

โปรแกรมควบคุมบราวเซอร์สำหรับช่วยเหลือคนพิการ

Assistive Browser for The Disable



นาย คมสัน จันท์เจริญ

นาย พงศกร พูนพิเชฐธรรม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

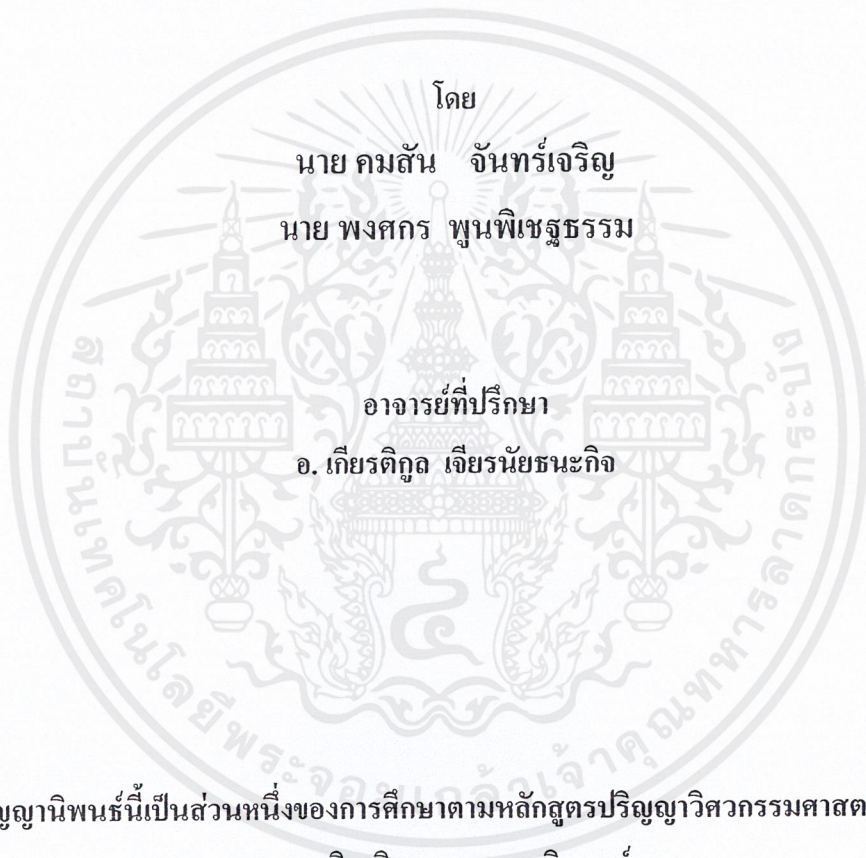
เลขที่.....
เลขทะเบียน...46190
วัน, เดือน, ปี...20 ส.ค. 2546

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมบราวเซอร์สำหรับช่วยเหลือคนพิการ

Assistive Browser for The Disable



โดย

นาย คมสัน จันทร์เจริญ

นาย พงศกร พูนพิเชฐธรรม

อาจารย์ที่ปรึกษา

อ. เกียรติภูม เจียรนัยชนะกิจ

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมควบคุมบราวเซอร์สำหรับช่วยเหลือคนพิการ

Assistive Browser for The Disable

ผู้จัดทำ

1. นาย คมสัน จันทร์เจริญ รหัสประจำตัว 41014055
2. นาย พงศกร พูนพิเชษฐธรรม รหัสประจำตัว 41014284



อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมบราวเซอร์สำหรับช่วยเหลือคนพิการ

นาย คมสัน จันท์เจริญ 41014055
 นาย พงศกร พูนพิเชษฐธรรม 41014284
 อ. เกียรติกุล เกียรตินัยชนะกิจ อาจารย์ที่ปรึกษา
 ปีการศึกษา 2544

บทคัดย่อ

เนื่องจากในยุคปัจจุบันนี้ คอมพิวเตอร์มีความสำคัญมากขึ้น และผู้พิการหรือคนทั่วไป ก็มีความจำเป็นต้องการใช้งานคอมพิวเตอร์ในการทำงานต่าง ๆ พอสมควร แต่ผู้พิการทางมือไม่สามารถใช้คอมพิวเตอร์โดยการกดแป้นพิมพ์และควบคุมเมาส์ได้สะดวกนัก ทางผู้จัดทำโครงการจึงได้มีความคิดที่จะช่วยเหลือให้ผู้พิการทางมือสามารถควบคุมการทำงานของคอมพิวเตอร์ได้ใช้งานสะดวกยิ่งขึ้น โดยมีอุปกรณ์เสริมเป็นปุ่มกด 2 ปุ่ม ซึ่งให้สามารถควบคุมการทำงานของโปรแกรมต่าง ๆ โดยผ่านบราวเซอร์ที่สร้างขึ้น ซึ่งจะสามารถส่งการทำงานไปยังระบบปฏิบัติการวินโดวส์ และแอปพลิเคชันต่างๆ บนวินโดวส์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Assistive Browser for The Disable

Mr. Khomsan Chanchaoren 41014055

Mr. Pongsakron Phoonphichaidthum 41014284

Mr. Kietkul Jeeranaitanakij Advisor

Year 2001

Abstract

Computers have become the tools people use to access information and communicate with other around the world. Nowadays computers are necessary not only for common people but also for people who have physical limitations. In this study we focus on people with physical limitations which have no both arms. They are also potential human resources in our country. Although the disable can use computers, some of them such as users with limited hand mobility cannot use a keyboard or control a mouse conveniently. Therefore, the purpose of this project is to propose the method that can help users with limited hand mobility. They can control the computer's operation more conveniently and easily by using the supportable device which has 2 buttons. With this device, users can control and manage software's operations with this browser is developed. This browser can command instructions to Windows Operating System and also many application softwares on Windows.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ สำเร็จลงด้วยดีได้ เพราะความกรุณาจากอาจารย์ต่างๆ ที่ให้การอบรมสั่งสอน วิชาความรู้ตั้งแต่ต้น จนถึงปัจจุบัน โดยเฉพาะอย่างยิ่ง อาจารย์ที่ปรึกษา อาจารย์เกียรติคุณ เจียรนัยธนะกิจ ที่คอยให้คำปรึกษา พร้อมทั้งแนวคิด ข้อเสนอแนะ ตลอดจนการแก้ไขบกพร่องต่างๆ โดยตลอด ผู้จัดทำรู้สึกซาบซึ้ง และขอกราบขอบพระคุณอย่างสูง มา ณ โอกาสนี้

ขอกราบขอบพระคุณ บิดา มารดา ที่คอยอบรมสั่งสอน รับฟังปัญหาต่างๆ และทุ่มเทกำลังกาย กำลังใจในการให้การศึกษาแก่ผู้จัดทำ และให้กำลังใจมาเป็นอย่างดีโดยตลอด ผู้จัดทำขอกราบขอบพระคุณมา ณ โอกาสนี้ด้วย

ขอบคุณถึงที่ช่วยแนะนำให้รู้จักโปรแกรมสเปย์และแนะนำหนังสือดีๆ ที่มีประโยชน์มาก
ขอบคุณภาคย์ เคง โอ๊ด พี่อ้อม และซานที่ช่วยเหลือด้านฮาร์ดแวร์
และขอบคุณเพื่อนๆทุกคนที่ไม่ได้กล่าวถึงแต่คอยช่วยเหลือและช่วยลุ้นให้งานสำเร็จลุล่วงได้ดี

นาย กมลสัน

จันทร์เจริญ

นาย พงศกร

พูนพิเชฐธรรม

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการวิจัย	1
1.3 ขอบเขตของโครงการวิจัย	2
1.4 โปรแกรมต่างๆ ที่ใช้งานในโครงการ	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีพื้นฐาน	3
2.1 Object Linking and Embedding	3
2.1.1 การเชื่อมข้อมูลแบบ Object Embedding	3
2.1.2 การเชื่อมต่อข้อมูลแบบ Object Linking	3
2.2 Windows Application Program Interface	4
2.2.1 รู้จักกับ Windows API	4
2.2.2 กลุ่มฟังก์ชันของ Windows API	5
2.2.3 เลือกฟังก์ชันมาใช้งานด้วย API Viewer	6
2.2.4 การประกาศฟังก์ชันใน Windows API	10
2.2.5 ข้อควรรู้ในการใช้ API	11
บทที่ 3 การเขียนโปรแกรมฐานข้อมูลด้วย ADO	12
3.1 ADO คืออะไร	12
3.2 เขียนโปรแกรมติดต่อฐานข้อมูล ผ่าน ADO ด้วยวิซวลเบสิก	13
บทที่ 4 การสื่อสารข้อมูลด้วยพอร์ตอนุกรม	21
4.1 การเขียนโปรแกรมควบคุมฮาร์ดแวร์ต่างๆ	21
4.1.1 พอร์ตอนุกรม (RS-232)	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
4.1.2 พอร์ตขนาน	21
4.1.3 TCP/IP	22
4.2 การสื่อสารข้อมูลแบบอนุกรม	22
4.2.1 มาตรฐาน EIA RS-232	22
4.2.2 มาตรฐาน RS-232C	23
4.2.2.1 คุณสมบัติของ RS-232C	23
4.2.2.2 การจัดหาสัญญาณของ RS-232 ทั้งแบบ 9 ขา และ 25 ขา	24
4.2.2.3 การเชื่อมต่อสัญญาณของ RS-232C	25
4.2.3 รูปแบบของข้อมูลอนุกรมและอัตราการส่งข้อมูลในการสื่อสารข้อมูลอนุกรม	26
4.2.4 การเชื่อมต่อแบบอนุกรมและ UART	26
4.2.5 MCS - 51 กับ การส่งข้อมูลแบบอนุกรม	29
4.2.6 Serial Port Control Register	30
4.2.7 Mode Of Operation	31
4.2.7.1 8-Bit Shift Register (Mode 0)	32
4.2.7.2 8-Bit UART with Variable Baud Rate (Mode 1)	32
4.2.7.3 9-Bit UART with Fixed Baud Rate (Mode 2)	33
4.2.7.4 9-Bit UART with Variable Baud Rate (Mode 3)	33
4.2.8 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล	33
4.2.9 อัตราการส่งข้อมูลทางพอร์ตอนุกรม	34
4.2.10 การเชื่อมต่อหน่วยความจำกับ MCS-51	37
4.2.10.1 Data Memory	37
4.2.10.2 Program Memory	37
4.2.10.3 การเชื่อมต่อหน่วยความจำกับ MCS-51	37
4.2.11 การถอดรหัสตำแหน่งของหน่วยความจำ	39
บทที่ 5 การเขียนโปรแกรมติดต่อพอร์ตอนุกรมด้วยวิซวลเบสิก	41
5.1 ควบคุมและติดต่อผ่านทางพอร์ตอนุกรมด้วย Com Port	41
5.2 การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรม	43
5.2.1 การติดต่อแบบอินเทอร์รัพต์	43
5.2.2 การติดต่อแบบ โพลลิง	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
5.3 องค์ประกอบในการใช้ MSCComm	43
5.3.1 การตั้งค่าติดต่อกับพอร์ต	43
5.3.2 การใช้ Buffer ในการรับส่งข้อมูล	43
5.3.3 ด้านฮาร์ดแวร์	44
5.4 การกำหนดคุณสมบัติของ MSCComm Control ให้สามารถติดต่อกับพอร์ตได้	44
5.5 วิธีของการรับส่งข้อมูลจากพอร์ตอนุกรม	45
บทที่ 6 แนวความคิดและการออกแบบ	50
6.1 Sequence Diagram และ Collaboration Diagram	50
6.1.1 แสดง Sequence Diagram ของสถานะของการกดปุ่ม	50
6.1.2 แสดง Collaboration Diagram ของสถานะของการกดปุ่ม	51
6.1.3 แสดง Sequence Diagram ของสถานะของการตั้งค่าต่าง ๆ	51
6.1.4 แสดง Collaboration Diagram ของสถานะของการตั้งค่าต่าง ๆ	52
6.2 Activity Diagram	52
6.2.1 แสดง Activity Diagram ของการทำงานของการทำงานหลัก	52
6.2.2 แสดง Activity Diagram ของการเก็บข้อมูลลงฐานข้อมูล	53
6.2.3 แสดง Activity Diagram ของการนำข้อมูลจากฐานข้อมูลมาแสดงในส่วน Predict Field	53
6.2.4 แสดง Activity Diagram ในการเปลี่ยนการตั้งค่า	54
6.3 แสดงการออกแบบวงจรของปุ่ม 2 ปุ่ม	54
บทที่ 7 บทสรุปและวิจารณ์	56
7.1 ปัญหาและอุปสรรคในการทำงาน	56
7.2 สรุปผลที่ได้จากการทำโครงการ	56
7.3 ข้อเสนอแนะ	56
7.4 แนวทางการพัฒนาต่อ	56
7.5 ผลการทดสอบ โปรแกรม	57
ภาคผนวก ก ตารางแอสกี	58
ภาคผนวก ข หน้าที่ของโปรซีเจอร์ และฟังก์ชันต่างๆ ในโปรแกรม	60
บรรณานุกรม	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้าที่
รูปที่ 2-1 แสดงหลักการทำงานของ Windows API	5
รูปที่ 2-2 แสดงการเพิ่มเติม API Viewer ลงในเมนู	6
รูปที่ 2-3 แสดงการเพิ่มเติมเมนูไอเท็มเข้าไปใหม่	7
รูปที่ 2-4 แสดงการเรียกใช้เท็กซ์ไฟล์	7
รูปที่ 2-5 แสดงการเลือกเท็กซ์ไฟล์ที่ต้องการ	8
รูปที่ 2-6 แสดงการเลือกชนิดของการดู	8
รูปที่ 2-7 แสดงการเลือกฟังก์ชันที่สนใจ	9
รูปที่ 2-8 แสดงการประกาศฟังก์ชันที่ถูกเลือก	9
รูปที่ 2-9 แสดงตัวอย่างการเลือกใช้ค่าคงที่ของ API Viewer	10
รูปที่ 3-1 แสดงการติดต่อกับฐานข้อมูลของ ADO	13
รูปที่ 3-2 แสดงรูปการเลือก ADODC Properties	14
รูปที่ 3-3 แสดงการเลือก Use Connection String ในหน้าของ Property Pages	14
รูปที่ 3-4 แสดงการเลือก Provider ในหน้าของ Data Link Properties	15
รูปที่ 3-5 แสดงการเลือก Connection ในหน้าของ Data Link Properties	15
รูปที่ 3-6 แสดงการเซตค่า RecordSource ใน Property Pages	16
รูปที่ 3-7 แสดงการเลือก Data Source	17
รูปที่ 3-8 การเซตค่าหน้า ODBC Microsoft Access Setup	17
รูปที่ 3-9 แสดงหน้าต่างการเลือก Select Database	18
รูปที่ 3-10 แสดงหน้าต่าง Property Pages	18
รูปที่ 3-11 แสดงรูปการเลือก Microsoft Data Link	19
รูปที่ 3-12 แสดงการเลือก Provider ในหน้าของ Data Link Properties	19
รูปที่ 3-13 แสดงการเซตค่า Connection ในหน้า Data Link Properties	20
รูปที่ 4-1 แสดงรายละเอียดของขาสัญญาณของ RS-232 แบบ 9 ขา และ 25 ขา	24
รูปที่ 4-2 แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ อุปกรณ์ DCE	25
รูปที่ 4-3 แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ อุปกรณ์ DTE	25
รูปที่ 4-4 แสดงรูปแบบของสัญญาณข้อมูลอนุกรมที่ใช้ในการสื่อสารข้อมูลอนุกรม	26
รูปที่ 4-5 แสดงการแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม	27
รูปที่ 4-6 แสดงเวิร์ดข้อมูลขนาด 8 บิตกับ บิตเริ่มต้นและบิตหยุด ที่ใช้ในการ โอนย้ายข้อมูลแบบอนุกรม	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้าที่
รูปที่ 4-7 แสดงการรับส่งข้อมูลระหว่างรีจิสเตอร์กับบัสภายใน	30
รูปที่ 4-8 แสดงการส่งข้อมูลออกโดยใช้รีพรีจิสเตอร์ช่วย	32
รูปที่ 4-9 แสดงการกำหนดอัตรากาการส่งข้อมูลในโหมดต่าง ๆ	35
รูปที่ 4-10 แสดงสัญญาณของ MCS-51 ที่ใช้ติดต่อกับตำแหน่งของหน่วยความจำ	38
รูปที่ 4-11 แสดงการต่อ MCS-51 กับหน่วยความจำ	38
รูปที่ 4-12 แสดงการจัดวางขาของ 74LS138	39
รูปที่ 5-1 แสดง ADD Components ให้โปรแกรมวิซวลเบสิก	41
รูปที่ 5-2 แสดงการ ADD Component ที่ชื่อ Microsoft Comm Control 6.0	42
รูปที่ 5-3 แสดงการ ADD Control Component ลง Form	42
รูปที่ 6-1 แสดง Sequence Diagram สถานะของการกดปุ่ม	50
รูปที่ 6-2 แสดง Collaboration Diagram สถานะของการกดปุ่ม	51
รูปที่ 6-3 แสดง Sequence Diagram สถานะของการตั้งค่าต่างๆ	51
รูปที่ 6-4 แสดง Collaboration Diagram สถานะของการตั้งค่าต่าง ๆ	52
รูปที่ 6-5 แสดง Activity Diagram การทำงานของโปรแกรมหลัก	52
รูปที่ 6-6 แสดง Activity Diagram ของการเก็บข้อมูลลงฐานข้อมูล	53
รูปที่ 6-7 แสดง Activity Diagram ของการนำข้อมูลจากฐานข้อมูลมาแสดงในส่วน Predict Field	53
รูปที่ 6-8 แสดง Activity Diagram ในการเปลี่ยนการตั้งค่า	54
รูปที่ 6-9 แสดงการต่อวงจรปุ่ม 2 ปุ่ม	55

สารบัญตาราง

	หน้าที่
ตารางที่ 2-1 แสดงตารางเกี่ยวกับไฟล์ DLL ที่สำคัญ	5
ตารางที่ 4-1 แสดงข้อมูลการเชื่อมต่ออนุกรม 9 ขา	23
ตารางที่ 4-2 อัตราการส่งข้อมูลและช่วงเวลาของแต่ละบิตข้อมูลในการสื่อสารข้อมูลอนุกรม	26
ตารางที่ 4-3 แสดงอัตราการส่งข้อมูลทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม	28
ตารางที่ 4-4 แสดงบิตต่างๆ ของรีจิสเตอร์	30
ตารางที่ 4-4 (ต่อ) แสดงบิตต่างๆ ของรีจิสเตอร์	31
ตารางที่ 4-5 แสดงโหมดต่างๆของการรับส่งแบบอนุกรม	31
ตารางที่ 4.6 แสดงความถี่สัญญาณนาฬิกาที่ใช้กำหนดอัตราการส่งข้อมูลของค่าต่างๆ	36
ตารางที่ 4-7 แสดงรหัสตำแหน่งหน่วยความจำขนาด 4 กิโลไบต์	39
ตารางที่ 4-8 แสดงค่าความจริงของ 74LS138	40
ตารางที่ 5-1 แสดง CommEvent เกี่ยวกับการเกิดสถานะเมื่อเกิดการผิดพลาดในการสื่อสาร	47
ตารางที่ 5-2 แสดง CommEvent เกี่ยวกับเกิดสถานะการเมื่อมีการสื่อสาร	47
ตารางที่ 5-3 แสดง MSComm Control Property	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในยุคปัจจุบันนี้ คอมพิวเตอร์มีความสำคัญมากพอสมควร และผู้พิการหรือคนทั่วไป ก็มีความจำเป็นที่ต้องการใช้งานคอมพิวเตอร์ในการทำงานต่างๆ พอสมควร เนื่องจากผู้พิการทางมือไม่สามารถใช้คอมพิวเตอร์โดยการกดแป้นพิมพ์ และควบคุมใช้เมาส์ได้สะดวกนัก ทางผู้จัดทำโครงการจึงได้มีความคิดที่จะทำอย่างไร ให้ผู้พิการทางมือควบคุมการทำงานของคอมพิวเตอร์ เพื่อที่จะใช้งานได้สะดวกยิ่งขึ้น โดยมีอุปกรณ์เสริมเป็นปุ่มกด 2 ปุ่ม หรือสามารถกำหนดปุ่มบนแป้นพิมพ์ 2 ปุ่ม เพื่อให้สามารถควบคุมการทำงานของโปรแกรมต่างๆ ได้ง่ายขึ้น

คนพิการทางมือในโครงการครั้งนี้จะระบุถึงผู้พิการทางมือที่ไม่มีแขนเลยแม้แต่ข้างเดียว แต่ยังมีเท้าเพื่อสามารถควบคุมปุ่ม 2 ปุ่มนี้ได้ สะดวกกว่าการใช้เท้าควบคุมแป้นพิมพ์หรือเมาส์ และการควบคุมนี้สามารถควบคุมโดยการนำเมาส์ไปวางไว้ที่ปุ่มใดๆ บนแป้นพิมพ์เสมือน หรือปุ่มใดๆ บนแอปพลิเคชันต่างๆ บนวินโดวส์ ก็จะสามารถคลิกได้เองในช่วงเวลากำหนด การคลิกนี้รวมถึงการคลิกลิงค์ต่างๆ บนอินเทอร์เน็ตด้วย และในตัวผู้ใช้งานสามารถเพิ่มคำศัพท์ลงฐานข้อมูลเองได้ เพื่อให้การพิมพ์ข้อมูล สะดวกรวดเร็วยิ่งขึ้นอีกทางหนึ่งด้วย

นอกจากนี้ ในกรณีที่ผู้พิการบางคนยังสามารถใช้งานเมาส์ได้ แต่ก็ยังสามารถใช้เมาส์ควบคุมโปรแกรมที่ได้โดยจะใช้โปรแกรมได้คล่องตัวมากขึ้น เพียงแค่นำเคอร์เซอร์ของเมาส์ไปวางไว้บนแป้นพิมพ์เสมือนก็จะมีการส่งคำสั่งไปที่โปรแกรมได้ หรือนำไปวางที่ปุ่มใดๆ ก็จะมีการคลิกในเวลาที่กำหนดด้วย ทำให้ไม่ต้องคลิก เนื่องจากบางทีผู้ใช้เหล่านั้นจะไม่สะดวกในการกดปุ่มคลิกบนเมาส์

และในปัจจุบันผู้พิการก็เป็นบุคลากรที่สำคัญต่อประเทศชาติอีกทางหนึ่ง ผู้จัดทำโครงการจึงมีความคิดที่จะทำให้ผู้พิการเหล่านี้สามารถใช้ศักยภาพของคอมพิวเตอร์ได้มากขึ้น

1.2 วัตถุประสงค์ของโครงการวิจัย

1. เพื่อศึกษาการเขียน โปรแกรมด้วยภาษาวิซวลเบสิก
2. เพื่อศึกษาการเขียน โปรแกรมที่ทำงานบนระบบปฏิบัติการวินโดวส์
3. เพื่อศึกษาการเขียน โปรแกรมที่ใช้ในการติดต่อกับฟังก์ชันของระบบปฏิบัติการวินโดวส์
4. เพื่อที่จะให้ผู้พิการทางมือใช้โปรแกรมต่างๆ บนวินโดวส์ได้สะดวกยิ่งขึ้น
5. เพื่อศึกษาการทำงานของพอร์ตอนุกรม (RS-232)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงการวิจัย

ปริญญานิพนธ์นี้จะกล่าวถึง การทำงานของแป้นพิมพ์เสมือนเพื่อคนพิการทางมือ โดยการควบคุมด้วยปุ่มเพียง 2 ปุ่ม โดยมีการทำงานหรือข้อกำหนดดังนี้

1. สามารถให้ใช้กับโปรแกรมต่างๆ ดังต่อไปนี้ Microsoft Office ,Internet Explorer ,MSN Messenger ,Notepad ,Edit Plus ได้
2. ควบคุมการทำงานของโปรแกรมต่างๆ ข้างต้น ด้วยปุ่มเพียง 2 ปุ่ม
3. สามารถควบคุมการทำงานของแป้นพิมพ์เสมือนให้สามารถใช้งานได้ทั้งภาษาไทยและภาษาอังกฤษ
4. สามารถที่จะใช้ฟังก์ชันของเมาส์ควบคุมโปรแกรมต่างๆ อาทิ คลิก(Click) ดับเบิลคลิก(Double Click) และลากแล้วปล่อย(Drag and Drop) หลังจากที่นำเมาส์ไปวางไว้บน(Mouse Over) วัตถุต่างๆ
5. มีการทำนายคำที่เกิดขึ้นจากฐานข้อมูลที่รวบรวมคำศัพท์ภาษาไทยและภาษาอังกฤษซึ่งผู้ใช้สามารถเพิ่มคำได้ เพื่อลดเวลาในการพิมพ์ และสามารถใช้งานได้รวดเร็วยิ่งขึ้น

1.4 โปรแกรมต่างๆ ที่ใช้งานในโครงการ

- Microsoft Visual Basic 6.0 (ใช้ในการเขียนโปรแกรม)
- Windows 98/Me /NT /2000 (ระบบปฏิบัติการสำหรับ Run โปรแกรม)
- Microsoft Access 2000 (ไว้เป็นฐานข้อมูล สำหรับเก็บข้อมูลที่เคยพิมพ์)
- Microsoft Office 97/2000 (สำหรับทดสอบการส่งข้อมูล และเข้าถึงข้อมูล)
- Internet Explorer (สำหรับทดสอบการส่งข้อมูล และเข้าถึงข้อมูล)
- Edit Plus (สำหรับทดสอบการส่งข้อมูล และเข้าถึงข้อมูล)
- Notepad (สำหรับทดสอบการส่งข้อมูล และเข้าถึงข้อมูล)
- Calculator (สำหรับทดสอบการส่งข้อมูล และเข้าถึงข้อมูล)

หมายเหตุ โปรแกรมนี้ยังสามารถใช้ร่วมกับโปรแกรมอื่นๆ ได้อีกหลายโปรแกรม โปรแกรมเหล่านี้ใช้สำหรับเพียงทดสอบเท่านั้น

1.5 วิธีการดำเนินงาน

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลักๆ อยู่ 4 เรื่องด้วยกัน คือ ทฤษฎีเกี่ยวกับ Windows API ของโปรแกรมวิซวลเบสิก, การเขียนโปรแกรมติดต่อกับฐานข้อมูลด้วย ADO, การทำงานของพอร์ตอนุกรม และการเขียนโปรแกรมเชื่อมต่อกับพอร์ตอนุกรมด้วยโปรแกรมวิซวลเบสิก ซึ่งจะมีรายละเอียดอยู่ในบทที่ 2, 3, 4 และ 5 ตามลำดับ หลังจากนั้นจะนำเอาความรู้ที่เรียบเรียงมา มาสู่ขั้นตอนพัฒนาโปรแกรม โดยจะกล่าวถึงองค์ประกอบที่สำคัญของระบบที่พัฒนาขึ้นมา และยังอธิบายไปถึงรูปแบบกับการติดต่อกับผู้ใช้ และการประมวลผลเบื้องต้น รวมถึงการทดสอบโปรแกรมที่สร้างขึ้นกับระบบปฏิบัติการต่างๆ เนื้อหาเหล่านี้จะอยู่ในบทที่ 6 และในบทที่ 7 จะเป็นเนื้อหาในส่วนสรุปของโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐาน

ภาษาวิซวลเบสิก (Visual Basic) เป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมนำมาใช้พัฒนาโปรแกรมบน วินโดวส์ เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize ซึ่งเพียงแค่เลือกคอนโทรลที่เหมาะสม แล้ววางลงบนฟอร์ม ก็สามารถสร้างจอภาพที่ใช้สำหรับติดต่อกับผู้ใช้ รวมทั้งการใช้เทคโนโลยี การเขียนโปรแกรมแบบ Event – Driven ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับคอนโทรลต่างๆ ที่สร้างขึ้นตามเหตุการณ์ (Event) ต่างๆ ที่เกิดขึ้น เช่น การเลื่อนเมาส์ หรือการรับข้อมูลจากแป้นพิมพ์ ฯลฯ เป็นต้น ประกอบกับภาษาที่ใช้เขียนโปรแกรมเป็นภาษาเบสิก ซึ่งเป็นภาษาคอมพิวเตอร์ จึงส่งผลให้ การพัฒนาโปรแกรมบนวินโดวส์ด้วย วิซวลเบสิกกระทำได้ง่ายและสะดวกต่อการใช้งาน วิซวลเบสิกนี้เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมขึ้นเพื่อใช้งาน ที่ใช้ได้ตั้งแต่ผู้ใช้ระดับต้น เพื่อใช้สร้างโปรแกรมง่ายๆ บนวินโดวส์ หรือโปรแกรมเมอร์ระดับกลาง ที่จะใช้เรียกฟังก์ชันการทำงานต่างๆ ของวิซวลเบสิก ได้อย่างมีประสิทธิภาพ ตลอดจนโปรแกรมเมอร์ในระดับมืออาชีพ ที่จะพัฒนาโปรแกรมในระดับสูง โดยการใช้ Object Linking and Embedding (OLE) และ Application Programming Interface(API) ของวินโดวส์ มาประกอบในการเขียนโปรแกรม

