สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมเครื่องใช้ไฟฟ้าในบ้านด้วยเสียง

Control of Household Electrical Appliances by sound

โดย

นายขจรศักดิ์ วิทยบุษราคัม 41012045

นายศรชัย ลำใย 41012074

ห้อง 2M

เลขหมู่.....
เลขทะเบียน 42168
วัน, เดือน, ปี 14 พ.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต

สาขาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	**การควบคุมเครื่องใช้ไฟฟ้าในบ้านด้วยเสียง**
ชื่อนักศึกษา	**นายขจรศักดิ์ วิทยบุษราคัม 41012045**
	นายศรชัย ลำใย 41012074
อาจารย์ที่ปรึกษา	**อาจารย์บุญชนะ ภู่ระหงษ์**
	อาจารย์พิทักษ์ ธรรมวาริน
ภาควิชา	**เทคนิคอุตสาหกรรม**
สาขาวิชา	**เทคโนโลยีโทรคมนาคม**
ปีการศึกษา	2542

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

_____ประธานกรรมการ
( )
_____กรรมการ
( )
_____กรรมการ
( )
_____กรรมการ
( )
_____กรรมการ
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การควบคุมเครื่องใช้ไฟฟ้าในบ้านด้วยเสียง
ชื่อนักศึกษา	นายขจรศักดิ์ วิทยบุษราคัม 41012045
	นายศรชัย ลำไย 41012074
อาจารย์ที่ปรึกษา	อาจารย์บุณย์ชนะ ภู่ระหงษ์
	อาจารย์พิทักษ์ ธรรมวาริน
ภาควิชา	เทคนิคอุตสาหกรรม
สาขาวิชา	เทคโนโลยีโทรคมนาคม
ปีการศึกษา	2542

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าในบ้านด้วยเสียง โดยผ่านคอมพิวเตอร์แสดงผลออกทางพอร์ต RS232 โดยใช้โปรแกรม VISUAL BASIC 6.0 ซึ่งใช้ ACTIVEX CONTROL ที่ทำการติดตั้งเพิ่มเข้าไปใน VISUAL BASIC ทำให้มีความสามารถในการจัดการเรื่องเสียงดีมากขึ้น โดย ACTIVEX CONTROL นี้ชื่อ MICROSOFT SPEECH SDK 4.0

ในการจัดการเกี่ยวกับสัญญาณเสียงนั้น ACTIVEX CONTROL จะทำหน้าที่เป็นตัวจัดการสัญญาณเสียงให้เกือบทั้งหมด เพียงแต่จะต้องทำการศึกษาวิธีการใช้และการเขียน CODE ของโปรแกรมโดย VISUAL BASIC เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TITLE	Control of household electrical appliances by sound		
STUDENT	Mr.Kajornsak	Witthayabussarakum	41012045
	Mr.Sonchai	Lamyai	41012074
ADVISOR	Mr.Boonchana	Poorahong	
	Mr.Pitak	Thumwarin	
DEGREE	Bachelor Degree of Industrial Telecommunication		
PROGRAMMER	Telecommunication Technology		
DEPARTMENT	Industrial Technology		
ACADEMIC YEAR	1999		

ABSTRACT

This project presents control of household electrical appliances by sound which uses computer to analyze and out to port RS232. The ActiveX Control (Microsoft Speech SDK 4.0) has capability for producing sound control by writing the code in Visual Basic 6.0 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ เพราะความร่วมมือของสมาชิกภายในกลุ่ม และอาจารย์ทุกท่านที่ได้ให้คำปรึกษา คำแนะนำที่ดี และกำลังใจอีกมากมายจากบิดา มารดา เพื่อนที่คอยถามไถ่ให้ความห่วงใยมาโดยตลอดในยามที่รู้สึกท้อถอย

สุดท้ายนี้ขอให้สิ่งดี ๆ ที่จงบังเกิดแด่ทุก ๆ ท่านที่กล่าวถึงตลอดไป

นายขจรศักดิ์ วิทย์บุษราคัม

นายศรชัย ลำใย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	
1.1 ความเป็นมาของปริญญานิพนธ์	1
1.2 จุดมุ่งหมายของปริญญานิพนธ์	1
1.3 ขอบเขต	1
1.4 ขั้นตอนและวิธีดำเนินงาน	2
1.5 เครื่องมือที่ใช้ในการทดลอง	2
บทที่ 2 ทฤษฎีที่ใช้ในการดำเนินงาน	
2.1 อวัยวะที่ใช้ในการเปล่งเสียง	3
2.2 ลำดับการเกิดเสียง	3
2.3 หน่วยการจำเสียงพูด	4
2.4 ActiveX	5
2.5 ActiveX Control	5
2.6 Microsoft Speech SDK	6
บทที่ 3 ฮาร์ดแวร์ (Hardware)	
3.1 การทำงานของบอร์ดอินเตอร์เฟส	19
3.2 UART	21
3.3 ส่วนควบคุมอุปกรณ์เครื่องใช้ไฟฟ้า	24
บทที่ 4 ผลการทดลอง	
4.1 ฮาร์ดแวร์ (Hardware)	25
4.2 ซอฟต์แวร์ (Software)	31
บทที่ 5 สรุปปัญหาและข้อเสนอแนะ	
5.1 สรุป	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญต่อ

เรื่อง	หน้า
5.2 ปัญหาที่เกิดขึ้นในการทำงาน	37
5.3 แนวทางการแก้ไข	38
5.4 ข้อเสนอแนะในโครงงาน	38
ภาคผนวก	
ภาคผนวก ก SOURCE CODE ของโปรแกรม	39
ภาคผนวก ข ลายวงจรและการวางอุปกรณ์	52
ภาคผนวก ค	
คอนโทรลตัวอื่น ๆ ที่อยู่ใน MICROSOFT SPEECH SDK 4.0	55
ภาคผนวก ง รูปภาพฮาร์ดแวร์	70
ภาคผนวก จ FLOW CHAT แสดงการทำงานของโปรแกรม	73
ภาคผนวก ฉ Data Cheet ของ IC UART CDP6402	75
บรรณานุกรม	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูป หน้า

รูปที่ 2.1 แสดงอวัยวะที่ใช้ในการออกเสียง 4
รูปที่ 3.1 แสดงการต่อพอร์ตอนุกรมแบบ DB9 และ DB25 19
รูปที่ 3.2 แสดงวงจรของบอร์ดอินเตอร์เฟส 20
รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงานพื้นฐานของ UART 21
รูปที่ 3.4 บล็อกไดอะแกรมแสดงการทำงานภายในไอซี 23
รูปที่ 3.5 แสดงวงจรขับรีเลย์โดยใช้โฟโต้ไดโอดเป็นตัวรับอินพุต 24
รูปที่ 4.1 แสดงเมนูของโปรแกรมทดสอบฮาร์ดแวร์ 31
รูปที่ 1 ภาคผนวก ก แสดงเมนูหลักของโปรแกรม 48
รูปที่ 2 ภาคผนวก ก แสดง HELP ของโปรแกรม 50
รูปที่ 3 ภาคผนวก ก แสดงเกี่ยวกับผู้จัดทำโปรแกรม 51
รูปที่ 1 ภาคผนวก ข แสดงการวางอุปกรณ์ของบอร์ดอินเตอร์เฟส 53
รูปที่ 2 ภาคผนวก ข แสดงลายวงจรของบอร์ดอินเตอร์เฟส 54
รูปที่ 1 ภาคผนวก ง แสดงฮาร์ดแวร์ภายนอก 71
รูปที่ 2 ภาคผนวก ง แสดงฮาร์ดแวร์วงจรภายใน 72
รูปที่ 1 ภาคผนวก จ FLOW CHAT แสดงการทำงานของโปรแกรม 74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 2.1 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ MenuCreate 10
ตารางที่ 2.2 แสดงค่าของ flags ของคำสั่ง MenuCreate 11
ตารางที่ 2.3 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ Activate 12
ตารางที่ 2.4 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ AddCommand 14
ตารางที่ 2.5 แสดงค่าของ flags ของคำสั่ง AddCommand 14
ตารางที่ 2.6 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ Deactivate 15
ตารางที่ 2.7 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ GetCommand 16
ตารางที่ 2.8 แสดงค่าของ flags ของคำสั่ง GetCommand 17
ตารางที่ 2.9 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ ReleaseMenu 17
ตารางที่ 2.10 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ CommandRecognize 18
ตารางที่ 3.1 แสดงการกำหนดโหมดเพื่อควบคุมการทำงานของไอซี 6402 22
ตารางที่ 4.1 ตารางการทดสอบการสั่งเปิดอุปกรณ์ 32
ตารางที่ 4.2 ตารางการทดสอบการสั่งปิดอุปกรณ์ 33
ตารางที่ 4.3 ตารางแสดงเปอร์เซ็นต์ความถูกต้องของบุคคลที่ 1 33
ตารางที่ 4.4 ตารางการทดสอบการสั่งเปิดอุปกรณ์ 34
ตารางที่ 4.5 ตารางการทดสอบการสั่งปิดอุปกรณ์ 35
ตารางที่ 4.6 ตารางแสดงเปอร์เซ็นต์ความถูกต้องของบุคคลที่ 2 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

ความเป็นมาของปริญญานิพนธ์

การวิเคราะห์เสียงที่ไม่ขึ้นกับผู้พูด เป็นการพัฒนาระบบการรับคำสั่งของคอมพิวเตอร์โดยไม่เจาะจงเสียงผู้ออกคำสั่ง เป็นการเปิดกว้างให้กับคนทั่วไปสามารถออกคำสั่งกับคอมพิวเตอร์ได้ เป็นจุดเริ่มต้นของการพัฒนาระบบการสื่อสารระหว่างมนุษย์กับคอมพิวเตอร์โดยใช้เสียงคำพูดเป็นสื่อ ทำให้เกิดความสะดวกในกลุ่มสังคมที่ใช้ประโยชน์จากคอมพิวเตอร์ ซึ่งสามารถประยุกต์ใช้กับงานได้มากมายเช่นระบบการป้อนข้อมูลด้วยเสียง ระบบสั่งงานด้วยเสียงพูดทางโทรศัพท์ การควบคุมอุปกรณ์ไฟฟ้าของคนพิการและผู้ป่วยอัมพาต โดยใช้เสียงเป็นต้น

จุดมุ่งหมายของปริญญานิพนธ์

1. ศึกษาหาความเป็นไปได้วิธีการรู้จำเสียงพูดคำไทย แบบไม่ขึ้นอยู่กับคำพูดโดยวิธีการเปรียบเทียบระหว่างแบบทดสอบกับแบบอ้างอิงโดยใช้ Visual Basic
2. เพื่อหาวิธีการวิเคราะห์เสียงคำพูดคำไทยที่มีความแม่นยำสูงขึ้น
3. เพื่อใช้เป็นแนวทางและพื้นฐานความรู้ในการพัฒนาระบบรู้จำเสียงพูดคำไทยโดยวิธีการอื่นที่ใช้ให้ประสิทธิภาพสูงกว่าในอนาคต

ขอบเขต

1. การสั่งงานจะใช้คำสั่งเป็นภาษาอังกฤษซึ่งจะเป็นบุคคลใดทำการสั่งงานก็ได้
2. การทำงานต้องมีการพูดคำสั่งที่ถูกต้องเหมือนกับภาษาอังกฤษ
3. ถ้าคำสั่งในการสั่งงานไม่ถูกต้องฮาร์ดแวร์จะไม่แสดงผลออกมา
4. เอาต์พุตมี 8 เอาต์พุตต่อออกทางพอร์ต RS-232 โดยมี LED เป็นตัวแสดงผลและรีเลย์ทำการเปิด-ปิดอุปกรณ์ทั้ง 8 ชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนและวิธีการดำเนินงาน

1. เริ่มต้นจากการศึกษาหาข้อมูลจากแหล่งต่าง ๆ ที่เกี่ยวข้อง
2. ศึกษาระบบการพูดและการฟังของมนุษย์โดยการสังเกตจากพฤติกรรมและข้อมูลที่เกี่ยวข้อง
3. กำหนดขอบเขตของการดำเนินงาน
4. เขียนโปรแกรมเพื่อการทดสอบและออกแบบฮาร์ดแวร์เพื่อใช้ในการทดสอบ
5. ทำการทดลองกับฮาร์ดแวร์และปรับปรุงแก้ไขให้เหมาะกับงานที่ใช้
6. บันทึกและสรุปผลการทดลอง

เครื่องมือที่ใช้ในการทดลอง

ฮาร์ดแวร์

1. ไมโครคอมพิวเตอร์ส่วนบุคคลใช้โปรเซสเซอร์รุ่น Pentium Celeron 366MHz., หน่วยความจำหลัก 64 MB.
2. Sound Card ขนาด 32 บิต ยี่ห้อ Creative Sound Blaster รุ่น Vibra 128
3. Hard Disk Drive ขนาด 6.4 Gbytes
4. ไมโครโฟน

ซอร์ฟแวร์

1. Windows 98 และ Microsoft Office
2. โปรแกรม Microsoft Visual Basic 6.0 และ Microsoft Speech SDK 4.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่ใช้ในการดำเนินงาน

โดยทั่วไปแล้วคนเราเปล่งเสียงออกมาตามสำเนียงในระบบภาษาของตนเอง แม้ว่าคนที่อยู่ในสังคมเดียวกัน ใช้ภาษาเดียวกัน แต่เสียงแต่ละคนที่เปล่งออกมานั้น จะมีลักษณะแตกต่างกันแม้คนๆเดียวกันเปล่งเสียงคำเดียวกันสองครั้ง สัญญาณที่ได้ก็ยังมีความแตกต่างกัน แต่คนก็ยังแยกแยะความหมายของคำที่เกิดจากการพูดของแต่ละคนได้ การจดจำและแยกแยะคำพูดต่างๆจึงดูเป็นเรื่องง่ายในคน แต่สำหรับที่จะให้คอมพิวเตอร์จำและแยกแยะเสียงพูดนั้นเป็นเรื่องที่ยากมาก ขั้นตอนแรกในการทำให้คอมพิวเตอร์จำเสียงพูดได้นั้น จึงต้องมีการศึกษาลักษณะของเสียงพูดเสียก่อน โดยเริ่มจากอวัยวะที่ใช้ในการเปล่งเสียงก่อน

2.1 อวัยวะที่ใช้ในการเปล่งเสียง (Articulation)

อวัยวะที่ใช้ในการเปล่งเสียงแบ่งเป็น 3 พวกใหญ่ๆ คือ

1. อวัยวะที่ใช้ในการสร้างลม คือส่วนที่ทำให้เกิดการเคลื่อนไหวของลม

2. อวัยวะที่เคลื่อนที่ได้ (Active Articulator) หมายถึงอวัยวะที่ติดกับกระดูกคางส่วนล่าง ได้แก่ ริมฝีปากและลิ้น

3. อวัยวะที่เคลื่อนที่ไม่ได้ (Passive Articulator) หมายถึงอวัยวะที่ติดกับกระดูกคางส่วนบน ได้แก่ ริมฝีปากบน ฟันบน ปุ่มเหงือก เพดานแข้ง เพดานอ่อน ลิ้นไก่ และอวัยวะที่เป็นช่วงว่าง ได้ได้แก่ ช่องคอ ปาก ช่องจมูก

2.2 ลำดับการเกิดเสียง

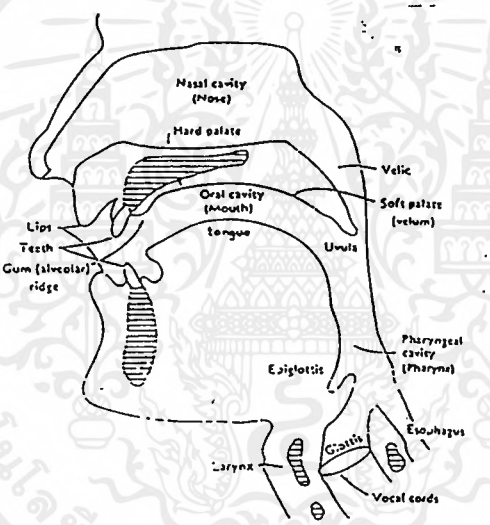
ลำดับการเกิดเสียงนั้นเราแบ่งออกได้เป็น 3 ขั้นตอนใหญ่ ๆ คือ

1. จุดเริ่มต้น (Initiation) อวัยวะที่ใช้ในขั้นตอนนี้คือปอด ที่ขั้นตอนนี้ลมจะถูกขับออกจากปอด

2. การดัดแปลงลมที่เส้นเสียง (Phonation) อวัยวะที่ใช้ในขั้นตอนนี้ก็คือ อวัยวะที่อยู่ต่อจากปอดขึ้นไปจนถึงกล่องเสียง เป็นขั้นตอนที่ลมจากปอดจะผ่านมาเข้าหลอดลมและกล่องเสียง ซึ่ง ณ ที่กล่องเสียงนี้เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิดและปิดทำให้เกิดเสียง 2 ชนิด คือ ถ้าเส้นเสียงเปิดตลอดเวลาที่ลมผ่าน ลมจะผ่านออกมาได้อย่างสะดวก ซึ่งจะทำให้เกิดเสียงไม่ก้อง แต่ถ้าเส้นเสียงปิดกั้นลมไว้ ลมที่ผ่านออกมาจะเพิ่มแรงดันมากขึ้นจนเส้นเสียงเปิดและปิดสลับกันไป ทำให้เกิดเสียงชนิดก้องและเรียกความถี่ในการเปิด ปิด ของเส้นเสียงว่า "ความถี่พื้นฐาน"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) การเปลี่ยนแปลงลักษณะเส้นเสียง (Articulation) อวัยวะที่ใช้ในขั้นตอนนี้ก็คือส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปากและช่องว่างใหญ่อีก 2 ช่อง คือ ช่องปากและช่องจมูก ขั้นตอนนี้ลมที่ออกจากกล่องเสียงจะถูกแปลงให้เกิดเสียงลักษณะต่างๆ ทั้งนี้เพราะช่องว่างภายในจะถูกเปลี่ยนแปลงขนาดตลอดเวลาที่เปลี่ยนแปลงเสียง และช่องจมูกจะถูกควบคุมด้วยเพดานอ่อน (Velume) ดังนั้นถ้าช่องว่างเหล่านี้เปลี่ยนขนาดไปจะทำให้เกิดความถี่กำทอน (Resonant Frequency) ของเสียงเปลี่ยนแปลงไปด้วย ซึ่งช่องว่างเหล่านี้เปรียบเสมือนท่อที่มีขนาดต่างๆ กัน



รูปที่ 2.1 แสดงอวัยวะที่ใช้ในการออกเสียง

2.3 หน่วยการจำเสียงพูด (Speed recognition Unit)

หน่วยการจำเสียงพูดมีวิธีการจำแนกตามลักษณะการพูดไว้ 4 แบบ

1.) ผู้พูดคนเดิม (Speaker Dependent Recognition) เป็นการที่ผู้พูดคนเดิมพูด เป็นพยางค์เพียงพยางค์เดียว เป็นคำ หรือเป็นประโยค โดยเราต้องสร้างหน่วยเปรียบเทียบที่เป็นเสียงของคนๆเดียว

2.) ผู้พูดเป็นใครก็ได้ (Speaker Independent Reconition System)เป็นการที่ใครก็ได้ที่พูด จะเป็นพยางค์เดียว เป็นคำหรือเป็นประโยค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สอนให้เครื่องรู้จักคำในลักษณะของหน่วยเสียงหรืออย่างมากหน่วยพยางค์เป็นการสอนให้เครื่องเข้าใจในเสียงที่มีการต่อเนื่องมากกว่าหนึ่งพยางค์ ต้นแบบจะถูกเก็บไว้ในหน่วยเสียงหรือพยางค์เท่านั้น
4. ผู้พูดสามารถพูดอย่างต่อเนื่องเหมือนการพูดแบบปกติ (Globolization) เครื่องจะต้องรู้จักและจำแนกคำที่มีการออกเสียงเชื่อมกัน การพูดแบบนี้ในมนุษย์เป็นเรื่องง่าย แต่สำหรับคอมพิวเตอร์เป็นเรื่องยากมาก เพราะการพูดแบบนี้ไม่สามารถหาเส้นแบ่งเขตระหว่างคำ หรือพยางค์ได้แม่นยำ ดังนั้นการรู้จักโดยการเปรียบเทียบกับต้นแบบจึงไม่ใช่ของง่าย วิธีการแบบนี้ใช้กับเครื่องมินิคอมพิวเตอร์ขึ้นไป

