

ระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติ  
Grade Checking Automation System by telephone



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2545

เลขหมู่.....

เลขทะเบียน..... 49974

วัน,เดือน,ปี 16 เม.ย. 2547

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

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นตามการค้า  
แม้ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16/04/2004

ปริญญาโท ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติ

Grade Checking Automation System by Telephone

คณะผู้จัดทำ นายชนเทพ เมฆทรัพย์ รหัส 42010070

นายทนงศักดิ์ นิเวชพงศ์ศักดิ์ รหัส 42010123



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติ

นายชนเทพ เมฆทรัพย์ รหัส 42010070

นายทงศักดิ์ นิเวชพงศ์ศักดิ์ รหัส 42010123

อาจารย์ที่ปรึกษา

อ.ประสาร ดังติสานนท์

อ.อวัชริน นาชิน

ปีการศึกษา 2545

### บทคัดย่อ

โครงการนี้เป็นระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติ โดยผู้ใช้จะใช้โทรศัพท์ชนิดปุ่มกดเป็นอุปกรณ์ในการติดต่อกับระบบ และทางสถานศึกษาจะมีเครื่องคอมพิวเตอร์ซึ่งมีโมเด็ม เป็นอุปกรณ์ในการติดต่อกับผู้ใช้ ในโครงการนี้แบ่งการทำงานได้เป็น 4 ส่วน ได้แก่ การควบคุม Modem โดยใช้ TAPI ( Telephony Application Programming Interface ) , การใช้งาน Text to Speech เพื่อใช้ในการแปลงข้อมูลที่เกี่ยวข้องอยู่ในฐานข้อมูลให้เป็นข้อมูลในรูปแบบเสียงเพื่อใช้ตอบกลับไปซึ่งเป็นของ ไมโครซอฟ , การติดต่อกับฐานข้อมูลเพื่อเพิ่มประสิทธิภาพในการทำงานและค้นหา , การใช้งาน Multimedia API ในการทำงานกับข้อมูลในรูปแบบเสียงให้ออกไปทางโมเด็ม ซึ่งเครื่องมือที่ใช้ในการเขียนโปรแกรมคือ Visual C++ และในโครงการนี้สามารถรองรับการติดต่อได้ 2 คู่สายพร้อมกัน

## Grade Checking Automation System by telephone

Mr. CHANATHEP      MAKSUP      42010070

Mr. THANONGSAK      NIVETPONGSAK      42010123

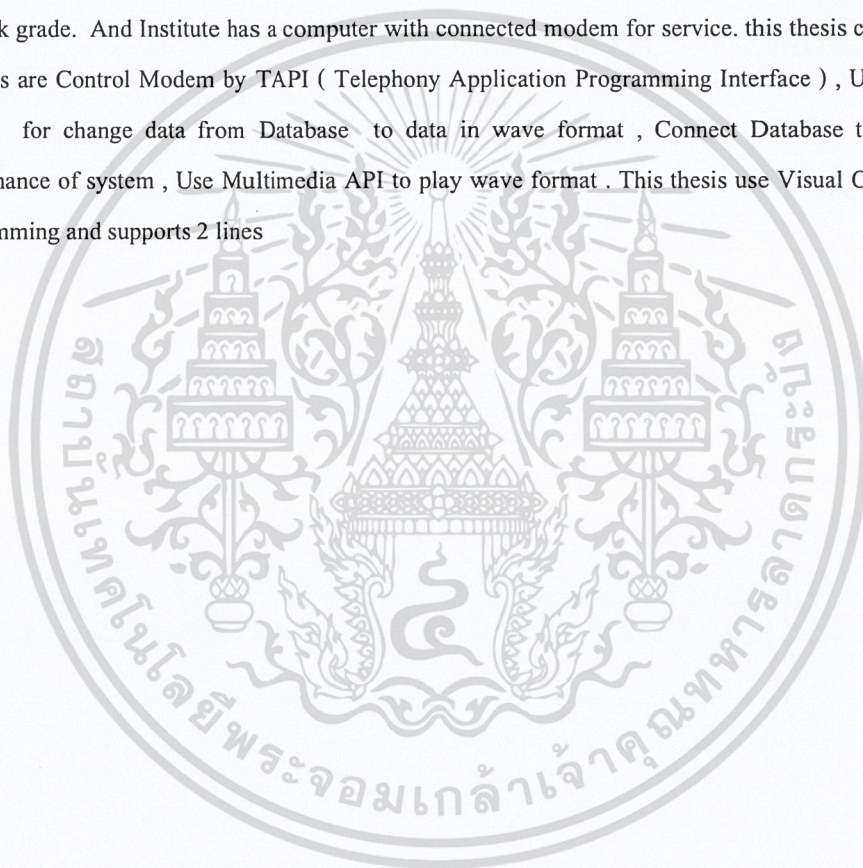
ADVISOR

Mr. PRASARN      TANGTISANON

Mr. AWACHARIN      NACHIN

### Abstraction

This thesis is Grade Checking Automation System by telephone. User can use a digit telephone to check grade. And Institute has a computer with connected modem for service. this thesis consist of 4 modules are Control Modem by TAPI ( Telephony Application Programming Interface ) , Use Text to Speech for change data from Database to data in wave format , Connect Database to increase performance of system , Use Multimedia API to play wave format . This thesis use Visual C++ tool in programming and supports 2 lines

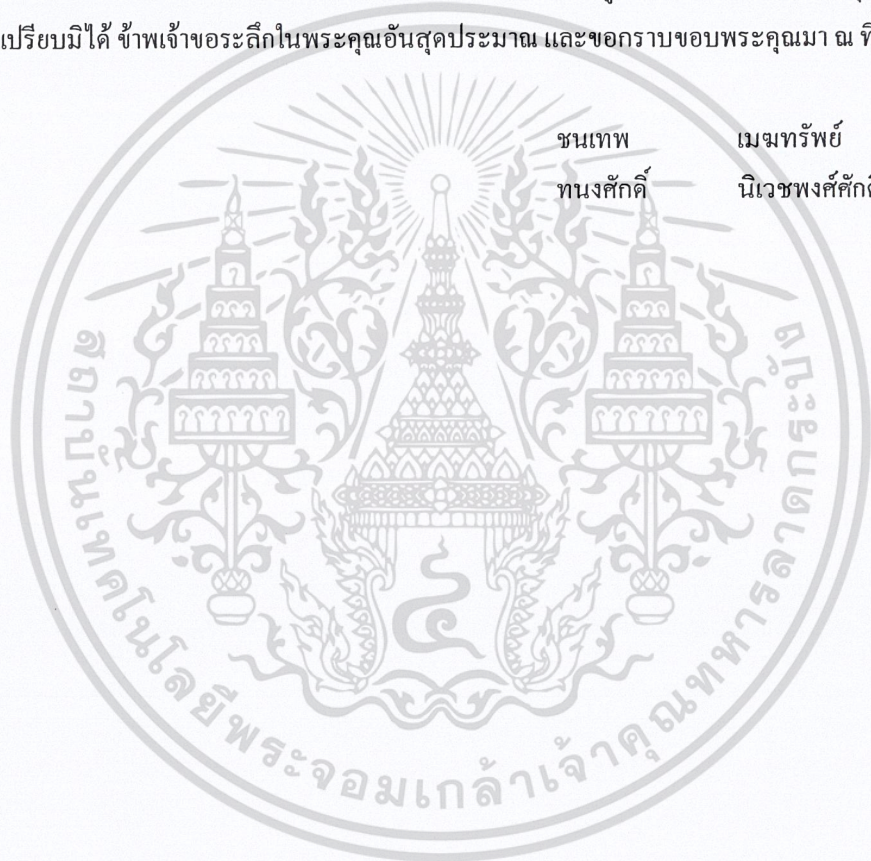


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญาบัตรนี้ เสร็จลงได้ก็คือ อาจารย์ อาจารย์ ประสาร ตังติสานนท์ และ อาจารย์ อวิชรีน นาซินอาจารย์ที่ปรึกษาปริญญาบัตร ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา นอกจากนั้นต้องขอขอบคุณเพื่อนๆ ทุกคนที่ให้คำแนะนำ เป็นเพื่อนคุย เพื่อนเล่น ระหว่างการทำงาน

และสุดท้ายที่ขาดไม่ได้คือ ครอบครัวของข้าพเจ้า อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ ดูแลเอาใจใส่เสมอมา ในทุกๆด้าน อันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	V
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 วิธีการดำเนินงาน	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 ทฤษฎีเกี่ยวกับโทรศัพท์	3
2.2 โมเด็ม	4
2.3 วิธีการเขียนโปรแกรมติดต่อกับโมเด็ม	5
2.4 โปรแกรมติดต่อกับฐานข้อมูล	9
2.5 วิธีการใช้งาน Text to Speech	9
2.6 การเขียนโปรแกรมโดยใช้ Thread	10
2.7 การเขียนโปรแกรมเพื่อใช้งานเกี่ยวกับการเล่น file wave	11
บทที่ 3 การใช้งาน TAPI 2.2	12
3.1 ข้อกำหนดการใช้งาน TAPI 2.2	12
3.2 ขั้นตอนการใช้งาน TAPI 2.2	12
3.3 ตัวอย่างการตรวจจับ Message ของ TAPI	15
3.4 การรับค่า Wave Device ID ของโมเด็ม	20
บทที่ 4 การใช้งาน Multimedia API	22
4.1 ข้อกำหนดการใช้งาน	22
4.2 การเตรียมข้อมูล	22
4.3 การเล่นเสียงออกโมเด็มและการจบการเล่น	24
บทที่ 5 การใช้งานโปรแกรมแปลงข้อความเป็นสัญญาณเสียง	26
5.1 ติดตั้งเพื่อให้สามารถใช้งานส่วนโปรแกรมแปลงข้อความเป็นสัญญาณเสียง	26
5.2 การทำงานภายในของตัวจัดการเกี่ยวกับสัญญาณเสียง	26
5.3 ตัวอย่างการเขียนโปรแกรมเรียกใช้ Text to Speech	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ ( ต่อ )

	หน้า
บทที่ 6 การทำงานและการออกแบบของระบบ	30
6.1 ภาพรวมของระบบ	30
6.2 ขอบเขตการใช้งาน	31
6.3 ความสามารถพื้นฐานของระบบ	32
6.4 สิ่งที่ระบบต้องการ	32
6.5 การทำงานของระบบ	33
6.6 ส่วนประกอบของระบบ	35
บทที่ 7 ผลการทดสอบ	39
7.1 ผลการทดสอบการใช้งาน TAPI	39
7.2 ผลการทดสอบการใช้งาน Text to Speech	39
7.3 ผลการทดสอบการเชื่อมต่อฐานข้อมูล	40
7.4 ผลการทดสอบใช้ Multimedia API	40
บทที่ 8 บทวิจารณ์และสรุป	41
8.1 บทวิจารณ์	41
8.2 สรุป	41
8.3 ข้อเสนอแนะสำหรับการพัฒนาในอนาคต	41
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน IV เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
6.1 ตารางแสดงโครงสร้างของ ตารางข้อมูล Student ในฐานข้อมูล	36
6.2 ตารางแสดงโครงสร้างของ ตารางข้อมูล Subject ในฐานข้อมูล	36
6.3 ตารางแสดงโครงสร้างของ ตารางข้อมูล Grade ในฐานข้อมูล	36
6.4 ตารางแสดงค่าที่เป็นไปได้ของผลการเรียนในแต่ละรายวิชา	37





## สารบัญญภาพ

รูปที่	หน้า
2.1 โทศัพท์แบบหมุน	3
2.2 การ Cross ความถี่	3
2.3 การทำงานของ TAPI	6
2.4 โครงสร้างการทำงานของ TAPI	8
2.5 โครงสร้างการทำงานของ Text to Speech	10
2.6 การเขียนโปรแกรมโดยใช้ Thread	11
5.1 โครงสร้างการทำงานของตัวจัดการเกี่ยวกับสัญญาณเสียง	26
6.1 ภาพรวมของระบบ	30
6.2 การทำงานของระบบ	33
6.3 ส่วนประกอบของระบบ	35



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเป็นที่ทราบกันว่าเป็นยุคแห่งข้อมูลข่าวสารและบริการ หลากขององค์กรต่างตระหนักถึงความสำคัญของข้อมูลข่าวสารและบริการ เพื่อประโยชน์หรือเพื่อแสดงศักยภาพการบริหารและจัดเก็บข้อมูลขององค์กรนั้นๆ เช่น เมื่อทำการจัดเก็บข้อมูล ต่างๆเราสามารถสร้างรายงานสรุปข้อมูล เพื่อเป็นประโยชน์ในการทำงานได้ หรืออาจจัดทำบริการค้นหาหรือสอบถามข้อมูลที่มีอยู่ได้ อาจค้นหาโดยใช้เครื่องคอมพิวเตอร์ผ่านระบบNetwork ค้นหาผ่าน Internet หากองค์กรใดมีระบบจัดการในการค้นหาหรือสอบถามข้อมูลที่ดีและมีประสิทธิภาพจะทำให้องค์กรนั้นๆ ใช้ประโยชน์จากข้อมูลที่มีได้อย่างมีประสิทธิภาพ

ดังนั้น ระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติจึงได้ถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกในการสอบถามผลการศึกษาได้อย่างถูกต้องและมีประสิทธิภาพ ซึ่งอุปกรณ์ที่ใช้ในนี้ได้แก่ โทรศัพท์ ซึ่งเป็นอุปกรณ์ที่หาได้ง่าย และมีอยู่ทั่วไป ทำให้ผู้ใช้ลดข้อจำกัดทางด้านอุปกรณ์การสืบค้นได้ โดยการเขียนโปรแกรมนั้นใช้ Visual C++ ในการเขียนโปรแกรม มีการใช้ TAPI ในการติดต่อกับ โมเด็ม และมีการนำเทคโนโลยี Text to Speech มาใช้ในการจัดเตรียมข้อมูลในรูปแบบเสียงเพื่อให้ง่ายในการนำไปใช้งาน

### 1.2 วัตถุประสงค์

ระบบที่พัฒนาขึ้นนี้มีจุดประสงค์เพื่อ

- เพื่อสร้างระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติ
- เพื่อช่วยบริการข้อมูลข่าวสารอย่าง สะดวก รวดเร็ว
- เพื่ออำนวยความสะดวกแก่ผู้ที่อยู่ไกล
- เพื่อเพิ่มประสิทธิภาพในการให้บริการข้อมูลข่าวสาร
- เพื่อเพิ่มวิธีในการรับข้อมูลข่าวสารของผู้ใช้บริการ

