการทำ DMA ได้สิ้นสุดลงแล้ว

ใน IBM PC จะมีแอดเดรสสำหรับใช้เป็นอุปกรณ์อินพุต/เอาต์พุตอยู่ทั้งสิ้น 1024 หรือ 1K แอดเดรส และมีแอดเดรสสำหรับหน่วยความจำจำนวน 1 เมกกะไบต์ การอ้างแอดเดรสของอุปกรณ์อินพุต/เอาต์พุตจะใช้สายสัญญาณแอดเดรสทั้งสิ้น 10 เส้นคือ A0-A9 ซึ่งแอดเดรสจำนวนนี้ถูกแบ่งออกเป็น 2 ส่วนด้วยกันคือใน 512 แอดเดรสแรก คือ 0000H ถึง 01FFH จะถูกใช้โดยอุปกรณ์ต่างๆบนเมนบอร์ด(Main board) ส่วนอีก 512 แอดเดรสที่เหลือคือ 0200H ถึง 03FFH จะถูกใช้โดยการ์ดต่างๆที่เสียบอยู่บนสล๊อตขยายระบบ โดยในตารางที่ 2.3 จะแสดงการแบ่งตำแหน่งแอดเดรสสำหรับอุปกรณ์ต่างๆที่ IBM PC ได้แบ่งไว้ จะเห็นว่าการ์ดต้นแบบได้ถูกกำหนดให้อยู่ในช่วงแอดเดรส 0300H ถึง 031FH เป็นจำนวนทั้งหมด 32 แอดเดรส

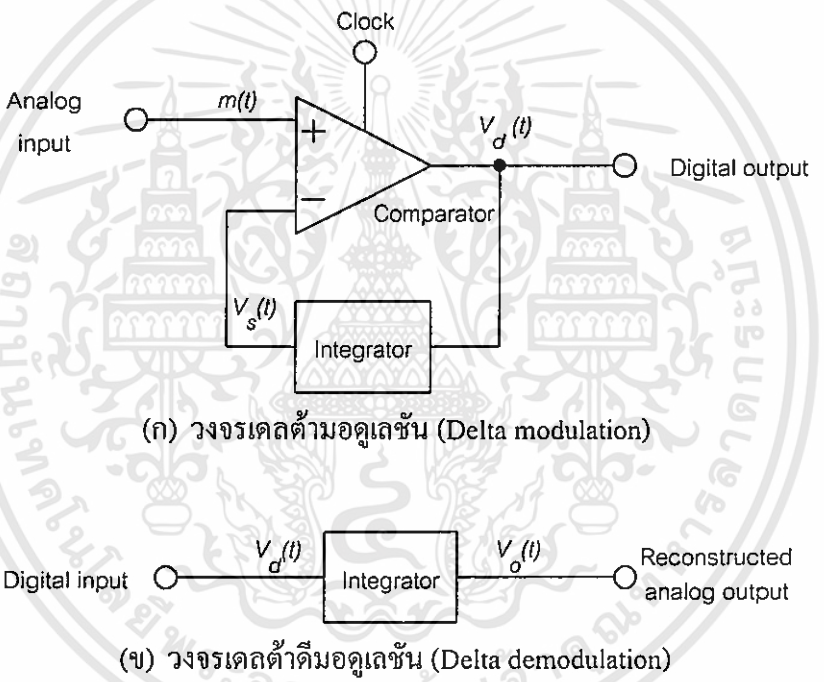
ตารางที่ 2.3 ตำแหน่งของอุปกรณ์อินพุต/เอาต์พุตของ IBM PC

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
100-1FF	Not usable
200-20F	Game control
210-217	Expansion unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous communications(2)
300-31F	Prototype card
320-32F	Fixed disk
378-37F	Printer
380-38C	SDLC communications
380-389	Binary synchronous communications(2)
3A0-3A9	Binary synchronous communications(1)
3B0-3BF	IBM monochrome display/printer
3C0-3CF	Reserved
3D0-3DF	Color/graphics
3E0-3F7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous communications(1)

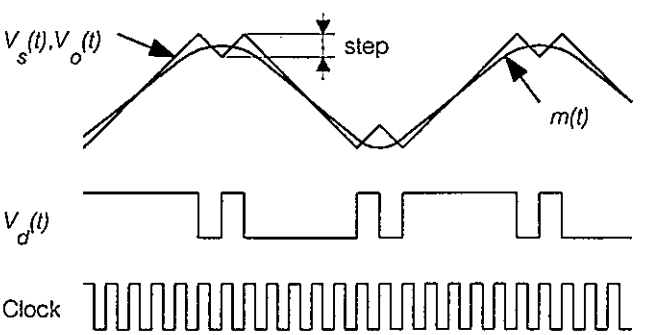
2.3 Delta Modulation (DM)

2.3.1 Linear Delta Modulation (LDM)

เดลตามอดูเลชันเป็นวิธีหนึ่งในการแปลงสัญญาณอนาลอกแป็นสัญญาณดิจิตอลหรือแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก โดยในการสุ่มแต่ละครั้งจะได้ข้อมูลดิจิตอลจำนวน 1 บิต ในรูปแบบปกติจะให้สัญญาณอนาลอกเอาต์พุตเป็นสัญญาณรูปขั้นบันได โดยมีความแตกต่างในการสุ่มแต่ละครั้งเป็น ±step เท่านั้น ถ้าสัญญาณอินพุตมีค่ามากกว่าสัญญาณที่ได้จากการสุ่มครั้งก่อนก็จะทำการประมาณสัญญาณเอาต์พุตเดิมขึ้นอีก +step ในทางกลับกันถ้าสัญญาณอินพุตมีค่าน้อยกว่าสัญญาณที่ได้จากการสุ่มครั้งก่อนก็จะทำการประมาณสัญญาณเอาต์พุตเดิมลงไป –step การทำงานดังกล่าวข้างต้นสามารถเขียนเป็นแผนภาพได้ดังรูปที่ 2.6



รูปที่ 2.6 แผนภาพการทำงานของระบบ LDM



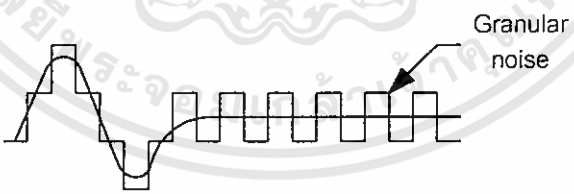
รูปที่ 2.7 สัญญาณอินพุตและเอาต์พุตของระบบ LDM

ระบบเดลต้ามอดูเลชันนั้นถ้ายิ่งมีความถี่ในการสุ่มสัญญาณมากก็จะทำให้สัญญาณเอาต์พุตที่ได้มีขนาดที่ใกล้เคียงกับสัญญาณจริงมากยิ่งขึ้นซึ่งก็คือจะได้เสียงที่คุณภาพดีขึ้นแต่ก็มีข้อเสียก็คือการสิ้นเปลืองหน่วยความจำ โดยในการเลือกความถี่ในการสุ่มนั้นต้องเลือกให้มีค่าเป็นอย่างน้อย 2 เท่าของความถี่ของสัญญาณอินพุต ซึ่งความถี่ที่ใช้ในการสุ่มนี้ก็คืออัตราการส่งข้อมูล (bit rate) นั่นเอง

จากการทำงานของระบบ LDM ถ้าสัญญาณอินพุตที่เข้ามามีการเปลี่ยนแปลงเมื่อเทียบกับเวลาไม่รวดเร็วจนเกินไปเมื่อเทียบกับขนาดของ step จะพบว่าสัญญาณเอาต์พุตโดยเฉลี่ยจะมีขนาดใกล้เคียงกับสัญญาณอินพุต นั่นก็หมายถึงว่าขนาดของ step เป็นสิ่งที่สำคัญของระบบเดลต้ามอดูเลชัน เพราะว่าถ้าขนาดของ step ที่เล็กจนเกินไปในขณะที่สัญญาณอินพุตมีการเปลี่ยนแปลงที่รวดเร็วจะทำให้สัญญาณที่ประมาณขึ้นตามสัญญาณอินพุตไม่ทัน ซึ่งจะเรียกปัญหานี้ว่า Slope overload ดังแสดงในรูปที่ 2.8

รูปที่ 2.8 ปัญหา Slope overload ของระบบ LDM

ในทางตรงกันข้ามถ้าขนาดของ step ใหญ่จนเกินไปในขณะที่สัญญาณอินพุตไม่ค่อยเปลี่ยนแปลงก็จะเกิดปัญหาที่เรียกว่า Granular noise ขึ้นดังแสดงในรูปที่ 2.9

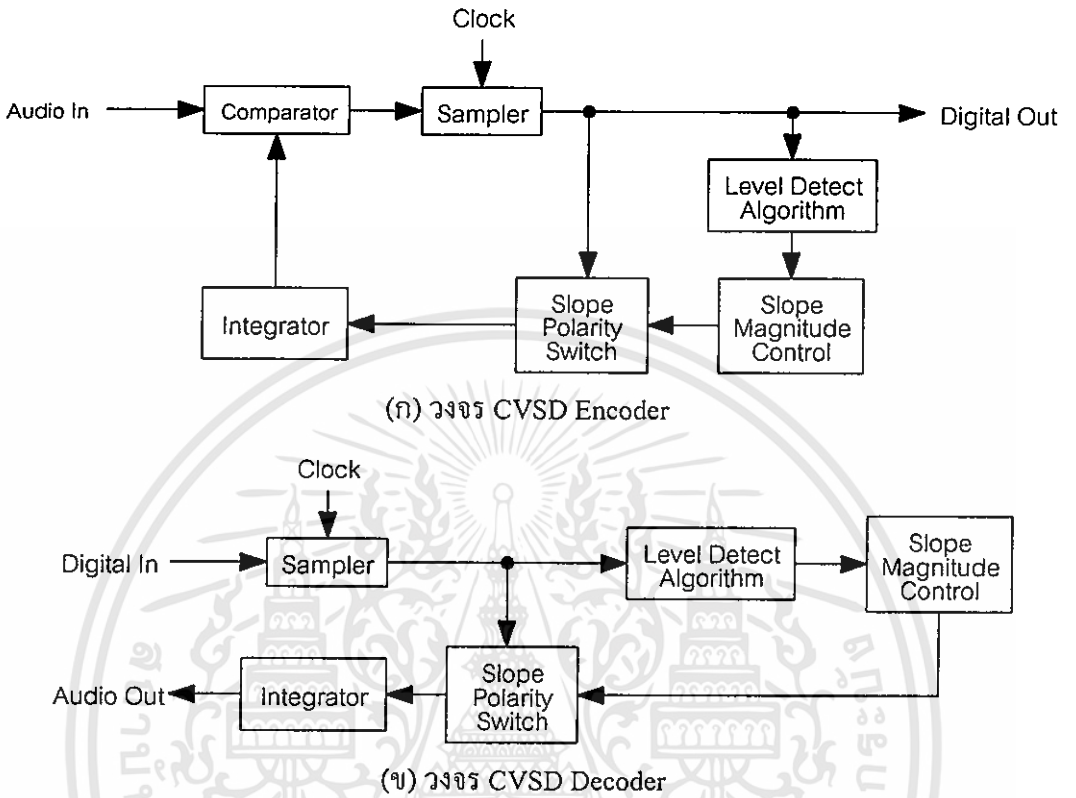


รูปที่ 2.9 ปัญหา Granular noise ของระบบ LDM

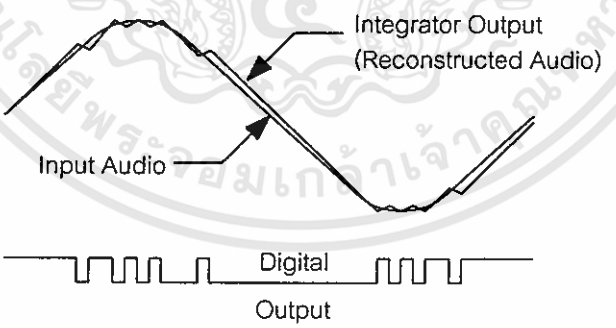
2.3.2 Continuously Variable Slope Delta Modulation (CVSD)

จากปัญหา Slope overload และ Granular noise ของระบบเดลต้ามอดูเลชันนั้นจะสามารถแก้ไขได้โดยการปรับเปลี่ยนขนาดของ step ไปตามการเปลี่ยนแปลงของสัญญาณอินพุต ซึ่งถ้าอินพุตมีการเปลี่ยนแปลงที่รวดเร็วก็ทำการเปลี่ยนขนาดของ step ให้มีค่ามากขึ้น ในทางตรงกันข้ามถ้าสัญญาณอินพุตมีการเปลี่ยนแปลงไม่มากนัก ก็ทำการเปลี่ยนขนาดของ step ให้มีค่าลดลง

หลักการข้างต้นนี้ก็คือหลักการของระบบ CVSD นั่นเอง ระบบ CVSD มีแผนภาพการทำงานดังรูปที่ 2.10



รูปที่ 2.10 แผนภาพการทำงานของระบบ CVSD



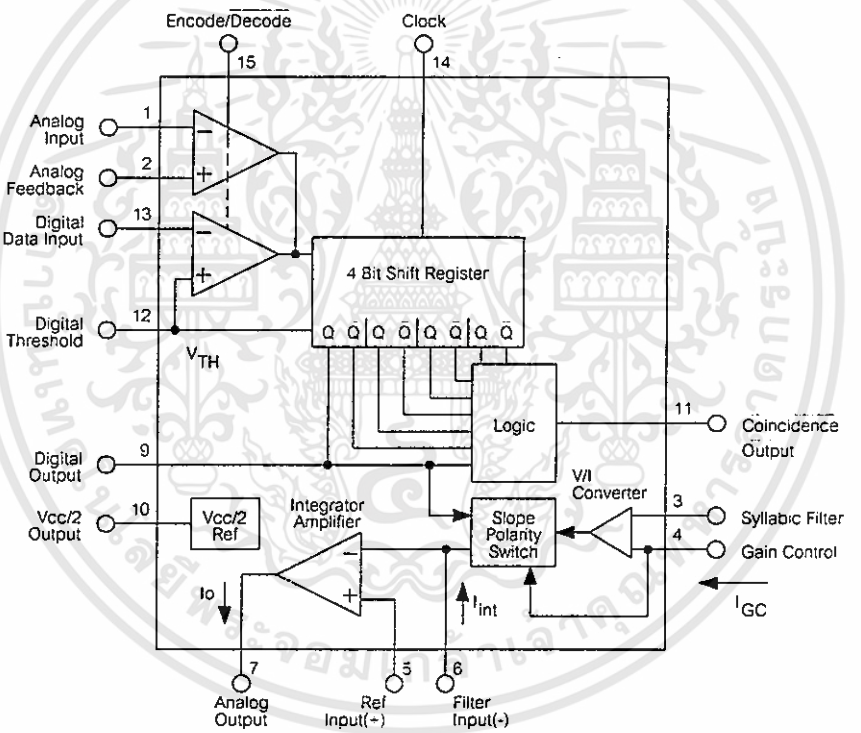
รูปที่ 2.11 สัญญาณอินพุตและเอาต์พุตของระบบ CVSD

ในการเปลี่ยนแปลงขนาดของ step ของระบบ CVSD ทำได้โดยการใช้ส่วนตรวจระดับสัญญาณ (Level detect algorithm) ซึ่งจะประกอบด้วย shift register ขนาด 3 หรือ 4 บิตในการตรวจสอบข้อมูลดิจิตอลว่ามีข้อมูลเป็นลอจิก '0' หรือ '1' ติดกันถึง 3 หรือ 4 บิตหรือไม่ ถ้าเกิดเหตุการณ์ดังกล่าวขึ้นหมายถึงว่าขนาดของ step เล็กจนเกินไปส่วนตรวจระดับสัญญาณจะส่ง

สัญญาณไปยังส่วนเปลี่ยนขนาดของความชัน (Slope magnitude control) เพื่อทำการเปลี่ยนขนาดของ step ให้มีค่าเพิ่มมากขึ้น ส่วนของ Slope polarity switch จะทำหน้าที่ควบคุมการเปลี่ยนแปลงเอาต์พุตของส่วน Integrator ว่าจะให้เปลี่ยนแปลงขึ้นหรือเปลี่ยนแปลงลง

2.3.3 คุณสมบัติของ MC3418

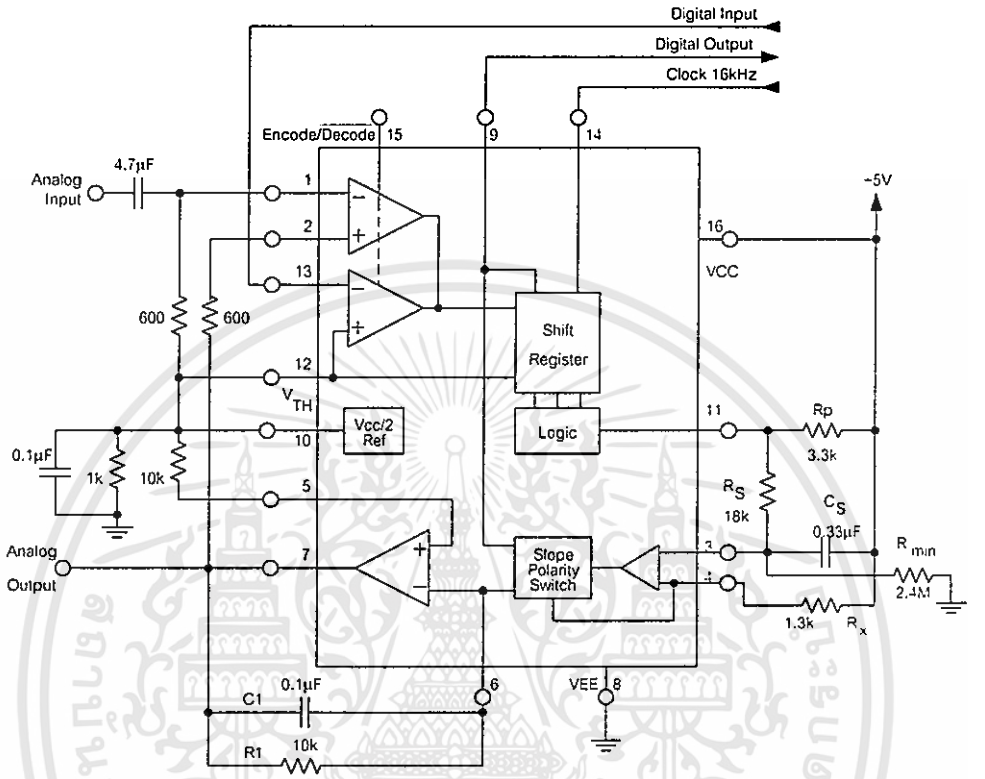
ไอซีเบอร์ MC3418 เป็นไอซีสำหรับการเข้ารหัสและถอดรหัสแบบ CVSD ที่ออกแบบมาสำหรับการใช้งานทางด้านการสื่อสารในทางทหารและการใช้งานทางด้านโทรศัพท์ โดยไอซีตัวเดียวสามารถเลือกได้ว่าจะเป็นตัวเข้ารหัสหรือตัวถอดรหัสโดยใช้สัญญาณดิจิตอลมาเลือก ภายในประกอบด้วย shift register ขนาด 4 บิต (ซึ่งเหมาะสำหรับงานทางด้านโทรศัพท์) มีแรงดันอ้างอิง Vcc/2 ในตัว ไอซีเบอร์ MC3418 มีโครงสร้างดังรูปที่ 2.12



รูปที่ 2.12 โครงสร้างของไอซีเบอร์ MC3418

Shift register ขนาด 4 บิตภายใน MC3418 ใช้สำหรับในการตรวจสอบสัญญาณเอาต์พุตและจะแสดงออกมาที่ขา $\overline{\text{Coincidence Output}}$ เมื่อภายใน shift register ประกอบด้วยลอจิก '1' หรือ '0' ทั้งหมด ซึ่งนั่นจะหมายถึงว่าความชันของเอาต์พุตของวงจรอินทิเกรตมีค่าที่น้อยเกินไปจะต้องมีการปรับค่าความชันหรืออัตราขยายของวงจรอินทิเกรตเพิ่มขึ้น โดยที่เอาต์พุตที่แสดงออกมาที่ขา $\overline{\text{Coincidence Output}}$ จะไปทำหน้าที่ charge วงจรกรองความถี่ต่ำผ่านที่เรียกว่า syllabic filter ที่อยู่ภายนอกตัวไอซีซึ่งมีผลให้ค่าความชันหรืออัตราขยายของวงจรอินทิเกรตเพิ่มขึ้น ซึ่งวงจร syllabic

filter นี้เองที่จะทำให้ความชันของวงจรอินทิเกรตมีค่าเปลี่ยนแปลงอย่างต่อเนื่อง ซึ่งก็จะทำให้ความชันของสัญญาณเอาต์พุตมีค่าที่เปลี่ยนแปลงตามสัญญาณอินพุตโดยวงจร syllabic filter ในที่นี้ก็คือวงจรที่ประกอบด้วย C_S และ R_S ในวงจรรูปที่ 2.13



รูปที่ 2.13 วงจรเข้ารหัสและถอดรหัส CVSD ที่ใช้ในวิทยานิพนธ์

ในการออกแบบวงจรเข้ารหัสและถอดรหัสที่ใช้ MC3418 นั้นต้องพิจารณาสิ่งต่างๆ เช่น สัญญาณนาฬิกา ซึ่งโดยปกติจะใช้สัญญาณนาฬิกาอยู่ระหว่าง 9.6 ถึง 64 kHz โดยที่สัญญาณนาฬิกาความถี่สูงๆจะทำให้ระบบมี S/N ที่ดีขึ้น ในการเลือกความถี่ของสัญญาณนาฬิกานั้นมักจะถูกกำหนดด้วยแบนด์วิทธ์ของตัวกลางในการส่งสัญญาณ ในระบบเสียงทั่วไปมักจะใช้ความถี่ที่ 9600 Hz ในระบบวิทยุบางประเภทจะใช้ความถี่ที่ 12 kHz ในระบบของโทรศัพท์มักจะใช้ความถี่ที่ 16 kHz หรือ 32 kHz หรือมากกว่านี้ ส่วนงานในวิทยานิพนธ์นี้จะใช้ความถี่ที่ 16 kHz ซึ่งหมายถึงว่าต้องใช้หน่วยความจำขนาด 16 kbits สำหรับเก็บข้อมูลสัญญาณเสียงความยาว 1 วินาที

บทที่ 3

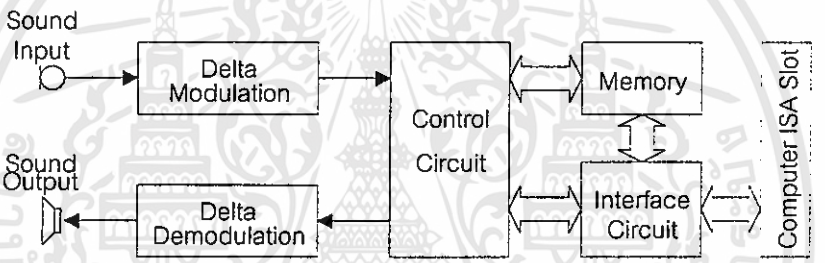
การออกแบบ

วิทยานิพนธ์นี้ประกอบด้วยงาน 2 ส่วนด้วยกัน คือ ส่วนของการ์ดบันทึกเสียงและส่วนของชุดฝึกพิมพ์ โดยในแต่ละส่วนจะประกอบด้วยส่วนของฮาร์ดแวร์และส่วนของซอฟแวร์ซึ่งจะได้แยกอธิบายเป็นส่วนๆต่อไป

3.1 การออกแบบทางฮาร์ดแวร์

3.1.1 ฮาร์ดแวร์ของส่วนการ์ดบันทึกเสียง

ในส่วนของการ์ดบันทึกเสียงมีโครงสร้างดังรูปที่ 3.1



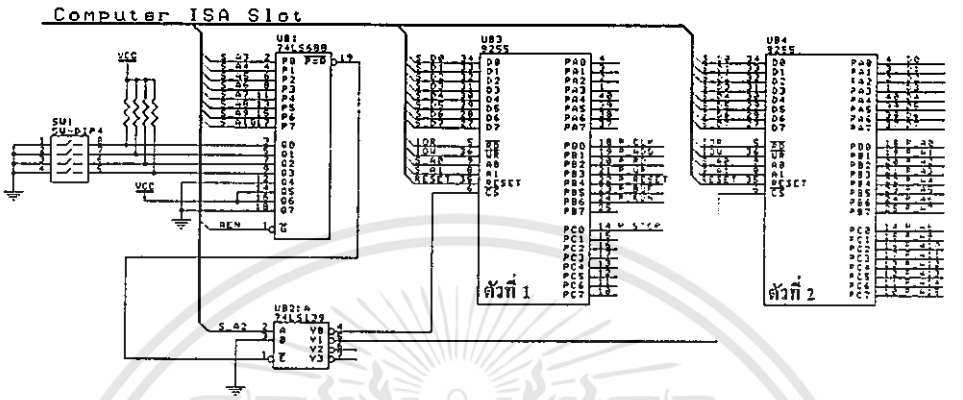
รูปที่ 3.1 โครงสร้างของการ์ดบันทึกเสียง

จากรูปที่ 3.1 ในแต่ละส่วนมีหน้าที่การทำงานดังนี้

Delta Modulation	ทำหน้าที่แปลงสัญญาณเสียงเป็นสัญญาณดิจิตอลขนาด 1 บิต
Delta Demodulation	ทำหน้าที่แปลงสัญญาณดิจิตอลกลับเป็นสัญญาณเสียง
Control Circuit	ทำหน้าที่คอยรับคำสั่งควบคุมจากโปรแกรมบนคอมพิวเตอร์ว่าจะสั่งบันทึกเสียง เล่นกลับ หรือหยุดการทำงาน เมื่อได้รับคำสั่งแล้วก็จะทำหน้าที่ควบคุมส่วนต่างๆให้ทำงานตามที่ต้องการ
Memory	ทำหน้าที่เก็บสัญญาณเสียงที่เป็นดิจิตอลที่ได้จากส่วนมอดูเลชันก่อนที่จะส่งขึ้นสู่คอมพิวเตอร์ หรือรับสัญญาณเสียงที่เป็นดิจิตอลที่บันทึกไว้บนคอมพิวเตอร์แล้วมาเก็บไว้เพื่อที่จะทำการเล่นกลับ
Interface Circuit	ทำหน้าที่เป็นอินพุต/เอาต์พุตพอร์ตสำหรับการสั่งงาน และอ่านเขียนข้อมูลจากคอมพิวเตอร์

3.1.1.1 วงจรอินเทอร์เฟส

วงจรอินเทอร์เฟสที่ใช้ประกอบด้วยส่วนถอดรหัสสัญญาณแอดเดรสและส่วนของ 8255 จำนวน 2 ตัวทำหน้าที่เป็นอินพุต/เอาต์พุตพอร์ต เพื่อทำหน้าที่ในการสั่งงานและอ่าน/เขียนข้อมูลลงสู่หน่วยความจำบนการ์ด ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 วงจรถอดรหัสที่ประกอบกับ 8255

วงจรถอดรหัสที่ใช้ประกอบด้วย 74LS688 และ 74LS139 ทำหน้าที่เปรียบเทียบสัญญาณแอดเดรสจากสล๊อตขยายระบบของคอมพิวเตอร์ โดยที่มีดิฟสวิทซ์ทำหน้าที่เลือกแอดเดรสที่ต้องการใช้ในกรณีที่เครื่องคอมพิวเตอร์นั้นมีการ์ดอื่นที่มีแอดเดรสตรงกันก็จะสามารถเปลี่ยนแอดเดรสของการ์ดที่สร้างขึ้นไม่ให้ตรงกับแอดเดรสของการ์ดอื่นได้ แอดเดรสของการ์ดที่สร้างขึ้นจะเป็นไปตามตารางที่ 3.1

ตารางที่ 3.1 แอดเดรสของการ์ดบันทึกเสียง

SW1-4	SW1-3	SW1-2	SW1-1	แอดเดรส
ON	ON	ON	ON	300-307
ON	ON	ON	OFF	308-30F
ON	ON	OFF	ON	310-317
ON	ON	OFF	OFF	318-31F
ON	OFF	ON	ON	320-327
ON	OFF	ON	OFF	328-32F
ON	OFF	OFF	ON	330-337
ON	OFF	OFF	OFF	338-33F
OFF	ON	ON	ON	340-347
OFF	ON	ON	OFF	348-34F
OFF	ON	OFF	ON	350-357
OFF	ON	OFF	OFF	358-35F
OFF	OFF	ON	ON	360-367
OFF	OFF	ON	OFF	368-36F
OFF	OFF	OFF	ON	370-377
OFF	OFF	OFF	OFF	378-37F

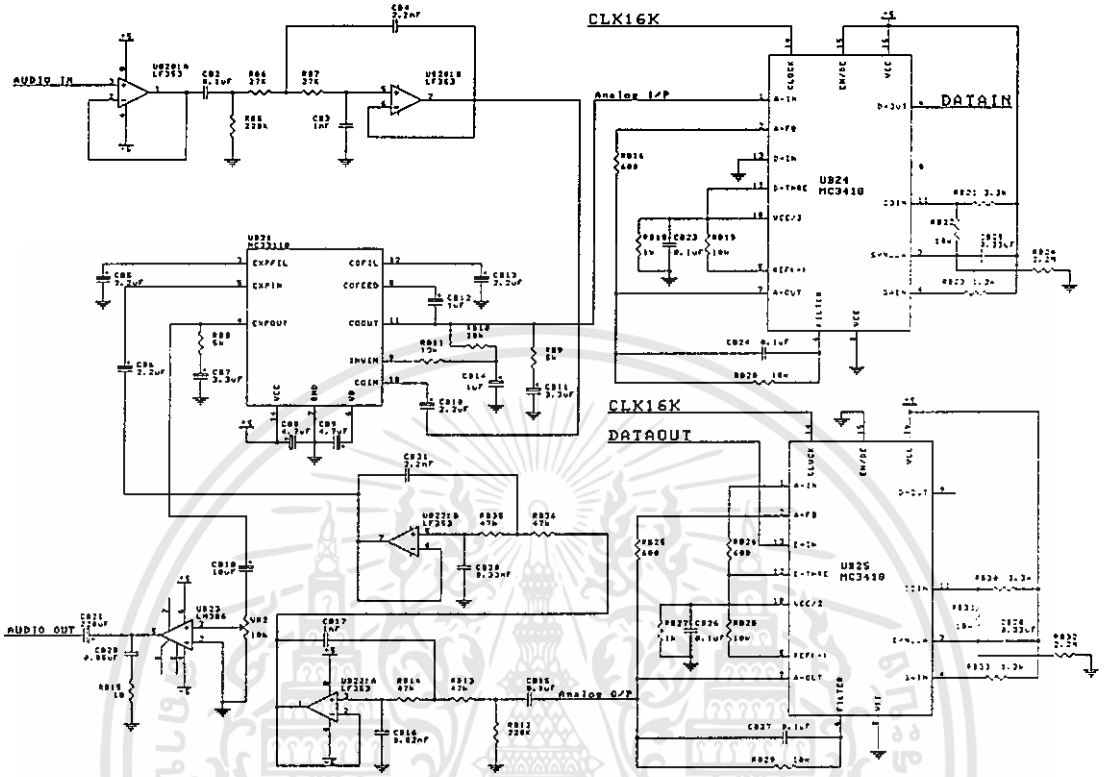
ส่วน 8255 ที่ใช้จำนวน 2 ตัว มีทั้งสิ้น 6 พอร์ต ทำหน้าที่อ่าน/เขียนข้อมูลต่างๆทั้งแอดเดรสและข้อมูลจากโปรแกรมบนคอมพิวเตอร์กับหน่วยความจำบนการ์ด และทำหน้าที่เป็นสัญญาณควบคุมการทำงานต่างๆของการ์ดให้เป็นไปตามต้องการ โดยสัญญาณในแต่ละบิตที่พอร์ตต่างๆจะทำหน้าที่ดังตารางที่ 3.2

ตารางที่ 3.2 หน้าที่ของพอร์ตต่างๆของ 8255

8255	พอร์ต	หน้าที่
ตัวที่ 1	PA0-PA7	ไม่ได้ใช้
	PB0 (P_CLK)	สัญญาณนาฬิกาของวงจรกำเนิดสัญญาณแอดเดรสที่สั่งงานโดยโปรแกรมบนคอมพิวเตอร์
	PB1 (P_ADD)	ใช้ในกรณีที่ต้องการอ่านค่าแอดเดรสที่ได้จากวงจรกำเนิดสัญญาณแอดเดรสขึ้นสู่คอมพิวเตอร์
	PB2 (P_RD)	สัญญาณอ่านข้อมูลในหน่วยความจำที่สั่งงานโดยโปรแกรมบนคอมพิวเตอร์
	PB3 (P_WR)	สัญญาณเขียนข้อมูลลงในหน่วยความจำที่สั่งงานโดยโปรแกรมบนคอมพิวเตอร์
	PB4 (P_RESET)	สัญญาณรีเซ็ตวงจรกำเนิดสัญญาณแอดเดรส
	PB5 (P_BUF)	ทำหน้าที่กั้นข้อมูลที่ได้จากวงจรมอดูเลชันไม่ให้เข้าสู่ Data bus ในขณะที่มีการเล่นกลับ
	PB6 (P_RUN)	ทำหน้าที่สั่งให้วงจรเริ่มหรือหยุดการทำงาน
	PB7	ไม่ได้ใช้
	PC0 (P_STOP)	เป็นสัญญาณที่ทำหน้าที่บอกคอมพิวเตอร์ว่าได้สิ้นสุดการทำงานของวงจรแล้วในกรณีที่บันทึกครบ 16 วินาทีหรือในกรณีที่ทำการเล่นกลับจบแล้ว
	PC1-PC7	ไม่ได้ใช้
ตัวที่ 2	PA0-PA7	(D0-D7) สัญญาณข้อมูลของหน่วยความจำ
	PB0-PB7 PC0-PC7	(P_A0-P_A15) สัญญาณแอดเดรสของหน่วยความจำ

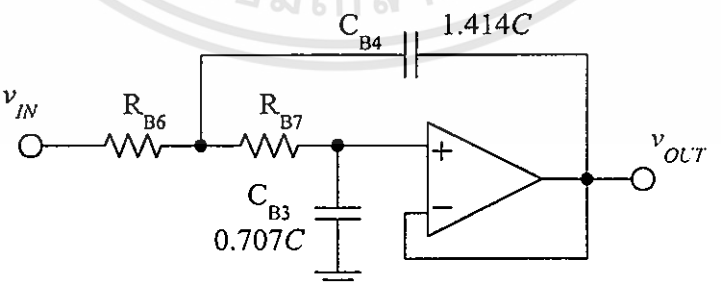
3.1.1.2 วงจรมอดูเลชันและดีมอดูเลชัน

วงจรมอดูเลชันและดีมอดูเลชันแสดงดังรูปที่ 3.3



รูปที่ 3.3 วงจรมอดูเลชันและดีมอดูเลชัน

ในส่วนภาคมอดูเลชันสัญญาณเสียงทางด้านอินพุตจะผ่านวงจรกรองความถี่ต่ำผ่านอันดับสองแบบบัตเตอร์เวิร์ตที่มีความถี่ตัดที่ 4 kHz เนื่องจากว่าสัญญาณเสียงคนพูดจะมีแบนวิดท์ที่ 4 kHz โดยวงจรกรองความถี่ต่ำผ่านอันดับสองแบบบัตเตอร์เวิร์ตมีลักษณะดังรูปที่ 3.4



รูปที่ 3.4 วงจรกรองความถี่ต่ำผ่านอันดับสองแบบบัตเตอร์เวิร์ต

จากวงจรในรูปที่ 3.4 ถ้ากำหนดให้ R_{B6} มีค่าเท่ากับ R_{B7} มีค่าเท่ากับ R จะได้ความถี่ตัดของวงจรดังสมการ (3.1)

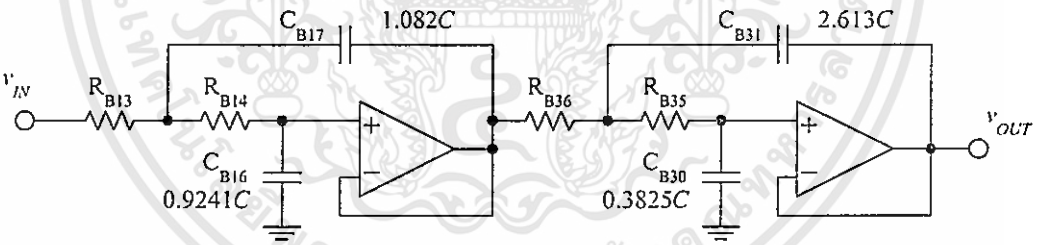
$$f_C = \frac{1}{2\pi RC} \tag{3.1}$$

จากสมการ (3.1) เมื่อกำหนดให้ความถี่ตัดของวงจรกรองในรูป 3.4 มีค่าเท่ากับ 4kHz และ C = 1.414nF จะได้ R = 27kΩ ดังนั้นจะได้ค่าของอุปกรณ์ทั้งหมดเป็น

$$\begin{aligned} R_{B6} &= R_{B7} = 27\text{k}\Omega \\ C_{B3} &= 0.707C = 1\text{nF} \\ C_{B4} &= 1.414C = 2\text{nF} \approx 2.2\text{nF} \end{aligned} \tag{3.2}$$

สัญญาณที่ผ่านวงจรกรองมาแล้วจะเข้าสู่วงจร Compander ที่ใช้ไอซีเบอร์ MC33110 ต่อไป โดยสัญญาณนี้จะเข้าไปในส่วนของการบีบอัดสัญญาณ(Compress) ก่อนโดยไอซี MC33110 นี้จะบีบอัดสัญญาณด้วยสมการ $V_o = 0.3162\sqrt{V_{in}}$ สัญญาณที่ได้บีบอัดแล้วนี้จะผ่านไปยังส่วนมอดูเลชันคือ MC3418 เปลี่ยนสัญญาณเสียงเป็นสัญญาณดิจิตอลเข้าไปสู่วงจรควบคุมต่อไป

ในส่วนของภาคดีมอดูเลชันสัญญาณดิจิตอลจากวงจรควบคุมจะถูกเปลี่ยนกลับเป็นสัญญาณเสียงโดย MC3418 ผ่านวงจรกรองความถี่ต่ำผ่านอันดับสี่แบบบัตเตอร์เวิร์ตที่มีความถี่ตัดเท่ากับ 4 kHz โดยวงจรกรองความถี่ต่ำผ่านอันดับสี่แบบบัตเตอร์เวิร์ตมีลักษณะดังรูปที่ 3.5



รูปที่ 3.5 วงจรกรองความถี่ต่ำผ่านอันดับสี่แบบบัตเตอร์เวิร์ต

จากวงจรในรูปที่ 3.5 ถ้ากำหนดให้ $R_{B13} = R_{B14} = R_{B35} = R_{B36}$ มีค่าเท่ากับ R จะได้ความถี่ตัดของวงจรดังสมการ (3.3)

$$f_C = \frac{1}{2\pi RC} \tag{3.3}$$

ถ้ากำหนดให้ความถี่ตัดของวงจรกรองในรูป 3.5 มีค่าเท่ากับ 4kHz และ C = 0.888nF จะได้ R = 45kΩ (เลือกใช้ค่า R จริง 47kΩ) ดังนั้นจะได้ค่าของอุปกรณ์ทั้งหมดเป็น

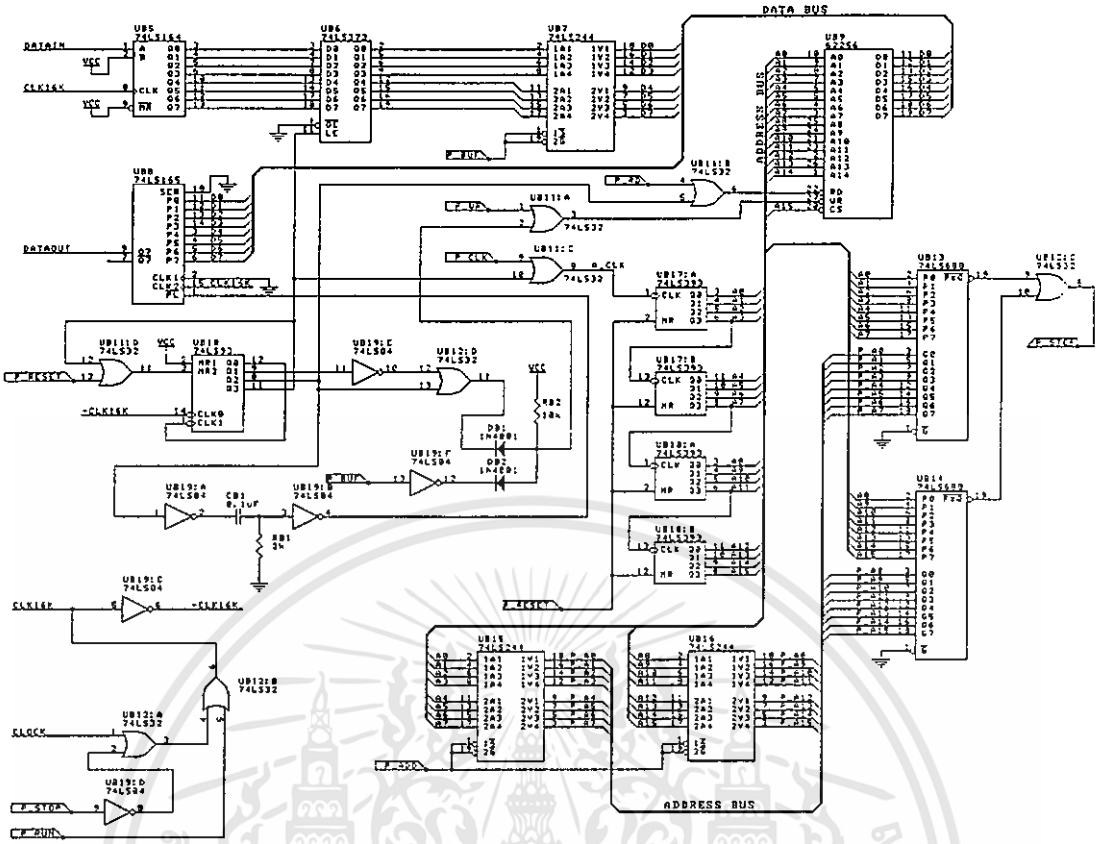
$$
\begin{aligned}
R_{B13} &= R_{B14} = R_{B35} = R_{B36} = 47k\Omega \\
C_{B16} &= 0.9241C = 0.82nF \\
C_{B17} &= 1.082C = 0.96nF \approx 1nF \\
C_{B30} &= 0.3825C = 0.34nF \approx 0.33nF \\
C_{B31} &= 2.613C = 2.32nF \approx 2.2nF
\end{aligned}
\tag{3.4}
$$

สัญญาณที่ได้จากวงจรกรองนี้จะนำมาทำการขยาย(Expand) ต่อไปโดยวงจรภายในไอซีเบอร์ MC33110 ด้วยสมการ $V_o = 10V_{in}^2$ และผ่านสู่วงจรขยายกำลังที่ใช้ LM386 เพื่อขับออกสู่ลำโพง

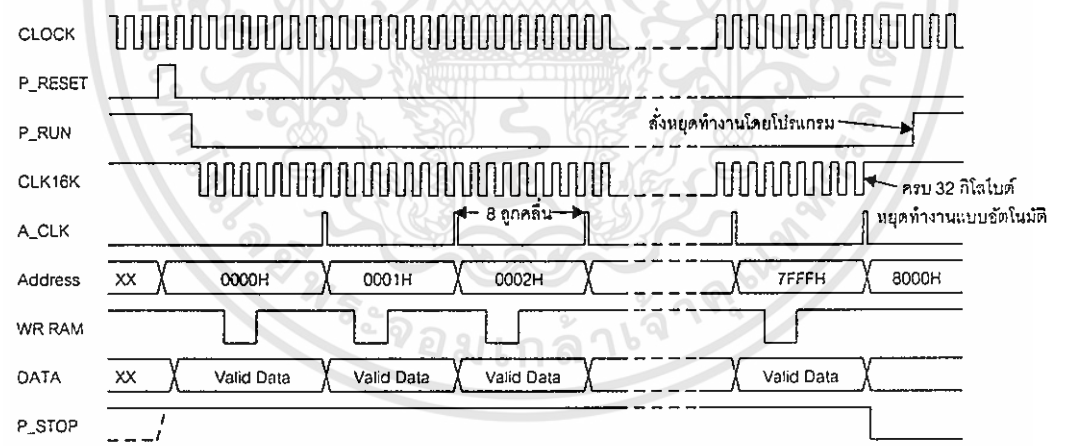
3.1.1.3 วงจรควบคุมและหน่วยความจำ

ในส่วนของวงจรควบคุมนี้จะทำหน้าที่รับคำสั่งควบคุมจากโปรแกรมคอมพิวเตอร์ที่ผ่าน 8255 ในส่วนของวงจรอินเทอร์เฟส (รูปที่ 3.2) เมื่อได้รับคำสั่งควบคุมก็จะทำหน้าที่ควบคุมการทำงานต่างๆได้แก่การสั่งเริ่มบันทึกเสียง การหยุดบันทึกเสียง การอ่านข้อมูลจากหน่วยความจำสู่คอมพิวเตอร์ การเขียนข้อมูลจากคอมพิวเตอร์ลงสู่หน่วยความจำและการเล่นกลับ โดยมีความถี่ที่ใช้ในการแปลงสัญญาณเท่ากับ 16kHz นั่นคือจะได้ข้อมูลจำนวน 16 กิโลบิตต่อ 1 วินาที หน่วยความจำที่ใช้มีขนาด 32 กิโลไบต์(หรือ 256 กิโลบิต) ดังนั้นการ์ดบันทึกเสียงที่สร้างขึ้นนี้สามารถบันทึกเสียงได้นานสูงสุดครั้งละ 16 วินาที วงจรรวมของส่วนวงจรควบคุมและหน่วยความจำแสดงดังรูป 3.6

และเมื่อมีการสั่งงานจากโปรแกรมบนคอมพิวเตอร์ก็จะเป็นการสั่งควบคุมบิตต่างๆของพอร์ต B ของ 8255 ตัวที่ 1 ในรูปที่ 3.2 ซึ่งจะมาทำการควบคุมการทำงานของวงจรส่วนต่างๆในรูปที่ 3.6 โดยการทำงานต่างๆของการ์ดนี้จะแสดงโดย Timing Diagram ดังรูปที่ 3.7 ถึง รูปที่ 3.11



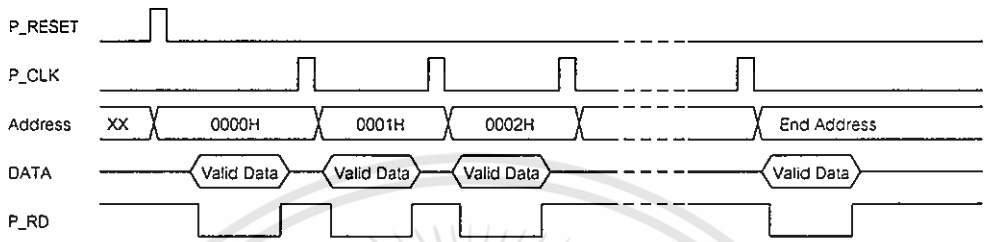
รูปที่ 3.6 วงจรรวมของส่วนควบคุมและหน่วยความจำ



รูปที่ 3.7 Timing diagram ของการสั่งบันทึกเสียง

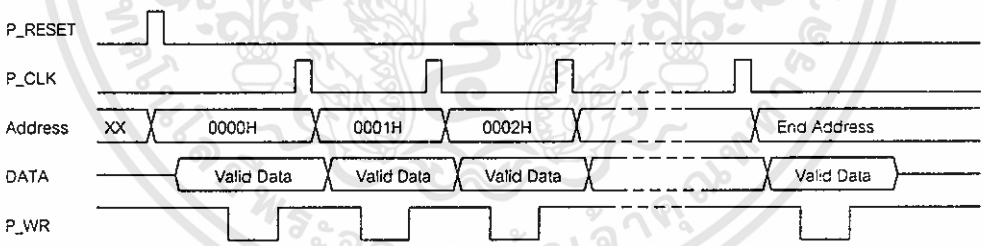
การสั่งบันทึกเสียงทำได้โดยสั่งรีเซ็ตแอดเดรสด้วยสัญญาณ P_RESET และตั้งค่าแอดเดรส P_A0 - P_A15 ที่พอร์ต B และ C ของ 8255 ตัวที่ 2 (ในรูปที่ 3.2) เป็น 8000H เพื่อกำหนดเป็นแอดเดรสสุดท้ายของการบันทึกเสียง แล้วจึงเริ่มการทำงานโดยสัญญาณ P_RUN ระหว่างนี้ส่วนควบคุมจะทำงานแบบอัตโนมัติคือนำข้อมูลที่ได้จากส่วนมอดูเลชันซึ่งเป็นข้อมูลอนุกรม 1 บิต เปลี่ยนเป็นข้อมูลขนาน 8 บิตเพื่อเขียนลงหน่วยความจำไปจนครบทุกแอดเดรสคือ 32 กิโลไบต์ (ถ้า

ไม่มีการสั่งหยุดบันทึก) เมื่อครบทุกแอดเดรสแล้ว คือสัญญาณแอดเดรสของหน่วยความจำมีค่าเท่ากับค่าที่ตั้งไว้ใน P_A0 - P_A15 สัญญาณ P_STOP จะกลายเป็นลอจิก '0' โดยอัตโนมัติ ซึ่งโปรแกรมบนคอมพิวเตอร์จะตรวจสอบพบและสั่งหยุดการทำงานอีกครั้งหนึ่งด้วยสัญญาณ P_RUN ในช่วงของการบันทึกนี้สัญญาณควบคุมที่ไม่ได้แสดงไว้คือสัญญาณ P_CLK, P_WR และ P_BUF จะเป็น ลอจิก '0' ส่วนสัญญาณ P_ADD และ P_RD จะเป็น ลอจิก '1'



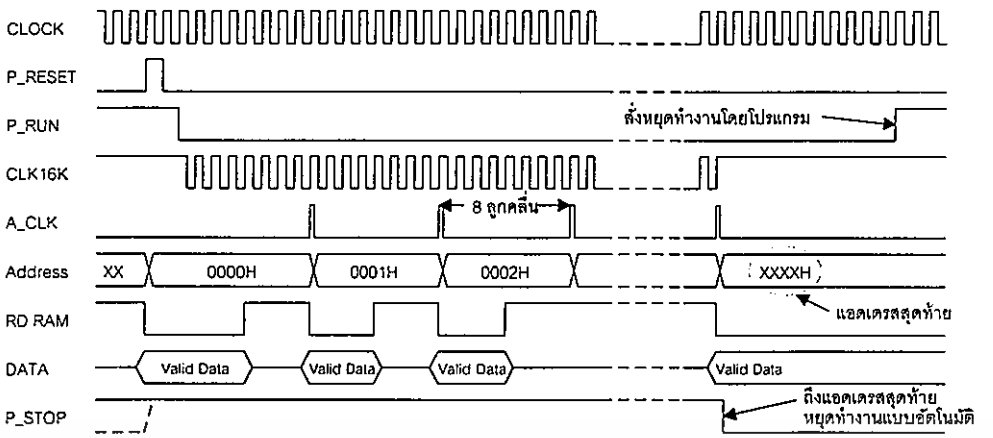
รูปที่ 3.8 Timing diagram ของการอ่านข้อมูลจากหน่วยความจำ

การอ่านข้อมูลจากหน่วยความจำทำได้โดยรีเซ็ตแอดเดรสด้วยสัญญาณ P_RESET แล้วจึงอ่านข้อมูลขึ้นมาบนคอมพิวเตอร์โดยใช้สัญญาณ P_RD และนับแอดเดรสขึ้น โดยใช้สัญญาณ P_CLK ทำการนับแอดเดรสขึ้นและอ่านข้อมูลสลับไปเรื่อยๆจนกระทั่งถึงแอดเดรสสุดท้าย ในช่วงของการอ่านข้อมูลจากหน่วยความจำนี้สัญญาณควบคุมที่ไม่ได้แสดงไว้คือสัญญาณ P_RUN, P_ADD, P_BUF และ P_WR จะเป็น ลอจิก '1' ทั้งหมด



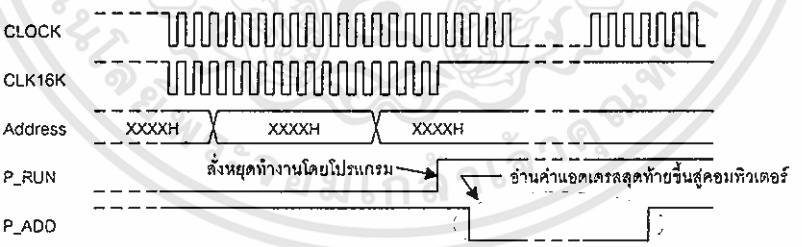
รูปที่ 3.9 Timing diagram ของการเขียนข้อมูลจากคอมพิวเตอร์ลงหน่วยความจำ

การเขียนข้อมูลจากคอมพิวเตอร์ลงหน่วยความจำทำได้โดยรีเซ็ตแอดเดรสด้วยสัญญาณ P_RESET จากนั้นก็วางข้อมูลไปที่ D0-D7 (พอร์ต A ของ 8255 ตัวที่ 2) แล้วจึงเขียนข้อมูลลงหน่วยความจำโดยใช้สัญญาณ P_WR และนับแอดเดรสขึ้น โดยใช้สัญญาณ P_CLK ทำการนับแอดเดรสขึ้น วางข้อมูลและเขียนข้อมูลสลับไปเรื่อยๆจนกระทั่งถึงแอดเดรสสุดท้าย ในช่วงของการเขียนข้อมูลลงหน่วยความจำนี้ สัญญาณควบคุมที่ไม่ได้แสดงไว้คือสัญญาณ P_RUN, P_ADD, P_BUF และ P_RD จะเป็น ลอจิก '1' ทั้งหมด



รูปที่ 3.10 Timing diagram ของการเล่นกลับ

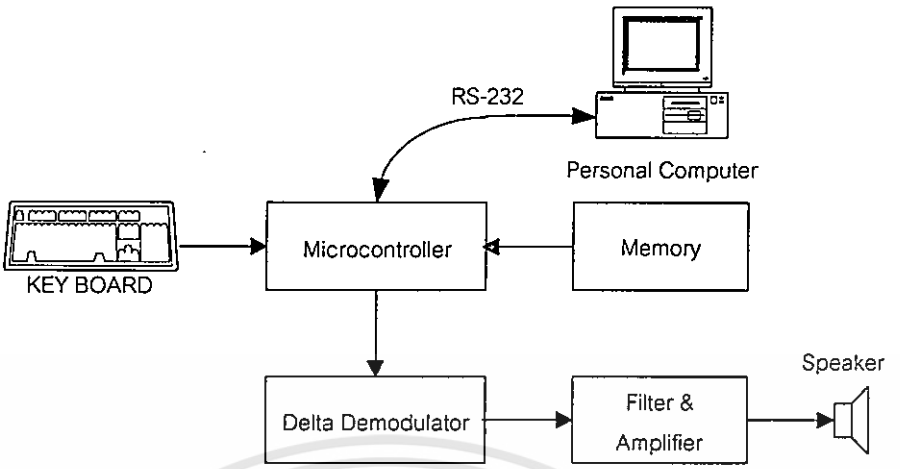
ในการสั่งงานการเล่นกลับจะเป็นการทำงาน 2 ขั้นตอนด้วยกันคือจะเขียนข้อมูลที่จะเล่นกลับลงในหน่วยความจำก่อนหลังจากนั้นจึงทำการเล่นกลับ เริ่มจากการกำหนดตั้งค่าแอดเดรส P_A0 - P_A15 ให้เป็นค่าที่ต้องการ แล้วจึงเริ่มการเล่นกลับโดยสัญญาณ P_RUN ระหว่างนี้ส่วนควบคุมจะทำงานแบบอัตโนมัติคือนำข้อมูลที่ได้จากหน่วยความจำขนาด 8 บิต มาเปลี่ยนเป็นข้อมูลอนุกรม 1 บิตส่งไปยังส่วนดีมอดูเลชัน เมื่อถึงแอดเดรสสุดท้ายแล้ว คือสัญญาณแอดเดรสของหน่วยความจำมีค่าเท่ากับค่าที่ตั้งไว้ใน P_A0 - P_A15 สัญญาณ P_STOP จะกลายเป็นลอจิก '0' โดยอัตโนมัติ ซึ่งโปรแกรมบนคอมพิวเตอร์จะตรวจสอบพบและสั่งหยุดการทำงานอีกครั้งหนึ่งด้วยสัญญาณ P_RUN ในช่วงของการเล่นกลับนี้สัญญาณควบคุมที่ไม่ได้แสดงไว้คือสัญญาณ P_CLK และ P_RD จะเป็นลอจิก '0' ส่วนสัญญาณ P_ADD, P_BUF และ P_WR จะเป็น ลอจิก '1'



รูปที่ 3.11 Timing diagram ของการสั่งหยุดการทำงาน

การหยุดการทำงานสามารถทำได้ทั้งในขณะที่ทำการบันทึกหรือทำการเล่นกลับโดยการสั่งสัญญาณ P_RUN เป็น ลอจิก '1' ก็เป็นอันหยุดการทำงานทั้งหมด แต่ถ้าทำการหยุดในขณะทำการบันทึกนั้นต้องทำการอ่านค่าแอดเดรสสุดท้ายก่อนที่จะหยุดทำงานขึ้นมาด้วยเพื่อที่จะได้ทราบว่าได้หยุดการทำงานที่แอดเดรสเท่าไรจะได้อ่านข้อมูลขึ้นมาจากหน่วยความจำถึงแอดเดรสสุดท้ายนั้นเท่านั้น การกระทำนี้ทำได้โดยสั่งสัญญาณ P_ADD เป็นลอจิก '0' ค่าของแอดเดรสที่หยุดนั้นก็สามารถอ่านได้โดยผ่านพอร์ต B และ C ของ 8255 ตัวที่ 2 นั่นเอง

