สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมรถเข็นไฟฟ้าด้วยเสียงพูด

Control of Electric Wheelchair by sound

โดย

นาย กฤษดา แสงอรุณบริสุทธิ์ 42015544

เลขหมู่.....
เลขทะเบียน 42247
วัน, เดือน, ปี 16 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีอิเล็กทรอนิกส์ ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวะกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การควบคุมรถเข็นไฟฟ้าด้วยเสียงพูด
ชื่อนักศึกษา	นาย กฤษดา แสงอรุณบริสุทธิ์ 42015544
อาจารย์ที่ปรึกษา	อาจารย์ สุธีรา พันธุ์ธีรานุรักษ์
	อาจารย์ บุญย์ชนะ ภู่ระหงษ์
ภาควิชา	เทคนิคอุตสาหกรรม
สาขาวิชา	เทคโนโลยีอิเล็กทรอนิกส์
ปีการศึกษา	2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

---------- ประธานกรรมการ
( )

---------- กรรมการ
( )

---------- กรรมการ
( )

---------- กรรมการ
( )

---------- กรรมการ
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	**การควบคุมรถเข็นไฟฟ้าด้วยเสียงพูด**
ชื่อนักศึกษา	นาย กฤษดา แสงอรุณบริสุทธิ์ 42015544
อาจารย์ที่ปรึกษา	อาจารย์ สุธีรา พันธุ์ธีรานุรักษ์
	อาจารย์ บุณย์ชนะ ภู่ระหงษ์
ภาควิชา	เทคนิคอุตสาหกรรม
สาขาวิชา	เทคโนโลยีอิเล็กทรอนิกส์
ปีการศึกษา	2544

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการควบคุมรถเข็นไฟฟ้าด้วยเสียงพูด โดยผ่านคอมพิวเตอร์แสดงผลออกทางพอร์ต RS232 โดยใช้โปรแกรม VISUAL BASIC 6.0 ซึ่งใช้ ACTIVE X CONTROL ที่ทำการติดตั้งเพิ่มเข้าไปใน VISUAL BASIC ทำให้มีความสามารถในการจัดการเรื่องเสียงดีมากขึ้น โดย ACTIVE X CONTROL นี้ชื่อ MICROSOFT SPEECH SDK 4.0

ในการจัดการเกี่ยวกับสัญญาณเสียงนั้น ACTIVE X CONTROL จะทำหน้าที่เป็นตัวจัดการสัญญาณเสียงเกือบทั้งหมด เพียงแต่จะต้องทำการศึกษาวิธีการใช้และเขียน CODE ของโปรแกรมโดย VISUAL BASIC เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TITLE	Control of electric wheelchair by sound
STUDENT	Mr. Kitsada Sangarunborisut 42015544
ADVISOR	Mis. Suteera Punteeranurak
	Mr. Boonchana Poorahong
DEGREE	Bachelor Degree of Industrial Electronic
PROGRAMMER	Electronic Technology
DEPARTMENT	Industrial Technology
ACADEMIC YEAR	2001

ABSTRACT

This project present control of electric wheelchair by sound wich uses computer to analyze and out 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ เพราะอาจารย์ทุกท่านที่ได้ให้คำปรึกษา คำแนะนำที่ดี และกำลังใจอีกมากมายจากบิดา มารดา เพื่อนที่คอยถามไถ่ให้ความห่วงใยมาโดยตลอด โดยเฉพาะเมื่อเวลารู้สึกท้อแท้เนื่องจากมีปัญหามากมาย

สุดท้ายนี้ขอให้สิ่งดีๆ จงบังเกิดแด่ทุกๆ ท่านที่กล่าวถึงตลอดไป

นาย กฤษดา แสงอรุณบริสุทธิ์

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
ABSTRACT	ข
กิตติกรรมประกาศ	ค
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 Speech Techmology	5
2.5 Active X	8
2.6 Active X control	8
2.7 Microsoft Speech SDK	9
บทที่ 3 ฮาร์ดแวร์	
3.1 การทำงานของบอร์ดอินเตอร์เฟส	23
3.2 UART	25
บทที่ 4 ผลการทดลอง	
4.1 ฮาร์ดแวร์ (Hardware)	29
4.2 ซอฟต์แวร์ (Software)	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญต่อ

เรื่อง	หน้า
บทที่ 5 สรุปปัญหาและข้อเสนอแนะ	
5.1 สรุป	43
5.2 ปัญหาที่เกิดขึ้นในการทำงาน	43
5.3 แนวทางแก้ไข	43
5.4 ข้อเสนอแนะในโครงงาน	44
บรรณานุกรม	
ภาคผนวก	
ภาคผนวก ก Source Code ของโปรแกรม	
ภาคผนวก ข คอนโทรล ตัวอื่น ๆ ที่อยู่ใน Microsoft Speech SDK	
ภาคผนวก ค Flow Chat แสดงการทำงานของโปรแกรม	
ภาคผนวก ง ลายปริ้นของวงจรอินเตอร์เฟส	
ภาคผนวก จ Data Cheet ของ IC UART CDP6402	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้า
รูปที่ 2.1 อวัยวะที่ใช้ในการออกเสียง	4
รูปที่ 3.1 การต่อพอร์ตอนุกรมแบบ DB9 และ DB25	24
รูปที่ 3.2 วงจรของบอร์ดอินเตอร์เฟส	24
รูปที่ 3.3 วรจร POWER SUPPY	25
รูปที่ 3.4 บล็อกไดอะแกรมการทำงานภายในไอซี	28
รูปที่ 4.1 เมนูของโปรแกรมทดสอบฮาร์ดแวร์	35
รูปที่ 4.2 สัญลักษณ์ของ Microsoft Voice	36
รูปที่ 4.3 เมนูการทำงานของ Microsoft Voice	37
รูปที่ 4.4 การใช้เมนู What Can I Say	37
รูปที่ 4.5 การใช้เมนู Listening Moden ใน Command options	38
รูปที่ 4.6 เมนู Voice Commands ใน Voice Command options	39
รูปที่ 4.7 การใช้คำสั่ง "start microsoft word"	40
รูปที่ 4.8 ผลของการใช้คำสั่ง "start microsoft word"	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 2.1 ความหมายของค่าพารามิเตอร์ต่างๆ ของ MenuCreate	13
ตารางที่ 2.2 ค่าของ flags ของคำสั่ง MenuCreate	14
ตารางที่ 2.3 ความหมายของพารามิเตอร์ต่างๆ ของ Activate	15
ตารางที่ 2.4 ความหมายของค่าพารามิเตอร์ต่างๆ ของ AddCommand	17
ตารางที่ 2.5 ค่า flage ของคำสั่ง AddCommand	17
ตารางที่ 2.6 ความหมายค่าพารามิเตอร์ต่างๆ ของ Deactivate	18
ตารางที่ 2.7 ความหมายของค่าพารามิเตอร์ต่างๆ ของ GetCommand	19
ตารางที่ 2.8 ค่าของ flags ของคำสั่ง GetCommand	20
ตารางที่ 2.9 ความหมายของค่าพารามิเตอร์ต่างๆ ของ ReleaseMenu	20
ตารางที่ 2.10 ความหมายของค่าพารามิเตอร์ต่างๆ ของ CommandRecogniZe	21
ตารางที่ 3.1 การกำหนดโหมดเพื่อควบคุมการทำงานของไอซี 6402	26
ตารางที่ 4.1 ตารางการทดสอบการใช้คำสั่งบังคับรถ	41
ตารางที่ 4.2 เปอร์เซ็นต์ เซ็นความถูกต้อง	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นของปริญญานิพนธ์

เนื่องจากในปัจจุบันได้มีผู้ป่วยที่พิการมีจำนวนมาก เช่น เป็นอัมพาต ขาขาด แขนขาด ไม่สามารถช่วยเหลือตัวเองได้ ทำให้เป็นผู้ที่ต้องการความช่วยเหลือจากสังคม และเป็นปัญหาสำหรับสังคมอย่างมาก ที่ต้องมีการแก้ไข จึงเกิดแนวคิดของปริญญานิพนธ์นี้ขึ้นมา ซึ่งใช้แนวการวิเคราะห์เสียงที่ไม่ขึ้นกับผู้พูด เป็นการพัฒนาระบบการรับคำสั่งของคอมพิวเตอร์โดยไม่เจาะจงเสียงผู้ออกคำสั่ง เป็นการเปิดกว้างให้กับใครก็ได้ที่ต้องการใช้อุปกรณ์นั้นสามารถออกคำสั่งควบคุมอุปกรณ์นั้นได้โดยผ่านระบบคอมพิวเตอร์ ทำให้เกิดความสะดวกสบายในกลุ่มของผู้ป่วย และเป็นการแก้ปัญหาสังคมด้วยโดยใช้ประโยชน์จากระบบคอมพิวเตอร์ ซึ่งในที่นี้ยังสามารถประยุกต์ใช้งานในด้านต่างๆ มากมาย เช่น ระบบการป้อนข้อมูลด้วยเสียง การสั่งงานยานพาหนะด้วยเสียง ควบคุมอุปกรณ์ต่างๆ ด้วยเสียง ได้ทั้งหมดโดยผ่านระบบคอมพิวเตอร์ โดยที่ไม่เลือกคนออกคำสั่งควบคุม

1.2 จุดมุ่งหมายของปริญญานิพนธ์

1. ศึกษาความเป็นไปได้วิธีการรู้จำเสียงพูด แบบไม่ขึ้นอยู่กับคำพูดโดยวิธีการเปรียบเทียบระหว่างแบบทดสอบกับแบบอ้างอิงโดยใช้ Visual Basic
2. เพื่อหาวิธีการวิเคราะห์เสียงพูดที่มีความแม่นยำสูงขึ้น
3. เพื่อใช้เป็นแนวทางและพื้นฐานความรู้ในการพัฒนาระบบรู้จำเสียงพูดที่เป็นคำพูดในภาษาไทย

1.3 ขอบเขต

1. การสั่งงานจะใช้คำสั่งเป็นคำสั่งภาษาอังกฤษซึ่งจะเป็นบุคคลใดทำการสั่งงานก็ได้
2. การทำงานต้องมีการพูดคำสั่งที่ถูกต้องเหมือนกับภาษาอังกฤษ
3. ถ้าคำสั่งในการสั่งไม่ถูกต้อง ฮาร์ดแวร์จะไม่แสดงผลออกมา
4. เอาต์พุตมี 8 เอาต์พุตต่อออกทางพอร์ต RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. เริ่มต้นจากการศึกษาหาข้อมูลจากแหล่งต่างๆ ที่เกี่ยวข้อง
2. ศึกษาระบบการพูดและการฟังของมนุษย์โดยการสังเกตุจากพฤติกรรมและข้อมูลที่เกี่ยวข้อง
3. กำหนดขอบเขตของการดำเนินงาน
4. เขียนโปรแกรมเพื่อการทดสอบและออกแบบฮาร์ดแวร์เพื่อใช้ในการทดสอบ
5. ทำการทดลองกับฮาร์ดแวร์และปรับปรุงแก้ไขให้เหมาะกับงานที่ใช้
6. บันทึกและสรุปผลการทดลอง

1.5 เครื่องมือที่ใช้ในการทดลอง

ฮาร์ดแวร์

1. ไมโครคอมพิวเตอร์ส่วนบุคคลใช้โปรเซสเซอร์รุ่น AMD K2 333MHz., หน่วยความจำหลัก 64 MB.
2. SOUND Card ขนาด 32 บิต ยี่ห้อ Creative Sound Blasster รุ่น Vibra 128
3. Hard Disk Drive ขนาด 2.1 Gbytes
4. ไมโครโฟน

ซอฟต์แวร์

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่ใช้ในการดำเนินงาน

โดยทั่วไปแล้วคนเราเปล่งเสียงออกมาตามสำเนียงในระบบภาษาของตนเอง แม้ว่าคนที่อยู่ในสังคมเดียวกัน ใช้ภาษาเดียวกัน แต่เสียงแต่ละคนที่เปล่งออกมานั้น จะมีลักษณะแตกต่างกันแม้คนๆ เดียวกันเปล่งเสียงคำเดียวกันสองครั้ง สัญญาณที่ได้ก็ยังมีความแตกต่างกัน แต่คนก็ยังแยกแยะความหมายของคำที่เกิดจากการพูดของแต่ละคนได้ การจดจำและแยกแยะคำพูดต่างๆ จึงดูเป็นเรื่องง่ายในคน แต่สำหรับที่จะให้คอมพิวเตอร์จำและแยกแยะเสียงพูดนั้นเป็นเรื่องยากมาก 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) หมายถึง อวัยวะที่ติดกับกระดูกคางส่วนบน ได้แก่ ริมฝีปากบน ฟันบน ปุ่มเหงือก เพดานแข็ง เพดานอ่อน ลิ้นไก่ และอวัยวะที่เป็นช่องว่าง ได้แก่ ช่องคอ ปาก ช่องจมูก

2.2 ลำดับการเกิดเสียง

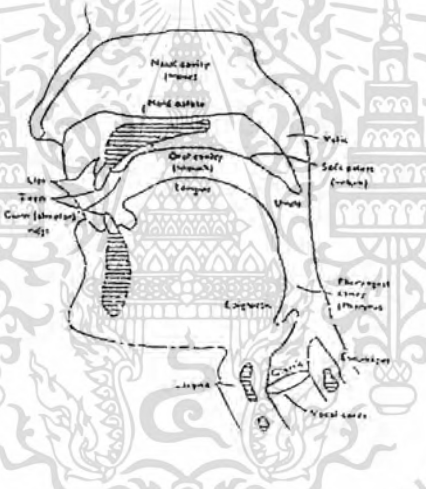
ลำดับการเกิดเสียงนั้นเราแบ่งออกได้เป็น 3 ขั้นตอนใหญ่ๆ คือ

1. จุดเริ่มต้น (Initiation) อวัยวะที่ใช้ในขั้นตอนนี้คือปอด ที่ขั้นตอนนี้ลมจะถูกขับออกจากปอด
2. การดัดแปลงลมที่เส้นเสียง (Phonation) อวัยวะที่ใช้ในขั้นตอนนี้ก็คือ อวัยวะที่อยู่ต่อจากปอดขึ้นไปจนถึงกล่อมเสียง เป็นขั้นตอนที่ลมจากปอดจะผ่านมาเข้าหลอดลมและกล่อมเสียงซึ่ง ณ ที่กล่อมเสียงนี้เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิดและปิดทำให้เกิดเสียง 2 ชนิด คือ ถ้าเส้นเสียงเปิดตลอดเวลาที่ลมผ่าน ลมจะผ่านออกมาได้อย่างสะดวก ซึ่งจะทำให้เกิดเสียงไม่ก้อง แต่ถ้าเส้นเสียงปิดกั้นลมไว้ ลมที่ผ่านออกมาจะเพิ่มแรงดันมากขึ้นจนเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นเสียงเปิดและปิดสลับกันไป ทำให้เกิดเสียงชนิดก้องและเรียกความถี่ในการเปิดและปิด ของเส้นเสียงว่า "ความถี่พื้นฐาน"

3. การเปลี่ยนแปลงลักษณะเส้นเสียง (Articulation) อวัยวะที่ใช้ในขั้นตอนนี้ก็คือ ส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปากและช่องว่างใหญ่อีก 2 ช่อง คือ ช่องปากและช่องจมูก ขั้นตอนนนี้ลมที่ออกจากกล่องเสียงจะถูกแปลงให้เกิดเสียงลักษณะต่างๆ ทั้งนี้เพราะช่องว่างภายในจะถูกเปลี่ยนแปลงขนาดตลอดเวลาที่เปลี่ยนแปลงเสียง และช่องจมูกจะถูกควบคุมด้วยเพดานอ่อน (Volume) ดังนั้นถ้าช่องว่างเหล่านี้เปลี่ยนขนาดไปจะทำให้เกิดความถี่กำทอน (Resonant Frequency) ของเสียงเปลี่ยนแปลงไปด้วย ซึ่งช่องว่างเหล่านี้เปรียบเสมือนท่อที่มีขนาดต่าง ๆ กัน



รูปที่ 2.1 อวัยวะที่ใช้ในการออกเสียง

2.3 หน่วยการจำเสียงพูด (Speed Recognition Unit)