### 1.3 ขอบเขตของโครงการ

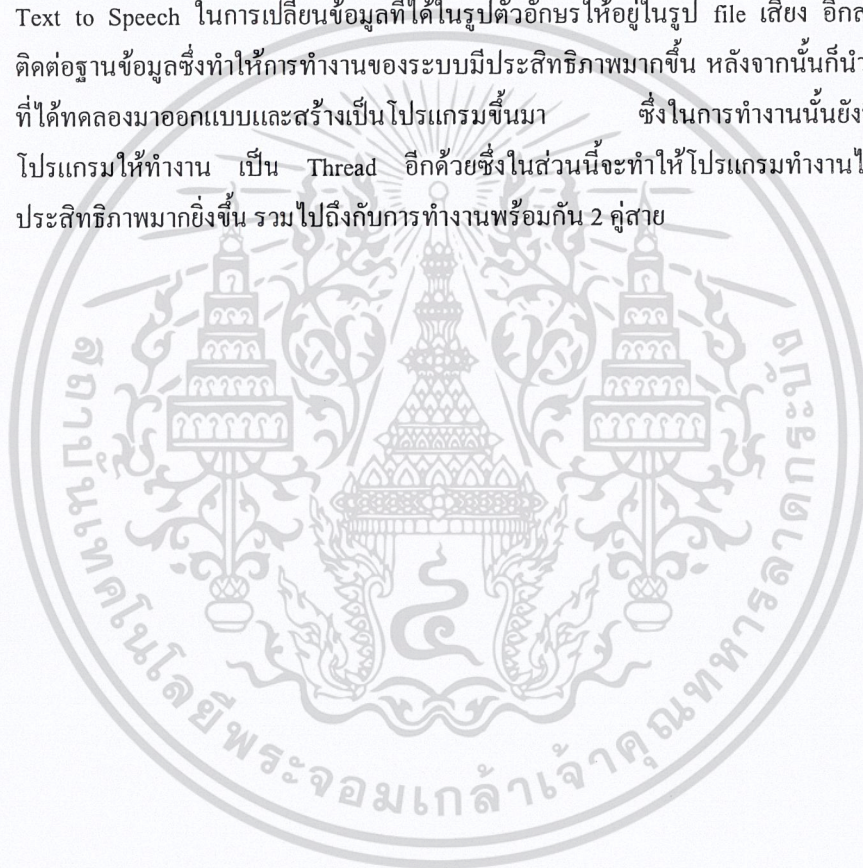
- ระบบสามารถรับ input จากผู้ใช้ทาง ปุ่มกดของโทรศัพท์ ( สัญญาณ DTMF )
- ระบบสามารถ output เป็นเสียงเพื่อตอบกลับไปหาผู้ใช้ได้
- ระบบสามารถเข้าไปค้นหาข้อมูลจากฐานข้อมูลได้
- ระบบสามารถตอบผลการเรียน ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผลการเรียนรู้เฉลี่ยของทุกเทอม
  - ผลการเรียนรู้เฉลี่ยเทอมที่ต้องการ
  - ผลการเรียนรู้รายวิชาของเทอมที่ต้องการ
- สามารถติดต่อได้อย่างน้อย 2 คู่สาย

#### 1.4 วิธีการดำเนินงาน

ปริญญาบัตรฉบับนี้เริ่มจากการศึกษาทฤษฎีขั้นพื้นฐานที่เกี่ยวข้องกับปริญญาบัตร ซึ่งมีหัวข้อต่าง ๆ ดังนี้ การเขียนโปรแกรมติดต่อกับโมเด็ม ตรวจสอบปุ่มกดจากโทรศัพท์ ส่งเสียงออกไปทาง Modem ตรวจสอบและวิเคราะห์ event ต่าง ๆ ของโมเด็มส่วนต่อไปคือทดลอง ใช้งาน Text to Speech ในการเปลี่ยนข้อมูลที่ได้ในรูปตัวอักษรให้อยู่ในรูป file เสียง อีกส่วนคือการติดต่อฐานข้อมูลซึ่งทำให้การทำงานของระบบมีประสิทธิภาพมากขึ้น หลังจากนั้นก็นำส่วนต่าง ๆ ที่ได้ทดลองมาออกแบบและสร้างเป็นโปรแกรมขึ้นมา ซึ่งในการทำงานนั้นยังมีการเขียนโปรแกรมให้ทำงาน เป็น Thread อีกด้วยซึ่งในส่วนนี้จะทำให้โปรแกรมทำงานได้อย่างเต็มประสิทธิภาพมากยิ่งขึ้น รวมไปถึงการทำงานพร้อมกัน 2 คู่สาย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### หลักการและทฤษฎี

#### 2.1 ทฤษฎีเกี่ยวกับโทรศัพท์

2.1.1 โทรศัพท์ที่ใช้งานกันอยู่ทั่วไปมีอยู่แบ่งเป็น 2 ประเภท

2.1.1.1 ระบบโทรศัพท์แบบหมุน

โทรศัพท์แบบหมุนจะใช้การขจัดจังหวะการไหลของกระแสโดยใช้การเปิดปิดสวิตช์ตามหมายเลขที่หมุน เช่น ถ้าหมุนหมายเลข 2 สวิตช์จะเปิดออก 2 ครั้งทำให้เกิด Pulse ออกไป 2 ลูก



รูปที่ 2.1 โทรศัพท์แบบหมุน

2.1.1.2 ระบบโทรศัพท์แบบกดปุ่ม (DTMF)

โทรศัพท์แบบกดปุ่มจะทำการส่งสัญญาณแบบคลื่นความถี่ โดยจะใช้การส่งสัญญาณความถี่ 2 ค่า Modulate ออกไปพร้อมกัน ทำให้รู้ได้ว่าผู้ใช้กดปุ่มอะไร เช่น หากมีการกดปุ่ม 1 จะทำการส่งความถี่ 697 Hz, 1209 Hz ออกไป

	1209 Hz	1336 Hz	1477 hZ
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

รูป 2.2 การ Cross ความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2 ส่วนประกอบอื่นๆของ วงจรโทรศัพท์

1. วงจรกำเนิดเสียงเรียก
2. วงจร switch hook
3. ไมล์

### 2.1.3 วงจรกำเนิดเสียงเรียก (ringer)

จะคอยส่งเสียงเมื่อมีการติดต่อเข้ามายังโทรศัพท์ ซึ่งวงจรมีจะติดต่อกับชุมสายโดยตรง

### 2.1.4 วงจร Switch hook

จะเป็นวงจรที่บอกทางฝั่งผู้ใช้พร้อมที่จะใช้ หรืออยู่ในสถานะวางสาย ในขณะที่วางสายจะให้กระแสไหลไม่ครบวงจร ในขณะที่ยกสายจะทำให้กระแสครบวงจรหลังจากที่กระแสไหลครบวงจรที่ชุมสายก็จะส่งสัญญาณ Dial Tone ไปยังผู้ใช้เพื่อเป็นการบอกให้สามารถกดหมายเลขได้

### 2.1.5 Microphone

สำหรับรับ Input ของเสียงเพื่อส่งไปยัง ผู้รับ

## 2.2 โมเด็ม

โมเด็ม (Modem) ย่อมาจาก modulate (แปลงสัญญาณ) กับ demodulate (คืนรูปสัญญาณ) การส่งข้อมูลเป็นดังนี้

Computer (DTE)  $\leftrightarrow$  UART  $\leftrightarrow$  Modem  $\leftrightarrow$  NETWORK

การส่งข้อมูลเริ่มจาก คอมพิวเตอร์ส่งข้อมูลซึ่งถูกส่งแบบขนานไปยัง UART ( Universal Asynchronous Receive Transmitter ) หลังจากนั้น UART จะทำการส่งข้อมูลที่รับมาและส่งออกในรูปแบบอนุกรม โดยส่งบิตที่มีนัยสำคัญต่ำสุดก่อน ไปให้ โมเด็ม หลังจากนั้นข้อมูลแต่ละบิตจะต้องถูกแปลงให้เป็นสัญญาณเสียงที่มีความถี่อยู่ในย่านที่สามารถส่งผ่านสายโทรศัพท์ได้ ( Modulate ) ก่อนที่จะส่งออก ไป เมื่อโมเด็มฝ่ายผู้รับ ได้รับก็จะทำการ Demodulate ก่อนแล้วส่งให้คอมพิวเตอร์ทางฝ่ายผู้รับอีกครั้ง

ชนิดการส่งข้อมูลของ โมเด็ม มี 3 ชนิดคือ

1. แบบ simplex ส่งข้อมูลที่ทางเดียว
2. แบบ half duplex ส่งข้อมูลได้ 2 ทาง แต่ต้องไม่ใช่เวลาเดียวกัน
3. แบบ full duplex ส่งข้อมูลได้ 2 ทาง และอาจเป็นเวลาเดียวกันก็ได้

ในการ Modulate จะทำการแปลง ข้อมูล "0" เป็นความถี่ 1070 Hz

ข้อมูล "1" เป็นความถี่ 1270 Hz

ในช่วงหลังโมเด็ม มีการพัฒนาให้มีความสามารถต่างๆเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Voice Modem

เป็น โมเด็ม ที่มีความสามารถในการ Modulate สัญญาณข้อมูลในลักษณะรูปแบบเสียงไว้ด้วยซึ่งเรียกว่า Voice Modem โดยทำการเพิ่มวงจรการแปลงข้อมูลเสียงทั้งจากสัญญาณอนาลอกเป็นดิจิทัล และจากสัญญาณดิจิทัลเป็นอนาลอก

### Detect DTMF

โดยโมเด็มนั้นสามารถจะตรวจสอบสัญญาณการกดปุ่มจากผู้ใช้อีกฝั่งได้

## 2.3 วิธีการเขียน โปรแกรมติดต่อกับ โมเด็ม

มีอยู่สามวิธีที่นิยมใช้ดังนี้

- ใช้ AT Command
- ใช้ API เช่น JTAPI , TAPI
- ใช้ Component

### AT Command

คำสั่งที่ใช้ควบคุมโมเด็ม ได้รับการพัฒนาโดยบริษัท Hayes ซึ่งเป็นบริษัทผลิตโมเด็มชั้นนำของโลก โดยนำชุดคำสั่งนี้ไปใช้กับโมเด็ม ของคนในเวลาต่อมาชุดคำสั่งนี้ก็เริ่มแพร่หลายใช้กับโมเด็ม จากทุกบริษัทจนมีการรับรองเป็นมาตรฐานของคำสั่งควบคุมการทำงานของ โมเด็ม ชุดคำสั่งเหล่านี้มีชื่อเรียกว่า Hayes Command หรือ AT Command มีลักษณะดังนี้ ATDT 123456 เป็นคำสั่งที่สั่งให้โมเด็ม โทรออกไปที่ เบอร์ 123456 , ATH จะสั่งให้วางหูโทรศัพท์ โดยคำสั่งเหล่านี้ไม่ได้สั่งให้โมเด็ม โดยตรง แต่จะสั่งให้ Driver ที่ควบคุม โมเด็ม ก่อน แล้ว Driver จึงจะนำคำสั่งนั้นไปสั่งงาน โมเด็ม อีกทีหนึ่ง

### ข้อดี

- ง่ายในการสร้างโปรแกรมขนาดเล็ก
- สามารถใช้คำสั่งที่เป็นคำสั่งเฉพาะของโมเด็ม แต่ละรุ่นได้

### ข้อเสีย

- ต้องทำการเขียน โปรแกรมกับ port โดยตรง

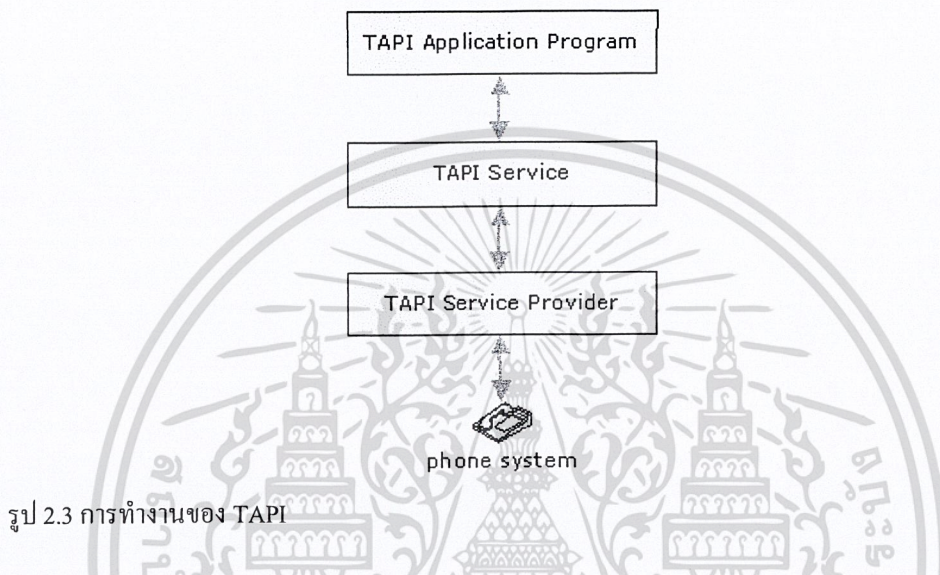
### API

เป็นเครื่องมือที่ถูกสร้างขึ้นมาเพื่อใช้ในการติดต่อกับ โมเด็ม ได้ง่ายขึ้นแทนแบบ AT Command ที่พบเห็นได้บ่อยคือ JTAPI จาก Sun เพื่อใช้งานกับภาษา JAVA และ TAPI จาก Microsoft โดยผู้ทำการเขียนโปรแกรมกับ API นั้น ไม่ต้องอ้างอิงถึงในระดับ physical เช่นต้องอ้างอิงกับ port ที่ใช้ต่อกับ โมเด็ม นั้น ผู้เขียนโปรแกรมจะอ้างอิงอุปกรณ์ทุกอย่างผ่าน API ไม่ว่าจะเป็นการค้นหา โมเด็ม หรือ Sound device ของ โมเด็ม ตัวนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของ Program ที่จะทำการควบคุมหรือสั่งงาน โมเด็ม โดยใช้ TAPI มีดังนี้

1. TAPI Application จะทำการ Load Tapi32.dll หรือ Tapi3.dll เพื่อจะใช้คำสั่งของ TAPI
2. TAPI จะทำการติดต่อกับ Tapisrv โดยส่งข้อมูลไปที่ TSP ( Telephony Service Provider )
3. TSP จะทำการสั่งงาน โมเด็ม