2.1 Object Linking and Embedding

Object Linking and Embedding (OLE) เป็นวิธีที่ใช้เชื่อมโยงข้อมูลของโปรแกรมที่ต่างกันที่ทำงานอยู่บนวินโดวส์ เข้าด้วยกัน ด้วยแนวความคิดของ OLE นี้ เอกสารหรือข้อมูลที่สร้างขึ้นด้วยโปรแกรมหนึ่ง สามารถที่จะจัดเก็บเอกสารหรือข้อมูลหลายประเภทเข้าด้วยกัน เช่น เอกสารที่สร้างขึ้นด้วย Microsoft Word นอกจากจะประกอบด้วยข้อความต่างๆ เท่านั้น ยังสามารถที่จะจัดเก็บข้อมูลในรูปแบบอื่น เช่น Spreadsheet รูปภาพ Bitmap ฯลฯ ได้อีกด้วย สำหรับเอกสารหรือข้อมูลในแนวความคิดของ OLE นี้ จะเรียกว่า “OLE Document” หรือ “Compound Document”

ในการเชื่อมโยงข้อมูลของโปรแกรมที่ต่างกันเข้าด้วยกัน ในแนวความคิดของ OLE มีอยู่ด้วยกัน 2 ลักษณะดังนี้

2.1.1 การเชื่อมข้อมูลแบบ Object Embedding

เป็นการสร้าง OLE Document ขึ้น โดยนำเอาเอกสารหรือข้อมูลที่สร้างขึ้นจากโปรแกรมอื่น มาเก็บไว้ในตัว OLE Document นั้นโดยตรง

2.1.2 การเชื่อมต่อข้อมูลแบบ Object Linking

เป็นการสร้าง OLE Document ขึ้นเช่นเดียวกันแต่แตกต่างจากการเชื่อมโยงข้อมูลแบบ Object Embedding กล่าวคือ เอกสารหรือข้อมูลที่สร้างขึ้นจากโปรแกรมอื่นที่ปรากฏอยู่ใน OLE Document ที่ใช้การเชื่อมโยงข้อมูลแบบ Object Linking นี้ จะไม่ปรากฏตัวอยู่ใน OLE Document นั้นโดยตรง แต่จะเก็บเพียง Path ที่ใช้อ้างอิงไปยังเอกสารหรือข้อมูลนั้นแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเชื่อมโยงข้อมูลในแบบ OLE จะต้องประกอบไปด้วยโปรแกรมที่ทำงานร่วมกันอยู่ด้วยกัน 2 ฟังก์ชันด้วยกัน ดังนี้

- **Container Application** ได้แก่โปรแกรมที่ผู้สร้าง OLE Document ซึ่งเป็นฟังก์ชันที่เรียกเอกสารหรือข้อมูลของโปรแกรมอื่นมาทำงาน โดย OLE Document ที่สร้างขึ้น อาจเรียกใช้เอกสารหรือข้อมูลของโปรแกรมอื่นในรูปแบบของ Object Embedding หรือ Object Linking
- **Server Application** ได้แก่ โปรแกรมที่ผู้สร้างเอกสารหรือข้อมูลที่จะถูกเรียกใช้โดย Container Application โปรแกรมที่เป็น Server Application นี้ จะต้องสนับสนุนการทำงานในลักษณะลากแล้วปล่อย (Drag and Drop) การนำข้อมูลไปยังคลิปบอร์ด เพื่อที่สามารถจะนำเอาเอกสารหรือข้อมูลที่สร้างขึ้นด้วย Server Application ไปใช้งานในรูปแบบ Object Embedding และ Object Linking ได้

Container Application และ Server Application นี้ จะไม่ได้ติดต่อกันโดยตรง แต่จะติดต่อผ่าน OLE ในรูปแบบของไฟล์ DLL ซึ่งเป็นไฟล์ที่จัดเก็บในวิธีการต่างๆ ที่ Container Application และ Server Application ใช้ในการติดต่อซึ่งกันและกัน และด้วยวิธีนี้ โปรแกรมทางฝั่ง Container Application จึงไม่จำเป็นต้องทราบถึงรายละเอียดของโปรแกรมทางฝั่ง Server Application แต่อย่างใด

2.2 Windows Application Program Interface

แม้ว่าวิซวลเบสิก จะเตรียมฟังก์ชันและความสามารถต่างๆ ไว้อย่างมากมาย แต่อาจไม่ตอบสนองความต้องการทั้งหมด ดังนั้นวิซวลเบสิกจึงยอมให้ผู้พัฒนาแอปพลิเคชัน สามารถเรียกใช้ความสามารถของระบบปฏิบัติการ Windows 95/98 ,NT Workstation ได้อย่างเต็มที่ โดยการเรียกผ่านฟังก์ชัน Windows API

ในหัวข้อนี้เราจะเรียนรู้วิธีการเรียกใช้ฟังก์ชัน API พร้อมทั้งบอกถึงฟังก์ชันที่ถูกนำมาใช้งาน

2.2.1 รู้จักกับ Windows API

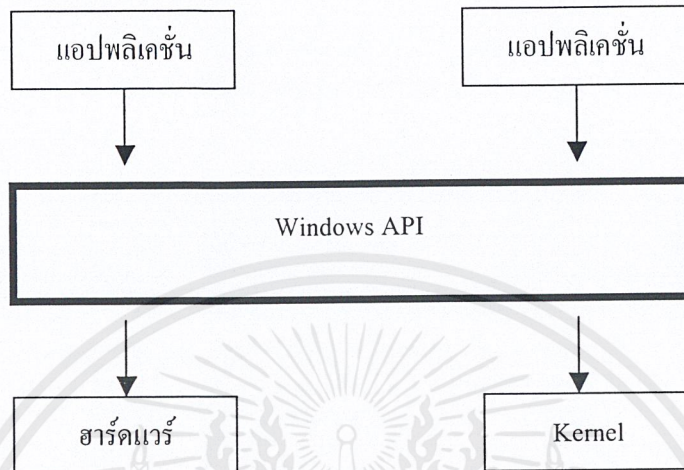
Windows API ย่อมาจาก Windows Application Program Interface ซึ่งมันก็คือ ชุดของฟังก์ชันที่พร้อมให้เราเรียกใช้งาน โดยที่ฟังก์ชันเหล่านั้นรับประกันได้ว่าทำงานอย่างมีประสิทธิภาพ เช่น เราไม่จำเป็นต้องเขียนโปรแกรมให้ยืดยาวเพื่อตรวจสอบว่าเนื้อที่ฮาร์ดดิสก์เหลือว่างอยู่เท่าไร แต่เรียกใช้ฟังก์ชัน GetDiskFreeSpace ฟังก์ชันเดียวก็รู้แล้ว เป็นต้น

Windows API นั้น จึงมีฟังก์ชันอยู่มากมายนับพันๆ ฟังก์ชัน ถูกแบ่งออกเป็นหมวดหมู่ตามประเภทของงาน เช่น API ด้านกราฟฟิก , API ด้านเน็ตเวิร์ค เป็นต้น

สำหรับสาเหตุที่ต้องมี Windows API ก็เพราะว่า เราจะได้มีวิธีการมาตรฐานในการใช้งาน วินโดวส์แบบเดียว ซึ่งจะช่วยลดเวลาของเราที่ต้องไปศึกษาอุปกรณ์ และองค์ประกอบต่างๆ ทำให้เราไม่จำเป็นต้องรู้ว่าจอภาพทำงานอย่างไร , เป็นพิมพ์ทำงานอย่างไร , เมาส์ทำงานอย่างไร แต่สามารถเรียกใช้งานผ่าน Windows API ได้ทันที

อีกหนึ่งข้อดีของ Windows API ที่น่าสนใจก็คือ ช่วยลดความผิดพลาดจากการเข้าถึงฮาร์ดแวร์โดยตรง เพราะเราจะสั่งงานฮาร์ดแวร์ผ่าน Windows API เท่านั้น ซึ่งช่วยลดความสับสนในการใช้งานฮาร์ดแวร์ที่ผิดวิธีไปได้มาก

นอกจากนี้ยังมีข้อดีของ Windows API ได้แก่ ความเป็นมาตรฐานที่ทุกคนรู้จัก จึงง่ายต่อการแก้ไขเปลี่ยนแปลง เช่น แม้ว่าเราจะเปลี่ยนไปใช้ฮาร์ดแวร์ใหม่ หรือแม้แต่วาระบบปฏิบัติการเวอร์ชันใหม่ๆ แล้วก็ตาม ชื่อของฟังก์ชันเหล่านี้ ยังคงใช้ชื่อเดิมทำงานได้เหมือนเดิม ทำให้เราไม่ต้องกังวลกับความเปลี่ยนแปลงที่จะเกิดขึ้นจนบ่อยเกินไปนัก



รูปที่ 2-1 แสดงหลักการการทำงานของ Windows API

2.2.2 กลุ่มฟังก์ชันของ Windows API

Windows 95/98, NY Workstations จะมีฟังก์ชันในรูปของ Windows API ให้ใช้มากกว่า 1000 ฟังก์ชัน ซึ่งฟังก์ชันต่างๆ จะถูกเก็บในไฟล์ .DLL ซึ่งอยู่ในโฟลเดอร์ C:\Windows\System32 ซึ่งมีไฟล์สำคัญๆ ได้แก่

ชื่อไฟล์	คำอธิบาย
ADVAPI32.DLL	มาจาก Advanced API ซึ่งจะเก็บฟังก์ชันเกี่ยวกับความปลอดภัย และ Registry
COMDLG.DLL	เก็บฟังก์ชันของ Common Dialog
GDI32.DLL	GDI ซึ่งจะเก็บฟังก์ชันเกี่ยวกับการแสดงผล และฟังก์ชันเกี่ยวกับกราฟฟิก
KERNEL32.DLL	เก็บฟังก์ชันที่ใช้จัดการทรัพยากรของระบบปฏิบัติการ
LZ32.DLL	เก็บฟังก์ชันที่ใช้เกี่ยวกับการบีบอัดข้อมูล
MPR.DLL	เก็บฟังก์ชันที่เกี่ยวกับ Multiple Provide Router
NETAPI32.DLL	เก็บฟังก์ชันที่เกี่ยวกับการจัดการเครือข่าย (Network)
SHELL32.DLL	เก็บฟังก์ชันที่เกี่ยวกับเชลล์ (Windows Explorer)
USER32.DLL	เก็บฟังก์ชันที่เกี่ยวกับการติดต่อกับผู้ใช้งาน (User Interface)
VERSION32.DLL	เก็บฟังก์ชันที่เกี่ยวกับการจัดการรุ่นของซอฟต์แวร์
WINMM.DLL	เก็บฟังก์ชันที่เกี่ยวกับมัลติมีเดีย
WINSPOOL.DRV	เก็บฟังก์ชันที่เกี่ยวกับการพิมพ์ผ่านเครื่องพิมพ์

ตารางที่ 2-1 แสดงตารางเกี่ยวกับไฟล์ DLL ที่สำคัญ

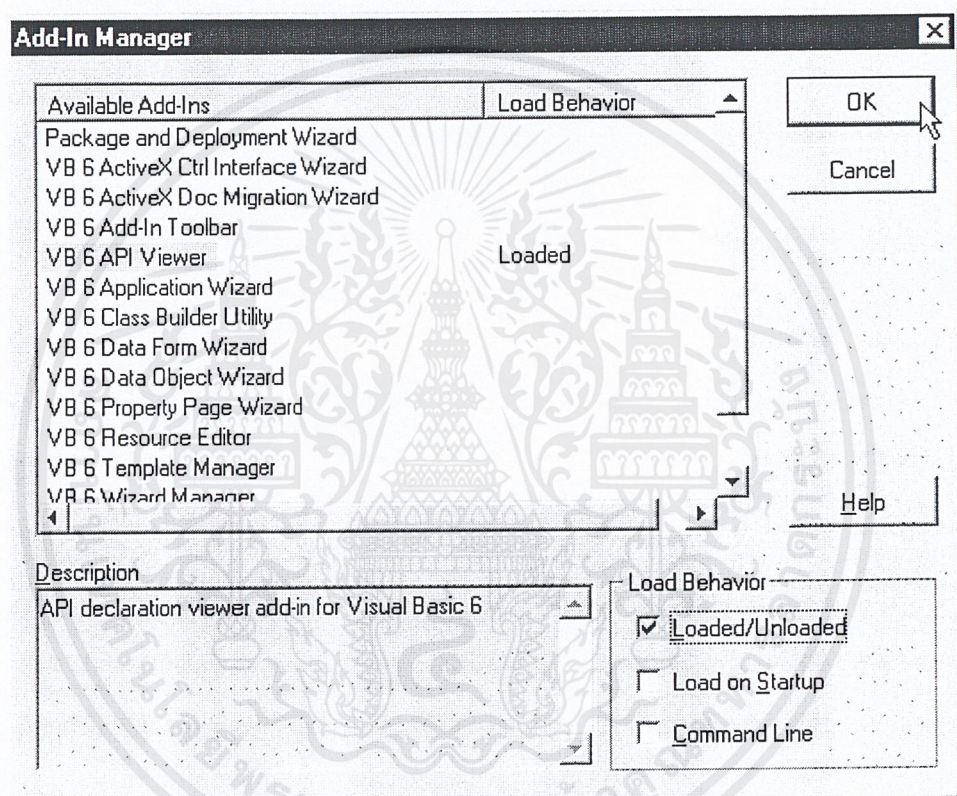
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 เลือกฟังก์ชันมาใช้งานด้วย API Viewer

เพราะว่ามีฟังก์ชันให้เลือกใช้เป็นจำนวนมาก วิศวกรเบสิกจึงเตรียมเครื่องมือสำหรับเรียกใช้งานฟังก์ชันเหล่านั้น ซึ่งมีชื่อว่า API Viewer

ขั้นตอนการใช้งาน API Viewer

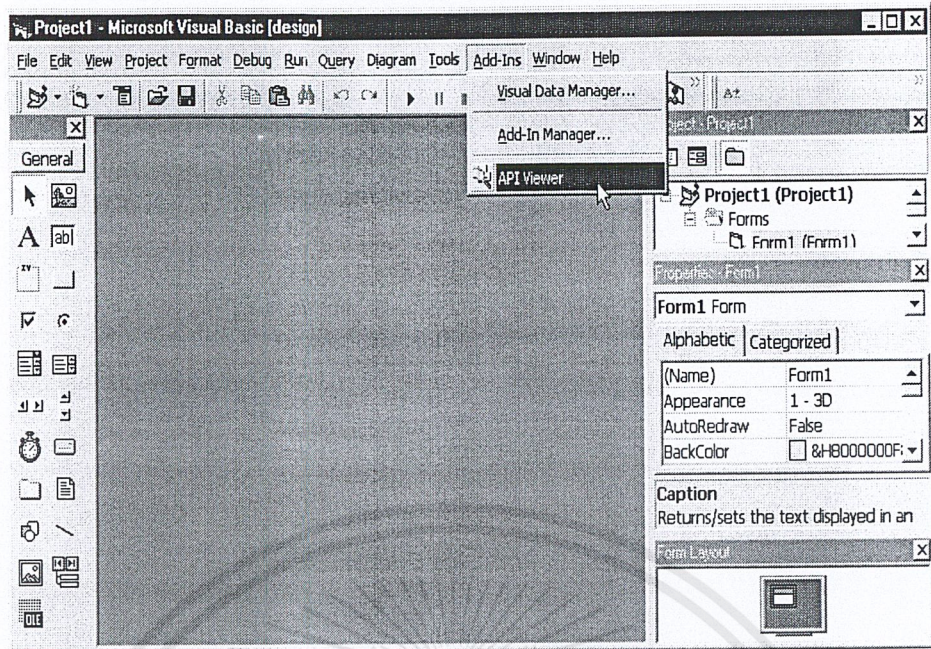
1. เริ่มจากเลือกเมนู Add-Ins > Add In Manager ... จะปรากฏไดอะล็อกบ็อกซ์ Add-In Manager
2. คลิกเลือกที่ VB API Viewer
3. เลือก CheckBox ในกรอบ Load Behavior ตามลักษณะการโหลด API Viewer ที่เราต้องการ จากนั้นคลิก OK



รูปที่ 2-2 แสดงการเพิ่มเติม API Viewer ลงในเมนู

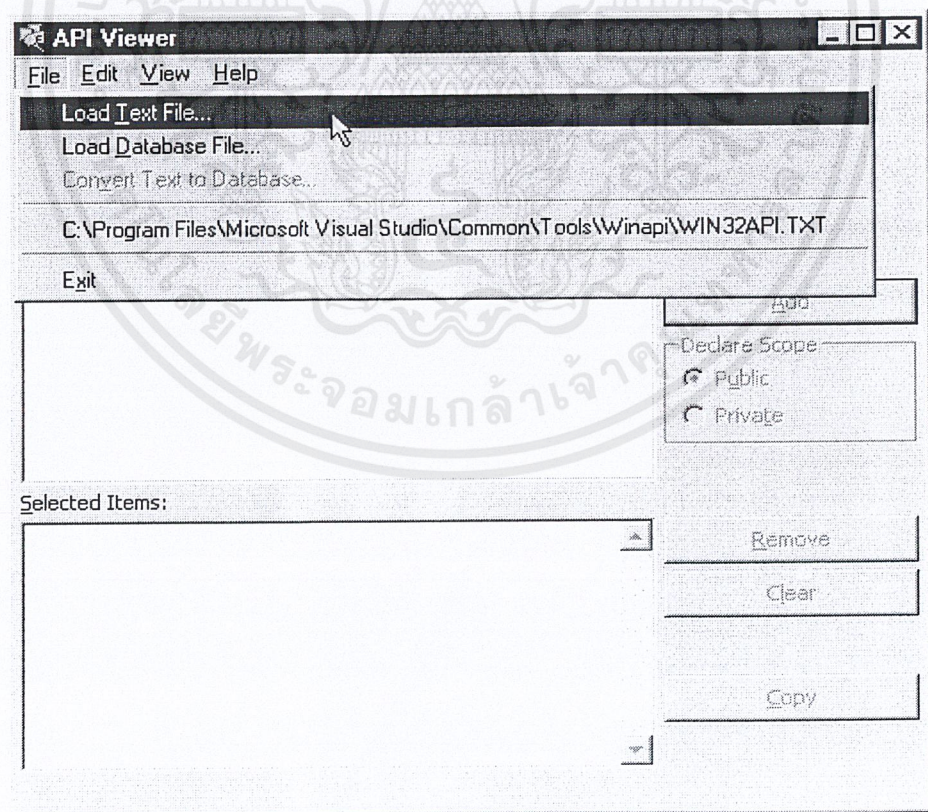
4. ภายใต้เมนู Add-Ins จะปรากฏเมนูไอ้เพิ่ม API Viewer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 แสดงการเพิ่มเติมเมนูไอเท็มเข้าไปใหม่

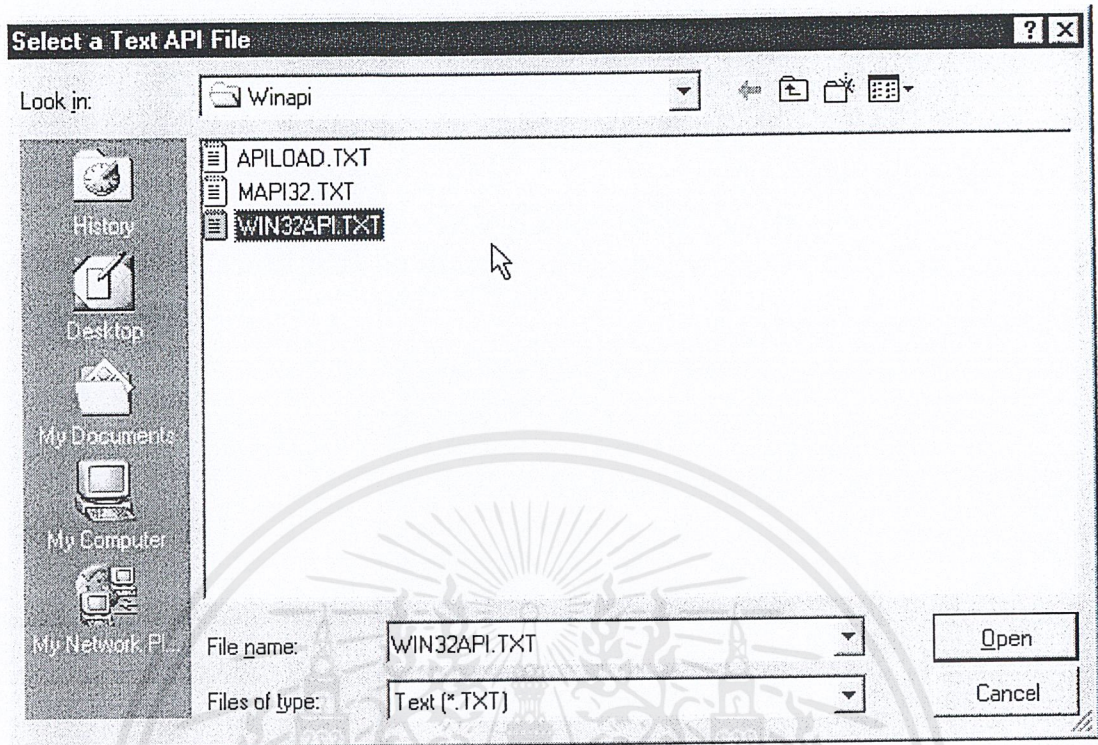
5. เมื่อเราเลือกเมนู Add-Ins > API Viewer ก็จะมีปรากฏหน้าต่าง API Viewer
6. ให้เลือกเมนู File > Load Text File เพื่อโหลดเท็กซ์ไฟล์ ซึ่งเกี่ยวข้องกับฟังก์ชันของ Windows API ขึ้นมาใช้งาน



รูปที่ 2-4 แสดงการเรียกใช้เท็กซ์ไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

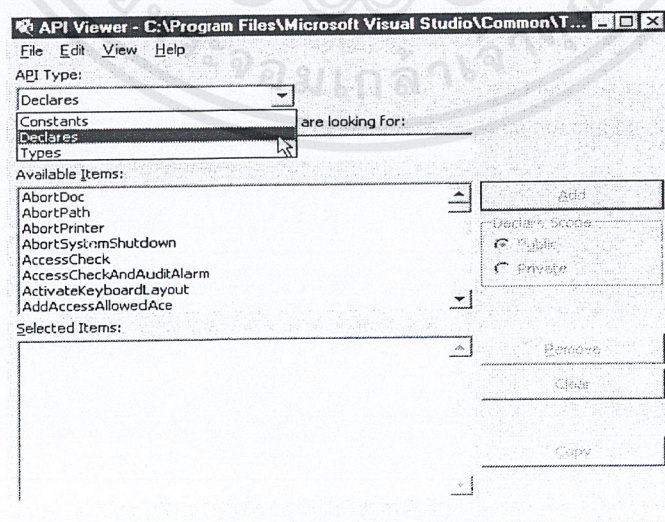
7. ให้เราเลือกไฟล์ Win32api.txt ขึ้นมาใช้งาน



รูปที่ 2-5 แสดงการเลือกเท็กซ์ไฟล์ที่ต้องการ

8. ในช่อง API Type นั้น เราสามารถ เลือกได้ว่าต้องการดูเกี่ยวกับอะไร ซึ่งประกอบไปด้วย

- การประกาศค่า (Declares)
- ค่าคงที่ (Constants)
- ชนิดข้อมูลที่ได้มีการประกาศไว้ (Types)

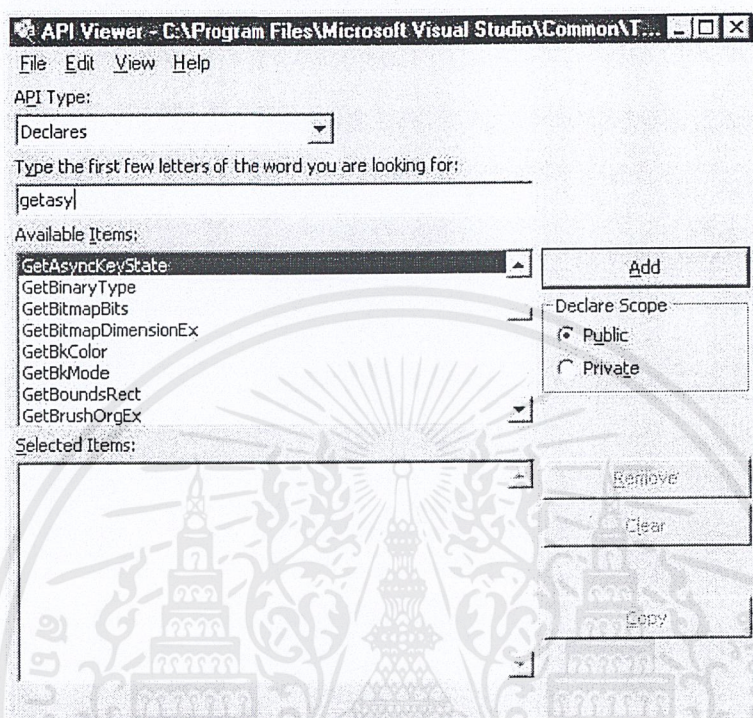


รูปที่ 2-6 แสดงการเลือกชนิดของการดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

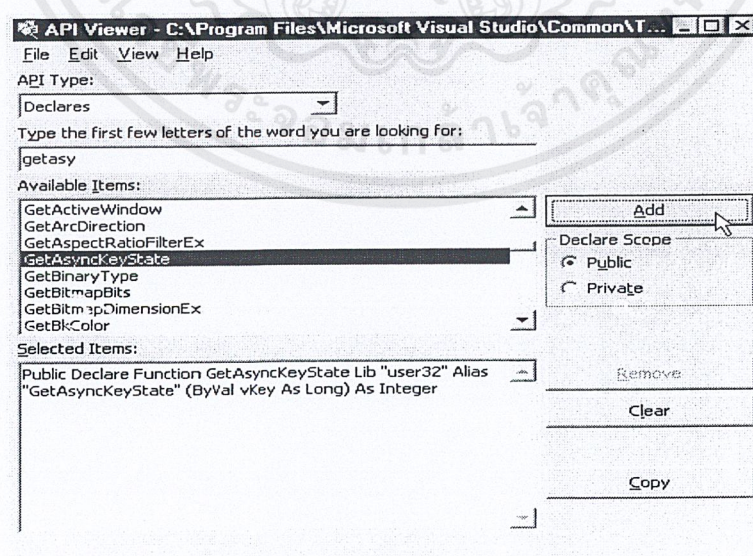
9. กรณีที่ต้องการดูการประกาศฟังก์ชัน เราเริ่มได้จากการพิมพ์ชื่อฟังก์ชันที่ต้องการในช่อง Type the first few letters of the word you are looking for : ซึ่งเราเพียงแต่พิมพ์ไม่กี่คำในช่องนี้ ที่ช่อง Available Items : ก็จะเลือกชื่อฟังก์ชันที่ใกล้เคียงมาให้เราเลือก

10. คลิกเลือกฟังก์ชันที่ต้องการ แล้วคลิกปุ่ม ADD



รูปที่ 2-7 แสดงการเลือกฟังก์ชันที่สนใจ

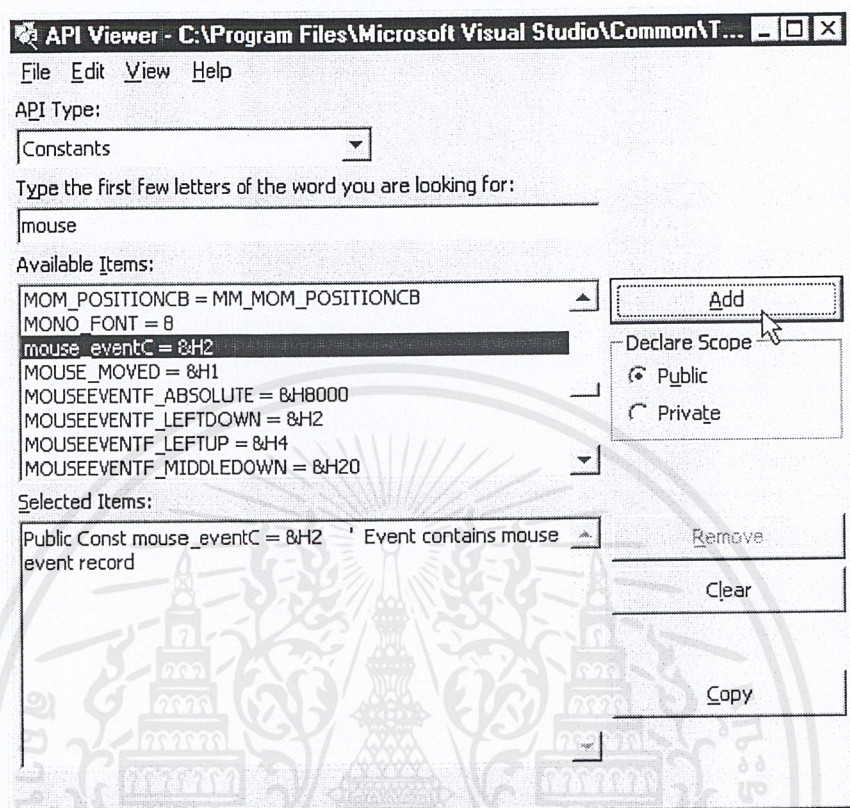
11. ฟังก์ชันที่เราเลือกจะมาอยู่ในช่อง Selected Items : จากนั้นเราก็สามารถคัดลอกไปยัง Code Window ของ โมดูลหรือฟอร์มที่เราต้องการได้ทันที