หน่วยการจำเสียงพูดมีวิธีการจำแนกตามลักษณะการพูดไว้ 4 แบบ

1. ผู้พูดเดิม (Speaker Dependent Recognition) เป็นการที่ผู้พูดคนเดิมพูด เป็นพยางค์เพียงพยางค์เดียว เป็นคำ หรือประโยค โดยเราต้องสร้างหน่วยเปรียบเทียบที่เป็นเสียง ของคนๆ เดียว
2. ผู้พูดเป็นใครก็ได้ (Speaker Independent Reconition System) เป็นการที่ใครก็ได้ที่พูดจะเป็นพยางค์เดียว เป็นคำหรือเป็นประโยค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สอนให้เครื่องรู้จักคำในลักษณะของหน่วยเสียงหรืออย่างมากหน่วยพยางค์เป็นการสอนให้
เครื่องเข้าใจในเสียงที่มีการต่อเนื่องมากกว่าหนึ่งพยางค์ ต้นแบบจะถูกเก็บไว้ในหน่วยเสียง หรือพยางค์เท่านั้น
4. ผู้พูดสามารถพูดอย่างต่อเนื่องเหมือนการพูดแบบปกติ (Globolization) เครื่องจะต้องรู้จักและจำแนกคำที่มีการออกเสียงเชื่อมกัน การพูดแบบนี้ในมนุษย์เป็นเรื่องง่าย แต่สำหรับคอมพิวเตอร์เป็นเรื่องยากมาก เพราะการพูดแบบนี้ไม่สามารถหาเส้นแบ่งเขตระหว่างคำหรือพยางค์ได้แม่นยำ ดังนั้นการรู้จักโดยการเปรียบเทียบกับต้นแบบจึงไม่ใช่ของง่าย วิธีการแบบนี้ใช้กับเครื่องมินิคอมพิวเตอร์ขึ้นไป

2.4 Speech Technology

1. Speech Analysis

เป็นการวิเคราะห์สัญญาณรูปคลื่นของเสียงพูดโดยมีรูปแบบทั่วไปอยู่ 2 แบบ คือ การวิเคราะห์ Time domain และ Frequency domain โดย Time domain มีประโยชน์คือ นำส่วนหนึ่งของรูปคลื่นมาวิเคราะห์ตามระยะคาบเวลาได้อย่างง่าย และรวดเร็ว ง่ายต่อการเข้าใจ หลายตัวแปลที่สำคัญอย่างเช่น Amplitude และ Energy สามารถอ่านได้อย่างเด่นชัด ขณะที่ Frequency domain เป็นการวิเคราะห์ให้เห็นความแตกต่างของเสียง ถึงแม้ว่า Time domain เร็วแต่จะเหมาะกับรูปคลื่นที่มี wave form ไม่ซับซ้อน ส่วน Frequency domain สามารถวิเคราะห์คลื่นที่ซับซ้อนได้ละเอียดแม่นยำกว่า

เป็นหมายสำคัญอย่างหนึ่ง ของ Speech Analysis คือ การลดจำนวนข้อมูลลงเพื่อให้ประหยัดทั้งการเก็บและการส่ง เสียงพูดที่ยอมรับได้จะมี Rate ได้ต่ำถึง 600 bits/s อย่างไรก็ตามถ้า Bitrate ยิ่งต่ำการบิดเบือนของเสียงเล็กๆ ก็จะค่อยๆเพิ่มขึ้นเมื่อตอนที่ทำการแปลงข้อมูลกลับมาเป็นเสียง

2. Digital Coding

เป็นการเข้ารหัส เก็บรักษา และส่งข้อมูลอย่างมีประสิทธิภาพ และสามารถแปลงสัญญาณกลับเป็นเสียงได้อย่างถูกต้อง เป้าหมายที่สำคัญคือการใช้ Bandwidth ของการส่ง ให้น้อยที่สุด พร้อมกับมีระบบลดข้อผิดพลาดในขณะส่งที่ดี

3. Speech Synthesis

คือการสังเคราะห์เสียงจากข้อมูลที่เป็น เสียงหรือจากข้อมูลที่เป็นตัวอักษร โดยพยายามใช้หน่วยความจำและการประมวลผลให้น้อยที่สุด โดยการเข้ารหัสอย่างง่ายที่สุดมีอัตรา บิทเรทได้ที่ 100 Kbits/s สำหรับ Synthesizer ที่มีข้อมูลคำศัพท์ขนาดเล็กมาก แต่สำหรับ Synthesizer ที่มีข้อมูลคำศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดใหญ่แล้วการลดขนาด ของหน่วยความจำจะต้องเกี่ยวกับรูปแบบ ของความถี่ ร่องเสียง และเสียงสูงต่ำ ซึ่งหมายถึง การลดคุณภาพของเสียงนั่นเอง

4. Speech Recognition

เป็นอีกรูปแบบหนึ่งของการสื่อสารระหว่างคนกับคอมพิวเตอร์ ซึ่งเกี่ยวกับ การจดจำคำพูดโดยอัตโนมัติหรือส่วนหนึ่งคือ การแปลงจากคำพูดไปเป็นตัวอักษรนั่นเอง ความสำเร็จในส่วนของระบบการจดจำเสียงมีค่อนข้างน้อยเพราะ ความไม่สมบูรณ์ของ การแปลงคำพูดมีข้อจำกัดหลายอย่าง ส่วนใหญ่แล้วระบบจดจำเสียงจะถูกจำกัดจำนวนของผู้พูดที่มันสามารถรับรู้ได้รวมถึงจำกัดจำนวนคำและโครงสร้างประโยคที่ผู้พูดสามารถใช้ได้ บ่อยครั้งที่ผู้พูดจะต้องหยุดในระหว่างคำเพื่อรอให้ระบบเข้าใจเป็นการยากที่จะทำให้ระบบสามารถจดจำเสียงของหลาย ๆ คนได้ ระบบส่วนใหญ่เป็นระบบที่จำกัดคนพูด

5. Text – to – Speech (TTS) Systems

การแปลงจาก ตัวอักษรเป็นคำพูด เป็นระบบการแปลงแบบอัตโนมัติ ของประโยคที่เก็บไว้ไปเป็นคำพูดสังเคราะห์ โดยผ่านส่วนต่างๆของระบบ TTS ที่ถูกที่ปรับปรุงขึ้นโดยการใช้ หลากหลายหลักการทางด้านวิทยาศาสตร์เชิงพฤติกรรมมนุษย์ คุณลักษณะพฤติกรรมการพูดของมนุษย์เป็นตัวแปลที่สำคัญที่จะทำให้นำเทคโนโลยีนี้ทำงานในด้านต่างๆได้เหมาะสม

การประยุกต์ใช้งานต่างๆ

เทคโนโลยีนี้มีประโยชน์ที่สำคัญคือ การที่ทำให้มนุษย์สามารถสื่อสารกับคอมพิวเตอร์ โดยการพูดอย่างธรรมชาติของมนุษย์ การนำไปใช้งานมีหลายด้านเช่น เป็นเครื่องฝึก ตรวจสอบการอ่านระบบแจ้งเตือน เครื่องช่วยพูดสำหรับผู้พิการ เครื่องช่วยอ่านหรับคนตาบอด หรือเป็นเครื่องที่อ่านข้อมูลจากฐานข้อมูลให้ผู้ใช้ทางโทรศัพท์ อย่างเช่น ชื่อที่อยู่ สมุดโทรศัพท์ บัญชีเงินฝาก ราคาหุ้น พยากรณ์อากาศ การจอง หรือซื้อขายของ ระบบจะมีประโยชน์มากเมื่อ ใช้กับระบบที่มีการเปลี่ยนแปลงข้อมูลบ่อย โดย TTS จะช่วยลดเนื้อที่การเก็บข้อมูล จาก 64 Kb ต่อ 1 วินาที สำหรับการบันทึกเสียงเหลือ เพียงแค่ ไม่ถึง 1 Kb ในรูปข้อความ

ขั้นตอนการแปลง Text to Speech

หัวข้อนี้จะกล่าวถึงสิ่งที่นักวิทยาศาสตร์ทางด้านพฤติกรรมได้สร้างสรรค์ระบบให้สนับสนุนการแปลงประโยคให้เป็นเสียงพูดขึ้น

ขั้นตอนพื้นฐานในการเปลี่ยนประโยคเป็นคำพูดคือ

- Text normalization เป็นการแยกแยะประโยคที่รับเข้าไป อย่างเช่นเป็นตัวย่อหรือตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Syntactic parsing ทำการวิเคราะห์โครงสร้างของประโยค ว่ามีส่วนไหนบ้างเช่น คำนาม คำกริยา
- Word pronunciation เป็นการเลือกออกเสียง โดยยึดหลักจากตารางที่กำหนดไว้ อย่างเช่น การเน้นเสียงหนักของพยางค์ การออกเสียงขึ้นต้นหรือลงท้าย
- Determination of prosody เป็นการปรับระยะระหว่างตัวอักษรและคำเพื่อให้การพูดเป็นธรรมชาติ
- Speech synthesis ข้อมูลเกี่ยวกับเสียง ที่ได้จากขั้นตอนที่ผ่านมา ถูกเปลี่ยนให้เป็นโครงสร้างเสียง ที่ให้ผลเป็นเสียงที่ได้ยินได้

ถึงแม้ระบบปัจจุบัน TTS สามารถที่จะพูดคำใดๆ ก็ได้ในภาษาอังกฤษ แต่มันก็ไม่เข้าใจความหมายทำให้บางครั้งคำที่คลุมเคลือจึงไม่สามารถจะออกเสียงได้ถูกต้อง

ระบบเพื่อใช้พัฒนา TTS (Evaluation)

เพื่อให้ได้มาซึ่งระบบ TTS ที่มีประสิทธิภาพดี จำเป็นต้องมีการวิเคราะห์ ข้อมูลเพื่อนำมาทำการปรับปรุงส่วนต่างๆของระบบ TTS

ในบางงานนั้นระบบทดสอบจะถูกสร้างขึ้นมาเพื่อ ทำให้ได้ TTS ที่เหมาะสม ระบบทดสอบที่สร้างขึ้นมาเฉพาะการประยุกต์ใช้งานนั้นถือได้ว่ามีประโยชน์มาก ผลที่ได้จากระบบทดสอบที่เจาะจงนี้จะทำให้ได้ TTS ที่มีประสิทธิภาพสูงโดยสิ่งที่จะวิเคราะห์นั้นจะต้องคำนึงถึงหลายด้าน สิ่งแรกก็คือด้านสิ่งแวดล้อมเช่น สัญญาณรบกวน ต้องพิจรณาว่าผู้ใช้อยู่ในที่ชุมชนหรือส่วนตัว ส่วนที่สองคือ ตัวทดสอบควรจะเป็นข้อมูลประเภทเดียวกับที่นำไปใช้ อย่างเช่น ชื่อ ที่อยู่ ชื่อผลิตภัณฑ์ หรือ Email เป็นต้น นอกจากนั้นต้องคำนึงถึงผลกระทบที่มีต่อความรู้สึกของผู้ใช้ ว่าจะยอมรับระบบ TTS ที่จะนำมาแทนระบบที่มีอยู่แล้วได้หรือไม่

ตัวอย่างเช่น ระบบของบริษัทTT&Tที่เป็นการให้บริการฝากข้อความโฆษณาทางโทรศัพท์ จะสามารถใช้งานได้ดีกว่าการใช้คนที่เป็นตัวแทนไปติดต่อเพราะมันทำงานโดยปราศจากความผิดพลาดและอารมณ์ของมนุษย ์คนส่วนใหญ่ให้การยอมรับเสียงสังเคราะห์ที่เกิดจากระบบ TTS แต่สำหรับคนที่ไม่ยอมรับ การที่มันทำงานได้ 24 ชั่วโมงและช่วยลดต้นทุนก็เริ่มทำให้เขายอมรับมากขึ้น

องค์ประกอบของมนุษย์ที่มีผลเกี่ยวกับ TTS สำหรับงานเฉพาะอย่าง

นอกจากสิ่งที่กล่าวมาแล้วยังมีสิ่งที่ต้องคำนึงถึงเพิ่มเติม คือการนำไปประยุกต์ใช้งานเฉพาะด้านจะต้องมีสิ่งที่ต้องปรับปรุงเพิ่มเติมอย่างเช่นถ้าระบบต้องใช้ออกเสียงชื่อ ที่อยู่ ระบบควรจะถูกปรับปรุงให้ออกเสียงชื่อได้ถูกต้อง รวมไปถึงจังหวะ การอ่าน ให้มีความเป็นธรรมชาติมากที่สุด

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wording and Tuning TTS Announcements

การหยุดและการเรียงการออกเสียง ในจังหวะที่ถูกต้อง เป็นปัญหาที่สำคัญอีกอย่างหนึ่งของผู้ที่นำระบบไปติดตั้งและเป็นคนออกแบบคำพูด จะต้องคำนึงถึงความสนใจของผู้ฟัง และให้ได้การพูดที่เป็นธรรมชาติที่สุด

TTS เป็นระบบที่ทำให้ มนุษย์และคอมพิวเตอร์สามารถติดต่อกันได้ในแบบธรรมชาติของเสียงพูดที่ได้ยินได้ จะมีประโยชน์มากโดยเฉพาะอย่างยิ่งใช้กับข้อมูลขนาดใหญ่ที่มีการเปลี่ยนแปลงตลอดเวลาโดยที่เป็นไปไม่ได้ที่จะเก็บเอาข้อมูลในรูปเสียง การพัฒนาระบบ TTS แตกแขนงออกไปหลายด้านโดยมีนักวิทยาศาสตร์ทางด้านพฤติกรรมมนุษย์เป็นตัวจักรสำคัญในการพัฒนา โดยมีกระบวนการพื้นฐาน 5 ขั้นตอนคือ Text normalization คือการหาตำแหน่งเริ่มและจบของประโยค การแปลงตัวย่อและสัญลักษณ์ที่ไม่ใช่ตัวอักษร, Sysntactic Parsingคือ การแบ่งคำในประโยคออกเป็นส่วนๆ , Word Pronunciation คือการปรับเสียงให้เป็นไปตามพจนานุกรมเสียงและกฎของการออกเสียง, Determination of Prosodyคือ ปรับรูปแบบความถี่ห่างของการออกเสียงซึ่งสำคัญต่อการออกเสียงให้เป็นธรรมชาติ Speech Synthesis คือการแปลงข้อมูลที่ได้จากขั้นตอนข้างต้นให้เป็นข้อมูลเสียงที่สามารถฟังได้

2.5 ActiveX

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

2.6 ActiveX Control

ActiveX Control หมายถึงคอนโทรลที่เพิ่มเติมจากออบเจ็คพื้นฐานที่ Visual Basic ได้เตรียมไว้ให้แล้วใน Toolbox โดย ActiveX Control เหล่านี้มีอยู่มากมายทั้งที่มาพร้อมกับ Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และที่พัฒนาออกมาขายโดยบริษัทต่างๆ ทั่วโลก ซึ่ง ActiveX Control เหล่านี้เองคือพลังผลักดันให้โปรแกรม Visual Basic และเครื่องมืออีกหลายตัวของไมโครซอฟท์เป็นที่นิยมอย่างแพร่หลายเนื่องจากผู้ใช้มองตัว ActiveX Control เป็นเพียงกล่องดำหรือกล่องวัตถุ ซึ่งเพียงแค่นำคอนโทรลที่มีการทำงานตามที่ต้องการมาแปะบนฟอร์มแล้วเรียกใช้งานเท่านั้น โดยไม่ต้องสนใจการทำงานภายในของ ActiveX Control เหล่านั้น เช่น ถ้าเราต้องการเขียนโปรแกรมเพื่อติดต่อกับอิเตอร์เน็ต (Internet) ถ้าเป็นในอดีตต้องศึกษาข้อมูลมากมาย เช่น Protocol การติดต่อสื่อสาร TCP/IP ซึ่งเป็นมาตรฐานของภาษาที่คอมพิวเตอร์ใช้สื่อสารกันในระบบอินเตอร์เน็ต แต่ถ้าใช้ ActiveX Control เพียงแค่นำ Internet ActiveX Control มาวางลงบนฟอร์มแล้วเรียนรู้วิธีใช้เท่านั้นก็สามารถเขียนโปรแกรมที่ติดต่อกับเครือข่ายอิเตอร์น็ตได้ในเวลาอันสั้น