2.4 ActiveX

ActiveX เป็นชื่อเรียกเทคโนโลยีที่ไมโครซอฟท์คิดค้นขึ้น โดยมีวัตถุประสงค์หลักเพื่อสร้างระบบซอฟท์แวร์ที่มุ่งเน้นการใช้งานและพัฒนาสำหรับใช้บนอินเตอร์เน็ต เพื่อทำให้ซอฟท์แวร์ซึ่งอยู่ต่าง Platform สามารถทำงานร่วมกัน โดยเทคโนโลยี ActiveX มีพื้นฐานมาจากแนวความคิดของ COM (Component Object Model) ซึ่งเป็นสถาปัตยกรรมการออกแบบเชิงวัตถุ ที่มองซอฟท์แวร์เสมือนเป็นวัตถุที่ประกอบด้วยชิ้นส่วนย่อยรวมกันขึ้นมาเป็นซอฟท์แวร์ที่ต้องการ เสมือนว่าซอฟท์แวร์เป็นวัตถุจริง ๆ แบบเดียวกันกับฮาร์ดแวร์ เช่น คอมพิวเตอร์จะประกอบไปด้วยวัตถุย่อย ๆ คือ CPU, ฮาร์ดดิสก์, ซีดีรอม และส่วนประกอบอื่น ๆ ที่ประกอบกันขึ้นมาเป็นคอมพิวเตอร์ แต่ COM จะมองว่าซอฟท์แวร์เสมือนประกอบด้วยส่วนย่อย โดยแต่ละส่วนอาจอยู่คนละ Platform แยกจากกัน โดยซอฟท์แวร์ที่สร้างบนสถาปัตยกรรม COM จะสามารถสื่อสารทำงานร่วมกันได้แม้ว่าจะถูกสร้างหรือทำงานบน Platform ที่ต่างกันก็ตาม

2.5 ActiveX Control

ActiveX Control หมายถึง คอนโทรลที่เพิ่มเติมจากออบเจ็คพื้นฐานที่ Visual Basic ได้เตรียมไว้ให้แล้วใน Toolbox โดย ActiveX Control เหล่านี้มีอยู่มากมายทั้งที่มาพร้อมกับ Visual Basic และที่พัฒนาออกมาขายโดยบริษัทต่าง ๆ ทั่วโลก ซึ่ง ActiveX Control เหล่านี้เองคือพลังผลักดันให้โปรแกรม Visual Basic และเครื่องมืออีกหลายตัวของไมโครซอฟท์เป็นที่นิยมอย่างแพร่หลายเนื่องจากผู้ใช้มองตัว ActiveX Control เป็นเพียงกล่องดำหรือกล่องวัตถุ ซึ่งเพียงแค่นำคอนโทรลที่มีการทำงานตามที่ต้องการมาแปะบนฟอร์มแล้วเรียกใช้งานเท่านั้น โดยไม่ต้องสนใจการทำงานภายในของ ActiveX Control เหล่านั้น เช่น ถ้าเราต้องการเขียนโปรแกรมเพื่อติดต่อกับอินเตอร์เน็ต (Internet) ถ้าเป็นในอดีตต้องศึกษาข้อมูลมากมาย เช่น Protocol การติดต่อสื่อสาร TCP/IP ซึ่งเป็นมาตรฐานของภาษาที่คอมพิวเตอร์ใช้สื่อสารกันในระบบอินเตอร์เน็ต แต่ถ้าใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ActiveX Control เพียงแค่นำ Internet ActiveX Control มาวางลงบนฟอร์มแล้วเรียนรู้วิธีใช้เท่านั้นก็สามารถเขียนโปรแกรมที่ติดต่อกับเครือข่ายอินเตอร์เน็ตได้ในเวลาอันสั้น

2.6 Microsoft Speech SDK

Microsoft Speech SDK (Software Developer's Kit) เป็น ActiveX Control อีกอย่างหนึ่งซึ่งไม่มีในออบเจ็คพื้นฐานของ Visual Basic ต้องหาซอฟท์แวร์ตัดนี้มาทำการติดตั้งเข้าไปในคอมพิวเตอร์ที่ใช้จึงจะสามารถเรียกใช้ ActiveX Control ตัวนี้ได้

ใน SDK จะมี ActiveX Control มาให้ 6 คอนโทรลด้วยกัน ซึ่งสามารถใช้ในการรู้จำเสียง (Speech Recognition), เขียนตามคำบอก (Dictation), การสังเคราะห์เสียง (Speech Synthesis) และโทรศัพท์ (Telephony) ในสภาพแวดล้อมต่าง ๆ เช่น Visual Basic, Microsoft Office และ Internet Explorer ทั้ง 6 คอนโทรลนั้นได้แก่

1. Voice Command Control
2. Voice Dictation Control
3. Voice Text Control
4. Voice Telephony Control
5. Direct Speech Recognition Control
6. Direct Speech Synthesis Control

ซึ่งทั้ง 6 คอนโทรลจะประกอบไปด้วย 2 คอนโทรลที่ทำหน้าที่รู้จำเสียง (Direct Speech Recognition และ Voice Command Control) มี 1 คอนโทรลทำหน้าที่เขียนตามคำบอก (Voice Dictation Control) มี 2 คอนโทรลทำหน้าที่สังเคราะห์เสียง (Direct Speech Synthesis Control และ Voice Text Control) และสุดท้ายอีก 1 คอนโทรลทำหน้าที่ติดต่อโทรศัพท์ (Voice Telephony Control)

ในคอนโทรล Direct Speech Recognition Control และ Direct Speech Synthesis Control นั้นตัวคอนโทรลได้จัดการการเข้าถึงข้อมูลในการติดต่อกับ Speech API (Application Programming Interface) อย่างเต็มที่ สำหรับการสังเคราะห์เสียงและการรู้จำมันจะติดต่อกับเครื่องมือเสียงในระดับต่ำทำให้ทำงานได้เร็วขึ้นและควบคุมได้ง่าย

Voice Command Control และ Voice Text Control จะจัดการการเข้าถึงข้อมูลโดยการแชร์ (Share) Speech API ในระดับสูงในการเชื่อมต่อจะมีขีดจำกัดและช้ากว่าแบบแรกที่ติดต่อในระดับต่ำ แต่จะสะดวกกว่าโดยการจัดเตรียมทรัพยากรและการแชร์หน่วยความจำ (Memory) ระหว่างโปรแกรมที่ทำงานเกี่ยวกับเสียงจะเป็นไปโดยอัตโนมัติ

Voice Dictation Control เป็นการง่ายที่จะทำการเขียนตามที่ผู้ใช้บอกอย่างถูกต้องและแปลงเสียงเป็นข้อความตามปกติโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Voice Telephony Control เป็นเครื่องมือในการติดต่อกับการสังเคราะห์เสียง, การรู้จำเสียงม การสังเคราะห์เสียงแบบ Wave และ DTMF บน Single หรือ Multi-line voice ของอุปกรณ์โทรศัพท์

โดยจะกล่าวถึงเฉพาะ Voice Command Control ซึ่งเป็นคอนโทรลที่นำมาใช้เท่านั้นส่วนคอนโทรลตัวอื่น ๆ จะไม่ขอกล่าวถึงรายละเอียดมากนัก แต่จะมีเนื้อหาคร่าว ๆ ในภาคผนวก

Voice Command Control

Properties

Voice Command control มีคุณสมบัติดังต่อไปนี้

- AutoGainEnable
- AwakeState
- CountCommands
- Device
- Enabled
- EnableMenu
- hWnd
- Initialized
- LastError
- MenuCreate
- Microphone
- Speaker
- SRMode
- SuppressExceptions
- Threshold

Methods

Voice Command control มี Method ดังต่อไปนี้

- Activate
- AddCommand
- CmdMimic
- Deactivate
- EnableItem
- GeneralDlg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GetCommand

LexiconDlg

ListGet

ListSet

MenuDelete

ReleaseMenu

Remove

SetCommand

TrainGeneralDlg

TrainMenuDlg

TrainMicDlg

Events

Voice Command control มี events ดังต่อไปนี้

AttribChanged

ClickIn

CommandOther

CommandRecognize

CommandStart

Interference

MenuActivate

UtteranceBegin

UtteranceEnd

VUMeter

ในที่นี้จะกล่าวถึงเฉพาะ Properties , Methods และ Events ที่ใช้ในการเขียนโปรแกรมเท่านั้น ส่วนที่ไม่ได้กล่าวถึงสามารถดูได้ใน Help ของ Microsoft Speech SDK 4.0 ซึ่งจะมีคำบรรยายต่าง ๆ โดยละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Properties ที่ใช้การเขียนโปรแกรมได้แก่

1. CountCommands
2. Enabled
3. Initialized
4. MenuCreate

CountCommand

รูปแบบคำสั่ง

CountCommand(menu As Long) As Long

คำอธิบาย

เป็นคำสั่งที่ใช้ในการรีเทิร์นค่าของคำสั่งบนเมนู ซึ่งเป็นคำสั่งที่สามารถอ่านได้อย่างเดียว ไม่สามารถทำอย่างอื่นได้

Enables

รูปแบบคำสั่ง

Enabled As Long

คำอธิบาย

TRUE ถ้าสามารถทำการรู้จำเสียงหรือ FALSE ถ้าไม่สามารถทำได้ ซึ่งต้องทำการเซ็ตเป็น TRUE เพื่อที่จะสามารถฟังเสียงจากไมโครโฟนได้

เมื่อการรู้จำเสียงไม่สามารถทำได้ตัวรู้จำก็จะไม่สามารถทำการรู้จำคำสั่งใด ๆ จากเมนูต่าง ๆ ได้ แอพลิเคชันหนึ่ง ๆ จะใช้คุณสมบัติ ENABLED ซึ่งจะอนุญาตใช้ผู้ใช้ทำการปิดความสามารถในการรู้จำเสียงได้เป็นการหยุดการรู้จำเสียงเป็นการชั่วคราว

ผู้ใช้สามารถใช้บัญชีรายชื่อคำสั่งควบคุมในการสั่งการเปิด-ปิดระบบเสียง เมื่อใดก็ตามที่เสียงถูกเปิดหรือปิด ข้อมูลข่าวสาร WM_SPEECHSTARTED หรือ WM_SPEECHENDED ทั้งหมดจะส่งไปที่ระบบวินโดว์ระดับบน แอพลิเคชันจะสามารถใช้ข้อมูลข่าวสารนี้ในการตัดสินใจเมื่อจะเปิดหรือปิดตัว VOICE COMMANDS หรือ VOICE TEXT

การเซ็ต ENABLED ให้เป็น FALSE นั้นจะอนุญาตใช้ผู้ใช้ปิดการร์จำเสียงดังนั้นจะไม่เกิดการรู้จำ สำหรับตัวอย่าง ผู้ใช้อาจจะต้องการปิดการรู้จำเสียงจากไมโครโฟนในระหว่างการประชุม ดังนั้นการรู้จำเสียงจะหยุดรอหรือใครบางคนจะเผลอพูดคำสั่งให้ปิดการรู้จำในเมนู

ถ้าแอพลิเคชัน VOICE-NAVIGATION ถูกติดตั้งบนเครื่องคอมพิวเตอร์ของผู้ใช้ แอพลิเคชันจะไม่ต้องการเซ็ตสถาวะ ENABLED อย่างไรก็ตามมันจะต้องเรียกใช้ฟังก์ชันเพื่อเริ่มที่จะทำการรู้จำเสียง สำหรับตัวอย่าง ถ้าแอพลิเคชันได้รับคำสั่ง "Do you want to print the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

document ?" มันจะการรู้จำเสียงและทำการตอบรับเสียงจากผู้ใช้

Initialized

รูปแบบคำสั่ง

Initialized As Long

คำอธิบาย

จะเท่ากับ 1 ถ้าตัวคอนโทรลทำการเริ่มต้น และจะเป็น 0 ถ้าไม่ ผู้ใช้สามรถควบคุมตัวคอนโทรลให้มันเริ่มต้นโดยการเซ็ตให้เป็น 1 (ตัวจัดการเสียงจะสามารถโหลดได้ช้า ดังนั้นผู้ใช้อาจจะต้องการปรับคอนโทรลมากกว่าเมื่อทำการเริ่มต้น)

ส่วนมากการใช้ METHODS และ PROPERTIES จะทำการเริ่มต้นคอนโทรลโดยอัตโนมัติถ้ามันไม่ได้ถูกผู้ใช้เซ็ตให้เริ่มต้นอยู่ก่อนแล้ว

MenuCreate

รูปแบบคำสั่ง

MenuCreate(Application As String, State As String, flags As Long) As Long

คำอธิบาย

จะทำการสร้างเมนูเสียงขึ้นมาใหม่หรือตัวเมนูเสียงอาจจะมีอยู่แล้วสำหรับแอพลิเคชันสามารถอ่านได้อย่างเดียว

พารามิเตอร์	คำบรรยาย
Application	เป็นชื่อของแอพลิเคชัน เช่น "Excel"
State	การกำหนดชื่อ เช่น "Main Menu" หรือ "File Open dialog box."
flags	Flags เป็นตัวบอกสถานะในการสร้างเมนูว่าสร้างลักษณะใด พารามิเตอร์นี้สามารถเป็นหนึ่งในค่าต่างของในตารางแสดงค่าข้างล่าง

ตารางที่ 2.1 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ MenuCreate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ flags	คำบรรยาย
VCMDMC_CREATE_ALWAYS	ทำการสร้างเมนูว่าง ๆ ขึ้นมาและตั้งชื่อ ถ้ามีเมนูอยู่แล้วในฐานข้อมูลของเมนูเสียง มันจะทำการลบออกก่อนแล้วทำการเก็บเมนูใหม่ลงไปในฐานข้อมูลเมื่อตัวเมนูเก่าถูกปล่อยออกจากหน่วยความจำ
VCMDMC_CREATE_NEW	ทำการสร้างเมนูว่าง ๆ ขึ้นมาและตั้งชื่อ ถ้ามีเมนูอยู่แล้วในฐานข้อมูลของเมนูเสียง ฟังก์ชันจะทำการรีเทิร์นค่า Error เมนูที่สร้างขึ้นใหม่จะถูกเก็บลงในฐานข้อมูล เมื่อ เมนูเก่าถูกปล่อยออกจากหน่วยความจำ
VCMDMC_CREATE_TEMP	ทำการสร้างเมนูว่าง ๆ ขึ้นมาและตั้งชื่อ ถ้ามีเมนูอยู่แล้วในฐานข้อมูลของเมนูเสียง ฟังก์ชันจะทำการรีเทิร์นค่า Error เมนูใหม่นี้จะเป็นเมนูชั่วคราวและจะถูกยกเลิกเมื่อตัวเมนูถูกปล่อยออกจากหน่วยความจำ
VCMDMC_OPEN_ALWAYS	ทำการเปิดเมนูที่มีอยู่แล้วและตั้งชื่อ ถ้าไม่มีเมนูอยู่ฟังก์ชันจะทำการสร้างเมนูว่างขึ้นมาใหม่ เมนูใหม่นี้จะถูกเก็บลงในฐานข้อมูล เมื่อตัวเมนูถูกปล่อยออกจาหน่วยความจำ
VCMDMC_OPEN_EXISTING	ทำการเปิดเมนูที่มีอยู่แล้ว ถ้าไม่มีเมนูอยู่ฟังก์ชันจะทำการรีเทิร์นค่า Error

ตารางที่ 2.2 แสดงค่าของ flags ของคำสั่ง MenuCreate

เมื่อแอพลิเคชันทำการสร้างเมนูเสียงขึ้นมาโดยใช้ Methods MenuCreate ซึ่งค่าของ VCMDMC_CREATE_NEW, VCMDMC_CREATE_ALWAYS, VCMDMC_OPEN_ALWAYS ตัวคำสั่งเสียงจะทำการเก็บเมนูใหม่ลงในฐานข้อมูล เมื่อแอพลิเคชันทำการเริ่มต้นใหม่มันจะสามารถรักษาเวลาในการโหลดเมนูจากฐานข้อมูลแทนที่จะสร้างเมนูขึ้นมาใหม่

แอพลิเคชันสามารถสร้างตัวเมนูเสียงโดยการโหลดเมนูเสียงที่มีอยู่แล้วจากฐานข้อมูลของเมนูเสียงหรือทำการสร้างเมนูเสียงขึ้นใหม่ เมนูเสียงตัวหนึ่งจะไม่ต้องถูกเก็บลงในฐานข้อมูล แอพลิเคชันสามารถเมนูเสียงเป็นการชั่วคราวโดยการเซ็ต dwFlags กับ VCMDMC_CREATE_TEMP เมนูเสียงชั่วคราวนี้จะอยู่ไปเรื่อย ๆ จนกว่าจถูกปล่อยออกจากหน่วยความจำ

แอพลเคชันสามารถสร้างตัวเมนูเสียงได้มากกว่าหนึ่งโดยทำเหมือนเดิม สำหรับตัวอย่างหนึ่งแอพลิเคชันต้องทำการระบุเมนูของแอพลิเคชันอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Methods ที่ใช้การเขียนโปรแกรมได้แก่

1. Activate
2. AddCommand
3. Deactivate
4. GetCommand
5. ReleaseMenu

Activate

รูปแบบคำสั่ง

Activate(Menu As Long)

คำอธิบาย

เมนูเสียงจะอยู่ในสถานะพร้อมที่จะรับคำสั่ง ดังนั้นมันจะสามารถรู้จำคำสั่งที่ป้อนเข้ามาได้ เมนูเสียงถูกสร้างเพียงครั้งเดียวและมีหลาย ๆ คำสั่ง มันจะต้องพร้อมก่อนที่จะทำการรู้จำคำสั่ง เมื่อเมนูเสียงพร้อม ตัวเช็คเมนูจะเช็คการแมทช์ ของคำสั่งที่รับเข้ามา เมื่อไรก็ตามที่ตัวประมวลผลออกเสียงจากคำสั่งเสียงจะไม่สามารถเช็คเมนูได้เมื่อมันไม่พร้อม

พารามิเตอร์	คำบรรยาย
Menu	คำสั่งเสียง

ตารางที่ 2.3 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ Activate

AddCommand

รูปแบบคำสั่ง

AddCommand(Menu As Long, id As Long, command As String, description As String, category As String, flags As Long, action As String)

คำอธิบาย

เป็นการเพิ่มคำสั่งกับเมนูที่ถูกสร้างโดย MenuCreate คำสั่งจะถูกเพิ่มเข้าไปต่อท้ายคำสั่งที่มีอยู่แล้วในเมนู

คำสั่งจะเป็นหมายเลขโดยเรียงลำดับจาก 1 ถึง n คำสั่งใหม่จะถูกเพิ่มเข้าไปในส่วนท้ายของเมนู ดังนั้นคำสั่งแรกที่ถูกเพิ่มเข้าไปจะมีหมายเลขเป็น 1+n คำสั่งเสียงจะมีการสนับสนุนการแสดงผลแบบลำดับรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับผลที่ดีที่สุดนั้นจะต้องทำการให้เมนูเสียงอยู่สภาวะไม่พร้อมที่ปฏิบัติการก่อนที่เรียกใช้ AddCommand อีกอย่างหนึ่งก็คือ เมนูจะต้อง deactivate, recompiled, และ reactivated ก่อนที่จะ AddCommand ถ้าเมนูไม่พร้อมที่จะปฏิบัติการอยู่แล้วเมื่อ AddCommand ถูกเรียก ตัวเมนูจะไม่ recompiled จนกว่าแอพลิเคชันจะอยู่ในสภาวะพร้อมที่จะปฏิบัติการอีกครั้ง

ถ้าคำสั่งข้อความประกอบด้วยลำดับรายชื่อ สามารถเรียกใช้ ListSet ทำการเซ็ตคำพูดที่ผู้ใช้สามารถใช้แทนรายชื่อเมื่อกำลังพูดคำสั่ง