รูป 2.3 การทำงานของ TAPI

การทำงานของแต่ละชั้นเป็นดังนี้

1. TAPI Application Program เป็นส่วนที่ทำให้ User ได้ทำการต่างๆ เช่น โทรศัพท์ รับโทรศัพท์ ส่งแฟกซ์ รับแฟกซ์
2. TAPI Service เป็นส่วน Function พื้นฐานให้สำหรับ Application เรียกใช้ เช่น dialing
3. TAPI Service Provider ทำหน้าที่ แปลงคำสั่ง สำหรับอุปกรณ์หรือคำสั่งมาตรฐานของระบบโทรศัพท์ TAPI Service Provider สำหรับ โมเด็ม คือ UniModem5 Service Provider มีประโยชน์คือ ทำให้เราไม่ต้องเขียน โปรแกรมโดยขึ้นกับ Hardware Vender นั้น หากมีการเปลี่ยนแปลงมาตรฐานใหม่ Windows ก็จะมี TAPI Service Provider ตัวใหม่มาให้เอง

ความสามารถของ TAPI ในการทำงานอื่นๆ

1. ใช้คอมพิวเตอร์เป็นเครื่องตอบรับอัตโนมัติ หรือ เครื่องบันทึกข้อความ
2. ใช้เครื่องคอมพิวเตอร์เป็นเครื่องส่งหรือรับ แฟกซ์
3. ใช้เครื่องคอมพิวเตอร์แทน โทรศัพท์
4. ทำระบบ log on ระยะไกล
5. ทำระบบ โทรศัพท์หลายคู่สาย หรือทำ PBX Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. จัดการ การรับฝากข้อความข้ามเครือข่ายขนาดใหญ่
7. จัดการบริการข้อมูลเสียงในการแจ้งข่าวสาร โดยผู้โทรจะทำการตั้งข้อมูลไปซึ่งระบบจะทำการใช้ Text to speech
- 8 อื่นๆ

#### ข้อดี

- สามารถทำงานได้กับหลายภาษาและนิยมใช้ในระบบที่ทำงานบน Windows
- มีเอกสารแนะนำและตัวอย่างการใช้งานมากกว่า JTAPI ทำให้ใช้งานได้สะดวกกว่า

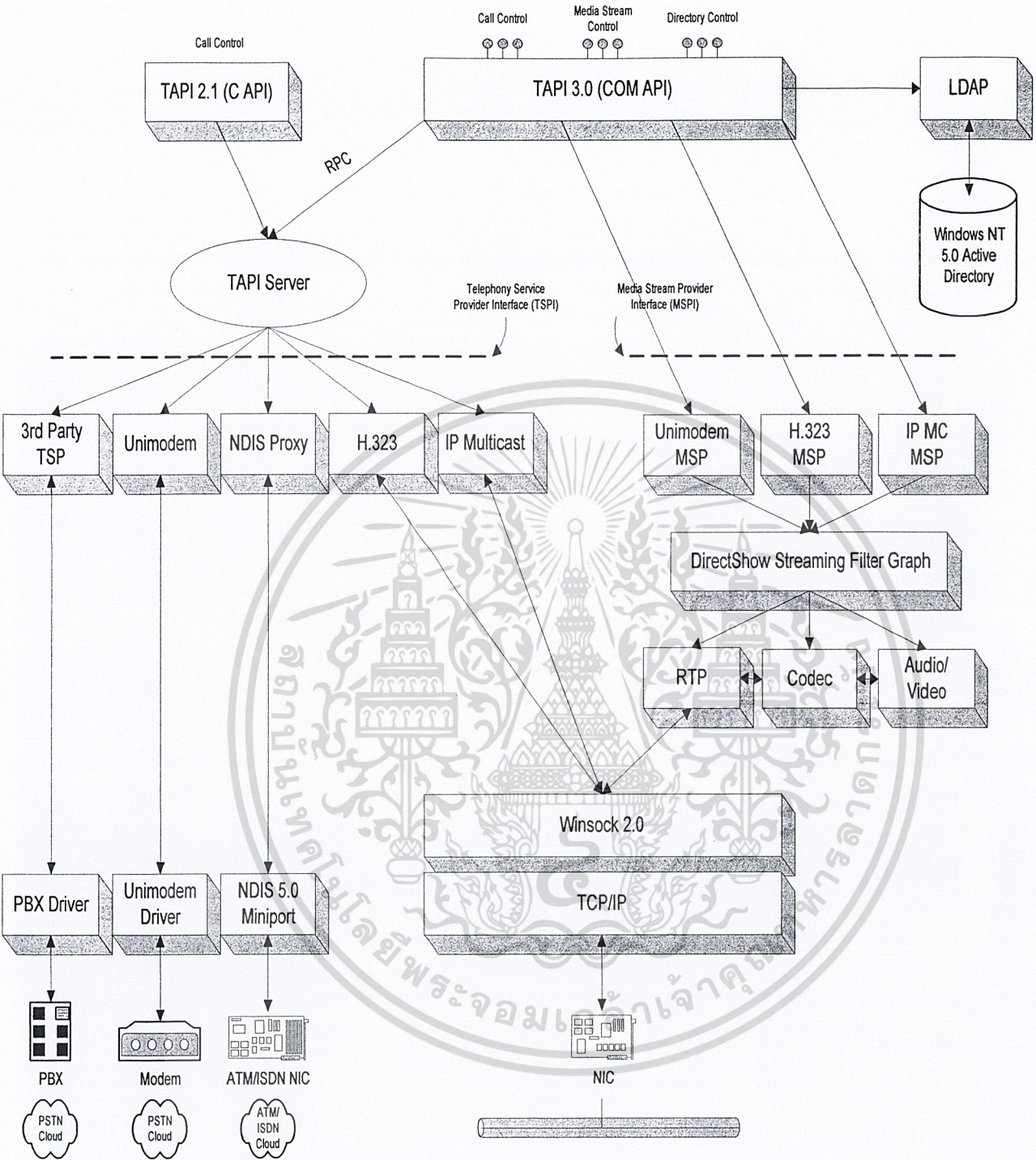
#### ข้อเสีย

- ต้องศึกษาการทำงานและคำสั่งของ TAPI ซึ่งมีขั้นตอนมาก
- คู่มือยังมีไม่มาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูป 2.4 โครงสร้างการทำงานของ TAPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Component

Component ส่วนใหญ่ที่มี จะเป็น Component ที่ใช้งานกับภาษา Visual Basic และทำงานบนระบบ Windows ซึ่งส่วนใหญ่จะมีการกำหนดราคาขายและมีให้เลือกใช้หลายรุ่น ซึ่งก็รวมทั้งรุ่นที่ไม่เสียค่าใช้จ่ายในการทดลองใช้ด้วยแต่ก็จะมีกำหนดระยะเวลาในการใช้งาน Component เช่น 15 วัน 1 เดือน หรือ 3 เดือน เป็นต้น โดย Component นั้นก็จะทำงานบน TAPI ต่างๆ อีกที่

### ข้อดี

- มีรุ่นฟรีให้ทดลองใช้ แม้จะมีการจำกัดระยะเวลาการใช้งาน
- ใช้งานง่ายและมีเอกสารประกอบการใช้งานพร้อมตัวอย่างการใช้งานอย่างครบถ้วน

### ข้อเสีย

- จำกัดเวลาทำให้ไม่สามารถนำมาใช้ในการทำงานจริงได้ ถ้าจะใช้ในระยะเวลานานจะต้องเสียค่าใช้จ่ายในการซื้อ Component
- ความสามารถในการทำงานจำกัด

จากข้อดีข้อเสียที่ได้ศึกษามา นี้ เป็นข้อดีข้อเสียที่เห็น ได้ชัดเจนในแต่ละแบบจึงเป็นเหตุผลให้เลือกใช้งาน TAPI กับระบบสอบตามผลการเรียนทางโทรศัพท์อัตโนมัติ เพราะระบบสอบตามผลการเรียนทางโทรศัพท์อัตโนมัติได้ออกแบบให้ทำงานบนระบบปฏิบัติการ Windows และการใช้งาน API จะช่วยให้การใช้งานทำงานได้ง่ายกว่าการใช้ AT Command โดยที่ไม่ต้องเสียค่าใช้จ่ายเพิ่มเติม

## 2.4 โปรแกรมติดต่อกับ ฐานข้อมูล

ระบบที่มีการจัดการข้อมูลขนาดใหญ่ มักจะใช้ระบบจัดการฐานข้อมูลเข้ามาช่วยเสมอเพื่อให้ง่ายและสะดวกในการเรียกใช้ข้อมูลต่างๆจากระบบ ระบบจัดการฐานข้อมูลในปัจจุบันมีให้เลือกใช้มากมาย อาทิเช่น ระบบจัดการฐานข้อมูล Oracle ระบบจัดการฐานข้อมูล MySQL ระบบจัดการฐานข้อมูล DB2 เป็นต้น

## 2.5 วิธีการใช้ Text to Speech

เนื่องจากระบบสอบตามผลการเรียนทาง โทรศัพท์อัตโนมัติ จะติดต่อกับผู้ใช้ให้ทราบผลการเรียน โดยการส่งข้อมูลกลับในรูปแบบสัญญาณเสียง ซึ่งข้อมูลที่ต้องการส่วนใหญ่จะถูกจัดเก็บอยู่ในรูปแบบของข้อความจึงมีความจำเป็นต้องแปลงข้อมูลจากข้อมูลในรูปแบบข้อความมาเป็นข้อมูลในรูปแบบสัญญาณเสียงก่อน Text to Speech จึงได้ถูกนำมาใช้ในหน้าที่นี้

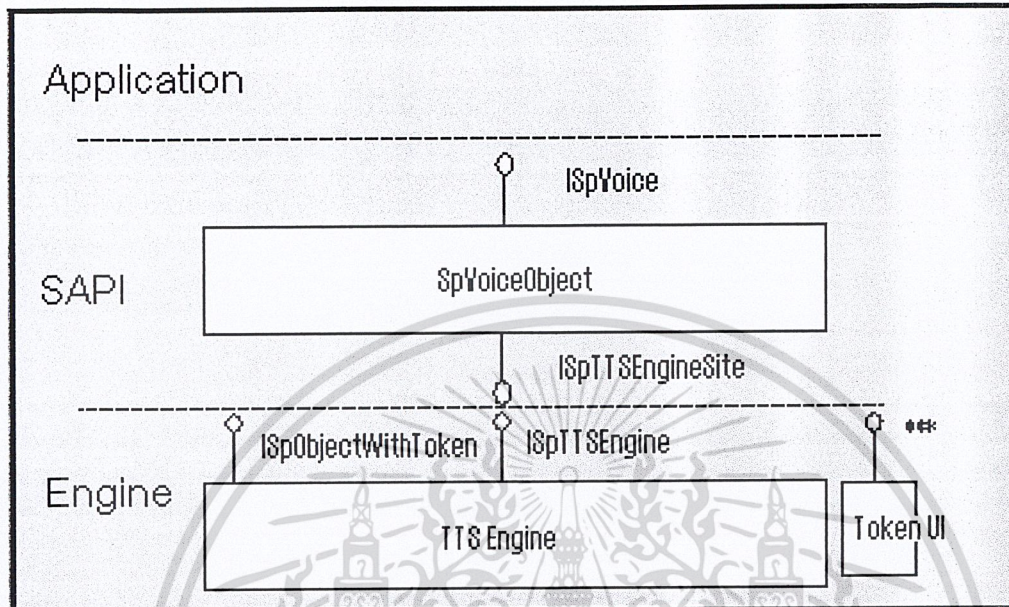
Text to Speech ที่เราเลือกใช้นี้เป็น Engine ที่ทางบริษัท Microsoft ได้พัฒนาขึ้น ซึ่งทำงานภายใต้ระบบปฏิบัติการ Windows ได้ดี

หลักการทำงานของ Text to Speech

การทำงานของ TTS นั้น จะมี TTS Engine ซึ่งเป็นตัวหลักที่ทำหน้าที่ในการแปลงข้อมูลในรูปแบบข้อความให้เป็นข้อมูลในรูปแบบสัญญาณเสียง โดยการทำงานจะควบคุมผ่านทาง ISpVoice

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถใช้อِسพีวอยซ์ ในการกำหนดลักษณะของเสียงที่ได้ เช่นคุณภาพเสียง คุณลักษณะของเสียงเป็นต้น และสามารถกำหนดผลการแปลงข้อมูลเป็นสัญญาณเสียงไว้หลายรูปแบบ โดยสามารถเก็บเป็นแฟ้มข้อมูลหรือสามารถนำมาเล่นเสียงออกลำโพงได้เลย



รูป 2.5 โครงสร้างการทำงานของ Text to Speech

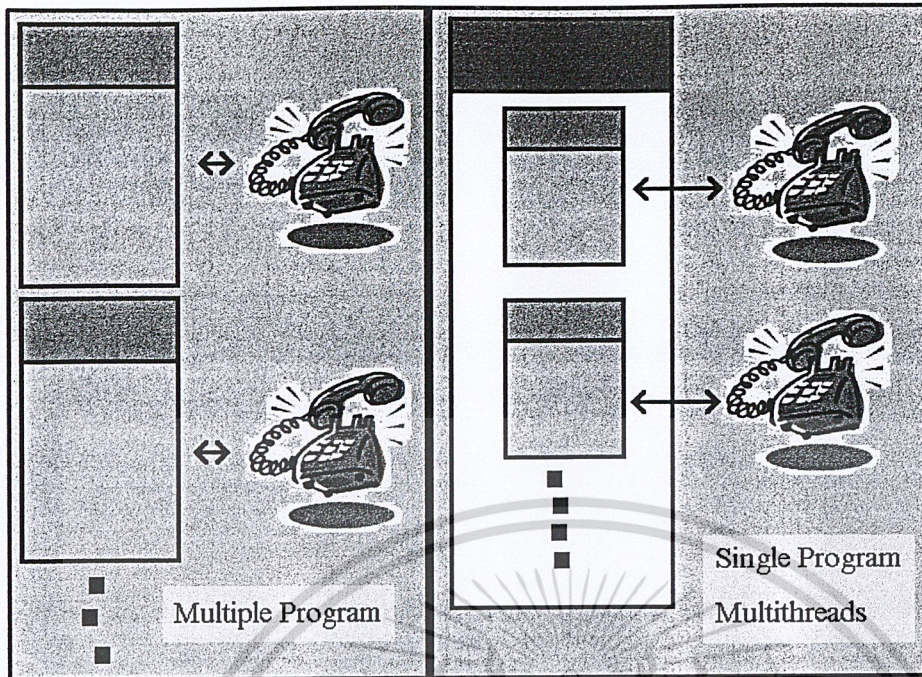
ข้อดี

- ทำงานร่วมกับระบบปฏิบัติการ Windows ได้ดี
- มีคู่มือเอกสารและตัวอย่างการใช้งานพอสมควร
- ไม่เสียค่าใช้จ่าย