รูปที่ 2-8 แสดงการประกาศฟังก์ชันที่ถูกเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. กรณีเลือกค่าคงที่ที่สามารถทำได้โดยเปลี่ยนที่ช่อง API Type ; ให้เป็นรูปแบบที่ต้องการ แล้วใช้วิธีการค้นหาคล้ายๆ กับการประกาศฟังก์ชัน



รูปที่ 2-9 แสดงตัวอย่างการเลือกใช้ค่าคงที่ของ API Viewer

2.2.4 การประกาศฟังก์ชันใน Windows API

ในการเรียกใช้ฟังก์ชันต่างๆ ของ Windows API นั้น มีขั้นตอนที่แตกต่างจากการใช้เรียกฟังก์ชันที่วิซวลเบสิกเตรียมไว้ให้ หรือฟังก์ชันที่เราสร้างขึ้นมานั่นคือ ต้องมีการประกาศค่าฟังก์ชันก่อนการใช้งาน

สำหรับการใช้งานฟังก์ชันนั้น เราจะคัดลอกมาจาก API Viewer ซึ่งฟังก์ชันที่คัดลอกมานั้น มีรูปแบบการประกาศดังนี้

Public **Declare Function** **GetMenu** **Lib "user32"** (**ByVal hwnd As Long**) **As Long**

Declare Function
ชื่อฟังก์ชัน
Lib
พารามิเตอร์
ค่ารีเทิร์น

จากตัวอย่าง เป็นการประกาศฟังก์ชัน GetMenu และฟังก์ชัน RegCloseKey จะเห็นว่าฟังก์ชันที่ประกาศนั้น ประกอบด้วยองค์ประกอบต่างๆ ดังนี้

- **Declare Function** เป็นคำสั่งแสดงว่านี่คือ การประกาศฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชื่อฟังก์ชัน เป็นชื่อของฟังก์ชัน จากตัวอย่างนี้ ชื่อของฟังก์ชัน คือ GetMenu และ RegClose
- Lib เป็นการบอกให้ทราบว่าให้ค้นหาจากไฟล์ .DLL ซึ่งจะต้องระบุชื่อไฟล์ .DLL นั้น ยกเว้นอยู่ในไฟล์ USER32.DLL, KERNEL32.DLL และ GDI32.DLL
- พารามิเตอร์ เป็นรายการของพารามิเตอร์ที่ต้องมีการผ่านค่าให้ฟังก์ชัน
- ค่าที่รีเทิร์น เป็นชนิดข้อมูลที่จะคืนค่าให้กับผู้เรียกใช้ จากตัวอย่างมีชนิดข้อมูลเป็น Long ในบางฟังก์ชันนั้น อาจมีองค์ประกอบอื่นเพิ่มเติม เช่น

Public Declare Function RegisterClass Lib "user32" Alias "RegisterClassA" (Class As WNDCLASS) As Long

Alias

- Alias เป็นชื่อที่ระบบปฏิบัติการเรียกใช้ฟังก์ชันนี้ จากตัวอย่างเราจะเรียกใช้งานในแอปพลิเคชันของเราในชื่อฟังก์ชัน RegisterClass แต่ Windows 95/98 จะรู้จักฟังก์ชันเดียวกันนี้ในชื่อ RegisterClassA เป็นต้น

2.2.5 ข้อควรรู้ในการใช้ API

1. Save โปรแกรมที่เขียนในวิชวลเบสิกก่อน รันโปรแกรม เสมอๆ
2. พารามิเตอร์ที่ยังไม่มีค่าสแกน ByVal อยู่ข้างหน้า โดยเฉพาะสตริงควรตรวจสอบก่อน
3. ให้พิมพ์ข้อความ Option Explicit ในส่วนของทุกๆ โมดูล เพื่อป้องกันตัวแปร Variant
4. ถ้าเกิด GPF (General Protection Fault) ขณะทดสอบโปรแกรมอยู่ให้ Restart PC ใหม่
5. ถ้าในวิชวลเบสิก ฟังก์ชันคล้ายๆ API ก็ควรหนีไปใช้ของวิชวลเบสิกจะดีกว่าครับ
6. ไม่ควรที่จะใช้ประกาศข้อมูลประเภท As Any ซึ่งจะส่งผลให้ข้อมูลผิดพลาด มีผลให้เกิด GPE
7. ข้อมูลในวิชวลเบสิกประเภทสตริง ไม่สามารถกำหนดพารามิเตอร์ใน API ได้ แต่สามารถใส่คำว่า ByVal ไว้ข้างหน้าก่อน
8. ถ้าพารามิเตอร์เป็นสตริงขนาดของสตริงที่กำหนดให้ในตัวแปรต้องมากกว่าขนาดสูงสุดที่ฟังก์ชัน ใช้อยู่ 1 Byte ของแอสกี 0 ที่ต่อท้ายสตริง
9. ต้องสร้างโมดูลไฟล์ไว้ในวิชวลเบสิก
10. การประกาศฟังก์ชันตอนแรก สามารถทำได้ทั้งแบบ Public และ Private

ถ้าเราเขียนโปรแกรมที่มีฟอร์มหลายๆ หน้าและต้องใช้ฟังก์ชัน API หลายฟอร์ม ควรที่จะประกาศฟังก์ชัน API ลักษณะ Public อยู่ใน โมดูลไฟล์ อีกทั้งสามารถนำกลับมาใช้ใหม่ได้ในกรณีที่เราเขียนโปรแกรมอื่นๆ

บทที่ 3

การเขียนโปรแกรมฐานข้อมูลด้วย ADO

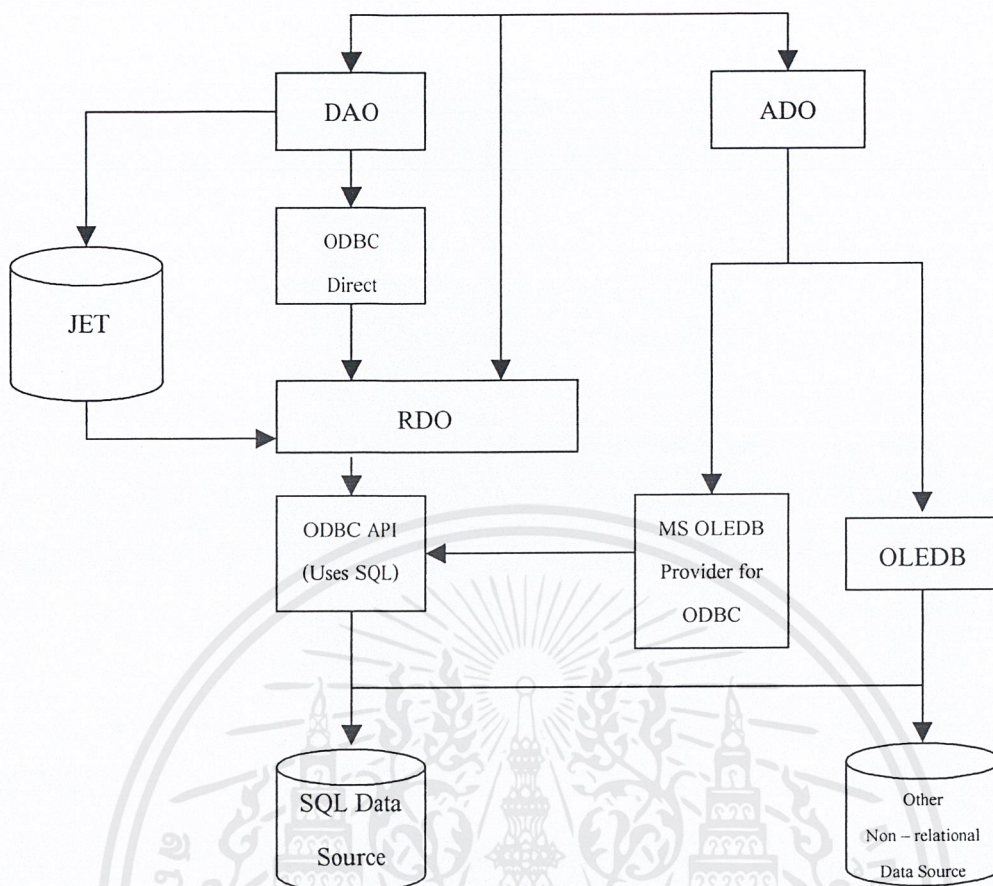
Active Data Object (ADO) เป็นเทคโนโลยีใหม่ที่ถูกนำมาใช้ในการติดต่อกับฐานข้อมูล และกำลังได้รับความสนใจ รวมทั้งมีแนวโน้มที่จะเป็นเทคโนโลยีที่ถูกนำมาใช้ในอนาคต เนื่องจากสามารถติดต่อกับข้อมูลได้หลายรูปแบบ ซึ่งวิซวลเบสิก 6.0 ได้นำเอาเทคโนโลยีมาใช้เช่นเดียวกัน โดยผ่านทางคอนโทรลที่ชื่อว่า “ADO Data” โดยจะเห็นได้จากฟอร์ม ที่สร้างขึ้นจาก VB Data Form Wizard ของวิซวลเบสิก 6.0 ล้วนแต่ใช้คอนโทรลนี้ในการติดต่อกับฐานข้อมูลแทน Data Control ที่ใช้ในวิซวลเบสิกเวอร์ชันก่อนๆ ที่ผ่านมา ดังนั้นในบทนี้ เราจะมาศึกษาการเขียนโปรแกรมกับฐานข้อมูลด้วย ADO Data Control กัน

3.1 ADO คืออะไร

ย่อมาจากคำว่า Active Data Object โดยเป็นเทคโนโลยีใหม่ของบริษัท Microsoft ที่ถูกพัฒนาขึ้นเพื่อใช้ติดต่อกับฐานข้อมูลในรูปแบบต่างๆ เนื่องจากในปัจจุบัน คอมพิวเตอร์มีความสามารถในการใช้งานได้หลากหลาย ซึ่งส่งผลให้ข้อมูลต่างๆ ที่ใช้อยู่ในระบบงานคอมพิวเตอร์ มีรูปแบบที่หลากหลายมากขึ้นไปด้วย เช่น ข้อมูลที่อยู่ในรูปของเท็กซ์ ฐานข้อมูลที่มีโครงสร้างในรูปแบบ Relational ข้อมูลที่อยู่ในรูปแบบของอีเมล ข้อมูลที่เป็นรูปภาพ ข้อมูลที่อยู่ในรูปแบบอินเทอร์เน็ต ฯลฯ เป็นต้น ประกอบกับรูปแบบของข้อมูลที่หลากหลายนี้ ได้ก่อให้เกิดปัญหาในการใช้ข้อมูลร่วมกันระหว่างแอปพลิเคชันต่างๆ ที่มีการใช้ข้อมูลที่ต่างรูปแบบกัน ดังนั้น ในปัจจุบัน จึงได้เกิดแนวความคิดที่ชื่อว่า “Universal Data Access” ขึ้น แนวความคิดนี้ มีจุดมุ่งหมายที่ต้องการให้เกิดศูนย์กลางของการติดต่อระหว่างแอปพลิเคชัน ที่มีการใช้ข้อมูลที่ต่างรูปแบบกัน เพื่อให้แอปพลิเคชันต่างๆ ที่ใช้ข้อมูลที่ต่างรูปแบบกันนั้น สามารถใช้ข้อมูลร่วมกันได้โดยไม่ต้องเปลี่ยนแปลงรูปแบบของข้อมูลเดิมของแต่ละแอปพลิเคชันแต่อย่างใด ซึ่งคล้ายกับการใช้ Open Database Connectivity (ODBC) แต่ต่างกันตรงที่ ODBC ถูกสร้างขึ้นโดยบริษัทที่มีข้อตกลงในการใช้ข้อมูลร่วมกัน ดังนั้น ODBC จึงไม่สามารถรองรับข้อมูลได้ทุกรูปแบบ

ในการติดต่อกับฐานข้อมูลของ ADO จะกระทำผ่าน OLE DB แทนการใช้ ODBC เนื่องจาก OLE DB เป็นเทคนิคใหม่ที่ถูกพัฒนาขึ้นเพื่อรองรับแนวความคิดแบบ Universal Data Access ซึ่งสามารถแสดงด้วยภาพ ดังรูปที่ 3-1

แต่เนื่องจากการใช้ OLE DB มีวิธีการที่ซับซ้อน และยุ่งยาก ดังนั้น จึงต้องกระทำผ่าน Object ต่างๆ ที่ ADO เตรียมไว้ให้ ซึ่งจะส่งผลให้สามารถติดต่อกับข้อมูลที่อยู่ในรูปของฐานข้อมูล เช่น Microsoft Access, SQL Server, Oracle ฯลฯ เป็นต้น หรือข้อมูลที่ไม่ได้อยู่ในรูปของฐานข้อมูล เช่น อีเมล ข้อมูลในรูปของวีดีโอ ข้อมูลในรูปแบบของอินเทอร์เน็ต ฯลฯ เป็นต้น



รูปที่ 3-1 แสดงการติดต่อกับฐานข้อมูลของ ADO

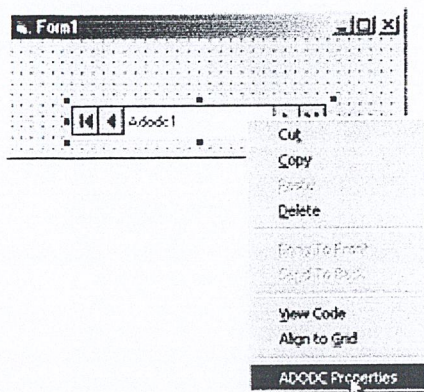
3.2 เขียนโปรแกรมติดต่อกับฐานข้อมูล ผ่าน ADO ด้วยวิซวลเบสิก

การติดต่อกับฐานข้อมูลโดยใช้ ADO นี้เป็นเทคโนโลยีใหม่ที่ ทาง Microsoft หมายมั่นให้มาแทนรูปแบบการติดต่อแบบเดิม ๆ ที่เราใช้กันอยู่ ADO จะใช้ OLEDB Provider เป็นกลไกในการเข้าถึงข้อมูลในฐานข้อมูลแทน JET Engine โดยเปลี่ยนมุมมองการติดต่อกับฐานข้อมูลที่เราต้องกำหนดชนิดของฐานข้อมูล มาเป็นมองที่รูปแบบการเชื่อมต่อ Connection แทน ในวิซวลเบสิกเราสามารถใช้อะไรได้ 2 แบบคือ

1. ADO Data Control ร่วมกับกลุ่มของ ActiveX Bound Control โดยเราจะสังเกตจากชื่อของคอนโทรลที่มีคำว่า OLEDB ต่อท้าย เช่น DataGrid , DataList เป็นต้น

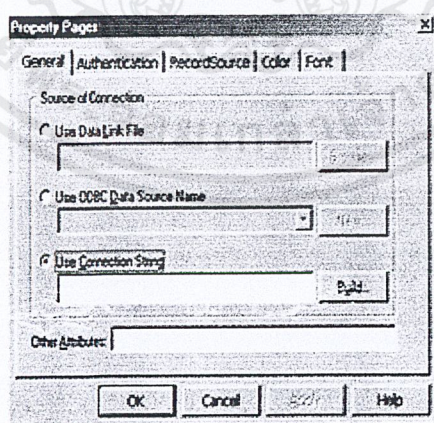
2. การเขียนโปรแกรมควบคุม ADO โดยตรง ซึ่งจะมีความเร็วมาก เพราะไม่ต้องใช้คอนโทรลแต่เราจะยุ่งยากกว่าในการเขียน Code ในตอนแรกเราจะแสดงรูปแบบโดยใช้ ADO Data Control เราจะต้องเพิ่ม ADO Data โดยเลือกที่เมนู Project > Component... แล้วเลือก Microsoft ADO Data Control 6.0 (Sp4) (OLEDB) ๕ เมื่อเราวางคอนโทรลที่โครงการของเราโดย ADO Control จะมีลักษณะเหมือนกับ Data Control เลย แล้วต่อไปเราจะต้องสร้าง Connection เพื่อติดต่อกับฐานข้อมูลเสียก่อนโดย

คลิกขวาที่ Control ADO แล้วเลือก ADODC Properties ดังรูป



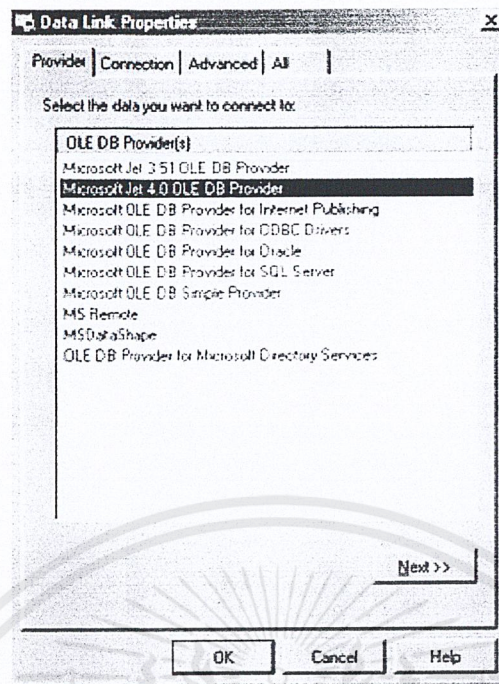
รูปที่ 3-2 แสดงรูปการเลือก ADODC Properties

โดยใน Tab General จะเป็นการกำหนด วิธีติดต่อ 3 แบบ คือ *Data Link* , *ODBC* , *Connection String* ดังรูปที่ 3-3 เราสามารถจะเลือกการติดต่อได้เพียง 1 อย่างจาก 3 อย่างตามความเหมาะสมของเรา โดยจะเริ่มจะวิธี Use Connection String ให้คลิกที่ Build ... เพื่อเข้าสู่ขั้นตอนต่อไปในหน้า Data Link Properties จะประกอบไปด้วย 4 ส่วน ดังรูปที่ 3-4 คือ *Provider* ใช้กำหนดว่าเราจะใช้ OLEDB Provider แบบใด Connect กับ ฐานข้อมูลของเรา ซึ่งก็ขึ้นกับว่าเราใช้ฐานข้อมูลแบบใดใน หน่วยงานของเรา ถ้าเราใช้ Access 2000 ก็ให้เลือก Provider แบบ Microsoft Jet 4.0 OLEDB Provider ส่วน Access97 ให้เลือกแบบ Provider แบบ Microsoft Jet 3.51 OLEDB Provider แต่ก็สามารถใช้ Provider แบบ Microsoft Jet 4.0 OLEDB Provider ร่วมกับ Access 97 ได้ไม่ผิดกติกาอะไรหลังจากที่เลือกแล้วก็ให้กด Next >> เพื่อไปยัง Tab Connection



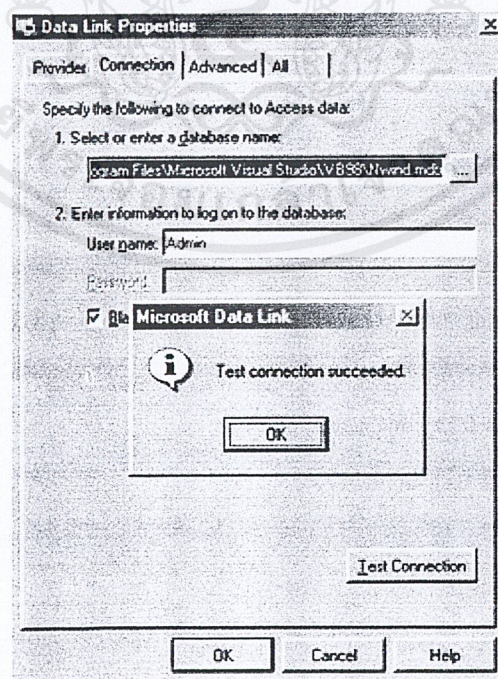
รูปที่ 3-3 แสดงการเลือก Use Connection String ในหน้าของ Property Pages

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 แสดงการเลือก Provider ในหน้าของ Data Link Properties

Connection ใช้กำหนดฐานข้อมูลที่เราติดต่อด้วย เมื่อเราเลือกฐานข้อมูล ที่ต้องการได้แล้วโดยในที่นี่จะเลือกใช้ Nwnd.mdb ใน Folder VB จากนั้น ก็ให้ลองคลิก Test Connection ดังรูปที่ 3-5 เพื่อทดสอบการติดต่อว่า สามารถจะติดต่อกับฐานข้อมูล ที่เราต้องการได้หรือไม่ ถ้าติดต่อสำเร็จก็จะมีไอคอนแสดงมาให้เราดู หลังจาก Test Connection แล้วก็ให้กด Ok เพื่อกลับมายังหน้า Property Pages เพื่อกำหนด Command



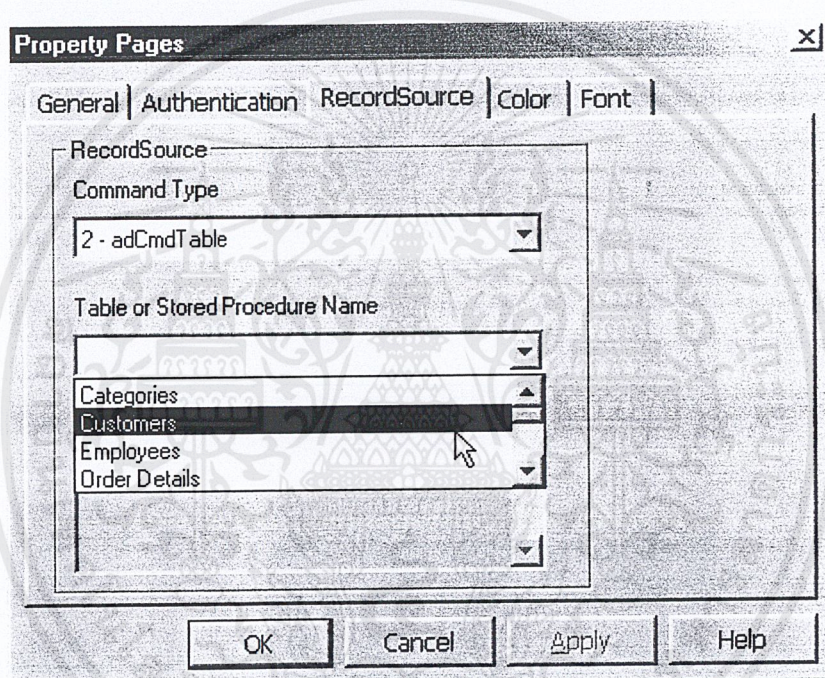
รูปที่ 3-5 แสดงการเลือก Connection ในหน้าของ Data Link Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่เรากำหนด Connection เรียบร้อยแล้ว ก็ถึงขั้นตอนที่ต้องกำหนด RecordSource ซึ่งเราจะทำผ่าน Object Command ของ ADO เราจะต้องกำหนดชนิดของ Command ทุกครั้งก่อนเข้าสู่ฐานข้อมูล เพราะมิฉะนั้นจะบอกว่า Error ชนิดของ Command จะมี 4 ชนิดดังนี้

- 8 - adCmdUnknow ไม่ระบุประเภทของ Command
- 1 - adCmdText ใช้คำสั่ง SQL ในการเข้าถึงฐานข้อมูล
- 2 - adCmdTable ให้เข้าถึงแบบตาราง
- 4 - adCmdStoredProc ให้เข้าโดยใช้ Stored Procedure ซึ่งใช้ใน SQL Server

โดยในที่นี้จะเลือกแบบ 2 - adCmdTable เพื่อเข้าถึงฐานข้อมูล แบบตาราง Customers จากนั้นก็คลิก Ok ได้เลย เท่านั้นก็เสร็จ การกำหนด Connection ดังรูปที่ 3-6



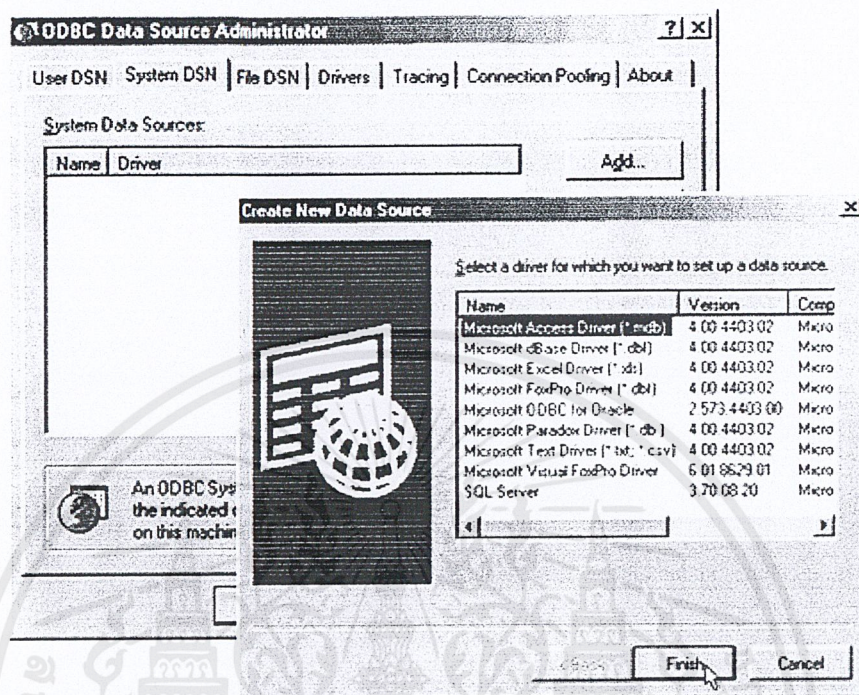
รูปที่ 3-6 แสดงการเซตค่า RecordSource ใน Property Pages

ต่อมาก็เป็นวิธีที่ 2 โดยเราจะเปลี่ยนมาลองใช้ ODBC บ้าง วิธีนี้ผู้ที่ใช้งาน Web App เช่น Asp คงจะคุ้นเคยมากเพราะเราจะกำหนด DSN สำหรับติดต่อกับฐานข้อมูลของเราโดยผ่าน ODBC เมื่อเราต้องการใช้งานฐานข้อมูลนั้นภายหลัง ก็จะใช้ชื่อ DSN ที่เรากำหนดมาเป็นที่อ้างอิงกับฐานข้อมูลที่เรากำหนด โดยมีขั้นตอน ดังนี้ ที่ Desktop ของ Windows 98 เลือก Start > Settings > Control Panel แล้วดับเบิลคลิกที่ไอคอน ODBC Data Source (32 Bit) หรือถ้าเป็น Win2000 แล้วเลือกที่ Administrative Tools แล้วเลือกที่ไอคอน ODBC Data Source จากนั้นก็เลือกที่ System DSN Tab เพื่อเลือก Data Source ที่ต้องการติดต่อกับฐานข้อมูลของเรา

จากหน้า System DSN ให้เราคลิกที่ Add เพื่อเพิ่ม DSN จากนั้นก็ให้เลือกชนิดฐาน ข้อมูลที่ต้องการ ติดต่อ ในกรณีที่เป็น Access ก็ให้ เลือก Microsoft Access Drivers (*.mdb) หรือถ้าใครใช้ฐานข้อมูล

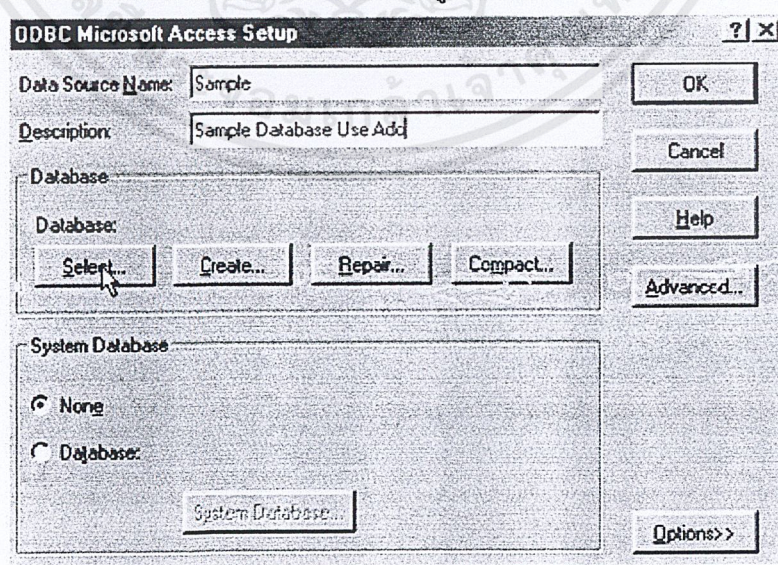
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลแบบอื่น เช่น DBase FoxPro Sql Server ก็สามารเลือก ได้ตามต้องการจากนั้นก็ให้คลิกที่ Finish เพื่อไปยังหน้าถัดไป ดังรูปที่ 3-7