พารามิเตอร์	คำบรรยาย
Menu	เมนูเสียง
Id	คำสั่งที่เหมือนกัน แอพลิเคชันสามารถระบุคำสั่งที่เหมือนกันแต่ละคำสั่งเสียงในเมนู คำสั่งที่เหมือนกันแอพลเคชันจะกำหนดค่าพิเศษของคำสั่งภายในเมนู พารามิเตอร์นี้สามารถแก้ไขคำสั่งที่เหมือนกัน หรือมันสามารถถูกใช้สำหรับเหตุการณ์เมื่อแอพลิเคชันรับเหตุการณ์แอพลิเคชันจะสามารถพิจารณาคำสั่งที่เหมือนกันในการตัดสินใจคำสั่งที่เป็นส่วนสำคัญของเหตุการณ์
Command	คำสั่งเสียงที่เป็นข้อความ ทุก ๆ คำสั่งเสียงจะต้องมีคำสั่งข้อความ เช่น “Open a file.” ในระหว่างกระบวนการรู้จำเสียงจะทำการแปลสัญญาณออดิโอที่รับเข้ามาไปเป็นข้อความและเปรียบเทียบมันกับคำสั่งข้อความในเมนูเสียงที่พร้อมที่จะทำงาน ภายในคำสั่งข้อความ จะมีสัญลักษณ์ที่มีความหมายพิเศษดังต่อไปนี้ < > แสดงรายชื่อของคำหรือถ้อยคำโดยสามารถพูดเป็นคำสั้น ๆ สำหรับตัวอย่าง คำสั่งข้อความ “Send mail to <name>” บรรจุรายชื่อที่ถูกเรียก “name” ซึ่งการเพิ่มคำพูดในรายการจะใช้ Method ListSet {} ถูกสำรองไว้ใช้ในอนาคต []ถูกสำรองไว้ใช้ในอนาคต ตัวรู้จำคำสั่งจะสนับสนุนสัญลักษณ์ที่ใช้แมนคำสั่งข้อความ การใช้สัญลักษณ์จะอนุญาตให้ผู้ใช้พูดคำบางคำในระหว่างคำสำคัญในถ้อยคำนั้น สำหรับตัวอย่าง ถ้าแอพลิเคชันนิยามคำสั่งเสียงของ “{wild-card} mail {wild-card} name”, ผู้ใช้สามารถพูด ,”Send mail to Fred,” “I want to mail Fred” หรือ “Mail, yes,I want it to go to Fred.”
Description	เป็นข้อความที่บอกถึงการกระทำของแอพลิเคชันที่ตอบสนองต่อคำสั่ง คำสั่งนี้จะบอกผู้ใช้ในจุดประสงค์ของคำสั่ง แอพลิเคชันจะเป็นตัวแสดงความหมายของคำสั่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

category	ข้อความที่บอกถึงลำดับชั้นซึ่งคำสั่งนี้จะเป็นส่วนหนึ่งของคำสั่งในเมนูเสียงจะทำการรวบรวมความแตกต่างของลำดับชั้นจะช่วยให้ผู้ใช้เลือกตลอดทั้งรายการของคำสั่งได้อย่างง่ายดาย ซึ่งมันจะมีแนวคิดคล้ายกับเมนูวินโดว์ซึ่งจะรวมเอาคำสั่งต่าง ๆ ภายใต้ชื่อเมนู เช่น "File," "Edit," "View," และอื่น ๆ สำหรับผลที่ดีที่สุดจะต้องใช้ การแบ่งลำดับชั้น 20 หรือ น้อยกว่านี้
flags	Flags จะแสดงข้อมูลข่าวสารเกี่ยวกับคำสั่ง ค่าต่าง ๆ ดูในตารางที่ 2.5
action	เป็นบล็อกของข้อมูลที่ถูกส่งไปให้แอพลิเคชันเมื่อพูดคำสั่ง ข้อมูลพร้อมที่จะให้แอพลิเคชันตีความหมายซึ่งข้อมูลจะรวมไปกับคำสั่งในเมนูเสียง ตัวจัดการจะส่งข้อมูลไปที่แอพลิเคชันเมื่อคำสั่งถูกการรู้จำเกิดขึ้น ข้อมูลที่พร้อมนี้จะมีลักษณะพิเศษ ดังนั้นแอพลิเคชันต้องรู้ว่าจะทำอย่างไรกับคำสั่งเมื่อมีการสั่ง ข้อมูลจะถูกส่งผ่านไปกับคำสั่งโดยคอนโทรลจะไม่สามารถแปลความหมาย มันขึ้นอยู่กับการตัดสินใจของแอพลิเคชัน

ตารางที่ 2.4 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ AddCommand

ค่าของ Flags	คำบรรยาย
VCMDCMD_DISABLED_PERM	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง PROPERTEIS EnableMemu ดังนั้นคำสั่งเสียงจะไม่รู้จำมัน คำสั่งจะไม่รวบรวมในเมนูคำสั่ง
VCMDCMD_DISABLED_TEMP	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง METHOD SetCommand คำสั่งจะถูกรวบรวมในเมนูเสียง อย่างไรก็ตามมันจะสามารถ ENABLED อีกครั้ง โดยปราศจากการรวบรวมของเมนู
VCMDCMD_VERIFY	แอพลิเคชันจะต้องพร้อม โดยผู้ใช้จะทำการพิสูจน์คำสั่งก่อนเอามันออก
VCMDCMD_CANTRENAME	บอกถึงคำสั่งอัตโนมัติทั่วไปโดยไม่อนุญาตให้ผู้ใช้เปลี่ยนชื่อคำสั่ง

ตารางที่ 2.5 แสดงค่าของ flags ของคำสั่ง AddCommand

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deactivate

รูปแบบคำสั่ง

Deactivate(Menu As Long)

คำอธิบาย

บอกให้ตัวรู้จำทำการหยุดฟังคำสั่งและคืนทรัพยากรไมโครโฟน ซาวด์การ์ด ถ้าผู้ใช้ยังออกคำสั่งและตัวรู้จำไม่ทำตามจะเกิด Error ขึ้น

พารามิเตอร์	คำบรรยาย
Menu	เมนูคำสั่ง

ตารางที่ 2.6 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ Deactivate

GetCommand

รูปแบบคำสั่ง

GetCommand(Menu As Long, index As Long, command As String, description As String, category As String, flags As Long, action As String)

คำอธิบาย

ข้อมูลข่าวสารเกี่ยวกับคำสั่งที่ได้รับกลับมา

พารามิเตอร์	คำบรรยาย
Menu	ถ้ามีค่าเป็น TRUE จะเป็นการอ่านคำสั่งจากคำสั่งโกลบอลมันจะถูกใช้ในทุก ๆ แอพลิเคชัน ถ้ามีค่าเป็น FALSE มันจะเข้าถึงข้อมูลคำสั่งสำหรับแอพลิเคชัน โดยทั่วไปแอพลิเคชันจะไม่ทำการเปลี่ยนแปลงแก้ไขคำสั่งโกลบอล
index	หมายเลขของคำสั่งที่ได้รับมา หมายเลขของคำสั่งจะเป็นลำดับจาก 1 ถึง n
Command	คำสั่งเสียงที่เป็นข้อความ ทุก ๆ คำสั่งเสียงจะต้องมีคำสั่งข้อความ เช่น “Open a file.” ในระหว่างกระบวนการรู้จำเสียงจะทำการแปลสัญญาณออดิโอที่รับเข้ามาไปเป็นข้อความและเปรียบเทียบมันกับคำสั่งข้อความในเมนูเสียงที่พร้อมที่จะทำงาน ภายในคำสั่งข้อความ จะมีสัญลักษณ์ที่มีความหมายพิเศษดังต่อไปนี้ < > แสดงรายชื่อของคำหรือถ้อยคำโดยสามารถพูดเป็นคำสั้น ๆ สำหรับตัวอย่าง คำสั่งข้อความ “Send mail to <name>” บรรจุรายชื่อที่ถูกเรียก “name” ซึ่งการเพิ่มคำพูดในรายการจะใช้ Method ListSet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	{} ถูกสำรองไว้ใช้ในอนาคต []ถูกสำรองไว้ใช้ในอนาคต ตัวรู้จำคำสั่งจะสนับสนุนสัญลักษณ์ที่ใช้แมนคำสั่งข้อความ การใช้สัญลักษณ์จะอนุญาตให้ผู้ใช้พูดคำบางคำในระหว่างคำสำคัญในถ้อยคำนั้น สำหรับตัวอย่าง ถ้าแอพลิเคชันนิยามคำสั่งเสียงของ "{wild-card} mail {wild-card} name", ผู้ใช้สามารถพูด ,"Send mail to Fred," "I want to mail Fred" หรือ "Mail, yes,I want it to go to Fred."
Description	เป็นข้อความที่บอกถึงการกระทำของแอพลิเคชันที่ตอบสนองต่อคำสั่ง คำสั่งนี้จะบอกผู้ใช้ในจุดประสงค์ของคำสั่ง แอพลิเคชันจะเป็นตัวแสดงความหมายของคำสั่งข้อความ
category	ข้อความที่บอกถึงลำดับชั้นซึ่งคำสั่งนี้จะเป็นส่วนหนึ่งของคำสั่งในเมนูเสียงจะทำการรวบรวมความแตกต่างของลำดับชั้นจะช่วยให้ผู้ใช้เลือกตลอดทั้งรายการของคำสั่งได้อย่างง่ายดาย ซึ่งมันจะมีแนวคิดคล้ายกับเมนูวินโดว์ซึ่งจะรวมเอาคำสั่งต่าง ๆ ภายใต้ชื่อเมนู เช่น "File," "Edit," "View," และอื่น ๆ สำหรับผลที่ดีที่สุดจะต้องใช้ การแบ่งลำดับชั้น 20 หรือ น้อยกว่านี้
flags	Flags จะแสดงข้อมูลข่าวสารเกี่ยวกับคำสั่ง ค่าต่าง ๆ ดูในตารางที่ 2.8
action	เป็นบล็อกของข้อมูลที่ถูกส่งไปให้แอพลิเคชันเมื่อพูดคำสั่ง ข้อมูลพร้อมที่จะให้แอพลิเคชันตีความหมายซึ่งข้อมูลจะรวมไปกับคำสั่งในเมนูเสียง ตัวจัดการจะส่งข้อมูลไปที่แอพลิเคชันเมื่อคำสั่งถูกการรู้จำเกิดขึ้น ข้อมูลที่พร้อมนี้จะมีลักษณะพิเศษ ดังนั้นแอพลิเคชันต้องรู้ว่าจะทำอย่างไรกับคำสั่งเมื่อมีการสั่ง ข้อมูลจะถูกส่งผ่านไปกับคำสั่งโดยคอนโทรลจะไม่สามารถแปลความหมาย มันขึ้นอยู่กับการตัดสินใจของแอพลิเคชัน

ตารางที่ 2.7 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ GetCommand

ค่าของ Flags	คำบรรยาย
VCMDCMD_DISABLED_PERM	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง PROPERTEIS EnableMenu ดังนั้นคำสั่งเสียงจะไม่รู้จำมัน คำสั่งจะไม่รวบรวมในเมนูคำสั่ง
VCMDCMD_DISABLED_TEMP	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง METHOD SetCommand คำสั่งจะถูกรวบรวมในเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

	เสียง อย่างไรก็ตามมันจะสามารถ ENABLED อีกครั้ง โดยปราศจากการรวบรวมของเมนู
VCMDCMD_VERIFY	แอพลิเคชันจะต้องพร้อม โดยผู้ใช้จะทำการพิสูจน์คำสั่งก่อนเอามันออก
VCMDCMD_CANTRENAME	บอกถึงคำสั่งอัตโนมัติทั่วไปโดยไม่อนุญาตให้ผู้ใช้เปลี่ยนชื่อคำสั่ง

ตารางที่ 2.8 แสดงค่าของ flags ของคำสั่ง GetCommand

ReleaseMenu

รูปแบบคำสั่ง

ReleaseMenu(Menu As Long)

คำอธิบาย

ปล่อยเมนูเสียงออกจากหน่วยความจำ ซึ่งสามารถเรียกคำสั่ง ReleaseMenu สำหรับทุก ๆ เมนูคำสั่งที่สร้างขึ้นได้

พารามิเตอร์	คำบรรยาย
Menu	เมนูคำสั่ง

ตารางที่ 2.9 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ ReleaseMenu

Events ที่ใช้การเขียนโปรแกรมได้แก่

1. CommandRecognize

CommandRecognize

รูปแบบคำสั่ง

CommandRecognize(ID As Long, CmdName As String, Flags As Long, Action As String, NumLists As Long, ListValues As String, command As String)

คำอธิบาย

เหตุการณ์นี้จะเกิดขึ้นเมื่อคำสั่งที่พูดเข้ามาถูกรู้จำ พร้อมด้วยเหตุการณ์แอพลิเคชันจะรับข้อความของคำพูดและข้อมูลซึ่งจะถูกป้อนให้แอพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42168

พารามิเตอร์	คำบรรยาย
ID	เช่นเดียวกับคำสั่งที่ถูกรู้จำ
CmdName	ชื่อของเมนูซึ่งบรรจุชื่อของคำสั่งที่ถูกรู้จำ
Flags	VCMDCMD_VERIFY ถ้าแอพลิเคชันต้องการพิสูจน์จากผู้ใช้หรือไม่มีผลถ้าการพิสูจน์นั้นถูกปฏิเสธ ความต้องการในการพิสูจน์นั้นแอพลิเคชันจะแสดงไดอะล็อกบ็อกซ์ขึ้นมา แอพลิเคชันจะทำการพิสูจน์สำหรับทำลายหรือกลับไม่ได้ของคำสั่ง เช่น “Format disk”
Action	ข้อความที่บรรจุข้อมูลมากับคำสั่งที่ถูกรู้จำ
NumLists	หมายเลขของรายการที่ถูกรู้จำ ถ้าคำสั่งไม่ได้บรรจุรายการอะไรมาเลย พารามิเตอร์นี้จะเป็นศูนย์
ListValues	รายการของหนึ่งข้อความหรือมากกว่าที่ถูกแยก สำหรับตัวอย่าง ถ้าคำสั่งเป็น “Set the time to number AM or PM,” พารามิเตอร์จะเป็น “Ten|PM”
command	คำสั่งข้อความสำหรับคำสั่งที่ถูกรู้จำ

ตารางที่ 2.10 แสดงความหมายของค่าพารามิเตอร์ต่าง ๆ ของ CommandRecognize

ผู้ใช้จะไม่ใช้ข้อความภายในของคำสั่งที่เหมือนกับคำสั่งที่ถูกรู้จำ แทนที่จะใช้ข้อมูลใน Action หรือ ข้อมูลที่เหมือนกันใน ID ในการตัดสินใจกับคำสั่งที่ถูกรู้จำ คำสั่งข้อความจะไม่บรรจุข้อความที่มีลักษณะเฉพาะ เพราะว่ามันเป็นไปได้สำหรับผู้ใช้ที่จะแก้ไขข้อความสำหรับคำสั่งสำหรับแอพลิเคชันที่ใช้ไมโครซอฟท์วอยซ์หรืออื่น ๆ

ถ้ามี 2 หรือมากกว่าเมนูคำสั่งที่บรรจุคำพูดและตัวรู้จำคำพูด ตัวจัดการจะทำการเรียก CommandRecognize สำหรับหนึ่งเมนูหรือ CommandOther สำหรับอื่น ๆ ตัวจัดการจะทำการตัดสินใจซึ่งเหตุการณ์จะเรียกเมนูอื่น ๆ แอพลิเคชันจะไม่สามารถตัดสินใจเหตุการณ์ที่ถูกเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ฮาร์ดแวร์ (Hardware)

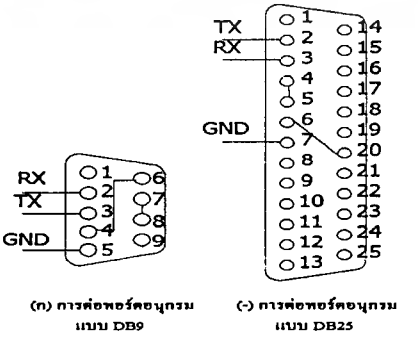
3.1 การทำงานของบอร์ดอินเตอร์เฟส

บอร์ดอินเตอร์เฟสจะประกอบด้วย 3 ส่วนคือส่วนที่หนึ่งเป็นส่วนที่ติดต่อกับพอร์ตอนุกรม RS-232 ใช้ไอซีเบอร์ MAX 232 ทำหน้าที่เป็นบัฟเฟอร์ให้กับพอร์ตอนุกรม RS-232 และจะเปลี่ยนระดับข้อมูลจากระดับแรงดัน 5 โวลท์ ไปเป็นระดับแรงดัน +10,-10 โวลท์ ในมาตรฐานการส่งข้อมูลแบบ RS-232C และในทางกลับกันก็จะทำหน้าที่เปลี่ยนระดับแรงดันมาตรฐาน RS-232 C ให้เป็นระดับแรงดัน 5 โวลท์โดยที่ภายใน IC1 จะมีวงจรทวีแรงดันและวงจรอินเวอร์เตอร์ซึ่งจะต้องต่อตัวเก็บประจุ C1,C2,C3 และ C4 ไว้ภายนอก

ในส่วนที่สองคือส่วนสร้างสัญญาณนาฬิกาโดยเราใช้ IC2 เบอร์ 74HCT4060 เพื่อเป็นตัวกำหนดค่าช่วงเวลา BAUD RATE ให้กับตัว UART เบอร์ 6402 เริ่มดูวงจรจากส่วนสร้างสัญญาณนาฬิกาคริสตอลค่า 2.4576 MHz สร้างความถี่ผ่านไอซีเบอร์ 74HCT4060 เพื่อหารความถี่และเลือกความถี่เพื่อให้ได้ค่า BAUD RATE ตรงกันกับที่คอมพิวเตอร์ ส่งมาซึ่งค่าที่หารออกมาจะมีค่า BAUD RATE ต่างๆ ที่ใช้งานกันบ่อยๆ คือ ค่า 9600, 4800 ,2400, 1200 บิตต่อวินาที ซึ่งปกติที่ใช้กันอยู่ก็คือ 9600 บิตต่อวินาที ดังนั้นเวลาใช้งานก็จัมพ์ SK ไว้ที่ 9600 บิตต่อวินาทีได้เลย

ในการทำงานส่วนที่สามใช้ IC3 เบอร์ UART 6402 ซึ่งจะทำหน้าที่แปลงการติดต่อข้อมูลแบบอนุกรมไปเป็นขนานและสามารถแปลงข้อมูลแบบขนานไปเป็นแบบอนุกรมได้ โดยจะใช้วิธีการส่งแบบอะซิงโครนัส

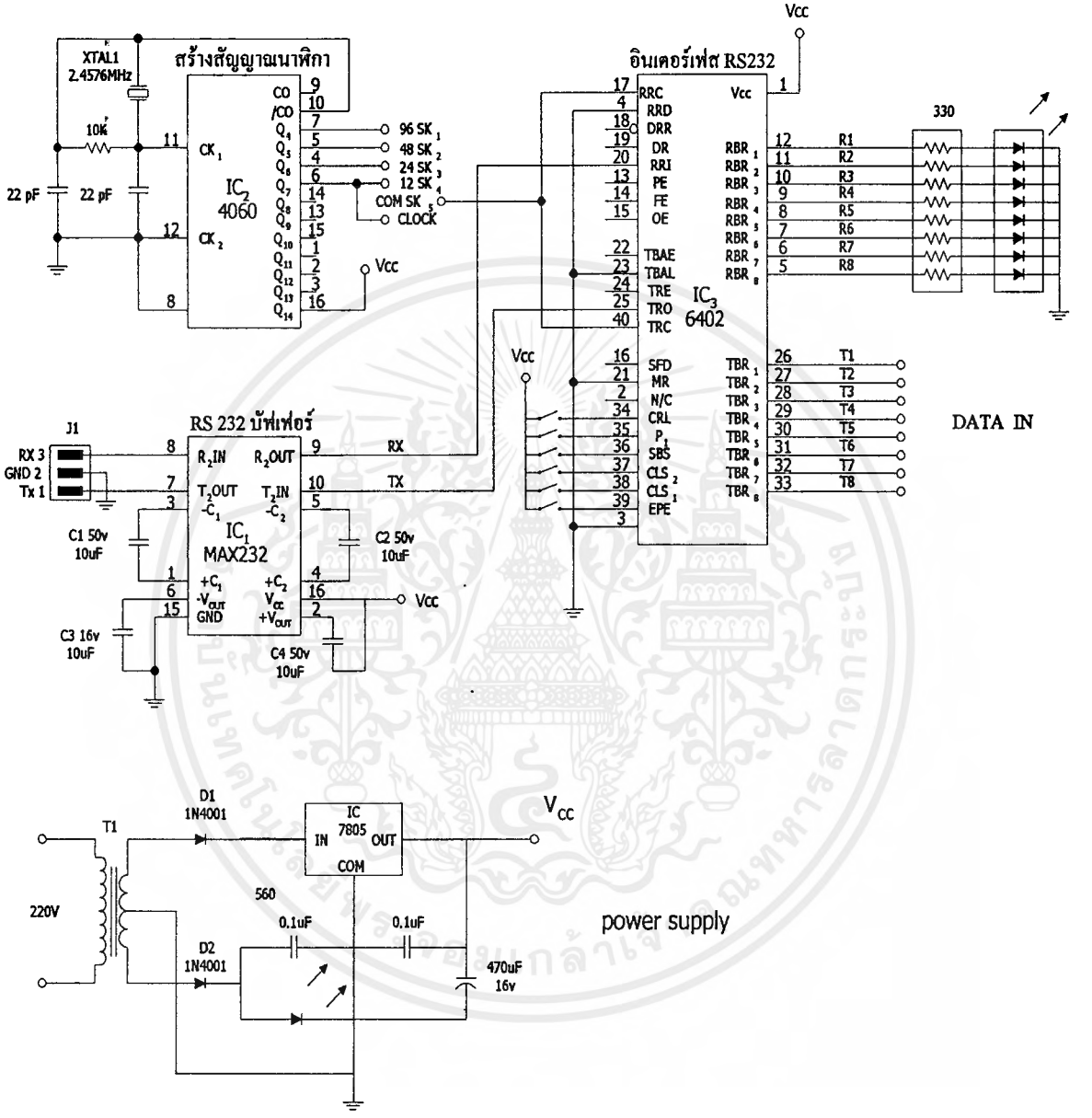
เมื่อเรารวมส่วนต่างๆ เข้าด้วยกันจะมีการทำงานดังนี้คือเราจะใช้ UART 6402 โดยการแปลงข้อมูลแบบอนุกรมให้เป็นแบบขนาน โดยจะรับข้อมูลที่ขา 20 แล้วทำการแปลงข้อมูลอนุกรมเป็นขนานออกแสดงผลที่ขา 5-12 โดยจะแสดงผลโดย LED ทั้ง 8 ตัว ซึ่งขึ้นอยู่กับว่าข้อมูลที่ได้มานั้นที่เป็นข้อมูล 8 บิทนั้นจะทำให้ LED แสดงผลอย่างไรตามข้อมูลที่รับมา