3.1.2 ฮาร์ดแวร์ของส่วนชุดฝึกพิมพ์

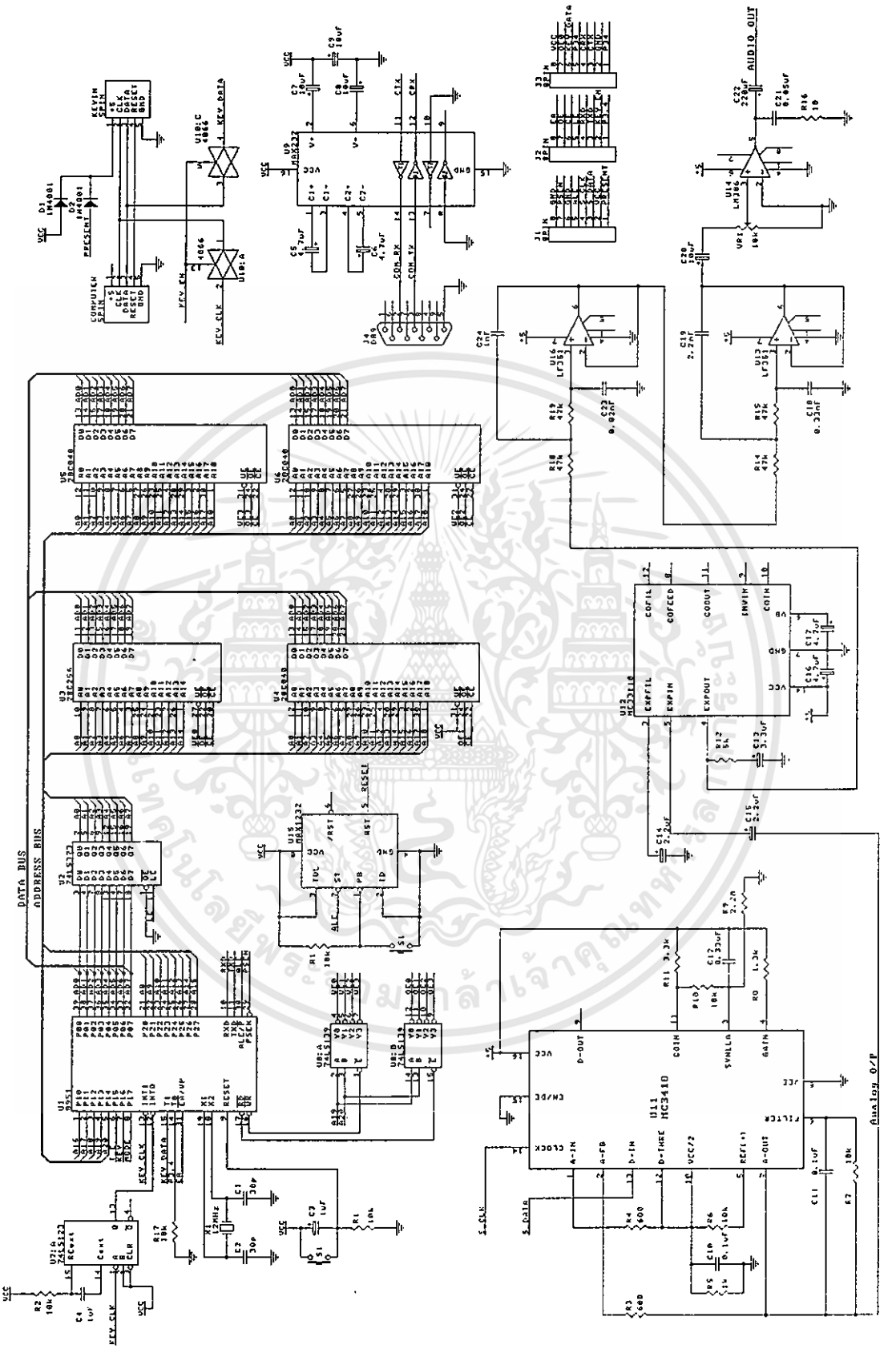


รูปที่ 3.12 โครงสร้างของชุดฝึกพิมพ์

ในส่วนของชุดฝึกพิมพ์จะประกอบด้วยส่วนของไมโครคอนโทรลเลอร์ที่ทำหน้าที่ควบคุมการทำงานทั้งหมด ในส่วนนี้จะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 รับสัญญาณจากคีย์บอร์ดมาทำการประมวลผลว่าเป็นคีย์ใด และมีส่วนของหน่วยความจำที่ใช้เก็บข้อมูลของเสียงที่ได้บันทึกไว้ก่อนแล้วเป็นข้อมูลดิจิตอลด้วยการ์ดบันทึกเสียงข้างต้น ส่วนสุดท้ายคือส่วนดีมอดูเลชันเพื่อเปลี่ยนข้อมูลดิจิตอลกลับมาเป็นสัญญาณเสียงผ่านวงจรกรองและวงจรขยายเพื่อขับออกลำโพงต่อไป

การทำงานของชุดฝึกพิมพ์จะมีสองโหมดการทำงานด้วยกันคือโหมดการทำงานปกติและโหมดการฝึกพิมพ์ ในโหมดการทำงานปกติจะทำงานดังนี้คือเมื่อมีการกดคีย์ใดๆบนคีย์บอร์ดจะมีเสียงบอกผู้ใช้ว่าได้กดคีย์ใดเช่นผู้ใช้กดคีย์ตัวอักษร ‘ก’ ก็จะมีเสียงบอกผู้ใช้ว่า ‘กอ-ไก่’ เป็นต้น การเปลี่ยนระหว่างคีย์ภาษาไทยและภาษาอังกฤษทำได้โดยกดสวิทซ์เปลี่ยนภาษาบนตัวเครื่อง

ส่วนโหมดการทำงานที่สองคือโหมดการฝึกพิมพ์จะเป็นโหมดการทำงานที่ให้ผู้ใช้สามารถฝึกฝนทักษะการพิมพ์ได้ทั้งภาษาไทยและภาษาอังกฤษโดยการพิมพ์คีย์ตามที่เครื่องบอก ซึ่งมีแบบฝึกหัดทั้งแบบบอกให้กดตามทีละคีย์และบอกเป็นคำให้พิมพ์ตามคำนั้นๆเช่นบอกคำว่า ‘ดาว’ ก็ให้พิมพ์คีย์ ‘ด’, ‘า’ และ ‘ว’ ตามลำดับ ซึ่งเมื่อจบการทำงานของแบบฝึกหัดแต่ละชุดจะมีการบอกผู้ใช้ด้วยว่าพิมพ์ได้ถูกต้องกี่เปอร์เซนต์และพิมพ์ด้วยความเร็วกี่คำต่อนาที โดยในส่วนของแบบฝึกหัดการพิมพ์นี้ผู้ใช้สามารถสร้างขึ้นด้วยตนเองได้โดยใช้โปรแกรมบนคอมพิวเตอร์ที่สร้างขึ้นมาดังจะได้กล่าวไว้ในส่วนของการออกแบบซอฟแวร์ต่อไป เมื่อสร้างแบบฝึกหัดเองได้แล้วก็สามารถโปรแกรมไปบนชุดฝึกพิมพ์ได้โดยผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์ ซึ่งจะเป็นการโปรแกรมลงบน EEPROM บนชุดฝึกพิมพ์ต่อไป ซึ่งการที่ให้ผู้ใช้สามารถสร้างแบบฝึกหัดขึ้นใช้เองได้นั้นจะเป็นการทำให้ในการใช้งานจริงๆมีความยืดหยุ่นในการใช้งานมากขึ้น ผู้ใช้สามารถออกแบบแบบฝึกหัดให้เหมาะสมกับผู้ฝึกพิมพ์และยุคสมัยได้โดยง่าย

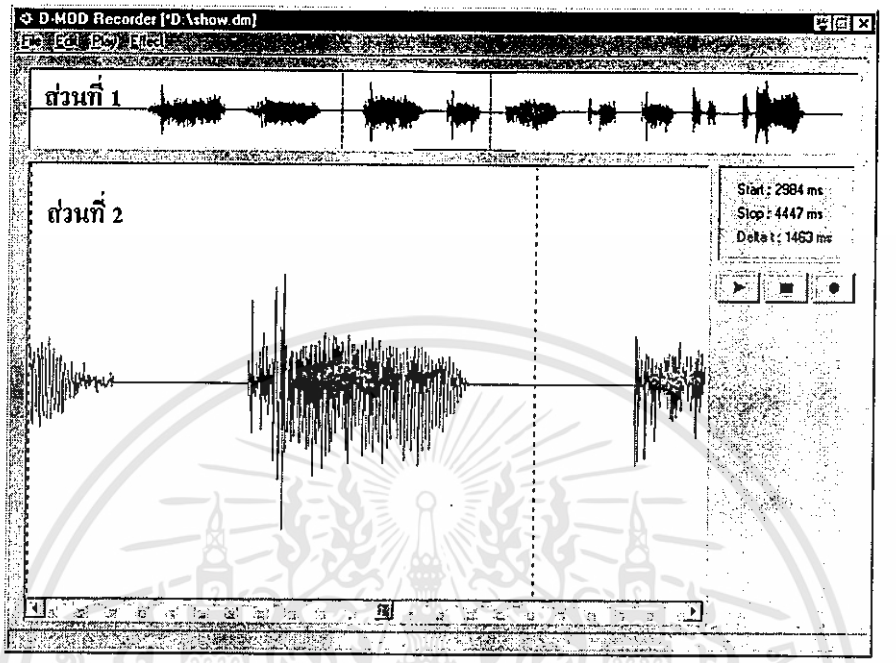


รูปที่ 3.13 วงจรรวมของชุดฝึกพิมพ์

3.2 การออกแบบทางซอฟแวร์

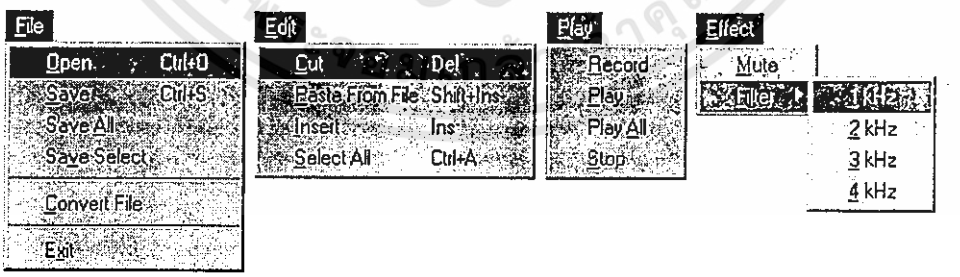
3.2.1 ซอฟแวร์ของส่วนการ์ดบันทึกเสียง

ซอฟแวร์ของส่วนบันทึกเสียงมีลักษณะดังรูปที่ 3.14



รูปที่ 3.14 หน้าจอของโปรแกรมบันทึกเสียง

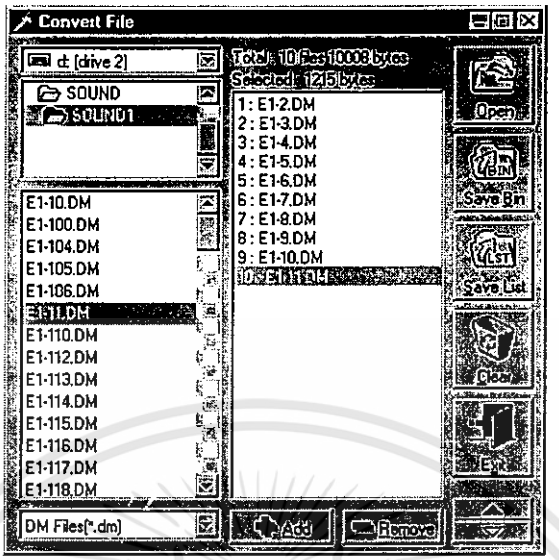
จากรูปที่ 3.14 จะเห็นว่าหน้าจอจะประกอบด้วย 2 หน้าต่างหลักๆคือส่วนที่ 1 จะเป็นการแสดงส่วนของเสียงทั้งหมดที่ได้ทำการบันทึกมีความยาวสูงสุด 16 วินาที และส่วนที่ 2 จะแสดงส่วนหนึ่งของเสียงที่ได้เลือกไว้ซึ่งหน้าต่างส่วนที่ 2 นี้จะมีความยาวเท่ากับ 1 วินาที ในส่วนของเมนูที่ใช้งานต่างๆมีดังรูปที่ 3.15



รูปที่ 3.15 เมนูต่างๆของโปรแกรมบันทึกเสียง

การใช้งานเมนูต่างๆจะกล่าวถึงโดยละเอียดในหัวข้อ 3.3 ต่อไป ซึ่งเมนูการใช้งานต่างๆโดยส่วนใหญ่จะมีความเหมือนกันกับโปรแกรมบันทึกเสียงทั่วๆไปอยู่แล้ว จะมีเพียงเมนู 2 เมนูที่ต่างไปจากโปรแกรมทั่วๆไปคือ เมนู File – Convert File และเมนู Effect – Filter ซึ่งจะมีการทำงานดังนี้

เมนู File – Convert File จะทำหน้าที่รวมไฟล์ที่ได้บันทึกไว้มารวมกันเป็นไฟล์เพียงไฟล์เดียวโดยเมื่อเข้าเมนูนี้หน้าจอของโปรแกรมจะเปลี่ยนไปดังรูปที่ 3.16



รูปที่ 3.16 หน้าจอของโปรแกรมเมื่อเข้าสู่เมนู File – Convert File

การบันทึกเสียงโดยโปรแกรมนี้จะบันทึกไฟล์เป็นนามสกุล DM ซึ่งจากรูปที่ 3.16 ในหน้าต่างทางด้านซ้ายมือจะเห็นว่ามีไฟล์ที่ได้บันทึกไว้แล้วเป็นจำนวนหนึ่ง เราสามารถทำการเลือกไฟล์โดยคลิกที่ไฟล์และกดปุ่ม Add ไฟล์ที่เลือกก็จะปรากฏที่หน้าต่างทางขวามือ เมื่อเลือกไฟล์ได้ครบตามต้องการแล้วก็ทำการกดปุ่ม Save List จะเป็นการบันทึกไฟล์ที่ทำหน้าที่บอกว่าเราได้ทำการเลือกไฟล์ใดไว้บ้าง(ซึ่งเราสามารถเปิดมาดูภายหลังได้โดยกดปุ่ม Open) หลังจากที่กดปุ่ม Save List แล้วก็ให้กดปุ่ม Save Bin จะเป็นการนำไฟล์ที่ได้เลือกไว้นั้นนำมาเรียงต่อกันเป็นไฟล์ไบนารีไฟล์เดียว โดยจะแทรกข้อมูลที่หัว(Header)ของไฟล์นั้นไว้เป็นลักษณะดังรูปที่ 3.17

รูปที่ 3.17 ลักษณะของหัวของไฟล์ไบนารี

จะเห็นว่าในหัวของไฟล์ไบนารีที่สร้างขึ้นจะทำการบอกตำแหน่งของไฟล์แต่ละไฟล์ที่ทำการบันทึกว่าอยู่ที่ตำแหน่งใดของไฟล์ไบนารีนี้ โดยการอ้างถึงแอดเดรสนี้จะใช้ข้อมูลทั้งหมด 3 ไบต์ดังนั้นจะสามารถอ้างแอดเดรสได้ถึง 16.7 ล้านแอดเดรส ทั้งนี้เพื่อเป็นการสะดวกเมื่อจะนำไฟล์นี้ไปใช้งานต่อไปคือเราจะทราบได้ว่าไฟล์ใดอยู่ที่ตำแหน่งใดบ้าง นอกจากที่จะบอกแอดเดรสไว้ที่หัวไฟล์แล้วโปรแกรมยังจะสร้างไฟล์รายงานที่บอกตำแหน่งของไฟล์ต่างๆไว้อีกทีหนึ่งด้วย ดังตัวอย่างในรูปที่ 3.18

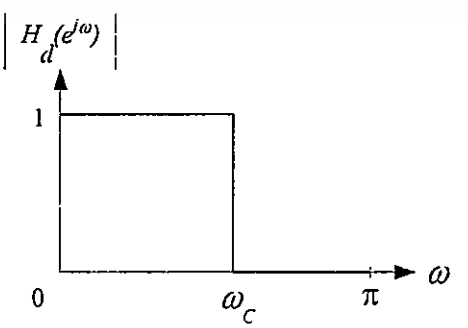
```
1 D:\SOUND\SOUND1\E1-2.DM   0002C1 - 000728
2 D:\SOUND\SOUND1\E1-3.DM   000729 - 000B96
3 D:\SOUND\SOUND1\E1-4.DM   000B97 – 001020
```

รูปที่ 3.18 ตัวอย่างบางส่วนของไฟล์รายงาน

ส่วนอีกเมนูหนึ่งคือ Effect – Filter ใช้สำหรับกรองสัญญาณที่ได้บันทึกไว้แล้วนั้นโดยกระบวนการทางดิจิตอล ซึ่งก็คือใช้วงจรกรองความถี่ต่ำผ่านแบบดิจิตอลที่มีความถี่ตัดต่างๆนั่นเอง สำหรับฟังก์ชันถ่ายโอนของวงจรกรองความถี่ต่ำผ่านในอุดมคติแบบดิจิตอลคือ

$$H_d\left(e^{j\omega}\right)=\begin{cases}1\cdot e^{-j\theta\omega} & ;|\omega|\le\omega_C\\ 0 & ;\omega_C<|\omega|\le\pi\end{cases} \qquad (3.5)$$

จากสมการที่ (3.5) ค่า θ คือค่าคงที่ใดๆและ ω_c คือความถี่ตัดที่ Normalized แล้ว (Normalized ด้วยความถี่ที่ใช้ในการสุ่ม ω_s) ผลตอบสนองของฟังก์ชันถ่ายโอนนี้จะได้ผลตอบสนองเฟสที่เป็นเชิงเส้นซึ่งจะทำให้สัญญาณที่ขาออกมีการหน่วงเวลาที่คงที่ตลอดทุกความถี่ซึ่งก็คือจะทำให้ไม่เกิดความเพี้ยนของสัญญาณที่เกิดจากเฟส โดยความผิดเพี้ยนของเฟสนี้จะมีผลในงานหลายๆด้านเช่นการสื่อสารข้อมูลและเสียงพูดหรือเสียงดนตรีต่างๆ ผลตอบสนองขนาดของสมการที่ (3.5) แสดงดังรูปที่ 3.19



รูปที่ 3.19 ผลตอบสนองความถี่ของวงจรกรองความถี่ต่ำผ่านแบบดิจิตอลในอุดมคติ

เราสามารถหาผลตอบสนองสัญญาณอิมพัลส์จากสมการที่ (3.5) โดยการแปลงกลับแบบฟูเรียร์ จะได้ผลตอบสนองสัญญาณอิมพัลส์ของวงจรกรองความถี่ต่ำผ่านในอุดมคติดังสมการที่ (3.6)

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega}) \cdot e^{j\omega n} d\omega$$

$$= \frac{1}{2\pi}\int_{-\omega_C}^{\omega_C} (1) \cdot e^{j\omega n} d\omega$$

$$= \left[\frac{e^{j\omega n}}{2\pi jn}\right]_{-\omega_C}^{\omega_C} = \frac{e^{j\omega_C n} - e^{-j\omega_C n}}{2\pi jn}$$

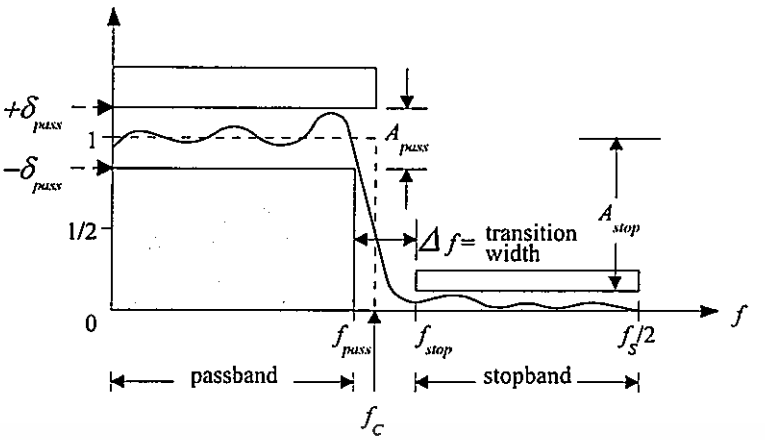
$$h_d(n) = \frac{\sin(\omega_C n)}{\pi n} \qquad \text{เมื่อ } -\infty \le n \le +\infty \qquad (3.6)$$

จะได้ผลตอบสนองสัญญาณอิมพัลส์แบบ 2 ด้าน คือ n มีค่าตั้งแต่ $-\infty$ ถึง $+\infty$ ซึ่งใช้จริงไม่ได้ในทางปฏิบัติ แต่สามารถแก้ไขได้โดยใช้วิธีวินโดว์(window) ได้ตามขั้นตอนดังนี้

กำหนดค่าคุณสมบัติของวงจรกรองตามที่ต้องการได้แก่ค่า δ_{pass} ค่า A_{stop} และค่า Δf เป็นต้น (ความหมายของค่าต่างๆให้ดูรูปที่ 3.20 ประกอบ) จากนั้นเลือกชนิดของวินโดว์ตามตารางที่ 3.3 เพื่อให้สอดคล้องกับค่าต่างๆที่ต้องการ รวมทั้งคำนวณหาค่าอันดับของวงจรกรอง(M)ที่จะใช้ เมื่อเลือกได้แล้วก็ทำการคำนวณหาค่าวินโดว์($w(n)$)จากตารางจะได้ค่า$w(n)$ ที่ n ตั้งแต่ 0ถึง M-1

ตารางที่ 3.3 แสดงวินโดว์ชนิดต่างๆและคุณสมบัติที่สำคัญ

Window	δ_{pass}	A_{stop} (dB)	Δf (normalized)	$w(n)$; n=0,1,…,M-1
Rectangular	8.9%	21	$2/M$	1
Hanning	0.63%	44	$4/M$	$0.5 - 0.5\cos\left(\frac{2\pi n}{M-1}\right)$
Hamming	0.22%	53	$4/M$	$0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right)$
Blackman	0.02%	74	$6/M$	$0.42 - 0.5\cos\left(\frac{2\pi n}{M-1}\right) + 0.08\cos\left(\frac{4\pi n}{M-1}\right)$



รูปที่ 3.20 แสดงค่าต่างๆในการกำหนดคุณสมบัติของวงจรกรอง

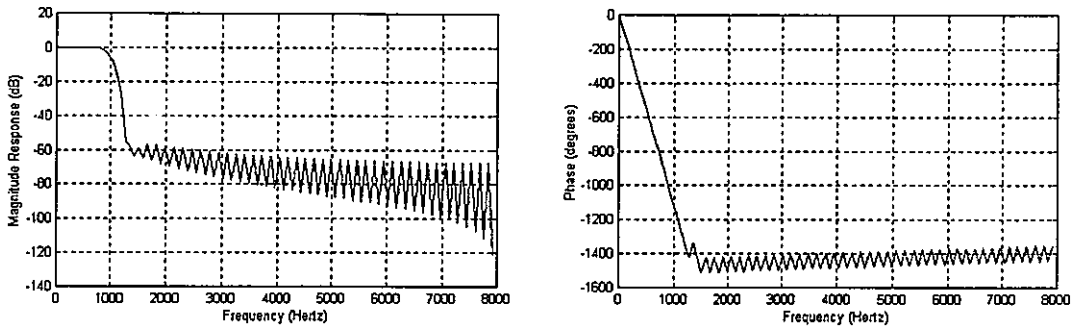
จากนั้นทำการหาผลตอบสนองสัญญาณอิมพัลส์โดยการคูณวินโดว์เข้ากับผลตอบสนองสัญญาณอินพัลส์ในอุดมคติที่มีการเลื่อนล้าหลังไปเท่ากับ α โดยที่ $\alpha = (M\text{-}1)/2$ แบบจุดต่อจุดดังสมการที่ (3.7)

$$h(n) = h_d(n-\alpha)\cdot w(n) \qquad \text{เมื่อ } n = 1,2,\ldots,M\text{-}1 \tag{3.7}$$

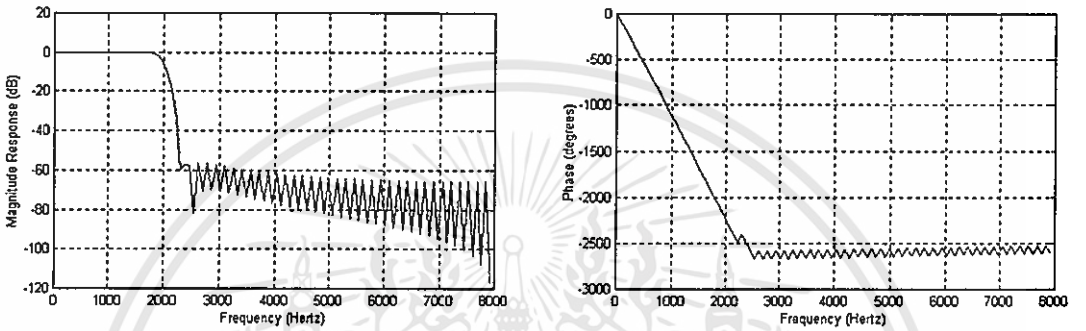
จะได้ผลตอบสนองสัญญาณอิมพัลส์ของวงจรกรองความถี่ต่ำผ่านที่มีคุณสมบัติตามที่ต้องการ เมื่อจะนำไปใช้ก็จะนำผลตอบสนองสัญญาณอิมพัลส์นี้ไปทำการคอนโวลูชันกับสัญญาณที่ต้องการดังสมการที่ (3.8)

$$y(n) = h(n) * x(n) = \sum_{k=-\infty}^{\infty} h(k)\cdot x(n-k) \tag{3.8}$$

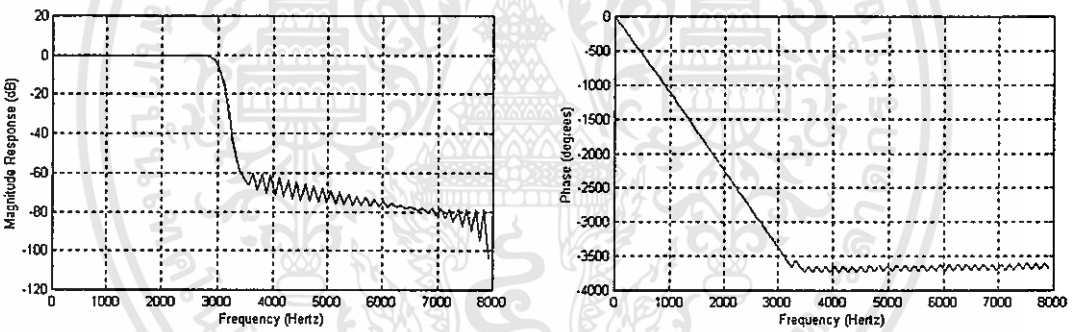
ในวิทยานิพนธ์นี้ได้ทำการเลือกวินโดว์แบบ Hamming โดยกำหนด M = 101 ซึ่งจะได้ค่า δ_{pass} = 0.22%, A_{stop} = 53dB และ $\Delta f = \frac{4}{M}$ = 0.04 และนำไปคำนวณหาค่าวินโดว์และผลตอบสนองสัญญาณอิมพัลส์ นำผลตอบสนองสัญญาณอิมพัลส์ที่ได้ออกแบบไว้ไปวาดกราฟของผลตอบสนองความถี่โดยโปรแกรม MATLAB จะได้ดังรูปที่ 3.21 ซึ่งเป็นการออกแบบวงจรกรองความถี่ต่ำผ่านที่มีความถี่ตัดเท่ากับ 1kHz, 2kHz, 3kHz และ 4kHz ตามลำดับ เพื่อเป็นการทดสอบการทำงานของวงจรกรองแบบดิจิตอลที่ความถี่ต่างๆ



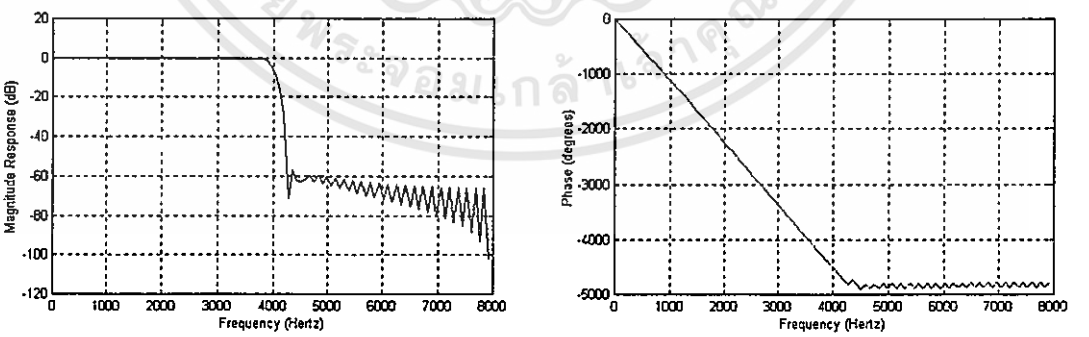
(ก) ความถี่ตัดเท่ากับ 1 kHz



(ข) ความถี่ตัดเท่ากับ 2 kHz



(ค) ความถี่ตัดเท่ากับ 3 kHz



(ง) ความถี่ตัดเท่ากับ 4 kHz

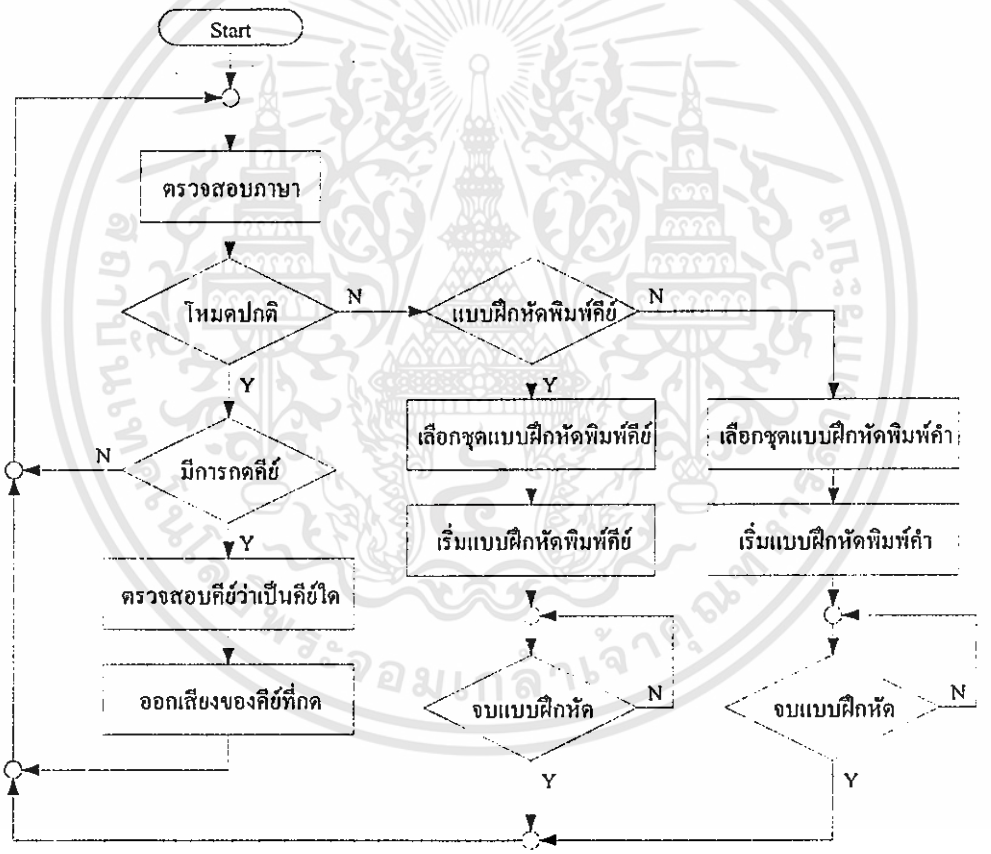
รูปที่ 3.21 ผลตอบสนองความถี่ของวงจรกรองแบบดิจิตอลที่ใช้งานจริง

3.2.2 ซอฟแวร์ของส่วนชุดฝึกพิมพ์

ในส่วนซอฟแวร์ของชุดฝึกพิมพ์จะแบ่งการทำงานเป็น 3 โหมดการทำงานใหญ่ดังนี้คือ

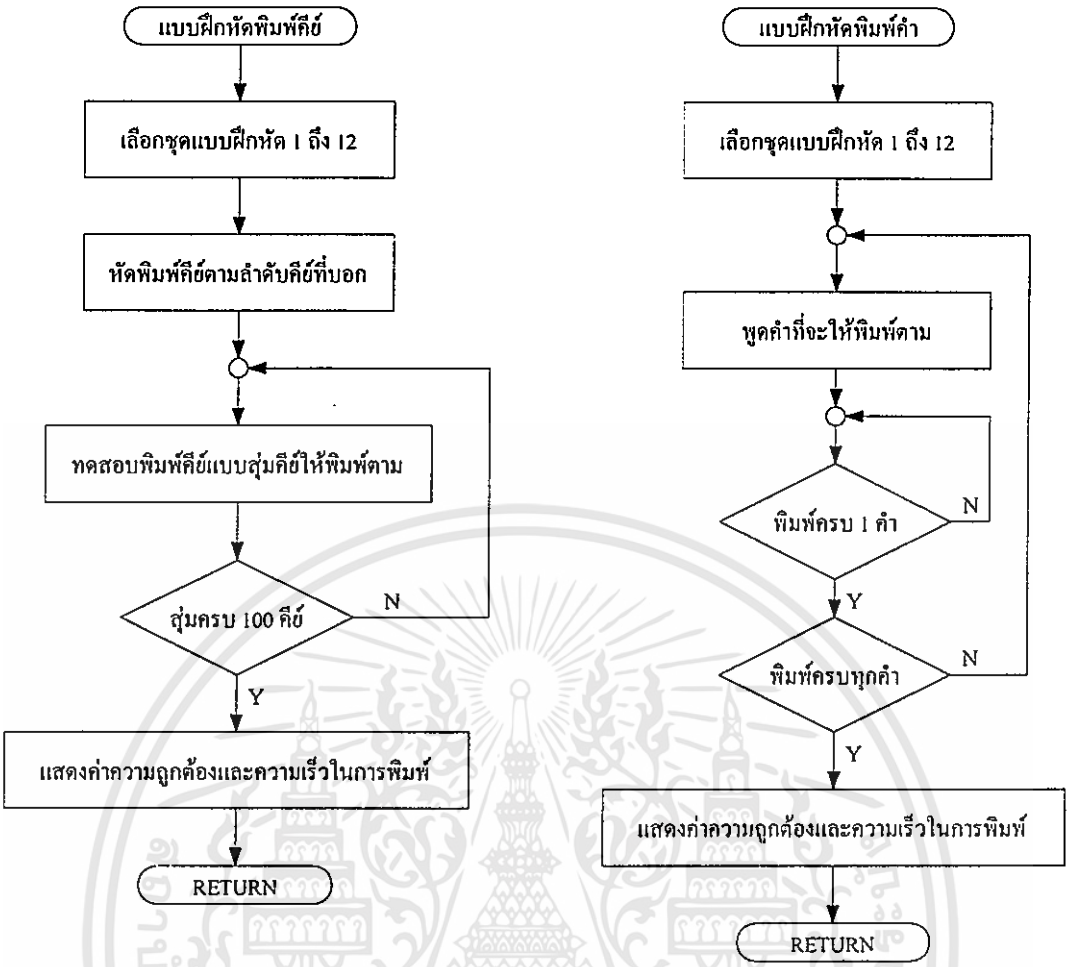
- **โหมดปกติ** เป็นโหมดที่ผู้ใช้กดคีย์ใดๆก็จะมีเสียงบอกผู้ใช้ว่าได้กดคีย์ใด
- **โหมดแบบฝึกหัดพิมพ์คีย์** เป็นโหมดที่ใช้ทดสอบความสามารถในการพิมพ์โดยที่จะมีเสียงบอกคีย์และให้ผู้ใช้กดคีย์ตามเสียงที่ได้ยินนั้น เมื่อจบแบบฝึกหัดก็จะมีเสียงบอกความถูกต้องและความเร็วในการพิมพ์ด้วย
- **โหมดแบบฝึกหัดพิมพ์คำ** เป็นโหมดที่ใช้ทดสอบความสามารถในการพิมพ์เช่นเดียวกันแต่จะให้พิมพ์เป็นคำโดยจะมีเสียงบอกคำออกมาและให้ผู้ใช้กดคีย์เพื่อสะกดเป็นคำนั้น เมื่อจบแบบฝึกหัดก็จะมีเสียงบอกความถูกต้องและความเร็วในการพิมพ์เช่นกัน

การทำงานโดยรวมของซอฟแวร์ทั้งหมดสามารถเขียนเป็นแผนผังได้ดังรูปที่ 3.22



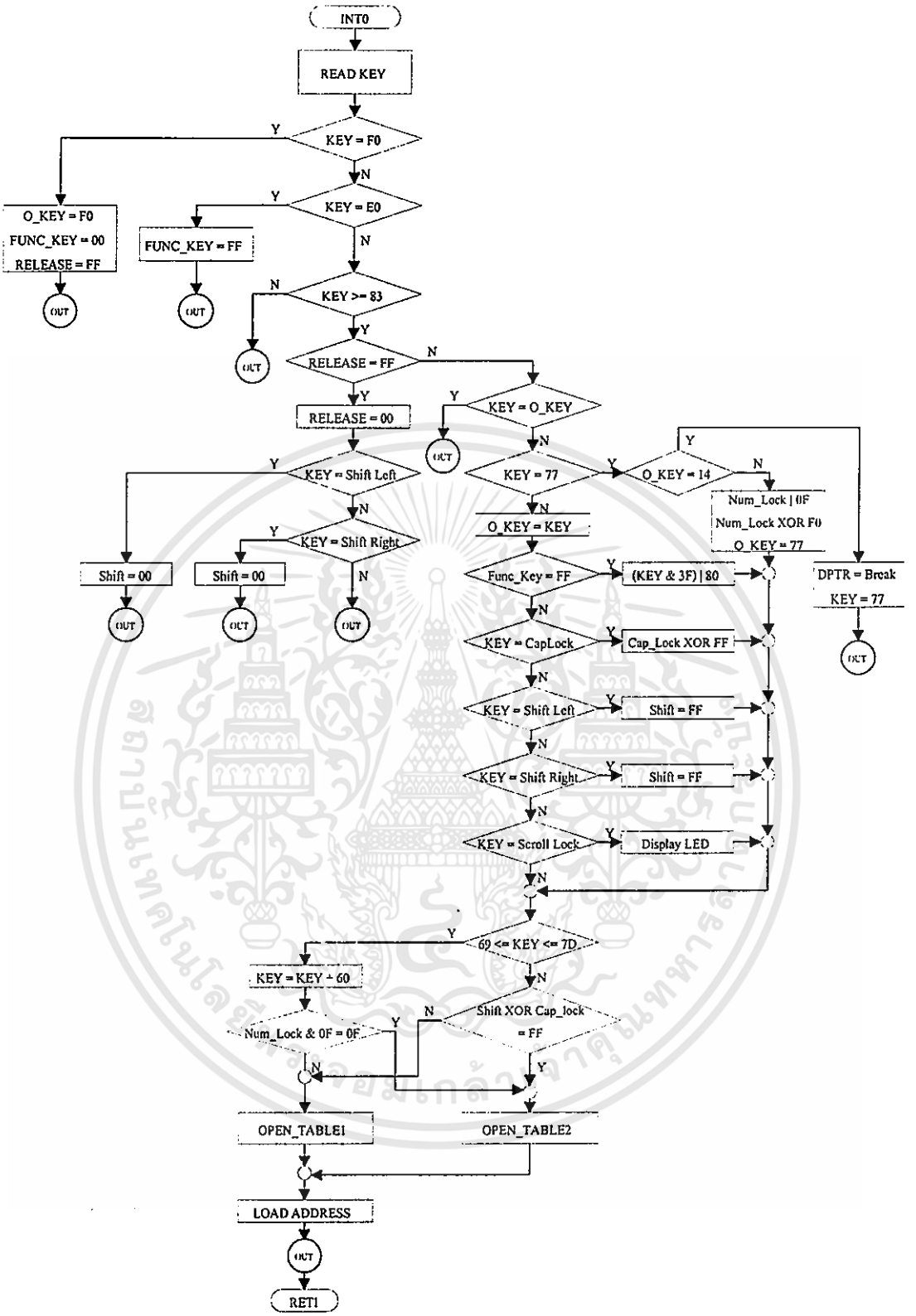
รูปที่ 3.22 แผนผังการทำงานของซอฟแวร์ส่วนชุดฝึกพิมพ์

เมื่อเริ่มการทำงานจะทำการตรวจสอบก่อนว่าเป็นภาษาไทยหรือภาษาอังกฤษ หลังจากนั้นจะตรวจสอบว่าเป็นโหมดปกติหรือไม่ ถ้าใช่จะตรวจสอบว่าถ้ามีการกดคีย์ก็จะทำการพูดคีย์นั้นออกมา แต่ถ้าเป็นโหมดแบบฝึกหัดก็จะให้เลือกชุดของแบบฝึกหัดก่อนซึ่งมีทั้งหมด 12 ชุดโดยการกดคีย์ F1 – F12 เมื่อจบแบบฝึกหัดก็จะกลับไปยังจุดเริ่มต้นอีกครั้ง



รูปที่ 3.23 แผนผังการทำงานของแบบฝึกหัดพิมพ์คำและแบบฝึกหัดพิมพ์คีย์

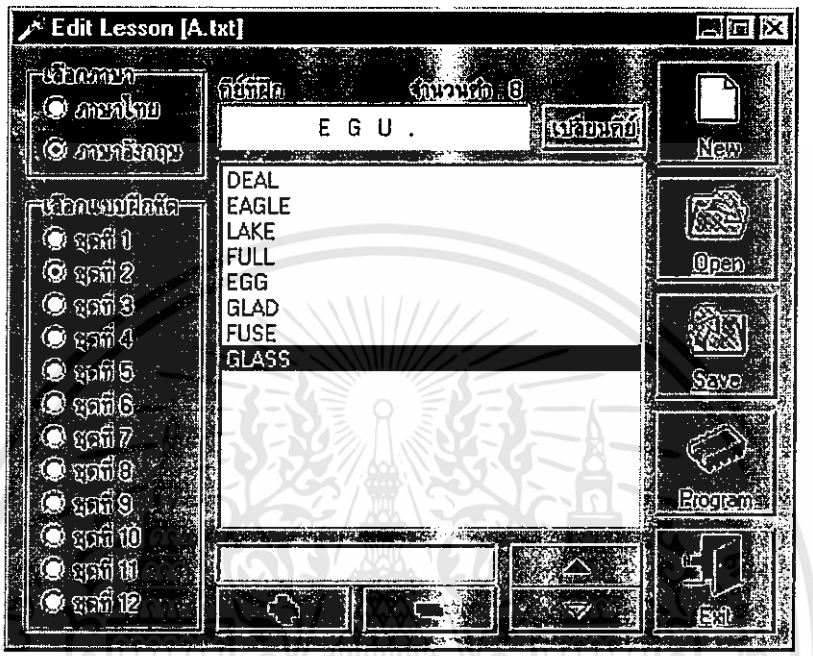
ในส่วนของการตรวจสอบคีย์เมื่อมีการกดคีย์ใดๆแล้วจะมีการอินเทอร์รัปท์เข้าทาง INT0 ของไมโครคอนโทรลเลอร์ ขั้นแรกจะทำการตรวจสอบก่อนว่าจะเป็นคีย์ที่นำหน้าด้วย E0 หรือ F0 หรือไม่ ถ้านำด้วย F0 หมายถึงมีการปล่อยคีย์ก็จะไม่นำคีย์ต่อไปมาคิด หรือถ้านำด้วย E0 หมายถึงว่าเป็นคีย์ที่แสดงด้วยตัวเลขจำนวน 2 ไบต์ก็จะทำการตรวจสอบค่าไบต์ต่อไปอีกครั้ง หลังจากนี้ก็เป็นการตรวจสอบคีย์พิเศษต่างๆก่อนเช่นเป็นการกดคีย์ Shift ค้างไว้หรือไม่ คีย์ Caps Lock หรือ Num Lock ได้ถูกกดอยู่หรือไม่ เพราะว่าคีย์ Shift และคีย์ Caps Lock จะมีผลต่อการกดคีย์อักษร และคีย์ Num Lock มีผลต่อการกดคีย์ตัวเลขต่างๆ เมื่อตรวจสอบสถานะของคีย์พิเศษต่างๆได้แล้วก็จะทราบได้ว่าคีย์ที่ได้กดนั้นเป็นคีย์ใดก็ทำการเปิดตารางเพื่อนำค่าแอดเดรสของสัญญาณเสียงที่ต้องการพูดออกมา ซึ่งแผนผังการทำงานของการตรวจสอบคีย์นี้แสดงได้ดังรูปที่ 3.24



รูปที่ 3.24 แผนผังการทำงานของการตรวจสอบคีย์

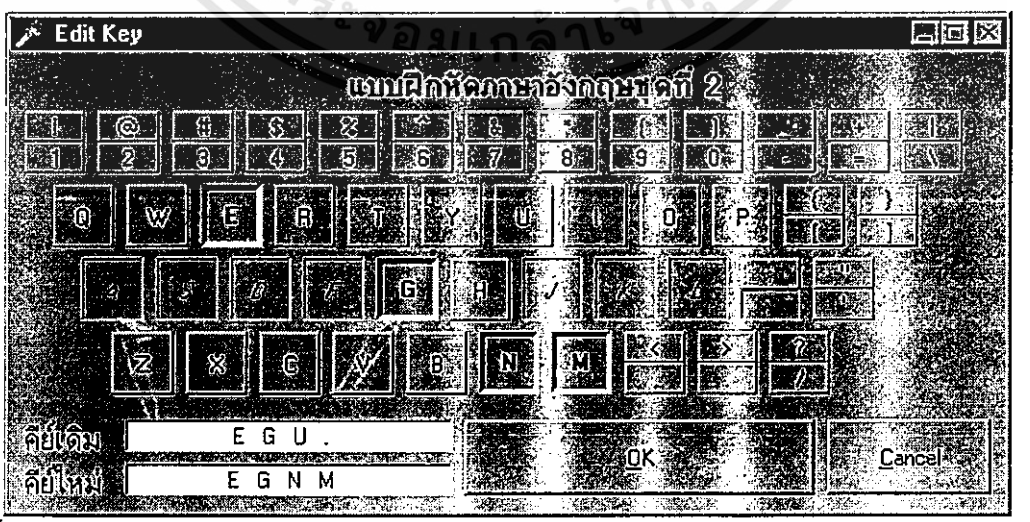
3.2.3 ซอฟต์แวร์ของส่วนแก้ไขแบบฝึกหัด

ในส่วนของการแก้ไขแบบฝึกหัดสามารถทำได้โดยซอฟต์แวร์บนคอมพิวเตอร์และดาวน์-โหลดข้อมูลของแบบฝึกหัดผ่านทางพอร์ตอนุกรมไปบนชุดฝึกพิมพ์ โดยลักษณะของโปรแกรมแก้ไขแบบฝึกหัดเป็นดังรูปที่ 3.25



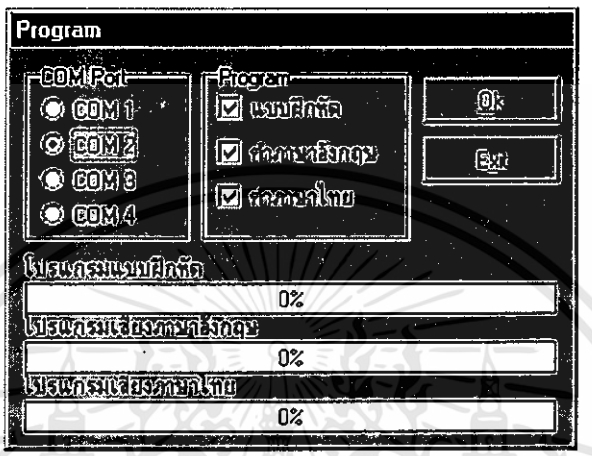
รูปที่ 3.25 หน้าจอของโปรแกรมแก้ไขแบบฝึกหัด

แบบฝึกหัดทั้งหมดจะมีทั้งสิ้นอย่างละ 12 ชุด แบ่งเป็นแบบฝึกหัดภาษาไทยและภาษาอังกฤษ แบบฝึกหัดพิมพ์คำและพิมพ์คีย์ ในส่วนของแบบฝึกหัดพิมพ์คีย์สามารถเปลี่ยนคีย์ที่จะฝึกได้โดยการกดที่ปุ่ม "เปลี่ยนคีย์" จะได้หน้าต่างดังรูปที่ 3.26



รูปที่ 3.26 หน้าต่างของการเปลี่ยนคีย์

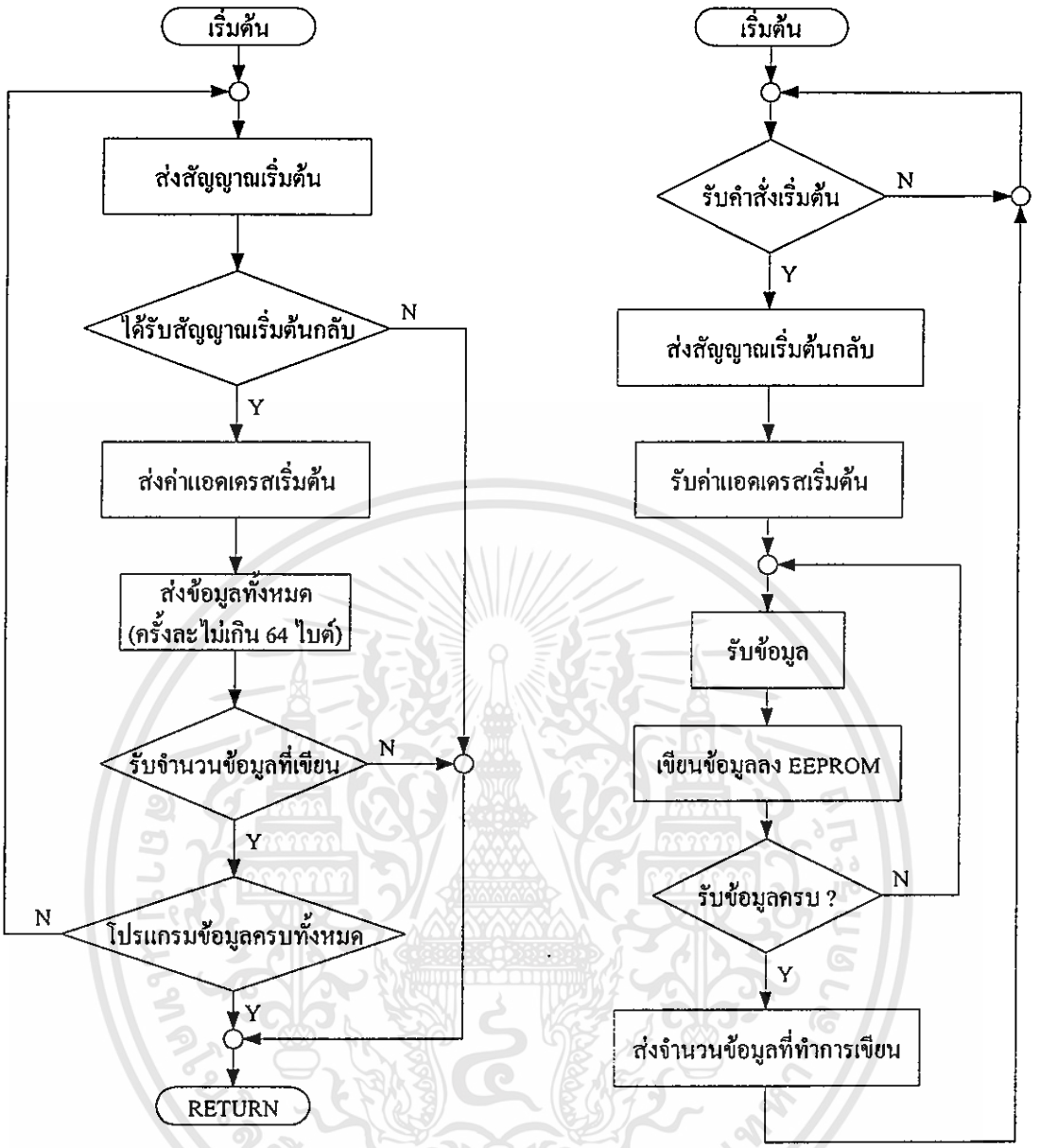
จากรูปที่ 3.26 เราสามารถเปลี่ยนคีย์ที่จะฝึกในแต่ละชุดของแบบฝึกหัดพิมพ์คีย์ได้โดยการกดที่คีย์ที่ต้องการเท่านั้นเอง ส่วนในส่วนของแบบฝึกหัดพิมพ์คำสามารถเปลี่ยนได้จากในหน้าต่างรูปที่ 3.25 โดยการเติมคำที่ต้องการลงในช่องว่างและกดปุ่ม "+" เป็นการเติมคำ หรือเลือกที่คำและกดปุ่ม "-" เป็นการลบคำที่ไม่ต้องการออก เมื่อแก้ไขได้แบบฝึกหัดตามที่ต้องการแล้วก็กดปุ่ม "Program" เพื่อโปรแกรมลงไปยังชุดฝึกพิมพ์ต่อไป โดยหน้าจอของการโปรแกรมเป็นดังรูปที่ 3.27



รูปที่ 3.27 หน้าจอของการโปรแกรมลงชุดฝึกพิมพ์

การโปรแกรมจะเป็นการโปรแกรมผ่านทางพอร์ตอนุกรมโดยเลือกพอร์ตได้ตามต้องการ โดยที่สิ่งที่จะโปรแกรมลงชุดฝึกพิมพ์มีทั้งสิ้น 3 อย่างด้วยกันคือ แบบฝึกหัด และเสียงของคำที่เป็นภาษาไทยและภาษาอังกฤษ ซึ่งสามารถเลือกได้ว่าจะโปรแกรมสิ่งใดลงไปยังชุดฝึกพิมพ์บ้าง

ขั้นตอนการส่งข้อมูลจากคอมพิวเตอร์ไปยังชุดฝึกพิมพ์และการรับข้อมูลของชุดฝึกพิมพ์เพื่อโปรแกรมลงสู่ EEPROM ภายในชุดฝึกพิมพ์แสดงดังแผนภูมิสายงานรูปที่ 3.28 (ก) และ (ข) ตามลำดับ โดยเมื่อเริ่มสั่งโปรแกรมแบบฝึกหัดและ/หรือเสียงลงบนชุดฝึกพิมพ์ ซอฟต์แวร์บนคอมพิวเตอร์จะทำการส่งสัญญาณเริ่มต้นให้กับชุดฝึกพิมพ์และรอรับสัญญาณเริ่มต้นที่ส่งไปนั้นกลับจากชุดฝึกพิมพ์ จากนั้นซอฟต์แวร์ก็ทำการส่งแอดเดรสให้กับชุดฝึกพิมพ์ตามด้วยข้อมูลที่จะโปรแกรมลงบนชุดฝึกพิมพ์ครั้งละไม่เกิน 64 ไบต์ เมื่อชุดฝึกพิมพ์โปรแกรมเสร็จแต่ละครั้งชุดฝึกพิมพ์ก็จะส่งจำนวนไบต์ที่เขียนลง EEPROM คืนกลับมา ซอฟต์แวร์ก็จะส่งข้อมูลชุดถัดไปเรื่อยๆ จนกระทั่งหมดข้อมูลที่จะเขียนลงสู่ EEPROM บนชุดฝึกพิมพ์ก็เป็นอันเสร็จขั้นตอนการโปรแกรมแบบฝึกหัดลงสู่ชุดฝึกพิมพ์

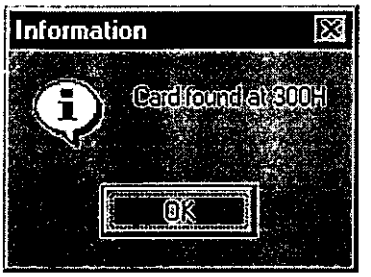


(ก) แผนภูมิสายงานส่วนซอฟต์แวร์บนคอมพิวเตอร์ (ข) แผนภูมิสายงานส่วนชุดฝึกพิมพ์

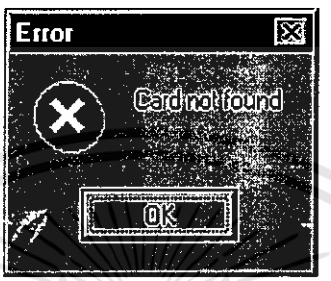
รูปที่ 3.28 แผนภูมิสายงานของการโปรแกรมแบบฝึกหัดลงสู่ชุดฝึกพิมพ์

3.3 การใช้งานโปรแกรมบันทึกเสียง

เมื่อเริ่มเรียกการทำงานของโปรแกรมบันทึกเสียงโปรแกรมจะทำการค้นหาการ์ดบันทึกเสียงว่าอยู่ที่ตำแหน่งแอดเดรสที่เท่าไร และจะแสดงออกมาที่หน้าจอดังรูปที่ 3.29 หรือถ้าไม่มีการ์ดอยู่โปรแกรมก็จะบอกออกมาว่าค้นหาการ์ดไม่พบดังรูปที่ 3.30