## 2.6 โปรแกรมโดยใช้ Thread

เพื่อให้ระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติสามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างเพียงพอและไม่ทำให้สิ้นเปลืองทรัพยากรของระบบพร้อมทั้งช่วยให้สามารถทำการควบคุมการทำงานของระบบโดยรวมได้สะดวกขึ้นกว่าการทำงานของระบบโดยไม่อาศัยการทำงานของ Thread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.6 การเขียนโปรแกรมโดยใช้ Thread

## 2.7 การเขียนโปรแกรมเพื่อใช้งานเกี่ยวกับการเล่น file wave

ในการที่จะทำให้มีเสียงตอบกลับไปหาผู้ใช้ได้นั้น จะต้องมีการเขียนโปรแกรมให้อ่านข้อมูลในรูปแบบของเสียง และสั่งให้อุปกรณ์การเล่นเสียง ของโมเด็มทำงาน ซึ่งต้องใช้ **Multimedia API** ช่วยในการเขียน โปรแกรม ซึ่งในการใช้งานผู้ให้จะต้องรู้ว่าเล่นข้อมูลในรูปแบบเสียงจบตอนไหนก็ให้ปิดการทำงานนั้นซึ่งมีวิธีในการแก้ปัญหาได้ 2 วิธีคือ

- วน loop รอจนเสร็จ วิธีนี้ทำให้ ซีพียูทำงานร้อยเปอร์เซ็นต์ทำให้ไม่สามารถรับ Event อื่นๆได้เสีย
- ใช้ Thread กับ Semaphor โดยในการเล่นข้อมูลในรูปแบบเสียงให้ใช้เป็น Thread และเมื่อเล่นอยู่ให้ Wait รอ ถ้าต้องการให้จบการเล่นก็ให้ Signal ก็จะทำให้การปิดการเล่นในทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การใช้งาน TAPI 2.2

#### 3.1 ข้อกำหนดการใช้งาน TAPI 2.2

- กำหนดให้ใช้ TAPI32.LIB ที่ Project -> setting
- ทำการ include file TAPI.H ในการเขียน โปรแกรม
- สามารถเข้าถึง file TAPI32.DLL

#### 3.2 ขั้นตอนการใช้งาน TAPI 2.2

- 3.2.1 กำหนดค่าเริ่มต้น
- 3.2.2 ตรวจสอบความสามารถของอุปกรณ์
- 3.2.3 เปิดใช้งานอุปกรณ์ และอนุญาตการรับ Message ต่างๆ
- 3.2.4 การรับสายและการค้นหาอุปกรณ์ในการเล่น file wave ของ โมเด็ม

##### 3.2.1 กำหนดค่าเริ่มต้น

`lineInitializeEx(`

`LPHLINEAPP,`

`HINSTANCE,`

`LINECALLBACK,`

`LPCSTR,`

`LPDWORD,`

`LPDWORD,`

`LPLINEINITIALIZEEXPARAMS`

`);`

เพื่อทำการเริ่มการติดต่อระหว่าง Application กับ TAPI SERV ซึ่งจะต้องทำการกำหนดรูปแบบการรับ Message ของ TAPI ที่ขั้นตอนนี้ด้วยซึ่งเป็นการกำหนดที่มีผลโดยตรงต่อ Program โดยกำหนดที่ `lpLineInitializeExParams.dwOption` สามารถกำหนดได้ 3 แบบ คือ Callback function , Event Handle , Completion Port และจะมีการ Return จำนวนของอุปกรณ์ที่ TAPI รู้จักมาให้ สมมุติถ้ามีการ Return มา 5 แสดงว่ามีอุปกรณ์ ID 0 – 4 ให้สามารถเรียกใช้ได้

##### 3.2.2 ตรวจสอบความสามารถของอุปกรณ์โดยใช้ ฟังก์ชันต่างๆดังนี้

`LONG WINAPI lineNegotiateAPIVersion(`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HLINEAPP,
DWORD,
DWORD,
DWORD,
LPDWORD,
);

```

ฟังก์ชันสำหรับ ตรวจสอบเช็คอุปกรณ์ว่า สนับสนุนการทำงานของ TAPI version ที่  
ต้องการไม่

```

LONG WINAPI lineGetDevCaps(

```

```

HLINEAPP,
DWORD,
DWORD,
DWORD,
LPLINEDEVCAPS
);

```

ฟังก์ชันสำหรับตรวจสอบความสามารถของอุปกรณ์ รวมถึงชื่อของอุปกรณ์นั้นๆ

3.2.3 เปิดใช้งานอุปกรณ์ และอนุญาตการรับ Message ต่างๆ

```

LONG WINAPI lineOpen(

```

```

HLINEAPP,
DWORD,
LPHLINE,
DWORD,
DWORD,
DWORD_PTR,
DWORD,
DWORD,
LPLINECALLPARAMS const
);

```

ฟังก์ชันสำหรับเปิดใช้งาน Line หรือ อุปกรณ์นั้นๆ โดยมีกำหนดความสามารถ  
ของอุปกรณ์ เช่น เป็นเจ้าของสามารถสั่งงาน ได้ทุกอย่าง หรือเป็นผู้สังเกตการณ์ และยังกำหนดโหมดการ  
ทำงานได้ที่ฟังก์ชันนี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**LONG WINAPI lineSetStatusMessages(**

**HLINE,**

**DWORD,**

**DWORD**

**);**

ฟังก์ชันที่ทำการ บอกว่าให้ Line นี้สามารถรับ Message หรือ Event ประเภทไหน  
ได้บ้าง สามารถกำหนดได้หลายอย่าง ใน ฟังก์ชันเดียว

### 3.2.4 การรับสายและการค้นหาอุปกรณ์ในการเล่น file wave ของ โมเด็ม

**LONG WINAPI lineAnswer(**

**HCALL,**

**LPCSTR,**

**DWORD**

**);**

ฟังก์ชันที่เปรียบเสมือนการรับ โทรศัพท์หรือยกหู โทรศัพท์ที่ขึ้นมา โดยทั่วไปจะใช้  
ฟังก์ชันนี้ที่ Event LineCallState -> lineoffering โดยนำค่า hCall จาก hDevice ของ Message ที่รับมา จะ  
ทำให้สามารถทำให้รับ โทรศัพท์ได้ หากไปใช้รับที่ Event อื่นอาจจะรับไม่ได้เนื่องจากว่าเป็น hCall คนละ  
ตัวกัน

**LONG WINAPI lineGetID(**

**HLINE,**

**DWORD,**

**HCALL,**

**DWORD,**

**LPVARSTRING,**

**LPCSTR**

**);**

ฟังก์ชันนี้ทำหน้าที่รับค่า WaveDevicID ของอุปกรณ์ เช่น WaveID ของอุปกรณ์ที่  
สามารถ เล่น file wave ของ โมเด็มซึ่งจะใช้ในการตอบเป็นเสียงกลับไปหาผู้โทรเข้ามา เมื่อได้ WaveID  
แล้วก็ทำการใช้ ฟังก์ชัน ของ Multimedia API ช่วยในการเล่นเสียงตอบกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 ตัวอย่างการตรวจจับ Message ของ TAPI

- แบบ Hidden Window
- แบบ Event Handle

3.3.1 วิธีการใช้ Hidden Window โดยทำการเปิดอุปกรณ์ id=0;  
ประกาศตัวแปรที่สำคัญคือ

HLINEAPP hlineapp;

HCALL hcall;

HLINE hline

LINEINITIALIZEEXPARAMS setbehave;

DWORD numdevice;

LINEEXTENSIONID extensionid;

ขอโค้ดที่ Win main มีดังนี้

```
=====1. Initial TAPI พร้อมกับส่งชื่อฟังก์ชัน Callback ไปให้=====
setbehave.dwTotalSize = sizeof(LINEINITIALIZEEXPARAMS);
setbehave.dwOptions =
LINEINITIALIZEEXOPTION_USEHIDDENWINDOW;
temp=lineInitializeEx(&hlineapp,hInstance,(LINECALLBACK)lineCal
lbackFunc,"TAPI",&numdevice,&version,&setbehave);
```

===== 2. ตรวจสอบความสามารถของอุปกรณ์ =====  
ในกรณีที่เรารู้ความสามารถของอุปกรณ์แล้วไม่ต้องตรวจสอบ

=====3. เปิดอุปกรณ์ที่ต้องการใช้ โดยเปิด ID = 0 =====

```
temp=lineOpen(hlineapp,0,&hline,0x00010003,NULL,NULL,LINECALLP
RVILEGE_OWNER,LINEMEDIAMODE_AUTOMATEDVOICE,NULL);
```

===== พร้อมกับตั้งค่าการรับ Message ด้วย =====

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
temp=lineSetStatusMessages(hline,LINEDEVSTATE_CONNECTED |
LINEDEVSTATE_INSERVICE |LINEDEVSTATE_MSGWAITON |
LINEDEVSTATE_RINGING | LINEDEVSTATE_SIGNAL ,
LINEADDRESSSTATE_INUSEONE );
```

ส่วนที่คอยตรวจจับ Message ของ TAPI มีดังนี้

```
OID FAR PASCAL lineCallbackFunc(DWORD hDevice,
                                DWORD dwMsg,
                                DWORD dwCallbackInstance,
                                DWORD dwParam1,
                                DWORD dwParam2,
                                DWORD dwParam3 )
{
    switch(dwMsg)
    {
    case LINE_CALLSTATE:
        switch(dwParam1)
        {
            case LINECALLSTATE_DISCONNECTED:
                MessageBox(hwnd,"Disconnect","test callback Initial",MB_OK);
                break;
            case LINECALLSTATE_IDLE: /* your logic here */;
                MessageBox(hwnd,"State_idle","test callback Initial",MB_OK);
                break;
            case LINECALLSTATE_CONNECTED:
                break;
            case LINECALLSTATE_BUSY: /* your logic here */;
                MessageBox(hwnd,"bosy","test callback Initial",MB_OK);
                break;
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
/* This occur When first incoming ring */
////////////////////////////////////

    case LINECALLSTATE_OFFERING :

        hcall = (HCALL)hDevice;

        lineAnswer(hcall,0,0); // รับสาย

        temp=lineMonitorDigits(hcall,LINEDIGITMODE_DTMF );

        case LINECALLSTATE_ACCEPTED :

            break;

        case LINECALLSTATE_RINGBACK :

            break;

        case LINECALLSTATE_PROCEEDING:

            break;

        default :

            break;

    }

    case LINE_REPLY:

        break;

}

```

3.3.2 วิธีการใช้ Event Handle โดยทำการเปิดอุปกรณ์ id=0;

ประกาศตัวแปรที่สำคัญคือ

```

HLINEAPP hlineapp;

HCALL    hcall;

HLINE    hline

LINEINITIALIZEEXPARAMS    setbehave;

DWORD    numdevice;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LINEEXTENSIONID    extensionid;
HANDLE    htapi;    // Handle ไว้คอยรับ Message ของ TAPI
```

### ขอโค้ดที่ Win main มีดังนี้

```
===== 1. Initial TAPI พร้อมทั้งรับค่า Event Handle =====
    setbehave.dwTotalSize = sizeof(LINEINITIALIZEEXPARAMS);
    setbehave.dwOptions = LINEINITIALIZEEXOPTION_USEEVENT;
    temp=lineInitializeEx(&hlineapp,hInstance,NULL,"TAPI",&numdevic
    e,&version,&setbehave);
    htapi = setbehave.Handles.hEvent    // รับค่า Event Handle
===== 2. ตรวจสอบความสามารถของอุปกรณ์ =====
    ในกรณีที่เรารู้ความสามารถของอุปกรณ์แล้วไม่ต้องตรวจสอบ
===== 3. เปิดอุปกรณ์ที่ต้องการใช้ โดยเปิด ID = 0 =====
    temp=lineOpen(hlineapp,0,&hline,0x00010003,NULL,NULL,LINECALLP
    RVILEGE_OWNER,LINEMEDIAMODE_AUTOMATEDVOICE,NULL);
===== พร้อมกับตั้งค่าการรับ Message ด้วย =====
    temp=lineSetStatusMessages(hline,LINEDEVSTATE_CONNECTED |
    LINEDEVSTATE_INSERTSERVICE |LINEDEVSTATE_MSGWAITON |
    LINEDEVSTATE_RINGING | LINEDEVSTATE_SIGNAL ,
    LINEADDRESSSTATE_INUSEONE );
===== การสร้าง Thread เพื่อมาจัดการกับ Message =====
    CreateThread(NULL, 0, zTapiEventThread, NULL, 0, 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ฟังก์ชันที่ทำหน้าที่เป็น Thread สำหรับคอยจัดการ Message แบบ Event

Handle คือ

```

Void zTapiEventThread()
{
    LINEMESSAGE lm;
    while(true)
    {
        WaitForSingleObject(htapi, 10);
        lineGetMessage(hlineapp, &lm, 0)
        switch( lm.dwMessageID)
        {
            case LINECALLSTATE_OFFERING:
                break;
            case LINECALLSTATE_CONNECTED:
                break;
        }
    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การรับค่า Wave Device ID ของโมเด็ม