รูปที่ 3.1 แสดงการต่อพอร์ตอนุกรมแบบ DB9 และ DB25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ขาเข้าของขา 26-33 จะเป็นข้อมูลแบบขนาน โดยเป็นข้อมูลที่ส่งมาจากบอร์ดเพื่อส่งสัญญาณไปที่คอมพิวเตอร์ได้ แล้วทำการจัดเรียงเป็นแบบอนุกรมออกทางขา 25 ผ่านทาง MAX232 เข้าสู่คอมพิวเตอร์

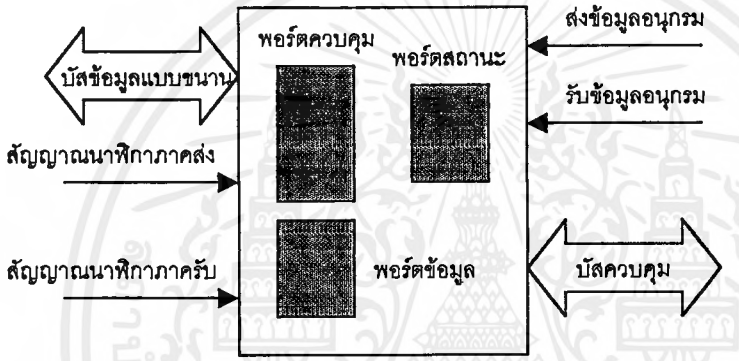


รูปที่ 3.2 แสดงวงจรของบอร์ดอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 UART

UART ย่อมาจาก Universal Asynchronous Receiver Transmiter เป็นระบบการเปลี่ยนการส่งข้อมูลจากแบบอนุกรมไปเป็นแบบขนาน หรือแบบขนานไปเป็นแบบอนุกรม โดยจะใช้การส่งแบบ อะซิงโครนัส ซึ่งไม่ต้องส่งสัญญาณนาฬิกาไปพร้อมกับตัวข้อมูล และจะใช้การส่งบิตสตาร์ต (Start Bit) ซึ่งโดยทั่วไปจะมีค่าเป็น "0" และปิดท้ายด้วยบิตสต็อป (Stop Bit) ซึ่งโดยทั่วไปจะมีค่าเป็น "1" บล็อกไดอะแกรมการทำงานโดยทั่วไปของมันเป็นดังรูปที่ 3.3 โดยมันจะสามารถรับข้อมูลเข้าทั้งแบบอนุกรมและแบบขนาน และส่งออกได้ทั้งแบบอนุกรมและแบบขนาน โดยในการแปลงสัญญาณจะใช้นาฬิกาจากภายนอกเป็นตัวกำหนดค่าคาบเวลาเพื่อเป็นมาตรฐานการส่งหรือมาตรฐานการรับ



รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงานพื้นฐานของ UART

การใช้งาน UART นั้นก็เพื่อจะทำให้การรับส่งข้อมูลสามารถส่งได้ไกลขึ้นโดยจะอาศัยการแปลงข้อมูลจากแบบอนุกรมมาเป็นแบบขนานแล้วใช้มาตรฐานการเชื่อมต่อข้อมูลต่างๆ เช่น RS-232 ส่งข้อมูลออกไปจากนั้นด้านรับก็จะทำการแปลงมาตรฐานการเชื่อมต่อเหล่านี้ออกมาเป็นสัญญาณระดับทีทีแอลให้ UART สามารถรับรู้และแปลงข้อมูลแบบอนุกรมไปเป็นขนานเพื่อใช้งานต่อไป เช่นเดียวกับบอร์ดนี้จะใช้ UART เบอร์ CDP6402 เป็นตัวทำหน้าที่แปลงข้อมูลจากแบบอนุกรมเป็นแบบขนาน

ทำความรู้จัก CDP6402

CDP6402 เป็นไอซี UART แบบ CMOS ที่ทำหน้าที่ได้ทั้งเปลี่ยนข้อมูลแบบอนุกรมไปเป็นขนานและเปลี่ยนข้อมูลแบบขนานไปเป็นแบบอนุกรม โดยจะสามารถใช้กับข้อมูลในรูปแบบต่างๆ กันได้ขึ้นอยู่กับการกำหนดโหมดการทำงานของมันสามารถกำหนดได้ดังตารางที่ 3.1 และสำหรับรายละเอียดภายในตัวมันก็สามารถแสดงได้ดังในรูปที่ 3.2 สามารถแบ่งการทำงานของมันออกเป็น 3 ส่วนด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาควบคุม					ขนาดข้อมูล	พาริตี้บิต	STOP BIT
CLS2	CLS1	PI	EPE	SBS			
L	L	L	L	L	5	คี่	1
L	L	L	L	H	5	คี่	1.5
L	L	L	H	L	5	คู่	1
L	L	L	H	H	5	คู่	1.5
L	L	H	X	L	5	ไม่ใช้	1
L	L	H	X	H	5	ไม่ใช้	1.5
L	H	L	L	L	6	คี่	1
L	H	L	L	H	6	คี่	2
L	H	L	H	L	6	คู่	1
L	H	L	H	H	6	คู่	2
L	H	H	X	L	6	ไม่ใช้	1
L	H	H	X	H	6	ไม่ใช้	2
H	L	L	L	L	7	คี่	1
H	L	L	L	H	7	คี่	2
H	L	L	H	L	7	คู่	1
H	L	L	H	H	7	คู่	2
H	L	H	X	L	7	ไม่ใช้	1
H	L	H	X	H	7	ไม่ใช้	2
H	H	L	L	L	8	คี่	1
H	H	L	L	H	8	คี่	2
H	H	L	H	L	8	คู่	1
H	H	L	H	H	8	คู่	2
H	H	H	X	L	8	ไม่ใช้	1
H	H	H	X	H	8	ไม่ใช้	2

ตารางที่ 3.1 แสดงการกำหนดโหมดเพื่อควบคุมการทำงานของไอซี 6402

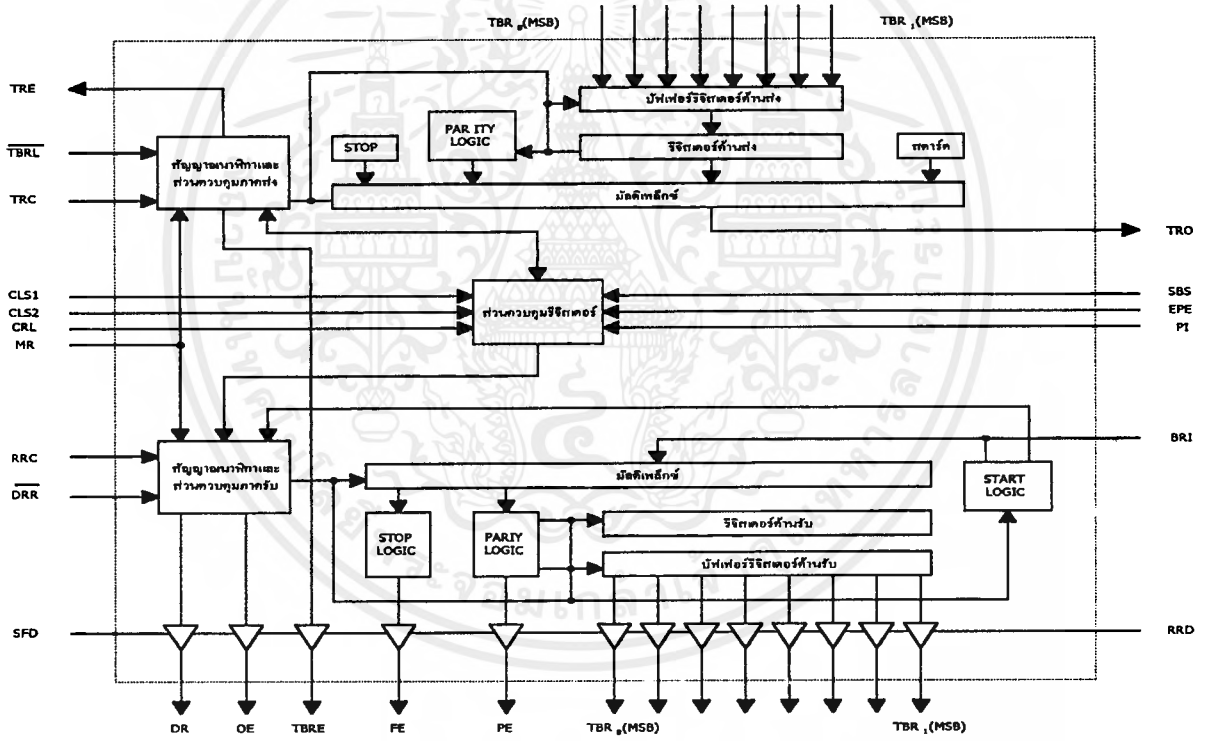
ส่วนแรกเป็นส่วนการแปลงข้อมูลแบบขนานไปเป็นแบบอนุกรมจะมีขา TER1 ถึง TER8 เป็นขารับข้อมูลแบบขนานเข้ามา ผ่านบัฟเฟอร์รีจิสเตอร์และรีจิสเตอร์ และจะมีส่วนของพาริตี้เพื่อคอยตรวจสอบค่าพาริตี้ด้วย จากนั้นก็จะใช้วงจรมัลติเพล็กซ์ทำการแปลงข้อมูลแบบขนานไปเป็นข้อมูลแบบอนุกรม โดยจะมีส่วนสตาร์ตบิตและสต็อปบิตเพิ่มขึ้นมาด้วย การแปลงข้อมูลนี้จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณนาฬิกาจากขา TRC มาเป็นตัวกำหนดมาตรฐานการส่งข้อมูลและเมื่อแปลงครบ 1 ไบต์ก็จะส่งพัลซ์ออกทางขา TRE สัญญาณที่แปลงได้จะส่งออกทางขา TRO

ส่วนที่สอง เป็นส่วนการแปลงข้อมูลแบบอนุกรมไปเป็นแบบขนาน เริ่มจากรับข้อมูลแบบอนุกรมจากขา RRI ส่วนแรกจะส่งไปที่ START LOGIC เพื่อบอกให้รู้ว่าจะส่งข้อมูลเข้ามาแล้ว จากนั้นก็ส่งข้อมูลเข้าส่วนของมัลติเพล็กซ์เพื่อทำการแปลงข้อมูลให้เป็นแบบขนานไปเก็บไว้ในรีจิสเตอร์และบัฟเฟอร์เป็นข้อมูลแบบขนานออกทางขา RBR1 ถึง RBR8 โดยจะใช้สัญญาณนาฬิกาในการแปลงจากขา RRC

ส่วนที่สาม เป็นส่วนของการกำหนดโหมดการทำงานเพื่อให้ UART แปลงข้อมูลได้ตามมาตรฐานต่างๆ โดยให้ "H" จะหมายถึงสภาวะ HIGH,"L" จะหมายถึง
สถานะ LOW และ "X" จะหมายถึงสามารถทำงานได้ทั้งสองสถานะ

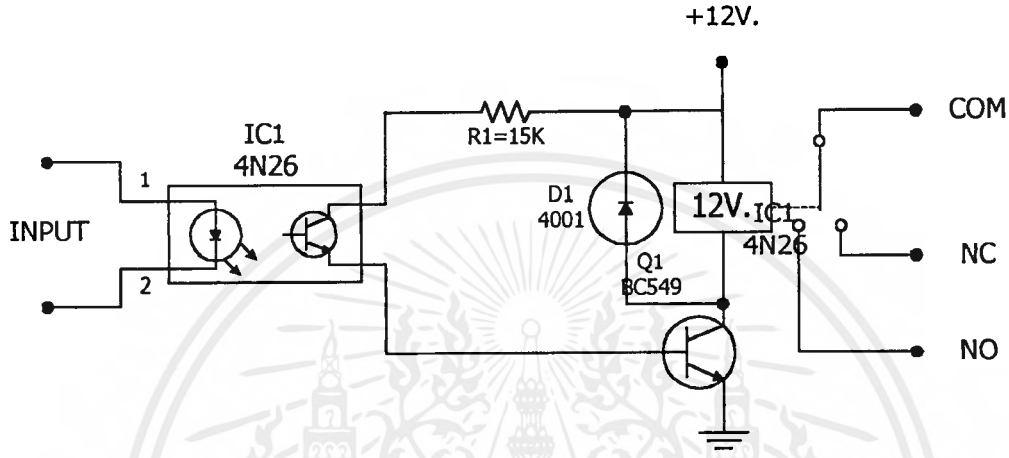


รูปที่ 3.4 บล็อกไดอะแกรมแสดงการทำงานภายในไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนควบคุมอุปกรณ์เครื่องใช้ไฟฟ้า

ส่วนควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าในบ้านนั้นจะใช้ตัวรีเลย์เป็นสวิทช์ในการเปิดปิดอุปกรณ์ซึ่งในที่นี้จะมีทั้งหมด 8 ตัว เพราะสามารถติดต่ออุปกรณ์ได้ทั้งหมด 8 อุปกรณ์ โดยรูปของวงจรขับรีเลย์จะแสดงได้ดังรูปที่ 3.5



รูปที่ 3.5 แสดงวงจรขับรีเลย์โดยใช้โฟโต้ไดโอดเป็นตัวรับอินพุต

การทำงานของวงจร IC1 เบอร์ 4N26 จะทำหน้าที่ในการรับอินพุตจาก CDP6402 เข้ามาที่ขา 1 จะเป็นไฟบวกที่ขา 2 จะเป็น GND จะทำให้โฟโต้ไดโอดทำงานซึ่งจะทำการขับให้ทรานซิสเตอร์ทำงานซึ่งจะเป็นอินพุตต่อไปให้ทรานซิสเตอร์ Q1 ทำงานซึ่งจะเป็นตัวขับให้รีเลย์ทำงาน เพื่อทำการสับสวิทช์ให้มีกระแสไฟไหลเพื่อเปิดอุปกรณ์ที่ทำการต่อไว้ โดยรีเลย์ต้องป้อนไฟเลี้ยง 12 โวลท์ เพื่อขับให้รีเลย์ทำงาน

เมื่อประกอบส่วนต่าง ๆ เข้าด้วยกันแล้ว ก็จะได้วงจรสมบูรณ์ที่สามารถรับข้อมูลจากคอมพิวเตอร์และสามารถแสดงผลออกมาเป็นการเปิด-ปิดอุปกรณ์ และแสดงผลผ่าน LED เพื่อบอกให้รู้ว่าอุปกรณ์ตัวใดทำงานอยู่ในขณะนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

โครงงานการควบคุมอุปกรณ์ไฟฟ้าในบ้านด้วยเสียง (Control of Household Electrical Appliances by sound) แบ่งออกเป็น 2 ส่วนด้วยกันคือ

1. ฮาร์ดแวร์ (Hardware)
2. ซอฟต์แวร์ (Software)

4.1 ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์ประกอบไปด้วย 2 ส่วนใหญ่ ๆ คือ ส่วนที่ทำหน้าที่ติดต่อกับคอมพิวเตอร์ในการรับข้อมูลและส่วนที่ทำหน้าที่ในการแสดงผลข้อมูลที่รับมาจากคอมพิวเตอร์โดยข้อมูลที่รับมานั้นจะเป็นข้อมูลอนุกรมขนาด 8 บิต ซึ่งถูกส่วนที่ทำหน้าที่ติดต่อกับคอมพิวเตอร์ทำการแปลงข้อมูลให้เป็นแบบขนานแล้วส่งไปยังส่วนแสดงผล

4.1.1 การทอดสอบฮาร์ดแวร์

ในการทดสอบฮาร์แวร์นั้นจะทำการทดสอบโดยการส่งข้อมูลจากคอมพิวเตอร์ไปที่ฮาร์ดแวร์แล้วฮาร์ดแวร์แสดงผลได้ตามที่ต้องการ โดยจะต้องมีโปรแกรมในการทดสอบฮาร์ดแวร์แบบง่าย ๆ เพื่อให้แน่ใจว่าฮาร์ดแวร์ที่ทำขึ้นนั้นสามารถใช้งานได้จริงตามที่ต้องการ โดยส่วนของซอฟต์แวร์ที่ทดลองฮาร์ดแวร์นั้นมีดังต่อไปนี้

```
Option Explicit
Private a As String
Private b As String
Private c As String
Private d As String

Private Sub Command1_Click()
   End
End Sub

Private Sub Command2_Click()
   If Text1.Text = "" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MsgBox "Please Input Data .....", vbOKOnly + vbCritical, "Error"
End If
a = Text1.Text
If a = "1" Then
   b = "1"
   d = Val(c) Or Val(b)
   c = d

ElseIf a = "2" Then
   b = "2"
   d = Val(c) Or Val(b)
   c = d

ElseIf a = "3" Then
   b = "4"
   d = Val(c) Or Val(b)
   c = d

ElseIf a = "4" Then
   b = "8"
   d = Val(c) Or Val(b)
   c = d

ElseIf a = "5" Then
   b = "16"
   d = Val(c) Or Val(b)
   c = d

ElseIf a = "6" Then
   b = "32"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    d = Val(c) Or Val(b)
    c = d

  ElseIf a = "7" Then
    b = "64"
    d = Val(c) Or Val(b)
    c = d

  ElseIf a = "8" Then
    b = "128"
    d = Val(c) Or Val(b)
    c = d

  Else
     MsgBox "Please Input Number Between 0-8.", vbOKOnly + vbCritical, "Error !!!"
  End If
  ' เซ็ตพอร์ต COM1.
  MSComm1.CommPort = 1
  ' เซ็ตค่าในการรับส่งข้อมูลระหว่าง Hardware กับ Software
  '9600 baud, no parity, 8 data, and 1 stop bit.
  MSComm1.Settings = "9600,N,8,1"
  ' เปิดพอร์ตให้พร้อมที่จะรับส่งข้อมูล.
  MSComm1.PortOpen = True
  ' ส่งข้อมูลออกทางพอร์ตซึ่งเป็นข้อมูลขนาด 8 บิต.
  MSComm1.Output = Chr$(d)
  ' ปิดพอร์ต.
  MSComm1.PortOpen = False
End Sub

Private Sub Command3_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' เซ็ตพอร์ต COM1.
MSComm1.CommPort = 1
' เซ็ตค่าในการรับส่งข้อมูลระหว่าง Hardware กับ Software
'9600 baud, no parity, 8 data, and 1 stop bit.
MSComm1.Settings = "9600,N,8,1"
' เปิดพอร์ตให้พร้อมที่จะรับส่งข้อมูล.
MSComm1.PortOpen = True
' ส่งข้อมูลออกทางพอร์ตซึ่งเป็นข้อมูลขนาด 8 บิต.
MSComm1.Output = Chr$(0)
' ปิดพอร์ต.
MSComm1.PortOpen = False
b = "0"
c = "0"
d = "0"
End Sub

Private Sub Command4_Click()
If Text1.Text = "" Then
    MsgBox "Please Input Data .....", vbOKOnly + vbCritical, "Error"
  End If
  a = Text1.Text
  If a = "1" Then
    b = "254"
    d = Val(c) And Val(b)
    c = d

  ElseIf a = "2" Then
    b = "252"
    d = Val(c) And Val(b)
    c = d
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ElseIf a = "3" Then
   b = "251"
   d = Val(c) And Val(b)
   c = d

ElseIf a = "4" Then
   b = "247"
   d = Val(c) And Val(b)
   c = d

ElseIf a = "5" Then
   b = "239"
   d = Val(c) And Val(b)
   c = d

ElseIf a = "6" Then
   b = "223"
   d = Val(c) And Val(b)
   c = d

ElseIf a = "7" Then
   b = "191"
   d = Val(c) And Val(b)
   c = d

ElseIf a = "8" Then
   b = "127"
   d = Val(c) And Val(b)
   c = d
```

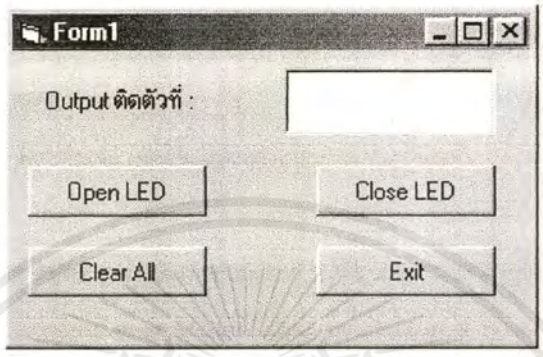
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Else
     MsgBox "Please Input Number Between 0-8.", vbOKOnly + vbCritical, "Error !!!"
  End If
  ' เซ็ตพอร์ต COM1.
  MSComm1.CommPort = 1
  ' เซ็ตค่าในการรับส่งข้อมูลระหว่าง Hardware กับ Software
  '9600 baud, no parity, 8 data, and 1 stop bit.
  MSComm1.Settings = "9600,N,8,1"
  ' เปิดพอร์ตให้พร้อมที่จะรับส่งข้อมูล.
  MSComm1.PortOpen = True
  ' ส่งข้อมูลออกทางพอร์ตซึ่งเป็นข้อมูลขนาด 8 บิต.
  MSComm1.Output = Chr$(d)
  ' ปิดพอร์ต.
  MSComm1.PortOpen = False
End Sub