2.7 Microsoft Speech SDK

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

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

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 คอนโทรลที่ทำหน้าที่ รู้จำเสียง (Direct Speech Recognition และ Voice Command Control) มี 1 คอนโทรล ทำหน้าที่เขียนตามคำบอก (Voice Dictation) มี 2 คอนโทรลทำหน้าที่สังเคราะห์เสียง (Direct Speech Synthesis Control และ Voice Text Control) และสุดท้ายอีก 1 คอนโทรล ทำหน้าที่ติดต่อโทรศัพท์ (Voice Telephony Control)

ในคอนโทรล Direct Speech Recognition Control และ Dricet Speech Synthesis Control นั้นตัวคอนโทรลได้จัดการ การเข้าถึงข้อมูลในการติดต่อกับ Speech API (Application Programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InterFace) อย่างเต็มที่ สำหรับการสังเคราะห์เสียงและการรู้จำมันจะติดต่อกับเครื่องมือเสียงในระดับต่ำทำให้ทำงานได้เร็วขึ้นและควบคุมได้ง่าย

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

Voice Dictation Control เป็นการง่ายที่จะทำการเขียนตามที่ผู้ใช้บอกอย่างถูกต้องและแปลงเสียงเป็นข้อความตามปกติโดยอัตโนมัติ

Voice Telephony Control เป็นเครื่องมือในการติดต่อกับการสังเคราะห์เสียง การรู้จำเสียง การสังเคราะห์เสียงแบบ Wave และ DTMF บน Single หรือ Multi-line Voice ของอุปกรณ์โทรศัพท์

โดยจะขอกล่าวถึงเฉพาะ Voice Command Control ซึ่งเป็นคอลโทรลที่นำมาใช้เท่านั้นส่วนคอนโทรลตัวอื่น ๆ จะไม่ขอกล่าวถึง

Voice Command control

Properties

Voice Command Control มีคุณสมบัติดังต่อไปนี้

AutoGainEnable

AwakeState

CountCommands

Device

Enabled

EnableMenu

Hwnd

Initialized

LastError

MenuCreate

Microphon

Speaker

SRMode

SuppressExceptions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Threshold

Methods

Voice Command Control มี Method ดังต่อไปนี้

Activate

AddCommand

CmdMimic

DeaCtivate

EnableItem

Generaldlg

GetCommand

LexiconDlg

ListGet

MenuDelete

ReleaseMenu

Remove

SetCommand

TrainGeneralDlg

TrainMenuDlg

TrainMicDig

Events

Voice Command Control มี Events ดังต่อไปนี้

AttribChanged

ClickIn

CommandOther

CommandRecognize

CommandStart

Interference

MenuActivate

UtteranceBegin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UtteranceEnd

VUMeter

ในที่นี้จะกล่าวถึงเฉพาะ Properties , Method และ Events ที่ใช้ในการเขียนโปรแกรมเท่านั้น ส่วนที่ไม่ได้กล่าวถึงสามารถดูได้ใน Help ของ Microsoft Speech SDK 4.0 ซึ่งจะมีคำบรรยายต่างๆ โดยละเอียด

Properties ที่ใช้ในการเขียนโปรแกรม ได้แก่

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

CountCommand

รูปแบบคำสั่ง

CountCommand(menu As Long) As Long

คำอธิบาย

เป็นคำสั่งที่ใช้ในการรีเทอร์ค่าของคำสั่งบนเมนู ซึ่งเป็นคำสั่งที่สามารถอ่านได้อย่างเดียว ไม่สามารถทำอย่างอื่นได้

Enables

รูปแบบคำสั่ง

Enabled As Long

คำอธิบาย

TRUE ถ้าสามารถทำการรู้จำเสียงหรือ FALSE ถ้าไม่สามารถทำได้ ซึ่งต้องทำการเซ็ตเป็น TRUE เพื่อที่จะสามารถฟังเสียงจากไมโครโฟนได้

เมื่อการรู้จำเสียงไม่สามารถทำได้ตัวรู้จำเสียงก็จะไมาสามารถทำการรู้จำคำสั่งใดๆ จากเมนูต่างๆ ได้ แอพลิเคชันหนึ่งๆ จะใช้คุณสมบัติ Enalebled ซึ่งจะอนุญาติให้ผู้ทำการปิดความสามารถในการรู้จำเสียงได้เป็นการหยุดการรู้จำเสียงเป็นการชั่วคราว

ผู้ใช้สามารถใช้บัญชีรายชื่อคำสั่งควบคุมในการสั่งการเปิดและ ปิดระบบเสียง เมื่อใดก็ตามที่เสียงถูกเปิดหรือปิด ข้อมูลข่าวสาร WM_SPEECHSTARTED หรือ WM_SPEECHENDED ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งไปที่ระบบวินโดรซว์ระดับบน แอพลิเคชันจะสามารถใช้ข้อมูลข่าวสารนี้ในการตัดสินใจเมื่อจะเปิดหรือปิดตัว VOICE COMMANDS หรือ VOICE TEXT

การเซ็ต ENABLED ให้เป็น FALSE นั้นจะอนุญาตให้ผู้ใช้ปิดการรู้จำเสียง ดังนั้นจะไม่เกิดการรู้จำเสียง สำหรับตัวอย่าง ผู้ใช้อาจจะต้องการปิดการรู้จำเสียงจากไมโครโฟนในระหว่างการประชุมดังนั้นการรู้จำเสียงจะหยุดรอหรือใครบางคนจะเผลอพูดคำสั่งให้ปิดการรู้จำในเมนู

ถ้าแอพลิเคชัน VOICE NAVIGTION ถูกติดตั้งบนเครื่องคอมพิวเตอร์ของผู้ใช้ แอพลิเคชันจะไม่ต้องการเซ็ตสภาวะ ENABLED อย่างไรก็ตามมันจะต้องเรียกใช้ฟังก์ชันเพื่อเริ่มที่จะทำการรู้จำเสียง สำหรับตัวอย่าง ถ้าแอพลิเคชันได้รับคำสั่ง "Do you want to print thedocumen:?" มันจะทำการรู้จำเสียงและทำการตอบรับเสียงจากผู้ใช้

Initialized

รูปแบบคำสั่ง

initialized As Long

คำอธิบาย

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

ส่วนมากการใช้ METHODS และ PROPERTIES จะทำการเริ่มต้นคอนโทรลโดยอัตโนมัติถ้ามันไม่ได้ถูกผู้ใช้เซ็ตให้เริ่มต้นอยู่ก่อนแล้ว

MenuCreate

รูปแบบคำสั่ง

Menucreate(Appication As String, State As String, Flage As Long) As Long

คำอธิบาย

จะทำการสร้างเมนูเสียงขึ้นมาใหม่ หรือ ตัวเมนูเสียงอาจจะมีอยู่แล้วสำหรับแอพลิเคชันสามารถค่านได้อย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ความหมายของค่าพารามิเตอร์ต่างๆ ของ MenuCreate

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

ตารางที่ 2.2 ค่าของ flags ของคำสั่ง MenuCreate

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแอพลิเคชันทำการสร้างเมนูเสียงขึ้นมาโดยใช้ Method MenuCreate ซึ่งค่าของ VCMDMC_CREATE_NEW, VCMDMC_CREATE_AWAYS, VCMDMC_OPEN_ALWAYS ตัวคำสั่งเสียงจะทำการเก็บเมนูใหม่ลงในฐานข้อมูล เมื่อแอพลิเคชันทำการเริ่มต้นใหม่มันจะสามารถรักษาเวลาในการโหลดเมนูจากฐานข้อมูลแทนที่สร้างเมนูขึ้นมาใหม่

แอพลิเคชันสามารถสร้างตัวเมนูเสียงโดยการโหลดเมนูเสียงที่มีอยู่แล้วจากฐานข้อมูลของเมนูเสียงหรือทำการสร้างเมนูเสียงขึ้นใหม่ เมนูเสียงตัวหนึ่งจะไม่ต้องถูกเก็บลงในฐานข้อมูล แอพลิเคชันสามารถสร้างเมนูเสียงชั่วคราวโดยการเซ็ต dwFlags กับ VCMDMC_CREATE_TEMP เมนูเสียงชั่วคราวนี้จะอยู่ไปเรื่อย ๆ จนกว่าจะถูกปล่อยออกจากหน่วยความจำ

แอพลิเคชันสามารถสร้างตัวเมนูเสียงได้มากกว่าหนึ่งโดยทำเหมือนเดิม สำหรับตัวอย่างหนึ่งแอพลิเคชันต้องทำการระบุเมนูของแอพลบิเคชันอื่น ๆ

Methods ที่ใช้ในการเขียนโปรแกรม ได้แก่

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

ตารางที่ 2.3 ความหมายของค่าพารามิเตอร์ต่าง ๆ ของ Activate

พารามิเตอร์	คำบรรยาย
Menu	คำสั่งเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AddCommand

รูปแบบคำสั่ง

AddCommand(Menu As Lond, id AsLong, 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

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

ถ้าคำสั่งข้อความประกอบด้วยลำดับรายชื่อ สามารถเรียกใช้ ListSet ทำการเซ็ตคำพูดที่ผู้ใช้สามารถใช้แทนรายชื่อเมื่อกำลังพูดคำสั่ง

ตารางที่ 2.4 ความหมายของค่าพารามิเตอร์ต่าง ๆ ของ AddCommand

พารามิเตอร์	คำบรรยาย
Menu	เมนูเสียง
Id	คำสั่งที่เหมือนกัน แอพลิเคชันสามารถระบุคำสั่งที่เหมือนกันแต่ละคำสั่งเสียงในเมนู คำสั่งที่เหมือนกัน แอพลิเคชันจะกำหนดค่าพิเศษของคำสั่งภายในเมนู พารามิเตอร์นี้สามารถแก้ไขคำสั่งที่เหมือนกัน หรือมันสามารถถูกใช้สำหรับเหตุการณ์เมื่อแอพลิเคชันรับเหตุการณ์ แอพลิเคชันจะสามารถพิจารณาคำสั่งที่เหมือนกันในการตัดสินใจคำสั่งที่เป็นส่วนสำคัญของเหตุการณ์
Command	คำสั่งเสียงที่เป็นข้อความทุก ๆ คำสั่งเสียงจะต้องมีคำสั่งข้อมความ เช่น "Open a file" ในระหว่างกระบวนการรู้จำเสียงจะทำการแปลสัญญาณออดิโอที่รับเข้ามาไปเป็นข้อความและเปรียบเทียบมันกับคำสั่งข้อความในเมนูเสียงที่พร้อมที่จะทำงาน ภายในคำสั่งข้อความ จะมีสัญลักษณ์ที่มีความหมายพิเศษดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

	<> แสดงรายชื่อของคำหรือถ้อยคำโดยสามารถพูดเป็นคำสั้น ๆ สำหรับตัวอย่างคำสั่งข้อความ "Send mail to <name>" บรรจุรายชื่อที่ถูกเรียก "name" ซึ่งการเพิ่มคำพูดในรายการจะใช้ Method ListSet {} ถูกสำรองไว้ในอานาคต [] ถูกสำรองไว้ใช้ในอนาคต ตัวรู้จำคำสั่งจะสนับสนุนสัญลักษณ์ที่ใช้แทนคำสั่งข้อความ การใช้สัญลักษณ์จะอนุญาตให้ผู้ใช้พูดคำบางคำในระหว่างคำสำคัญในถ้อยคำนั้น สำหรับตัวอย่าง ถ้าแอพลิเคชันนิยายคำสั่งเสียงของ "{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 แอพลิชันจะเป็นตัวแสดงความหมายของคำสั่งข้อความ
Category	ข้อความที่บอกถึงลำดับชั้นซึ่งคำสั่งนี้จะเป็นส่วนหนึ่งของคำสั่งในเมนูเสียงจะทำการรวบรวมความแตกต่างของลำดับชั้นจะช่วยให้ผู้ใช้เลือกตลอดทั้งรายการของคำสั่งอย่างง่ายดาย ซึ่งมันจะมีแนวคิดคล้ายกับเมนูวินโดว์ซึ่งจะรวมเอาคำสั่งต่าง ๆ ภายใต้ชื่อเมนู เช่น "File" , "Edit" , "View" และอื่น ๆ สำหรับผลที่ดีที่สุดจะต้องใช้การแบ่งลำดับชั้น 20 หรือ น้อยกว่านี้
Flage	Flage จะแสดงข้อมูลข่าวสารเกี่ยวกับคำสั่ง ค่าต่าง ๆ ดูในตารางที่ 5
Action	เป็นบล็อกของข้อมูลที่ถูกส่งไปให้แอพลิเคชันเมื่อพูดคำสั่ง ข้อมูลพร้อมที่จะให้แอพลิเคชันตีความหมายซึ่งข้อมูลจะรวมไปกับคำสั่งในเมนูเสียง ตัวจัดการจะส่งข้อมูลไปที่แอพลิเคชันเมื่อคำสั่งถูกการรู้จำเกิดขึ้น ข้อมูลที่พร้อมนี้จะมีลักษณะพิเศษดังนั้นแอพลิเคชันต้องรู้ว่าจะทำอย่างไรกับคำสั่งเมื่อมีการสั่ง ข้อมูลจะถูกส่งผ่านไปกับคำสั่งโดยคอนโทรลจะไม่สามารถแปลความหมาย มันขึ้นอยู่กับการตัดสินใจของแอพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42247

ตารางที่ 2.5 ค่าของ Flage ของคำสั่ง AddCommand

ค่าของ Flage	คำบรรยาย
VCMDCMD_DISABLED_PERM	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง PROPERTIES EnableMenu ดังนั้นคำสั่งเสียงจะไม่รู้จำมัน คำสั่งจะไม่รวบรวมในเมนูคำสั่ง
VCMDCMD_DISABLED_TEMP	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง METHOD SetCommand คำสั่งจะถูกรวบรวมในเมนูเสียง อย่างไรก็ตามมันจะสามารถ ENABLED อีกครั้งโดยปราศจากการรวบรวมของเมนู
VCMDCMD_VERIFY	แอพลิเคชันจะต้องพร้อม โดยผู้ใช้จะทำการพิสูจน์คำสั่งก่อนเอามันออก
VCMDCMD_CANTRENAME	บอกถึงคำสั่งอัตโนมัติทั่วไปโดยไม่อนุญาตให้ผู้ใช้เปลี่ยนชื่อคำสั่ง

Deactivate

รูปแบบคำสั่ง

Deactivate(Meu As Long)

คำอธิบาย

บอกให้ตัวรู้จำทำการหยุดฟังคำสั่งและคืนทรัพยากรไมโครโฟน ซาวด์การ์ด ถ้าผู้ใช้ยังออกคำสั่งและตัวรู้จำไม่ทำตามจะเกิด Error ขึ้น

ตารางที่ 2.6 ความหมายของค่าพารามิเตอร์ต่าง ๆ ของ Deactivate

พารามิเตอร์	คำบรรยาย
Menu	เมนูคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ตารางที่ 2.7 ความหมายของค่าพารามิเตอร์ต่าง ๆ ของ GetCommand