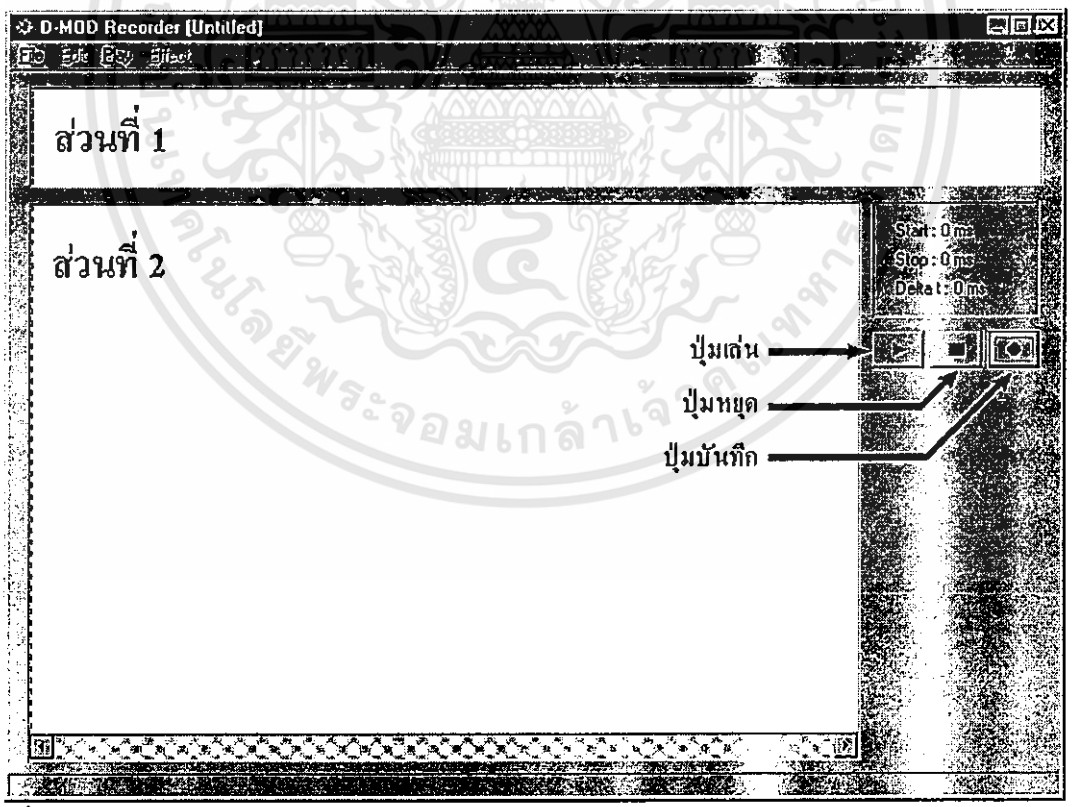


รูปที่ 3.29 หน้าต่างแสดงตำแหน่งของการ์ดที่โปรแกรมค้นพบ



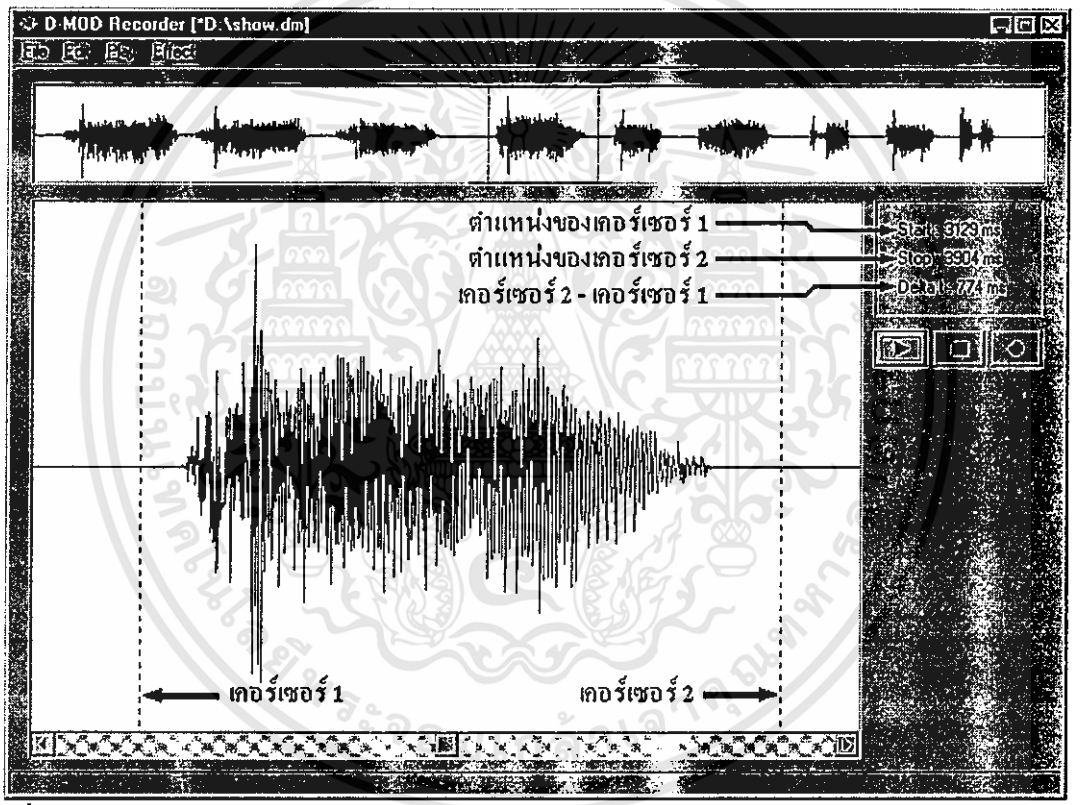
รูปที่ 3.30 หน้าต่างแสดงว่าโปรแกรมค้นหาการ์ดบันทึกเสียงไม่พบ

เมื่อเข้าสู่โปรแกรมแล้ว หน้าต่างหลักของโปรแกรมจะเป็นดังรูปที่ 3.31



รูปที่ 3.31 หน้าต่างหลักของโปรแกรมบันทึกเสียง

หน้าต่างของโปรแกรมจะมี 2 ส่วนด้วยกัน คือส่วนที่ 1 เป็นส่วนที่ใช้แสดงสัญญาณเสียงทั้งหมดที่ได้บันทึกไว้ซึ่งมีความยาวสูงสุดได้ 16 วินาที ในส่วนที่ 2 จะเป็นส่วนย่อยที่แสดงสัญญาณเสียงจากหน้าต่างในส่วนที่ 1 ซึ่งจะมีความยาว 1 วินาที ซึ่งในหน้าต่างส่วนที่ 2 นี้เราสามารถเลื่อนไปยังส่วนต่างๆของสัญญาณเสียงที่ต้องการดูได้โดยการเลื่อน scroll bar ใต้หน้าต่างส่วนที่ 2 หรือสามารถคลิ๊กไปยังหน้าต่างส่วนที่ 1 ไปยังส่วนที่ต้องการเลื่อนไปได้ทันที ซึ่งในรูปที่ 3.31 นี้เป็นการเริ่มต้นเข้าสู่โปรแกรมดังนั้นหน้าต่างทั้ง 2 ส่วนจะยังคงว่างเปล่าอยู่ เมื่อเรากดปุ่มบันทึกโปรแกรมจะทำการเริ่มต้นบันทึกเสียง จนกระทั่งครบ 16 วินาที โปรแกรมจะหยุดการทำงานอัตโนมัติ หรือจนกระทั่งกดปุ่มหยุด โปรแกรมก็จะทำการแสดงสัญญาณเสียงออกมาทางหน้าจอทั้ง 2 ส่วน ดังรูปที่ 3.32



รูปที่ 3.32 หน้าต่างของโปรแกรมบันทึกเสียงเมื่อได้บันทึกเสียงแล้ว

โปรแกรมนี้จะมีเคอร์เซอร์อยู่ 2 เส้นใช้สำหรับเลือกช่วงของสัญญาณคือเคอร์เซอร์ 1 และเคอร์เซอร์ 2 ซึ่งสามารถเลื่อนได้โดยการคลิ๊กลงบนหน้าต่างส่วนที่ 2 โดยการคลิ๊กซ้ายจะหมายถึงการเลื่อนเคอร์เซอร์ที่ 1 และการคลิ๊กขวาจะหมายถึงการเลื่อนเคอร์เซอร์ที่ 2 ซึ่งเคอร์เซอร์ทั้งสองนี้จะแสดงอยู่ในหน้าต่างส่วนที่ 1 ด้วย โดยตำแหน่งของเคอร์เซอร์ทั้งสองนี้จะแสดงอยู่ทางด้านขวามือของหน้าต่างของโปรแกรมด้วย เคอร์เซอร์ทั้งสองนี้ใช้ทำหน้าที่เลือกช่วงข้อมูลของสัญญาณเสียงที่ได้บันทึกไว้ ซึ่งเราสามารถเล่นเสียงในช่วงที่ได้เลือกไว้โดยการกดปุ่มเล่น

สำหรับเมนูต่างๆในโปรแกรมมีการทำงานต่างๆดังนี้

เมนู File

- Open ใช้สำหรับเปิดไฟล์ที่ได้เคยบันทึกไว้
- Save ใช้บันทึกไฟล์ที่ได้เปิดไว้อยู่
- Save All ใช้บันทึกไฟล์ที่ได้เปิดไว้โดยบันทึกช่วงสัญญาณทั้งหมด
- Save Select ใช้บันทึกไฟล์ในช่วงสัญญาณที่เลือกไว้โดยเคอร์เซอร์ทั้งสอง
- Convert File ใช้สำหรับบันทึกไฟล์หลายๆไฟล์รวมกัน
- Exit ออกจากโปรแกรมบันทึกเสียง

เมื่อเข้าสู่เมนู Convert File หน้าต่างของโปรแกรมจะกลายเป็นดังรูปที่ 3.33

รูปที่ 3.33 หน้าต่างเมื่อเข้าสู่เมนู File – Convert File

หน้าต่างทางด้านซ้ายของรูปที่ 3.33 เป็นหน้าต่างที่ใช้สำหรับเลือกไดรฟ์และไดเร็กทอรีที่มีไฟล์ที่ได้บันทึกไว้แล้ว สำหรับการเลือกไฟล์นั้นทำได้โดยการดับเบิ้ลคลิ๊กที่ไฟล์ที่ต้องการเลือก หรือคลิ๊กที่ชื่อไฟล์แล้วกดปุ่ม Add ซึ่งไฟล์ที่ได้เลือกไว้จะไปปรากฏอยู่ที่หน้าต่างทางด้านขวา การเอาไฟล์ที่ได้เลือกไว้นี้ออกไปนั้นทำได้โดยการดับเบิ้ลคลิ๊กที่ไฟล์ที่ต้องการเอาออก หรือคลิ๊กที่ชื่อไฟล์ที่ต้องการเอาออกแล้วกดปุ่ม Remove ไฟล์ที่ได้เลือกไว้จะมีลำดับของแต่ละไฟล์บอกไว้ซึ่งเราสามารถเลื่อนลำดับของไฟล์ขึ้นหรือลงได้โดยการกดปุ่มลูกศรขึ้นหรือลง ในหน้าต่างการเลือกไฟล์นี้สามารถเลือกไฟล์ได้ไม่เกิน 256 ไฟล์ หรือไม่เกิน 1 เมกกะไบต์ โดยด้านบนของหน้าต่างจะ

แสดงจำนวนและขนาดของไฟล์ทั้งหมดที่ได้เลือกไว้ด้วย สำหรับปุ่มต่างๆทางด้านขวามีการทำงานต่างๆดังนี้

- Open เปิดไฟล์ของรายชื่อไฟล์ที่บันทึกไว้
- Save Bin บันทึกไฟล์ที่ได้เลือกไว้ทั้งหมดเป็นไฟล์ไบนารีไฟล์เดียว
- Save List บันทึกรายชื่อไฟล์ที่ได้เลือกไว้
- Clear ลบรายชื่อไฟล์ทั้งหมดที่ได้เลือกไว้
- Exit ออกจากเมนู Convert File

เมนู Edit

- Cut ตัดข้อมูลเสียงส่วนที่เลือกไว้ออก
- Paste From File ต่อข้อมูลเสียงจากไฟล์ที่เลือกไปที่ท้ายไฟล์ที่เปิดอยู่
- Insert แทรกข้อมูลเสียงจากไฟล์ที่เลือกไปที่ตำแหน่งของเคอร์เซอร์ 1
- Select All เลื่อนเคอร์เซอร์ 1 และ 2 ไปที่ตำแหน่งเริ่มต้นและสุดท้ายของข้อมูลเสียงตามลำดับ

เมนู Play

- Record เริ่มบันทึกเสียง (หรือกดปุ่มบันทึก)
- Play เล่นเสียงในช่วงที่เลือกไว้ (หรือกดปุ่มเล่น)
- Play All เล่นเสียงในช่วงทั้งหมด
- Stop หยุดบันทึกหรือหยุดเล่นเสียง (หรือกดปุ่มหยุด)

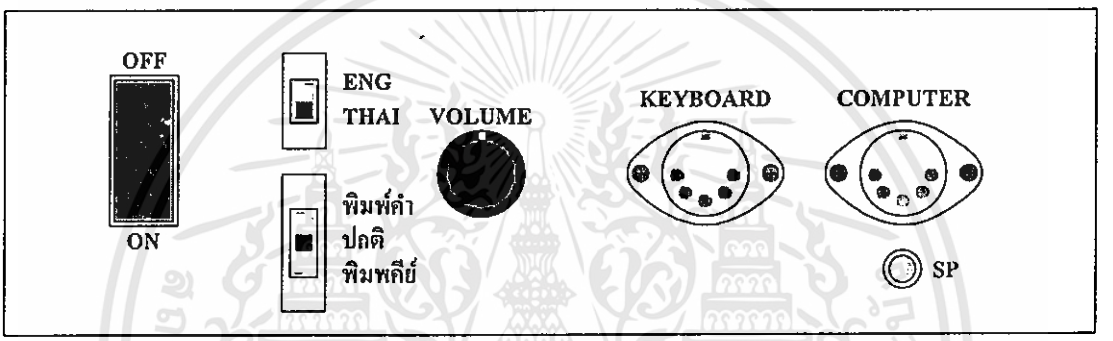
เมนู Effect

- Mute ทำสัญญาณเงียบในช่วงที่เลือกไว้
- Filter การกรองสัญญาณที่ได้บันทึกไว้ทั้งหมดด้วยวิธีทางดิจิตอลทางซอฟท์แวร์โดยเป็นการกรองแบบความถี่ต่ำผ่านที่ความถี่ 1kHz, 2kHz, 3kHz และ 4kHz

3.4 การใช้งานชุดฝึกพิมพ์

ชุดฝึกพิมพ์มีโหมดการทำงานอยู่ 3 โหมดด้วยกันคือโหมดปกติ โหมดแบบฝึกหัดพิมพ์คีย์ และโหมดแบบฝึกหัดพิมพ์คำโดยแต่ละโหมดมีการทำงานดังนี้

- **โหมดปกติ** เป็นโหมดที่เมื่อผู้ใช้พิมพ์คีย์ใดๆจะมีเสียงบอกตามคีย์ที่กดทั้งภาษาไทยและภาษาอังกฤษ
- **โหมดแบบฝึกหัดพิมพ์คีย์** เป็นโหมดหัดพิมพ์แบบทีละคีย์ คือจะมีเสียงพูดคีย์ต่างๆออกมาและให้พิมพ์ตามที่บอก เช่นชุดฝึกพิมพ์พูดคำว่า "กอ-ไก่" ก็ให้พิมพ์คีย์ "ก.ไก่" เป็นต้น เมื่อจบแบบฝึกหัดก็จะมีเสียงบอกความถูกต้องและความเร็วในการพิมพ์ด้วย
- **โหมดแบบฝึกหัดพิมพ์คำ** เป็นโหมดหัดพิมพ์แบบทีละหลายๆคีย์ คือจะมีเสียงพูดคำต่างๆออกมาและให้พิมพ์ตามคำที่บอก เช่นชุดฝึกพิมพ์พูดคำว่า "ดาว" ก็ให้พิมพ์คีย์ "ด" , "า" และ "ว" ตามลำดับเป็นต้น เมื่อจบแบบฝึกหัดก็จะมีเสียงบอกความถูกต้องและความเร็วในการพิมพ์ด้วยเช่นกัน



รูปที่ 3.34 หน้าปัดของชุดฝึกพิมพ์สำหรับคนตาบอด

บนชุดฝึกพิมพ์จะมีสวิตซ์อยู่ 2 สวิตซ์ คือสวิตซ์สำหรับเลือกภาษาไทยหรือภาษาอังกฤษ โดยภาษาที่เลือกนั้นจะใช้ในโหมดการทำงานทั้ง 3 ของชุดฝึกพิมพ์ และอีกหนึ่งสวิตซ์สำหรับเลือกโหมดการทำงาน

ถ้าเลือกการทำงานไปยังโหมดแบบฝึกหัดพิมพ์คีย์จะเป็นการฝึกพิมพ์คีย์ตามภาษาที่ได้เลือกไว้ซึ่งแต่ละภาษายังมีแบบฝึกหัดย่อยๆอีกภาษาละ 12 ชุดแบบฝึกหัด ซึ่งชุดฝึกพิมพ์จะบอกให้กดคีย์ F1 ถึง F12 เพื่อเลือกชุดแบบฝึกหัดย่อยๆนั้น แบบฝึกหัดพิมพ์คีย์ที่ออกแบบไว้แต่ละชุดนั้นแสดงดังตารางที่ 3.4

ตารางที่ 3.4 แบบฝึกหัดพิมพ์คีย์

ชุดที่	คีย์ที่ใช้ฝึกภาษาไทย	คีย์ที่ใช้ฝึกภาษาอังกฤษ
1	ฟ ห ก ด ่ า ส ว	A S D F J K L ;
2	เ ้ ง	E G U .
3	พ ะ ั ี ่ ำ ร	R H O ,
4	อ ิ ท ื แ ม	T C I M :
5	ไ ป น ใ ๆ ผ ย	W V Y P
6	บ ล ฝ	Q N X /
7	ถ ภ ค ต ็ ๊ ๋	B ? Z -
8	จ ข ช - /	ไม่มี
9	โ ฌ ็ ๊ ๋ ฑ ธ ์ ฌ	ไม่มี
10	ฎ ฏ ษ ฆ " ศ ฯ	ไม่มี
11	ฮ ฉ ฒ - ฿ ์ ?	ไม่มี
12	ฤ ( ) ฬ ฐ ญ ฦ ฐ	ไม่มี

แบบฝึกหัดชุดแรกของแต่ละภาษาจะเป็นคีย์พื้นฐานที่ใช้ในการฝึกพิมพ์ดีดโดยปกติคือเป็นคีย์ที่ใช้วางนิ้วทั้งหมดที่เรียกว่าเป็นแป้นเหย้า ซึ่งขั้นแรกชุดฝึกพิมพ์จะบอกคีย์ตามลำดับให้กดคีย์ตามที่เครื่องบอกเป็นการหัดพิมพ์ก่อน และเมื่อบอกจบจำนวน 5 รอบแล้วชุดฝึกพิมพ์จะเริ่มเข้าสู่แบบฝึกหัดคือจะบอกคีย์ที่ได้ฝึกไปแต่จะไม่ได้เรียงตามลำดับเป็นจำนวน 100 คีย์ เมื่อเราพิมพ์ตามจนครบชุดฝึกพิมพ์จะบอกเปอร์เซนต์ความถูกต้องในการพิมพ์และความเร็วในการพิมพ์เป็นคำต่อนาทีด้วย ส่วนแบบฝึกหัดตั้งแต่ชุดที่ 2 เป็นต้นไปจะเป็นการเริ่มฝึกคีย์ที่ไม่ใช่แป้นเหย้าซึ่งเมื่อเข้าสู่ชุดแบบฝึกหัดเหล่านี้เริ่มแรกชุดฝึกพิมพ์จะเริ่มฝึกให้ก้าวนิ้วจากแป้นเหย้าไปยังคีย์ที่ต้องการเช่นคีย์ "สระเอ" เป็นคีย์ที่ต้องใช้นิ้วชี้ก้าวไปจากคีย์ "ด.เด็ก" ชุดฝึกพิมพ์ก็จะให้พิมพ์คีย์ "ด.เด็ก" ก่อนที่จะพิมพ์คีย์ "สระเอ" ดังนี้ "ด ด ด เ" จำนวน 5 รอบก่อนที่จะไปฝึกคีย์ถัดไป เมื่อฝึกพิมพ์จนครบทุกคีย์แล้วก็จะเริ่มเข้าสู่แบบฝึกหัดเช่นเดียวกับแบบฝึกหัดชุดแรก โดยคีย์ที่จะสุ่มมาให้ฝึกพิมพ์จะเป็นคีย์ที่ใช้ฝึกมาตั้งแต่แบบฝึกหัดแรกจนถึงแบบฝึกหัดปัจจุบัน เช่นเมื่อฝึกพิมพ์อยู่ที่ชุดที่ 5 แบบฝึกหัดจะเป็นการสุ่มคีย์ตั้งแต่แบบฝึกหัดชุดที่ 1 ถึงแบบฝึกหัดชุดที่ 5 ซึ่งเมื่อเราพิมพ์ตามจนครบแล้วชุดฝึกพิมพ์จะบอกเปอร์เซนต์ความถูกต้องในการพิมพ์และความเร็วในการพิมพ์เช่นเดียวกัน

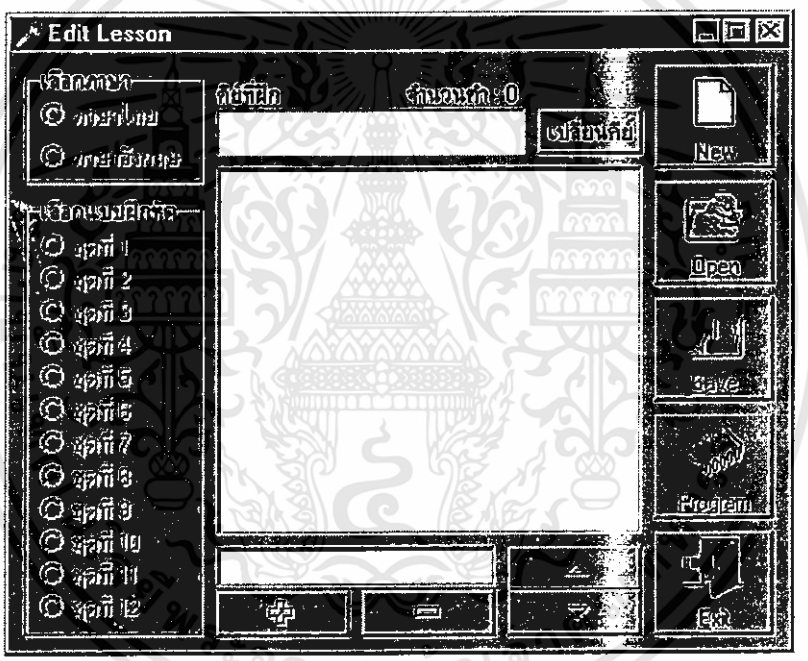
ในส่วนของแบบฝึกหัดพิมพ์คำนั้นจะเป็นการพิมพ์ผสมเป็นคำคือชุดฝึกพิมพ์จะบอกเป็นคำออกมาเช่นคำว่า "ดาว" เราก็พิมพ์ "ด" "า" และ "ว" ตามลำดับ ซึ่งแต่ละภาษาก็จะมีแบบฝึกหัดพิมพ์คำย่อยๆ 12 ชุดแบบฝึกหัดซึ่งสามารถเลือกได้โดยการกดคีย์ F1 ถึง F12 เช่นเดียวกันกับแบบ

ฝึกหัดพิมพ์คีย์ ซึ่งคำที่จะใช้ฝึกพิมพ์นี้ได้เลือกมาให้สอดคล้องกับแบบฝึกหัดพิมพ์คีย์ด้วยยกตัวอย่างเช่นแบบฝึกหัดพิมพ์คำในชุดที่ 6 จะมีคำที่ผสมกันของคีย์ที่ได้ฝึกในแบบฝึกหัดพิมพ์คีย์ชุดที่ 1 ถึงชุดที่ 6 เท่านั้น และเมื่อเราพิมพ์ตามจนครบทุกคำแล้วชุดฝึกพิมพ์จะบอกเปอร์เซนต์ความถูกต้องในการพิมพ์และความเร็วในการพิมพ์เช่นกัน

แบบฝึกหัดพิมพ์คีย์และแบบฝึกหัดพิมพ์คำนี้สามารถแก้ไขเปลี่ยนแปลงได้ด้วยตนเองด้วยซอฟต์แวร์บนคอมพิวเตอร์ซึ่งจะได้กล่าวต่อไปในหัวข้อ 3.5

3.5 การใช้งานโปรแกรมแก้ไขแบบฝึกหัดของชุดฝึกพิมพ์

แบบฝึกหัดบนชุดฝึกพิมพ์ทั้งแบบฝึกหัดพิมพ์คำและแบบฝึกหัดพิมพ์คีย์สามารถทำการแก้ไขได้โดยซอฟท์แวร์บนคอมพิวเตอร์โดยเมื่อเรียกโปรแกรมจะมีหน้าต่างดังรูปที่ 3.35

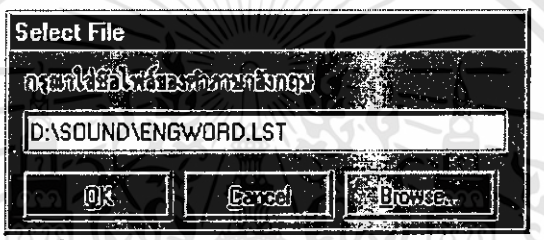


รูปที่ 3.35 หน้าต่างของโปรแกรมแก้ไขแบบฝึกหัด

เมื่อเข้าสู่โปรแกรมเราสามารถเลือกไฟล์ข้อมูลของแบบฝึกหัดที่เคยบันทึกไว้มาแก้ไขใหม่ได้โดยการกดปุ่ม Open หรือถ้าต้องการสร้างไฟล์ข้อมูลของแบบฝึกหัดขึ้นมาใหม่สามารถทำได้โดยการกดปุ่ม New จะได้หน้าต่างดังรูปที่ 3.36

รูปที่ 3.36 หน้าต่างเมื่อกดปุ่ม New

หน้าต่างในรูปที่ 3.36 จะเป็นการให้เลือกไฟล์ที่มีนามสกุลเป็น LST ที่ได้จากการบันทึกเสียงโดยโปรแกรมบันทึกเสียง โดยไฟล์ดังกล่าวจะต้องเป็นไฟล์ของคำที่จะใช้เป็นแบบฝึกหัดภาษาไทย เมื่อใส่ชื่อไฟล์ของคำภาษาไทยแล้วโปรแกรมจะถามชื่อไฟล์ที่มีนามสกุล LST ของไฟล์ของคำที่จะใช้เป็นแบบฝึกหัดภาษาอังกฤษต่อไปดังรูปที่ 3.37



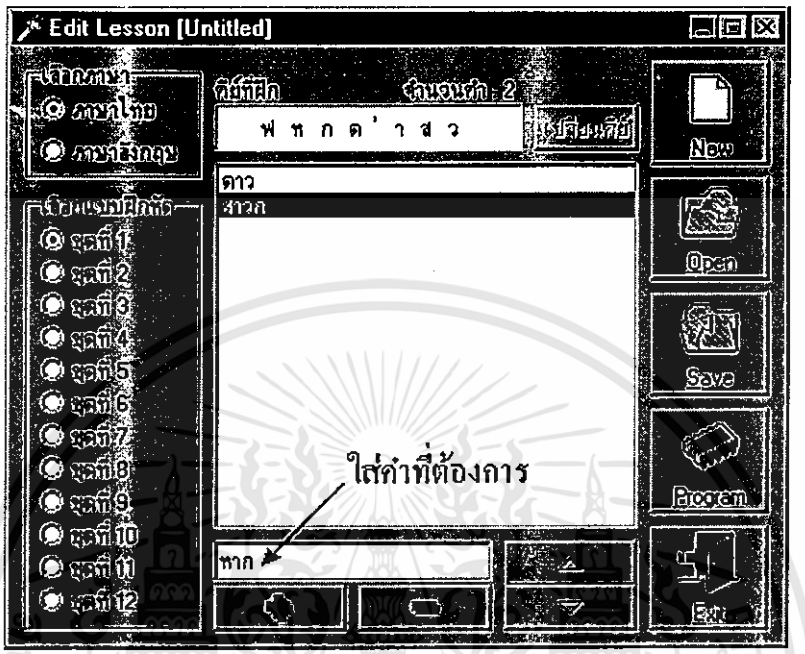
รูปที่ 3.37 หน้าต่างสำหรับใส่ชื่อไฟล์ของคำภาษาอังกฤษ

เมื่อใส่ชื่อไฟล์สำหรับคำภาษาไทยและภาษาอังกฤษเรียบร้อยแล้ว แบบฝึกหัดที่ได้จะเป็นแบบฝึกหัดที่ยังคงว่างเปล่าอยู่ ยกเว้นแบบฝึกหัดพิมพ์ดีย์ชุดที่ 1 ของทั้งภาษาไทยและภาษาอังกฤษ ซึ่งกำหนดไว้เป็นคีย์ **"A S D F J K L :"** สำหรับคีย์ภาษาอังกฤษและคีย์ **"ฟหกด่าสว"** สำหรับคีย์ภาษาอังกฤษ ซึ่งแบบฝึกหัดพิมพ์คีย์ชุดที่ 1 นี้ไม่สามารถทำการแก้ไขได้ส่วนแบบฝึกหัดพิมพ์คีย์ชุดอื่นๆสามารถทำการแก้ไขได้โดยการกดปุ่ม "เปลี่ยนคีย์" จะได้หน้าต่างดังรูปที่ 3.38

รูปที่ 3.38 หน้าต่างสำหรับแก้ไขแบบฝึกหัดพิมพ์คีย์

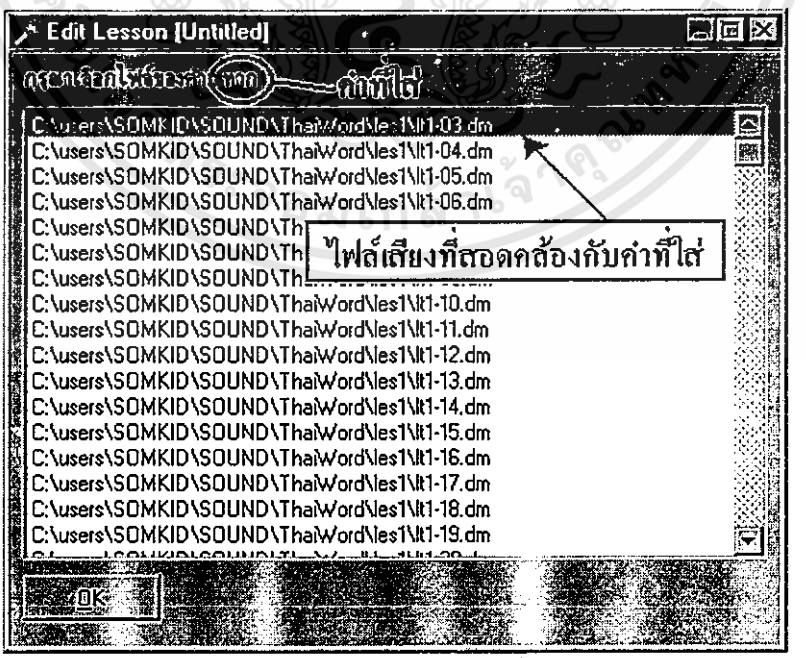
การเลือกคีย์ที่จะใช้ฝึกสามารถทำได้โดยการกดปุ่มคีย์นั้นๆบนหน้าจอได้ทันที ซึ่งสามารถเลือกคีย์ในแบบฝึกหัดพิมพ์คีย์แต่ละชุดได้ไม่เกินชุดละ 8 คีย์

ส่วนการแก้ไขแบบฝึกหัดพิมพ์คำทำได้โดยการใส่คำลงในช่องว่างและกดปุ่ม "+" ดังรูปที่ 3.39



รูปที่ 3.39 หน้าต่างสำหรับแก้ไขแบบฝึกหัดพิมพ์คำ

เมื่อใส่คำที่ต้องการแล้วโปรแกรมจะให้เลือกไฟล์ของเสียงที่สอดคล้องกับคำที่ใส่ลงไปข้างต้นดังรูปที่ 3.40



รูปที่ 3.40 หน้าต่างสำหรับเลือกไฟล์เสียงที่สอดคล้องกับคำที่ใส่

เมื่อกดปุ่ม OK ก็เป็นอันเสร็จขั้นตอนการเพิ่มคำในแบบฝึกหัดชุดนั้นๆ ส่วนการลบคำที่ไม่ต้องการก็เพียงแค่เลือกคำที่ไม่ต้องการแล้วกดปุ่ม "-" หรือสามารถทำการเลื่อนลำดับของคำที่ต้องการขึ้นหรือลงได้โดยการกดปุ่มขึ้นหรือลง ส่วนปุ่ม SAVE เอาไว้สำหรับบันทึกแบบฝึกหัดเก็บไว้เพื่อที่จะสามารถทำการแก้ไขได้ในภายหลัง

เมื่อทำการแก้ไขแบบฝึกหัดทั้งหมดเรียบร้อยแล้ว เราสามารถทำการโปรแกรมลงบนชุดฝึกพิมพ์ได้โดยการกดปุ่ม Program จะได้หน้าต่างดังรูปที่ 3.41 ซึ่งก่อนจะโปรแกรมเราจะต้องเลือกพอร์ตอนุกรมที่ชุดฝึกพิมพ์ต่ออยู่และเลือกส่วนที่เราต้องการที่จะโปรแกรมลงบนชุดฝึกพิมพ์ซึ่งสามารถเลือกโปรแกรมเฉพาะส่วนใดส่วนหนึ่งได้ คืออาจจะโปรแกรมเฉพาะแบบฝึกหัด หรือโปรแกรมเฉพาะเสียงของคำภาษาอังกฤษหรือเสียงของคำภาษาไทยได้ เมื่อตั้งค่าต่างๆเสร็จก็กดปุ่ม OK โปรแกรมก็จะทำการโปรแกรมส่วนที่เลือกไว้ลงบนชุดฝึกพิมพ์โดยที่จะมีแถบที่บอกเปอร์เซนต์ที่ได้โปรแกรมลงไปด้วย

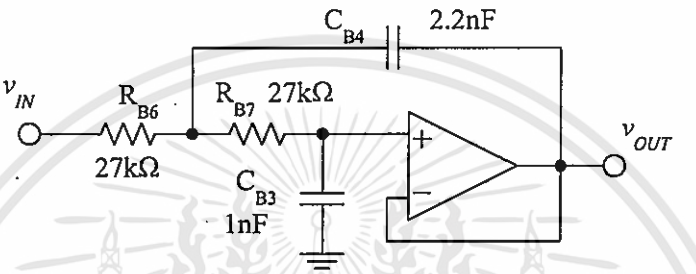
รูปที่ 3.41 หน้าต่างสำหรับโปรแกรมลงชุดฝึกพิมพ์

บทที่ 4

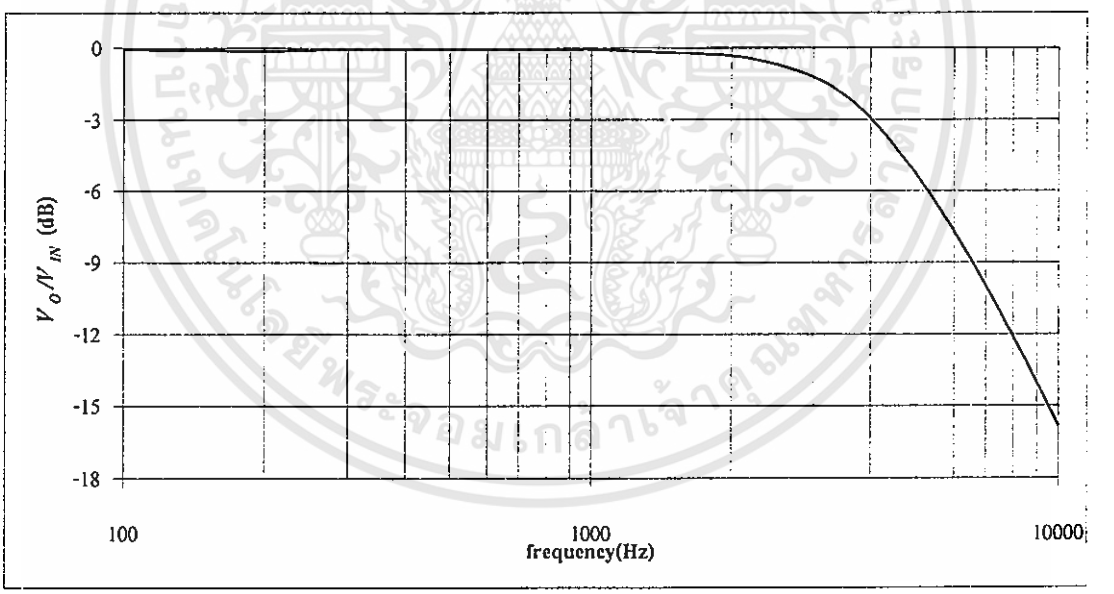
การทดลองและผลการทดลอง

4.1 การทดลองส่วน Delta Modulation และ Delta Demodulation

ทดสอบผลตอบสนองความถี่ของวงจรกรองทางด้านอินพุตของการ์ดบันทึกเสียง ซึ่งได้ออกแบบไว้เป็นวงจรกรองความถี่ต่ำผ่านอันดับสองแบบบัตเตอร์เวิร์ตดังรูปที่ 4.1 ซึ่งได้ผลตอบสนองความถี่ดังรูปที่ 4.2

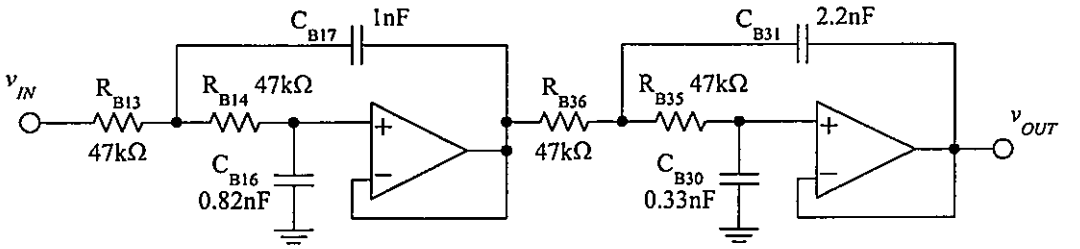


รูปที่ 4.1 วงจรกรองทางด้านอินพุตของการ์ดบันทึกเสียง

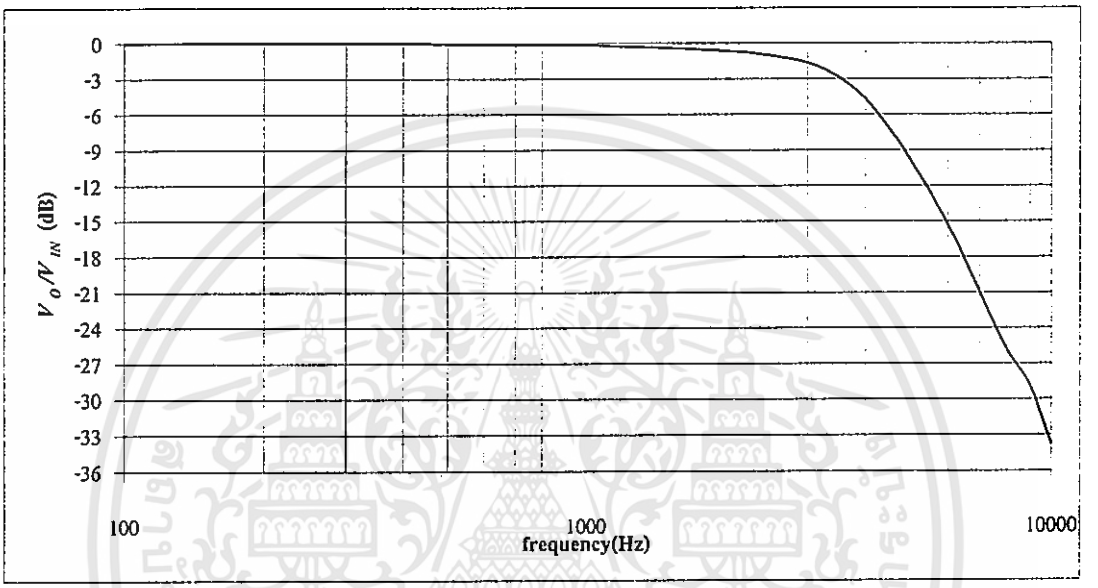


รูปที่ 4.2 ผลตอบสนองความถี่ของวงจรกรองในรูปที่ 4.1

วงจรกรองสัญญาณทางด้านเอาต์พุตของการ์ดบันทึกเสียงที่ใช้กรองสัญญาณที่ออกจากภาคดีมอดูเลชันได้ออกแบบเป็นวงจรกรองความถี่ต่ำผ่านอันดับสี่แบบบัตเตอร์เวิร์ตดังรูปที่ 4.3

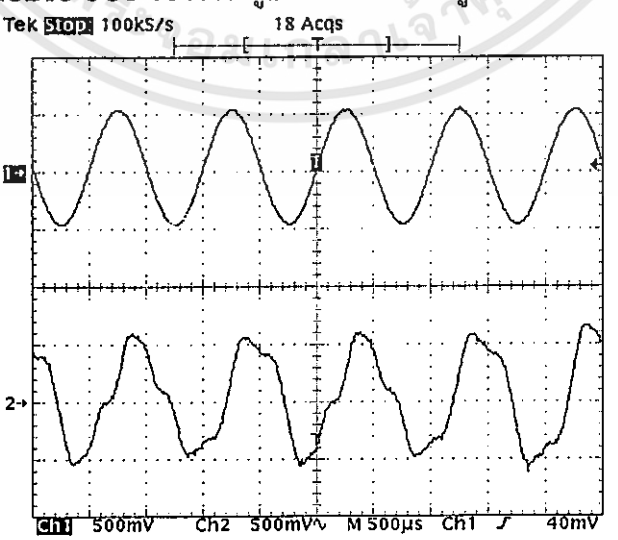


รูปที่ 4.3 วงจรกรองทางด้านเอาต์พุตของการ์ดบันทึกเสียง

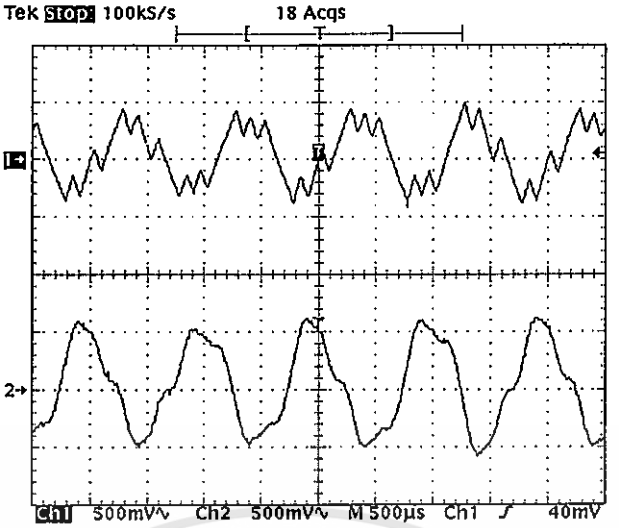


รูปที่ 4.4 ผลตอบสนองความถี่ของวงจรกรองในรูปที่ 4.3

ทำการทดสอบในส่วนของภาคมอดูเลชันและดีมอดูเลชันของการ์ดบันทึกเสียงโดยการป้อนสัญญาณซายน์ขนาด 2 Vpp ความถี่ 1 kHz ให้กับวงจรและวัดสัญญาณที่จุด AUDIO IN , ANALOG O/P และ AUDIO OUT ของวงจรรูปที่ 3.3 ได้ผลดังรูปที่ 4.5 และ 4.6



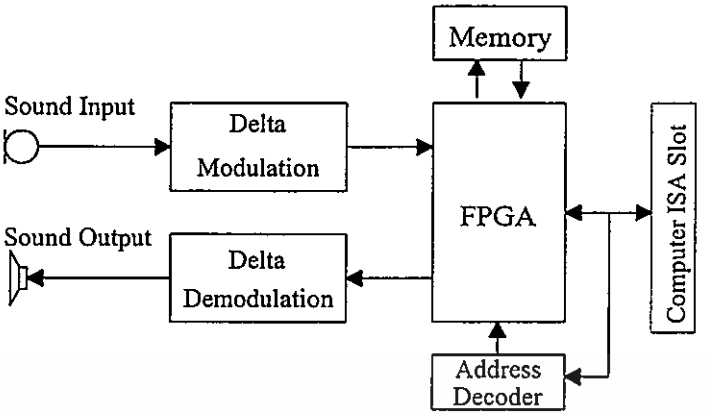
รูปที่ 4.5 สัญญาณ AUDIO IN (รูปบน) เทียบกับสัญญาณ AUDIO OUT (รูปล่าง)



รูปที่ 4.6 สัญญาณ ANALOG O/P (รูปบน) เทียบกับสัญญาณ AUDIO OUT (รูปล่าง)

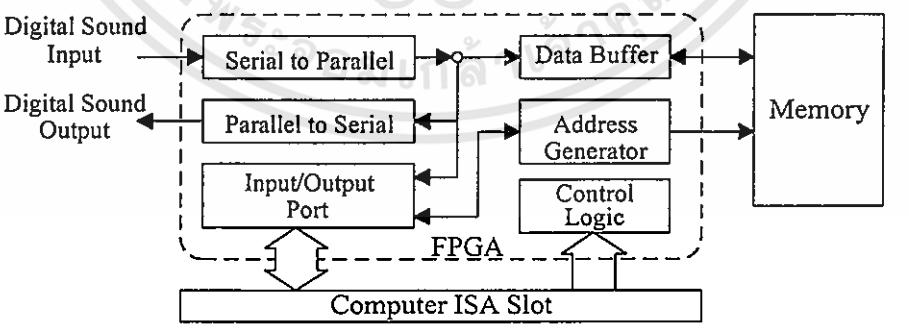
รูปที่ 4.5 และ 4.6 สัญญาณอินพุตของวงจร(AUDIO IN) ซึ่งมีขนาด 2 Vpp จะถูกนำไปแปลงเป็นสัญญาณดิจิตอลโดยวงจรมอดูเลชันและจะถูกแปลงกลับเป็นสัญญาณอนาลอก (ANALOG O/P) อีกครั้งหนึ่งโดยวงจรดีมอดูเลชัน ก่อนที่จะนำสัญญาณนี้ไปผ่านวงจรกรองและวงจรขยายเพื่อขับลำโพง (AUDIO OUT) ต่อไป ซึ่งจะเห็นได้ว่าสัญญาณ ANALOG O/P ที่ได้จากการดีมอดูเลชันนี้มีรูปร่างของสัญญาณที่ต่างไปจากสัญญาณอินพุตไปบ้าง เนื่องจากภาคเอาต์พุตของการดีมอดูเลชันจะเป็นอินทิเกรเตอร์ซึ่งทำหน้าที่อินทิเกรตสัญญาณอินพุตดิจิตอลให้กลับเป็นสัญญาณอนาลอก ซึ่งการอินทิเกรตสัญญาณดิจิตอลนี้เองที่ทำให้ลักษณะของสัญญาณเอาต์พุตมีลักษณะเป็นยอดแหลมรูปสามเหลี่ยม การที่จะทำให้สัญญาณกลับคืนมาเหมือนเดิมให้ได้นั้นควรจะต้องใช้วงจรกรองที่มีประสิทธิภาพสูงเพื่อที่จะกำจัดองค์ประกอบความถี่สูงของสัญญาณที่ได้ให้มีค่าลดลง เพื่อที่จะทำให้สัญญาณมีลักษณะที่ใกล้เคียงกับสัญญาณเดิมมากที่สุด ดังนั้นในวิทยานิพนธ์นี้จึงได้เลือกวงจรกรองทางด้านเอาต์พุตเป็นวงจรกรองความถี่ต่ำผ่านอันดับสี่แบบบัตเตอร์-เวิร์ต เนื่องจากให้ผลตอบสนองในแถบผ่านที่ค่อนข้างราบเรียบ และใช้อันดับของวงจรกรองถึงอันดับสี่เพื่อที่จะได้มีอัตราการลดทอนที่ความถี่สูงได้ดียิ่งขึ้นกว่าวงจรกรองที่มีอันดับต่ำกว่า

4.2 การประยุกต์ใช้ FPGA ในการออกแบบการ์ดบันทึกเสียง

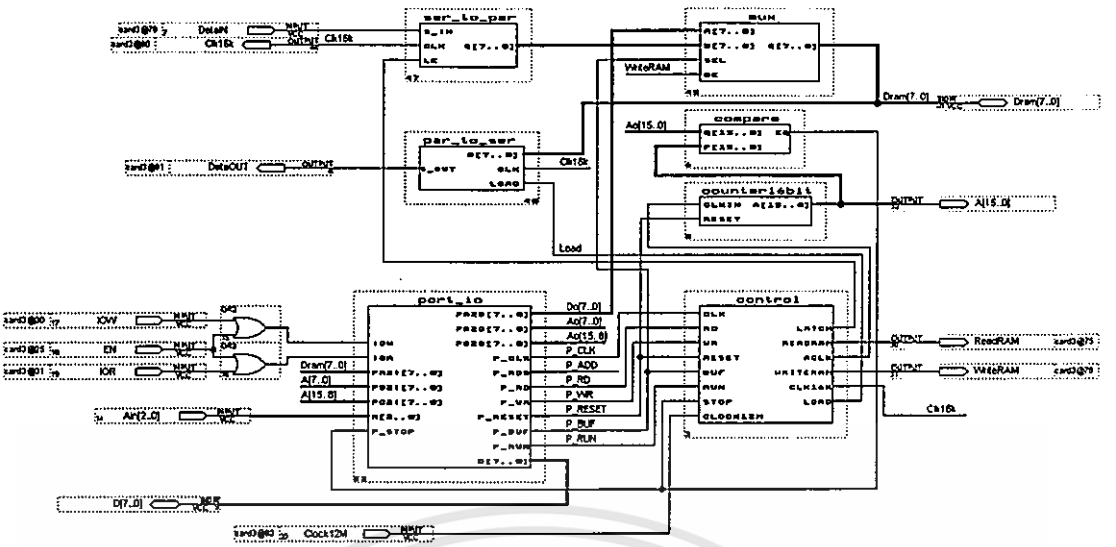


รูปที่ 4.7 โครงสร้างของการ์ดบันทึกเสียงที่สร้างขึ้นใหม่โดยใช้ FPGA

จากวงจรของการ์ดบันทึกเสียงที่ได้จะพบว่าวงจรมีขนาดที่ค่อนข้างใหญ่ใช้ไอซีดิจิตอลจำนวนมาก จึงได้คิดออกแบบวงจรในส่วนของวงจรควบคุมการทำงานของการ์ดบันทึกเสียงขึ้นใหม่โดยใช้ FPGA ซึ่งการออกแบบวงจรด้วยชิพ FPGA นี้จะอาศัยการออกแบบวงจรดิจิตอลโดยใช้ Schematic ผสมกับการเขียนวงจรด้วยภาษา AHDL โดยใช้โปรแกรม Max+Plus II ของบริษัท Altera ซึ่งจะมีความยืดหยุ่นในการออกแบบ สามารถออกแบบได้อย่างรวดเร็ว แก้ไขและตรวจสอบผลการทำงานของวงจรให้ถูกต้องได้ก่อนที่จะนำไปใช้งานจริงได้และสามารถเลือกเบอร์ชิพ FPGA ที่เหมาะสมก่อนที่จะนำวงจรที่ออกแบบไปโปรแกรมลงชิพและทดสอบการทำงานกับวงจรที่ต่อร่วมต่อไป ซึ่งโครงสร้างของการ์ดบันทึกเสียงที่สร้างขึ้นใหม่โดยใช้ FPGA แสดงดังรูปที่ 4.7 โดยจะเห็นว่าโครงสร้างแตกต่างกับการ์ดบันทึกเสียงเดิมเฉพาะในส่วนของวงจรควบคุม ซึ่งได้ใช้ชิพ FPGA มาแทนนั่นเอง โดยโครงสร้างภายในของชิพ FPGA ที่ออกแบบแสดงดังรูปที่ 4.8



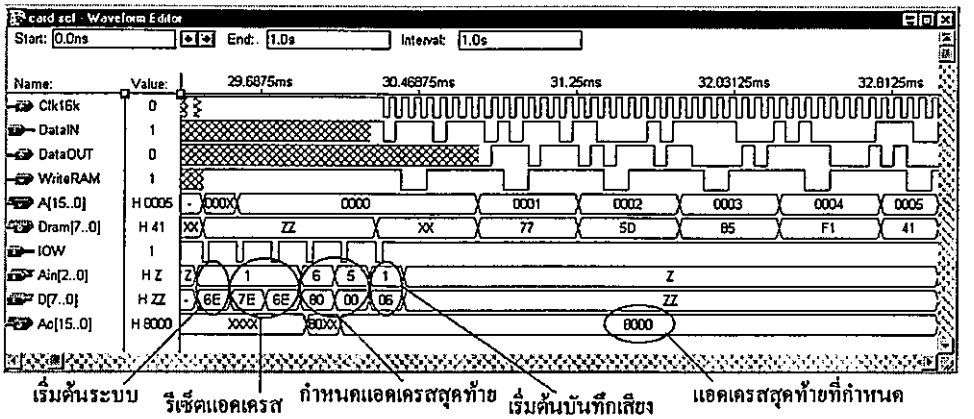
รูปที่ 4.8 โครงสร้างภายใน FPGA ที่ออกแบบ



รูปที่ 4.9 วงจรที่ใช้จำลองการทำงานด้วย Max+Plus II

การทำงานของวงจรภายในชิพ FPGA ที่ออกแบบจะเริ่มโดยรับสัญญาณเสียงที่เป็นดิจิตอลขนาด 1 บิตที่ได้จากส่วนมอดูเลชันมาเลื่อนข้อมูลเป็น 8 บิต โดยวงจรแปลงสัญญาณอนุกรมเป็นขนาน (Serial to Parallel) ส่วนวงจรแปลงสัญญาณขนานเป็นอนุกรม (Parallel to Serial) จะทำหน้าที่เลื่อนสัญญาณขนาด 8 บิต เป็นสัญญาณ 1 บิตอีกครั้งเพื่อส่งสัญญาณไปยังส่วนดีมอดูเลชันต่อไป โดยมีวงจรควบคุมทำหน้าที่คอยควบคุมจังหวะการทำงานของวงจรทั้งหมด ซึ่งจะเป็นวงจรที่คอยรับคำสั่งต่างๆเช่น คำสั่งในการเริ่มบันทึกเสียง หรือหยุดบันทึกเสียงเป็นต้นจากซอฟต์แวร์บนคอมพิวเตอร์ แล้วสร้างสัญญาณควบคุมต่างๆออกมาเช่นสัญญาณนาฬิกา สัญญาณที่ใช้เลื่อนข้อมูล สัญญาณในการอ่าน/เขียนหน่วยความจำ สัญญาณในการนับแอดเดรสขึ้นเป็นต้น โดยรายละเอียดของโครงสร้างในส่วนต่างๆในรูป 4.9 แสดงรายละเอียดไว้ในภาคผนวก จ.