Private Sub Form_Load()
  ' เซ็ตพอร์ต COM1.
  MSComm1.CommPort = 1
  ' เซ็ตค่าในการรับส่งข้อมูลระหว่าง Hardware กับ Software
  '9600 baud, no parity, 8 data, and 1 stop bit.
  MSComm1.Settings = "9600,N,8,1"
  ' เปิดพอร์ตให้พร้อมที่จะรับส่งข้อมูล.
  MSComm1.PortOpen = True
  ' ส่งข้อมูลออกทางพอร์ตซึ่งเป็นข้อมูลขนาด 8 บิต.
  MSComm1.Output = Chr$(0)
  ' ปิดพอร์ต.
  MSComm1.PortOpen = False
  b = "0"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
c = "0"
d = "0"
End Sub
```



รูปที่ 4.1 แสดงเมนูของโปรแกรมทดสอบฮาร์ดแวร์

จากรูปที่ 4.1 เมื่อป้อนอินพุตซึ่งเป็นเลข 1-8 ในช่องรับอินพุตแล้วทำการกดปุ่ม Open LED โปรแกรมจะทำการส่งข้อมูลไปที่ฮาร์ดแวร์ตามที่ต้องการซึ่งที่อุปกรณ์ตัวตรงกับที่ป้อนเข้าไปตามหมายเลขก็จะทำงานซึ่งสามารถตรวจเช็คได้ว่าฮาร์ดแวร์นั้นทำงานหรือไม่ได้ ถ้ากดปุ่ม Close LED อินพุตที่ป้อนให้โปรแกรมก็จะถูกส่งไปที่ฮาร์ดแวร์เพื่อทำการปิดอุปกรณ์นั้นได้ ส่วนปุ่ม Clear All นั้นถ้าถูกกดโดยจะไม่สนใจอินพุตที่ป้อนให้โปรแกรมมันจะส่งข้อมูลไปให้ฮาร์ดแวร์ทำการปิดอุปกรณ์ทุกชนิดที่กำลังเปิดอยู่ ซึ่งเงื่อนไขต่าง ๆ ที่อยู่ในการทดสอบนั้นจะตรงตามความต้องการ คือ ถ้าสั่งให้อุปกรณ์ชนิดใดทำงานแล้วยังไม่สั่งให้หยุดมันจะทำงานต่อไปเรื่อย ๆ จนกว่าจะมีการสั่งงานมาจากคอมพิวเตอร์ โดยที่แม้จะสั่งอุปกรณ์ตัวอื่น ๆ ให้ทำงานก็จะไม่ส่งผลกระทบต่ออุปกรณ์ตัวที่ไม่ได้ถูกสั่งเลย ส่วนปุ่ม Exit เมื่อกดปุ่มนี้แล้วก็จะออกจากโปรแหรมโดยที่จะทำการปิดอุปกรณ์ฮาร์ดแวร์ทุกตัวด้วยเมื่อไม่มีการทำงาน

4.2 ซอฟต์แวร์ (Software)

ซอฟต์แวร์ที่เขียนขึ้นโดยโปรแกรม Visual Basic นั้นจะมีการทำงานเมื่อมีการพูดสั่งงานจากไมโครโฟนซึ่งโปรแกรมจะรับสัญญาณเข้ามาทำการเปรียบเทียบกับคำสั่งที่ป้อนไว้แล้วในโปรแกรมว่าตรงกับคำสั่งใด เมื่อเปรียบเทียบเสร็จแล้วก็จะทำงานตามเงื่อนไขที่ต้องการ คือ เมื่อตรงกับคำสั่งในโปรแกรม เช่น สั่งว่า Open Light โปรแกรมก็จะส่งข้อมูลไปบอกฮาร์ดแวร์ว่าให้ทำการเปิดไฟ เป็นต้น ถ้าคำสั่งนั้นไม่ตรงกับคำสั่งที่มีอยู่โปรแกรมก็จะไม่ส่งข้อมูลใด ๆ ไปที่ฮาร์ดแวร์ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวฮาร์ดแวร์ก็จะไม่มีการเปลี่ยนแปลงไปจากเดิม จนกว่าซอฟต์แวร์จะส่งข้อมูลไปที่ฮาร์ดแวร์เท่านั้น

การทดสอบซอฟต์แวร์นั้นจะทำการทดสอบโดยการพูดตามคำสั่งที่กำหนดโดยโปรแกรม ซึ่งจะเป็นใครก็ได้ที่จะเป็นผู้สั่งงานโดยไม่จำกัดเพศ อายุ เพียงแต่สั่งงานให้ตรงตามคำสั่งที่ตั้งไว้เท่านั้น โดยคำสั่งที่ใช้ในการทดสอบจะมีด้วยกัน 19 คำสั่ง ซึ่งเป็นคำสั่งในการเปิดอุปกรณ์ 9 คำสั่งและปิดอุปกรณ์ 9 คำสั่งดังต่อไปนี้

คำสั่งในการเปิดอุปกรณ์ Open Light, Open TV, Open Door, Open Window, Open Fan, Open Video, Open PC, Open Pump และ Open All

คำสั่งในการปิดอุปกรณ์ Close Light, Close TV, Close Door, Close Window, Close Fan, Close Video, Close PC, Close Pump และ Close All

ทดสอบการเปิดอุปกรณ์ของบุคคลที่ 1

คำสั่ง/ ครั้งที่	Open Light	Open TV	Open Door	Open Window	Open Fan	Open Video	Open PC	Open Pump	Open All
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✗	✓	✓	✓	✓	✗	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	✓	✓	✓	✓	✓	✓	✓	✗	✓
10	✓	✓	✓	✓	✓	✓	✓	✓	✓

ตารางที่ 4.1 ตารางการทดสอบการสั่งเปิดอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบการปิดอุปกรณ์ของบุคคลที่ 1

คำสั่ง/ ครั้งที่	Close Light	Close TV	Close Door	Close Window	Close Fan	Close Video	Close PC	Close Pump	Close All
1	✓	✓	✓	✓	✓	✓	✓	✗	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✗	✗	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	✓	✓	✓	✓	✓	✓	✓	✗	✓
9	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	✓	✓	✓	✓	✓	✓	✓	✓	✓

ตารางที่ 4.2 ตารางการทดสอบการสั่งปิดอุปกรณ์

คำสั่งเปิด อุปกรณ์	Open Light	Open TV	Open Door	Open Window	Open Fan	Open Video	Open PC	Open Pump	Open All
(%) ความถูกต้อง	100	100	90	100	100	100	100	80	100
คำสั่งปิด อุปกรณ์	Close Light	Close TV	Close Door	Close Window	Close Fan	Close Video	Close PC	Close Pump	Close All
(%) ความถูกต้อง	100	100	100	100	100	100	90	70	100

ตารางที่ 4.3 ตารางแสดงเปอร์เซ็นต์ความถูกต้องของบุคคลที่ 1

จากการทดสอบจะเห็นว่าคำสั่งที่ใช้ในการทดสอบนั้นจะมีการออกเสียงที่ต่างกันจึงทำให้การสั่งงานแล้วเกิดการทำงานผิดพลาดนั้นมีน้อย แต่การที่คำสั่ง Open Pump และ Close Pump นั้นมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การผิดพลาดเกิดขึ้นบ่อยเพราะว่าเวลาพูดนั้นจะต้องเน้นการออกเสียงที่ชัดเจนเหมือนเจ้าของภาษา ซึ่งในการเน้นเสียงนี้คนไทยส่วนใหญ่จะไม่ค่อยมีการเน้นเท่าไร ส่วนมากจะออกเสียงตามธรรมชาติมากกว่าอาจทำให้เวลาสั่งงานเกิดความผิดพลาดขึ้นได้

ทดสอบการเปิดอุปกรณ์บุคคลที่ 2

คำสั่ง/ ครั้งที่	Open Light	Open TV	Open Door	Open Window	Open Fan	Open Video	Open PC	Open Pump	Open All
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✗	✓
3	✓	✓	✗	✓	✓	✓	✗	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✗	✓
7	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	✓		✓	✓	✓	✓	✓	✓	✓
10	✓	✓	✓	✓	✓	✓	✓	✓	✓

ตารางที่ 4.4 ตารางการทดสอบการสั่งเปิดอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบการปิดอุปกรณ์ของบุคคลที่ 2

คำสั่ง/ ครั้งที่	Close Light	Close TV	Close Door	Close Window	Close Fan	Close Video	Close PC	Close Pump	Close All
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✗	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✗	✓
7	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	✓	✓	✗	✓	✓	✓	✓	✓	✓
10	✓	✓	✓	✓	✓	✓	✓	✗	✓

ตารางที่ 4.5 ตารางการทดสอบการสั่งปิดอุปกรณ์

คำสั่งเปิด อุปกรณ์	Open Light	Open TV	Open Door	Open Window	Open Fan	Open Video	Open PC	Open Pump	Open All
(%) ความถูกต้อง	100	100	90	100	100	100	90	80	100
คำสั่งปิด อุปกรณ์	Close Light	Close TV	Close Door	Close Window	Close Fan	Close Video	Close PC	Close Pump	Close All
(%) ความถูกต้อง	100	90	90	100	100	100	100	80	100

ตารางที่ 4.6 ตารางแสดงเปอร์เซ็นต์ความถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบของบุคคลที่ 2 นั้นก็มีลักษณะคล้ายกับบุคคลที่ 1 เนื่องจากว่าคำสั่งในการสั่งงานแต่ละคำนั้นมีการออกเสียงที่คล้ายกันน้อยจึงทำให้เวลาสั่งงานมีความถูกต้องมาก แต่ก็จะมีบางคำเท่านั้นที่มีความการออกเสียงคล้ายกัน เช่น Open PC กับ Open TV เป็นต้น

การทดลองสามารถให้ผลออกมาเป็นที่น่าพอใจ แต่ถ้าคำสั่งนั้นมีการออกเสียงที่คล้ายกันหลาย ๆ คำสั่งออกทำให้ผลที่ออกมาคงไม่น่าพอใจเท่าไดนัก เนื่องจากการออกเสียงของบุคคลสั่งงานนั้นจะไม่สามารถพูดได้เหมือนเจ้าของภาษามากนัก ซึ่งถ้าเจ้าของภาษาพูดเองอาจมีความถูกต้องที่สูงยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปปัญหาและข้อเสนอแนะ

5.1 สรุป

การสั่งงานด้วยเสียงพูดโดยใช้ MICROSOFT SPEECH SDK 4.0 ของ VISUAL BASIC ในการจัดการเกี่ยวกับเรื่องเสียงนั้น จะมีความถูกต้องแม่นยำที่สูงขึ้นถ้าคำสั่งเสียงนั้นมีการออกเสียงที่ไม่คล้ายกันมากนัก แต่จะมีความผิดพลาดสูงในกรณีที่คำสั่งเสียงมีการออกเสียงที่คล้ายกัน ซึ่งสำหรับคนไทยนั้นการแก้ไขตรงจุดนี้คงจะต้องเรียนรู้วิธีพูดภาษาอังกฤษและการออกเสียงที่ถูกต้องจึงจะทำให้เกิดความแม่นยำที่สูงขึ้นได้

แต่จากการทดลองพูดสั่งงานนี้ได้ผลที่น่าพอใจ เพราะคำสั่งที่สั่งงานนั้น มีการออกเสียงที่ไม่คล้ายกันเท่าไรนัก ทางด้านฮาร์ดแวร์ก็สามารถแสดงผลได้ตามที่กำหนด คือ สั่งคำสั่งใดไปฮาร์ดแวร์นั้น ๆ ก็จะทำงานตามที่สั่งจะแสดงผิดพลาดก็ต่อเมื่อมีการสั่งงานในคำสั่งที่คล้ายกันเท่านั้นเพราะฮาร์ดแวร์นั้นทำงานตามข้อมูลที่ซอฟต์แวร์เป็นตัวส่ง ข้อมูลมาให้ฮาร์ดแวร์

5.2 ปัญหาที่เกิดขึ้นในการทำงาน

1. การศึกษาหาข้อมูลเกี่ยวกับ MICROSOFT SPEECH SDK 4.0 นั้นทำได้ช้าเนื่องจากต้นฉบับเป็นภาษาอังกฤษและข้อมูลทางด้านอินเตอร์เน็ตนั้นก็มีของ MICROSOFT ที่เดียวเท่านั้น ส่วนทฤษฎีต่าง ๆ ที่ใช้ในการทำงานของ MICROSOFT SPEECH SDK 4.0 นั้นก็ไม่มีการเปิดเผยเพียงแต่มีแค่วิธีใช้งานและ HELP เกี่ยวกับตัวคอนโทรลเท่านั้น
2. อุปกรณ์ฮาร์ดแวร์คือ IC UART CDP6402 นั้นหาซื้อได้ยากเพราะต้องสั่งจากทางต่างประเทศซึ่งทางร้านที่สั่งก็มีเพียง 2-3 ร้านเท่านั้น ถ้าของขาดตลาด ก็จะหาซื้อไม่ได้เลย
3. ราคาของฮาร์ดแวร์มีราคาสูงโดยเฉพาะส่วนของรีเลย์ ซึ่งเป็นวงจรขับรีเลย์โดยใช้ออปโต้เป็นตัวรับอินพุตที่ป้อนให้กับรีเลย์
4. ไมโครโฟนที่ใช้นั้นต้องมีความไวพอสมควรและไม่ก่อให้เกิดเสียงรบกวนมากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการแก้ไข

1. ศึกษาภาษาอังกฤษเพิ่มเติมเพื่อให้สามารถอ่านและแปรตำราภาษาอังกฤษได้ถูกต้องมากขึ้น ปรึกษาผู้ที่มีความรู้ด้าน VISUAL BASIC เพื่อการเขียนโปรแกรมที่ถูกต้อง
2. ควรใช้อุปกรณ์ที่สามารถหาซื้อได้ง่ายแทน เช่น ใช้ไมโครคอนโทรลเลอร์ เป็นต้น
3. ควรศึกษาการออกแบบวงจรเพื่อให้สามารถออกแบบวงจรได้เองและช่วยลดค่าใช้จ่ายลงได้
4. ควรใช้ไมโครโฟนที่ความไวพอสมควรและไม่มีเสียงรบกวนมากนัก ซึ่งราคาของไมโครโฟนก็จะแพงตามไปด้วย

5.4 ข้อเสนอแนะในโครงงาน

ควรจะสามารถสั่งงานด้วยภาษาไทยได้ในอนาคต โดยศึกษา MICROSOFT SPEECH SDK 4.0 และ VISUAL BASIC ให้เข้าใจมากยิ่งขึ้น เพราะเป็นหัวใจในการทำงานของโครงงานนี้ เพื่อทำให้สามารถพัฒนาโปรแกรมให้ทำงานได้ตามวัตถุประสงค์ที่ต้องการมากยิ่งขึ้น และสามารถนำไปใช้ในการประกอบอาชีพได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

Source Code ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOURCE CODE ของ FORM1

```
Public gMymenu As Long
Private buffdata1 As String
Private buffdata2 As String
Private buffdata3 As String

Private Sub UpdateList()
Dim i As Long
Dim tcount As Long
Dim command As String
Dim description As String
Dim category As String
Dim flags As Long
Dim action As String

List1.Clear
tcount = Vcommand1.CountCommands(gMymenu)
For i = 1 To tcount
  Vcommand1.GetCommand gMymenu, i, command, description, category, flags, action
  List1.AddItem command
Next i
End Sub

Private Sub cmdListen_Click()
If cmdListen.Caption = "รอฟังคำสั่ง" Then
   Vcommand1.Activate gMymenu
   cmdListen.Caption = "หยุดฟังคำสั่ง"
   List1.Enabled = True
Else
   Vcommand1.Deactivate gMymenu
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    cmdListen.Caption = "รอฟังคำสั่ง"
    List1.Enabled = False
End If
End Sub

Private Sub cmdExit_Click()
    Vcommand1.Deactivate gMymenu
    Vcommand1.ReleaseMenu gMymenu
    End
End Sub

Private Sub Form_Load()
' เซ็ตค่าเริ่มต้นของตัวพักข้อมูล
buffdata1 = "0"
buffdata2 = "0"
buffdata3 = "0"
 ' เซ็ตพอร์ต COM1 เป็นพอร์ตในการติดต่อกับฮาร์ดแวร์.
MSComm1.CommPort = 1
' เซ็ตค่าในการรับส่งข้อมูลระหว่าง Hardware กับ Software
'9600 baud, no parity, 8 data, and 1 stop bit.
MSComm1.Settings = "9600,N,8,1"
' เปิดพอร์ตให้พร้อมที่จะรับส่งข้อมูล.
   MSComm1.PortOpen = True
' ส่งข้อมูลออกทางพอร์ตซึ่งเป็นข้อมูลขนาด 8 บิต.
MSComm1.Output = Chr$(0)
' ปิดพอร์ต.
MSComm1.PortOpen = False
Vcommand1.Initialized = 1
gMymenu = Vcommand1.MenuCreate(App.EXEName, "state1", 4)
Vcommand1.Enabled = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

' คำสั่งในการเปิดอุปกรณ์ทุกอุปกรณ์

Vcommand1.AddCommand gMymenu, 1, "open all", "when you say" + "open all", "listen list", 0, ""

' คำสั่งในการเปิดอุปกรณ์ต่าง ๆ

Vcommand1.AddCommand gMymenu, 1, "open light", "when you say" + "open light", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "open tv", "when you say" + "open tv", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "open door", "when you say" + "open door", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "open window", "when you say" + "open windwo", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "open fan", "when you say" + "open fan", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "open video", "when you say" + "open video", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "open pc", "when you say" + "open pc", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "open pump", "when you say" + "open pump", "listen list", 0, ""

' คำสั่งในการปิดอุปกรณ์ทุกอุปกรณ์

Vcommand1.AddCommand gMymenu, 1, "close all", "when you say" + "closr all", "listen list", 0, ""

' คำสั่งในการปิดอุปกรณ์ต่าง ๆ

Vcommand1.AddCommand gMymenu, 1, "close light", "when you say" + "close light", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "close tv", "when you say" + "close tv", "listen list", 0, ""

Vcommand1.AddCommand gMymenu, 1, "close door", "when you say" + "close door", "listen list", 0, ""

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Vcommand1.AddCommand gMymenu, 1, "close window", "when you say" + "close window",
"listen list", 0, ""
Vcommand1.AddCommand gMymenu, 1, "close fan", "when you say" + "close fan", "listen list",
0, ""
Vcommand1.AddCommand gMymenu, 1, "close video", "when you say" + "close video", "listen
list", 0, ""
Vcommand1.AddCommand gMymenu, 1, "close pc", "when you say" + "close pc", "listen list",
0, ""
Vcommand1.AddCommand gMymenu, 1, "close pump", "when you say" + "close pump", "listen
list", 0, ""
UpdateList
Vcommand1.Activate gMymenu
End Sub

Private Sub Form_Unload(Cancel As Integer)
Vcommand1.Deactivate gMymenu
Vcommand1.ReleaseMenu gMymenu
End Sub

Private Sub mnuAbout_Click()
   Form3.Show
End Sub

Private Sub mnuExit_Click()
   End
End Sub

Private Sub mnuHelp_Click()
   Form2.Show
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Vcommand1_CommandRecognize(ByVal ID As Long, ByVal CmdName As String,
ByVal flags As Long, ByVal action As String, ByVal NumLists As Long, ByVal ListValues As
String, ByVal command As String)
lblHear.Caption = command
' เช็คคำสั่งในการเปิดอุปกรณ์
If lblHear.Caption = "open light" Then
    buffdata1 = "1"
    buffdata3 = Val(buffdata2) Or Val(buffdata1)
    buffdata2 = buffdata3

  ElseIf lblHear.Caption = "open tv" Then
    buffdata1 = "2"
    buffdata3 = Val(buffdata2) Or Val(buffdata1)
    buffdata2 = buffdata3

  ElseIf lblHear.Caption = "open door" Then
    buffdata1 = "4"
    buffdata3 = Val(buffdata2) Or Val(buffdata1)
    buffdata2 = buffdata3

  ElseIf lblHear.Caption = "open window" Then
    buffdata1 = "8"
    buffdata3 = Val(buffdata2) Or Val(buffdata1)
    buffdata2 = buffdata3

  ElseIf lblHear.Caption = "open fan" Then
    buffdata1 = "16"
    buffdata3 = Val(buffdata2) Or Val(buffdata1)
    buffdata2 = buffdata3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ElseIf lblHear.Caption = "open video" Then
   buffdata1 = "32"
   buffdata3 = Val(buffdata2) Or Val(buffdata1)
   buffdata2 = buffdata3

ElseIf lblHear.Caption = "open pc" Then
   buffdata1 = "64"
   buffdata3 = Val(buffdata2) Or Val(buffdata1)
   buffdata2 = buffdata3

ElseIf lblHear.Caption = "open pump" Then
   buffdata1 = "128"
   buffdata3 = Val(buffdata2) Or Val(buffdata1)
   buffdata2 = buffdata3

ElseIf lblHear.Caption = "open all" Then
   buffdata1 = "255"
   buffdata3 = Val(buffdata2) Or Val(buffdata1)
   buffdata2 = buffdata3

' เช็คคำสั่งในการปิดอุปกรณ์
ElseIf lblHear.Caption = "close light" Then
   buffdata1 = "254"
   buffdata3 = Val(buffdata2) And Val(buffdata1)
   buffdata2 = buffdata3