พารามิเตอร์	คำบรรยาย
Menu	ถ้ามีค่าเป็น TRUE จะเป็นการอ่านคำสั่งจากคำสั่งโกลบอลมันจะถูกใช้ในทุก ๆ แอพลิเคชัน ถ้ามีค่าเป็น FALSE มันจะเข้าถึงข้อมูลคำสั่งสำหรับแอพลิเคชัน โดยทั่วไปแอพลิเคชันจะไม่ทำการเปลี่ยนแปลงแก้ไขคำสั่งโกลบอล
Index	หมายเลขของคำสั่งที่ได้รับมา หมายเลขของคำสั่งจะเป็นลำดับจาก 1 ถึง n
Command	คำสั่งเสียงที่เป็นข้อความทุก ๆ คำสั่งเสียงจะต้องมีคำสั่งข้อความ เช่น “Open a file” ในระหว่างกระบวนการรู้จำเสียงจะทำการแปลสัญญาณออดิโอที่รับเข้ามาไปเป็นข้อความและเปรียบเทียบมันกับคำสั่งข้อความในเมนูเสียงที่พร้อมที่จะทำงาน ภายในคำสั่งข้อความจะมีสัญลักษณ์ที่มีความหมายพิเศษดังต่อไปนี้ < > แสดงรายชื่อของคำหรือถ้อยคำโดยสามารถพูดเป็นคำสั้น ๆ สำหรับตัวอย่าง คำสั่งข้อความ “Send mail to <name>” บรรจุรายชื่อที่ถูกเรียก “name” ซึ่งการเพิ่มคำพูดในรายการจะใช้ Method ListSet {} ถูกสำรองไว้ใช้ในอนาคต [] ถูกสำรองไว้ใช้ในอนาคต ตัวรู้จำคำสั่งจะสนับสนุนสัญลักษณ์ที่ใช้แทนคำสั่งข้อความ 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Category	ข้อความที่บอกถึงลำดับชั้นซึ่งคำสั่งนี้จะเป็นส่วนหนึ่งของคำสั่งในเมนูเสียงจะทำการรวบรวมความแตกต่างของลำดับชั้นจะช่วยให้ผู้เลือกตลอดทั้งรายการของคำสั่งได้อย่างง่ายดาย ซึ่งมันจะมีแนวคิดคล้ายกับเมนูวินโดว์ซึ่งจะรวมเอาคำสั่งต่าง ๆ ภายใต้ชื่อเมนู เช่น "File", "Edit" , "View" และอื่น ๆ สำหรับผลที่ดีที่สุดจะต้องใช้กรแบ่งลำดับชั้น 20 หรือ น้อยกว่านี้
Flags	Flags จะแสดงข้อมูลข่าวสารเกี่ยวกับคำสั่ง ค่าต่างๆ ดูในตารางที่ 8
Action	เป็นบล็อกของข้อมูลที่ถูกส่งไปให้แอพลิเคชันเมื่อพูดคำสั่ง ข้อมูลพร้อมที่จะให้แอพลิเคชันตีความหมายซึ่งข้อมูลจะรวมไปกับคำสั่งในเมนูเสียง ตัวจัดการจะส่งข้อมูลไปที่แอพลิเคชันเมื่อคำสั่งถูกการรู้จำเกิดขึ้น ข้อมูลที่พร้อมนี้จะมีลักษณะพิเศษ ดังนั้นแอพลิเคชันต้องรู้ว่าจะทำอย่างไรกับคำสั่งเมื่อมีการสั่ง ข้อมูลจะถูกส่งผ่านไปกับคำสั่งโดยคอนโทรลจะไม่สามารถแปลความหมาย มันขึ้นอยู่กับการตัดสินใจของแอพลิเคชัน

ตารางที่ 2.8 ค่าของ Flags ของคำสั่ง GetCommand

ค่าของ Flags	บรรยาย
VCMDCMD_DISABLED_PERM	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง PROPERTIES EnableMenu ดังนั้นคำสั่งเสียงจะไม่รู้จำมัน คำสั่งจะไม่รวบรวมในเมนูคำสั่ง
VCMDCMD_DISABLED_TEMP	คำสั่งจะถูก DISABLED โดยจะใช้ตัวควบคุมของคำสั่งเสียง METHOD SetCommand คำสั่งจะถูกรวบรวมในเมนูเสียง อย่างไรก็ตามมันจะสามารถ ENABLED อีกครั้งโดยปราศจากการรวบรวมของเมนู
VCMDCMD_VERIFY	แอพลิเคชันจะต้องพร้อม โดยผู้ใช้จะทำการพิสูจน์คำสั่งก่อนรเอามันออก
VCMDCMD_CANTRENAME	บอกคำสั่งอัตโนมัติทั่วไปโดยไม่อนุญาตให้ผู้ใช้เปลี่ยนชื่อคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ReleaseMenu

รูปแบบคำสั่ง

ReleaseMenu(Menu As Long)

คำอธิบาย

ปล่อยเมนูเสียงออกจากหน่วยความจำ ซึ่งสามารถเรียกคำสั่ง ReleaseMenu สำหรับทุก ๆ เมนูคำสั่งที่สร้างขึ้นได้

ตารางที่ 2.9 ความหมายของค่าพารามิเตอร์ต่าง ๆ ของ ReleaseMenu

พารามิเตอร์	คำบรรยาย
Menu	เมนูคำสั่ง

Events ที่ใช้การเขียนโปรแกรม ได้แก่ 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 ความหมายของค่าพารามิเตอร์ต่าง ๆ ของ CommandRecognize

พารามิเตอร์	คำบรรยาย
ID	เช่นเดียวกับคำสั่งที่ถูกรู้จำ
CmdName	ชื่อของเมนูซึ่งบรรจุชื่อของคำสั่งที่ถูกรู้จำ
Flags	VCMDCMD_VERIFY ถ้าแอพลิเคชันต้องการพิสูจน์จากผู้ใช้หรือไม่มีผลถ้าการพิสูจน์นั้นถูกปฏิเสธ ความต้องการในการพิสูจน์นั้นแอพลิเคชันจะแสดงไดอะล็อกบล็อกขึ้นมา แอพลิเคชันจะทำการพิสูจน์สำหรับทำลายหรือกลับไม่ได้ของคำสั่ง เช่น “Format disk”
Action	ข้อความที่บรรจุข้อมูลมากับคำสั่งที่ถูกรู้จำ
NumLists	หมายเลขของรายการที่ถูกรู้จำ ถ้าคำสั่งไม่ได้บรรจุรายการอะไรมาเลย พารามิเตอร์นี้จะเป็นศูนย์
ListValues	รายการของหนึ่งข้อความหรือมากกว่าที่ถูกแยก สำหรับตัวอย่าง ถ้าคำสั่งเป็น “Set the time to numble AM or PM” พารามิเตอร์จะเป็น “ten/PM”
Command	คำสั่งข้อความสำหรับคำสั่งที่ถูกรู้จำ

ผู้ใช้จะไม่ใช้ข้อความภายในของคำสั่งที่เหมือนกับคำสั่งที่ถูกรู้จำ แทนที่จะใช้ข้อมูลใน Action หรือ ข้อมูลที่เหมือนกันใน ID ในการตัดสินใจกับคำสั่งที่ถูกรู้จำ คำสั่งข้อความจะไม่บรรจุข้อความที่มีลักษณะเฉพาะ เพราะว่ามันเป็นไปได้สำหรับผู้ใช้ที่จะแก้ไขข้อความสำหรับคำสั่ง สำหรับแอพลิเคชันที่ใช้ไมโครซอฟท์วอยซ์หรือ ๆ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ฮาร์ดแวร์ (Hardware)

3.1 การทำงานของบอร์ดอินเตอร์เฟต

บอร์ดอินเตอร์เฟตจะประกอบด้วย 3 ส่วนคือส่วนที่หนึ่งเป็นส่วนที่ติดต่อกับพอร์ตอนุกรม RS-232 ใช้ไอซีเบอร์ MAX 232 ทำหน้าที่เป็นบัฟเฟอร์ให้กับพอร์ตอนุกรม RS-232 และจะแลกเปลี่ยนระดับข้อมูลจากระดับแรงดัน 5 โวลท์ ไปเป็นแรงดัน +10, -10 โวลท์ ในมาตรฐานการส่งข้อมูลแบบ RS-232C และในทางกลับกันก็จะทำหน้าที่เปลี่ยนระดับมาตรฐาน RS-232Cให้เป็นระดับแรงดัน5โวลท์โดยที่ภายในIC1จะมีวงจรทวีแรงดันและวงจรอินเวอร์เตอร์ซึ่งจะต้องต่อกับตัวเก็บประจุ C1,C2,C3 และ C4 ไว้ภายนอก

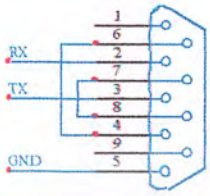
ในส่วนที่สองคือส่วนสร้างสัญญาณนาฬิกาโดยเราใช้ IC2 บอร์ 74HCT4060 เพื่อเป็นตัวกำหนด
ค่าช่วงเวลา BAUD RATEให้กับตัว UART เบอร์ 6402 เริ่มดูวงจรจากส่วนสร้างสัญญาณนาฬิกา คริสตอลค่า 2.4576 MHz สร้างความถี่ผ่านไอซีเบอร์74HCT4060 เพื่อหารความถี่และเลือกความถี่เพื่อให้ได้ค่า BAUD RATE ตรงกันกับที่คอมพิวเตอร์ ส่งมาซึ่งค่าที่หารออกมาจะมีค่า BAUD RATE ต่างๆ ที่ใช้งานกันบ่อยๆ คือ ค่า 9600, 4800, 2400, 1200 บิตต่อวินาที ซึ่งปกติที่ใช้อยู่ก็คือ 9600 บิตต่อวินาที ดังนั้นเวลาใช้งานก็จัมพ์ SK ไว้ที่ 9600 บิตต่อวินาทีได้เลย

ในการทำงานส่วนที่สามใช้ IC3 เบอร์ UART 6402 ซึ่งจะทำหน้าที่แปลงการติดต่อข้อมูลแบบอนุกรมไปขนานและสามารถแปลงข้อมูลแบบขานไปเป็นแบบอนุกรม โดยจะใช้วิธีการส่งแบบอะซิโครนัส

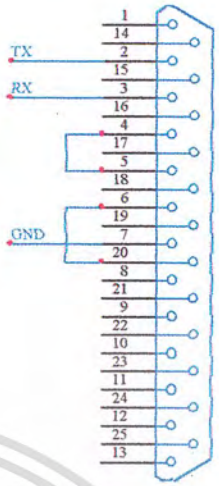
เมื่อเรารวมส่วนต่างๆ เข้าไว้ด้วยกันจะมีการกำงานดังนี้ คือเราจะใช้ UART 6402 โดยการแปลงข้อมูลแบบอนุกรมให้เป็นขนาน โดยจะรับข้อมูลที่ขา 20 แล้วทำการแปลงข้อมูลอนุกรมเป็นขนานออกแสดงผลทรี่ขา 5-12 โดยจะแสดงผลโดย LED ทั้ง 8 ตัว ซึ่งขึ้นอยู่กับว่าข้อมูลที่ได้มานั้นที่เป็นข้อมูล 8 บิท จะทำให้ LED แสดงผลอ่อย่างไรตามข้อมูลที่รับมา

สัญญาณที่ขาข้าวของขา 26-33 จะเป็นข้อมูลแบบขนาน โดยเป็นข้อมูลที่ส่งมาจากบอร์ดเพื่อส่งสัญญาณไปที่คอมพิวเตอร์ได้ แล้วทำการจัดเรียงเป็นแบบอนุกรมออกทางขา 25 ผ่านทาง MAX232 เข้าสู่คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

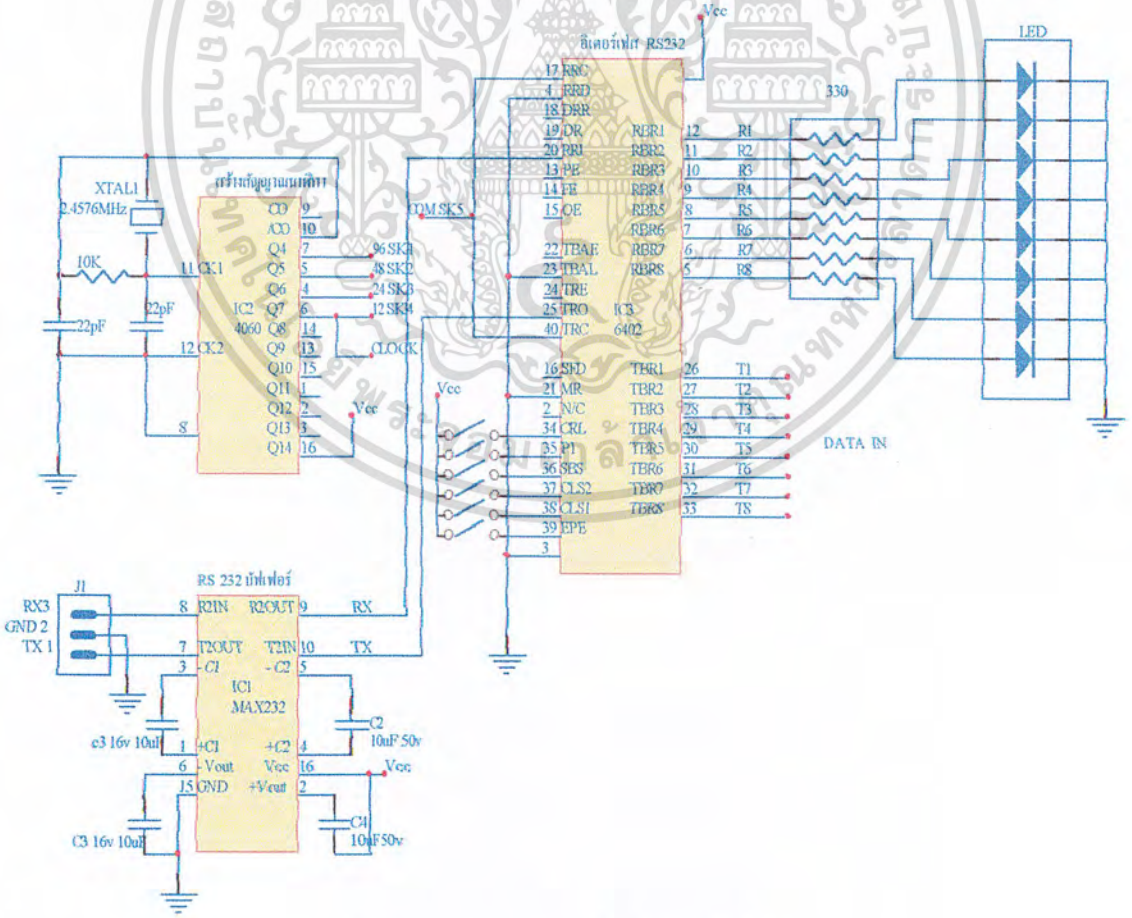


(ก) การต่อพอร์ตอนุกรมแบบ DB9



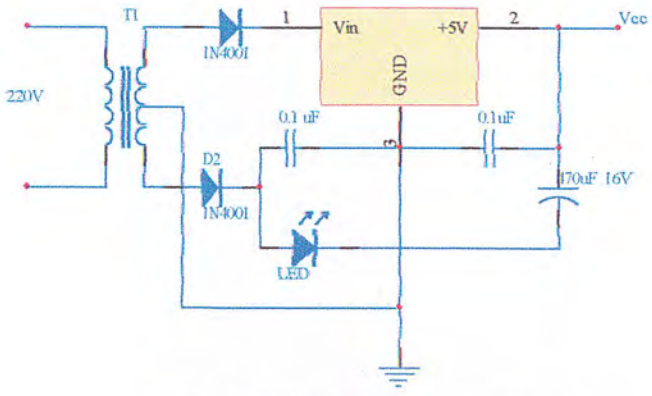
(ข) การต่อพอร์ตอนุกรมแบบ DB25

รูปที่ 3.1 การต่อพอร์ตอนุกรมแบบ DB9 และ DB25



รูปที่ 3.2 วงจรของบอร์ดอินเตอร์เฟส

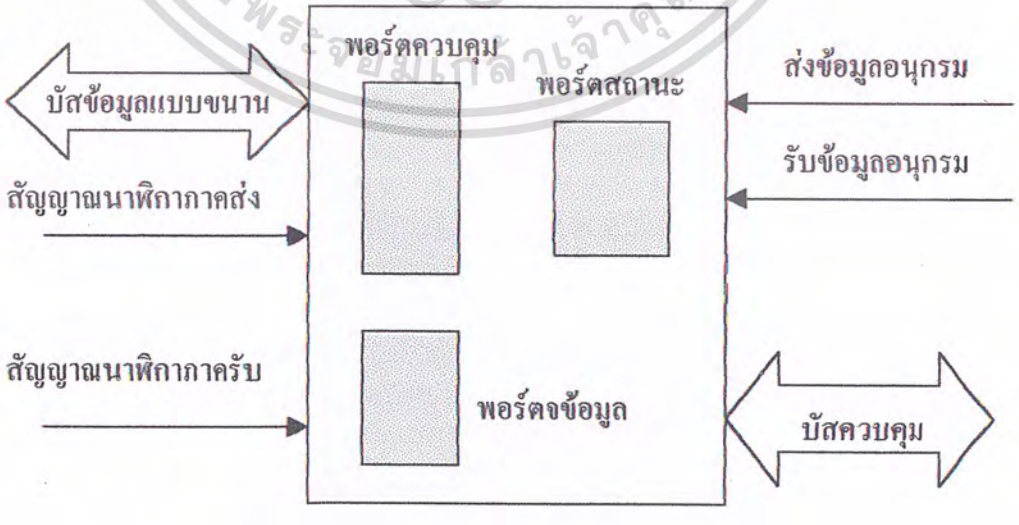
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจร POWER SUPPLY