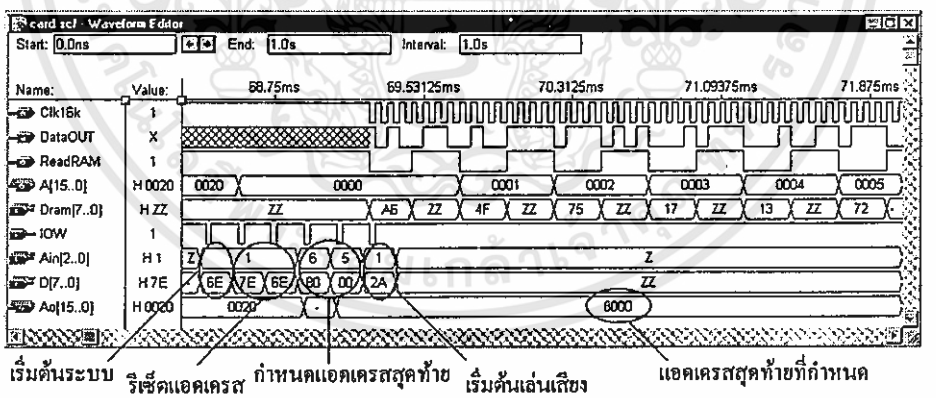
ผลการจำลองการทำงานของวงจรควบคุมที่สร้างขึ้นบน FPGA ในรูปที่ 4.9 โดยโปรแกรม Max+Plus II ได้ผลดังรูป 4.10 ถึงรูปที่ 4.14 โดยในรูปที่ 4.10 เป็นการจำลองการทำงานขณะสั่งเริ่มบันทึกเสียงโดยเริ่มจากการส่งคำสั่ง 6EH ออกมาก่อน ซึ่งเป็นคำสั่งที่ใช้ในการเริ่มต้นระบบหรือก็คือคำสั่งที่สั่งให้การ์ดอยู่ในสภาวะเตรียมพร้อมใช้งาน หลังจากนั้นก็ทำการรีเซ็ตแอดเดรสของหน่วยความจำ และกำหนดแอดเดรสสุดท้ายที่จะทำการบันทึกซึ่งในรูปก็คือแอดเดรส 8000H (หน่วยความจำที่ใช้มีขนาด 16 กิโลไบต์ คือมีแอดเดรสตั้งแต่ 0000H – 7FFFH) และสั่งเริ่มต้นการบันทึกเสียงโดยส่งคำสั่ง 06H ออกไป การ์ดก็จะเริ่มทำการบันทึกเสียงซึ่งจะเห็นจากในรูปว่ามีการส่งสัญญาณนาฬิกาเข้าสู่ระบบ ข้อมูลเสียงขนาด 1 บิต(DataIN) ก็จะถูกเลื่อนไปเป็นข้อมูลขนาด 8 บิต(Dram[7..0]) และเขียนลงสู่หน่วยความจำ และแอดเดรสของหน่วยความจำ(A[15..0]) ก็จะถูกนับขึ้นเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงแอดเดรสสุดท้ายที่ได้กำหนดไว้



รูปที่ 4.10 การจำลองการทำงานขณะสั่งบันทึกเสียง

อนึ่งจากรูปที่ 4.10 จะสังเกตได้อย่างหนึ่งว่าข้อมูลเสียงทางอินพุต(DataIN) และข้อมูลเสียงเอาต์พุต(DataOUT) จะเหลื่อมกันไปอยู่เท่ากับ 8 สัญญาณนาฬิกา

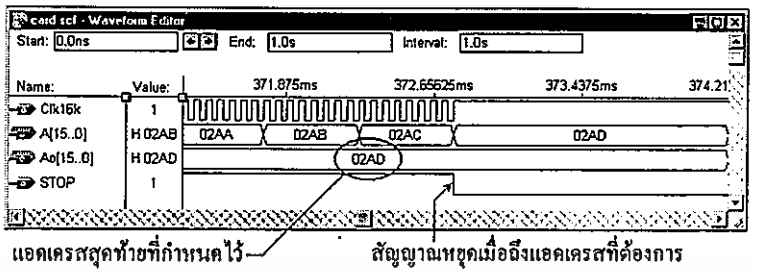
รูปที่ 4.11 เป็นการจำลองการทำงานเมื่อสั่งเล่นข้อมูลเสียงที่เก็บอยู่ในหน่วยความจำออกทางลำโพง ซึ่งการทำงานก็คือสั่งเริ่มต้นระบบและรีเซ็ตแอดเรสให้เป็น 0000H หลังจากนั้นก็ทำการกำหนดแอดเดรสสุดท้ายที่จะเล่น และส่งคำสั่ง 2AH ออกไปโปรแกรมก็จะทำการอ่านข้อมูลเสียงที่อยู่ในหน่วยความจำขนาด 8 บิตเลื่อนข้อมูลเป็นขนาด 1 บิตเพื่อส่งออกไปยังภาคดีมอดูเลชันต่อไปและแอดเดรสของหน่วยความจำก็จะถูกนับขึ้นเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงแอดเดรสสุดท้ายที่ได้กำหนดไว้



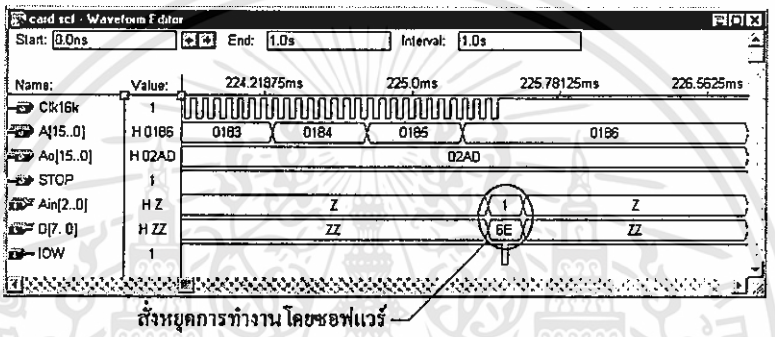
รูปที่ 4.11 การจำลองการทำงานขณะสั่งเล่นข้อมูลเสียงที่อยู่ในหน่วยความจำออกทางลำโพง

การที่การ์ดจะหยุดการทำงานคือหยุดบันทึกเสียงหรือหยุดเล่นข้อมูลเสียงที่อยู่ในหน่วยความจำเป็นไปได้ 2 กรณีคือ กรณีแรกกรณีที่มีการอ่านหรือเขียนข้อมูลหน่วยความจำถึงแอดเดรสสุดท้ายที่ได้กำหนดไว้วงจรควบคุมก็จะสร้างสัญญาณ STOP เพื่อหยุดการทำงานเองโดยอัตโนมัติและสัญญาณเดียวกันนี้จะจะถูกส่งขึ้นไปยังซอฟต์แวร์บนคอมพิวเตอร์เพื่อบอกคอมพิวเตอร์ด้วยว่าการ์ดได้หยุดการทำงานแล้ว ดังแสดงในรูปที่ 4.12(ก) ส่วนการหยุดการทำงานในกรณีที่สองคือ

การสั่งหยุดการทำงานด้วยการสั่งโดยซอฟต์แวร์ในกรณีที่แอดเดรสยังไม่ถึงแอดเดรสสุดท้ายที่กำหนดไว้ โดยการส่งคำสั่ง 6EH ออกมาการ์ดก็จะหยุดทำงานทันทีดังรูปที่ 4.12(ข)



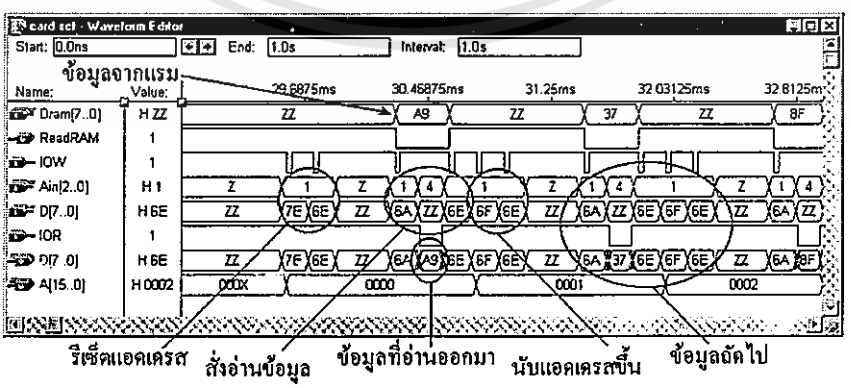
(ก) หยุดการทำงานแบบอัตโนมัติ



(ข)หยุดการทำงานโดยสั่งจากซอฟต์แวร์

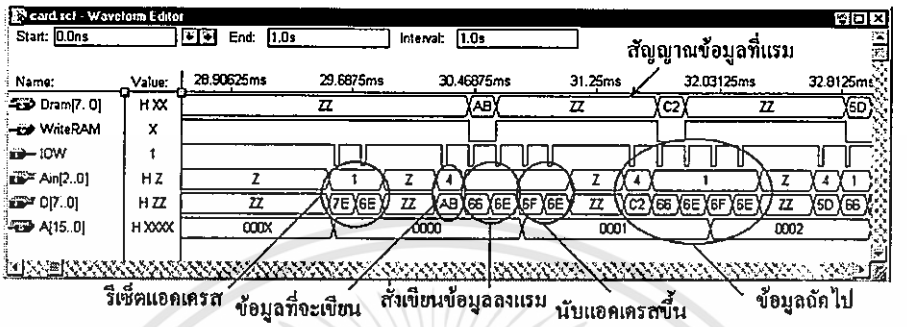
รูปที่ 4.12 การจำลองการทำงานขณะหยุดการทำงาน

เมื่อบันทึกข้อมูลเสียงเสร็จแล้วข้อมูลเสียงที่ได้จะอยู่ในหน่วยความจำซึ่งซอฟต์แวร์จะทำการอ่านข้อมูลเสียงนี้ขึ้นมาเพื่อทำการประมวลผล โดยสามารถทำได้ดังรูปที่ 4.13 ก็คือเริ่มต้นจากการสั่งรีเซ็ตแอดเดรส สั่งอ่านข้อมูลจากหน่วยความจำและสั่งนับแอดเดรสขึ้นสลับกันไปเรื่อยๆ จนกระทั่งถึงแอดเดรสสุดท้ายที่ต้องการอ่าน



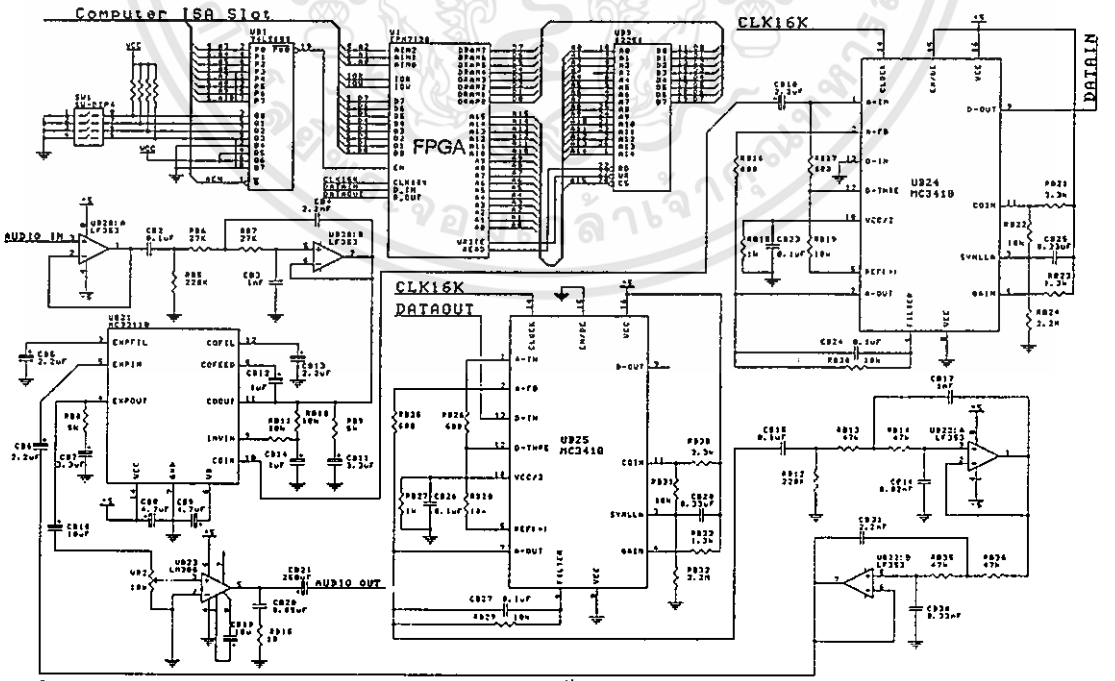
รูปที่ 4.13 การจำลองการทำงานขณะสั่งอ่านข้อมูลจากหน่วยความจำ

ส่วนรูปที่ 4.14 เป็นการจำลองการทำงานขณะที่สั่งเขียนข้อมูลลงหน่วยความจำบนการ์ดใช้เมื่อต้องการเอาข้อมูลเสียงที่เก็บไว้บนคอมพิวเตอร์เขียนลงสู่หน่วยความจำบนการ์ดเพื่อทำการเล่นเสียงที่ได้บันทึกไว้นั้น ซึ่งขั้นตอนการทำงานก็คือสั่งรีเซ็ตแอดเดรส สั่งเขียนข้อมูลลงหน่วยความจำและสั่งนับแอดเดรสขึ้นสลับกันไปเรื่อยๆจนกระทั่งหมดข้อมูลที่จะเขียน



รูปที่ 4.14 การจำลองการทำงานขณะสั่งเขียนข้อมูลลงหน่วยความจำ

เมื่อนำวงจรที่ได้ออกแบบไว้นี้ไปโปรแกรมลงบนชิพ FPGA โดยชิพ FPGA ที่ใช้เป็นเบอร์ EPM7128SLC-10 ซึ่งใช้จำนวนลอจิกเซลล์ไปจำนวน 122 ลอจิกเซลล์ โดยจะมีจำนวนลอจิกเซลล์ว่างอยู่อีกเล็กน้อย ซึ่งจะทำให้สามารถพัฒนาวงจรส่วนควบคุมนี้ต่อไปได้อีกในอนาคตโดยไม่ต้องเปลี่ยนแปลงวงจรแต่อย่างใดเพียงแค่โปรแกรมลงบนชิพ FPGA ใหม่เท่านั้นเอง เมื่อนำชิพ FPGA ที่ได้โปรแกรมแล้วนี้ไปทดลองกับวงจรส่วนอื่นๆผลปรากฏว่าได้ผลเหมือนกับวงจรดั้งเดิมทุกประการ โดยวงจรรวมของการ์ดบันทึกเสียงที่สร้างขึ้นใหม่เป็นดังรูปที่ 4.15



รูปที่ 4.15 วงจรรวมของการ์ดบันทึกเสียงที่พัฒนาขึ้นบน FPGA

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ก่อนอื่นจะกล่าวสรุปถึงคุณสมบัติและขอบเขตการใช้งานชุดฝึกพิมพ์และการ์ดบันทึกเสียงที่ได้ออกแบบไว้ดังนี้

- ชุดฝึกพิมพ์
    - ใช้เชื่อมต่อกับคีย์บอร์ดของคอมพิวเตอร์ทั่วๆไปได้ทางขั้วต่อ DIN
    - มีภาษาที่ใช้งาน 2 ภาษาคือภาษาไทยและภาษาอังกฤษสามารถเลือกได้โดยใช้สวิตช์บนชุดฝึกพิมพ์
    - มีแบบฝึกหัดทั้งภาษาไทยและภาษาอังกฤษอย่างละ 24 ชุดแบบฝึกหัด คือ 12 ชุดแบบฝึกหัดพิมพ์คีย์ และ 12 ชุดแบบฝึกหัดพิมพ์คำ สำหรับการฝึกฝนทักษะในการพิมพ์และทดสอบความสามารถในการพิมพ์
    - สามารถใช้งานได้โดยที่ไม่ต้องเชื่อมต่อกับคอมพิวเตอร์
    - มีซอฟต์แวร์บนไมโครซอฟต์วินโดว์เพื่อใช้ในการแก้ไขชุดแบบฝึกหัด และสามารถเชื่อมต่อกับชุดฝึกพิมพ์โดยผ่านทางพอร์ตอนุกรมเพื่อดาวน์โหลดข้อมูลของแบบฝึกหัดและเสียงที่ใช้ในแบบฝึกหัดไปยังชุดฝึกพิมพ์
- การ์ดบันทึกเสียง
    - บันทึกเสียงโดยใช้ไอซีเบอร์ MC3418 ที่ใช้หลักการ CVSD
    - สามารถบันทึกเสียงได้นานครั้งละ 16 วินาที โดยที่ใช้อัตราในการสุ่ม 16 kHz
    - มีซอฟต์แวร์ที่ใช้ตัด-ต่อ และปรับปรุงคุณภาพของเสียงบนไมโครซอฟต์วินโดว์
    - สามารถนำไฟล์เสียงที่ได้บันทึกไว้ไปใช้แก้ไขแบบฝึกหัดบนชุดฝึกพิมพ์ได้

อย่างไรก็ตามจะเห็นว่าวิทยานิพนธ์นี้ยังมีข้อจำกัดอยู่บ้างเช่นความถี่ที่ใช้ในการสุ่มของการ์ดบันทึกเสียงซึ่งกำหนดไว้คงที่ที่ 16 kHz และรูปแบบวงจรมอดูเลชันและดีมอดูเลชันที่กำหนดไว้ตายตัวไม่สามารถปรับเปลี่ยนได้ ซึ่งการออกแบบไว้เช่นนี้ก็เพราะพยายามที่จะออกแบบวงจรไว้เพื่อการนำไปใช้งานจริงได้อย่างสะดวกและประหยัด กล่าวคือใช้อุปกรณ์ที่ค่อนข้างน้อยไม่ยุ่งยาก ใช้ความถี่ที่ปานกลางคือไม่สูงไปหรือต่ำไปนัก วงจรโดยรวมของการ์ดบันทึกเสียงมีขนาดที่ค่อนข้างใหญ่ซึ่งถ้าสร้างวงจรทั้งหมดเป็นการ์ดจริงๆจะได้การ์ดที่ค่อนข้างใหญ่ไม่สะดวกในการเสียบลงบนสล็อตภายในเครื่องคอมพิวเตอร์จึงได้มีการแยกส่วนการ์ดบันทึกเสียงออกเป็น 2 ส่วน ส่วนหนึ่งอยู่ภายในเครื่องคอมพิวเตอร์และอีกส่วนหนึ่งอยู่ภายนอก หรืออาจแก้ไขได้อีกอย่างหนึ่งคือการพัฒนาวงจรควบคุมการทำงานบน FPGA ซึ่งแสดงไว้ในบทที่ 4 โดยจะเห็นว่าวงจรมีขนาด

เล็กลงค่อนข้างมาก แต่ว่าชิพ FPGA เองยังมีราคาที่ค่อนข้างแพงซึ่งคงไม่เหมาะถ้าจะนำไปผลิตในเชิงพาณิชย์ ส่วนซอฟต์แวร์ที่ใช้กับการ์ดบันทึกเสียงอาจจะยังมีฟังก์ชันในการใช้งานที่ยังไม่มากนักซึ่งอาจจะต้องมีการพัฒนาซอฟต์แวร์ต่อไปอีกในอนาคต

ส่วนในส่วนของชุดฝึกพิมพ์นั้นสิ่งที่เป็นข้อจำกัดก็คือขนาดของหน่วยความจำที่ใช้เก็บแบบฝึกหัดและเก็บสัญญาณเสียงโดยหน่วยความจำที่ใช้ในการเก็บสัญญาณเสียงในแต่ละภาษามีขนาดเท่ากับ 512 กิโลไบต์และหน่วยความจำที่จองไว้สำหรับแบบฝึกหัดแต่ละชุดมีขนาดเท่ากับ 255 ไบต์ ซึ่งถ้าต้องการขยายหน่วยความความจำในส่วนนี้ก็ต้องมีการแก้ไขซอฟต์แวร์ของชุดฝึกพิมพ์อีกเล็กน้อย ส่วนในส่วนของแบบฝึกหัดของชุดฝึกพิมพ์ก็ได้มีซอฟต์แวร์ที่ใช้ในการแก้ไขแบบฝึกหัดอยู่แล้วดังนั้นผู้ใช้ก็สามารถทำการแก้ไขแบบฝึกหัดให้มีความเหมาะสมสำหรับแต่ละบุคคลได้เพียงแต่ว่าถ้าต้องการแก้ไขแบบฝึกหัดพิมพ์คำก็ต้องมีการ์ดที่ใช้บันทึกเสียงด้วย

ปัญหาที่พบในวิทยานิพนธ์นี้ก็มีอยู่บ้างเช่นหน่วยความจำ EEPROM ขนาด 4 เมกกะบิตที่ใช้ในชุดฝึกพิมพ์เบอร์ 29C040 ซึ่งหาอุปกรณ์ได้ค่อนข้างยากต้องสั่งนำเข้ามาจากต่างประเทศและเรื่องคุณภาพของเสียงที่ยังไม่ค่อยดีเท่าที่ควรทั้งนี้อาจจะสรุปสาเหตุได้จากหลายกรณีเช่นในส่วนของวงจรกรองซึ่งอาจจะกรองสัญญาณรบกวนที่เกิดจากการแปลงสัญญาณดิจิตอลมาเป็นอนาลอกได้ไม่ดีเท่าที่ควร ซึ่งก็อาจต้องใช้วงจรกรองที่มีประสิทธิภาพดีขึ้น หรือการออกแบบระบบกราวน์ที่อาจจะยังไม่ดีนัก คือการแยกกราวน์กันระหว่างกราวน์อนาลอก กราวน์ดิจิตอล และกราวน์ของแหล่งจ่ายไฟยังไม่ดีพอ ทำให้อาจมีสัญญาณรบกวนจากส่วนดิจิตอลเข้ามารบกวนส่วนอนาลอกได้ ซึ่งการออกแบบควรจะแยกกราวน์ อนาลอกและดิจิตอลให้ชัดเจนและนำมาเชื่อมต่อกันใต้ตัวไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอล แต่อย่างไรก็ตามในการออกแบบจริงการจัดวางอุปกรณ์ต่างๆเพื่อให้การเดินลายทองแดงเป็นไปได้อย่างง่ายและไม่เดินลายทองแดงยาวไปนักทำให้การเดินระบบกราวน์ยังทำได้ไม่ดีเท่าที่ควร

โดยรวมแล้ววิทยานิพนธ์นี้ได้ออกแบบและสร้างชุดฝึกพิมพ์สำหรับคนตาบอดและการ์ดบันทึกเสียงขึ้นมาใช้งานเองได้ในประเทศแม้ว่าอาจจะยังมีข้อบกพร่องอยู่บ้างแต่ก็สามารถใช้เป็นแนวทางที่จะพัฒนาต่อไปได้ทั้งในส่วนของซอฟต์แวร์และฮาร์ดแวร์ และคงจะเป็นประโยชน์สำหรับคนตาบอดและคนปกติที่จะใช้ฝึกฝนทักษะการพิมพ์ให้ดียิ่งขึ้น และสำหรับการ์ดบันทึกเสียงก็คงจะเป็นทางเลือกหนึ่งในการเลือกใช้ไอซีบันทึกเสียงเบอร์ MC3418 แทนการใช้งานไอซีบันทึกเสียงเบอร์อื่นๆที่มีในท้องตลาด

เอกสารอ้างอิง

[1] วิศิษฐ์ สุขจิตร และคณะ. "ชุดฝึกพิมพ์สำหรับคนตาบอด". การประชุมวิชาการทางไฟฟ้าครั้งที่ 19, พฤศจิกายน 2539. หน้า 163-267.

[2] ดุลยา งามฤทธิ์. **พิมพ์ดีดไทย(ฉบับครบถ้วนสมบูรณ์).** กรุงเทพมหานคร : บริษัทอมรินทร์พริ้นติ้ง แอนด์ พับลิชชิ่ง จำกัด(มหาชน).

[3] ดุลยา งามฤทธิ์. **พิมพ์ดีดอังกฤษ(ฉบับครบถ้วนสมบูรณ์).** กรุงเทพมหานคร : บริษัทอมรินทร์พริ้นติ้ง แอนด์ พับลิชชิ่ง จำกัด(มหาชน).

[4] Motorola Inc. **MC3418 Motorola Communication data book.** 1995

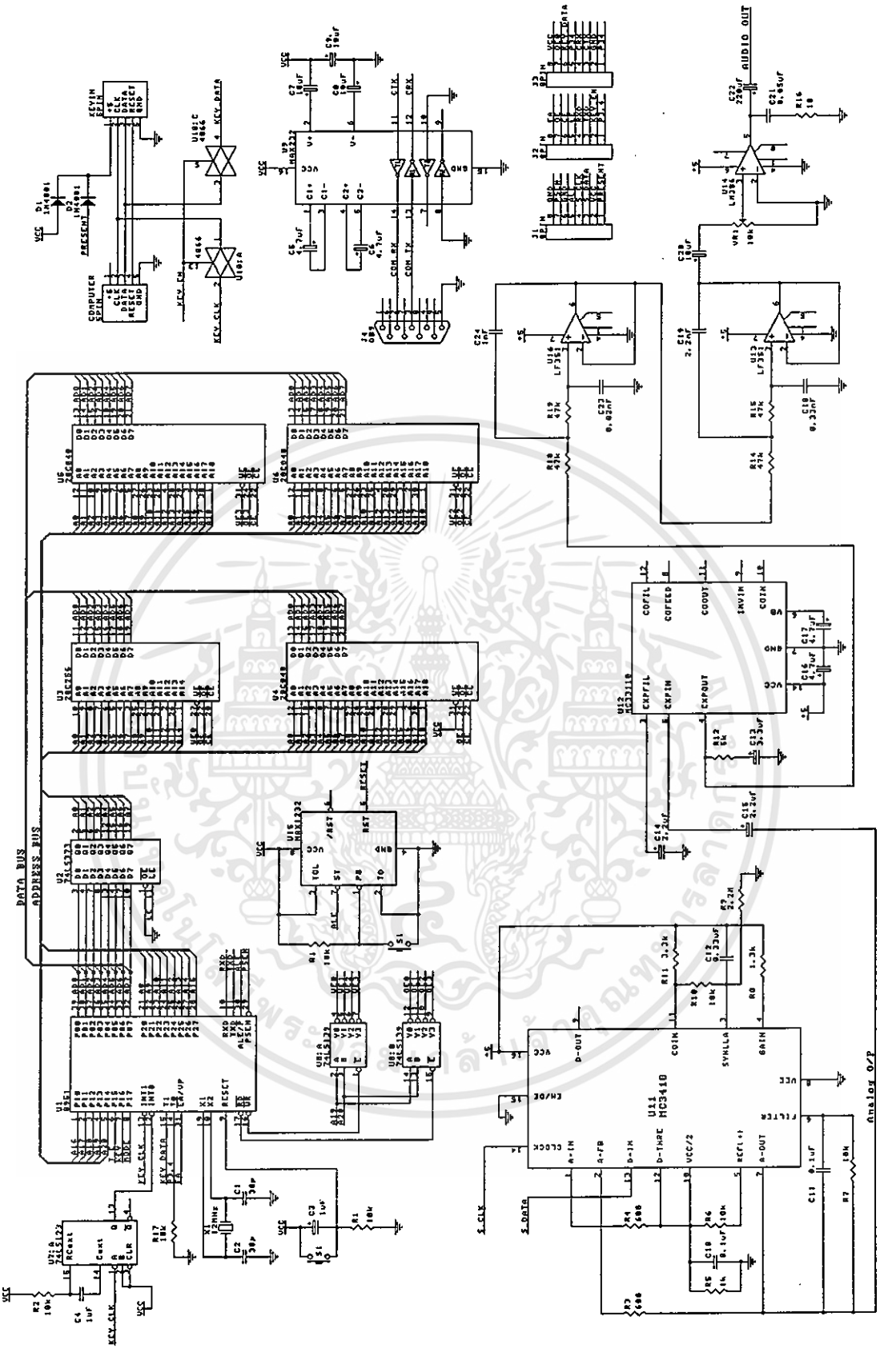
[5] Willis J. Tompkins, John G. Webster, Editors. **Interfacing Sensors to the IBM PC.** New Jersey : Prentice-Hall Inc. 1988.

[6] Altera Corporation. **MAX7000 . Altera Data Book.** 1998.

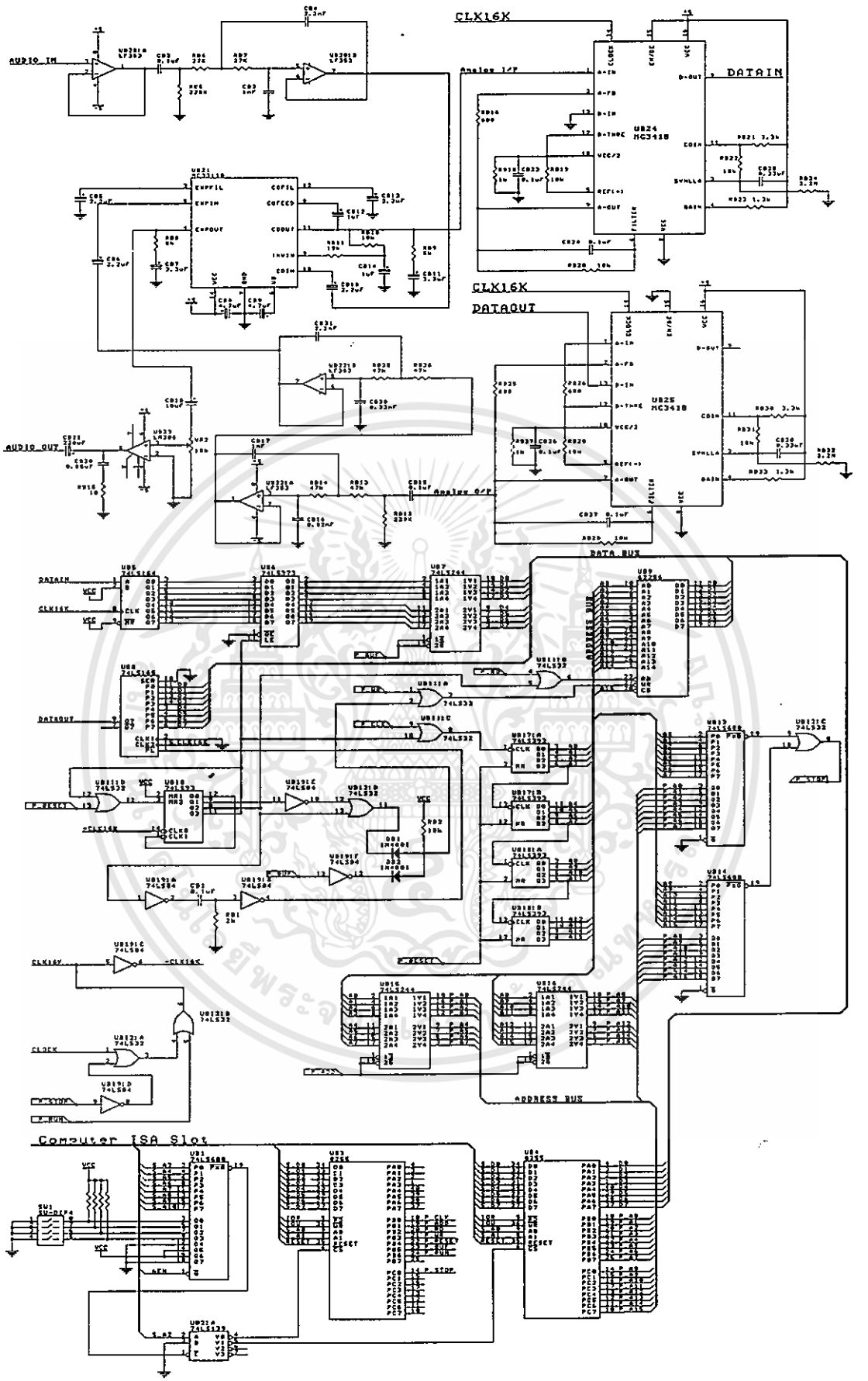
[7] Donald A. Neaman. **Electronic Circuit Analysis and Design.** McGraw-Hill Higher Education. 2000.

[8] พรชัย ภววงษ์ศักดิ์. **การประมวลผลสัญญาณดิจิตอลเบื้องต้น.** กรุงเทพมหานคร : โครงการตำราวิชาการมหาวิทยาลัยเทคโนโลยีมหานคร. 1999.

ภาคผนวก ก.
วงจรสมบูรณ์ของส่วนต่างๆ



รูปที่ ก.1 วงจรรวมของชุดฝึกพิมพ์



รูปที่ ก.2 วงจรรวมของการ์ดบันทึกเสียง

ภาคผนวก ข.

โปรแกรมของส่วนบันทึกเสียง

```
program DREC;
uses
  Forms,
  MainDREC in 'MainDREC.pas' {Form1},
  CnvFile in 'CnvFile.pas' {Form2};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.Run;
end.

//=============================================
unit MainDREC;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,  ExtCtrls, StdCtrls,
  ComCtrls, ToolWin, Menus, Buttons ;
type
  TForm1 = class(TForm)
    OpenDialog1 : TOpenDialog;
    SaveDialog1 : TSaveDialog;
    Edit1    : TEdit;
    Image1   : TImage;
    Image2   : TImage;
    Image3   : TImage;
    Cursor1  : TImage;
    Cursor2  : TImage;
    Cstart   : TShape;
    Cstop    : TShape;
    Timer1   : TTimer;
    BtnPlay  : TBitBtn;
    BtnStop  : TBitBtn;
    BtnRec   : TBitBtn;
    StartTimeText : TLabel;
    StopTimeText  : TLabel;
    DeltaTText    : TLabel;
    ScrollBar1    : TScrollBar;
    MainMenu1       : TMainMenu;
    MnuFile         : TMenuItem;
    MnuFileOpen     : TMenuItem;
    MnuFileSave     : TMenuItem;
    MnuFileSaveAll  : TMenuItem;
    MnuFileSaveSelect: TMenuItem;
    MnuFileConvert  : TMenuItem;
    MnuFileExit     : TMenuItem;
    MnuEdit         : TMenuItem;
    MnuEditCut      : TMenuItem;
    MnuEditPaste    : TMenuItem;
    MnuEditInsert   : TMenuItem;
    MnuEditSelectAll : TMenuItem;
    MnuPlay         : TMenuItem;
    MnuPlayRecord   : TMenuItem;
    MnuPlayPlay     : TMenuItem;
    MnuPlayPlayAll  : TMenuItem;
    MnuPlayStop     : TMenuItem;
    MnuEffect       : TMenuItem;
    MnuEffectMute   : TMenuItem;
    MnuEffectFilter : TMenuItem;
    MnuEffectFilter1 : TMenuItem;
    MnuEffectFilter2 : TMenuItem;
    MnuEffectFilter3 : TMenuItem;
    MnuEffectFilter4 : TMenuItem;
    MenuLine        : TBevel;
    StatusBar1      : TStatusBar;
    Bevel1 : TBevel;
    Bevel2 : TBevel;
    Bevel3 : TBevel;

    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject;
        var Action: TCloseAction);
    procedure ScrollBar1Change(Sender: TObject);
    procedure Image1MouseDown(Sender: TObject;
        Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Image2MouseDown(Sender: TObject;
        Button: TMouseButton; Shift: TShiftState;
        X, Y: Integer);
    procedure Image2MouseUp(Sender: TObject;
        Button: TMouseButton; Shift: TShiftState;
        X, Y: Integer);
    procedure Initial;
    procedure ResetCounter;
    procedure Starting;
    procedure SetEndPoint(EndAddress:integer);
    procedure PlayBack;
    procedure StopRunning;
    procedure ReadData;
    procedure WriteData(Start,Stop : word);
    procedure PlotAll;
    procedure Plot(Time : Integer);
    procedure InitGraph;
    procedure ConvertDMODToPCM;
    procedure ConvertPCMToDMOD;
    procedure Filter(Freq : Integer);
    procedure FindIOAddress;
    procedure MnuFileSaveAllClick(Sender: TObject);
    procedure MnuFileOpenClick(Sender: TObject);
    procedure MnuFileSaveSelectClick(Sender:
        TObject);
    procedure MnuFileConvertClick(Sender: TObject);
    procedure MnuFileExitClick(Sender: TObject);
    procedure MnuEditCutClick(Sender: TObject);
    procedure MnuEditPasteClick(Sender: TObject);
    procedure MnuEditInsertClick(Sender: TObject);
    procedure MnuEditSelectAllClick(Sender: TObject);
    procedure MnuPlayRecordClick(Sender: TObject);
    procedure MnuPlayPlayClick(Sender: TObject);
    procedure MnuPlayPlayAllClick(Sender: TObject);
    procedure MnuPlayStopClick(Sender: TObject);
    procedure MnuEffectMuteClick(Sender: TObject);
    procedure MnuEffectFilter1Click(Sender: TObject);
    procedure MnuEffectFilter2Click(Sender: TObject);
    procedure MnuEffectFilter3Click(Sender: TObject);
    procedure MnuEffectFilter4Click(Sender: TObject);
    procedure MnuFileSaveClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
    Form1: TForm1;
    Data : array[0..32767] of Byte;
    DataPCM : array[0..262144] of Word;
    MaxData,DataLength : Word;
    RightClick,LeftClick : Boolean;
    StartPoint,StopPoint : Word;
    StartTime,StopTime : Real;
```

```
    DataPortA1 ,DataPortB1 ,DataPortC1,
    DataPortA2 ,DataPortB2 ,DataPortC2 : Byte;
    Recording,Playing : Boolean;
    CurrentFile : string;
    Modify : Boolean;
    CardFound : Boolean;
    PortA1      : word;
    PortB1      : word;
    PortC1      : word;
    PortControl1 : word;
    PortA2      : word;
    PortB2      : word;
    PortC2      : word;
    PortControl2 : word;
implementation
uses CnvFile;
{$R *.DFM}

//---------------------------------------------------------------
Procedure TForm1.FormCreate(Sender: TObject);
begin
    Form1.Left := 1;
    Form1.Top := 1;
    Form1.Width := Screen.Width*11 div 12;
    Form1.Height := Screen.Height*11 div 12;
    MenuLine.Top := 0;
    MenuLine.Left := 0;
    MenuLine.Width := Form1.ClientWidth-5;
    Image1.Top := 15;
    Image1.Left := 15;
    Image1.Height := round((Form1.ClientHeight-75)*
                           0.15);
    Image1.Width := Form1.ClientWidth-30;
    Image1.Canvas.Create;
    Bevel1.Top := Image1.Top-2;
    Bevel1.Left := Image1.Left-2;
    Bevel1.Height := Image1.Height+4;
    Bevel1.Width := Image1.Width+4;
    Image2.Top := Image1.Top + Image1.Height + 15;
    Image2.Left := Image1.Left;
    Image2.Height := round((Form1.ClientHeight-75)*
                           0.85);
    Image2.Width := Form1.ClientWidth - 155;
    Image2.Canvas.Create;
    Bevel2.Top := Image2.Top-2;
    Bevel2.Left := Image2.Left-2;
    Bevel2.Height := Image2.Height + 6 +
                     ScrollBar1.Height;
    Bevel2.Width := Image2.Width+4;
    Image3.Top := Image1.Top-1;
    Image3.Left := Image1.Left;
    Image3.Height := Image1.Height+2;
    Image3.Width := Image1.Width;
    Image3.Canvas.Create;
    ScrollBar1.Top := Image2.Top + Image2.Height + 2;
    ScrollBar1.Left := Image2.Left;
    ScrollBar1.Width := Image2.Width;
    BtnPlay.Left := 25 + Image2.Width ;
    BtnStop.Left := 65 + Image2.Width;
    BtnRec.Left := 105 + Image2.Width;
    BtnPlay.Top := 140 + Image1.Height ;
    BtnStop.Top := 140 + Image1.Height;
    BtnRec.Top := 140 + Image1.Height;
    Cursor1.Top := Image2.Top;
    Cursor1.Left := Image2.Left;
    Cursor1.Height := Image2.Height;
    Cursor1.Width := 1;
    Cursor1.Canvas.Pen.Color := clBlue;
    Cursor1.Canvas.Pen.Style := psDot;
    Cursor1.Canvas.Rectangle(0,0,Cursor1.Width,
                             Cursor1.Height);
    Cursor2.Top := Image2.Top;
    Cursor2.Left := Image2.Left;
    Cursor2.Height := Image2.Height;
    Cursor2.Width := 1;
    Cursor2.Canvas.Pen.Color := clGreen;
    Cursor2.Canvas.Pen.Style := psDot;
    Cursor2.Canvas.Rectangle(0,0,Cursor2.Width,
                             Cursor2.Height);
    Cstart.Left := Image1.Left;
    Cstop.Left := Image1.Left;
    Cstart.Top := Image1.Top;
    Cstop.Top := Image1.Top;
    Cstart.Height := Image1.Height;
    Cstop.Height := Image1.Height;
    Edit1.Visible := False;
    Edit1.Left := 40+ Image2.Width;
    StartTimeText.Left := 40+ Image2.Width;
    StopTimeText.Left := 40+ Image2.Width;
    DeltaTText.Left := 40+ Image2.Width;
    Edit1.Top := 100+ Image1.Height;
    StartTimeText.Top := 40+ Image1.Height;
    StopTimeText.Top := 60+ Image1.Height;
    DeltaTText.Top := 80+ Image1.Height;
    Bevel3.Top := Bevel2.Top;
    Bevel3.Left := StartTimeText.Left - 15;
    Bevel3.Height := 80;
    Bevel3.Width := 15 + Image1.Width - Bevel3.Left;
    BtnPlay.Top := Bevel3.top + Bevel3.Height + 10;
    BtnStop.Top := Bevel3.top + Bevel3.Height + 10;
    BtnRec.Top := Bevel3.top + Bevel3.Height + 10;
    MaxData := 2047;
    ScrollBar1.Enabled := False;
    RightClick := False;
    LeftClick :=False;
    StartPoint := 0;
    StopPoint := 0;
    StartTime := 0;
    StopTime :=0;
    Recording := False;
    Playing := False;
    FindIOAddress;
    Initial;
    CurrentFile := '';
    Modify := False;
end;

//---------------------------------------------------------------
Procedure TForm1.FormClose(Sender: TObject;
                           var Action: TCloseAction);
var  ret : integer;
begin
    if modify = true then
    begin
        ret:=MessageDlg('Current file does not save. Save
                         it? ', mtWarning, [mbYes,mbNo], 0);
        if ret = 6 then MnuFileSaveClick(Sender);
    end;
end;

//---------------------------------------------------------------
Procedure TForm1.InitGraph;
begin
    ScrollBar1.Max := round(10*MaxData/2048)-10;
```

```
    if ScrollBar1.Max > 1 then
       ScrollBar1.Enabled :=True
    Else ScrollBar1.Enabled := False;
    ScrollBar1.Position := 0;
    Form1.Cursor := crHourGlass;
    StatusBar1.SimpleText := ' Ploting Graph Please
                              Wait';
    PlotAll;
    Plot(0);
    StatusBar1.SimpleText := '';
    Form1.Cursor := crDefault;
    if CardFound then
    begin
       MnuPlayStop.Enabled := False;
       MnuPlayRecord.Enabled := True;
       MnuPlayPlay.Enabled := True;
       MnuPlayPlayAll.Enabled := True;
       BtnPlay.Enabled := True;
       BtnStop.Enabled := False;
       BtnRec.Enabled := True;
    end;
end;

//---------------------------------------------------------
Procedure TForm1.ConvertPCMToDMOD;
var  x,i: Word;
    a,b,c,d : Byte;
    Dir : Real;
    A0,A1,Vc,Vo,Vi,dt : Real;
    Mul1,Mul2,Mul3 : Real;
begin
    StatusBar1.SimpleText := ' Convert Data ';
    Vc := 0;
    Vo := 2.5;
    dt := 62.5e-6;
    a := 1;
    b := 0;
    c := 1;
    Mul1 := Exp(-dt/(18000*0.33e-6));
    Mul2 := Exp(-dt/(21300*0.33e-6));
    Mul3 := Exp(-dt/(10000*0.1e-6));
    for x:=0 to DataLength-1 do
    begin
       Data[x] := 0;
       for i:=0 to 7 do
       begin
          Vi := 5*DataPCM[x*8+i]/65536;
          if Vi<Vo then d := 0 else d:=1;
          Data[x] := Data[x] or (d shl (7-i));
               if ((a=b)and(a=c)and(a=d)) then
          begin
             A0 := Vc-5;
                  Vc := Mul1*A0 + 5.0;
          end
               else
               begin
             A0 := Vc;
             Vc := Mul2*A0;
          end;
          if (d=1) then Dir:=1 else Dir:=-1;
          A1 := Vo - 2.5 - Dir*(Vc/0.13);
               Vo := A1*Mul3 + 2.5 + Dir*(Vc/0.13);
          a := b;
          b := c;
          c := d;
       end;
    end;
    StatusBar1.SimpleText := '';
end;

//---------------------------------------------------------
Procedure TForm1.ConvertDMODToPCM;
var  x,i: Word;
    a,b,c,d : Byte;
    Dir : Real;
    A0,A1,Vc,Vo,dt : Real;
    Mul1,Mul2,Mul3 : Real;
begin
    StatusBar1.SimpleText := ' Convert Data ';
    Vc := 0;
    Vo := 2.5;
    dt := 62.5e-6;
    b := 0;
    c := 1;
    d := 0;
    Mul1 := Exp(-dt/(18000*0.33e-6));
    Mul2 := Exp(-dt/(21300*0.33e-6));
    Mul3 := Exp(-dt/(10000*0.1e-6));
    for x:=0 to DataLength-1 do
    begin
       for i:=0 to 7 do
       begin
          a := b;
          b := c;
          c := d;
               d := (data[x] shr (7-i)) and $01;
               if ((a=b)and(a=c)and(a=d)) then
          begin
             A0 := Vc-5;
                  Vc := Mul1*A0 + 5.0;
          end
               else
               begin
             A0 := Vc;
             Vc := Mul2*A0;
          end;
          if (d=1) then Dir:=1 else Dir:=-1;
          A1 := Vo - 2.5 - Dir*(Vc/0.13);
               Vo := A1*Mul3 + 2.5 + Dir*(Vc/0.13);
          DataPCM[x*8+i] := round(65536*Vo/5);
       end;
    end;
    StatusBar1.SimpleText := '';
end;

//---------------------------------------------------------
Procedure TForm1.PlotAll;
var  x: Word;
    y : integer;
    Rect : TRect;
    Mul5 : Real;
begin
    Rect.Left := 0;
    Rect.Top := 0;
    Rect.Right := Image1.Width;
    Rect.Bottom := Image1.Height;
    Image1.Canvas.FillRect(Rect);
    Image1.Canvas.Pen.Color := clRed ;
    Image1.Canvas.MoveTo(0,round(Image1.Height/2));
    Mul5 := Image1.Width/(MaxData*8);
    for x:=0 to DataLength-1 do
    begin
       y := Image1.Height - round(Image1.Height*
                                  DataPCM[x*8]/65536);
```

```
      Image1.Canvas.lineto(round((x*8)*Mul5),y);
    end;
end;

//---------------------------------------------------------------
Procedure TForm1.Plot(time : integer);
var  x : Word;
    Vo,Mul1,Mul2 : Real;
    y,Point : integer;
    Rect : TRect;
begin
    Mul1 := Image2.Height/5;
    Mul2 := Image2.Width/(2048*8);
    Rect.Top := 0;
    Rect.Left := 0;
    Rect.Right := Image2.Width;
    Rect.Bottom := Image2.Height;
    Image2.Canvas.FillRect(Rect);
    Image2.Canvas.Pen.Color := clRed ;
    Image2.Canvas.MoveTo(0,round(Image2.Height/2));
    for x:=0 to 2048 do
    if x+round(time*204.8)<DataLength then
    begin
       Vo:= 5*DataPCM[x*8+round(8*Time*204.8)]/
            65536;
       y := Image2.Height - round(Mul1*Vo);
       Image2.Canvas.lineto(round((x*8)*Mul2),y);
    end;
    Image3.Canvas.Pen.Color := clBlue;
    Image3.Left := Image1.Left+round(
            (Image1.Width*Time/10)/(MaxData/2048));
    Image3.Width := round(Image1.Width*2048/
                          MaxData);
    Image3.Canvas.Rectangle(0,0,Image3.Width,
                            Image3.Height);
    Point := round(Time*2048/10);
    if StartPoint < Point then
       Cursor1.Left := Image2.Left else
    if StartPoint > Point+2048 then
       Cursor1.Left := Image2.Left+Image2.Width
    else Cursor1.Left := Image2.Left+round((StartPoint-
                        Point)*Image2.Width/2048);
    if StopPoint < Point then
       Cursor2.Left := Image2.Left else
    if StopPoint > Point+2048 then
       Cursor2.Left := Image2.Left+Image2.Width
    else Cursor2.Left := Image2.Left+round((StopPoint-
                        Point)*Image2.Width/2048);
end;

//---------------------------------------------------------------
Procedure TForm1.ScrollBar1Change(Sender:
                                  TObject);
begin
    Plot(ScrollBar1.Position);
end;

//---------------------------------------------------------------
Procedure TForm1.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var Time : Integer;
begin
    Time := round(10*X*MaxData/(Image1.Width*
                  2048))-1;
    if Time < 0 Then Time := 0;
    if Time > ScrollBar1.Max then
       Time := ScrollBar1.Max;
    ScrollBar1.Position := Time;
end;

//---------------------------------------------------------------
Procedure TForm1.Image2MouseDown(Sender:
                  TObject; Button: TMouseButton;
                  Shift: TShiftState; X, Y: Integer);
var  Time,DeltaT : Real;
    Point : Integer;
begin
    Time := (ScrollBar1.Position/10) +
            (X/Image2.Width);
    Point := round(Time * 2048);
    if (Button = mbLeft) and (Point<StopPoint)then
    begin
       LeftClick := True;
       Cursor1.Left := Image2.Left + X;
       Cstart.Left := round(Image1.Width*Point/
                  MaxData)+Image1.Left;
       StartPoint := Point;
       StartTime := Time;
       StartTimeText.Caption := 'Start : '+inttostr(round(
                              Time*1000))+' ms';
       DeltaT := StopTime-StartTime;
       DeltaTText.Caption := 'Delta t : '+inttostr(round(
                           DeltaT*1000))+' ms';
    end else
    if (Button = mbRight) and (point>StartPoint) and
      (Point<DataLength) Then
    begin
       RightClick := True;
       Cursor2.Left := Image2.Left + X;
       Cstop.Left := round(Image1.Width*Point/
                  MaxData)+Image1.Left;
       StopPoint := Point;
       StopTime := Time;
       StopTimeText.Caption := 'Stop : '+inttostr(round(
                              Time*1000))+' ms';
       DeltaT := StopTime-StartTime;
       DeltaTText.Caption := 'Delta t : '+inttostr(round(
                           DeltaT*1000))+' ms';
    end;
end;

//---------------------------------------------------------------
Procedure TForm1.Image2MouseUp(Sender: TObject;
                  Button: TMouseButton;
                  Shift: TShiftState; X, Y: Integer);
begin
    RightClick := False;
    LeftClick := False;
end;

//---------------------------------------------------------------
Procedure TForm1.Initial;
begin
  asm
     mov AL,$99
     mov DX,PortControl1
     out DX,AL      (* 8255#1 A-IN B-OUT C-IN *)
     mov AL,$90
     mov DX,PortControl2
     out DX,AL    (* 8255#2 A-IN B-OUT C-OUT *)
  end;
  DataPortB1 := $6E;
  asm
```

```
      mov AL,DataPortB1
      mov DX,PortB1
      out DX,AL
  end;
  ResetCounter;
end;

//---------------------------------------------------------------
Procedure TForm1.ResetCounter;
begin
   asm
      or  DataPortB1,$10
      mov DX,PortB1
      mov AL,DataPortB1
      out DX,AL        (* Send Hi To Reset Counter *)
      and DataPortB1,$EF;
      mov DX,PortB1
      mov AL,DataPortB1
      out DX,AL          (* Reset Counter Complete *)
   end;
end;

//---------------------------------------------------------------
Procedure TForm1.Starting;
begin
  ResetCounter;
  asm
     mov DataPortB1,$06
     mov AL,DataPortB1
     mov DX,PortB1
     out DX,AL
  end;
end;

//---------------------------------------------------------------
Procedure TForm1.SetEndPoint(EndAddress:integer);
begin
   DataPortC2 := (EndAddress and $FF00) shr 8;
   DataPortB2 := EndAddress and $FF ;
   asm
      mov AL,DataPortB2
      mov DX,PortB2
      out DX,AL          (* Send High Byte Address *)
      mov AL,DataPortC2
      mov DX,PortC2
      out DX,AL          (* Send Low Byte Address *)
   end;
end;

//---------------------------------------------------------------
Procedure TForm1.PlayBack;
begin
   ResetCounter;
   DataPortB1 := $2A;
   asm
      mov AL,DataPortB1
      mov DX,PortB1
      out DX,AL
   end;
end;

//---------------------------------------------------------------
Procedure TForm1.StopRunning;
begin
   DataPortB1 := $6E;
   asm
      mov AL,DataPortB1
      mov DX,PortB1
      out DX,AL
   end;
end;

//---------------------------------------------------------------
Procedure TForm1.ReadData;
var  i : word;
   dat : byte;
begin
   StatusBar1.SimpleText := ' Reading Data ';
   ResetCounter;
   for i:=0 to DataLength-1 do
   begin
      asm
         and DataPortB1,$FB;    (* Reset Bit P_RD *)
         mov AL,DataPortB1
         mov DX,PortB1
         out DX,AL
         mov DX,PortA2
         in  AL,DX
         mov dat,AL             (* Read Data *)
         or  DataPortB1,$04;    (* Set Bit P_RD *)
         mov AL,DataPortB1
         mov DX,PortB1
         out DX,AL
         or  DataPortB1,$01;    (* Set Bit P_CLK *)
         mov AL,DataPortB1
         mov DX,PortB1
         out DX,AL
         and DataPortB1,$FE;    (* Reset Bit P_CLK *)
         mov AL,DataPortB1
         mov DX,PortB1
         out DX,AL
      end;
      data[i] := dat;
   end;
   StatusBar1.SimpleText := '';
   ConvertDMODToPCM;
end;

//---------------------------------------------------------------
Procedure TForm1.WriteData(Start,Stop:word);
var  i : word;
   dat : byte;
begin
   asm
   mov AL,$80 (* 8255#2  A-OUT  B-OUT  C-OUT*)
   mov DX,PortControl2
   out DX,AL
   end;
   ResetCounter;
   for i:=Start to Stop do
   begin
      dat := data[i];
      asm
         mov AL,dat             (* Write Data *)
         mov DX,PortA2
         out DX,AL
         and DataPortB1,$F7     (* Reset Bit P_WR *)
         mov AL,DataPortB1
         mov DX,PortB1
         out DX,AL
         or  DataPortB1,$08     (* Set Bit P_WR *)
         mov AL,DataPortB1
         mov DX,PortB1
         out DX,AL
```

```
        or DataPortB1,$01        (* Set Bit P_CLK *)
        mov AL,DataPortB1
        mov DX,PortB1
        out DX,AL
        and DataPortB1,$FE    (* Reset Bit P_CLK *)
        mov AL,DataPortB1
        mov DX,PortB1
        out DX,AL
      end;
  end;
  asm
    mov AL,$90 (* 8255#2 A-IN B-OUT C-OUT*)
    mov DX,PortControl2
    out DX,AL
  end;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuFileOpenClick(Sender:
                                        TObject);
var  FileName : string;
  LoadFile : file;
  FSize : Word;
  ret : integer;
begin
  if modify = true then
  begin
    ret:=MessageDlg('Current file does not save. Save
                it? ', mtWarning, [mbYes,mbNo], 0);
    if ret = 6 then MnuFileSaveClick(Sender);
  end;
  OpenDialog1.Filename := '';
  OpenDialog1.Execute;
  FileName := OpenDialog1.Filename;