รูปที่ 3-7 แสดงการเลือก Data Source

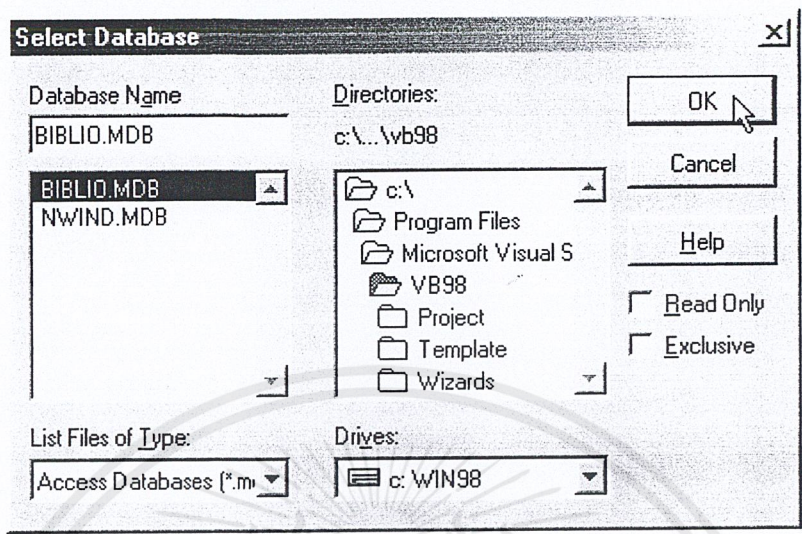
ใน Dialog ODBC Microsoft Access Setup มีรายละเอียดดังนี้ ช่อง Data Source Name ให้เราตั้งชื่อ DSN ที่ต้องการ ควรจะเป็นชื่อเดียวกับฐานข้อมูลของเรา แต่ในที่นี้จะใช้ชื่อ Sample และในช่อง Description ก็ให้ใส่รายละเอียดต่าง จะใส่หรือไม่ใส่ก็ได้ ดังรูปที่ 3-8



รูปที่ 3-8 การเซตค่าหน้า ODBC Microsoft Access Setup

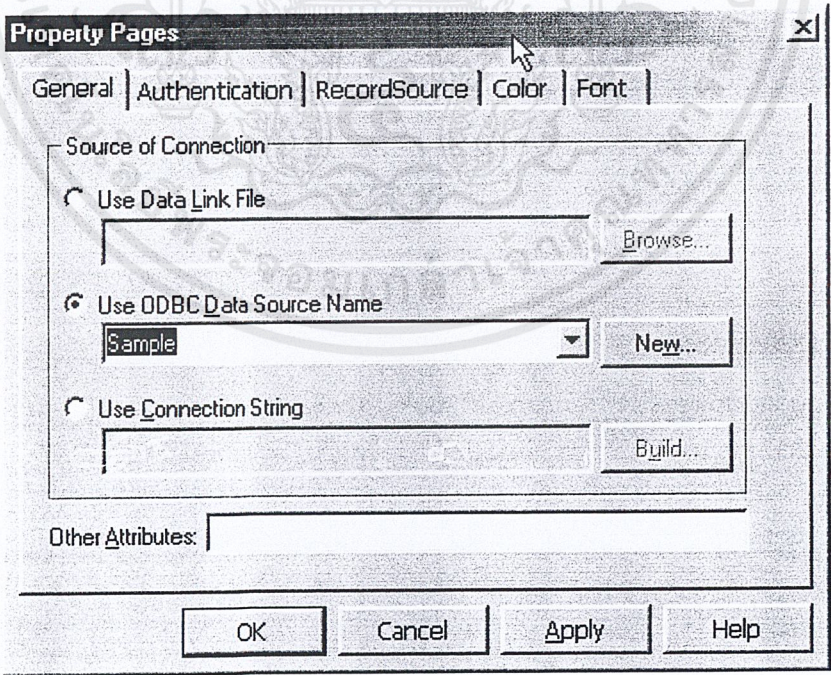
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นคลิกที่ Select ... เพื่อเลือกฐานข้อมูลที่ต้องการติดตั้งด้วย ให้เราเลือกฐานข้อมูลที่ต้องการเชื่อมต่อจากนั้นก็คลิก Ok ในหน้าต่าง Select Database และคลิก Ok ใน ODBC Microsoft Access Setup



รูปที่ 3-9 แสดงหน้าต่างการเลือก Select Database

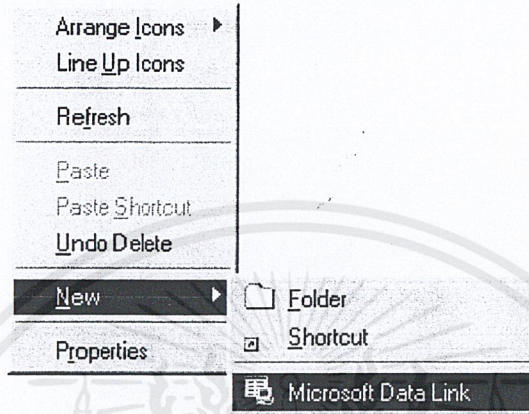
จะเห็นได้ว่าใน Dialog ODBC Data Source Administrator มีชื่อ DSN ที่เราเพิ่มเข้ามาแล้วกด OK จากตัวอย่างที่เราสร้างขึ้น ถ้าเราต้องการเลือกวิธี ODBC Data Source Name ให้เราเลือก DSN ที่ต้องการใช้ ให้เราเลือก Sample ดังรูปที่ 3-10



รูปที่ 3-10 แสดงหน้าต่าง Property Pages

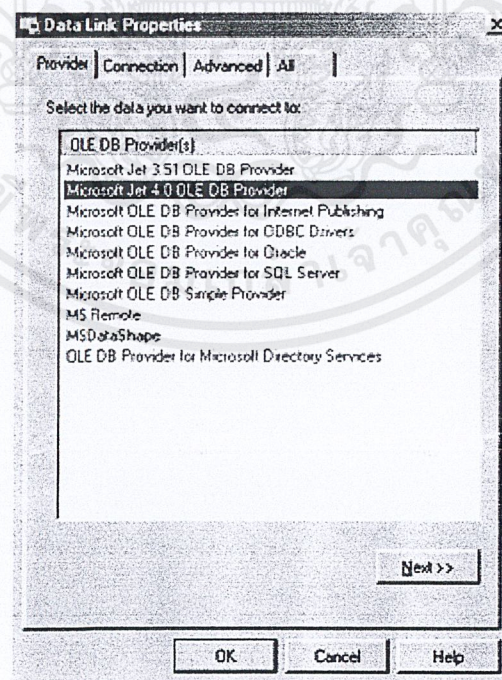
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถึงวิธีสุดท้ายก็คือการใช้ โดยจะเริ่มที่ใส่ Data link ก่อนโดยคลิกที่ Folder ที่ต้องการเก็บ .udl โดยเลือก คำสั่ง New > Microsoft Data Link ดังรูปที่ 3-11 และให้เรา ตั้งชื่อไฟล์โดยควรจะต้องตั้งชื่อเดียวกับฐานข้อมูล และถ้าใครใช้ 2000 หรือ Update Service Pack ก็คงไม่มี Microsoft Data Link ใน New ดังนั้น เราต้องใช้ลูกเล่น นิดหน่อย คือ โดยให้เรา New File เอาเป็น Text Document แล้วเปลี่ยนType ของ File ให้เป็น . udl เท่านั้นเราก็จะได้ Data Link File ได้เหมือนกัน



รูปที่ 3-11 แสดงรูปการเลือก Microsoft Data Link

หลังจากที่เราได้ Data Link File ดังรูป  เรียบร้อยแล้ว เราก็ต้องผูก Data Link File นี้กับฐานข้อมูลที่เราต้องการจะ Connect ด้วย เมื่อเรา คืบเบิ้ลคลิก ที่ไฟล์ Data Link จะได้ดังรูปข้างล่างนี้



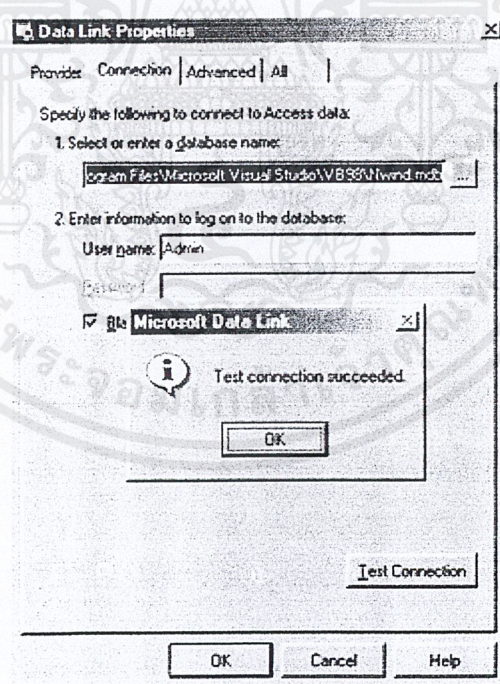
รูปที่ 3-12 แสดงการเลือก Provider ในหน้าของ Data Link Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน Tab Provider ใช้สำหรับกำหนดว่า เราจะใช้ OLEDB Provider แบบใด ซึ่ง ก็จะขึ้นกับว่า เราใช้ฐานข้อมูลแบบใด เช่น สมมติว่าเราใช้ Access 2000 ก็ให้เลือก Provider แบบ Microsoft Jet 4.0 OLEDB Provider ส่วน Access97 ให้เลือก Provider แบบ Microsoft Jet 3.51 OLEDB Provider แต่ก็สามารถใช้ Provider แบบ Microsoft Jet 4.0 OLEDB Provider ร่วมกับ Access 97 ได้ ส่วน Provider อื่น เช่น Microsoft OLE DB Provider For ODBC Drivers ก็บอกในตัวอย่างแล้วว่าให้ติดต่อผ่าน ODBC ดังนั้น ใครใช้ฐานข้อมูลอะไรก็ให้ใช้ตามนั้น จะเห็นว่าฐานข้อมูลของ SQL Server กับ Oracle สามารถติดต่อได้ 2 แบบคือผ่าน ODBC และ OLE DB Provider เฉพาะตัว ทั้งนี้เพราะทั้ง SQL Server กับ Oracle เป็น RDBMS ที่เป็นระดับ Server ดังนั้น ทาง Microsoft จึงออก OLE DB Provider แยก ออกมาเพื่อประสิทธิภาพสูงสุด

หลังจากเลือก Provider ที่ต้องการ ได้แล้วก็คลิก Next >> ก็จะเปลี่ยนไปหน้า Connection โดยในหน้านี้หลังจากที่เราเลือก Provider แล้วก็ต้องมาเลือกฐานข้อมูลที่จะติดต่อผ่าน Provider ที่เรากำหนด โดยจะใช้ฐานข้อมูล Biblio ที่มาพร้อมกับวิชวลเบสิก ถ้าใครอยากใช้ฐานข้อมูลอื่นอื่นก็ตามสะดวก ไม่บังคับ

เมื่อเราเลือกฐานข้อมูลที่ต้องการ ได้แล้วก็ให้ลองคลิก Test Connection เพื่อทดสอบการติดต่อว่า สามารถจะติดต่อกับฐานข้อมูลที่เราต้องการ ได้หรือไม่ ถ้าติดต่อสำเร็จก็จะมี Dialog แสดงมาให้เราดู อย่างรูปที่ 3-13 หลังจากนั้นก็ให้เรากลับไปหน้าจอทำงานที่ยังค้างของ เราตอนนี้ เราสามารถใช้วิธี Use Data Link File โดยเลือกไฟล์ Udl ของเราได้



รูปที่ 3-13 แสดงการเซตค่า Connection ในหน้า Data Link Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การสื่อสารข้อมูลด้วยพอร์ตอนุกรม

4.1 การเขียนโปรแกรมควบคุมฮาร์ดแวร์ต่างๆ

การใช้ VB เขียนโปรแกรมติดต่อ I/O ผ่านทางพอร์ตของเครื่องคอมพิวเตอร์ ไม่ว่าจะเป็นทาง พอร์ตอนุกรม(RS-232) หรือที่รู้จักในชื่อ Com1, Com2 และ พอร์ตขนาน หรือ Printer Port นั้นเอง หรือเราอาจใช้ Card I/O 8255 ซึ่งเป็นการขยายPort I/O ของพอร์ตขนาน ก็สามารถทำการติดต่อกับฮาร์ดแวร์ ภายนอกผ่านพอร์ต ดังที่กล่าวมาได้ อีกทั้งสามารถติดต่อผ่านระบบเน็ตเวิร์ค โดยผ่านช่องทางการติดต่ออย่าง TCP/IP จะเห็นได้ว่า VB ก็สามารถทำงานด้านฮาร์ดแวร์อินเทอร์เฟต ได้ไม่แพ้ภาษาตัวอื่นๆและง่ายกว่า จึงทำให้สามารถทำความเข้าใจเพื่อจะนำไปใช้งานได้สะดวกและรวดเร็ว

4.1.1 พอร์ตอนุกรม (RS-232)

สามารถติดต่อกับอุปกรณ์ต่างๆ ที่มีการติดต่อกับอุปกรณ์ภายนอกผ่านทาง RS-232 อย่างเช่น เครื่องชั่งน้ำรวมถึง โคลดเซล(เป็นเซ็นเซอร์ชนิดหนึ่งใช้สำหรับวัดน้ำหนักซึ่งที่ชุดแสดงผลภายในเป็นชุดไมโครคอนโทรลเลอร์ จะมีสัญญาณรับส่งทาง RS-232), เครื่องวัดงานทางด้านไฟฟ้า, ไมโครคอนโทรลเลอร์, ควบคุมอุปกรณ์ไฟฟ้า, โอนถ่ายข้อมูลในฮาร์ดดิสก์ระหว่างเครื่องคอมพิวเตอร์ด้วยกัน, ควบคุมแปดปีงมอเตอร์ เป็นต้น ข้อดีของการติดต่อข้อมูลกันผ่านทาง RS-232 ก็คือสามารถใช้ได้ในระยะทางไกลๆระหว่างอุปกรณ์ ที่ติดต่อกัน

เนื่องจากที่ Microsoft Visual Basic 5,6 จะมีตัวคอนโทรลชื่อ MS Comm ที่ใช้ติดต่อกับพอร์ตอนุกรม (RS-232) ให้ไว้อยู่แล้วไม่จำเป็นต้องเขียนโค้ดให้ยุ่งยากทำให้การพัฒนาโปรแกรมในด้านนี้ได้เร็วและเป็นมาตรฐานในทิศฐานการเขียนโปรแกรมเดียวกันของทุกโปรแกรมเมอร์ ลักษณะเมื่อทำงานร่วมกันเป็นทีม

4.1.2 พอร์ตขนาน

สามารถทำการประยุกต์ใช้งานได้ดีเพราะสามารถทำงานได้ที่ละ 8 บิตในการติดต่อข้อมูลกับอุปกรณ์ภายนอกให้ระดับแรงดันที่ใช้กับอุปกรณ์ TTL ที่สัญญาณลอจิกเป็น "1" เท่ากับ +5 โวลต์ และลอจิกเป็น "0" เท่ากับ 0 โวลต์ในเมื่อที่ข้อดีก็ต้องมีข้อเสีย ก็คือไม่สามารถทำงานในระยะทางที่ไกลๆ ระหว่างอุปกรณ์ที่ติดต่อกันจะเกิดความเสี่ยงของข้อมูลขึ้นง่ายเนื่องจากระดับแรงดันไม่สม่ำเสมออีกทั้งสิ้นเปลืองค่าใช้จ่ายในการต้องใช้สายสัญญาณหลายเส้น โครงการที่นำมาใช้เช่น ควบคุมแปดปีงมอเตอร์ 3 แกน XYZ, อิเล็กทรอนิกส์ควบคุมโซลินอยวาล์ว, ควบคุมอุปกรณ์ภายในบ้านหรือสถานที่พิเศษ เป็นต้น

Microsoft Visual Basic 4,5,6 นั้นจะไม่มีฟังก์ชันสำหรับติดต่อพอร์ตโดยตรง เมื่อกับตอนยังอยู่บน DOS ดังเช่น BASIC,QBASIC ที่ใช้ฟังก์ชัน เช่น OUT เป็นต้น ผมลองค้นหาจากใน MSDN ของ Visual Studio สามารถติดต่อทางพอร์ตได้โดยใช้ API(Application Programming Interface) โดยจะต้องมี

ไฟล์ DLL ไว้สำหรับเรียกฟังก์ชันเพื่อติดต่อพอร์ต โดยจะเขียนด้วยภาษา VC++, C++, Pascal เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 TCP/IP

ยุคสมัยนี้ถ้าไม่ใช้การติดต่อด้านนี้คงจะหนีไม่พ้น TCP/IP ซึ่ง เป็นระบบเครือข่ายเอาเป็นว่าใช้คำว่า Internet ก็แล้วกันที่ใช้โพรโทคอลมาตรฐานชื่อ TCP/IP ในการสื่อสารผ่านระบบเพื่อติดต่อกับเครื่องคอมฯ อื่นๆ โพรโทคอล TCP/IP นั้นประกอบด้วยส่วนที่สำคัญ 2 ส่วนก็คือ TCP(Transmission Control Protocol) และ IP(Internet Protocol) ในการติดต่อกันจริงๆแล้วเราคงจะไม่สามารถเห็นขั้นตอนการทำงาน ของระบบ ได้เพราะเป็นการทำงานของซอฟต์แวร์และฮาร์ดแวร์แต่เราจะอธิบายเพื่อความเข้าใจของ โพรโทคอล TCP/IP ให้ดูกันดังมีส่วนประกอบดังนี้

- IP Address : สำหรับการรับส่งข้อมูลในระบบอินเทอร์เน็ต จะถูกกำหนดและอ้างอิงด้วยหมายเลข ประจำเครื่องนั้นก็เครื่อง IP Address ซึ่งในระบบอินเทอร์เน็ต จะมีเครื่องคอมพิวเตอร์เป็นจำนวนมาก ที่อยู่ในระบบ ในการที่จะใช้ IP Address อาจจะไม่สะดวก จึงได้มีการเปลี่ยนมาใช้เป็น ชื่อ ในความ เข้ากันก็คือ Domain name โดยทั้งหมดนี้อยู่ในระบบ Name Services ซึ่งเป็นการอ้างอิงชื่อแทนหมายเลข นั้นเอง
- Routing Configuration : ชื่อของโพรโทคอล TCP/IP ก็คือในการกำหนดเส้นทางสำหรับการรับส่ง ที่ สามารถเลือกเส้นทางในการรับส่งข้อมูลได้อย่างอัตโนมัติหากถ้าเกิดเส้นทาง บางเส้นทางเสียหาย ระบบกลไกในการกำหนดเส้นทางสำหรับการรับส่งข้อมูลของโพรโทคอล TCP/IP ก็จะเลือกเส้นทาง ให้เหมาะสมถูกต้องให้สามารถรับส่งข้อมูลได้
- Protocol, Ports, Sockets : เป็นช่องทางสำหรับกำหนดทิศทางของการรับส่งข้อมูลนอกเหนือจากที่จะ ต้องกำหนดหลังจาก IP Address

ในงานควบคุมระยะไกลระบบนี้เป็นระบบที่มีประสิทธิภาพดี ซึ่งสามารถจะเขียนโปรแกรมไปควบคุม I/O ของระบบคอมพิวเตอร์ หรือฮาร์ดแวร์ที่มีการติดต่อกับระบบเดียวกัน สำหรับภาษาวิซวลเบสิก มีเครื่องมือช่วยในการเขียนโปรแกรมด้านนี้อยู่แล้วมีชื่อว่า Winsock หรือจะใช้แบบฟังก์ชัน API ก็มีให้ดูซึ่งอยู่ในเว็บไซต์ <http://www.ThaiIO.com> ก็มีอธิบายพร้อมให้ศึกษากัน

เราสามารถเขียนโปรแกรมได้ง่ายขึ้นซึ่งจะมีตัวคอนโทรลชื่อ MSComm สามารถติดต่อผ่าน RS-232 ได้ และ Winsock ที่สามารถติดต่อผ่าน TCP/IP

ในส่วนต่อไปจะขอพูดถึงเกี่ยวกับเนื้อหาที่ต้องใช้ คือ การติดต่อข้อมูลผ่านพอร์ตอนุกรม (RS-232)

4.2 การสื่อสารข้อมูลแบบอนุกรม

การเชื่อมต่อระหว่างพอร์ตอนุกรมแต่ละพอร์ตมีมาตรฐานอยู่สามแบบ คือ มาตรฐาน EIA RS-422, ระบบวงรอบกระแส และมาตรฐาน EIA RS-232 แต่เราส่งสัญญาณอนุกรมโดยใช้มาตรฐาน EIA RS-232 มากที่สุด โดยที่จะมีความยาวของสายได้ไม่เกิน 50 เมตร

4.2.1 มาตรฐาน EIA RS-232

มาตรฐาน EIA RS-232 หรือเรียกอีกอย่างว่า RS-232 กำหนดให้ระดับศักดาไฟฟ้าที่มีค่าเท่ากับ 3 โวลต์หรือสูงกว่า มีค่าทางตรรกะ เป็น 1 กำหนดให้ระดับศักดาไฟฟ้าที่มีค่าเท่ากับ -3 โวลต์หรือต่ำกว่า มีค่าทางตรรกะ เป็น 0 วงจร ไอซีที่ใช้สร้างสัญญาณเหล่านี้ต้องการใช้แหล่งจ่ายไฟขนาด +12 โวลต์ RS-232 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะใช้สาย 1 เส้นสำหรับส่งข้อมูลและอีก 1 เส้นสำหรับรับข้อมูล ตารางต่อไปนี้แสดงตัวเชื่อมต่อแบบ D ชนิด 25 ขา (25 pin D-connector) และ สัญญาณต่างๆของ RS-232 บ่อยครั้งที่สัญญาณต่างๆของ RS-232 จะใช้สาย 1 เส้น สำหรับส่งข้อมูลและสายอีกหนึ่งเส้นจะใช้สำหรับรับข้อมูล โดยสัญญาณในแต่ละสายนี้ จะถูกอ้างอิงเทียบกับกราวด์ (ขาเบอร์ 7) มาตรฐาน RS-232 นี้ยังได้กำหนดสัญญาณ RS-232 ถึงแม้ว่า บ่อยครั้งที่สายสัญญาณส่ง RS-232 นี้ได้กำหนดไว้สัญญาณตอบรับเพื่อใช้ในการควบคุมการรับส่งข้อมูลด้วย

ในการเชื่อมต่อแบบ D ชนิด 25 ขา (25 pin D – connector) และสัญญาณต่างๆของ RS-232 ถึงแม้ว่าบ่อยครั้งที่สายส่งสัญญาณ RS-232 จะใช้ควบคู่กับตัวเชื่อมต่อแบบ 25 ขานี้ แต่ก็ยังไม่เป็นรูปแบบที่มาตรฐาน เนื่องจากตัวเชื่อมต่อแบบใหม่ที่มีขาเพียง 9 ขา (ดังตารางด้านล่างนี้) มาใช้แทน โดยเราจะพบได้ในเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่นใหม่

มาตรฐาน RS-232 จะสามารถใช้ส่งข้อมูลได้ไกลที่สุด 50 เมตร ด้วยอัตราการส่งข้อมูล 9600 อัตร่า ถ้าหากว่าจะส่งข้อมูลมากกว่า 50 เมตรขึ้นไปจะต้องทำการส่งที่เร็วกว่า 9600 อัตร่า

9 Pin	EIA RS-232 Circuit	CCIT V.24 Circuit	RS-232 Description	Signal type & Direction
5	AB	102	Signal ground/common return	Ground/common
2	BB	104	Received data	Data from DCE
3	AA	103	Transmitted data	Data to DCE
1	CF	109	Received line signal detector	Control from DCE
4	CD	108,2	Data terminal ready	Control to DCE
6	CC	107	Data set ready	Control from DCE
7	CA	105	Request to send	Control to DCE
8	CB	106	Clear to send	Control from DCE
9	CE	125	Ring indicator	Control from DCE

ตารางที่ 4-1 แสดงข้อมูลการเชื่อมต่ออนุกรม 9 ขา

4.2.2 มาตรฐาน RS-232C

4.2.2.1 คุณสมบัติของ RS-232C

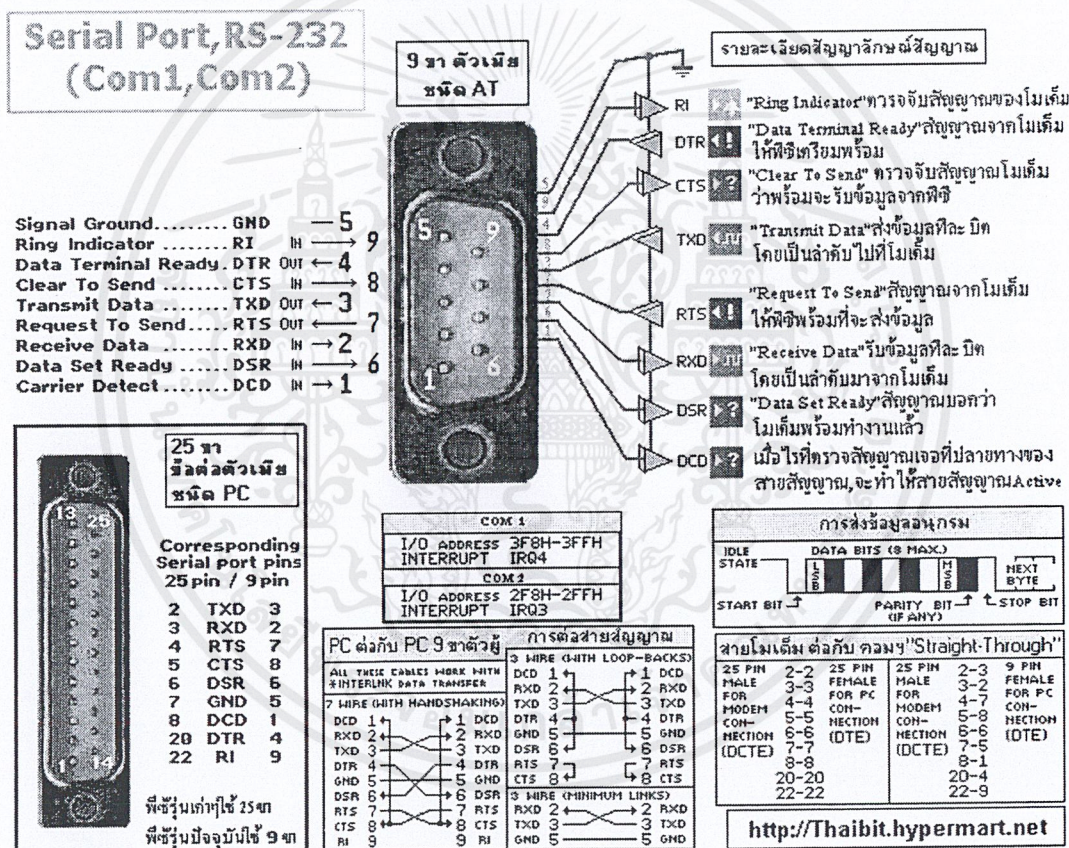
- อัตราการรับส่งข้อมูล : 0-20000 บิต/วินาที
- ระดับ แรงดันไฟฟ้าเอาต์พุตสูงสุดในภาวะไม่มีโหลด : -25 โวลต์ (ลอจิก 1)
+25 โวลต์ (ลอจิก 0)
- ระดับแรงดันไฟฟ้าเอาต์พุตสำหรับโหลด 3-7 กิโลโอห์ม : ลอจิก“1” -15 โวลต์ (7 กิโลโอห์ม)
-5 โวลต์ (3 กิโลโอห์ม)
ลอจิก“0”+15 โวลต์ (7 กิโลโอห์ม)
+5 โวลต์ (3 กิโลโอห์ม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กระแสเอาต์พุตเมื่อลัดวงจร : สูงสุด 500 มิลลิแอมป์เอาต์พุตอิมพีแดนซ์
เมื่อไม่จ่ายไฟเลี้ยงต่ำสุด 300 โอห์ม
- สล็อตเรตทางเอาต์พุตสูงสุด : 30 โวลต์/ไมโครวินาที
- ความต้านทานอินพุตของภาครับ : สูงสุด 7 กิโลโอห์ม
ต่ำสุด 3 กิโลโอห์ม
- ค่าความจุอินพุตของภาครับ : สูงสุด 2500 พิโคฟาร์ด
- ย่านแรงดันอินพุตของภาครับ : -25 โวลต์ ถึง +25 โวลต์