จะต้องทำการเรียกใช้ฟังก์ชัน lineGetID() ดังตัวอย่างด้านล่าง

```

VARSTRING *vs;
DWORD dwSize;
DWORD dwWaveDev;

    // allocate memory for structure
vs = (VARSTRING *) calloc (1, sizeof(VARSTRING));

    // set structure size
vs->dwTotalSize = sizeof(VARSTRING);
do
{
temp2=lineGetID(hline2,0l,hcall2,LINECALLSELECT_CALL,vs,"wave/out");
    _ltoa(temp2,buffer2,16);
    if(temp2!=0)
    {
        MessageBox(hwnd,buffer2,"Return from Line_GetID -> 0 is
Complete",MB_OK);
    }
    if(temp2)
        MessageBox(hwnd,"!!!! LineGetID Error !!!!","Error",MB_OK);
    if (vs->dwTotalSize < vs->dwNeededSize) {
        dwSize = vs->dwNeededSize;
        free (vs);
        vs = (VARSTRING *) calloc(1, dwSize);
        vs->dwTotalSize = dwSize;
        continue;
    } /* end if (need more space) */
break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}while(true);

DWORD dwWaveOutID;

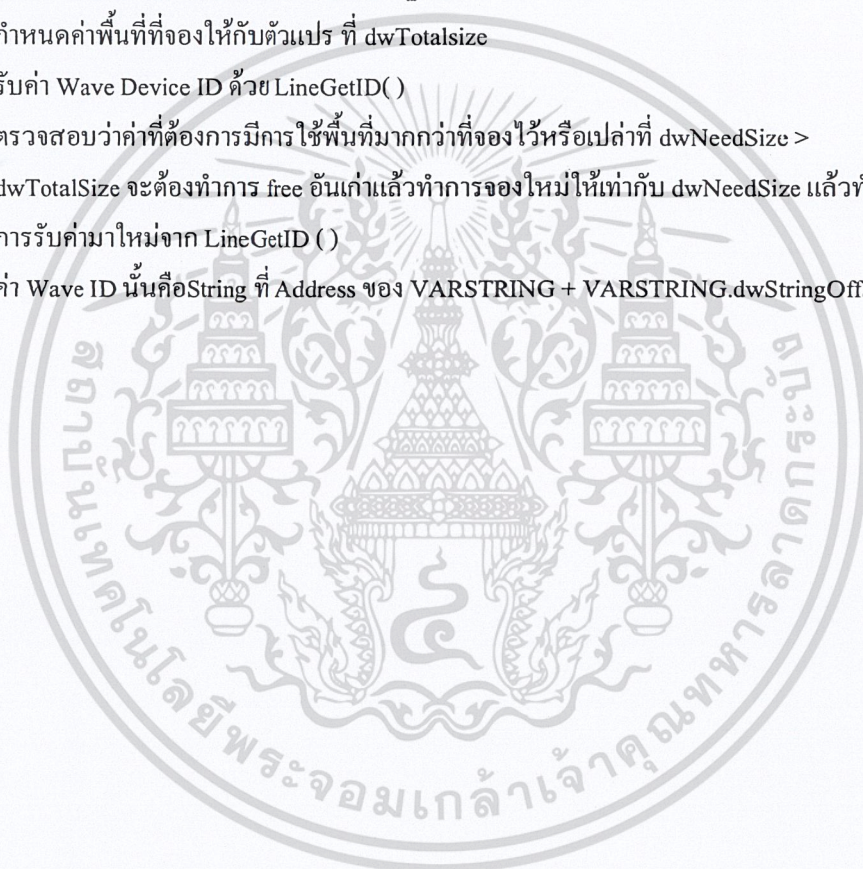
dwWaveOutID=(DWORD)*((DWORD*)((LPSTR)vs + vs->dwStringOffset));

free (vs);

```

### หลักการทำงานคือ

- ในการรับค่าของ Wave Device ID นั้นจะต้องใช้ตัวแปรชนิด VARSTRING ในการรับค่ามา ดังนั้นจึงต้องสร้างตัวแปรนี้ขึ้นมา
- ทำการจองพื้นที่ในการใช้งานในขนาดพื้นฐาน
- กำหนดค่าพื้นที่ที่จองให้กับตัวแปร ที่ dwTotalsize
- รับค่า Wave Device ID ด้วย LineGetID ( )
- ตรวจสอบว่าค่าที่ต้องการมีการใช้พื้นที่มากกว่าที่จองไว้หรือเปล่าที่ dwNeedSize > dwTotalSize จะต้องทำการ free อันเก่าแล้วทำการจองใหม่ให้เท่ากับ dwNeedSize แล้วทำการรับค่ามาใหม่จาก LineGetID ( )
- ค่า Wave ID นั้นคือString ที่ Address ของ VARSTRING + VARSTRING.dwStringOffset



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การใช้งาน Multimedia API

การใช้งาน Multimedia API เพื่อเล่น ข้อมูลเสียงออกทางโมเด็ม

- ข้อกำหนดการใช้งาน
- การเตรียมข้อมูล
- การเล่นเสียงออกโมเด็ม และการจบการเล่น

#### 4.1 ข้อกำหนดการใช้งาน

1. Include "Mmsystem.h"
2. ใช้ Winmm.lib กำหนดที่ Project -> setting -> link
3. ในเครื่องต้องมี Winmm.dll
4. Include Process.h สำหรับในกรณีที่มีการใช้ Thread
5. Include winbash.h , windows.h และใช้ kernel32.dll สำหรับการใช้งาน Semaphore

#### 4.2 การเตรียมข้อมูล

ข้อมูลเสียงที่มีนั้นอยู่ในรูปของ file.wav ข้อมูลนี้ไม่สามารถนำไปเล่นที่ โมเด็มได้ ต้องทำการเตรียมให้อยู่ใน Buffer ก่อน โดยทำการแยกส่วนที่เป็น wave จริงออกมา พร้อมทั้งมีการอ่านค่า Format ของ file.wav เพื่อเตรียมไว้ตรวจสอบกับ โมเด็มว่า สามารถใช้งานได้หรือไม่ โดย Format ที่รับได้คือ PCM 16 bit mono

ตัวอย่างการเตรียมข้อมูลเสียง จาก Source Code ที่ได้ค่าที่ได้จะอยู่ใน bufferin ซึ่งจะนำไปใช้ในการส่งไปให้โมเด็มเพื่อเล่นอีกทีหนึ่ง

```
HWAVEOUT hWaveOut;
```

```
UINT num;
```

```
HMMIO hmmio;
```

```
long hmmioIn,ret;
```

```
int errorCode;
```

```
//===== Open Wavefile by mmio\Open Window API =====//
```

```
hmmio = mmioOpen(filename2,0, MMIO_READ | MMIO_ALLOCBUF );
```

```
if( hmmio==NULL )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MessageBox(hwnd,"!! Error mmioOpen"," mmio Error ",MB_OK);
//===== Call function to find wave data =====//
MMCKINFO parent;
memset(&parent,0,sizeof(MMCKINFO));

parent.fccType = mmioFOURCC('W','A','V','E');
mmioDescend(hmmio,&parent,0,MMIO_FINDRIFF);

//===== find format data =====//
MMCKINFO child;
memset(&child,0,sizeof(MMCKINFO));
child.fccType = mmioFOURCC('f','m','t',' ');
mmioDescend(hmmio,&child,&parent,0);

//===== Read format wave file =====//
WAVEFORMATEX wavefmt; // Variable wave Format
mmioRead(hmmio,(char*)&wavefmt,sizeof(wavefmt));
if(wavefmt.wFormatTag != WAVE_FORMAT_PCM) ;
        MessageBox(hwnd,"LoadDirectSound mmioOpen","ERRO
WARNING",MB_OK);

//===== find data =====//
mmioAscend(hmmio,&child,0);
child.ckid = mmioFOURCC('d','a','t','a');
mmioDescend(hmmio,&child,&parent,MMIO_FINDCHUNK);

//===== Allocate soundbuffer and read sound data =====//
HGLOBAL hmem;
VOID *bufferin;
GlobalFree(hmem);
hmem = GlobalAlloc(0x40,child.cksize );
bufferin = GlobalLock(hmem);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
ret = mmioRead(hmmio,(char*)bufferin,child.cksize );
long numsample;
numsample = child.cksize / wavfmt.nBlockAlign ;
mmioClose(hmmio, 0);
```

#### 4.3 การเล่นเสียงออกโมเด็ม และการจบการเล่น

- 4.3.1 ในขั้นแรกคือการเปิด Wave Device ID พร้อมทั้งตรวจสอบ Format ว่าโมเด็มสามารถรองรับได้หรือไม่ ซึ่งในขั้นตอนนี้จะได้ Handle ของการส่งงานอุปกรณ์ออกมา โดยใช้ฟังก์ชัน waveOutOpen ( )
- 4.3.2 จัดเตรียมการเล่นด้วย waveOutPrepareHeader ( )
- 4.3.3 ทำการเล่นเสียงด้วย waveOutWrite ( )
- 4.3.4 ทำการจบการเล่นด้วยการปิดการทำงาน

ตัวอย่างโปรแกรมในการเล่นและปิดการทำงานที่ 1 ซึ่งได้มีการเตรียมข้อมูลมาจากโปรแกรมข้างบน

```
if(errorCode=waveOutOpen(&hWaveOut,(UINT)dwWaveOutID2,
&wavfmt,(DWORD)waveOutProc2 , 0,CALLBACK_FUNCTION ))
{
    _itoa(errorCode,buffer,10);
    MessageBox(hwnd,buffer," Wave outopen Error ",MB_OK);
}
WAVEHDR WaveHeader;
WaveHeader.dwBufferLength = numsample * wavfmt.nBlockAlign ;
WaveHeader.lpData = (LPSTR) bufferin ;
WaveHeader.dwFlags = WaveHeader.reserved = 0;

waveOutPrepareHeader(hWaveOut, &WaveHeader, sizeof(WaveHeader));
waveOutWrite(hWaveOut, &WaveHeader, sizeof(WaveHeader));
do {}
while (!(WaveHeader.dwFlags & WHDR_DONE));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
waveOutUnprepareHeader(hWaveOut, &WaveHeader, sizeof(WaveHeader));
```

```
waveOutClose(hWaveOut);
```

จากโปรแกรมข้างบน โปรแกรมจะต้องทำการวน loop รอจนกว่าจะเล่น file.wav จนจบวิธีนี้จะทำให้ CPU ทำงาน 100 เปอร์เซ็นต์ เครื่องจะไม่สามารถทำงานต่อได้เลยเป็นวิธีที่ไม่ดี

วิธีแก้ไขคือ

1. ทำให้เป็น Thread แล้วใช้ Sleep แทรกเข้าไปจะทำให้ CPU ว่าง แต่ไม่สามารถสั่งให้หยุดเล่นได้ต้องรอจนเล่นจบจบ
2. ทำให้เป็น Thread แล้วใช้ Semaphore โดยการ Wait รอ และถ้าต้องการสั่งให้หยุดการเล่นจึงทำการ Signal จะหลุดออกมาปิดการเล่นได้

วิธีการทำให้ฟังก์ชันเป็น Thread โดยการใช้ฟังก์ชัน

```
_beginthread(playwave_to_modem2,0,NULL);
```

Playwave\_to\_mdem2 คือฟังก์ชันที่ต้องการให้เป็น Thread

วิธีการใช้งาน Semaphore

1. ประกาศตัวแปร Handle เพื่อไว้เป็น Semaphore
2. สร้างตัวแปร Semaphore โดยใช้ฟังก์ชันดังนี้  

```
semaphor1 = CreateSemaphore(NULL,0,1,"sem1");
```
3. การ Wait รอ โดยตื่นขึ้นมาตรวจดูเป็นระยะจากฟังก์ชันดังนี้  

```
while (WaitForSingleObject(semaphor1,300) != WAIT_OBJECT_0);
```
4. การ Signal เพื่อให้หลุดจากการ Wait ใช้ฟังก์ชันดังนี้  

```
ReleaseSemaphore(semaphore1,1,NULL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### การใช้งานโปรแกรมแปลงข้อความเป็นสัญญาณเสียง

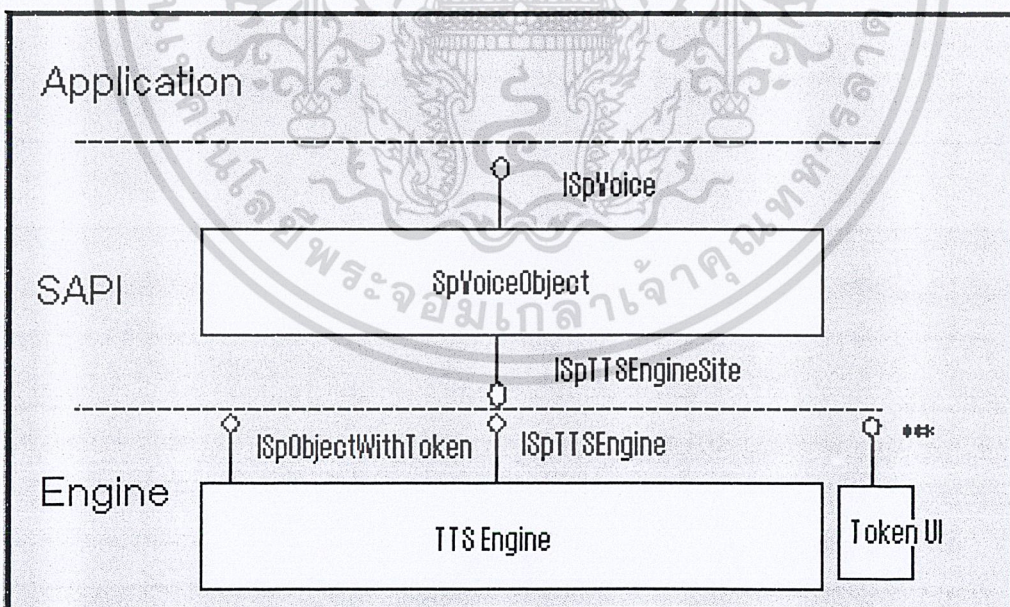
ความสามารถในการแปลงข้อความเป็นสัญญาณเสียงได้นั้น เป็นประโยชน์อย่างมาก เพราะเป็นการช่วยให้การนำข้อมูลที่เป็นข้อความ มาแปลงเป็นสัญญาณเสียงแล้วส่งผ่านไปให้กับผู้ใช้งานทางโทรศัพท์ได้ และที่สำคัญยังช่วยลดพื้นที่ในการเก็บข้อมูลสัญญาณเสียงที่ต้องเก็บไว้เล่นโดยเฉพาะซึ่งจะใช้พื้นที่ในการจัดเก็บเป็นจำนวนมากเพื่อให้มีข้อมูลเสียงเพียงพอกับที่ต้องการ

แต่การแปลงข้อความเป็นสัญญาณเสียงนี้เป็นเพียงส่วนหนึ่งในความสามารถของ ตัวจัดการเกี่ยวกับสัญญาณเสียง หรือ Speech API ซึ่งมีความสามารถเพียงพอที่จะจดจำสัญญาณเสียงเพื่อแปลงกลับเป็นข้อความได้ แต่ความสามารถการทำงานยังไม่น่าพอใจ

#### 5.1 การติดตั้งเพื่อให้สามารถใช้งานส่วน โปรแกรมแปลงข้อความเป็นสัญญาณเสียง

เนื่องจาก การใช้งานในส่วน โปรแกรมแปลงข้อความเป็นสัญญาณเสียงนี้นั้น ต้องการการประกาศชนิดข้อมูลให้ตัวแปลภาษาเข้าใจ ทาง Microsoft จึง ได้มีการจัดเตรียมไว้ให้ พร้อมกับข้อมูลช่วยเหลือในการใช้งานด้วย โดยสามารถหาข้อมูลได้จาก [www.microsoft.com](http://www.microsoft.com) ในหัวข้อเกี่ยวกับ SAPI หรือ Speech API โดยที่ใช้ในการทำโครงการนี้คือ SAPI 5.1 เมื่อทำการติดตั้งแล้วจะได้ Library และ เอกสารช่วยเหลือสำหรับการใช้งาน ซึ่งก็พอช่วยในการค้นหาได้ แต่ก็ยังคงต้องอาศัยการสุ่มใช้งานเพื่อให้ได้ผลลัพธ์ที่ต้องการ

#### 5.2 การทำงานภายในของตัวจัดการเกี่ยวกับสัญญาณเสียง



รูป 5.1 โครงสร้างการทำงานของตัวจัดการเกี่ยวกับสัญญาณเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวจัดการเกี่ยวกับสัญญาณเสียงนั้นมีการแบ่งระดับ ของส่วนของการทำงานออกเป็น 3 ส่วน คือ

- Application แทน โปรแกรมต่าง ๆ ที่จะมาขอติดต่อกับให้ทำการแปลงข้อความเป็นสัญญาณเสียง
- SAPI เป็นส่วนที่แต่ละ โปรแกรมจะใช้งานเพื่อทำการติดต่อกับส่วนหลักของตัวจัดการ
- Engine เป็นส่วนการทำงานหลักของตัวจัดการเกี่ยวกับสัญญาณเสียง ที่จะทำงานเป็นการบริการของระบบปฏิบัติการ Windows เพื่อรอให้บริการ

การใช้งาน โปรแกรมแปลงข้อความเป็นสัญญาณเสียงนั้น เราเริ่มจากการใช้งาน ISpVoice ซึ่งเป็นส่วนสำคัญที่เราจะต้องใช้สำหรับการแปลงข้อมูลจากข้อความเป็นสัญญาณเสียงนี้ และเป็นตัวหลักเพียงตัวเดียวที่เราจะใช้สำหรับการติดต่อกับส่วนการทำงานหลักของตัวจัดการเกี่ยวกับสัญญาณเสียง

### 5.3 ตัวอย่างการเขียน โปรแกรมเรียกใช้ Text to Speech

นี่เป็นตัวอย่างบางส่วนสำหรับการใช้งานให้ตัวจัดการเกี่ยวกับสัญญาณเสียงช่วยในการแปลงข้อความให้เป็นสัญญาณเสียง