ElseIf lblHear.Caption = "close tv" Then
   buffdata1 = "252"
   buffdata3 = Val(buffdata2) And Val(buffdata1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    buffdata2 = buffdata3

ElseIf lblHear.Caption = "close door" Then
    buffdata1 = "251"
    buffdata3 = Val(buffdata2) And Val(buffdata1)
    buffdata2 = buffdata3

ElseIf lblHear.Caption = "close window" Then
    buffdata1 = "247"
    buffdata3 = Val(buffdata2) And Val(buffdata1)
    buffdata2 = buffdata3

ElseIf lblHear.Caption = "close fan" Then
    buffdata1 = "239"
    buffdata3 = Val(buffdata2) And Val(buffdata1)
    buffdata2 = buffdata3

ElseIf lblHear.Caption = "close video" Then
    buffdata1 = "223"
    buffdata3 = Val(buffdata2) And Val(buffdata1)
    buffdata2 = buffdata3

ElseIf lblHear.Caption = "close pc" Then
    buffdata1 = "191"
    buffdata3 = Val(buffdata2) And Val(buffdata1)
    buffdata2 = buffdata3

ElseIf lblHear.Caption = "close pump" Then
    buffdata1 = "127"
    buffdata3 = Val(buffdata2) And Val(buffdata1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    buffdata2 = buffdata3

  ElseIf lblHear.Caption = "close all" Then
    buffdata1 = "0"
    buffdata3 = Val(buffdata2) And Val(buffdata1)
    buffdata2 = buffdata3
  Else
    MsgBox "คำสั่งไม่ถูกต้อง.....!!!", vbOKOnly + vbCritical, "คำสั่งผิดพลาด"
End If
' เซ็ตพอร์ต COM1.
MSComm1.CommPort = 1
' เซ็ตค่าในการรับส่งข้อมูลระหว่าง Hardware กับ Software
'9600 baud, no parity, 8 data, and 1 stop bit.
MSComm1.Settings = "9600,N,8,1"
' เปิดพอร์ตให้พร้อมที่จะรับส่งข้อมูล.
MSComm1.PortOpen = True
' ส่งข้อมูลออกทางพอร์ตซึ่งเป็นข้อมูลขนาด 8 บิต.
MSComm1.Output = Chr$(buffdata3)
' ปิดพอร์ต.
MSComm1.PortOpen = False
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1 ภาคผนวก ก แสดงเมนูหลักของโปรแกรม

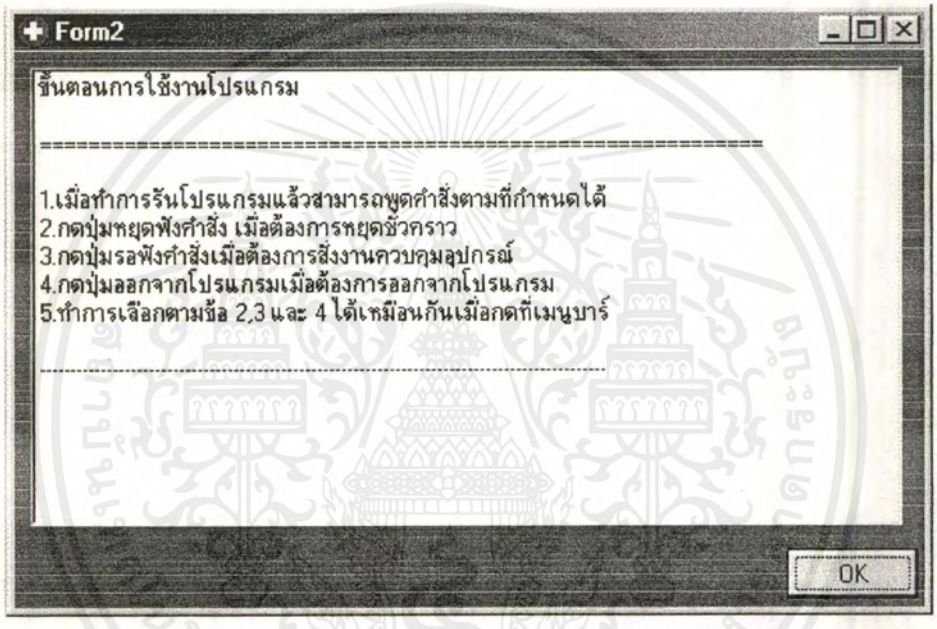
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOURCE CODE ของ FORM2

```
Private Sub cmdOK2_Click()
   Unload Me
End Sub

Private Sub Form_Load()
   lstHelp.Clear
   lstHelp.AddItem "ขั้นตอนการใช้งานโปรแกรม"
   lstHelp.AddItem ""
   lstHelp.AddItem "========================================================== &_
   ==========="
   lstHelp.AddItem ""
   lstHelp.AddItem "1.เมื่อทำการรันโปรแกรมแล้วสามารถพูดคำสั่งตามที่กำหนดได้"
   lstHelp.AddItem "2.กดปุ่มหยุดฟังคำสั่ง เมื่อต้องการหยุดชั่วคราว"
   lstHelp.AddItem "3.กดปุ่มรอฟังคำสั่งเมื่อต้องการสั่งงานควบคุมอุปกรณ์"
   lstHelp.AddItem "4.กดปุ่มออกจากโปรแกรมเมื่อต้องการออกจากโปรแกรม"
   lstHelp.AddItem "5.ทำการเลือกตามข้อ 2,3 และ 4 ได้เหมือนกันเมื่อกดที่เมนูบาร์"
   lstHelp.AddItem ""
   lstHelp.AddItem "---------------------------------------------------------------------- &_
   --------"
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

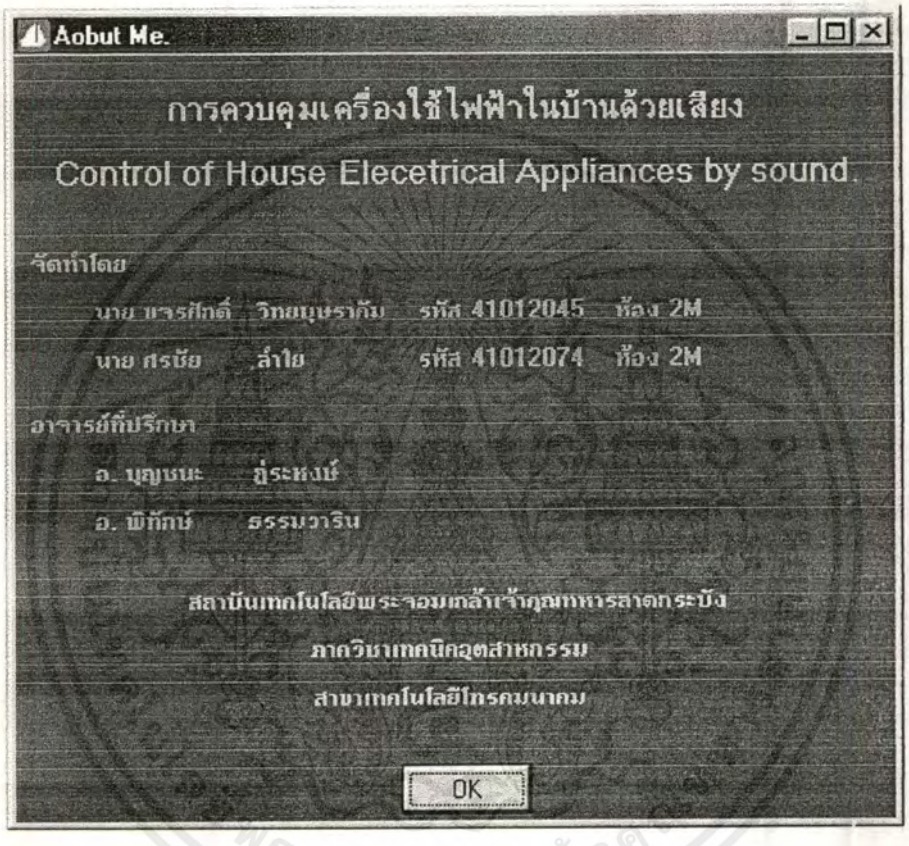


รูปที่ 2 ภาคผนวก ก แสดง HELP ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOURCE CODE ของ FORM3