4.2.2.2 การจัดขาสัญญาณของ RS-232 ทั้งแบบ 9 ขา และ 25 ขา

มีด้วยกัน 2 แบบ คือ แบบ 9 ขา และ แบบ 25 ขา ดังรูปนี้



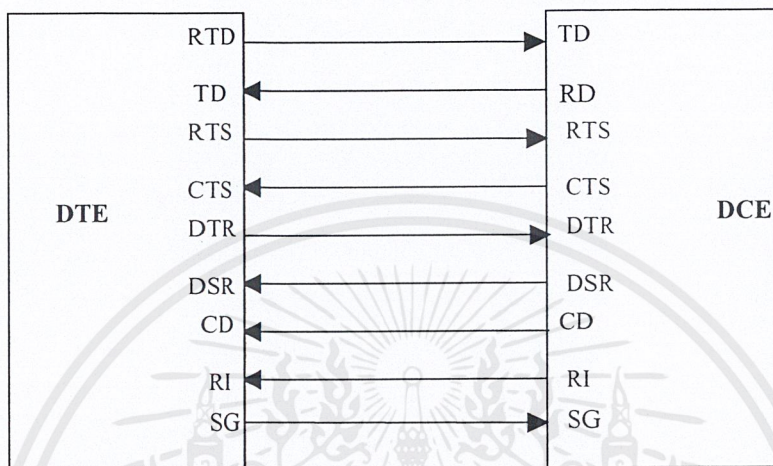
รูปที่ 4-1 แสดงรายละเอียดของขาสัญญาณของ RS-232 แบบ 9 ขา และ 25 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.3 การเชื่อมต่อสัญญาณของ RS-232C

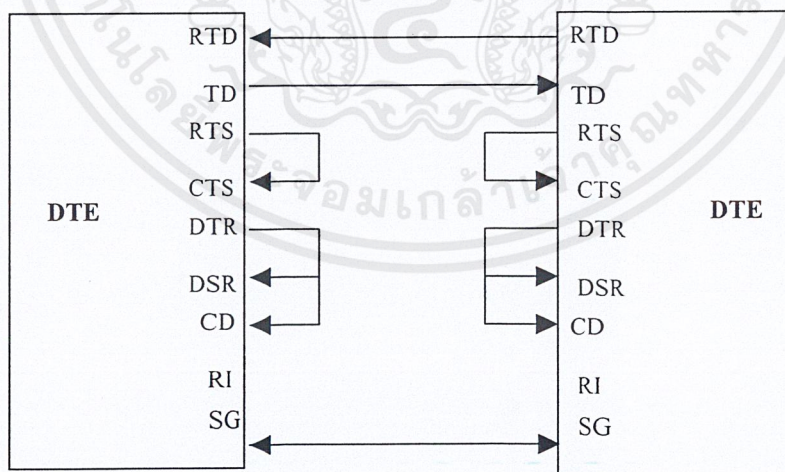
มีลักษณะเชื่อมต่อ 2 แบบด้วยกัน

1. การเชื่อมต่อกันระหว่างอุปกรณ์ DTE (Data terminal Equipment) เช่นคอมพิวเตอร์กับอุปกรณ์ DCE (Data Circuit Terminal) เช่น โมเด็ม แสดงดังรูปต่อไปนี้



รูปที่ 4-2 แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ อุปกรณ์ DCE

2. การเชื่อมต่อกันระหว่างอุปกรณ์ DTE (Data terminal Equipment) กับอุปกรณ์ DTE แสดงดังรูปต่อไปนี้

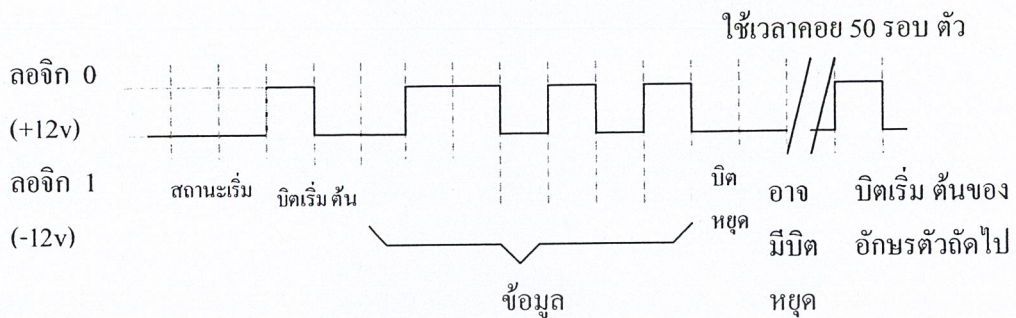


รูปที่ 4-3 แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ อุปกรณ์ DTE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 รูปแบบของข้อมูลอนุกรมและอัตราการส่งข้อมูลในการสื่อสารข้อมูลอนุกรม

อัตราการส่งข้อมูล (baud rate) คือความเร็วในการส่งข้อมูลอนุกรม มีหน่วยเป็นบิตต่อวินาที



รูปที่ 4-4 แสดงรูปแบบของสัญญาณข้อมูลอนุกรมที่ใช้ในการสื่อสารข้อมูลอนุกรม

เมื่อเราไม่ต้องการให้อุปกรณ์เชื่อมต่อภายนอกเชื่อมต่อโดยตรงกับระบบที่ไม่มีโครโปรเซสเซอร์เป็นพื้นฐาน เราจะต้องนำ Optoisolator มาใช้เป็นตัวป้องกันไมโครโปรเซสเซอร์ UART และวงจรอื่นๆในระบบที่มีศักดาไฟฟ้าสูงๆ เพื่อให้กระแสไหลเข้าที่สายกราวด์

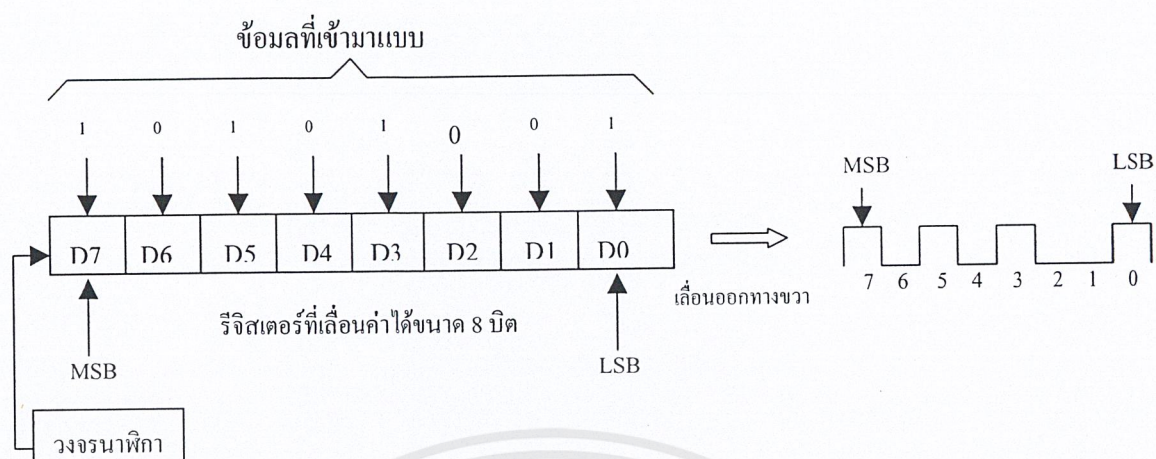
อัตราการส่งข้อมูล	ช่วงเวลาของแต่ละบิต
110	9.91 ms
150	6.67 ms
300	3.33 ms
600	1.67 ms
1200	0.833 ms
2400	0.417 ms
4800	0.208 ms
9600	0.104 ms
19200	0.052 ms

ตารางที่ 4-2 อัตราการส่งข้อมูลและช่วงเวลาของแต่ละบิตข้อมูลในการสื่อสารข้อมูลอนุกรม

4.2.4 การเชื่อมต่อแบบอนุกรมและ UART

ในรูป 4-5 จะแสดงการแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม โดยเริ่มแรกข้อมูลแบบขนานจะถูกนำไปเก็บไว้ในรีจิสเตอร์ Shift Register หลังจากนั้นจะใช้สัญญาณนาฬิกาในการเลื่อนค่ารีจิสเตอร์ออกมาทีละบิต โดยเลื่อนค่าไปทางขวามือ บิตแรกที่ถูกเลื่อนออกมาคือบิต LSB ของข้อมูลและบิตที่สองที่ถูกเลื่อนออกมาคือบิตที่อยู่ถัดจาก LSB และบิตต่อไป สำหรับบิตสุดท้ายที่ถูกเลื่อนออกมาคือบิต MSB ของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

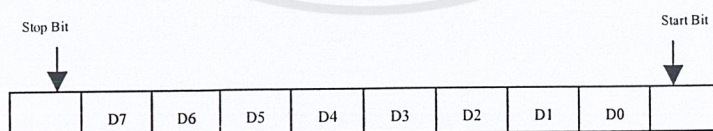


รูปที่ 4-5 แสดงการแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม

การเปลี่ยนแปลงข้อมูลแบบอนุกรมไปเป็นข้อมูลแบบขนานนั้นจะมีขั้นตอนตรงกันข้ามกับที่กล่าวมา นั่นคือข้อมูลแบบอนุกรมจะถูกเลื่อนเข้าเก็บใน Shift Register โดยใช้สัญญาณนาฬิกาเป็นตัวควบคุม และหลังจากที่ได้มีการเลื่อนข้อมูลทุกบิตเข้าไปใน Shift Register ได้หมดแล้ว ข้อมูลในรีจิสเตอร์นี้จะถูกนำออกมาเป็นแบบขนานเพื่อนำไปใช้งานต่อไป

อุปกรณ์ที่ทำหน้าที่แปลงข้อมูลแบบอนุกรมเป็นแบบขนาน และแปลงข้อมูลแบบขนานเป็นแบบอนุกรม เรียกว่า UART (Universal Asynchronous Receiver-Transmitter) ซึ่งเป็นวงจรรวม LSI ซึ่งนอกจากจะมีหน้าที่ในการเปลี่ยนแปลงข้อมูลแล้ว UART ยังมีหน่วยควบคุมและตรวจสอบการทำงานอีกด้วย

ในการขนส่งข้อมูลขนาด 8 บิตแบบอนุกรมนี้จะต้องมีบิตเริ่มต้น (Start Bit) และบิตหยุด (Stop Bit) เพิ่มขึ้นมาซึ่งจะทำให้ข้อมูลที่ถูกส่งไปจริงๆ มีขนาด 10 บิต ในรูปต่อไปนี้ แสดงข้อมูลที่มีขนาด 8 บิต บิตเริ่มต้น 1 บิต และบิตหยุด 1 บิต โดยที่บิตเริ่มต้นมีค่าเป็น 0 บอก UART ที่ทำหน้าที่รับข้อมูลให้รู้ว่าข้อมูลกำลังเข้ามาและบิตสตอปมีค่า 1 บอกให้รู้ว่าการส่งข้อมูลได้เสร็จสิ้นลงแล้ว



รูปที่ 4-6 แสดงเวิร์ดข้อมูลขนาด 8 บิตกับ บิตเริ่มต้นและบิตหยุด ที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

เราเรียกความเร็วของการส่งข้อมูลแบบ UART ว่าอัตราการส่งข้อมูล (Baud Rate) มีหน่วยเป็นจำนวนบิตต่อวินาที (Bit per Second) ซึ่งจะบอกจำนวนบิตที่รับส่งในเวลา 1 วินาที เช่นการส่งข้อมูลด้วยอัตราการส่งข้อมูล 1200 อัตรา ก็คือการส่งข้อมูลด้วยอักขระขนาด 10 บิต (บิตเริ่มต้น 1 บิต บิตข้อมูลที่มี

ขนาด 8 บิต และบิตหยุด 1 บิต) ได้ 120 ตัวอักษรใน 1 วินาที ซึ่งในตารางต่อไปนี้ แสดงอัตราการส่งข้อมูลของ UART ที่ใช้กันโดยทั่วไป

อัตราการส่งข้อมูล	ไบต์/วินาที
110	10
150	15
300	30
600	60
1200	120
2400	240
4800	480
9600	960
19200	1920
38400	3840

ตารางที่ 4-3 แสดงอัตราการส่งข้อมูลทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

การส่งข้อมูลด้วยอัตราการส่งข้อมูล 110 อัตรานั้น จะมีรูปแบบแตกต่างกับอัตราการส่งข้อมูลในอัตราอื่นๆ ซึ่งในอัตรานี้จะต้องใช้บิตเริ่มต้น 1 บิต และบิตหยุด 2 บิต ดังนั้นจึงทำให้ต้องส่งข้อมูลที่มีขนาด 11 บิต

เมื่อนำบิตที่ 8 ของข้อมูลมาใช้ตรวจสอบความผิดพลาดในการสื่อสารข้อมูล ซึ่งเรียกว่าบิตพาริตี (Parity Bit) UART ส่วนใหญ่สามารถสร้างและทำการตรวจสอบข้อมูลนั้นว่าเป็นพาริตีคู่หรือคี่ได้ ในการสร้างพาริตีคู่ UART จะทำการเซตหรือเคลียร์ค่าในบิตพาริตีเพื่อให้ข้อมูลทั้ง 8 บิต มีตัวเลข 1 เป็นจำนวนคู่ และในการสร้างพาริตีคี่ UART จะทำการเซตหรือเคลียร์ค่าในบิตพาริตีเพื่อให้ข้อมูลทั้ง 8 บิต มีตัวเลข 1 เป็นจำนวนคี่

เราสามารถใช่พาริตีคู่หรือคี่ในการตรวจสอบความผิดพลาดในการรับส่งข้อมูล โดยเมื่อ UART ได้รับความข้อมูลก็จะทำการทดสอบว่าข้อมูลที่ได้รับมานั้นเป็นพาริตีคู่หรือคี่ ถ้า UART ตรวจสอบพบว่าข้อมูลที่ได้รับมานั้นมีพาริตีไม่ตรงตามที่กำหนด บิตพาริตีในรีจิสเตอร์ของ UART จะถูกเซตเพื่อขอให้มีการส่งข้อมูลชุดเดิมนั้นกลับมาใหม่อีกครั้ง

เราสามารถแบ่ง UART ออกเป็นส่วนๆได้ 4 ส่วน ได้แก่

1. ส่วนที่ทำหน้าที่ส่งข้อมูล
2. ส่วนรับข้อมูล
3. ส่วนกำหนดสถานะ
4. ส่วนที่เป็นวงจรควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ทำหน้าที่ส่งข้อมูลของ UART แยกออกได้เป็น 2 ส่วน ซึ่งได้แก่บัฟเฟอร์ส่งข้อมูล (Transmitter data output buffer) กับรีจิสเตอร์ส่งข้อมูล (Transmitter register) โดยบิตข้อมูล 8 บิตจะถูกนำไปเก็บที่บัฟเฟอร์ส่งข้อมูล เมื่อสัญญาณควบคุมค่าได้อินพุตสโตรบ (Data input strobe) เปลี่ยนค่าจาก 1 เป็น 0 และการส่งข้อมูลแบบอนุกรมจะเริ่มขึ้นเมื่อสัญญาณนี้เปลี่ยนค่าจาก 0 เป็น 1 หลังจากนั้น รีจิสเตอร์ส่งข้อมูลจะทำหน้าที่เลื่อนข้อมูลส่งออกไปยังเส้นส่งข้อมูลอนุกรม โดยเริ่มจากบิตเริ่มต้นถัดมาเป็นบิตข้อมูล D0 ถึง D7 และบิตหยุด

ส่วนที่ทำหน้าที่รับข้อมูลของ UART จะมีการทำงานที่ตรงกันข้ามกับส่วนที่ทำหน้าที่ส่งข้อมูล ข้อมูลที่เข้ามาทางเส้นรับข้อมูลอนุกรม (เข้า) จะถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูล โดยการเลื่อนค่า 10 หรือ 11 ครั้ง การทำงานจะเริ่มเมื่อบิตเริ่มต้นเข้ามา และเมื่อข้อมูลทั้งหมดถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูล (Receiver Register) แล้ว ข้อมูลในรีจิสเตอร์รับข้อมูลจะถูกนำไปเก็บบัฟเฟอร์รับข้อมูล (Receive-data Output Buffer) เมื่อมีสัญญาณควบคุมค่าได้อินพุตสโตรบ ทั้งรีจิสเตอร์รับข้อมูลและรีจิสเตอร์ส่งข้อมูลจะได้รับสัญญาณนาฬิกาขนาด 16 หรือ 64 เท่าของอัตราการส่งข้อมูลที่ใช้ในการเลื่อนค่ารีจิสเตอร์ ทั้งรีจิสเตอร์รับข้อมูลและส่งข้อมูลจะได้รับสัญญาณนาฬิกาขนาด 16 หรือ 64 เท่าของอัตราการส่งข้อมูลที่ใช้ในการเลื่อนค่ารีจิสเตอร์

รีจิสเตอร์ควบคุมการทำงานของ UART ทำให้เราสามารถกำหนดโหมดการทำงานของ UART ได้ โดยบิตควบคุมจำนวนข้อมูลทั้ง 2 บิตจะช่วยให้เราสามารถเลือกจำนวนของข้อมูลจริงที่รับมาว่าเป็น 5,6,7 หรือ 8 บิต ก็ได้

ในปัจจุบัน UART ถูกนำมาพร้อมกับไอซีพิเศษที่ทำหน้าที่สร้างสัญญาณอัตราการส่งข้อมูล โดยไอซีนี้จะสามารถสร้างสัญญาณที่มีอัตราการส่งข้อมูลเป็น 16 หรือ 64 เท่าของอัตราการส่งข้อมูลมาตรฐาน โดยสัญญาณนี้จะถูกสร้างจากวงจรกำเนิดสัญญาณแบบคริสตอล ทำให้ได้อัตราการส่งข้อมูลที่มีอัตราคงที่และเที่ยงตรงมาก

4.2.5 MCS - 51 กับการส่งข้อมูลแบบอนุกรม

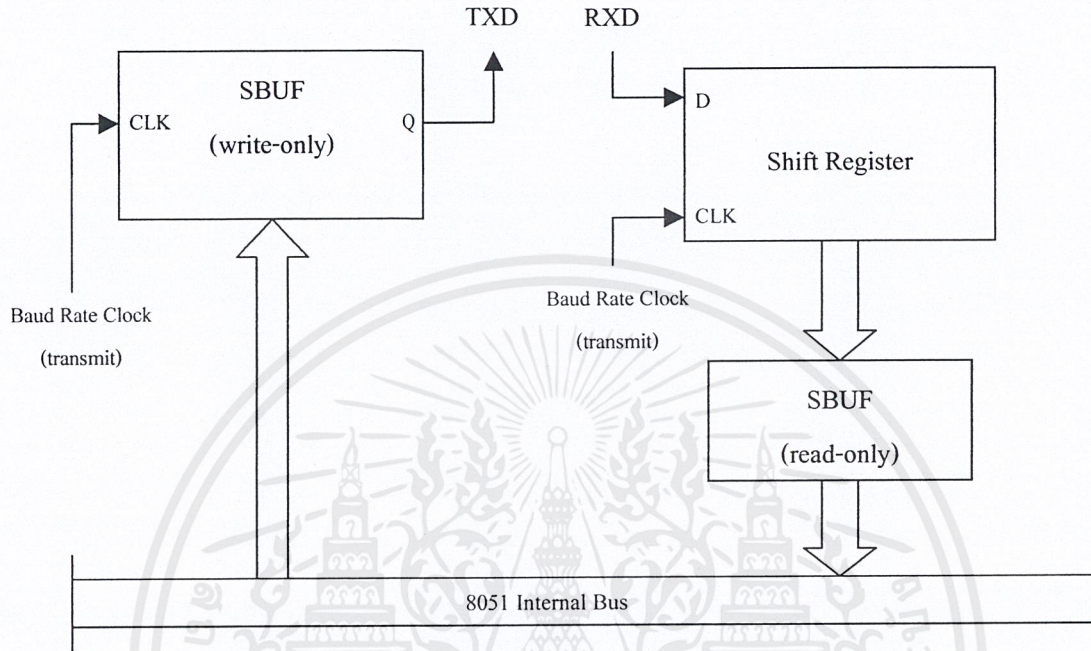
การรับส่งข้อมูลแบบอนุกรมกับไมโครคอนโทรลเลอร์ MCS-51 นั้นภายในชิพ MCS-51 จะมี UART อยู่ในตัวซึ่งเป็นข้อดีของไมโครคอนโทรลเลอร์ ถ้าเป็นไมโครโพรเซสเซอร์ เช่น เบอร์ Z-80 ต้องการส่งข้อมูลแบบอนุกรมจะต้องนำชิพ UART มาประกอบกับพอร์ตอนุกรมของ MCS-51 ซึ่งจะใช้เวลา TXD และ RXD ในการรับส่งข้อมูล โดยขาทั้งสองข้างจะอยู่ในพอร์ต 3 คือ P3.1 หรือขา 11 เป็น TXD และ P3.0 หรือขา 10 เป็น RXD พอร์ตอนุกรมของ MCS-51 สามารถทำงานแบบ Full Duplex ได้ คือสามารถส่งและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูล จะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

รีจิสเตอร์ที่สำคัญในการรับส่งข้อมูลคือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน Special Function Registers โดยรีจิสเตอร์ Serial Port Buffer (SBUF) จะอยู่ตำแหน่งที่ 99H ถ้าเขียนข้อมูลลงไปตำแหน่งนี้ จะเป็นการส่งข้อมูลออกทางพอร์ตอนุกรม และถ้าอ่านข้อมูลออกจากตำแหน่งนี้ จะเป็นการรับข้อมูลจากพอร์ตอนุกรม โดยใน จะประกอบด้วยบัฟเฟอร์สองตัวสำหรับส่งและรับข้อมูลคั้งรูปที่ 4-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Serial Port Control Register (SCON) ซึ่งอยู่ในตำแหน่ง 98H จะเป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลในระดับบิตได้ รีจิสเตอร์นี้จะทำหน้าที่ควบคุมและบอกสถานะต่างๆของการรับส่งข้อมูลแบบอนุกรม สำหรับความเร็วของการส่งข้อมูล (Baud Rate) สามารถหาได้จากตารางหาสัญญาณนาฬิกาที่ใช้กับ MCS-51



รูปที่ 4-7 แสดงการรับส่งข้อมูลระหว่างรีจิสเตอร์กับบัฟภายใน

4.2.6 Serial Port Control Register

MCS-51 มีโหมดการทำงานของพอร์ตอนุกรมหลายโหมด ซึ่งสามารถโปรแกรมโหมดการทำงานได้โดยเขียนข้อมูลไปยังรีจิสเตอร์ SCON โดยความหมายของแต่ละบิตและโหมดการทำงานต่างๆ จะแสดงดังตารางต่อไปนี้

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.7	SM0	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON.6	SM1	9EH	บิตเลือกโหมดการทำงานบิต 1
SCON.5	SM2	9DH	บิตเลือกโหมดการทำงานบิต 2
SCON.4	REN	9CH	บิตแฟลคกำหนดยอมให้มีการรับข้อมูล
SCON.3	TB8	9BH	ค่าของบิต 9 สำหรับส่งข้อมูลในโหมด 2 และ 3 สามารถตั้งค่าและเคลียร์ได้ โดยใช้ ซอฟต์แวร์
SCON.2	RB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา

ตารางที่ 4-4 แสดงบิตต่างๆ ของรีจิสเตอร์

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.1	TI	99H	บิตแฟลคแสดงการอินเทอร์รัพท์ ภายหลังจากส่งข้อมูลออกไป โดยจะตั้งค่าเมื่อส่งข้อมูลออกไปหมดแล้ว และสามารถเคลียร์ด้วยการใช้ซอฟต์แวร์
SCON.0	RI	98H	บิตแฟลคแสดงการอินเทอร์รัพท์ ภายหลังรับข้อมูลเข้ามา สามารถเคลียร์ได้ด้วยซอฟต์แวร์

ตารางที่ 4-4 (ต่อ) แสดงบิตต่างๆ ของรีจิสเตอร์

SM0	SM1	MODE	ความหมาย	BAUD RATE
0	0	0	Shift Register	เปลี่ยนแปลงไม่ได้
0	1	1	8-bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจาก Timer
1	0	2	9-bit UART	เปลี่ยนแปลงไม่ได้
1	1	3	9-bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจาก Timer

ตารางที่ 4-5 แสดงโหมดต่างๆของการรับส่งแบบอนุกรม

ก่อนที่เราจะใช้พอร์ตอนุกรมจะต้องโปรแกรมให้กับ SCON เสียก่อน เพื่อกำหนดโหมดการทำงานในลักษณะต่างๆ เช่น

MOV SCON,#01010010 B

เป็นการกำหนดให้พอร์ตอนุกรมทำงานในโหมด 1 และกำหนดให้มีการรับข้อมูล พร้อมทั้งกำหนด TI ให้เป็น 1

ในการส่งข้อมูลทุกโหมดสามารถทำได้โดยเขียนข้อมูลไปยัง SBUF เมื่อข้อมูลถูกส่งไปแล้ว บิต TI จะถูกตั้งค่าเป็น 1 ในการส่งข้อมูล จะต้องคอยตรวจสอบบิต TI ถ้า TI ยังไม่เปลี่ยนเป็น 1 แสดงว่ายังส่งข้อมูลไปไม่หมด หากมีการเขียนข้อมูลไปต่อกันไป ไปยัง SBUF จะทำให้เกิดข้อผิดพลาดขึ้น สำหรับการรับข้อมูล บิต REN จะต้องตั้งค่าให้เป็น 1 ยกเว้นโหมด 0 เพื่ออนุญาตให้รับข้อมูลได้ เมื่อข้อมูลรับเข้ามาเรียบร้อยแล้วบิต RI จะถูกตั้งค่าเป็น 1

4.2.7 Mode Of Operation

ใน MCS-51 การสื่อสารทางพอร์ตอนุกรมจะมีอยู่ 4 โหมดโดยจะกำหนดที่บิต SM0 และ SM1 ใน SCON โดยจะมี 3 โหมดเป็นการสื่อสารแบบอะซิงโครนัส (Asynchronous) จะมีบิตเริ่มต้น (Start bit) และบิตจบ (Stop bit) คล้ายการสื่อสารแบบ RS-232 ในระบบคอมพิวเตอร์ อีกโหมดหนึ่งจะเป็นการใช้พอร์ตอนุกรมในลักษณะซีพรีจิสเตอร์

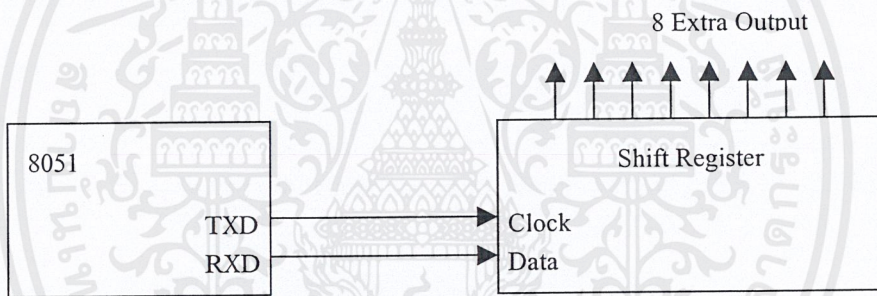
4.2.7.1 8-Bit Shift Register (Mode 0)

การทำงานในโหมดนี้จะใช้ขา RXD ในการรับส่งข้อมูลโดยต่อกับ Shift Register ภายนอก ส่วนขา TXD จะเป็น Output Shift Clock เพื่อกระตุ้นรีจิสเตอร์ภายนอกให้เลื่อนบิต ถ้ามีการส่งข้อมูลหรือรับข้อมูล 8 บิต จะเริ่มที่บิตต่ำสุดก่อน โดยมีค่าอัตราการส่งข้อมูลเท่ากับ 1/12 ของความถี่ที่ใช้บนชิพ

ในการส่งข้อมูลจะทำได้โดยเขียนข้อมูลลงไปทีรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งออกมาทางขา RXD โดยจะสอดคล้องกับสัญญาณที่ออกมาจากขา TXD ซึ่งสัญญาณที่ถูกส่งออกมาจากขา TXD นี้จะส่งออกมาทุกๆ Machine Cycle โดยจะเป็นลอจิก 0 ใน S3P1 และจะกลับเป็นลอจิก S6P1

สำหรับการรับข้อมูลจะรับได้เมื่อตั้งค่าขา Receiver Enable Bit (REN) เป็น 1 และ เคลียร์ขา Receiver Interrupt Bit (RI) เป็น 0 ข้อมูลจะเข้าสู่ MCS-51 เมื่อ Clock Shift ถูกส่งออกไปทาง TXD ที่ขอบขาขึ้นของ Clock Shift บิตต่ำจะถูกส่งเข้ามาก่อน

ในการประยุกต์ใช้งานในโหมดนี้ จะต้องมี IC Shift Register มาต่อภายนอก เช่น หากต้องการส่งข้อมูลออกทางพอร์ตอนุกรม อาจต้องจรได้ดังรูปด้านล่างนี้ โดยใช้ไอซี Serial-to-parallel Shift Register ช่วย โดยข้อมูลจะส่งออกมาทาง RXD และ TXD เป็น clock