```
ISpVoice *pVoice= NULL;
HRESULT hr = CoCreateInstance(CLSID_SpVoice,
                             NULL, CLSCTX_ALL, IID_ISpVoice, (void **)&pVoice);
if(SUCCEEDED(hr)){
    hr = pVoice->Speak(CharPtr, 0, NULL);
    pVoice->Release();
    pVoice = NULL;
}
```

เริ่มต้นด้วยการประกาศให้การบริการของตัวจัดการเกี่ยวกับสัญญาณเสียงรู้จัก ISpVoice ของเราเสียก่อน แล้วจึงสั่งให้มีการแปลงข้อความเป็นสัญญาณเสียงได้ แล้วจบด้วยการยกเลิก ISpVoice

แต่การทำงาน โดยปกติของตัวจัดการเกี่ยวกับสัญญาณเสียงจะกำหนดให้มีทางออกของสัญญาณเสียงคือ ลำโพง ทำให้ข้อความที่เราส่งให้ตัวจัดการเกี่ยวกับสัญญาณเสียงแปลงออกมาเป็นสัญญาณเสียงให้ นั่นก็จะถูกปล่อยออกมาที่ลำโพงทันที

ถึงแม้จากรูปแสดงการทำงานภายในของตัวจัดการเกี่ยวกับสัญญาณเสียงนั้นจะไม่แสดงถึงตัวช่วยอื่น ๆ ที่จะมาช่วยในการกำหนดลักษณะต่าง ๆ ของการแปลง เช่น ทิศทางการส่งออกของสัญญาณเสียง หรือแม้แต่ รูปแบบของสัญญาณเสียง เช่น ความถี่ ระดับความละเอียดของสัญญาณเสียง เป็นต้น แต่จากการติดตั้งที่ผ่านมาในตอนต้นนั้น ได้มีการเพิ่มตัวช่วยไว้ให้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถกำหนดให้ตัวจัดการเกี่ยวกับสัญญาณเสียง สามารถเปลี่ยนทิศทางการส่งออกของสัญญาณเสียงไปบันทึกเก็บไว้ได้ โดยการใช้ตัวช่วยตัวอื่นที่มีมาพร้อมกับการติดตั้งนั้นด้วย ดังตัวอย่างโปรแกรมต่อไปนี้

```

CComPtr <ISpVoice>          cpVoice;
CComPtr <ISpStream>         cpStream;
CSpStreamFormat             cAudioFmt;
HRESULT hr = cpVoice.CoCreateInstance(CLSID_SpVoice); //Set the audio format
if(SUCCEEDED(hr)){
    hr = cAudioFmt.AssignFormat(SPSF_8kHz16BitMono);
    //Call SPBindToFile, a SAPI helper method, to bind the audio stream to the file
    if(SUCCEEDED(hr)){
        hr = SPBindToFile(fileName,SPFM_CREATE_ALWAYS,
            &cpStream,&cAudioFmt.FormatId(),cAudioFmt.WaveFormatExPtr());
        //set the output to cpStream so that the output audio data will be stored in cpStream
        if(SUCCEEDED(hr)){
            hr = cpVoice->SetOutput( cpStream, TRUE );
            //Speak the text "hello world" synchronously
            if(SUCCEEDED(hr)){
                hr = cpVoice->Speak(CharPtr,SPF_DEFAULT,NULL);
                //close the stream
                if(SUCCEEDED(hr)){
                    hr = cpStream->Close();
                    //Release the stream and voice object
                }
            }
        }
    }
}
cpStream.Release ();
cpVoice.Release();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในแบบนี้เริ่มจากการประกาศ ISpVoice ให้บริการของตัวจัดการเกี่ยวกับสัญญาณเสียง ทำการสร้างรูปแบบของสัญญาณเสียงที่ต้องการ แล้วทำการกำหนดทิศทางการส่งออกของสัญญาณเสียง ไปทำการจัดเก็บไว้พร้อมกับการตั้งค่ารูปแบบของสัญญาณที่ได้สร้างไว้แล้วด้วย

ต่อจากนั้นสั่งให้ทำการแปลงข้อมูลแล้วตามด้วยการส่งยกเลิก ISpVoice พร้อมกับตัวช่วยอีกสองตัวคือ cpStream ที่จัดการเกี่ยวกับทิศทางการส่งออกของสัญญาณเสียงและ cpVoice ที่จัดการเกี่ยวกับรูปแบบของสัญญาณเสียง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

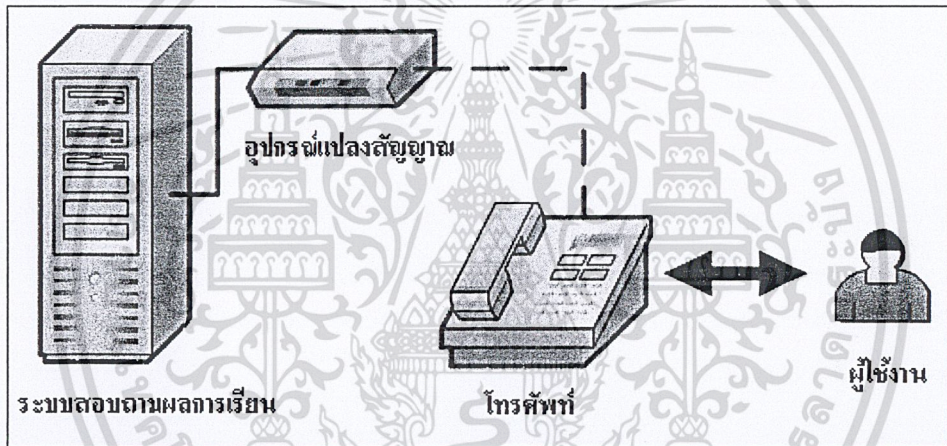
## บทที่ 6

### การทำงานและการออกแบบ

เราได้ทำการสร้างระบบขึ้นเพื่ออำนวยความสะดวกให้กับผู้ใช้งาน ให้สามารถสอบถามผลการเรียนได้อย่างสะดวกและง่าย เพราะโทรศัพท์มีอยู่ทั่วไป สามารถหาใช้งานได้สะดวก รวดเร็ว

ปัจจุบันคอมพิวเตอร์มีบทบาทมากในชีวิต เพราะมีความสามารถในการเก็บข้อมูลได้อย่างถูกต้อง และยังมีแนวโน้มที่จะสามารถเก็บข้อมูลได้มากขึ้น ระบบนี้ยังเป็นการเพิ่มความสามารถให้คอมพิวเตอร์ในด้านของการกระจายข้อมูล โดยที่ระบบไม่ต้องการคอมพิวเตอร์คุณภาพสูง ระบบทำงานโดยใช้โทรศัพท์ยกรต่ำ บำรุงรักษาง่าย และใช้ต้นทุนน้อย

#### 6.1 ภาพรวมของระบบ



รูปที่ 6.1 ภาพรวมของระบบ

ระบบจะทำการจองสายโทรศัพท์ที่เตรียมไว้สำหรับการสอบถามผลการเรียน เพื่อเตรียมพร้อมรับการติดต่อจากผู้ใช้งานที่จะติดต่อเข้ามา เมื่อมีผู้ใช้งานติดต่อเข้ามา ระบบจะส่งสัญญาณเสียงตอบกลับเป็นการติดต่อกับผู้ใช้งาน การทำงานของระบบอาจมีบางขั้นตอนที่ต้องการให้ผู้ใช้งานเลือก และระบบจะใช้สัญญาณการกดปุ่มจากผู้ใช้งาน เป็นวิธีการตอบรับจากผู้ใช้งานสู่ระบบของเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.2 ขอบเขตการใช้งาน

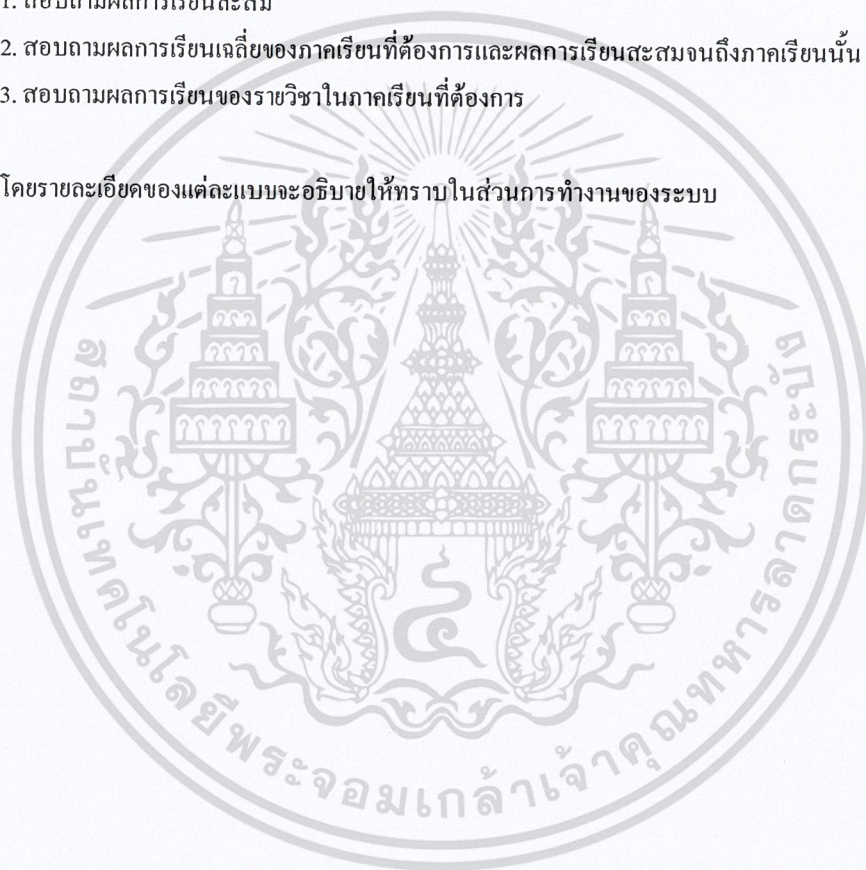
ผู้ใช้งานสามารถโทรศัพท์เข้ามาใช้งานได้ตลอดเวลาที่ระบบเปิดให้บริการอยู่ โดยแต่ละสายจะสามารถทำการติดต่อได้เพียงท่านเดียว เมื่อผู้ใช้งานเข้าติดต่อกับระบบและได้รับการตอบรับจากระบบแล้วผู้ใช้งานสามารถติดต่อกับระบบด้วยวิธีการดังนี้

1. ฟังเสียงจากโทรศัพท์เพื่อทราบรายละเอียด ข้อมูล และขั้นตอนการทำงาน
2. กดปุ่มโทรศัพท์เพื่อเลือกตัวเลือกในการทำงาน และป้อนข้อมูลที่ระบบต้องการ

เพียงเท่านี้ผู้ใช้งานก็สามารถใช้งานระบบได้ โดยมีแบบการสอบถามให้ผู้ใช้งานสามารถใช้งานระบบดังนี้

1. สอบถามผลการเรียนสะสม
2. สอบถามผลการเรียนเฉลี่ยของภาคเรียนที่ต้องการและผลการเรียนสะสมจนถึงภาคเรียนนั้น
3. สอบถามผลการเรียนของรายวิชาในภาคเรียนที่ต้องการ

โดยรายละเอียดของแต่ละแบบจะอธิบายให้ทราบในส่วนการทำงานของระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 6.3 ความสามารถพื้นฐานของระบบ