  if FileExists(FileName) then
  begin
    MnuFileSave.Enabled := true;
    MnuFileSaveAll.Enabled := true;
    MnuFileSaveSelect.Enabled := true;
    MnuEdit.Enabled := true;
    CurrentFile := FileName;
    Modify := False;
    Form1.Caption := 'D-MOD Recorder
                    ['+FileName+']';
    AssignFile(LoadFile,FileName);
    Reset(LoadFile,1);
    FSize := FileSize(LoadFile);
    DataLength := FSize;
    MaxData := 0;
    repeat
      MaxData := MaxData +2048;
    until MaxData >= DataLength;
    BlockRead(LoadFile, Data, DataLength);
    CloseFile(LoadFile);
    ConvertDMODToPCM;
    InitGraph;
    MnuEditSelectAllClick(Sender);
    MnuEffect.Enabled := True;
  end
  else if FileName <> '' then
  begin
    MessageDlg('File does not Exist', mtError,
              [mbok], 0);
  end;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuFileSaveClick(Sender:
                                        TObject);
var  SaveFile : file;
begin
  Modify := False;
  Form1.Caption := 'D-MOD Recorder
                    ['+CurrentFile+']';
  AssignFile(SaveFile,CurrentFile);
  Rewrite(SaveFile,1);
  BlockWrite(SaveFile, Data, DataLength);
  CloseFile(SaveFile);
end;

//---------------------------------------------------------------
Procedure TForm1.MnuFileSaveAllClick(Sender:
                                        TObject);
var  FileName : string;
  SaveFile : file;
  ret : integer;
begin
  SaveDialog1.Filename := '';
  SaveDialog1.Execute;
  if SaveDialog1.Filename <> '' then
  begin
    ret := 6;
    FileName := SaveDialog1.Filename;
    if ExtractFileExt(FileName) ='' then
      Filename := FileName + '.dm';
    if FileExists(FileName) then
      ret := MessageDlg('Do you want to replace the
        existing '''+ExtractFileName(FileName)+''',
        mtConfirmation, [mbYes,mbNo], 0);
    if ret=6 then
    begin
      Modify := False;
      CurrentFile := FileName;
      Form1.Caption := 'D-MOD Recorder
                    ['+FileName+']';
      AssignFile(SaveFile,FileName);
      Rewrite(SaveFile,1);
      BlockWrite(SaveFile, Data, DataLength);
      CloseFile(SaveFile);
    end;
  end;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuFileSaveSelectClick(Sender:
                                        TObject);
var  FileName : string;
  SaveFile : file of Byte;
  ret,i : word;
begin
  SaveDialog1.Filename := '';
  SaveDialog1.Execute;
  if SaveDialog1.Filename<>'' then
  begin
    ret := 6;
    FileName := SaveDialog1.Filename;
    if ExtractFileExt(FileName) ='' then
      Filename := FileName + '.dm';
    if FileExists(FileName) then
      ret := MessageDlg('Do you want to replace the
        existing '''+ExtractFileName(FileName)+''',
        mtConfirmation, [mbYes,mbNo], 0);
    if ret=6 then
    begin
```

```
            AssignFile(SaveFile,FileName);
            Rewrite(SaveFile);
            for i:=StartPoint to StopPoint do
               write(SaveFile,Data[i]);
            CloseFile(SaveFile);
         end;
      end;
end;

//---------------------------------------------------------------
procedure TForm1.MnuFileConvertClick(Sender:
                                          TObject);
begin
   Application.CreateForm(TForm2,Form2);
   Form2.Show;
   Form1.Visible := False;
   MnuFileConvert.Enabled := False;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuFileExitClick(Sender: TObject);
var  ret : integer;
begin
   if modify = true then
   begin
      ret:=MessageDlg('Current file does not save. Save
                  it? ', mtWarning, [mbYes,mbNo], 0);
      if ret = 6 then MnuFileSaveClick(Sender);
   end;
   Halt(1);
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEditCutClick(Sender: TObject);
var  i : integer;
begin
   for i:= StopPoint to DataLength do
      Data[StartPoint+(i - StopPoint)] := Data[i];
   DataLength := DataLength - (StopPoint-StartPoint);
   Edit1.Text := IntToStr(DataLength);
   ConvertDMODToPCM;
   MaxData := 0;
   repeat
      MaxData := MaxData +2048;
   until MaxData >= DataLength;
   InitGraph;
   MnuEditSelectAllClick(Sender);
   Modify := True;
   Form1.Caption := 'D-MOD Recorder
                  [*'+CurrentFile+']';
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEditPasteClick(Sender:
                                          TObject);
var  FileName : string;
   LoadFile : File of byte;
   FSize,i : integer;
begin
   OpenDialog1.Filename := '';
   OpenDialog1.Execute;
   FileName := OpenDialog1.Filename;
   if FileExists(FileName) then
   begin
      AssignFile(LoadFile,FileName);
      Reset(LoadFile);
      FSize := FileSize(LoadFile);
      if FSize+DataLength <= $8000 then
      begin
         Modify := true;
         Form1.Caption := 'D-MOD Recorder
                  [*'+CurrentFile+']';
         DataLength := DataLength + FSize;
         MaxData := 0;
         repeat
            MaxData := MaxData +2048;
         until MaxData >= DataLength;
         i := 0;
         repeat
            Read(LoadFile,Data[DataLength-Fsize+i]);
            i := i+1;
         until eof(LoadFile);
         CloseFile(LoadFile);
         ConvertDMODToPCM;
         InitGraph;
         MnuEditSelectAllClick(Sender);
      end
      else
         MessageDlg('File is too big !', mtError,
                  [mbok], 0);
   end
   else if FileName <> '' then
   begin
      MessageDlg('File does not Exist', mtError,
                  [mbok], 0);
   end;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEditInsertClick(Sender:
                                          TObject);
var  FileName : string;
   LoadFile : File of byte;
   FSize,i : integer;
begin
   OpenDialog1.Filename := '';
   OpenDialog1.Execute;
   FileName := OpenDialog1.Filename;
   if FileExists(FileName) then
   begin
      AssignFile(LoadFile,FileName);
      Reset(LoadFile);
      FSize := FileSize(LoadFile);
      if FSize+DataLength <= $8000 then
      begin
         Modify := true;
         Form1.Caption := 'D-MOD Recorder
                  [*'+CurrentFile+']';
         DataLength := DataLength + FSize;
         MaxData := 0;
         repeat
            MaxData := MaxData +2048;
         until MaxData >= DataLength;
        for i:=0 to Datalength - (StartPoint+Fsize-1) do
         begin
            Data[DataLength-i] := Data[DataLength-
                              Fsize-i]
         end;
         i := 0;
         repeat
            Read(LoadFile,Data[StartPoint+i]);
            i := i+1;
         until eof(LoadFile);
         CloseFile(LoadFile);
```

```
            ConvertDMODToPCM;
            InitGraph;
            MnuEditSelectAllClick(Sender);
        end
        else
            MessageDlg('File is too big !', mtError,
                        [mbok], 0);
    end
    else if FileName <> '' then
    begin
        MessageDlg('File does not Exist', mtError,
                    [mbok], 0);
    end;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEditSelectAllClick(Sender:
                                        TObject);
var  Point : integer;
begin
    StartPoint := 0;
    StopPoint := DataLength;
    StartTime := 0;
    StopTime := DataLength/2048;
    Cursor1.Left := Image2.Left;
    Point := round(ScrollBar1.Position*2048/10);
    if StopPoint > Point+2048 then
        Cursor2.Left := Image2.Left+Image2.Width
    else Cursor2.Left := Image2.Left+round((StopPoint-
                            Point)*Image2.Width/2048);
    CStart.Left := Image1.Left;
    CStop.Left := Image1.Left+round(image1.Width*
                    DataLength/MaxData);
    StartTimeText.Caption := 'Start : 0 ms';
    StopTimeText.Caption := 'Stop : '+inttostr(round(
                        DataLength*1000/2048))+' ms';
    DeltaTText.Caption := 'Delta t : '+inttostr(round(
                        DataLength*1000/2048))+' ms';
end;

//---------------------------------------------------------------
Procedure TForm1.MnuPlayRecordClick(Sender:
                                        TObject);
begin
    MnuFileSave.Enabled := true;
    MnuFileSaveAll.Enabled := true;
    MnuFileSaveSelect.Enabled := true;
    MnuEdit.Enabled := true;
    CurrentFile := '';
    Modify := True;
    Form1.Caption := 'D-MOD Recorder [Untitled]';
    MaxData := $8000;
    DataLength := $8000;
    SetEndPoint(MaxData);
    Recording := True;
    MnuPlayStop.Enabled := True;
    MnuPlayRecord.Enabled := False;
    MnuPlayPlay.Enabled := False;
    MnuPlayPlayAll.Enabled := False;
    BtnPlay.Enabled := False;
    BtnStop.Enabled := True;
    BtnRec.Enabled := False;
    Timer1.Enabled := True;
    Starting;
    MnuEffect.Enabled := True;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuPlayPlayClick(Sender:
TObject);
begin
    WriteData(StartPoint,StopPoint);
    SetEndPoint(StopPoint-StartPoint+1);
    Playing := True;
    MnuPlayStop.Enabled := True;
    MnuPlayRecord.Enabled := False;
    MnuPlayPlay.Enabled := False;
    MnuPlayPlayAll.Enabled := False;
    BtnPlay.Enabled := False;
    BtnStop.Enabled := True;
    BtnRec.Enabled := False;
    Timer1.Enabled := True;
    PlayBack;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuPlayPlayAllClick(Sender:
                                        TObject);
begin
    WriteData(0,DataLength);
    SetEndPoint(DataLength);
    Playing := True;
    MnuPlayStop.Enabled := True;
    MnuPlayRecord.Enabled := False;
    MnuPlayPlay.Enabled := False;
    MnuPlayPlayAll.Enabled := False;
    BtnPlay.Enabled := False;
    BtnStop.Enabled := True;
    BtnRec.Enabled := False;
    Timer1.Enabled := True;
    PlayBack;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuPlayStopClick(Sender:
                                        TObject);
begin
    if Recording Then
    begin
      StopRunning;
      asm
        mov AL,$9B
        mov DX,PortControl2
        out DX,AL    (* 8255#2 A-IN B-IN C-IN *)
        and DataPortB1,$FD   (* Reset Bit P_ADD *)
        mov AL,DataPortB1
        mov DX,PortB1
        out DX,AL
        mov DX,PortB2
        in  AL,DX            (* Read Low Address *)
        mov CL,AL
        mov DX,PortC2
        in  AL,DX            (* Read High Address *)
        mov CH,AL
        mov DataLength,CX
        or  DataPortB1,$02       (* Set Bit P_ADD *)
        mov AL,DataPortB1
        mov DX,PortB1
        out DX,AL
        mov AL,$90
        mov DX,PortControl2
       out DX,AL (* 8255#2 A-IN B-OUT C-OUT *)
      end;
      ReadData;
```

```
        MaxData := 0;
        repeat
            MaxData := MaxData +2048;
        until MaxData >= DataLength;
        Recording := False;
        InitGraph;
        MnuEditSelectAllClick(Sender);
    end;
    if Playing then
    begin
        StopRunning;
        Playing := False;
        MnuPlayStop.Enabled := False;
        MnuPlayRecord.Enabled := True;
        MnuPlayPlay.Enabled := True;
        MnuPlayPlayAll.Enabled := True;
        BtnPlay.Enabled := True;
        BtnStop.Enabled := False;
        BtnRec.Enabled := True;
    end;
    Timer1.Enabled := False;
end;

//---------------------------------------------------------------
Procedure TForm1.Timer1Timer(Sender: TObject);
begin
    if Playing then
    begin
        asm
            mov DX,PortC1
            in  AL,DX
            mov DataPortC1,AL
        end;
        if (DataPortC1 and $01) = 0 then
        begin
            Playing := False;
            StopRunning;
            MnuPlayStop.Enabled := False;
            MnuPlayRecord.Enabled := True;
            MnuPlayPlay.Enabled := True;
            MnuPlayPlayAll.Enabled := True;
            BtnPlay.Enabled := True;
            BtnStop.Enabled := False;
            BtnRec.Enabled := True;
            Timer1.Enabled := False;
        end;
    end;
    if Recording then
    begin
        asm
            mov DX,PortC1
            in  AL,DX
            mov DataPortC1,AL
        end;
        if (DataPortC1 and $01) = 0 then
        begin
            Recording := False;
            StopRunning;
            ReadData;
            InitGraph;
            MnuEditSelectAllClick(Sender);
            Timer1.Enabled := False;
        end;
    end;
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEffectMuteClick(Sender:
                                    TObject);
var  i: longint;
begin
    Modify := True;
    Form1.Caption := 'D-MOD Recorder
                    [*'+CurrentFile+']';
    for i:=StartPoint*8 to StopPoint*8 do
        DataPCM[i] := 32768 ;
    ConvertPCMToDMOD;
    PlotAll;
    Plot(ScrollBar1.Position);
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEffectFilter1Click(Sender:
                                    TObject);
begin
    Filter(1000);
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEffectFilter2Click(Sender:
                                    TObject);
begin
    Filter(2000);
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEffectFilter3Click(Sender:
                                    TObject);
begin
    Filter(3000);
end;

//---------------------------------------------------------------
Procedure TForm1.MnuEffectFilter4Click(Sender:
                                    TObject);
begin
    Filter(4000);
end;

//---------------------------------------------------------------
Procedure TForm1.Filter(Freq : Integer);
const M=101;
var  h : array[0..M-1] of real;
    tmp : array[1..M] of integer;
    i,n,k,MM : integer;
    wc : real;
    Result : real;
begin
    (* Generated Impulse Response of Ideal Low Pass
       Filter at Wc *)
    Modify := True;
    Form1.Caption := 'D-MOD Recorder
                    [*'+CurrentFile+']';
    MM := DataLength*8;
    wc := Freq*2*Pi/16000;
    for i := 0 to M-1 do
    begin
        h[i] := 0.6366*sin(wc*(i-50+0.00001))/
                (2*(i-50+0.00001)) ;
        tmp[i+1]:=0;
    end;
    (* Multiply by Hammimg window *)
    for i := 0 to M-1 do
    begin
```

```
      h[i] := h[i] * ( 0.54-0.46*cos(2*Pi*i/(M-1)));
      tmp[i+1]:=0;
    end;
    for n:=1 to M+MM-(M div 2) do
    begin
      (* Save data into tmp  and shift data *)
      for i:=1 to M do
        if i<>M then tmp[i] := tmp[i+1] else
          if n < MM+1 then tmp[i]:= DataPCM[n]
            else tmp[i]:=0;
      (* Convolution *)
      if (n > (M div 2) )and( n-(M div 2)<= MM) then
      begin
        result := 0;
        for k:=0 to M-1 do
          if (n-k > 0) then
            result := result + h[k]*tmp[M-k];
        DataPCM[n-(M div 2)] := round(result);
      end;
    end;
    ConvertPCMToDMOD;
    PlotAll;
    Plot(ScrollBar1.Position);
end;

//-------------------------------------------------------------
Procedure TForm1.FindIOAddress;
var  i,j : integer;
    addr : byte;
begin
    i:=0;
    repeat
      CardFound := true;
      PortControl1 := $303 + i*8;
      PortControl2 := $307 + i*8;
      PortB1 := $301 + i*8;
      PortB2 := $305 + i*8;
      DataPortB1 := $6C;
      asm
        mov AL,$99
        mov DX,PortControl1
        out DX,AL (* 8255#1  A-IN  B-OUT  C-IN *)
        mov AL,$9B
        mov DX,PortControl2
        out DX,AL   (* 8255#2  A-IN  B-IN  C-IN *)
        mov AL,DataPortB1
        mov DX,PortB1
        out DX,AL
      end;
      ResetCounter;
      j := 0;
      repeat
        asm  { Count Up 1 Address}
          or  DataPortB1,$01;      (* Set Bit P_CLK *)
          mov AL,DataPortB1
          mov DX,PortB1
          out DX,AL
          and DataPortB1,$FE; (* Reset Bit P_CLK *)
          mov AL,DataPortB1
          mov DX,PortB1
          out DX,AL
          mov DX,PortB2
          in  AL,DX              (* Read Low Address *)
          mov addr,AL
        end;
        j := j+1;
        if addr <> j then CardFound := false;
      until (j=5) or (not CardFound);
      i := i+1;
    until CardFound or (i=8) ;
    i := i-1;
    if CardFound then
    begin
      PortA1 := $300 + i*8;
      PortB1 := $301 + i*8;
      PortC1 := $302 + i*8;
      PortA2 := $304 + i*8;
      PortB2 := $305 + i*8;
      PortC2 := $306 + i*8;
      PortControl1 := $303 + i*8;
      PortControl2 := $307 + i*8;
      MessageDlg('Card found at '+inttohex(
            PortA1,3)+'H', mtInformation, [mbOK], 0);
    end
    else
    begin
      BtnRec.Enabled := false;
      BtnPlay.Enabled := false;
      BtnStop.Enabled := false;
      MnuPlay.Enabled := false;
      MessageDlg('Card not found ', mtError, [mbOK],
                  0);
    end;
end;
end.

//=============================================================
unit CnvFile;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,ComCtrls, FileCtrl, StdCtrls,
  Buttons, Gauges, ExtCtrls;

type
  TForm2 = class(TForm)
    DriveComboBox1: TDriveComboBox;
    DirectoryListBox1: TDirectoryListBox;
    FileListBox1: TFileListBox;
    FilterComboBox1: TFilterComboBox;
    ListBox1: TListBox;
    BtnRemove: TBitBtn;
    BtnAdd: TBitBtn;
    BtnUp: TBitBtn;
    BtnDown: TBitBtn;
    Label1: TLabel;
    BtnSaveList: TBitBtn;
    BtnClear: TBitBtn;
    BtnExit: TBitBtn;
    Label2: TLabel;
    SaveDialog1: TSaveDialog;
    BtnOpen: TBitBtn;
    SaveDialog2: TSaveDialog;
    BtnSaveBin: TBitBtn;
    OpenDialog1: TOpenDialog;
    ProgressPanel: TPanel;
    Gauge1: TGauge;

    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject;
      var Action: TCloseAction);
    procedure BtnAddClick(Sender: TObject);
    procedure BtnRemoveClick(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
```

```
    procedure BtnOpenClick(Sender: TObject);
    procedure BtnSaveBinClick(Sender: TObject);
    procedure BtnSaveListClick(Sender: TObject);
    procedure BtnClearClick(Sender: TObject);
    procedure BtnExitClick(Sender: TObject);
    procedure BtnUpClick(Sender: TObject);
    procedure BtnDownClick(Sender: TObject);
    procedure PrintDetail;

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;
  Filename : array[0..255] of string;
  NoOfFile : integer;
  TotalSize : longint;

implementation
uses MainDREC;
{$R *.DFM}

//---------------------------------------------------------------
Procedure TForm2.FormCreate(Sender: TObject);
begin
    Form2.Left := Form1.Left;
    Form2.Top  := Form1.Top;
    NoOfFile := 0;
    BtnUp.Enabled := False;
    BtnDown.Enabled := False;
    TotalSize := 0;
end;

//---------------------------------------------------------------
procedure TForm2.FormClose(Sender: TObject;
                           var Action: TCloseAction);
begin
    Form1.MnuFileConvert.Enabled := true;
    Form1.Visible := True;
end;

//---------------------------------------------------------------
Procedure TForm2.BtnAddClick(Sender: TObject);
var  i : integer;
    dup : boolean;
    f : file of byte;
begin
    if FileListBox1.FileName<>'' then
    begin
        dup := false;
        for i := 0 to NoOfFile-1 do
        begin
            if FileListBox1.FileName = FileName[i] then
                dup := true;
        end;
        if (not dup) and (NoOfFile<=255) then
        begin
            AssignFile(f,FileListBox1.FileName);
            Reset(f);
            if (TotalSize + FileSize(f))<=$100000 then
            begin
                FileName[NoOfFile] :=
                                FileListBox1.FileName;
                NoOfFile := NoOfFile+1;
                BtnSaveBin.Enabled := true;
                BtnSaveList.Enabled := true;
                BtnRemove.Enabled := true;
                ListBox1.Items.Add(inttostr(NoOfFile) +
                        ' : ' + ExtractFileName(FileName[
                        NoOfFile-1]));
                PrintDetail;
                ListBox1.SetFocus;
                ListBox1.ItemIndex := NoOfFile-1;
                ListBox1Click(Sender);
            end
            else
                MessageDlg('Cannot select more than
                            1MBytes', mtError, [mbok], 0);
            CloseFile(f);
        end
        else
        if dup then
            MessageDlg('File already selected', mtError,
                        [mbok], 0)
        else
            MessageDlg('Cannot select more than 256
                        files', mtError, [mbok], 0)
    end;
end;

//---------------------------------------------------------------
Procedure TForm2.BtnRemoveClick(Sender: TObject);
var  i,index : integer;
begin
    index := Listbox1.ItemIndex;
    if index>=0 then
    begin
        for i:=index to NoOfFile do
            FileName[i] := FileName[i+1];
        NoOfFile := NoOfFile-1;
        if NoOfFile=0 then
        begin
            BtnSaveBin.Enabled := False;
            BtnSaveList.Enabled := False;
            Label2.Caption := 'Selected : 0 bytes';
        end
        else
        begin
            BtnSaveBin.Enabled := true;
            BtnSaveList.Enabled := true;
        end;
        PrintDetail;
        ListBox1.Clear;
        if NoOfFile = 0 then BtnRemove.Enabled := false
        else
            for i:=0 to NoOfFile-1 do
                ListBox1.Items.Add(inttostr(i+1) + ' : ' +
                            ExtractFileName(FileName[i]));
        if index < NoOfFile then
            Listbox1.ItemIndex := index
        else Listbox1.ItemIndex := index-1;
        ListBox1Click(Sender);
    end;
end;

//---------------------------------------------------------------
Procedure TForm2.ListBox1Click(Sender: TObject);
var  f : file of byte;
    size : string;
begin
    if Listbox1.ItemIndex < ListBox1.Items.Count-1
```

```
        then BtnDown.Enabled := true;
    if Listbox1.ItemIndex = ListBox1.Items.Count-1
        then BtnDown.Enabled := false;
    if Listbox1.ItemIndex > 0 Then
        BtnUp.Enabled := true;
    if Listbox1.ItemIndex = 0 Then
        BtnUp.Enabled := False;
    if NoOfFile >0 then
    begin
        AssignFile(f,FileName[ListBox1.ItemIndex]);
        reset(f);
        size := inttostr(FileSize(f));
        CloseFile(f);
        Label2.Caption := 'Selected : ' + size + ' bytes';
    end
    else Label2.Caption := 'Selected : 0 bytes';
end;

//-------------------------------------------------------------------
Procedure TForm2.BtnUpClick(Sender: TObject);
var  tmp : string;
    index : integer;
begin
    index := ListBox1.ItemIndex;
    if index > 0 then
    begin
        tmp := FileName[index-1];
        FileName[index-1] := FileName[index];
        FileName[index] := tmp;
        ListBox1.Items.Delete(Index);
        ListBox1.Items.Insert(Index,inttostr(Index) + ' : '
                + ExtractFileName(FileName[Index-1]));
        ListBox1.Items.Delete(Index-1);
        ListBox1.Items.Insert(Index,inttostr(Index+1)+' : '
                + ExtractFileName(FileName[Index]));
        ListBox1.ItemIndex := Index-1;
        if ListBox1.Itemindex = 0 then
            BtnUp.Enabled := false;
        BtnDown.Enabled := true;
    end;
end;

//-------------------------------------------------------------------
Procedure TForm2.BtnDownClick(Sender: TObject);
var  tmp : string;
    index : integer;
begin
    index := ListBox1.ItemIndex;
    if index < NoOfFile-1 then
    begin
        tmp := FileName[index+1];
        FileName[index+1] := FileName[index];
        FileName[index] := tmp;
        ListBox1.Items.Delete(Index);
        ListBox1.Items.Insert(Index+1,inttostr(Index+2)
            + ' : ' + ExtractFileName(FileName[Index+1]));
        ListBox1.Items.Delete(Index);
        ListBox1.Items.Insert(Index,inttostr(Index+1)+' : '
                + ExtractFileName(FileName[Index]));
        ListBox1.ItemIndex := index+1;
        if ListBox1.Itemindex = ListBox1.Items.Count-1
            then BtnDown.Enabled := false;
        BtnUp.Enabled := true;
    end;
end;

//-------------------------------------------------------------------
Procedure TForm2.PrintDetail;
var  i : integer;
    size : longint;
    f : file of byte;
begin
    size := 0;
    for i:=0 to NoOfFile-1 do
    begin
        AssignFile(f,FileName[i]);
        Reset(f);
        size := size + FileSize(f);
        CloseFile(f);
    end;
    Label1.Caption := 'Total : ' + inttostr(NoOfFile) +
                        ' files ';
    Label1.Caption := Label1.Caption + inttostr(size) +
                        ' bytes';
    TotalSize := size;
end;

//-------------------------------------------------------------------
Procedure TForm2.BtnExitClick(Sender: TObject);
begin
    Close;
end;

//-------------------------------------------------------------------
Procedure TForm2.BtnClearClick(Sender: TObject);
var  ret : integer;
begin
    ret:=MessageDlg('Do you want to clear all list ?',
                mtConfirmation, [mbYes,mbNo], 0);
    if ret = 6 then
    begin
        NoOfFile := 0;
        ListBox1.Clear;
        PrintDetail;
        Label2.Caption := 'Selected : 0 bytes';
        BtnSaveBin.Enabled := False;
        BtnSaveList.Enabled := False;
    end;
end;

//-------------------------------------------------------------------
Procedure TForm2.BtnOpenClick(Sender: TObject);
var  FiName : string;
    LoadFile : TextFile;
begin
    OpenDialog1.Filename := '';
    OpenDialog1.Execute;
    FiName := OpenDialog1.Filename;
    if FileExists(FiName) then
    begin
        NoOfFile := 0;
        ListBox1.Clear;
        AssignFile(LoadFile,FiName);
        Reset(LoadFile);
        Repeat
            Readln(LoadFile,FileName[NoOfFile]);
            ListBox1.Items.Add(inttostr(NoOfFile+1)+':'
                +ExtractFileName(FileName[NoOfFile]));
            NoOfFile := NoOfFile + 1;
        Until eof(LoadFile);
        CloseFile(LoadFile);
        PrintDetail;
        BtnSaveBin.Enabled := true;
        BtnSaveList.Enabled := true;
```

```
    end
    else if FiName <> '' then
    begin
       MessageDlg('File does not Exist', mtError,
                  [mbok], 0);
    end;
end;

//---------------------------------------------------------------
Procedure TForm2.BtnSaveListClick(Sender: TObject);
var  FiName : string;
    ret,i : integer;
    SaveFile : TextFile;
begin
    SaveDialog1.Filename := '';
    SaveDialog1.Execute;
    if SaveDialog1.Filename <> '' then
    begin
       ret := 6;
       FiName := SaveDialog1.Filename;
       if ExtractFileExt(FiName) ='' then
         Finame := FiName + '.lst';
       if FileExists(FiName) then
          ret := MessageDlg('Do you want to replace the
               existing "'+ExtractFileName(FiName)+'"',
               mtConfirmation, [mbYes,mbNo], 0);
       if ret=6 then
       begin
          Modify := False;
          AssignFile(SaveFile,FiName);
          Rewrite(SaveFile);
          for i := 0 to NoOfFile-1 do
             writeln(SaveFile,FileName[i]);
          CloseFile(SaveFile);
       end;
    end;
end;

//---------------------------------------------------------------
Procedure TForm2.BtnSaveBinClick(Sender: TObject);
var  FiName,RepName : string;
    ret,i,j : integer;
    RepFile : TextFile;
    LoadFile,SaveFile : File of byte;
    FiSize : integer;
    Address : longint;
    data : byte;
begin
    SaveDialog2.Filename := '';
    SaveDialog2.Execute;
    if SaveDialog2.Filename <> '' then
    begin
       ret := 6;
       FiName := SaveDialog2.Filename;
       if ExtractFileExt(FiName) ='' then
         Finame := FiName + '.Bin';
       if FileExists(FiName) then
          ret := MessageDlg('Do you want to replace the
               existing "'+ExtractFileName(FiName)+'"',
               mtConfirmation, [mbYes,mbNo], 0);
       if ret=6 then
       begin
          ProgressPanel.visible := true;
          Gauge1.MaxValue := NoOfFile;
          RepName := Finame;
          RepName[Length(RepName)-2] := 'R';
          RepName[Length(RepName)-1] := 'E';
          RepName[Length(RepName)-0] := 'P';
          AssignFile(RepFile,RepName);
          rewrite(RepFile);
          AssignFile(SaveFile,FiName);
          Rewrite(SaveFile);
          Address := 3*(NoOfFile+1);
          for i := 0 to NoOfFile-1 do
          begin
             AssignFile(LoadFile,FileName[i]);
             Reset(LoadFile);
             FiSize := FileSize(LoadFile);
             write(RepFile,inttostr(i+1)+' '+
                   FileName[i]+' '+inttohex(address,6));
             writeln(RepFile,' - ' + inttohex(
                              Address+FiSize-1,6));
             data := (Address and $0FF0000) shr 16;
             write(SaveFile,data);
             data := (Address and $000FF00) shr 8;
             write(SaveFile,data);
             data := (Address and $00000FF);
             write(SaveFile,data);
             address := address + FiSize;
             CloseFile(LoadFile);
          end;
          CloseFile(RepFile);
          data := (Address and $0FF0000) shr 16;
          write(SaveFile,data);
          data := (Address and $000FF00) shr 8;
          write(SaveFile,data);
          data := (Address and $00000FF);
          write(SaveFile,data);
          for i := 0 to NoOfFile-1 do
          begin
             Gauge1.Progress := i+1;
             AssignFile(LoadFile,FileName[i]);
             Reset(LoadFile);
             FiSize := FileSize(LoadFile);
             for j :=0 to FiSize-1 do
             begin
                read(LoadFile,data);
                write(SaveFile,data);
             end;
             CloseFile(LoadFile);
          end;
          CloseFile(SaveFile);
          ProgressPanel.visible := false;
       end;
    end;
end;

end.
```