รูปที่ 4-8 แสดงการส่งข้อมูลออกโดยใช้รีจิสเตอร์ช่วย

4.2.7.2 8-Bit UART with Variable Baud Rate (Mode 1)

ในโหมดนี้จะเป็นการส่งข้อมูลแบบ 10 บิต ประกอบด้วยบิตเริ่มต้น (0) ข้อมูล 8 บิต และบิตจบ (1) นอกจากนี้ยังสามารถกำหนดค่าอัตราการส่งข้อมูล ได้โดยค่าอัตราการส่งข้อมูลนี้จะแปรตามตัวจับเวลาตัวที่ 1 ในโหมดนี้จะส่งข้อมูลออกทาง TXD และรับเข้ามาทาง RXD ถ้าเป็นการรับข้อมูลเข้าตัวของบิตหยุดจะเข้ามายังบิต RB8 ใน SCON

ถ้าอัตราการส่งข้อมูลที่ใช้ในการรับส่งข้อมูลจะกำหนดโดย Timer 1 หลังจากโปรแกรมไปใน Timer 1 แล้วสามารถเลือกค่าอัตราการส่งข้อมูลได้อีกสองค่าคือ ค่าจาก Timer 1 Overflowหาร 32 กับค่าจาก Timer 1 Overflowหาร 16

การส่งข้อมูลทำได้โดยการเขียนข้อมูล 8 บิต ไปที่ SBUF โดยบิตที่ 9 (บิตหยุด) ให้เขียนลงใน TB8 ใน SCON จากนั้น ข้อมูลจะถูกส่งออกมาทางขา TXD โดยส่งบิตเริ่มต้นออกมาก่อน ตามด้วยข้อมูล 8 บิต และจบด้วยบิตหยุด เมื่อข้อมูลถูกส่งออกไปหมดแล้วบิต Interrupt Flag (TI) จะเป็น 1 ดังนั้นก่อนการเขียนข้อมูลใหม่ลงไปจะต้องตรวจสอบบิตนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับข้อมูล จะเริ่มจากเมื่อมีการเปลี่ยนลอจิกจาก 1 เป็น 0 ที่ขา RXD หมายความว่าเริ่มรับบิตเริ่มต้น จากนั้นข้อมูลอีก 8 บิตจะถูกเก็บลงใน SUBF และ บิตหยุดจะถูกเก็บลงใน RB8 ของรีจิสเตอร์ SCON เมื่อข้อมูลเข้ามาครบแล้ว บิต Interrupt Flag (RI) จะถูกตั้งค่า ดังนั้นการอ่านบิตข้อมูลจะทำได้หลังจากบิต RI ถูกตั้งค่าแล้ว เมื่ออ่านข้อมูลไปแล้วจะต้องเคลียร์บิตนี้

4.2.7.3 9-Bit UART with Fixed Baud Rate (Mode 2)

การทำงานในโหมดนี้ ไม่สามารถกำหนดค่าในอัตราการส่งข้อมูลได้ ซึ่งค่าอัตราการส่งข้อมูลจะมีสองค่าคือ 1/64 และ 1/32 ของสัญญาณนาฬิกาบนชิพ การรับส่งข้อมูลจะเป็นชุดข้อมูล 9 บิต รวมบิตเริ่มต้นและหยุดเป็น 11 บิต โดยในข้อมูล 9 บิต จะเป็นข้อมูลจริงๆเพียง 8 บิต ส่วนอีก 1 บิตที่เหลือคือบิตที่สามารถโปรแกรมได้ ซึ่งจะใช้เป็นพาริตีบิต ในการส่งข้อมูลจะต้องเขียนไปที่บิต TB8 ในรีจิสเตอร์ SCON สำหรับการรับข้อมูลบิตที่ 9 จะถูกเก็บในบิต RB8

4.2.7.4 9-Bit UART with Variable Baud Rate (Mode 3)

การทำงานของโหมดนี้จะคล้ายกับโหมด 2 ต่างกันตรงที่สามารถกำหนดอัตราการส่งข้อมูลได้ โดยการโปรแกรมไปที่ Timer 1 หลังจากโปรแกรมแล้วยังจะสามารถเลือกได้อีก 2 ค่าคือ ความถี่การ Overflow ของ Timer 1 ทหารด้วย 32 หรือ 16

4.2.8 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล

การรับข้อมูลถ้าจะให้ MCS-51 รับข้อมูลทางพอร์ตอนุกรมจะต้องโปรแกรมไปที่บิต Receiver Enable (REN) ในรีจิสเตอร์ SCON ให้เป็นลอจิก 1 ซึ่งทำได้ 2 วิธีดังนี้

SETB REN

เป็นคำสั่งในการตั้งค่า บิต REN ให้เป็น 1 หรืออาจทำได้โดยใช้คำสั่ง

MOV SCON,#xxx1xxxxB

ซึ่งเป็นการย้ายข้อมูลที่ทำให้บิต REN เป็น 1 สำหรับค่า x หมายความว่า เป็นอะไรก็ได้ขึ้นกับการใช้งานในโหมดต่างๆ

- ข้อมูลแบบ 9 บิต ในการรับส่งข้อมูลที่มีบิตข้อมูลแบบ 9 บิต ได้แก่ การใช้งานในโหมด 2 และ โหมด 3 ซึ่งการส่งข้อมูลบิตที่ 9 จะถูกเขียนใน TB8 โดยการเขียนโปรแกรมสำหรับการรับข้อมูล เมื่อข้อมูลเข้ามาถึงบิตที่ 9 จะถูกเขียนลงในบิต RB8

การเพิ่มพาริตีบิต การส่งข้อมูลแบบ 9 บิต สามารถใช้บิตที่ 9 เป็นพาริตีบิตได้ ซึ่งพาริตีบิตจะอยู่ใน Program Status Word (PSW) โดยจะถูกตั้งค่าหรือเคลียร์ทุกๆ Machine Cycle ที่เกี่ยวข้องกับ Accumulator เช่น ถ้าจะส่งข้อมูลแบบ 8 บิต ตามด้วย Even Parity bit เป็นบิตที่ 9 สามารถเขียนโปรแกรมได้ดังนี้

MOV C, P	;	อ่านค่าบิต P มาเก็บใน C
MOV TB8, C	;	นำค่าพาริตีบิตมาเขียนลงใน TB8
MOV SBUF, A	;	ส่งข้อมูลไปทางพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเป็นแบบ Odd Parity ให้แก้ไขข้อมูลที่อ่านได้จากพาริตีบิต เสียก่อนที่จะส่งออกไป ซึ่งเขียนโปรแกรมได้ดังนี้

```
MOV C, P           ; อ่านค่าบิต P มาเก็บใน C
CLP C             ; กลับค่าให้เป็น Odd Parity
MOV TB8, C        ; เขียนค่าลงใน TB8
MOV SBUF, A       ; ส่งข้อมูลไปทางพอร์ตอนุกรม
```

การส่งข้อมูลแบบมีพาริตีบิตด้วย สามารถส่งได้ในโหมด 1 ซึ่งส่งข้อมูลแบบ 8 บิต ได้ด้วย เช่น การส่งรหัส ASCII จะใช้บิตข้อมูล 7 บิต ส่วนบิตที่เหลืออีก 1 บิตนั้นจะเป็นพาริตีบิต สามารถเขียนโปรแกรมได้ดังนี้

```
CLR ACC.7         ; เคลียร์ค่าบิต 7 เพื่อใช้เป็นพาริตีบิต
MOV C, P          ; อ่านค่าบิต P มาเก็บใน C
MOV TB8, C        ; นำค่าบิต Parity มาเขียนลงใน TB8
MOV SBUF, A       ; ส่งข้อมูลไปทางพอร์ตอนุกรม
```

แฟลกอินเทอร์รัพท์ เมื่อมีการรับส่งข้อมูลเสร็จสิ้นจะมีผลต่อแฟลกอินเทอร์รัพท์ (RI, TI) ในรีจิสเตอร์ SCON ซึ่งบิตเหล่านี้จะถูกตั้งค่าโดย ฮาร์ดแวร์แต่ต้องเคลียร์ โดย ซอฟต์แวร์

บิต RI ถ้าถูกตั้งค่า หมายความว่าบัพเฟอร์ที่ใ้รับข้อมูลเต็มให้อ่านไปได้แล้ว และบิตนี้สามารถใช้อินเทอร์รัพท์ MCS-51 ได้ แต่ถ้าเขียนโปรแกรม จะใช้วิธีตรวจเช็คบิตนี้เป็น 1 หมายความว่าให้อ่านข้อมูลมาเก็บในรีจิสเตอร์ A ได้ แต่ก่อนอ่านต้อง เคลียร์ RI เสียก่อน เพื่อที่จะได้รับข้อมูลถัดไปซึ่งเขียนโปรแกรมได้ดังนี้

```
WAIT: JNB RI, WAIT ; ถ้าไม่เป็น 1 จะทำงานอยู่ที่เดิม
CLR RI             ; เคลียร์ RI
MOV SBUF, A        ; อ่านค่ามาเก็บใน A
```

บิต TI เมื่อส่งข้อมูลออกไปแล้ว บิตนี้จะถูกตั้งค่าเป็นการแจ้งว่าบัพเฟอร์ส่งข้อมูลว่างแล้ว ให้ส่งข้อมูลใหม่เขาไปได้ ซึ่งสามารถใช้บิตอื่นเทอร์รัพท์ MCS-51 เช่นกัน ถ้าจะเขียนโปรแกรมคอยตรวจเช็ค อาจเขียนนำได้ดังนี้

```
WAIT: JNB TI, WAIT ; ตรวจสอบว่า TI เป็น 1 แล้วหรือยัง
CLR TI             ; เคลียร์ TI
MOV SBUF, A        ; เขียนข้อมูลลงไป
```

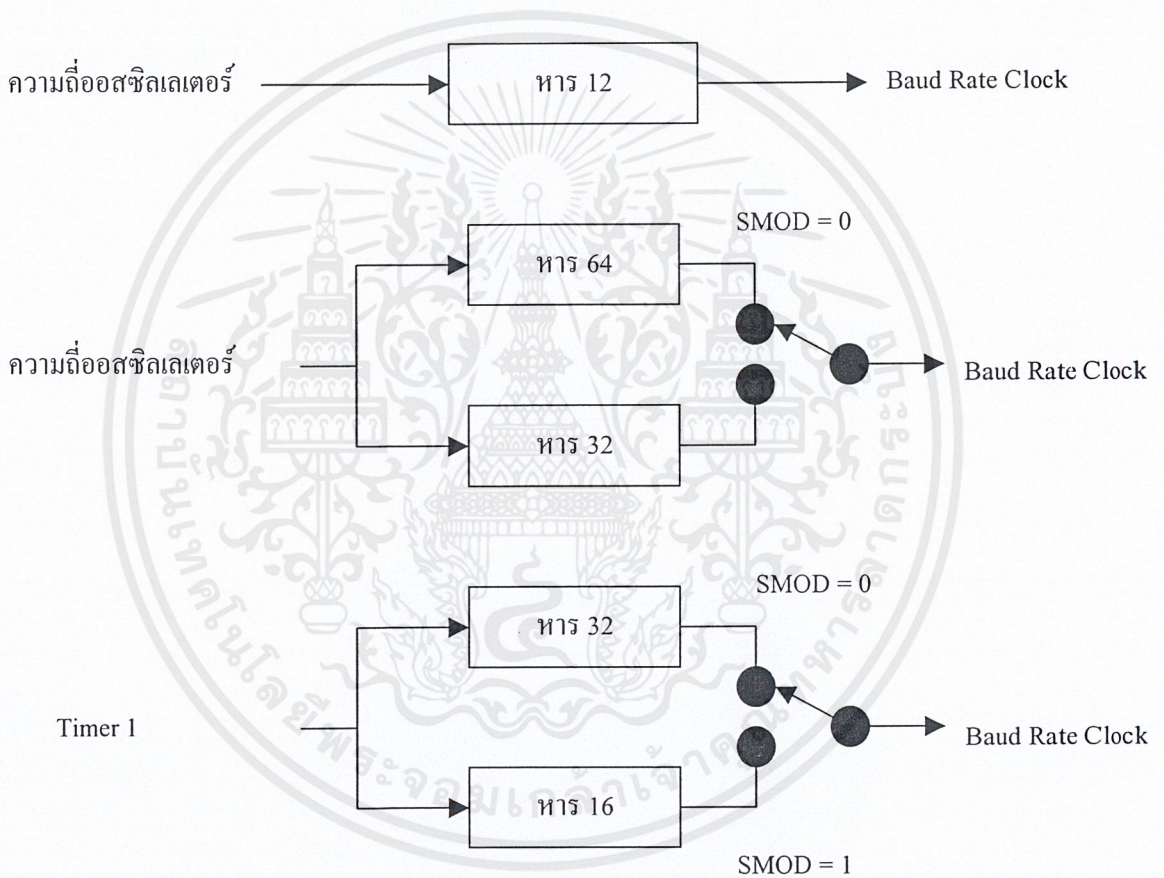
4.2.9 อัตราการส่งข้อมูลทางพอร์ตอนุกรม

จากการศึกษาการส่งข้อมูลข้อมูลในโหมดต่างๆ พบว่าโหมด 0 และ 2 ไม่สามารถกำหนดอัตราการส่งเองได้ โดยในโหมด 0 ค่าอัตราการส่งจะมีค่าเท่ากับความถี่ของ Oscillator หารด้วย 12 ในโหมด 1 จะมีสองค่าคือ ความถี่ Oscillator หารด้วย 32 และหารด้วย 64 สองค่านี้เรียกว่า SMOD0 และ SMOD1 ซึ่งสามารถกำหนดได้จากรีจิสเตอร์ PCON บิตที่เจ็ด ในรีจิสเตอร์นี้ไม่สามารถเข้าถึงข้อมูลในระดับบิตได้ ถ้าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนข้อมูลลงไปทีละบิตต้องใช้วิธี Read-Modify-Write คืออ่านค่าขึ้นมาแล้วแก้ไขเขียนลงไปใหม่ อย่างเช่น

```
MOV A,PCON      ; อ่านค่าจาก PCON มาเก็บใน رجิสเตอร์ A
SETB ACC.7      ; เซ็ตบิต 7 (SMOD)
MOV PCON,A      ; เขียนค่าลงไปใหม่ใน PCON
```

สำหรับโหมด 1 และโหมด 3 สามารถกำหนดค่าอัตราการส่งได้ด้วยวิธีการโปรแกรมลงใน Timer 1 ในการโปรแกรมแต่ละครั้ง จะมี SMOD สองค่าเช่นกัน ค่าอัตราการส่งของโหมดต่างๆแสดงได้ดังรูปต่อไปนี้



รูปที่ 4-9 แสดงการกำหนดอัตราการส่งข้อมูลในโหมดต่างๆ

- การใช้ Timer 1 กำหนด Baud Rate Clock

การกำหนดค่าลงใน Timer 1 ทำได้โดยการโปรแกรมไปที่ TMOD ให้ทำงานแบบ 8-bit Auto Reload Mode (Mode 2) โดยเขียนค่าไปที่ TH1 ซึ่งสามารถโปรแกรมลงไปที่รีจิสเตอร์ TMOD ได้ดังนี้

```
MOD TMOD, #0010xxxxB   (ค่า x เป็นค่าอะไรก็ได้เพราะบิตเหล่านี้ใช้ใน Timer 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการให้อัตราการส่งข้อมูลต่างๆเราก็สามารถใช้ 16-bit Mode ได้ โดยโปรแกรมเข้าไปเป็น TMOD = 0001xxxxB ค่าของอัตราการส่งข้อมูลที่ส่งออกมาจะมีค่าเท่ากับความถี่ของ Timer 1 เกิด Overflow หารด้วย 32 (หรือ 16 ถ้าเป็น SMOD = 1)

รูปแบบทั่วไปของการหาค่าอัตราการส่งข้อมูลในโหมด 1 และ 3 สามารถหาได้ดังนี้

$$\text{Baud Rate} = \text{Timer 1 Overflow Rate}/32$$

ถ้าต้องการอัตราการส่งข้อมูลเท่ากับ 1200 สามารถคำนวณค่าความถี่ Overflow ของ Timer 1 ได้ดังนี้

$$1200 = \text{Timer 1 Overflow} / 32$$

$$\text{จะได้ Timer 1 Overflow} = 38.4 \text{ KHz}$$

ถ้าระบบ MCS-51 ใช้ความถี่สัญญาณนาฬิกาจาก Crystal เท่ากับ 12 MHz ตัว Timer 1 จะได้รับ clock เท่ากับ 1 MHz หรือ 100 KHz ถ้าเราต้องการ Timer 1 Overflow เท่ากับ 38.4 KHz ดังนั้นค่าอัตรา Overflow มีค่าเท่ากับ $1000 / 38.4 = 26.04$ Clock โดยค่า Overflow จะเกิดเมื่อมีการเปลี่ยนจาก FFH เป็น 00H ดังนั้นจะต้องให้ Timer 1 นับไป 26 Count ดังนั้น ค่าที่จะให้รีจิสเตอร์ TH1 มีค่าเท่ากับ -26 ซึ่งใช้เป็นค่า Reload จึงเขียนคำสั่งได้ดังนี้