3.2 UART

UART ย่อมาจาก Universal Asynchronous Receiver Transmiter เป็นระบบการเปลี่ยนการส่งข้อมูลจากแบบอนุกรมไปเป็นขนาน หรือแบบขนานไปเป็นแบบอนุกรม โดยจะใช้การส่งแบบอะซิงโครนัส ซึ่งไม่ต้องส่งสัญญาณนาฬิกาไปพร้อมกับจตัวข้อมูลและจะใช้การส่งบิตสตาร์ต (Start Bit) ซึ่งโดยทั่วไปจะมีค่าเป็น "0" และปิดท้ายด้วยบิตสต็อป (Stop Bit) ซึ่งโดยทั่วไปจะมีค่าเป็น "1" บล็อกไดอะแกรมการทำงานโดยทั่วไปของมันเป็นดังรูปที่ 3.3 โดยมันจะสามารถรับข้อมูลเข้าทั้งแบบอนุกรมและแบบขนาน และส่งออกได้ทั้งแบบอนุกรมและแบบขนาน โดยในการแปลงสัญญาณจะใช้สัญญาณนาฬิกาจากภายนอกเป็นตัวกำหนดค่าคาบเวลาเพื่อเป็นมาตรฐานการส่งหรือมาตรฐานการรับ



รูปที่ 3.3 บล็อกไดอะแกรมการทำงานพื้นฐานของ UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน UART นั้นจะทำให้การรับส่งข้อมูลสามารถส่งได้ไกลขึ้นโดยอาศัยการแปลงข้อมูลจากแบบอนุกรมมาเป็นแบบขนานแล้วใช้มาตรฐานการเชื่อมต่อจข้อมูลต่างๆ เช่น RS-232 ส่งข้อมูลออกไปจากนั้นด้านรับก็จะทำการแปลงมาตรฐานการเชื่อมต่อเหล่านี้ออกมาเป็นสัญญาณระดับทีทีแอล ให้ UART สามารถรับรู้และแปลงข้อมูลแบบอนุกรมไปเป็นขนานเพื่อใขช้งานต่อไป เช่นเดียวกับบอร์ดนี้จะใช้ UART เบอร์ CDP6402 เป็นตัวทำหน้าที่แปลงข้อมูลจากแบบอนุกรมเป็นขนาน

ทำความรู้จัก CDP6402

CDP6402 เป็นไอซี UART แบบ CMOS ที่ทำหน้าได้ทั้งเปลี่ยนข้อมูลแบบอนุกรมไปเป็นขนานและเปลี่ยนข้อมูลแบบขนานไปเป็นแบบอนุกรม โดยจะสามารถใช้กับข้อมูลในรูปแบบต่างๆ กันได้ขึ้นอยู่กับการกำหนดโหมดการทำงานของมันสามารถกำหนดได้ดังตารางที่ 3.1 และสำหรับรายละเอียดภายในตัวมันก็สามารถแสดงได้ดังในรูปที่ 3.4 สามารถแบ่งการทำงานของมันออกเป็น 3 ส่วนด้วนกัน

ตารางที่ 3.1 การกำหนดโหมดเพื่อควบคุมการทำงานของไอซี 6402

ขาควบคุม							
CLS2	**CLS1**	**PI**	**EPE**	**SBS**	**ขนาดข้อมูล**	**พาริตี้บิต**	**STOP BIT**
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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

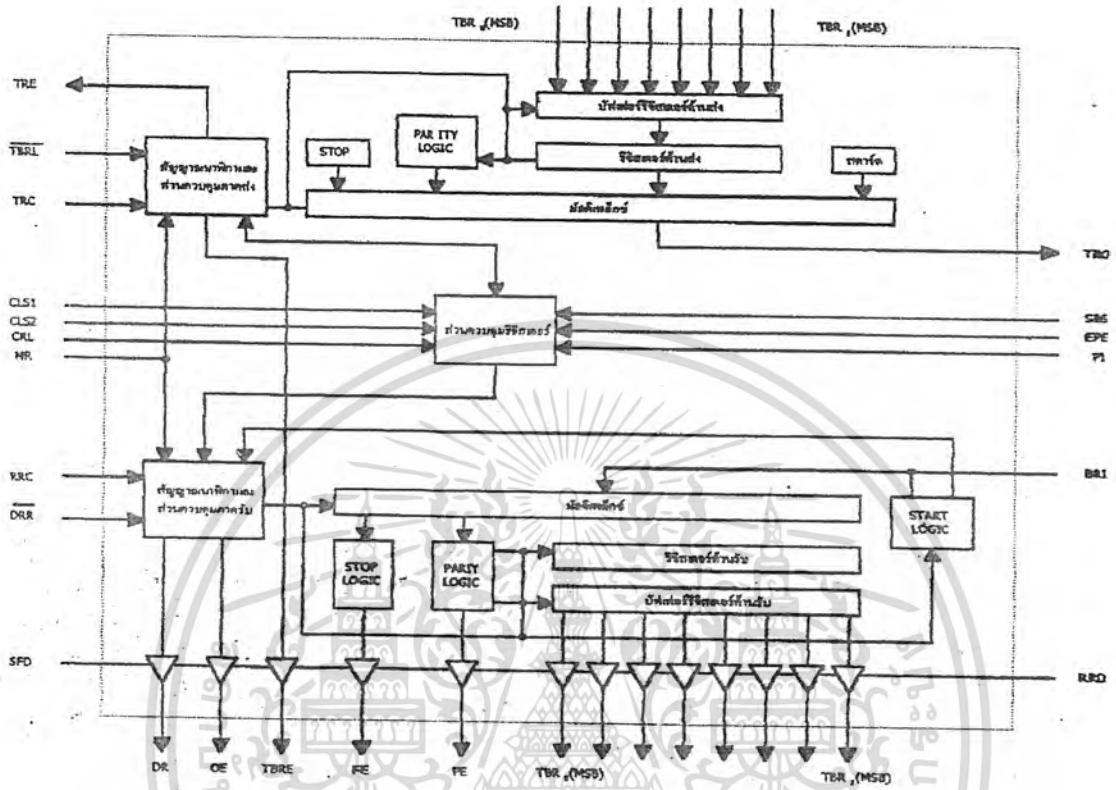
H	L	L	L	H	7	คี่	2
H	L	L	H	L	7	คู่	1
H	L	L	H	H	7	คู่	2
H	L	H	X	L	7	ไม่ใช้	1
H	L	H	X	H	7	ไม่ใช้	2
H	H	L	L	L	8	คี่	1
H	H	L	L	H	8	คี่	2
H	H	L	H	L	8	คู่	1
H	H	L	H	H	8	คู่	2
H	H	H	X	L	8	ไม่ใช้	1
H	H	H	X	H	8	ไม่ใช้	2

ส่วนแรก เป็นส่วนการแปลงข้อมูลแบบขานนไปเป็นแบบอนุกรมจะมีขา TER1 ถึง TER8 เป็นขารับข้อมูลแบบขนานเข้ามาผ่านบัฟเฟอร์รีจิสเตอร์และรีจีสเตอร์ และจะมีส่วนชิงพาริตี้เพื่อคอยตรวจสอบค่าพาริตี้ด้วย จากนั้นก็จะใช้จรมัลติเพล็กซ์ทำการแปลงข้อมูลแบบขานไปเป็นข้อมูลแบบอนุกรม โดยจะมีส่วนสตาร์ตและสต็อปบิตเพิ่มขึ้นมาด้วย การข้อมูลนี้จะใช้สัญญาณนาฬิกาจากขา TRC มาเป็นตัวกำหนดมาตรฐานการส่งข้อมูลและเมื่อแปลงครบ 1 ไบต์ก็จะส่งพัลซ์ออกทางขา TRE สัญญาณที่แปลงได้จะถูกส่งออกทาสงขา TRO

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

ส่วนที่สาม เป็นส่วนของการกำหนดโหมดการทำงานเพื่อให้ UART แปลงข้อมูลได้ตามมาตรฐานต่างๆ โดยให้ "H" จะหมายถึงสภาวะ HIGH และ "L" จะหมายถึงสภาวะ LOW และ "X" จะหมายถึง สามารถทำงานได้ทั้งสองสภาวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 บล็อกไดอะแกรมการทำงานภายในไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

โครงงานการควบคุมรถเข็นไฟฟ้าด้วยเสียงพูด (Control of electric wheelchair by sound) แบ่งออกเป็น 2 ส่วนด้วยกัน คือ

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

4.1 ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์ประกอบไปด้วย 2 ส่วนใหญ่ ๆ คือ ส่วนที่ทำหน้าที่ติดต่อกับคอมพิวเตอร์ในการรับข้อมูลและส่วนที่ทำหน้าที่ในการแสดงผลข้อมูลที่รับมาจากคอมพิวเตอร์โดยข้อมูลที่รับมานั้นจะเป็นข้อมูลอนุกรมขนาด 8 บิต ซึ่งถูกส่วนที่ทำหน้าที่ติดต่อกับคอมพิวเตอร์ทำการแปลงข้อมูลให้เป็นแบบขนานแล้วส่งไปยังส่วนแสดงผล

4.1.1 การทดสอบฮาร์ดแวร์

ในการทดสอบฮาร์ดแวร์นั้นจะทำการทดสอบโดยการส่งข้อมูลจากคอมพิวเตอร์ไปที่ฮาร์ดแวร์แล้วฮาร์ดแวร์แสดงผลได้ตามต้องการ โดยจะต้องมีโปรแกรมในการทดสอบฮาร์ดแวร์แบบง่ายๆ เพื่อให้แน่ใจว่าฮาร์ดแวร์ที่ทำขึ้นนั้นสามารถใช้งานได้จริงตามต้องการ 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้มีลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Private Sub Command3_Click()
  ' 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

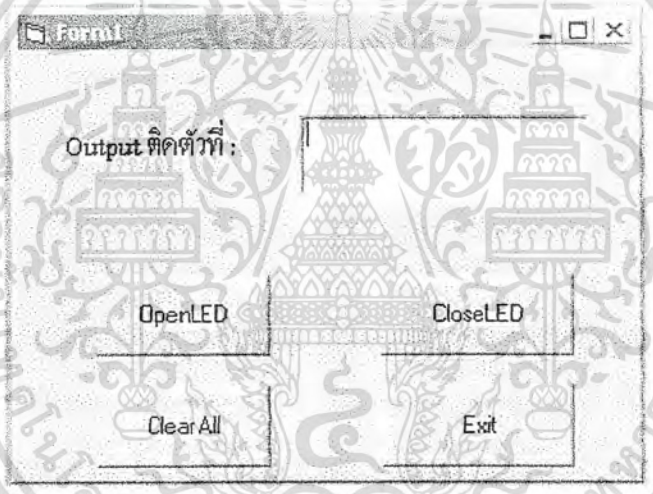
```
    b = "252"
    d = Val(c) And Val(b)
    c = d

  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"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
d = Val(c) And 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 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm1.Output = Chr$(0)
' ปิดพอร์ต
MSComm1.PortOpen = False

   b = "0"
   c = "0"
   d = "0"

End Sub
```



รูปที่ 4.1 เมนูของโปรแกรมทดสอบฮาร์ดแวร์

จากรูปที่ 4.1 เมื่อป้อนอินพุตซึ่งเป็นเลข 1-8 ในช่องรับอินพุตแล้วทำการกดปุ่ม Open LED โปรแกรมจะทำการส่งข้อมูลไปที่ฮาร์ดแวร์ตามที่ต้องการซึ่งที่อุปกรณ์ตัวตรงกับที่ป้อนเข้าไปตามหมายเลขก็จะทำงานซึ่งสามารถตรวจเช็คได้ว่าฮาร์ดแวร์นั้นทำงานหรือไม่ ถ้ากดปุ่ม Close LED อินพุตที่ป้อนให้กับโปรแกรมก็จะถูกส่งไปฮาร์ดแวร์เพื่อทำการปิดอุปกรณ์นั้นได้ ส่วนปุ่ม Clear All นั้นถ้าถูดกดโดยจะไม่สนใจอินพุตที่ป้อนให้ โปรแกรมจะส่งข้อมูลไปให้ฮาร์ดแวร์ทำการปิดอุปรณ์ทุกชนิดที่กำลังเปิดอยู่ ซึ่งเงื่อนไขต่างๆ ที่อยู่ในการทดสอบนั้นจะตรงตามความต้องการ คือ ถ้าสั่งให้อุปกรณ์ชนิดใดทำงานแล้วยังไม่สั่งให้หยุด อุปกรณ์จะทำงานต่อไปเรื่อย ๆ จนกว่าจะมีการสั่งงานมาจากคอมพิวเตอร์ โดยที่แม้จะสั่งอุปกรณ์ตัวอื่น ๆ ให้ทำงานก็จะไม่ส่งผลกระทบต่ออุปกรณ์ตัวที่ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ถูกสั่งเลย ส่วนปุ่ม Exit เมื่อกดปุ่มนี้แล้วก็จะออกจากโปรแกรมโดยที่จะทำการปิดอุปกรณ์ฮาร์ดแวร์ทุกตัวด้วยเมื่อไม่มีการทำงาน

4.2 ซอฟต์แวร์ (Software)

Microsoft Voice

Microsoft Voice เป็นโปรแกรมสำเร็จรูปของ Microsoft speech SDK ที่ทำให้คอมพิวเตอร์สามารถเข้าใจคำสั่งเกี่ยวกับคำพูดได้อย่างง่ายดาย เช่น สั่งเปิด window ต่าง ๆ ของแต่ละโปรแกรม ที่ทำงานอยู่บนวินโดว์ โดยใช้เสียงพูดสั่งผ่านไมโครโฟน และเข้าไปในคอมพิวเตอร์ ตัว Microsoft Voice จะทำตามคำสั่งที่ผู้สั่งได้สั่งไว้

วิธีการใช้งาน Microsoft Voice มีขั้นตอนดังต่อไปนี้

1. ลงโปรแกรม Microsoft speech SDK จะเป็นการติดตั้งพร้อมด้วย Microsoft Voice
2. เปิดโปรแกรม Microsoft Voice จะมีสัญลักษณ์เป็นรูปตัว V บน Taskbar ของ Windows

รูปที่ 4.2 สัญลักษณ์ของ Microsoft Voice

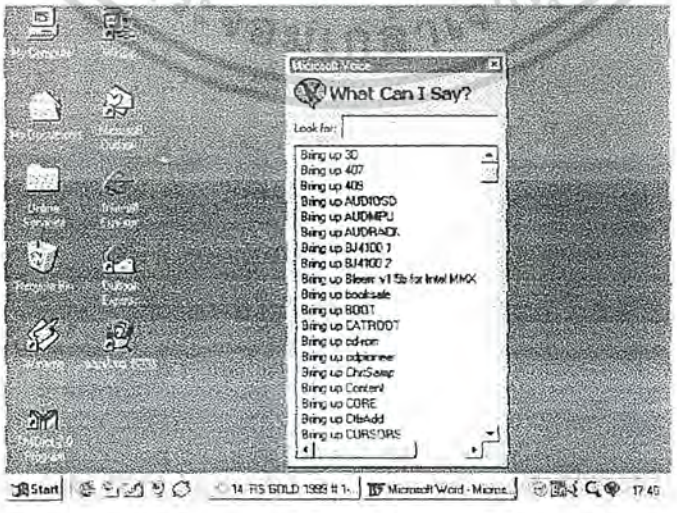
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิ๊ก Mouse ด้านซ้ายที่ Icon จะเป็นการปิดเปิดการรับฟังคำสั่งเสียง และถ้าคลิ๊ก Mouse ด้านขวาจะปรากฏเป็นวินโดว์ตามรูปข้างล่าง

รูปที่ 4.3 เมนูการทำงานของ Microsoft Voice

แต่ละเมนูสามารถอธิบายได้ดังนี้

- Listening for Voice Command เมื่อเลือกเมนูนี้เป็นการเปิดให้ Microsoft Voice รับฟังคำสั่งเสียง
- Not Listening เป็นการปิดไม่ให้รับฟังคำสั่งเสียง
- Paused Listening เป็นการหยุดรับฟังคำสั่งเสียงชั่วคราว
- What can I say เมื่อเลือกเมนูนี้ หรือใช้คำสั่งเสียง จะได้ผลดังนี้