ภาคผนวก ค.

โปรแกรมของส่วนชุดฝึกพิมพ์

```
CPU"8051.TBL"
   HOF"INT8"
   INCL "HEAD51.H"

   ORG     0000H
   LJMP    MAIN

   ORG     0003H
   LJMP    INT0ISR

   ORG     000BH
   LJMP    T0ISR

   ORG     0013H
   LJMP    INT1ISR

   ORG     001BH
   LJMP    T1ISR

SHIFT:        EQU   10H
RELEASE:      EQU   11H
O_KEY:        EQU   12H
CAP_LOCK:     EQU   13H
FUNC_KEY:     EQU   14H
NUM_LOCK:     EQU   15H
KEY:          EQU   16H
LED_STA:      EQU   17H
H_ADD:        EQU   18H
M_ADD:        EQU   19H
L_ADD:        EQU   1AH
CARRY:        EQU   1BH
X_DPTR:       EQU   1CH
X_DPH:        EQU   1DH
X_DPL:        EQU   1EH
T_E:          EQU   1FH
S_DATA:       EQU   20H
INDEX:        EQU   21H
RIGHT:        EQU   22H
PRESS:        EQU   23H
TEST_KEY:     EQU   24H
LOOP:         EQU   25H
LESSON:       EQU   26H
T_DPH:        EQU   27H
T_DPL:        EQU   28H
T_INDEX:      EQU   29H
T_LESSON:     EQU   2AH
RANDOM:       EQU   2BH
N_LOOP:       EQU   2CH
T_RIGHT:      EQU   2DH
KEY_HIT:      EQU   2EH
RESULT:       EQU   2FH
WORD:         EQU   30H
MIN:          EQU   31H
MIN100:       EQU   32H
MIN60:        EQU   33H

MAIN:
   LCALL   DELAY
   MOV     SP,#34H
   LCALL   INITIAL

   MOV     P1,#0FFH
   JNB     P1.5,SAY_THAI
   MOV     A,#232
               ; LOAD ENGLISH ADDRESS
   SJMP    SAY_ENGLISH
SAY_THAI:
   MOV     T_E,#02H
   MOV     A,#233
SAY_ENGLISH:
   LCALL   LOAD_ADDRESS
   MOV     P1,X_DPTR
   MOV     R1,#08H
   SETB    TR0
   JB      TR0,$

CHECK_T_E:
   MOV     P1,#0FFH
   JB      P1.7,NOT_TEST
   LCALL   DELAY
   MOV     P1,#0FFH
   JB      P1.7,NOT_TEST
   LCALL   PRACTICE
NOT_TEST:
   MOV     P1,#0FFH
   JNB     P1.5,THAI
               ; CHECK THAI OR ENGLISH
   LCALL   DELAY
   MOV     R3,T_E
   CJNE    R3,#02H,CHECK_T_E
   MOV     P1,#0FFH
   JNB     P1.5,CHECK_T_E
                  ; CHECK ENGLISH AGAIN
   ANL     IEC,#7FH
   MOV     A,#232
                  ; LOAD ENGLISH ADDRESS
   LCALL   LOAD_ADDRESS
   MOV     P1,X_DPTR
   MOV     T_E,#00H
   MOV     R1,#08H
   ORL     IEC,#80H
   SETB    TR0
   SJMP    CHECK_T_E
THAI:
   LCALL   DELAY
   MOV     R3,T_E
   CJNE    R3,#00H,CHECK_T_E
   MOV     P1,#0FFH
   JB      P1.5,CHECK_T_E
                  ; CHECK THAI AGAIN
   ANL     IEC,#7FH
   MOV     A,#233 ; LOAD THAI ADDRESS
   LCALL   LOAD_ADDRESS
   MOV     P1,X_DPTR
   MOV     T_E,#02H
   MOV     R1,#08H
   ORL     IEC,#80H
   SETB    TR0
   SJMP    CHECK_T_E

;************************************
; INITIAL
;************************************
INITIAL:
   MOV     SHIFT,#00H
   MOV     RELEASE,#00H
   MOV     O_KEY,#00H
   MOV     CAP_LOCK,#00H
   MOV     FUNC_KEY,#00H
   MOV     NUM_LOCK,#00H
   MOV     KEY,#0FFH
   MOV     LED_STA,#80H
   LCALL   LED_STATUS
   MOV     T_E,#00H
   MOV     PRESS,#00H
   MOV     WORD,#08H
```

```
    CLR     P3.0          ; Clock Output
    CLR     P3.1          ; Data Output

    MOV     TMOD,#00010010B
       ; Timer1 Mode 1 & Timer0 Mode 2
    MOV     TH1,#0D8H
             ; 10000 Pulse to Interupt
             ; = 1/100 sec
    MOV     TL1,#0F0H
    MOV     TH0,#-62
                ; 62 Pulse to Interupt
    MOV     TL0,#-62

    MOV     R4,#09H

    SETB    IT0     ; FALLING EDGE TRIG
    SETB    IT1
    MOV     IEC,#10001111B
            ; Interupt Timer1 & INT1 &
            ; Timer0 & INT0
    MOV     IPC,#00000100B
               ; INT1 HIGHEST PRIORITY
    RET
;************************************

;************************************
; Delay  0.5ms
;************************************
DELAY:
    MOV     R0,#250
    DJNZ    R0,$

    RET
;************************************

;************************************
; Delay  10ms
;************************************
DELAY10MS:
    MOV     R3,#20
DELAY10:
    MOV     R0,#250
    DJNZ    R0,$
    DJNZ    R3,DELAY10
    RET
;************************************

;************************************
; BEEP
;************************************
BEEP:
    CLR     TR0
    MOV     A,#253
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LCALL   DELAY
    LCALL   DELAY
    RET
;************************************

;************************************
; PRACTICE KEY
;************************************
PRACTICE:
    CLR     TR0
    MOV     A,#243
    LCALL   LOAD_ADDRESS
    SETB    TR0

    LCALL   SEL_LESSON
    JB      TR0,$
    MOV     R3,LESSON
    CJNE    R3,#0FFH,BEGIN_LESSON
                                  ; CHECK EXIT
    RET

BEGIN_LESSON:
    MOV     P1,#0FFH
    JB      P1.6,KEY_PRAC
    LCALL   WORD_PRACTICE
    SJMP    END_LESSON
KEY_PRAC:
    LCALL   KEY_PRACTICE
END_LESSON:
    MOV     R3,LESSON
    CJNE    R3,#0FFH,SHOW_PER
    CLR     TR1
    RET

SHOW_PER:
    MOV     A,#250
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LCALL   SHOW_PERCENT

SHOW_SPEED:
    LCALL   KEY_SPEED
    RET
    ;************************************

;************************************
;  WORD PRACTICE
;************************************
WORD_PRACTICE:
    MOV     DPTR,#2000H           ;
LOAD START WORD ADDRESS
    MOV     R3,T_E          ; CHECK THAI
OR ENG
    CJNE    R3,#02,WTEST_ENG
WTEST_THAI:
    MOV     DPTR,#3000H
WTEST_ENG:
    MOV     A,LESSON
    ORL     DPH,A
    MOV     A,#00H
    MOVC    A,@A+DPTR

    JNZ     WSAVE_DPTR
    MOV     LESSON,#0FFH    ; NO LESSON
    MOV     A,#252
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    RET

WSAVE_DPTR:
    MOV     KEY_HIT,A
    INC     DPTR
    MOV     T_DPH,DPH
```

```
    MOV     T_DPL,DPL
WSAY_SEL_LESSON:  ; SAY SELECT LESSON
    INC     LESSON
    MOV     A,#244
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    MOV     A,LESSON
    ADD     A,#246
    JNC     WLESS_THAN_10
    MOV     A,#237
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    MOV     A,LESSON
    CJNE    A,#12,WCHK_11
    MOV     A,#184
    SJMP    WSAY_NEXT
WCHK_11:
    CJNE    A,#11,WNOT_11
    MOV     A,#239
    SJMP    WSAY_NEXT
WNOT_11:
    MOV     A,#255
WSAY_NEXT:
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    SJMP    BEGIN_WORD_TEST

WLESS_THAN_10:
    MOV     DPTR,#NUM
    MOV     A,LESSON
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

BEGIN_WORD_TEST:
    MOV     A,#247
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

    MOV     LOOP,#00H
    MOV     N_LOOP,#00H
    MOV     RIGHT,#00H
    MOV     MIN,#00H
    MOV     MIN100,#00H
    MOV     MIN60,#00H
    SETB    TR1

LOAD_WORD:
    JB      TR0,$
    MOV     DPH,T_DPH
    MOV     DPL,T_DPL
    MOV     A,N_LOOP
    MOVC    A,@A+DPTR
    CJNE    A,#255,SAY_WORD
    CLR     TR1
    RET
SAY_WORD:
    PUSH    A
    MOV     WORD,#10H
    MOV     A,T_E
    RL      A
    RL      A
    XRL     WORD,A
    POP     A
    LCALL   LOAD_ADDRESS
    SETB    TR0
    INC     N_LOOP

LOAD_LETTER:
    MOV     T_RIGHT,RIGHT
    INC     LOOP
    MOV     PRESS,#0F0H
    MOV     DPH,T_DPH
    MOV     DPL,T_DPL
    MOV     A,N_LOOP
    MOVC    A,@A+DPTR
    MOV     TEST_KEY,A
    LCALL   LOAD_ASCII
    MOV     TEST_KEY,A
    CJNE    A,#255,W_WAIT1
    INC     N_LOOP
    DEC     LOOP
    LJMP    LOAD_WORD

W_WAIT1:
    JB      TR0,$
    MOV     P1,#0FFH
                ;CHECK FOR QUIT LESSON
    JNB     P1.7,W_WAIT2
    LCALL   DELAY
    MOV     P1,#0FFH
    JNB     P1.7,W_WAIT2
    MOV     PRESS,#00H
    MOV     A,#251
    LCALL   LOAD_ADDRESS
    SETB    TR0
    MOV     LESSON,#0FFH
    RET
W_WAIT2:
    MOV     R3,PRESS
    CJNE    R3,#0FFH,W_WAIT1
    MOV     A,T_LESSON
    LCALL   LOAD_ADDRESS
    SETB    TR0
    INC     N_LOOP
    MOV     A,RIGHT
    XRL     A,T_RIGHT
    JNZ     LOAD_LETTER
    LCALL   BEEP
    LJMP    LOAD_LETTER
;************************************

;************************************
; KEY PRACTICE
;************************************
KEY_PRACTICE:
    MOV     DPTR,#2E00H
                ; LOAD NUMBER OF KEY
    MOV     R3,T_E ; CHECK THAI OR ENG
    CJNE    R3,#02,TEST_ENG0
TEST_THAI0:
    MOV     DPTR,#3E00H
TEST_ENG0:
    MOV     A,LESSON
    MOVC    A,@A+DPTR
    MOV     INDEX,A
    MOV     T_INDEX,INDEX

    MOV     A,INDEX
```

```
    JNZ     LOAD_SRT_ADD
    MOV     LESSON,#0FFH
                    ; NO PRACTICE KEY
    MOV     A,#252
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    RET

LOAD_SRT_ADD:
    MOV     DPTR,#2C00H
              ; LOAD START KEY ADDRESS
    MOV     R3,T_E ; CHECK THAI OR ENG
    CJNE    R3,#02,TEST_ENG1
TEST_THAI1:
    MOV     DPTR,#3C00H
TEST_ENG1:
    MOV     A,LESSON
    SWAP    A
    ORL     DPL,A

SAVE_DPTR:
    MOV     T_DPH,DPH
    MOV     T_DPL,DPL

SAY_SEL_LESSON:
    INC     LESSON
    MOV     A,#245
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    MOV     A,LESSON
    ADD     A,#246
    JNC     LESS_THAN_10
    MOV     A,#237
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    MOV     A,LESSON
    CJNE    A,#12,CHK_11
    MOV     A,#184
    SJMP    SAY_NEXT
CHK_11:
    CJNE    A,#11,NOT_11
    MOV     A,#239
    SJMP    SAY_NEXT
NOT_11:
    MOV     A,#255
SAY_NEXT:
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    SJMP    SAY_TEST_KEY

LESS_THAN_10:
    MOV     DPTR,#NUM
    MOV     A,LESSON
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

SAY_TEST_KEY:
    MOV     A,#246
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    MOV     DPTR,#2E00H
    MOV     A,T_E
    JZ      SAY_T_KEY
    MOV     DPTR,#3E00H
SAY_T_KEY:
    DEC     LESSON
    MOV     A,LESSON
    MOVC    A,@A+DPTR
    RL      A
    DEC     A
    MOV     R3,A
SAY_T_NEXT:
    MOV     DPL,T_DPL
    MOV     DPH,T_DPH
    MOV     A,R3
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    DEC     R3
    DEC     R3
    CJNE    R3,#255,SAY_T_NEXT

SAY_BEGIN:
    LCALL   DELAY10MS
    MOV     A,#248
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

    MOV     RIGHT,#00H
    MOV     A,LESSON
    JNZ     ADV_KEY

    MOV     LOOP,#5
BASIC_KEY:
    MOV     INDEX,T_INDEX
OPEN_BASIC_KEY:
    DEC     INDEX
    MOV     DPH,T_DPH
    MOV     DPL,T_DPL
    MOV     PRESS,#0F0H
    MOV     T_RIGHT,RIGHT
    MOV     A,INDEX
    RL      A
    MOVC    A,@A+DPTR
    MOV     TEST_KEY,A
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

WAIT1:
    MOV     P1,#0FFH;
                CHECK FOR QUIT LESSON
    JNB     P1.7,N_WAIT1
    LCALL   DELAY
    MOV     P1,#0FFH
    JNB     P1.7,N_WAIT1
    MOV     PRESS,#00H
    MOV     A,#251
    LCALL   LOAD_ADDRESS
    SETB    TR0
    MOV     LESSON,#255
    RET
N_WAIT1:
    MOV     R3,PRESS
    CJNE    R3,#0FFH,WAIT1
    JB      P3.2,$
```

```
    MOV     A,T_RIGHT
    XRL     A,RIGHT
    JNZ     CHK_LOOP1
    LCALL   BEEP

CHK_LOOP1:
    MOV     R3,INDEX
    CJNE    R3,#00H,OPEN_BASIC_KEY
    DEC     LOOP
    MOV     R3,LOOP
    CJNE    R3,#00H,BASIC_KEY
             ; END OF LESSON BASIC KEY
    LJMP    PRACTICE2

ADV_KEY:
    MOV     INDEX,T_INDEX
    MOV     LOOP,#5
    MOV     N_LOOP,#4
OPEN_ADV_KEY:
    DEC     INDEX
PRAC_1KEY:
    MOV     T_RIGHT,RIGHT
    MOV     DPH,T_DPH
    MOV     DPL,T_DPL
    MOV     PRESS,#0F0H
    MOV     A,INDEX
    RL      A
    MOV     R3,N_LOOP
    CJNE    R3,#01H,NOT_ADV_KEY
    INC     A
NOT_ADV_KEY:
    MOVC    A,@A+DPTR
    MOV     TEST_KEY,A
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

WAIT2:
    MOV     P1,#0FFH
             ; CHECK FOR QUIT LESSON
    JNB     P1.7,N_WAIT2
    LCALL   DELAY
    MOV     P1,#0FFH
    JNB     P1.7,N_WAIT2
    MOV     PRESS,#00H
    MOV     A,#251
    LCALL   LOAD_ADDRESS
    SETB    TR0
    MOV     LESSON,#255
    RET
N_WAIT2:
    MOV     R3,PRESS
    CJNE    R3,#0FFH,WAIT2
    JB      P3.2,$

    MOV     A,T_RIGHT
    XRL     A,RIGHT
    JNZ     CHK_LOOP2
    LCALL   BEEP

CHK_LOOP2:
    DEC     N_LOOP
    MOV     R3,N_LOOP
    CJNE    R3,#00H,PRAC_1KEY

    MOV     N_LOOP,#4
    DEC     LOOP
    MOV     R3,LOOP
    CJNE    R3,#00H,PRAC_1KEY

    MOV     LOOP,#5
    MOV     R3,INDEX
    CJNE    R3,#00H,OPEN_ADV_KEY
           ; END OF LESSON ADVANCE KEY

PRACTICE2:     ; BEGIN TEST RANDOM KEY
    MOV     A,#249
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

    MOV     RIGHT,#00
    SETB    TR1
    MOV     DPTR,#2E10H
                     ; LOAD SUM OF KEY
    MOV     R3,T_E ; CHECK THAI OR ENG
    CJNE    R3,#02,SUM_KEY_ENG
    MOV     DPTR,#3E10H
SUM_KEY_ENG:
    MOV     A,LESSON
    MOVC    A,@A+DPTR
    MOV     INDEX,A

    MOV     LOOP,#100
    MOV     KEY_HIT,#100

    MOV     MIN,#00H
    MOV     MIN100,#00H
    MOV     MIN60,#00H

RANDOM_KEY:
    MOV     DPTR,#2D00H
             ; LOAD START KEY ADDRESS
    MOV     R3,T_E ; CHECK THAI OR ENG
    CJNE    R3,#02H,RAND_ENG
    MOV     DPTR,#3D00H
RAND_ENG:
    MOV     T_RIGHT,RIGHT
    MOV     A,RANDOM
    MOV     B,INDEX
    DIV     AB
    MOV     A,B
    MOVC    A,@A+DPTR
    MOV     TEST_KEY,A
    LCALL   LOAD_ADDRESS
    SETB    TR0
    MOV     PRESS,#0F0H
    JB      TR0,$
WAIT_3:
    MOV     P1,#0FFH
             ; CHECK FOR QUIT LESSON
    JNB     P1.7,N_WAIT3
    LCALL   DELAY
    MOV     P1,#0FFH
    JNB     P1.7,N_WAIT3
    MOV     PRESS,#00H
    MOV     A,#251
    LCALL   LOAD_ADDRESS
    SETB    TR0
    MOV     LESSON,#255
    RET
N_WAIT3:
    MOV     R3,PRESS
    CJNE    R3,#0FFH,WAIT_3
    JB      P3.2,$
```

```
    MOV     A,T_RIGHT
    XRL     A,RIGHT
    JNZ     CHK_LOOP3
    LCALL   BEEP

CHK_LOOP3:
    DEC     LOOP
    MOV     R3,LOOP
    CJNE    R3,#00H,RANDOM_KEY
            ; END OF LESSON RANDOM KEY

    CLR     TR1
    RET
;************************************


;************************************
;  SELECT LESSON
;************************************
SEL_LESSON:
    MOV     T_INDEX,T_E
    MOV     T_E,#00
RCV_KEY:
    MOV     PRESS,#0F0H
WAIT_KEY:
    MOV     P1,#0FFH
            ; CHECK FOR QUIT LESSON
    JNB     P1.7,N_WAIT_K
    LCALL   DELAY
    MOV     P1,#0FFH
    JNB     P1.7,N_WAIT_K
    MOV     PRESS,#00H
    MOV     LESSON,#0FFH
    CLR     TR0
    MOV     T_E,T_INDEX
    MOV     A,#251
    LCALL   LOAD_ADDRESS
    SETB    TR0
    RET
N_WAIT_K:
    MOV     R3,PRESS
    CJNE    R3,#0FFH,WAIT_KEY
    MOV     R3,T_LESSON
    CJNE    R3,#76,NOT_L0
    MOV     LESSON,#00
    LJMP    EXIT_SEL
NOT_L0:
    CJNE    R3,#77,NOT_L1
    MOV     LESSON,#01
    LJMP    EXIT_SEL
NOT_L1:
    CJNE    R3,#78,NOT_L2
    MOV     LESSON,#02
    LJMP    EXIT_SEL
NOT_L2:
    CJNE    R3,#79,NOT_L3
    MOV     LESSON,#03
    LJMP    EXIT_SEL
NOT_L3:
    CJNE    R3,#80,NOT_L4
    MOV     LESSON,#04
    LJMP    EXIT_SEL
NOT_L4:
    CJNE    R3,#81,NOT_L5
    MOV     LESSON,#05
    LJMP    EXIT_SEL
NOT_L5:
    CJNE    R3,#82,NOT_L6
    MOV     LESSON,#06
    LJMP    EXIT_SEL
NOT_L6:
    CJNE    R3,#83,NOT_L7
    MOV     LESSON,#07
    LJMP    EXIT_SEL
NOT_L7:
    CJNE    R3,#84,NOT_L8
    MOV     LESSON,#08
    LJMP    EXIT_SEL
NOT_L8:
    CJNE    R3,#85,NOT_L9
    MOV     LESSON,#09
    LJMP    EXIT_SEL
NOT_L9:
    CJNE    R3,#86,NOT_L10
    MOV     LESSON,#10
    LJMP    EXIT_SEL
NOT_L10:
    CJNE    R3,#87,NOT_L11
    MOV     LESSON,#11
    LJMP    EXIT_SEL
NOT_L11:
    CJNE    R3,#88,NOT_L12
    MOV     LESSON,#12
    LJMP    EXIT_SEL
NOT_L12:
    LJMP    RCV_KEY
EXIT_SEL:
    MOV     T_E,T_INDEX
    RET
;************************************

;************************************
;  SPEAK SPEED
;************************************
KEY_SPEED:
    MOV     A,#240          ; Say Speed
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
;                KEY_HIT *100
;    SPEED = ------------------------
;            (4,5) * (MIN*100 + MIN100)
    MOV     A,MIN
            ; MULTIPLY MINUTE BY 100
    MOV     B,#100
    MUL     AB
    MOV     R6,A
    MOV     R7,B
    ADD     A,MIN100    ; ADD BY MIN100
    JNC     NOT_INC
    INC     R7
NOT_INC:
    MOV     MIN,A
    MOV     MIN100,R7

    MOV     A,T_E ; MULTIPLY BY 4 OR 5
    RR      A
    ADD     A,#04H
    MOV     B,MIN100
    MUL     AB
    MOV     MIN100,A
```

```
    MOV     A,T_E
    RR      A
    ADD     A,#04H
    MOV     B,MIN
    MUL     AB
    MOV     MIN,A
    MOV     A,B
    ADD     A,MIN100
    MOV     MIN100,A

    MOV     A,KEY_HIT
            ; MULTIPLY KEY_HIT BY 100
    MOV     B,#100
    MUL     AB
    MOV     R6,A
    MOV     R7,B
START_DIV:
    CLR     C
          ; T_DPH+T_DPL = MIN100+MIN/2
    MOV     A,MIN100
    RRC     A
    MOV     T_DPH,A
    MOV     A,MIN
    RRC     A
    MOV     T_DPL,A
    CLR     C

    MOV     RESULT,#00H
CHK_HI:
    MOV     A,T_DPH
    SUBB    A,R7
    JC      INC_RESULT
CHK_LO:
    MOV     A,T_DPL
    SUBB    A,R6
    JC      INC_RESULT
    SJMP    DIV_COMPLETE

INC_RESULT:
    INC     RESULT
    MOV     A,T_DPH
    ADD     A,MIN100
    MOV     T_DPH,A
    MOV     A,T_DPL
    ADD     A,MIN
    JNC     NOT_INC_TDPH
    INC     T_DPH
NOT_INC_TDPH:
    MOV     T_DPL,A
    SJMP    CHK_HI

DIV_COMPLETE:
S_CHECK_ONE:
    MOV     A,RESULT
    ADD     A,#246
    JC      S_DIVIDE
    MOV     A,RESULT
    MOV     DPTR,#NUM
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LJMP    SPEED

S_DIVIDE:
    MOV     A,RESULT
    MOV     B,#10
    DIV     AB
    MOV     INDEX,B

    MOV     DPTR,#TEN         ; SAY TEN
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LCALL   DELAY10MS

    MOV     A,#237          ; SAY "TEN"
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LCALL   DELAY10MS

    MOV     DPTR,#ONE         ; SAY ONE
    MOV     A,INDEX
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

SPEED:
    MOV     A,#241
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

    RET
;************************************
;************************************
;  SPEAK CORRECT PERCENT
;************************************
SHOW_PERCENT:
    MOV     IEC,#10001010B

    MOV     RESULT,#00
    MOV     A,RIGHT
    MOV     B,#100
    MUL     AB
    MOV     R6,A
    MOV     R7,B
    MOV     A,R7
    JZ      COMPLT_DIV
LOOP_DIV:
    MOV     A,R6
    SUBB    A,KEY_HIT
    JNC     NOT_BORR
    DEC     R7
NOT_BORR:
    MOV     R6,A
    INC     RESULT
    CJNE    R7,#00,LOOP_DIV
COMPLT_DIV:
    MOV     A,R6
    SUBB    A,KEY_HIT
    JC      COMPLT_DIV2
    MOV     R6,A
    INC     RESULT
    SJMP    COMPLT_DIV
COMPLT_DIV2:
    MOV     A,KEY_HIT
    CLR     C
    RRC     A
    SUBB    A,R6
    JNC     SPEAK_PERCENT
```

```
    INC     RESULT

SPEAK_PERCENT:
    MOV     A,#242
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

    MOV     A,RESULT
    CJNE    A,#100,CHECK_ONE
    MOV     A,#236
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LJMP    PERCENT

CHECK_ONE:
    MOV     A,RESULT
    ADD     A,#246
    JC      DIVIDE
    MOV     A,RESULT
    MOV     DPTR,#NUM
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LJMP    PERCENT

DIVIDE:
    MOV     A,RESULT
    MOV     B,#10
    DIV     AB
    MOV     INDEX,B

    MOV     DPTR,#TEN       ; SAY TEN
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LCALL   DELAY10MS

    MOV     A,#237          ; SAY "TEN"
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$
    LCALL   DELAY10MS

    MOV     DPTR,#ONE       ; SAY ONE
    MOV     A,INDEX
    MOVC    A,@A+DPTR
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

PERCENT:
    MOV     A,#96           ; SAY PERCENT
    LCALL   LOAD_ADDRESS
    SETB    TR0
    JB      TR0,$

    MOV     PRESS,#00H
    MOV     IEC,#10001111B
    RET
;************************************

;************************************
; LOAD ASCII TABLE
;************************************
LOAD_ASCII:
    MOV     DPTR,#ASCII_TABLE
    MOV     A,TEST_KEY
    MOVC    A,@A+DPTR
    MOV     TEST_KEY,A
    RET
;************************************


;************************************
; Interupt Service Routine of INT1
; Rotate Key Data
;************************************
**************************
INT1ISR:
    CJNE    R4,#09H,NEXT
    MOV     C,P3.5
    JC      OUT_INT1
    DEC     R4
    SJMP    OUT_INT1
NEXT:
    CJNE    R4,#00H,NEXT2
    SJMP    OUT_INT1
NEXT2:
    MOV     A,KEY
    MOV     C,P3.5
    RRC     A
    MOV     KEY,A
    DEC     R4
OUT_INT1:
    RETI
;************************************

;************************************
; Interupt Service Routine of INT0
; Check Key
;************************************
INT0ISR:
    MOV     R4,#09H
    MOV     A,KEY
    CJNE    A,#0F0H,CHK_E0
                    ; CHECK FOR RELEASE KEY
    MOV     O_KEY,#0F0H
    MOV     FUNC_KEY,#00H
    MOV     RELEASE,#0FFH
    LJMP    OUT_INT0

CHK_E0: CJNE    A,#0E0H,CHK
    MOV     FUNC_KEY,#0FFH
    LJMP    OUT_INT0

CHK:    CLR     C
    ADD     A,#7CH
    JC      EXIT_INT
                    ; CHECK DATA <= 83H
    MOV     A,RELEASE
    CJNE    A,#0FFH,CHK_OKEY

REL:
    MOV     RELEASE,#00H
    MOV     A,KEY

CHK_SHFL:
    CJNE    A,#12H,CHK_SHFR
                    ; CHECK SHIFT LEFT
    MOV     SHIFT,#00H
```

```
   LJMP    OUT_INT0

CHK_SHFR:
   CJNE    A,#59H,CHK_CAP
                  ; CHECK SHIFT RIGHT
   MOV     SHIFT,#00H
   LJMP    OUT_INT0

CHK_CAP:
   CJNE    A,#58H,NOT_CAP
                  ; CHECK CAPS LOCK
   LJMP    OUT_INT0

NOT_CAP:
   ORL     PRESS,#0FH
EXIT_INT0:
   LJMP    OUT_INT0

CHK_OKEY:
   MOV     A,KEY
   CJNE    A,O_KEY,CHK_BREAK
   LJMP    OUT_INT0

CHK_BREAK:
   CJNE    A,#77H,CHK_FKEY
   MOV     A,O_KEY
   CJNE    A,#14H,NUM_L
   MOV     O_KEY,#77H
   MOV     A,#90          ; KEY = BREAK
   LJMP    NOT_FF

NUM_L:
   ORL     NUM_LOCK,#0FH
   XRL     NUM_LOCK,#0F0H
   XRL     LED_STA,#81H
   LCALL   LED_STATUS
   MOV     O_KEY,#77H
   SJMP    NOT_S_LOCK

CHK_FKEY:
   MOV     O_KEY,KEY
   MOV     A,FUNC_KEY
   CJNE    A,#0FFH,CHK_CAP2
   ANL     KEY,#3FH
   ORL     KEY,#80H
   SJMP    NOT_S_LOCK

CHK_CAP2:
   MOV     A,KEY
   CJNE    A,#58H,CHK_SHFL2
   XRL     CAP_LOCK,#0FFH
   XRL     LED_STA,#82H
   LCALL   LED_STATUS
   SJMP    NOT_S_LOCK

CHK_SHFL2:
   CJNE    A,#12H,CHK_SHFR2
                  ; CHECK SHIFT LEFT
   MOV     SHIFT,#0FFH
   SJMP    NOT_S_LOCK

CHK_SHFR2:
   CJNE    A,#59H,CHK_S_LOCK
                  ; CHECK SHIFT RIGHT
   MOV     SHIFT,#0FFH
   SJMP    NOT_S_LOCK

CHK_S_LOCK:
   CJNE    A,#7EH,NOT_S_LOCK
   XRL     LED_STA,#80H
   LCALL   LED_STATUS

NOT_S_LOCK:
   MOV     A,KEY
   CLR     C
   ADD     A,#97H  ; CHECK KEY >= 69H
   JNC     NOT_NUM_KEY
   MOV     A,KEY
   CLR     C
   ADD     A,#82H  ; CHECK KEY <= 7DH
   JC      NOT_NUM_KEY

NUM_KEY:
   MOV     A,KEY
   ADD     A,#60H
   MOV     KEY,A
   MOV     A,NUM_LOCK
   ANL     A,#0F0H
   CJNE    A,#0F0H,OPEN_TABLE2
   SJMP    OPEN_TABLE1

NOT_NUM_KEY:
   MOV     A,SHIFT
   XRL     A,CAP_LOCK
   CJNE    A,#0FFH,OPEN_TABLE1

OPEN_TABLE2:
   MOV     CARRY,#01H
   SJMP    ADD

OPEN_TABLE1:
   MOV     CARRY,#00H
ADD:
   MOV     DPTR,#TABLE
   MOV     A,KEY
   MOV     B,#04H
   MUL     AB
   ADD     A,DPL
   JC      ADD1
   SJMP    ADD2
ADD1:
   INC     DPH
ADD2:
   MOV     DPL,A
   MOV     A,B
   ADD     A,DPH
   MOV     DPH,A

LOAD_INDEX:
   MOV     A,CARRY
   ADD     A,T_E
   MOVC    A,@A+DPTR      ; LOAD INDEX

   CJNE    A,#0FFH,NOT_FF
   CLR     TR0
   LJMP    OUT_INT0

NOT_FF:
   MOV     KEY,A
   MOV     A,PRESS
   ANL     A,#0F0H
   JNZ     KEY_PRESS

   MOV     A,KEY
   LCALL   LOAD_ADDRESS
   MOV     R1,#08H
```

```
    MOV     P1,X_DPTR
    SETB    TR0
    SJMP    OUT_INT0

KEY_PRESS:
    MOV     A,KEY
    CJNE    A,#53,SHIFT_R_T
             ; CHECK SHIFT RIGHT (ENG)
    SJMP    OUT_INT0
SHIFT_R_T:
    CJNE    A,#157,SHIFT_L
            ; CHECK SHIFT RIGHT (THAI)
    SJMP    OUT_INT0
SHIFT_L:
    CJNE    A,#41,SHIFT_L_T
              ; CHECK SHIFT LEFT (ENG)
    SJMP    OUT_INT0
SHIFT_L_T:
    CJNE    A,#146,CAP
             ; CHECK SHIFT LEFT (THAI)
    SJMP    OUT_INT0
CAP:
    CJNE    A,#27,CHK_KEY
                    ; CHECK CAPS LOCK
    SJMP    OUT_INT0
CHK_KEY:
    CJNE    A,TEST_KEY,WRONG
    INC     RIGHT
WRONG:
    MOV     TEST_KEY,#255
    MOV     T_LESSON,A
OUT_INT0:
    MOV     KEY,#0FFH
    RETI
;************************************

;************************************
; Load Adress from Index   Index = A
;************************************
LOAD_ADDRESS:
    MOV     B,#03H
    MUL     AB

    MOV     P1,WORD
    MOV     DPL,A
    MOV     DPH,B
                    ; START HIGH BYTE
    MOVX    A,@DPTR
    MOV     R5,A

    INC     DPTR        ; START MID BYTE
    MOVX    A,@DPTR
    MOV     R6,A

    INC     DPTR        ; START LOW BYTE
    MOVX    A,@DPTR
    MOV     R7,A

    INC     DPTR         ; END HIGH BYTE
    MOVX    A,@DPTR
    MOV     H_ADD,A

    INC     DPTR          ; END MID BYTE
    MOVX    A,@DPTR
    MOV     M_ADD,A

    INC     DPTR          ; END LOW BYTE
    MOVX    A,@DPTR
    MOV     L_ADD,A

    MOV     X_DPTR,R5
    MOV     X_DPH,R6
    MOV     X_DPL,R7

    MOV     A,WORD
    ORL     X_DPTR,A
    ORL     H_ADD,A
    MOV     WORD,#08H
    RET
;************************************

;************************************
; Interupt Service Routine of Timer0
; Send Sound Data
;************************************
T0ISR:
    PUSH    DPH
    PUSH    DPL
    MOV     DPL,X_DPL
    MOV     DPH,X_DPH
    DJNZ    R1,OUTDATA
    MOV     A,H_ADD
    CJNE    A,X_DPTR,NEXTDATA
    MOV     A,M_ADD
    CJNE    A,DPH,NEXTDATA
    MOV     A,L_ADD
    CJNE    A,DPL,NEXTDATA
    CLR     TR0             ; STOP TIMER0
NEXTDATA:
    MOV     A,DPL
    CJNE    A,#0FFH,INC_DPTR
    MOV     A,DPH
    CJNE    A,#0FFH,INC_DPTR
    INC     X_DPTR
INC_DPTR:
    INC     DPTR
    MOV     X_DPH,DPH
    MOV     X_DPL,DPL
NEXTDATA2:
    MOV     P1,X_DPTR
    MOVX    A,@DPTR
    MOV     S_DATA,A
    MOV     R1,#08H
OUTDATA:
    MOV     A,S_DATA
    RLC     A
    MOV     S_DATA,A
    MOV     P3.1,C       ; Out Data
    SETB    P3.0         ; Send 1 Clock
    CLR     P3.0
    CLR     C
    POP     DPL
    POP     DPH
    RETI
;************************************

;************************************
; Interupt Service Routine of Timer1
; Random Number
;************************************
T1ISR:
    PUSH    A
```

```
    INC     MIN60
    MOV     A,MIN60
    CJNE    A,#60,OUT_T1

INC_SEC:
    MOV     MIN60,#00H
    INC     MIN100
    MOV     A,MIN100
    CJNE    A,#100,OUT_T1

INC_MIN:
    clr     tr0
    mov     a,#00h
    lcall   load_address
    setb    tr0
    MOV     MIN100,#00H
    INC     MIN

OUT_T1:
    POP     A
    INC     RANDOM
    MOV     TH1,#0D8H
    MOV     TL1,#0F0H
    RETI
;************************************


;************************************
; SEND DATA TO DISPLAY LED STATUS
;************************************
LED_STATUS:
    MOV     IEC,#00000000B
CP33:
    MOV     C,P3.3
    JNC     CP33
CP35:
    MOV     C,P3.5
    JNC     CP35

    CLR     P3.3    ; CLK
    CLR     P3.5    ; DATA
    MOV     R0,#10
    DJNZ    R0,$
    SETB    P3.3

    MOV     A,#0F6H
    LCALL   SEND

WA0:
    MOV     C,P3.2
    JC      WA0
WA1:
    MOV     C,P3.2
    JNC     WA1
WA2:
    MOV     C,P3.2
    JC      WA2

    CLR     P3.3
    MOV     R0,#100
    DJNZ    R0,$
    CLR     P3.5
    SETB    P3.3

    MOV     A,LED_STA
    LCALL   SEND

WA10:
    MOV     C,P3.2
    JC      WA10
WA11:
    MOV     C,P3.2
    JNC     WA11
WA12:
    MOV     C,P3.2
    JC      WA12

    MOV     IEC,#10001111B
    RET
;************************************


;************************************
; SEND DATA TO KEYBOARD
;************************************
SEND:
    MOV     R5,#08H
    MOV     R2,#01H
    MOV     B,A

PARITY:
    XRL     A,R2    ; CALCULATE PARITY
    MOV     R2,A
    RRC     A
    DJNZ    R5,PARITY

W1:
    MOV     C,P3.3
    JC      W1
    MOV     A,R2
    RRC     A
    MOV     P3.5,C      ; SEND PARITY
W2:
    MOV     C,P3.3
    JNC     W2

    MOV     A,B
    MOV     R5,#08H
LOOP1:
W3:
    MOV     C,P3.3
    JC      W3
    RRC     A
    MOV     P3.5,C      ; SEND DATA
W4:
    MOV     C,P3.3
    JNC     W4
    DJNZ    R5,LOOP1

W5:
    MOV     C,P3.3
    JC      W5
    SETB    C
    MOV     P3.5,C  ; SEND STOP BIT 1
W6:
    MOV     C,P3.3
    JNC     W6

W7:     MOV     C,P3.3
    JC      W7
    CLR     P3.5    ; SEND STOP BIT 0
W8:
    MOV     C,P3.3
    JNC     W8
    SETB    P3.5
```

```
   RET                                  ;       |    /---------- ENGLISH(CAP)
;************************************   ;       |    |    /-------- THAI
                                        ;       |    |    |    /--- THAI(CAP)
ASCII_TABLE:                            ;       |    |    |    |
   DFB                                          DFB 255 ,255 ,255 ,255  ;00
   000,000,000,000,000,000,000,000,             DFB  84 , 84 ,179 ,179  ;01      F9
   000,000,000,000,000,000,000,000 ;0           DFB 255 ,255 ,255 ,255  ;02
   DFB                                          DFB  80 , 80 ,175 ,175  ;03      F5
   000,000,000,000,000,000,000,000,             DFB  78 , 78 ,173 ,173  ;04      F3
   000,000,000,000,000,000,000,000 ;1           DFB  76 , 76 ,171 ,171  ;05      F1
   DFB                                          DFB  77 , 77 ,172 ,172  ;06      F2
   000,000,000,000,000,000,000,000,             DFB  87 , 87 ,182 ,182  ;07     F12
   000,000,000,000,000,000,000,000 ;2           DFB 255 ,255 ,255 ,255  ;08
   DFB                                          DFB  85 , 85 ,180 ,180  ;09     F10
   000,000,000,000,000,000,000,000,             DFB  83 , 83 ,178 ,178  ;0A      F8
   000,000,000,000,000,000,000,000 ;3           DFB  81 , 81 ,176 ,176  ;0B      F6
   DFB                                          DFB  79 , 79 ,174 ,174  ;0C      F4
   000,028,047,045,030,016,031,032,             DFB  13 , 13 , 13 , 13  ;0D     TAB
   033,021,034,035,036,049,048,022 ;4           DFB 255 , 91 ,255 ,255  ;0E      ~
   DFB                                          DFB 255 ,255 ,255 ,255  ;0F
   023,014,017,029,018,020,046,015,
   044,019,043,000,000,000,000,000 ;5           DFB 255 ,255 ,255 ,255  ;10
   DFB                                          DFB  55 , 55 ,159 ,159  ;11   L ALT
   000,028,047,045,030,016,031,032,             DFB  41 , 41 ,146 ,146  ;12  L SHFT
   033,021,034,035,036,049,048,022 ;6           DFB 255 ,255 ,255 ,255  ;13
   DFB                                          DFB  54 , 54 ,158 ,158  ;14  L CTRL
   023,014,017,029,018,020,046,015,             DFB  14 , 14 ,123 ,194  ;15      Q
   044,019,043,000,000,000,000,000 ;7           DFB  00 , 92 ,142 ,170  ;16      1
   DFB                                          DFB 255 ,255 ,255 ,255  ;17
   000,000,000,000,000,000,000,000,             DFB 255 ,255 ,255 ,255  ;18
   000,000,000,000,000,000,000,000 ;8           DFB 255 ,255 ,255 ,255  ;19
   DFB                                          DFB  43 , 43 ,147 ,217  ;1A      Z
   000,000,000,000,000,000,000,000,             DFB  29 , 29 ,136 ,207  ;1B      S
   000,000,000,000,000,000,000,000 ;9           DFB  28 , 28 ,135 ,206  ;1C      A
   DFB                                          DFB  15 , 15 ,124 ,195  ;1D      W
   000,137,121,000,118,000,207,145,             DFB  01 , 93 ,112 ,183  ;1E      2
   120,219,122,215,210,202,196,208 ;A           DFB 255 ,255 ,255 ,255  ;1F
   DFB
   203,197,223,200,138,119,115,153,             DFB 255 ,255 ,255 ,255  ;20
   198,131,133,148,147,156,126,135 ;B           DFB  45 , 45 ,149 ,219  ;21      C
   DFB                                          DFB  44 , 44 ,148 ,218  ;22      X
   114,154,132,130,206,134,225,144,             DFB  30 , 30 ,137 ,208  ;23      D
   214,213,143,136,224,150,220,000 ;C           DFB  16 , 16 ,125 ,196  ;24      E
   DFB                                          DFB  03 , 95 ,114 ,185  ;25      4
   127,128,142,125,151,129,117,152,             DFB  02 , 94 ,113 ,184  ;26      3
   116,187,000,000,000,000,000,000 ;D           DFB 255 ,255 ,255 ,255  ;27
   DFB                                          DFB 255 ,255 ,255 ,255  ;28
   139,149,209,155,124,000,123,211,             DFB  56 , 56 , 56 , 56  ;29   SPACE
   141,140,199,212,221,000,000,000 ;E           DFB  46 , 46 ,150 ,220  ;2A      V
   DFB                                          DFB  31 , 31 ,138 ,209  ;2B      F
   000,000,000,000,000,000,000,000,             DFB  18 , 18 ,127 ,198  ;2C      T
   000,000,000,000,116,187,000,255 ;F           DFB  17 , 17 ,126 ,197  ;2D      R
                                                DFB  04 , 96 ,115 ,186  ;2E      5
TEN:                                            DFB 255 ,255 ,255 ,255  ;2F
   DFB
   255,255,238,185,186,189,190,191,             DFB 255 ,255 ,255 ,255  ;30
   192,193     ; NUM 0-9                        DFB  48 , 48 ,152 ,221  ;31      N
ONE:                                            DFB  47 , 47 ,151 ,255  ;32      B
   DFB                                          DFB  33 , 33 ,140 ,211  ;33      H
   255,239,184,185,186,189,190,191,             DFB  32 , 32 ,139 ,210  ;34      G
   192,193     ; NUM 0-9                        DFB  19 , 19 ,128 ,255  ;35      Y
NUM:                                            DFB  05 , 97 ,116 ,187  ;36      6
   DFB                                          DFB 255 ,255 ,255 ,255  ;37
   194,183,184,185,186,189,190,191,             DFB 255 ,255 ,255 ,255  ;38
   192,193     ; NUM 0-9                        DFB 255 ,255 ,255 ,255  ;39
                                                DFB  49 , 49 ,153 ,222  ;3A      M
TABLE:                                          DFB  34 , 34 ,141 ,212  ;3B      J
;       /---------------- ENGLISH               DFB  20 , 20 ,129 ,199  ;3C      U
```

```
DFB  06 , 98 ,117 ,188  ;3D       7
DFB  07 , 99 ,118 ,189  ;3E       8
DFB 255 ,255 ,255 ,255  ;3F

DFB 255 ,255 ,255 ,255  ;40
DFB  50 ,109 ,154 ,223  ;41  COMMA
DFB  35 , 35 ,142 ,213  ;42       K
DFB  21 , 21 ,130 ,200  ;43       I
DFB  22 , 22 ,131 ,201  ;44       O
DFB  09 ,101 ,120 ,191  ;45       0
DFB  08 ,100 ,119 ,190  ;46       9
DFB 255 ,255 ,255 ,255  ;47
DFB 255 ,255 ,255 ,255  ;48
DFB  51 ,110 ,155 ,224  ;49  . DOT
DFB  52 ,111 ,156 ,225  ;4A  SLASH
DFB  36 , 36 ,143 ,214  ;4B       L
DFB  37 ,106 ,144 ,215  ;4C SCOLON
DFB  23 , 23 ,132 ,202  ;4D       P
DFB  10 ,102 ,121 ,192  ;4E       -
DFB 255 ,255 ,255 ,255  ;4F

DFB 255 ,255 ,255 ,255  ;50
DFB 255 ,255 ,255 ,255  ;51
DFB  38 ,107 ,145 ,168  ;52
                        ; APOSTROPHE
DFB 255 ,255 ,255 ,255  ;53
DFB  24 ,104 ,133 ,203  ;54       [
DFB  11 ,103 ,122 ,193  ;55       =
DFB 255 ,255 ,255 ,255  ;56
DFB 255 ,255 ,255 ,255  ;57
DFB  27 , 27 , 27 , 27  ;58 C LOCK
DFB  53 , 53 ,157 ,157  ;59 R SHFT
DFB  40 , 40 , 40 , 40  ;5A  ENTER
DFB  25 ,105 ,134 ,204  ;5B       ]
DFB 255 ,255 ,255 ,255  ;5C
DFB  39 ,108 ,234 ,235  ;5D B SLSH
DFB 255 ,255 ,255 ,255  ;5E
DFB 255 ,255 ,255 ,255  ;5F

DFB 255 ,255 ,255 ,255  ;60
DFB  42 , 42 , 42 , 42  ;61  MACRO
DFB 255 ,255 ,255 ,255  ;62
DFB 255 ,255 ,255 ,255  ;63
DFB 255 ,255 ,255 ,255  ;64
DFB 255 ,255 ,255 ,255  ;65
DFB  12 , 12 , 12 , 12  ;66  B SPC
DFB 255 ,255 ,255 ,255  ;67
DFB 255 ,255 ,255 ,255  ;68
DFB 255 ,255 ,255 ,255  ;69
DFB 255 ,255 ,255 ,255  ;6A
DFB 255 ,255 ,255 ,255  ;6B
DFB 255 ,255 ,255 ,255  ;6C
DFB 255 ,255 ,255 ,255  ;6D
DFB 255 ,255 ,255 ,255  ;6E
DFB 255 ,255 ,255 ,255  ;6F

DFB 255 ,255 ,255 ,255  ;70
DFB 255 ,255 ,255 ,255  ;71
DFB 255 ,255 ,255 ,255  ;72
DFB 255 ,255 ,255 ,255  ;73
DFB 255 ,255 ,255 ,255  ;74
DFB 255 ,255 ,255 ,255  ;75
DFB 255 ,255 ,255 ,255  ;76
DFB 255 ,255 ,255 ,255  ;77
DFB 255 ,255 ,255 ,255  ;78
DFB 255 ,255 ,255 ,255  ;79
DFB 255 ,255 ,255 ,255  ;7A
DFB 255 ,255 ,255 ,255  ;7B
DFB 255 ,255 ,255 ,255  ;7C
DFB 255 ,255 ,255 ,255  ;7D
DFB  89 , 89 , 89 , 89  ;7E S LOCK
DFB 255 ,255 ,255 ,255  ;7F

DFB 255 ,255 ,255 ,255  ;80
DFB 255 ,255 ,255 ,255  ;81
DFB 255 ,255 ,255 ,255  ;82
DFB  82 , 82 ,177 ,177  ;83      F7
DFB 255 ,255 ,255 ,255  ;84
DFB 255 ,255 ,255 ,255  ;85
DFB 255 ,255 ,255 ,255  ;86
DFB 255 ,255 ,255 ,255  ;87
DFB 255 ,255 ,255 ,255  ;88
DFB 255 ,255 ,255 ,255  ;89
DFB  70 , 70 ,166 ,166  ;8A  NUM /
DFB 255 ,255 ,255 ,255  ;8B
DFB 255 ,255 ,255 ,255  ;8C
DFB 255 ,255 ,255 ,255  ;8D
DFB 255 ,255 ,255 ,255  ;8E
DFB 255 ,255 ,255 ,255  ;8F

DFB 255 ,255 ,255 ,255  ;90
DFB  57 , 57 ,160 ,160  ;91  R ALT
DFB 255 ,255 ,255 ,255  ;92
DFB 255 ,255 ,255 ,255  ;93
DFB  58 , 58 ,161 ,161  ;94 R CTRL
DFB 255 ,255 ,255 ,255  ;95
DFB 255 ,255 ,255 ,255  ;96
DFB 255 ,255 ,255 ,255  ;97
DFB 255 ,255 ,255 ,255  ;98
DFB 255 ,255 ,255 ,255  ;99
DFB  40 , 40 , 40 , 40  ;9A  N ENT
DFB 255 ,255 ,255 ,255  ;9B
DFB 255 ,255 ,255 ,255  ;9C
DFB 255 ,255 ,255 ,255  ;9D
DFB 255 ,255 ,255 ,255  ;9E
DFB 230 ,230 ,226 ,226  ;9F
                        START LEFT

DFB 255 ,255 ,255 ,255  ;A0
DFB 255 ,255 ,255 ,255  ;A1
DFB 255 ,255 ,255 ,255  ;A2
DFB 255 ,255 ,255 ,255  ;A3
DFB 255 ,255 ,255 ,255  ;A4
DFB 255 ,255 ,255 ,255  ;A5
DFB 255 ,255 ,255 ,255  ;A6
DFB 231 ,231 ,227 ,227  ;A7
                        START RIGHT
DFB 255 ,255 ,255 ,255  ;A8
DFB  63 , 63 , 63 , 63  ;A9    END
DFB 255 ,255 ,255 ,255  ;AA
DFB  61 , 61 ,162 ,162  ;AB   LEFT
DFB  62 , 62 , 62 , 62  ;AC   HOME
DFB 255 ,255 ,255 ,255  ;AD
DFB 255 ,255 ,255 ,255  ;AE
DFB 228 ,228 ,228 ,228  ;AF   MENU

DFB  59 , 59 , 59 , 59  ;B0 INSERT
DFB  60 , 60 , 60 , 60  ;B1 DELETE
DFB  65 , 65 ,164 ,164  ;B2   DOWN
DFB 255 ,255 ,255 ,255  ;B3
DFB  68 , 68 ,165 ,165  ;B4  RIGHT
DFB  64 , 64 ,163 ,163  ;B5     UP
DFB 255 ,255 ,255 ,255  ;B6
DFB 255 ,255 ,255 ,255  ;B7
DFB 255 ,255 ,255 ,255  ;B8
DFB 255 ,255 ,255 ,255  ;B9
```

```
    DFB  67 , 67 , 67 , 67  ;BA  PG DW
    DFB 255 ,255 ,255 ,255  ;BB
    DFB  88 , 88 , 88 , 88  ;BC
                          PRINT SCREEN
    DFB  66 , 66 , 66 , 66  ;BD  PG UP
    DFB 255 ,255 ,255 ,255  ;BE
    DFB 255 ,255 ,255 ,255  ;BF

    DFB 255 ,255 ,255 ,255  ;C0
    DFB 255 ,255 ,255 ,255  ;C1
    DFB 255 ,255 ,255 ,255  ;C2
    DFB 255 ,255 ,255 ,255  ;C3
    DFB 255 ,255 ,255 ,255  ;C4
    DFB 255 ,255 ,255 ,255  ;C5
    DFB 255 ,255 ,255 ,255  ;C6
    DFB 255 ,255 ,255 ,255  ;C7
    DFB 255 ,255 ,255 ,255  ;C8
    DFB  00 , 63 ,183 , 63  ;C9   NUM1
    DFB 255 ,255 ,255 ,255  ;CA
    DFB  03 , 61 ,186 ,162  ;CB   NUM4
    DFB  06 , 62 ,191 , 62  ;CC   NUM7
    DFB 255 ,255 ,255 ,255  ;CD
    DFB 255 ,255 ,255 ,255  ;CE
    DFB 255 ,255 ,255 ,255  ;CF

    DFB  09 , 59 ,194 , 59  ;D0   NUM0
    DFB  72 , 60 ,168 , 60  ;D1   NUM.
    DFB  01 , 65 ,184 ,164  ;D2   NUM2
    DFB  04 ,255 ,189 ,255  ;D3   NUM5
    DFB  05 , 68 ,190 ,165  ;D4   NUM6
    DFB  07 , 64 ,192 ,163  ;D5   NUM8
    DFB  75 , 75 , 75 , 75  ;D6    ESC
    DFB  69 , 69 , 69 , 69  ;D7 N LOCK
    DFB  86 , 86 ,181 ,181  ;D8    F11
    DFB  74 , 74 ,170 ,170  ;D9   NUM+
    DFB  02 , 67 ,185 , 67  ;DA   NUM3
    DFB  73 , 73 ,169 ,169  ;DB   NUM-
    DFB  71 , 71 ,167 ,167  ;DC   NUM*
    DFB  08 , 66 ,193 , 66  ;DD   NUM9

    ORG  2000H
EWORD_TEST0:  ; A S D F J K L ;
    DFB  30
    DFB  000, "FALL"      ,255
    DFB  001, "ASK"       ,255
    DFB  002, "SALAD"     ,255
    DFB  003, "SAD"       ,255
    DFB  004, "JASS"      ,255
    DFB  005, "ALL"       ,255
    DFB  006, "AS"        ,255
    DFB  007, "ADD"       ,255
    DFB  008, "DAD"       ,255
    DFB  255      ; END OF EWORD_TEST0

    ORG  2100H
EWORD_TEST1:  ; E G U .
    DFB  80
    DFB  009, "DEAL"      ,255
    DFB  010, "EAGLE"     ,255
    DFB  011, "LAKE"      ,255
    DFB  012, "FULL"      ,255
    DFB  013, "EGG"       ,255
    DFB  014, "GLAD"      ,255
    DFB  015, "FUSE"      ,255
    DFB  016, "GLASS"     ,255
    DFB  017, "AGE"       ,255
    DFB  018, "FUEL"      ,255
    DFB  019, "US"        ,255
    DFB  020, "ELSE"      ,255
    DFB  021, "LEAD"      ,255
    DFB  022, "GUESS"     ,255
    DFB  023, "USE"       ,255
    DFB  024, "DESK"      ,255
    DFB  025, "JUG"       ,255
    DFB  026, "FAKE"      ,255
    DFB  027, "DEAD"      ,255
    DFB  028, "SEEK"      ,255
    DFB  029, "GAS"       ,255
    DFB  255      ; END OF EWORD_TEST1

    ORG  2200H
EWORD_TEST2:  ; R H O ,
    DFB  114
    DFB  030, "HAS"       ,255
    DFB  031, "LOOK"      ,255
    DFB  032, "DOG"       ,255
    DFB  033, "READ"      ,255
    DFB  034, "FRESH"     ,255
    DFB  035, "HER"       ,255
    DFB  036, "FOR"       ,255
    DFB  037, "GOLD"      ,255
    DFB  038, "SO"        ,255
    DFB  039, "JOKE"      ,255
    DFB  040, "ROAD"      ,255
    DFB  041, "GOD"       ,255
    DFB  042, "HOOK"      ,255
    DFB  043, "RULE"      ,255
    DFB  044, "RED"       ,255
    DFB  045, "GOOD"      ,255
    DFB  046, "OR"        ,255
    DFB  047, "FOLK"      ,255
    DFB  048, "HOUSE"     ,255
    DFB  049, "HALF"      ,255
    DFB  050, "ORDER"     ,255
    DFB  051, "GOLF"      ,255
    DFB  052, "HELLO"     ,255
    DFB  053, "FAR"       ,255
    DFB  054, "AGO"       ,255
    DFB  055, "LARGE"     ,255
    DFB  056, "EAR"       ,255
    DFB  057, "JAR"       ,255
    DFB  058, "ERROR"     ,255
    DFB  059, "LEADER"    ,255
    DFB  255      ; END OF EWORD_TEST2

    ORG  2300H
EWORD_TEST3:  ; T C I M :
    DFB  140
    DFB  060, "TEST"      ,255
    DFB  061, "CHECK"     ,255
    DFB  062, "STILL"     ,255
    DFB  063, "MAIL"      ,255
    DFB  064, "LETTER"    ,255
    DFB  065, "SIT"       ,255
    DFB  066, "FAST"      ,255
    DFB  067, "TIME"      ,255
    DFB  068, "CREAM"     ,255
    DFB  069, "HEAT"      ,255
    DFB  070, "TICKET"    ,255
    DFB  071, "CAKE"      ,255
    DFB  072, "COURSE"    ,255
    DFB  073, "MOTHER"    ,255
    DFB  074, "FACE"      ,255
    DFB  075, "JACKET"    ,255
    DFB  076, "COME"      ,255
    DFB  077, "FORGET"    ,255
```

```
    DFB  078, "KILL"      ,255
    DFB  079, "HAIR"      ,255
    DFB  080, "FAT"       ,255
    DFB  081, "HOME"      ,255
    DFB  082, "GAME"      ,255
    DFB  083, "MARKET"    ,255
    DFB  084, "ROOM"      ,255
    DFB  085, "DIRECT"    ,255
    DFB  086, "DOCTOR"    ,255
    DFB  087, "GIRL"      ,255
    DFB  088, "THAI"      ,255
    DFB  089, "FIRST"     ,255
    DFB  255       ; END OF EWORD_TEST3

    ORG  2400H
EWORD_TEST4:  ; W V Y P
    DFB  140
    DFB  090, "TODAY"     ,255
    DFB  091, "VIEW"      ,255
    DFB  092, "APPLE"     ,255
    DFB  093, "WAY"       ,255
    DFB  094, "PAPER"     ,255
    DFB  095, "JULY"      ,255
    DFB  096, "PEOPLE"    ,255
    DFB  097, "VICTORY"   ,255
    DFB  098, "VALUE"     ,255
    DFB  099, "WORD"      ,255
    DFB  100, "PLACE"     ,255
    DFB  101, "VISIT"     ,255
    DFB  102, "YOU"       ,255
    DFB  103, "WALK"      ,255
    DFB  104, "WELCOME"   ,255
    DFB  105, "GIVE"      ,255
    DFB  106, "FLY"       ,255
    DFB  107, "PARK"      ,255
    DFB  108, "PARTY"     ,255
    DFB  109, "MOVE"      ,255
    DFB  110, "PLAY"      ,255
    DFB  111, "HOW"       ,255
    DFB  112, "LOVELY"    ,255
    DFB  113, "CAPITAL"   ,255
    DFB  114, "SKY"       ,255
    DFB  115, "COPY"      ,255
    DFB  116, "SPEED"     ,255
    DFB  117, "GROUP"     ,255
    DFB  118, "PERFECT"   ,255
    DFB  119, "TYPE"      ,255
    DFB  255       ; END OF EWORD_TEST4

    ORG  2500H
EWORD_TEST5:  ; Q N X /
    DFB  152
    DFB  120, "QUIET"     ,255
    DFB  121, "TAXI"      ,255
    DFB  122, "ANIMAL"    ,255
    DFB  123, "AXE"       ,255
    DFB  124, "QUEEN"     ,255
    DFB  125, "NAME"      ,255
    DFB  126, "AXIAL"     ,255
    DFB  127, "QUICKLY"   ,255
    DFB  128, "NATION"    ,255
    DFB  129, "GUN"       ,255
    DFB  130, "QUOTA"     ,255
    DFB  131, "JUNE"      ,255
    DFB  132, "NEVER"     ,255
    DFB  133, "SQUARE"    ,255
    DFB  134, "DONE"      ,255
    DFB  135, "SIXTEEN"   ,255
    DFB  136, "NARROW"    ,255
    DFB  137, "LEARN"     ,255
    DFB  138, "QUIT"      ,255
    DFB  139, "CONTROL"   ,255
    DFB  140, "NEXT"      ,255
    DFB  141, "HAND"      ,255
    DFB  142, "GOING"     ,255
    DFB  143, "NEW"       ,255
    DFB  144, "EXAMPLE"   ,255
    DFB  145, "CHANGE"    ,255
    DFB  146, "NONE"      ,255
    DFB  147, "EXCITE"    ,255
    DFB  148, "ENGLISH"   ,255
    DFB  149, "AGAIN"     ,255
    DFB  255       ; END OF EWORD_TEST5

    ORG  2600H
EWORD_TEST6:  ; B ? Z -
    DFB  147
    DFB  150, "BOX"       ,255
    DFB  151, "ZEBRA"     ,255
    DFB  152, "BASIC"     ,255
    DFB  153, "KEYBOARD"  ,255
    DFB  154, "HOBBY"     ,255
    DFB  155, "BEFORE"    ,255
    DFB  156, "ZOO"       ,255
    DFB  157, "NUMBER"    ,255
    DFB  158, "BODY"      ,255
    DFB  159, "PROBLEM"   ,255
    DFB  160, "BEGIN"     ,255
    DFB  161, "LAZY"      ,255
    DFB  162, "BOOK"      ,255
    DFB  163, "OBEY"      ,255
    DFB  164, "TABLE"     ,255
    DFB  165, "BREAK"     ,255
    DFB  166, "ZOOM"      ,255
    DFB  167, "CRAB"      ,255
    DFB  168, "BUG"       ,255
    DFB  169, "DOZEN"     ,255
    DFB  170, "RUBBER"    ,255
    DFB  171, "SIZE"      ,255
    DFB  172, "BETTER"    ,255
    DFB  173, "MOBILE"    ,255
    DFB  174, "ZERO"      ,255
    DFB  175, "BUILDING"  ,255
    DFB  176, "LOBBY"     ,255
    DFB  177, "BIBLE"     ,255
    DFB  178, "ZONE"      ,255
    DFB  179, "BABY"      ,255
    DFB  255       ; END OF EWORD_TEST6

    ORG  2700H
EWORD_TEST7:
    DFB  0
    ORG  2800H
EWORD_TEST8:
    DFB  0
    ORG  2900H
EWORD_TEST9:
    DFB  0
    ORG  2A00H
EWORD_TEST10:
    DFB  0
    ORG  2B00H
EWORD_TEST11:
    DFB  0

    ORG  3000H
```

```
TWORD_TEST0: ;ฟ ห ก ด อ่ า ส ว
  DFB 54
  DFB 000, "ดาว",255
  DFB 001, "สาวก",255
  DFB 002, "หาก",255
  DFB 003, "สาว",255
  DFB 004, "กว่า",255
  DFB 005, "ฟ่า",255
  DFB 006, "วาด",255
  DFB 007, "ฟัก",255
  DFB 008, "กวด",255
  DFB 009, "ว่า",255
  DFB 010, "หาว",255
  DFB 011, "สวด",255
  DFB 012, "ดก",255
  DFB 013, "หา",255
  DFB 014, "สาด",255
  DFB 015, "หก",255
  DFB 016, "กา",255
  DFB 017, "ฟาด",255
  DFB 018, "กด",255
  DFB 019, "วา",255
  DFB 255                    ; END OF TWORD_TEST0

  ORG 3100H
TWORD_TEST1: ;เ อ้ ง
  DFB 85
  DFB 020, "ฟากฟ้า",255
  DFB 021, "สว่าง",255
  DFB 022, "ห้าง",255
  DFB 023, "ดวง",255
  DFB 024, "กางเกง",255
  DFB 025, "ห่วง",255
  DFB 026, "ด่าง",255
  DFB 027, "ฟาง",255
  DFB 028, "เดา",255
  DFB 029, "ส่ง",255
  DFB 030, "กาหว่า",255
  DFB 031, "หาง",255
  DFB 032, "ก้าง",255
  DFB 033, "ห้า",255
  DFB 034, "ว้าเหว่",255
  DFB 035, "ว่าง",255
  DFB 036, "เก่า",255
  DFB 037, "เก้ง",255
  DFB 038, "ดาดฟ้า",255
  DFB 039, "งด",255
  DFB 255                    ; END OF TWORD_TEST1

  ORG 3200H
TWORD_TEST2: ;พ ะ อี อ้ า ร
  DFB 94
  DFB 040, "พระ",255
  DFB 041, "วัด",255
  DFB 042, "พี่สาว",255
  DFB 043, "เพราะ",255
  DFB 044, "กัด",255
  DFB 045, "รัก",255
  DFB 046, "ฟัง",255
  DFB 047, "ดีด",255
  DFB 048, "ก้าวร้าว",255
  DFB 049, "ก้าพร้า",255
  DFB 050, "วงรี",255
  DFB 051, "หัดพา",255
  DFB 052, "รกร้าง",255
  DFB 053, "สีกา",255
  DFB 054, "ระดำ",255
  DFB 055, "เกะกะ",255
  DFB 056, "สีสรร",255
  DFB 057, "หัวเราะ",255
  DFB 058, "สำเหร่",255
  DFB 059, "ระวัง",255
  DFB 255                    ; END OF TWORD_TEST2

  ORG 3300H
TWORD_TEST3: ;อ อิ ท อึ แ ม
  DFB 103
  DFB 060, "ทอดแห",255
  DFB 061, "หิว",255
  DFB 062, "เสือ",255
  DFB 063, "แมวมอง",255
  DFB 064, "เรื่อง",255
  DFB 065, "ทหาร",255
  DFB 066, "แก้ว",255
  DFB 067, "เอะอะ",255
  DFB 068, "สื่อสาร",255
```

```
  DFB 069, "กระดิก",255
  DFB 070, "อาทร",255
  DFB 071, "เสมอ",255
  DFB 072, "เทวดา",255
  DFB 073, "อีดอาด",255
  DFB 074, "หัวหอม",255
  DFB 075, "แพ้",255
  DFB 076, "มืดมิด",255
  DFB 077, "เดือด",255
  DFB 078, "แรกเริ่ม",255
  DFB 079, "พ่อแม่",255
  DFB 255                    ; END OF TWORD_TEST3

  ORG 3400H
TWORD_TEST4: ; ไ ป น ไ ๆ ผ ย
  DFB 112
  DFB 080, "ปัดเป่า",255
  DFB 081, "ไฟไหม้",255
  DFB 082, "แผน",255
  DFB 083, "น้ำนม",255
  DFB 084, "ยุ่งยาก",255
  DFB 085, "ใส่",255
  DFB 086, "เขี่ยม",255
  DFB 087, "ผ่อนส่ง",255
  DFB 088, "สดใส",255
  DFB 089, "เปิด",255
  DFB 090, "อะไร",255
  DFB 091, "แนะนำ",255
  DFB 092, "เผื่อแผ่",255
  DFB 093, "ยอกย้อน",255
  DFB 094, "ปีใหม่",255
  DFB 095, "เพื่อน",255
  DFB 096, "ดอกไม้",255
  DFB 097, "น้อย",255
  DFB 098, "ฟ้าผ่า",255
  DFB 099, "นักเรียน",255
  DFB 255                    ; END OF TWORD_TEST4

  ORG 3500H
TWORD_TEST5: ; บ ล ฝ
  DFB 109
  DFB 100, "เบาบาง",255
  DFB 101, "ฝาแฝด",255
  DFB 102, "หลบหลีก",255
  DFB 103, "ปรับ",255
  DFB 104, "ฝ่าฟัน",255
  DFB 105, "เวลา",255
  DFB 106, "เหลือง",255
  DFB 107, "บิดา",255
  DFB 108, "ไฝฝ้าน",255
  DFB 109, "เปลือก",255
  DFB 110, "บ่อย",255
  DFB 111, "ทะเล",255
  DFB 112, "ลำบาก",255
  DFB 113, "กระทบ",255
  DFB 114, "ฝึกใฝ่",255
  DFB 115, "สบาย",255
  DFB 116, "เงียบเหงา",255
  DFB 117, "บิดเบือน",255
  DFB 118, "ปลายปี",255
  DFB 119, "ใกล้",255
  DFB 255                    ; END OF TWORD_TEST5

  ORG 3600H
TWORD_TEST6: ; ถ ภ ค ค อ อ
  DFB 110
  DFB 120, "สถิติ",255
  DFB 121, "ภักดี",255
  DFB 122, "ภาคใต้",255
  DFB 123, "สึกหรอ",255
  DFB 124, "พุทรา",255
  DFB 125, "ถอดถอน",255
  DFB 126, "สำเภา",255
  DFB 127, "รอคอย",255
  DFB 128, "ตึกแถว",255
  DFB 129, "คุกคาม",255
  DFB 130, "สุภาพ",255
  DFB 131, "ถนัด",255
  DFB 132, "นึกคิด",255
  DFB 133, "ติดตาม",255
  DFB 134, "คนไทย",255
  DFB 135, "ถามไถ่",255
  DFB 136, "ประเภท",255
  DFB 137, "ทึกทัก",255
  DFB 138, "คัดค้าน",255
  DFB 139, "ตลอด",255
```

```
  DFB 255                    ; END OF TWORD_TEST6

  ORG 3700H
TWORD_TEST7: ; จ ข ช ข ฝ - /
  DFB 119
  DFB 140, "จำใจ",255
  DFB 141, "แก้ไข",255
  DFB 142, "ชุกชุม",255
  DFB 143, "แจกจ่าย",255
  DFB 144, "ข้างขวา",255
  DFB 145, "ชมเชย",255
  DFB 146, "จุกจิก",255
  DFB 147, "แข่งขัน",255
  DFB 148, "ประชุม",255
  DFB 149, "บรรจุ",255
  DFB 150, "ชาวเขา",255
  DFB 151, "ชุ่มชื่น",255
  DFB 152, "จริงจัง",255
  DFB 153, "ตาข่าย",255
  DFB 154, "เชียงราย",255
  DFB 155, "จำหน่าย",255
  DFB 156, "ขยัน",255
  DFB 157, "วิชา",255
  DFB 158, "จุดหมาย",255
  DFB 159, "ชนิด",255
  DFB 255                    ; END OF TWORD_TEST7

  ORG 3800H
TWORD_TEST8: ; โ ฌ อ็ อ ฑ ธ อ็ ฌ
  DFB 111
  DFB 160, "โมโห",255
  DFB 161, "อัชฌา",255
  DFB 162, "ขามเขี้ยน",255
  DFB 163, "กระป๋อง",255
  DFB 164, "บัณฑิต",255
  DFB 165, "ธรณี",255
  DFB 166, "กิ๊บ",255
  DFB 167, "โบราณ",255
  DFB 168, "เข้าฌาน",255
  DFB 169, "ผัดเผ็ด",255
  DFB 170, "กระเป๋า",255
  DFB 171, "กรีฑา",255
  DFB 172, "ธิดา",255
  DFB 173, "มณฑา",255
  DFB 174, "โรงแรม",255
  DFB 175, "ฌาปนกิจ",255
  DFB 176, "แข็งแรง",255
  DFB 177, "ปุ๋ย",255
  DFB 178, "ธานี",255
  DFB 179, "ก๊อกน้ำ",255
  DFB 255                    ; END OF TWORD_TEST8

  ORG 3900H
TWORD_TEST9: ; ฎ ฎ ข ฆ " ฯ
  DFB 101
  DFB 180, "กฎหมาย",255
  DFB 181, "ปฏิบัติ",255
  DFB 182, "ภาษี",255
  DFB 183, "ระฆัง",255
  DFB 184, "ศาสนา",255
  DFB 185, "ฎีกา",255
  DFB 186, "ปฏัก",255
  DFB 187, "โฆษณา",255
  DFB 188, "สุภาษิต",255
  DFB 189, "ศิริยะ",255
  DFB 190, "ชฎา",255
  DFB 191, "กุฎี",255
  DFB 192, "พรรษา",255
  DFB 193, "พิฆาต",255
  DFB 194, "ศาลา",255
  DFB 195, "ดุษฎี",255
  DFB 196, "ปรากฏ",255
  DFB 197, "ขาทิษ",255
  DFB 198, "เฆี่ยนตี",255
  DFB 199, "ศึกษา",255
  DFB 255                    ; END OF TWORD_TEST9

  ORG 3A00H
TWORD_TEST10: ; ฮ ฉ ฒ อ ฮ อ์ ?
  DFB 126
  DFB 200, "ฮีโร่",255
  DFB 201, "โฉนด",255
  DFB 202, "พัฒนา",255
  DFB 203, "ภูเก็ต",255
  DFB 204, "พิมพ์ดีด",255
  DFB 205, "เฮโรอีน",255
```

```
DFB 206,"ปลาฉลาม",255
DFB 207,"วุฒิ",255
DFB 208,"ปู่ย่า",255
DFB 209,"อัฒจันทร์",255
DFB 210,"ฮองเฮา",255
DFB 211,"ฉายา",255
DFB 212,"ผู้เฒ่า",255
DFB 213,"รูปเทียน",255
DFB 214,"อาจารย์",255
DFB 215,"ฮึดฮัด",255
DFB 216,"เฉลิมกรุง",255
DFB 217,"วัฒนธรรม",255
DFB 218,"จมูก",255
DFB 219,"รถยนต์",255
DFB 255                    ; END OF TWORD_TEST10

ORG 3B00H
TWORD_TEST11:;ฤ()ฬฯญฦฐ
DFB 107
DFB 220,"ฤดูฝน",255
DFB 221,"ฤๅษา",255
DFB 222,"ฬุกฬน",255
DFB 223,"ผู้หญิง",255
DFB 224,"ฐานะ",255
DFB 225,"ทฤษฎี",255
DFB 226,"กีฬา",255
DFB 227,"ซ่อมแซม",255
DFB 228,"ปัญญา",255
DFB 229,"รัฐบาล",255
DFB 230,"ฤทัย",255
DFB 231,"ภาพทวีป",255
DFB 232,"ซอกซอย",255
DFB 233,"สัญญา",255
DFB 234,"ปฐม",255
DFB 235,"ฤกษ์ดี",255
DFB 236,"กระเซ็น",255
DFB 237,"สำคัญ",255
DFB 238,"ภูมิฐาน",255
DFB 239,"เชษฐา",255
DFB 255                    ; END OF TWORD_TEST11

   ORG 2C00H
ENG_KEY0:
   DFB
037,037,036,036,035,035,034,034,031,0
31,030,030,029,029,028,028
; LESSON 0   (;)   ; (L)   L (K)   K
(J)   J (F)   F (D)   D (S)   S (A)
A
   ORG 2C10H
ENG_KEY1:
   DFB
036,051,034,020,031,032,030,016
; LESSON 1   (L)   . (J)   U (F)   G
(D)   E
   ORG 2C20H
ENG_KEY2:
   DFB
035,050,036,022,034,033,031,017
; LESSON 2   (K)   , (L)   O (J)   H
(F)   R
   ORG 2C30H
ENG_KEY3:
   DFB
037,106,034,049,035,021,030,045,031,0
18
; LESSON 3   (;)   : (J)   M (K)   I
(D)   C (F)   T
   ORG 2C40H
ENG_KEY4:
   DFB
037,023,034,019,031,046,029,015
; LESSON 4   (;)   P (J)   Y (F)   V
(S)   W
   ORG 2C50H
ENG_KEY5:
   DFB
037,052,029,044,034,048,028,014
; LESSON 5   (;)   / (S)   X (J)   N
(A)   Q
   ORG 2C60H
ENG_KEY6:
   DFB
037,010,028,043,037,111,031,047
; LESSON 6   (;)   - (A)   Z (;)   ?
(F)   B
   ORG 2C70H
ENG_KEY7:
   DFB
031,005,031,004,031,003,030,002,029,0
01,028,000
; LESSON 7   (F)   6 (F)   5 (F)   4
(D)   3 (S)   2 (A)   1
   ORG 2C80H
ENG_KEY8:
   DFB
037,009,036,008,035,007,034,006
; LESSON 8   (;)   0 (L)   9 (K)   8
(J)   7

   ORG 2E00H
NO_OF_ENG:
   DFB 8, 4, 4, 5, 4, 4, 4, 6, 4, 0,
0 ,0
   ORG 2E10H
SUM_OF_ENG:
   DFB
8,12,16,21,25,29,33,39,43,43,43,43

   ORG 3C00H
THAI_KEY0:
```

```
   DFB
144,144,143,143,142,142,141,141,138,1
38,137,137,136,136,135,135
; LESSON 0   (ว) ว(ส) ส(า) า(อ่) อ่(ด) ด(ก) ก
(ห) ห(ฟ) ฟ
   ORG  3C10H
THAI_KEY1:
   DFB  144,145,141,140,138,139
; LESSON 1   (ว) ง(อ่) อ้(ค) เ
   ORG  3C20H
THAI_KEY2:
   DFB
142,130,137,125,141,128,141,129,138,1
27,138,126
; LESSON 2   (า) ร(ก) ำ(อ่) อ้(อ่) อี(ด) ะ(ด) พ
   ORG  3C30H
THAI_KEY3:
   DFB
142,154,137,149,141,152,141,153,138,1
51,138,150
; LESSON 3   (า) ม(ก) แ(อ่) อื(อ่) ท(ด) อิ(ด) อ
   ORG  3C40H
THAI_KEY4:
   DFB
144,132,135,147,135,123,143,155,143,1
31,136,148,136,124
; LESSON 4   (ว) ข(ฟ) ผ(ฟ) ๆ(ส) ไ(ส) น(ห)
ป(ห) ไ
   ORG  3C50H
THAI_KEY5:
   DFB  144,156,144,134,144,133
; LESSON 5   (ว) ฝ(ว) ล(ว) บ
   ORG  3C60H
THAI_KEY6:
   DFB
138,116,141,117,142,119,141,118,137,1
14,138,115
; LESSON 6   (ด) อุ(อ่) อื(า)ต (อ่)ค (ก)ภ (ด) ถ
   ORG  3C70H
THAI_KEY7:
   DFB
135,112,136,113,144,122,144,121,143,1
20
; LESSON 7   (ฟ) /(ห) -(ว) ช(ว) ข(ส) จ
   ORG  3C80H
THAI_KEY8:
   DFB
142,200,141,199,138,198,138,197,141,2
12,141,211,138,210,138,209
; LESSON 8   (า) ณ(อ่) อี(ด) ธ(ด) ฑ(อ่) อ๋(อ่) อี๋
(ด) ฌ(ด) โ
   ORG  3C90H
THAI_KEY9:
   DFB
143,201,143,214,136,195,136,207,142,2
13,137,208,137,196
; LESSON 9   (ส) ฯ(ส) ศ(ห) "(ห) ฆ(า) บ(ก)
ฎ(ก) ฎ
   ORG  3CA0H
THAI_KEY10:
   DFB
141,222,141,221,141,188,138,187,142,2
23,137,219,138,220
; LESSON 10  (อ่) ?(อ่) อ์(อ่) ฺ(ด) อู(า) ฒ(ก) ฉ
(ด) ฮ
   ORG  3CB0H
THAI_KEY11:
   DFB
144,203,144,225,144,202,144,215,143,2
24,136,218,135,217,135,206
; LESSON 11  (ว) ฐ(ว) ฦ(ว) ญ(ว) ซ(ส) ห(ห)
)(ฟ) ((ฟ) ฤ

   ORG  2D00H
RND_KEY_ENG:
   DFB
037,036,035,034,031,030,029,028
; LESSON 0     ;   L   K   J   F   D
S   A
   DFB  051,020,032,016
; LESSON 1     .   U   G   E
   DFB  050,022,033,017
; LESSON 2     ,   O   H   R
   DFB  106,049,021,045,018
; LESSON 3     :   M   I   C   T
   DFB  023,019,046,015
; LESSON 4     P   Y   V   W
   DFB  052,044,048,014
; LESSON 5     /   X   N   Q
   DFB  010,043,111,047
; LESSON 6     -   Z   ?   B
   DFB  005,004,003,002,001,000
; LESSON 7     6   5   4   3   2   1
   DFB  009,008,007,006
; LESSON 8     0   9   8   7

   ORG  3D00H
RND_KEY_THAI:
   DFB
144,143,142,141,138,137,136,135
; LESSON 0     ว ส า อ่ ด ก ห ฟ
   DFB  145,140,139
; LESSON 1     ง อ้ เ
   DFB  130,125,128,129,127,126
; LESSON 2     ร ำ อั อี ะ พ
   DFB  154,149,152,153,151,150
; LESSON 3     ม แ อื ท อิ อ
   DFB  132,147,123,155,131,148,124
; LESSON 4     ข ผ ๆ ใ น ป ไ
   DFB  156,134,133
; LESSON 5     ฝ ล บ
   DFB  116,117,119,118,114,115
; LESSON 6     อุ อื ต ค ภ ถ
   DFB  112,113,122,121,120
; LESSON 7     /   -   ช ข จ
   DFB
200,199,198,197,212,211,210,209
; LESSON 8     ณ อั ธ ฑ อ๋ อี๋ ฌ โ
   DFB  201,214,195,207,213,208,196
; LESSON 9     ฯ ศ " ฆ บ ฎ ฎ
```

```
   DFB  222,221,188,187,223,219,220
; LESSON 10    ?อ๊ ๒ อู ฒ ฉ ฮ
   DFB
203,225,202,215,224,218,217,206
; LESSON 11    ฐ ฦ ญ ช ฬ ) ( ฤ

   ORG  3E00H
NO_OF_THAI:
   DFB  8, 3, 6, 6, 7, 3, 6, 5, 8, 7,
7, 8
   ORG  3E10H
SUM_OF_THAI:
   DFB
8,11,17,23,30,33,39,44,52,59,66,74

   END
```