```
MOV TH1, #-26
```

ตัวโปรแกรมแอสเซมบลอร์ทั่วไปจะแปลงค่า -26 เป็น 0E6H เอง จากที่ผ่านมาจะเห็นว่าความถี่อัตราการส่งข้อมูล จะมีความสัมพันธ์กับค่าสัญญาณนาฬิกาจาก Crystal ในตารางด้านล่างนี้เป็นค่าที่ต้องกำหนดใน Timer 1 เมื่อต้องการอัตราการส่งข้อมูลที่ต่างๆ กัน

ค่า Baud Rate	Crystal	SMOD โหมด	ค่าใน TH1	ค่า Baud Rate ที่ได้	ผิดพลาด
9600	12.000	1	-7 (F9H)	8923	7%
2400	12.000	0	-13 (F3H)	2404	0.16%
1200	12.000	0	-16 (E6H)	1202	0.16%
13200	11.059	1	-3 (FDH)	19200	0
9600	11.059	0	-3 (FDH)	9600	0
2400	11.059	0	-12 (F4H)	2400	0
1200	11.059	0	-24 (E8H)	1200	0

ตารางที่ 4-6 แสดงความถี่สัญญาณนาฬิกาที่ใช้กำหนดอัตราการส่งข้อมูลของค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.10 การเชื่อมต่อหน่วยความจำกับ MCS-51

การจัดหน่วยความจำสำหรับ MCS-51 แบ่งได้เป็น 2 ส่วนใหญ่ๆ ได้แก่ Data Memory และ Program Memory

4.2.10.1 Data Memory

เป็นหน่วยความจำที่ใช้เก็บข้อมูล มีความจุได้ถึง 64 กิโลไบต์ เป็นหน่วยความจำประเภทอ่านและเขียนได้ (RAM) สัญญาณจาก MCS-51 ที่ใช้ในการอ่านข้อมูลจาก RAM คือ RD \setminus และสัญญาณที่ใช้ในการเขียนข้อมูลลง RAM คือ WR \setminus

- *RAM (Random Access Memory)*

เป็นหน่วยความจำที่สามารถเขียนหรืออ่านข้อมูลลงไปได้ ข้อมูลจะสูญหายทันทีที่ขาดไฟเลี้ยง แบ่งออกเป็น 2 ชนิดคือ Static RAM และ Dynamic RAM โดย Static RAM เป็นหน่วยความจำที่ง่ายต่อการนำมาใช้งานแต่จะราคาสูงเมื่อเทียบกับ Dynamic RAM แต่จะนำใช้งานง่ายกว่า Static RAM เพราะไม่ต้องคอย refresh หน่วยความจำตลอดเวลาเหมือน Dynamic RAM

- *Static RAM*

เป็นหน่วยความจำชนิดหนึ่งที่ยากต่อการนำมาใช้งาน แต่ราคาสูงเมื่อเทียบกับ Dynamic RAM แต่ Dynamic RAM จะยุ่งยากขึ้นในการใช้งาน คือจะต้องทำการรีเฟรชหน่วยความจำตลอดเวลา ส่วน Static RAM ซึ่งมีใช้กันอย่างแพร่หลายมีอยู่ด้วยกันหลายเบอร์ ตามขนาดความจุของหน่วยความจำ

4.2.10.2 Program Memory

หน่วยความจำที่ใช้เก็บโปรแกรม เป็นหน่วยความจำซึ่งสามารถอ่านข้อมูลได้เพียงอย่างเดียว (ROM) โดยสัญญาณจาก MCS-51 ที่ใช้ในการอ่าน ROM คือ PSEN \setminus

- *ROM (Read Only Memory)*

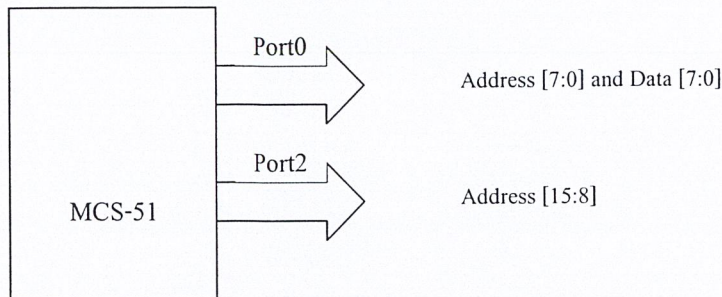
คือหน่วยความจำแบบถาวร สามารถรักษาข้อมูลไว้ได้โดยไม่ต้องใช้ไฟเลี้ยง ไม่สามารถเขียนข้อมูลลงไปได้สามารถอ่านข้อมูลได้เพียงอย่างเดียว

- *EPROM (Erasable Programmable read-only Memory)*

เป็นหน่วยความจำแบบถาวรชนิดหนึ่ง ซึ่งผู้ใช้นำมาประกอบเองได้ และสามารถใช้แสง UV ลบข้อมูลได้

4.2.10.3 การเชื่อมต่อหน่วยความจำกับ MCS-51

การเชื่อมต่อกับหน่วยความจำจะต้องใช้สัญญาณหลายชนิดประกอบด้วย แอดเดรสบัส บัสข้อมูล และบัสควบคุมใน MCS-51 สามารถติดต่อกับสัญญาณภายนอกได้ 64 กิโลไบต์ โดยจะใช้สายสัญญาณแอดเดรสบัส 16 เส้น ส่งออกมาทางพอร์ต 0 และ 2 โดยพอร์ต 0 จะใช้ Multiplex ระหว่างแอดเดรสบัสและบัสข้อมูล โดยแอดเดรสบัสจะเป็นบิตต่ำ A0-A7 ดังรูปด้านล่าง

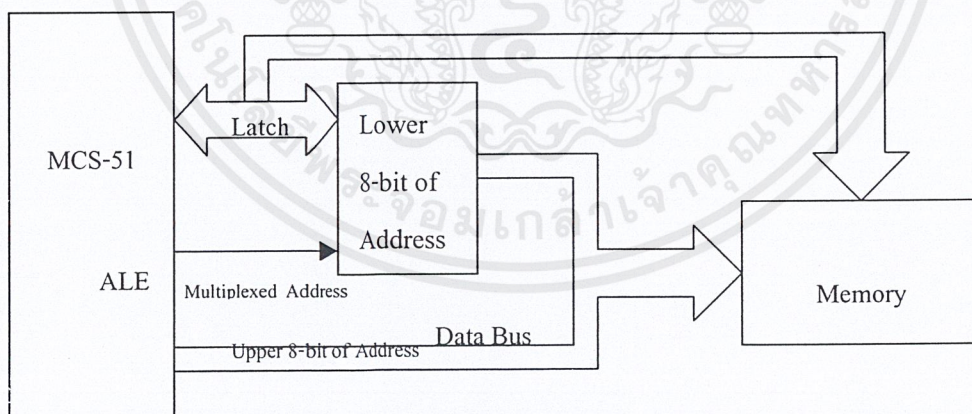


รูปที่ 4-10 แสดงสัญญาณของ MCS-51 ที่ใช้ติดต่อกับตำแหน่งของหน่วยความจำ

ในการเชื่อมต่อกับหน่วยความจำจะต้องมีอุปกรณ์ภายนอกมา Latch สัญญาณแอดเดรสบิตที่ได้จากพอร์ต 0 เพื่อที่จะใช้พอร์ต 0 เป็นบัสข้อมูลต่อไป ในรูปต่อไปนี้จะเป็นการต่อหน่วยความจำประเภท ROM และ RAM กับ MCS-51 โดยมีอุปกรณ์ภายนอกมา Latch ค่าแอดเดรสไบต์ต่ำเอาไว้เรียกว่า “Address Latching”

อุปกรณ์ที่นิยมใช้ได้แก่ ไอซี TTL เบอร์ 74LS373 โคนสัญญาณที่ใช้ Latch คือสัญญาณ ALE จาก MCS-51

การอ่านหน่วยความจำโปรแกรมและการอ่านเขียนหน่วยความจำข้อมูล การเชื่อมต่อขาแอดเดรสบิตและบัสข้อมูลจะเหมือนกัน แม้หน่วยความจำทั้งสองอยู่ในตำแหน่งเดียวกัน แต่จะแยกด้วยสัญญาณควบคุม



รูปที่ 4-11 แสดงการต่อ MCS-51 กับหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

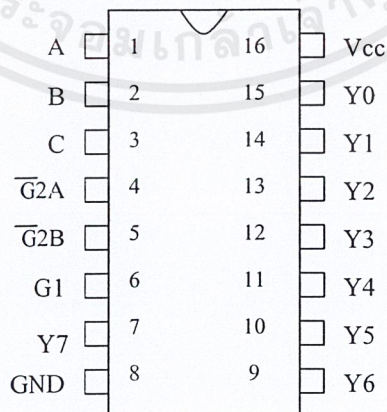
4.2.11 การถอดรหัสตำแหน่งของหน่วยความจำ

หน่วยความจำที่ถูกนำมาใช้ใน MCS-51 มีขนาดเล็กกว่า 64 กิโลไบต์ ดังนั้นจึงต้องนำมาเรียงกันแล้ว ถอดรหัสของหน่วยความจำแยกออกเป็นแต่ละช่วง ถ้าใช้หน่วยความจำขนาด 4 กิโลไบต์ จำนวน 8 ตัวจะ ได้เท่ากับ 32 กิโลไบต์ ถ้าให้ตัวแรกเริ่มที่แอดเดรส 8000H ตัวที่สองก็จะเริ่มที่แอดเดรส 9000H ดังรายละเอียดในตารางด้านล่าง

EPROM ตัวที่	แอดเดรส	A15	A14	A13	A12	A11-----A0
1	8000-8FFF	1	0	0	0	X-----X
2	9000-9FFF	1	0	0	1	X-----X
3	A000-AFFF	1	0	1	0	X-----X
4	B000-BFFF	1	0	1	1	X-----X
5	C000-CFFF	1	1	0	0	X-----X
6	D000-DFFF	1	1	0	1	X-----X
7	E000-EFFF	1	1	1	0	X-----X
8	F000-FFFF	1	1	1	1	X-----X

ตารางที่ 4-7 แสดงรหัสตำแหน่งหน่วยความจำขนาด 4 กิโลไบต์

การถอดรหัสตำแหน่งของหน่วยความจำจะใช้ IC to 8-line Decoder เบอร์ 74LS138 ดังมีการจัดวางขา ดังรูป



รูปที่ 4-12 แสดงการจัดวางขาของ 74LS138

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Enable Inputs		Select Input			Outputs							
G1	G2	C	B	D	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

ตารางที่ 4-8 แสดงค่าความจริงของ 74LS138

เมื่อพิจารณาจากตารางความจริงของ 74LS138 พบว่าจะต้องทำให้ขา G1 เป็น high และ G2A/ และ G2B/ เป็น low แล้วนำอินพุตเข้าทางขา A,B,C ซึ่งอินพุตนี้ก็คือแอดเดรสของ MCS-51 นั่นเอง จากตารางที่ 1 พบว่าตารางที่ 1 เป็น high ตลอดให้นำมาต่อที่ขา G1 ได้เลย และจากตารางความจริงของ 74LS138 พบว่าขา A,B,C ทั้งสามขานี้เป็นศูนย์ทั้งหมด ขา Y0 จะแอกทีฟ low ดังนั้นเมื่อพิจารณาจากตารางที่ 1 พบว่า A15,A14,A13 เป็นศูนย์ทั้งหมดซึ่งตรงกันพอดี ดังนั้นจึงได้การติดต่อกันอย่างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

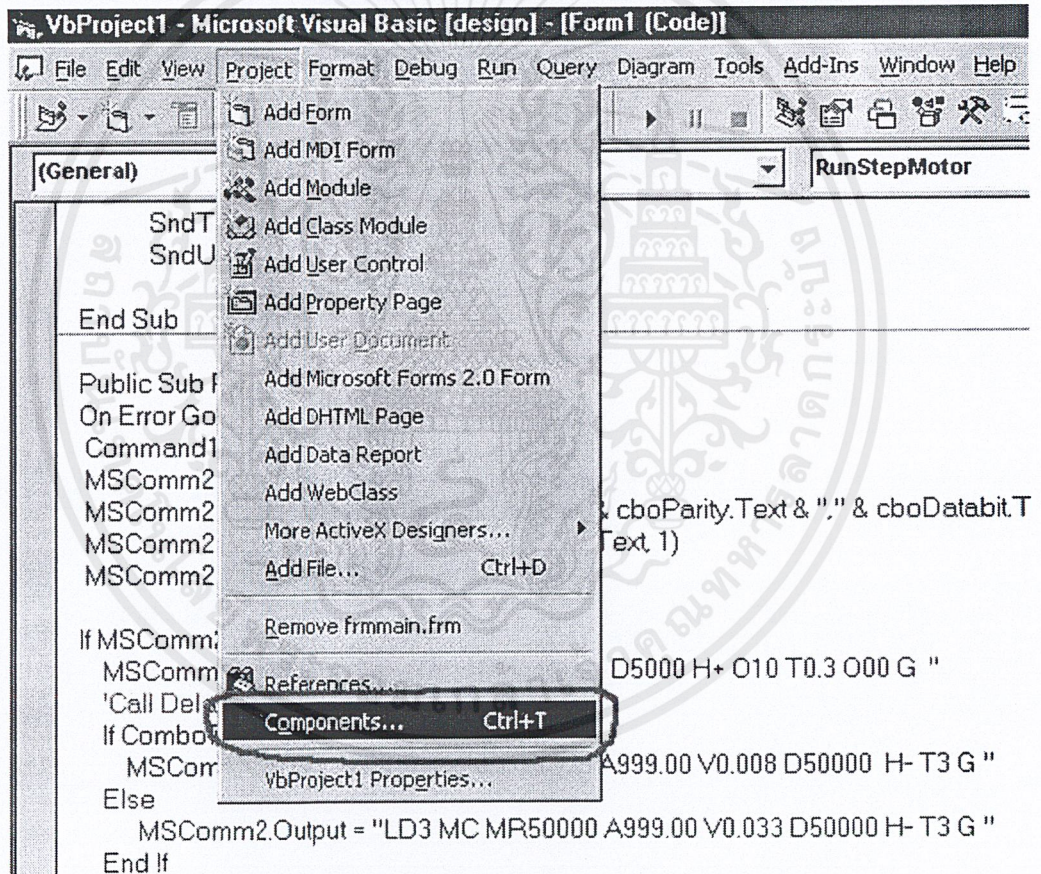
บทที่ 5

การเขียนโปรแกรมติดต่อพอร์ตอนุกรมด้วยวิซวลเบสิก

5.1 ควบคุมและติดต่อผ่านทางพอร์ตอนุกรมด้วย Com Port

สามารถทำได้โดยใช้ VB Control ที่ชื่อว่า MSComm โดยที่คุณต้อง กำหนด Custom Control เข้าไปที่ เมนู Project-->Components แล้วเลือกที่ช่อง MSComm ก็จะปรากฏ เป็นรูปไอคอนโทรศัพท์สีเหลือง ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของโปรแกรม เรา โดยสามารถทำตามวิธีที่กล่าวมา ได้ดังรูปต่อไปนี้

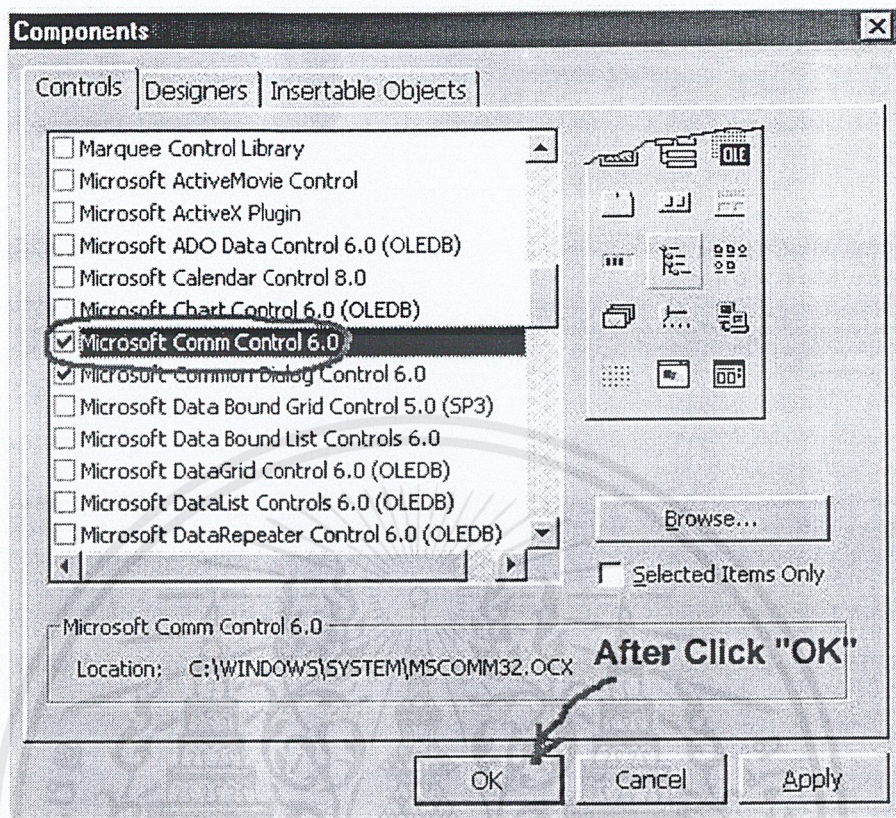
1. ขั้นที่ตอนแรก เลือกที่เมนูบาร์ด้านบนของโปรแกรมวิซวลเบสิก ดังรูปด้านล่าง



รูปที่ 5-1 แสดง ADD Components ให้โปรแกรมวิซวลเบสิก

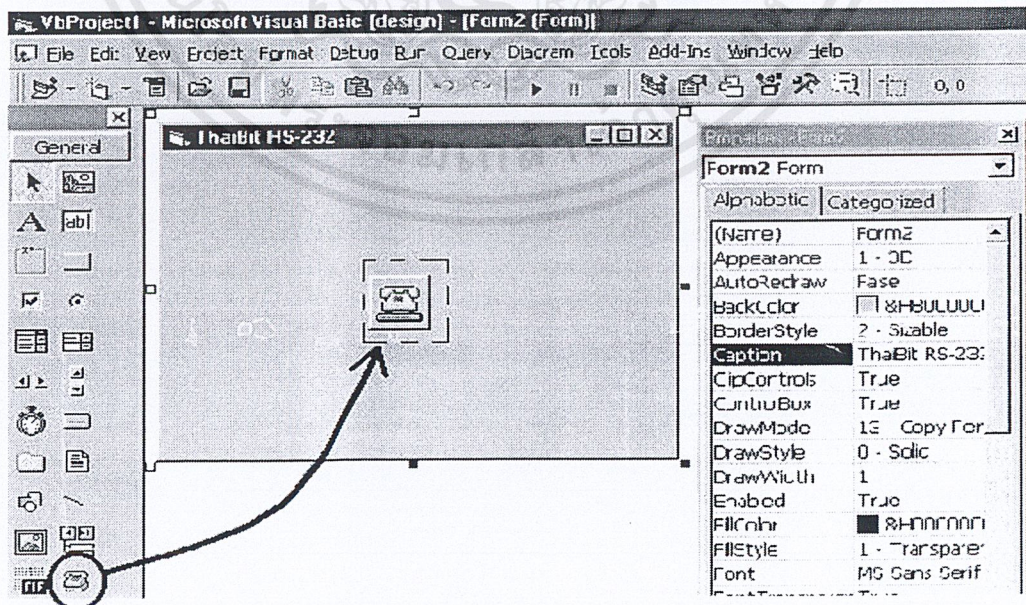
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขั้นที่สอง เลือกชื่อ Control ชื่อ Microsoft Comm Control 6 ดังรูปด้านล่าง



รูปที่ 5-2 แสดงการ ADD Component ที่ชื่อ Microsoft Comm Control 6.0

3. ลาก Control ชื่อ Microsoft Comm จาก Toolbox มาไว้บน Form ดังรูปด้านล่าง



รูปที่ 5-3 แสดงการ ADD Control Component ลง Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรม

สามารถทำได้ 2 วิธี คือ

5.2.1 การติดต่อแบบอินเทอร์รัพต์

ขบวนการอินเทอร์รัพต์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้ แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงการงานอุปกรณ์ จะส่งเป็นรหัสแอสกี (ASCII) เราจะเขียนโปรแกรมอินเทอร์รัพต์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

5.2.2 การติดต่อแบบโพลลิ่ง

ในระบบพีซี การโพลลิ่งที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็น ประเภทไบนารีที่ส่งจากแป้นพิมพ์ โดยวิธีการนี้จะตรวจสอบแป้นพิมพ์ว่ามีข้อมูลส่งมาหรือเปล่า โดยจะ ตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูล ที่ส่งเข้ามาทางแป้นพิมพ์ การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า "Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การโพลแบบ "Round Robin" แทน แต่ในวิซวลเบสิก เราจะใช้การตรวจสอบข้อมูลที่มาจากพอร์ตตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do...Loop ก็ได้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่ง จะใช้ในการติดต่อแบบอินเทอร์รัพต์ การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรมแบบธรรมดา จะใช้ comEvent เพียง comEvReceive,comEvSend ถ้าเป็นการติดต่อสื่อสารแบบ โมเด็มจะใช้หลายตัวในการ ตรวจสอบสัญญาณ ส่วนในจะมีรายละเอียดอยู่ใน Help ของวิซวลเบสิกอยู่แล้ว

5.3 องค์ประกอบในการใช้ MSComm

5.3.1 การตั้งค่าติดต่อกับพอร์ต

- **ComPort** คือ เราต้องกำหนดหมายเลขพอร์ตที่ใช้ต่อRS-232 (Com1,Com2)รายละเอียดดูในเมนูด้าน ซ้าย Serial Port Detail
 - **Setting** คือ เราต้องกำหนดอัตราการส่งข้อมูล , พาริตีบิต , Data(จำนวนบิต),Stop ตัวอย่าง 1200,n,8,1 เป็นต้น
 - **HandShaking** คือ เราจะกำหนดได้ 4 แบบ
1. comNone 2. comXonXoff 3. comRTS 4.comTRSXonXoff

5.3.2 การใช้ Buffer ในการรับส่งข้อมูล

- **InBuffersize** คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา
- **OutBuffersize** คือ การกำหนด Buffer ในการส่งข้อมูลออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Rthreshold** คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา
- **Sthreshold** คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลออกไป
- **Inputlen** คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล
- **EOFEnable** คือ การที่บอกว่สิ้นสุดของไฟล์(EOF) End of File

5.3.3 ด้านฮาร์ดแวร์

- **ParityReplace** คือ ค่าของคาเลกเตอร์ที่จะแทนในเมื่อเกิด Parity Error
- **NullDiscard** คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER
- **RTSEnable** คือ ทำให้มีสัญญาณ RTS (Request To Send)
- **DTSEnable** คือ ทำให้มีสัญญาณ DTR(Data Terminal Ready)

5.4 การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้

- **Property ชื่อ CommPort** คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน ตัวอย่างการเขียนโค้ดเป็นดังนี้

ตัวอย่าง MSComm1.CommPort=1

ในที่นี้เลือกจะใช้ Com1 อยู่ที่ด้านหลังเครื่องคอมพิวเตอร์

- **Property ชื่อ Settings** คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราการรับส่งข้อมูลของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

MSComm1.Settings="Baud(อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใช่ใส่ N,จำนวนบิตข้อมูล,บิตหยุด)"

มาดูตัวอย่างการเขียน โค้ดเป็นดังนี้

ตัวอย่าง MSComm1.Settings="1200,N,8,1"

- **Property ชื่อ InputLen** คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ มาดูตัวอย่างการเขียน โค้ดเป็นดังนี้

ตัวอย่าง MSComm1.InputLen=1

- **Property ชื่อ PortOpen** คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด = True ถ้าปิด = False มาดูตัวอย่างการเขียนโค้ดเป็นดังนี้

ตัวอย่าง MSComm1.PortOpen=True

- **Property ชื่อ Rthreshold** คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล(Comport) มันทำให้เกิดCommEvent ใน OnComm Event มาดูตัวอย่างการเขียนโค้ดเป็นดังนี้

ตัวอย่าง MSComm1.Rthreshold =1

จากรายละเอียดที่กล่าวมาข้างต้น เราจะมาเขียนในโปรแกรม Visual Basic ซึ่งจะไว้ที่ Sub Form_Load() หรือจะสร้าง Sub ขึ้นใหม่ในกรณีที่จะเรียกใช้ภายหลัง

```

Private Sub Form_Load()
MSComm1.Settings="1200,N,8,1"
MSComm1.CommPort=1
MSComm1.InputLen=1
MSComm1.PortOpen=True
MSComm1.Rthreshold =1
End Sub

```

5.5 วิธีของการรับส่งข้อมูลจากพอร์ตอนุกรม

จากวิธีเขียน โค้ดด้านบนเป็นการกำหนดค่าเริ่มต้นให้กับคอมพอร์ต และเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้นก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้

Output = ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต

Input = เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event

Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่ง จะอ่านข้อมูลเข้ามาจากทางพอร์ต RS-232 นั้นเอง

ตัวอย่าง

เช่น ถ้าต้องการที่จะพิมพ์ข้อมูลส่งออกพร้อมๆ กับขณะที่เราพิมพ์ไปด้วยคุณก็เพียง ไปเขียนโค้ดไว้ที่ Event KeyPress ของ Control TextBox ที่เราจะให้เป็นตัวส่งข้อมูล โดยเขียนดังนี้

```

Sub txtRXTX_KeyPress(KeyAscii As Integer)
MSComm1.Output=Chr$(KeyAscii)
End Sub

```

ส่วนการใช้ Property Input ต้องนำมาไว้ที่ Event OnComm ดังนี้

```

Private Sub MSComm1_OnComm()
Dim StrData As Variant 'กำหนดชนิดตัวแปรเพราะต้องการให้เป็นอะไรก็ได้
Str=MSComm1.Input
Text1.Text=StrData
End Sub

```

จากผลการทดลองครั้งแรกที่ผมทำในส่วนของการรับข้อมูล Property Input ซึ่งถ้าเขียน โค้ดดังกล่าวข้างต้นนั้นจะยังใช้ไม่ในการที่จะอ่านข้อมูลและแสดงออกมาซึ่งจะต้องเขียน ฟังก์ชันขึ้นชื่อ DataShow มาตรวจสอบซึ่ง โค้ดดังนี้

- ฟังก์ชัน DataShow

```

Public Static Sub DataShow(TextShow As Control, Data As String)
Const SpeedBaud = 16000
Dim lngSize As Long, X

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LngSize = Len(TextShow.Text)
If TermSize > SpeedBaud Then
TextShow.Text = Mid$(TextShow.Text, 4097)
LngSize = Len(TextShow.Text)
End If
TextShow.SelStart = SpeedBaud
Do
X = InStr(Data, Chr$(8))
If X Then
If X = 1 Then
TextShow.SelStart = SpeedBaud - 1
TextShow.SelLength = 1
Data = Mid$(Data, X + 1)
Else
Data = Left$(Data, X - 2) & Mid$(Data, X + 1)
End If
End If
Loop While X
Do
X = InStr(Data, Chr$(10))
If X Then
Data = Left$(Data, X - 1) & Mid$(Data, X + 1)
End If
Loop While X
X = 1
Do
X = InStr(X, Data, Chr$(13))
If X Then
Data = Left$(Data, X) & Chr$(10) & Mid$(Data, X + 1)
X = X + 1
End If
Loop While X
TextShow.SelText = Data

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TextShow.SelStart = Len(TextShow.Text)

End Sub

- การเรียกใช้ฟังก์ชัน DataShow :

Call DataShow Text1,(StrConv((StrData),vbUnicode))

ซึ่งจะทำให้มีตัวอักษรแสดงออกที่ TextBox หลักการเขียนโปรแกรมนี้ใช้ติดต่อกับอุปกรณ์ที่สามารถส่งค่า Acscii ทางSerial RS-232 ได้ อย่างเช่น การ์ดแสดงผลของเครื่องชั่งน้ำหนัก การ์ดวัดอุณหภูมิ เพียงเราตั้งค่าของคอมพิวเตอร์ให้ตรงกับที่อุปกรณ์กำหนดมา

- การใช้ Even ในโพธิ์เซอร์ OnComm()

ชื่อ Property	คำอธิบาย
ComEventBreak	การได้รับสัญญาณเบรก
ComEventCDTO	เมื่อเกิด ไทม์เอาต์ ขณะที่กำลังคอยสัญญาณ CD(Carrier Detect)
ComEventCTSTO	เมื่อเกิด ไทม์เอาต์ ขณะที่กำลังคอยสัญญาณ CTS(Carrier To Send)
ComEventDSRTO	เมื่อเกิด ไทม์เอาต์ ขณะที่กำลังคอยสัญญาณ DSR (Data Set Ready)
ComEventFrame	การที่เกิดความผิดพลาดทางเฟรม เป็นลักษณะที่ไม่พบปิดจบตามที่ควรจะเป็น
ComEventOverrun	การที่เกิดความผิดพลาดโอเวอร์รัน เป็นลักษณะที่รับข้อมูลไม่ทันในการประมวลผล
ComEventRxOver	บัฟเฟอร์ที่รับข้อมูลเกิดโอเวอร์โฟลล์ ก็คือรับตัวอักษรหลังจากการรับ EOF Char
ComEventRxParity	การที่เกิดความผิดพลาดทางพาริตี เป็นลักษณะที่ตัวอักษรที่รับ ได้มีพาริตีไม่ถูกต้อง
ComEventTxFull	ตัวบัฟเฟอร์ที่ส่งข้อมูลเต็ม
ComEventDCB	การที่เกิดความผิดพลาดขึ้น โดยไม่ได้คาดถึง

ตารางที่ 5-1 แสดง CommEvent เกี่ยวกับการเกิดสถานะเมื่อเกิดการผิดพลาดในการสื่อสาร

ชื่อ Property	คำอธิบาย
ComEvCD	CD(Carrier Detect) เมื่อเปลี่ยนซึ่งคือสายของสัญญาณ Receive Line Signal Detect(RLSD)
ComEvCTS	RCTS(Carrier To Send)เมื่อมีการเปลี่ยนสถานะเกิดขึ้น
ComEvDSR	DSR(Data Set Ready) เมื่อมีการเปลี่ยนสถานะเกิดขึ้น
ComEvRing	เมื่อตรวจจับสัญญาณ Ring Indicator ได้
ComEvReceive	เมื่อได้รับข้อมูลเก็บลงใน InputBuffer
ComEvSend	เมื่อส่งข้อมูลออกจาก OutputBuffer
ComEvEof	เมื่อพบอักขระ EOF(End Of File)

ตารางที่ 5-2 แสดง CommEvent เกี่ยวกับเกิดสถานะการเมื่อมีการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ Property	คำอธิบาย
Break	ในการที่เรากำหนดหรือเคลียร์สัญญาณเบรก
CDHolding	ตรวจสอบสัญญาณ Carrier Detect(CD)ดูว่ายังคงมีสถานะอยู่หรือเปล่า
CDTimeout	การกำหนดค่าหรือทำให้ค่าของเวลา(หน่วย mSec) ที่รอสัญญาณ Carrier Detect
CommEvent	จะให้ผลของการเกิด Event ของ Communication
CommID	จะให้ผลของการเสดเคลิขของ Communication ที่เปิดใ้ช้อยู่
CommPort	การกำหนดหรือว่าอ้างอิงของหมายเลขคอมพอร์ต ที่เปิดใ้ช้อยู่ เช่น Com1=1 , Com2=2
CTSHolding	เป็นการตรวจสอบสัญญาณของ Clear To Send ว่ายังคงมีสถานะอยู่หรือเปล่า
CTSTimeout	การกำหนดค่าหรือทำให้ค่าของเวลา(หน่วย mSec) ที่รอสัญญาณ Data Set Ready
DSRHoldIng	เป็นการตรวจสอบสัญญาณของ Data Set Ready ว่ายังคงมีสถานะอยู่หรือเปล่า
DSRTimeout	การกำหนดค่าหรือทำให้ค่าของเวลา(หน่วย mSec) ที่รอสัญญาณ Clear To Send
DTREnable	ใ้เปิดการใช้สายของสัญญาณ Data Terminal Ready(DTR)
Handshaking	กำหนดการแฮนเช็ดทางฮาร์ดแวร์ เพื่อที่ค้อยตรวจสอบการรับส่งข้อมูล
InBufferCount	ใ้ค่าของจำนวนข้อมูลที่อยู่ภายในบัฟเฟอร์รับข้อมูล
InBufferSize	กำหนดหรือว่าใ้ค่าของขนาดในบัฟเฟอร์รับข้อมูล
Input	เป็นการใ้ค่าหรือว่าเคลื่อนย้ายข้อมูลจากบัฟเฟอร์รับข้อมูล
InputLen	การกำหนดหรือว่าใ้ของจำนวนข้อมูลที่นำมาจากบัฟเฟอร์รับข้อมูล
Interval	เป็นการกำหนดอัตราความเร็วของการใ้งานในโหมดโพลลิ่ง
NullIDiscard	เป็นการกำหนดใ้มีการรับ Null Character เก็บลงในบัฟเฟอร์รับข้อมูล
OutBufferCount	เป็นจำนวนข้อมูลที่ค้อยอยู่ในบัฟเฟอร์ส่งข้อมูล
OutBufferSize	การกำหนดหรือว่าใ้ค่าขนาดของบัฟเฟอร์ส่งข้อมูล
Output	เป็นการส่งข้อมูลใ้กับบัฟเฟอร์ส่งข้อมูลเพื่อทำการส่งข้อมูลออก
ParityReplace	เป็นการกำหนดใ้ส่งอักขระที่กำหนดนี้แทนหากเกิดการผิดพลาดในข้อมูล
PortOpen	เป็นการกำหนดหรือว่าใ้ค่าของสถานะพอร์ตว่าเปิดหรือปิดอยู่
Rthreshold	การกำหนดหรือว่าใ้ค่าของจำนวนข้อมูลที่เก็บลงในบัฟเฟอร์รับข้อมูลก่อนการเกิด CommEvent ในการรับข้อมูล
RTSEnable	ใ้เปิดการใช้สายสัญญาณ Request To Send(RTS)
Settings	เป็นการกำหนดอัตราการส่งข้อมูล พาริตีข้อมูล บิตหยุด
Sthreshold	การกำหนดหรือว่าใ้ผลของจำนวนข้อมูลที่เก็บลงในของบัฟเฟอร์ส่งข้อมูลก่อนการเกิด CommEvent ในการใ้ส่งข้อมูล

ตารางที่ 5-3 แสดง MSComm Control Property

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใ้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตใ้ให้นำไปใ้ประโยชน์ด้านการค้าไม่ว่ากรณีใ้ใดๆ ทั้งสิ้น อีกทั้งห้ามมิใ้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใ้

