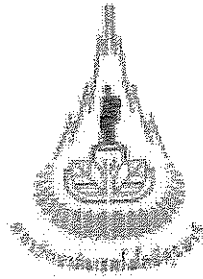


CONTRIBUTION



ระบบรักษาความปลอดภัย และฐานข้อมูลโดยใช้ระบบ
ระบุลักษณะของวัตถุด้วยคลื่นความถี่วิทยุ (RFID)

นายวาณิช โสภาก

นายเฉลิมพงศ์ ชวนาห้วยตะโก

นายชนบรพ์ เจริญภิญโญภาพ

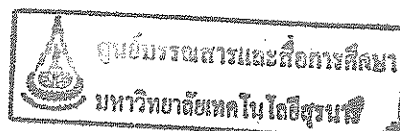
มหาวิทยาลัยเทคโนโลยีสุรนารี

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2548



โครงการ	ระบบรักษาความปลอดภัยและฐานข้อมูล โดยใช้ระบบระบุลักษณะของวัตถุด้วยคลื่นความถี่วิทยุ (RFID)
ผู้พัฒนาโครงการ	นายวาณิช โสกา นายเฉลิมพงศ์ ชวนนาทไวทยะโก นายชรรณบรรพ์ เจริญกัญญาภาท
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ชาญชัย ทองโสกา
สาขาวิชา	วิศวกรรมโทรคมนาคม
ภาคการศึกษาที่	3/2548

บทคัดย่อ

โครงการระบบรักษาความปลอดภัยและฐานข้อมูล โดยใช้ระบบระบุลักษณะของวัตถุด้วยคลื่นความถี่วิทยุ (RFID) นั้น ได้ทำขึ้นเพื่อมุ่งหวังที่จะสามารถช่วยเพิ่มความปลอดภัยแก่บริษัทหรือสำนักงานให้มีระบบรักษาความปลอดภัยภายในอาคารที่มีความปลอดภัยมากยิ่งขึ้นและสามารถที่จะจัดเก็บข้อมูลของพนักงานไว้เพื่อตรวจสอบข้อมูลการทำงานและเวลาในการเข้าหรือออกจากบริษัทซึ่งการจัดเก็บข้อมูลโดยใช้ระบบฐานข้อมูล (DATABASE) นั้นจะทำให้ง่ายต่อการใช้งานและยังสามารถกรอกข้อมูลของพนักงานใหม่หรือลบข้อมูลของพนักงานเก่าก็ทำได้โดยง่าย อีกทั้งระบบฐานข้อมูลนี้ยังช่วยป้องกันไม่ให้บุคคลภายนอกที่ไม่ได้รับอนุญาตเข้ามาในบริษัทหรือสำนักงานได้อีกด้วย

โครงการระบบรักษาความปลอดภัยและฐานข้อมูล โดยใช้ระบบระบุลักษณะของวัตถุด้วยคลื่นความถี่วิทยุ (RFID) นี้ มีองค์ประกอบที่เป็นส่วนสำคัญในการทำงานคือ วงจรเครื่องส่งและวงจรเครื่องรับซึ่งวงจรเครื่องส่งจะทำหน้าที่ในการส่งข้อมูลของพนักงานแต่ละคน ส่วนวงจรเครื่องรับจะบอกรหัสข้อมูลของพนักงานแต่ละคน โดยวงจรเครื่องรับซึ่งติดตั้งที่ประตูและจะส่งรหัสที่อ่านได้ไปยังไมโครคอนโทรลเลอร์เบอร์ AT89S8252 ซึ่งจะทำหน้าที่ในการประมวลผลเพื่อที่จะตรวจสอบว่าเป็นข้อมูลของพนักงานคนใด จากนั้นก็จะส่งรหัสข้อมูลของพนักงานคนดังกล่าวเข้าไปยังคอมพิวเตอร์และจะเรียกข้อมูลจากฐานข้อมูลที่ได้นับที่กไว้ออกมาแสดงที่จอคอมพิวเตอร์

กิตติกรรมประกาศ

โครงการนี้สำเร็จได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์ในการให้คำปรึกษาในด้านต่างๆ ในระหว่างการดำเนินการจากบุคคลหลายท่านที่ได้ให้ความช่วยเหลือและให้คำปรึกษา รวมทั้งข้อเสนอแนะที่เป็นประโยชน์ในการทำโครงการครั้งนี้

บุคคลเหล่านี้ประกอบไปด้วย

อ.ดร. ชาญชัย ทองโสภิต อาจารย์ที่ปรึกษาที่ให้คำปรึกษาในด้านต่างๆทั้งทางวิชาการและ การปฏิบัติงานและควบคุมการทำงานอย่างใกล้ชิด และคอยให้กำลังใจเสมอมา.คุณประพล จาระคะคู หัวหน้าอาคารศูนย์เครื่องมือ3ที่ช่วยดูแลและคิดคือประสานงานเรื่องเอกสารและ ให้ความช่วยเหลือ ในการดำเนินงานเกี่ยวกับงบประมาณตลอดจนอุปกรณ์และเครื่องมือต่างๆที่ใช้ดำเนินโครงการ, นายประสิทธิ์ บุญเอนก ที่ให้คำปรึกษาในด้านการเขียนโปรแกรมไมโครคอนโทรลเลอร์, นายชิตพล หนูบรรจง สาขาวิชาเทคโนโลยีสารสนเทศ ที่ให้คำปรึกษาในด้านการเขียนโปรแกรม วิชาเว็บสีก และ สร้างฐานข้อมูล SQL server 2000, บุคลากรสาขาวิชาวิศวกรรมโทรคมนาคมที่ สนับสนุนด้านอุปกรณ์และการดำเนินงานและพี่น้องชาววิศวกรรมโทรคมนาคมทุกคนซึ่งให้กำลังใจเสมอมา

สุดท้ายนี้คณะผู้จัดทำขอกราบขอบพระคุณบิดา มารดา ที่อบรมเลี้ยงดู และให้โอกาสทางการ ศึกษาและคอยสนับสนุนด้วยดีตลอดมาอย่างหาที่เปรียบมิได้

จึงเห็นสมควรที่จะมอบคุณคุณความดี และเกียรติคุณเหล่านี้แก่ท่านที่กล่าวมานี้ รวมถึงบุคคลที่ มิได้กล่าวนามมา ณ ที่นี้ด้วย

นายวาณิช โสภิต

นายเฉลิมพงศ์ ชวนาห้วยคะโก

นายธรณบรรพ์ เจริญกัญญาภาพ

สารบัญ

เรื่อง	หน้า
-บทคัดย่อ	ก
-ถิตติกรรมประกาศ	ข
-สารบัญ	ค
-สารบัญภาพ	ง
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ระบบสื่อสาร	3
2.2 องค์ประกอบขั้นพื้นฐานของระบบสื่อสาร	4
2.3 สัญญาณและความถี่	5
2.4 รูปแบบของการส่งสัญญาณข้อมูล	7
2.5 การส่งสัญญาณข้อมูล	8
2.6 สื่อกลางในการสื่อสารข้อมูล	8
2.7 ระบบสื่อสารแอนะล็อก	13
2.8 ระบบสื่อสารดิจิทัล	15
2.9 Visual basic	19
2.10 SQL Server	28
บทที่ 3 การสร้างและการออกแบบ	30
3.1 การออกแบบวงจรเครื่องส่งสัญญาณคลื่นวิทยุ	30
3.2 การออกแบบวงจรเครื่องรับสัญญาณคลื่นวิทยุ	32
3.3 การเขียนโปรแกรมเพื่อรับ-ส่งข้อมูล ระหว่างตัวส่งและตัวรับสัญญาณวิทยุ	35
3.4 การออกแบบเพื่อเขียน โปรแกรม Visual basic	43

3.5 การสร้างฐานข้อมูลด้วย SQL server 2000	61
3.6 การสร้างวงจรเครื่องรับ-เครื่องส่ง	81
บทที่ 4 การทดลอง	87
4.1 การต่ออุปกรณ์	87
4.2 การใช้โปรแกรม Visual Basic เพื่อเรียกฐานข้อมูลมาแสดงบนจอคอมพิวเตอร์	91
4.3 ผลการทดลอง	102
4.4 วิเคราะห์การทดลอง	106
4.5 สรุปการทดลอง	108
บทที่ 5 สรุปผลและข้อเสนอแนะของโครงการ	109
5.1 สิ่งที่ได้จากจากโครงการ	109
5.2 ปัญหาและอุปสรรคในการทำโครงการ	110
5.3 ปัญหาของโครงการ	111
5.4 การแก้ปัญหาของโครงการ	113
5.5 ข้อจำกัดของโครงการ	114
5.6 ข้อเสนอแนะ	115
-ภาคผนวก ก	116
-ภาคผนวก ข	119
-ภาคผนวก ค	125
-ภาคผนวก ง	138
-ภาคผนวก จ	141
-เอกสารอ้างอิง	146
-ประวัติผู้เขียน	147



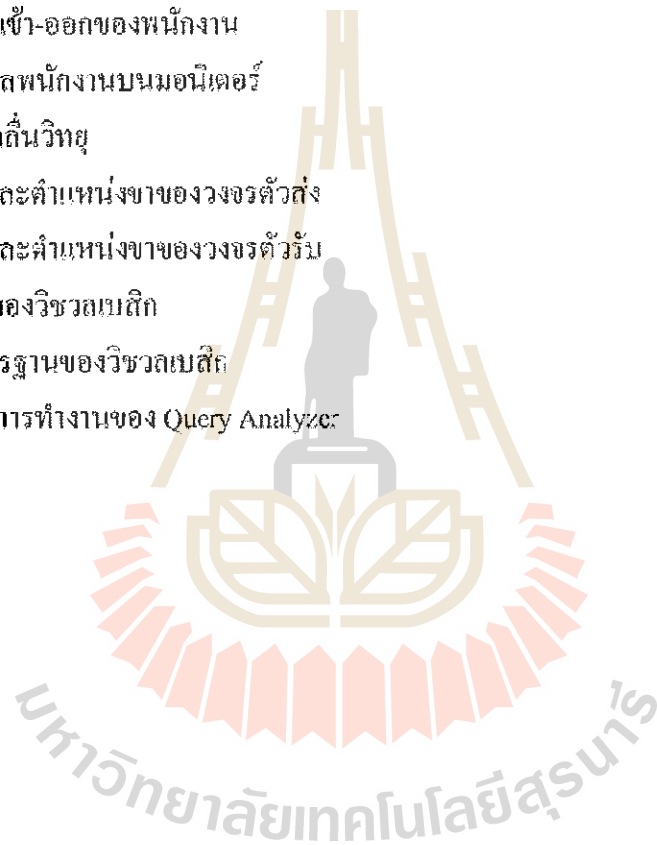
สารบัญภาพ

เรื่อง	หน้า
รูปที่ 2.1 องค์ประกอบขั้นพื้นฐานของระบบสื่อสาร	5
รูปที่ 2.2 รูปแบบของการสื่อสารในการรับส่งสัญญาณ	14
รูปที่ 2.3 การมอดูเลตแบบ Amplitude – Shift Keying (ASK)	17
รูปที่ 2.4 การมอดูเลตแบบ Frequency – Shift Keying (FSK)	18
รูปที่ 2.5 การมอดูเลตแบบ Phase – Shift Keying (PSK)	19
รูปที่ 2.6 โมเดลของสถาปัตยกรรม UDA ที่เกี่ยวข้องกับ Visual Basic	21
รูปที่ 2.7 แสดงการเพิ่มคอมโพเนนต์ MSComm	23
รูปที่ 2.8 แสดงบล็อกที่รายการ MSComm	24
รูปที่ 2.9 คอนโทรล MSComm พร้อมทำงาน	25
รูปที่ 3.1 ตั้งค่าอะแดปเตอร์ของวงจรเครื่องส่งสัญญาณคลื่นวิทยุ	30
รูปที่ 3.2 ลำดับการทำงานของโปรแกรมการส่งข้อมูล	31
รูปที่ 3.3 ตั้งค่าอะแดปเตอร์ของวงจรเครื่องส่งสัญญาณคลื่นวิทยุ	32
รูปที่ 3.4 ลำดับการทำงานของโปรแกรมการรับข้อมูล	33
รูปที่ 3.5 ลำดับการทำงานของโปรแกรม Visual Basic	43
รูปที่ 3.6 การสร้างฟอร์มเพื่อแสดงบนจอภาพ	
รูป(ก) ฟอร์ม 1	44
รูป(ข) ฟอร์ม 2	45
รูป(ค) ฟอร์ม 3	45
รูป(ง) MDI ฟอร์ม	46
รูปที่ 3.7 การเรียกเครื่องมือ Enterprise Manager ขึ้นมา	62
รูปที่ 3.8 แสดงฐานข้อมูลทั้งหมดที่มีอยู่ในเซิร์ฟเวอร์	63
รูปที่ 3.9 แสดงการสร้างฐานข้อมูล	63
รูปที่ 3.10 แสดงการตั้งชื่อฐานข้อมูล	64
รูปที่ 3.11 แสดงไดอะล็อกซ์ Database Properties	65
รูปที่ 3.12 แสดงการกำหนดค่าพารามิเตอร์	66

รูปที่ 3.13 แสดงการกำหนดคุณสมบัติต่างๆ ที่ไฟล์ทราบนเช็ทชั้นล็อก	67
รูปที่ 3.14 แสดงฐานข้อมูล Orders ที่เราได้ทำการสร้างขึ้น	68
รูปที่ 3.15 แสดงส่วนประกอบต่างๆ ของเครื่องมือ Query Analyzer	69
รูปที่ 3.16 แสดงการลบฐานข้อมูล โดยใช้ Enterprise Manager	71
รูปที่ 3.17 แสดงทำการเลือกฐานข้อมูลที่ต้องการ	72
รูปที่ 3.18 แสดงโคโอะลิตี้ Database Properties	73
รูปที่ 3.19 แสดงการกำหนดคุณสมบัติของไฟล์กลุ่ม	74
รูปที่ 3.20 แสดงการสร้างตารางด้วย SQL Enterprise Manager	
รูป(ก) แสดงการเลือกเซิร์ฟเวอร์ และเลือกฐานข้อมูลที่ต้องการจะสร้างฐานข้อมูล	76
รูป(ข) แสดงการกำหนดฟิลด์แต่ละฟิลด์, กำหนดชนิดข้อมูล, ขนาดข้อมูล, และอื่นๆ	77
รูป(ค) แสดงการกรอกข้อมูลลงในฐานข้อมูล	78
รูป(ง) แสดงฐานข้อมูลที่ได้ออกแบบเรียบร้อยแล้ว	78
รูปที่ 3.21 แสดงการลบตารางด้วย SQL Enterprise Manager	79
รูปที่ 3.22 แสดงการแก้ไขตารางด้วย Enterprise Manager	80
รูปที่ 3.23 แผนภาพชุดวงจรเครื่องส่ง	82
รูปที่ 3.24 รูปสัญญาณข้อมูลขณะที่ไม่มีการส่งสัญญาณจากวงจรเครื่องส่งไปยังวงจรเครื่องรับ	83
รูปที่ 3.25 รูปสัญญาณข้อมูลขณะมีการส่งสัญญาณจากวงจรเครื่องส่งไปยังวงจรเครื่องรับ	83
รูปที่ 3.26 แผนภาพชุดวงจรเครื่องรับ	86
รูปที่ 4.1 วงจรเครื่องส่งสัญญาณคลื่นวิทยุ	87
รูปที่ 4.2 โครงสร้างภายนอกของวงจรเครื่องรับสัญญาณคลื่นวิทยุ	89
รูปที่ 4.3 โครงสร้างภายในของวงจรเครื่องรับสัญญาณคลื่นวิทยุ	89
รูปที่ 4.4 การต่อพอร์ตอนุกรมของคอมพิวเตอร์กับวงจรเครื่องรับสัญญาณคลื่นวิทยุ	90
รูปที่ 4.5 การสร้างโปรเจกต์บน Desktop เป็น file.exe	91
รูปที่ 4.6 กำหนดชื่อโปรเจกต์ที่สร้างขึ้น	92
รูปที่ 4.7 ปรัชญาโปรเจกต์ บน Desktop	93
รูปที่ 4.8 หน้าค่างแสดงข้อมูลของพนักงาน	94
รูปที่ 4.9 แสดงข้อมูลของพนักงาน	95
รูปที่ 4.10 แสดงการเรียกฟอร์มสำหรับเพิ่ม, ลบ และแก้ไขข้อมูลพนักงานเข้าไปในฐานข้อมูล	96
รูปที่ 4.11 แสดงการเพิ่มข้อมูลพนักงานในฐานข้อมูล	97

รูปที่ 3.13 แสดงการกำหนดคุณสมบัติต่างๆ ที่ไฟล์ที่ฐานข้อมูล	67
รูปที่ 3.14 แสดงฐานข้อมูล Orders ที่เราได้ทำการสร้างขึ้น	68
รูปที่ 3.15 แสดงส่วนประกอบต่างๆ ของเครื่องมือ Query Analyzer	69
รูปที่ 3.16 แสดงการลบฐานข้อมูล โดยใช้ Enterprise Manager	71
รูปที่ 3.17 แสดงทำการเลือกฐานข้อมูลที่ต้องการ	72
รูปที่ 3.18 แสดงไดอะล็อกซ์ Database Properties	73
รูปที่ 3.19 แสดงการกำหนดคุณสมบัติของไฟล์ที่รีป	74
รูปที่ 3.20 แสดงการสร้างตารางด้วย SQL Enterprise Manager	
รูป(ก) แสดงการเลือกเซิร์ฟเวอร์ และเลือกฐานข้อมูลที่ต้องการจะสร้างฐานข้อมูล	76
รูป(ข) แสดงการกำหนดฟิลด์แต่ละฟิลด์, กำหนดชนิดข้อมูล, ขนาดข้อมูล, และอื่นๆ	77
รูป(ค) แสดงการกรอกข้อมูลลงในฐานข้อมูล	78
รูป(ง) แสดงฐานข้อมูลที่ได้ออกแบบเรียบร้อยแล้ว	78
รูปที่ 3.21 แสดงการลบตารางด้วย SQL Enterprise Manager	79
รูปที่ 3.22 แสดงการแก้ไขตารางด้วย Enterprise Manager	80
รูปที่ 3.23 แผนภาพชุดวงจรเครื่องส่ง	82
รูปที่ 3.24 รูปสัญญาณข้อมูลขณะที่ไม่มีการส่งสัญญาณจากวงจรเครื่องส่งไปยังวงจรเครื่องรับ	83
รูปที่ 3.25 รูปสัญญาณข้อมูลขณะมีการส่งสัญญาณจากวงจรเครื่องส่งไปยังวงจรเครื่องรับ	83
รูปที่ 3.26 แผนภาพชุดวงจรเครื่องรับ	86
รูปที่ 4.1 วงจรเครื่องส่งสัญญาณคลื่นวิทยุ	87
รูปที่ 4.2 โครงสร้างภายนอกของวงจรเครื่องรับสัญญาณคลื่นวิทยุ	89
รูปที่ 4.3 โครงสร้างภายในของวงจรเครื่องรับสัญญาณคลื่นวิทยุ	89
รูปที่ 4.4 การต่อพอร์ตอนุกรมของคอมพิวเตอร์กับวงจรเครื่องรับสัญญาณคลื่นวิทยุ	90
รูปที่ 4.5 การสร้างโปรเจกต์บน Desktop เป็น file.exe	91
รูปที่ 4.6 กำหนดชื่อโปรเจกต์ที่สร้างขึ้น	92
รูปที่ 4.7 ปรากฏโปรเจกต์ บน Desktop	93
รูปที่ 4.8 หน้าค่างแสดงข้อมูลของพนักงาน	94
รูปที่ 4.9 แสดงข้อมูลของพนักงาน	95
รูปที่ 4.10 แสดงการเรียกฟอร์มสำหรับเพิ่ม, ลบ และแก้ไขข้อมูลพนักงานเข้าไปในฐานข้อมูล	96
รูปที่ 4.11 แสดงการเพิ่มข้อมูลพนักงานในฐานข้อมูล	97

รูปที่ 4.12 แสดงการลบข้อมูลพนักงานในฐานข้อมูล	98
รูปที่ 4.13 แสดงการแก้ไขข้อมูลพนักงานในฐานข้อมูล	99
รูปที่ 4.14 แสดงการตรวจสอบเวลาเข้า-ออกของพนักงาน	100
รูปที่ 4.15 แสดงข้อมูลของพนักงานเมื่อมีพนักงานเข้า-ออกที่ทำงาน	101
รูปที่ 4.16 แสดงสถานะประตูเปิด	103
รูปที่ 4.17 แสดงสถานะประตูปิด	104
รูปที่ 4.18 แสดงข้อมูลพนักงานบนมอนิเตอร์	104
รูปที่ 4.19 แสดงเวลาเข้า-ออกของพนักงาน	105
รูปที่ 4.20 แสดงข้อมูลพนักงานบนมอนิเตอร์	105
วงจรรีจิสเตอร์สัญญาณคลื่นวิทยุ	116
โครงสร้างภายนอกและตำแหน่งขาของวงจรรีจิสเตอร์	117
โครงสร้างภายนอกและตำแหน่งขาของวงจรรีจิสเตอร์	118
แสดงส่วนประกอบของวงจรเบสิก	138
แสดงคอนโทรลมาตรฐานของวงจรเบสิก	140
ทูลบาร์ ที่ใช้ควบคุมการทำงานของ Query Analyzer	145



บทที่ 1

บทนำ

1.1 ความเป็นมา

เทคโนโลยี RFID (Radio Frequency Identification) กำลังมีบทบาทในชีวิตประจำวันของเราเป็นอย่างมากซึ่งปัจจุบันนี้ได้มีการนำเทคโนโลยีRFIDมาใช้ในระบบการรักษาความปลอดภัยและการจัดการฐานข้อมูลภายในบริษัทและหน่วยงานไทยทั้งภาครัฐและเอกชน โดย RFID มีหลักการการทำงานดังนี้ คือ RFID card จะมีรหัสเฉพาะของแต่ละบุคคลบรรจุอยู่โดยรหัสดังกล่าวเป็นไบนารี8บิตซึ่งรหัสดังกล่าวจะถูกส่งไปในรูปของคลื่นวิทยุไปที่เครื่องรับที่ติดอยู่กับประตู และจะทำให้ประตูเปิดเองโดยอัตโนมัติ จากนั้นรหัสนี้จะถูกแปลงให้เป็นข้อมูลของพนักงานเพื่อเก็บประวัติการเข้า-ออกบริษัท รวมทั้งข้อมูลอื่นๆ เช่น ตำแหน่ง เงินเดือน เป็นต้น ซึ่งข้อมูลทั้งหมดจะถูกแสดงที่จอมอนิเตอร์ของเจ้าหน้าที่รักษาความปลอดภัย และข้อมูลเหล่านี้จะถูกเก็บในระบบฐานข้อมูลของบริษัทด้วย

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการส่ง-รับข้อมูลแบบรหัสดังกล่าวโดยผ่านคลื่นวิทยุ
2. เพื่อศึกษาการใช้ Microcontroller ในการกำหนดรหัสเฉพาะของพนักงานแต่ละคนและทำการควบคุมการเปิด-ปิดประตู
3. เพื่อศึกษาโปรแกรมVisual basicเพื่อสร้างและแสดงฐานข้อมูลผ่านจอมอนิเตอร์
4. เพื่อศึกษาและออกแบบสถานการณ์จำลอง ในการควบคุมการเปิด-ปิดประตูและแสดงฐานข้อมูลผ่านจอมอนิเตอร์โดยใช้ RFID ได้
5. เพื่อรวบรวมความรู้ที่ได้ศึกษาจากภาคทฤษฎี มาใช้ในการทำงานได้จริง

1.3 ขอบเขตของโครงการ

- 1.สร้างวงจรเครื่องรับ-ส่งคลื่นวิทยุ เพื่อให้เครื่องส่งทำหน้าที่เป็นบัตรประจำตัวพนักงาน และเครื่องรับทำหน้าที่เป็นตัวอ่านข้อมูลจากบัตรประจำตัวพนักงาน
- 2 .เขียน โปรแกรมไมโครคอนโทรลเลอร์ลงในเครื่องส่งเพื่อกำหนดรหัสเฉพาะของพนักงานแต่ละคนและควบคุมเครื่องรับให้ถอดรหัสนี้เฉพาะของพนักงาน จากนั้นใช้ไมโครคอนโทรลเลอร์ควบคุมในการเปิด-ปิดประตู
- 3.เขียน โปรแกรมVisual basicเพื่อแสดงรายละเอียดของพนักงานแต่ละคนบนจอมอนิเตอร์ โดยโปรแกรมนี้จะเป็นตัวเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับฐานข้อมูลในMicrosoft Access

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาและทำการทดลองส่ง-รับข้อมูลแบบรหัสไบนารีโดยผ่านคลื่นวิทยุ
2. เขียนโปรแกรมเพื่อใช้งาน Microcontroller ในการควบคุมการเปิด-ปิดประตู
3. ทำการทดลองส่งรหัสไบนารีโดยผ่านคลื่นวิทยุไปที่เครื่องรับที่ติดอยู่กับประตูเพื่อให้ประตูเปิด-ปิดเองโดยอัตโนมัติ
4. เขียนโปรแกรม Visual basic เพื่อสร้างและแสดงฐานข้อมูลผ่านจอมอนิเตอร์
5. นำส่วนประกอบต่างๆมาทำการทดลองและวิเคราะห์ร่วมกันในสถานการณ์จำลอง
6. ตรวจสอบเช็คความถูกต้องและแก้ไขข้อผิดพลาด
7. ศึกษาและออกแบบเพื่อติดตั้งอุปกรณ์การควบคุมการเปิด-ปิดประตูอีกจำนวน 2 ประตูให้เป็นทำงานร่วมกันแบบหลายประตู(Multiple)
8. ทำการทดลองและวิเคราะห์ให้ทำงานร่วมกันแบบหลายประตู(Multiple)ในสถานการณ์จำลอง
9. ตรวจสอบเช็คความถูกต้องและแก้ไขข้อผิดพลาด
10. สรุปและประเมินผล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้การส่ง-รับข้อมูลแบบรหัสไบนารีโดยผ่านคลื่นวิทยุ
2. ได้เรียนรู้การใช้งาน Microcontroller และนำไปใช้งานได้จริง
3. ได้เรียนรู้การใช้โปรแกรม Visual basic เพื่อสร้างและแสดงฐานข้อมูล
4. สามารถนำผลงานไปใช้งานได้จริง

บทที่ 2 ทฤษฎีและหลักการ

2.1 ระบบสื่อสาร [1]

การติดต่อสื่อสารของมนุษย์มีมาช้านานนับแต่มีการก่อกำเนิดขิงมนุษยชาติ การสื่อสารยุคแรกๆ มีรูปแบบแตกต่างกันไป ไม่ว่าจะเป็นการใช้ภาษา การแสดงอาการปฏิกิริยา ท่าทาง รูปภาพ สัญญาณควัน และวิธีอื่นๆ ในอดีตการสื่อสารเกิดขึ้นได้เฉพาะในขอบเขตที่จำกัดกล่าวคือ การสื่อสารของคนสองคนจะมีประสิทธิภาพได้จะต้องอยู่ในระยะที่สามารถมองเห็นกันหรืออยู่ใกล้กันเท่านั้น หากบุคคลทั้งสองอยู่ห่างไกลกันการสื่อสารก็จะยากลำบากมากขึ้น การติดต่อสื่อสารของมนุษย์ได้เปลี่ยนแปลงอย่างมาก นับแต่การคิดค้นระบบโทรเลข (telegraph) การคิดค้นนี้ช่วยให้มนุษย์สามารถส่งสัญญาณไฟฟ้ารูปพัลส์ผ่านสายนำสัญญาณทองแดง (copper wire) ระยะทางไกลได้ ต่อมาได้มีการคิดค้นระบบโทรศัพท์ (telephone) ขึ้นทำให้มนุษย์สามารถติดต่อสนทนากันแม้จะอยู่ห่างกันในระยะทางไกลได้ โดยอาศัยหลักการแปลงสัญญาณเสียงให้อยู่ในรูปของสัญญาณไฟฟ้าและส่งสัญญาณไฟฟ้านั้นผ่านสายนำสัญญาณทองแดงไปได้ระยะทางไกลๆ จากนั้นที่ปลายทางก็จะแปลงสัญญาณไฟฟ้านั้นกลับมาเป็นสัญญาณเสียงอีกครั้ง การค้นพบนี้เป็นจุดเริ่มต้นของการพัฒนาระบบโทรศัพท์ที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบันนั่นเอง

เป็นที่ยอมรับว่าในอนาคตอันใกล้นี้ เราจะได้พบกับการสื่อสารที่มีให้เลือกมากมาย การติดต่อข้ามประเทศอาจทำได้ง่ายและรวดเร็วขึ้น และระบบสื่อสารยังได้อำนวยความสะดวกสบายเป็นอย่างมากด้วย

2.2 องค์ประกอบขั้นพื้นฐานของระบบสื่อสาร [2]

ระบบสื่อสารส่วนใหญ่แบ่งองค์ประกอบพื้นฐานของระบบสื่อสารออกเป็น 6 อย่าง ดังรูปที่ 2.1

สามารถจำแนกออกเป็นส่วนประกอบได้ดังนี้

1. ผู้ส่งข่าวสารหรือแหล่งกำเนิดข่าวสาร(source) อาจจะเป็นสัญญาณต่างๆ เช่น สัญญาณภาพ ข้อมูล และเสียง เป็นต้น ในการติดต่อสื่อสารสมัยก่อนอาจใช้แสงไฟ คandles ไฟ ก็นับว่าเป็นแหล่งกำเนิดข่าวสาร เช่นกัน

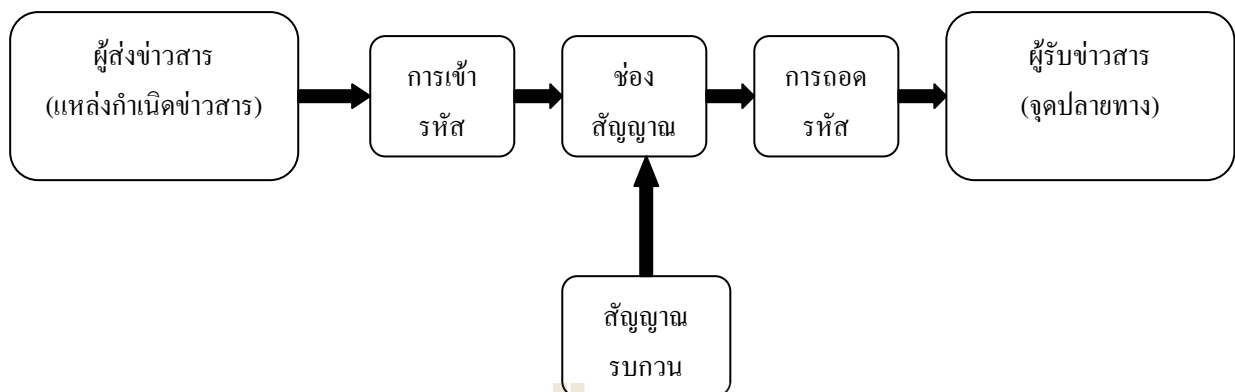
2. ผู้รับข่าวสารหรือจุดหมายปลายทางของข่าวสาร(sink) ซึ่งจะรับรู้จากสิ่งที่ผู้ส่งข่าวสารหรือแหล่งกำเนิดข่าวสารส่งผ่านมาให้ ถ้าผู้รับสารหรือจุดหมายปลายทางไม่ได้รับข่าวสาร ก็แสดงว่าไม่มีการสื่อสารเกิดขึ้นนั่นเอง

3. ช่องสัญญาณ(Channel) ในที่นี้อาจหมายถึงสื่อกลางหรือตัวกลางที่ข่าวสารเดินทางผ่าน อาจจะเป็นอากาศ สายนำสัญญาณต่างๆ หรือแม้กระทั่งของแข็งของเหลว เช่น น้ำ น้ำมัน เป็นต้น เปรียบเสมือนเป็นสะพานที่จะให้ข่าวสารข้ามจากอีกฝั่งหนึ่ง ไปอีกฝั่งหนึ่งนั่นเอง

4. การเข้ารหัส(encoding) เป็นการช่วยให้ผู้ส่งข่าวสารและผู้รับข่าวสารมีความเข้าใจตรงกันในการสื่อความหมาย จึงมีความจำเป็นต้องแปลงความหมายนี้ การเข้ารหัสจึงหมายถึงการแปลงข่าวสารให้อยู่ในรูปพลังงานที่พร้อมจะส่งไปในสื่อกลาง ทางผู้ส่งมีความสามารถในการแปลงพลังงานที่พร้อมจะส่งไปในสื่อกลาง และ ทางผู้รับมีความสามารถในการแปลงพลังงานให้กลับไปอยู่ในรูปข่าวสารที่สามารถเข้าใจได้ ข้อสำคัญคือความเข้าใจต้องตรงกันระหว่างผู้ส่งและผู้รับ หรือมีรหัสเดียวกันการสื่อสารจึงจะเกิดขึ้น

5. การถอดรหัส(decoding) หมายถึงการที่ผู้รับข่าวสารแปลงพลังงานจากสื่อกลางให้ไปอยู่ในรูปของข่าวสารที่ส่งมาจากผู้ส่งข่าวสาร โดยมีความเข้าใจหรือรหัสตรงกัน

6. สัญญาณรบกวน(noise) เป็นสิ่งที่มีอยู่ในธรรมชาติ มักจะลดทอนหรือรบกวนระบบ อาจเกิดขึ้นได้ทั้งทางด้านผู้ส่งข่าวสาร ผู้รับข่าวสาร และช่องสัญญาณ แต่ในการศึกษาขั้นพื้นฐานมักจะสมมติให้ทางด้านผู้ส่งข่าวสารและ ผู้รับข่าวสารไม่มีความผิดพลาด ตำแหน่งที่ใช้วิเคราะห์ มักจะเป็นที่ตัวกลางหรือช่องสัญญาณ เมื่อไรที่รวมสัญญาณรบกวนด้านผู้ส่งข่าวสารและ ผู้รับข่าวสาร สิ่งที่จะส่งมาจะเป็นข่าวสารจากแหล่งกำเนิดรวมกับสัญญาณรบกวนทางด้านส่ง ในทางปฏิบัติมักจะใช้วงจรกรอง (filter) กรองสัญญาณรบกวนแต่ต้นทาง เพื่อให้การสื่อสารมีคุณภาพดียิ่งขึ้นแล้วค่อยดำเนินการ เช่น การเข้ารหัสแหล่งข้อมูล เป็นต้น



รูปที่ 2.1 องค์ประกอบขั้นพื้นฐานของระบบสื่อสาร

2.3 สัญญาณและความถี่ [2]

สัญญาณที่ใช้ในระบบสื่อสารแบ่งออกได้เป็น 2 ประเภทคือ สัญญาณแอนะล็อก และสัญญาณดิจิทัล สัญญาณแอนะล็อกเป็นสัญญาณที่มีขนาดเป็นค่าต่อเนื่อง ส่วนสัญญาณดิจิทัลเป็นสัญญาณที่มีขนาดเปลี่ยนแปลงเป็นค่าของเลขลงตัว โดยปกติมักแทนด้วยระดับแรงดันที่แดงสถานะ “ 0 ” และ “ 1 ” หรืออาจจะมีหลายสถานะ ในระบบสื่อสารดิจิทัล มีค่าที่ตั้งไว้ (threshold) เป็นค่าบอกสถานะ ถ้าสูงกว่าค่าที่ตั้งไว้ สถานะเป็น “ 1 ” ถ้าต่ำกว่าค่าที่ตั้งไว้ สถานะเป็น “ 0 ” ซึ่งมีข้อดีในการทำให้เกิดความผิดพลาดน้อยลง เนื่องจากสัญญาณรบกวนต้องมีค่าสูงกว่าค่าที่ตั้งไว้สถานะจึงจะเปลี่ยน ในขณะที่ระบบแอนะล็อก สัญญาณรบกวนจะเติมเข้าไปในสัญญาณจริงโดยตรง กล่าวคือ สัญญาณจริงบวกสัญญาณรบกวนเป็นสัญญาณขณะนั้น ทำให้สัญญาณรบกวนมีผลต่อสัญญาณจริงและมีความผิดพลาดเกิดขึ้น ตามปกติแล้วคลื่นที่ใช้ในการติดต่อสื่อสารได้แก่คลื่นแม่เหล็กไฟฟ้า ซึ่งมีความสัมพันธ์ระหว่างความถี่ ความยาวคลื่น และความเร็วในการเคลื่อนที่ของคลื่นดังนี้

$$\text{ความถี่} = \frac{\text{ความเร็ว}}{\text{ความยาวคลื่น}}$$

ความยาวคลื่นนี้เป็นการวัดระหว่างจุดสูงสุดหรือจุดต่ำสุดของคลื่นทั้งสองคลื่นที่อยู่ติดกัน ความเร็วของแสงในอากาศเป็น 3×10^8 เมตรต่อวินาที ความถี่มีหน่วยเป็นเฮิรตซ์ (hertz หรือ Hz) ช่วงของความถี่เรียกว่า สเปกตรัม (spectrum) และในสเปกตรัมนี้แบ่งเป็นช่วงๆ เรียกว่า แบนด์ (band) อันเป็นที่มาของความกว้างของช่วงคลื่นหรือแถบความถี่เรียกว่า แบนด์วิดท์ (bandwidth) ตารางที่ 2.1 แสดงสเปกตรัมที่ความถี่ต่างๆ ตามมาตรฐานข้อตกลงระหว่างประเทศ ซึ่งกำหนดโดยองค์การโทรคมนาคมนานาชาติ (International Telecommunication Union หรือ ITU) คุณสมบัติของคลื่นต่างๆ ขึ้นอยู่กับความถี่ที่ใช้งาน ความถี่สูงจะมีความยาวคลื่นที่สั้นกว่า แต่มีอำนาจทะลุทะลวงได้ดีกว่า ในขณะที่ความถี่ต่ำจะมีความยาวคลื่นที่มากกว่า ทำให้คลื่นเดินทางได้ไกลกว่า ความรู้เรื่องความถี่เป็นสิ่งสำคัญมากเนื่องจากการใช้งานประยุกต์ใช้งานที่ความถี่ย่านต่างๆ ในระบบสื่อสาร

แถบความถี่	ชื่อเรียก (ชื่อย่อ) ของแถบความถี่
3-10 kHz	Extremely low frequency (ELF)
10-30 kHz	Very low frequency (VLF)
30-300 kHz	Low frequency (LF)
3-10 MHz	Medium frequency (MF)
3-30 MHz	High frequency (HF) (หรือ short wave)
30-300 MHz	Very high frequency (VHF)
300-3000 MHz	Ultra high frequency (UHF) (หรือ microwave)
3-30 GHz	Super high frequency (SHF)
30-300 GHz	Extremely high frequency (EHF)

โดยที่ $1000 \text{ kHz} = 1 \text{ MHz}$

$1000 \text{ MHz} = 1 \text{ GHz}$

ตารางที่ 2.1 แสดงสเปกตรัมที่ความถี่ต่างๆ

2.4 รูปแบบของการส่งสัญญาณข้อมูล [3]

ในกระบวนการส่งสัญญาณข้อมูล จากเครื่องผู้ส่ง ไปยังเครื่องผู้รับนั้นสามารถแบ่งรูปแบบการส่ง

สัญญาณข้อมูล ออกเป็น 4 รูปแบบดังนี้

1. **แบบทิศทางเดียว (Simplex)** การส่งสัญญาณประเภทนี้ จะเป็นการส่งไปในทิศทางเดียวกันเท่านั้น และตลอดเวลา ยกตัวอย่างเช่นการส่งสัญญาณกระจายเสียงของวิทยุ หรือ

การแพร่ภาพทางโทรทัศน์ ซึ่งจะเห็นได้ว่าทางสถานีได้ส่งสัญญาณเสียง หรือสัญญาณภาพมายังเครื่องรับที่เป็นวิทยุ หรือ โทรทัศน์แต่ในขณะที่เดียวกันทางเครื่องรับนั้น ๆ ไม่สามารถที่จะส่งสัญญาณ หรือข้อมูลกลับมายังสถานีได้

2. **แบบกึ่งทางคู่ (Half Duplex)** สำหรับการส่งสัญญาณในรูปแบบกึ่งทางคู่นี้ จะเป็นลักษณะเช่นเดียวกับวิทยุสื่อสารกล่าวคือรูปแบบของการส่งสัญญาณ ในลักษณะนี้จะเป็นแบบต้องผลัดกันส่งสัญญาณและรับสัญญาณ ยกตัวอย่างเช่นวิทยุสื่อสารของตำรวจ หรือทหารที่ต้องผลัดกันพูดบางครั้งมักเรียกการส่งสัญญาณแบบนี้ว่าการสื่อสารสายคู่ (Two – Wire Line)
3. **แบบทางคู่ (Full Duplex)** การสื่อสารแบบทางคู่นี้ ลักษณะการ สื่อสารนั้นจะสามารถส่งข้อมูลได้พร้อม ๆ กันทั้งสองทาง เสมือนกับการพูดคุยสนทนากัน หรือการใช้ โทรศัพท์ ซึ่งโดยปกติแล้วลักษณะการพูดคุยจะสลับกัน แต่การสนทนา หรือการพูดโทรศัพท์นั้นจะกลายเป็น การสื่อสารแบบทางคู่ ได้ นั้น ในกรณีที่เสียง หรือทะเลาะกัน จะเห็นได้ว่าทั้ง 2 ฝ่ายนั้นสามารถพูด พร้อมกันได้ ซึ่งการสื่อสารแบบนี้บางครั้ง เรียกการสื่อสารแบบนี้ว่าการสื่อสาร 4 สาย (Four Wire Line)
4. **แบบสะท้อนสัญญาณ (Echo – Plex)** การสื่อสารแบบนี้จะเป็นการรวมการสื่อสารแบบครึ่งคู่เพล็กซ์ หรือคู่เพล็กซ์เต็มเข้าไว้ด้วยกัน ซึ่งการทำงานจะเป็นลักษณะสะท้อนแสดงให้เห็นภาพ ในขณะที่ยังทำงานควบคู่ไปด้วย ตัวอย่างเช่น การใช้เครื่อง ATM (Auto Teller Machine) ที่เมื่อผู้ใช้ป้อนคำสั่งผ่าน Keyboard เพื่อสั่งให้ Host Computer ทำงาน ขณะเดียวกันหน้าจอก็จะ แสดงคำสั่งที่ผู้ใช้สั่งป้อนเข้าไป (การทำงานนี้จะเป็นลักษณะของ Full Duplex)

2.5 การส่งสัญญาณข้อมูล [2], [3]

การส่งสัญญาณข้อมูลเปรียบเสมือนกับพนักงานไปรษณีย์ที่ต้องคอยรับจดหมายจากไปรษณีย์ไปยังผู้รับตามหน้าของจดหมาย ซึ่งพนักงานไปรษณีย์นี้เอง ถือว่าเป็นสื่อหรือตัวกลางในการดำเนินการของ กระบวนการ ทั้งหมด ซึ่งในการดำเนินการของสื่อ หรือตัวกลาง อาจจะอยู่ในรูปของสัญญาณเสียง คลื่นแม่เหล็กไฟฟ้า หรือแสงก็ได้ นอกจากนี้ตัวกลาง หรือสื่อยังสามารถจำแนกออกเป็นได้ 2 ประเภทคือ

1. กลุ่มที่สามารถกำหนดเส้นทางของสัญญาณได้ (Guided Media) ในกลุ่มนี้จะได้แก่ กลุ่มที่มีสื่อ หรือตัวกลาง เป็นจำพวกสายที่ต่อเชื่อมกัน เช่นต่อเชื่อมระหว่างผู้รับและผู้ส่งด้วยสายคู่ตีเกลียว (Twisted Pair Cable) หรือแม้แต่เชื่อมด้วย สายใยแก้วนำแสง (Fiber Optic) เป็นต้น

2. กลุ่มที่ไม่สามารถกำหนดเส้นทางของสัญญาณได้ (Unguided Media) กลุ่มนี้เป็นกลุ่มจำพวกที่ไม่สามารถกำหนดเส้นทางของการสื่อสารได้เลย หรือเรียกว่าเป็นสื่อกลางไร้สาย ซึ่งสื่อกลางประเภทนี้ ได้แก่ น้ำ สุญญากาศ หรือแม้แต่ชั้นบรรยากาศ

2.6 สื่อกลางในการสื่อสารข้อมูล [3]

สื่อกลางในการสื่อสารข้อมูล หรือตัวกลางนั้น สามารถแบ่งได้ออกเป็น 2 กรณี ดังที่ได้กล่าวมาแล้วข้างต้น ว่าประกอบไปด้วย จำพวกกำหนดเส้นทางได้ และจำพวกที่ไม่สามารถกำหนดเส้นทางได้ ซึ่งพวกที่สามารถ กำหนดเส้นทางได้ จะเป็นพวกที่ใช้สายสื่อสาร เป็นสื่อกลาง กล่าวคือสามารถกำหนดว่าให้ข้อมูล ไปยังทิศทางใดและส่วนพวกที่ไม่สามารถกำหนดเส้นทางได้ การส่งสัญญาณจะเป็นลักษณะเช่นเดียวกับการแพร่กระจาย การเลือกใช้ว่าจะให้ระบบเครือข่ายใช้ตัวกลางหรือสื่อชนิดใดจำเป็นต้องขึ้นอยู่กับตัวแปรหลายอย่าง ยกตัวอย่างเช่น ราคา อัตราค่าบริการ ความเร็วในการรับ-ส่งข้อมูล ภูมิประเทศ รวมถึงเทคโนโลยีที่มีอยู่ด้วย ต่อจากนี้จะแบ่งอธิบายชนิดของสื่อกลางในการสื่อสาร ออกเป็น 2 ประเภทดังนี้

2.6.1 สื่อกลางที่สามารถกำหนดเส้นทางได้[3]

1. สายคู่ตีเกลียว (Twisted Pair Cable) จำพวกสื่อกลางที่สามารถกำหนดเส้นทางได้นั้น สายคู่ตีเกลียว นี้จะมีราคาต่ำที่สุด ในสายสัญญาณจะประกอบไปด้วยสายทองแดง 2 เส้น และแต่ละเส้นจะมีฉนวนหุ้มจับคู่ พันกันเป็นเกลียว สายคู่ตีเกลียว 1 คู่ สามารถใช้แทนช่องทางได้ 1 ช่องทาง ยกตัวอย่างสายคู่ตีเกลียวที่สามารถ พบเห็นได้ทั่ว ๆ ไป เช่น สายโทรศัพท์ เป็นต้น สายคู่ตีเกลียวแต่ละเส้นจะมีฉนวนหุ้ม ซึ่งช่วยในการลด การรบกวนจากสนามแม่เหล็กไฟฟ้าได้ แต่ไม่สามารถป้องกันการสูญเสียพลังงานจากการแผ่รังสีความร้อน นอกจากนี้สายคู่ตีเกลียวยังสามารถใช้รับ – ส่งสัญญาณได้ทั้งแบบ Analog และแบบ Digital

2. สายโคแอกเชียล (Coaxial Cable) เป็นสายที่มีคุณภาพดีกว่าสายคู่ตีเกลียว และเป็นที่ยอมรับใช้เช่นเดียวกัน เนื่องจากเป็นสายที่มีคุณสมบัติเด่นอยู่ 2 ประการคือ มีฉนวนภายนอกที่มีความคงทนต่อการใช้งานสูง และเป็นช่องทางการสื่อสารที่มีความกว้างมาก ข้างในของสายประกอบไปด้วยลวดทองแดงอยู่ตรงกลาง เส้นทองแดงเส้นใหญ่ตรงกลาง เรียกว่า “ สายนำสัญญาณภายใน ” (Inner Conductor) มีชั้นโลหะเหนียวนำ หุ้มอยู่ 2 ชั้น ชั้นในสุดเป็นฟืนเกลียว หรือชั้นแข็ง ชั้นนอกเป็นฟืนเกลียว และคั่นระหว่างชั้นด้วยฉนวนหนา ชั้นนอกสุดเป็นฉนวน สายโคแอก หรือโคแอกเชียลนั้นสามารถม้วน โค้งงอได้ง่าย และมีอยู่ 2 แบบคือแบบ 70 โอห์ม และ 50 โอห์ม การจัดกลุ่มของสายทำได้โดยการกำหนดตัวเลข RG สายโคแอกสามารถส่งสัญญาณ ข้อมูลได้ทั้งช่องทางแบบ Broadband และ Baseband สำหรับการส่งสัญญาณจะทำได้เพียง 1 ช่องทาง และเป็นแบบ Half Duplex ตัวอย่างการใช้สายโคแอกในปัจจุบันที่ใช้กันมากได้แก่ สายเคเบิลทีวี และสายโทรศัพท์ทางไกล สายส่งข้อมูลในระบบเครือข่าย LAN แบบ Thick และ Thin Ethernet หรือใช้ในการเชื่อมโยงสั้น ๆ ระหว่างอุปกรณ์อิเล็กทรอนิกส์
3. สายใยแก้วนำแสง (Fiber - Optic Cable) สายใยแก้วนำแสงเป็นสายสัญญาณชนิดใหม่ ซึ่งการทำงานจะใช้สัญญาณแสงเป็นตัวส่งนำข้อมูล (โดยการเปลี่ยนสัญญาณไฟฟ้า ให้เป็นคลื่นแสงก่อน) เมื่อได้สัญญาณแสงแล้วจึงค่อยส่งแสงออกไป แกนของสายใยแก้วประกอบไปด้วย ท่อขนาดเล็กจำนวนมาก ทำด้วยแก้วบริสุทธิ์ หรือวัสดุใยแก้ว ซึ่งท่อแต่ละอันเรียกว่า “ ท่อใยแก้วนำแสง ” มีขนาดประมาณเล็กเท่าเส้นผม สายใยแก้วนำแสง 1 อันจะประกอบไปด้วยท่อใยแก้วนำแสงจำนวนมาก ซึ่งเรียกชื่อสายใยแก้วตามจำนวนท่อที่อยู่ข้างใน เช่นถ้ามีท่อใยแก้ว 10 เส้น ก็เรียกว่าสาย 10 Core เนื่องจากสายใยแก้วมีขนาดเล็ก แต่มีความกว้างของช่องทางการสื่อสารข้อมูล ทำให้สามารถ ถ่ายทอดข้อมูลปริมาณสูง ด้วยเวลาที่สั้นกว่าสายประเภทอื่น อัตราความเร็วจะเริ่มต้นที่ 100 ล้านบิตต่อวินาที รวมไปถึงการถ่ายทอดข้อมูลผ่านสายลวดทองแดงนั้น มีความปลอดภัยต่ำ เนื่องจากการขโมยสัญญาณ (Tapping) สามารถกระทำได้ง่าย โดยที่ผู้ใช้เองก็ไม่สามารถรู้ได้ เทคนิคการเคลือบท่อสายใยแก้ว นี้ที่เรียกว่า Heat Fused Cladding ทำให้การ Tapping เกือบเป็นไปไม่ได้ แม้ว่าจะ Tapping ได้ ผู้ใช้ก็สามารถตรวจจับที่ปลายทางได้ เพราะการ Tapping ทำให้ความเข้มของลำแสงนั้นเปลี่ยนไป