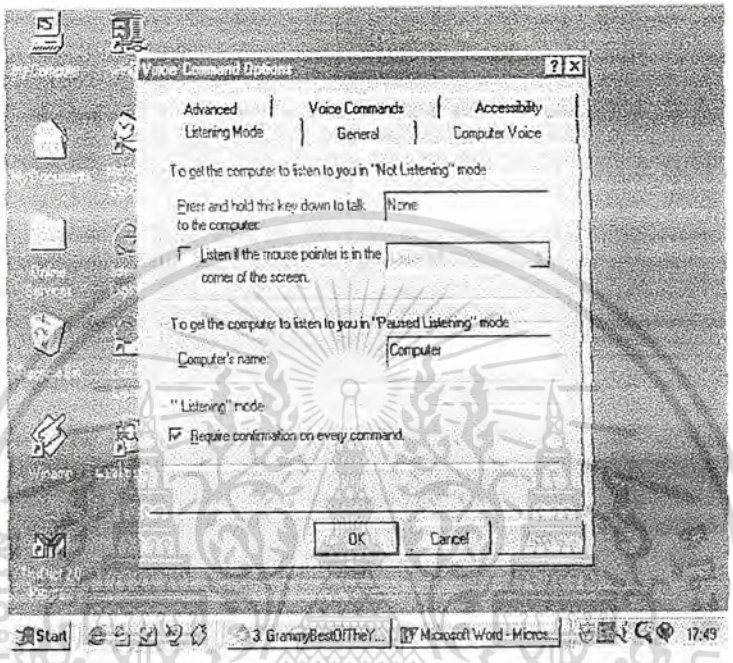


รูปที่ 4.4 การใช้เมนู What can I say

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะเป็นตารางแสดงให้รู้ว่าจะสามารถใช้คำสั่งเสียงอะไรได้บ้าง กับ Microsoft Voice

- Voice Command Options เป็นการเซ็ตอัพในส่วนต่าง ๆ ของ Microsoft Voice เมื่อเลือกเมนูนี้แล้วจะได้ผลดังรูปข้างล่างนี้



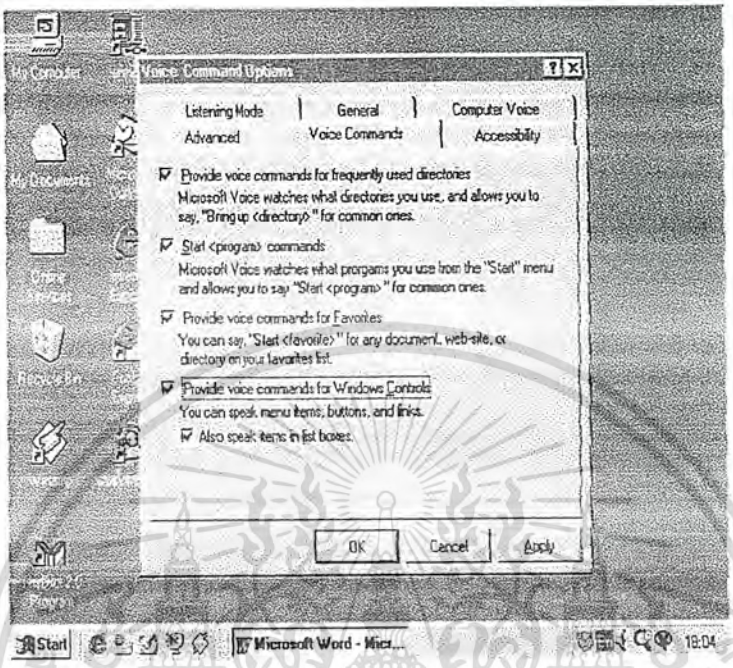
รูปที่ 4.5 การใช้เมนู Listening Modenใน Voice Command Options

ในเมนูนี้จะกล่าวถึงเพียงในส่วนที่สำคัญดังนี้

ส่วนสำคัญของ Listening Mode คือ ในส่วนของ "Listening" mode ถ้าเลือกจะเป็นการยืนยันคำสั่งเสียงอีกครั้งหนึ่ง โดยต้องพูดว่า "Do it" เป็นการยืนยัน ถ้าไม่เลือกจะไม่ต้องมีการยืนยันโปรแกรมจะทำงานทันทีเมื่อมีการสั่งเกิดขึ้น

ในส่วนของ Voice Commands เมนูข้างในจะถูกเลือกไว้เสร็จเรียบร้อยแล้วยกเว้น Provide voice commands for windows control เมื่อเลือกเมนูนี้ จะสามารถใช้คำสั่งเสียงควบคุมการทำงานพื้นฐานของวินโดว์ได้ และเมนูย่อย Also speak items in list boxes เป็นการเลือกเพื่อสามารถใช้คำสั่งเสียงกับเมนูข้างในโปรแกรมที่ได้เลือกไว้แล้วได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 เมนู Voice Commands ใน Voice Command Option

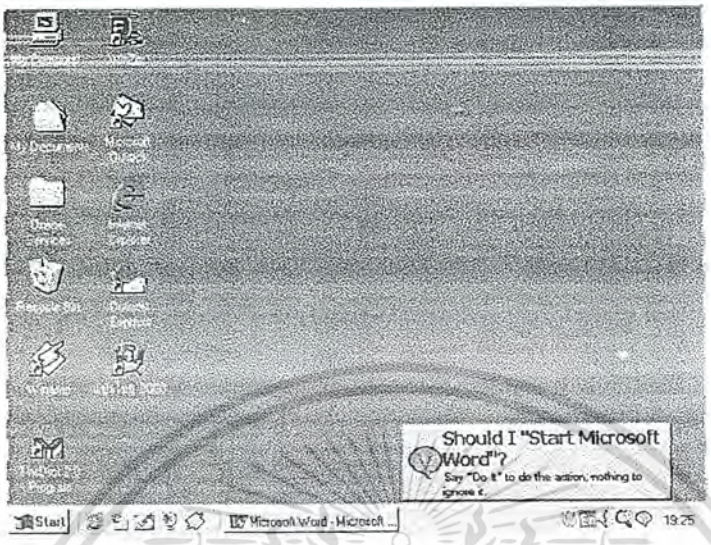
- Help เป็นส่วนอธิบายเกี่ยวกับการทำงานของโปรแกรม
- Exit Microsoft Voice เป็นการยกเลิกการใช้งานแล้วออกจากโปรแกรม

4. เมื่อเซ็ตค่าต่าง ๆ ตามที่ต้องการแล้วเริ่มการใช้งานโดยผู้สั่งงานสั่งผ่านไมโครโฟน เป็นเสียงพูด ผู้สั่งจะต้องพูดให้ตรงตามหลักเน้นเสียงให้ถูกต้อง และMicrosoft Voice นี้สามารถให้ใครก็ได้ใช้งานตามต้องการ

ตัวอย่างการ ใช้งานเมื่อต้องการเปิดโปรแกรม Microsoft word

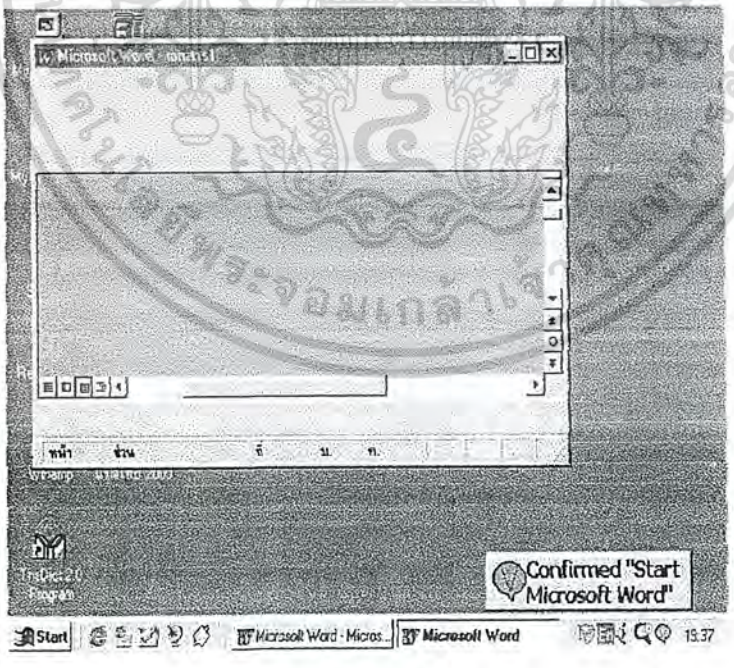
- เริ่มต้นโดยการสั่งด้วยเสียงว่า "start microsoft word" จะได้ผลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 การใช้คำสั่ง "start microsoft word"

- จากนั้นยืนยันคำสั่งโดยการพูดว่า "do it" จะได้ผลดังนี้



รูปที่ 4.8 ผลของการใช้คำสั่ง "start microsoft word"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Software Visual Basic

เป็นซอฟต์แวร์ที่เขียนขึ้นโดยโปรแกรม Visual Basic ซึ่งจะทำงานร่วมอยู่กับโปรแกรม Microsoft Voice โดยจะมีการทำงานเมื่อมีการพูดสั่งงานจากไมโครโฟน ซึ่งโปรแกรมจะรับสัญญาณเข้ามาทำการเปรียบเทียบกับคำสั่งที่ป้อนไว้แล้วในโปรแกรมว่าตรงกับคำสั่งใด เมื่อเปรียบเทียบเสร็จแล้วก็จะทำงานตามเงื่อนไขที่ต้องการ ถ้าคำสั่งใดไม่ตรงกับคำสั่งที่มีอยู่โปรแกรมก็จะไม่ส่งข้อมูลใดๆ ไปที่ฮาร์ดแวร์ซึ่งตัวฮาร์ดแวร์ก็จะไม่มีการเปลี่ยนแปลงไปจากเดิม จนกว่าซอฟต์แวร์จะส่งข้อมูลไปที่ฮาร์ดแวร์เท่านั้น

การทดสอบซอฟต์แวร์นั้นจะทำการทดสอบโดยการพูดตามคำสั่งที่กำหนดโดยโปรแกรมซึ่งจะเป็นใครก็ได้ที่จะเป็นผู้สั่งงานโดยไม่จำกัดเพศ อายุ เพียงแต่ สั่งงานให้ตรงตามคำสั่งที่ตั้งไว้เท่านั้น โดยคำสั่งที่ใช้ในการทดสอบจะมีด้วยกัน 5 คำสั่ง

ตารางที่ 4.1 ตารางการทดสอบการใช้คำสั่งบังคับรถ

คำสั่ง / ครั้งที่	Forward	Turnleft	Turnright	Backward	Stop
1	/	/	/	/	/
2	X	/	/	/	/
3	/	/	/	X	/
4	/	/	/	/	/
5	/	/	/	/	/
6	/	/	/	/	/
7	/	/	/	/	/
8	/	/	/	/	/
9	/	/	/	X	/
10	/	/	/	/	/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 เปอร์เซ็นต์ความถูกต้อง

คำสั่ง	Forward	Turnleft	Turnright	Backward	Stop
(%) ความถูกต้อง	90	100	100	80	100

การทดลองสามารถให้ผลออกมาเป็นที่น่าพอใจ แต่ถ้าเราให้คนอื่นมาลองทดลองบ้างปรากฏว่าผลออกมาใกล้เคียงกัน แต่จะแตกต่างกันบ้าง ขึ้นอยู่กับผู้พูดว่าจะออกเสียงได้ถูกต้องแค่ไหน และบางคำที่มีเสียงใกล้เคียงกันบางครั้ง ซอฟต์แวร์ก็ทำงานผิดคำสั่งบ้าง เพราะฉะนั้นการออกคำสั่งจะต้องให้พูดให้เหมือนภาษาอังกฤษให้มากที่จะได้ผลออกมีความถูกต้องสูง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปปัญหาและข้อเสนอแนะ

5.1 สรุป

การสั่งงานด้วยเสียงพูดโดยใช้ MICROSOFT SPEECH SDK 4.0 ของ VISUAL BASIC ในการจัดการเกี่ยวกับเรื่องเสียงนั้น จะมีความถูกต้องแม่นยำที่สูงขึ้นถ้าคำสั่งเสียงนั้นมีการออกเสียงที่ไม่คล้ายกันมากนัก แต่จะมีความผิดพลาดสูงในกรณีที่คำสั่งเสียงมีการออกเสียงคล้ายกันซึ่งสำหรับคนไทยนั้นการแก้ไขตรงจุดนี้คงต้องเรียนรู้วิธีพูดภาษาอังกฤษและการออกเสียงที่ถูกต้องจึงจะทำให้เกิดความแม่นยำที่สูงขึ้นได้

แต่จากการทดลองพูดสั่งงานนี้ได้ผลที่น่าพอใจ เพราะคำสั่งที่สั่งงานนั้น มีการออกเสียงที่ไม่คล้ายกันเท่าไรนัก ทางด้านฮาร์ดแวร์ก็สามารถแสดงผลได้ตามที่กำหนด คือ สั่งคำสั่งใดไปฮาร์ดแวร์นั้นๆ ก็จะทำงานตามที่สั่ง จะผิดพลาดก็ต่อเมื่อมีการสั่งงานในคำสั่งที่คล้ายกันเท่านั้นเพราะฮาร์ดแวร์นั้นทำงานตามข้อมูลที่ซอฟต์แวร์เป็นตัวส่งข้อมูลมาให้ฮาร์ดแวร์

5.2 ปัญหาที่เกิดขึ้นในการทำงาน

1. การทำโครงงานนี้ได้เกิดปัญหาใหญ่ขึ้น คือ คู่ โครงงาน มีปัญหาส่วนตัวทำให้ไม่ได้มาเรียน และรวมไปถึงการไม่ได้ร่วมทำโครงงานนี้ด้วย

2. การศึกษาข้อมูลเกี่ยวกับ MICROSOFT SPEECH SDK 4.0 นั้นทำได้ช้าเนื่องจากต้นฉบับเป็นภาษาและข้อมูลทางด้านอินเตอร์เน็ตนั้นก็มีของ MICROSOFT ที่เดียวเท่านั้น ส่วนทฤษฎีต่างๆ ที่ใช้ในการทำงานของ MICROSOFT SPEECH SDK 4.0 นั้นก็ไม่มีการเปิดเผย เพียงแต่มีแค่วิธีการใช้งานและ HELP เกี่ยวกับตัวคอนโทรลเท่านั้น

3. อุปกรณ์ฮาร์ดแวร์คือ IC UART CDP6402 นั้นหาซื้อได้ยากเพราะต้องสั่งจากทางต่างประเทศซึ่งทางร้านที่สั่งก็มีเพียง 2-32 ร้านเท่านั้น ถ้าของขาดตลาดก็จะซื้อไม่ได้เลย

4. ไมโครโฟนที่ใช้นั้นต้องมีความไวพอสมควรและไม่ก่อให้เกิดเสียงรบกวนมากนัก

5.3 แนวทางการแก้ไข

1. พยายามศึกษาและปฏิบัติด้วยด้วยตัวเอง เนื่องจากคู่ โครงงานประสบปัญหาส่วนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ศึกษาภาษาอังกฤษเพิ่มเติมเพื่อให้สามารถอ่านและแปรคำราภาษาอังกฤษได้ถูกต้องมากขึ้น ปรึกษาผู้ที่มีความรู้ด้าน VISUAL BASIC เพื่อการเขียนโปรแกรมที่ถูกต้อง

3. ควรใช้อุปกรณ์ที่สามารถหาซื้อได้ง่ายแทน เช่น ใช้ไมโครคอนโทรลเลอร์ เป็นต้น

4. ควรใช้ไมโครโฟนที่มีความไวพอสมควรและไม่มีเสียงรบกวนมากนัก ซึ่งราคาของไมโครโฟนก็จะแพงตามไปด้วย