เราสามารถนำ Event ในตารางดังกล่าวมาใช้เขียนใน โฟชั่นเคอร์ OnComm() โดยการใส่เงื่อนไขแบบ Select case ซึ่งสามารถเขียนได้ดังนี้

```
Private Sub MSComm1_OnComm()
Select Case MSComm1.CommEvent
Case comEvReceive
Dim Buffer As Variant
Buffer = MSComm1.Input
ShowData txtRXTX, (StrConv(Buffer, vbUnicode))
Case comEvSend : 'ส่วนนี้จะใส่เงื่อนไขให้ทำอะไรก็ได้ตามที่คุณว่างคอนเซปไว้
Case comEvCTS
Case comEvDSR
Case comEvCD
Case comEvRing
Case comEvEOF
Case comBreak
Case comCDTO
Case comCTSTO
Case comDCB
Case comDSRTO
Case comFrame
Case comOverrun
Case comRxOver
Case comRxParity
Case comTxFull
End Select
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

แนวคิดและการออกแบบ

6.1 Sequence Diagram และ Collaboration Diagram

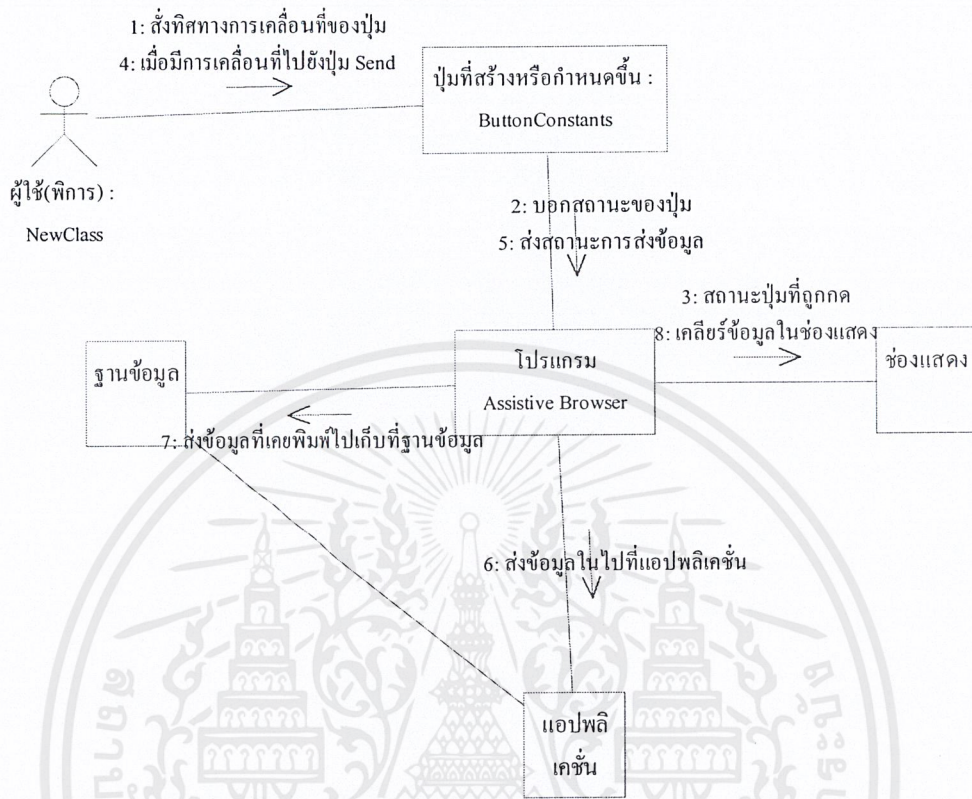
6.1.1 แสดง Sequence Diagram ของสถานะของการกดปุ่ม



รูปที่ 6-1 แสดง Sequence Diagram สถานะของการกดปุ่ม

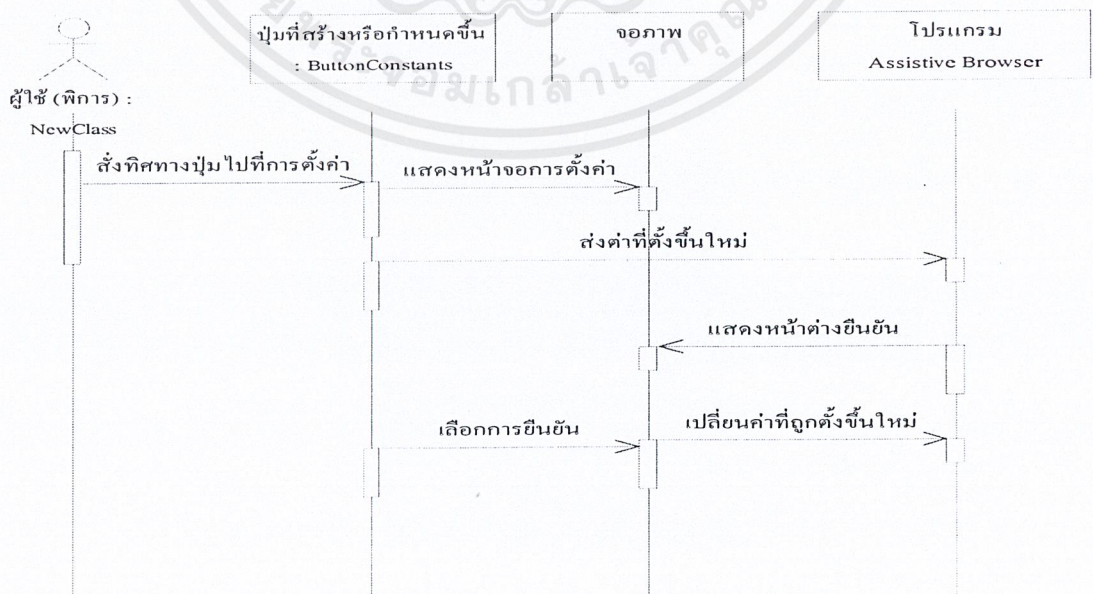
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 แสดง Collaboration Diagram ของสถานะของการกดปุ่ม



รูปที่ 6-2 แสดง Collaboration Diagram สถานะของการกดปุ่ม

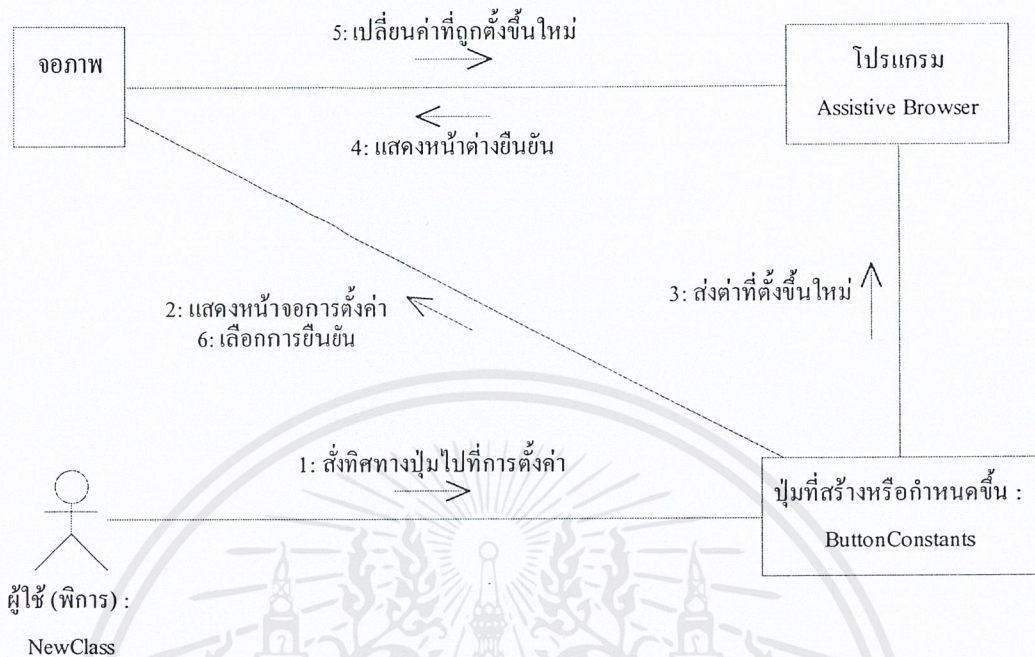
6.1.3 แสดง Sequence Diagram ของสถานะของการตั้งค่าต่าง ๆ



รูปที่ 6-3 แสดง Sequence Diagram สถานะของการตั้งค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

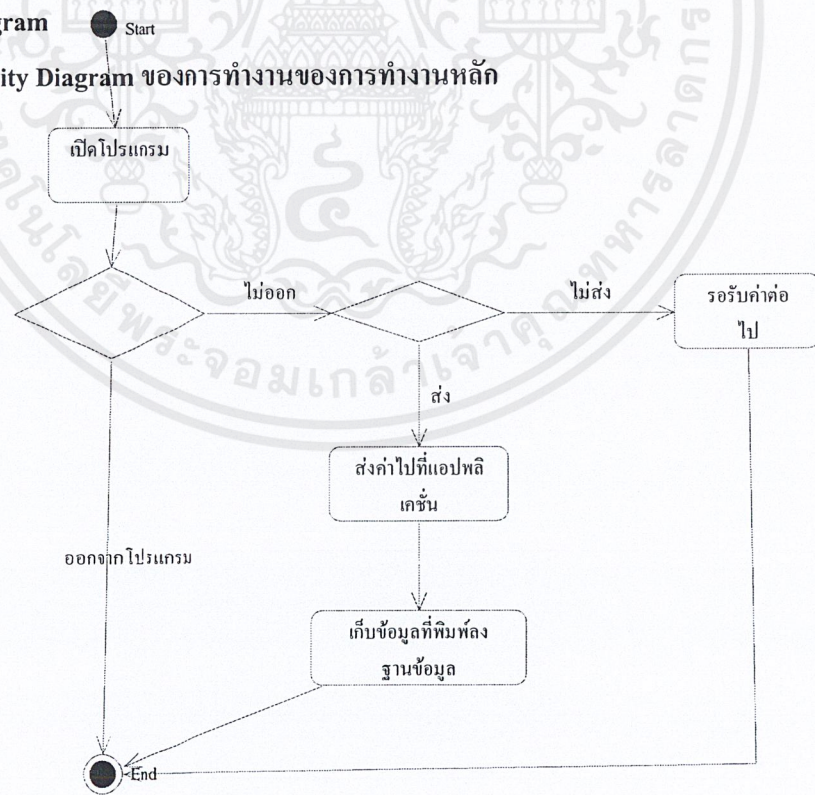
6.1.4 แสดง Collaboration Diagram ของสถานะของการตั้งค่าต่าง ๆ



รูปที่ 6-4 แสดง Collaboration Diagram สถานะของการตั้งค่าต่าง ๆ

6.2 Activity Diagram

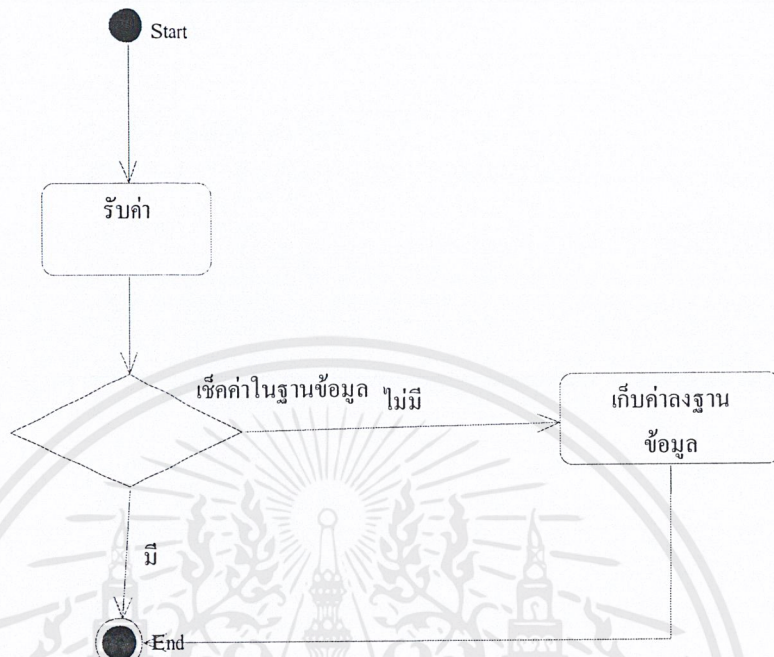
6.2.1 แสดง Activity Diagram ของการทำงานของการทำงานหลัก



รูปที่ 6-5 แสดง Activity Diagram การทำงานของโปรแกรมหลัก

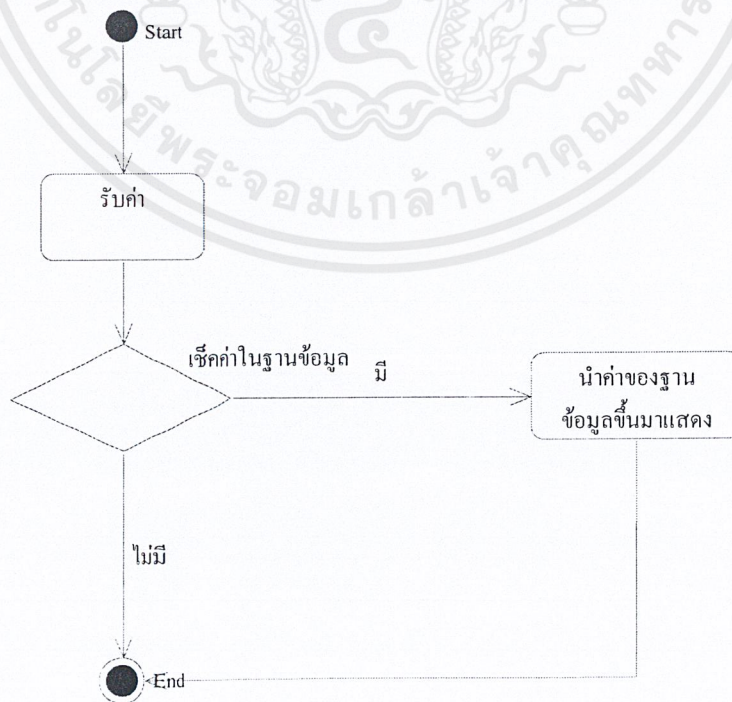
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 แสดง Activity Diagram ของการเก็บข้อมูลลงฐานข้อมูล



รูปที่ 6-6 แสดง Activity Diagram ของการเก็บข้อมูลลงฐานข้อมูล

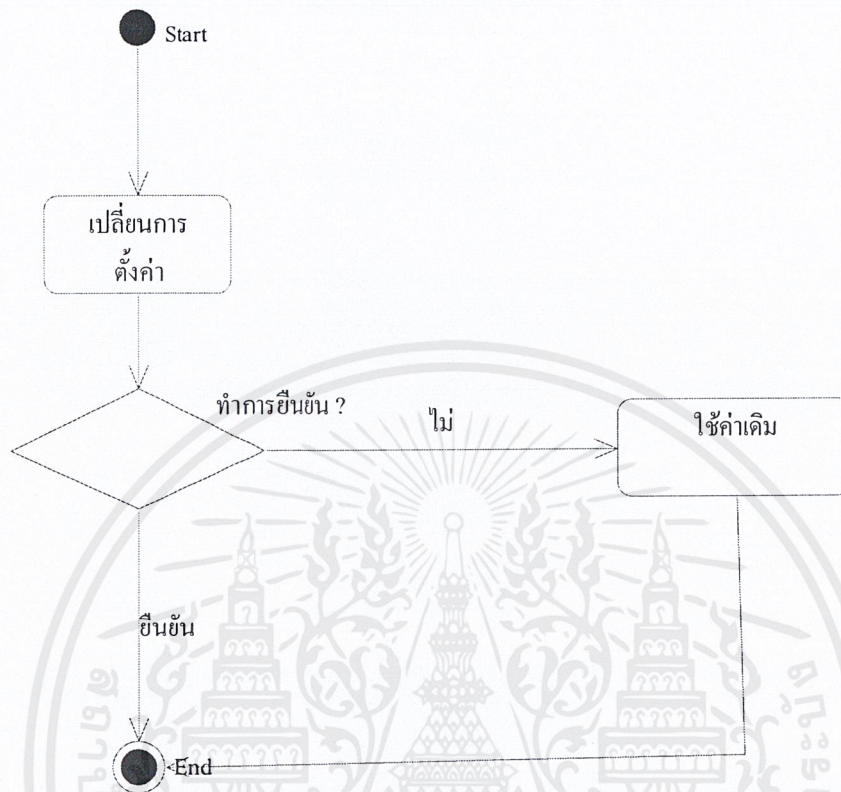
6.2.3 แสดง Activity Diagram ของการนำข้อมูลจากฐานข้อมูลมาแสดงในส่วน Predict Field



รูปที่ 6-7 แสดง Activity Diagram ของการนำข้อมูลจากฐานข้อมูลมาแสดงในส่วน Predict Field

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนการคำนวณ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 แสดง Activity Diagram ในการเปลี่ยนการตั้งค่า

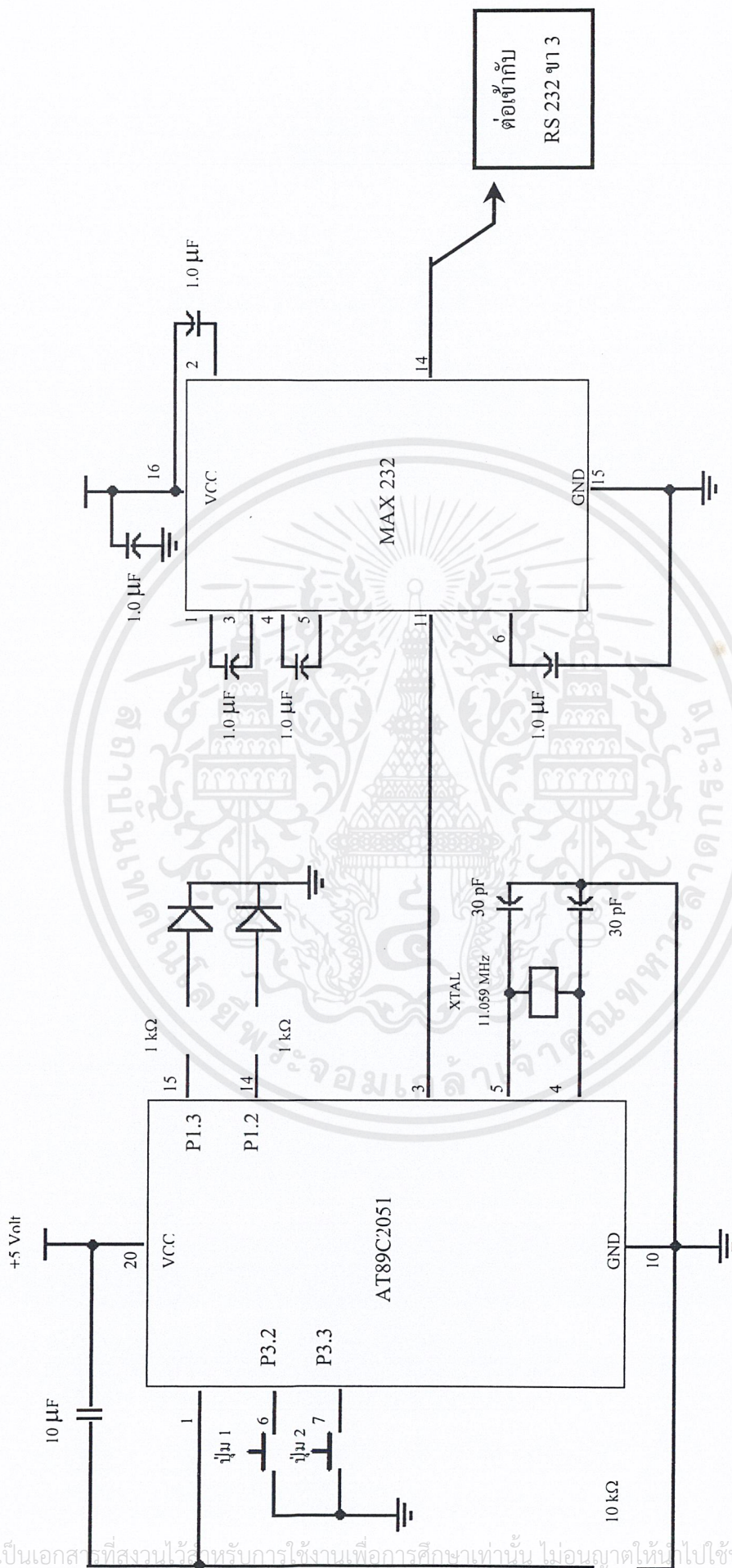


รูปที่ 6-8 แสดง Activity Diagram ในการเปลี่ยนการตั้งค่า

6.3 แสดงการออกแบบวงจรของปุ่ม 2 ปุ่ม

เป็นการออกแบบของการใช้ปุ่ม 2 ปุ่ม เป็นวงจรดังรูปที่ 6-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-9 แสดงการต่อวงจรปุ่ม 2 ปุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทสรุปและวิจารณ์

7.1 ปัญหาและอุปสรรคในการทำงาน

1. หนังสือที่เป็นข้อมูลค้นคว้าเกี่ยวกับ WIN32 API ยังมีน้อย ต้องทำการค้นคว้าทางอินเทอร์เน็ตแทน
2. ขาดความรู้ทางด้านฮาร์ดแวร์ทำให้เสียเวลาในการค้นคว้าเกี่ยวกับทางด้านฮาร์ดแวร์มาก

7.2 สรุปผลที่ได้จากการทำโครงการ

1. ได้รับความรู้และความชำนาญเกี่ยวกับการเขียน โปรแกรมด้วยภาษาวิซวลเบสิก
2. ได้รับความรู้และความชำนาญในการเขียน โปรแกรมเชิงประยุกต์โดยใช้ WIN32 API
3. ได้รับความรู้และความชำนาญในการเขียน โปรแกรมติดต่อกับฐานข้อมูล
4. ได้รับความรู้และความชำนาญเกี่ยวกับการเขียน โปรแกรมติดต่อกับระบบปฏิบัติการ
5. ได้รับความรู้จักการทำงานของพอร์ตอนุกรม
6. ได้เรียนรู้การทำงานเป็นกลุ่ม

7.3 ข้อเสนอแนะ

การนำโปรแกรมไปใช้ในการทำงานสามารถ ควบคุมโดยการใช้ปุ่มเพียง 2 ปุ่มผ่าน โปรแกรมที่สร้างขึ้น สำหรับผู้พิการทางมือที่ไม่มีแขนแม่แต่ข้างเดียวโดยใช้เท้ากดปุ่ม 2 ปุ่มซึ่งเป็นอุปกรณ์ฮาร์ดแวร์ที่สร้างขึ้น หรือถ้าเป็นผู้พิการทางมือที่ยังมีแขนเหลืออยู่ โดยที่มีความสามารถในการควบคุมเมาส์ได้ ก็จะทำให้การใช้โปรแกรมนี้ได้ และสามารถสะดวกยิ่งขึ้นด้วย

7.4 แนวทางการพัฒนาต่อ

มีความคิดที่จะเอาแผ่นเต้น (Dance Board) มาทำการต่อเข้าด้วยกัน ให้แผ่นเต้นสามารถควบคุมการทำงานของโปรแกรมได้ เพื่อที่จะสามารถควบคุมได้ง่ายขึ้น

7.5 ผลการทดสอบโปรแกรม

บนระบบปฏิบัติการต่างๆ

ระบบปฏิบัติการ	ผลการทดสอบ
Windows 98	ใช้งานได้
Windows ME	ใช้งานได้
Windows 2000	ใช้งานได้
Windows NT	ใช้งานได้

บนแอปพลิเคชันต่างๆ

แอปพลิเคชันต่างๆ	ผลการทดสอบ
Microsoft Office 97/2000	ใช้งานได้ดี ซึ่งสามารถพิมพ์ข้อความต่างๆ ได้โดยไม่ต้องเก็บไว้ที่ Text Field ก่อน
Notepad	ใช้งานได้ดี ซึ่งสามารถพิมพ์ข้อความต่างๆ ได้โดยไม่ต้องเก็บไว้ที่ Text Field ก่อน
Edit Plus	ใช้งานได้ดี ซึ่งสามารถพิมพ์ข้อความต่างๆ ได้โดยไม่ต้องเก็บไว้ที่ Text Field ก่อน
Calculator	สามารถใช้ปุ่ม Num Pad ที่ถูกสร้างขึ้นด้วยโปรแกรมเป็นพิมพ์เสมือน ซึ่งสามารถทำงานได้ดี
Internet Explorer	ใช้งานได้ดี ซึ่งสามารถพิมพ์ข้อความต่างๆ ได้แต่ต้องเก็บไว้ที่ Text Field ก่อนที่จะส่งข้อความ
MSN Messenger	ใช้งานได้ดี ซึ่งสามารถพิมพ์ข้อความต่างๆ ได้โดยไม่ต้องเก็บไว้ที่ Text Field ก่อน

หมายเหตุ ยังมีแอปพลิเคชันอีกหลายตัวที่ยังใช้งานได้ อีก แต่โปรแกรมเหล่านี้ เป็น โปรแกรมที่เราทดสอบขึ้นมา ซึ่งเห็นว่าเป็นแอปพลิเคชันที่ใช้งานอยู่บ่อยครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก ตารางแอสกี

8	ปุ่มลบ	59	;	89	Y	121	y
9	Tab	60	<	90	Z	122	z
10	Shift	61	=	91	[123	{
13	Enter	62	>	92	\	124	
27	Esc	63	?	93]	125	}
33	!	64	@	94	^	126	~
35	#	65	A	97	a	128	ฐ
36	\$	66	B	98	b	129	
37	%	67	C	99	c	130	
38	&	68	D	100	d	131	
39	/	69	E	101	e	132	
40	(70	F	102	f	133	...
41)	71	G	103	g	134	
42	*	72	H	104	h	135	
43	+	73	I	105	i	136	
44	,	74	J	106	j	137	
45	-	75	K	107	k	138	
46	.	76	L	108	l	139	
47	/	77	M	109	m	140	
48	0	78	N	110	n	141	
49	1	79	O	111	o	142	
50	2	80	P	112	p	143	
51	3	81	Q	113	q	144	ญ
52	4	82	R	114	r	145	'
53	5	83	S	115	s	146	,
54	6	84	T	116	t	147	"
55	7	85	U	117	u	148	"
56	8	86	V	118	v	149	.
57	9	87	W	119	w	152	"
58	:	88	X	120	x	153	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

161	ก	162	ข	163	ฅ	164	ค
165	ค	166	ฅ	167	ง	168	จ
169	ฉ	170	ช	171	ฌ	172	ฉ
173	ญ	174	ฎ	175	ฏ	176	ฐ
177	ท	178	ฒ	179	ณ	180	ด
181	ต	182	ถ	183	ท	184	ธ
185	น	186	บ	187	ป	188	ผ
189	ฝ	190	พ	191	ฟ	192	ภ
193	ม	194	ย	195	ร	196	ฤ
197	ล	198	ฬ	199	ว	200	ศ
201	ษ	202	ส	203	ห	204	พ
205	อ	206	ฮ	207	๑	208	๒
210	๓	211	๔	226	๖	227	๗
228	๘	229	๙	230	๑	240	1
241	1	242	2	243	3	244	4
245	5	246	6	247	7	248	8
249	9						

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

หน้าที่ของโพรซีเจอร์ และฟังก์ชันต่างๆ ในโปรแกรม

อธิบายหน้าที่ของโพรซีเจอร์และฟังก์ชันต่างๆในโปรแกรม

- Sub SetStartValueFunction()
เรียก โพรซีเจอร์ SetStartValue และกำหนดเวลาวนซ้ำของ Timer
- Public Sub SetStartValue()
กำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆ เช่น ทิศทางเริ่มต้นของเมาส์ ความเร็วของการตรวจจับการคลิกและดับเบิลคลิกปุ่ม เป็นต้น
- Sub SetKeyEng()
แสดงค่า Caption ของปุ่มให้อยู่ในรูปแบบของแป้นพิมพ์ภาษาอังกฤษที่ไม่ได้กด Shift
- Sub SetKeyShiftEng()
แสดงค่า Caption ของปุ่มให้อยู่ในรูปแบบของแป้นพิมพ์ภาษาอังกฤษที่ถูกกด Shift
- Sub SetKeyThai()
แสดงค่า Caption ของปุ่มให้อยู่ในรูปแบบของแป้นพิมพ์ภาษาไทยที่ไม่ได้กด Shift
- Sub SetKeyShiftThai()
แสดงค่า Caption ของปุ่มให้อยู่ในรูปแบบของแป้นพิมพ์ภาษาไทยที่ถูกกด Shift
- Sub SpyWindow()
จะทำหน้าที่ตรวจการรับเหตุการณ์ที่ผู้ใช้ส่งเข้ามาแล้วตรวจสอบ ถ้าเป็นการกดปุ่มที่ใช้ควบคุมเมาส์ก็จะเรียกฟังก์ชัน MoveMouseCursor และถ้ามีการมีการคลิกเกิดขึ้นจะเรียกฟังก์ชัน TrackMouse เพื่อหาค่า Handle ของหน้าต่างที่ถูกคลิก
- Function TrackMouse(IHwnd As Long) As Long
จะหาค่า Handle โดยตรวจสอบจากตำแหน่งปัจจุบันของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Sub MoveMouseCursor()

จะตรวจสอบการกดปุ่มที่ใช้บังคับทิศทางเมาส์ว่าเป็นการกดปุ่มใดและเรียกฟังก์ชัน CheckKeyUpDown หรือ CheckKeyLeftRight ซึ่งจะส่งค่าทิศทางกลับมาให้ จากนั้นจะทำการเลื่อนตำแหน่งเมาส์โดยเรียกโปรซีเจอร์ DoneMoveCursor
- Function CheckKeyUpDown() As String

จะตรวจสอบปุ่มที่ใช้ควบคุมตำแหน่งเมาส์ ขึ้น-ลง ว่าเป็นการคลิกปุ่มหรือดับเบิลคลิกปุ่ม แล้วเรียกฟังก์ชัน Direction ซึ่งจะคำนวณหาทิศทางจากทิศทางล่าสุดและพิจารณาจากสถานะการกดปุ่มแล้วส่งค่าทิศทางกลับไปให้
- Function CheckKeyLeftRight() As String

จะตรวจสอบปุ่มที่ใช้ควบคุมตำแหน่งเมาส์ ซ้าย-ขวา ว่าเป็นการคลิกปุ่มหรือดับเบิลคลิกปุ่ม แล้วเรียกฟังก์ชัน Direction ซึ่งจะคำนวณหาทิศทางจากทิศทางล่าสุดและพิจารณาจากสถานะการกดปุ่มแล้วส่งค่าทิศทางกลับไปให้
- Sub DoneMoveCursor()

จะเลื่อนตำแหน่งของเมาส์โดยพิจารณาการเลื่อนจากตัวแปร Direction
- Sub CheckDirection()

จะคำนวณทิศทางของเมาส์และสั่งให้แสดงภาพทิศทางของเมาส์
- Sub SendChar(ch As String)

ส่งตัวอักษรไปยังหน้าต่างที่กำลังทำงาน
- Sub send_function_key(fKey As Byte)

ส่งงานปุ่มฟังก์ชันไปยังหน้าต่างที่กำลังทำงาน
- Sub SendDisplay(text_t As String)

ส่งตัวอักษรทั้งหมดในกล่องข้อความของ Assistive Browser ไปยังหน้าต่างที่กำลังทำงาน พร้อมทั้งเรียกฟังก์ชัน History เพื่อตรวจสอบว่าเป็นคำที่มีอยู่ใน รายการของกล่องข้อความหรือไม่ ถ้าไม่มี จะเพิ่มเข้าไปในกล่องข้อความ (Combo Box)
- Function Check_history(text_t As String) As Boolean

ตรวจสอบว่า text_t มีอยู่ในรายการของกล่องข้อความหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Private Sub TimerStartProgram_Timer()
เป็นตัวจับเวลาให้เริ่มต้นทำโปรแกรม SpyWindow
- Private Sub TimerClickMouse_Timer()
เป็นตัวจับเวลาเพื่อทำการคลิก ดับเบิลคลิก หรือการลากแล้วปล่อย(Drag and Drop)
- Private Sub TimerLoopMousemove_Timer()
วนตรวจว่า เมาส์เคลื่อนที่หรือไม่ ถ้าหยุดเคลื่อนที่แล้วให้ตั้ง TimerClickMouse_timer ทำงาน
- Private Sub TimerDB_Timer()
ใช้นับเวลาหลังจากการกดปุ่มครั้งแรกและครั้งที่สองเพื่อใช้พิจารณาว่าเป็นการคลิกหรือดับเบิลคลิก
- Private Sub DoMouseEvent()
ใช้ตรวจสอบว่ามีการส่งงานเมาส์ คลิก ดับเบิลคลิก หรือลากเมาส์ และเป็นการสั่งให้ทำที่เมาส์ปุ่มขวาหรือปุ่มซ้าย
- Sub StayOnTop(frmHwnd As Long, bAction As Boolean)
ใช้สั่งให้หน้าต่างที่ต้องการแสดงอยู่ ณ ตำแหน่งหน้าสุด(อยู่เหนือหน้าต่างที่ใช้งานอยู่)
- Sub mouse_press_left()
สั่งให้กดเมาส์ซ้าย
- Sub mouse_press_right()
สั่งให้กดเมาส์ขวา
- Sub mouse_release_left()
สั่งให้คลายเมาส์การกดเมาส์ซ้าย
- Sub mouse_release_right()
สั่งให้คลายเมาส์การกดเมาส์ขวา
- Function TrackMouse(lHwnd As Long) As Long

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาค่า Handle ของหน้าต่างจากตำแหน่งปัจจุบันของเมาส์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] กิตติ ภัคดีวัฒนะกุล – จำลอง ครูอุตสาหะ : “Visual Basic 6 ฉบับฐานข้อมูล” , บ.เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด ,2521
- [2] กิตติ ภัคดีวัฒนะกุล – จำลอง ครูอุตสาหะ : “Visual Basic 6 ฉบับโปรแกรมเมอร์” , บ.เคทีพี คอมพ์แอนด์ คอนซัลท์ จำกัด ,2521
- [3] สัจจะ จรัสรุ่งรวีร : “คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6” , สำนักพิมพ์ อินโฟเพรส ,2544
- [4] สุขชัย สมพานิช : “Database Programming ด้วย Visual Basic ฉบับมืออาชีพ” , สำนักพิมพ์อินโฟเพรส ,2543
- [5] ชาริน ศิทธิธรรมขารี : “คู่มือการเขียนโปรแกรม Microsoft Visual Basic Version 6.0” , บริษัท ส.เอเชียเพรส(1989) จำกัด , 1989

แหล่งค้นคว้าทางอินเทอร์เน็ต

- [1] <http://www.microsoft.com>
- [2] <http://thai.ch.net>
- [3] <http://www.thairinker.com/vbth>
- [4] <http://www.thaib.com>
- [5] <http://www.google.com>
- [6] <http://se-ed.net/thai/vb6/>
- [7] <http://www.thindex.com>
- [8] <http://www.abstract.h.com>
- [9] <http://xhopencode.onweb.cz>
- [10] <http://www.siam.su.net>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้