สายใยแก้วนำแสงนั้นสามารถแบ่งออกได้ 3 ประเภท โดยแบ่งตามวิธีการส่งสัญญาณดังนี้

1. Multimode Step Index เป็นสายใยแก้วชนิดแรก วัสดุที่เคลือบท่อใยแก้วเป็นสารประเภทพลาสติก หรือสารสะท้อนแสง ทำหน้าที่สะท้อนแสงที่ส่งออกมาให้อยู่ในท่อจนกระทั่งไปทะลุออกทางปลายสาย
2. Multimode Graded Index สายชนิดนี้จะไม่มีการใช้สายเคลือบที่ผิว แต่อาศัยคุณสมบัติที่แตกต่างกัน ของวัสดุที่นำมาสร้างเป็นสายใยแก้ว แต่หลักการทำงานก็เช่นเดียวกันกับแบบ Step กล่าวคือจะช่วยให้การ สะท้อนแสงที่ส่งออกมาให้อยู่ในท่อจนกระทั่งไปทะลุออกทางปลายสาย
3. Single Mode เป็นสายที่มีความเร็วในการทำงานสูงสุดของทั้ง 3 แบบ สายชนิดนี้จะเป็นสาย ที่มีเส้นผ่าศูนย์กลางขนาดเล็กมาก มีหลักการทำงานคือส่งแสงออกไปเป็นแนวเส้นตรง ไม่ ต้องกระทบ หรือสะท้อนใด ๆ ใด

2.6.2 สื่อกลางที่ไม่สามารถกำหนดเส้นทางได้ [3]

เป็นสื่อกลางที่เป็นลักษณะของการกระจายคลื่น ซึ่งไม่มีลักษณะทางกายภาพ สื่อประเภทกระจายคลื่น (Radiated Media) เรียกอีกอย่างว่า สื่อประเภทไร้สาย (Wireless Media) ซึ่งสามารถส่งข้อมูล ผ่านได้ทั้งทางอากาศ น้ำ หรือแม้แต่ในสุญญากาศได้ รูปแบบของการสื่อแบบไร้สายนี้ที่รู้จักกันมีอยู่มากมาย อาทิเช่น คลื่นวิทยุ สัญญาณไมโครเวฟ ทั้งแบบภาคพื้น และแบบดาวเทียม วิทยุ เซลลูลาร์ สัญญาณอินฟราเรด ซึ่งสัญญาณแต่ละชนิดมีความถี่แตกต่างกัน ซึ่งสัญญาณบางตัวอาจจะมีความถี่ใกล้เคียงกันทำให้เกิดปัญหา สัญญาณทับซ้อนกันได้ ดังนั้นจึงจำเป็นต้องมีองค์กรทำหน้าที่ควบคุม และอนุญาตในการใช้คลื่นสัญญาณ ที่ต้องการใช้

1. คลื่นวิทยุ (Broadcast Radio) คลื่นวิทยุนี้เป็นคลื่นที่นิยมใช้กันทั่วไป ข้อดีของการใช้คลื่นวิทยุจะอยู่ที่มีการเชื่อมโยง และติดต่อได้ง่าย ซึ่งทำเพียงแค่ต่ออุปกรณ์รับ – ส่งวิทยุเข้ากับอุปกรณ์คอมพิวเตอร์ ก็สามารถติดต่อสื่อสารข้อมูลได้ทั้งภายใน และภายนอกองค์กรได้ การส่งสัญญาณคลื่นวิทยุนี้ จะทำโดยการ แพร่กระจายคลื่นไปยังทุกทิศทาง ดังนั้นตัวรับสัญญาณ หรือเสารับสัญญาณ จึงไม่จำเป็นต้องชี้มายังจุดที่ส่ง สัญญาณแต่อย่างใด
2. คลื่นไมโครเวฟ (Microwave) การเดินทางของสัญญาณไมโครเวฟจะเดินทางเป็นลักษณะเส้นตรง จึงเรียกได้ว่าเป็นสัญญาณทิศทางเดียว และเป็นการส่งข้อมูลแบบรับช่วงต่อ ๆ กัน เป็นหอน (สถานี) ซึ่งความห่างของแต่ละสถานีจะห่างกัน ประมาณ 40-48 กิโลเมตร และอาจจะไกลถึง 88 กิโลเมตรได้ ถ้าสถานีที่อยู่สูงห่างจากพื้นดินมาก ๆ นั่นก็แสดงให้เห็นว่า ยิ่งสถานี (หอน) มีความสูงมาก การส่งสัญญาณก็สามารถส่ง ได้ไกลขึ้น ประกอบกับการถ่ายทอดสัญญาณจะมีความถี่สูงมากทำให้สามารถส่งข้อมูลด้วยอัตราความเร็ว ที่สูงมา

เช่นเดียวกัน ในการวางตำแหน่งจำเป็นต้องคำนึงถึงทิศทางของเสาอากาศ รวมถึงสภาพดินฟ้าอากาศอีกด้วย ปัจจุบันสื่อกลางไมโครเวฟเป็นที่นิยมใช้กันมากขึ้น เนื่องจากมีราคาถูกกว่าการสื่อสารสายใยแก้วนำแสง หรือดาวเทียม แต่ให้ประสิทธิภาพในการรับ – ส่งข้อมูลรวดเร็วกว่ากันนั่นเอง

3. ดาวเทียม (Satellite) ดาวเทียมถือเป็นคลื่นไมโครเวฟชนิดหนึ่ง หรือก็คือสถานีไมโครเวฟลอยฟ้า นั่นเอง การทำงานจะประกอบไปด้วยสถานีภาคพื้นดินตั้งแต่ 2 สถานีขึ้นไป การทำงานจะทำการทบทวนและขยายสัญญาณข้อมูล รับ-ส่งสัญญาณข้อมูลไปยังดาวเทียม ซึ่งการส่งสัญญาณจากภาคพื้นดิน ไปยัง ดาวเทียม เรียกว่า สัญญาณขาขึ้น (Uplink) ในทางกลับกันถ้าส่งสัญญาณจากดาวเทียมมายัง พื้นดิน เรียกว่า สัญญาณขาลง (Downlink) ข้อเสียของการใช้สัญญาณดาวเทียมจะอยู่ตรงที่สัญญาณข้อมูล อาจจะถูกรบกวนจากสัญญาณภาคพื้นอื่น ๆ ได้ รวมถึงจะมีเวลาประวิง (Delay Time) ในการส่งสัญญาณ ยกตัวอย่างเช่น การรับสัญญาณ โทรทัศน์จากดาวเทียม และจากเสาอากาศ จะเห็นได้ ชัดเจนว่าเครื่องที่รับสัญญาณ โทรทัศน์จากดาวเทียมจะแพร่ภาพช้ากว่าเครื่องที่รับสัญญาณจากเสาอากาศนั่นเอง ซึ่งทำให้ระบบการส่งสัญญาณดาวเทียมไม่เหมาะสำหรับการสื่อสารข้อมูลแบบ On-Line หรือแบบ Real Time และที่สำคัญคือมีการลงทุนในการติดตั้งสูงการสื่อสารข้อมูลโดยผ่านทางดาวเทียม นั้นที่นิยมใช้ใน ปัจจุบันได้แก่ การใช้ระบบเครือข่ายสถานีดาวเทียมขนาดเล็ก หรือเครือข่ายวีแซท (VSAT : Very Small Aperture Terminal) ซึ่งการติดตั้งและเคลื่อนย้ายทำได้โดยง่าย และเสียค่าใช้จ่ายน้อยกว่า การลงทุนทำ เครือข่ายดาวเทียมเองมาก
4. แสงอินฟราเรด (Infrared) ลำแสงอินฟราเรดเป็นคลื่นแม่เหล็กไฟฟ้าที่มีความถี่ระหว่างแสงที่ตามองเห็น กับ คลื่นสัญญาณวิทยุ การสื่อสารข้อมูลโดยอินฟราเรดนั้นจริง ๆ มักคุ้นเคยกันเป็นอย่างดีในชีวิตประจำวัน เช่นการใช้รีโมทคอนโทรลกับเครื่องรับโทรทัศน์ การส่งสัญญาณอินฟราเรดนั้นสามารถ ส่งด้วยอัตราความเร็ว สูงสุด ประมาณ 4 Mbps ดังนั้นการใช้แสงอินฟราเรดจึงเหมาะกับงานที่ไม่เน้นการส่งข้อมูลปริมาณมาก ๆ นอกจากนั้นข้อจำกัดของการใช้อุปกรณ์แสงอินฟราเรดคือต้องไม่มีสิ่งกีดขวางทางเดินของแสง และระยะ การส่งก็ได้ไม่ไกลนัก จึงเหมาะกับการใช้อุปกรณ์ภายในห้องเดียวกัน (ระยะการส่งสัญญาณได้ประมาณ 30-80 ฟุต)
5. วิทยุเซลลูลาร์ (Cellular Radio) ถ้าจะกล่าวถึงการส่งสัญญาณด้วยวิทยุเซลลูลาร์อาจจะดูไม่คุ้นเคยนัก แต่ถ้าเรียกเสียใหม่ว่า โทรศัพท์เคลื่อนที่ หรือ โทรศัพท์เคลื่อนที่ โทรศัพท์มือถือ คงจะคุ้นเคยมากกว่า ซึ่งการทำงานของวิทยุเซลลูลาร์นี้จะใช้สื่อประเภทคลื่นสัญญาณวิทยุซึ่งการรับส่ง สัญญาณจะต้องอยู่ในพื้นที่หนึ่งที่เรียกว่า Cell ซึ่งมีลักษณะเป็นรูปร่างสี่เหลี่ยมหรือการรับส่งสัญญาณ ในแต่ละเซลล์นั้นจะมีเสาอากาศเป็นของตัวเอง และใช้คลื่นสัญญาณที่มี พลังงานต่ำเพื่อป้องกันการรบกวน ระหว่าง

กัน การใช้งานจะเป็นลักษณะการติดต่อระหว่างเสาอากาศของแต่ละเซลล์ กับเครื่องโทรศัพท์มือถือ ซึ่งจะมีการโอนติดต่อระหว่างกัน Cell ที่มีพื้นที่อยู่ติดกัน(Roaming)ไม่สามารถใช้คลื่นความถี่ในสัญญาณ อันเดียวกันได้ เพราะจะทำให้เกิดสัญญาณ รบกวน แต่ใน Cell ที่มีพื้นที่ไม่ติดกันนั้นสามารถ ใช้ความถี่เดียวกันได้ ยกตัวอย่างเช่น Cell A ติดกับ Cell B จะไม่สามารถใช้ความถี่อันเดียวกัน แต่สำหรับ Cell C สามารถใช้คลื่นความถี่อันเดียวกันกับ Cell A ได้เนื่องจากว่ามีพื้นที่ ๆ ไม่ติดกัน

6. วิทยุแบบสเปกตรัมแผ่กระจาย (Spread Spectrum Radio) วิทยุแบบสเปกตรัมแผ่กระจาย นี้จะใช้วิธีการส่ง สัญญาณออกไปหลายคลื่นความถี่พร้อม ๆ กัน ภายในแถบคลื่นที่กำหนด แรกทีเดียวเทคนิคแบบนี้ ได้รับการพัฒนาขึ้น เพื่อใช้ในกิจการทางทหารซึ่งต้องการ ความเชื่อถือได้ในระดับสูงมากในระหว่างการรบ ซึ่งระบบนี้จะทำสัญญาณให้ถูกส่งออกไปในหลายความถี่พร้อมกันทำให้การดักฟังได้ยากขึ้น รวมทั้งการรบกวนการสื่อสารก็ยากมากขึ้นด้วย เพราะต้องค้นหาความถี่ทั้งหมดให้ได้ปัจจุบันได้นำเทคนิคแบบนี้มาใช้ สำหรับ การสื่อสารไร้สายของระบบเครือข่ายเฉพาะบริเวณ [LAN]

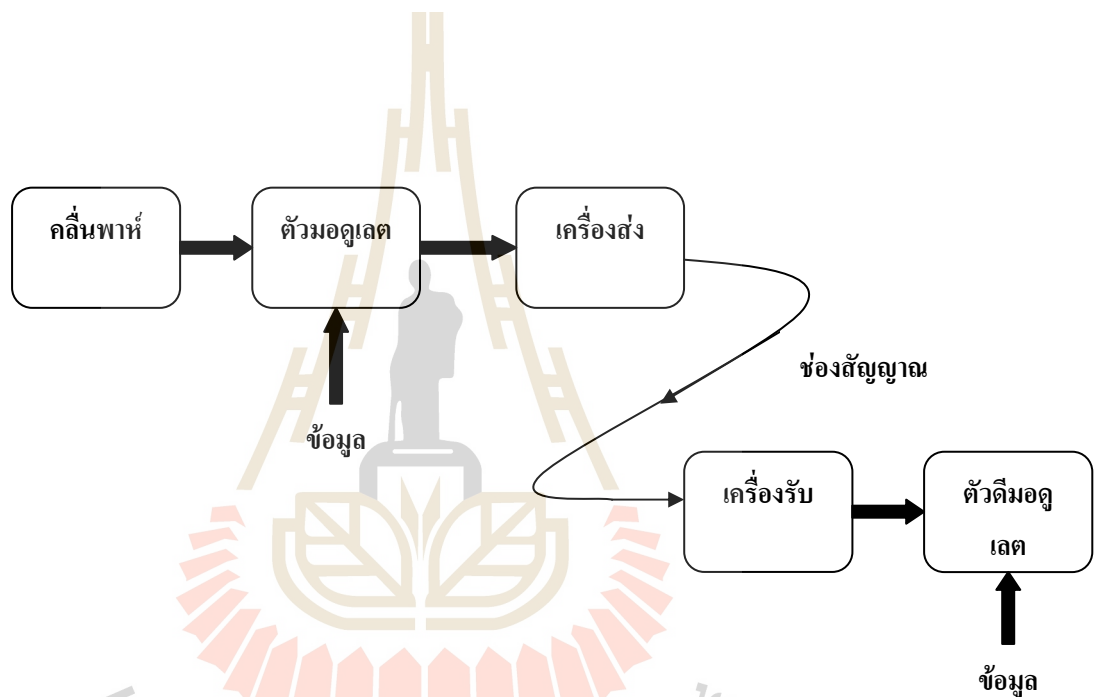
2.7 ระบบสื่อสารแอนะล็อก [2]

การสื่อสารแอนะล็อกเป็นระบบที่ออกแบบให้ส่งสัญญาณแอนะล็อก เช่น สัญญาณเสียง ปัญหาสำคัญสำหรับการสื่อสารแอนะล็อกก็คือ เรื่องสัญญาณรบกวน ซึ่งบางครั้งอาจทำให้ระบบไม่สามารถใช้งานได้เลย ดังนั้นการสื่อสารดิจิทัลจึงได้เริ่มเข้ามาแทนที่ อย่างไรก็ดี การสื่อสารดิจิทัลก็ยังไม่สามารถแทนที่การสื่อสารแอนะล็อกได้ทั้งหมด เนื่องจากสัญญาณในธรรมชาติทั้งหมดเป็นสัญญาณแอนะล็อกจึงยังคงเห็นการพัฒนาการสื่อสารแบบแอนะล็อกอยู่ในปัจจุบัน ซึ่งรูปแบบการสื่อสารที่มีประสิทธิภาพบางรูปแบบก็ยังคงใช้อยู่ในปัจจุบัน เช่น การมอดูเลตแอมพลิจูด (amplitude modulation หรือ AM) การมอดูเลตความถี่ (frequency modulation หรือ FM) สัญญาณ โทรศัพท์ เป็นต้น

2.7.1 การมอดูเลต [3], [4]

การมอดูเลต (modulation) เป็นการผสมสัญญาณของข้อมูลเข้าไปกับสัญญาณอีกสัญญาณหนึ่งเรียกว่า คลื่นพาห์ (carrier) ซึ่งสัญญาณนี้มีความถี่ที่เหมาะสมกับช่องสัญญาณนั้นๆ เพื่อให้ข้อมูลที่ส่งเข้าไปในช่องสัญญาณเดินทางได้ไกลมากขึ้น รูป

ที่ 2.2 แสดงรูปแบบของการสื่อสารในการรับส่งสัญญาณอย่างง่าย โดยคลื่นพาห้ผสม สัญญาณข้อมูลที่ตัวมอดูเลต (modulator) แล้วส่งไปที่เครื่องส่ง จากเครื่องส่งไปยัง เครื่องรับจะเป็นช่องสัญญาณสำหรับลำเลียงสัญญาณผสมนี้ สัญญาณผสมจาก เครื่องรับจะไปเข้าตัวดีมอดูเลต (demodulator) เพื่อแยกสัญญาณข้อมูลออกมา การ เลือกรูปมอดูเลตขึ้นอยู่กับปัจจัยหลายประการเช่น ชนิดของช่องสัญญาณ แบนด์วิดท์ ประสิทธิภาพของระบบที่ต้องการ และความต้านทานต่อสัญญาณรบกวน เป็นต้น เมื่อ มีการมอดูเลตที่ภาคส่ง การที่จะรับข้อมูลได้นั้นต้องมีการดีมอดูเลต (demodulation) ที่ ภาครับด้วย



รูปที่ 2.2 รูปแบบของการสื่อสารในการรับส่งสัญญาณ

1. การมอดูเลตแอมพลิจูด (amplitude modulation หรือ AM) วิธีนี้เป็นวิธีการที่ดั้งเดิม และ สะดวกสบายที่สุด เนื่องจากว่าความถี่ของ สัญญาณคลื่นพาห้จะคงที่และสูงกว่าความถี่ของ สัญญาณข้อมูล แต่วิธีการนี้ก็มีข้อเสียอยู่ตรงที่ว่า แบนด์วิดท์ของสัญญาณ AM นั้นเป็นย่าน ความถี่ที่ไม่ค่อยสูงนัก ทำให้เกิด Noise เข้ามารบกวนได้โดยง่าย นอกจากนี้ยังสิ้นเปลือง พลังงานในการส่งสูงอีกด้วย
2. การมอดูเลตความถี่ (frequency modulation หรือ FM) เป็นวิธีการที่ตรงกันข้ามกับ AM ใน การ Modulation แบบ FM นี้จะมี Amplitude คงที่ แต่ความถี่ของสัญญาณจะไม่คงที่ เปลี่ยนไปตามความถี่ของสัญญาณข้อมูล แต่ข้อเสียของสัญญาณ FM นี้จะอยู่ที่ว่า มีการใช้

แบนด์วิดท์ที่มีขนาดกว้าง เนื่องจากสัญญาณข้อมูลมีหลาย ความถี่ ทำให้ต้องการสายสื่อสารที่มี แบนด์วิดท์กว้าง และทำให้ราคาของสายสูงขึ้นตามไปด้วย คุณภาพของสัญญาณ FM นั้นจะดีกว่าแบบ AM แต่ระบบการทำงานจะซับซ้อนมากกว่า ปัจจุบันมีการเพิ่มวงจร Phase-Lock-Loop เข้าไปในระบบเพื่อปรับความถี่ และเฟสของสัญญาณ ทำให้สัญญาณมีคุณภาพ และถูกต้องยิ่งขึ้น

3. การมอดูเลตเฟส (phase modulation หรือ PM) วิธีการ Modulation แบบ AM และ FM นิยมใช้ในการกระจายเสียง ของวิทยุ แต่แบบ PM กลับนิยมใช้ในการแพร่ภาพทางโทรทัศน์ ซึ่งวิธีการ PM นี้สามารถเปลี่ยนแปลง เฟสของสัญญาณได้ง่ายกว่าแบบ FM แต่เป็นวิธีที่มีวงจรค่อนข้างซับซ้อนมากกว่าจึงไม่ค่อยเป็นที่นิยม ในการส่งข้อมูลแบบ Analog ในการ Modulation แบบ PM นี้ให้ครีกรอบของสัญญาณเป็นมุมเฟสเท่ากับ 180 องศา และเมื่อครบ 1 รอบ ก็เท่ากับ 360 องศา โดยเมื่อสัญญาณ Modulate จะมีการเปลี่ยนกลับมุมเฟส ทุกครั้งที่สัญญาณคลื่นพาห์ เท่ากับ 180 องศา

2.8 ระบบสื่อสารดิจิทัล [2]

ความต้องการรับส่งข้อมูลข่าวสารในปัจจุบันที่มีมากทั้งปริมาณและชนิดของข้อมูล ทำให้เกิดการพัฒนาระบบการขยายตัวของระบบสื่อสารข้อมูลเป็นอย่างมาก ในยุคที่เริ่มมีการรับส่งข้อมูลระบบต่างๆ ทำงานแบบแอนะล็อกทั้งหมด ต่อมาเมื่อเทคโนโลยีทางด้านดิจิทัลได้ก้าวหน้าขึ้นมา จึงได้เริ่มมีการนำเทคโนโลยีดิจิทัลเข้าไปทดแทนแบบแอนะล็อกเดิม ทั้งการนำไปทดแทนทั้งหมด สัญญาณดิจิทัลมักจะหมายถึงสัญญาณในระบบเลขฐานสองได้แก่ “0” และ “1” เท่านั้น “0” และ “1” นี้แทนค่าความหมายต่างๆ ของข้อมูลได้ แต่ในความเป็นจริง สัญญาณดิจิทัลเป็นสัญญาณที่มีการแบ่งค่าต่างๆ ออกเป็น 2, 4, 6, 8 หรือ 16 ระดับก็ได้ การเพิ่มระดับเป็นการช่วยในการใช้ประโยชน์จากช่องสัญญาณ และทำให้สมรรถนะที่มีความผิดพลาดต่ำอันเป็นเป้าหมายสูงสุดของระบบ ข้อได้เปรียบของสัญญาณดิจิทัลที่สำคัญคือ เนื่องจากสามารถนำสัญญาณไปประมวลผลได้หลายรูปแบบ โดยที่มีคุณภาพเท่ากันหรือดีกว่า สัญญาณแอนะล็อก สัญญาณแบบดิจิทัลจึงใช้งานได้หลากหลายกว่า อย่างไรก็ตาม สัญญาณดิจิทัลก็คือ สัญญาณแอนะล็อกที่มีการเปลี่ยนแปลงแบบทันทีทันใด ดังนั้นระบบที่ใช้กับแบบแอนะล็อกจะสามารถใช้กับสัญญาณแบบดิจิทัลได้ด้วย แต่จากการที่สัญญาณดิจิทัลก็คือ สัญญาณแอนะล็อกที่มีรูปแบบเฉพาะ การใช้เทคนิคต่างๆ ของระบบแอนะล็อกจึงเป็นการสิ้นเปลือง และได้มีเทคนิคของระบบดิจิทัลโดยเฉพาะขึ้นมา

2.8.1 การเข้ารหัส

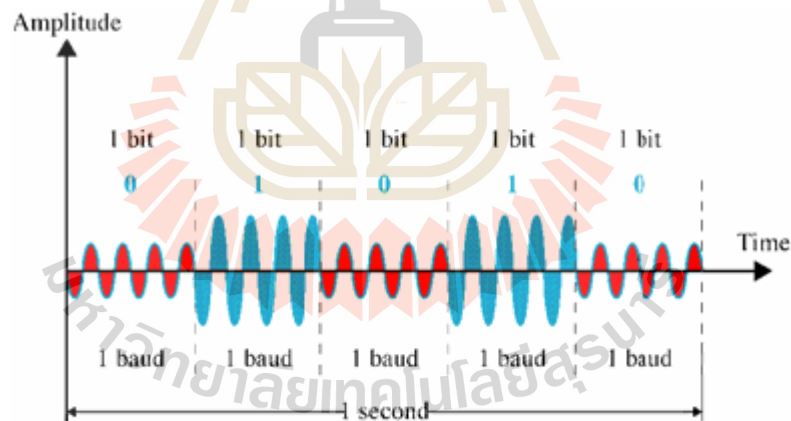
แหล่งกำเนิดข้อมูล (source) เช่น ภาพ เสียง หรือสัญญาณพูดที่ใช้ในการสื่อสารมักจะเป็นแหล่งกำเนิดสัญญาณซึ่งมีคุณสมบัติเฉพาะ การที่จะส่งสัญญาณเหล่านั้นไปได้มักจะใช้วิธีการเข้ารหัส (encoding) เพื่อเปลี่ยนรูปแบบของสัญญาณจากรูปแบบหนึ่งไปยังอีกรูปแบบหนึ่ง ตัวอย่างของการเข้ารหัสที่ใช้กันมากได้แก่

1. การเข้ารหัสแหล่งข้อมูล (source encoding) ทำหน้าที่ในการลดขนาดของข้อมูลดิจิทัลที่จะส่งให้มีการสูญเสียของข่าวสารน้อย ซึ่งรวมไปถึงการซ้ำซ้อนของข้อมูล โดยปกติแล้วสัญญาณดิจิทัลที่ได้จากการเข้ารหัสแหล่งข้อมูลมักจะน้อยกว่าสัญญาณดิจิทัลขาเข้าเริ่มแรก
2. การเข้ารหัสช่องสัญญาณ (channel encoding) หลักการนี้มักจะใช้เพื่อให้สัญญาณดิจิทัลขาออกมีความอ่อนไหวหรือแปรเปลี่ยนตามคุณภาพของช่องสัญญาณในการส่งน้อยลง โดยทั่วไปจะใช้รหัสตรวจจับและแก้ไขความผิดพลาดเข้ามาช่วย การเข้ารหัสช่องสัญญาณอาจจะรวมไปถึงการเปลี่ยนข้อมูลให้เป็นสัญญาณไฟฟ้าที่เหมาะสมกับสายนำสัญญาณ โดยการกระจายกำลังของสัญญาณออกไป
3. การเข้ารหัสสายหรือไลน์โค้ดดิ้ง (line coding) เป็นการเข้ารหัสสัญญาณดิจิทัลที่เป็นตัวเลขให้เป็นสัญญาณทางไฟฟ้า ในบางกรณีสัญญาณทางไฟฟ้าเหล่านี้สามารถที่จะส่งออกไปที่สายนำสัญญาณได้เลย ไม่จำเป็นต้องนำไปมอดูเลตกับคลื่นพาห้ด้วยวิธีการต่างๆ เพื่อส่งไปในช่องสัญญาณก็ได้ วิธีนี้เป็นการปรับสัญญาณให้เข้ากับสมบัติทางกายภาพของสายนำสัญญาณ
4. การเข้ารหัสลับและการถอดรหัสลับ (encryption and decryption) เป็นการป้องกันข่าวสารที่ส่งออกไปให้ผู้ที่ไม่ได้รับอนุญาตรับรู้ข้อมูลได้ โดยปกติมักเป็นการสื่อสารในรูปแบบเครือข่ายที่เป็นสาธารณะ เช่น เคเบิลทีวี เป็นต้น รูปแบบการเข้ารหัสมักจะเก็บไว้เป็นความลับ การเข้ารหัสแบบนี้อาจนำไปใช้กับงานที่ต้องการความเชื่อถือของข้อมูลสูงได้

2.8.2 รูปแบบการมอดูเลตสัญญาณดิจิทัล[5], [6]

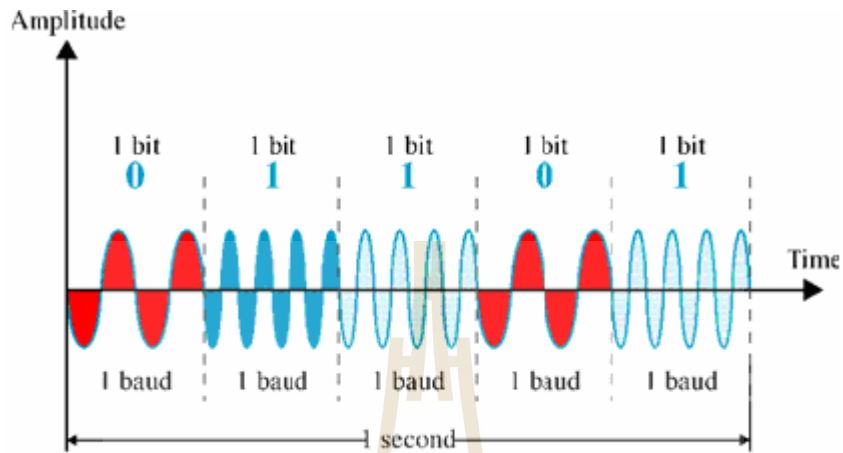
การมอดูเลตสัญญาณดิจิทัล (Digital modulation) เข้ากับคลื่นพาห้ที่เป็นสัญญาณไซน์ (sine) นั้นมีอยู่หลายรูปแบบ ทั้งนี้ก็เพื่อต้องการให้สัญญาณดิจิทัลเหล่านั้นสามารถส่งผ่านตัวกลางที่ออกแบบมาสำหรับสัญญาณแอนะล็อกได้ เช่น โครงข่ายโทรศัพท์พื้นฐาน ไมโครเวฟ เป็นต้น การมอดูเลตที่ใช้กันทั่วไปได้แก่

1. การมอดูเลตเชิงเลขทางแอมพลิจูดหรือ ASK (Amplitude – Shift Keying) เป็นวิธี modulation วิธีแรก ๆ ที่ทำให้สามารถส่งข่าวสารในรูปแบบของ electric pulse ไปในระยะไกล ๆ ได้ ASK ใช้เทคนิคของการปิด-เปิดสวิตช์ รูปแบบที่ง่ายที่สุดของ carrier คือสัญญาณ DC ที่มีระดับแรงดัน 0 โวลต์ หรือ ระดับที่สูงสุดที่จะเป็นไปได้ ในปัจจุบันไม่ว่าจะเป็นการส่งโดยใช้สายส่งหรือไม่ใช้สายส่ง จะใช้สัญญาณ sinusoidal ที่มีความถี่ในย่าน AF ของ HF เป็น carrier การเปิด-ปิดสวิตช์ของสัญญาณ carrier อาจจะใช้เครื่องกำเนิดโดยตรง หรืออาจจะใช้สัญญาณ clock ควบคุมเครื่องกำเนิดสัญญาณ carrier อีกชั้นหนึ่ง



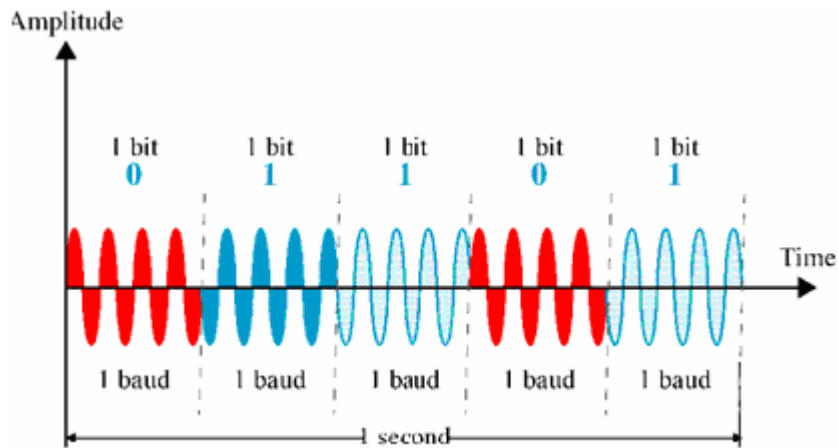
รูปที่ 2.3 การมอดูเลตแบบ Amplitude – Shift Keying (ASK)

2. การมอดูเลตเชิงเลขทางความถี่หรือ FSK (Frequency – Shift Keying) การ Modulate เชิงเลขทาง ความถี่ ขนาดของคลื่นพาห้ Carrier Wave จะไม่เปลี่ยน แต่ ความถี่ของคลื่นจะเปลี่ยนแทน โดยเมื่อบิตมีค่าเป็น 1 ความถี่จะสูงกว่าปกติ และเมื่อบิตมีค่าเป็น 0 ความถี่ของคลื่นจะต่ำกว่าปกติ



รูปที่ 2.4 การมอดูเลตแบบ Frequency – Shift Keying (FSK)

3. การมอดูเลตเชิงเลขทางเฟสหรือ PSK (Phase – Shift Keying) เป็นการ Modulate เชิงเลขทางเฟส คือค่าของขนาด และความถี่ของ คลื่นพาห้จะไม่มีการเปลี่ยนแปลง แต่เฟสของสัญญาณจะเป็นตัวเปลี่ยนแปลง กล่าวคือเมื่อสภาวะของบิตเป็น 0 หรือเป็น 1 เฟสของคลื่นจะเปลี่ยน (Shift) ไป 180 องศาด้วย ซึ่งวิธีนี้จะมีสัญญาณรบกวนเกิดขึ้นน้อยที่สุด ทำให้ได้สัญญาณที่มีคุณภาพดีที่สุด แต่ว่าจริงในการทำงานจะซับซ้อนกว่า และราคาอุปกรณ์ ก็แพงกว่าปกติ อีกด้วย



รูปที่ 2.5 การมอดูเลตแบบ Phase – Shift Keying (PSK)

2.9 Visual basic [7]

Visual Basic เป็นภาษาคอมพิวเตอร์ (Computer Language) ที่พัฒนาโดยบริษัท ไมโครซอฟท์ ซึ่งเป็นบริษัทยักษ์ใหญ่ที่สร้างปฏิบัติการ Window 95/98 และ Window NT ที่เราใช้กันอยู่ในปัจจุบัน โดยตัวภาษาเองมีรากฐานมาจากภาษา Basic ซึ่งย่อมาจาก Beginner's All Purpose Symbolic Instruction ถ้าแปลให้ได้ความหมายก็คือ “ชุดคำสั่งหรือภาษาคอมพิวเตอร์สำหรับผู้เริ่มต้น” ภาษา Basic มีจุดเด่นคือผู้ไม่มีพื้นฐานเรื่องการเขียนโปรแกรมเลยก็สามารถเรียนรู้และนำไปใช้งานได้ค่อนข้างง่ายและรวดเร็ว เมื่อเทียบกับการเรียนภาษาคอมพิวเตอร์อื่นๆ เช่น ภาษาซี (C), ปาสคาล (Pascal), ฟอรัแทรน (Fortran), หรือ แอสเซมบลี (Assembler) สาเหตุที่ Visual Basic เป็นภาษาที่เหมาะสมสำหรับการเรียนรู้ในการเขียนโปรแกรมนั้นเนื่องจาก Visual Basic มีข้อดีหลายประการคือ

1. ง่ายต่อการเรียนรู้เหมาะสำหรับผู้เริ่มต้นทั้งในเรื่องไวยากรณ์ของภาษาเองและเครื่องมือการใช้งาน ดังชื่อที่บอกว่า basic ซึ่งเหมาะสำหรับผู้เริ่มต้น

2. ความนิยมของตัวภาษา โดยอาจกล่าวได้ว่าภาษา Basic นั้นเป็นภาษาที่มีคนเรียนรู้และใช้งานมากที่สุดในประวัติศาสตร์ของคอมพิวเตอร์

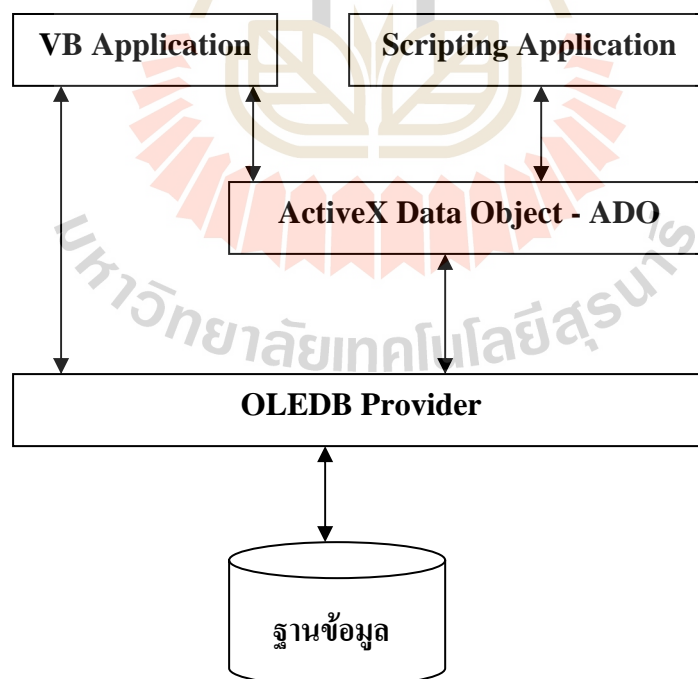
3. การพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วของการประมวลผล และในเรื่องของความสามารถใหม่ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต

4. ผู้พัฒนาสำคัญของ Visual Basic คือบริษัทไมโครซอฟท์ซึ่งจัดว่าเห็นยักษ์ของวงการคอมพิวเตอร์ ในปัจจุบัน เราจึงสามารถมั่นใจได้ว่า Visual Basic จะยังมีการพัฒนา ปรับปรุงและคงอยู่ไปอีกนาน

ถ้าจะกล่าวโดยสรุปถึงข้อดีของการศึกษา Visual Basic ก็คือเป็นภาษาที่ใช้ง่าย สะดวก มีผู้ใช้เยอะ มีการพัฒนาอย่างต่อเนื่อง มีประโยชน์มากมาย และมีความสามารถไม่แพ้ภาษาใดๆ ถ้าคิดจะเรียนรู้ภาษาคอมพิวเตอร์สักภาษาหนึ่งแล้วละก็ รับรองว่าไม่ผิดหวังที่เรียนรู้ Visual Basic

ทำความรู้จักกับเทคโนโลยี ADO [9]

เทคโนโลยี ADO ถูกออกแบบมาโดยอาศัยสถาปัตยกรรมที่เรียกว่า Universal Data Access เรียกสั้น ๆ ว่า UDA โดยมีแนวความคิดว่า ต้องการให้เป็นเทคโนโลยีที่จะสามารถเข้าถึงข้อมูลในฐานข้อมูลได้หลายชนิด เช่น ฐานข้อมูลประเภท Relation Database Management System (RDBMS), ข้อมูลที่ถูกจัดเก็บในรูปแบบอื่น ๆ เช่น ไฟล์ข้อความ ฐานข้อมูลของมาตรฐาน Index Sequential Access Method (ISAM) ซึ่งรวมถึงการเข้าถึงข้อมูลในฐานข้อมูลผ่านทางเครือข่ายอินเทอร์เน็ตด้วย การเขียนโปรแกรมด้านฐานข้อมูลโดยใช้เทคโนโลยี UDA จะอาศัย OLE Provider ประเภทต่างๆ ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลแต่ละชนิด โดยอาศัยสถาปัตยกรรม ADO ทำหน้าที่เป็นกลไกในการทำงาน



รูปที่ 2.6 โมเดลของสถาปัตยกรรม UDA ที่เกี่ยวข้องกับ Visual Basic

จากรูปที่ 2.5 จะเห็นได้ว่า ADO ทำหน้าที่เป็นตัวกลางที่จะไปสั่งให้ OLE Provider เข้าถึงข้อมูลในฐานข้อมูล ซึ่งเรียกโดยรวมว่า Microsoft Data Access Component (MDAC) โดยที่ Scripting Application เป็นการเข้าถึงข้อมูลในฐานข้อมูลผ่านทางเครือข่ายอินเทอร์เน็ต โดยอาศัยภาษาสคริปต์เป็นตัวเรียกการใช้งานคือ ภาษา Active Server Pages (ASP)

การเขียนโปรแกรมติดต่อและควบคุม Serial Port[10]

ถึงแม้ว่าการสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้ก็เพราะว่าการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตนานสามารถครั้งละหลาย ๆ บิต พร้อมกันส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน แต่ว่าการส่งข้อมูลแบบอนุกรมนั้นมีข้อที่เหนือกว่าการส่งข้อมูลแบบขนานคือ การสามารถส่งข้อมูลได้ในระยะที่ไกลแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย การสื่อสารแบบอนุกรมสามารถแบ่งออกเป็น 3 รูปแบบดังนี้

1. Simplex สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว
2. Half-Duplex สามารถส่งข้อมูลไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่ไม่สามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน
3. Full-Duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

นอกจากนี้แล้วยังสามารถแบ่งประเภทของการสื่อสารแบบอนุกรมตามลักษณะสัญญาณในการส่งได้อีก 2 แบบคือ

- การสื่อสารแบบซิงโครนัส (Synchronous) สำหรับการสื่อสารแบบซิงโครนัสนี้จะใช้สัญญาณนาฬิกาควบคุมการรับส่งสัญญาณ เช่น สายเคเบิลคอมพิวเตอร์ โดยจะมีสายสัญญาณเส้นหนึ่งเป็นสายสัญญาณนาฬิกา ส่วนอีกเส้นหนึ่งเป็นสายของข้อมูล (และมักจะมีสายกราวด์ด้วย)

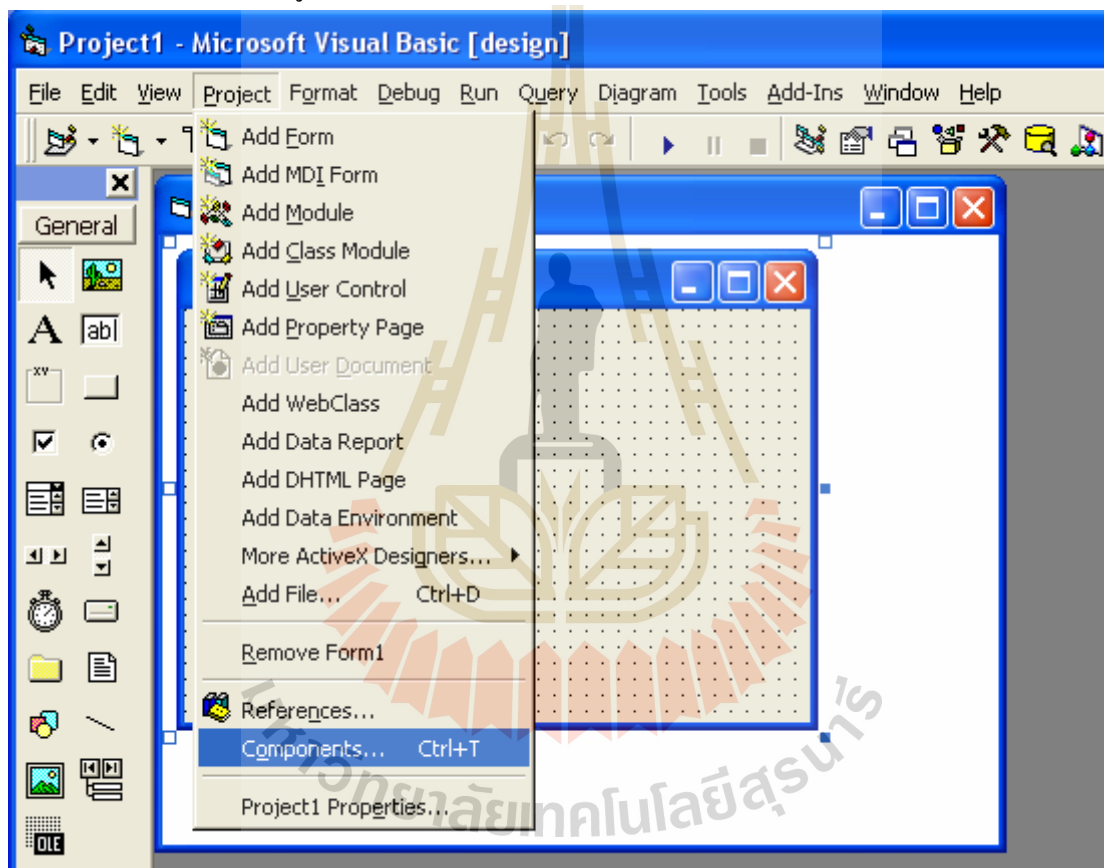
สำหรับการสื่อสารแบบซิงโครนัสนี้เหมาะสำหรับการทำงานในระยะใกล้ข้อมูลที่จะส่งมีไม่มากนัก เพราะถ้าระยะไกลขึ้นจะทำให้สัญญาณนาฬิกามีปัญหา อีกทั้งต้องมีสายหลายเส้นทำให้สิ้นเปลืองมาก

- การสื่อสารแบบอะซิงโครนัส (Asynchronous) สำหรับการสื่อสารแบบอะซิงโครนัสนั้นจะใช้สายข้อมูลเพียงตัวเดียว แต่จะใช้รูปแบบการส่งข้อมูล หรือ Bit Pattern เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล, ส่วนไหนเป็นตัวข้อมูล, ส่วนไหนจะเป็นตัวตรวจสอบความถูกต้องของข้อมูล และส่วนไหนเป็นส่วนปิด

ท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาคส่ง และภาครับ
ซึ่งจะมีอุปกรณ์พิเศษที่ชื่อว่า UART หรือ Universal Asynchronous
Receive/Transmitter คอยควบคุมการรับ และส่งข้อมูล

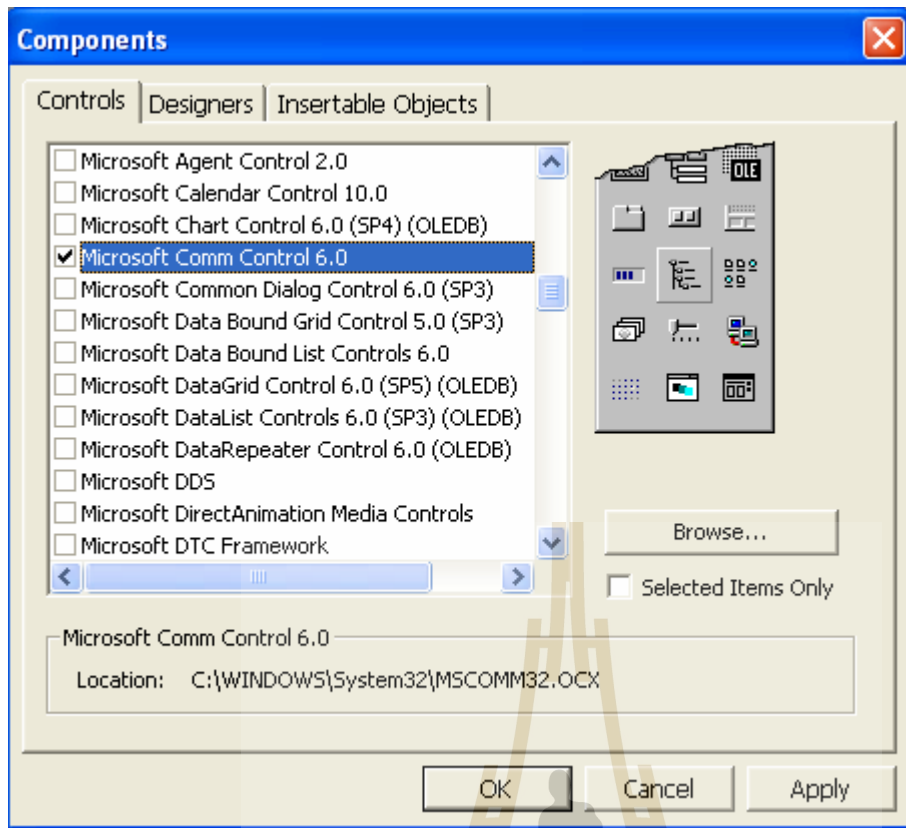
การเขียนโปรแกรมติดต่อ และควบคุม Serial Port กับ Visual Basic [10]

คอนโทรลที่สำคัญในการทำให้ Visual Basic สามารถสื่อสารผ่านพอร์ตอนุกรมได้นั้นก็คือ
คอนโทรล MSComm ซึ่งไม่ใช่คอนโทรลมาตรฐาน ดังนั้นถ้าเราต้องการใช้งาน MSComm เรา
จะต้องทำการเพิ่มคอนโทรลนี้เข้าไปใน ToolBox ซึ่งสามารถกระทำได้โดยคลิกขวาที่ ToolBox
แล้วเลือก Component ดังรูป



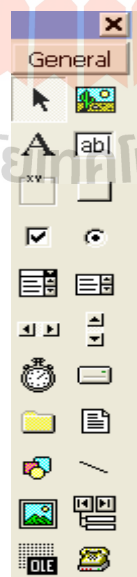
รูปที่2.7 แสดงการเพิ่มคอมโพเนนต์ MSComm

จากนั้นจะปรากฏไดอะล็อก Component ขึ้นมา จากนั้นให้คลิกเลือกที่ Microsoft Comm
Control6.0 คลิกปุ่ม ดังรูป



รูปที่ 2.8 แสดงเลือกที่รายการ MSComm

จากนั้นก็ปรากฏภายใน Toolbox จะมีไอคอนรูปโทรศัพท์ ซึ่งเป็นไอคอนของคอนโทรล MSComm ปรากฏขึ้นมาให้เลือกใช้งาน



รูปที่ 2.9 คอนโทรล MSComm พร้อมทำงาน

พรีอพเพอร์ตีที่สำคัญในการใช้งาน MSComm

CommPort ใช้ในการกำหนดหมายเลขของพอร์ตอนุกรมที่เราต้องการจะติดต่อรูปแบบของการใช้งานดังนี้

```
object.CommPort [ = value ]
```

ยกตัวอย่างเช่น เรากำหนดให้การเขียนโปรแกรมติดต่อกับพอร์ต Com 1 จะเขียนเป็น

```
MSComm1.CommPort = 1
```

Settings ใช้ในการกำหนดอัตราบอด (Baud Rate) หรือความเร็วในการส่งข้อมูล มีหน่วยเป็นบิตต่อวินาที, พาริตี, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย โดยมีรูปแบบของการใช้งานดังนี้

```
object.Setting [ = value ]
```

ยกตัวอย่างเช่น เรากำหนดให้มีการเขียนโปรแกรมใช้งานที่ Baud Rate = 9,600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูลเท่ากับ 8 บิต และมีบิตปิดท้าย 1 บิต

```
MSComm1.Setting = "9600, N, 8, 1"
```

PortOpen ใช้สำหรับเปิดและปิดการใช้งานพอร์ตอนุกรม โดยมีรูปแบบของการทำงานดังนี้

```
Object.PortOpen [ = value ]
```

ยกตัวอย่างเช่น เราจะใช้งานพอร์ตอนุกรม ให้กำหนดค่า value เป็น True เขียนโค้ดได้ดังนี้

```
MSComm 1.PortOpen = True
```

แต่ถ้าต้องการปิดพอร์ตอนุกรม ให้กำหนดค่า value เป็น False

```
MSComm 1.PortOpen = False
```

InBuffersize เป็นการกำหนดของ Buffer ในการรับข้อมูลเข้ามา โดยมีรูปแบบการทำงานดังนี้

```
Object.InBuffersize [ = value ]
```


OutBufferSize เป็นการกำหนดของ Buffer ในการส่งข้อมูลเข้ามา โดยมีรูปแบบการทำงานดังนี้

```
Object.OutBufferSize [= value ]
```

Inputlen เป็นการกำหนดค่าของข้อมูลที่อ่านจาก Buffer ภาครับ โดยมีรูปแบบการกำหนดค่าดังนี้

```
Object.Inputlen [= value ]
```

▪ InputMode เป็นการกำหนดค่าชนิดของข้อมูลที่ได้รับมา โดยมีรูปแบบการกำหนดดังนี้

```
Object.InputMode [= value ]
```

โดยที่เราสามารถเลือกชนิดของข้อมูลได้ 2 ประเภทคือ

- comInputMode Text ข้อมูลที่รับเข้ามาเป็นข้อความปกติ เราสามารถตั้งค่าให้อยู่ในโหมดนี้ได้โดยกำหนดค่า Value ให้เป็น "0"
- comInputModeBinary ข้อมูลที่รับเข้ามาเป็นข้อมูลไบนารี เราสามารถตั้งค่าให้อยู่ในโหมดนี้ได้โดยกำหนดค่า Value ให้เป็น "1"
- Input ใช้ในการอ่านค่าข้อมูลจากพอร์ตอนุกรม โดยมีรูปแบบการอ่านค่าดังนี้

```
Object.Input
```

ตัวอย่างเช่น เราอ่านค่าจากบัพเฟอร์ของพอร์ตอนุกรม แล้วนำมาเก็บไว้ในตัวแปรที่ชื่อว่า Data เราจะเขียนโค้ดดังนี้

```
Data = MSComm1.Input
```

- Output ใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบการอ่านค่าดังนี้

```
Object.Output [= value ]
```

- EOFEnable เป็นการบอกว่าสิ้นสุดของไฟล์ End of File [EOF] โดยมีรูปแบบการใช้งานดังนี้

```
Object.EOFEnable [= value ]
```

2.10 SQL Server 2000 [8]

ในปัจจุบัน ระบบงานด้านสารสนเทศต่างๆ ภายในองค์กร เช่น ระบบการลงทะเบียน, เว็บไซต์, ระบบงานบัญชีได้ มีการใช้งานระบบฐานข้อมูลในการจัดเก็บข้อมูลแทนระบบไฟล์แบบเดิม เป็นผลทำให้ข้อมูลที่กระจัดกระจายอยู่ตามไฟล์ข้อมูลต่างๆ ในระบบงานอยู่รวมกันเป็นศูนย์กลาง ซึ่งทำให้ประสิทธิภาพในการจัดเก็บ และทำงานร่วมกับข้อมูลนั้นจะได้รับการออกแบบไว้อย่างดี เพียงใดก็ตาม ถ้าไม่มีซอฟต์แวร์ที่ใช้จัดการระบบฐานข้อมูล (Database Management System: DBMS) ที่มีประสิทธิภาพ ระบบฐานข้อมูลนั้นก็ไม่ได้แตกต่างไปจากการจัดเก็บไฟล์ธรรมดา

ทางบริษัท Microsoft นั้นได้พัฒนาโปรแกรม SQL Server 2000 ซึ่งเป็น DBMS ที่ใช้จัดการระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System หรือ RDBMS) ในท้องตลาดที่มีประสิทธิภาพสูงตัวหนึ่ง มาถึงตรงนี้อาจจะสงสัยกันว่าโปรแกรม Microsoft Access ซึ่งเป็นโปรแกรมจำพวก RDBMS เหมือนกันนั้นมีความแตกต่างกับ SQL Sever 2000 อย่างไร ตรงนี้ต้องขอตอบว่าแตกต่างกันตรงลักษณะการใช้งาน ซึ่งโปรแกรม Microsoft Access นั้นออกแบบมาเพื่อให้ใช้ในงานเดสก์ท็อปขนาดเล็กๆ มีลักษณะการใช้งานส่วนบุคคล เช่น การใช้เก็บข้อมูลส่วนตัว แต่ทว่า SQL Server 2000 ถูกออกแบบมาเพื่อให้ใช้งานใหญ่ๆ ที่ต้องการใช้งานฐานข้อมูลจากผู้ใช้งานหลายๆ คน ดังนั้น SQL Server 2000 จึงต้องมีความสามารถในการจัดฐานข้อมูลที่มีมากกว่า Microsoft Access ดังต่อไปนี้

1. เป็น DBMS แบบ Multiuser Client-Server Database Management System ซึ่งมีความสามารถในการรองรับผู้ใช้ที่เข้ามาใช้งานฐานข้อมูลได้หลายๆ คนในเวลาเดียวกัน ดังนั้น SQL Server 2000 จึงกับงานเล็กๆ ในองค์กรไปจนถึงระดับใหญ่ โดยขึ้นอยู่กับประสิทธิภาพของระบบคอมพิวเตอร์ที่เราใช้ว่า จะสนับสนุนได้ถึงระดับไหน

2. มีความสามารถสำหรับเพิ่มประสิทธิภาพการทำงานให้กับระบบฐานข้อมูลอย่างครบถ้วน เช่น การสร้างวิว, การสร้างอินเด็กซ์ และการสร้างฟังก์ชันเพิ่มเติมนอกเหนือจากฟังก์ชันที่มีไว้ให้แล้ว เป็นต้น

3. ระบบความปลอดภัย SQL Server 2000 ที่ออกแบบเป็นหลายระดับตั้งแต่ระดับการล็อกอินเข้าใช้งานระบบฐานข้อมูล จนถึงระดับการตรวจสอบสิทธิการใช้งานออบเจกต์ต่างๆ ที่อยู่ในระบบฐานข้อมูล ทำให้สามารถจัดการความปลอดภัยของระบบได้อย่างมีประสิทธิภาพ และมีความยืดหยุ่น

4. สนับสนุนการทำงานแบบมัลติโปรเซสเซอร์ ซึ่งทำให้สามารถจัดการข้อมูลได้รวดเร็วขึ้น โดยจะกระจายงานไปให้ซีพียูแต่ละตัว แล้ นำผลลัพธ์ที่ได้มารวมกัน รวมทั้งยังสามารถสร้าง

ระบบการทำงานแบบกระจาย (Distributed Query) ได้อีกด้วย โดยการกระจายฐานข้อมูลไปอยู่บนเซิร์ฟเวอร์หลายๆ ตัว

5. มีความสามารถในการสำรองข้อมูล และยังสามารถคืนสภาพฐานข้อมูลได้อย่างอัตโนมัติ เมื่อระบบทำงานล้มเหลวเนื่องจาก SQL Server 2000 จะมีการเก็บข้อมูลที่เกี่ยวข้องกับการแก้ไขข้อมูลนั้นไว้ ทำให้ข้อมูลในระบบฐานข้อมูลมีเสถียรภาพน่าเชื่อถือมากยิ่งขึ้น

6. มีเครื่องมือต่างๆ ที่ช่วยในการจัดการระบบฐานข้อมูล เช่น Query Analyzer, SQL Enterprise Manager, SQL Profiler ทำให้การจัดการฐานข้อมูลที่ยุ่งยากซับซ้อนกลายเป็นเรื่องที่ยางขึ้น

7. สนับสนุนการทำงาน OLTP (Online Transaction Processing), Data Mining, Data Warehousing และแอปพลิเคชันทางด้านอีคอมเมิร์ซที่ได้รับความนิยมอยู่ในขณะนี้ การจัดการกับฐานข้อมูลเบื้องต้น

ในหัวข้อนี้จะกล่าวถึงการจัดการฐานข้อมูลเบื้องต้น เช่น การสร้างฐานข้อมูล, การลบ, การนำเข้าฐานข้อมูล เป็นต้น สำหรับการจัดการฐานข้อมูลที่ใช้ทั่วไปใน SQL Server จะมี 2 วิธีคือ

1. ใช้เครื่องมือ Enterprise Manager ซึ่งเป็นเครื่องมือที่ใช้ในการจัดการระบบฐานข้อมูลเป็นแบบกราฟิกยูสเซอร์อินเตอร์เฟซ ทำให้เราสามารถทำงานต่างๆ ในระบบฐานข้อมูล เช่น การสร้างฐานข้อมูล, การสร้างตารางโดยไม่ต้องเรียนรู้ และจดจำคำสั่ง Transact-SQL ต่างๆ ที่มีอยู่มากมาย แต่ Enterprise Manager ก็ยังต้องแปลงการทำงานของเรานั้นเป็นคำสั่ง Transact-SQL อยู่ดี ดังนั้น Enterprise Manager อาจจะไม่สามารถทำบางสิ่งที่คำสั่ง Transact-SQL สามารถทำได้ ในหัวข้อนี้จะทำการสร้าง และจัดการฐานข้อมูล โดยใช้ Enterprise Manager อย่างเดียว สำหรับวิธีใช้คำสั่ง Transact-SQL จะแสดงรายละเอียดคร่าวๆ เท่านั้น

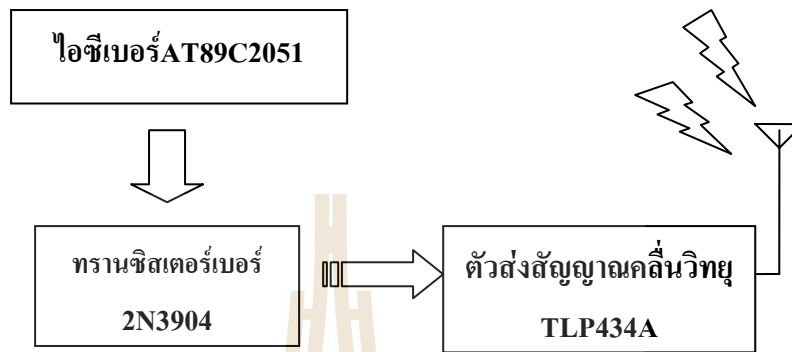
2.. ใช้คำสั่ง Transact-SQL ซึ่งเป็นชุดคำสั่งที่ทางไมโครซอฟท์สร้างขึ้นเพื่อใช้ในการทำงานร่วมกับฐานข้อมูลต่างๆ ในระบบฐานข้อมูลเชิงสัมพันธ์ เช่น การสร้างฐานข้อมูล, การเรียกดูข้อมูลในฐานข้อมูล, การลบข้อมูลในฐานข้อมูล เป็นต้น โดย SQL Server นั้นได้เตรียมเครื่องมือ Query Analyzer ซึ่งเป็นเครื่องมือรวมกับคำสั่ง Transact-SQL ไว้ให้แล้ว

บทที่ 3

การสร้างและการออกแบบ

3.1 การออกแบบวงจรเครื่องส่งสัญญาณคลื่นวิทยุ

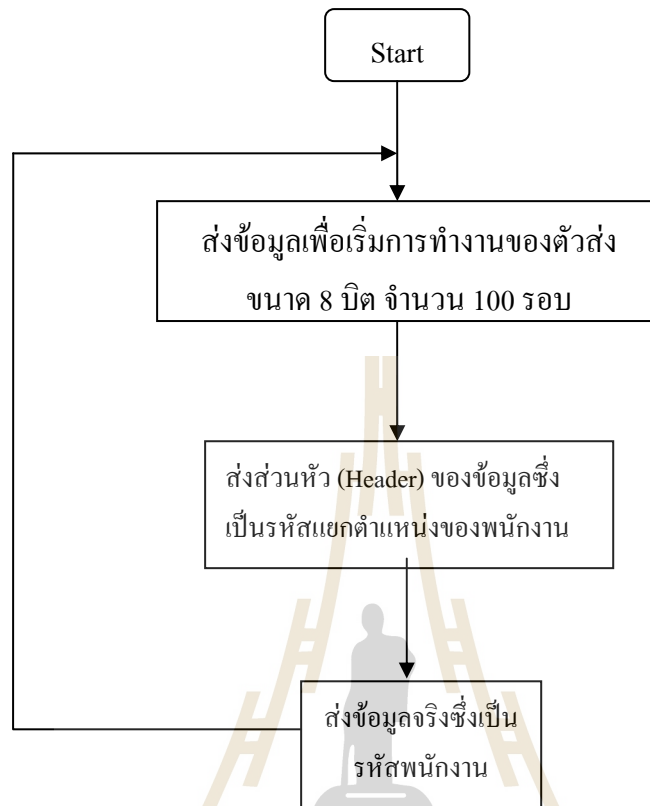
3.1.1 การออกแบบทางฮาร์ดแวร์



รูปที่ 3.1 ผังโคะแกรมของวงจรเครื่องส่งสัญญาณคลื่นวิทยุ

การออกแบบวงจรเครื่องส่งสัญญาณคลื่นวิทยุ นั้นเริ่มจากการเลือกใช้ไอซีที่สามารถบันทึกข้อมูลได้ และส่งข้อมูลดังกล่าวออกมาใช้งานได้ ซึ่งเราเลือกใช้ไอซีเบอร์ AT89C2051 แล้วนำไอซีเบอร์ดังกล่าวมาบันทึกข้อมูลโดยใช้บอร์ด ET-AFP V1.0(R1) ข้อมูลที่ได้บันทึกลงไปจะถูกส่งผ่านออกทางขาอินพุตสำหรับส่งข้อมูล(TDx)ของไอซีเบอร์ AT89C2051 และจะใช้ทรานซิสเตอร์เบอร์ 2N3904 เป็นตัวควบคุมจังหวะในการส่งข้อมูลของวงจร แล้วข้อมูลดังกล่าวจะถูกส่งไปยังตัวส่งสัญญาณคลื่นวิทยุ LAIPAC เบอร์ TLP434A ที่จะเป็นตัวรับข้อมูลเข้ามาเพื่อทำการเข้ารหัสสัญญาณข้อมูล (Modulation) แบบ Amplitude – Shift Keying (ASK) ซึ่งเป็นการมอดูเลตสัญญาณดิจิทัล เป็นสัญญาณอนาล็อก (Digital To Analog : D/A) โดยทำงานในความถี่ 433.92 เมกะเฮิร์ตซ์ (MHz) แล้วสัญญาณข้อมูลดังกล่าวจะถูกส่งไปยังเครื่องรับสัญญาณคลื่นวิทยุผ่านสายอากาศ (Antenna) ดังแสดงในรูปที่ 3.1

3.1.2 การออกแบบทางซอฟต์แวร์



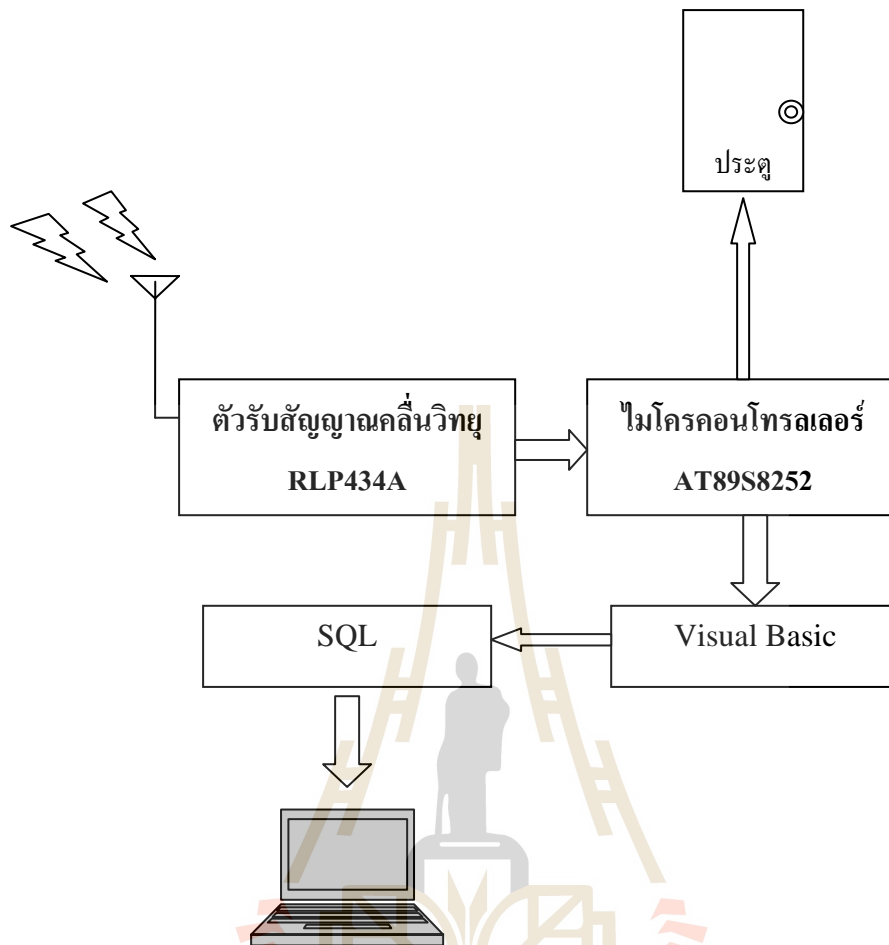
รูปที่ 3.2 ลำดับการทำงานของโปรแกรมการส่งข้อมูล

การทำงานของโปรแกรมที่ได้ออกแบบไว้มีการทำงานดังนี้

1. ส่งข้อมูลเพื่อเริ่มการทำงานของตัวส่งและตัวรับซึ่งเป็นข้อมูลขนาด 8 บิต จำนวน 100 รอบ การส่งข้อมูลลักษณะนี้เป็นการส่งเพื่อให้ตัวส่งและตัวรับของวงจรเครื่องรับ-ส่งเริ่มทำงาน
2. ส่งส่วนหัว (Header) ของข้อมูลซึ่งเป็นรหัสเพื่อใช้แยกข้อมูลจริงออกจากข้อมูลเริ่มการทำงาน of ตัวส่งและตัวรับ อีกทั้งยังเป็นรหัสที่ใช้แยกตำแหน่งของพนักงานเพื่อให้ง่ายต่อการตรวจสอบข้อมูลและช่วยให้การตรวจสอบรวดเร็วและมีประสิทธิภาพยิ่งขึ้น
3. ส่งข้อมูลจริงซึ่งเป็นรหัสของข้อมูลของพนักงานแต่ละคน และรหัสข้อมูลจริงนี้จะถูกแปลงเป็นข้อมูลแล้วทำการแสดงยังมอเนเตอร์ต่อไป
4. วนรอบการทำงานใหม่จนกระทั่งมีการปิดสวิตช์หยุดการทำงานของวงจรเครื่องส่ง การทำงานจึงจะหยุดลง

3.2. การออกแบบวงจรเครื่องรับสัญญาณคลื่นวิทยุ

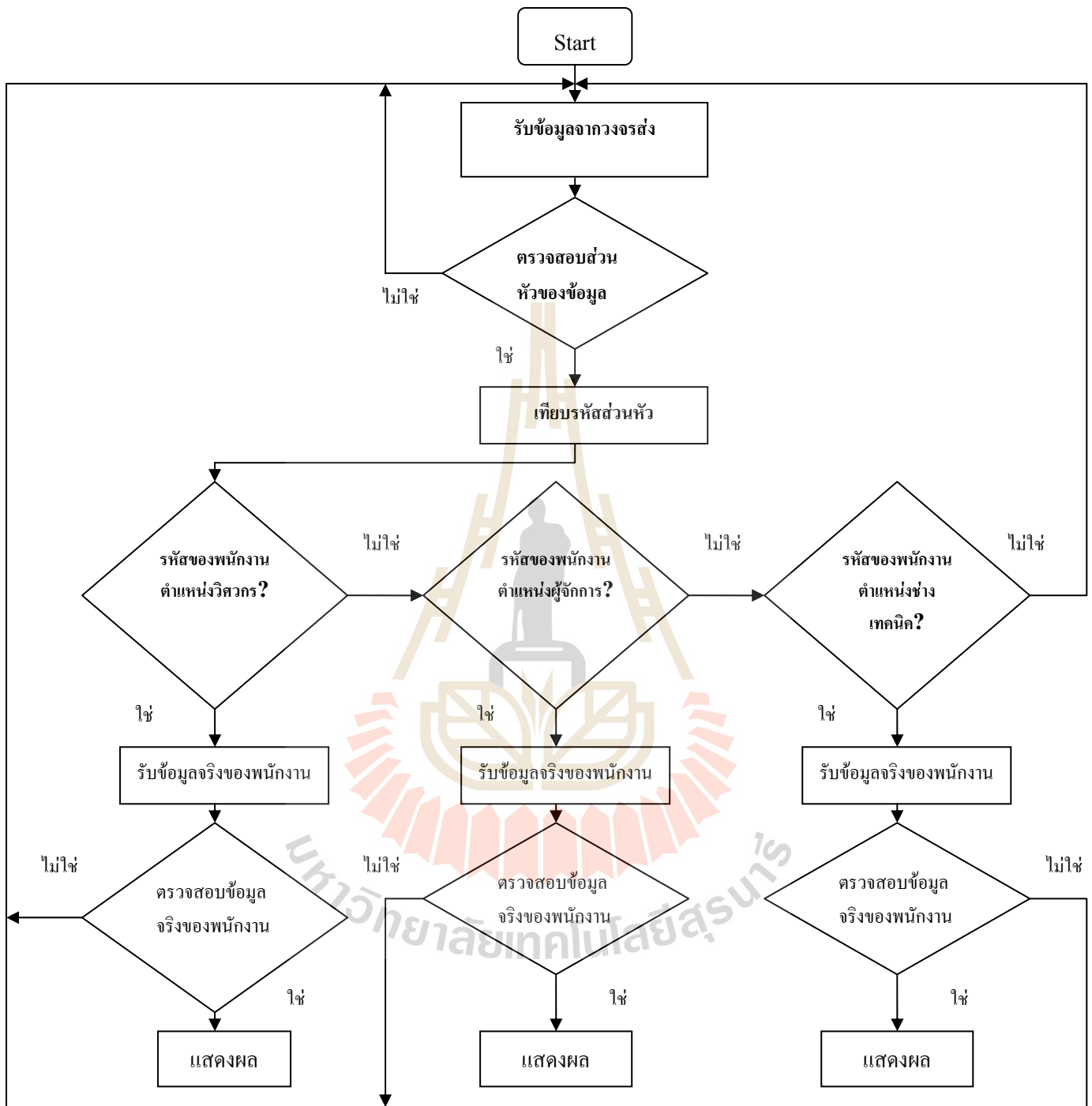
3.2.1 การออกแบบทางฮาร์ดแวร์



รูปที่ 3.3 ผังโคดะแกรมของวงจรเครื่องส่งสัญญาณคลื่นวิทยุ

การออกแบบวงจรเครื่องรับสัญญาณคลื่นวิทยุ นั้นจะใช้ตัวรับสัญญาณคลื่นวิทยุ LAIPAC เบอร์ RPL434A รับสัญญาณข้อมูลจากเครื่องส่งสัญญาณคลื่นวิทยุผ่านสายอากาศ (Antenna) ของตัวรับสัญญาณคลื่นวิทยุ ที่ทำงานในย่านความถี่ 433.92 เมกะเฮิรตซ์ (MHz) จากนั้นตัวรับสัญญาณคลื่นวิทยุ จะทำการถอดรหัสสัญญาณข้อมูล (Demodulation) สัญญาณข้อมูลที่ถูกถอดรหัสแล้ว จะถูกส่งไปที่ขาอินพุตสำหรับรับข้อมูล (RXD) ของไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 เมื่อได้รับข้อมูลไมโครคอนโทรลเลอร์จะสั่งให้ประตู่เปิด-ปิด ซึ่งจะแสดงสถานะการเปิดประตู่ด้วยหลอดLEDกระพริบ และจะแสดงสถานะการปิดประตู่ด้วยหลอดLEDหยุดกระพริบ ส่วนข้อมูลที่ได้รับมาจะถูกส่งออกมาผ่านพอร์ตอนุกรมเข้าสู่เครื่องคอมพิวเตอร์ เพื่อที่จะแสดงข้อมูลของพนักงานทางจอมอนิเตอร์ต่อไป

3.2.2 การออกแบบทางซอฟต์แวร์



รูปที่ 3.4 ลำดับการทำงานของโปรแกรมการรับข้อมูล

การทำงานของโปรแกรมที่ได้ออกแบบไว้มีการทำงานดังนี้

1. รับข้อมูลจากวงจรเครื่องส่ง ข้อมูลที่ได้รับจะเป็นข้อมูลขนาด 8 บิตทั้งหมด สามารถแบ่งลักษณะของข้อมูลที่ได้รับเป็น 3 ลักษณะคือ ข้อมูลเพื่อเริ่มการทำงานของตัวส่ง และตัวรับ ส่วนหัว (Header) ของข้อมูลและข้อมูลจริง
2. ตรวจสอบส่วนหัว (Header) ของข้อมูลว่าเป็นรหัสที่กำหนดไว้หรือไม่ ถ้าใช่จะทำการเทียบส่วนหัวของข้อมูลว่าเป็นรหัสตำแหน่งของพนักงานในตำแหน่งใด แต่ถ้าไม่ใช่รหัสที่เป็นส่วนหัวของข้อมูลก็แสดงว่าข้อมูลที่ได้รับมานั้นยังเป็นส่วนข้อมูลเพื่อเริ่มการทำงานของตัวส่งและตัวรับ ดังนั้นวงจรเครื่องรับก็จะทำการรับข้อมูลใหม่อีกครั้ง และจะตรวจสอบจนกว่าจะได้ส่วนหัวของข้อมูล
3. จากข้อ 2. เมื่อเทียบส่วนหัวของข้อมูลจนทราบว่าเป็นรหัสตำแหน่งของพนักงานในตำแหน่งใดแล้วก็จะรอรับข้อมูลจริงที่ถูกส่งเข้ามา แต่ถ้าเทียบแล้วไม่ใช่รหัสที่เป็นรหัสตำแหน่งของพนักงานในตำแหน่งใด ก็แสดงว่าข้อมูลที่ได้รับมานั้นยังเป็นส่วนข้อมูลเพื่อเริ่มการทำงานของตัวส่งและตัวรับ ดังนั้นวงจรเครื่องรับก็จะทำการรับข้อมูลใหม่อีกครั้งและจะตรวจสอบจนกว่าจะได้ส่วนหัวของข้อมูลและเป็นรหัสตำแหน่งของพนักงานในบริษัท
4. รับข้อมูลจริงซึ่งเป็นรหัสของข้อมูลของพนักงานแต่ละคน และตรวจสอบว่ารหัสข้อมูลจริงที่ได้รับมานี้เป็นรหัสข้อมูลจริงหรือไม่ ถ้าใช่รหัสข้อมูลนี้จะถูกส่งไปที่คอมพิวเตอร์เพื่อแปลงเป็นข้อมูลแล้วทำการแสดงยังมอเนิเตอร์ต่อไป แต่ถ้าไม่ใช่รหัสข้อมูลจริง วงจรเครื่องรับก็จะทำการรับข้อมูลใหม่อีกครั้งและจะตรวจสอบจนกระทั่งทราบว่ารหัสข้อมูลจริงที่ได้เขียนไว้ในวงจรเครื่องส่งจึงจะส่งออกไปแสดงผล
5. วงจรเครื่องรับจะทำการรับข้อมูล ตรวจสอบข้อมูลและส่งไปแสดงผล อยู่ตลอดเวลา จนกระทั่งมีการปิดสวิทช์หยุดการทำงานของวงจรเครื่องรับ การทำงานจึงจะหยุดลง

จากการออกแบบโปรแกรมพบว่ามี การตรวจสอบข้อมูลหลายขั้นตอนก็เพราะว่า การส่งข้อมูลผ่านอากาศจากวงจรเครื่องส่งมายังวงจรเครื่องรับนั้น ในบางครั้งวงจรเครื่องรับจะรับข้อมูลบางอย่างที่อยู่ในอากาศซึ่งมีความถี่เดียวกันกับตัวรับในวงจรเครื่องรับเข้ามาทำให้การแสดงผลที่ได้ไม่เป็นไปตามที่ต้องการ ซึ่งการตรวจสอบหลายขั้นตอนสามารถช่วยกำจัดปัญหาเหล่านี้ได้และยังง่ายต่อการตรวจสอบข้อมูลและช่วยให้การตรวจสอบรวดเร็วและมีประสิทธิภาพยิ่งขึ้น อีกทั้งยังช่วยลดการทำงานของวงจรเครื่องรับโดยที่ไม่ต้องทำการตรวจสอบข้อมูลที่ละรายการอีกด้วย

3.3 การเขียนโปรแกรมเพื่อรับ-ส่งข้อมูล ระหว่างตัวส่งและตัวรับสัญญาณวิทยุ

3.3.1 การเขียนโปรแกรมในการส่งข้อมูล

1.เขียนคำสั่งสำหรับเปิดพอร์ตการสื่อสารแบบอนุกรมของวงจรเครื่องส่ง

```
org    0000h
mov    scon,#50h           ; สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov    tmod,#20h
mov    th1,#0fbh
setb   tr1
clr    ti
clr    ri
```

2.เขียนคำสั่งเพื่อส่งข้อมูลเพื่อเริ่มการทำงานของตัวส่ง เนื่องจากวงจรของตัวส่งนี้จะต้องทำการส่งข้อมูลสำหรับติดต่อระหว่างวงจรเครื่องส่งและวงจรเครื่องรับ เพื่อเริ่มการทำงานก่อนส่งข้อมูลจริง จากโปรแกรมนี้ จะทำการส่งข้อมูลชุดนี้จำนวน 100 รอบ

```
mov    r7,#0100           ; นำค่า 100 เก็บไว้ที่ R7
aa:    mov    sbuf,#55h     ; นำค่า 55 H เก็บไว้ที่ SBUF
jnb    ti,$               ; รอส่งข้อมูลที่ละบิตจนกระทั่งข้อมูลครบ 8 บิต
clr    ti                  ; เคลียร์ ti เพื่อส่งค่าในรอบต่อไป
djnz   r7,aa              ; วนรอบส่งจนกระทั่งค่า R7 ลดจาก 100 ให้เป็น 0
```

3.เขียนคำสั่งเพื่อส่งส่วนหัวของข้อมูล (Header) ซึ่งส่วนหัวของข้อมูลนี้เป็นข้อมูลขนาด 8 บิต และจะแตกต่างกับข้อมูลที่ส่งเพื่อเริ่มการทำงานของวงจรเครื่องส่ง เพื่อตรวจสอบที่วงจรเครื่องรับว่าวงจรเครื่องส่งได้ส่งข้อมูลจริงมาหรือไม่ และส่วนหัวของข้อมูลนี้ยังเป็นรหัสที่ใช้แทนตำแหน่งของพนักงานในบริษัทเพื่อใช้จัดกลุ่มพนักงาน จึงทำให้การเรียกข้อมูลของพนักงานทำได้ง่ายและรวดเร็ว อีกทั้งยังช่วยลดการทำงานของวงจรให้ทำงานน้อยลง

```
mov    sbuf,#06h         ; ส่งส่วนหัวของข้อมูลซึ่งเป็นรหัสที่ใช้แทนตำแหน่งของพนักงาน
jnb    ti,$
clr    ti
```

4.เริ่มทำการส่งข้อมูลจริง ซึ่งข้อมูลนี้จะเป็นรหัสข้อมูลของพนักงานและถูกนำไปแปลงเป็นข้อมูลเพื่อแสดงผ่านมอนิเตอร์โดยใช้โปรแกรมวิซวลเบสิก ต่อไป

```
mov  sbuf,#61h           ;ส่งส่วนของข้อมูลจริง
jnb  ti,$
clr  ti
```

5.เริ่มทำการส่งข้อมูลใหม่โดยจะส่งข้อมูลเพื่อเริ่มการทำงานของวงจรเครื่องส่งและจะทำงานวนรอบไปเรื่อยๆจนกว่าจะปิดแหล่งจ่ายไฟแก่วงจร วงจรเครื่องส่งจึงจะหยุดทำงาน

ตัวอย่างโปรแกรมในการส่งข้อมูล ของพนักงาน ไปยังวงจรรับข้อมูล

```
org  0000h
mov  scon,#50h           ; สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov  tmod,#20h
mov  th1,#0fbh
setb tr1
clr  ti
clr  ri
clr  p3.7                ;เคลียร์ขา p3.7 เพื่อป้องกันการรบกวนจากสถานะ High ของ
ข้อมูล
loop: mov  r7,#0100       ; นำค่า 100 เก็บไว้ที่ R7
aa:  mov  sbuf,#55h       ; นำค่า55 H เก็บไว้ที่ SBUF
     jnb  ti,$           ; รอส่งข้อมูลที่ละบิตจนกระทั่งข้อมูลครบ 8 บิต
     clr  ti              ;เคลียร์ ti เพื่อส่งค่าในรอบต่อไป
     djnz r7,aa          ; วนรอบส่งจนกระทั่งค่า R7 ลดจาก 100 ให้เป็น0
     mov  r6,#04h        ; นำค่า 04h เก็บไว้ที่ R6
xx1: mov  sbuf,#06h       ;ส่งส่วนหัวของข้อมูล
     jnb  ti,$
     clr  ti
     mov  sbuf,#61h      ;ส่งส่วนของข้อมูลจริงของพนักงานคนที่ 1
     jnb  ti,$
```

```

clr    ti
djnz  r6,xx1          ;วนรอบส่งจนกระทั่งค่า R6 ลดจาก 4 ให้เป็น0
setb  p3.7           ;เซตขา p3.7 เพื่อเริ่มทำงานรอบใหม่
call  delay          ;เรียกส่วนประวิงเวลา
jmp   loop           ;วนรอบเพื่อเริ่มการทำงานใหม่

delay:  setb  p3.7      ;ส่วนประวิงเวลา
        mov  r3,#0ch
delay 1: mov  r4,#0ffh
delay 2: mov  r5,#0ffh
        djnz r5,$
        djnz r4, delay 2
        djnz r3, delay 1
        clr  p3.7
        ret   ;ย้อนกลับไปหลังคำสั่งที่เรียกใช้
        end   ;จบการทำงาน

```

3.3.2 การเขียนโปรแกรมในการรับข้อมูล

1.เขียนคำสั่งสำหรับเปิดพอร์ตการสื่อสารแบบอนุกรมของวงจรเครื่องรับ

```

org    0000h
mov    scon,#50h      ; สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov    tmod,#20h
mov    th1,#0fbh
setb   tr1
clr    ti
clr    ri

```

2.เขียนคำสั่งเพื่อรับข้อมูลทั้งหมดจากวงจรส่ง ซึ่งข้อมูลส่วนนี้จะเป็นข้อมูลที่ใช้เริ่มการทำงานของวงจร และส่วนหัวของข้อมูล

jnb	ri,\$;รับข้อมูลที่ละบิตจนครบ 8 บิตแล้วเก็บไว้ใน sbuf
mov	a,sbuf	;เก็บข้อมูลจาก sbuf ไว้ในรีจิสเตอร์ (Register) a
clr	ri	;เคลียร์ ri เพื่อรับค่าในรอบต่อไป

3.เขียนคำสั่งเพื่อตรวจสอบว่าข้อมูลที่รับเข้ามานั้น เป็นส่วนหัวของข้อมูลหรือไม่ ถ้าใช่ ก็จะรอรับข้อมูลจริงต่อไป แต่ถ้าไม่ใช่ส่วนหัวของข้อมูล จะทำการเริ่มรับข้อมูลใหม่ ซึ่งคำสั่งในส่วนนี้จะเป็นการแยกข้อมูลจริงออกจากข้อมูลสำหรับเริ่มทำงานนั่นเอง อีกทั้งส่วนหัวของข้อมูลนี้ยังเป็นรหัสที่ใช้แทนตำแหน่งของพนักงานในบริษัทเพื่อใช้จัดกลุ่มพนักงาน ทำให้ส่วนหัวของข้อมูลนี้มีอยู่หลายรหัสตามตำแหน่งของพนักงาน ดังนั้นจะมีการเทียบส่วนหัวทุกรหัส กล่าวคือถ้าไม่ใช่รหัสของตำแหน่งพนักงานแรกก็จะทำการเทียบไปเรื่อยๆจนกว่าจะตรงกับรหัสของตำแหน่งพนักงานที่ถูกเขียนไว้ เมื่อเทียบตรงแล้วจึงรอรับข้อมูลจริงของพนักงานต่อไป แต่ถ้าเทียบแล้วไม่ตรงกับรหัสของพนักงานตำแหน่งใดเลยก็จะทำการเริ่มรับข้อมูลใหม่

cjne	a,#06h,loop5	;ตรวจสอบส่วนหัวของข้อมูลว่าใช่หรือไม่ ถ้าใช่จะรอรับข้อมูลจริง ถ้าไม่ใช่จะทำการเทียบรหัสไปเรื่อยๆถ้าตรงจะรอรับข้อมูลจริง แต่ถ้าไม่ตรงเริ่มรับข้อมูลใหม่
jnb	ri,\$;รอรับข้อมูลจริง
mov	a,sbuf	
clr	ri	

4.เขียนคำสั่งเพื่อตรวจสอบว่าข้อมูลที่รับเข้ามานั้นเป็นข้อมูลจริงที่ส่งมาจากวงจรส่งจริงหรือไม่ ถ้าใช่ก็จะส่งข้อมูลไปยังส่วนแสดงผลต่อไป แต่ถ้าไม่ใช่ข้อมูลจริงจะทำการเริ่มรับข้อมูลใหม่ ซึ่งส่วนตรวจสอบข้อมูลจริงที่เขียนขึ้นมาเพื่อป้องกันการผิดพลาดจากการส่งข้อมูล เนื่องจากคุณสมบัติของวงจรรับ-ส่ง เป็นแบบ ASK ซึ่งใช้ความถี่ในการส่งเดียวกัน เมื่อมีวงจรส่งหลายตัวในพื้นที่เดียวกันแล้ว จะทำให้ข้อมูลที่รับได้เกิดข้อผิดพลาดขึ้นมาได้

cjne	a,#61h,loop	;ตรวจสอบข้อมูลจริงว่าใช่หรือไม่ ถ้าใช่จะส่งข้อมูลจริงไปยังส่วนแสดงผล ถ้าไม่ใช่จะเริ่มรับข้อมูลใหม่
mov	sbuf,a	;ส่งข้อมูลไปยังส่วนแสดงผล
jnb	ti,\$	
clr	ti	

5.เริ่มทำการรับข้อมูลใหม่โดยจะรับข้อมูลแล้วทำการตรวจสอบส่วนหัวของข้อมูล ตรวจสอบข้อมูลจริง จากนั้นจึงส่งข้อมูลไปยังส่วนแสดงผล และจะทำงานวนรอบไปเรื่อยๆจนกว่าจะไม่มีสัญญาณข้อมูลจากวงจรส่ง

ตัวอย่างโปรแกรมในการรับและตรวจสอบข้อมูลของพนักงานจำนวน 3 คนในตำแหน่งวิศวกร

```

org    0000h
mov    scon,#50h           ; สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov    tmod,#20h
mov    th1,#0fbh
setb   tr1
clr    ti
clr    ri
setb   p3.6               ; เปิดการแสดงสถานะของหลอดไฟ
setb   p3.7               ; เปิดการแสดงสถานะของหลอดไฟ
mov    p0,#11111111b      ; หยุดการทำงานของประตู
loop:  mov    p0,#11111111b ; หยุดการทำงานของประตู
      jnb    ri,$          ; รับข้อมูลที่ละบิตจนครบ 8 บิตแล้วเก็บไว้ใน sbuf
      mov    a,sbuf        ; เก็บข้อมูลจาก sbuf ไว้ในรีจิสเตอร์ (Register) a
      clr    ri           ; เคลียร์ ri เพื่อรับค่าในรอบต่อไป
loop1: cjne  a,#06h,loop5   ; ตรวจสอบส่วนหัวของข้อมูลว่าใช่หรือไม่ ถ้าใช่จะรอรับข้อมูล
จริง                                     ถ้าไม่ใช่จะ ทำการเทียบรหัสไปเรื่อยๆ ถ้าตรงจะรอรับข้อมูลจริง
                                       แต่ถ้าไม่ตรงเริ่มรับข้อมูลใหม่(06hเป็นรหัสในตำแหน่งวิศวกร)
      jnb    ri,$          ; ส่งข้อมูลไปยังส่วนแสดงผล
      mov    a,sbuf
      clr    ri
      jmp   loop          ; วนรอบเพื่อเริ่มการทำงานใหม่

loop2: cjne  a,#61h,loop3   ; ตรวจสอบข้อมูลจริงว่าเป็นข้อมูลของพนักงานคนที่ 1 ใช่หรือไม่
                                       ถ้าใช่จะส่งข้อมูลนั้นไปส่วนแสดงผล แต่ถ้าไม่ใช่จะตรวจข้อมูล
                                       พนักงานคนต่อไป
      mov    sbuf,a        ; ส่งข้อมูลไปยังส่วนแสดงผล
      jnb    ti,$
      clr    ti

```

```

mov    p0,#1111110b      ;เปิดประตู
call   delay500ms        ;ประวิงเวลาประมาณ 0.5 วินาที
mov    p0,#1111101b      ;เปิดประตู
jmp    loop              ;วนรอบเพื่อเริ่มการทำงานใหม่
loop3: cjne  a,#62h,loop4 ;ตรวจสอบข้อมูลจริงว่าเป็นข้อมูลของพนักงานคนที่ 2 ใช่
        หรือไม่          ;ถ้าใช่จะส่งข้อมูลนั้นไปส่วนแสดงผล แต่ถ้าไม่ใช่จะ
                        ;ตรวจสอบข้อมูลพนักงานคนต่อไป
mov    sbuf,a            ;ส่งข้อมูลไปยังส่วนแสดงผล
jnb    ti,$
clr    ti
mov    p0,#1111110b      ;เปิดประตู
call   delay500ms        ;ประวิงเวลาประมาณ 0.5 วินาที
mov    p0,#1111101b      ;เปิดประตู
jmp    loop              ;วนรอบเพื่อเริ่มการทำงานใหม่
loop4: cjne  a,#63h,loop ;ตรวจสอบข้อมูลจริงว่าเป็นข้อมูลของพนักงานคนที่ 3 ใช่
        หรือไม่          ;ถ้าใช่จะส่งข้อมูลนั้นไปส่วนแสดงผล แต่ถ้า
        ไม่ใช่จะเริ่มรับข้อมูลใหม่
mov    sbuf,a            ;ส่งข้อมูลไปยังส่วนแสดงผล
jnb    ti,$
clr    ti
mov    p0,#1111110b      ;เปิดประตู
call   delay500ms        ;ประวิงเวลาประมาณ 0.5 วินาที
mov    p0,#1111101b      ;เปิดประตู
jmp    loop              ;วนรอบเพื่อเริ่มการทำงานใหม่
delay500ms: call   delay100ms ;ส่วนประวิงเวลาประมาณ 0.5 วินาที
              call   delay100ms
              call   delay100ms
              call   delay100ms
              call   delay100ms
              ret
delay100ms:  mov    r6,#100
dly100ms:   call   delay500u

```

```

        djnz    r6, delay100ms
        ret
delay500u:  mov    r5,#230
            djnz    r5,$
            end
            ;จบการทำงาน

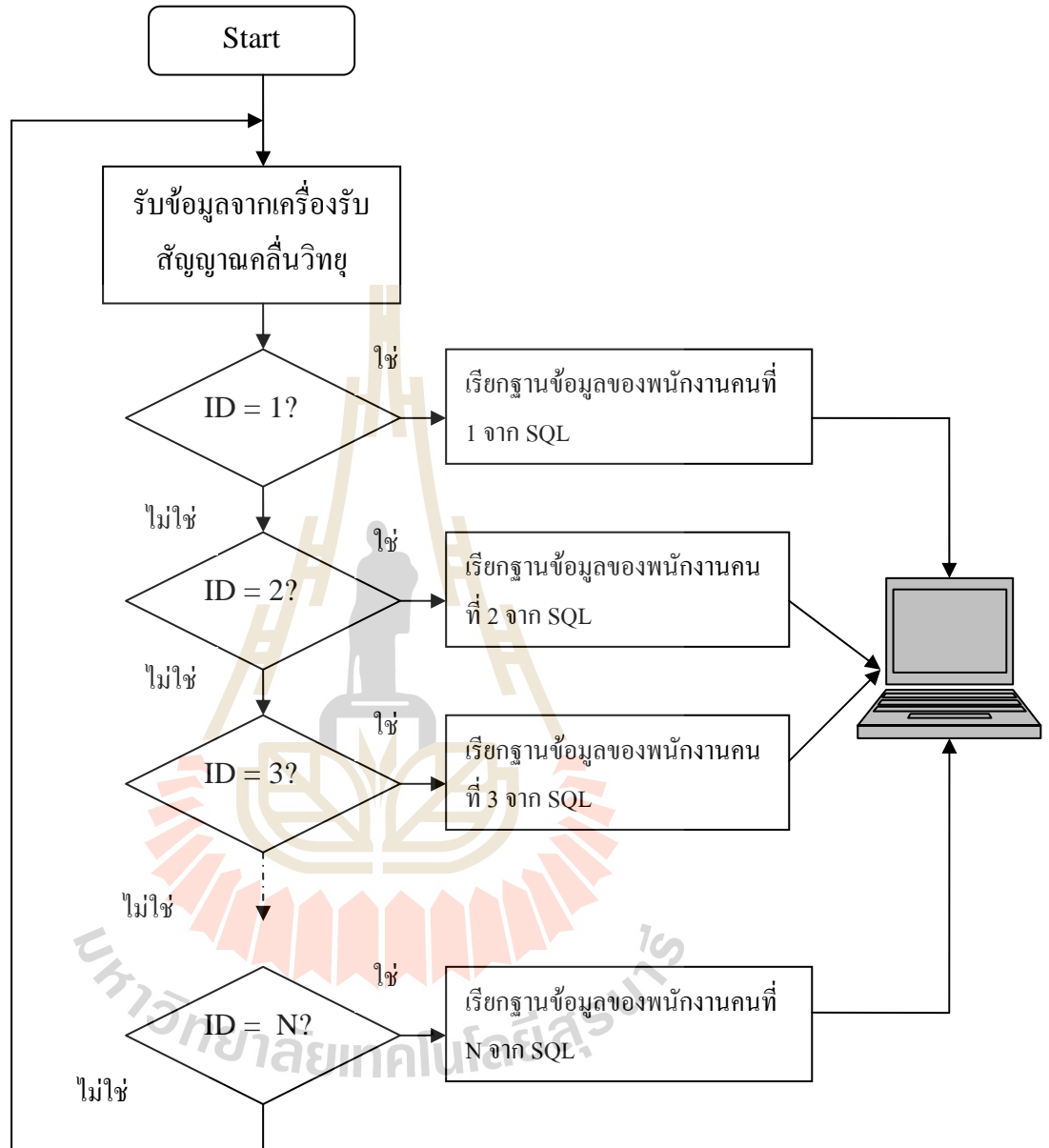
```

จากคำสั่งต่างๆของโปรแกรมในการส่งและรับข้อมูลนั้น พบว่ามีการเพิ่มส่วนของการประวิงเวลา เพื่อให้สามารถสังเกตการแสดงผลการทดลองได้ เช่น การเปิด-ปิด หลอดไฟ การเปิด-ปิด ประตู เป็นต้น แต่ช่วงการประวิงเวลานี้จะมีผลต่อการรับและส่งข้อมูลคือ จะทำให้การรับและส่งข้อมูลได้ช้าลง ส่วนที่แสดงผลของข้อมูลที่จอมอนิเตอร์ก็จะช้าลงหรืออาจจะไม่สามารถแสดงผลของข้อมูลได้บ้างค่า ดังนั้นการเพิ่มเวลาในการประวิงให้มากขึ้น แม้จะสามารถแสดงผลการทดลองได้ แต่ก็มีผลเสียคือการแสดงผลของข้อมูลได้ช้าลง



3.4 การออกแบบเพื่อเขียนโปรแกรม Visual basic

1. ลำดับการทำงานของโปรแกรม Visual basic ดังแสดงในแผนภาพที่ 3.5



แผนภาพที่ 3.5 ลำดับการทำงานของ โปรแกรม Visual Basic

2. การเขียนโปรแกรม Visual Basic

1. สร้างฟอร์มเพื่อแสดงบนจอภาพ ดังรูป

The screenshot shows a Visual Basic form titled "Form1" with a blue background and a grid pattern. The form contains several input fields and controls:

- Fields for "รหัสพนักงาน" (Employee ID), "ตำแหน่ง" (Position), "ชื่อ" (Name), "นามสกุล" (Surname), "เพศ" (Gender), "วันเดือนปีเกิด" (Date of Birth), "สัญชาติ" (Nationality), "สถานภาพ" (Status), "ศาสนา" (Religion), "ภูมิภาค" (Region), "โทรศัพท์" (Phone Number), "วุฒิการศึกษา" (Education Level), "วันเข้าทำงาน" (Start Date), "เงินเดือน" (Salary), "เวลาเข้า" (Time In), and "เวลาออก" (Time Out).
- Navigation buttons: Home, Previous, Next, End.
- Control buttons: "ค้นหา" (Search), "K", "<", ">", ">|".
- Text boxes labeled "Text17" through "Text30".
- System tray icons: Clock, Volume, Network, and a custom icon.

รูป (ก) ฟอร์ม 1

The screenshot shows a Visual Basic form titled "ประวัติการเข้า-ออกงาน" (Attendance History) with a blue background and a grid pattern. The form contains:

- Fields for "รหัสพนักงาน" (Employee ID) and "วันที่" (Date).
- Buttons: "ค้นหา" (Search), "ปิด" (Close).
- A table with columns: "รหัสพนักงาน" (Employee ID), "วันที่เข้า" (Date In), "เวลาที่เข้า" (Time In), "วันที่ออก" (Date Out), and "เวลาที่ออก" (Time Out).
- Buttons: "ดูเข้าทั้งหมด" (View All In), "ดูออกทั้งหมด" (View All Out), and "ดู" (View).
- Navigation buttons: Home, Previous, Next, End.

รูป (ข) ฟอร์ม 2

Form3

ข้อมูลพนักงาน

รหัสพนักงาน:

ตำแหน่ง:

ชื่อ:

นามสกุล:

เพศ:

วันเดือนปีเกิด:

สัญชาติ:

สถานภาพ:

ศาสนา:

วุฒิตำแหน่ง:

โทรศัพท์:

วุฒิการศึกษา:

วันเข้าทำงาน:

เงินเดือน:

ที่อยู่จริง:

ยืนยัน ตกลง ยกเลิก

ยืนยัน ตกลง ยกเลิก

ยืนยัน ตกลง ยกเลิก

Navigation: [Back] [Forward] [Home] [End] [K] [<] [>] [>|]

รูป (ค) ฟอรั่ม 3

MDIForm1

ข้อมูลพนักงาน การทำงาน คณิตศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูป (ง) MDI ฟอรั่ม

รูปที่ 3.6 การสร้างฟอรั่มเพื่อแสดงบนจอภาพ

2. คำสั่งที่ใช้ในแต่ละฟอร์ม

2.1 ประกาศตัวแปรและกำหนดประเภทของตัวแปรทั้งหมด ภายในฟอร์ม 1 ซึ่งเป็นฟอร์มที่แสดงข้อมูลของพนักงาน เมื่อมีพนักงานเข้า-ออกบริษัท

Form 1

Option Explicit

Dim Data As String ;ตัวแปร Data เป็นตัวแปรประเภท String

Dim recdata As String ;ตัวแปร recdata เป็นตัวแปรประเภท String

Dim x, y, z, x1, x2 As Long ;ตัวแปร x, y, z, x1, x2 เป็นตัวแปรประเภท Long

2.2 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดแรกของข้อมูล

Private Sub Command1_Click()

Adodc2.Recordset.MoveFirst ;เลื่อนไปเรคคอร์ดแรกของข้อมูล

Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ;แสดงรูปจากฐานข้อมูล

End Sub

2.3 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดก่อนหน้าของข้อมูล

Private Sub Command2_Click()

If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF

Adodc2.Refresh ; Refresh ข้อมูล

Else

Adodc2.Recordset.MovePrevious ; เลื่อนไปเรคคอร์ดก่อนหน้า

If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF

Else

Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล

End If

End If

End Sub

2.4 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดถัดไปของข้อมูล

Private Sub Command3_Click()

If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF

```

Adodc2.Recordset.MoveLast ;เลื่อนไปเรคคอร์ดสุดท้าย
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
Else
Adodc2.Recordset.MoveNext ;เลื่อนไปเรคคอร์ดถัดไป
If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF
Picture1.Refresh ; Refresh รูป
Else
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
End If
End If
End Sub

```

2.5 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดสุดท้ายของข้อมูล

```

Private Sub Command4_Click()
Adodc2.Recordset.MoveLast ;เลื่อนไปเรคคอร์ดสุดท้าย
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ;แสดงรูปจากฐานข้อมูล
End Sub

```

2.6 เขียนคำสั่งเพื่อค้นหาข้อมูลของพนักงาน โดยการพิมพ์รหัสพนักงานถ้ารหัสที่พิมพ์ไม่ตรงกับรหัสของพนักงานปรากฏ Message Box แจ้งให้ทราบว่าป็นรหัสนี้ไม่ตรงกับรหัสของพนักงาน

```

Private Sub Command5_Click()
Dim id As String ; ตัวแปร id เป็นตัวแปรประเภท String
id = InputBox("พิมพ์รหัสพนักงานที่ต้องการค้นหา", "รหัสพนักงาน")
If id = "" Then ; ตรวจสอบ id
Exit Sub ; จบการทำงาน
Else
Adodc2.RecordSource = "select * from employee where employee.emp_id like " & id & " "
; ค้นหาข้อมูลจาดตาราง employee โดยค้นหาจาก id
Adodc2.Refresh ; Refresh ข้อมูล
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ;แสดงรูปจากฐานข้อมูล
End If

```

End Sub

2.7 เขียนคำสั่งเพื่อเปิดพอร์ตอนุกรมเพื่อใช้งาน

```
Private Sub Form_Load()  
MSComm1.PortOpen = True ;เปิดพอร์ตอนุกรม พอร์ตที่ 1  
MSComm2.PortOpen = True ;เปิดพอร์ตอนุกรม พอร์ตที่ 2  
x = 0 ; กำหนดให้ x = 0  
y = 0 ; กำหนดให้ y = 0  
z = 0 ; กำหนดให้ z = 0  
x1 = 0 ; กำหนดให้ x1 = 0  
x2 = 0 ; กำหนดให้ x2 = 0  
End Sub
```

2.8 เขียนคำสั่งเพื่อปิดพอร์ตอนุกรม

```
Private Sub Form_Unload(Cancel As Integer)  
MSComm1.PortOpen = False ; ปิดพอร์ตอนุกรม พอร์ตที่ 1  
MSComm2.PortOpen = False ; ปิดพอร์ตอนุกรม พอร์ตที่ 2  
End Sub
```

2.9 เขียนคำสั่งเพื่อแสดงเวลาตามเวลาจริง

```
Private Sub Timer1_Timer()  
Label15.Caption = Format(Time(), "hh:mm:ss") ;เขียนคำสั่งเพื่อแสดงเวลาตามเวลาจริง  
End Sub
```

2.10 เขียนคำสั่งเพื่อแสดงวันตามเวลาจริง

```
Private Sub Timer2_Timer()  
Label16.Caption = Format(Date, "DD/MM/YY") ; เขียนคำสั่งเพื่อแสดงวันตามเวลาจริง  
End Sub
```

2.11 เขียนคำสั่งเพื่อรับข้อมูลที่พอร์ต 1 เข้าเปรียบเทียบแล้วจะเรียกข้อมูลของพนักงานแต่ละคน และเขียนคำสั่งเพื่อแสดงเวลาเข้า-ออกของพนักงาน

```

Private Sub Timer3_Timer()
Dim id As String ; ตัวแปร id เป็นตัวแปรประเภท String
Dim Data As String ; ตัวแปร Data เป็นตัวแปรประเภท String
Data = MSComm1.Input ; อ่านค่าจากบัพเฟอร์ของพอร์ตอนุกรม เก็บไว้ที่ตัวแปร Data
If Data = "" Then ; ตรวจสอบ Data
Load Form2 ; โหลดฟอร์ม 2
Else
Form2.Hide ; ซ่อนฟอร์ม 2
Form1.Show ; โหว้ฟอร์ม 1
id = Val(Asc(Data) - 96) ; คำนวณค่า Data เก็บไว้ที่ตัวแปร id
Adodc2.RecordSource = "select * from employee where employee.emp_id like " & id & " "
; ค้นหาข้อมูลจากราง employee โดยค้นหาจาก id
Adodc2.Refresh ; Refresh ข้อมูล
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; ตั้งให้ load รูปในตำแหน่งที่
Adodc2อยู่
Adodc1.RecordSource = "select * from timetable where timetable.emp_id like " & "" &
& Text1.Text & "%" & " "
; ค้นหาข้อมูลจากราง timetable โดยค้นหาจาก id ที่ปรากฏใน Text1.Text
Adodc1.Refresh ; Refresh ข้อมูล
Adodc1.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
Text24.Text = Adodc1.Recordset.Fields("timeout").Value ; ให้ Text24 รับค่าเวลาออก
If Text24.Text <> "-----" Then ; ตรวจสอบว่าเวลาออกว่างหรือไม่
Adodc3.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
Adodc1.Recordset.AddNew ; ให้ทำการเพิ่มเรคคอร์ดใหม่
Text17 = Text23 + 1
Text18.Text = Text1.Text
Text19.Text = Label16.Caption ; ให้เก็บค่าวันที่เข้า
Text20.Text = Label15.Caption ; ให้เก็บค่าเวลาที่เข้า
Text21.Text = "-----" ; ให้วันที่ออกเป็นค่าว่าง
Text22.Text = "-----" ; ให้เวลาออกเป็นค่าว่าง
Adodc1.Recordset.Update ; บันทึกค่าลงฐานข้อมูล

```

Adodc3.Refresh

Adodc1.Refresh

Adodc1.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย

If Text1.Text = "1" Then

x = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้

End If

If Text1.Text = "2" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้

y = (Int(Text17.Text) - 101)

End If

If Text1.Text = "3" Then

z = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้

End If

If Text1.Text = "4" Then

x1 = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้

End If

If Text1.Text = "5" Then

x2 = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้

End If

Else

Adodc4.Refresh

If Text1.Text = "1" Then

Adodc4.Recordset.Move (x) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้

End If

If Text1.Text = "2" Then

Adodc4.Recordset.Move (y) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้

End If

If Text1.Text = "3" Then

Adodc4.Recordset.Move (z) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้

End If

If Text1.Text = "4" Then

Adodc4.Recordset.Move (x1) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้

End If

```

If Text1.Text = "5" Then
    Adodc4.Recordset.Move (x2) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
Adodc4.Recordset.Fields("timeout").Value = Label15.Caption ; ลงค่าวันที่ออก
Adodc4.Recordset.Fields("dateout").Value = Label16.Caption ; ลงค่าเวลาที่ออก
Text29.Text = Label16.Caption ; แสดงวันที่ออก
Text30.Text = Label15.Caption ; แสดงเวลาที่ออก
Text16.Text = Text30.Text
Adodc4.Recordset.Update ; บันทึกค่าลงฐานข้อมูล
End If
End If
End Sub

```

2.12 เขียนคำสั่งเพื่อรับข้อมูลที่พอร์ต 2 เข้าเปรียบเทียบกับแล้วจะเรียกข้อมูลของพนักงานแต่ละคน และเขียนคำสั่งเพื่อแสดงเวลาเข้า-ออกของพนักงาน

```

Private Sub Timer4_Timer()
Dim id As String ; ตัวแปร id เป็นตัวแปรประเภท String
Dim Data As String ; ตัวแปร Data เป็นตัวแปรประเภท String
Data = MSComm2.Input ; อ่านค่าจากบัฟเฟอร์ของพอร์ตอนุกรม เก็บไว้ที่ตัวแปร Data
If Data = "" Then ; ตรวจสอบ Data
Load Form2 ; โหลดฟอร์ม 2
Else
Form2.Hide ; ซ่อนฟอร์ม 2
Form1.Show ; โจมฟอร์ม 1
id = Val(Asc(Data) - 96) ; คำนวณค่า Data เก็บไว้ที่ตัวแปร id
Adodc2.RecordSource = "select * from employee where employee.emp_id like " & id & " "
; ค้นหาข้อมูลจาดตาราง employee โดยค้นหาจาก id
Adodc2.Refresh ; Refresh ข้อมูล
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; สั่งให้loadรูปในตำแหน่งที่
Adodc2อยู่Adodc1.RecordSource = "select * from timetable where timetable.emp_id like " & ""
& Text1.Text & "% " & " "
; ค้นหาข้อมูลจาดตาราง timetable โดยค้นหาจาก id ที่ปรากฏใน Text1.Text

```



```

Adodc1.Refresh ; Refresh ข้อมูล
Adodc1.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
Text24.Text = Adodc1.Recordset.Fields("timeout").Value ; ให้ Text24 รับค่าเวลาออก
If Text24.Text <> "-----" Then ; ตรวจสอบว่าเวลาออกว่างหรือไม่
Adodc3.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
Adodc1.Recordset.AddNew ; ให้ทำการเพิ่มเรคคอร์ดใหม่
Text17 = Text23 + 1
Text18.Text = Text1.Text
Text19.Text = Label16.Caption ; ให้เก็บค่าวันที่เข้า
Text20.Text = Label15.Caption ; ให้เก็บค่าเวลาที่เข้า
Text21.Text = "-----" ; ให้วันที่ออกเป็นค่าว่าง
Text22.Text = "-----" ; ให้เวลาออกเป็นค่าว่าง
Adodc1.Recordset.Update ; บันทึกค่าลงฐานข้อมูล
Adodc3.Refresh
Adodc1.Refresh
Adodc1.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
If Text1.Text = "1" Then
x = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
End If
If Text1.Text = "2" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
y = (Int(Text17.Text) - 101)
End If
If Text1.Text = "3" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
z = (Int(Text17.Text) - 101)
End If
If Text1.Text = "4" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
x1 = (Int(Text17.Text) - 101)
End If
If Text1.Text = "5" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
x2 = (Int(Text17.Text) - 101)
End If
Else

```

```

Adodc4.Refresh
If Text1.Text = "1" Then
    Adodc4.Recordset.Move (x) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "2" Then
    Adodc4.Recordset.Move (y) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "3" Then
    Adodc4.Recordset.Move (z) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "4" Then
    Adodc4.Recordset.Move (x1) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "5" Then
    Adodc4.Recordset.Move (x2) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
Adodc4.Recordset.Fields("timeout").Value = Label15.Caption ; ลงค่าวันที่ออก
Adodc4.Recordset.Fields("dateout").Value = Label16.Caption ; ลงค่าเวลาที่ออก
Text29.Text = Label16.Caption ; แสดงวันที่ออก
Text30.Text = Label15.Caption ; แสดงเวลาที่ออก
Text16.Text = Text30.Text
Adodc4.Recordset.Update ; บันทึกค่าลงฐานข้อมูล
End If
End If
End Sub

```

Form 2

2.13 เขียนคำสั่งเพื่อปิดฟอร์มที่ 2

```

Private Sub Command1_Click() ; ปุ่มปิด
Form2.Hide ; ซ่อนฟอร์มที่ 2
End Sub

```

2.14 เขียนคำสั่งเพื่อค้นหาเวลาเข้า-ออกและจำนวนการเข้า-ออกของพนักงานในแต่ละวัน

```
Private Sub Command2_Click() ; ปุ่มค้นหา
Dim a As Integer ; ตัวแปร a เป็นตัวแปรประเภท Integer
Adodc2.RecordSource = "select timetable.emp_id , timetable.date ,
timetable.time,timetable.dateout , timetable.timeout from timetable where timetable.emp_id
like " & "'" & Text1.Text & "%" & " and timetable.date like " & "'" & Text2.Text & "%" & "
and timetable.timeout like " & "'" & "-----" & "%" & "
; ทำการเลือกค่าจาดตาราง timetable
Adodc2.Refresh
a = Int(Adodc2.Recordset.RecordCount) ; ทำการนับเรคคอร์ด
Adodc1.RecordSource = "select timetable.emp_id ,
timetable.dat,timetable.time,timetable.dateout , timetable.timeout from timetable where
timetable.emp_id like " & "'" & Text1.Text & "%" & " and timetable.date like " & "'" &
Text2.Text & "%" & "
; ทำการเลือกค่าจาดตาราง timetable
Adodc1.Refresh
DataGrid1.Refresh ; ให้ DataGrid1รับค่าจาก Adodc1
ใหม่
Label3.Caption = Int(Adodc1.Recordset.RecordCount) ; แสดงผลการนับเวลาเข้า
Label6.Caption = Int(Adodc1.Recordset.RecordCount) - a ; แสดงผลการนับเวลาออก
End Sub
```

Form 3

2.15 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดแรกของข้อมูล

```
Private Sub Command1_Click() ; ปุ่มเลื่อนเรคคอร์ด ไปยังเรคคอร์ด
แรก
Adodc2.Recordset.MoveFirst ; เลื่อนเรคคอร์ด ไปยังเรคคอร์ด
แรก
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; ตั้งให้loadรูปในตำแหน่งที่
Adodc2อยู่
End Sub
```

2.16 เขียนคำสั่งเพื่อยกเลิกการลบข้อมูลพนักงาน

```
Private Sub Command10_Click() ; ปุ่มยกเลิกการลบข้อมูลพนักงาน
Adodc2.Refresh
Frame2.Visible = False ; ให้เฟรม 2 มีคุณสมบัติในการแสดงเป็น
False
End Sub
```

2.17 เขียนคำสั่งเพื่อยืนยันการลบข้อมูลพนักงาน

```
Private Sub Command11_Click() ; ปุ่มยืนยันการลบข้อมูลพนักงาน
Adodc2.Recordset.Delete ; ลบข้อมูลพนักงาน
Adodc2.Refresh
Frame2.Visible = False ; ให้เฟรม 2 มีคุณสมบัติในการแสดงเป็น
False
MsgBox ("ข้อมูลถูกลบแล้ว") ; แสดงข้อความ("ข้อมูลถูกลบแล้ว ")
End Sub
```

2.18 เขียนคำสั่งเพื่อยืนยันการแก้ไขข้อมูลพนักงาน

```
Private Sub Command12_Click() ; ปุ่มยืนยันการแก้ไขข้อมูล
Adodc2.Recordset.Update ; แก้ไขข้อมูลลงฐานข้อมูล
Adodc2.Refresh
Frame3.Visible = False ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น
False
MsgBox ("ข้อมูลถูกแก้ไขแล้ว") ; แสดงข้อความ("ข้อมูลถูกแก้ไขแล้ว ")
End Sub
```

2.19 เขียนคำสั่งเพื่อยกเลิกการแก้ไขข้อมูลพนักงาน

```
Private Sub Command13_Click() ; ปุ่มยกเลิกการแก้ไขข้อมูลพนักงาน
Adodc2.Refresh
```

```

Frame3.Visible = False ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น
False
End Sub

```

2.20 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดก่อนหน้า

```

Private Sub Command2_Click()
If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF
    Adodc2.Refresh ; Refresh ข้อมูล
Else
    Adodc2.Recordset.MovePrevious ; เลื่อนไปเรคคอร์ดก่อนหน้า
If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF
Else
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
End If
End If
End Sub

```

2.21 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดถัดไป

```

Private Sub Command3_Click()
If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF
    Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
    Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
Else
    Adodc2.Recordset.MoveNext ; เลื่อนไปเรคคอร์ดถัดไป
If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF
    Picture1.Refresh ; Refresh รูป
Else
    Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
End If
End If
End Sub

```

2.22 เขียนคำสั่งเพื่อเลื่อนไปเรคคอร์ดสุดท้าย

```
Private Sub Command4_Click()  
    Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย  
    Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล  
End Sub
```

2.23 เขียนคำสั่งเพื่อเพิ่มข้อมูลพนักงาน

```
Private Sub Command5_Click() ; ปุ่มเพิ่มข้อมูลพนักงาน  
    Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย  
    Adodc2.Recordset.AddNew ; เพิ่มเรคคอร์ดใหม่  
    Frame1.Visible = True ; ให้เฟรม 1 มีคุณสมบัติในการแสดงเป็น  
    True  
End Sub
```

2.24 เขียนคำสั่งเพื่อลบข้อมูลพนักงาน

```
Private Sub Command6_Click() ; ปุ่มลบข้อมูลพนักงาน  
    Frame2.Visible = True ; ให้เฟรม 2 มีคุณสมบัติในการแสดงเป็น  
    True  
End Sub
```

2.25 เขียนคำสั่งเพื่อแก้ไขข้อมูลพนักงาน

```
Private Sub Command7_Click() ; ปุ่มแก้ไขข้อมูลพนักงาน  
    Frame3.Visible = True ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น  
    True  
End Sub
```

2.26 เขียนคำสั่งเพื่อยืนยันการแก้ไขข้อมูลพนักงาน

```
Private Sub Command8_Click() ; ปุ่มยืนยันการแก้ไขข้อมูล  
    Adodc2.Recordset.Update ; แก้ไขข้อมูลลงฐานข้อมูล
```

```
Adodc2.Refresh
Frame1.Visible = False
End Sub
```

2.27 เขียนคำสั่งเพื่อยกเลิกข้อมูลพนักงาน

```
Private Sub Command9_Click() ; ปุ่มยกเลิกการแก้ไขข้อมูล
Adodc2.Refresh
Frame1.Visible = False ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น
False
End Sub
```

MDI Form1

2.28 เขียนคำสั่งเพื่อแสดงหน้าต่างข้อมูลพนักงาน

```
Private Sub emp_Click()
Form3.Show ; แสดงหน้าต่างข้อมูลพนักงาน
Form2.Hide ; ซ่อนหน้าต่างค้นหาพนักงาน
Form1.Hide ; ซ่อนหน้าต่างข้อมูลการเข้าทำงานของพนักงาน
End Sub
```

2.29 เขียนคำสั่งเพื่อแสดงหน้าต่างค้นหาพนักงาน

```
Private Sub find_Click()
Form1.Show ; แสดงหน้าต่างค้นหาพนักงาน
Form2.Hide ; ซ่อนหน้าต่างข้อมูลการเข้าทำงานของพนักงาน
Form3.Hide ; ซ่อนหน้าต่างข้อมูลพนักงาน
End Sub
```

2.30 เขียนคำสั่งเพื่อแสดงหน้าต่างข้อมูลการเข้าทำงานของพนักงาน

```
Private Sub work_Click()
Form2.Show ; แสดงหน้าต่างข้อมูลการเข้าทำงานของพนักงาน
Form1.Hide ; ซ่อนหน้าต่างค้นหาพนักงาน
Form3.Hide ; ซ่อนหน้าต่างข้อมูลพนักงาน
End
```

3.5 การสร้างฐานข้อมูลด้วย SQL server 2000[5]

3.5.1 การสร้างฐานข้อมูลโดยใช้ Enterprise Manager

การใช้เครื่องมือ Enterprise Manager ในการสร้างฐานข้อมูล เราสามารถนิยามคุณสมบัติต่างๆ ของไฟล์ข้อมูลและไฟล์ทรานแซกชันล็อกได้สะดวก และรวดเร็ว การสร้างฐานข้อมูลโดยใช้ Enterprise Manager มีขั้นตอนดังต่อไปนี้ ซึ่งจะเป็นการสร้างฐานข้อมูล Order ขึ้นมา

1. ให้เราทำการเรียกเครื่องมือ Enterprise Manager โดย Click mouse ที่ปุ่ม



จากนั้นให้เลือก Program>Microsoft SQL Server> Enterprise Manager จะปรากฏเครื่องมือ Enterprise Manager ขึ้นมา

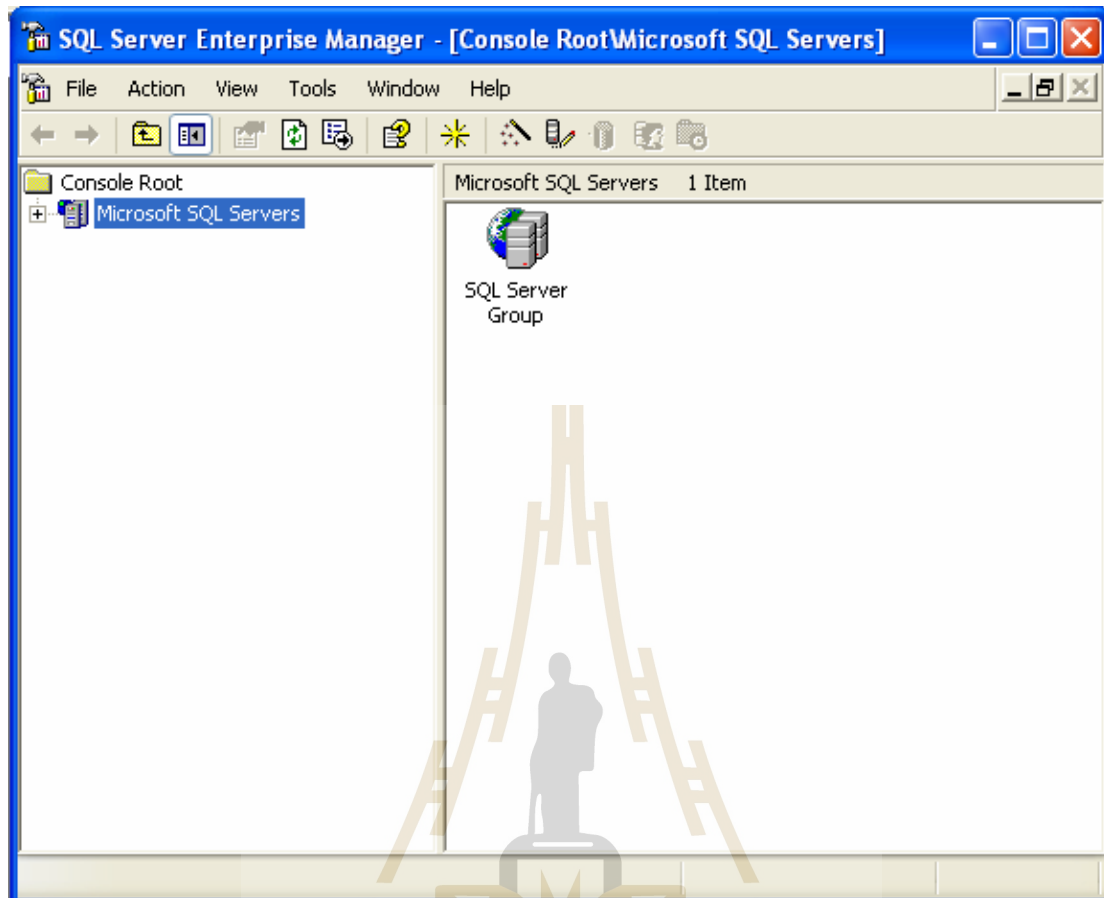
2. ให้เราเลือกว่า จะทำการสร้างฐานข้อมูลลงบนเซิร์ฟเวอร์ใด และอยู่ที่เซิร์ฟเวอร์กรุ๊ปไหน โดยการ Click mouse เลือกตามต้องการ ซึ่งตัวอย่างในรูปจะมีแค่เซิร์ฟเวอร์กรุ๊ปเดียวชื่อ SQL Server Group และมีเซิร์ฟเวอร์เดียวคือ KIM(Window NT) เมื่อเราเลือกแล้ว จะปรากฏฐานข้อมูลต่างๆ ทั้งหมดที่มีอยู่ในเซิร์ฟเวอร์ตัวที่เราเลือกขึ้นมา

3. เราสามารถทำการสร้างฐานข้อมูลได้โดย Click mouse ที่พื้นที่ว่างด้านขวา แล้วเลือก New Database หรือให้เลือกเมนู Action>New Database ก็ได้

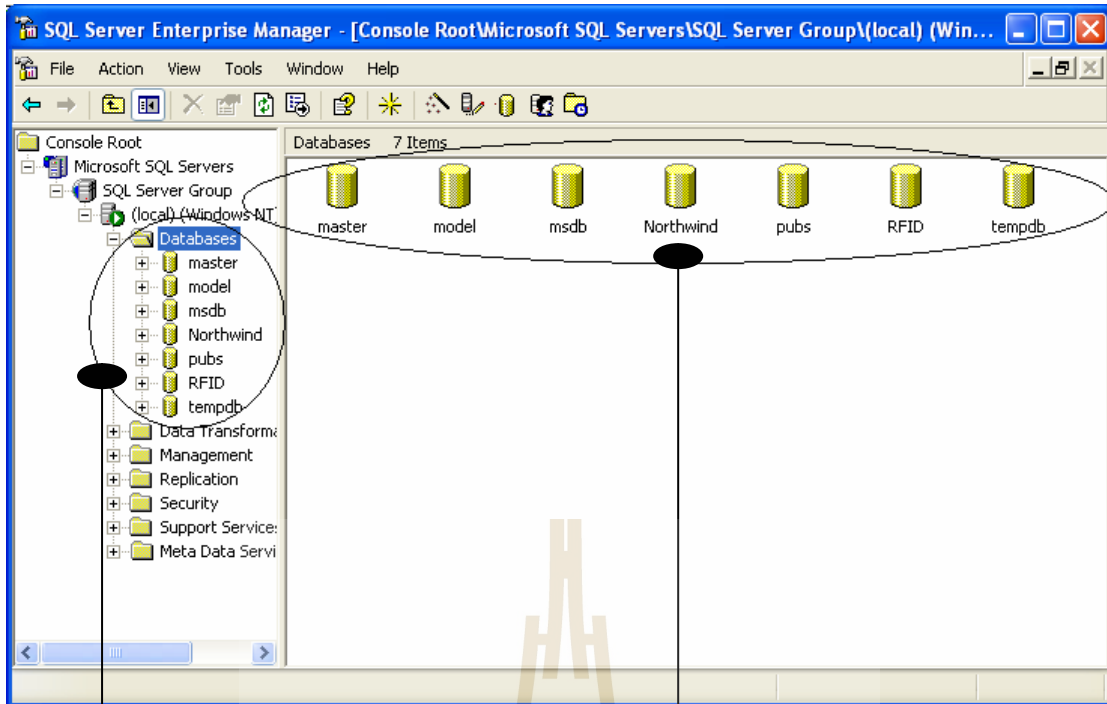
- 4.จากนั้นจะปรากฏไดอะล็อกซ์ Database Properties ที่แท็บ General ให้เราทำการตั้งชื่อฐานข้อมูลที่เราต้องการสร้างลงไป เสร็จแล้วให้ Click mouse ที่แท็บ Data File เพื่อทำการนิยามคุณสมบัติต่างๆ ไฟล์ข้อมูลในฐานข้อมูลของเรา



1. ให้เราทำการเรียกเครื่องมือ Enterprise Manager ขึ้นมา



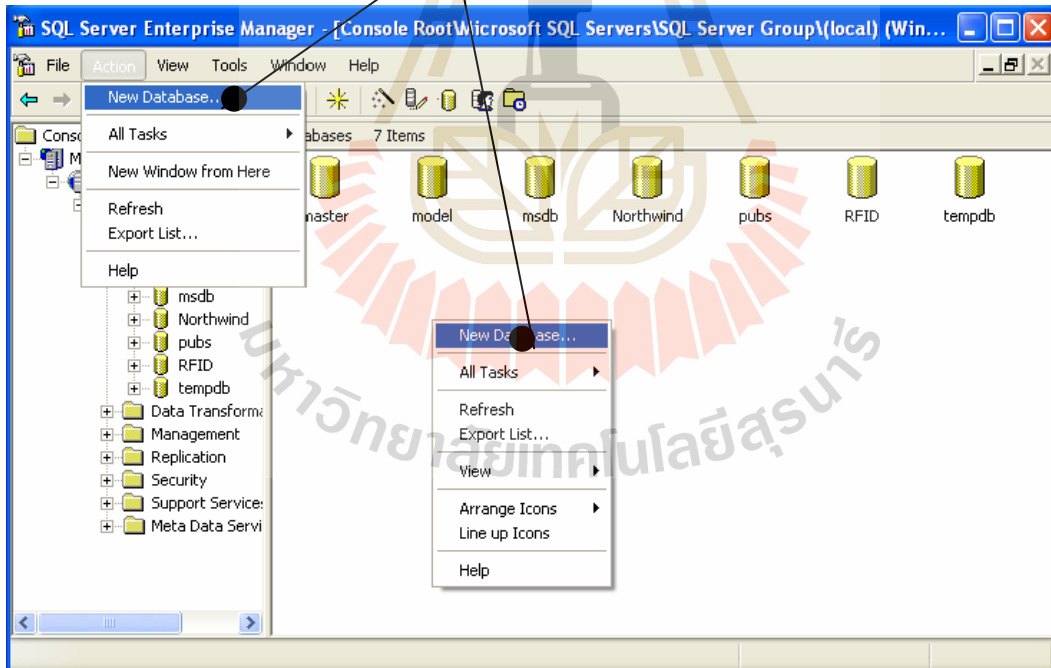
รูปที่ 3.7 การเรียกเครื่องมือ Enterprise Manager ขึ้นมา



2.1 Click Mouse เลือกเซิร์ฟเวอร์ที่ต้องการสร้างข้อมูล 2.2 แสดงฐานข้อมูลทั้งหมดที่มีอยู่ในเซิร์ฟเวอร์

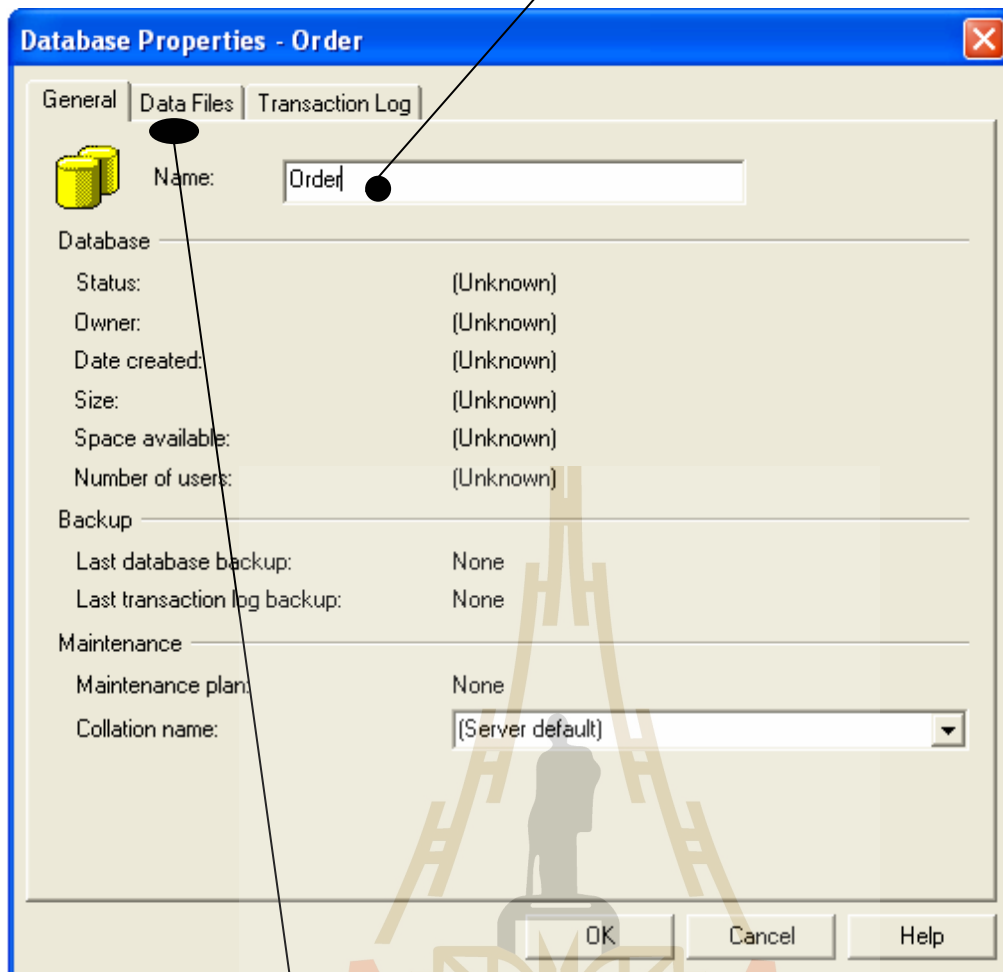
รูปที่ 3.8 แสดงฐานข้อมูลทั้งหมดที่มีอยู่ในเซิร์ฟเวอร์

3.1 ให้เรา Click Mouse ปุ่มขวา และเลือก New Database... หรือเลือกเมนู Action>New Database.. ก็ได้



รูปที่ 3.9 แสดงการสร้างฐานข้อมูล

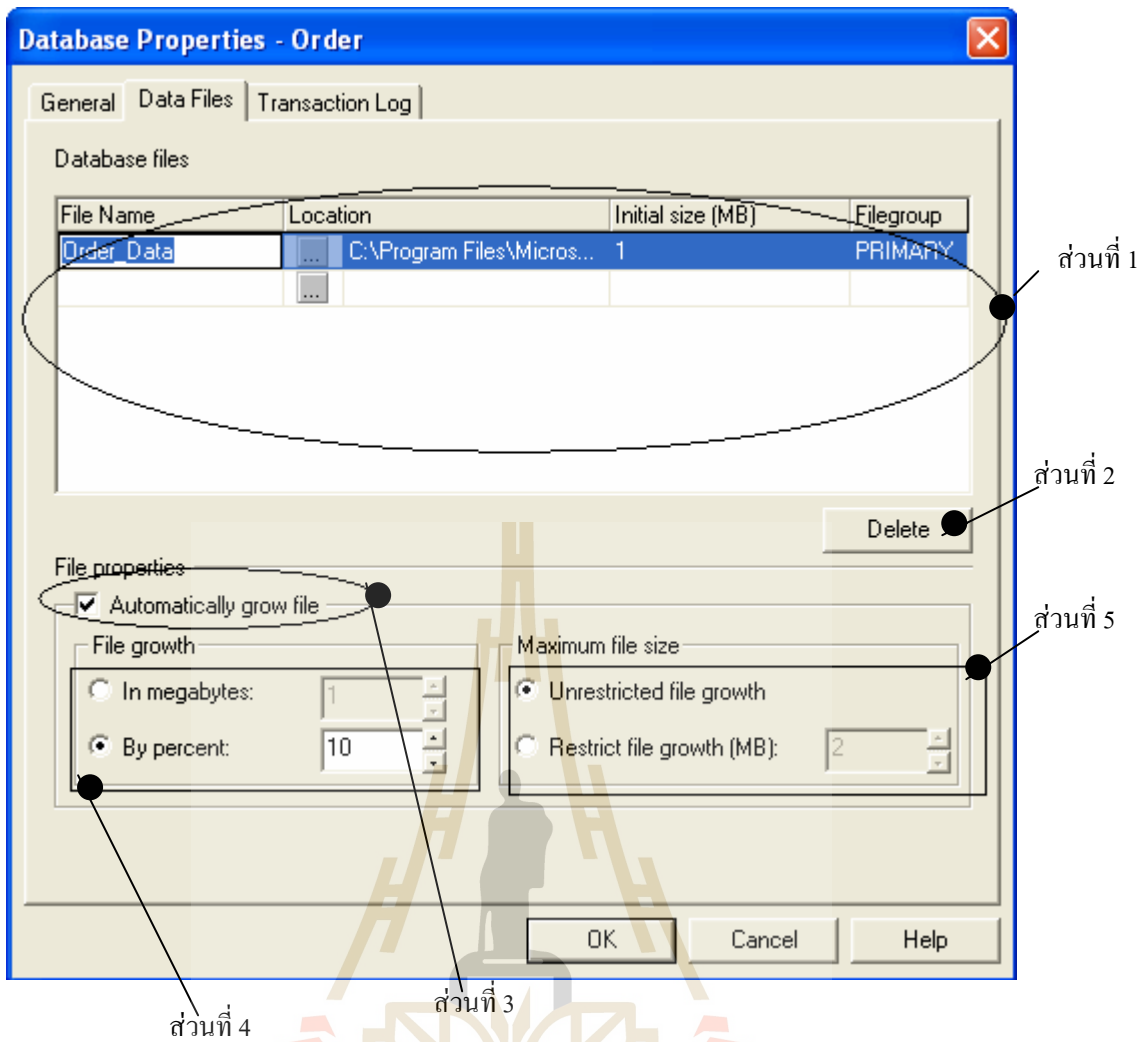
4.1 กำหนดชื่อฐานข้อมูลที่เราต้องการ



4.2 Click Mouse ที่แท็บ Data File

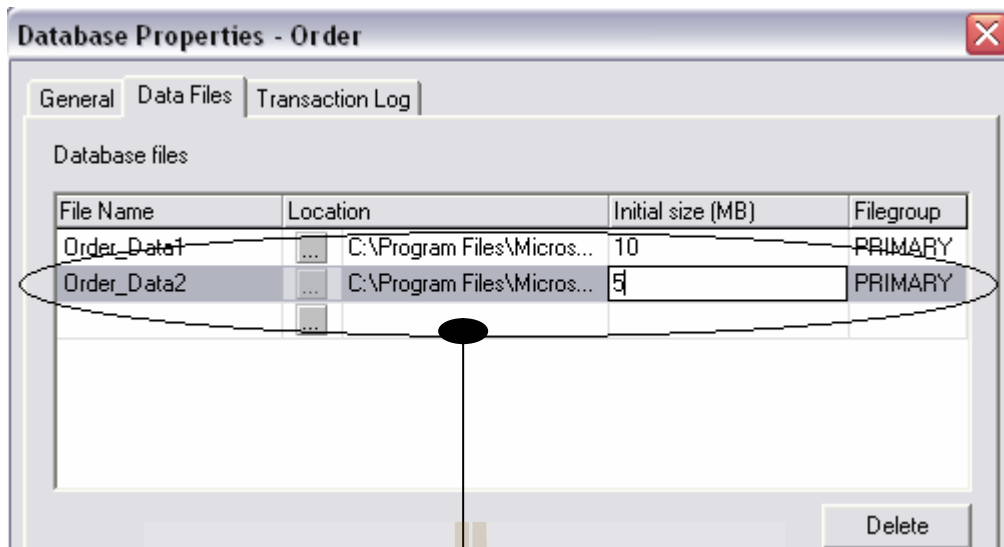
รูปที่ 3.10 แสดงการตั้งชื่อฐานข้อมูล

5. จะปรากฏไดอะล็อกซ์ส่วนที่นิยามคุณสมบัติต่างๆ ของไฟล์ข้อมูล ดังรูป โดยในไดอะล็อกซ์นี้สามารถนิยามได้ทั้งจำนวนไฟล์ในฐานข้อมูล, ขนาดของไฟล์ข้อมูล, รูปแบบการขยายขนาดไฟล์ข้อมูล ซึ่งไดอะล็อกซ์นี้ประกอบไปด้วย 5 ส่วนดังต่อไปนี้



รูปที่ 3.11 แสดงไดอะล็อกซ์ Database Properties

ส่วนที่ 1 เป็นส่วนที่ใช้กำหนดไฟล์ว่าจะให้มีกี่ไฟล์, ชื่ออะไร, เก็บอยู่ที่ไหน, มีขนาดเริ่มต้นเท่าไร และเก็บอยู่ในไฟล์กรุปใด โดยเราสามารถเข้าไปแก้ไขเหล่านี้ได้โดยการ Click Mouse ที่ช่องต่างๆ แล้วเพิ่มค่าที่เราต้องการลงไป และเราสามารถเพิ่มจำนวนไฟล์ข้อมูลโดย Click Mouse ที่ช่องว่างสีขาวแล้วทำการใส่ค่าต่างๆ ที่นิยามไฟล์ข้อมูลที่ต้องการเพิ่มลงไปดังรูป



1. ทำการ Click Mouse ที่ช่องว่างสีขาวแล้วใส่พารามิเตอร์ที่ต้องการลงไป
รูปที่ 3.12 แสดงการกำหนดค่าพารามิเตอร์

ส่วนที่ 2 เป็นส่วนของปุ่ม **Delete** ใช้ในการลบไฟล์ข้อมูลที่เราเลือกไว้ (ซึ่งจะปรากฏแถบสีน้ำเงินบนไฟล์ข้อมูลนั้นในส่วนที่ 1)

ส่วนที่ 3 เป็นส่วนที่ใช้กำหนดว่า จะให้มีการขยายขนาดของไฟล์ข้อมูลแบบอัตโนมัติ เมื่อไฟล์ข้อมูลที่เกิดข้อมูลเต็มขึ้นมาหรือไม่ ถ้าเลือกไว้จะทำให้ตัวเลือกต่างๆ ในส่วนที่ 4 และส่วนที่ 5 เลือกได้ด้วย

ส่วนที่ 4 เป็นส่วนที่ใช้กำหนดว่า การขยายขนาดของไฟล์ข้อมูลแบบอัตโนมัติจะเพิ่มขึ้นในรูปแบบใด เมื่อไฟล์ข้อมูลนั้นเกิดเต็มขึ้นมา ซึ่งจะมีอยู่ 2 ตัวเลือกด้วยกัน ดังต่อไปนี้

1. In megabytes เป็นการกำหนดว่า จะขยายไฟล์ข้อมูลเพิ่มขึ้นทีละเท่าไรในหน่วยเมกะไบต์ ซึ่งเราสามารถได้ในช่องทางขวามือ

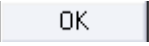
2. By percent เป็นการกำหนดว่า จะขยายเพิ่มทีละกี่เปอร์เซ็นต์ของขนาดไฟล์ข้อมูลเดิม ซึ่งเราสามารถกำหนดเปอร์เซ็นต์ได้ทางช่องทางขวามือ

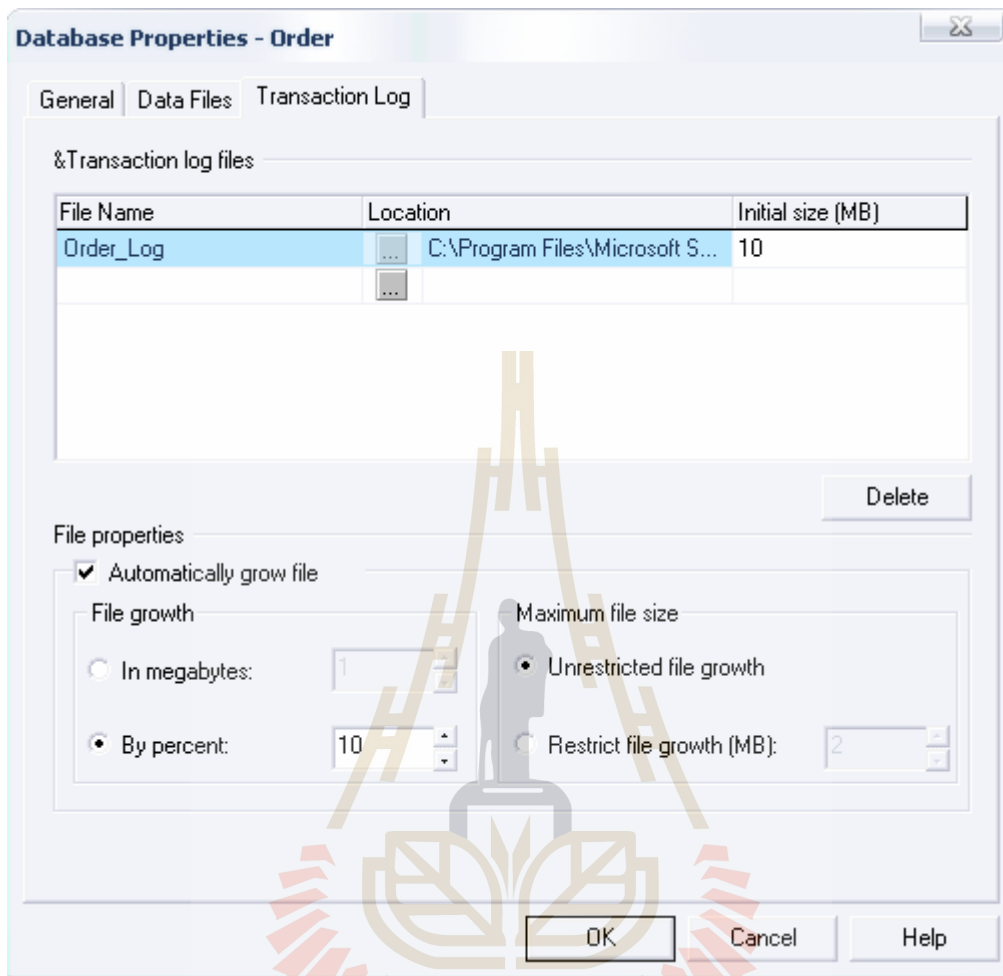
ส่วนที่ 5 เป็นการกำหนดขนาดของไฟล์ข้อมูลสูงสุด ซึ่งมี 2 ตัวเลือกดังต่อไปนี้

1. Unrestricted file growth จะเป็นการกำหนดขนาดของไฟล์ข้อมูลสามารถขยายได้อย่างไม่จำกัด

2. Restrict file growth (MB) จะเป็นการกำหนดขนาดสูงสุดของไฟล์ ซึ่งเมื่อขนาดไฟล์นั้นถึงขนาดที่เรากำหนดแล้ว ก็จะไม่สามารถขยายต่อไป

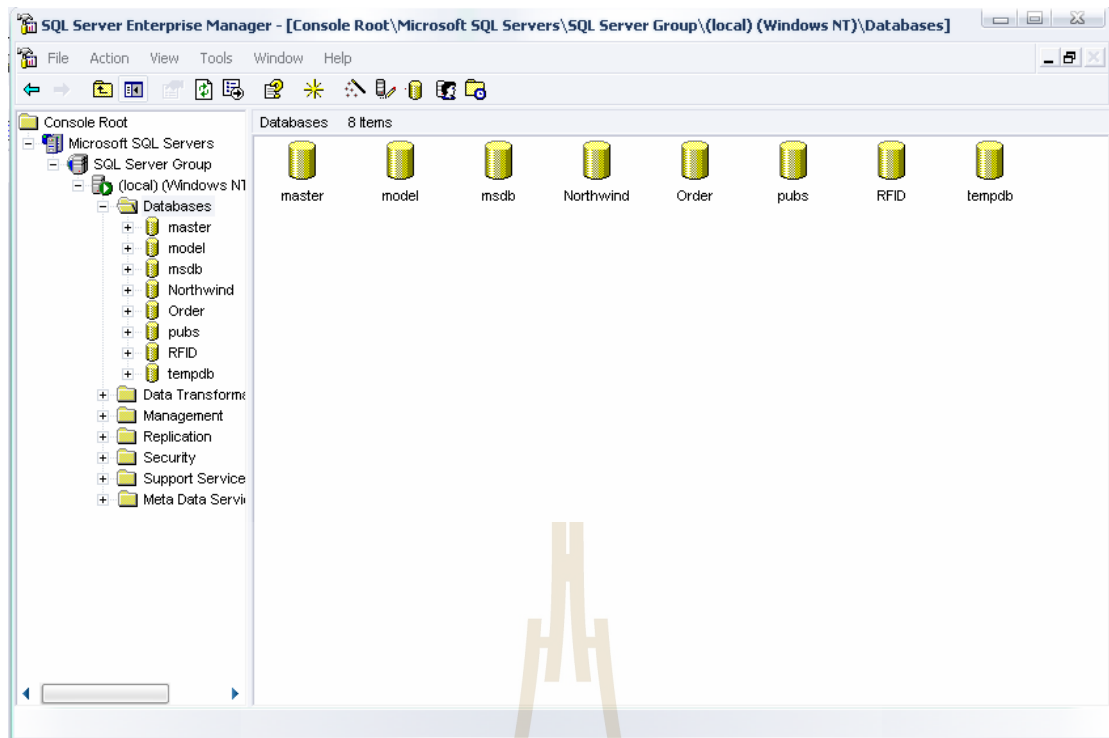
เมื่อทำการนิยามไฟล์ข้อมูลเสร็จแล้ว ให้เรา Click Mouse ที่แท็บ Transaction Log เพื่อทำการนิยามคุณสมบัติต่างๆ ไฟล์ทรานแซคชันล็อกต่อไป

6. จะปรากฏไดอะล็อกซ์ให้กำหนดค่าต่างๆ ที่ใช้นิยามไฟล์ทรานแซ็คชั่นล็อก ซึ่งคล้ายกับการกำหนดคุณสมบัติต่างๆ เสร็จแล้วให้ Click Mouse ที่ปุ่ม 



รูปที่ 3.13 แสดงการกำหนดคุณสมบัติต่างๆ ที่ไฟล์ทรานแซ็คชั่นล็อก

7. จะปรากฏฐานข้อมูล Orders ที่เราได้ทำการสร้างขึ้นมาดังรูปข้างล่าง

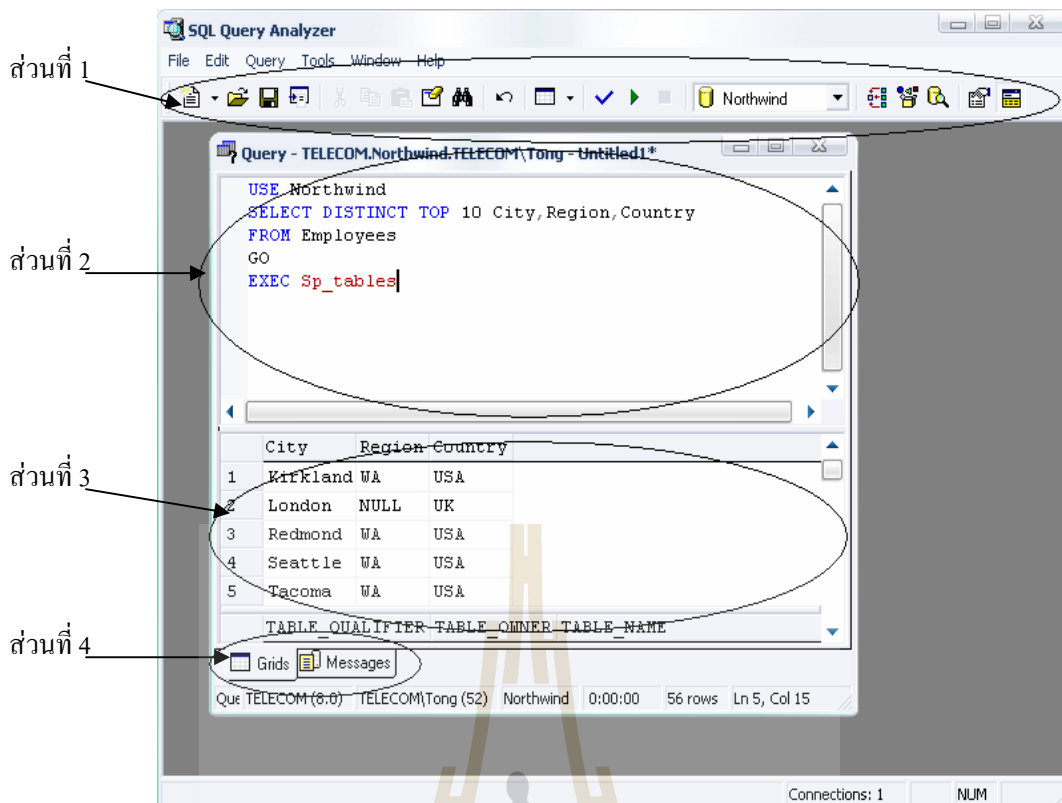


รูปที่ 3.14 แสดงฐานข้อมูล Orders ที่เราได้ทำการสร้างขึ้น

รู้จักกับ Query Analyzer

การสร้างและจัดการกับฐานข้อมูลนั้นนอกจากการใช้เครื่องมือ Enterprise Manager แล้วเรายังสร้างได้โดยใช้คำสั่ง Transact-SQL ได้อีกด้วย แต่ก่อนที่เราจะกล่าวถึงวิธีการสร้างนั้นเราต้องทำความรู้จักกับเครื่องมือ Query Analyzer ซึ่งเป็นเครื่องมือที่ใช้ในการทำงานร่วมกับคำสั่ง Transact-SQL เสียก่อน

Query Analyzer เป็นเครื่องมือที่ใช้ทำงานร่วมกับคำสั่ง Transact-SQL โดยการรับคำสั่ง Transact-SQL จากผู้ใช้แล้วส่งไปให้ SQL Server เพื่อประมวลผล จากนั้นก็จะรับผลลัพธ์ที่ได้จาก SQL Server แสดงผลออกมาทางหน้าจอ ซึ่งเราสามารถจะใช้เครื่องมือ Query Analyzer นี้ในการวิเคราะห์คำสั่ง Transact-SQL นี้ก่อนที่จะนำคำสั่ง Transact-SQL ไปใช้งานร่วมกับแอปพลิเคชันจริงได้อีกด้วย และนอกจากนี้เครื่องมือ Query Analyzer ยังมีความสามารถต่างๆ การใช้เทมเพลต, Object Browser, Object Search ซึ่งจะทำให้การใช้งาน Query Analyzer จัดการฐานข้อมูลมีความสะดวกมากขึ้น โดยหน้าต่างของเครื่องมือ Query Analyzer เป็นดังรูป



รูปที่ 3.15 แสดงส่วนประกอบต่างๆ ของเครื่องมือ Query Analyzer

โดยเครื่องมือ Query Analyzer ประกอบด้วยส่วนต่างๆ ดังนี้

- ส่วนที่ 1 ทูลบาร์ ที่ใช้ควบคุมการทำงานของ Query Analyzer
- ส่วนที่ 2 หน้าต่างคำสั่ง ที่ใช้ในการพิมพ์คำสั่ง Transact-SQL
- ส่วนที่ 3 หน้าต่างแสดงผล ใช้แสดงผลลัพธ์ของคำสั่ง Transact-SQL ซึ่งส่งประมวลผลที่เซิร์ฟเวอร์
- ส่วนที่ 4 แท็บเลือกแสดงผลข้อมูล

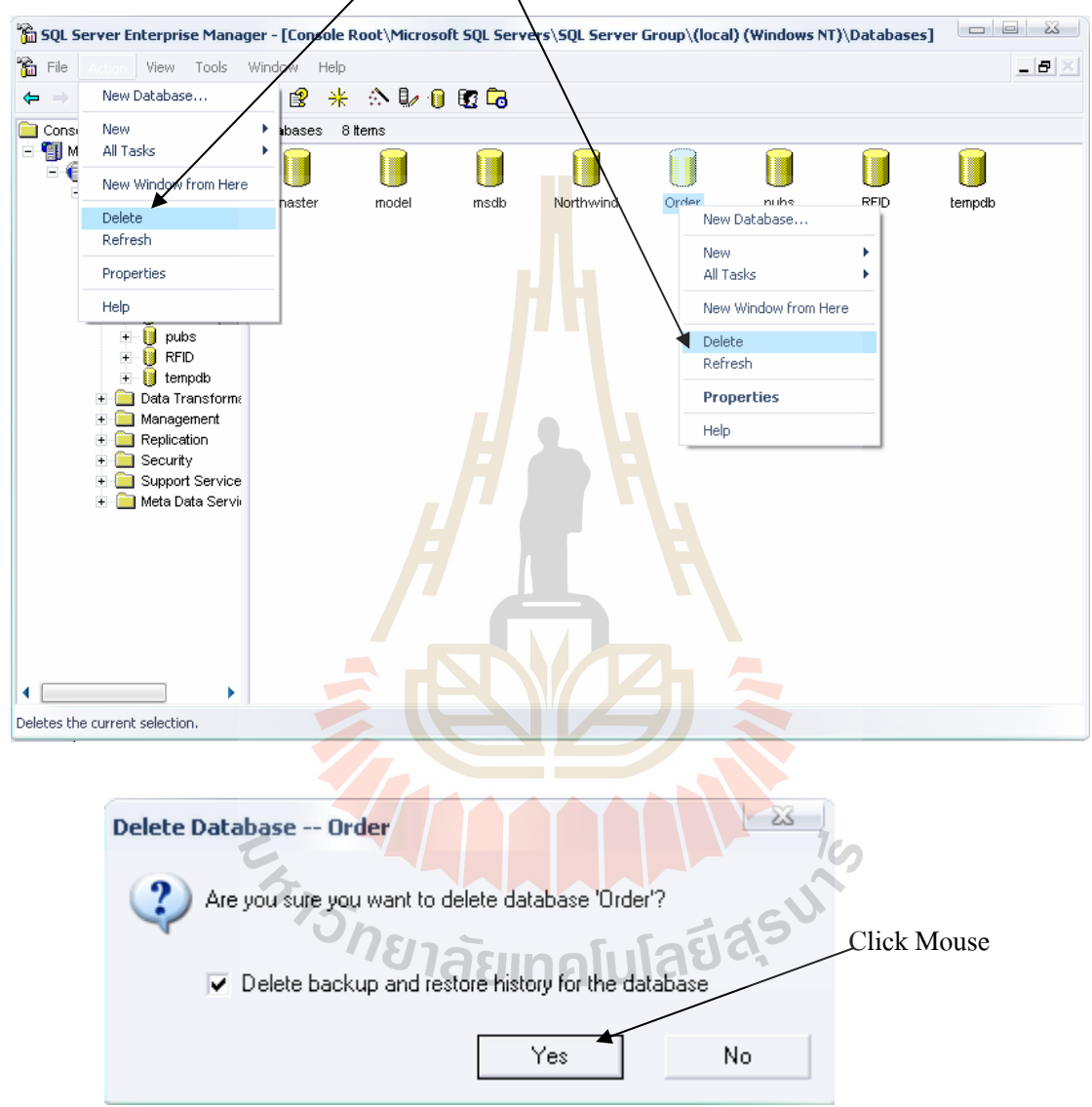
การลบฐานข้อมูลโดยใช้ Enterprise Manager

เราสามารถใช้อุปกรณ์ Enterprise Manager ในการลบฐานข้อมูลที่ไม่ต้องการใช้งานอีกแล้วในระบบฐานข้อมูลออกไปได้ โดยทำตามขั้นตอนดังต่อไปนี้

1. ทำการเลือกฐานข้อมูลที่ต้องการจากนั้นให้ Click Mouse ที่ปุ่มขวาแล้วเลือกคำสั่ง Delete หรือ Click Mouse เลือกที่ เมนู Action>Delete ก็ได้
2. จากนั้นจะปรากฏไดอะล็อกยืนยันว่าจะลบฐานข้อมูลแน่หรือไม่ และมีเช็ทซ์บ็อกซ์ว่าจะได้ทำการข้อมูลที่เราได้ทำการแบ็กอัพ หรือรีโอสต์ด้วยหรือไม่ ถ้าเลือกจะทำให้ไม่สามารถกู้คืนมาได้

เมื่อเลือกเสร็จแล้วให้เรา Click Mouse ที่ปุ่ม **Yes** เป็นอันว่า ฐานข้อมูลได้ถูกลบไปเรียบร้อยแล้ว

1. เลือกฐานข้อมูลที่ต้องการลบ จากนั้น Click Mouse ปุ่มแล้วเลือกคำสั่ง Delete หรือเลือกเมนู Action>Delete ก็ได้



รูปที่ 3.16 แสดงการลบฐานข้อมูลโดยใช้ Enterprise Manager

การแก้ไของค์ประกอบต่างๆ ในฐานข้อมูล

การแก้ไขฐานข้อมูลใน SQL Server เราสามารถแก้ไของค์ประกอบต่างๆ ของฐานข้อมูลได้ดังต่อไปนี้

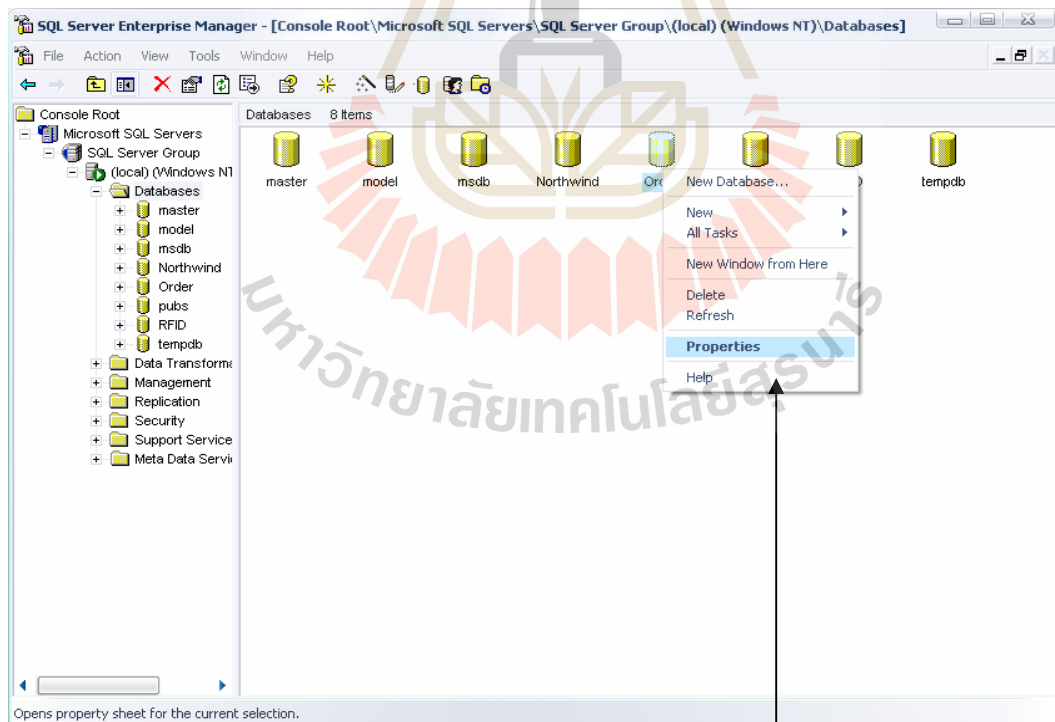
1. การเพิ่ม, การลบ, การแก้ไขคุณสมบัติต่างๆ ไฟล์ข้อมูล หรือไฟล์ทรานแซ็คชั่นล็อก
2. การกำหนดคุณสมบัติต่างๆ ของไฟล์กลุ่ม

ซึ่งการแก้ไขของค้ประกอบต่างๆ ในฐานข้อมูลนั้นทำให้เราได้ฐานข้อมูลที่มีความเหมาะสมกับระบบงานที่เปลี่ยนแปลงไปได้ โดยใน SQL Server นั้นสามารถใช้ได้ทั้งเครื่องมือ Enterprise Manager และคำสั่ง Transact – SQL แต่เนื้อหาในหัวข้อนี้จะกล่าวถึงการใช้เครื่องมือ Enterprise Manager วิธีเดียวกันนั้น

การแก้ไขฐานข้อมูลโดยใช้ Enterprise Manager

การแก้ไขฐานข้อมูลด้วย Enterprise Manager สามารถทำได้ค่อนข้างสะดวกและรวดเร็ว โดยมีขั้นตอนต่างๆ ดังต่อไปนี้

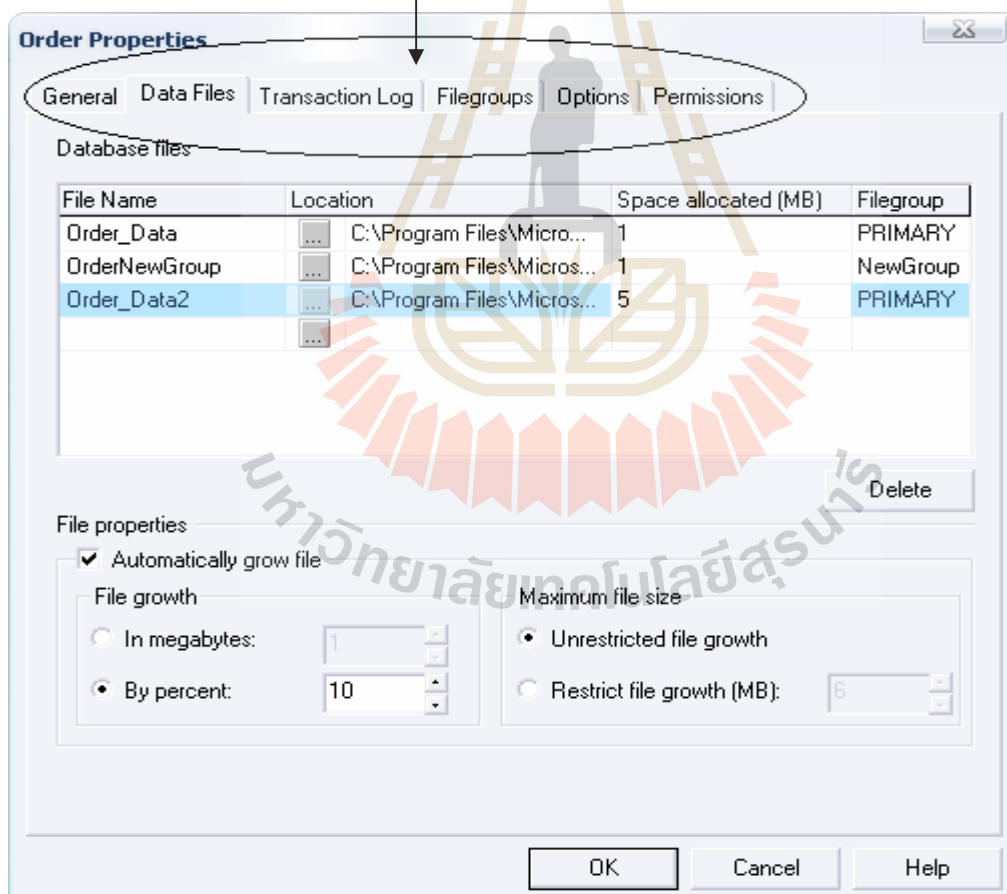
1. ทำการเลือกฐานข้อมูลที่ต้องการ จากนั้น Click Mouse ปุ่มขวาแล้วเลือก Properties
2. จะปรากฏไดอะล็อกซ์ Database Properties ดังรูป ซึ่งไดอะล็อกซ์นี้จะมีแท็บต่างๆ ให้ทำการแก้ไขข้อมูลได้ตามต้องการ ซึ่งไดอะล็อกซ์นี้จะคล้ายๆ กับไดอะล็อกซ์ที่ใช้สร้างฐานข้อมูล แต่จะแท็บเพิ่มขึ้นมาอีก 3 แท็บ คือ Filegroups, Option, Permissions



2.1 .ให้ Click Mouse ปุ่มขวาเลือกฐานข้อมูล จากนั้นให้เลือกคำสั่ง Properties

รูปที่ 3.17 แสดงทำการเลือกฐานข้อมูลที่ต้องการ

2.2 แท็บต่างๆ ที่ใช้ในการแก้ไขฐานข้อมูลตามที่เราต้องการ




รูปที่ 3.18 แสดงไดอะล็อก Database Properties

โดยในไดอะล็อกซ์ข้างบน สามารถแก้ไขฐานข้อมูล โดยการ Click Mouse ที่แท็บต่างๆ แล้วเข้าไปกำหนดค่าตามที่ต้องการ ซึ่งแท็บต่างๆ แต่ละแท็บมีความหมายดังนี้

- Data Files เป็นแท็บที่ใช้ในการจัดการกับไฟล์ข้อมูล เช่น การเพิ่มไฟล์ข้อมูล, การเปลี่ยนไฟล์ข้อมูล, การเปลี่ยนแปลงไฟล์ข้อมูล ซึ่งมีวิธีการกำหนดค่าเหมือนกับการสร้างฐานข้อมูลที่ได้อธิบายไปแล้ว

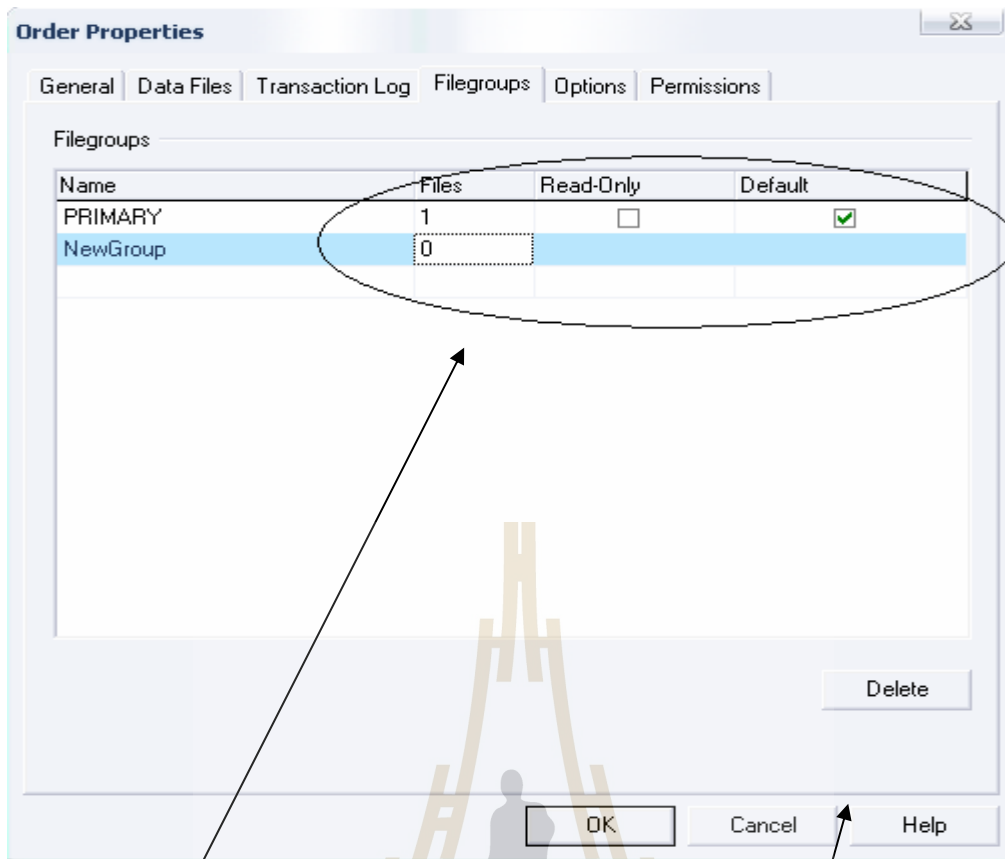
- Transaction Logs เป็นแท็บที่ใช้ในการจัดการกับไฟล์ทรานแซกชันล็อก ซึ่งการใช้งานจะคล้ายๆ กับแท็บ Data Files

- File Group เป็นแท็บที่ใช้จัดการการลบ และเปลี่ยนคุณสมบัติต่างๆ ของไฟล์กรุป ซึ่งสามารถลบไฟล์กรุปได้โดย Click Mouse เลือกไฟล์กรุปที่ต้องการแล้ว Click Mouse ที่ปุ่ม  ส่วนคุณสมบัติของไฟล์กรุปมีอยู่ 2 คุณสมบัติดังนี้

โดยสามารถ Click Mouse ที่เช็คบ็อกซ์ เพื่อกำหนดคุณสมบัติของไฟล์กรุปได้

1. เช็คบ็อกซ์ Read-Only ถ้าทำการเลือกไว้ จะทำให้ไฟล์ข้อมูลทั้งหมดในไฟล์กรุปนั้นเป็นแบบอ่านได้อย่างเดียว แต่ถ้าไม่เลือกไว้ สามารถที่จะอ่าน หรือเขียนไฟล์ข้อมูลทั้งหมดที่อยู่ในไฟล์กรุปนี้ได้

2. เช็คบ็อกซ์ Default ไฟล์กรุปที่จะทำการเลือกเช็คบ็อกซ์นี้ได้ จะมีเพียงไฟล์กรุปเดียวเท่านั้น ซึ่งเมื่อเลือกแล้ว เวลาที่ทำการสร้างออบเจกต์ต่างๆ ในระบบฐานข้อมูล ถ้าไม่ได้กำหนดว่าสร้างไว้บนไฟล์กรุปใด SQL Server จะทำการสร้างบนไฟล์กรุปที่ได้กำหนดคุณสมบัติ Default เอาไว้



2. เลือกเช็ทบ็อกซ์เพื่อกำหนดคุณสมบัติต่างๆ ของไฟล์กรุป

1. Click Mouse เพื่อเลือกไฟล์กรุปที่เลือกไว้

รูปที่ 3.19 แสดงการกำหนดคุณสมบัติของไฟล์กรุป

- Options กับ Permission เป็นแท็บที่ใช้เซตค่าอุปชั้นต่างๆ ฐานข้อมูล และเซตค่าสิทธิต่างๆ ในการจัดการฐานข้อมูล

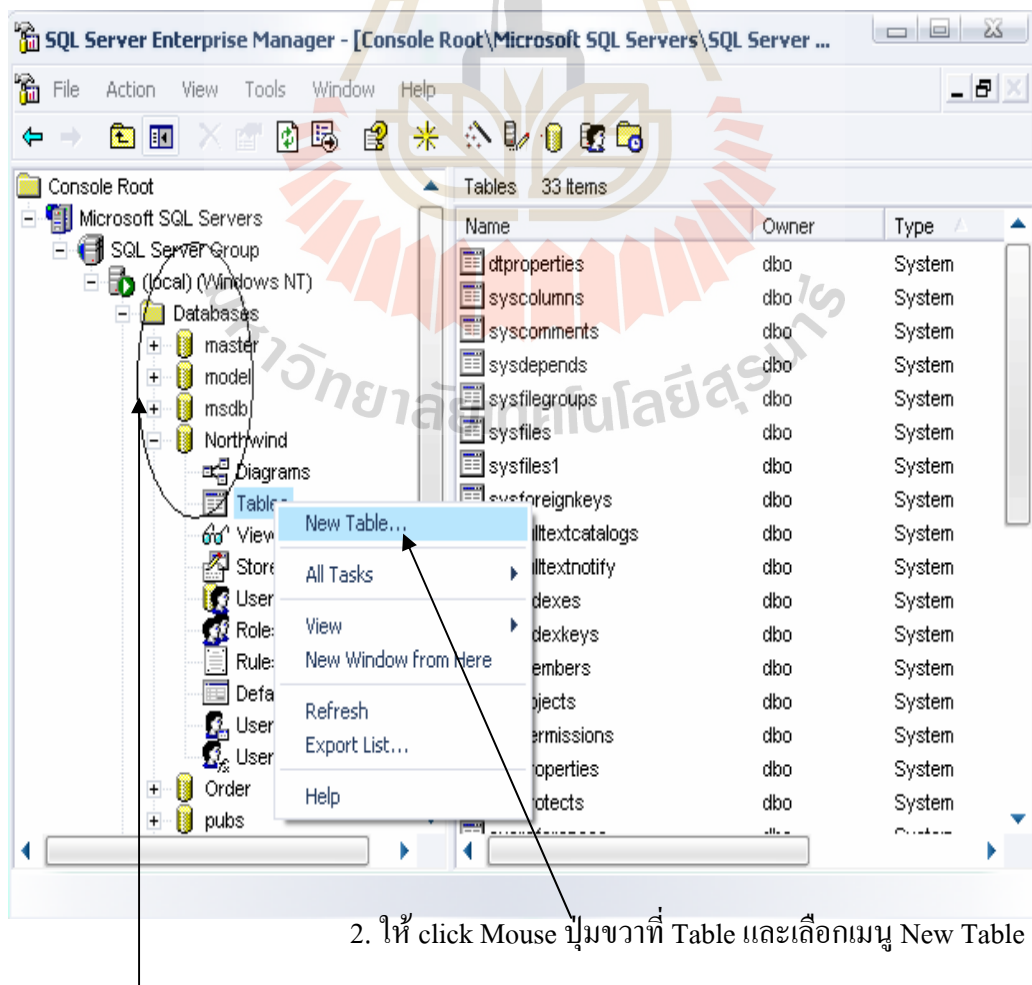
การสร้างตาราง

ตารางในระบบฐานข้อมูลเชิงสัมพันธ์เป็นส่วนที่เก็บข้อมูลในรูปแบบของแถว(ที่เราเรียกว่าเรคอร์ด) และคอลัมน์(ที่เราเรียกว่า ฟิวด์) ซึ่งใน SQL Server นั้นสามารถจะสร้างตารางได้มากถึง 2,000 ล้านตารางใน 1 ฐานข้อมูล ซึ่งงานหลักๆ ในการสร้างตารางจะเป็นการนิยามคุณสมบัติต่างๆ ของฟิวด์ การกำหนดชนิดข้อมูล การกำหนด Primary Key การกำหนด Foreign Key และอื่นๆ เช่น การสร้างอินเด็กซ์, การกำหนด Data Integrity เป็นต้น

การสร้างตารางด้วย SQL Enterprise Manager

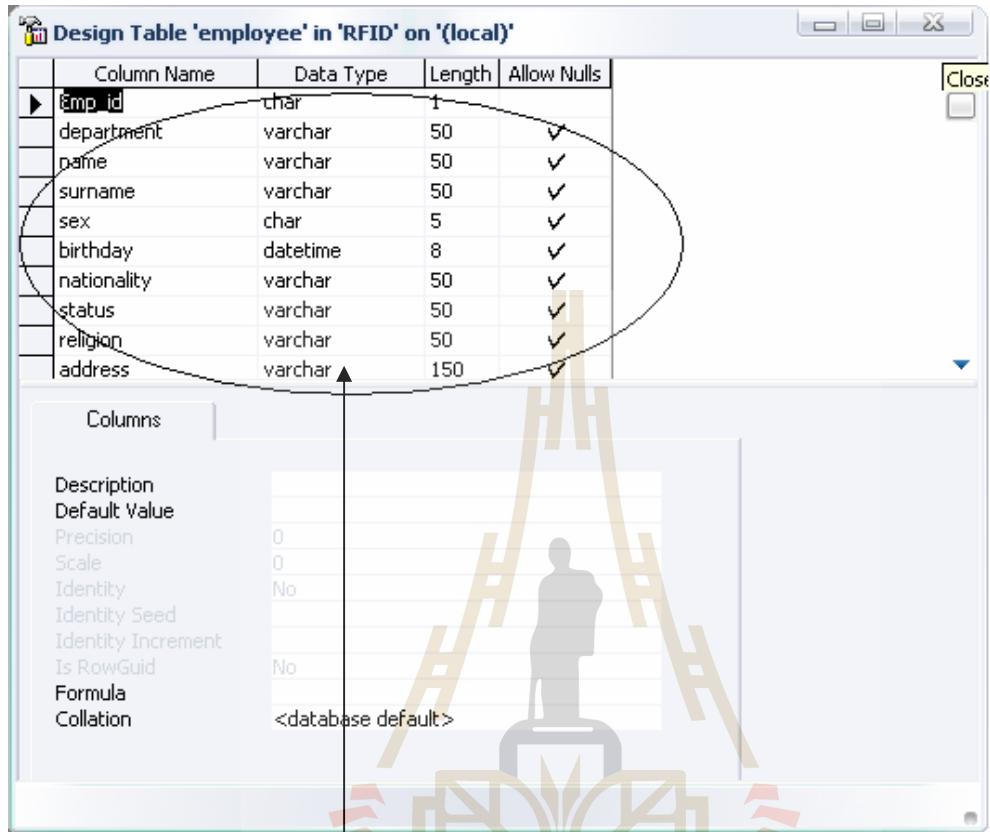
สามารถใช้ SQL Enterprise Manager ในการตารางได้ SQL Enterprise Manager นี้เป็นเครื่องมือที่ทำให้สามารถกำหนดสิ่งต่างๆ ได้สะดวกกว่าการสร้างตารางด้วยคำสั่ง SQL ที่จะกล่าวถึงต่อไป ซึ่งสามารถสร้างตารางได้ ตามขั้นตอนต่อไปนี้

1. ให้เลือกเซิร์ฟเวอร์ และเลือกฐานข้อมูลที่ต้องการจะสร้างฐานข้อมูล
2. ให้ Click Mouse ปุ่มขวาที่ Table และเลือกเมนู New table
3. จะปรากฏหน้าต่าง Table Design ให้กำหนดฟิลด์แต่ละฟิลด์, กำหนดชนิดข้อมูล, ขนาดข้อมูล, และอื่นๆ
4. กรอกข้อมูลลงในฐานข้อมูลตามที่ต้องการ



1. ให้เลือกเซิร์ฟเวอร์ และเลือกฐานข้อมูลที่ต้องการจะสร้างฐานข้อมูล

รูป(ก) แสดงการเลือกเซิร์ฟเวอร์ และเลือกฐานข้อมูลที่ต้องการจะสร้างฐานข้อมูล



3. จะปรากฏหน้าต่าง Table Design ให้กำหนดฟิลด์แต่ละฟิลด์, กำหนดชนิดข้อมูล, ขนาดข้อมูล และอื่นๆ

รูป(ข) แสดงการกำหนดฟิลด์แต่ละฟิลด์, กำหนดชนิดข้อมูล, ขนาดข้อมูล, และอื่นๆ

Data in Table 'employee' in 'RFID' on '(local)'

Emp_id	department	name	surname	sex	birthday	nationality	status	religion
1	วิศวกร	เฉลิมพงศ์	ชานานำวัยตะโก	ชาย	4/8/2526	ไทย	โสด	พุทธ
2	วิศวกร	ธรรเบรนท์	เจริญกัลโยภภาพ	ชาย	20/2/2527	ไทย	โสด	พุทธ
3	วิศวกร	วาณิช	โสภกา	ชาย	13/11/2526	ไทย	โสด	พุทธ
4	เลขา	ราตรี	รักงาน	หญิง	16/5/2527	ไทย	โสด	พุทธ

รูป(ค) แสดงการกรอกข้อมูลลงในฐานข้อมูล

Console Root\Microsoft SQL Servers\SQL Server Group\ (local) (Windows NT)\Databases\RFID\Tables

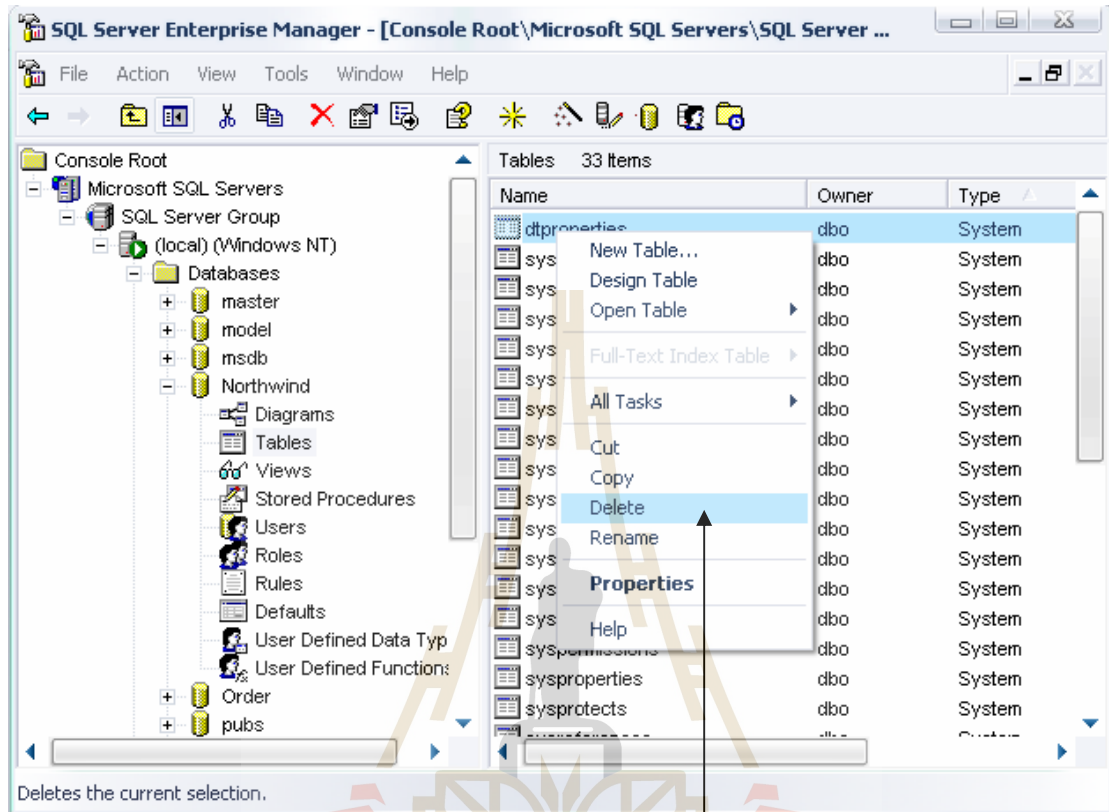
Name	Owner	Type	Create Date
dtproperties	dbo	System	27/2/2549 10:35:38
syscolumns	dbo	System	6/8/2543 1:29:12
syscomments	dbo	System	6/8/2543 1:29:12
sysdepends	dbo	System	6/8/2543 1:29:12
sysfilegroups	dbo	System	6/8/2543 1:29:12
sysfiles	dbo	System	6/8/2543 1:29:12
sysfiles1	dbo	System	6/8/2543 1:29:12
sysforeignkeys	dbo	System	6/8/2543 1:29:12
sysfulltextcatalogs	dbo	System	6/8/2543 1:29:12
sysfulltextnotify	dbo	System	6/8/2543 1:29:12
sysindexes	dbo	System	6/8/2543 1:29:12
sysindexkeys	dbo	System	6/8/2543 1:29:12
sysmembers	dbo	System	6/8/2543 1:29:12
sysobjects	dbo	System	6/8/2543 1:29:12
syspermissions	dbo	System	6/8/2543 1:29:12
sysproperties	dbo	System	6/8/2543 1:29:12
sysprotects	dbo	System	6/8/2543 1:29:12
sysreferences	dbo	System	6/8/2543 1:29:12
systypes	dbo	System	6/8/2543 1:29:12
sysusers	dbo	System	6/8/2543 1:29:12
employee	dbo	User	6/3/2549 12:10:54
timetable	dbo	User	6/3/2549 12:38:37

รูป(ง) แสดงฐานข้อมูลที่ได้ออกแบบเรียบร้อยแล้ว

รูปที่ 3.20 แสดงการสร้างตารางด้วย SQL Enterprise Manager

การลบตารางด้วย SQL Enterprise Manager

การลบตาราง จะเป็นการลบตารางซึ่งจะรวมทั้งข้อมูลในตาราง และสิ่งที่ได้กำหนดไว้ ตารางทิ้งไป สามารถลบตารางได้ด้วย SQL Enterprise Manager โดยให้ Click Mouse ปุ่มขวาเลือก ตารางที่เราต้องการลบ และเลือกเมนู Delete



Click Mouse ปุ่มขวาเลือกตารางที่เราต้องการลบ และเลือกเมนู

รูปที่ 3.21 แสดงการลบตารางด้วย SQL Enterprise Manager

การแก้ไขตารางที่สร้างไปแล้ว

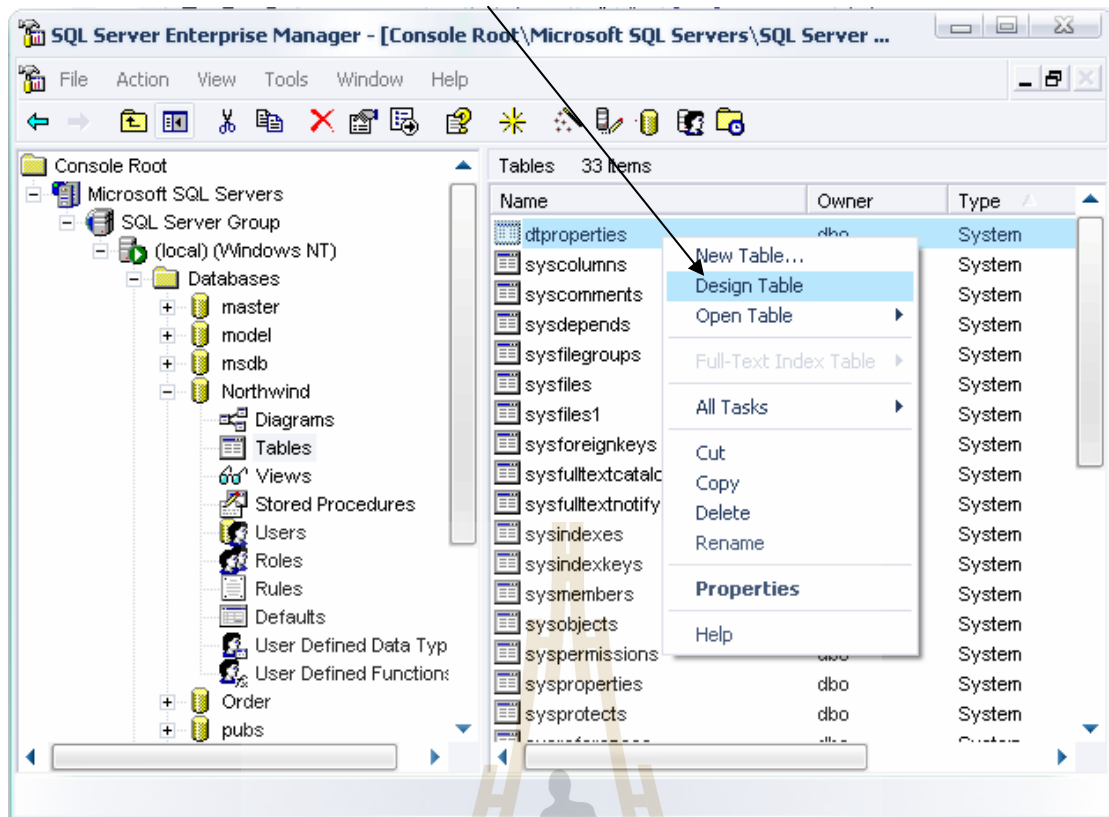
เราสามารถแก้ไขตารางที่ได้สร้างไปแล้ว เพื่อปรับเปลี่ยนให้เหมาะสมกับการใช้งานได้ โดยใช้ Enterprise Manager หรือคำสั่ง Transact-SQL ซึ่งจะได้อธิบายต่อไปนี้

การแก้ไขตารางด้วย Enterprise Manager

สำหรับตารางที่มีอยู่แล้ว สามารถเข้าหน้าต่าง Table Design เพื่อเปลี่ยนแปลงแก้ไข โครงสร้างของตารางได้โดยให้ทำขั้นตอนต่อไปนี้

ให้ Click Mouse ปุ่มขวาเลือกตารางที่ต้องการออกแบบใหม่ และเลือกเมนู Table Design

ให้ Click Mouse ปุ่มขวาเลือกตารางที่ต้องการแก้ไข และเลือกเมนู Table Design



รูปที่ 3.22 แสดงการแก้ไขตารางด้วย Enterprise Manager

3.6 การสร้างวงจรเครื่องรับ-เครื่องส่ง

3.5.1 วงจรเครื่องส่งประกอบไปด้วย

- ไอซีเบอร์ AT89C2051 ซึ่งมีหน้าที่เก็บข้อมูลและส่งข้อมูลแบบดิจิตอลจากขา Tx ไปยังขา ดาต้าอินพุต(Data In)ของวงจรตัวส่งสัญญาณคลื่นวิทยุ (LAIPAC เบอร์ TLP43)
- บอร์ด ET-AFP V1.0(R1) สำหรับบันทึกข้อมูลลงใน ไอซีเบอร์ AT89C2051
- วงจรตัวส่งสัญญาณคลื่นวิทยุ (LAIPAC เบอร์ TLP43) ที่จะเป็นตัวรับข้อมูลเข้ามาเพื่อทำการเข้ารหัสสัญญาณข้อมูล (Modulation) แบบ Amplitude - Shift Keying (ASK) และส่งข้อมูลในรูปสัญญาณวิทยุ ผ่านสายอากาศ(Antenna) ซึ่งทำงานที่ความถี่ 433.92 เมกะเฮิร์ตซ์ (MHz)
- Crystal Oscillator ความถี่ 18.432 เมกะเฮิร์ตซ์ (MHz) ทำหน้าที่สร้างสัญญาณนาฬิกาในการ กำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

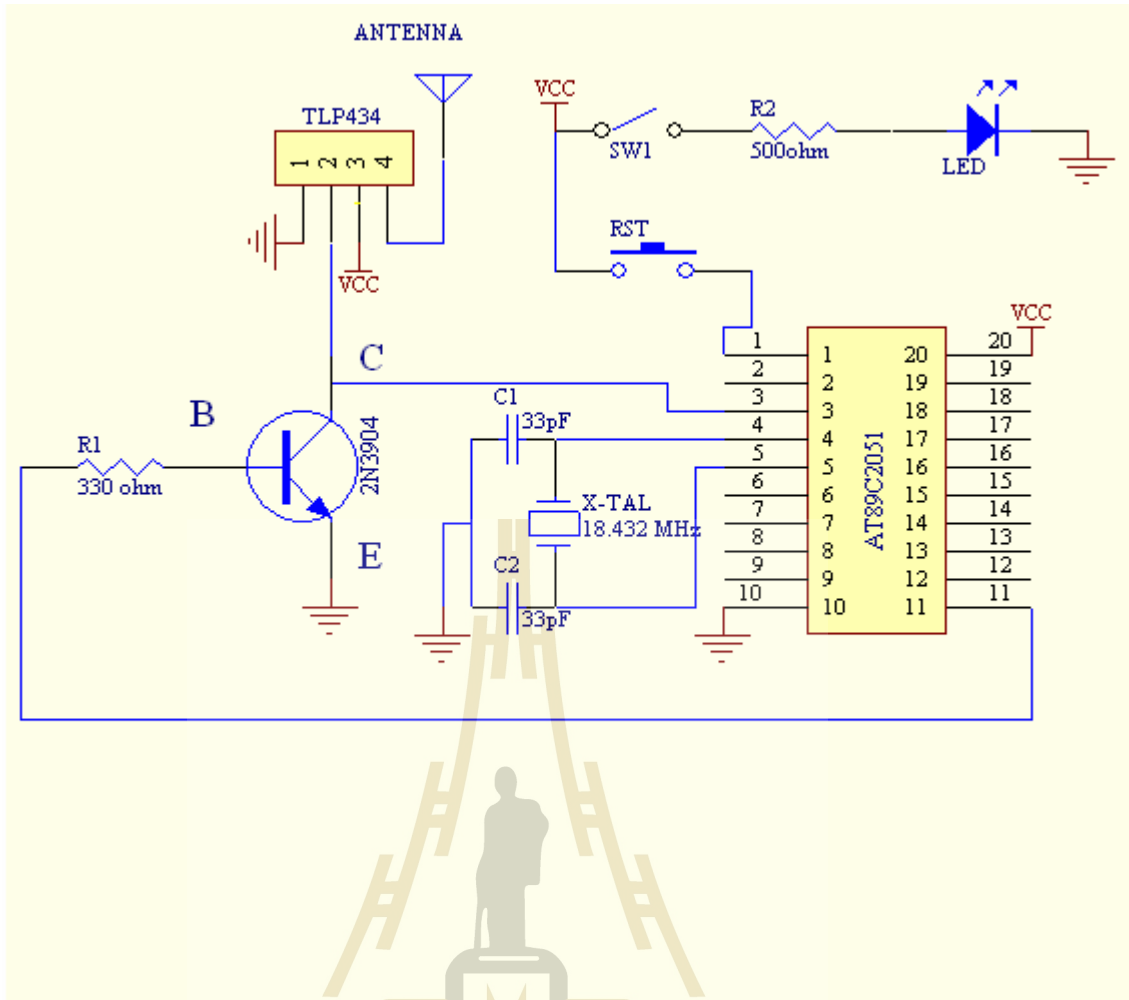
ทรานซิสเตอร์เบอร์ 2N3904 ทำหน้าที่เป็นตัวควบคุม

อธิบายการทำงานของวงจรเครื่องส่ง

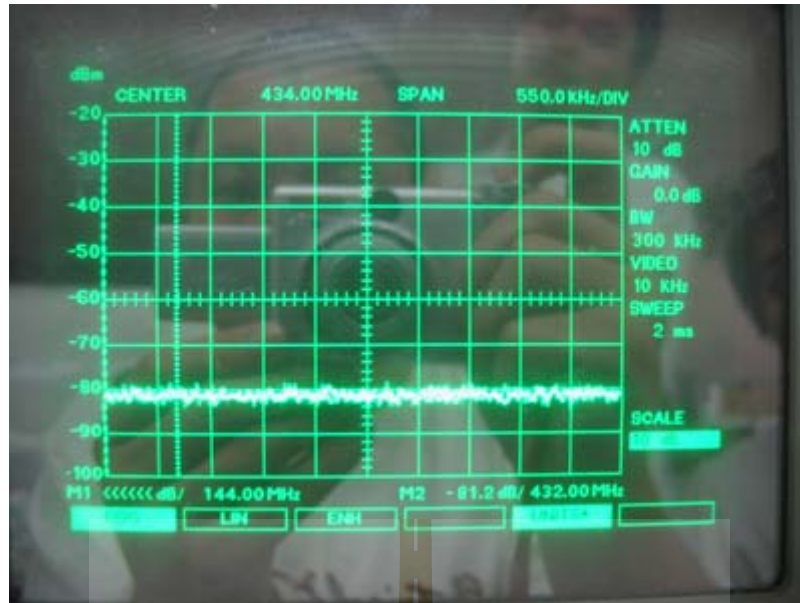
เมื่อเปิดสวิตช์จ่ายไฟจากแหล่งจ่ายไฟซึ่งเป็นถ่านนาฬิกาให้กับวงจรเครื่องส่ง และ จะทำให้วงจรเครื่องส่งเริ่มทำงาน โดยจะมีการแสดงสถานะของหลอดLEDซึ่งจะสว่างขึ้น เมื่อมีการจ่ายไฟให้กับวงจรและมีวงจรเครื่องส่งการทำงานดังนี้คือ ไมโครคอนโทรลเลอร์

เบอร์ AT89C2051 จะทำการส่งข้อมูลซึ่งข้อมูลนี้ได้ถูกโปรแกรมเข้าไปในไมโครคอนโทรลเลอร์ เบอร์ AT89C2051 โดยโปรแกรมที่เขียนไว้จะมีการทำงานเป็นขั้นตอนดังนี้คือ จะส่งข้อมูลเพื่อเริ่มการทำงานของวงจรับ-ส่ง จากนั้นจะส่งข้อมูลที่เป็นส่วนหัวของข้อมูลจริง และสุดท้ายเป็นข้อมูลจริง ซึ่งได้อธิบายคุณสมบัติไว้ในข้อ 3.1 เรียบร้อยแล้ว และข้อมูลทั้งหมดนี้จะถูกส่งออกมาผ่านขาที่ 3 (ขาTx) ของไมโครคอนโทรลเลอร์ เบอร์ AT89C2051ไปที่วงจรตัวส่งสัญญาณคลื่นวิทยุ (LAIPAC เบอร์ TLP43) ผ่านขาที่2 (Data In) ของตัวส่งสัญญาณคลื่นวิทยุ โดยข้อมูลจะถูกส่งออกอากาศโดยผ่านสายอากาศไปยังวงจรเครื่องรับสัญญาณ ข้อมูลที่ส่งผ่านอากาศไปยังวงจรเครื่องรับสัญญาณนี้ จะเป็นข้อมูล 8 บิต โดยจะถูกแยกให้ส่งทีละบิต และจะทำการรวมข้อมูลใหม่ให้เป็นข้อมูล 8 บิตอีกครั้งที่วงจรเครื่องรับสัญญาณ วงจรเครื่องส่งสัญญาณนี้จะทำการส่งข้อมูลอยู่ตลอดเวลาจนกระทั่งปิดสวิตช์ของวงจรวงจรเครื่องส่งจึงจะหยุดส่งข้อมูลและจะเริ่มส่งข้อมูลอีกครั้งเมื่อเปิดสวิตช์จ่ายไฟ จากวงจรเครื่องส่งจะพบว่า การทำงานของไมโครคอนโทรลเลอร์จะถูกกำหนดจังหวะการทำงานโดยวงจร Crystal Oscillator และวงจรถานซิสเตอร์เบอร์ 2N3904 จะควบคุมจังหวะในการส่งข้อมูลทั้งหมดของวงจรเครื่องส่ง

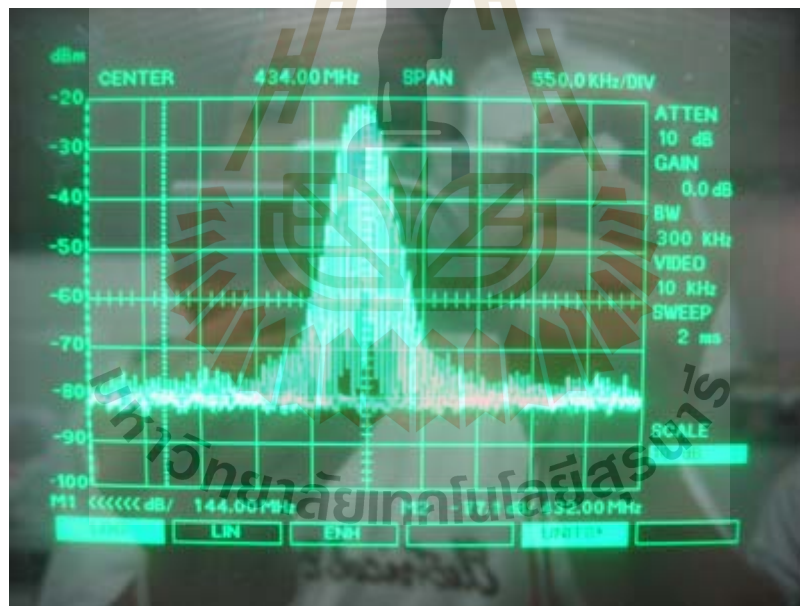




รูปที่ 3.23 แผนภาพชุดวงจรเครื่องส่ง



รูปที่ 3.24 รูปสัญญาณข้อมูลขณะที่ไม่มีการส่งสัญญาณจากวงจรเครื่องส่งไปยังวงจรเครื่องรับ



รูปที่ 3.25 รูปสัญญาณข้อมูลขณะมีการส่งสัญญาณจากวงจรเครื่องส่งไปยังวงจรเครื่องรับ

3.5.2 วงจรเครื่องรับประกอบไปด้วย

- ไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 ซึ่งมีหน้าที่เอาสัญญาณที่เข้ามาจากพอร์ต Serial แล้วทำการเลือกค่าลอจิก

- ทรานซิสเตอร์ เบอร์ 7805 เป็นส่วนประกอบของวงจรปรับแรงดัน (Voltage Regulator) เพื่อทำการเปลี่ยนระดับแรงดันจาก $\pm 15\text{ V}$ เป็นระดับของสัญญาณ TTL หรือ $\pm 5\text{ V}$
- ไอซี เบอร์ Max232 เป็นส่วนประกอบของวงจรที่มีหน้าที่เชื่อมต่อกันระหว่าง วงจรเครื่องรับและคอมพิวเตอร์ โดยวงจรมีหน้าที่ทำการส่งข้อมูลจริง 8 บิต ผ่านพอร์ต Serial เพื่อทำการแสดงผลที่จอมอนิเตอร์
- Crystal Oscillator ความถี่ 18.432 เมกะเฮิร์ตซ์ (MHz) ทำหน้าที่สร้างสัญญาณนาฬิกาในการ กำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์
- วงจรตัวรับสัญญาณคลื่นวิทยุ(LAIPACเบอร์RLP434A)ทำหน้าที่รับข้อมูลในรูปแบบสัญญาณวิทยุ ซึ่งทำงานที่ความถี่ 433.92 เมกะเฮิร์ตซ์ (MHz)

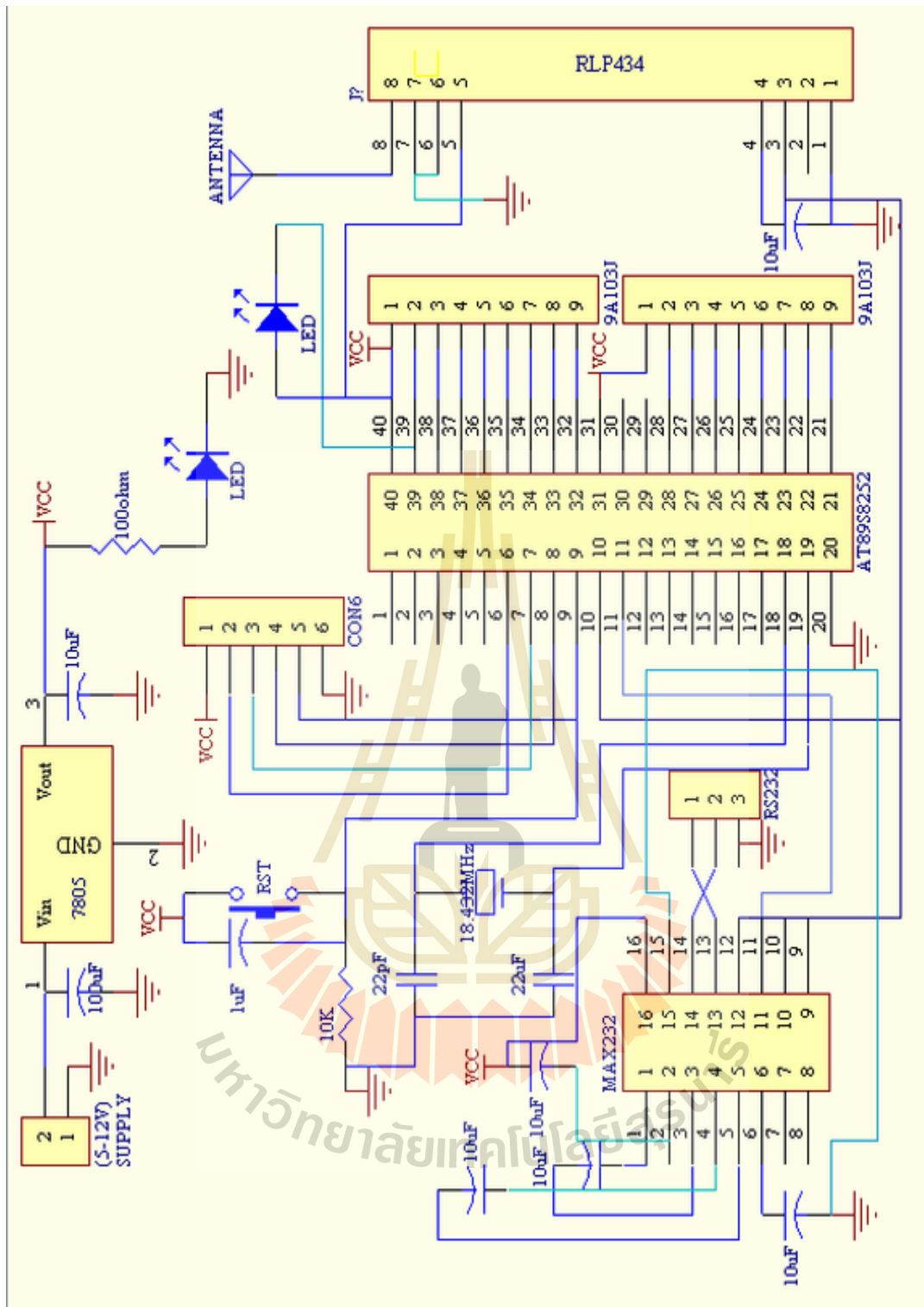
อธิบายการทำงานของวงจรเครื่องรับ

เมื่อเปิดสวิตช์จ่ายไฟจากแหล่งจ่ายไฟให้กับวงจรเครื่องส่ง และจะทำให้วงจรเครื่องรับเริ่มทำงาน โดยจะมีการแสดงสถานะของหลอดLEDซึ่งจะสว่างขึ้นเมื่อมีการจ่ายไฟให้กับวงจรและมีวงจรเครื่องรับการทำงานดังนี้คือ วงจรสัญญาณคลื่นวิทยุ (LAIPACเบอร์RLP434A)จะรับข้อมูลที่มาจากวงจรเครื่องส่งผ่านสายอากาศเข้ามาจากนั้นก็ส่งข้อมูลไปที่ขาที่ 2 ซึ่งเป็นขาที่ส่งข้อมูลออก(Digital Data Output)ของวงจรสัญญาณคลื่นวิทยุ และข้อมูลจะถูกส่งไปยังขาที่ 10(P3.0) ของไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 ซึ่งเป็นขาที่รับข้อมูลเข้า (Rx) ซึ่งข้อมูลที่ได้อาจจะเป็นข้อมูล 8 บิต ที่ถูกแยกให้ส่งทีละบิตมาจากวงจรเครื่องส่ง จากนั้นไมโครคอนโทรลเลอร์ จะทำการรวมข้อมูลใหม่ให้เป็นข้อมูล 8 บิตอีกครั้ง แล้วจึงทำการประมวลผลข้อมูลที่ได้มาตามคำสั่งของโปรแกรมที่ถูกเขียนไว้ โดยโปรแกรมที่เขียนไว้จะมีการทำงานเป็นขั้นตอนดังนี้คือ จะทำการตรวจสอบข้อมูลว่าเป็นส่วนหัวของข้อมูลจริงหรือไม่ ถ้าใช่ก็จะทำการรับข้อมูลจริงที่ถูกส่งมาแล้วนำไปแสดงผลที่จอมอนิเตอร์ แต่ถ้าไม่ใช่ก็จะรอรับข้อมูลใหม่อีกครั้งและทำการตรวจสอบหาส่วนหัวของข้อมูลจริง โดยวงจรเครื่องรับจะทำการรอรับสัญญาณของข้อมูลและตรวจสอบอยู่ตลอดเวลาจนกระทั่งปิดสวิตช์ของวงจรวงจรเครื่องรับจึงจะหยุดรอรับสัญญาณของข้อมูลและตรวจสอบข้อมูล และจะเริ่มทำงานอีกครั้งเมื่อเปิดสวิตช์จ่ายไฟ ซึ่งการตรวจสอบข้อมูลได้อธิบายอย่างละเอียดไว้ในข้อ 3.2 เรียบร้อยแล้วเมื่อได้ข้อมูลจริงที่ต้องการ ข้อมูลนี้ก็จะถูกส่งออกไปที่ขาที่ 11(P3.1) ของ ไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 ซึ่งเป็นขาส่งข้อมูลออก (Tx) ข้อมูลดังกล่าวจะถูกส่งผ่านพอร์ต Serial เพื่อทำการแสดงผลที่จอมอนิเตอร์

ส่วนวงจรอื่นๆจะมีหน้าที่ต่างกันได้แก่ วงจรปรับแรงดัน (Voltage Regulator) ที่มีทรานซิสเตอร์ เบอร์ 7805 เป็นส่วนประกอบนั้น จะทำหน้าที่เปลี่ยนระดับแรงดันจาก $\pm 15\text{ V}$ เป็นระดับของสัญญาณ TTL หรือ $\pm 5\text{ V}$ เพื่อใช้เป็นแหล่งจ่ายไฟให้แก่วงจร

และวงจรส่งข้อมูลไปแสดงผลนั้น จะมีไอซี เบอร์ Max232 เป็นส่วนประกอบ ซึ่งวงจรส่วนนี้จะเป็นส่วนที่เชื่อมต่อกันระหว่าง วงจรเครื่องรับและคอมพิวเตอร์ โดยวงจรนี้จะทำการส่งข้อมูลจริง 8 บิต ผ่านพอร์ต Serial เพื่อทำการแสดงผลที่จอมอนิเตอร์ ส่วนวงจร Crystal Oscillator และวงจรทรานซิสเตอร์เบอร์ 2N3904 จะควบคุมจังหวะในการรับส่งข้อมูลทั้งหมดของวงจรเครื่องรับ





รูปที่ 3.26 แผนภาพชุดวงจรเครื่องรับ

บทที่ 4

การทดลอง

4.1 การต่ออุปกรณ์

4.1.1. เครื่องส่งสัญญาณคลื่นวิทยุ

เมื่อทำการสร้างและต่อวงจรเครื่องส่งสัญญาณคลื่นวิทยุเรียบร้อยแล้ว จากนั้นทำการเขียนโปรแกรมลงในไมโครคอนโทรลเลอร์เบอร์ AT89C2051 เพื่อใช้ในการส่งข้อมูลแล้วนำมาประกอบเข้ากับวงจรเครื่องส่งสัญญาณคลื่นวิทยุเสร็จแล้วก็นำถ่านนาฬิกาไปใส่ในแปลงถ่านเพื่อใช้เป็นแหล่งจ่ายไฟของวงจรเครื่องส่ง เมื่อต้องการส่งสัญญาณก็สามารถกดเปิดสวิตซ์วงจรและกดสวิตซ์รีเซต 1 ครั้งเพื่อให้วงจรเริ่มการทำงานได้ ซึ่งข้อมูลที่ส่งออกมานั้นจะมีข้อมูล 3 ชนิดคือ ข้อมูลเริ่มการทำงานของวงจรรับสัญญาณ ส่วนหัวของข้อมูล และส่วนข้อมูลจริง เมื่อต้องการหยุดส่งข้อมูลเพียงแคंपิดสวิตซ์ วงจรเครื่องส่งสัญญาณคลื่นวิทยุก็จะหยุดส่งสัญญาณข้อมูลทันที ซึ่งจะเห็นว่าการใช้งานวงจรเครื่องส่งสัญญาณคลื่นวิทยุนั้นทางคณะผู้จัดทำได้ทำการออกแบบให้มีขนาดเล็กกะทัดรัด ง่ายต่อการพกพาอีกทั้งยังสะดวกต่อการใช้งานอีกด้วย



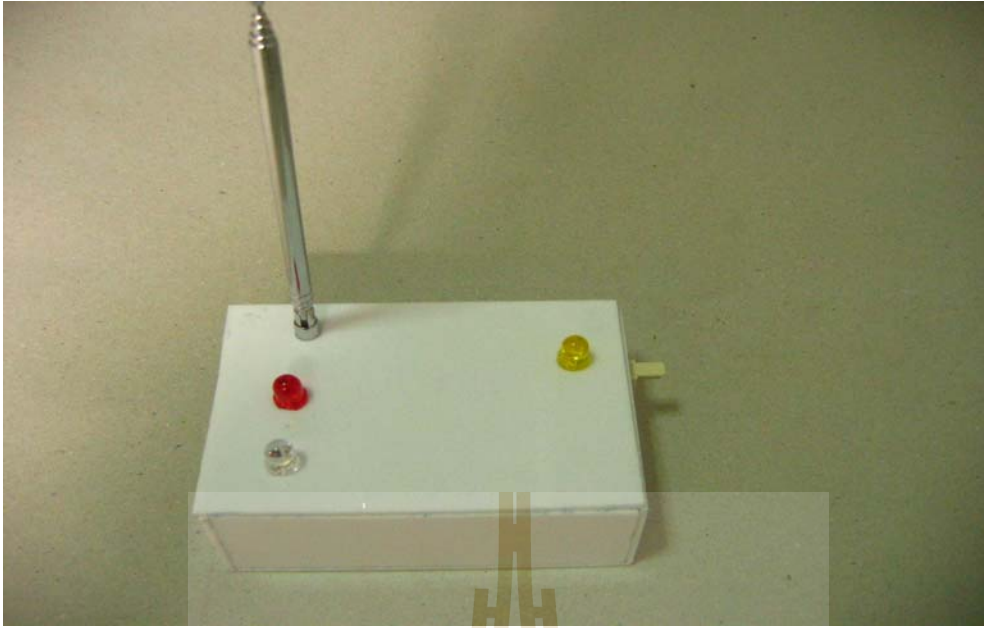
รูปที่ 4.1 วงจรเครื่องส่งสัญญาณคลื่นวิทยุ

4.1.2.เครื่องรับสัญญาณคลื่นวิทยุ

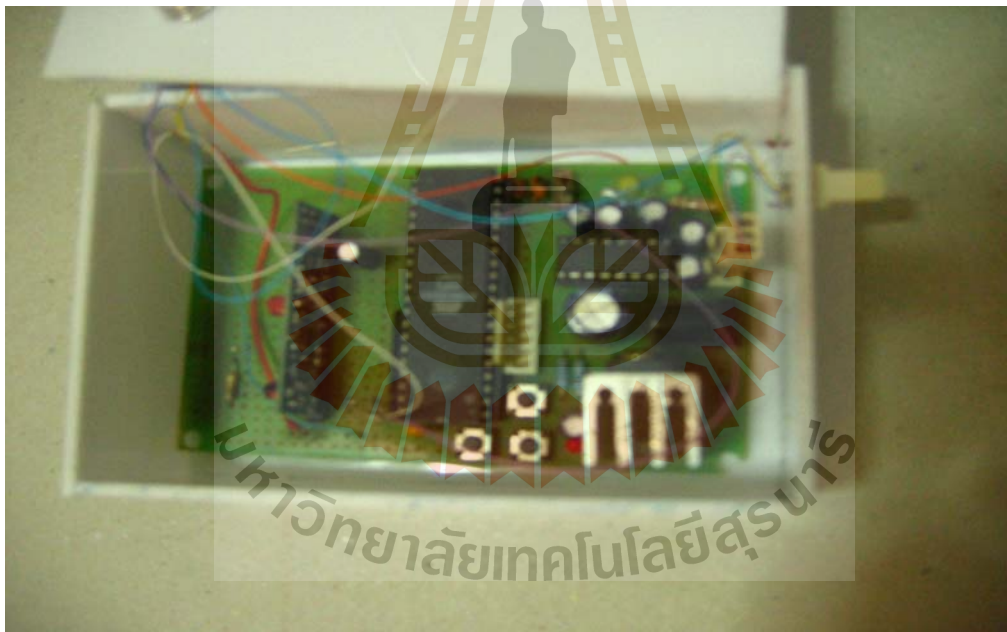
ในส่วนของวงจรเครื่องรับสัญญาณคลื่นวิทยุนั้นจะมีความยุ่งยากและซับซ้อนกว่าวงจรเครื่องส่งสัญญาณคลื่นวิทยุ เนื่องจากเป็นวงจรที่ใช้ในการรับข้อมูล ตรวจสอบข้อมูลและแสดงผล ซึ่งในส่วนแสดงผลนี้จะแสดงผลทั้งข้อมูลออกทางมอนิเตอร์โดยผ่านโปรแกรม Visual basic และการแสดงผลการเปิด-ปิดประตูโดยการแสดงผ่านสถานะของหลอดLED การต่ออุปกรณ์ของวงจรเครื่องรับอธิบายได้ดังนี้

1. เมื่อสร้างและต่อวงจรเครื่องรับเรียบร้อยแล้วก็ทำการเขียนโปรแกรมลงในไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 เพื่อใช้ในการรับข้อมูล ตรวจสอบข้อมูลและแสดงผล แล้วนำมาประกอบกับวงจรเครื่องรับสัญญาณคลื่นวิทยุ
2. ต่อวงจรเครื่องรับสัญญาณคลื่นวิทยุเข้ากับคอมพิวเตอร์เพื่อส่งข้อมูลออกมาแสดงผลทางจอมอนิเตอร์ ซึ่งการต่อนี้จะใช้พอร์ตอนุกรมของคอมพิวเตอร์โดยใช้สายซีเรียล(Serial) ต่อจากขา Rx และ Tx ของวงจรเครื่องรับเข้ากับพอร์ต Com 1 หรือ Com 2 ของเครื่องคอมพิวเตอร์แล้วแต่การใช้งาน โดยการต่อวงจรเครื่องรับสัญญาณคลื่นวิทยุเข้ากับคอมพิวเตอร์ได้แสดงไว้ดังรูปที่ 4.1.4
3. ต่อแหล่งจ่ายไฟเข้ากับวงจรเครื่องรับซึ่งทางคณะผู้จัดทำได้เลือกใช้หม้อแปลงที่จ่ายไฟขนาด 12 โวลต์ ให้แก่วงจร
4. กดสวิทช์รีเซ็ต(Reset) ของวงจรเครื่องรับ 1 ครั้ง เพื่อให้วงจรเริ่มทำการรรับข้อมูลใหม่และส่งข้อมูลที่ได้ไปแสดงผล
5. ทำการเปิดโปรแกรม Visual basic เพื่อดูข้อมูลของพนักงานผ่านจอมอนิเตอร์

จากการต่ออุปกรณ์ของวงจรเครื่องรับที่ได้อธิบายข้างต้นนี้เป็นเพียงการใช้วงจรเครื่องรับเพียงเครื่องเดียว ซึ่งโครงการนี้จะใช้วงจรเครื่องรับ 2 เครื่อง ดังนั้นทางคณะผู้จัดทำจึงเลือกใช้พอร์ตอนุกรมของคอมพิวเตอร์ทั้ง 2 พอร์ตคือพอร์ต Com 1 และพอร์ต Com 2 การต่ออุปกรณ์ของทั้งสองใช้วิธีการต่ออุปกรณ์เหมือนกันทุกประการแต่จะต่างกันตรงการเลือกใช้พอร์ตอนุกรมเท่านั้นและเพื่อให้ง่ายต่อการใช้งานทางคณะผู้จัดทำจึงเลือกใช้พอร์ต Com 1 สำหรับวงจรเครื่องรับเครื่องที่ 1 และติดตั้งไว้ที่ประตูที่ 1 ในทำนองเดียวกันใช้พอร์ต Com 2 สำหรับวงจรเครื่องรับเครื่องที่ 2 และติดตั้งไว้ที่ประตูที่ 2 เช่นกัน ส่วนการแสดงผลของข้อมูลนั้นจะถูกเขียนโปรแกรมกำหนดลำดับการแสดงผลของข้อมูลที่เข้ามาโดยใช้โปรแกรม Visual basic เนื่องจากข้อมูลที่เข้ามาทางพอร์ตอนุกรมทั้งสองพอร์ตนั้นเป็นข้อมูลที่ได้จากวงจรเครื่องรับคนละเครื่องกัน แต่จะถูกนำมาแสดงออกทางมอนิเตอร์เดียวกันนั่นเอง



รูปที่ 4.2 โครงสร้างภายนอกของวงจรเครื่องรับสัญญาณคลื่นวิทยุ



รูปที่ 4.3 โครงสร้างภายในของวงจรเครื่องรับสัญญาณคลื่นวิทยุ



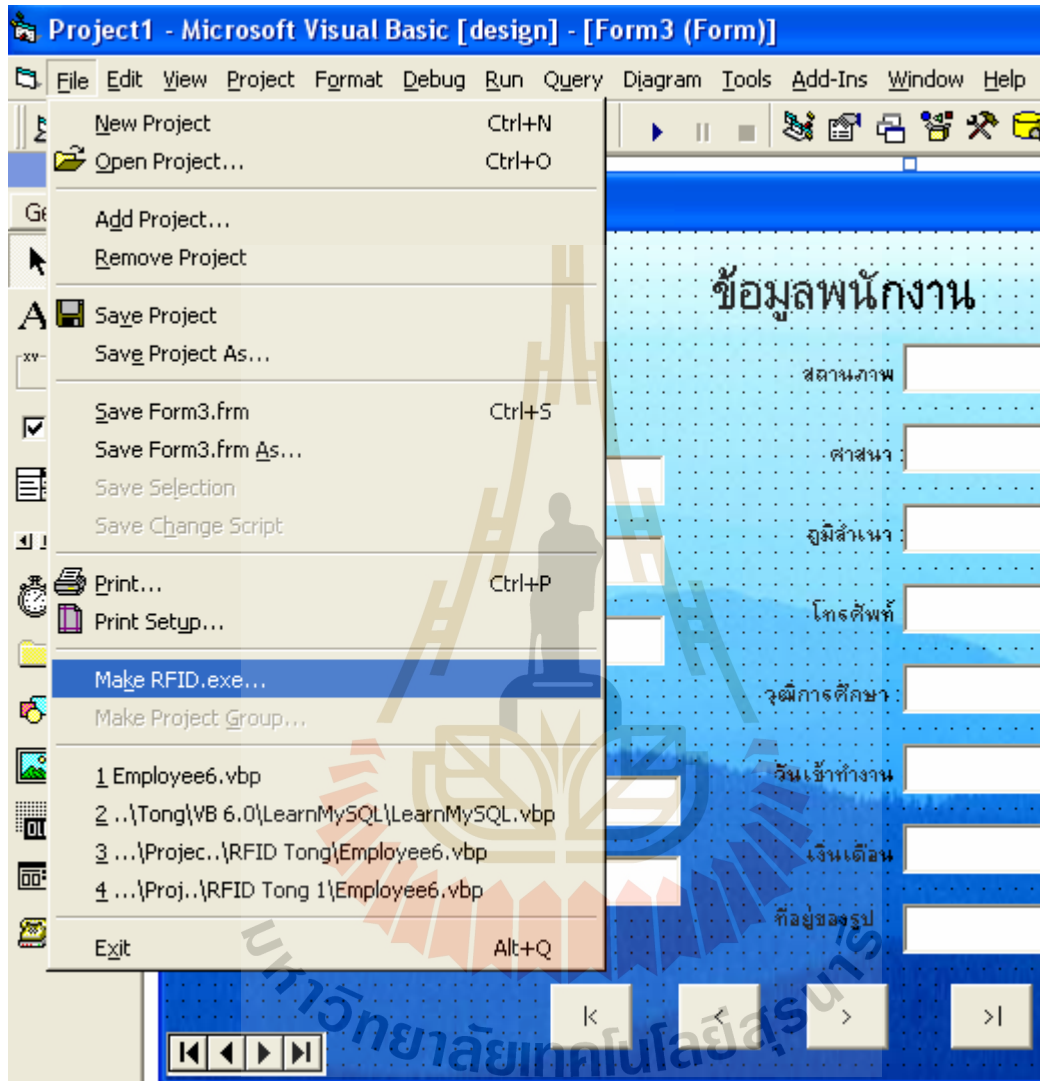
รูปที่ 4.4 การต่อพอร์ตอนุกรมของคอมพิวเตอร์กับวงจรเครื่องรับสัญญาณคลื่นวิทยุ



4.2 การใช้โปรแกรม Visual Basic เพื่อเรียกฐานข้อมูลมาแสดงบนจอมอนิเตอร์

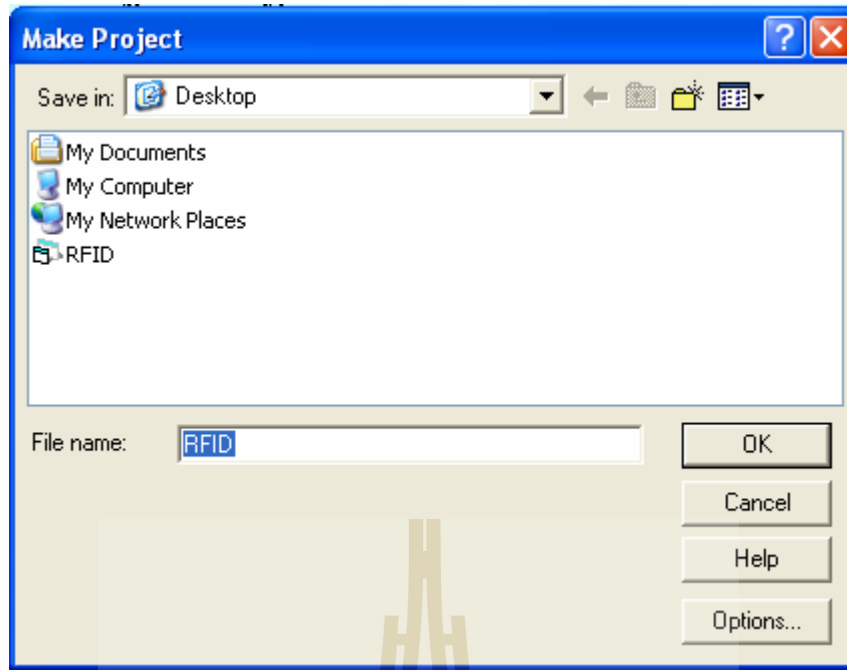
4.2.1 วิธีการใช้งาน

1. หลังจากออกแบบและสร้างฐานข้อมูลใน SQL server และเขียนโปรแกรม Visual Basic แล้วทำการสร้างโปรเจกต์บน Desktop เป็น file.exe ดังรูป



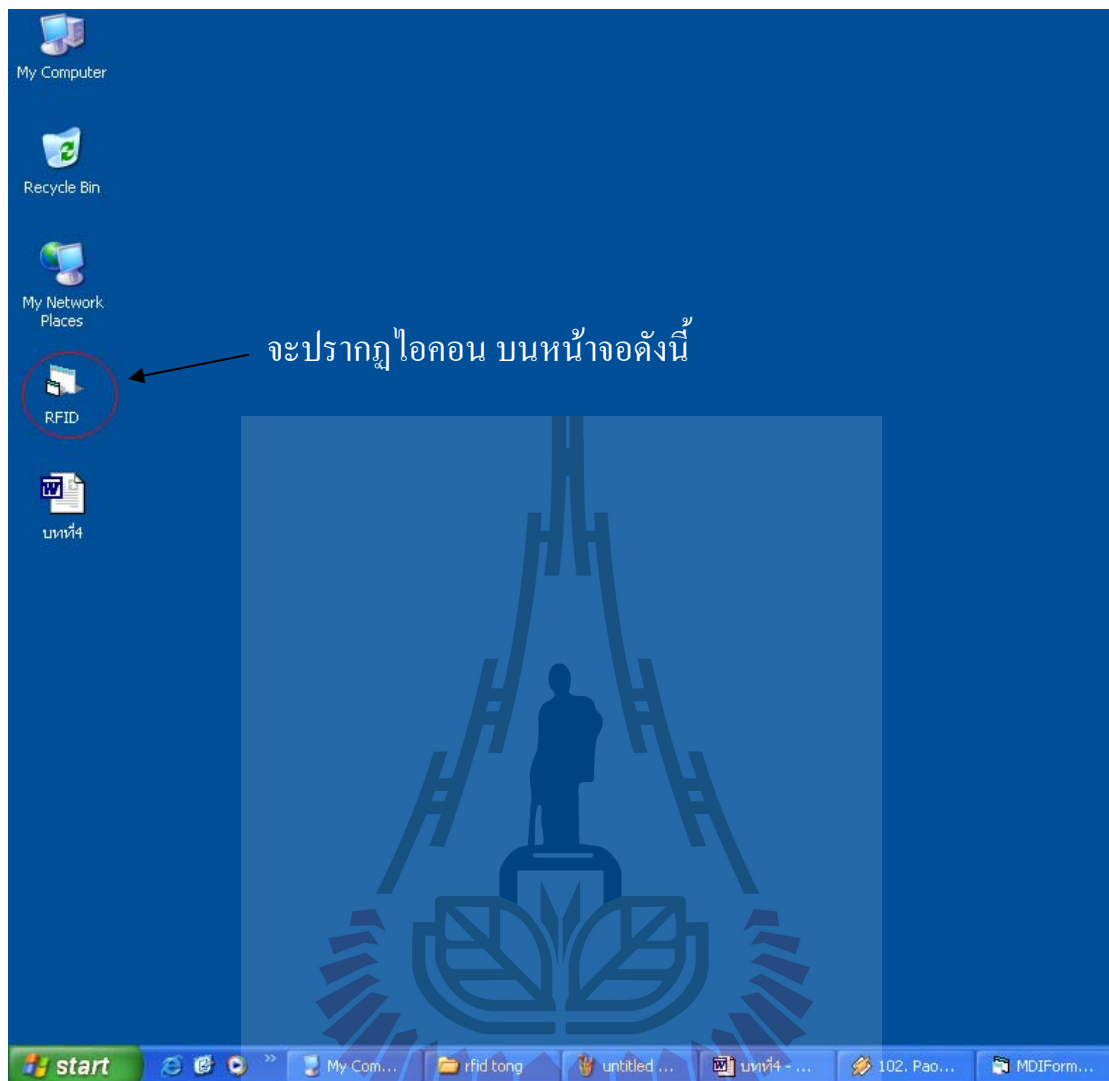
รูปที่ 4.5 การสร้างโปรเจกต์บน Desktop เป็น file.exe

2. จากนั้นจะปรากฏไดอะล็อกบ็อกซ์เพื่อให้กำหนดชื่อโปรเจกต์ที่สร้างขึ้น โดยตัวอย่างนี้จะกำหนดชื่อโปรเจกต์นี้ว่า RFID.exe ดังรูป



รูปที่ 4.6 กำหนดชื่อโปรเจกต์ที่สร้างขึ้น

3. จะปรากฏโปรเจกต์ บน Desktop ดังรูป



รูปที่4.7 ปรากฏโปรเจกต์ บน Desktop

4. เมื่อ Double Click ไอคอน RFID ปรากฏฟอร์มดังรูป

MDIForm1 - [Form1]
ข้อมูลพนักงาน การทำงาน ตำแหน่งงาน


รหัสพนักงาน :	1	สถานภาพ :	โสด	
แผนก :	วิศวกร	ศาสนา :	พุทธ	
ชื่อ :	เฉลิมพงศ์	ภูมิลำเนา :	11/5 ต.หนองกบ อ.บ้านโป่ง จ.ราช	
นามสกุล :	ชานานศิริยะระโก	โทรศัพท์ :	06-6521204	
เพศ :	ชาย	วุฒิการศึกษา :	ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม	ค้นหา ปิด
วันเดือนปีเกิด :	4/8/2526	วันเริ่มทำงาน :	7/7/2549	วันที่ 03/03/49
สัญชาติ :	ไทย	เงินเดือน :	25000	เวลา 14:49:33
เวลาเข้า :	12:12:10	เวลาออก :	12:20:12	

K < > >I

รูปที่ 4.8 หน้าต่างแสดงข้อมูลของพนักงาน

5. เมื่อมีพนักงานเข้ามาจะแสดงข้อมูลของพนักงานบนหน้าต่างแสดงข้อมูลของพนักงาน ดังรูป

The screenshot shows a software application window titled "MDIForm1 - [Form1]". The window has a menu bar with "ข้อมูลพนักงาน", "การหางาน", and "ค้นหาพนักงาน". The main area contains a form with the following fields:

รหัสพนักงาน :	2	สถานภาพ :	โสด		
แผนก :	วิศวกรรม	ศาสนา :	พุทธ		
ชื่อ :	ธรรมบรรพ์	วุฒิสำเร็จ :	121/5ต.วังเหนือ อ.วังเหนือ จ.ลำปาง		
นามสกุล :	เจริญภูมิโยภาพ	โทรศัพท์ :	09-2526548		
เพศ :	ชาย	วุฒิการศึกษา :	ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม	ค้นหา	ปิด
วันเดือนปีเกิด :	20/2/2526	วันเข้าทำงาน :	5/5/2548	วันที่ 03/03/49	
สัญชาติ :	ไทย	เงินเดือน :	26000	เวลา 14:26:29	
เวลาเข้า :	12:12:10	เวลาออก :	12:20:12		

At the bottom of the form, there are navigation buttons: "k", "<", ">", and ">|".

รูปที่ 4.9 แสดงข้อมูลของพนักงาน

4.2.2 รูปแบบการแสดงผล

ฟอร์มที่ 1 ข้อมูลพนักงาน

คลิกปุ่มข้อมูลพนักงาน

MDIForm1 - [Form3]

ข้อมูลพนักงาน การทำงาน ค้นหาพนักงาน

ข้อมูลพนักงาน

รหัสพนักงาน	2	สถานภาพ	โสด
แผนก	วิศวกร	ศาสนา	พุทธ
ชื่อ	ธรรมบรรพ์	วุฒิสาขา	121/5ต.รังเหนือ อ.รังเหนือ จ.ลำปาง
นามสกุล	เจริญกิจโยธภาพ	โทรศัพท์	09-2526548
เพศ	ชาย	วุฒิการศึกษา	ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม
วันเดือนปีเกิด	20/2/2526	วันเข้าทำงาน	5/5/2548
สัญชาติ	ไทย	เงินเดือน	26000
		ที่อยู่จริงรูป	C:\Documents and Settings\Tong\My

เพิ่มข้อมูลพนักงาน (ส่วนที่ 1)
ลบข้อมูลพนักงาน (ส่วนที่ 2)
แก้ไขข้อมูลพนักงาน (ส่วนที่ 3)

K < > >|

รูปที่ 4.10 แสดงการเรียกฟอร์มสำหรับเพิ่ม, ลบ และแก้ไขข้อมูลพนักงานเข้าไปในฐานข้อมูล

เมื่อต้องการเพิ่ม, ลบ, แก้ไขข้อมูลของพนักงานให้คลิกเมาส์ที่เมนูบาร์ “ข้อมูลพนักงาน” จะปรากฏฟอร์มเพื่อให้กรอกข้อมูลสำหรับเพิ่มข้อมูลพนักงาน หรือทำการค้นหาข้อมูลพนักงานเพื่อลบหรือแก้ไขข้อมูลพนักงาน

ส่วนที่1 ส่วนที่ใช้เพิ่มข้อมูลพนักงาน

The screenshot shows a web browser window with the title 'MDIForm1 - [Form3]'. The address bar shows 'ข้อมูลพนักงาน การทำงาน ค้นหาพนักงาน'. The main content area is titled 'ข้อมูลพนักงาน' and contains the following fields:

รหัสพนักงาน	5	สถานภาพ	
แผนก	บัญชี	ศาสนา	
ชื่อ	พรทศิ์	วุฒิสาขา	
นามสกุล		โทรศัพท์	
เพศ		วุฒิการศึกษา	
วันเดือนปีเกิด		วันเข้าทำงาน	
สัญชาติ		เงินเดือน	
		ที่อยู่จริงรูป	

On the right side, there is a photo upload section with a photo of a person and buttons for 'ยกเลิก' (Cancel) and 'ตกลง' (OK). Below the photo are two buttons: 'ลบข้อมูลพนักงาน' (Delete Employee Information) and 'แก้ไขข้อมูลพนักงาน' (Edit Employee Information). At the bottom of the form, there are navigation buttons: 'K', '<', '>', and '>|'.


รูปที่ 4.11 แสดงการเพิ่มข้อมูลพนักงานในฐานข้อมูล

เราสามารถกรอกข้อมูลของพนักงานใหม่ทั้งหมดได้เลย เมื่อกรอกครบแล้วให้คลิกเมาส์ที่ปุ่ม “เพิ่มข้อมูลพนักงาน” จะปรากฏปุ่มเพื่อยืนยันการเพิ่มข้อมูล โดยที่ข้อมูลของพนักงานทั้งหมดจะถูกนำไปเก็บไว้ในฐานข้อมูลของบริษัท

ส่วนที่ 2 ส่วนที่ใช้ลบข้อมูลพนักงาน

MDIForm1 - [Form3]
ข้อมูลพนักงาน การทำงาน ค้นหาพนักงาน

ข้อมูลพนักงาน

รหัสพนักงาน	2	สถานภาพ	โสด	
แผนก	วิศวกร	ศาสนา	พุทธ	
ชื่อ	ธรรมบรรพ์	ภูมิลำเนา	121/5 ต.วังเหนือ อ.วังเหนือ จ.ลำปาง	
นามสกุล	เจริญกิจโยธภาพ	โทรศัพท์	09-2526548	
เพศ	ชาย	วุฒิการศึกษา	ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม	
วันเดือนปีเกิด	20/2/2526	วันเข้าทำงาน	5/5/2548	
สัญชาติ	ไทย	เงินเดือน	26000	
		ที่อยู่จริงรูป	C:\Documents and Settings\Tong\My	

เพิ่มข้อมูลพนักงาน
ยืนยัน
ตกลง ยกเลิก
แก้ไขข้อมูลพนักงาน

K < > >|


รูปที่ 4.12 แสดงการลบข้อมูลพนักงานในฐานข้อมูล

เราสามารถลบข้อมูลของพนักงานได้โดยการคลิกที่ปุ่ม  เพื่อค้นหาข้อมูลพนักงานที่ต้องการลบ เมื่อพบแล้วจากนั้นให้คลิกที่ปุ่ม “ลบข้อมูลพนักงาน” จะปรากฏปุ่มเพื่อยืนยันการลบข้อมูลพนักงาน ทำให้ข้อมูลของพนักงานในฐานข้อมูลจะถูกลบออกจากฐานข้อมูล

ส่วนที่3 ส่วนที่ใช้แก้ไขข้อมูลพนักงาน

MDIForm1 - [Form3]
ข้อมูลพนักงาน การทำงาน ค้นหาพนักงาน

ข้อมูลพนักงาน

รหัสพนักงาน	1	สถานภาพ	โสด	
แผนก	วิศวกรรม	ศาสนา	พุทธ	
ชื่อ	เฉลิมพงศ์	ภูมิลำเนา	11/5 ต.หนองกบ อ.บ้านโป่ง จ.ราชบุรี	
นามสกุล	ชาวหนาหัวตะกอก	โทรศัพท์	06-6521204	
เพศ	ชาย	วุฒิการศึกษา	ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม	เพิ่มข้อมูลพนักงาน
วันเดือนปีเกิด	4/8/2526	วันเข้าทำงาน	7/7/2549	ลบข้อมูลพนักงาน
สัญชาติ	ไทย	เงินเดือน	25000	ยืนยัน
		ที่อยู่จริงรูป	C:\Documents and Settings\Tong\My	ตกลง ยกเลิก

Navigation: |< < > >|

รูปที่ 4.13 แสดงการแก้ไขข้อมูลพนักงานในฐานข้อมูล

เราสามารถลบข้อมูลของพนักงานได้โดยการคลิกที่ปุ่ม  เพื่อค้นหาข้อมูลพนักงานที่ต้องการแก้ไข เมื่อพบแล้วจากนั้นให้คลิกที่ปุ่ม “แก้ไขข้อมูลพนักงาน” จะปรากฏปุ่มเพื่อยืนยันการแก้ไขข้อมูลพนักงาน ทำให้ข้อมูลของพนักงานในฐานข้อมูลจะถูกแก้ไข



ฟอร์มที่ 2 การทำงานของพนักงาน

คลิกปุ่มการทำงาน

รหัสพนักงาน	วันที่เข้า	เวลาที่เข้า	วันที่ออก	เวลาที่ออก
2	26/2/49	13:12:15	26/2/49	13:12:45

คุณเข้าทั้งหมด
1 ครั้ง

คุณออกทั้งหมด
1 ครั้ง


รูปที่ 4.14 แสดงการตรวจสอบเวลาเข้า-ออกของพนักงาน

เราสามารถตรวจสอบเวลาเข้า-ออกของพนักงานได้โดยการคลิกเมาส์ที่เมนูบาร์ “การทำงาน” จากนั้นจะปรากฏฟอร์มที่แสดงเวลาเข้า-ออกของพนักงาน ถ้าต้องการที่จะตรวจสอบวันและเวลาเข้า-ออกของพนักงานคนใด ให้กรอกรหัสพนักงานและกรอกรวันที่ที่ต้องตรวจสอบแล้วคลิกเมาส์ที่ปุ่ม “ค้นหา” จากนั้นจะปรากฏเวลาเข้า-ออกของพนักงานที่ต้องการค้นหาตามวันที่ที่ถูกกำหนดและจำนวนครั้งที่พนักงานคนนั้นเข้า-ออก

ฟอร์มที่ 3 ค้นหาพนักงานของบริษัท

คลิกปุ่มค้นหาพนักงาน

The screenshot shows a web application window titled "MDIForm1 - [Form1]". The window has a menu bar with "ข้อมูลพนักงาน", "การทำงาน", and "ค้นหาพนักงาน". The "ค้นหาพนักงาน" menu item is highlighted, and an arrow points to it with the text "คลิกปุ่มค้นหาพนักงาน". The form contains the following fields:

รหัสพนักงาน :	2	สถานภาพ :	โสด		
แผนก :	วิศวกร	ศาสนา :	พุทธ		
ชื่อ :	ธรรมบรรพ์	ผู้มีสำเนา :	121/5ต.วังเหนือ อ.วังเหนือ จ.ลำปาง		
นามสกุล :	เจริญวิมลโยภภาพ	โทรศัพท์ :	09-2526548		
เพศ :	ชาย	วุฒิการศึกษา :	ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม	ค้นหา	ปิด
วันเดือนปีเกิด :	20/2/2526	วันเข้าทำงาน :	5/5/2548	วันที่	03/03/49
สัญชาติ :	ไทย	เงินเดือน :	26000	เวลา	14:26:29
เวลาเข้า :	12:12:10	เวลาออก :	12:20:12		

At the bottom of the form, there are navigation buttons: "K", "<", ">", and ">|".

รูปที่ 4.15 แสดงข้อมูลของพนักงานเมื่อมีพนักงานเข้า-ออกที่ทำงาน

เมื่อมีพนักงานคนใดเข้า-ออกประตูนี้ก็จะแสดงข้อมูลของพนักงานคนนั้นบนจอมอนิเตอร์ โดยที่ข้อมูลของเวลาเข้า-ออกของพนักงานคนนั้นจะถูกเก็บที่ฐานข้อมูลเพื่อให้สามารถตรวจสอบเวลาเข้า-ออกได้และสามารถคำนวณจำนวนครั้งในการเข้า-ออกได้ นอกจากนี้ยังสามารถค้นหาพนักงานในวันนั้นได้อีกด้วย(ในเวลานั้นต้องไม่มีพนักงานคนใดเข้า-ออก) โดยการคลิกเมาส์ที่ปุ่ม “ค้นหา” จากนั้นจะปรากฏ Message Box เพื่อให้กรอกรหัสพนักงาน จะปรากฏข้อมูลของพนักงานที่ต้องการค้นหาในวันนั้น

4.3 ผลการทดลอง

หลังจากทำการต่ออุปกรณ์เครื่องรับ-ส่งสัญญาณวิทยุและติดตั้งโปรแกรม Visual basic เรียบร้อยแล้ว เมื่อทดลองส่งสัญญาณจากเครื่องส่ง 1 เครื่อง พบว่าเมื่อเครื่องรับรับสัญญาณได้ทำให้หลอดไฟสีขาวและสีเหลืองจะกระพริบ(แสดงสถานะประตูเปิด) ดังรูปที่ 4.3.1 จนกระทั่งเครื่องรับรับสัญญาณไม่ได้ เมื่อเครื่องหยุดส่งสัญญาณจะทำให้หลอดไฟสีแดงจะสว่างขึ้น(แสดงสถานะประตูปิด) ดังรูปที่ 4.3.2 ซึ่งหลอดไฟสีแดงนี้จะติดอยู่ตลอดเวลาจนกระทั่งเครื่องรับเริ่มรับสัญญาณได้อีกครั้ง การทำงานของเครื่องรับจะทำงานวนรอบเช่นนี้ตลอดเวลาและในการทำงานของระบบนี้เครื่องรับจะใช้เวลาในการตรวจสอบข้อมูลเล็กน้อยก่อนที่จะแสดงผลออกมาทางมอนิเตอร์โดยใช้โปรแกรม Visual basic ดังรูป 4.16 รูปที่ 4.17 และรูปที่ 4.18

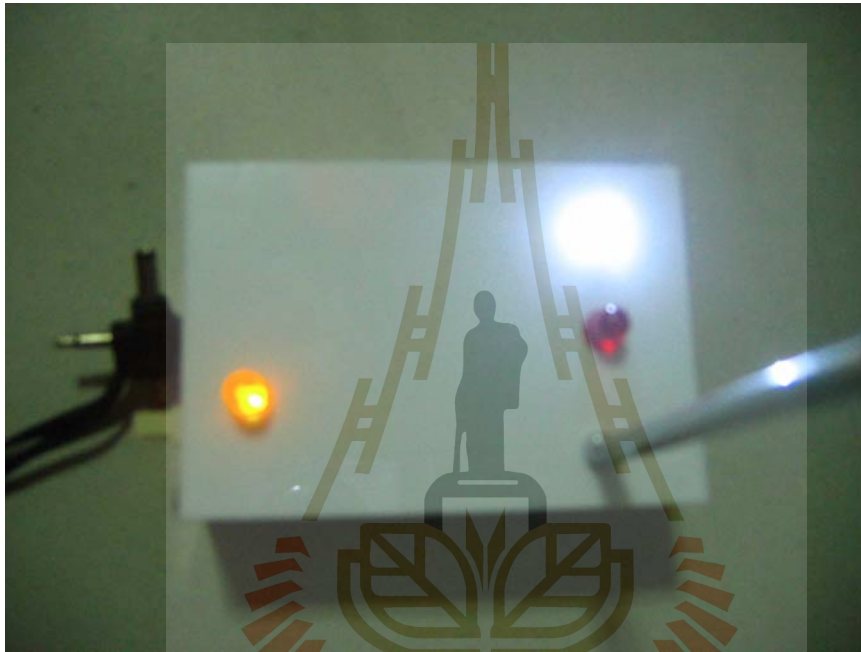
จากนั้นทดลองส่งสัญญาณจากเครื่องส่ง 2 เครื่องพร้อมกัน พบว่าเครื่องรับจะใช้เวลาในการตรวจสอบข้อมูลมากขึ้น และใช้เวลาเพื่อที่จะแสดงผลออกมาทางมอนิเตอร์มากขึ้นและพบว่าข้อมูลที่แสดงนั้นจะเปลี่ยนแปลงบนหน้าต่างแสดงผลตามสัญญาณที่รับเข้ามาได้กล่าวคือเมื่อรับสัญญาณจากตัวส่งตัวที่ 1 ได้ก็จะแสดงข้อมูลของพนักงานคนที่ 1 และเมื่อรับสัญญาณจากตัวส่งตัวที่ 2 ได้ก็จะแสดงข้อมูลของพนักงานคนที่ 2 ในการแสดงผลนั้นจะมีการเปลี่ยนแปลงเฉพาะข้อมูลของแต่ละคนเท่านั้นและจะมีบางช่วงที่ไม่แสดงข้อมูลที่เป็นเช่นนี้เนื่องจากเครื่องรับไม่สามารถรับสัญญาณได้ เช่นเดียวกันเมื่อทดลองส่งสัญญาณจากเครื่องส่ง 3 เครื่องพร้อมกัน พบว่าเครื่องรับจะใช้เวลาในการตรวจสอบข้อมูลมากขึ้นและการแสดงผลจะเหมือนการทดลองใช้เครื่องส่ง 2 เครื่องพร้อมกัน แต่ข้อมูลจะเปลี่ยนแปลงบนหน้าต่างแสดงผลช้าลง ซึ่งการแสดงผลนี้จะขึ้นอยู่กับเครื่องรับที่จะสามารถรับข้อมูลจากเครื่องส่งเครื่องใดก่อนและรับข้อมูลได้ในช่วงเวลาใด และจากการทดลองพบว่าเครื่องรับจะรับสัญญาณจากเครื่องส่งที่อยู่ใกล้และมีความแรงของสัญญาณสูงได้ดีกว่าเครื่องส่งที่อยู่ไกลและมีความแรงของสัญญาณต่ำ

จากนั้นทดลองส่งสัญญาณจากเครื่องส่ง 2 เครื่องและใช้เครื่องรับ 2 เครื่องโดยให้เครื่องส่ง 1 เครื่องส่งสัญญาณให้เครื่องรับ 1 เครื่องและเครื่องรับ-ส่งทั้งสองชุดนี้อยู่ในบริเวณที่ห่างกันประมาณ 12 เมตรซึ่งจะทำให้เครื่องส่งแต่ละเครื่องจะไม่รบกวนกันและเพื่อให้เข้าใจได้ง่าย คณะผู้จัดทำจึงเลือกใช้พอร์ต Com 1 สำหรับวงจรเครื่องรับเครื่องที่ 1 และติดตั้งไว้ที่ประตูที่ 1 ซึ่งประตูนี้จะอยู่ใกล้กับเครื่องคอมพิวเตอร์ และใช้พอร์ต Com 2 สำหรับวงจรเครื่องรับเครื่องที่ 2 และติดตั้งไว้ที่ประตูที่ 2 ซึ่งประตูนี้จะอยู่ไกลจากเครื่องคอมพิวเตอร์และประตูแรกประมาณ 12 เมตร จากการทดลองพบว่าผลที่ได้นั้นไม่ต่างกันคือ มอนิเตอร์จะแสดงผลข้อมูลของเครื่องส่งที่ 1 จากเครื่องรับที่ 1 สลับกันกับแสดงผลข้อมูลของเครื่องส่งที่ 2 จากเครื่องรับที่ 2 แต่ข้อมูลจากเครื่องรับที่ 2 นั้นจะแสดงได้ช้ากว่าเล็กน้อย

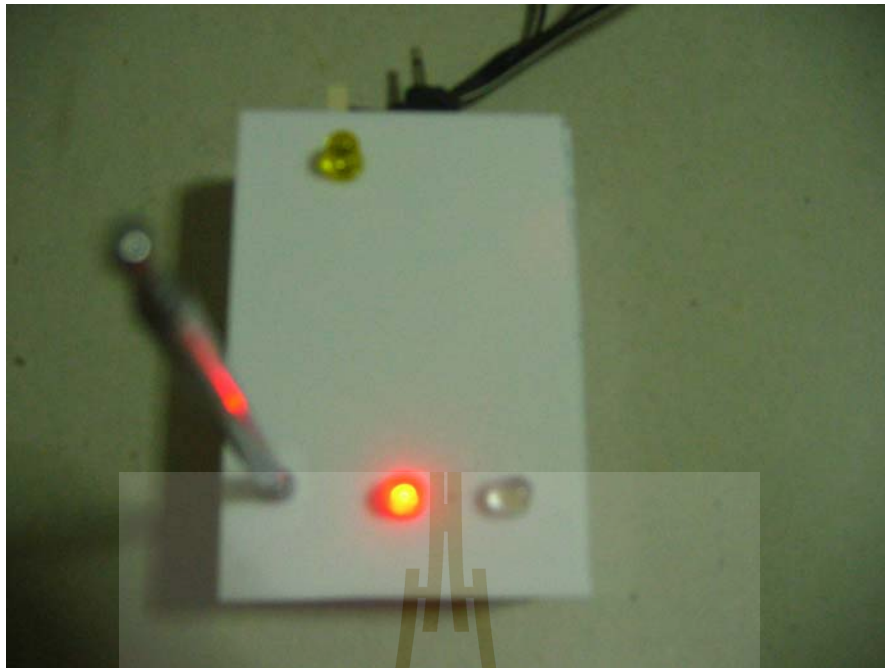
จากผลที่แสดงบนจอมอนิเตอร์โดยโปรแกรม Visual basic นั้น พบว่า หน้าจอที่แสดงวันและเวลาเข้า-ออกของพนักงานในรูปที่ 4.3.4 จะเก็บเวลาเข้า-ออกของพนักงาน รวมทั้งนับจำนวน

ครั้งที่เข้าและออกของพนักงาน โดยจำแนกตามรหัสพนักงานและวันที่ ข้อมูลการเข้า-ออกของพนักงานทุกคนจะถูกเก็บไว้ในฐานข้อมูลทั้งหมด และใช้โปรแกรม Visual basic เรียกฐานข้อมูลดังกล่าวมาแสดงบนจอมอนิเตอร์

ส่วนรูป 4.3.5 จะเป็นการแสดงการค้นหาข้อมูลของพนักงานแต่ละคน โดยใช้รหัสพนักงานเป็นตัวชี้ เมื่อกรอกรหัสพนักงานที่ต้องการค้นหา โปรแกรม โปรแกรม Visual basic จะทำการเรียกฐานข้อมูลของรหัสพนักงานคนดังกล่าวมาแสดงบนจอมอนิเตอร์เช่นกัน



มหาวิทยาลัยเทคโนโลยีสุรนารี
รูปที่ 4.16 แสดงสถานะประตูเปิด



รูปที่ 4.17 แสดงสถานะประตูปิด

MDIForm1 - [Form3]

ข้อมูลพนักงาน การทำงาน คำนวณพนักงาน

ข้อมูลพนักงาน

รหัสพนักงาน	2	สถานภาพ	โสด
แผนก	วิศวกร	ศาสนา	พุทธ
ชื่อ	ธรรมบรรพ์	วุฒิสาขา	121/5 ต.วังเหนือ อ.วังเหนือ จ.ลำปาง
นามสกุล	เจริญภูมิโยภภาพ	โทรศัพท์	09-2526548
เพศ	ชาย	วุฒิการศึกษา	ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม
วันเดือนปีเกิด	20/2/2526	วันเข้าทำงาน	5/5/2548
สัญชาติ	ไทย	เงินเดือน	26000
		ที่อยู่จริง	C:\Documents and Settings\Tong\My

เพิ่มข้อมูลพนักงาน
ลบข้อมูลพนักงาน
แก้ไขข้อมูลพนักงาน

K < > |

รูปที่ 4.18 แสดงข้อมูลพนักงานบนมอนิเตอร์

MDIForm1 - [ประวัติการเข้า-ออกงาน]

ข้อมูลพนักงาน การทำงาน คำนวณพนักงาน

รหัสพนักงาน: 2

วันที่: 26/2/49 เวลา: 13:12:15

รหัสพนักงาน	วันที่เข้า	เวลาที่เข้า	วันที่ออก	เวลาที่ออก
2	26/2/49	13:12:15	26/2/49	13:12:45

ปุ่ม: ค้นหา, ปิด

รวมเข้าทั้งหมด: 1 ครั้ง
รวมออกทั้งหมด: 1 ครั้ง

รูปที่ 4.19 แสดงเวลาเข้า-ออกของพนักงาน

MDIForm1 - [Form1]

ข้อมูลพนักงาน การทำงาน คำนวณพนักงาน

รหัสพนักงาน: 2 สถานภาพ: โสด

แผนก: วิศวกรรม สาขา: พุทธ

ชื่อ: ธรรมบรรพ์ อภิสำเนา: 121/5 ต.วังเหนือ อ.วังเหนือ จ.ลำปาง

นามสกุล: เจริญวิทย์โยภาส โทรศัพท์: 09-2526548

เพศ: ชาย จุดการศึกษา: ปริญญาตรีสาขาวิศวกรรมโทรคมนาคม

วันเดือนปีเกิด: 20/2/2526 วันเริ่มทำงาน: 5/5/2548

สัญชาติ: ไทย เงินเดือน: 26000

เวลาเข้า: 12:12:10 เวลาออก: 12:20:12

วันที่: 03/03/49 เวลา: 14:26:29

ปุ่ม: ค้นหา, ปิด

ปุ่ม: k, <, >, >|

รูปที่ 4.20 แสดงข้อมูลพนักงานบนมอนิเตอร์

4.4 วิเคราะห์การทดลอง

จากการทดลองทั้งหมดพบว่า เครื่องรับสามารถรับสัญญาณจากเครื่องส่ง 1 เครื่องได้โดยไม่เกิดปัญหานั้นเนื่องจากการที่ไม่มีสัญญาณรบกวนที่มีความถี่เดียวกันในบริเวณนั้นๆ ทำให้สัญญาณที่รับได้มีความต่อเนื่องที่สุดแต่การที่ส่วนแสดงผลมีการแสดงผลที่ช้าเนื่องจากทางคณะผู้จัดทำได้เขียนโปรแกรมให้เครื่องส่งมีการประวิงเวลาในการส่งเล็กน้อยเพื่อแก้ไขปัญหาการส่งสัญญาณ โดยเครื่องส่งหลายตัวที่ใช้ความถี่เดียวกันในบริเวณเดียวกันและในส่วนของเครื่องรับเองก็ได้เขียนโปรแกรมให้มีการตรวจสอบข้อมูลหลายขั้นตอนเนื่องจากป้องกันสัญญาณรบกวนที่มีความถี่เดียวกันกับเครื่องส่งที่ใช้งานอยู่นั่นเอง

เมื่อทดลองส่งสัญญาณจากเครื่องส่ง 2 เครื่องพร้อมกัน พบว่าเครื่องรับจะใช้เวลาในการตรวจสอบข้อมูลมากขึ้น และใช้เวลาเพื่อที่จะแสดงผลออกมาทางมอนิเตอร์มากขึ้น ซึ่งปัญหานี้เกิดจากการเพิ่มการประวิงเวลาและการตรวจสอบข้อมูลหลายขั้นตอนดังที่กล่าวมาแล้วและเหตุผลอีกประการหนึ่งคือ การที่มีเครื่องส่งสัญญาณหลายตัวที่ใช้ความถี่เดียวกันในบริเวณเดียวกันส่งสัญญาณพร้อมกันทำให้เครื่องรับไม่สามารถแยกสัญญาณข้อมูลได้ แม้ว่าทางคณะผู้จัดทำได้เขียนโปรแกรมให้เครื่องส่งมีการประวิงเวลาในการส่งเล็กน้อย เพื่อให้เครื่องส่งส่งสัญญาณคนละช่วงเวลาแต่ก็แก้ปัญหาได้เพียงบางช่วงเวลาเท่านั้น เมื่อเพิ่มเครื่องส่งสัญญาณเป็น 3 เครื่องพบว่าผลที่ได้เหมือนการทดลองส่งสัญญาณจากเครื่องส่ง 2 เครื่องพร้อมกันซึ่งปัญหาที่เกิดขึ้นก็เป็นปัญหาเดียวกันนั่นเอง

จากนั้นทดลองส่งสัญญาณจากเครื่องส่ง 2 เครื่องและใช้เครื่องรับ 2 เครื่องโดยให้เครื่องส่ง 1 เครื่องส่งสัญญาณให้เครื่องรับ 1 เครื่องและเครื่องรับ-ส่งทั้งสองชุดนี้อยู่ในบริเวณที่ห่างกันประมาณ 12 เมตรซึ่งจะทำให้เครื่องส่งแต่ละเครื่องจะไม่รบกวนกัน พบว่าเครื่องรับแต่ละเครื่องนั้นสามารถรับข้อมูลจากเครื่องส่งได้โดยไม่มีปัญหาซึ่งผลที่ได้จะเหมือนกันการใช้เครื่องรับ-ส่งเพียงชุดเดียวนั่นเอง ส่วนการแสดงผลที่มอนิเตอร์นั้นจะเกิดปัญหาล็กน้อยคือ ข้อมูลจากเครื่องรับที่ 2 ที่อยู่ห่างจากเครื่องคอมพิวเตอร์ 12 เมตร นั้นจะแสดงได้ช้ากว่าเล็กน้อย กล่าวคือ ผลจากเครื่องรับที่ 1 นั้นจะแสดงผลค้ำงานกว่าเครื่องที่ 2 เป็นเช่นนี้เนื่องจากสายซีเรียลที่ใช้ต่อจากวงจรเครื่องรับที่ 2 นั้นมีความยาวกว่านั่นเอง

ส่วนการแสดงผลบนจอมอนิเตอร์โดยใช้ โปรแกรม Visual basic นั้นจะพบว่า โดยรวมแล้วการแสดงผลเป็นไปตามโปรแกรมที่ได้เขียนไว้ แต่มีปัญหาเกิดขึ้นในส่วนของการบินทีกเวลาเข้า-ออก และการนับจำนวนครั้งที่เข้า-ออก คือ มีการบินทีกเวลาผิดพลาด และนับจำนวนครั้งบ่อยเกินไป ซึ่งสาเหตุเกิดมาจากวงจรเครื่องรับมีการส่งข้อมูลไปแสดงที่มอนิเตอร์แบบไม่ต่อเนื่อง เพราะว่าการเพิ่มการประวิงเวลาให้กับวงจรตัวรับ อีกทั้งมีการตรวจสอบข้อมูลหลายขั้นตอนยกตัวอย่างเช่น เมื่อพนักงานเดินผ่านประตูครั้งแรก รหัสข้อมูลจากวงจรเครื่องส่งถูกส่งไปยังวงจรเครื่องรับ เมื่อวงจรเครื่องรับรับข้อมูลได้ก็จะส่งต่อไปยังส่วนแสดงผล โปรแกรม Visual basic จะ

ทำการบันทึกเวลาที่รับข้อมูลได้ครั้งแรกเป็นเวลาเช้า แต่เนื่องจากความไม่ต่อเนื่องของการส่งข้อมูล ทำให้มีช่วงเวลาที่หยุดส่งข้อมูล เมื่อมีการส่งข้อมูลอีกครั้ง โปรแกรมจะทำการบันทึกเวลาเมื่อได้รับข้อมูลเป็นเวลาออกทันที จากนั้นจะทำการบันทึกเวลาเช้าใหม่เมื่อได้รับข้อมูลอีกครั้ง โดยโปรแกรม Visual basic จะทำการบันทึกเวลาเข้า-ออกเช่นนี้ไปเรื่อยๆ จนกระทั่งไม่มีการส่งสัญญาณจากวงจรเครื่องส่ง และส่วนการนับจำนวนครั้งการเข้า-ออก ก็จะบันทึกจำนวนครั้งการเข้า-ออก ตามจำนวนครั้งที่ได้รับข้อมูลไปด้วย ซึ่งการบันทึกข้อมูลทั้งสองชนิดนี้ไม่เป็นไปตามที่ต้องการ คือในการเดินผ่านประตูหนึ่งครั้ง แต่มีการบันทึกเวลาเข้า-ออก และจำนวนครั้งการเข้าออกมากกว่าหนึ่งครั้งนั่นเอง

จากการทดลองทั้งหมดกล่าวได้ว่าปัญหาหลักของโครงการนี้คือการใช้เครื่องส่งที่ใช้การมอดูเลตสัญญาณแบบ ASK ซึ่งใช้ความถี่เดียวกันเมื่อส่งสัญญาณพร้อมกันก็จะทำให้เครื่องรับไม่สามารถแยกสัญญาณข้อมูลได้ และเหตุผลที่คณะผู้จัดทำได้เลือกใช้เครื่องส่งที่ใช้การมอดูเลตสัญญาณแบบ ASK นั้นเนื่องมาจากเครื่องส่งที่ใช้การมอดูเลตสัญญาณแบบ FSK นั้น เมื่อใช้เครื่องส่งหลายๆเครื่องก็จะต้องใช้เครื่องรับหลายเครื่องตามไปด้วยซึ่งจะทำให้เกิดปัญหาเมื่อนำไปใช้งานจริง ส่วนเครื่องส่งที่ใช้การมอดูเลตสัญญาณแบบ PSK นั้นมีราคาสูงเกินไป ดังนั้นเพื่อความเหมาะสมคณะผู้จัดทำจึงได้เลือกใช้เครื่องส่งที่ใช้การมอดูเลตสัญญาณแบบ ASK นั้นเองและปัญหาอีกประการหนึ่งคือกรณีที่เครื่องรับจะรับสัญญาณจากเครื่องส่งที่อยู่ใกล้และมีความแรงของสัญญาณสูงได้ดีกว่าเครื่องส่งที่อยู่ไกลและมีความแรงของสัญญาณต่ำซึ่งปัญหานี้จะทำให้ส่วนแสดงผลจะแสดงได้เฉพาะเครื่องส่งที่อยู่ใกล้เครื่องรับมากกว่าหรือส่งสัญญาณได้แรงกว่า ปัญหานี้ทางผู้จัดทำได้แก้ปัญหาเขียนโปรแกรมให้เครื่องรับโดยการพิจารณารับสัญญาณจากเครื่องส่งที่ละเครื่อง แต่ก็ทำให้ส่วนแสดงผลแสดงผลช้าลงหรืออาจไม่แสดงผลเนื่องจากมีการประวิงเวลามากเกินไปทำให้เครื่องรับไม่สามารถรับสัญญาณได้เนื่องจากรอรับสัญญาณคนละช่วงเวลากับเครื่องส่งนั่นเอง

4.5 สรุปการทดลอง

จากการทดลองสรุปได้ว่าปัญหาหลักของโครงการนี้คือการใช้เครื่องส่งที่ใช้การมอดูเลตสัญญาณแบบ ASK ซึ่งใช้ความถี่เดียวกันเมื่อส่งสัญญาณพร้อมกันก็จะทำให้เครื่องรับไม่สามารถแยกสัญญาณข้อมูลได้ ทำให้ทางคณะผู้จัดทำต้องแก้ปัญหาโดยการเพิ่มการประวิงเวลาในส่วนของเครื่องส่งเพื่อให้เครื่องส่งส่งสัญญาณคนละช่วงเวลา แต่ก็ส่งผลให้ส่วนแสดงผลแสดงข้อมูลได้ช้าลงตามไปด้วย

ดังนั้นทางผู้จัดทำจึงหวังเป็นอย่างยิ่งว่าจะสามารถแก้ปัญหาที่เกิดขึ้นเพื่อให้การส่งสัญญาณข้อมูลสมบูรณ์ขึ้นและส่วนแสดงผลมีการตอบสนองต่อสัญญาณข้อมูลได้ดีขึ้นด้วย



บทที่ 5

สรุปผลและข้อเสนอแนะของโครงการ

5.1 สิ่งที่ได้จากจากโครงการ

1. ได้รับความรู้เกี่ยวกับการส่ง-รับข้อมูลแบบรหัสไบนารีโดยผ่านคลื่นวิทยุ
2. ได้รับความรู้เกี่ยวกับการเข้ารหัส(Encode)และการถอดรหัส(Decode)เพื่อใช้ในการส่งข้อมูลผ่านคลื่นวิทยุ
3. ได้รับความรู้เกี่ยวกับการทำงานของเครื่องรับและเครื่องส่งสัญญาณผ่านคลื่นวิทยุ
4. ได้รับความรู้ในการใช้งาน Microcontroller และนำไปใช้งานได้จริง
5. ได้รับความรู้ในการใช้งานโปรแกรม ภาษา Assembly
6. ได้รับความรู้ในการใช้งานโปรแกรม Visual Basic เพื่อสร้างและแสดงฐานข้อมูล
7. ได้รับความรู้ในการประกอบอุปกรณ์ต่างๆ และสามารถเลือกใช้อุปกรณ์ที่เหมาะสมในโครงการ
8. ทำให้สามารถนำความรู้ที่ได้รับจากการเรียนทฤษฎีมาใช้ในการปฏิบัติจริง
9. ทำให้สามารถรู้จักแก้ปัญหาเฉพาะหน้าได้
10. ทำให้รู้จักการทำงานร่วมกับผู้อื่น
11. ทำให้รู้จักการแบ่งเวลาในการทำงานและสามารถปฏิบัติงานตามตารางเวลาได้
12. สามารถนำความรู้ที่ได้จากโครงการมาประยุกต์ใช้งานได้

5.2 ปัญหาและอุปสรรคในการทำโครงการ

1. มีความรู้ในการเขียนโปรแกรมภาษา Assembly น้อย จึงต้องใช้เวลาในการศึกษาและทำความเข้าใจมาก ทำให้เสียเวลาในส่วนนี้มาก
2. มีความรู้ในการเขียนโปรแกรม Visual Basic น้อย จึงต้องใช้เวลาในการศึกษาและทำความเข้าใจมากกว่ารวมทั้งต้องอาศัยผู้ที่มีความเชี่ยวชาญช่วยให้คำแนะนำ ทำให้เสียเวลาในส่วนนี้มาก
3. ใช้เวลานานในการศึกษาการใช้โปรแกรมร่วมกันระหว่างภาษา Assembly และ โปรแกรม Visual Basic
4. ใช้เวลาในการขอเบิกงบประมาณในการซื้ออุปกรณ์นานพอสมควรซึ่งมีผลทำให้การทำโครงการออกมาล่าช้า
5. มีความรู้ในการเลือกใช้อุปกรณ์น้อย จึงต้องอาศัยผู้ที่มีความเชี่ยวชาญช่วยให้คำแนะนำและใช้เวลาในการเลือกและสั่งซื้ออุปกรณ์นานพอสมควร
6. มีความรู้ในการประกอบวงจรบนแผ่น PCB น้อย ทำให้วงจรมีความเสียหายบ่อย และใช้งานไม่ได้ ทำให้ต้องสร้างวงจรหลายครั้งจึงจะสามารถนำมาใช้งานได้
7. ขาดอุปกรณ์ในการเขียนโปรแกรมลงในไมโครคอนโทรลเลอร์เพื่อใช้งานเนื่องจากอุปกรณ์มีราคาสูง
8. ใช้เวลาในการทดลองและตรวจสอบการใช้งานร่วมกันระหว่างวงจรเครื่องรับและวงจรเครื่องส่งนานพอสมควร
9. ใช้เวลาในการทดลองและแก้ปัญหาเรื่องการใช้เครื่องส่งสัญญาณที่ใช้ความถี่เดียวกันหลายตัวในบริเวณเดียวกันนานพอสมควร
10. ใช้เวลาในการทดลองและแก้ปัญหาเรื่องการแสดงข้อมูลต่างๆโดยใช้โปรแกรม Visual Basic นานพอสมควร

5.3 ปัญหาของโครงการงาน

จากการที่คณะผู้จัดทำได้เลือกใช้เครื่องรับ-ส่งสัญญาณวิทยุที่ใช้การเข้ารหัสสัญญาณข้อมูล (Modulation) แบบ Amplitude – Shift Keying (ASK) นั้น เพื่อให้โครงการนี้สามารถนำไปประยุกต์ใช้งานจริงได้ เพราะ การเข้ารหัสสัญญาณข้อมูลแบบ ASK นั้นจะใช้ความถี่ในการส่งสัญญาณเพียงความถี่เดียวทำให้ไม่ต้องใช้วงจรรับสัญญาณหลายวงจร และมีราคาถูกกว่า แบบ FSK และ PSK และคณะผู้จัดทำได้เลือกใช้โปรแกรม Visual basic สำหรับเขียนเพื่อแสดงผลออกทางจอมอนิเตอร์ ซึ่งโปรแกรม Visual basic เป็นโปรแกรมที่ใช้งานง่ายและใช้งานได้ดีกับอุปกรณ์ภายนอก จากการเลือกใช้อุปกรณ์และโปรแกรมที่ผิดพลาดมาแล้วทำให้เกิดปัญหาของโครงการงานดังนี้

1. ในการส่งสัญญาณโดยใช้เครื่องส่งสัญญาณเกิน 1 ตัว ภายในบริเวณเดียวกัน และใช้เครื่องรับสัญญาณเพียง 1 ตัว นั้นจะทำให้เครื่องรับไม่สามารถแยกสัญญาณและรับสัญญาณพร้อมๆกันได้
2. เนื่องจากต้องทำวงจรเครื่องส่งสัญญาณให้มีขนาดเล็ก เพื่อต่อการพกพา ทำให้ต้องใช้แหล่งจ่ายไฟขนาดเล็ก คือ ถ่านนาฬิกา ซึ่งเมื่อนำมาใช้แล้วพบว่า มีอายุการใช้งานสั้นมาก คือ ใช้งานได้ไม่ถึง 2 วัน และสัญญาณที่ได้มีความแรงของสัญญาณต่ำทำให้วงจรเครื่องรับรับสัญญาณได้ช้าลง
3. วงจรเครื่องรับจะรับสัญญาณที่มีความแรงของสัญญาณสูงกว่าได้ดีกว่ากล่าวคือ ถ้ามีเครื่องส่งสัญญาณ 2 ตัว ส่งสัญญาณพร้อมกัน และเครื่องส่งตัวแรกส่งสัญญาณที่มีความแรงของสัญญาณสูงกว่า อาจเนื่องมาจากแหล่งจ่ายไฟแรงกว่า หรือ อยู่ใกล้เครื่องรับมากกว่า จะทำให้เครื่องรับรับสัญญาณได้จากเครื่องส่งเพียงเครื่องเดียว ทำให้ส่วนแสดงผลแสดงข้อมูลของพนักงานคนแรกเท่านั้น ซึ่งจุดประสงค์ของโครงการนั้นต้องการระบุข้อมูลของทุกคนที่เดินผ่านเครื่องรับสัญญาณ
4. ในส่วนของโปรแกรมที่เขียนลงในไมโครคอนโทรลเลอร์นั้น เนื่องจากตัวโปรแกรมมีการตรวจสอบหลายขั้นตอน มีส่วนแสดงผลไปยังประตูและยังมีส่วนของการประวิงเวลาทำให้ในการทำงานแต่ละครั้งจะใช้เวลานานซึ่งจะส่งผลต่อการรับและส่งข้อมูลด้วย
5. ในส่วนของโปรแกรม Visual basic นั้น ตัวโปรแกรมถูกเขียนให้รอรับข้อมูลจากเครื่องรับสัญญาณอยู่ตลอดเวลา ดังนั้นเมื่อเครื่องรับไม่สามารถรับสัญญาณได้จะทำให้ไม่สามารถแสดงผลของข้อมูลที่จอมอนิเตอร์ และโปรแกรม Visual basic จะมีการประมวลผลที่เร็วมากทำให้ไม่สามารถดูข้อมูลที่แสดงผ่านจอมอนิเตอร์ของแต่ละคนได้ทันทีเมื่อมีการส่งสัญญาณข้อมูลจากเครื่องส่งหลายเครื่องพร้อมกัน
6. ในโปรแกรม Visual basic มีปัญหาเกิดขึ้นในส่วนของการบินที่เวลาเข้า-ออก และการนับจำนวนครั้งที่เข้า-ออก คือ มีการบินที่เวลาผิดพลาด และนับจำนวนครั้งบ่อยเกินไป ซึ่งสาเหตุเกิดมาจากวงจรเครื่องรับมีการส่งข้อมูลไปแสดงที่มอนิเตอร์แบบไม่ต่อเนื่อง

เพราะว่าการเพิ่มการประวิงเวลาให้กับวงจรตัวรับ อีกทั้งมีการตรวจสอบข้อมูลหลาย
ขั้นตอน จึงส่งผลให้ในการเดินผ่านประตูหนึ่งครั้ง แต่มีการบันทึกเวลาเข้า-ออก และ
จำนวนครั้งการเข้าออกมากกว่าหนึ่งครั้ง นั่นเอง



5.4 การแก้ปัญหาของโครงการ

1. การแก้ปัญหาในส่วนของการส่งข้อมูลเมื่อมีวงจรเครื่องส่งส่งสัญญาณพร้อมกันเกิน 1 ตัวภายในบริเวณเดียวกันนั้น ทำได้โดยการเขียนโปรแกรมเพิ่มเวลาประวิง(Delay) ในวงจรเครื่องส่งแต่ละตัวซึ่งจะทำให้วงจรเครื่องส่งส่งสัญญาณไม่พร้อมกันและทำให้วงจรเครื่องรับรับสัญญาณของแต่ละเครื่องได้แต่เวลาประวิงที่เพิ่มขึ้นจะส่งผลให้ส่วนแสดงผลมีการแสดงผลได้ช้าลง
2. การเพิ่มสวิตช์ให้แก่วงจรจะช่วยแก้ปัญหาเรื่องอายุการใช้งานของถ่านนาฬิกาได้กล่าวคือเมื่อพนักงานเดินเข้าใกล้ประตูจึงเปิดสวิตช์ให้วงจรทำงานและเมื่อผ่านประตูไปแล้วจึงปิดสวิตช์ซึ่งวิธีการนี้จะช่วยยืดอายุการใช้งานของถ่านนาฬิกาได้อีกทั้งทำให้วงจรเครื่องรับรับสัญญาณได้ดี
3. ในส่วนแสดงผลโดยใช้โปรแกรม Visual basic นั้นเนื่องจากต้องรับสัญญาณตลอดเวลาจึงจะแสดงผลได้อย่างต่อเนื่องทางคณะผู้จัดทำได้แก้ไขโดยการเขียนโปรแกรมให้แสดงผลของข้อมูลของพนักงานที่ผ่านประตูคนสุดท้ายค้างไว้บนจอมอนิเตอร์จนกระทั่งมีพนักงานคนอื่นเดินผ่านประตูจอมอนิเตอร์จึงจะแสดงข้อมูลของพนักงานคนนั้นๆ
4. การแก้ปัญหาเกิดขึ้นในส่วนของการบันทึกเวลาเข้า-ออก และการนับจำนวนครั้งที่เข้า-ออกนั้นทำได้โดยการเพิ่มส่วนบันทึกข้อมูล (Buffer) ขึ้นมา ส่วนนี้จะทำการบันทึกข้อมูลทุกค่าที่ได้รับและจะนำข้อมูลสุดท้ายไปแสดงและเก็บไว้ในฐานข้อมูล ส่วนข้อมูลที่เหลือนั้นจะลบทิ้ง เมื่อมีการรับข้อมูลได้อีกก็จะเริ่มบันทึกค่าและจะนำข้อมูลสุดท้ายไปแสดงและเก็บไว้ในฐานข้อมูลอีกครั้ง ซึ่งส่วนบันทึกข้อมูลที่เพิ่มเข้ามานี้จะช่วยแก้ปัญหาอธิบายได้ดังนี้ คือ เมื่อมีการเดินผ่านประตูครั้งแรก รหัสข้อมูลจากวงจรเครื่องส่งถูกส่งไปยังวงจรเครื่องรับ เมื่อวงจรเครื่องรับรับข้อมูลได้ก็จะส่งต่อไปยังส่วนแสดงผล ส่วนบันทึกข้อมูลจะทำการบันทึกเวลาที่รับข้อมูลได้ครั้งแรกเป็นเวลาเข้า แต่เนื่องจากความไม่ต่อเนื่องของการส่งข้อมูล ทำให้มีช่วงเวลาที่หยุดส่งข้อมูล เมื่อมีการส่งข้อมูลอีกครั้ง ส่วนบันทึกข้อมูลก็จะทำการบันทึกเวลาอีกครั้ง ดังนั้นเมื่อได้รับข้อมูล จะทำการบันทึกเวลาเข้า-ออกเช่นนี้ไปเรื่อยๆ จนกระทั่งไม่มีการส่งสัญญาณจากวงจรเครื่องส่ง จากนั้นส่วนบันทึกข้อมูลจะนำข้อมูลสุดท้ายไปแสดงและเก็บไว้ในฐานข้อมูล ส่วนข้อมูลที่เหลือนั้นจะลบทิ้งและจะทำงานเช่นนี้ไปเรื่อยๆ ในส่วนการนับจำนวนครั้งที่เข้า-ออกนั้นจะนับจำนวนครั้งของข้อมูลตามที่ถูกบันทึกในฐานข้อมูลจึงแก้ปัญหาส่วนนี้ไปด้วย ดังนั้นการเพิ่มส่วนบันทึกข้อมูลจึงสามารถช่วยแก้ปัญหาการบันทึกเวลาเข้า-ออก และการนับจำนวนครั้งที่เข้า-ออกได้

5.5 ข้อจำกัดของโครงการ

1. เนื่องจากการใช้เครื่องส่งสัญญาณที่ใช้การมอดูเลตแบบ ASK ซึ่งใช้ความถี่เดียวกันส่งสัญญาณพร้อมกันหลายตัวในบริเวณเดียวกัน ทำให้เครื่องรับไม่สามารถแยกสัญญาณเพื่อนำข้อมูลไปใช้งานได้
2. การเพิ่มการประวิงเวลาให้แก่เครื่องส่งให้ส่งคนละช่วงเวลานั้น แม้จะช่วยแก้ปัญหาเครื่องรับไม่สามารถแยกสัญญาณเพื่อนำข้อมูลไปใช้งานได้ แต่จะทำให้เครื่องรับรับสัญญาณข้อมูลได้น้อยและช้าลง ทำให้ส่วนแสดงผลแสดงข้อมูลได้ช้าลงด้วย
3. การเพิ่มการประวิงเวลาให้แก่เครื่องรับในส่วนของควบคุมการเปิด-ปิดประตูนั้น ทำให้เสียเวลาในการตรวจสอบสัญญาณมากและส่งผลให้เครื่องรับรับสัญญาณข้อมูลจากเครื่องส่งมาตรวจสอบได้ที่ละเครื่องเท่านั้น อีกทั้งยังทำให้ส่วนแสดงผลแสดงข้อมูลได้ช้าลงด้วย
4. เนื่องจากวงจรของเครื่องส่งมีขนาดเล็ก ทำให้ต้องใช้แหล่งจ่ายไฟเป็นถ่านนาฬิกาขนาดเล็กซึ่งมีอายุการใช้งานสั้น ต้องเปลี่ยนบ่อยครั้ง
5. การใช้แหล่งจ่ายไฟขนาดเล็กทำให้ความแรงและระยะการส่งสัญญาณของเครื่องส่งสั้นลง
6. เครื่องรับจะรับสัญญาณจากเครื่องส่งที่อยู่ใกล้และมีความแรงของสัญญาณสูงได้ดีกว่าเครื่องส่งที่อยู่ไกลและมีความแรงของสัญญาณต่ำ
7. เนื่องจากผลของโปรแกรมที่มีการตรวจสอบหลายขั้นตอนทำให้การทำงานของระบบโดยรวมช้าลง
8. ถ้าเครื่องส่งส่งข้อมูลไม่ต่อเนื่องเครื่องรับจะไม่สามารถรับข้อมูลได้ทำให้โปรแกรมวิชวลเบสิกไม่สามารถแสดงผลที่จอมอนิเตอร์ อีกทั้งจะส่งผลให้มีการบันทึกเวลาเข้า-ออก และการนับจำนวนครั้งที่เข้า-ออก ที่ผิดพลาดได้
9. โปรแกรมวิชวลเบสิกจะแสดงผลข้อมูลที่จอมอนิเตอร์เร็วมากคือข้อมูลจะเปลี่ยนแปลงอย่างรวดเร็วจนบางครั้งไม่สามารถอ่านข้อมูลได้ทัน เมื่อมีการส่งข้อมูลมาพร้อมกันจากเครื่องส่งหลายเครื่อง

5.6 ข้อเสนอแนะ

โครงการนี้เป็นโครงการที่มีการประยุกต์ใช้งานข้อมูลที่ถูกส่งผ่านคลื่นวิทยุ ซึ่งได้นำข้อมูลเหล่านี้ไปใช้งานในด้านการรักษาความปลอดภัยและการจัดเก็บข้อมูล ทำให้สามารถพัฒนาเพื่อให้ใช้งานที่มีลักษณะเดียวกันกับโครงการนี้ได้ดังข้อเสนอแนะต่อไปนี้

1. สามารถพัฒนาให้ทำงานเปิด-ปิดประตูที่จอดรถในสำนักงานและเพิ่มข้อมูลสำหรับรถยนต์รวมถึงตำแหน่งที่จอดรถเฉพาะพนักงานแต่ละคนได้
2. สามารถพัฒนาให้ใช้งานในหอพักหรือโรงแรมได้เช่น ทำงานเปิด-ปิดประตูห้องพักและบันทึกข้อมูลของผู้พักอาศัยแต่ละห้อง หรือแม้กระทั่งควบคุมการทำงานของลิฟต์ ซึ่งสามารถใช้ได้เฉพาะลูกค้าที่เข้ามาพักเท่านั้นได้
3. สามารถพัฒนาให้เครื่องส่งมีการทำงานแบบ Passive หรือ เครื่องส่งทำการเหนี่ยวนำพลังงานจากเครื่องรับ ซึ่งจะทำให้เครื่องส่งมีขนาดเล็กและประหยัดพลังงาน
4. สามารถเปลี่ยนโปรแกรมแสดงผลจากโปรแกรม Visual Basic ที่มีการตอบสนองต่อการประวิงเวลาไม่ดีนักเป็นโปรแกรมอื่นเช่นโปรแกรม Java ที่มีการตอบสนองต่อการประวิงเวลาที่ดีกว่าจะทำให้การแสดงผลดีขึ้นได้
5. สามารถแก้ไขการประวิงเวลาให้มีช่วงการประวิงเวลาที่แน่นอนและสั้นลงเพื่อลดผลกระทบต่อการทำงานของส่วนแสดงผลให้น้อยลง
6. สามารถเพิ่มส่วนบันทึกข้อมูล (Buffer) เพื่อใช้ในการแก้ไขปัญหาการบันทึกเวลาเข้า-ออก และการนับจำนวนครั้งที่เข้า-ออก ที่ผิดพลาดได้

เอกสารอ้างอิง

- [1] ดร. ประสิทธิ์ ทีชาพุด. การสื่อสารโทรคมนาคม.ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.
- [2] ไซโรจน์ ไววานิชกิจ .กมล เขมะรังสี .เปิดโลกการสื่อสารไร้สาย กรุงเทพฯ 2539.
- [3] URL: <http://dusithost.dusit.ac.th/~phitsanulok/e-learning/>
- [4] จิตติมา พุกบัวขาว. [Homepage], available from
URL:<http://student.bsru.ac.th/~s452229018/technology.htm>
- [5] URL:http://cptd.chandra.ac.th/selfstud/datacom2/Contents/Chapters/Encoding_Modulation.htm
- [6] URL: <http://media.en.kku.ac.th/178341/dtc03.htm>
- [7] ฉัททวุฒิ พิษผล และคณะ, คู่มือเรียน Visual Basic 6.0(ฉบับปรับปรุง). กรุงเทพฯ: โปรวิชั่น, 2547.
- [8] ชาริน สิทธิธรรมชาตรี, โปรแกรม SQL Server 2000, กรุงเทพฯ. ชัคเซส มีเดีย
- [9] สุภชัย สมพานิช, เรียนรู้และฝึกฝนการเขียน โปรแกรมใช้งานฐานข้อมูลด้วย Visual Basic ฉบับมืออาชีพ. นนทบุรี: ไอดีซี 2547.
- [10] อภิชาติ ภูพลับ, เริ่มต้นเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ด้วย Visual Basic.นนทบุรี อินโฟเพรส, 2546.

ภาคผนวก ก.
ชุดรับ-ส่งสัญญาณคลื่นวิทยุ

วงจรตัวส่งสัญญาณคลื่นวิทยุ (LAIPAC เบอร์ TLP43)

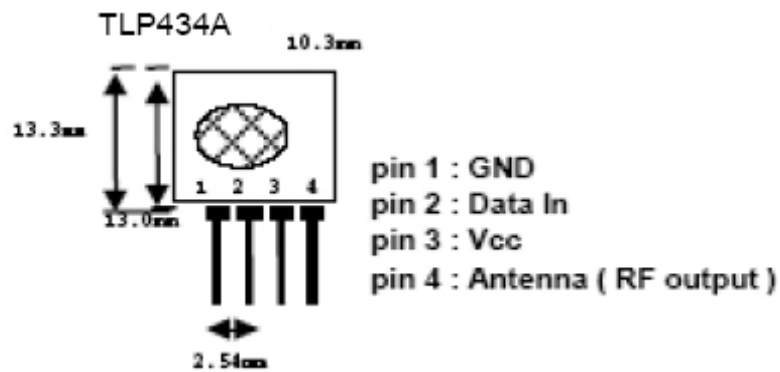


วงจรตัวส่งสัญญาณคลื่นวิทยุ

-คุณสมบัติของวงจรตัวส่งสัญญาณคลื่นวิทยุ

1. ทำงานที่ความถี่ 433.92 เมกะเฮิร์ตซ์ (MHz)
2. มีการเข้ารหัสสัญญาณข้อมูล (Modulation) สัญญาณดิจิทัลให้เป็นอนาล็อก แบบ Amplitude – Shift Keying (ASK)
3. สามารถทำงานได้ในช่วงแรงดันอินพุต 2-12 โวลต์ (Vdc)
4. กำลังงานด้านเอาต์พุต 8 มิลลิวัตต์ ที่แรงดันอินพุต 3.6 โวลต์
5. มีอัตราการส่งข้อมูล (Data Rate) เท่ากับ 2.4 กิโลบิตต่อวินาที (KB/s)

-โครงสร้างภายนอกและตำแหน่งขา



โครงสร้างภายนอกและตำแหน่งขาของวงจรตัวส่ง

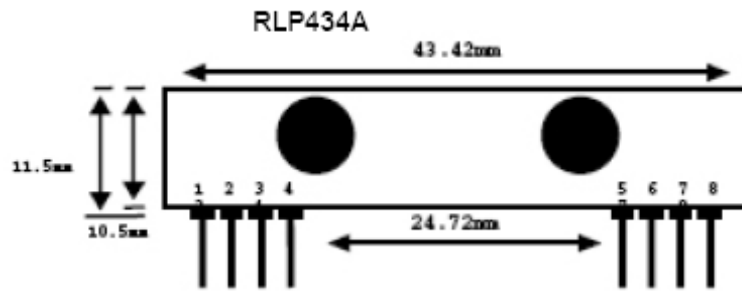
-รายละเอียดของขาวงจรตัวส่งมีดังนี้

- ขา 1 เป็นขา GND ใช้สำหรับเป็นกราวด์ของระบบ
- ขา 2 เป็นขา ที่รับข้อมูลเข้ามา (Data In)
- ขา 3 เป็นขา VCC ใช้สำหรับต่อไฟเลี้ยง
- ขา 4 เป็นขา ที่ต่อกับสายอากาศ(RF output)

วงจรตัวรับสัญญาณคลื่นวิทยุ (LAIPAC เบอร์ RLP434A)

- คุณสมบัติของวงจรตัวส่งสัญญาณคลื่นวิทยุ
- 1. ทำงานที่ความถี่ 433.92 เมกะเฮิร์ตซ์ (MHz)
- 2. มีการเข้ารหัสสัญญาณข้อมูล (Modulation) สัญญาณดิจิทัลให้เป็นอนาล็อก แบบ Amplitude – Shift Keying (ASK)
- 3. สามารถทำงานได้ในช่วงแรงดันอินพุต 4.5-5.5 โวลต์ (Vdc)
- 4. ให้เอาท์พุตเป็นสัญญาณดิจิทัล (Digital) และเป็นเชิงเส้น (Linear)
- 5. มีความไว 3 ไมโครวัตต์ (uVrms)
- 6. มีอัตราการส่งข้อมูล (Data Rate) เท่ากับ 3 กิโลบิตต่อวินาที (KB/s)

-โครงสร้างภายนอกและตำแหน่งขา



- pin 1 : Gnd
- pin 2 : Digital Data Output
- pin 3 : Linear Output /Test
- pin 4 : Vcc
- pin 5 : Vcc
- pin 6 : Gnd
- pin 7 : Gnd
- pin 8 : Antenna

โครงสร้างภายนอกและตำแหน่งขาของวงจรตัวรับ

-รายละเอียดของขาวงจรตัวรับมีดังนี้

- ขา 1,6 และ ขา 7 เป็นขา GND ใช้สำหรับเป็นกราวด์ของระบบ
- ขา 2 เป็นขา ที่ส่งข้อมูลออก(Digital Data Output)
- ขา 3 เป็นขา เอาต์พุตเชิงเส้นและใช้เป็นขาทดสอบ
- ขา 4 และขา 5 เป็นขา VCC ใช้สำหรับต่อไฟเลี้ยง
- ขา 8 เป็นขา ที่ต่อกับสายอากาศ

ภาคผนวก ข.

โปรแกรมควบคุมการทำงานเครื่องรับ-ส่ง

1.โปรแกรมวงจรเครื่องส่ง

```
org    0000h
mov    scon,#50h           ; สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov    tmod,#20h
mov    th1,#0fbh
setb   tr1
clr    ti
clr    ri
mov    r7,#0100           ; นำค่า 100 เก็บไว้ที่ R7
aa:   mov    sbuf,#55h     ; นำค่า 55 H เก็บไว้ที่ SBUF
      jnb   ti,$          ; รอส่งข้อมูลที่ละบิตจนกระทั่งข้อมูลครบ 8 บิต
      clr   ti           ; เคลียร์ ti เพื่อส่งค่าในรอบต่อไป
      djnz  r7,aa        ; วนรอบส่งจนกระทั่งค่า R7 ลดจาก 100 ให้เป็น 0
      mov   sbuf,#06h    ; ส่งส่วนหัวของข้อมูลซึ่งเป็นรหัสที่ใช้แทนตำแหน่งของพนักงาน
      jnb   ti,$
      clr   ti
      mov   sbuf,#61h    ; ส่งส่วนของข้อมูลจริง
      jnb   ti,$
      clr   ti
```

```
org    0000h
mov    scon,#50h           ; สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov    tmod,#20h
mov    th1,#0fbh
setb   tr1
clr    ti
clr    ri
```

```

        clr    p3.7          ;เคลียร์ขา p3.7 เพื่อป้องกันการรบกวนจากสถานะ High ของข้อมูล
loop:   mov    r7,#0100      ; นำค่า 100 เก็บไว้ที่ R7
aa:     mov    sbuf,#55h     ; นำค่า 55 H เก็บไว้ที่ SBUF
        jnb   ti,$          ; รอส่งข้อมูลที่ละบิตจนกระทั่งข้อมูลครบ 8 บิต
        clr   ti            ; เคลียร์ ti เพื่อส่งค่าในรอบต่อไป
        djnz  r7,aa         ; วนรอบส่งจนกระทั่งค่า R7 ลดจาก 100 ให้เป็น 0

```

```

        mov    r6,#04h      ; นำค่า 04h เก็บไว้ที่ R6
xx1:   mov    sbuf,#06h     ; ส่งส่วนหัวของข้อมูล
        jnb   ti,$
        clr   ti
        mov    sbuf,#61h    ; ส่งส่วนของข้อมูลจริงของพนักงานคนที่ 1
        jnb   ti,$
        clr   ti
        djnz  r6,xx1       ; วนรอบส่งจนกระทั่งค่า R6 ลดจาก 4 ให้เป็น 0
        setb  p3.7         ; เซตขา p3.7 เพื่อเริ่มทำงานรอบใหม่
        call  delay        ; เรียกส่วนประวิงเวลา
        jmp   loop         ; วนรอบเพื่อเริ่มการทำงานใหม่

```

```

delay:   setb  p3.7         ; ส่วนประวิงเวลา

```

```

        mov    r3,#0ch

```

```

delay 1:  mov    r4,#0ffh

```

```

delay 2:  mov    r5,#0ffh

```

```

        djnz  r5,$

```

```

        djnz  r4, delay 2

```

```

        djnz  r3, delay 1

```

```

clr    p3.7
ret    ;ย้อนกลับไปหลังคำสั่งที่เรียกใช้
end    ;จบการทำงาน

```

2.1. โปรแกรมวงจรถูกเครื่องรับ

```

org    0000h
mov    scon,#50h    ;สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov    tmod,#20h
mov    th1,#0fbh
setb   tr1
clr    ti
clr    ri

jnb    ri,$    ;รับข้อมูลที่ละบิตจนครบ 8 บิตแล้วเก็บไว้ใน sbuf
mov    a,sbuf    ;เก็บข้อมูลจาก sbuf ไว้ในรีจิสเตอร์ (Register) a
clr    ri    ;เคลียร์ ri เพื่อรับค่าในรอบต่อไป

cjne   a,#06h,loop5    ;ตรวจสอบส่วนหัวของข้อมูลว่าใช่หรือไม่ ถ้าใช่จะรอรับข้อมูล
จริง ถ้าไม่ใช่จะ ทำการเทียบรหัสไปเรื่อยๆถ้าตรงจะรอรับข้อมูลจริง แต่ถ้าไม่ตรงเริ่มรับ
ข้อมูลใหม่

jnb    ri,$    ;รอรับข้อมูลจริง
mov    a,sbuf
clr    ri

cjne   a,#61h,loop    ;ตรวจสอบข้อมูลจริงว่าใช่หรือไม่ ถ้าใช่จะส่งข้อมูลจริงไปส่วน
แสดงผล ถ้าไม่ใช่จะเริ่มรับข้อมูลใหม่
mov    sbuf,a    ;ส่งข้อมูลไปยังส่วนแสดงผล
jnb    ti,$
clr    ti
org    0000h

```

```

mov    scon,#50h           ; สำหรับเปิดพอร์ตการสื่อสารแบบอนุกรม
mov    tmod,#20h
mov    th1,#0fbh
setb   tr1
clr    ti
clr    ri
setb   p3.6                ; เปิดการแสดงสถานะของหลอดไฟ
setb   p3.7                ; เปิดการแสดงสถานะของหลอดไฟ
mov    p0,#11111111b       ; หยุดการทำงานของประตู
loop:  mov    p0,#11111111b ; หยุดการทำงานของประตู
jnb    ri,$                ; รับข้อมูลที่ละบิตจนครบ 8 บิตแล้วเก็บไว้ใน sbuf
mov    a,sbuf              ; เก็บข้อมูลจาก sbuf ไว้ในรีจิสเตอร์ (Register) a
clr    ri                  ; เคลียร์ ri เพื่อรับค่าในรอบต่อไป
loop1: cjne   a,#06h,loop5   ; ตรวจสอบส่วนหัวของข้อมูลว่าใช่หรือไม่ ถ้าใช่จะรอรับข้อมูล
จริง
                                ; ถ้าไม่ใช่จะ ทำการเทียบรหัสไปเรื่อยๆ ถ้าตรงจะรอรับข้อมูลจริง
                                ; แต่ถ้าไม่ตรงเริ่มรับข้อมูลใหม่(06hเป็นรหัสในตำแหน่งวิศวกร)
jnb    ri,$                ; ส่งข้อมูลไปยังส่วนแสดงผล
mov    a,sbuf
clr    ri
jmp    loop                ; วนรอบเพื่อเริ่มการทำงานใหม่

```

```

loop2: cjne   a,#61h,loop3   ; ตรวจสอบข้อมูลจริงว่าเป็นข้อมูลของพนักงานคนที่ 1 ใช่
หรือไม่                        ; ถ้าใช่จะส่งข้อมูลนั้นไปส่วนแสดงผล แต่ถ้า
                                ; ไม่ใช่จะตรวจข้อมูลพนักงานคนต่อไป
mov    sbuf,a                ; ส่งข้อมูลไปยังส่วนแสดงผล
jnb    ti,$
clr    ti
mov    p0,#11111110b        ; เปิดประตู

```

```

call    delay500ms        ;ประวิงเวลาประมาณ 0.5 วินาที
mov     p0,#11111101b     ;ปิดประตู
jmp     loop              ;วนรอบเพื่อเริ่มการทำงานใหม่
loop3:  cjne    a,#62h,loop4      ;ตรวจสอบข้อมูลจริงว่าเป็นข้อมูลของพนักงานคนที่
        2 ใช่หรือไม่                ถ้าใช่จะส่งข้อมูลนั้นไปส่วนแสดงผล
                                แต่ถ้าไม่ใช่จะตรวจสอบข้อมูลพนักงานคนต่อไป
mov     sbuf,a            ;ส่งข้อมูลไปยังส่วนแสดงผล
jnb     ti,$
clr     ti
mov     p0,#11111110b     ;เปิดประตู
call    delay500ms        ;ประวิงเวลาประมาณ 0.5 วินาที
mov     p0,#11111101b     ;ปิดประตู
jmp     loop              ;วนรอบเพื่อเริ่มการทำงานใหม่
loop4:  cjne    a,#63h,loop      ;ตรวจสอบข้อมูลจริงว่าเป็นข้อมูลของพนักงานคนที่ 3
        ใช่หรือไม่                ถ้าใช่จะส่งข้อมูลนั้นไปส่วน
        แสดงผล แต่ถ้าไม่ใช่จะเริ่มรับข้อมูลใหม่
mov     sbuf,a            ;ส่งข้อมูลไปยังส่วนแสดงผล
jnb     ti,$
clr     ti
mov     p0,#11111110b     ;เปิดประตู
call    delay500ms        ;ประวิงเวลาประมาณ 0.5 วินาที
mov     p0,#11111101b     ;ปิดประตู
jmp     loop              ;วนรอบเพื่อเริ่มการทำงานใหม่
delay500ms: call    delay100ms    ;ส่วนประวิงเวลาประมาณ 0.5 วินาที
        call    delay100ms
        call    delay100ms
        call    delay100ms
        call    delay100ms
        ret
delay100ms: mov     r6,#100

```

```
dly100ms:    call    delay500u
             djnz   r6, delay100ms
             ret
delay500u:   mov    r5,#230
             djnz   r5,$
             end           ;จบการทำงาน
```



ภาคผนวก ค.
โปรแกรมวิชาเว็บติก

Form 1

Option Explicit

Dim Data As String ;ตัวแปร Data เป็นตัวแปรประเภท String

Dim recdata As String ;ตัวแปร recdata เป็นตัวแปรประเภท String

Dim x, y, z, x1, x2 As Long ;ตัวแปร x, y, z, x1, x2 เป็นตัวแปรประเภท Long

Private Sub Command1_Click()

Adodc2.Recordset.MoveFirst ;เลื่อนไปเรคคอร์ดแรกของข้อมูล

Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ;แสดงรูปจากฐานข้อมูล

End Sub

Private Sub Command2_Click()

If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF

Adodc2.Refresh ; Refresh ข้อมูล

Else

Adodc2.Recordset.MovePrevious ; เลื่อนไปเรคคอร์ดก่อนหน้า

If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF

Else

Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล

End If

End If

End Sub

Private Sub Command3_Click()

If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF

Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย

Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล

Else

Adodc2.Recordset.MoveNext ; เลื่อนไปเรคคอร์ดถัดไป

If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF

Picture1.Refresh ; Refresh รูป

Else


```

Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจาก
ฐานข้อมูล
End If
End If
End Sub
Private Sub Command4_Click()
Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
End Sub
Private Sub Command5_Click()
Dim id As String ; ตัวแปร id เป็นตัวแปรประเภท
String
id = InputBox("พิมพ์รหัสพนักงานที่ต้องการค้นหา", "รหัสพนักงาน")
If id = "" Then ; ตรวจสอบ id
Exit Sub ; จบการทำงาน
Else
Adodc2.RecordSource = "select * from employee where employee.emp_id like " & id & " "
; ค้นหาข้อมูลจากราย employee โดยค้นหาจาก id
Adodc2.Refresh ; Refresh ข้อมูล
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
End If
End Sub
Private Sub Form_Load()
MSComm1.PortOpen = True ; เปิดพอร์ตอนุกรม พอร์ตที่ 1
MSComm2.PortOpen = True ; เปิดพอร์ตอนุกรม พอร์ตที่ 2
x = 0 ; กำหนดให้ x = 0
y = 0 ; กำหนดให้ y = 0
z = 0 ; กำหนดให้ z = 0
x1 = 0 ; กำหนดให้ x1 = 0
x2 = 0 ; กำหนดให้ x2 = 0
End Sub
Private Sub Form_Unload(Cancel As Integer)

```

```

MSComm1.PortOpen = False ; ปิดพอร์ตอนุกรม พอร์ตที่ 1
MSComm2.PortOpen = False ; ปิดพอร์ตอนุกรม พอร์ตที่ 2
End Sub

Private Sub Timer1_Timer()
    Label15.Caption = Format(Time(), "hh:mm:ss") ; เขียนคำสั่งเพื่อแสดงเวลาตามเวลาจริง
End Sub

Private Sub Timer2_Timer()
    Label16.Caption = Format(Date, "DD/MM/YY") ; เขียนคำสั่งเพื่อแสดงวันตามเวลาจริง
End Sub

Private Sub Timer3_Timer()
    Dim id As String ; ตัวแปร id เป็นตัวแปรประเภท String
    Dim Data As String ; ตัวแปร Data เป็นตัวแปรประเภท String
    Data = MSComm1.Input ; อ่านค่าจากบัฟเฟอร์ของพอร์ตอนุกรม เก็บไว้ที่ตัวแปร
    Data
    If Data = "" Then ; ตรวจสอบ Data
        Load Form2 ; โหลดฟอร์ม 2
    Else
        Form2.Hide ; ซ่อนฟอร์ม 2
        Form1.Show ; โหว้ฟอร์ม 1
    End If
    id = Val(Asc(Data) - 96) ; คำนวณค่า Data เก็บไว้ที่ตัวแปร id
    Adodc2.RecordSource = "select * from employee where employee.emp_id like " & id & " "
    ; ค้นหาข้อมูลจากตาราง employee โดยค้นหาจาก id
    Adodc2.Refresh ; Refresh ข้อมูล
    Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; ส่งให้ load รูปในตำแหน่งที่ Adodc2
    อยู่
    Adodc1.RecordSource = "select * from timetable where timetable.emp_id like " & "" &
    Text1.Text & "% " & " "
    ; ค้นหาข้อมูลจากตาราง timetable โดยค้นหาจาก id ที่ปรากฏใน Text1.Text
    Adodc1.Refresh ; Refresh ข้อมูล
    Adodc1.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
    Text24.Text = Adodc1.Recordset.Fields("timeout").Value ; ให้ Text24 รับค่าเวลาออก
    If Text24.Text <> "-----" Then ; ตรวจสอบว่าเวลาออกว่างหรือไม่
        Adodc3.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
    End If

```

```

Adodc1.Recordset.AddNew ; ให้ทำการเพิ่มเรคคอร์ดใหม่
Text17 = Text23 + 1
Text18.Text = Text1.Text
Text19.Text = Label16.Caption ; ให้เก็บค่าวันที่เข้า
Text20.Text = Label15.Caption ; ให้เก็บค่าเวลาที่เข้า
Text21.Text = "-----" ; ให้วันที่ออกเป็นค่าว่าง
Text22.Text = "-----" ; ให้เวลาออกเป็นค่าว่าง
Adodc1.Recordset.Update ; บันทึกค่าลงฐานข้อมูล
Adodc3.Refresh
Adodc1.Refresh
Adodc1.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
If Text1.Text = "1" Then
    x = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
End If
If Text1.Text = "2" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
    y = (Int(Text17.Text) - 101)
End If
If Text1.Text = "3" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
    z = (Int(Text17.Text) - 101)
End If
If Text1.Text = "4" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
    x1 = (Int(Text17.Text) - 101)
End If
If Text1.Text = "5" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
    x2 = (Int(Text17.Text) - 101)
End If
Else
Adodc4.Refresh
If Text1.Text = "1" Then
    Adodc4.Recordset.Move (x) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "2" Then

```

```

Adodc4.Recordset.Move (y) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "3" Then
    Adodc4.Recordset.Move (z) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "4" Then
    Adodc4.Recordset.Move (x1) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
If Text1.Text = "5" Then
    Adodc4.Recordset.Move (x2) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If
Adodc4.Recordset.Fields("timeout").Value = Label15.Caption ; ลงค่าวันที่ออก
Adodc4.Recordset.Fields("dateout").Value = Label16.Caption ; ลงค่าเวลาที่ออก
Text29.Text = Label16.Caption ; แสดงวันที่ออก
Text30.Text = Label15.Caption ; แสดงเวลาที่ออก
Text16.Text = Text30.Text
Adodc4.Recordset.Update ; บันทึกค่าลงฐานข้อมูล
End If
End If
End Sub

Private Sub Timer4_Timer()
Dim id As String ; ตัวแปร id เป็นตัวแปรประเภท String
Dim Data As String ; ตัวแปร Data เป็นตัวแปรประเภท String
Data = MSComm2.Input ; อ่านค่าจากบัฟเฟอร์ของพอร์ตอนุกรม เก็บไว้ที่ตัวแปร
Data
If Data = "" Then ; ตรวจสอบ Data
Load Form2 ; โหลดฟอร์ม 2
Else
Form2.Hide ; ซ่อนฟอร์ม 2
Form1.Show ; โช่วฟอร์ม 1
id = Val(Asc(Data) - 96) ; คำนวณค่า Data เก็บไว้ที่ตัวแปร id

```

```

Adodc2.RecordSource = "select * from employee where employee.emp_id like " & id & " "
; ค้นหาข้อมูลจากตาราง employee โดยค้นหาจาก id
Adodc2.Refresh ; Refresh ข้อมูล
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; ส่งให้ load รูปในตำแหน่งที่
Adodc2 อยู่
Adodc1.RecordSource = "select * from timetable where timetable.emp_id like " & ""
& Text1.Text & "%" & " "
; ค้นหาข้อมูลจากตาราง timetable โดยค้นหาจาก id ที่ปรากฏใน Text1.Text
Adodc1.Refresh ; Refresh ข้อมูล
Adodc1.Recordset.MoveLast ; เลื่อน ไปเรคคอร์ดสุดท้าย
Text24.Text = Adodc1.Recordset.Fields("timeout").Value ; ให้ Text24 รับค่าเวลาออก
If Text24.Text <> "-----" Then ; ตรวจสอบว่าเวลาออกว่างหรือไม่
Adodc3.Recordset.MoveLast ; เลื่อน ไปเรคคอร์ดสุดท้าย
Adodc1.Recordset.AddNew ; ให้ทำการเพิ่มเรคคอร์ดใหม่
Text17 = Text23 + 1
Text18.Text = Text1.Text
Text19.Text = Label16.Caption ; ให้เก็บค่าวันที่เข้า
Text20.Text = Label15.Caption ; ให้เก็บค่าเวลาที่เข้า
Text21.Text = "-----" ; ให้วันที่ออกเป็นค่าว่าง
Text22.Text = "-----" ; ให้เวลาออกเป็นค่าว่าง
Adodc1.Recordset.Update ; บันทึกค่าลงฐานข้อมูล
Adodc3.Refresh
Adodc1.Refresh
Adodc1.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
If Text1.Text = "1" Then
    x = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
End If
If Text1.Text = "2" Then ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
    y = (Int(Text17.Text) - 101)
End If
If Text1.Text = "3" Then
    z = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
End If

```

```

If Text1.Text = "4" Then
    x1 = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
End If

If Text1.Text = "5" Then
    x2 = (Int(Text17.Text) - 101) ; กำหนดค่าให้ตัวแปรเพื่อระบุผู้ใช้
End If

Else
Adodc4.Refresh

If Text1.Text = "1" Then
    Adodc4.Recordset.Move (x) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If

If Text1.Text = "2" Then
    Adodc4.Recordset.Move (y) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If

If Text1.Text = "3" Then
    Adodc4.Recordset.Move (z) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If

If Text1.Text = "4" Then
    Adodc4.Recordset.Move (x1) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If

If Text1.Text = "5" Then
    Adodc4.Recordset.Move (x2) ; ให้เลื่อนเรคคอร์ดไปยังเวลาผู้ใช้
End If

Adodc4.Recordset.Fields("timeout").Value = Label15.Caption ; ลงค่าวันที่ออก
Adodc4.Recordset.Fields("dateout").Value = Label16.Caption ; ลงค่าเวลาที่ออก
Text29.Text = Label16.Caption ; แสดงวันที่ออก
Text30.Text = Label15.Caption ; แสดงเวลาที่ออก
Text16.Text = Text30.Text
Adodc4.Recordset.Update ; บันทึกค่าลงฐานข้อมูล
End If
End If
End Sub

```

Form 2

```
Private Sub Command1_Click() ; ปุ่มปิด
Form2.Hide ; ซ่อนฟอร์มที่ 2
End Sub

Private Sub Command2_Click() ; ปุ่มค้นหา
Dim a As Integer ; ตัวแปร a เป็นตัวแปรประเภท
Integer

Adodc2.RecordSource = "select timetable.emp_id , timetable.date ,
timetable.time,timetable.dateout , timetable.timeout from timetable where timetable.emp_id
like " & "" & Text1.Text & "%" & " and timetable.date like " & "" & Text2.Text & "%" & "
and timetable.timeout like " & "" & "-----" & "%" & ""
; ทำการเลือกค่าจากตาราง timetable
Adodc2.Refresh
a = Int(Adodc2.Recordset.RecordCount) ; ทำการนับเรคคอร์ด
Adodc1.RecordSource = "select timetable.emp_id ,
timetable.dat,timetable.time,timetable.dateout , timetable.timeout from timetable where
timetable.emp_id like " & "" & Text1.Text & "%" & " and timetable.date like " & "" &
Text2.Text & "%" & ""
; ทำการเลือกค่าจากตาราง timetable
Adodc1.Refresh
DataGrid1.Refresh ; ให้ DataGrid1 รับค่าจาก
Adodc1 ใหม่
Label3.Caption = Int(Adodc1.Recordset.RecordCount) ; แสดงผลการนับเวลาเข้า
Label6.Caption = Int(Adodc1.Recordset.RecordCount) - a ; แสดงผลการนับเวลาออก
End Sub
```

Form 3

```
Private Sub Command1_Click() ; ปุ่มเลื่อนเรคคอร์ดไป
; ซ่อนเรคคอร์ดแรก
Adodc2.Recordset.MoveFirst ; เลื่อนเรคคอร์ดไปยังเรคค
; ออร์ดแรก
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; สั่งให้loadรูปในตำแหน่งที่
Adodc2อยู่
```

```

End Sub
Private Sub Command10_Click() ; ปุ่มยกเลิกการลบข้อมูลพนักงาน
Adodc2.Refresh
Frame2.Visible = False ; ให้เฟรม 2 มีคุณสมบัติในการแสดงเป็น False
End Sub
Private Sub Command11_Click() ; ปุ่มยืนยันการลบข้อมูลพนักงาน
Adodc2.Recordset.Delete ; ลบข้อมูลพนักงาน
Adodc2.Refresh
Frame2.Visible = False ; ให้เฟรม 2 มีคุณสมบัติในการแสดงเป็น False
MsgBox ("ข้อมูลถูกลบแล้ว") ; แสดงข้อความ("ข้อมูลถูกลบแล้ว ")
End Sub
Private Sub Command12_Click() ; ปุ่มยืนยันการแก้ไขข้อมูล
Adodc2.Recordset.Update ; แก้ไขข้อมูลลงฐานข้อมูล
Adodc2.Refresh
Frame3.Visible = False ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น False
MsgBox ("ข้อมูลถูกแก้ไขแล้ว") ; แสดงข้อความ("ข้อมูลถูกแก้ไขแล้ว ")
End Sub
Private Sub Command13_Click() ; ปุ่มยกเลิกการแก้ไขข้อมูลพนักงาน
Adodc2.Refresh
Frame3.Visible = False ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น False
End Sub
Private Sub Command2_Click()
If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF
Adodc2.Refresh ; Refresh ข้อมูล
Else
Adodc2.Recordset.MovePrevious ; เลื่อนไปเรคคอร์ดก่อนหน้า
If Adodc2.Recordset.BOF = True Then ; ถ้าจุดนี้เป็น BOF
Else
Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
End If
End If
End Sub

```



```

Private Sub Command3_Click()
    If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF
        Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
        Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
    Else
        Adodc2.Recordset.MoveNext ; เลื่อนไปเรคคอร์ดถัดไป
        If Adodc2.Recordset.EOF = True Then ; ถ้าจุดนี้เป็น EOF
            Picture1.Refresh ; Refresh รูป
        Else
            Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจาก
ฐานข้อมูล
        End If
    End If
End Sub

Private Sub Command4_Click()
    Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
    Picture1.Picture = LoadPicture(Adodc2.Recordset.Fields("pic")) ; แสดงรูปจากฐานข้อมูล
End Sub

Private Sub Command5_Click() ; ปุ่มเพิ่มข้อมูลพนักงาน
    Adodc2.Recordset.MoveLast ; เลื่อนไปเรคคอร์ดสุดท้าย
    Adodc2.Recordset.AddNew ; เพิ่มเรคคอร์ดใหม่
    Frame1.Visible = True ; ให้เฟรม 1 มีคุณสมบัติในการแสดงเป็น True
End Sub

Private Sub Command6_Click() ; ปุ่มลบข้อมูลพนักงาน
    Frame2.Visible = True ; ให้เฟรม 2 มีคุณสมบัติในการแสดงเป็น True
End Sub

Private Sub Command7_Click() ; ปุ่มแก้ไขข้อมูลพนักงาน
    Frame3.Visible = True ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น True
End Sub

Private Sub Command8_Click() ; ปุ่มยืนยันการแก้ไขข้อมูล
    Adodc2.Recordset.Update ; แก้ไขข้อมูลลงฐานข้อมูล
    Adodc2.Refresh

```

Frame1.Visible = False

End Sub

Private Sub Command9_Click() ; ปุ่มยกเลิกการแก้ไขข้อมูล

Adodc2.Refresh

Frame1.Visible = False ; ให้เฟรม 3 มีคุณสมบัติในการแสดงเป็น False

End Sub

MDI Form1

Private Sub emp_Click()

Form3.Show ; แสดงหน้าต่างข้อมูลพนักงาน

Form2.Hide ; ซ่อนหน้าต่างค้นหาพนักงาน

Form1.Hide ; ซ่อนหน้าต่างข้อมูลการเข้าทำงานของพนักงาน

End Sub

Private Sub find_Click()

Form1.Show ; แสดงหน้าต่างค้นหาพนักงาน

Form2.Hide ; ซ่อนหน้าต่างข้อมูลการเข้าทำงานของพนักงาน

Form3.Hide ; ซ่อนหน้าต่างข้อมูลพนักงาน

End Sub

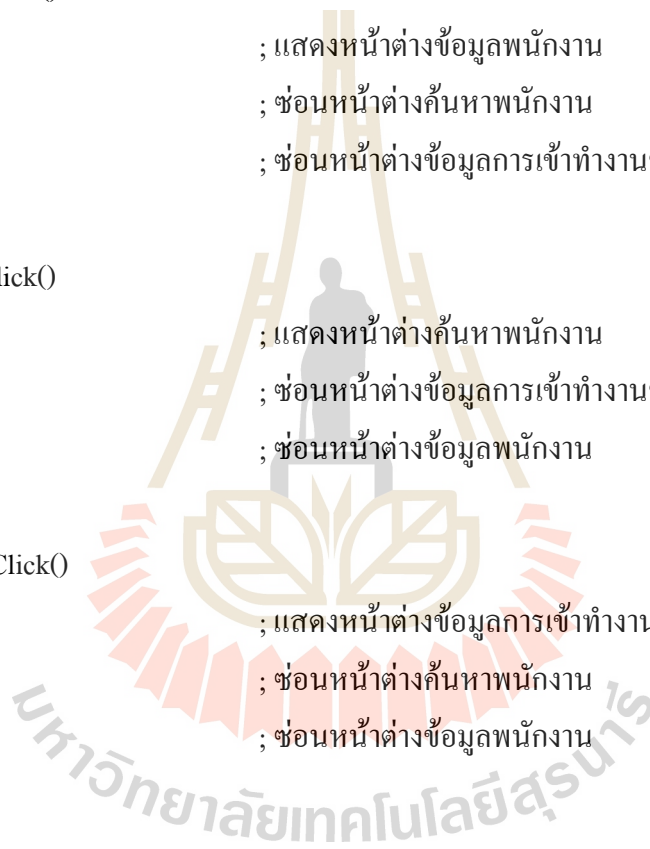
Private Sub work_Click()

Form2.Show ; แสดงหน้าต่างข้อมูลการเข้าทำงานของพนักงาน

Form1.Hide ; ซ่อนหน้าต่างค้นหาพนักงาน

Form3.Hide ; ซ่อนหน้าต่างข้อมูลพนักงาน

End



ภาคผนวก ง.
ส่วนประกอบของวิซวลเบสิก



แสดงส่วนประกอบของวิซวลเบสิก

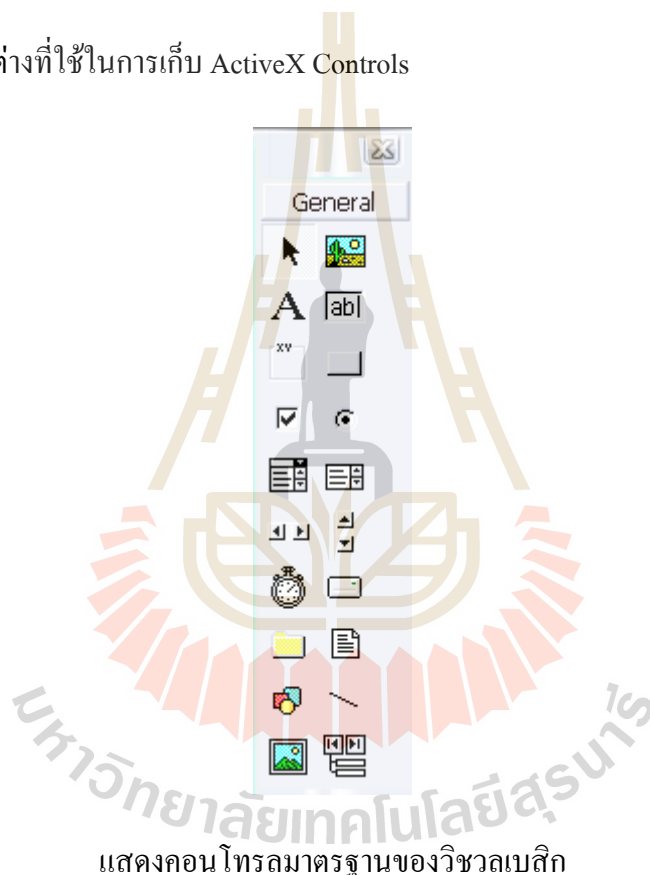
1. Menu Bar เป็นที่รวบรวมคำสั่งเพื่อควบคุมการทำงานของโปรแกรมทั้งหมด
 2. Tool Bar เป็นที่รวบรวมคำสั่งเพื่อควบคุมการทำงานของโปรแกรมในรูปแบบช็อตคัต
 3. Tool Box เป็นที่รวบรวมเครื่องมือมาตรฐานต่างๆ ในการสร้าง Application
 4. Form Designer เป็นหน้าต่างของโปรแกรมที่ต้องการออกแบบ
 5. Code Window เป็นส่วนที่ใช้เขียนคำสั่งควบคุมการทำงานของ Application
 6. Project Window เป็นเครื่องมือที่ใช้ควบคุมการทำงานของ Project
 7. Properties Window เป็นส่วนที่กำหนด Properties ให้กับ object ต่างๆ ใน Application
 8. Form Layout เป็นส่วนที่บอกตำแหน่งคร่าวๆ ของฟอร์มที่ได้จากการรัน Application
- การพัฒนา Application ด้วย ActiveX Controls

ActiveX Controls นั้นทำให้สามารถสร้าง Application ได้อย่างสะดวกและง่ายดาย เพราะเป็นคอนโทรลสำเร็จรูป ที่สามารถ ดึงนำมาใช้งานได้เลย ActiveX Controls ทุกตัวสามารถใช้

งานได้กับ Development Program ที่สนับสนุนการทำงานกับ ActiveX Controls ทุกโปรแกรม เช่น Visual Basic, Visual C++ และ Delphi เป็นต้น




ActiveX Controls จะถูกเก็บไว้ที่หน้าต่าง Tool Box หรือจะกล่าวกลับกันก็คือ Tool Box เป็นหน้าต่างที่ใช้ในการเก็บ ActiveX Controls นั้นเอง ในการเขียนโปรแกรมด้วย Visual basic นั้น จะเรียกใช้งานคอนโทรลชนิดต่างๆ เข้ามาประกอบเป็น Application ซึ่งคอนโทรลชนิดต่างๆ นั้น บางตัวก็เคยใช้งานผ่านโปรแกรมต่างๆ ของ Windows มาแล้ว อาทิเช่น ปุ่มกด, Text Box และ List Box เป็นต้น แต่คอนโทรลบางตัวก็ไม่ได้ปรากฏให้เห็น แต่จะถูกเรียกใช้งานภายหลัง อาทิเช่น Common Dialog, MSChart เป็นต้น








Tool Box เป็นหน้าต่างที่ใช้ในการเก็บ ActiveX Controls



แสดงคอนโทรลมาตรฐานของ Visual Basic

สำหรับคอนโทรลมาตรฐานที่ควรรู้จักได้แก่

ActiveX Controls	ไอคอน	คำอธิบาย
PictureBox		เป็นคอนโทรลที่ใช้อ่านไฟล์รูปภาพมาแสดงบนฟอร์ม
Label		เป็นแถบข้อความ มักใช้เขียนข้อความให้อ่านอย่างเดียว
TextBox		เป็นช่องให้ผู้ใช้งานกรอกข้อความ

Frame		เป็นกรอบที่จัดกลุ่มคอนโทรลต่างๆ ออกเป็นพวกๆ
CommandButton		เป็นปุ่มกดให้ใช้งานกด หรือคลิกที่ปุ่มนี้
CheckBox		เป็นปุ่มให้ใช้งานคลิกเลือก ซึ่งจะเลือกก็ได้
OptionButton		เป็นปุ่มให้ใช้งานคลิกเลือก ซึ่งเลือกได้เพียง 1 ตัวเท่านั้น
ComboBox		เป็นรายการข้อมูลให้ใช้งานเลือก
ListBox		เป็นรายการข้อมูลให้ใช้งานเลือก
HScrollBar		เป็นแถบเลื่อนตามแนวนอน
VScrollBar		เป็นแถบเลื่อนตามแนวตั้ง
Timer		เป็นตัวจับเวลา
DriveListBox		เป็น ActiveX Control ที่ใช้ติดต่อกับฮาร์ดไดรฟ์ หรือซีดีรอม
DirListBox		เป็น ActiveX Control ที่ใช้ติดต่อกับโฟลเดอร์ หรือไดเรกทอรี
FileListBox		เป็น ActiveX Control ที่ใช้ติดต่อกับไฟล์
Shape		ใช้วาดรูปภาพทางเลขาคณิต
Line		เป็นคอนโทรลที่ใช้วาดเส้น
Image		เป็นคอนโทรลที่ใช้อ่านไฟล์รูปภาพมาแสดงบนฟอร์ม
Data Control		เป็นคอนโทรลที่ใช้ติดต่อกับฐานข้อมูล
OLE Control		ใช้สำหรับเรียกใช้งานโปรแกรมอื่นที่ทำงานบนระบบ Windows

ภาคผนวก จ.
SQL server 2000

ชนิดของข้อมูล

ในการสร้างตารางนั้น ส่วนหลักๆก็คือเราจะต้องนิยามคอลัมน์แต่ละคอลัมน์ ซึ่งในการนิยามนั้นเราจำเป็นต้องกำหนดชนิดข้อมูลให้กับคอลัมน์นั้น สำหรับใน SQL Server นั้นจะมีชนิดข้อมูลให้เราเลือกใช้มากมาย เช่น ตัวเลข, ข้อความ, ตัวอักษร นอกจากนี้ สำหรับตารางหนึ่งตาราง เราจะต้องกำหนดอย่างน้อย 1 คอลัมน์ แต่จะมีได้สูงสุดไม่เกิน 1024 คอลัมน์ในหนึ่งตาราง

ข้อมูลชนิดเลขจำนวนเต็ม

ชนิดข้อมูล	คำอธิบาย
TINYINT	เป็นชนิดข้อมูลแบบตัวเลขที่สามารถเก็บข้อมูลได้ในช่วง 0-255 ซึ่งจะใช้เนื้อที่ในการเก็บข้อมูล 1 ไบต์
SMALLINT	เป็นชนิดข้อมูลแบบตัวเลขที่สามารถเก็บข้อมูลได้ในช่วง -32,768 ถึง 32,767 ซึ่งจะใช้เนื้อที่ในการเก็บข้อมูล 2 ไบต์
INT หรือ INTEGER	เป็นชนิดของข้อมูลแบบตัวเลขที่สามารถเก็บข้อมูลได้ตั้งแต่ $-(2^{31})$ ถึง $(2^{31})-1$ ซึ่งจะเป็นตัวเลขในช่วง -2,147,483,648 ถึง 2,147,483,647 ซึ่งจะใช้เนื้อที่ในการเก็บข้อมูล 4 ไบต์
BIGINT	เป็นชนิดของข้อมูลแบบตัวเลขที่สามารถเก็บข้อมูลได้ตั้งแต่ $-(2^{63})$ ถึง $(2^{63})-1$ ซึ่งจะใช้เนื้อที่ในการเก็บข้อมูล 8 ไบต์

ข้อมูลชนิดเลขทศนิยม

ชนิดข้อมูล	คำอธิบาย						
FLOAT[(n)]	<p>เป็นชนิดข้อมูลที่มีค่าอยู่ในช่วงระหว่าง $1.7E-308$ ถึง $1.7E+308$ ซึ่งจะใช้ 8 ไบต์ในการเก็บข้อมูล ถ้าเราไม่ได้กำหนดค่า n ความละเอียดของข้อมูลจะเท่ากับ 15 หลัก นอกจากนี้เรายังสามารถกำหนดค่า n ให้มีค่าเป็น 1 ถึง 53 ได้ซึ่งจะมีความสัมพันธ์กับความละเอียดดังต่อไปนี้</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td>ค่า n</td> <td>ความละเอียด</td> <td>ขนาดที่เก็บ</td> </tr> <tr> <td>1-24</td> <td>7 หลัก</td> <td>4 ไบต์</td> </tr> </table>	ค่า n	ความละเอียด	ขนาดที่เก็บ	1-24	7 หลัก	4 ไบต์
ค่า n	ความละเอียด	ขนาดที่เก็บ					
1-24	7 หลัก	4 ไบต์					

ชนิดข้อมูล	คำอธิบาย
REAL	<p>ค่า n ความละเอียด ขนาดที่เก็บ</p> <p>25-53 15 หลัก 8 ไบต์</p> <p>เป็นชนิดของข้อมูลที่มีช่วงระหว่าง $-3.4 \text{ E} +38$ ถึง $3.4 \text{ E} +38$ ซึ่งจะใช้ 4 ไบต์ในการเก็บข้อมูลทำให้มีความละเอียดถึง 7 หลัก ซึ่งเหมือนกับ FLOAT(24)</p>
DECIMAL(p,s)], NUMERIC(p,s)],	<p>สำหรับชนิดข้อมูลนี้ เราสามารถใช้คำว่า DECIMAL หรือ NUMERIC แทนก็ได้ ซึ่งชนิดข้อมูลนี้จะอนุญาตให้เราสามารถที่จะกำหนดความละเอียดได้ด้วยพารามิเตอร์ p กับ s และสามารถเก็บข้อมูลได้ในช่วง $10^{38}-1$ จนถึง $-10^{38}+1$ โดยใช้ 2-17 ไบต์ในการเก็บข้อมูล ซึ่งค่า</p> <p>-p จะเป็นการกำหนดจำนวนหลักที่สามารถเก็บได้ทั้งซ้ายและขวาของจุดทศนิยม รวมกัน ค่าดีฟอลต์ของ p จะเท่ากับ 18 ค่า p จะมีได้ตั้งแต่ 1-38 หลัก</p> <p>-s จะเป็นการกำหนดจำนวนหลักหลังทศนิยมมี ค่าดีฟอลต์ของ s จะเท่ากับ 0 และจะมีค่าสูงสุดเท่ากับ p</p>

ข้อมูลชนิดข้อความ(Character)

ชนิดข้อมูล	คำอธิบาย
CHAR	<p>เป็นชนิดของข้อมูลแบบข้อความที่แต่ละตัวอักษรจะใช้เนื้อที่ 1 ไบต์ และเราสามารถกำหนดจำนวนอักขระที่เก็บได้ เราสามารถกำหนดให้ข้อมูลนี้เก็บข้อมูลได้ถึง 8,000 ตัวอักษรในแบบ ASCII</p>
VARCHAR	<p>เป็นชนิดข้อมูลที่คล้ายกับ CHAR โดยที่แต่ละตัวอักษรจะใช้เนื้อที่ 1 ไบต์ และเราสามารถกำหนดขนาดของชนิดข้อมูลได้ด้วย เราสามารถกำหนดให้ข้อมูลนี้เก็บข้อมูลได้ถึง 8,000 ตัวอักษรในแบบ ASCII</p>

ข้อมูลประเภท Unicode

ชนิดข้อมูล	คำอธิบาย
NCAR	เป็นชนิดข้อมูลที่คล้ายกับ CHAR โดยที่แต่ละตัวอักษรจะใช้เนื้อที่ 2 ไบต์ เราสามารถกำหนดขนาดของชนิดข้อมูลได้ด้วย และเราสามารถกำหนดให้ข้อมูลนี้เก็บข้อมูลได้ถึง 4,000 ตัวอักษร
NVARCHAR	เป็นชนิดข้อมูลที่คล้ายกับ NCHAR แต่เราสามารถกำหนดให้ข้อมูลชนิดนี้สามารถเก็บข้อมูลได้ถึง 4,000 ตัวอักษร และถ้าเราใส่ตัวอักษรน้อยกว่าที่เรากำหนด SQL server จะไม่ใส่ช่องว่างลงไป

ข้อมูลชนิดไบนารี

ชนิดข้อมูล	คำอธิบาย
BINARY(n)	เราสามารถใช้ชนิดข้อมูลนี้ในการเก็บข้อมูลไบนารีได้ถึง 8,000 ไบต์ และเราจะต้องกำหนดค่า n เป็นขนาดไบต์ที่ต้องการจาก 1- 8000 และหากข้อมูลที่เราเก็บที่ขนาดเกินจากที่เรากำหนด ก็จะถูกตัดตอนให้เหลือ n ไบต์แรกเท่านั้น และสำหรับข้อมูลที่มีขนาดน้อยกว่าที่กำหนด SQL server ก็จะใส่ช่องว่างเพิ่มลงไปจนครบจำนวน
VARBINARY(n)	เราสามารถใช้ชนิดข้อมูลนี้ในการเก็บข้อมูลได้ถึง 8,000 ไบต์และเราจะต้องกำหนดค่า n เป็นขนาดไบต์ที่ต้องการจาก 1 ถึง 8,000 แต่จะแตกต่างกับข้อมูลชนิด BINARY ตรงที่ข้อมูลจะถูกเก็บจริงตามจำนวนไบต์ที่รับเข้ามาหากมีขนาดน้อยกว่าจำนวนที่กำหนดไว้ ก็จะไม่มีการเพิ่มช่องว่างให้ครบตามจำนวนแต่อย่างไร

ข้อมูลชนิดเลขทศนิยมที่ใช้ในด้านการเงิน

ชนิดข้อมูล	คำอธิบาย
MONEY	จะใช้ในการเก็บข้อมูลที่เกี่ยวข้องกับจำนวนเงิน โดยที่ข้อมูลชนิดนี้จะเก็บเป็นส่วนหนึ่งของจำนวนเต็ม 8 ไบต์ และส่วนของทศนิยม 8 ไบต์ รวมเป็น 16 ไบต์ ซึ่งจะสามารถเก็บข้อมูลได้ในช่วง - 922,337,203,685,477.5808 ถึง 922,337,203,685,477.5807
SMALLMONEY	จะใช้ในการเก็บข้อมูลที่เกี่ยวข้องกับจำนวนเงินเช่นเดียวกับชนิด MONEY โดยที่ความแม่นยำมากถึง 0.1 ถึง 0.001

	โดยที่ข้อมูลชนิดนี้จะเก็บเป็นส่วนองจำนวนเต็ม 4 ไบต์ และส่วนของทศนิยม 4 ไบต์ รวมเป็น 8 ไบต์ ซึ่งจะสามารถเก็บข้อมูลได้ในช่วง - 214,748.3648 ถึง 214,748.3647
--	--

ชนิดข้อมูลชนิดพิเศษ




















ชนิดข้อมูล	คำอธิบาย
BIT	เราสามารถใช้งานข้อมูลนี้ในการเก็บข้อมูลที่แทนได้ด้วยสถานะเพียง 2 สถานะ คือ 0 หรือ 1 ซึ่งอาจแทนค่าได้ด้วย True หรือ False On หรือ Off ก็ได้ โดยถ้าในตารางหนึ่งมีคอลัมน์ที่ใช้ข้อมูลชนิด Bit นี้ น้อยกว่าหรือเท่ากับ 8 คอลัมน์ มันก็จะเก็บในขนาด 1 ไบต์ แต่ถ้าอยู่ระหว่าง 9 ถึง 16 คอลัมน์ มันก็จะใช้ 2 ไบต์ ในการเก็บข้อมูล จะเป็นอย่างไรไปเรื่อยๆเมื่อนับครบ 8 บิต นั่นเอง
TIMESTAMP	สำหรับคอลัมน์ที่ใช้ชนิดข้อมูลนี้ เมื่อเราเพิ่มเรคอร์ดใหม่เข้าไป ก็จะมีค่าใหม่เพิ่มเข้าไปในคอลัมน์ TIMESTAMP โดยอัตโนมัติ เราจะมาสามารถกำหนดค่าในคอลัมน์นี้ได้ด้วยตัวเอง โดยค่าที่ใส่เข้าไปจะเป็นค่าที่เก็บเป็นชนิดข้อมูล VARBINARY(8) และยังสามารถเป็นค่า NULL ได้ด้วย
Sql_variant	ชนิดข้อมูลนี้สามารถเก็บค่าของชนิดข้อมูลพื้นฐานที่เราได้กล่าวมาแล้วเกือบทุกชนิด ชนิดข้อมูลนี้เป็นชนิดข้อมูลใหม่ที่เพิ่มเข้ามาใน SQL server 2000



มหาวิทยาลัยเทคโนโลยีสุรนารี



ทูลบาร์ ที่ใช้ควบคุมการทำงานของ Query Analyzer

ปุ่ม	คำอธิบาย
                  	<p>เปิดหน้าต่างคำสั่ง Transact-SQL ขึ้นมาใหม่</p> <p>เปิดไฟล์ที่ใช้เก็บคำสั่ง Transact-SQL</p> <p>บันทึกคำสั่ง Transact-SQL ลงไฟล์</p> <p>แทรกเทมเพลตของคำสั่ง Transact-SQL ลงไป</p> <p>ใช้ในการตัดข้อความที่เลือกไว้ลงในคลิปบอร์ด</p> <p>ใช้ในการคัดลอกข้อความที่เลือกไว้ลงในคลิปบอร์ด</p> <p>ใช้ในการวางข้อมูลจากคลิปบอร์ดลงหน้าต่างคำสั่ง</p> <p>ลบคำสั่ง Transact-SQL ทั้งหมดในหน้าต่างคำสั่ง</p> <p>ใช้ในการหาคำหรือข้อความที่ต้องการ</p> <p>ทำการ UNDO</p> <p>เลือกรูปแบบในการแสดงผลข้อมูลในหน้าต่างแสดงผล</p> <p>ตรวจสอบไวยากรณ์ของคำสั่ง Transact-SQL</p> <p>ส่งคำสั่ง Transact-SQL ไปประมวลผลที่เซิร์ฟเวอร์</p> <p>หยุดการประมวลคำสั่ง Transact-SQL แล้วส่งผลลัพธ์เท่าที่ประมวลผลเสร็จกลับคืนมา</p> <p>เลือกฐานข้อมูลที่ต้องการจัดการ</p> <p>กำหนดให้มีการแสดง Execution Plan ขึ้นมา</p> <p>เรียก Object Browser ขึ้นมาใช้งาน</p> <p>เรียก Object Search ขึ้นมาใช้งาน</p> <p>ทำการเรียกไดอะล็อกซ์ Current Connection Properties ขึ้นมา</p> <p>เรียกหน้าต่างแสดงผลขึ้นมาเพื่อดูผลลัพธ์ของคำสั่ง Transact-SQL</p>