```
Private Sub cmdOK3_Click()
   Unload Me
End Sub
```

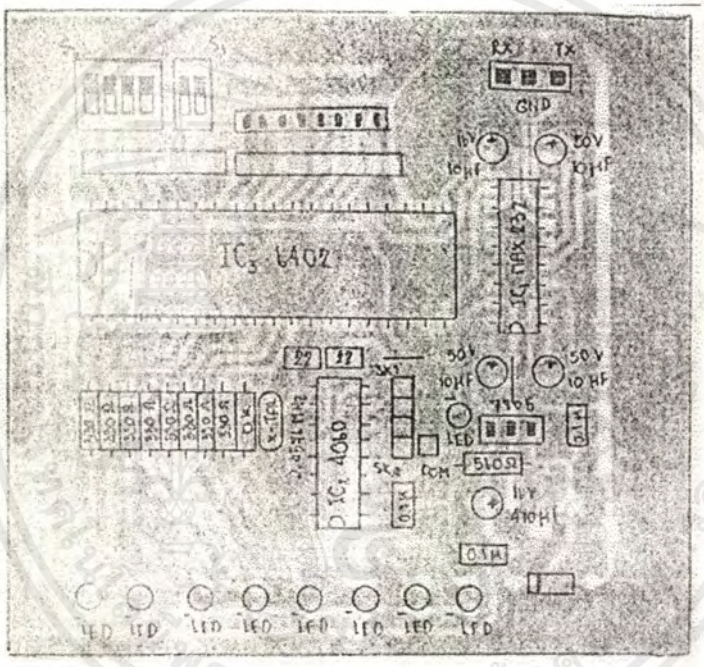


รูปที่ 3 ภาคผนวก ก แสดงเกี่ยวกับผู้จัดทำของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

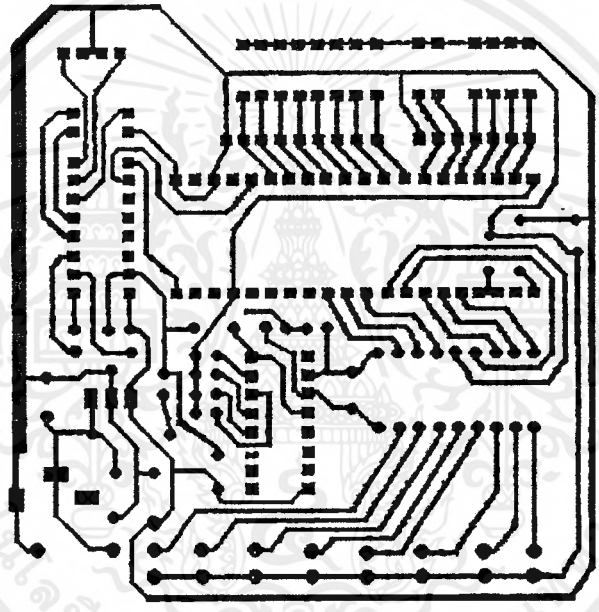
ภาคผนวก ข
ลายวงจรและการวางอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 ภาคผนวก ข แสดงการวางอุปกรณ์ของบอร์ดอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 ภาคผนวก ข แสดงลายวงจรของบอร์ดอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

คอนโทรลตัวอื่น ๆ ที่อยู่ใน Microsoft Speech SDK 4.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Direct Speech Recognition Control

Properties

Direct Speech Recognition control มีคุณสมบัติดังต่อไปนี้

AutoGain

CompleteTimeOut

CountEngines

CreateResultsObject

Dialect

Echo

EnergyFloor

EngineFeatures

EngineID

Features

FileName

Find

FindEngine

FlagsGet

GetPhraseScore

Grammars

hWnd

Identify

IncompleteTimeOut

Initialized

Interfaces

LanguageID

LastError

LastHeard MaxAutoGain

MaxCompleteTimeOut

MaxEnergyFloor

MaxIncompleteTimeOut

MaxRealTime

MaxThreshold

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MaxWordsState

MaxWordsVocab

MfgName

Microphone

MinAutoGain

MinCompleteTimeOut

MinEnergyFloor

MinIncompleteTimeOut

MinRealTime

MinThreshold

ModeID

ModeName

Phrase

ProductName

RealTime

ReEvaluate

Sequencing

Speaker

SRMode

SuppressExceptions

Threshold

Wave

Methods

Direct Speech Recognition control มี methods ดังต่อไปนี้

AboutDlg

Activate

ActivateAndAssignWindow

Archive

Correction

Deactivate

DeleteArchive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DestroyResultsObject

GeneralDlg

GrammarDataSet

GrammarFromFile

GrammarFromMemory

GrammarFromResource

GrammarFromStream

GrammarFromString

GrammarToMemory

InitAudioSourceDirect

InitAudioSourceObject

LexiconDlg

Listen

Pause

PosnGet

Resume

Select

SelectEngine

TimeGet

TrainGeneralDlg

TrainMicDlg

Validate

Events

Direct Speech Recognition control มี events ดังต่อไปนี้

AttribChanged BookMark

ClickIn

Interference

Paused

PhraseFinish

PhraseHypothesis

PhraseStart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ReEvaluate

Sound

Training

UnArchive

UtteranceBegin

UtteranceEnd

VUMeter

Direct Speech Synthesis Control

Properties

Direct Speech Synthesis Control มีคุณสมบัติดังต่อไปนี้

Age

CallBacksEnabled

CountEngines

CurrentMode

Dialect

EngineFeatures

EngineID

Features

FileName

Find

FindEngine

Gender

hWnd

Initialized

Interfaces

JawOpen

LanguageID

LastError

LastWordPosition

LipTension

LipType

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MaxPitch

MaxRealTime

MaxSpeed

MaxVolumeLeft

MaxVolumeRight

MfgName

MinPitch

MinRealTime

MinSpeed

MinVolumeLeft

MinVolumeRight

ModeID

ModeName

MouthEnabled

MouthHeight

MouthUpturn

MouthWidth

Phonemes

Pitch

ProductName

RealTime

Sayit

Speaker

Speaking

Speed

Style

SuppressExceptions

Tagged

TeethLowerVisible

TeethUpperVisible

TonguePosn

VolumeLeft

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VolumeRight

Methods

Direct Speech Synthesis Control มี methods ดังต่อไปนี้

AboutDlg

AudioPause

AudioReset

AudioResume

GeneralDlg

GetPronunciation

InitAudioDestDirect

InitAudioDestMM

InitAudioDestObject

Inject

LexiconDlg

PosnGet

Select

Speak

TextData

TranslateDlg

Events

Direct Speech Synthesis control มี events ดังต่อไปนี้

AttribChanged

AudioStart

AudioStop

BookMark

ClickIn

TextDataDonc

TextDataStarted

Visual

WordPosition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Voice Dictation Control

Properties

Voice Dictation control มี properties ดังต่อไปนี้

Attributes

AttributeMemory

AttributeString

AutoGainEnable

BookmarkQuery

CountBookmarks

CountCommands

CountGlossary

CountSpeakers

CreateDocFile

CreateStream

Echo

EnergyFloor

Flags

hWnd

Initialized

IsAnyoneDictating

LastError

Microphone

Mode

Option

OptionsEnum

RealTime

Speaker

SpeakerGet

SuppressExceptions

Threshold

TimeoutComplete

TimeoutIncomplete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wLeft

wTop

Methods

Voice Dictation control มี methods ดังต่อไปนี้

AboutDlg

Activate

ActivateAndAssignWindow

AddCommand

AddGlossary

BookmarkAdd

BookmarkRemove

CFGSet

CopyToBin

CopyToMemory

Deactivate

FX

GeneralDlg

GetBookMark

GetChanges

GetCommand

GetGlossary

GlobalKeyHook

GlobalMouseHook

Hint

ITNApply

ITNExpand

LexiconDlg

Lock

MemoryGet

MemorySet

PasteFromBin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PasteFromMemory

ReadStreamFont

ReleaseStore

ReleaseStream

RemoveCommand

RemoveGlossary

ResultsGet

ResultsGet2

ResultsSet

SessionDeserialize

SessionSerialize

SetCommand

SetGlossary

SetSelRect

SetSize

SpeakerDelete

SpeakerNew

SpeakerQuery

SpeakerRevert

SpeakerSelect

StreamRead

StreamWrite

TextGet

TextRemove

TextSelGet

TextSelSet

TextSet

TrainGeneralDlg

TrainMicDlg

Unlock

Words

WriteStreamFont

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Events

Voice Dictation control มี events ดังต่อไปนี้

AttribChanged

ClickIn

CommandBuiltIn

CommandOther

CommandRecognize

Dictating

GlobalKey

GlobalMouse

Interference

PhraseFinish

PhraseHypothesis

PhraseStart

TextBookmarkChanged

TextChanged

TextSelChanged

Training

UtteranceBegin

UtteranceEnd

VUMeter

Voice Text Control

Properties

Voice Text control มี properties ดังต่อไปนี้

Age

CountEngines

CurrentMode

Device

Dialect

Enabled

EngineFeatures

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

Find

FindEngine

Gender

hWnd

Initialized

Interfaces

IsSpeaking

JawOpen

LanguageID

LastError

LipTension

LipType

MfgName

ModeID

ModeName

MouthHeight

MouthUpturn

MouthWidth

ProductName

Speaker

Speed

Style

SuppressExceptions

TeethLowerVisible

TeethUpperVisible

TonguePosn

TTSMode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Methods

Voice Text control มี methods ดังต่อไปนี้

AboutDlg

FastForward

GeneralDlg

LexiconDlg

Pause

Resume

Rewind

Select

Speak

StopSpeaking

TranslateDlg

Events

Voice Text control มี events ดังต่อไปนี้

AttribChanged

ClickIn

Speak

SpeakingDone

SpeakingStarted

Visual

Voice Telephony Control

Properties

Voice Telephony control มี properties ดังต่อไปนี้

AnswerAfterRings

Initialized

MaxLines

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Methods

Voice Telephony control มี methods ดังต่อไปนี้

CallDialog

ChooseEngineDialog

DateFromFile

DateFromResource

DateFromString

DWORDGet

DWORDSet

ExtensionFromFile

ExtensionFromResource

ExtensionFromString

FreeWave

GrammarFromFile

GrammarFromResource

GrammarFromString

PhoneFromFile

PhoneFromResource

PhoneFromString

RecordFromFile

RecordFromResource

RecordFromString

SendAbort

SendDTMF

SendDTMFToLine

Speak

TimeFromFile

TimeFromResource

TimeFromString

WaveAddFromListFile

WaveAddFromListResource

WaveAddFromListString

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WaveAddFromMemory

YesNoFromFile

YesNoFromResource

YesNoFromString

Events

Voice Telephony control มี events ดังต่อไปนี้

ClickIn

DoPhoneCall

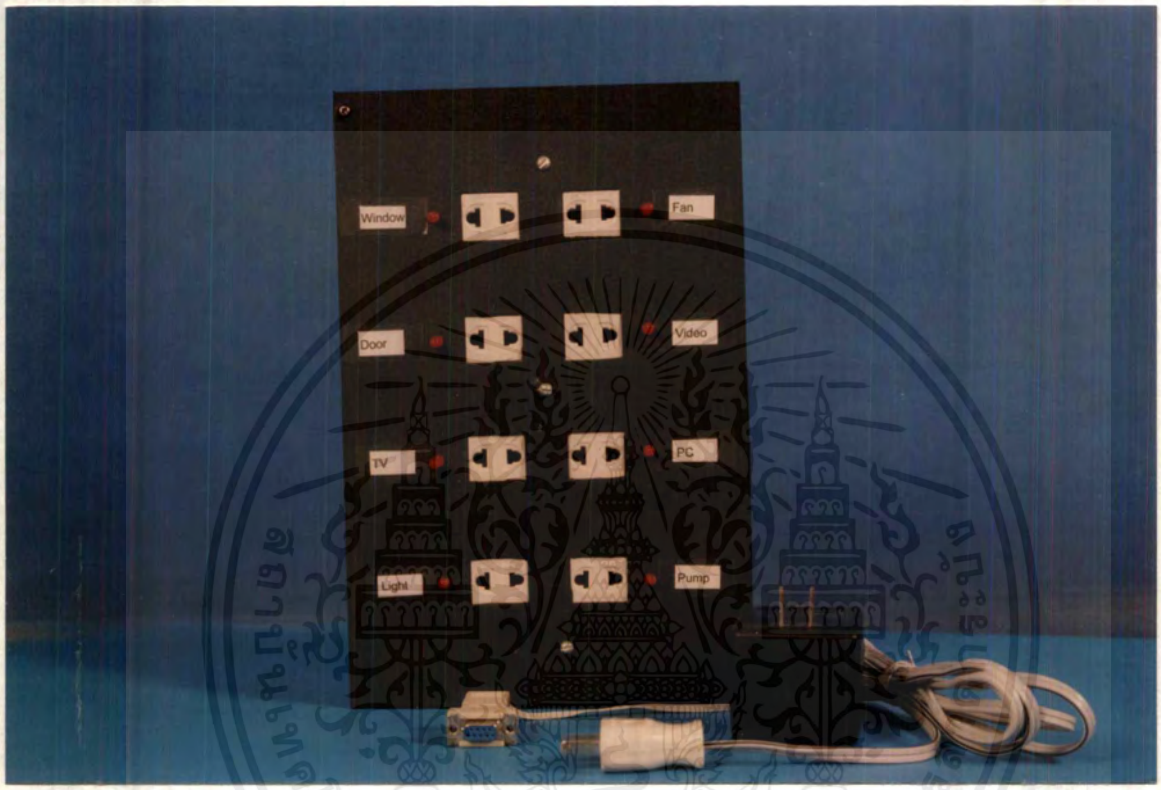
Initialize

Shutdown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

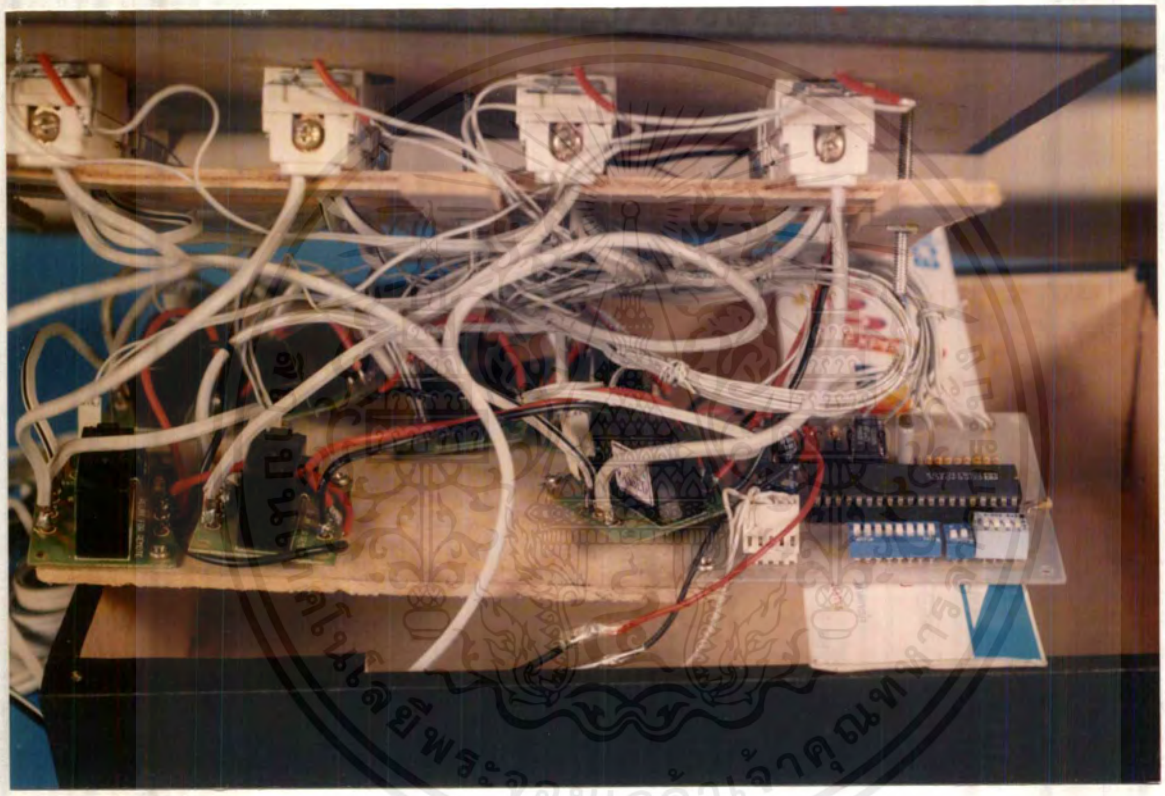
ภาคผนวก ง
รูปภาพฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 ภาคผนวก ง แสดงฮาร์ดแวร์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



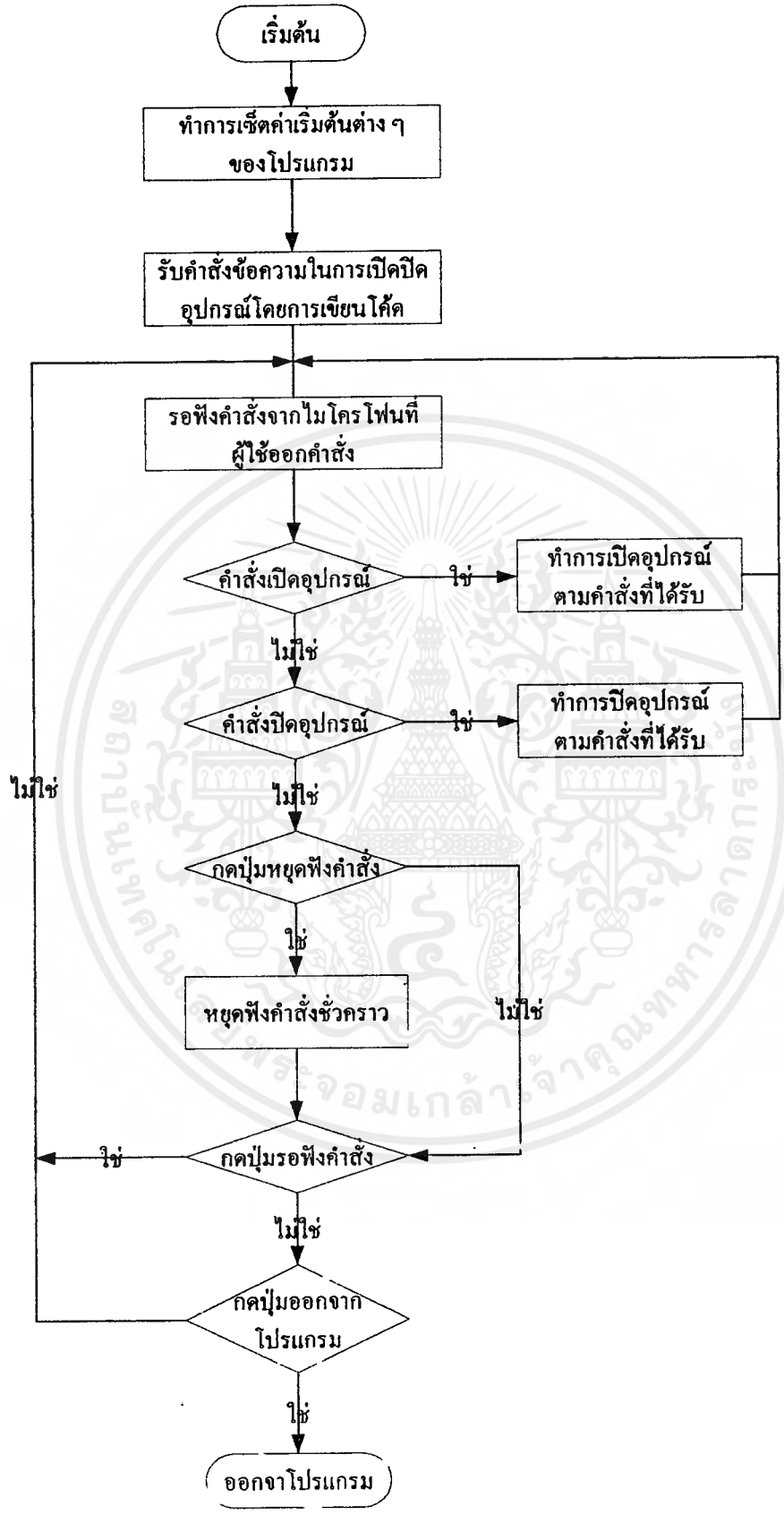
รูปที่ 2 ภาคผนวก ง แสดงฮาร์ดแวร์วงจรภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

FLOW CHAT แสดงการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 ภาคผนวก จ FLOW CHAT แสดงการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

DATA CHEET ของ IC UART CDP6402

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HARRIS SEMICONDUCTOR

CDP6402, CDP6402C

CMOS Universal Asynchronous Receiver/Transmitter (UART)

August 1996

Features

- Low Power CMOS Circuitry. . . . . 7.5mW (Typ) at 3.2MHz (Max Freq.) at V_{DD} = 5V
- Baud Rate
  - DC to 200K Bits/s (Max) at. . . . . 5V, 85°C
  - DC to 400K Bits/s (Max) at. . . . . 10V, 85°C
- 4V to 10.5 Operation
- Automatic Data Formatting and Status Generation
- Fully Programmable with Externally Selectable Word Length (5 - 8 Bits), Parity Inhibit, Even/Odd Parity, and 1, 1-1/2, or 2 Stop Bits
- Operating Temperature Range
  - CDP6402D, CD . . . . . -55°C to +125°C
  - CDP6402E, CE . . . . . -40°C to +85°C
- Replaces Industry Type IM6402 and Compatible with HD6402

Ordering Information

PACK-AGE	TEMP. RANGE	5V/200K BAUD	10V/400K BAUD	PKG. NO.
PDIP	-40°C to +85°C	CDP6402CE	CDP6402E	E40.6
Burn-In		CDP6402CEX	-	
SBDIP	-40°C to +85°C	CDP6402CD	CDP6402D	D40.6
Burn-In		CDP6402CDX	CDP6402DX	

Description

The CDP6402 and CDP6402C are silicon gate CMOS Universal Asynchronous Receiver/Transmitter (UART) circuits for interfacing computers or microprocessors to asynchronous serial data channels. They are designed to provide the necessary formatting and control for interfacing between serial and parallel data channels. The receiver converts serial start, data, parity, and stop bits to parallel data verifying proper code transmission, parity and stop bits. The transmitter converts parallel data into serial form and automatically adds start parity and stop bits.

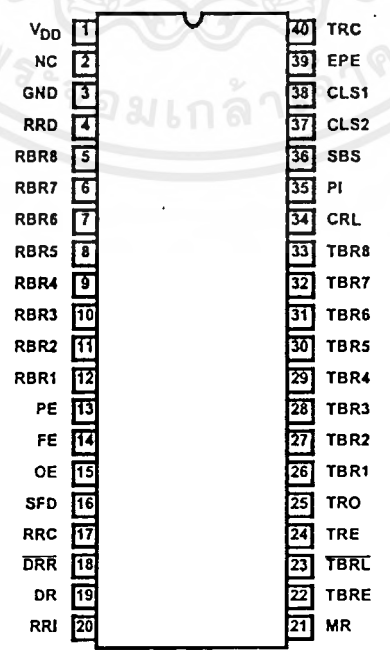
The data word can be 5, 6, 7 or 8 bits in length. Parity may be odd, even or inhibited. Stop bits can be 1, 1-1/2, or 2 (when transmitting 5-bit code).

The CDP6402 and CDP6402C can be used in a wide range of applications including modems, printers, peripherals, video terminals, remote data acquisition systems, and serial data links for distributed processing systems.

The CDP6402 and CDP6402C are functionally identical. They differ in that the CDP6402 has a recommended operating voltage range of 4V to 10.5V, and the CDP6402C has a recommended operating voltage range of 4V to 6.5V.

Pinout

(40 LEAD PDIP, SBDIP)
TOP VIEW



CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.

Copyright © Harris Corporation 1996

ไมวากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

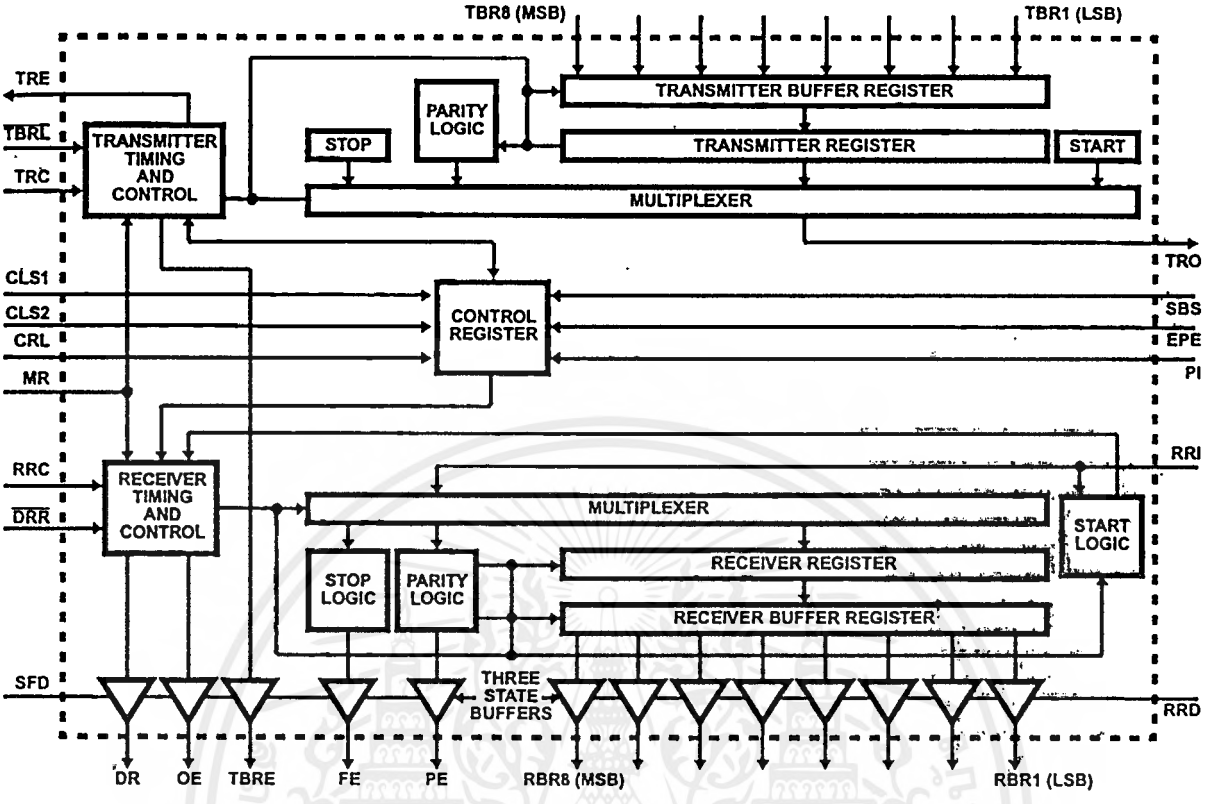


FIGURE 1. FUNCTIONAL BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

DC Supply-Voltage Range, (V_{DD})
- CDP6402 ..... -0.5 to +11V
- CDP6402C ..... -0.5 to +7V

Input Voltage Range, All Inputs ..... -0.5 to V_{DD} +0.5V
DC Input Current, Any One Input ..... ±100µA
Device Dissipation Per Output Transistor
- For T_A = Full Package-Temperature Range
- (All Package Types) 100mW

Operating-Temperature Range (T_A)
- Package Type D (SBDIP) ..... -55°C to +125°C
- Package Type E (PDIP) ..... -40°C to +85°C

Thermal Information

Thermal Resistance (Typical, Note 1)	θ_{JA} (°C/W)	θ_{JC} (°C/W)
PDIP Package	50	N/A
SBDIP Package	55	15

Maximum Storage Temperature Range (T_{STG}) ... -65°C to +150°C
Maximum Lead Temperature (Soldering 10s):
- At Distance 1/16 ±1/32 inch (1.59 ±0.79mm) ..... +265°C

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

1. θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

Operating Conditions At T_A = Full Package-Temperature Range. For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

PARAMETER	LIMITS				UNITS
	CDP6402		CDP6402C		
	MIN	MAX	MIN	MAX	
DC Operating Voltage Range	4	10.5	4	6.5	V
Input Voltage Range	V_{SS}	V_{DD}	V_{SS}	V_{DD}	V

Static Electrical Specifications at T_A = -40°C to +85°C, V_{DD} ±10%, Except as noted

PARAMETER		CONDITIONS			LIMITS						UNITS
					CDP6402			CDP6402C			
		V_O (V)	V_{IN} (V)	V_{DD} (V)	MIN	(NOTE 1) TYP	MAX	MIN	(NOTE 1) TYP	MAX	
Quiescent Device Current	I_{DD}	-	0, 5	5	-	0.01	50	-	0.02	200	µA
		-	0,10	10	-	1	200	-	-	-	µA
Output Low Drive (Sink) Current	I_{OL}	0.4	0,5	5	2	4	-	1.2	2.4	-	mA
		0.5	0,10	10	5	7	-	-	-	-	mA
Output High Drive (Source) Current	I_{OH}	4.6	0, 5	5	-0.55	-1.1	-	-0.55	-1.1	-	mA
		9.5	0,10	10	-1.3	-2.6	-	-	-	-	mA
Output Voltage Low-Level (Note 2)	V_{OL}	-	0, 5	5	-	0	0.1	-	0	0.1	V
		-	0, 10	10	-	0	0.1	-	-	-	V
Output Voltage High Level (Note 2)	V_{OH}	-	0, 5	5	4.9	5	-	4.9	5	-	V
		-	0, 10	10	9.9	10	-	-	-	-	V
Input Low Voltage	V_{IL}	0.5, 4.5	-	5	-	-	0.8	-	-	0.8	V
		0.5, 9.5	-	10	-	-	0.2 V_{DD}	-	-	-	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Static Electrical Specifications at T_A = -40°C to +85°C, V_{DD} ±10%, Except as noted (Continued)

PARAMETER		CONDITIONS			LIMITS						UNITS
					CDP6402			CDP6402C			
		V_O (V)	V_{IN} (V)	V_{DD} (V)	MIN	(NOTE 1) TYP	MAX	MIN	(NOTE 1) TYP	MAX	
Input High Voltage	V_{IH}	0.5, 4.5	-	5	V_{DD}-2	-	-	V_{DD}-2	-	-	V
		0.5, 9.5	-	10	7	-	-	-	-	-	V
Input Leakage Current	I_{IN}	Any Input	0,5	5	-	$\pm10^{-4}$	±1	-	-	±1	µA
			0,10	10	-	$\pm10^{-4}$	±2	-	-	-	µA
Three-State Output Leakage Current	I_{OUT}	0, 5	0, 5	5	-	$\pm10^{-4}$	±1	-	$\pm10^{-4}$	±1	µA
		0, 10	0,10	10	-	$\pm10^{-4}$	±10	-	-	-	µA
Operating Current (Note 2)	I_{DD1}	-	0, 5	5	-	1.5	–	-	1.5	-	mA
		-	0,10	10	-	10	–	-	-	-	mA
Input Capacitance	C_{IN}	-	-	-	-	5	7.5	-	5	7.5	pF
Output Capacitance	C_{OUT}	-	-	-	-	10	15	-	10	15	pF

NOTES:

1. Typical values are for T_A = 25°C and nominal V_{DD}
2. I_{OL} = I_{OH} = 1µA.
3. Operating current is measured at 200kHz or V_{DD} = 5V and 400kHz for V_{DD} = 10V, with open outputs (worst-case frequencies for CDP1802A system operating at maximum speed of 3.2MHz).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Description of Operation

Initialization and Controls