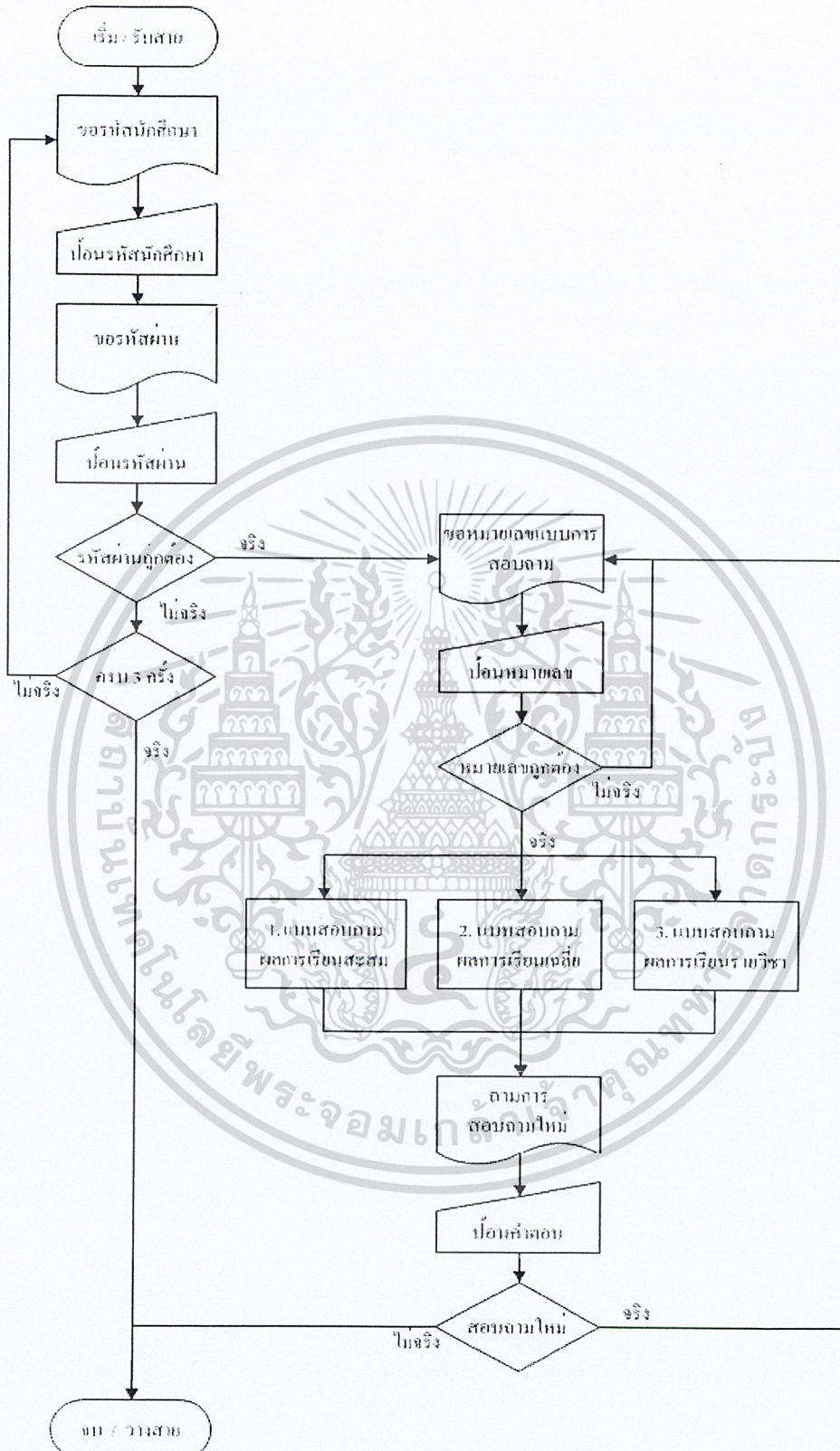
ระบบต้องสามารถทำการติดต่อกับผู้ใช้งานเพื่อตอบรับการสอบถามของผู้ใช้งาน โดยทำการค้นข้อมูลจากฐานข้อมูลเพื่อแจ้งให้กับผู้ใช้งานให้ทราบอย่างถูกต้อง ในเรื่องของเวลาการให้บริการนั้น ระบบควรจะสามารถให้บริการได้ตลอดเวลา ซึ่งทำให้ระบบต้องมีการทำงานที่ถูกต้องอยู่เสมอและพร้อมที่จะทนรับกับความผิดพลาดที่อาจเกิดโดยไม่คาดคิดได้ เช่น สามารถรู้ถึงสภาพการติดต่อกับผู้ใช้งานว่าเป็นการติดต่อแบบปกติ หรือเกิดความผิดพลาดต่าง ๆ ขึ้นแล้ว ซึ่งอาจมีการส่งวางสายถ้าเห็นว่าการติดต่อนั้นไม่อยู่ในความเหมาะสมเพื่อให้สามารถรองรับการติดต่อใช้งานของผู้ใช้งานได้อย่างทั่วถึงและถูกต้อง

### 6.4 สิ่งที่ระบบต้องการ

1. ระบบต้องการการติดต่อทางโทรศัพท์โดยผ่านทางอุปกรณ์แปลงสัญญาณ Modem โดยระบบจะทำการติดต่อกับอุปกรณ์โดยใช้ Telephony Application Programming Interface (TAPI) เป็นตัวติดต่อกับอุปกรณ์เพื่อติดต่อกับผู้ใช้งานผ่านทางระบบโทรศัพท์
2. ในการติดต่อกับผู้ใช้งานผ่านทางระบบโทรศัพท์นั้นระบบจำเป็นต้องส่งข้อมูลให้ผู้ใช้งานทราบในรูปแบบสัญญาณเสียงเพียงแบบเดียวเท่านั้น ดังนั้นระบบจึงต้องสามารถแปลงข้อความจากฐานข้อมูลเป็นสัญญาณเสียงเพื่อติดต่อส่งข้อมูลให้ผู้ใช้งานได้รับฟังได้ โดยใช้ Speech Application Programming Interface (SAPI)
3. ระบบต้องสามารถค้นหาข้อมูลจากฐานข้อมูลเพื่อทำการตอบรับการสอบถามจากผู้ใช้งานได้
4. ระบบการจัดการฐานข้อมูลที่สามารถใช้งานผ่านทางตัวจัดการ ODBC ได้
5. เลขหมายโทรศัพท์เพื่อเป็นการติดต่อระหว่างระบบและระบบโทรศัพท์
6. อุปกรณ์แปลงสัญญาณโมเด็ม สำหรับการแปลงสัญญาณเพื่อส่งข้อมูลผ่านไปในระบบโทรศัพท์ จากสิ่งที่ต้องการข้างต้นทำให้ระบบ ซึ่งสร้างขึ้นด้วย Visual C++ 6.0 ซึ่งเป็นของ Microsoft ต้องทำงานอยู่บนระบบปฏิบัติการ Windows โดยเป็นรุ่น Windows 2000 ขึ้นไป เพื่อความสามารถในการรองรับสิ่งที่ระบบต้องการดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.5 การทำงานของระบบ



รูปที่ 6.2 การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ผู้ใช้งานโทรศัพท์ติดต่อเข้าใช้งานกับระบบได้แล้ว ระบบจะทำการร้องขอ รหัสนักศึกษา และรหัสผ่านจากผู้ใช้งาน ผู้ใช้งานจะต้องทำการป้อนรหัสนักศึกษาเป็นตัวเลขจำนวน 8 หลัก และป้อนรหัสผ่านเป็นตัวเลขจำนวน 6 หลัก เมื่อระบบได้รับรหัสทั้งสองแล้ว ระบบจะทำการตรวจสอบความถูกต้องของรหัสนักศึกษา และรหัสผ่านที่ได้รับกับข้อมูลที่มีอยู่ในฐานข้อมูล

ถ้ารหัสนักศึกษา และ รหัสผ่านที่ป้อนเข้ามานั้นตรงกับข้อมูลที่มีอยู่ในฐานข้อมูลระบบจะทำงานต่อในขั้นต่อไป แต่ถ้ารหัสทั้งสองที่ป้อนเข้ามานั้น ไม่มีอยู่ในฐานข้อมูล ระบบจะให้ทำการป้อนข้อมูลทั้งสองใหม่ ผู้ใช้งานจะสามารถป้อนรหัสผิดได้สองครั้ง ถ้าครั้งที่สามผู้ใช้งานยังป้อนรหัสผิดอีก ระบบจะทำการตัดการติดต่อกับผู้ใช้งานในครั้งนั้นทันที

เมื่อผ่านการตรวจสอบรหัสนักศึกษาและรหัสผ่านมาแล้ว ระบบจะให้ผู้ใช้งานเลือกแบบการสอบถามซึ่งมีอยู่ 3 แบบ โดยมีรายละเอียดของการใช้งานในแต่ละแบบดังนี้

#### 1. การสอบถามผลการเรียนสะสม

เป็นการสอบถามผลการเรียนสะสมของผู้ติดต่อใช้งานระบบ จากข้อมูลทั้งหมดของผู้ติดต่อใช้งานที่มีอยู่ในฐานข้อมูล ตั้งแต่เริ่มเข้าศึกษาจนถึงปัจจุบัน ซึ่งจะทำให้ผู้ใช้งานทราบผลการเรียนสะสมนี้ได้อย่างสะดวก

#### 2. การสอบถามผลการเรียนเฉลี่ยของภาคเรียนที่ต้องการ

จะทำการร้องขอให้ผู้เข้าใช้งานระบบ ทำการป้อนปีการศึกษา และภาคเรียนที่ต้องการเพื่อเป็นการกรองข้อมูล ให้เหลือเฉพาะข้อมูลของผู้ใช้งานระบบที่ต้องการ โดยจะทำการแจ้งผลการเรียนเฉลี่ยของภาคเรียนที่ต้องการ และตามด้วยผลการเรียนสะสมตั้งแต่ผู้ใช้งานเข้าศึกษา จนถึงภาคเรียนที่ระบุด้วย

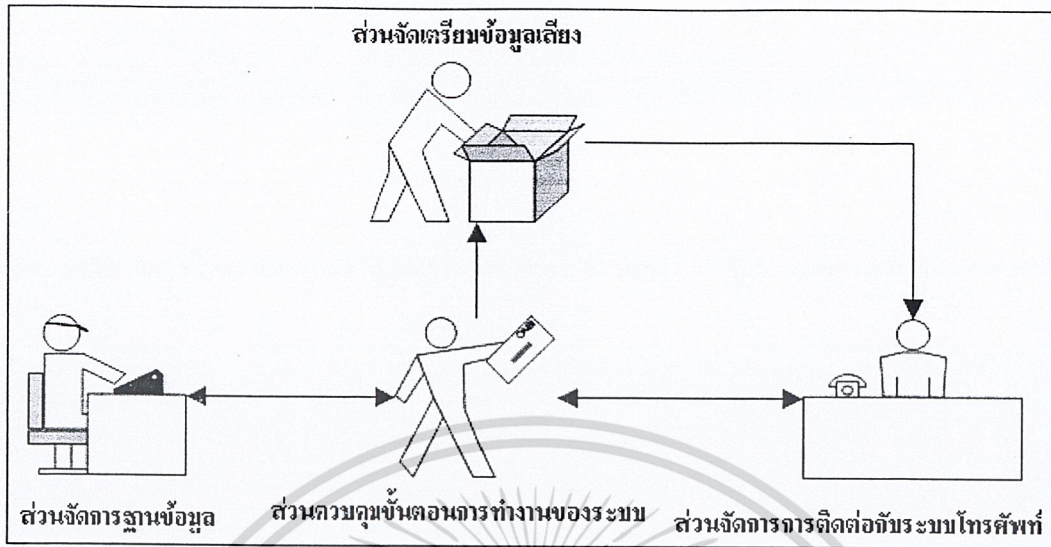
#### 3. การสอบถามผลการเรียนของรายวิชาในภาคเรียนที่ต้องการ

ในแบบนี้ ระบบจะทำงานคล้ายกับในแบบที่สอง คือผู้ใช้งานต้องทำการป้อนปีการศึกษา และ ภาคเรียนที่ต้องการฟังผลการเรียนให้กับระบบ แล้วระบบจะทำการค้นข้อมูลจากฐานข้อมูล โดยระบบจะบอกรายชื่อวิชา และผลการเรียนของแต่ละรายวิชา ที่ละรายวิชาจนครบภาคเรียนนั้น ๆ และจะทำการบอกผลการเรียนในข้อสองด้วย คือทำการบอกผลการเรียนเฉลี่ยของภาคเรียนที่ต้องการนั้น และผลการเรียนสะสมตั้งแต่เริ่มเข้ารับการศึกษา จนถึงภาคเรียนนั้น ต่อท้ายไปด้วย

เมื่อผู้ใช้งานเลือกสอบถามแบบใด ๆ แล้วระบบจะดำเนินการตามแบบที่ต้องการ หลังจากนั้นผู้ใช้งานสามารถเลือกที่จะสอบถามในแบบอื่นต่อ หรือเลือกจบการสอบถามก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.6 ส่วนประกอบของระบบ



รูปที่ 6.3 ส่วนประกอบของระบบ

เราสามารถแบ่งส่วนประกอบของระบบ ออกเป็นส่วน ๆ การทำงาน ดังนี้

1. ส่วนควบคุมขั้นตอนการทำงานของระบบ
2. ส่วนจัดการฐานข้อมูล
3. ส่วนจัดเตรียมข้อมูลเสียง
4. ส่วนจัดการการติดต่อด้วยระบบ โทรศัพท์

ทุกส่วนจะทำงาน โดยอาศัยการควบคุมจากส่วนควบคุมขั้นตอนการทำงานของระบบ เพื่อทำการตอบ การสอบถามของผู้ใช้งานตามขั้นตอน โดยแต่ละส่วนจะมีหน้าที่การทำงานดังนี้

1. ส่วนควบคุมขั้นตอนการทำงานของระบบ  
เป็นส่วนควบคุมขั้นตอนการทำงานของระบบ โดยเรียกใช้งานส่วนประกอบอื่น ๆ ให้ทำงานตามลำดับ ขั้นตอน เพื่อให้ระบบสามารถทำการตอบรับการสอบถามจากผู้ใช้งานได้อย่างถูกต้อง ในส่วนนี้เราต้องทำการออกแบบและสร้างขึ้นเองทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. ส่วนจัดการฐานข้อมูล

ส่วนจัดการฐานข้อมูลนี้จะทำงานเมื่อส่วนควบคุมขั้นตอนการทำงานของระบบต้องการค้นหาข้อมูลที่เก็บอยู่ในฐานข้อมูล ส่วนนี้จะทำการค้นหาเพื่อให้ได้คำตอบที่ส่วนควบคุมขั้นตอนการทำงานของระบบต้องการ

การทำงานของส่วนนี้จะทำการติดต่อกับระบบจัดการฐานข้อมูลผ่านทาง Open DataBase Connectivity (ODBC) เพื่ออำนวยความสะดวก สำหรับการเปลี่ยนแปลงระบบจัดการฐานข้อมูลในภายหลัง ก็สามารถทำได้โดยไม่ต้องทำการแก้ไขโปรแกรมของระบบเลย

### โครงสร้างตารางฐานข้อมูลที่ระบบใช้งาน

ระบบจะทำการค้นหาข้อมูลจาก ODBC ที่ชื่อ Student โดยฐานข้อมูลนี้ เราได้ออกแบบให้โครงสร้างประกอบด้วย 3 ตารางดังนี้

#### - ตาราง Student

ใช้สำหรับเก็บรายละเอียดของนักศึกษา รวมทั้งรหัสผ่านในการเข้าใช้งานระบบด้วย มีโครงสร้างดังนี้

ชื่อคอลัมน์	ชนิด	ขนาด (Byte)	รายละเอียด
ID	Integer	4	รหัสประจำตัวของนักศึกษา
Fname	String	-	ชื่อนักศึกษา
Lname	String	-	นามสกุลนักศึกษา
Pass	Integer	4	รหัสผ่าน เลขจำนวน 6 หลัก