ภาคผนวก ง.
โปรแกรมของส่วนการแก้ไขแบบฝึกหัด

```
program ExEdit;
uses
  Forms,
  MainExEdit in 'MainExEdit.pas' {Form1},
  KeyEdit in 'KeyEdit.pas' {Form2},
  SelFile in 'SelFile.pas' {Form3},
  ExProg in 'ExProg.pas' {Form4};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm4, Form4);
  Application.Run;
end.

//=============================================================
unit MainExEdit;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,  StdCtrls, ExtCtrls, Buttons;
type
  TForm1 = class(TForm)
    SelectLesson  : TRadioGroup;
    SelectLang    : TRadioGroup;
    KeyToPractice : TPanel;
    PracticeKey   : TLabel;
    WordList      : TListBox;
    BtnChangeKey  : TButton;
    BtnSave       : TBitBtn;
    BtnOpen       : TBitBtn;
    BtnProgram    : TBitBtn;
    BtnExit       : TBitBtn;
    BtnUp         : TBitBtn;
    BtnDown       : TBitBtn;
    BtnAdd        : TBitBtn;
    BtnRemove     : TBitBtn;
    BtnNew        : TBitBtn;
    OpenDialog1   : TOpenDialog;
    SaveDialog1   : TSaveDialog;
    WordEdit      : TEdit;
    TotalWord     : TLabel;
    Panel1        : TPanel;
    SelectFile    : TLabel;
    BtnOk         : TButton;
    EngWordList   : TListBox;
    ThaiWordList  : TListBox;

    procedure ShowKey;
    procedure NewFile;
    procedure BtnChangeKeyClick(Sender: TObject);
    procedure BtnNewClick(Sender: TObject);
    procedure BtnOpenClick(Sender: TObject);
    procedure BtnSaveClick(Sender: TObject);
    procedure BtnProgramClick(Sender: TObject);
    procedure BtnExitClick(Sender: TObject);
    procedure SelectLangClick(Sender: TObject);
    procedure SelectLessonClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BtnAddClick(Sender: TObject);
    procedure BtnRemoveClick(Sender: TObject);
    procedure BtnUpClick(Sender: TObject);
    procedure BtnDownClick(Sender: TObject);
    procedure WordListClick(Sender: TObject);
    procedure WordEditChange(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure BtnOkClick(Sender: TObject);
    procedure WordListDblClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  Language : Byte;
  Lesson   : Byte;
  Key      : array[1..2,1..12,1..8] of char;
  NoOfKey  : array[1..2,1..12]  of byte;
  Word     : array[1..2,1..12,1..30] of string;
  WordNo   : array[1..2,1..12,1..30] of byte;
  NoOfWord : array[1..2,1..12]  of byte;
  FileName : array[1..2,1..256] of string[80];
  CurrentFileName : string;
  OldWord  : string;
  AsciiTable : array[1..2,1..255] of byte;

implementation
uses KeyEdit, SelFile, ExProg;
{$R *.DFM}

//-------------------------------------------------------------
procedure TForm1.ShowKey;
var  i : integer;
begin
    KeyToPractice.caption := '';
    for i:=1 to NoOfKey[Language,Lesson] do
    begin
      KeyToPractice.caption := KeyToPractice.caption +
                               Key[Language,Lesson,i];
      KeyToPractice.caption := KeyToPractice.caption +
                               '  ';
    end;
    WordList.Items.Clear;
    for i:=1 to NoOfWord[Language,Lesson] do
      WordList.Items.Add(Word[Language,Lesson,i]);
    if NoOfWord[Language,Lesson] = 0 then
      BtnRemove.Enabled := false
        else BtnRemove.Enabled := true;
    if NoOfKey[Language,Lesson] = 0 then
      BtnAdd.Enabled := false
        else BtnAdd.Enabled := true;
    WordList.ItemIndex := 0;
    BtnUp.Enabled := false;
    if NoOfWord[Language,Lesson] < 2 then
      BtnDown.Enabled := false
        else BtnDown.Enabled := true;
    if Lesson = 1 then BtnChangeKey.Enabled := false
      else BtnChangeKey.Enabled := true;
    OldWord := '';
    WordEdit.Text := '';
    if NoOfWord[Language,Lesson]=30 then
       BtnAdd.Enabled := false;
    TotalWord.Caption := 'จำนวนคำ : ' + inttostr(
                         NoOfWord[Language,Lesson]);
    if (Language=1)and(ThaiWordList.Items.Count = 0)
       then BtnAdd.Enabled := false;
    if (Language=2)and(EngWordList.Items.Count = 0)
       then BtnAdd.Enabled := false;
end;
```

```
//---------------------------------------------------------------
procedure TForm1.BtnChangeKeyClick(Sender:
TObject);
begin
   Form1.Enabled := False;
   Application.CreateForm(TForm2,Form2);
   Form2.Show;
end;

//---------------------------------------------------------------
procedure TForm1.BtnNewClick(Sender: TObject);
begin
   Form1.Enabled := False;
   Application.CreateForm(TForm3,Form3);
   Form3.Show;
end;

//---------------------------------------------------------------
procedure TForm1.NewFile;
var  i,j,k : integer;
   LoadFile : textfile;
begin
   SelectLang.Enabled := true;
   SelectLesson.Enabled := true;
   BtnChangeKey.Enabled := true;
   WordList.Enabled := true;
   BtnAdd.Enabled := true;
   BtnRemove.Enabled := true;
   BtnUp.Enabled := true;
   BtnDown.Enabled := true;
   BtnSave.Enabled := true;
   BtnProgram.Enabled := true;
   for i:=1 to 2 do
     for j:=1 to 255 do
     AsciiTable[i,j] := 255;
   Language := 1;
   Lesson := 1;
   SelectLang.ItemIndex := 0;
   SelectLesson.ItemIndex := 0;
   Form1.Caption := 'Edit Lesson [Untitled]';
   CurrentFileName := '';
   for i:=1 to 12 do
   begin
     NoofKey[1,i] := 0;
     NoofKey[2,i] := 0;
   end;
   for i:=1 to 2 do
     for j:=1 to 12 do
       for k:=1 to 8 do
       Key[i,j,k] := ' ';
   for i:=1 to 2 do
     for j:=1 to 12 do
        NoOfWord[i,j] := 0;
   for i:=1 to 2 do
     for j:=1 to 12 do
       for k:=1 to 30 do
       begin
         Word[i,j,k] := '';
         WordNo[i,j,k] := 255;
       end;
   for i:=1 to 2 do
     for j:=1 to 256 do
       FileName[i,j] := '';
   AssignFile(LoadFile,ThaiFile);
   Reset(LoadFile);
   i := 0;
   ThaiWordList.Items.Clear;
   While not(eof(LoadFile)) do
   begin
     inc(i);
     readln(Loadfile,FileName[1,i]);
     ThaiWordList.Items.Add(FileName[1,i]);
   end;
   CloseFile(LoadFile);
   AssignFile(LoadFile,EngFile);
   Reset(LoadFile);
   i := 0;
   EngWordList.Items.Clear;
   While not(eof(LoadFile)) do
   begin
     inc(i);
     readln(Loadfile,FileName[2,i]);
     EngWordList.Items.Add(FileName[2,i]);
   end;
   CloseFile(LoadFile);
   NoOfKey[1,1] := 8;
   NoOfKey[2,1] := 8;
   Key[1,1] := ('ฟหกดำสว');
   Key[2,1] := ('ASDFJKL;');
   for i:= 1 to 2 do
     for k:= 1 to 8 do
   AsciiTable[i,ord(Key[i,1,k])] := 1;
   ShowKey;
end;

//---------------------------------------------------------------
procedure TForm1.BtnOpenClick(Sender: TObject);
var  LoadFile : textfile;
   i,j,k : integer;
   tmp : string;
   Ok : Boolean;
begin
   OpenDialog1.FileName := CurrentFileName;
   OpenDialog1.Execute;
   if OpenDialog1.Filename = '' then exit;
   for i:=1 to 2 do
     for j:=1 to 255 do
     AsciiTable[i,j] := 255;
   CurrentFileName := OpenDialog1.Filename;
   SelectLang.Enabled := true;
   SelectLesson.Enabled := true;
   BtnChangeKey.Enabled := true;
   WordList.Enabled := true;
   BtnAdd.Enabled := true;
   BtnRemove.Enabled := true;
   BtnUp.Enabled := true;
   BtnDown.Enabled := true;
   BtnSave.Enabled := true;
   BtnProgram.Enabled := true;
   Form1.Caption := 'Edit Lesson ['+ExtractFileName(
                CurrentFileName)+']';
   AssignFile(LoadFile,CurrentFileName);
   Reset(LoadFile);
   for i:=1 to 2 do
     for j:=1 to 12 do
     begin
       readln(LoadFile,tmp);
       NoOfKey[i,j] := strtoint(tmp);
     end;
   for i:=1 to 2 do
     for j:=1 to 12 do
       for k:=1 to 8 do
```

```
        begin
          Readln(LoadFile,tmp);
          Key[i,j,k] := tmp[1];
          AsciiTable[i,ord(Key[i,j,k])] := j;
        end;
    for i:=1 to 2 do
      for j:=1 to 12 do
      begin
        readln(LoadFile,tmp);
        NoOfWord[i,j] := strtoint(tmp);
      end;
    for i:=1 to 2 do
      for j:=1 to 12 do
        for k:=1 to 30 do
          Readln(LoadFile,Word[i,j,k]);
    for i:=1 to 2 do
      for j:=1 to 12 do
        for k:=1 to 30 do
        begin
          Readln(LoadFile,tmp);
          if tmp<>'' then WordNo[i,j,k] := strtoint(tmp)
            else WordNo[i,j,k] := 255;
        end;
    for i:=1 to 2 do
      for j:=1 to 256 do
        Readln(LoadFile,FileName[i,j]);
    Readln(LoadFile,ThaiFile);
    Readln(LoadFile,EngFile);
    CloseFile(LoadFile);
    ThaiWordList.Items.Clear;
    EngWordList.Items.Clear;
    for i:=1 to 256 do
    begin
      Ok := true;
      if FileName[1,i]<>'' then
      begin
        for j:=1 to 12 do
          for k:=1 to NoOfWord[1,j] do
            if((i-1)=WordNo[1,j,k]) then Ok := false;
        if Ok then ThaiWordList.Items.Add(
                                    FileName[1,i]);
      end;
      Ok := true;
      if FileName[2,i]<>'' then
      begin
        for j:=1 to 12 do
          for k:=1 to NoOfWord[2,j] do
            if((i-1)=WordNo[2,j,k]) then Ok := false;
        if Ok then EngWordList.Items.Add(
                                    FileName[2,i]);
      end;
    end;
    if ThaiWordList.Items.Count = 0 then
       BtnAdd.Enabled := false;
    Language := 1;
    Lesson := 1;
    SelectLang.ItemIndex := 0;
    SelectLesson.ItemIndex := 0;
    ShowKey;
end;

//---------------------------------------------------------------
procedure TForm1.BtnSaveClick(Sender: TObject);
var  SaveFile : textfile;
    i,j,k : integer;
begin
    SaveDialog1.FileName := CurrentFileName;
    SaveDialog1.Execute;
    if SaveDialog1.Filename = '' then exit;
    CurrentFileName := SaveDialog1.Filename;
    Form1.Caption := 'Edit Lesson ['+ExtractFileName(
                        CurrentFileName)+']';
    AssignFile(SaveFile,CurrentFileName);
    Rewrite(SaveFile);
    for i:=1 to 2 do
      for j:=1 to 12 do
        writeln(SaveFile,inttostr(NoOfKey[i,j]));
    for i:=1 to 2 do
      for j:=1 to 12 do
        for k:=1 to 8 do
        writeln(SaveFile,Key[i,j,k]);
    for i:=1 to 2 do
      for j:=1 to 12 do
        writeln(SaveFile,inttostr(NoOfWord[i,j]));
    for i:=1 to 2 do
      for j:=1 to 12 do
        for k:=1 to 30 do
        writeln(SaveFile,Word[i,j,k]);
    for i:=1 to 2 do
      for j:=1 to 12 do
        for k:=1 to 30 do
        writeln(SaveFile,inttostr(WordNo[i,j,k]));
    for i:=1 to 2 do
      for j:=1 to 256 do
        writeln(SaveFile,FileName[i,j]);
    writeln(SaveFile,ThaiFile);
    writeln(SaveFile,EngFile);
    CloseFile(SaveFile);
end;

//---------------------------------------------------------------
procedure TForm1.BtnProgramClick(Sender: TObject);
begin
    Form1.Enabled := False;
    Application.CreateForm(TForm4,Form4);
    Form4.Show;
end;

//---------------------------------------------------------------
procedure TForm1.BtnExitClick(Sender: TObject);
begin
    Halt(1);
end;

//---------------------------------------------------------------
procedure TForm1.SelectLangClick(Sender: TObject);
begin
    if SelectLang.ItemIndex = 0 then Language := 1
      else Language := 2;
    ShowKey;
end;

//---------------------------------------------------------------
procedure TForm1.SelectLessonClick(Sender:
TObject);
begin
    Lesson := SelectLesson.ItemIndex+1;
    ShowKey;
end;

//---------------------------------------------------------------
procedure TForm1.FormCreate(Sender: TObject);
begin
    Lesson := 2;
```

```
    Language := 2;
    Panel1.Left := 0;
    Panel1.Top := 0;
    Panel1.Width := Form1.ClientWidth;
    Panel1.Height := Form1.ClientHeight;
    ThaiWordList.Width := Panel1.Width - 16;
    ThaiWordList.Height := Panel1.Height - 80;
    EngWordList.Left := ThaiWordList.Left;
    EngWordList.Width := ThaiWordList.Width;
    EngWordList.Height := ThaiWordList.Height;
    BtnOk.Top := ThaiWordList.Top +
                    ThaiWordList.Height +8;
end;

//---------------------------------------------------------------
procedure TForm1.BtnAddClick(Sender: TObject);
var  AddText : string;
begin
    AddText := WordEdit.Text;
    if length(AddText)=0 then
    begin
      MessageDlg('Empty Input',MtError,[MbOk],1);
      exit;
    end;
    Panel1.Visible := True;
    SelectFile.Caption := 'กรุณาเลือกไฟล์ของคำ : ' +
                          AddText;
    if SelectLang.ItemIndex = 0 then
    begin
      ThaiWordList.Visible := True;
      EngWordList.Visible := False;
    end
    else
    begin
      ThaiWordList.Visible := False;
      EngWordList.Visible := True;
    end;
    WordList.Items.Add(AddText);
    NoOfWord[Language,Lesson] := NoOfWord[
                                 Language,Lesson] +1;
    TotalWord.Caption := 'จำนวนคำ : ' + inttostr(
                         NoOfWord[Language,Lesson]);
    Word[Language,Lesson,NoOfWord[
                         Language,Lesson]] := AddText;
    WordEdit.Text := '';
    WordList.ItemIndex := NoOfWord[
                          Language,Lesson]-1;
    if NoOfWord[Language,Lesson] = 30 then
        BtnAdd.Enabled := false;
    if NoOfWord[Language,Lesson] > 0 then
        BtnRemove.Enabled := true;
end;

//---------------------------------------------------------------
procedure TForm1.BtnOkClick(Sender: TObject);
var  FName : string;
    i : integer;
begin
    if Language = 1 then
    begin
      if ThaiWordList.Itemindex = -1 then exit;
      FName := ThaiWordList.Items.Strings[
                  ThaiWordList.Itemindex];
      ThaiWordList.Items.Delete(
                              ThaiWordList.Itemindex);
    end
    else
    begin
      if EngWordList.Itemindex = -1 then exit;
      FName := EngWordList.Items.Strings[
                  EngWordList.Itemindex];
      EngWordList.Items.Delete(
                              EngWordList.Itemindex);
    end;
    i := 1;
    repeat
      if FName = FileName[Language,i] then
      begin
        WordNo[Language,Lesson,NoOfWord[
                              Language,Lesson]] := i-1;
        i := 255;
      end;
      i := i+1;
    until i>255;
    Panel1.Visible := False;
    if (Language=1)and(ThaiWordList.Items.Count = 0)
        then BtnAdd.Enabled := false;
    if (Language=2)and(EngWordList.Items.Count = 0)
        then BtnAdd.Enabled := false;
end;

//---------------------------------------------------------------
procedure TForm1.BtnRemoveClick(Sender: TObject);
var  i,index : integer;
begin
    index := WordList.ItemIndex;
    WordList.Items.Delete(index);
    WordList.ItemIndex := index;
    if NoOfWord[Language,Lesson] > 0 then
    begin
      if Language = 1 then
      begin
        ThaiWordList.Items.add(FileName[Language,
              WordNo[Language,Lesson,Index+1]+1]);
      end
      else
      begin
        EngWordList.Items.add(FileName[Language,
              WordNo[Language,Lesson,Index+1]+1]);
      end;
      dec(NoOfWord[Language,Lesson]);
      for i:=index+1 to NoOfWord[Language,Lesson] do
      begin
        Word[Language,Lesson,i] := Word[Language,
                                  Lesson,i+1];
        WordNo[Language,Lesson,i] := WordNo[
                                  Language,Lesson,i+1];
      end;
      Word[Language,Lesson,NoOfWord[Language,
                                  Lesson]+1] := '';
      WordNo[Language,Lesson,NoOfWord[Language,
                                  Lesson]+1] := 255;

      TotalWord.Caption := 'จำนวนคำ : ' + inttostr(
                          NoOfWord[Language,Lesson]);
    end;
    if NoOfWord[Language,Lesson] = 0 then
        BtnRemove.Enabled := false;
    if index >0 then WordList.ItemIndex := index-1;
    if NoOfWord[Language,Lesson] < 30 then
BtnAdd.Enabled := true;
end;
```

```
//---------------------------------------------------------------------
procedure TForm1.BtnUpClick(Sender: TObject);
var  index : integer;
    tmp : string;
    tmp2 : byte;
begin
    index := WordList.ItemIndex;
    if index > 0 then
    begin
      WordList.Items.Move(index,index-1);
      WordList.ItemIndex := index-1;
      tmp := Word[Language,Lesson,index+1];
      Word[Language,Lesson,index+1] := Word[
                              Language,Lesson,index];
      Word[Language,Lesson,index] := tmp;
      tmp2 := WordNo[Language,Lesson,index+1];
      WordNo[Language,Lesson,index+1] := WordNo[
                              Language,Lesson,index];
      WordNo[Language,Lesson,index] := tmp2;
      WordListClick(Sender);
    end;
end;

//---------------------------------------------------------------------
procedure TForm1.BtnDownClick(Sender: TObject);
var  index : integer;
    tmp : string;
    tmp2 : byte;
begin
    index := WordList.ItemIndex;
    if index < NoOfWord[Language,Lesson]-1 then
    begin
      WordList.Items.Move(index,index+1);
      WordList.ItemIndex := index+1;
      tmp := Word[Language,Lesson,index+1];
      Word[Language,Lesson,index+1] := Word[
                            Language,Lesson,index+2];
      Word[Language,Lesson,index+2] := tmp;
      tmp2 := WordNo[Language,Lesson,index+1];
      WordNo[Language,Lesson,index+1] := WordNo[
                            Language,Lesson,index+2];
      WordNo[Language,Lesson,index+2] := tmp2;
      WordListClick(Sender);
    end;
end;

//---------------------------------------------------------------------
procedure TForm1.WordListClick(Sender: TObject);
begin
    if WordList.ItemIndex = 0 then
      BtnUp.Enabled := false
        else BtnUp.Enabled := true;
    if WordList.ItemIndex = NoOfWord[
                                  Language,Lesson]-1
      then BtnDown.Enabled := false
        else BtnDown.Enabled := true;
end;

//---------------------------------------------------------------------
procedure TForm1.WordEditChange(Sender: TObject);
var  NewWord : string;
    i : integer;
    TLesson : integer;
begin
    NewWord := WordEdit.Text;
    for i := 1 to length(Newword) do
    begin
      TLesson := AsciiTable[Language,ord(
                                    NewWord[i])];
      if TLesson > Lesson then
      begin
        WordEdit.Text := OldWord;
        NewWord := OldWord;
      end;
    end;
    WordEdit.SelStart := length(NewWord);
    OldWord := NewWord;
end;

//---------------------------------------------------------------------
procedure TForm1.FormActivate(Sender: TObject);
begin
    ShowKey;
end;

//---------------------------------------------------------------------
procedure TForm1.WordListDblClick(Sender:
                                        TObject);
var i : integer;
    s : string;
begin
    i := WordList.ItemIndex+1;
    s := Word[Language,Lesson,i]+' = ';
    s := s + inttostr(WordNo[Language,Lesson,i])+':';
    s := s + FileName[Language,WordNo[
                                Language,Lesson,i]+1];
    MessageDlg(s,mtInformation,[mbOk],0);
end;
end.

//=====================================================================
unit KeyEdit;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Buttons, ExtCtrls;
type
  TForm2 = class(TForm)
    BtnOK     : TButton;
    BtnCancel  : TButton;
    NoOfLesson : TLabel;
    OldKeyList : TPanel;
    OldKey     : TLabel;
    NewKeyList : TPanel;
    NewKey     : TLabel;
    Pn001 : TPanel;   Pn002 : TPanel;   Pn003 : TPanel;
    Pn004 : TPanel;   Pn005 : TPanel;   Pn006 : TPanel;
    Pn007 : TPanel;   Pn008 : TPanel;   Pn009 : TPanel;
    Pn010 : TPanel;   Pn011 : TPanel;   Pn012 : TPanel;
    Pn013 : TPanel;   Pn014 : TPanel;   Pn015 : TPanel;
    Pn016 : TPanel;   Pn017 : TPanel;   Pn018 : TPanel;
    Pn019 : TPanel;   Pn020 : TPanel;   Pn021 : TPanel;
    Pn022 : TPanel;   Pn023 : TPanel;   Pn024 : TPanel;
    Pn025 : TPanel;   Pn026 : TPanel;   Pn027 : TPanel;
    Pn028 : TPanel;   Pn029 : TPanel;   Pn030 : TPanel;
    Pn031 : TPanel;   Pn032 : TPanel;   Pn033 : TPanel;
    Pn034 : TPanel;   Pn035 : TPanel;   Pn036 : TPanel;
    Pn037 : TPanel;   Pn038 : TPanel;   Pn039 : TPanel;
    Pn040 : TPanel;   Pn041 : TPanel;   Pn042 : TPanel;
    Pn043 : TPanel;   Pn044 : TPanel;   Pn045 : TPanel;
    Pn046 : TPanel;   Pn047 : TPanel;   Pn048 : TPanel;
    Pn049 : TPanel;   Pn050 : TPanel;   Pn051 : TPanel;
    Pn052 : TPanel;   Pn053 : TPanel;   Pn054 : TPanel;
    Pn055 : TPanel;   Pn056 : TPanel;   Pn057 : TPanel;
```

```
Pn058 : TPanel;  Pn059 : TPanel;  Pn060 : TPanel;
Pn061 : TPanel;  Pn062 : TPanel;  Pn063 : TPanel;
Pn064 : TPanel;  Pn065 : TPanel;  Pn066 : TPanel;
Pn067 : TPanel;  Pn068 : TPanel;  Pn069 : TPanel;
Pn070 : TPanel;  Pn071 : TPanel;  Pn072 : TPanel;
Pn073 : TPanel;  Pn074 : TPanel;  Pn075 : TPanel;
Pn076 : TPanel;  Pn077 : TPanel;  Pn078 : TPanel;
Pn079 : TPanel;  Pn080 : TPanel;  Pn081 : TPanel;
Pn082 : TPanel;  Pn083 : TPanel;  Pn084 : TPanel;
Pn085 : TPanel;  Pn086 : TPanel;  Pn087 : TPanel;
Pn088 : TPanel;  Pn089 : TPanel;  Pn090 : TPanel;
Pn091 : TPanel;  Pn092 : TPanel;  Pn093 : TPanel;
Pn094 : TPanel;  Pn095 : TPanel;  Pn096 : TPanel;
Pn097 : TPanel;  Pn098 : TPanel;  Pn099 : TPanel;
Pn100 : TPanel;  Pn101 : TPanel;  Pn102 : TPanel;
Pn103 : TPanel;  Pn104 : TPanel;  Pn105 : TPanel;
Pn106 : TPanel;  Pn107 : TPanel;  Pn108 : TPanel;
Pn109 : TPanel;  Pn110 : TPanel;  Pn111 : TPanel;
Pn112 : TPanel;  Pn113 : TPanel;  Pn114 : TPanel;
Pn115 : TPanel;  Pn116 : TPanel;  Pn117 : TPanel;
Pn118 : TPanel;  Pn119 : TPanel;  Pn120 : TPanel;
Pn121 : TPanel;  Pn122 : TPanel;  Pn123 : TPanel;
Pn124 : TPanel;  Pn125 : TPanel;  Pn126 : TPanel;
Pn127 : TPanel;  Pn128 : TPanel;  Pn129 : TPanel;
Pn130 : TPanel;  Pn131 : TPanel;  Pn132 : TPanel;
Pn133 : TPanel;  Pn134 : TPanel;  Pn135 : TPanel;
Pn136 : TPanel;  Pn137 : TPanel;  Pn138 : TPanel;
Pn139 : TPanel;  Pn140 : TPanel;  Pn141 : TPanel;
Pn142 : TPanel;  Pn143 : TPanel;  Pn144 : TPanel;
Pn145 : TPanel;  Pn146 : TPanel;  Pn147 : TPanel;
Pn148 : TPanel;  Pn149 : TPanel;  Pn150 : TPanel;
Pn151 : TPanel;  Pn152 : TPanel;  Pn153 : TPanel;
Pn154 : TPanel;  Pn155 : TPanel;  Pn156 : TPanel;
Pn157 : TPanel;  Pn158 : TPanel;

procedure Initial;
procedure SelectBtn(Sel:byte; Number:integer);
procedure PressBtn(KeyIn:byte;Number:byte);
procedure BtnCancelClick(Sender: TObject);
procedure BtnOKClick(Sender: TObject);
procedure FormClose(Sender: TObject;
    var Action: TCloseAction);
procedure FormCreate(Sender: TObject);
procedure Pn001Click(Sender: TObject);
procedure Pn002Click(Sender: TObject);
procedure Pn003Click(Sender: TObject);
procedure Pn004Click(Sender: TObject);
procedure Pn005Click(Sender: TObject);
procedure Pn006Click(Sender: TObject);
procedure Pn007Click(Sender: TObject);
procedure Pn008Click(Sender: TObject);
procedure Pn009Click(Sender: TObject);
procedure Pn010Click(Sender: TObject);
procedure Pn011Click(Sender: TObject);
procedure Pn02Click(Sender: TObject);
procedure Pn013Click(Sender: TObject);
procedure Pn014Click(Sender: TObject);
procedure Pn015Click(Sender: TObject);
procedure Pn016Click(Sender: TObject);
procedure Pn017Click(Sender: TObject);
procedure Pn018Click(Sender: TObject);
procedure Pn019Click(Sender: TObject);
procedure Pn020Click(Sender: TObject);
procedure Pn021Click(Sender: TObject);
procedure Pn022Click(Sender: TObject);
procedure Pn023Click(Sender: TObject);
procedure Pn024Click(Sender: TObject);
procedure Pn025Click(Sender: TObject);
procedure Pn026Click(Sender: TObject);
procedure Pn027Click(Sender: TObject);
procedure Pn028Click(Sender: TObject);
procedure Pn029Click(Sender: TObject);
procedure Pn030Click(Sender: TObject);
procedure Pn031Click(Sender: TObject);
procedure Pn032Click(Sender: TObject);
procedure Pn033Click(Sender: TObject);
procedure Pn034Click(Sender: TObject);
procedure Pn035Click(Sender: TObject);
procedure Pn036Click(Sender: TObject);
procedure Pn037Click(Sender: TObject);
procedure Pn038Click(Sender: TObject);
procedure Pn039Click(Sender: TObject);
procedure Pn040Click(Sender: TObject);
procedure Pn041Click(Sender: TObject);
procedure Pn042Click(Sender: TObject);
procedure Pn043Click(Sender: TObject);
procedure Pn044Click(Sender: TObject);
procedure Pn045Click(Sender: TObject);
procedure Pn046Click(Sender: TObject);
procedure Pn047Click(Sender: TObject);
procedure Pn048Click(Sender: TObject);
procedure Pn049Click(Sender: TObject);
procedure Pn050Click(Sender: TObject);
procedure Pn051Click(Sender: TObject);
procedure Pn052Click(Sender: TObject);
procedure Pn053Click(Sender: TObject);
procedure Pn054Click(Sender: TObject);
procedure Pn055Click(Sender: TObject);
procedure Pn056Click(Sender: TObject);
procedure Pn057Click(Sender: TObject);
procedure Pn058Click(Sender: TObject);
procedure Pn059Click(Sender: TObject);
procedure Pn060Click(Sender: TObject);
procedure Pn061Click(Sender: TObject);
procedure Pn062Click(Sender: TObject);
procedure Pn063Click(Sender: TObject);
procedure Pn064Click(Sender: TObject);
procedure Pn065Click(Sender: TObject);
procedure Pn066Click(Sender: TObject);
procedure Pn067Click(Sender: TObject);
procedure Pn068Click(Sender: TObject);
procedure Pn069Click(Sender: TObject);
procedure Pn070Click(Sender: TObject);
procedure Pn071Click(Sender: TObject);
procedure Pn072Click(Sender: TObject);
procedure Pn073Click(Sender: TObject);
procedure Pn074Click(Sender: TObject);
procedure Pn075Click(Sender: TObject);
procedure Pn076Click(Sender: TObject);
procedure Pn077Click(Sender: TObject);
procedure Pn078Click(Sender: TObject);
procedure Pn079Click(Sender: TObject);
procedure Pn080Click(Sender: TObject);
procedure Pn081Click(Sender: TObject);
procedure Pn082Click(Sender: TObject);
procedure Pn083Click(Sender: TObject);
procedure Pn084Click(Sender: TObject);
procedure Pn085Click(Sender: TObject);
procedure Pn086Click(Sender: TObject);
procedure Pn087Click(Sender: TObject);
procedure Pn088Click(Sender: TObject);
procedure Pn089Click(Sender: TObject);
procedure Pn090Click(Sender: TObject);
procedure Pn091Click(Sender: TObject);
```

```
procedure Pn092Click(Sender: TObject);
procedure Pn093Click(Sender: TObject);
procedure Pn094Click(Sender: TObject);
procedure Pn095Click(Sender: TObject);
procedure Pn096Click(Sender: TObject);
procedure Pn097Click(Sender: TObject);
procedure Pn098Click(Sender: TObject);
procedure Pn099Click(Sender: TObject);
procedure Pn100Click(Sender: TObject);
procedure Pn101Click(Sender: TObject);
procedure Pn102Click(Sender: TObject);
procedure Pn103Click(Sender: TObject);
procedure Pn104Click(Sender: TObject);
procedure Pn105Click(Sender: TObject);
procedure Pn106Click(Sender: TObject);
procedure Pn107Click(Sender: TObject);
procedure Pn108Click(Sender: TObject);
procedure Pn109Click(Sender: TObject);
procedure Pn110Click(Sender: TObject);
procedure Pn111Click(Sender: TObject);
procedure Pn112Click(Sender: TObject);
procedure Pn113Click(Sender: TObject);
procedure Pn114Click(Sender: TObject);
procedure Pn115Click(Sender: TObject);
procedure Pn116Click(Sender: TObject);
procedure Pn117Click(Sender: TObject);
procedure Pn118Click(Sender: TObject);
procedure Pn119Click(Sender: TObject);
procedure Pn120Click(Sender: TObject);
procedure Pn121Click(Sender: TObject);
procedure Pn122Click(Sender: TObject);
procedure Pn123Click(Sender: TObject);
procedure Pn124Click(Sender: TObject);
procedure Pn125Click(Sender: TObject);
procedure Pn126Click(Sender: TObject);
procedure Pn127Click(Sender: TObject);
procedure Pn128Click(Sender: TObject);
procedure Pn129Click(Sender: TObject);
procedure Pn130Click(Sender: TObject);
procedure Pn131Click(Sender: TObject);
procedure Pn132Click(Sender: TObject);
procedure Pn133Click(Sender: TObject);
procedure Pn134Click(Sender: TObject);
procedure Pn135Click(Sender: TObject);
procedure Pn136Click(Sender: TObject);
procedure Pn137Click(Sender: TObject);
procedure Pn138Click(Sender: TObject);
procedure Pn139Click(Sender: TObject);
procedure Pn140Click(Sender: TObject);
procedure Pn141Click(Sender: TObject);
procedure Pn142Click(Sender: TObject);
procedure Pn143Click(Sender: TObject);
procedure Pn144Click(Sender: TObject);
procedure Pn145Click(Sender: TObject);
procedure Pn146Click(Sender: TObject);
procedure Pn147Click(Sender: TObject);
procedure Pn148Click(Sender: TObject);
procedure Pn149Click(Sender: TObject);
procedure Pn150Click(Sender: TObject);
procedure Pn151Click(Sender: TObject);
procedure Pn152Click(Sender: TObject);
procedure Pn153Click(Sender: TObject);
procedure Pn154Click(Sender: TObject);
procedure Pn155Click(Sender: TObject);
procedure Pn156Click(Sender: TObject);
procedure Pn157Click(Sender: TObject);
procedure Pn158Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;
  Pn : array[1..158] of TPanel;
  CurrNoOfKey : integer;
  CurrKey : array[1..10] of char;
implementation
uses MainExEdit;
{$R *.DFM}

//---------------------------------------------------------------------
procedure TForm2.Initial;
var  i : integer;
begin
  Pn[01] := Pn001;  Pn[02] := Pn002;  Pn[03] := Pn003;
  Pn[04] := Pn004;  Pn[05] := Pn005;  Pn[06] := Pn006;
  Pn[07] := Pn007;  Pn[08] := Pn008;  Pn[09] := Pn009;
  Pn[10] := Pn010;  Pn[11] := Pn011;  Pn[12] := Pn012;
  Pn[13] := Pn013;  Pn[14] := Pn014;  Pn[15] := Pn015;
  Pn[16] := Pn016;  Pn[17] := Pn017;  Pn[18] := Pn018;
  Pn[19] := Pn019;  Pn[20] := Pn020;  Pn[21] := Pn021;
  Pn[22] := Pn022;  Pn[23] := Pn023;  Pn[24] := Pn024;
  Pn[25] := Pn025;  Pn[26] := Pn026;  Pn[27] := Pn027;
  Pn[28] := Pn028;  Pn[29] := Pn029;  Pn[30] := Pn030;
  Pn[31] := Pn031;  Pn[32] := Pn032;  Pn[33] := Pn033;
  Pn[34] := Pn034;  Pn[35] := Pn035;  Pn[36] := Pn036;
  Pn[37] := Pn037;  Pn[38] := Pn038;  Pn[39] := Pn039;
  Pn[40] := Pn040;  Pn[41] := Pn041;  Pn[42] := Pn042;
  Pn[43] := Pn043;  Pn[44] := Pn044;  Pn[45] := Pn045;
  Pn[46] := Pn046;  Pn[47] := Pn047;  Pn[48] := Pn048;
  Pn[49] := Pn049;  Pn[50] := Pn050;  Pn[51] := Pn051;
  Pn[52] := Pn052;  Pn[53] := Pn053;  Pn[54] := Pn054;
  Pn[55] := Pn055;  Pn[56] := Pn056;  Pn[57] := Pn057;
  Pn[58] := Pn058;  Pn[59] := Pn059;  Pn[60] := Pn060;
  Pn[61] := Pn061;  Pn[62] := Pn062;  Pn[63] := Pn063;
  Pn[64] := Pn064;  Pn[65] := Pn065;  Pn[66] := Pn066;
  Pn[67] := Pn067;  Pn[68] := Pn068;  Pn[69] := Pn069;
  Pn[70] := Pn070;  Pn[71] := Pn071;  Pn[72] := Pn072;
  Pn[73] := Pn073;  Pn[74] := Pn074;  Pn[75] := Pn075;
  Pn[76] := Pn076;  Pn[77] := Pn077;  Pn[78] := Pn078;
  Pn[79] := Pn079;  Pn[80] := Pn080;  Pn[81] := Pn081;
  Pn[82] := Pn082;  Pn[83] := Pn083;  Pn[84] := Pn084;
  Pn[85] := Pn085;  Pn[86] := Pn086;  Pn[87] := Pn087;
  Pn[88] := Pn088;  Pn[89] := Pn089;  Pn[90] := Pn090;
  Pn[91] := Pn091;  Pn[92] := Pn092;  Pn[93] := Pn093;
  Pn[94] := Pn094;  Pn[95] := Pn095;  Pn[96] := Pn096;
  Pn[97] := Pn097;  Pn[98] := Pn098;  Pn[99] := Pn099;
  Pn[100] := Pn100;
 Pn[101] := Pn101;Pn[102] := Pn102;Pn[103] := Pn103;
 Pn[104] := Pn104;Pn[105] := Pn105;Pn[106] := Pn106;
 Pn[107] := Pn107;Pn[108] := Pn108;Pn[109] := Pn109;
 Pn[110] := Pn110;Pn[111] := Pn111;Pn[112] := Pn112;
 Pn[113] := Pn113;Pn[114] := Pn114;Pn[115] := Pn115;
 Pn[116] := Pn116;Pn[117] := Pn117;Pn[118] := Pn118;
 Pn[119] := Pn119;Pn[120] := Pn120;Pn[121] := Pn121;
 Pn[122] := Pn122;Pn[123] := Pn123;Pn[124] := Pn124;
 Pn[125] := Pn125;Pn[126] := Pn126;Pn[127] := Pn127;
 Pn[128] := Pn128;Pn[129] := Pn129;Pn[130] := Pn130;
 Pn[131] := Pn131;Pn[132] := Pn132;Pn[133] := Pn133;
 Pn[134] := Pn134;Pn[135] := Pn135;Pn[136] := Pn136;
 Pn[137] := Pn137;Pn[138] := Pn138;Pn[139] := Pn139;
```

```
  Pn[140] := Pn140;Pn[141] := Pn141;Pn[142] := Pn142;
  Pn[143] := Pn143;Pn[144] := Pn144;Pn[145] := Pn145;
  Pn[146] := Pn146;Pn[147] := Pn147;Pn[148] := Pn148;
  Pn[149] := Pn149;Pn[150] := Pn150;Pn[151] := Pn151;
  Pn[152] := Pn152;Pn[153] := Pn153;Pn[154] := Pn154;
  Pn[155] := Pn155;Pn[156] := Pn156;Pn[157] := Pn157;
  Pn[158] := Pn158;
    if Language = 1 then
      for i:=1 to 66 do
        Pn[i].Visible := false
    else
      for i:=67 to 158 do
        Pn[i].Visible := false;
    for i:= 67 to 158 do
      Pn[i].tag := ord(Pn[i].Caption[length(Pn[
                  i].Caption)]);
end;

//---------------------------------------------------------------
procedure TForm2.SelectBtn(Sel:byte;
                           Number:integer);
begin
    if Sel = 0 then
    begin
      Pn[Number].font.Color := ClGray;
      Pn[Number].font.Style := [fsItalic];
      Pn[Number].BevelOuter := bvLowered;
      Pn[Number].Enabled := false;
    end;
    if Sel = 1 then
    begin
      Pn[Number].font.Color := ClBlue;
      Pn[Number].font.Style := [fsBold];
      Pn[Number].BevelOuter := bvLowered;
      Pn[Number].BevelWidth := 3;
    end;
    if Sel = 2 then
    begin
      Pn[Number].font.Color := ClBlack;
      Pn[Number].font.Style := [];
      Pn[Number].BevelOuter := bvRaised;
      Pn[Number].BevelWidth := 1;
    end;
end;

//---------------------------------------------------------------
procedure TForm2.PressBtn(KeyIn:byte;Number:byte);
var i,j : integer;
begin
    if Pn[Number].BevelOuter = bvLowered then
    begin
      for i:=1 to CurrNoOfKey do
        if Chr(KeyIn)=CurrKey[i] then j:=i;
      CurrNoOfKey := CurrNoOfKey-1;
      for i:=j to CurrNoOfKey do
        CurrKey[i] := CurrKey[i+1];
      CurrKey[CurrNoOfKey+1] := ' ';
      SelectBtn(2,Number)
    end
    else
    begin
      if CurrNoOfKey <8 then
      begin
        SelectBtn(1,Number);
        CurrNoOfKey := CurrNoOfKey +1;
        CurrKey[CurrNoOfKey] := Chr(KeyIn);
      end
      else
        MessageDlg('Can not select more than 8
                    key',MtError,[mbok],1);
    end;
    NewKeyList.Caption := '';
    for i:=1 to CurrNoOfKey do
      NewKeyList.Caption := NewKeyList.Caption +
                            CurrKey[i]+'  ';
end;

//---------------------------------------------------------------
procedure TForm2.BtnOKClick(Sender: TObject);
var  i: integer;
begin
    for i:=1 to NoOfKey[Language,Lesson] do
      AsciiTable[Language,ord(Key[Language,
                 Lesson,i])] := 255;
    for i:=1 to CurrNoOfKey do
      AsciiTable[Language,ord(CurrKey[i])] := Lesson;
    NoOfKey[Language,Lesson] := CurrNoOfKey;
    for i:=1 to 8 do
      Key[Language,Lesson,i] := CurrKey[i];
    Form2.Close;
end;

//---------------------------------------------------------------
procedure TForm2.BtnCancelClick(Sender: TObject);
begin
    Form2.Close;
end;

//---------------------------------------------------------------
procedure TForm2.FormClose(Sender: TObject; var
                           Action: TCloseAction);
begin
    Form1.Enabled := true;
end;

//---------------------------------------------------------------
procedure TForm2.FormCreate(Sender: TObject);
var  i,j,k : integer;
begin
    if Language = 1 then NoOfLesson.Caption :=
                         'แบบฝึกหัดภาษาไทยชุดที่ '
    else NoOfLesson.Caption :=
                         'แบบฝึกหัดภาษาอังกฤษชุดที่ ';
    NoOfLesson.Caption := NoOfLesson.Caption +
                          inttostr(Lesson);
    OldKeyList.Caption := '';
    for i:=1 to NoOfKey[Language,Lesson] do
      OldKeyList.Caption := OldKeyList.Caption +
                            Key[Language,Lesson,i]+'  ';
    NewKeyList.Caption := OldKeyList.Caption;
    Initial;
    for i:=1 to 8 do
      CurrKey[i] := Key[Language,Lesson,i];
    CurrNoOfKey := NoOfKey[Language,Lesson];
    for i:=1 to 12 do
      for j:=1 to 255 do
      begin
        if AsciiTable[Language,j] = Lesson then
        begin
          for k:=1 to 158 do
            if Pn[k].tag = j then SelectBtn(1,k);
        end
```

```
        else
        if AsciiTable[Language,j] <>255  then
          for k:=1 to 158 do
            if Pn[k].tag = j then SelectBtn(0,k);
      end;
end;

//---------------------------------------------------------------
procedure TForm2.Pn001Click(Sender: TObject);
begin
   PressBtn(Pn001.tag,1);
end;
procedure TForm2.Pn002Click(Sender: TObject);
begin
   PressBtn(Pn002.tag,2);
end;
procedure TForm2.Pn003Click(Sender: TObject);
begin
   PressBtn(Pn003.tag,3);
end;
procedure TForm2.Pn004Click(Sender: TObject);
begin
   PressBtn(Pn004.tag,4);
end;
procedure TForm2.Pn005Click(Sender: TObject);
begin
   PressBtn(Pn005.tag,5);
end;
procedure TForm2.Pn006Click(Sender: TObject);
begin
   PressBtn(Pn006.tag,6);
end;
procedure TForm2.Pn007Click(Sender: TObject);
begin
   PressBtn(Pn007.tag,7);
end;
procedure TForm2.Pn008Click(Sender: TObject);
begin
   PressBtn(Pn008.tag,8);
end;
procedure TForm2.Pn009Click(Sender: TObject);
begin
   PressBtn(Pn009.tag,9);
end;
procedure TForm2.Pn010Click(Sender: TObject);
begin
   PressBtn(Pn010.tag,10);
end;
procedure TForm2.Pn011Click(Sender: TObject);
begin
   PressBtn(Pn011.tag,11);
end;
procedure TForm2.Pn02Click(Sender: TObject);
begin
   PressBtn(Pn012.tag,12);
end;
procedure TForm2.Pn013Click(Sender: TObject);
begin
   PressBtn(Pn013.tag,13);
end;
procedure TForm2.Pn014Click(Sender: TObject);
begin
   PressBtn(Pn014.tag,14);
end;
procedure TForm2.Pn015Click(Sender: TObject);
begin
   PressBtn(Pn015.tag,15);
end;
procedure TForm2.Pn016Click(Sender: TObject);
begin
   PressBtn(Pn016.tag,16);
end;
procedure TForm2.Pn017Click(Sender: TObject);
begin
   PressBtn(Pn017.tag,17);
end;
procedure TForm2.Pn018Click(Sender: TObject);
begin
   PressBtn(Pn018.tag,18);
end;
procedure TForm2.Pn019Click(Sender: TObject);
begin
   PressBtn(Pn019.tag,19);
end;
procedure TForm2.Pn020Click(Sender: TObject);
begin
   PressBtn(Pn020.tag,20);
end;
procedure TForm2.Pn021Click(Sender: TObject);
begin
   PressBtn(Pn021.tag,21);
end;
procedure TForm2.Pn022Click(Sender: TObject);
begin
   PressBtn(Pn022.tag,22);
end;
procedure TForm2.Pn023Click(Sender: TObject);
begin
   PressBtn(Pn023.tag,23);
end;
procedure TForm2.Pn024Click(Sender: TObject);
begin
   PressBtn(Pn024.tag,24);
end;
procedure TForm2.Pn025Click(Sender: TObject);
begin
   PressBtn(Pn025.tag,25);
end;
procedure TForm2.Pn026Click(Sender: TObject);
begin
   PressBtn(Pn026.tag,26);
end;
procedure TForm2.Pn027Click(Sender: TObject);
begin
   PressBtn(Pn027.tag,27);
end;
procedure TForm2.Pn028Click(Sender: TObject);
begin
   PressBtn(Pn028.tag,28);
end;
procedure TForm2.Pn029Click(Sender: TObject);
begin
   PressBtn(Pn029.tag,29);
end;
procedure TForm2.Pn030Click(Sender: TObject);
begin
   PressBtn(Pn030.tag,30);
end;
procedure TForm2.Pn031Click(Sender: TObject);
begin
   PressBtn(Pn031.tag,31);
end;
procedure TForm2.Pn032Click(Sender: TObject);
begin
```

```
    PressBtn(Pn032.tag,32);
end;
procedure TForm2.Pn033Click(Sender: TObject);
begin
    PressBtn(Pn033.tag,33);
end;
procedure TForm2.Pn034Click(Sender: TObject);
begin
    PressBtn(Pn034.tag,34);
end;
procedure TForm2.Pn035Click(Sender: TObject);
begin
    PressBtn(Pn035.tag,35);
end;
procedure TForm2.Pn036Click(Sender: TObject);
begin
    PressBtn(Pn036.tag,36);
end;
procedure TForm2.Pn037Click(Sender: TObject);
begin
    PressBtn(Pn037.tag,37);
end;
procedure TForm2.Pn038Click(Sender: TObject);
begin
    PressBtn(Pn038.tag,38);
end;
procedure TForm2.Pn039Click(Sender: TObject);
begin
    PressBtn(Pn039.tag,39);
end;
procedure TForm2.Pn040Click(Sender: TObject);
begin
    PressBtn(Pn040.tag,40);
end;
procedure TForm2.Pn041Click(Sender: TObject);
begin
    PressBtn(Pn041.tag,41);
end;
procedure TForm2.Pn042Click(Sender: TObject);
begin
    PressBtn(Pn042.tag,42);
end;
procedure TForm2.Pn043Click(Sender: TObject);
begin
    PressBtn(Pn043.tag,43);
end;
procedure TForm2.Pn044Click(Sender: TObject);
begin
    PressBtn(Pn044.tag,44);
end;
procedure TForm2.Pn045Click(Sender: TObject);
begin
    PressBtn(Pn045.tag,45);
end;
procedure TForm2.Pn046Click(Sender: TObject);
begin
    PressBtn(Pn046.tag,46);
end;
procedure TForm2.Pn047Click(Sender: TObject);
begin
    PressBtn(Pn047.tag,47);
end;
procedure TForm2.Pn048Click(Sender: TObject);
begin
    PressBtn(Pn048.tag,48);
end;
procedure TForm2.Pn049Click(Sender: TObject);
begin
    PressBtn(Pn049.tag,49);
end;
procedure TForm2.Pn050Click(Sender: TObject);
begin
    PressBtn(Pn050.tag,50);
end;
procedure TForm2.Pn051Click(Sender: TObject);
begin
    PressBtn(Pn051.tag,51);
end;
procedure TForm2.Pn052Click(Sender: TObject);
begin
    PressBtn(Pn052.tag,52);
end;
procedure TForm2.Pn053Click(Sender: TObject);
begin
    PressBtn(Pn053.tag,53);
end;
procedure TForm2.Pn054Click(Sender: TObject);
begin
    PressBtn(Pn054.tag,54);
end;
procedure TForm2.Pn055Click(Sender: TObject);
begin
    PressBtn(Pn055.tag,55);
end;
procedure TForm2.Pn056Click(Sender: TObject);
begin
    PressBtn(Pn056.tag,56);
end;
procedure TForm2.Pn057Click(Sender: TObject);
begin
    PressBtn(Pn057.tag,57);
end;
procedure TForm2.Pn058Click(Sender: TObject);
begin
    PressBtn(Pn058.tag,58);
end;
procedure TForm2.Pn059Click(Sender: TObject);
begin
    PressBtn(Pn059.tag,59);
end;
procedure TForm2.Pn060Click(Sender: TObject);
begin
    PressBtn(Pn060.tag,60);
end;
procedure TForm2.Pn061Click(Sender: TObject);
begin
    PressBtn(Pn061.tag,61);
end;
procedure TForm2.Pn062Click(Sender: TObject);
begin
    PressBtn(Pn062.tag,62);
end;
procedure TForm2.Pn063Click(Sender: TObject);
begin
    PressBtn(Pn063.tag,63);
end;
procedure TForm2.Pn064Click(Sender: TObject);
begin
    PressBtn(Pn064.tag,64);
end;
procedure TForm2.Pn065Click(Sender: TObject);
begin
    PressBtn(Pn065.tag,65);
end;
```

```
procedure TForm2.Pn066Click(Sender: TObject);
begin
   PressBtn(Pn066.tag,66);
end;
procedure TForm2.Pn067Click(Sender: TObject);
begin
   PressBtn(Pn067.tag,67);
end;
procedure TForm2.Pn068Click(Sender: TObject);
begin
   PressBtn(Pn068.tag,68);
end;
procedure TForm2.Pn069Click(Sender: TObject);
begin
   PressBtn(Pn069.tag,69);
end;
procedure TForm2.Pn070Click(Sender: TObject);
begin
   PressBtn(Pn070.tag,70);
end;
procedure TForm2.Pn071Click(Sender: TObject);
begin
   PressBtn(Pn071.tag,71);
end;
procedure TForm2.Pn072Click(Sender: TObject);
begin
   PressBtn(Pn072.tag,72);
end;
procedure TForm2.Pn073Click(Sender: TObject);
begin
   PressBtn(Pn073.tag,73);
end;
procedure TForm2.Pn074Click(Sender: TObject);
begin
   PressBtn(Pn074.tag,74);
end;
procedure TForm2.Pn075Click(Sender: TObject);
begin
   PressBtn(Pn075.tag,75);
end;
procedure TForm2.Pn076Click(Sender: TObject);
begin
   PressBtn(Pn076.tag,76);
end;
procedure TForm2.Pn077Click(Sender: TObject);
begin
   PressBtn(Pn077.tag,77);
end;
procedure TForm2.Pn078Click(Sender: TObject);
begin
   PressBtn(Pn078.tag,78);
end;
procedure TForm2.Pn079Click(Sender: TObject);
begin
   PressBtn(Pn079.tag,79);
end;
procedure TForm2.Pn080Click(Sender: TObject);
begin
   PressBtn(Pn080.tag,80);
end;
procedure TForm2.Pn081Click(Sender: TObject);
begin
   PressBtn(Pn081.tag,81);
end;
procedure TForm2.Pn082Click(Sender: TObject);
begin
   PressBtn(Pn082.tag,82);
end;
procedure TForm2.Pn083Click(Sender: TObject);
begin
   PressBtn(Pn083.tag,83);
end;
procedure TForm2.Pn084Click(Sender: TObject);
begin
   PressBtn(Pn084.tag,84);
end;
procedure TForm2.Pn085Click(Sender: TObject);
begin
   PressBtn(Pn085.tag,85);
end;
procedure TForm2.Pn086Click(Sender: TObject);
begin
   PressBtn(Pn086.tag,86);
end;
procedure TForm2.Pn087Click(Sender: TObject);
begin
   PressBtn(Pn087.tag,87);
end;
procedure TForm2.Pn088Click(Sender: TObject);
begin
   PressBtn(Pn088.tag,88);
end;
procedure TForm2.Pn089Click(Sender: TObject);
begin
   PressBtn(Pn089.tag,89);
end;
procedure TForm2.Pn090Click(Sender: TObject);
begin
   PressBtn(Pn090.tag,90);
end;
procedure TForm2.Pn091Click(Sender: TObject);
begin
   PressBtn(Pn091.tag,91);
end;
procedure TForm2.Pn092Click(Sender: TObject);
begin
   PressBtn(Pn092.tag,92);
end;
procedure TForm2.Pn093Click(Sender: TObject);
begin
   PressBtn(Pn093.tag,93);
end;
procedure TForm2.Pn094Click(Sender: TObject);
begin
   PressBtn(Pn094.tag,94);
end;
procedure TForm2.Pn095Click(Sender: TObject);
begin
   PressBtn(Pn095.tag,95);
end;
procedure TForm2.Pn096Click(Sender: TObject);
begin
   PressBtn(Pn096.tag,96);
end;
procedure TForm2.Pn097Click(Sender: TObject);
begin
   PressBtn(Pn097.tag,97);
end;
procedure TForm2.Pn098Click(Sender: TObject);
begin
   PressBtn(Pn098.tag,98);
end;
procedure TForm2.Pn099Click(Sender: TObject);
begin
```

```
    PressBtn(Pn099.tag,99);
end;
procedure TForm2.Pn100Click(Sender: TObject);
begin
    PressBtn(Pn100.tag,100);
end;
procedure TForm2.Pn101Click(Sender: TObject);
begin
    PressBtn(Pn101.tag,101);
end;
procedure TForm2.Pn102Click(Sender: TObject);
begin
    PressBtn(Pn102.tag,102);
end;
procedure TForm2.Pn103Click(Sender: TObject);
begin
    PressBtn(Pn103.tag,103);
end;
procedure TForm2.Pn104Click(Sender: TObject);
begin
    PressBtn(Pn104.tag,104);
end;
procedure TForm2.Pn105Click(Sender: TObject);
begin
    PressBtn(Pn105.tag,105);
end;
procedure TForm2.Pn106Click(Sender: TObject);
begin
    PressBtn(Pn106.tag,106);
end;
procedure TForm2.Pn107Click(Sender: TObject);
begin
    PressBtn(Pn107.tag,107);
end;
procedure TForm2.Pn108Click(Sender: TObject);
begin
    PressBtn(Pn108.tag,108);
end;
procedure TForm2.Pn109Click(Sender: TObject);
begin
    PressBtn(Pn109.tag,109);
end;
procedure TForm2.Pn110Click(Sender: TObject);
begin
    PressBtn(Pn110.tag,110);
end;
procedure TForm2.Pn111Click(Sender: TObject);
begin
    PressBtn(Pn111.tag,111);
end;
procedure TForm2.Pn112Click(Sender: TObject);
begin
    PressBtn(Pn112.tag,112);
end;
procedure TForm2.Pn113Click(Sender: TObject);
begin
    PressBtn(Pn113.tag,113);
end;
procedure TForm2.Pn114Click(Sender: TObject);
begin
    PressBtn(Pn114.tag,114);
end;
procedure TForm2.Pn115Click(Sender: TObject);
begin
    PressBtn(Pn115.tag,115);
end;
procedure TForm2.Pn116Click(Sender: TObject);
begin
    PressBtn(Pn116.tag,116);
end;
procedure TForm2.Pn117Click(Sender: TObject);
begin
    PressBtn(Pn117.tag,117);
end;
procedure TForm2.Pn118Click(Sender: TObject);
begin
    PressBtn(Pn118.tag,118);
end;
procedure TForm2.Pn119Click(Sender: TObject);
begin
    PressBtn(Pn119.tag,119);
end;
procedure TForm2.Pn120Click(Sender: TObject);
begin
    PressBtn(Pn120.tag,120);
end;
procedure TForm2.Pn121Click(Sender: TObject);
begin
    PressBtn(Pn121.tag,121);
end;
procedure TForm2.Pn122Click(Sender: TObject);
begin
    PressBtn(Pn122.tag,122);
end;
procedure TForm2.Pn123Click(Sender: TObject);
begin
    PressBtn(Pn123.tag,123);
end;
procedure TForm2.Pn124Click(Sender: TObject);
begin
    PressBtn(Pn124.tag,124);
end;
procedure TForm2.Pn125Click(Sender: TObject);
begin
    PressBtn(Pn125.tag,125);
end;
procedure TForm2.Pn126Click(Sender: TObject);
begin
    PressBtn(Pn126.tag,126);
end;
procedure TForm2.Pn127Click(Sender: TObject);
begin
    PressBtn(Pn127.tag,127);
end;
procedure TForm2.Pn128Click(Sender: TObject);
begin
    PressBtn(Pn128.tag,128);
end;
procedure TForm2.Pn129Click(Sender: TObject);
begin
    PressBtn(Pn129.tag,129);
end;
procedure TForm2.Pn130Click(Sender: TObject);
begin
    PressBtn(Pn130.tag,130);
end;
procedure TForm2.Pn131Click(Sender: TObject);
begin
    PressBtn(Pn131.tag,131);
end;
procedure TForm2.Pn132Click(Sender: TObject);
begin
    PressBtn(Pn132.tag,132);
end;
```

```
procedure TForm2.Pn133Click(Sender: TObject);
begin
   PressBtn(Pn133.tag,133);
end;
procedure TForm2.Pn134Click(Sender: TObject);
begin
   PressBtn(Pn134.tag,134);
end;
procedure TForm2.Pn135Click(Sender: TObject);
begin
   PressBtn(Pn135.tag,135);
end;
procedure TForm2.Pn136Click(Sender: TObject);
begin
   PressBtn(Pn136.tag,136);
end;
procedure TForm2.Pn137Click(Sender: TObject);
begin
   PressBtn(Pn137.tag,137);
end;
procedure TForm2.Pn138Click(Sender: TObject);
begin
   PressBtn(Pn138.tag,138);
end;
procedure TForm2.Pn139Click(Sender: TObject);
begin
   PressBtn(Pn139.tag,139);
end;
procedure TForm2.Pn140Click(Sender: TObject);
begin
   PressBtn(Pn140.tag,140);
end;
procedure TForm2.Pn141Click(Sender: TObject);
begin
   PressBtn(Pn141.tag,141);
end;
procedure TForm2.Pn142Click(Sender: TObject);
begin
   PressBtn(Pn142.tag,142);
end;
procedure TForm2.Pn143Click(Sender: TObject);
begin
   PressBtn(Pn143.tag,143);
end;
procedure TForm2.Pn144Click(Sender: TObject);
begin
   PressBtn(Pn144.tag,144);
end;
procedure TForm2.Pn145Click(Sender: TObject);
begin
   PressBtn(Pn145.tag,145);
end;
procedure TForm2.Pn146Click(Sender: TObject);
begin
   PressBtn(Pn146.tag,146);
end;
procedure TForm2.Pn147Click(Sender: TObject);
begin
   PressBtn(Pn147.tag,147);
end;
procedure TForm2.Pn148Click(Sender: TObject);
begin
   PressBtn(Pn148.tag,148);
end;
procedure TForm2.Pn149Click(Sender: TObject);
begin
   PressBtn(Pn149.tag,149);
end;
procedure TForm2.Pn150Click(Sender: TObject);
begin
   PressBtn(Pn150.tag,150);
end;
procedure TForm2.Pn151Click(Sender: TObject);
begin
   PressBtn(Pn151.tag,151);
end;
procedure TForm2.Pn152Click(Sender: TObject);
begin
   PressBtn(Pn152.tag,152);
end;
procedure TForm2.Pn153Click(Sender: TObject);
begin
   PressBtn(Pn153.tag,153);
end;
procedure TForm2.Pn154Click(Sender: TObject);
begin
   PressBtn(Pn154.tag,154);
end;
procedure TForm2.Pn155Click(Sender: TObject);
begin
   PressBtn(Pn155.tag,155);
end;
procedure TForm2.Pn156Click(Sender: TObject);
begin
   PressBtn(Pn156.tag,156);
end;
procedure TForm2.Pn157Click(Sender: TObject);
begin
   PressBtn(Pn157.tag,157);
end;
procedure TForm2.Pn158Click(Sender: TObject);
begin
   PressBtn(Pn158.tag,158);
end;
end.

//=============================================
unit SelFile;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,  StdCtrls;
type
  TForm3 = class(TForm)
    FilenameEdit: TEdit;
    Title    : TLabel;
    BtnOk    : TButton;
    BtnCancel : TButton;
    BtnBrowse : TButton;
    BrowseDialog : TOpenDialog;
    procedure BtnBrowseClick(Sender: TObject);
    procedure BtnCancelClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BtnOkClick(Sender: TObject);
    procedure FormClose(Sender: TObject;
        var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form3: TForm3;
  Select : Boolean;
```

```
  ThaiFile,EngFile : string;
 implementation
 uses MainExEdit;
 {$R *.DFM}

 //-------------------------------------------------------------------
 procedure TForm3.BtnBrowseClick(Sender: TObject);
 begin
    BrowseDialog.execute;
    FileNameEdit.text := BrowseDialog.Filename;
 end;

 //-------------------------------------------------------------------
 procedure TForm3.BtnCancelClick(Sender: TObject);
 begin
    Form3.Close;
 end;

 //-------------------------------------------------------------------
 procedure TForm3.FormCreate(Sender: TObject);
 begin
    Form3.Left := Form1.Left+70;
    Form3.Top := Form1.Top+100;
    Select := False;
 end;

 //-------------------------------------------------------------------
 procedure TForm3.BtnOkClick(Sender: TObject);
 begin
    if Select = False then
    begin
      Select := True;
      ThaiFile := FileNameEdit.Text;
      if not(FileExists(ThaiFile)) then
      begin
        Select := False;
        messagedlg('File not exists',mterror,[mbok],0);
      end
      else
      Title.Caption := 'กรุณาใส่ชื่อไฟล์ของคำภาษาอังกฤษ';
      FileNameEdit.Text := '';
    end
    else
    begin
      EngFile := FileNameEdit.Text;
      if not(FileExists(EngFile)) then
        messagedlg('File not exists',mterror,[mbok],0)
      else
      begin
        Form1.NewFile;
        Form3.Close;
      end;
    end;
 end;

 //-------------------------------------------------------------------
 procedure TForm3.FormClose(Sender: TObject;
                                  var Action: TCloseAction);
 begin
    Form1.Enabled := True;
 end;
 end.

 //===================================================================
 unit ExProg;
 interface
```

```
uses
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Gauges, Wsc;
type
  TForm4 = class(TForm)
    SelectCOM: TRadioGroup;
    SelectProg: TGroupBox;
    CheckLesson: TCheckBox;
    CheckEng: TCheckBox;
    CheckThai: TCheckBox;
    BtnOk: TButton;
    BtnExit: TButton;
    Gauge1: TGauge;
    Gauge2: TGauge;
    Gauge3: TGauge;
    LabelProgress1: TLabel;
    LabelProgress2: TLabel;
    LabelProgress3: TLabel;
    function OpenComPort(P: integer): integer;
    procedure CloseComPort;
    procedure LessonPGM;
    procedure SoundProgram(FName : string; lang : byte;
Gau : TGauge);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure BtnCancelClick(Sender: TObject);
    procedure BtnExitClick(Sender: TObject);
    procedure BtnOkClick(Sender: TObject);
  private
    { Private declarations }
    Port : Integer;
    Baud : Integer;
    Parity : Integer;
    DataBits : Integer;
    StopBits : Integer;
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
  BaseKey : array[1..2,1..256] of char =
    (('','','','','','','','','','','','','','','','',
     '','','','','','','','','','','','','','','','',
     '','ห','','','','','','ฟ','ห','','','','','ห','','ฟ','',
     '','','','','','','','','','','','','','','่','',
     '','','','','','','','','','','','','','','','',
     '','','','','','','','','','','','','','','','',
     '','','','','','','','','','','','','','','','',
     '','','','','','','','','','','','','','','','',
     '','','','','','','','','','','','','','','','',
     '','','','','','','','','','','','','','','','',
     'ก','ว','ว','่','ว','ห','ว','ส','ก','ว','ว','ด','ว','ก','ก','ว',
     'ด','า','า','ด','า','ด','่','ด','ส','ว','ห','ฟ','ว','ด','ฟ','ก',
     'า','ว','า','ฟ','ว','ว','ว','ส','า','ส','ห','ส','ด','ด','ส','ด',
     '่','า','ก','ด','่','่','่','ด','ด','','','','','','่','ด',
     'ก','ด','ส','ห','','ฟ','่','่','่','่','่','','','','','',
     '','','','','','','','','','','','','','','','' ),
```

```
  ( ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    'A',';','D','F','F','J',';','L',';','K',';','K',';','L',';','L',
    'A','S','D','F','F','F','J','K','L',';',';','K',';','L',';',' ',
    'A','F','D','D','D','F','F','J','K','J','K','L','J','J','L',';',
    'A','F','S','F','J','F','S','S','J','A',';',';',';','F',';',' ',
    'A','F','D','D','D','F','F','J','K','J','K','L','J','J','L',';',
    'A','F','S','F','J','F','S','S','J','A',';',';',';','A',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
    ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ' ));
MapASCII : array[1..2,1..256] of byte =
  (( 000,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,195,255,255,255,255,255,217,218,255,
     255,204,113,216,112,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,222,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     137,121,234,118,235,207,145,120,219,122,
     215,210,202,196,208,203,
     197,223,200,138,119,115,153,198,131,133,
     148,147,156,126,135,114,
     154,132,130,206,134,225,144,214,213,143,
     136,224,150,220,201,127,
     128,142,125,151,129,117,152,116,187,255,
     255,255,255,255,188,139,
     149,209,155,124,255,123,211,141,140,199,
     212,221,255,255,255,194,
     183,184,185,186,189,190,191,192,193,255,
     255,116,187,255,255,255),
   ( 000,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,050,010,051,052,009,
     000,001,002,003,004,005,006,007,008,106,
     037,255,255,255,111,255,
     028,047,045,030,016,031,032,033,021,034,
     035,036,049,048,022,023,
     014,017,029,018,020,046,015,044,019,043,
     255,255,255,255,255,255,
     028,047,045,030,016,031,032,033,021,034,
     035,036,049,048,022,023,
     014,017,029,018,020,046,015,044,019,043,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255,
     255,255,255,255,255,255,255,255,255,255,
     255,255,255,255,255,255));
implementation
uses MainExEdit, SelFile;
{$R *.DFM}

//---------------------------------------------------------------
procedure TForm4.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
  Form1.Enabled := true;
end;

//---------------------------------------------------------------
procedure TForm4.BtnCancelClick(Sender: TObject);
begin
  Form4.Close;
end;

//---------------------------------------------------------------
procedure TForm4.BtnExitClick(Sender: TObject);
begin
  Form4.Close;
end;

//---------------------------------------------------------------
procedure TForm4.BtnOkClick(Sender: TObject);
var Code,P : integer;
  FName : string;
begin
  P := COM2;
  if SelectCOM.ItemIndex = 0 then P:=COM1
    else if SelectCOM.ItemIndex = 1 then P:=COM2
      else if SelectCOM.ItemIndex = 2 then P:=COM3
        else if SelectCOM.ItemIndex = 3 then
          P:=COM4;
  Code := OpenComPort(P);
  if Code <> 0 then
  begin
    MessageDlg('Error '+inttostr(Code)+' cannot start
              communication',MtError,[mbOk],1);
    exit;
  end;
  if CheckLesson.State <> cbChecked then
    Gauge1.Progress := Gauge1.MaxValue
      else Gauge1.Progress := 0;
  if CheckEng.State <> cbChecked then
    Gauge2.Progress := Gauge2.MaxValue
      else Gauge2.Progress := 0;
  if CheckThai.State <> cbChecked then
    Gauge3.Progress := Gauge3.MaxValue
      else Gauge3.Progress := 0;
  SelectCOM.Enabled := false;
  SelectProg.Enabled := false;
  CheckLesson.Enabled := False;
```