5.4 ข้อเสนอแนะในโครงงาน

ควรจะสามารถสั่งงานด้วยภาษาไทยได้ในอนาคต โดยศึกษา MICROSOFT SPEECH SDK 4.0 และ VISUAL BASIC ให้เข้าใจมากยิ่งขึ้น เพราะเป็นหัวใจในการทำงานของโครงการนี้เพื่อทำให้สามารถพัฒนาโปรแกรมให้ทำงานได้วัตถุประสงค์ที่ต้องการมากยิ่งขึ้น และสามารถนำไปใช้งานในการประกอบอาชีพได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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/

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

Source Code ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOURCE CODE ของ FORM1

```
Public mainmenu As Long
Private buffdata As String

Private Sub UpdateList()
Dim i As Long
Dim counter As Long
Dim command As String
Dim description As String
Dim category As String
Dim Flags As Long
Dim Action As String
List1.Clear
counter = Vcommand1.CountCommands(mainmenu)
For i = 1 To counter
  Vcommand1.GetCommand mainmenu, i, command, description, category, Flags, Action
  List1.AddItem command
Next i
End Sub

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Vcommand1.AddCommand mainmenu, 1, "Toiy Char", "when you say" + "Toiy Char",
"listen list", 0, ""
UpdateList
Vcommand1.Activate mainmenu
End Sub

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

Private Sub Mnuเกี่ยวกับโครงงานItem_Click()
Form3.Show
End Sub

Private Sub mnuรอฟังคำสั่งItem_Click()
  Vcommand1.Activate mainmenu
  cmdListen.Caption = "หยุดฟังคำสั่ง"
  Hear.Caption = "รับคำสั่ง"
  List1.Enabled = True
End Sub

Private Sub Mnuระบบช่วยเหลือ_Click()
    Form2.Show
End Sub

Private Sub Mnuหยุดฟังคำสั่งItem_Click()
  Vcommand1.Deactivate mainmenu
  cmdListen.Caption = "รอฟังคำสั่ง"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Vcommand1.Deactivate mainmenu
    cmdListen.Caption = "รอฟังคำสั่ง"
    Hear.Caption = "หยุดรับคำสั่ง"
    List1.Enabled = False
End If
End Sub

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

Private Sub Form_Load()
buffdata = "0"
    MSComm2.CommPort = 2
    MSComm2.Settings = "9600,N,8,1"
    MSComm2.PortOpen = True
    MSComm2.Output = Chr$(0)
    MSComm2.PortOpen = False
    Vcommand1.Initialized = 1
mainmenu = Vcommand1.MenuCreate(App.EXEName, "state1", 4)
Vcommand1.Enabled = True
Vcommand1.AddCommand mainmenu, 1, "Run", "when you say" + "Run", "listen list", 0,
""
Vcommand1.AddCommand mainmenu, 1, "Turn Right", "when you say" + "Turn Right",
"listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Turn Lefts", "when you say" + "Turn Lefts",
"listen list", 0, ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Vcommand1.AddCommand mainmenu, 1, "Back ward", "when you say" + "Back ward",
"listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Stop", "when you say" + "Stop", "listen list", 0,
""
Vcommand1.AddCommand mainmenu, 1, "Turn around Lefts", "when you say" + "Turn
around Lefts", "listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Slow", "when you say" + "Slow", "listen list", 0,
""
Vcommand1.AddCommand mainmenu, 1, "Turn around Right", "when you say" + "Turn
around Right", "listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Back ward Slow", "when you say" + "Back ward
Slow", "listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Real Sine", "when you say" + "Real Sine", "listen
list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Wing", "when you say" + "Wing", "listen list", 0,
""
Vcommand1.AddCommand mainmenu, 1, "Real Phar", "when you say" + "Real Phar",
"listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Yoot", "when you say" + "Yoot", "listen list", 0,
""
Vcommand1.AddCommand mainmenu, 1, "Toiy", "when you say" + "Toiy", "listen list", 0,
""
Vcommand1.AddCommand mainmenu, 1, "Wing Char", "when you say" + "Wing Char",
"listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Moon Sine", "when you say" + "Moon Sine",
"listen list", 0, ""
Vcommand1.AddCommand mainmenu, 1, "Moon Phar", "when you say" + "Moon Phar",
"listen list", 0, ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Hear.Caption = "หยุดรับคำสั่ง"
    List1.Enabled = False
End Sub


Private Sub MnuออกจากโปรแกรมItem_Click()
End
End Sub

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)
Hear.Caption = command
    If Hear.Caption = "Run" Then
        buffdata = "1"
        buffdata = Val(buffdata)
    ElseIf Hear.Caption = "Turn Right" Then
        buffdata = "2"
        buffdata = Val(buffdata)
    ElseIf Hear.Caption = "Turn Lefts" Then
        buffdata = "4"
        buffdata = Val(buffdata)
    ElseIf Hear.Caption = "Back ward" Then
        buffdata = "8"
        buffdata = Val(buffdata)
    ElseIf Hear.Caption = "Stop" Then
        buffdata = "16"
        buffdata = Val(buffdata)
    ElseIf Hear.Caption = "Slow" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    buffdata = "32"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Turn around Right" Then
    buffdata = "64"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Turn around Lefts" Then
    buffdata = "128"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Back ward Slow" Then
    buffdata = "3"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Wing" Then
        Hear.Caption = "วิ่ง"
    buffdata = "1"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Real Phar" Then
        Hear.Caption = "เลี้ยวขวา"
    buffdata = "2"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Real Sine" Then
        Hear.Caption = "เลี้ยวซ้าย"
    buffdata = "4"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Yoot" Then
        Hear.Caption = "หยุด"
    buffdata = "16"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Toiy" Then
        Hear.Caption = "ถอย"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    buffdata = "8"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Wing Char" Then
        Hear.Caption = "วิ่งช้า"
    buffdata = "32"
    buffdata = Val(buffdata)
   ElseIf Hear.Caption = "Moon Phar" Then
        Hear.Caption = "หมุนขวา"
    buffdata = "64"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Moon Sine" Then
        Hear.Caption = "หมุนซ้าย"
    buffdata = "128"
    buffdata = Val(buffdata)
  ElseIf Hear.Caption = "Toiy Char" Then
        Hear.Caption = "ถอยช้า"
    buffdata = "3"
    buffdata = Val(buffdata)
  Else
    MsgBox "คำสั่งไม่ถูกต้อง...!!!", vbOKOnly + vbCritical, "คำสั่งผิดพลาด"
  End If
  MSComm2.CommPort = 2
  MSComm2.Settings = "9600,N,8,1"
  MSComm2.PortOpen = True
  MSComm2.Output = Chr$(buffdata)
  MSComm2.PortOpen = False
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

Source Code ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
คอนโทรลตัวอื่นๆ ที่อยู่ใน MICROSOFT SPEECH SDK 4.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Direct Speech Recognition Control

Properties

Direct Speech Recognition control มีคุณสมบัติดังต่อไปนี้

AutoGain

Complete TimeOut

CountEngines

CreateResultsObject

Dialect

Echo

EnergyFloor

EngineFeatures

EngineID

Features

FileName

Find

FindEngine

FlagsGet

GetPhraseScore

Grammars

HWnd

Identify

IncompleteTimeOut

Initialized

Interfaces

LanguageID

LastError

LastHeard MaxAotuGain

MaxCompleteTimeOut

MaxEnergyFloor

MaxIncompleteTimeOut

MaxRealTime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MaxThreshold

MaxWordsState

MaxWordsVocab

MfgName

Microphone

MinAutoGain

MincompleteTimeOut

MinEnergyFloor

MinIncompleteTiomeOut

MinRealTime

MinThreshold

ModelD

ModeName

Phrase

ProductName

RealTime

ReEvaluate

Sequencing

Speaker

SRMode

SuppressExceptions

Threshold

Wave

Methods

Direct Speech Recognition control มี methods ดังต่อไปนี้

AboiutDlg

Activate

ActivateAndAssignWindow

Archive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Correction

Deaction

DeleteArchive

DestroyResultsObject

GeneralDlg

GrammarDataset

GrammarFormMemory

GrammarFormResource

GrammarFormStream

GrammarFormString

GrammarToMemory

InitAudioSourceDirect

InitAudioSourceObject

LexiconDlg

Listen

Pause

PosnGet

Resume

Select

SelectEngine

TimeGet

TrainGeneralDlg

TrainMicDlg

Validatr

Events

Direct Speech Recognition control มี events ดังต่อไปนี้

AttribChanged BookMark

ClickIn

InterFerence

Paused

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PhraseFinish

PhraseHypothesis

PhraseStart

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LanguageID

LastError

LastWordPosition

LipTension