A positive pulse on the MASTER RESET (MR) input resets the control, status, and receiver buffer registers, and sets the serial output (TRO) High. Timing is generated from the clock inputs RRC and TRC at a frequency equal to 16 times the serial data bit rate. The RRC and TRC inputs may be driven by a common clock, or may be driven independently by two different clocks. The CONTROL REGISTER LOAD (CRL) input is strobed to load control bits for PARITY INHIBIT (PI), EVEN PARITY ENABLE (EPE), STOP BIT SELECTS (SBS), and CHARACTER LENGTH SELECTS (CLS1 and CLS2). These inputs may be hand wired to V_{SS} or V_{DD} with CRL to V_{DD}. When the initialization is completed, the UART is ready for receiver and/or transmitter operations.

Transmitter Operation

The transmitter section accepts parallel data, formats it, and transmits it in serial form (Figure 2) on the TRO terminal.

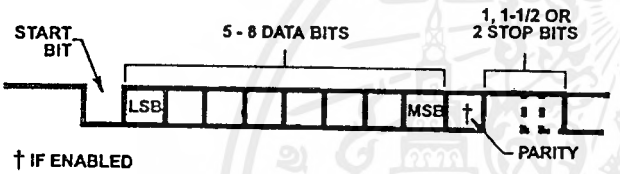


FIGURE 2. SERIAL DATA FORMAT

Transmitter timing is shown in Figure 3. (A) Data is loaded into the transmitter buffer register from the inputs TBR1 through TBR8 by a logic low on the $\overline{TBRL}$ input. Valid data must be present at least t_{DT} prior to, and t_{TD} following, the rising edge of $\overline{TBRL}$. If words less than 8-bits are used, only the least significant bits are used. The character is right justified into the least significant bit, TBR1. (B) The rising edge of $\overline{TBRL}$ clears TBRE. 1/2 to 11/2 cycles later, depending on when the $\overline{TBRL}$ pulse occurs with respect to TRC, data is transferred to the transmitter register and TRE is cleared. TBRE is set to a logic High one cycle after that.

Output data is clocked by TRC. The clock rate is 16 times the data rate. (C) A second pulse on $\overline{TBRL}$ loads data into the transmitter buffer register. Data transfer to the transmitter register is delayed until transmission of the current character is complete. (D) Data is automatically transferred to the transmitter register and transmission of that character begins.

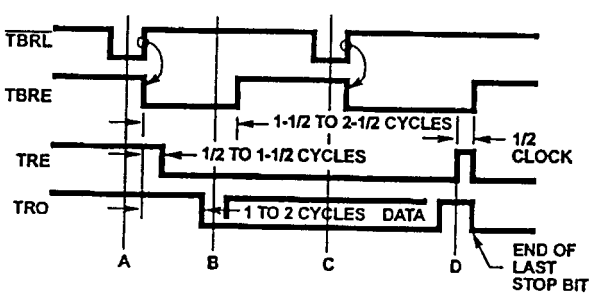


FIGURE 3. TRANSMITTER TIMING WAVEFORMS

Receiver Operation

Data is received in serial form at the RRI input. When no data is being received, RRI input must remain high. The data is clocked through the RRC. The clock rate is 16 times the data rate. Receiver timing is shown in Figure 4.

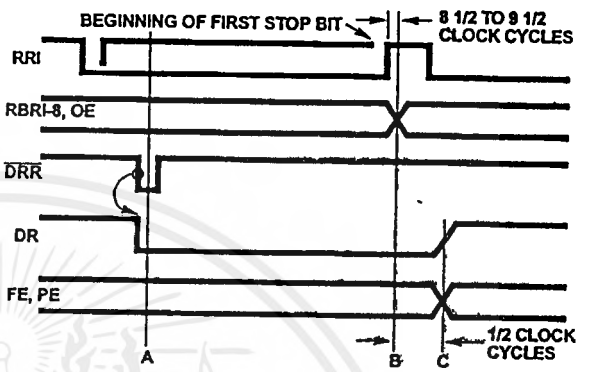


FIGURE 4. RECEIVER TIMING WAVEFORMS

(A) A low level on $\overline{DRR}$ clears the DR line. (B) During the first stop bit data is transferred from the receiver register to the RB Register. If the word is less than 8 bits, the unused most significant bits will be a logic low. The output character is right justified to the least significant bit RBR1. A logic high on OE indicates overruns. An overrun occurs when DR has not been cleared before the present character was transferred to the RBR. (C) 1/2 clock cycle later DR is set to a logic high and FE is evaluated. A logic high on FE indicates an invalid stop bit was received. A logic high on PE indicates a parity error.

Start Bit Detection

The receiver uses a 16X clock for timing (Figure 5). The start bit could have occurred as much as one clock cycle before it was detected, as indicated by the shaded portion. The center of the start bit is defined as clock count 7 1/2. If the receiver clock is a symmetrical square wave, the center of the start bit will be located within ±1/2 clock cycle ±1/32 bit or ±3.125%. The receiver begins searching for the next start bit at 9 clocks into the first stop bit.

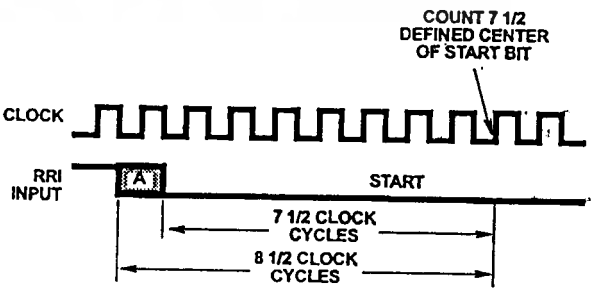


FIGURE 5. START BIT TIMING WAVEFORMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1. CONTROL WORD FUNCTION

CONTROL WORD					DATA BITS	PARITY BIT	STOP BIT (S)
CLS2	CLS1	PI	EPE	SBS			
L	L	L	L	L	5	ODD	1
L	L	L	L	H	5	ODD	1.5
L	L	L	H	L	5	EVEN	1
L	L	L	H	H	5	EVEN	1.5
L	L	H	X	L	5	DISABLED	1
L	L	H	X	H	5	DISABLED	1.5
L	H	L	L	L	6	ODD	1
L	H	L	L	H	6	ODD	2
L	H	L	H	L	6	EVEN	1
L	H	L	H	H	6	EVEN	2
L	H	H	X	L	6	DISABLED	1
L	H	H	X	H	6	DISABLED	2
H	L	L	L	L	7	ODD	1
H	L	L	L	H	7	ODD	2
H	L	L	H	L	7	EVEN	1
H	L	L	H	H	7	EVEN	2
H	L	H	X	L	7	DISABLED	1
H	L	H	X	H	7	DISABLED	2
H	H	L	L	L	8	ODD	1
H	H	L	L	H	8	ODD	2
H	H	L	H	L	8	EVEN	1
H	H	L	H	H	8	EVEN	2
H	H	H	X	L	8	DISABLED	1
H	H	H	X	H	8	DISABLED	2

NOTE: X = Don't Care

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 2. FUNCTION PIN DEFINITION

PIN	SYMBOL	DESCRIPTION
1	V_{DD}	Positive Power Supply
2	N/C	No Connection
3	GND	Ground (V_{SS})
4	RRD	A high level on RECEIVER REGISTER DISABLE forces the receiver holding register ouputs RBR1-RBR8 to a high impedance state.
5	RBR8	The contents of the RECEIVER BUFFER REGISTER appear on these three-state outputs. Word formats less than 8 characters are right justified to RBR1.
6	RBR7	See Pin 5 - RBR8
7	RBR6	
8	RBR5	
9	RBR4	
10	RBR3	
11	RBR2	
12	RBR1	
13	PE	A high level on PARITY ERROR indicates that the received parity does not match parity programmed by control bits. The output is active until parity matches on a succeeding character. When parity is inhibited, this output is low.
14	FE	A high level on FRAMING ERROR indicates the first stop bit was invalid. FE will stay active until the next valid character's stop bit is received.
15	OE	A high level on OVERRUN ERROR indicates the data received flag was not cleared before the last character was transferred to the receiver buffer register. The Error is reset at the next character's stop bit if DRR has been performed (i.e., DRR; active low).
16	SFD	A high level on STATUS FLAGS DISABLE forces the outputs PE, FE, OE, DR, TBRE to a high impedance state.
17	RRC	The RECEIVER REGISTER CLOCK is 16X the receiver data rate.
18	$\overline{DRR}$	A low level on DATA RECEIVED RESET clears the data received output (DR), to a low level.
19	DR	A high level on DATA RECEIVED indicates a character has been received and transferred to the receiver buffer register.
20	RRI	Serial data on RECEIVER REGISTER INPUT is clocked into the receiver register.
21	MR	A high level on MASTER RESET (MR) clears PE, FE, OE and DR, and sets TRE, TBRE, and TRO. TRE is actually set on the first rising edge of TRC after MR goes high. MR should be strobed after power-up.
22	TBRE	A high level on TRANSMITTER BUFFER REGISTER EMPTY indicates the transmitter buffer register has transferred its data to the transmitter register and is ready for new data.
23	$\overline{TBRL}$	A low level on TRANSMITTER BUFFER REGISTER LOAD transfers data from inputs TBR1-TBR8 into the transmitter buffer register. A low to high transition on $\overline{TBRL}$ requests data transfer to the transmitter register. If the transmitter register is busy, transfer is automatically delayed so that the two characters are transmitted end to end.
24	TRE	A high level on TRANSMITTER REGISTER EMPTY indicates completed transmission of a character including stop bits.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 2. FUNCTION PIN DEFINITION (Continued)

PIN	SYMBOL	DESCRIPTION
25	TRO	Character data, start data and stop bits appear serially at the TRANSMITTER REGISTER OUTPUT.
26	TBR1	Character data is loaded into the TRANSMITTER BUFFER REGISTER via inputs TBR1-TBR8. For character formats less than 8 bits, the TBR8, 7, and 6 inputs are ignored corresponding to the programmed word length.
27	TBR2	See Pin 26 - TBR1
28	TBR3	
29	TBR4	
30	TBR5	
31	TBR6	
32	TBR7	
33	TBR8	
34	CRL	A high level on CONTROL REGISTER LOAD loads the control register.
35	PI†	A high level on PARITY INHIBIT inhibits parity generation, parity checking and forces PE output low.
36	SBS†	A high level on STOP BIT SELECT selects 1.5 stop bits for a 5 character format and 2 stop bits for other lengths.
37	CLS2†	These inputs program the CHARACTER LENGTH SELECTED. (CLS1 low CLS2 low 5 bits) (CLS1 high CLS2 low 6 bits) (CLS1 low CLS2 high 7 bits) (CLS1 high CLS2 high 8 bits).
38	CLSI†	See Pin 37 - CLS2
39	EPE†	When PI is low, a high level on EVEN PARITY ENABLE generates and checks even parity. A low level selects odd parity.
40	TRC	The TRANSMITTER REGISTER CLOCK is 16X the transmit data rate.

† See Table 1 (Control Word Function)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dynamic Electrical Specifications at T_A = -40°C to +85°C, V_{DD} ±5%, t_R, t_F = 20ns, V_{IH} = 0.7 V_{DD}, V_{IL} = 0.3 V_{DD}, C_L = 100pF

(NOTE 1) PARAMETER		V_{DD} (V)	LIMITS				UNITS
			CDP6402		CDP6402C		
			(NOTE 2) TYP	(NOTE 3) MAX	(NOTE 2) TYP	(NOTE 3) MAX	
SYSTEM TIMING (See Figure 6)							
Minimum Pulse Width CRL	t_{CRL}	5	50	150	50	150	ns
		10	40	100	-	-	ns
Minimum Setup Time Control Word to CRL	t_{CWC}	5	20	50	20	50	ns
		10	0	40	-	-	ns
Minimum Hold Time Control Word after CRL	t_{CCW}	5	40	60	40	60	ns
		10	20	30	-	-	ns
Propagation Delay Time SFD High to SOD	t_{SFDH}	5	130	200	130	200	ns
		10	100	150	-	-	ns
SFD Low to SOD	t_{SFDL}	5	130	200	130	200	ns
		10	40	60	-	-	ns
RRD High to Receiver Register High Impedance	t_{RRDH}	5	80	150	80	150	ns
		10	40	70	-	-	ns
RRD Low to Receiver Register Active	t_{RRDL}	5	80	150	80	150	ns
		10	40	70	-	-	ns
Minimum Pulse Width MR		5	200	400	200	400	ns
		10	100	200	-	-	ns

NOTES:

1. All measurements are made at the 50% point of the transition except three-state measurements.
2. Typical values for T_A = 25°C and nominal V_{DD}.
3. Maximum limits of minimum characteristics are the values above which all devices function.

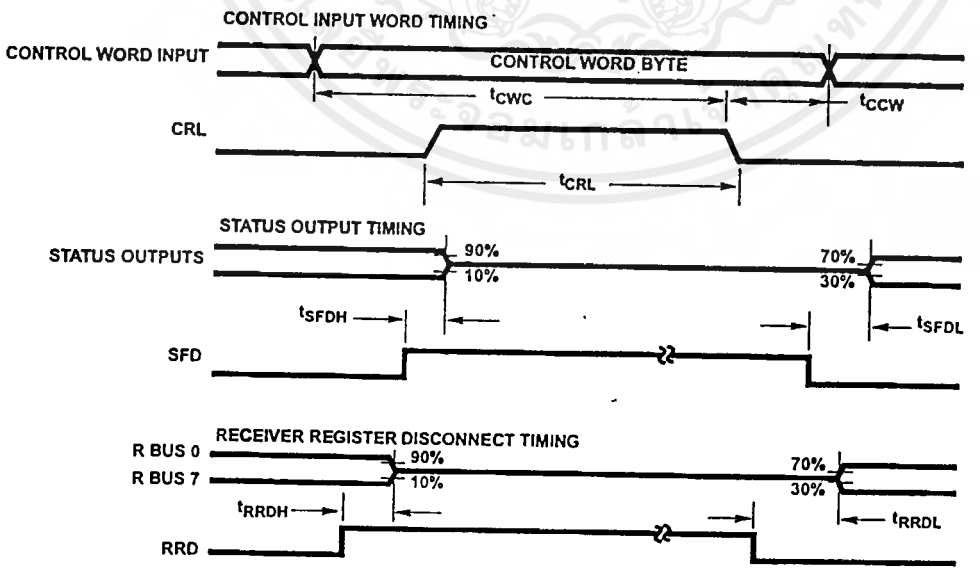


FIGURE 6. SYSTEM TIMING WAVEFORMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

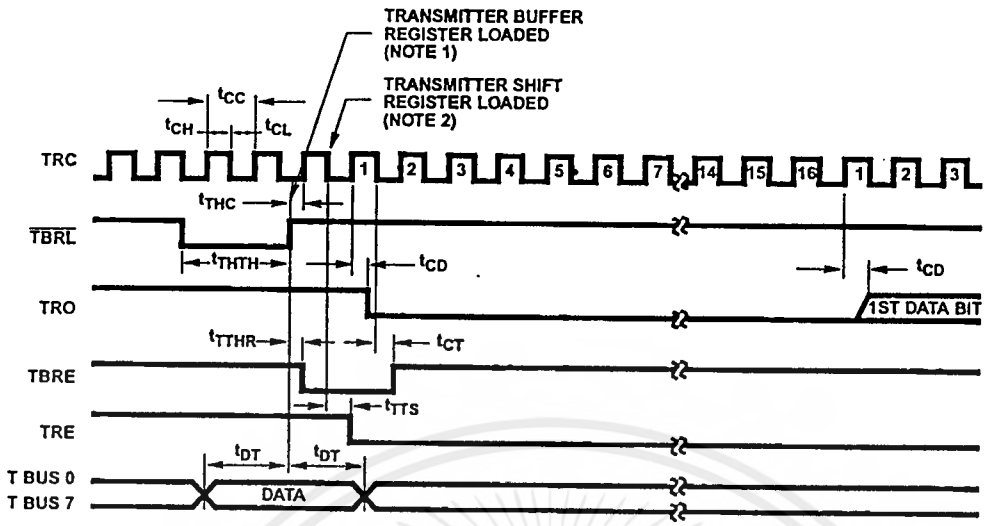
Dynamic Electrical Specifications at T_A = -40°C to +85°C, V_{DD} ±5%, t_R, t_F = 20ns, V_{IH} = 0.7 V_{DD}, V_{IL} = 0.3 V_{DD}, C_L = 100pF

(NOTE 1) PARAMETER		V_{DD} (V)	LIMITS				UNITS
			CDP6402		CDP6402C		
			(NOTE 2) TYP	(NOTE 3) MAX	(NOTE 2) TYP	(NOTE 3) MAX	
TRANSMITTER TIMING (See Figure 7)							
Minimum Clock Period (TRC)	t_{CC}	5	250	310	250	310	ns
		10	125	155	-	-	ns
Minimum Pulse Width							
Clock Low Level	t_{CL}	5	100	125	100	125	ns
		10	75	100	-	-	ns
Clock High Level	t_{CH}	5	100	125	100	125	ns
		10	75	100	-	-	ns
$\overline{TBRL}$	t_{THTH}	5	80	200	80	200	ns
		10	40	100	-	-	ns
Minimum Setup Time							
$\overline{TBRL}$ to Clock	t_{THC}	5	175	275	175	275	ns
		10	90	150	-	-	ns
Data to $\overline{TBRL}$ ↗	t_{DT}	5	20	50	20	50	ns
		10	0	40	-	-	ns
Minimum Hold-Time							
Data after $\overline{TBRL}$ ↗	t_{TD}	5	40	60	40	60	ns
		10	20	30	-	-	ns
Propagation Delay Time							
Clock to Data Start Bit	t_{CD}	5	300	450	300	450	ns
		10	150	225	-	-	ns
Clock to TBRE	t_{CT}	5	330	400	330	400	ns
		10	100	150	-	-	ns
$\overline{TBRL}$ to TBRE	t_{TTHR}	5	200	300	200	300	ns
		10	100	150	-	-	ns
Clock to TRE	t_{TTS}	5	330	400	330	400	ns
		10	100	150	-	-	ns

NOTES:

1. All measurements are made at the 50% point of the transition except three-state measurements.
2. Typical values for T_A = 25°C and nominal V_{DD}.
3. Maximum limits of minimum characteristics are the values above which all devices function.

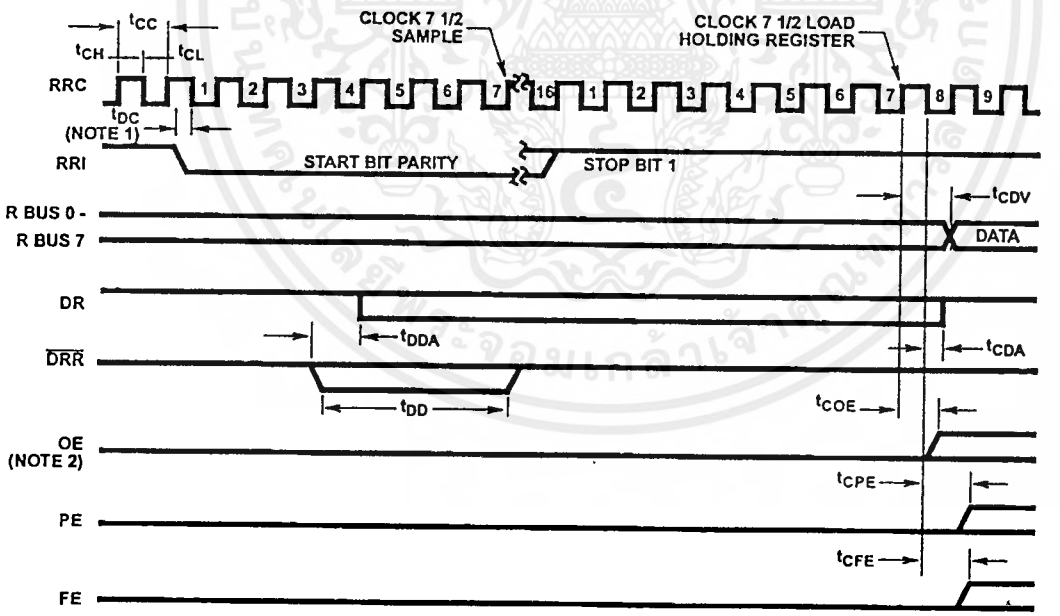
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



NOTES:

1. The holding register is loaded on the trailing edge of $\overline{TBRL}$.
2. The transmitter shift register, if empty , is loaded on the first high-to-low transition of the clock which occurs at least 1/2 clock period + t_{THC} after the trailing edge of $\overline{TBRL}$ and transmission of a start bit occurs 1/2 clock period + t_{CD} later.

FIGURE 7. TRANSMITTER TIMING WAVEFORMS



NOTES:

1. If a start bit occurs at a time less than t_{DC} before a high-to-low transition of the clock, the start bit may not be recognized until the next high-to-low transition of the clock. The start bit may be completely asynchronous with the clock.
2. If a pending DA has not been cleared by a read of the receiver holding register by the time a new word is loaded into the receiver holding register, the OE signal will come true..

FIGURE 8. RECEIVER TIMING WAVEFORMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dynamic Electrical Specifications at T_A = -40°C to +85°C, V_{DD} ±5%, t_R, t_F = 20ns, V_{IH} = 0.7 V_{DD}, V_{IL} = 0.3 V_{DD}, C_L = 100pF

(NOTE 1) PARAMETERS		V_{DD} (V)	LIMITS				UNITS
			CDP6402		CDP6402C		
			(NOTE 2) TYP	(NOTE 3) MAX	(NOTE 2) TYP	(NOTE 3) MAX	
RECEIVER TIMING (See Figure 8)							
Minimum Clock Period (RRC)	t_{CC}	5	250	310	250	310	ns
		10	125	155	-	-	ns
Minimum Pulse Width							
Clock Low Level	t_{CL}	5	100	125	100	125	ns
		10	75	100	-	-	ns
Clock High Level	t_{CH}	5	100	125	100	125	ns
		10	75	100	-	-	ns
Data Received Reset	t_{DD}	5	50	75	50	75	ns
		10	25	40	-	-	ns
Minimum Setup Time							
Data Start Bit to Clock	t_{DC}	5	100	150	100	150	ns
		10	50	75	-	-	ns
Propagation Delay Time							
Data Received Reset to Data Received	t_{DDA}	5	150	250	150	250	ns
		10	75	125	-	-	ns
Clock to Data Valid	t_{CDV}	5	275	400	275	400	ns
		10	110	175	-	-	ns
Clock to DR	t_{CDA}	5	275	400	275	400	ns
		10	110	175	-	-	ns
Clock to Overrun Error	t_{COE}	5	275	400	275	400	ns
		10	100	150	-	-	ns
Clock to Parity Error	t_{CPE}	5	240	375	240	375	ns
		10	120	17	-	-	ns
Clock to Framing Error	t_{CFE}	5	200	300	200	300	ns
		10	100	150	-	-	ns

NOTES:

1. All measurements are made at the 50% point of the transition except three-state measurements.
2. Typical values for T_A = 25°C and nominal V_{DD}.
3. Maximum limits of minimum characteristics are the values above which all devices function.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ฉัททวุฒิ พืชผล, พิชิต สันติกุลานนท์, "คู่มือเรียน VISUAL BASIC" , PROVISION, 2542

2. Dan Appleman, "Visual Basic Programmer's Guide to the WIN32API", Ziff-Davis Press, 1996

3. Http://MSDN.MICROSOFT.COM/VBASIC/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้