ตาราง 6.1 ตารางแสดงโครงสร้างของ ตารางข้อมูล Student ในฐานข้อมูล

#### - ตาราง Subject

ใช้เก็บรายละเอียดเกี่ยวกับวิชาต่าง ๆ โดยมีโครงสร้างดังนี้

ชื่อคอลัมน์	ชนิด	ขนาด (Byte)	รายละเอียด
ID	Integer	4	รหัสวิชา
Subject	String	128	ชื่อวิชา ภาษาอังกฤษ
Weight	Integer	4	จำนวนหน่วยกิต

ตาราง 6.2 ตารางแสดงโครงสร้างของ ตารางข้อมูล Subject ในฐานข้อมูล

#### - ตาราง Grade

ใช้เก็บผลการเรียนทุกวิชาของนักศึกษาทุกคน โดยมีโครงสร้างดังนี้

ชื่อคอลัมน์	ชนิด	ขนาด (Byte)	รายละเอียด
StdID	Integer	4	รหัสนักศึกษาที่มีจริงจาก [Student]
SubID	Integer	4	รหัสวิชาที่มีจริงจาก [Subject]
Year	Integer	2	ปีการศึกษาที่ลงทะเบียน
Term	Integer	1	ภาคเรียนที่ลงทะเบียน
Grade	String	-	ผลการเรียน

ตาราง 6.3 ตารางแสดงโครงสร้างของ ตารางข้อมูล Grade ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Grade จะมีค่าที่เป็นไปได้ ดังนี้

ผลการเรียน	อธิบาย	ค่า
ไม่มีข้อมูล(ว่าง)	สำหรับรายวิชาที่ยังไม่ทราบผลการเรียน	
A	Excellent	4.00
B+	Very Good	3.50
B	Good	3.00
C+	Fairly Good	2.50
C	Fair	2.00
D+	Poor	1.50
D	Very Poor	1.00
F	Failed	0.00
Fa	Failed , insufficient Attendance	0.00
Fe	Failed , absent from Examination	0.00
Fw	Failed , late Withdrawal	0.00
W	Withdrawal	
I	Incomplete	
S	Satisfactory	
U	Unsatisfactory	

ตาราง 6.4 ตารางแสดงค่าที่เป็นไปได้ของผลการเรียนในแต่ละรายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. ส่วนจัดการข้อมูลเสียง

เพราะการสื่อสารกับมนุษย์ผ่านทางระบบโทรศัพท์ทำให้การสื่อสารกลับไปหามนุษย์นั้น ต้องเป็นในรูปแบบสัญญาณเสียง แต่ข้อมูลที่เก็บอยู่ในฐานข้อมูลไม่ได้เก็บอยู่ในรูปสัญญาณเสียง ระบบจึงต้องแปลงข้อมูลเหล่านั้นเป็นสัญญาณเสียงเสียก่อน เพื่อที่จะส่งกลับไปให้มนุษย์เข้าใจได้

ในการทำงานของส่วนนี้ เมื่อส่วนควบคุมขั้นตอนการทำงานของระบบต้องการแปลงข้อมูลให้อยู่ในรูปแบบสัญญาณเสียงส่งงานมา ซึ่งแบ่งการส่งงานออกได้เป็นสามส่วนคือ

#### (1) บอกให้ผู้ใช้งานทราบถึงขั้นตอนการทำงาน

ในส่วนแรก ถ้าส่วนควบคุมขั้นตอนการทำงานของระบบต้องการบอกให้ผู้ใช้งานทราบถึงขั้นตอนการทำงาน เพื่อให้ผู้ใช้งาน ทำตามขั้นตอน ได้อย่างถูกต้อง ในการบอกขั้นตอนการทำงานนั้น ขั้นตอนที่ต้องบอกให้ผู้ใช้งานทราบอยู่ไม่มาก และต้องทำการบอกทุกครั้งกับผู้ใช้งานเมื่อผู้ใช้งาน ได้ผ่านไปจนถึงขั้นตอนนั้น ๆ เราจึงสามารถที่จะบันทึกการบอกขั้นตอนนี้เป็นข้อมูลเสียงเตรียมไว้ได้ เมื่อต้องการใช้ก็ใช้ได้ทันที

#### (2) บอกผลการเรียน

แบบต่อมาคือการบอกผลการเรียนให้ผู้ใช้งานทราบ สัญญาณเสียงที่ใช้ในการบอกผลการเรียนนั้นก็จะมีจำนวนจำกัด เพราะผลการเรียนประกอบด้วยตัวเลข 0 ถึง 9 , จุด , ตัวอักษรภาษาอังกฤษ และ + เท่านั้น เราจึงสามารถทำการบันทึกข้อมูลเสียงสำหรับแต่ละตัวแล้วนำมาต่อกันเพื่อบอกให้ผู้ใช้งานทราบถึงผลการเรียนได้

#### (3) บอกรายชื่อของแต่ละวิชา

ในการบอกรายชื่อของวิชาเรียนนั้น เราจะพบว่าวิชาเรียนทั้งหมด มีอยู่เป็นจำนวนมาก ทั้งที่การบอกรายวิชานั้นใช้เพียงส่วนน้อยของรายวิชาทั้งหมด ทำให้การบันทึกข้อมูลเสียงของชื่อรายวิชานั้นจะเป็นการใช้พื้นที่ในการจัดเก็บมากเกินไป เราจึงให้ระบบสามารถทำการแปลงข้อมูลชื่อรายวิชาจากข้อความ ไปเป็นข้อมูลเสียงก่อนที่จะทำการส่งผลให้กับผู้ใช้งาน

ส่วนนี้จะใช้ SAPI ในการแปลงข้อมูลที่ได้จากฐานข้อมูล จากข้อความเป็นสัญญาณเสียง

### 4. ส่วนจัดการการติดต่อกับระบบโทรศัพท์

ระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัตินี้ใช้อุปกรณ์แปลงสัญญาณ โมเด็ม ในการเชื่อมต่อคอมพิวเตอร์เข้ากับระบบโทรศัพท์ ทำให้เราต้องสามารถใช้งาน โมเด็ม ได้

เราใช้ TAPI ในการจัดการการติดต่อกับระบบ โทรศัพท์โดยมีสิ่งที่ต้องการตั้งการดังนี้

- การตรวจจับสัญญาณการเรียกเข้า
- การรับสาย-วางสาย เพื่อเริ่มและสิ้นสุดการติดต่อ
- การส่งสัญญาณเสียงเพื่อแจ้งขั้นตอนให้กับผู้ใช้งาน
- การตรวจจับสัญญาณการกดปุ่ม โทรศัพท์จากผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### ผลการทดลอง

ระบบที่ใช้ทดลองคือ

- Microsoft Window XP
- โมเด็ม US Robotic 2 ตัว

#### 7.1 การทดสอบการใช้งาน TAPI

TAPI ที่นำมาใช้นั้นเพื่อนำมาช่วยในการควบคุม โมเด็ม ซึ่ง TAPI นั้นตามคุณสมบัติคือสามารถใช้กับ ระบบปฏิบัติการจำพวก Windows ได้ เช่น Window98 , Window me , Window 2000 , Window XP แต่ก็มีข้อกำหนดอีกอย่างคือ โมเด็ม ที่ TAPI จะรู้จักนั้นต้องทำมาสนับสนุน TAPI ด้วยจึงไม่สามารถใช้กับ โมเด็ม ได้ทุกตัว นอกจากนั้นแต่ละระบบปฏิบัติการในจำพวก Window TAPI ก็มีความสามารถในการรู้จัก หรือเรียกใช้ โมเด็ม ได้ไม่เท่ากันเช่น โมเด็ม U.S Robotics หากใช้ใน Window XP TAPI สามารถเรียกใช้งาน โมเด็ม ได้ตามปกติ แต่ถ้าเป็น Window 2000 TAPI ไม่สามารถใช้งาน U.S. Robotics ได้

จากที่ได้ทดลองใช้งาน TAPI บน Window XP สามารถ ควบคุม โมเด็ม ได้เป็นอย่างดีปัญหาส่วนใหญ่เป็นเรื่องการใช้งานที่ลำบาก และ TAPI มีการใช้งานหลายรูปแบบทำให้ผู้ศึกษาในตอนเริ่มต้นสับสนได้ สิ่งที่ได้ทดลองคือ

- การConnect กับ โมเด็ม
- การตรวจจับปุ่มกด
- การตรวจจับเสียงเรียกของโทรศัพท์
- การตอบรับกลับไปทาง โมเด็ม
- การวางสาย

จากที่ได้ทดลองสามารถใช้งาน ได้เป็นอย่างดี ยกเว้นการวางสาย (disconnect ) TAPI ทำได้ช้า ทำให้การเริ่มรับสายใหม่ต้องรอสักครู่

#### 7.2 การทดสอบการใช้งาน Text to Speech

Text to Speech เป็น API ที่ Microsoft พัฒนาจนเป็นชุด SDK ( Software Development Kid ) มีตัวอย่างพร้อมทั้งวิธีการใช้งาน ทำให้ศึกษาและทดลองได้ง่าย โดยได้ทดลองต่างๆดังนี้

- ทดลองการติดต่อกับ Text to Speech
- ทดลองการแปลงจาก Text เป็น เสียงออกทางลำโพง
- ทดลองการแปลงจาก Text เป็น file ในรูปแบบ wave
- ทดลองปรับแต่งลักษณะของเสียง เช่น ความเร็ว , รูปแบบของเสียง , คุณภาพของเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากที่ได้ทดลองใช้ Text to Speech นั้นสามารถทำงานได้ แต่เสียงที่ได้ค่อนข้างจะฟังยากสำหรับการแปลงบางประโยค

### 7.3 การทดสอบการเชื่อมต่อกับฐานข้อมูล

เนื่องจากในโครงการนี้ใช้ Visual C++ ในการเขียนจึงได้ทดลองเขียน โปรแกรมในการ เชื่อมต่อกับฐานข้อมูล 2 แบบคือ

- เชื่อมต่อในสภาวะที่ใช้ MFC ( Microsoft Fundamental Class )
- เชื่อมต่อในสภาวะที่ใช้ Win 32 API

ในการเชื่อมต่อแบบ MFC นั้นมี Class ซึ่งคอยจัดการเกี่ยวกับการเชื่อมต่อกับฐานข้อมูล การดึงข้อมูลมาแสดงผลได้เป็นอย่างดี ทำให้ง่ายต่อการเขียน โปรแกรม ส่วนการเชื่อมต่อในสภาวะ Win 32 API นั้นทำได้ค่อนข้างยากเพราะฟังก์ชันส่วนใหญ่ทำงานได้ในระดับล่าง และหลังจากที่ได้ทดลองใช้ในการติดต่อใช้แล้ว ผลที่ได้คือสามารถเชื่อมต่อได้เป็นบางครั้งเท่านั้น อาจเนื่องมาจากว่าใช้ ฟังก์ชันไม่ถูกเท่าที่ควรก็เป็นไปได้

### 7.4 ผลการทดสอบการใช้งาน Multimedia API

เนื่องจากในโครงการนี้ต้องการ เล่น file wave ไปที่ โมเด็ม จึงได้ทดลองใช้งาน Multimedia API ดังนี้

- จัดเตรียม Buffer
- อ่านรูปแบบของ file wave เพื่อนำไปตรวจสอบว่า โมเด็ม สนับสนุนในรูปแบบนี้หรือไม่
- ตั้งให้ โมเด็ม เล่น file wave
- ตั้งให้หยุดเล่น

จากการทดลองสามารถใช้งานได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

# บทวิจารณ์และสรุป

### 8.1 บทวิจารณ์

จากการทดลองทำได้พบข้อจำกัดต่างๆดังต่อไปนี้

1. TAPI ไม่สามารถทำงานได้ครบถ้วนบนระบบปฏิบัติการ Window ที่ต่างกันออกไป ส่วนใหญ่จะทำงานได้ดีที่ Window XP
2. การเขียนโปรแกรมเชื่อมต่อกับฐานข้อมูลบนสถานะแวดล้อมแบบ Win 32 API นั้นทำได้ลำบากมาก เพราะฉะนั้นควรเขียนโปรแกรมบนสถานะแวดล้อมแบบ MFC

### 8.2 บทสรุป

โครงการระบบสอบถามผลการเรียนอัตโนมัติสรุปได้ดังหัวข้อต่อไปนี้

1. การติดต่อกับโมเด็มนั้นสามารถทำงานได้ค่อนข้างดี ช่วยทำให้การเขียนโปรแกรมง่ายขึ้น แต่ยังมีปัญหาเรื่องการวางสาย และใช้งานกับโมเด็มได้ยังไม่หลายรุ่นเท่าไร และทำงานได้ดีบน Window XP

2. การใช้งาน Text to Speech นั้นเป็น API ที่ใช้งานและมีคู่มือการใช้งานเป็นอย่างดีทำให้การเขียนโปรแกรมสะดวก แต่เสียงที่ได้ออกมานั้นยังฟังไม่ค่อยรู้เรื่อง

โดยรวมโครงการนี้ถือว่าประสบความสำเร็จพอสมควรเนื่องจากทำงานพื้นฐานได้ค่อนข้างครบ

### 8.3 แนวทางในการพัฒนาต่อ

1. เนื่องจากขั้นตอนต่างๆในระบบเป็นขั้นตอนที่กำหนดมาไม่สามารถเปลี่ยนแปลงขั้นตอนการตอบรับได้ น่าจะทำให้ขั้นตอนต่างๆมีการเปลี่ยนแปลงได้โดยไม่ต้องแก้ไขโปรแกรมทำให้สะดวกขึ้นในการนำไปประยุกต์ใช้งานต่างๆ
2. ควรนำ Text to Speech ภาษาไทยมารวมเข้าด้วยทำให้สามารถใช้งานข้อความภาษาไทยได้
3. ควรปรับปรุงให้ทำงานได้หลายคู่สายมากขึ้น โดยมีส่วนในการตั้งค่าของระบบว่าจะให้ทำงานหลายคู่สาย
4. ควรทำโปรแกรมให้สามารถเลือกอุปกรณ์ได้อัตโนมัติ
5. ใช้ TAPI version 3 ซึ่งเป็น Component C++ ทำให้ออกแบบโปรแกรมได้ง่าย ส่วน version 2.2 นั้น เป็น C API
6. เพิ่มส่วนของการอัดเสียงเข้าไปในโปรแกรมด้วยเพื่อให้ง่ายต่อผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. MSDN Library Help

ในหัวข้อต่อไปนี้

- To Learn About the Voice Modem Extensions for Windows 95, Press 1 Now!
- For the Telephony API, Press 1; For Unimodem, Press 2; or Stay on the Line
- Telephony Application Programming Interface (TAPI) Programmer's Reference

2. Microsoft Speech SDK, version 5.1 Help

3. จาก Web Site ต่างๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้