LipType

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SuppressException

Tagged

TeethLowerVisible

TeerthUpperVisible

TonguePosn

VolumeLeft

VolumeRight

Methode

Direct Speech Synthesis Control มี methods ดังต่อไปนี้

AboutDlg

AudioPause

AudioReset

AudioResume

GeneralDlg

DetPronunciation

InitAudioDestDriect

InitAudioDestMM

InitAudioDestObject

InJect

LexiconDlg

PosnGet

Select

Speak

TextData

TranslateDlg

Events

Direct Speech Synthesis Control มี events ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AttribChanged

AudioStart

AudioStop

BookMark

ClickIn

TextDataDone

TextDataStarted

Visual

WordPosition

Voice Dictation Control

Properties

Voice Dictation control มี properties ดังต่อไปนี้

Attributes

AttributeMemory

AttributeString

AutoGainEnble

BookmarkQuery

CoutBookmarks

CountCommands

CountGlossary

CountSpeakers

CreateDocFile

CreateStream

Echo

EnergyFloor

Flags

HWnd

Initialized

IsAnyoneDictating

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LastError

Microphone

Mode

Option

OptionsEnum

Realtime

Speaker

SpeakerGet

SuppressExceptions

Threshold

TimeoutComplete

TimeoutIncomplete

WLeft

WTop

Methods

Voice Dictation control มี methods ดังต่อไปนี้

AboutDlg

Activate

ActivateAndAssignWindow

AddCommand

AddGlossary

BookmarkAdd

BookmarkRemove

CFGSet

CopyToBin

CopyToMemory

Deactivate

FX

GeneralDlg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GetBookMark

GetChanges

GetCommand

GetGlossary

GlobalkeyHook

Hint

ITNApply

ITNExpand

LexiconDlg

Lock

MemoryGet

MemorySet

PasteFromBin

PasteFromMemory

ReadStreamFont

ReleaseStore

ReleaseStream

RemoveCommand

RemoveGlossary

Resultsget

ResultsGet2

ResultsSet

SessionDeserialize

SessionSerialize

SetCommand

SetGlossary

SetSelRect

SetSize

SpeakerDelete

SpeakerNew

SpeakerQuery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SpeakerRevert

SpeakerSelect

StreamRead

StreamWrite

TextGet

TextRemove

TextSelGet

TextSelSet

TextSet

TrainGeneralDlg

TrainMicDlg

Unlock

Words

WriteSteamFont

Events

Voice Dictation control มี events ดังต่อไปนี้

AttribChanged

ClickIn

CommandBuiltIn

Commamd Oter

CommandRecognize

Dictaing

GlobaKey

GlobalMouse

Interference

PhraseFinish

PhraseHypothesis

PhraseStart

TextBookmarkChanged

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TextChanged

TextSelChanged

Training

UtteranceBegin

UtteranceEnd

VUMeter

Voice Text Control

Properties

Voice Text contro มี Properties ดังต่อไปนี้

Age

CountEngines

CurrentMode

Device

Dialect

Enable

EngineFeatures

Features

Find

FindEngine

Gender

HWnd

Initialized

Interfaces

IsSpeaking

JawOpen

LanguageID

LastError

เอกสารนี้เป็นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LipTension

LipType

MfgName

ModeID

ModeName

MouthHeight

MouthUpturn

MouthWidth

ProductName

Speaker

Speed

Style

SuppressExceptions

TeethLowerVisible

TeethUpperVisible

TonguePosn

TTSMode

Methods

Voice Text control มี methods ดังต่อไปนี้

AboutDlg

FastForward

GeneralDlg

LexiconDlg

Pause

Resume

Rewind

Select

Speak

StopSpeaking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Methods

Voice Telephony control มี methods ดังต่อไปนี้

CallDialog

ChooseEngineDialog

DateFromFile

DateFromResource

DateFromString

DWORDGet

DWORDSet

ExtensionFromFile

ExtensionFromResource

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ExtensionFromString

FreeWave

GrammarFromFile

GrammarFromResource

GrammarFromString

PhoneFromFile

PhoneFromResource

PhoneFromString

RecordFromFile

RecordFromResource

RecordFromString

SendAbort

SendDTMF

SendDTMFToLine

Speak

TimeFromFile

TimeFromeResource

TimeFromString

WaveAddFromListFile

WaveAddFromListResource

WaveAddFromListString

WaveAddFromMemory

YesNoFromFile

YesNoFromResource

YesNoFromString

Events

Voice Telephony control มี events ดังต่อไปนี้

ClickIn

DoPhoneCall

Initalize

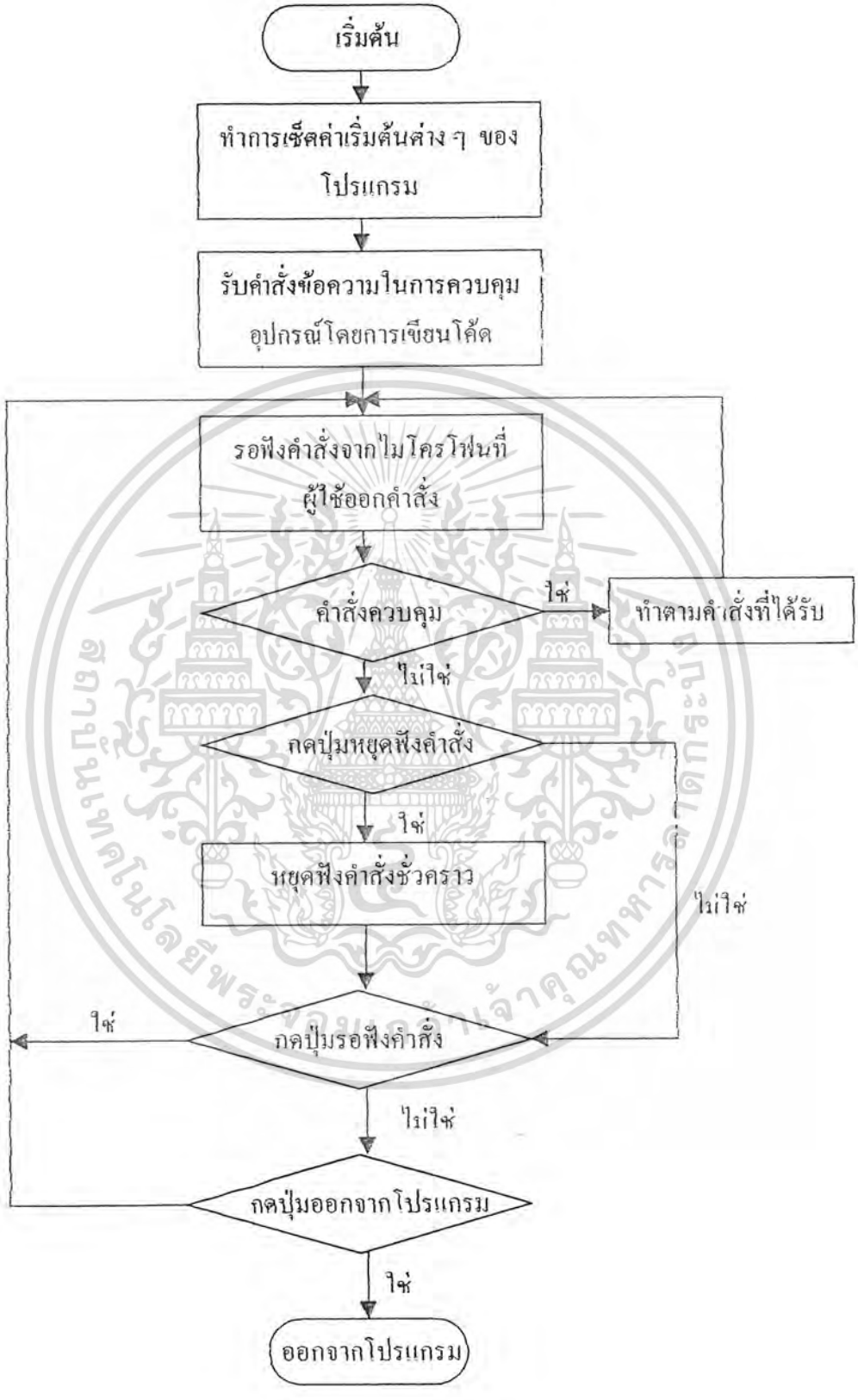
Shutdown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

FLOW CHAT แสดงการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

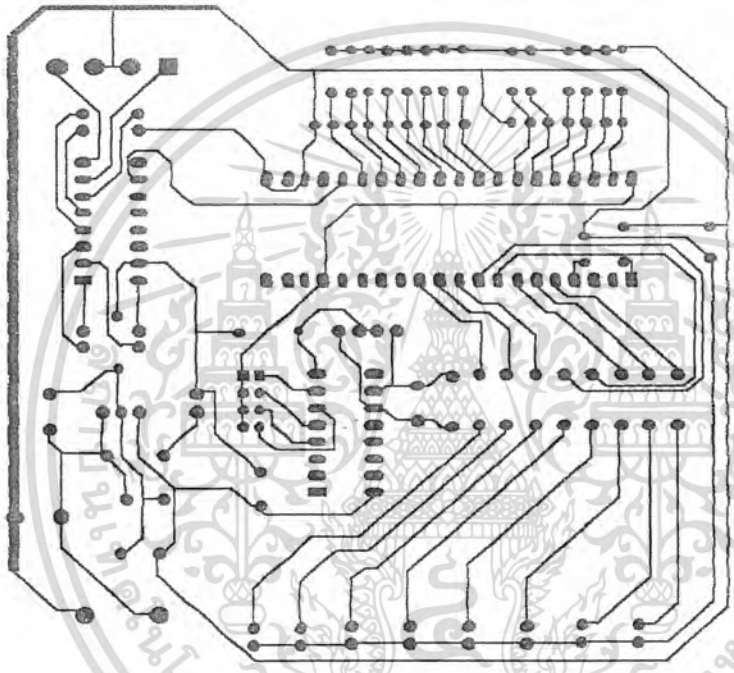


รูปที่ 1 ภาคผนวก ค FLOW CHAT แสดงการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
ลายปริ้นของจงจรอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ลายปริ้นของวงจรอิเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

DATACHEET ของ IC UART CDP6402

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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
  - ÷16, ÷1, ÷0.5 Operation
- Automatic Data Formatting and Status Generation
- Word Length 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
- Pin for Pin Industry Type IM6402 and Compatible with AY-5-1402

Ordering Information

TEMP. RANGE	5V/200K BAUD	10V/400K BAUD	PKG. NO.
-40°C to +85°C	CDP6402CE	CDP6402E	E40.6
	CDP6402CEX	-	
-40°C to +85°C	CDP6402CD	CDP6402D	D40.6
	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.

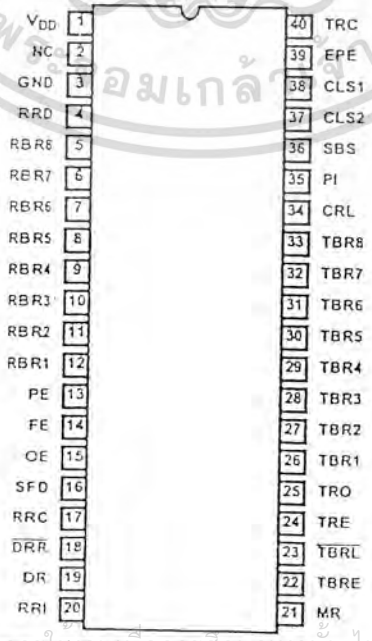
The data word can be 5, 6, 7 or 8 bits in length. Parity may be odd, even or inhibited. Stop bits can be 1, 1-1/2, or 2 (when transmitting 5-bit code).

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

File Number 1328.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

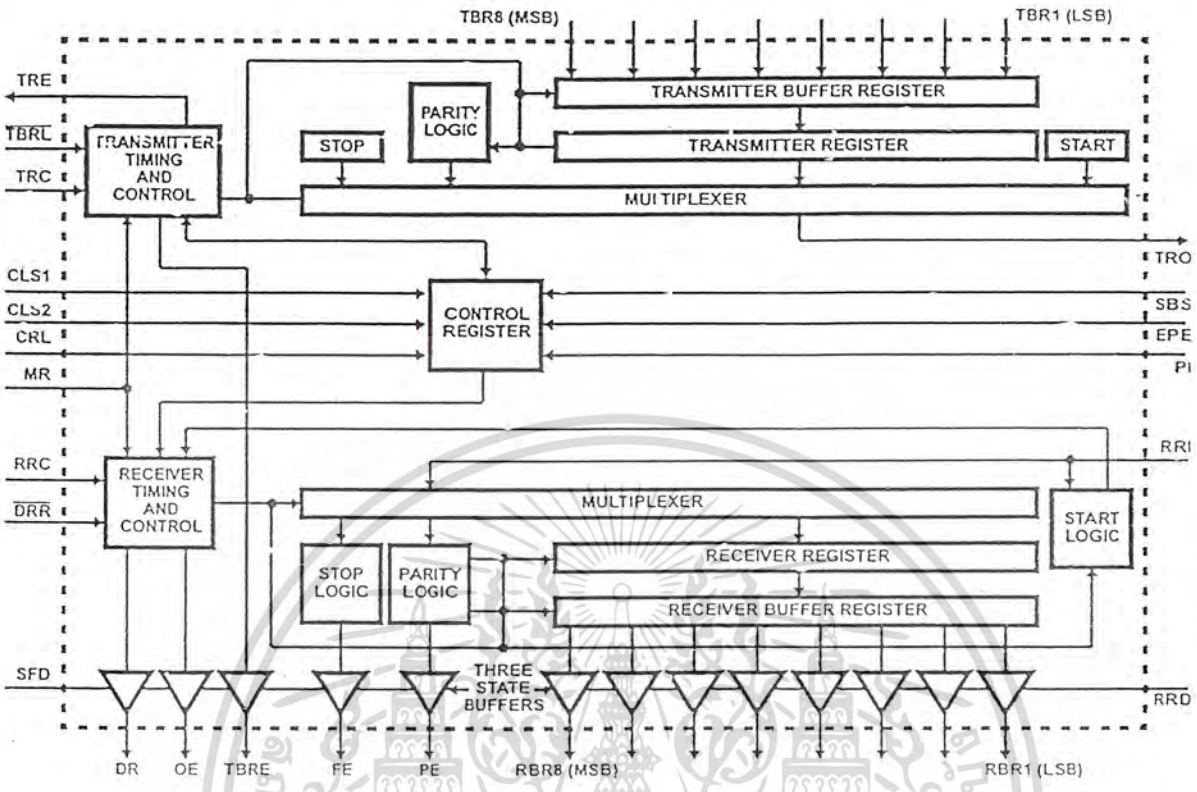


FIGURE 1. FUNCTIONAL BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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	
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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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.

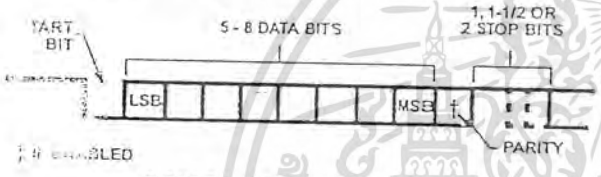


FIGURE 2. SERIAL DATA FORMAT

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.

Serial 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.

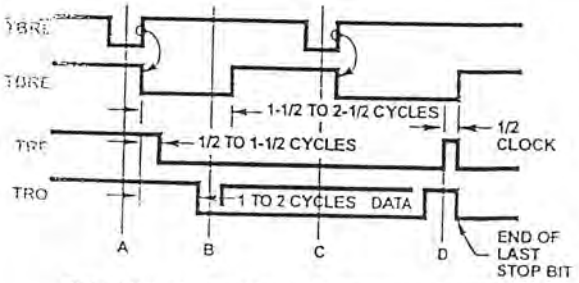


FIGURE 3. TRANSMITTER TIMING WAVEFORMS

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.

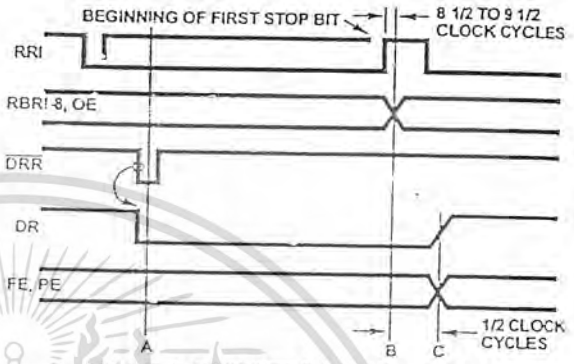


FIGURE 4. RECEIVER TIMING WAVEFORMS

(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.

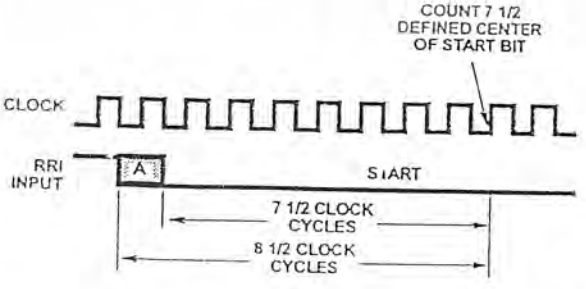


FIGURE 5. START BIT TIMING WAVEFORMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 2. FUNCTION PIN DEFINITION

PIN	SYMBOL	DESCRIPTION
7	V_{DD}	Positive Power Supply
4	N/C	No Connection
	GND	Ground (V_{SS})
	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 tha. 8 characters are right justified to RBR1.
	RBR7	See Pin 5 - RBR8
	RBR6	
	RBR5	
2	RBR4	
	RBR3	
11	RBR2	
	RBR1	
17	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.
16	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).
18	SFD	A high level on STATUS FLAGS DISABLE forces the outputs PE, FE, OE, DR, TBRE to a high impedance state.
19	RRC	The RECEIVER REGISTER CLOCK is 16X the receiver data rate.
20	$\overline{DRR}$	A low level on DATA RECEIVED RESET clears the data received output (DR), to a low level.
13	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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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	CLS1†	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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dynamic Electrical Specifications at $T_A = -40^oC$ to $+85^oC$, $V_{DD} \pm 5\%$, 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 CDP6402 (NOTE 2) TYP	LIMITS CDP6402 (NOTE 3) MAX	LIMITS CDP6402C (NOTE 2) TYP	LIMITS CDP6402C (NOTE 3) MAX	UNITS
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 $T_A = 25^oC$ and nominal V_{DD}.
3. Maximum limits of minimum characteristics are the values above which all devices function.

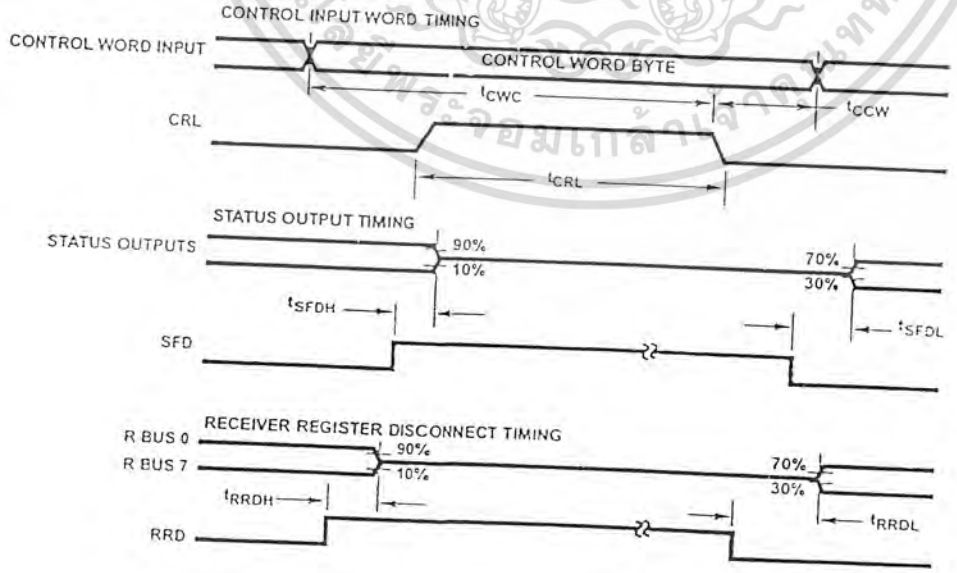


FIGURE 6. SYSTEM TIMING WAVEFORMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

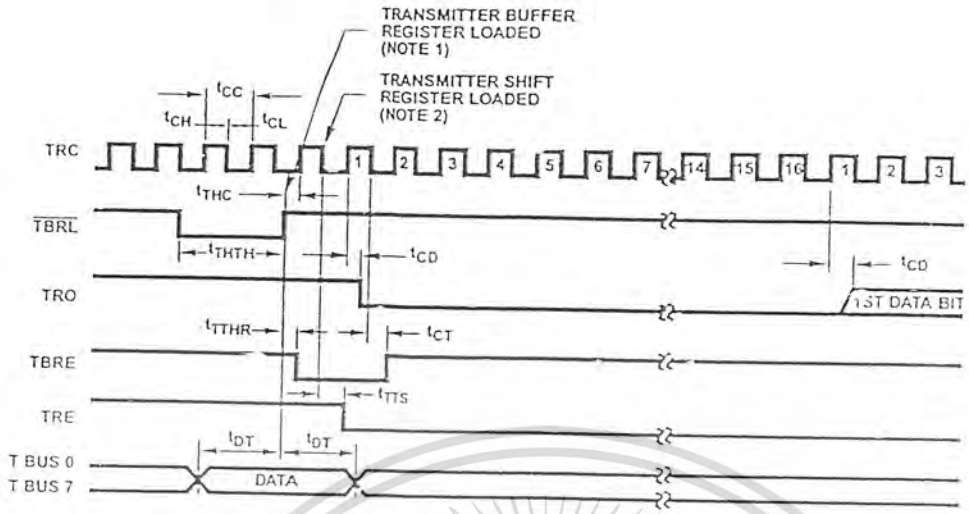
Dynamic Electrical Specifications at T_A = -40°C to +85°C, V_{DD} ±5%, t_R, t_F = 20ns, V_{IH} = 0.7 V_{DD}, V_{IL} = 0.3 V_{DD}, C_L = 100pF

(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.

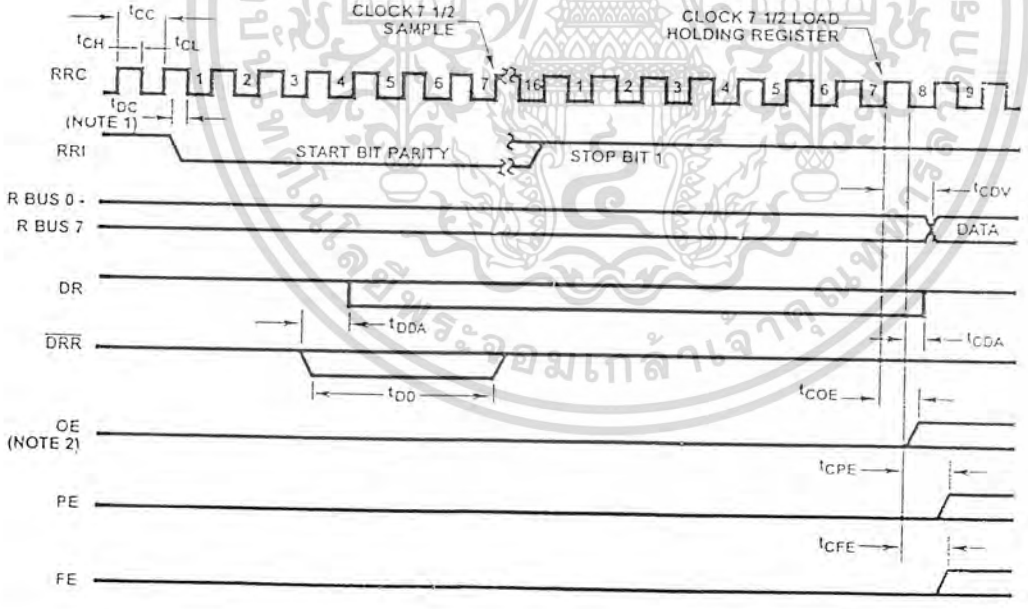
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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
$\overline{\text{Data Received Reset}}$	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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้