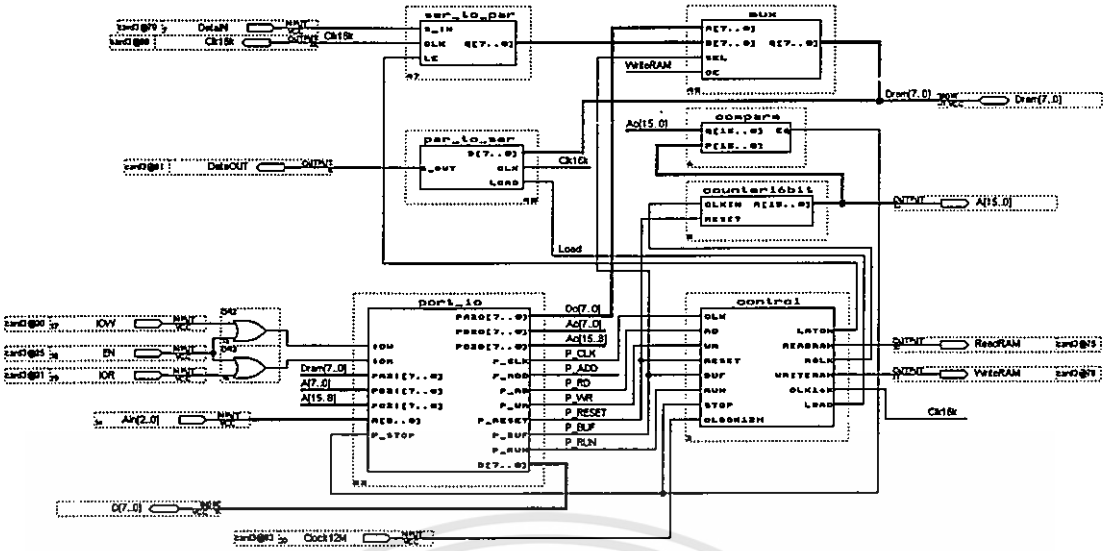
```
    CheckEng.Enabled := False;
    CheckThai.Enabled := False;
    BtnOk.Enabled := false;
    BtnExit.Enabled := false;
    if Gauge1.Progress <> Gauge1.MaxValue then
       LessonPGM;
    if Gauge2.Progress <> Gauge2.MaxValue then
    begin
      FName := EngFile;
      FName[Length(FName)-2]:= 'b';
      FName[Length(FName)-1]:= 'i';
      FName[Length(FName)-0]:= 'n';
      if FileExists(FName) then
        SoundProgram(FName,$10,Gauge2)
          else messagedlg('ไม่พบไฟล์ชื่อ "'+
                          FName+'"',mterror,[mbOk],1);
    end;
    if Gauge3.Progress <> Gauge3.MaxValue then
    begin
      FName := ThaiFile;
      FName[Length(FName)-2]:= 'b';
      FName[Length(FName)-1]:= 'i';
      FName[Length(FName)-0]:= 'n';
      if FileExists(FName) then
        SoundProgram(FName,$18,Gauge3)
          else messagedlg('ไม่พบไฟล์ชื่อ "'+
                          FName+'"',mterror,[mbOk],1);
    end;
    CloseComPort;
    SelectCOM.Enabled := true;
    SelectProg.Enabled := true;
    CheckLesson.Enabled := true;
    CheckEng.Enabled := true;
    CheckThai.Enabled := true;
    BtnOk.Enabled := true;
    BtnExit.Enabled := true;
end;

//-------------------------------------------------------------
function TForm4.OpenComPort(P : integer):integer;
var  Code : Integer;
begin
    Port := P;
    Baud := Baud9600;
    Parity := NoParity;
    DataBits := WordLength8;
    StopBits := OneStopBit;
    Code := SioReset(Port,2048,512);
    if Code <> 0 then
    begin
      OpenComPort := Code;
      exit;
    end;
    SioBaud(Port,Baud);
    SioParms(Port, Parity, StopBits, DataBits);
    SioDTR(Port,'S');
    SioRTS(Port,'S');
    OpenComPort := 0;
end;

//-------------------------------------------------------------
procedure TForm4.CloseComPort;
begin
    SioDone(Port);
end;

//-------------------------------------------------------------
procedure TForm4.SoundProgram(FName : string; lang
: byte; Gau : TGauge);
var  LoadFile : File of byte;
    i,j,k,Code : integer;
    Size : Longint;
    buf : array [1..256] of byte;
    Start,Now : longint;
begin
    AssignFile(LoadFile,FName);
    Reset(LoadFile);
    Size := filesize(LoadFile);
    i := 0;
    Gau.Progress := 0;
    Gau.MaxValue := Size div 256 + 1;
    while (i*256) < Size do
    begin
      j := 0;
      repeat
        read(LoadFile,buf[j+1]);
        j := j+1;
      until eof(LoadFile) or (j=256) ;
      SioPutc(Port,chr(55));
      Start := SioTimer;
      repeat
        Code := SioGetc(Port);
        Now := SioTimer;
        if (Now-Start) > 2000 then
        begin
          messagedlg('ไม่สามารถติดต่อกับ Port ได้',mterror,
                     [mbok],1);
          exit;
        end;
      until code = 55;
      SioPutc(Port,chr(lang or ((i and $FF00) shr 8) ) );
      SioPutc(Port,chr(i and $FF) );       // send address
      SioPutc(Port,chr(0));
      SioPutc(Port,chr(0));                // send start byte
      SioPutc(Port,chr(j-1));              // send end byte
      for k:=1 to j do
      begin
        SioPutc(Port,chr(buf[k]));
      end;
      Start := SioTimer;
      repeat
        Code := SioGetc(Port);
        Now := SioTimer;
        if (Now-Start) > 2000 then
        begin
          messagedlg('ไม่สามารถติดต่อกับ Port ได้',mterror,
                     [mbok],1);
          exit;
        end;
      until code = (j-1);
      i := i+1;
      Gau.Progress := i;
    end;
end;

//-------------------------------------------------------------
procedure TForm4.LessonPGM;
var  i,j,k,l,loop,sum : integer;
    Address : longint;
    buf : array[1..256] of char;
    Code : integer;
    Start,Now : longint;
```

```
begin
  for i:=1 to 2 do          // Program Lesson 1-12
  for j:=1 to 12 do
  if NoOfKey[i,j]<>0 then
  begin
    Address := $4C00 - i*$1000 + (j-1)*$10;
    SioPutc(Port,chr(55));
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',mterror,
                   [mbok],1);
        exit;
      end;
    until code = 55;
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((Address and $FF00) shr 8));
                                        // send address
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((Address and $00FF)));
                                        // send start byte
    SioPutc(Port,chr((Address and $00FF) +
            NoOfKey[i,j]*2 - 1));       // send end byte
    for k:=NoOfkey[i,j] downto 1 do
    begin
      SioPutc(Port,chr(MapASCII[i,ord(BaseKey[i,
                                ord(Key[i,j,k])])]));
      SioPutc(Port,chr(MapASCII[i,ord(Key[i,j,k])]));
    end;
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้ '+
                   inttostr(Code),mterror,[mbok],1);
        exit;
      end;
    until code = (NoOfKey[i,j]*2 - 1);
    Gauge1.Progress := (i-1)*12 + j;
  end;
  for i:=1 to 2 do          // Program Number of Key
  begin
    Address := $4E00 - i*$1000;
    SioPutc(Port,chr(55));
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',mterror,
                   [mbok],1);
        exit;
      end;
    until code = 55;
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((Address and $FF00) shr 8));
                                        // send address
    SioPutc(Port,chr($00));
    SioPutc(Port,chr($00));             // send start byte
    SioPutc(Port,chr($0B));             // send end byte
    for j:=1 to 12 do
      SioPutc(Port,chr(NoOfKey[i,j]));
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',mterror,
                   [mbok],1);
        exit;
      end;
    until code = $0B;
    Gauge1.Progress := 24+i;
  end;
  for i:=1 to 2 do          // Program Sum Number of Key
  begin
    Address := $4E10 - i*$1000;
    SioPutc(Port,chr(55));
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',mterror,
                   [mbok],1);
        exit;
      end;
    until code = 55;
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((Address and $FF00) shr 8));
                                        // send address
    SioPutc(Port,chr($00));
    SioPutc(Port,chr($10));             // send start byte
    SioPutc(Port,chr($1B));             // send end byte
    sum := 0;
    for j:=1 to 12 do
    begin
      sum := sum + NoOfKey[i,j];
      SioPutc(Port,chr(sum));
    end;
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',mterror,
                   [mbok],1);
        exit;
      end;
    until code = $0B;
    Gauge1.Progress := 26+i;
  end;
  for i:=1 to 2 do                  // program random key
  begin
    Gauge1.Progress := 28+i;
    Address := $4D00 - i*$1000;
    for j:=1 to 256 do
      buf[j] := chr($01);
    sum := 0;
    for j:=1 to 12 do
      for k:=NoOfKey[i,j] downto 1 do
```

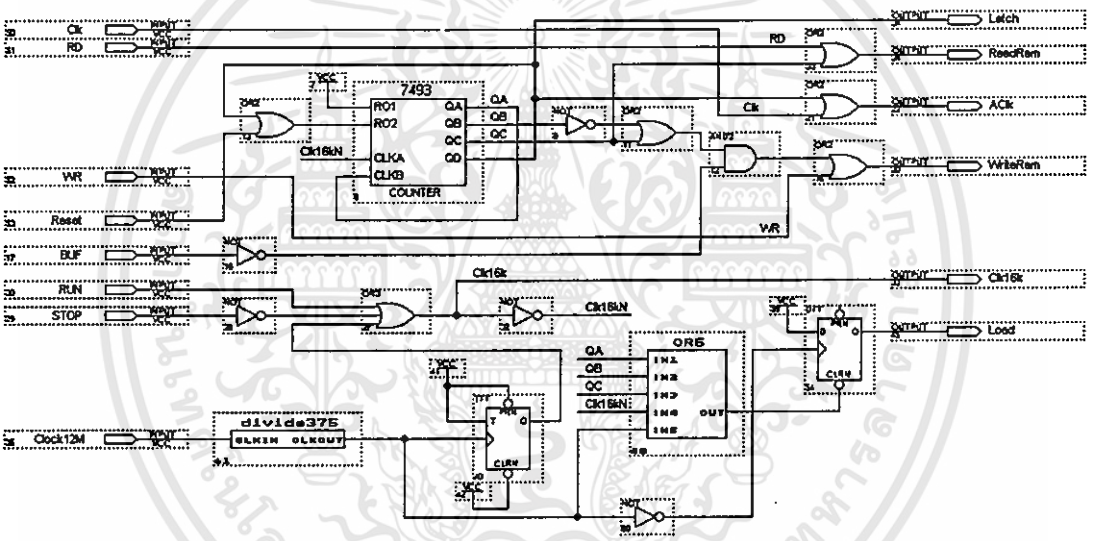
```
    begin
      sum:=sum+1;
      buf[sum] := Key[i,j,k]
    end;
  loop := sum div 64;
  if sum > loop*64 then loop := loop +1;
  for j:=1 to loop do
  begin
    SioPutc(Port,chr(55));
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',
                   mterror,[mbok],1);
        exit;
      end;
    until code = 55;
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((Address and $FF00) shr 8));
                                    // send address
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((j-1)*64));      // send start byte
    SioPutc(Port,chr((j-1)*64 + 63));
                                    // send end byte
    for k:=1 to 64 do
      SioPutc(Port,chr(MapASCII[i,ord(buf[(j-1)*
                                64 +k])]));
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',
                   mterror,[mbok],1);
        exit;
      end;
    until code = 63;
    for k:=1 to 15 do      // delay
      CheckLesson.Caption := 'แบบฝึกหัด';
  end;
end;
Gauge1.Progress := 31;
for i:=1 to 2 do          // Program Word Lesson 1-12
for j:=1 to 12 do
begin
  for l:=1 to 256 do buf[l] := chr($00);
  Gauge1.Progress := 30+((i-1)*24)+j*2;
  if NoOfWord[i,j]<>0 then
  begin
    sum := 1;
    for k:=1 to NoOfWord[i,j] do
    begin
      sum := sum+1;
      buf[sum] := chr(WordNo[i,j,k]);
      for l:=1 to length(Word[i,j,k]) do
      begin
        sum:=sum+1;
        buf[sum] := Word[i,j,k][l];
      end;
      sum := sum+1;
      buf[sum] := chr($FF);
    end;
    sum := sum+1;
    buf[sum] := chr($FF);
    buf[1] := chr(sum-NoOfWord[i,j]*2-2);
    loop := sum div 64;
    if sum > loop*64 then loop := loop +1;
  end
  else loop := 1;
  Address := $4000 - i*$1000 + (j-1)*$100;
  for l := 1 to loop do
  begin
    SioPutc(Port,chr(55));
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',
                   mterror,[mbok],1);
        exit;
      end;
    until code = 55;
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((Address and $FF00) shr 8));
                                    // send address
    SioPutc(Port,chr($00));
    SioPutc(Port,chr((l-1)*64));      // send start byte
    SioPutc(Port,chr((l-1)*64 +63));
                                    // send end byte
    for k:=1 to 64 do
      SioPutc(Port,buf[(l-1)*64+k]);
    Start := SioTimer;
    repeat
      Code := SioGetc(Port);
      Now := SioTimer;
      if (Now-Start) > 2000 then
      begin
        messagedlg('ไม่สามารถติดต่อกับ Port ได้',
                   mterror,[mbok],1);
        exit;
      end;
    until code = 63;
    for k:=1 to 15 do      // delay
      CheckLesson.Caption := 'แบบฝึกหัด';
  end;
end;

end.
```

ภาคผนวก จ.

วงจรควบคุมของการ์ดบันทึกเสียงที่เขียนบน Max+Plus II



รูปที่ จ.1 โครงสร้างภายในชิพ FPGA ที่ออกแบบ



รูปที่ จ.2 โครงสร้างในส่วนบล็อก CONTROL

ส่วนในบล็อกส่วนอื่นๆภายใน FPGA นี้เขียนด้วยภาษา AHDL ดังนี้

```
SUBDESIGN SER_TO_PAR
(
        Q[7..0] : output;
        S_IN,CLK,LE : input;
)
VARIABLE
        in_ff[7..0] : DFF;
        out_ff[7..0] : DFF;
BEGIN
        in_ff[].clk = CLK;
        in_ff[7..0].d = ( in_ff[6..0].q ,S_IN );
        out_ff[].clk = LE;
        out_ff[].d = in_ff[].q;
        Q[] = out_ff[].q;
END;
```

```
SUBDESIGN PAR_TO_SER
(
        S_OUT : output;
        D[7..0],CLK,LOAD : input;
)
VARIABLE
        in_ff[7..0] : DFF;
        out_ff : DFF;
BEGIN
        in_ff[].clk = CLK;
        IF LOAD == GND THEN
           in_ff[].prn = !D[];
           in_ff[].clrn = D[];
        ELSE
           in_ff[].prn = VCC;
```

```
        in_ff[].clrn = VCC;
    END IF;
    in_ff[7..0].d = ( in_ff[6..0].q ,GND );
    out_ff.clk = CLK;
    out_ff.d = in_ff[7].q;
    S_OUT = out_ff.q;
END;

SUBDESIGN PORT_IO
(
    PA2O[7..0], PB2O[7..0], PC2O[7..0] : output;
    P_CLK, P_ADD, P_RD, P_WR : output;
    P_RESET, P_BUF, P_RUN : output;
    D[7..0] : BIDIR;
    IOW, IOR : input;
    PA2I[7..0], PB2I[7..0], PC2I[7..0] : input;
    A[2..0], P_STOP : input;
)
VARIABLE
    D_TRI[7..0] : TRI;
    PA[7..0], PB[7..0], PC[7..0] : DFF;
    P_CON[6..0] : DFF;
BEGIN
    IF IOR==GND THEN
      CASE (A[2..0]) IS
       WHEN H"02" => D_TRI[0].IN = P_STOP;
       WHEN H"04" => D_TRI[].IN = PA2I[];
       WHEN H"05" => D_TRI[].IN = PB2I[];
       WHEN H"06" => D_TRI[].IN = PC2I[];
       WHEN OTHERS => D_TRI[].IN = GND;
      END CASE;
    END IF;
    D_TRI[].OE = !IOR;
    D[] = D_TRI[].out;

    IF IOW==GND THEN
      CASE (A[2..0]) IS
    WHEN 1 => P_CON[6..0].D = D[6..0];
              PA[].D = PA[].Q;
              PB[].D = PB[].Q;
              PC[].D = PC[].Q;
    WHEN 4 => PA[].D = D[];
              PB[].D = PB[].Q;
              PC[].D = PC[].Q;
              P_CON[].D = P_CON[].Q;
    WHEN 5 => PA[].D = PA[].Q;
              PB[].D = D[];
              PC[].D = PC[].Q;
              P_CON[].D = P_CON].Q;
    WHEN 6 => PA[].D = PA[].Q;
              PB[].D = PB[].Q;
              PC[].D = D[];
              P_CON[].D = P_CON].Q;
      END CASE;
    END IF;

    PA[].CLK = IOW;
    PB[].CLK = IOW;
    PC[].CLK = IOW;
    P_CON[].CLK = IOW;

    PA2O[] = PA[].Q;
    PB2O[] = PB[].Q;
    PC2O[] = PC[].Q;

    P_CLK  = P_CON[0].Q;
    P_ADD  = P_CON[1].Q;
    P_RD   = P_CON[2].Q;
    P_WR   = P_CON[3].Q;
    P_RESET = P_CON[4].Q;
    P_BUF  = P_CON[5].Q;
    P_RUN  = P_CON[6].Q;

END;

SUBDESIGN MUX
(
    A[7..0],B[7..0] : input;
    SEL,OE : input;
    Q[7..0] : output;
)
VARIABLE
    TNODE[7..0]  : TRI;
BEGIN
    IF SEL == VCC THEN TNODE[].IN = A[];
      ELSE TNODE[].IN = B[];
    END IF;
    TNODE[].OE = !OE;
    Q[] = TNODE[].out;
END;

SUBDESIGN COUNTER16BIT
(
    A[15..0] : output;
    clkin,reset : input;
)
VARIABLE
    ff[15..0] : DFF;
BEGIN
    ff[].clk = clkin;
    ff[].clrn = !reset;
    ff[] = ff[]+1;
    A[] = ff[];
END;

SUBDESIGN Compare
(
    Q[15..0],P[15..0] : input;
    EQ : output;
)
BEGIN
    IF P[] == Q[] THEN EQ = GND;
      ELSE EQ = VCC;
    END IF;
END;

SUBDESIGN Divide375
(
    clkout : output;
    clkin : input;
)
VARIABLE
    ff[9..1] : DFF;
BEGIN
    ff[].clk = clkin;
    IF ff[] < 374 THEN ff[] = ff[]+1;
      ELSE ff[] =0;
    END IF;
    clkout = ff[9];
END;
```

ภาคผนวก ฉ.
ผลงานที่ได้รับการตีพิมพ์

ขอรับรองว่าผลงานวิจัย

เรื่อง

ชุดฝึกพิมพ์สำหรับคนตาบอดและชุดพัฒนาระบบ

โดย

สมคิด จรัสกิจวิกัยกุล และ พลผดุง ผดุงกุล

ได้ผ่านการพิจารณาจากคณะกรรมการผู้ทรงคุณวุฒิ สาขาวิศวกรรมศาสตร์ และได้นำเสนอในการประชุมทางวิชาการของมหาวิทยาลัยเกษตรศาสตร์ ครั้งที่ 39 ระหว่างวันที่ 5-7 กุมภาพันธ์ 2544

(ศาสตราจารย์ ต่อกุล กาญจนาลัย)
รองอธิการบดีฝ่ายวิชาการ
ประธานคณะกรรมการดำเนินการจัดการประชุมทางวิชาการ ครั้งที่ 39

ชุดฝึกพิมพ์สำหรับคนตาบอดและชุดพัฒนาระบบ

Blinds-Typing Training Set and Its Developing Set

สมคิด จรัสกิจวิกัยกุล [1] และ พลผดุง ผดุงกุล [1]

Somkid Jaruskitwikaikul [1] and Polphadung Phadungkul [1]

บทคัดย่อ

บทความนี้นำเสนอชุดฝึกพิมพ์สำหรับคนตาบอดและชุดพัฒนาระบบ ในส่วนของชุดฝึกพิมพ์สำหรับคนตาบอดจะเป็นชุดที่เชื่อมต่อกับคีย์บอร์ดที่ใช้กับเครื่องคอมพิวเตอร์ทั่วไป ใช้สำหรับการฝึกฝนการพิมพ์ทั้งภาษาไทยและภาษาอังกฤษโดยใช้เสียงเป็นตัวช่วยบอกผู้ใช้ว่าได้พิมพ์คีย์ใด ในชุดฝึกพิมพ์นี้ยังประกอบด้วยแบบฝึกหัดประเมินผลการฝึกพิมพ์และบอกผลการประเมินในรูปของเสียงอีกด้วย อีกทั้งผู้ใช้ยังสามารถทำการแก้ไขแบบฝึกหัดในแต่ละส่วนได้ด้วยตนเองโดยโปรแกรมบนคอมพิวเตอร์ พร้อมทั้งสามารถเปลี่ยนแปลงเสียงต่าง ๆที่ใช้ในชุดฝึกพิมพ์โดยชุดพัฒนาระบบที่สร้างขึ้น โดยในส่วนของชุดพัฒนาระบบจะสร้างเป็นการ์ดที่เชื่อมต่อกับเครื่องคอมพิวเตอร์เพื่อทำหน้าที่รับสัญญาณเสียงมาแปลงเป็นสัญญาณดิจิตอล และนำข้อมูลดิจิตอลที่ได้นี้นำมาประมวลผลด้วยโปรแกรมบนคอมพิวเตอร์ โปรแกรมนี้จะสามารถทำการตัด-ต่อเสียง, ปรับปรุงคุณภาพของเสียงได้ และสามารถบันทึกข้อมูลเสียงนี้ให้อยู่ในรูปของไฟล์ข้อมูล ซึ่งไฟล์ข้อมูลที่ได้นี้สามารถนำใช้โปรแกรมลงบนชุดฝึกพิมพ์หรือนำไปประยุกต์ใช้งานกับวงจรอื่น ๆที่ต้องการใช้เสียงได้อีกด้วย

ABSTRACT

This paper presents a blinds-typing training set and its developing set. The typing training set, connecting to computer keyboard, is used for typing-training both Thai and English language. When the blind presses any key in the keyboard, a training set will produce a sound according to that key. So the blind can recognize which key that he presses. Moreover this training set includes exercises for practice and evaluating typing skill and it can also report the outcome of evaluation in form of sound. Additionally, a trainer can edit the exercises by using computer software and can change any sound in training set by using the developing set. The developing set, connecting to computer, functions as a converter, which transforms sound signal to digital data. The data from the developing set is processed (realign, reform) by computer software and is recorded in file later. This file can be recorded in training set or can be used further with other sound applications.

---

1 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang

คำนำ

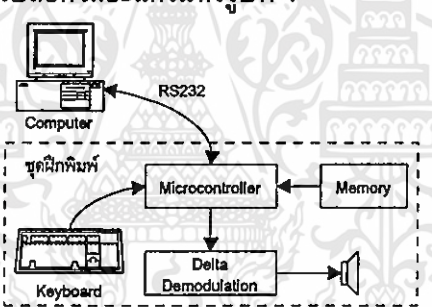
ในปัจจุบันคอมพิวเตอร์นับได้ว่าเป็นสิ่งที่สำคัญอย่างหนึ่งในชีวิตประจำวัน ไม่ว่าจะเป็นการศึกษาหาความรู้ต่างๆ , การค้าขายผ่านอินเตอร์เนตเป็นต้น โดยการใช้งานคอมพิวเตอร์นั้นยังจำกัดอยู่ที่บุคคลที่มีตาปกติ เพราะคนที่ตาบอดจะไม่สามารถพิมพ์คีย์บอร์ดให้ถูกต้องได้โดยที่ไม่มีใครช่วยเหลือ จึงได้คิดชุดฝึกพิมพ์สำหรับคนตาบอดขึ้น โดยใช้เสียงมาใช้ให้เป็นประโยชน์ให้คนตาบอดได้รับทราบถึงคีย์ที่พิมพ์ไป สามารถทำงานได้โดยไม่ต้องใช้เครื่องคอมพิวเตอร์ มีแบบฝึกหัดฝึกฝนและทดสอบความสามารถในการพิมพ์ โดยแบบฝึกหัดต่างๆนี้สามารถแก้ไขได้ด้วยตนเองโดยซอฟต์แวร์บนคอมพิวเตอร์ และสามารถบันทึกเสียงเข้าไปในชุดฝึกพิมพ์ใหม่ได้โดยใช้ชุดพัฒนาระบบที่สร้างขึ้น

ส่วนชุดพัฒนาระบบนั้นใช้งานไอซีเสียงที่เป็นที่นิยมใช้กันคือไอซีเบอร์ MC3418 (Motorola Inc., 1995) ซึ่งเป็นไอซีที่ทำหน้าที่แปลงสัญญาณเสียงเป็นสัญญาณดิจิตอลที่ใช้หลักการ Continuously Variable Slope Delta Modulation (CVSD) ซึ่งจะนำสัญญาณดิจิตอลที่ได้จาก MC3418 นี้ขึ้นไปประมวลผลโดยซอฟต์แวร์บนเครื่องคอมพิวเตอร์และบันทึกเป็นไฟล์ข้อมูลเก็บไว้ เพื่อจะได้นำไปใช้ประโยชน์ในการนำข้อมูลเสียงที่ได้บันทึกไว้ไปใช้กับชุดฝึกพิมพ์หรือวงจรประยุกต์อื่นๆต่อไป

หลักการทำงาน

ชุดฝึกพิมพ์สำหรับคนตาบอด

ชุดฝึกพิมพ์สำหรับคนตาบอดมีบล็อกไดอะแกรมดังรูปที่ 1

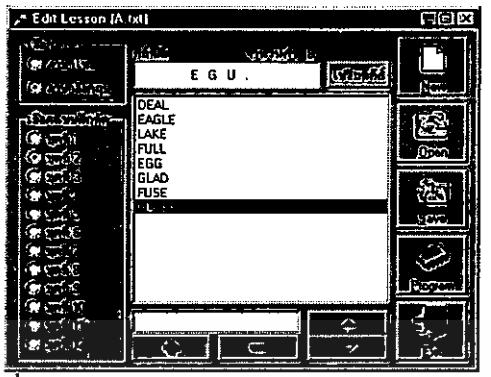


รูปที่ 1 บล็อกไดอะแกรมของชุดฝึกพิมพ์สำหรับคนตาบอด

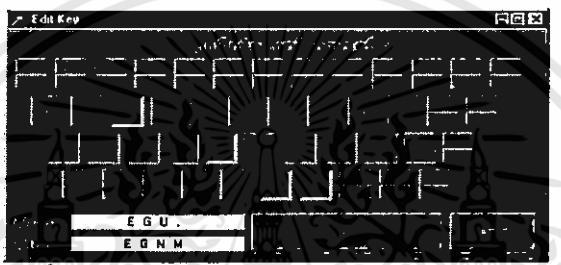
ชุดฝึกพิมพ์สำหรับคนตาบอดจะมีไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานหลักทำหน้าที่ประมวลผลสัญญาณจากคีย์บอร์ด และอ่านข้อมูลเสียงที่เก็บอยู่ในหน่วยความจำออกไปยังส่วนดีมอดูเลชันแปลงเป็นสัญญาณเสียงออกสู่ลำโพงต่อไป ซึ่งสามารถทำงานปกติได้โดยไม่ต้องเชื่อมต่อกับคอมพิวเตอร์ ส่วนของการเชื่อมต่อกับคอมพิวเตอร์จะเชื่อมต่อผ่านทางพอร์ตอนุกรมใช้เพื่อทำการแก้ไขแบบฝึกหัดหรือเปลี่ยนแปลงเสียงในชุดฝึกพิมพ์ได้ด้วยตัวผู้ใช้เอง

การทำงานของชุดฝึกพิมพ์จะประกอบด้วย 2 โหมดการทำงานด้วยกันคือ โหมดแรกจะเป็นโหมดที่เมื่อผู้ใช้มีการพิมพ์คีย์ใดก็จะมีเสียงบอกออกมาว่าคีย์ใดถูกพิมพ์ ส่วนโหมดที่สองจะเป็นโหมดของแบบฝึกหัดซึ่งจะมีแบบฝึกหัด 2 แบบด้วยกัน แบบฝึกหัดแบบแรกจะเป็นแบบฝึกหัดฝึกพิมพ์ทีละคีย์โดยให้ผู้ใช้พิมพ์คีย์ตามเสียงคีย์ที่บอกออกมา ส่วนแบบฝึกหัดแบบที่สองเป็นแบบฝึกหัดพิมพ์เป็นคำโดยชุดฝึกพิมพ์จะบอกคำออกมาและให้ผู้ใช้พิมพ์ตามให้เป็นคำนั้นๆ แบบฝึกหัดแต่ละแบบมีได้สูงสุด 12 ชุดแบบฝึกหัดภาษาไทยและ 12 ชุดแบบฝึกหัดภาษาอังกฤษ ซึ่งรวมทั้งหมดแล้วมีแบบฝึกหัดได้สูงสุดทั้งสิ้น 48 ชุดด้วยกัน เมื่อจบแบบฝึกหัดแต่ละชุดจะมีเสียงบอกค่าความถูกต้องในการพิมพ์และความเร็วในการพิมพ์ด้วย แบบฝึกหัดแต่ละชุดนี้สามารถทำการแก้ไขได้ด้วยตนเองโดยซอฟต์แวร์บนคอมพิวเตอร์โดยทำการเชื่อมต่อชุดฝึกพิมพ์กับ

คอมพิวเตอร์ทางพอร์ตอนุกรม ผ่านทางพอร์ตอนุกรมที่เชื่อมต่อฝึกหัดพิมพ์คีย์ดังรูปที่ 3 เมื่อแก้ไขแบบฝึกหัดได้แล้วก็สามารถทำการโหลดข้อมูลลงชุดฝึกพิมพ์โดยโดยมีหน้าต่างสำหรับแก้ไขแบบฝึกหัดพิมพ์คำดังรูปที่ 2 และแก้ไขแบบ

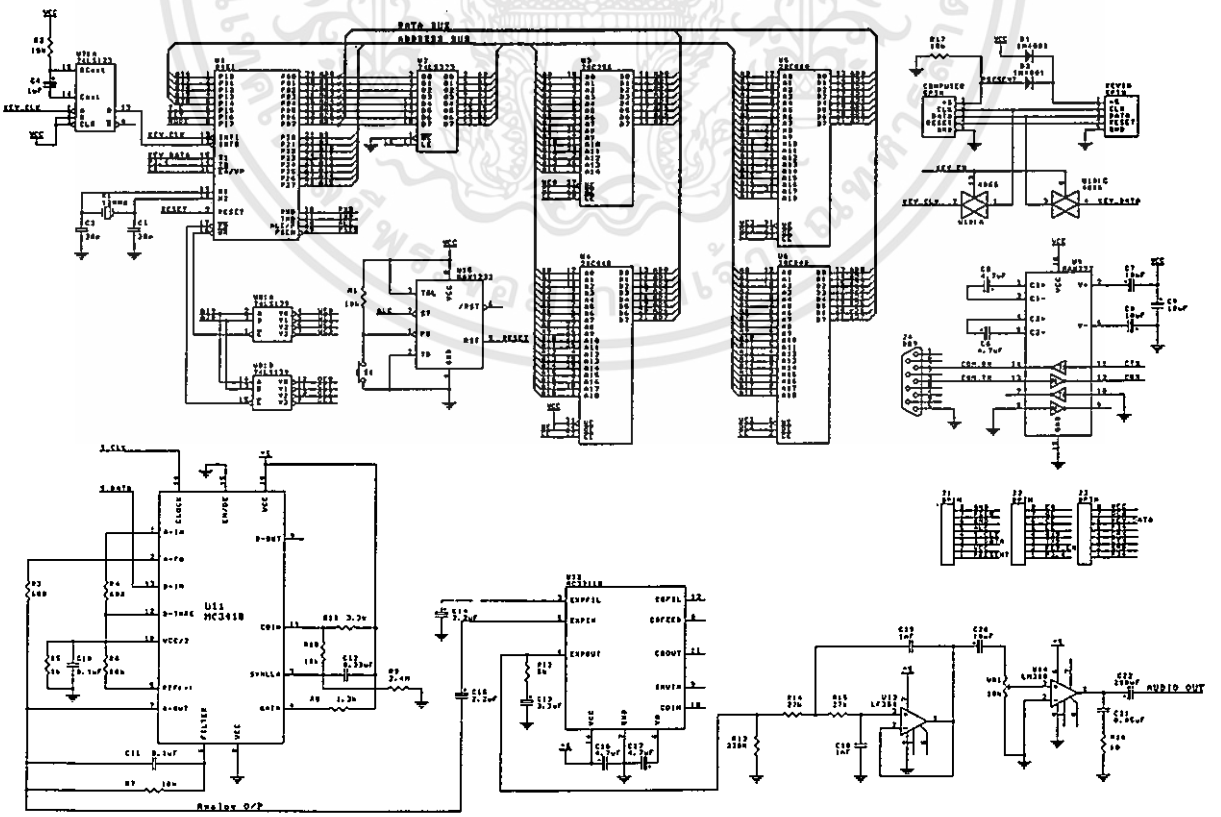


รูปที่ 2 โปรแกรมสำหรับแก้ไขแบบฝึกหัดพิมพ์คำ



รูปที่ 3 โปรแกรมสำหรับแก้ไขแบบฝึกหัดพิมพ์คีย์

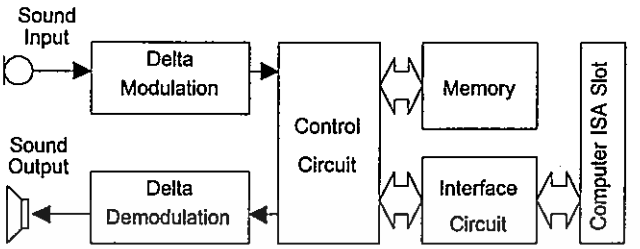
นอกจากนี้เสียงต่างๆที่ใช้ในชุดฝึกพิมพ์ยังสามารถแก้ไขโปรแกรมลงไปในชุดฝึกพิมพ์ใหม่ได้โดยชุดพัฒนาระบบที่สร้างขึ้นร่วมกับซอฟต์แวร์ข้างต้นได้อีกด้วย



รูปที่ 4 วงจรรวมของชุดฝึกพิมพ์สำหรับคนตาบอด

ชุดพัฒนาระบบ

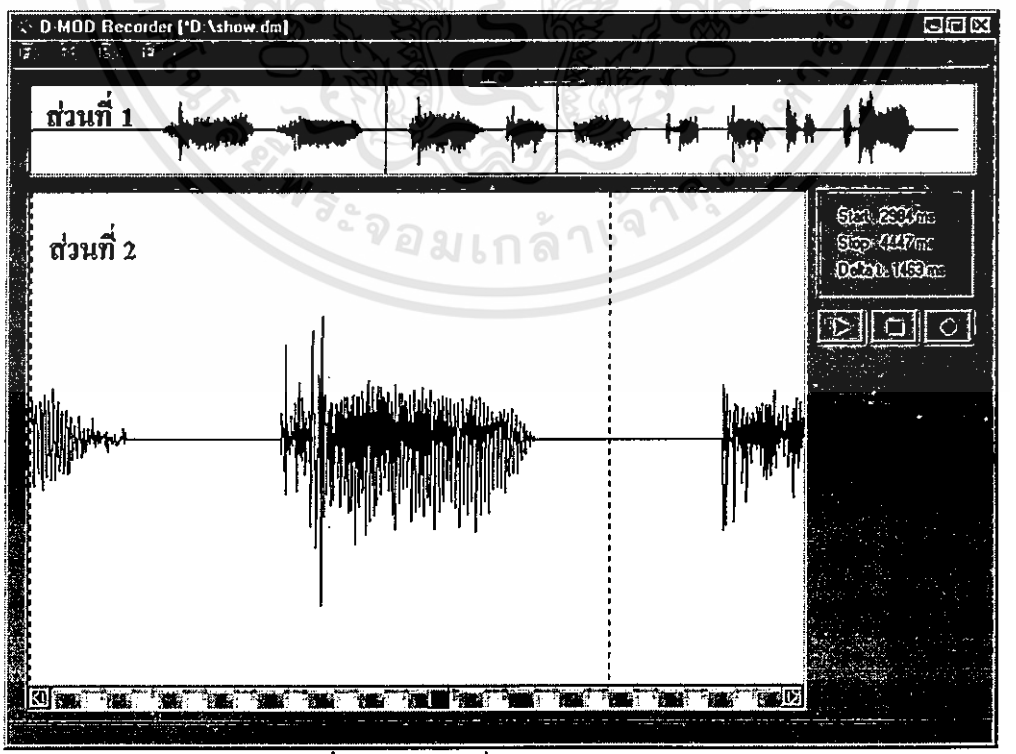
ชุดพัฒนาระบบนี้สร้างขึ้นมาเป็นการ์ดที่เสียบกับสล็อตขยายระบบของคอมพิวเตอร์ทำหน้าที่รับสัญญาณเสียงแปลงเป็นสัญญาณดิจิตอลบันทึกเก็บไว้เป็นไฟล์ข้อมูลบนคอมพิวเตอร์ ซึ่งไฟล์เสียงที่บันทึกไว้นี้สามารถนำไปโปรแกรมลงบนชุดฝึกพิมพ์ได้โดยซอฟต์แวร์ในรูปที่ 2 หรืออาจจะนำไปใช้ในวงจรประยุกต์อื่นๆ ที่ต้องการให้มีเสียงพูดได้ บล็อกไดอะแกรมของชุดพัฒนาระบบที่สร้างขึ้นแสดงดังรูปที่ 5



รูปที่ 5 บล็อกไดอะแกรมของชุดพัฒนาระบบ

จากบล็อกไดอะแกรมในรูปที่ 5 สัญญาณเสียงทางด้านอินพุตจะถูกแปลงเป็นสัญญาณดิจิตอล โดยผ่านวงจร Delta Modulation ที่ใช้ไอซี MC3418 สัญญาณดิจิตอลที่ได้จะส่งผ่านไปยังวงจรควบคุมเพื่อจัดเก็บข้อมูลดิจิตอลนี้ลงในหน่วยความจำต่อไป โดยหน่วยความจำที่ใช้มีขนาด 32 กิโลไบต์ สามารถเก็บสัญญาณเสียงได้นาน 16 วินาทีที่อัตราการสุ่ม 16 kHz นอกจากนี้วงจรควบคุมยังทำหน้าที่ในการควบคุมติดต่อกับซอฟต์แวร์บนคอมพิวเตอร์ผ่านวงจรอินเตอร์เฟส (Tompkins และ Webster,1988) เพื่อทำการเริ่ม/หยุดบันทึกเสียงทำหน้าที่รับส่งข้อมูลเสียงระหว่างคอมพิวเตอร์กับหน่วยความจำ และส่งข้อมูลดิจิตอลไปยังส่วน Delta Demodulation เพื่อแปลงกลับเป็นสัญญาณเสียงเพื่อส่งออกสู่ลำโพงต่อไป

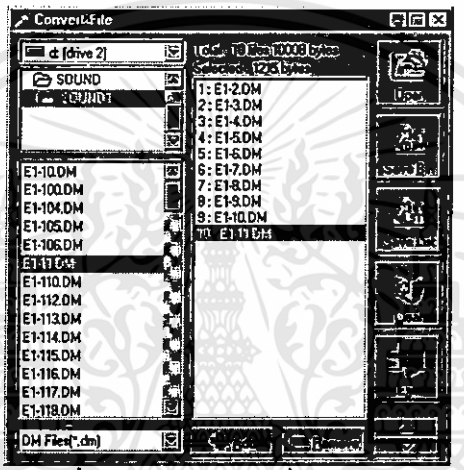
ซอฟต์แวร์ของชุดพัฒนาระบบที่สร้างขึ้นนี้มีหน้าจอดังรูปที่ 6 โดยมีหน้าต่างที่ 1 แสดงข้อมูลเสียงที่ได้บันทึกไว้ทั้งหมด ส่วนหน้าต่างที่ 2 จะเป็นหน้าต่างย่อยที่ใช้แสดงข้อมูลเสียงเป็นความยาว 1 วินาที



รูปที่ 6 โปรแกรมที่ใช้กับชุดพัฒนาระบบ

ซอฟต์แวร์นี้สามารถทำงานต่าง ๆได้ดังนี้

- สั่งเริ่ม-หยุดบันทึกเสียง (ได้นานสูงสุด 16 วินาที)
- มีเคอร์เซอร์ 2 เคอร์เซอร์ เพื่อทำการเลือกช่วงข้อมูลสัญญาณเสียงที่บันทึกไว้
- ทำการบันทึกข้อมูลเสียงช่วงที่เลือกหรือช่วงทั้งหมดให้อยู่ในรูปของไฟล์ข้อมูล
- เปิดไฟล์ข้อมูลที่ได้บันทึกไว้มาเพื่อทำการแก้ไข
- ทำการตัด-ต่อสัญญาณเสียง
- ส่วนของการทำสัญญาณเงียบ(Mute)
- ส่วนซอฟต์แวร์การกรองสัญญาณแบบดิจิตอล (Digital Filter) กรองสัญญาณเสียงที่ได้บันทึกไว้แล้วด้วยกระบวนการทางดิจิตอลเพื่อขจัดสัญญาณรบกวนอีกครั้งหนึ่ง
- ส่วนของการเลือกไฟล์ข้อมูลหลาย ๆไฟล์เพื่อบันทึกข้อมูลรวมเป็นไฟล์ไบนารีไฟล์เดียว โดยส่วนนี้มีหน้าต่างดังรูปที่ 7



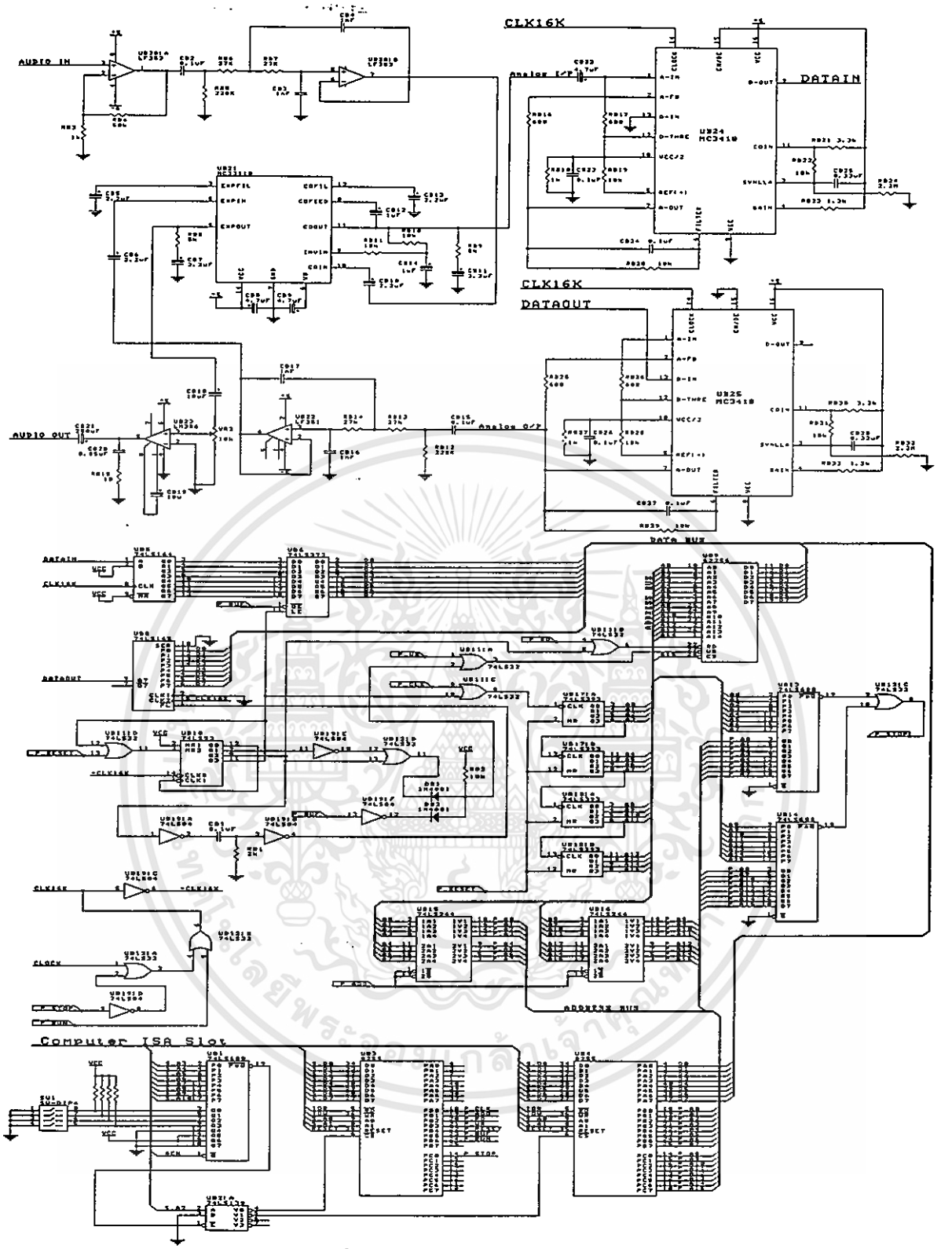
รูปที่ 7 แสดงหน้าต่างที่ใช้เลือกไฟล์

ในส่วนของการเลือกไฟล์นี้สามารถเลือกไฟล์เสียงย่อยได้ไม่เกิน 256 ไฟล์ หรือไม่เกิน 1Mbytes เมื่อเลือกเสร็จแล้วก็สามารถบันทึกข้อมูลรวมเป็นไฟล์ไบนารีไฟล์เดียว และโปรแกรมจะสร้างไฟล์รายงานเพื่อบอกด้วยว่าไฟล์เสียงย่อย ๆใดอยู่ที่แอดเดรสที่เท่าไรถึงเท่าไรในไฟล์รวม เพื่อความสะดวกในการโปรแกรมเพื่ออ่านข้อมูลเสียงต่อไป ตัวอย่างของไฟล์รายงานแสดงในรูปที่ 8

```
1   D:\SOUND\SOUND1\E1-2.DM     0002C1 - 000728
2   D:\SOUND\SOUND1\E1-3.DM     000729 - 000B96
3   D:\SOUND\SOUND1\E1-4.DM     000B97 - 001020
```

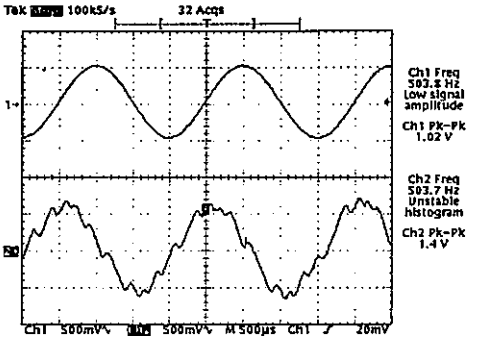
รูปที่ 8 ตัวอย่างบางส่วนของไฟล์รายงาน

ถ้าจะนำไฟล์เสียงรวมนี้ไปโปรแกรมลงในชุดฝึกพิมพ์ก็ทำได้โดยใช้ซอฟต์แวร์ในรูปที่ 2 ได้ทันที แต่ถ้าจะนำไปใช้กับวงจรประยุกต์อื่น ๆที่ต้องการให้มีเสียงพูดก็ทำได้เช่นกัน โดยนำไฟล์เสียงที่ได้นี้ไปโปรแกรมลงใน EPROM ได้โดยเครื่องโปรแกรม EPROM ทั่ว ๆไป และอาจใช้ไมโครคอนโทรลเลอร์อ่านข้อมูลเสียงไฟล์ย่อย ๆต่าง ๆได้ เช่นถ้าต้องการเล่นเสียงที่บันทึกอยู่ในไฟล์ E1-2.DM แล้ว ก็ต้องทำการเขียนโปรแกรมอ่านข้อมูลตั้งแต่แอดเดรส 0002C1 ถึงแอดเดรส 000728 ของ EPROM ส่งผ่านไปยังวงจรดีมอดูเลชันที่ใช้ MC3418 เป็นต้น



รูปที่ 9 วงจรรวมของชุดพัฒนาระบบ

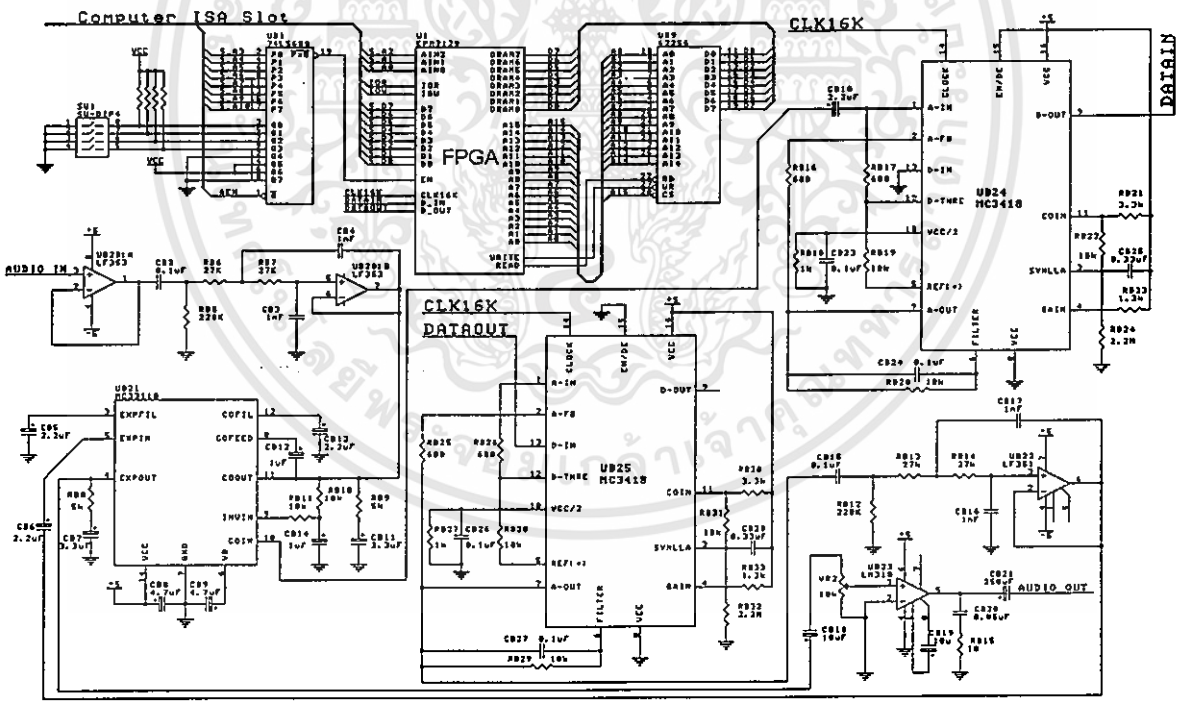
จากการทดลองป้อนสัญญาณอินพุตเป็นสัญญาณซายน์ความถี่ 500Hz ให้กับชุดพัฒนาระบบและวัดสัญญาณเอาต์พุตที่ออกไปขับลำโพงได้ผลดังรูปที่ 10 โดยสัญญาณรูปบนเป็นสัญญาณอินพุต และสัญญาณรูปล่างเป็นสัญญาณเอาต์พุต จะพบว่าสัญญาณเอาต์พุตที่ได้มีการหน่วงเวลาเล็กน้อยอันเนื่องมาจากวงจรควบคุมและสัญญาณมีความเพี้ยนอันเนื่องมาจากกระบวนการ CVSD แต่ว่ายังถือว่าอยู่ในระดับที่ยอมรับได้



รูปที่ 10 สัญญาณอินพุตและเอาต์พุตที่ทำการทดสอบชุดพัฒนาระบบ

การพัฒนาชุดพัฒนาระบบบนชิพ FPGA

เนื่องจากชุดพัฒนาระบบมีการใช้งานไอซีที่ทำหน้าที่เฉพาะต่าง ๆจำนวนมากทำให้วงจรที่ได้มีขนาดที่ค่อนข้างใหญ่ จึงได้คิดพัฒนาเปลี่ยนวงจรดิจิตอลต่าง ๆที่เป็นวงจรควบคุม มาเขียนโปรแกรมขึ้นเพื่อที่จะใช้งานชิพ FPGA เพียงตัวเดียว ซึ่งการออกแบบวงจรบนชิพ FPGA นี้จะออกแบบวงจรดิจิตอลโดยใช้ Schematic ผสมกับการเขียนวงจรด้วยภาษา AHDL โดยใช้โปรแกรม Max+Plus II ของบริษัท Altera ซึ่งจะมีความยืดหยุ่นในการออกแบบ สามารถออกแบบได้อย่างรวดเร็ว แก้ไขและตรวจสอบผลการทำงานของวงจรให้ถูกต้องได้ก่อนที่จะนำไปใช้งานจริงได้และสามารถเลือกเบอร์ชิพ FPGA ที่เหมาะสมก่อนที่จะนำวงจรที่ออกแบบไปโปรแกรมลงชิพและทดสอบการทำงานกับวงจรที่ต่อร่วมต่อไป



รูปที่ 11 วงจรรวมของชุดพัฒนาระบบที่พัฒนาด้วยชิพ FPGA

จะเห็นว่าชุดพัฒนาระบบที่สร้างขึ้นใหม่นี้จะมีวงจรที่เล็กลงเหลือเพียงชิพ FPGA ตัวเดียวที่ใช้ควบคุมการทำงานทั้งหมด ซึ่งสามารถทำงานได้เหมือนกับวงจรเดิมทุกประการ ชิพ FPGA ที่ใช้คือเบอร์ EPM7128 (Altera Corporation.,1998) ซึ่งใช้จำนวนลอจิกเซลล์ไปจำนวน 122 ลอจิกเซลล์ โดยจะมีจำนวนลอจิกเซลล์ว่างอยู่อีกเล็กน้อย ซึ่งจะทำให้สามารถพัฒนาชุดพัฒนาระบบนี้ต่อไปได้อีกโดยไม่ต้องเปลี่ยนแปลงวงจรแต่อย่างใดเพียงแค่โปรแกรมลงบนชิพ FPGA ใหม่เท่านั้นเอง

สรุป

ชุดฝึกพิมพ์สำหรับคนตาบอดเป็นตัวอย่างการนำเสียงมาประยุกต์ใช้ให้เป็นประโยชน์สำหรับคนตาบอดและคนตาปกติได้รับรู้ถึงคีย์ที่พิมพ์ลงไปได้ อีกทั้งยังมีแบบฝึกหัดที่ใช้ฝึกฝนทักษะในการพิมพ์ให้มีความรวดเร็วและถูกต้องอีกด้วย และยังสามารถแก้ไขชุดของแบบฝึกหัดได้ด้วยตนเองได้ด้วยเพื่อเป็นการปรับเปลี่ยนแบบฝึกหัดให้เหมาะสมกับแต่ละบุคคลได้และมีความทันสมัยอยู่ตลอดเวลา

ส่วนชุดพัฒนาระบบที่สร้างขึ้นนอกเหนือจากการใช้ปรับปรุงเสียงที่ใช้ในชุดฝึกพิมพ์แล้ว ยังสามารถนำเสียงที่ได้บันทึกไว้ไปประยุกต์ใช้กับวงจรอื่นๆที่ต้องการใช้เสียงได้อีกด้วย อีกทั้งยังเป็นการนำไอซีเบอร์ MC3418 มาใช้ได้อย่างสะดวกมากขึ้น คือเราสามารถบันทึก ,ตัด-ต่อคำ และปรับปรุงคุณภาพเสียงก่อนได้จนเป็นที่พอใจก่อนที่จะนำไปใช้งาน แทนที่ในอดีตเมื่อบันทึกไปแล้วไม่ได้อย่างที่ต้องการก็ต้องทำการบันทึกใหม่หมดไม่สามารถแก้ไขได้ และยังสามารถใช้แทนไอซีบันทึกเสียงอื่นๆในท้องตลาดที่มีราคาที่สูงกว่า อีกทั้งยังไม่มีชุดพัฒนาระบบการบันทึกเสียงอีกด้วย หรือไอซีบันทึกเสียงของบางบริษัทอาจก็มีชุดพัฒนาระบบการบันทึกเสียงแต่ว่าต้องนำเข้าจากต่างประเทศซึ่งมีราคาแพงและหาใช้ยากอีกด้วย ดังนั้นชุดพัฒนาระบบการบันทึกเสียงที่ได้สร้างขึ้นนี้จึงนับว่ามีประโยชน์มากเพราะว่ามีราคาที่ถูกและไอซีเบอร์ MC3418 ก็เป็นไอซีที่มีราคาถูกและหาใช้ได้ง่าย

เอกสารอ้างอิง

Motorola Inc. 1995. MC3418. Motorola Communication Device Data.

W.J.Tompkins, J.G.Webster. 1988. Interfacing sensors to the IBM PC. Prentice-Hall Inc.

Altera Corporation. 1998. MAX7000 . Altera Data Book.

ประวัติผู้เขียน

นายสมคิด จรัสกิจวิกัยกุล เกิดเมื่อวันที่ 27 สิงหาคม 2518 ที่จังหวัดสมุทรสาคร สำเร็จการศึกษาวิศวกรรมศาสตร์บัณฑิต (อิเล็กทรอนิกส์) จากมหาวิทยาลัยเทคโนโลยีมหานคร เมื่อปีการศึกษา 2538 ปัจจุบันดำรงตำแหน่งอาจารย์ประจำภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร