

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาเครื่องตรวจนับการใช้โทรศัพท์

DEVELOPMENT OF TELEPHONE COUNTER



นายจำลอง เลือจุ่น 43015766

นายสิงหา พันภัยพาด 43015790

เลขหมู่.....
เลขทะเบียน..... 50278
วัน,เดือน,ปี 28 เม.ย. 2547

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

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ปีการศึกษา 2545

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๒๕๔๕

หัวข้อปริญญานิพนธ์	การพัฒนาเครื่องตรวจนับการใช้โทรศัพท์		
ชื่อนักศึกษา	นายจำลอง	เสื่อจุ่น	รหัส 43015766
	นายสิงหา	พັນภัยพาล	รหัส 43015790
อาจารย์ที่ปรึกษา	อาจารย์สุธีรา	พินธุ์ธีรานุกฤษ์	
	อาจารย์ณภพินทุ์	อนันตรศิริชัย	
ภาควิชา	วิศวกรรมสารสนเทศ		
ปีการศึกษา	2545		

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาหลักการทํางานของเครื่องโทรศัพท์และมาตรฐานของสัญญาณในระบบโทรศัพท์เพื่อนำมาประยุกต์สร้าง และปรับปรุงเครื่องตรวจนับการใช้โทรศัพท์ คุณลักษณะที่สำคัญของโครงการนี้ คือ กำจัดสัญญาณเสียงของคํานโทรออกในขณะที่ยังไม่มีกรับสายที่ปลายทาง และทำการบันทึกจำนวนการใช้โทรศัพท์พร้อมรายละเอียดข้อมูลเกี่ยวกับการใช้โทรศัพท์ผ่านระบบคอมพิวเตอร์ โดยสามารถแสดงผลทางจอภาพ และเครื่องพิมพ์

เครื่องตรวจนับการใช้งานโทรศัพท์นี้ จะมีไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวควบคุมการทำงาน โดยรับสัญญาณสถานะขงหู วางหู สัญญาณการกดหมายเลข สัญญาณเรียกกลับ สัญญาณสายไม่ว่าง สัญญาณถอดรหัส DTMF และสัญญาณการถอดรหัสคีย์เพื่อการประมวลผล และแสดงออกทางจอ

ภาคแสดงผลบนจอ จะแสดง วันที่ เวลา และจำนวนครั้งในการเรียกออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Development Telephone counter		
Name	Mr. Jumlong	Suejoon	ID 43015766
	Mr. Singha	Pronpriprn	ID 43015790
Project Advisor	Ms. Suthera	Puntheeranurak	
	Ms. Noppin	Anantrasirichai	
Department	INFORMATION ENGINEERING		
Academic Year	2002		

ABSTARCT

This project present the study of telephone and standard signal in telephone system. The purpose of this project is to development and application of telephone counter . The characteristics of this project is the cancellation in voice signal during waiting signal from end terminal. It can be recorded the dial numbers and the detail of using telephone on computer system with displayed on screen and print out.

This telephone counter used microcontroller MCS-51 to control the operation system such as hand-check signal, dial tone, ringing tone, DTMF decoder signal and decoder signal for processing and displayed on screen.

The results from telephone counter will be displayed on monitor such as date, time and number of calling to end terminal.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากการอนุเคราะห์ เสนอแนะข้อมูล และให้คำปรึกษาที่ดียิ่งจากอาจารย์ที่ปรึกษาปริญญาบัตร คือ อาจารย์สุธีรา พันธุ์ธีรานุรักษ์ และ อาจารย์ นภพินท์ อนันตรศิริชัย จึงขอขอบพระคุณทุกท่านไว้ ณ ที่นี้ ที่ได้ให้ความช่วยเหลือ และอำนวยความสะดวกในการจัดทำโครงการให้สำเร็จได้ด้วยดี



คณะผู้จัดทำ

นาย จำลอง เสือจุ่น

นาย สิงหา พันภัยพาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
บทที่ 1 บทนำ	
1.1 บทนำ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญาานิพนธ์	2
บทที่ 2 ระบบชุมสายโทรศัพท์	
บทที่ 3 ระบบควบคุม	
3.1 ไมโครคอนโทรลเลอร์ตระกูล MCS 51	10
3.2 การขยายพอร์ตอินพุทหรือเอาต์พุทของ ไมโครคอนโทรลเลอร์ กับ IC 8255	18
3.3 การเชื่อมต่อ IC 8255 กับ MCS 51	22
3.4 ภาครตรวจจับสัญญาณเสียง	23
3.5 วงจรถอดรหัสเลขหมาย (DTMF decoder)	25
3.6 วงจรถอดรหัสคีย์แพค	32
3.7 การขับ LCD โมดูลแสดงผลแบบผลึกเหลว	34
3.8 การสื่อสารอนุกรม	47
3.9 การรับส่งข้อมูลแบบอนุกรม	69
3.10 อัตราค่าบริการโทรศัพท์ต่างจังหวัด	84
3.11 อัตราค่าบริการโทรศัพท์ภายในพื้นที่	87
บทที่ 4 โปรแกรมใช้งาน	
4.1 ส่วนแสดงผลทาง Visual Basic	88
4.2 โปรแกรมในการคอนโทรล Visual Basic - From 1	94
4.3 โปรแกรมในการคอนโทรล Visual Basic - From 2	104
4.4 โปรแกรมในการคอนโทรล Visual Basic - From 3	106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 การออกแบบและการสร้าง

5.1 ส่วนวงจรดีเทคเตอร์	108
5.2 การออกแบบวงจรตรวจจับสัญญาณเสียง	109
5.3 การออกแบบวงจรถอดรหัส DTMF	111
5.4 การออกแบบวงจรถอดรหัสสี่เหลี่ยม	112
5.5 ส่วนควบคุมการทำงาน	112
5.6 การออกแบบส่วนซอฟต์แวร์	113
5.7 โปรแกรมประยุกต์	116
5.8 ส่วนแสดงผล	116
5.9 การทำงานของวงจรรวม	116

บทที่ 6 การทดลองและผลการทดลอง

6.1 วงจรตรวจสอบสถานะขงหนู วางหนู	117
6.2 วงจรตรวจสอบสัญญาณเรียกกลับ	117
6.3 สัญญาณการตรวจสอบหมายเลข	117
6.4 สัญญาณสายไม่ว่าง	117
6.5 วงจร Decode สัญญาณ DTMF	120
6.6 วงจรถอดรหัสการกดคีย์	120
6.7 การทดลองการแสดงผลของ Visual Basic	121

บทที่ 7 บทสรุป ปัญหาและข้อเสนอแนะ

บรรณานุกรม	127
------------	-----

ภาคผนวก

ภาคผนวก ก	วงจรรวมและแผ่นปริ๊น
ภาคผนวก ข	SOURCE CODE ของ VISUAL BASIC
ภาคผนวก ค	SOURCE CODE ของ MICROCONTROLLER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 การเชื่อมต่อระหว่างขุมสายโทรศัพท์ที่ใช้ในการติดต่อระหว่างเครื่องโทรศัพท์ผู้ใช้ 2 ประเภท	3
รูปที่ 2.2 แสดงบล็อกไดอะแกรมของเครื่องรับโทรศัพท์	5
รูปที่ 2.3 แสดงเป็นกคหมายเลขของเครื่องโทรศัพท์แบบกดปุ่มและค่าความถี่ที่ใช้	7
รูปที่ 3.1 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051	11
รูปที่ 3.2 โครงสร้างของ MCS-8051	12
รูปที่ 3.3 แสดงโครงสร้างภายในพอร์ตอินพุตเอาต์พุต MCS-8051	14
รูปที่ 3.4 แสดงหน่วยความจำโปรแกรมของ MCS-8051	15
รูปที่ 3.5 แสดงหน่วยความจำข้อมูลของ MCS-8051	16
รูปที่ 3.6 แสดงหน่วยความจำข้อมูลภายใน	17
รูปที่ 3.7 แสดงรายละเอียดของรีจิสเตอร์ฟังก์ชันพิเศษ	18
รูปที่ 3.8 แสดงขาสัญญาณต่างๆและโครงสร้างภายใน 8255	21
รูปที่ 3.9 แสดงความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ 8255	24
รูปที่ 3.10 ลักษณะของสัญญาณ โทรศัพท์	24
รูปที่ 3.11 โครงสร้างภายในไอซี XR-2211	25
รูปที่ 3.12 วงจรถอดรหัสเลขหมาย	26
รูปที่ 3.13 แสดงความถี่ที่ได้จากภาคกรองความถี่	28
รูปที่ 3.14 แสดงแสดงการต่อวงจรควบคุมสัญญาณเบื้องต้นของไอซี MT-8870	30
รูปที่ 3.15 การต่อวงจรภาคอินพุตของไอซี MT-8870	30
รูปที่ 3.16 การต่อวงจรผลิตความถี่	31
รูปที่ 3.17 ไดอะแกรมสัญญาณต่างๆของไอซี MT-8870	32
รูปที่ 3.18 โครงสร้างภายในไอซี 74C922	33
รูปที่ 3.19 รูปร่างการจัดขาสัญญาณของ LCD โมดูล	35
รูปที่ 3.20 ไดอะแกรมการทำงานของ LCD โมดูลแบบอักขระ	37
รูปที่ 3.21 รูปแบบอย่างง่ายสุดของข้อมูลแบบอนุกรม	47

รูปที่ 3.22	รูปแบบอย่างง่ายสุดของข้อมูลแบบอนุกรมแบบ อะซิงโครนัส	48
รูปที่ 3.23	การจัดการขาคอนเนคเตอร์ RS- 232 แบบ DB9,DB10	51
รูปที่ 3.24	การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของ คอมพิวเตอร์ในลักษณะต่างๆ	53
รูปที่ 3.25	ไดอะแกรมการทำงานภายในของพอร์ตอนุกรมเครื่อง คอมพิวเตอร์	56
รูปที่ 3.26	ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ต อนุกรม	64
รูปที่ 3.27	หน้าต่างโปรแกรม DEBUG แสดงตำแหน่งของพอร์ต อนุกรม	66
รูปที่ 3.28	แสดงตำแหน่งของพอร์ตอนุกรมบนวินโดวส์ 95/98	67
รูปที่ 4.1	หน้าจอสำหรับเปิด Project ใหม่	88
รูปที่ 4.2	แสดงจอภาพของ Visual Basic	89
รูปที่ 4.3	แสดงการเลือก Control ต่าง ๆ ที่ต้องการ	90
รูปที่ 4.4	แถบเครื่องมือที่ประกอบ Icon ต่าง ๆ	90
รูปที่ 4.5	แสดงตัวอย่างการเขียน Form และ Control มาใช้งาน	93
รูปที่ 4.6	แสดงรูป Form 1	94
รูปที่ 4.7	แสดงรูป Form 2	104
รูปที่ 4.8	แสดงรูป Form 3	106
รูปที่ 5.1	บล็อกไดอะแกรมแสดงส่วนประกอบของโครงการ	108
รูปที่ 5.2	วงจรตรวจสอบการยกหูโทรศัพท์	108
รูปที่ 5.3	วงจรตรวจจับสัญญาณเสียง	110
รูปที่ 5.4	แสดงวงจร Decode DTMF	111
รูปที่ 5.5	แสดงวงจร Scan keypads	112
รูปที่ 5.6	แสดงบล็อกไดอะแกรมของไมโครคอนโทรลเลอร์	112
รูปที่ 5.7	รูปแสดงการควบคุมโดยใช้ไมโครคอนโทรลเลอร์เมนู	114
รูปที่ 5.8	รูปแสดงการควบคุมโดยใช้ไมโครคอนโทรลเลอร์	114
รูปที่ 6.1	แสดงรูปสัญญาณที่ได้จากวงจรตรวจสอบการยกหู	118
รูปที่ 6.2	แสดงรูปสัญญาณเรียกกลับ	118

เอกสารนี้เป็นรูปที่ 6.3 แสดงสัญญาณการกดหมายเลขการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีรูปที่ 6.4 แสดงสัญญาณสายไม่ว่างเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.5 แสดงผลการทดลองของ From 1	122
รูปที่ 6.6 แสดงผลการทดลองของ From 2	123
รูปที่ 6.7 แสดงผลการทดลองของ From 3	124
รูปที่ 6.8 แสดงผลการพิมพ์ของ From 1	125



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 1 แสดงหน้าที่พิเศษของพอร์ต 3	13
ตารางที่ 2 การกำหนดรูปแบบการทำงานของพอร์ต 8255	19
ตารางที่ 3 แสดงหน้าที่การทำงานของขาสัญญาณไอซี 8255	20
ตารางที่ 4 ระบบรีจิสเตอร์ของ 8255	22
ตารางที่ 5 แสดงข้อมูลทางขาสัญญาณ	22
ตารางที่ 6 ตำแหน่งของรีจิสเตอร์	23
ตารางที่ 7 แสดงการรหัสที่ได้จากควมถี่ต่างๆ	28-29
ตารางที่ 8 แสดงขาสัญญาณของแอลซีดีโมดูล	36
ตารางที่ 9 ตารางชุดคำสั่ง	39
ตารางที่ 10 แสดงกำหนดค่า Clear display	41
ตารางที่ 11 แสดงการกำหนดค่า DDRAM	41
ตารางที่ 12 แสดงการกำหนดค่าโหมดเซต	42
ตารางที่ 13 แสดงการกำหนดค่า ON/OFF	42
ตารางที่ 14 แสดงการกำหนดค่าการเลื่อน	43
ตารางที่ 15 แสดงการกำหนดค่าเซตฟังก์ชัน	43
ตารางที่ 16 แสดงการกำหนดค่า CGRAM	44
ตารางที่ 17 แสดงการกำหนดตำแหน่งของ DDRAM	44
ตารางที่ 18 แสดงการกำหนดตำแหน่งของ DDRAM ที่ใช้งาน	45
ตารางที่ 19 แสดงการกำหนดค่า	45
ตารางที่ 20 แสดงการเขียนข้อมูลลง DDRAM และ CGRAM	45
ตารางที่ 21 แสดงการอ่านข้อมูลลง DDRAM และ CGRAM	46
ตารางที่ 22 แสดงบิตพาร์ตีข้อมูล	49
ตารางที่ 23 แสดงรีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์	57
ตารางที่ 24 แสดงรีจิสเตอร์โหมดและสถานะการอินเตอร์รัปต์	57

เอกสารนี้เป็นตารางที่ 25 แสดงรีจิสเตอร์กำหนดรูปแบบของข้อมูลนั้น ไม่อนุญาตให้นำไปใช้ 59 ยืนยันด้านการค้า
ไม่ว่ากรณีใดตารางที่ 26 แสดงรีจิสเตอร์ควบคุมไม่เต็ม จะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง 60 มีการนำไปใช้

ตารางที่ 27 แสดงรีจิสเตอร์สถานะการรับและการส่งข้อมูล	61
ตารางที่ 28 แสดงรีจิสเตอร์สถานะของโมเด็ม	62
ตารางที่ 29 แสดงข้อมูลในแอดเดรส 0000H : 0411H ที่ใช้แจ้งจำนวนพอร์ต	65
ตารางที่ 30 แสดงรายละเอียดของรีจิสเตอร์ AL	68
ตารางที่ 31 แสดงมาตรฐานการกำหนดคิพพริต RS-232	72
ตารางที่ 32 แสดงค่าคงที่คุณสมบัติ Handshake	80
ตารางที่ 33 แสดงค่าคงที่สำหรับคุณสมบัติ OnComm	81
ตารางที่ 34 ค่าคงที่สำหรับคุณสมบัติ Error	81
ตารางที่ 35 แสดงค่าคงที่สำหรับคุณสมบัติ InputMode	82
ตารางที่ 36 แสดงการอ่านค่าลอจิกจากพอร์ตอนุกรม RS-232	84
ตารางที่ 37 แสดงอัตราค่าบริการโทรศัพท์ทางไกล	85-87
ตารางที่ 38 แสดงรายละเอียดการใช้งาน Toolbar ต่าง ๆ	91-92
ตารางที่ 39 แสดงการทดสอบการถอดรหัสเลขหมายโดยใช้ไอซี MT8870	120
ตารางที่ 40 แสดงเอาต์พุตที่ได้จากการทดสอบวงจรถอดรหัสการกดคีย์	121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 บทนำ

เนื่องจากปัจจุบันเป็นยุคแห่งการติดต่อสื่อสาร ซึ่งโทรศัพท์เป็นสื่อสารชนิดหนึ่งที่มีความสำคัญมากต่อมนุษย์เราในการติดต่อสื่อสารถึงกัน การใช้โทรศัพท์แต่ละครั้งที่มีการโทรออกไปยังหมายเลขอื่นจะมีการบันทึกการโทรออกเป็นจำนวนครั้ง ซึ่งกระทำโดยของจักรโทรศัพท์ ปัญหาที่เกิดขึ้นก็คือผู้ใช้ไม่สามารถทราบความถูกต้องของการนับจำนวนครั้งการโทรออกได้เลย จากเหตุการณ์ดังกล่าวจึงมีแนวคิดในการสร้างเครื่องนับการใช้งานโทรศัพท์ขึ้น

เครื่องดังกล่าวได้มีการสร้างและทดลองใช้งานมาแล้วครั้งหนึ่ง ซึ่งเป็นโครงการของปีการศึกษา 2544 สำหรับโครงการที่จัดทำครั้งนี้ จะเป็นการศึกษาและปรับปรุงแก้ไขจากโครงการปีการศึกษา 2544 โดยจะทำการปรับปรุงแก้ไข 2 เรื่อง คือ

1. การกำจัดสัญญาณเสียงพูดของผู้โทรออกหรือเสียงอื่น ๆ ที่อยู่ใกล้เคียงที่อาจเล็ดลอดเข้าไปในโทรศัพท์ในขณะที่ยังไม่มีการรับสายปลายทาง ซึ่งจะเป็นการป้องกันโดยไม่ให้อัตโนมัติเป็นการนับจำนวนครั้งของการใช้โทรศัพท์
2. การบันทึกข้อมูลการใช้โทรศัพท์ลงในคอมพิวเตอร์ โดยที่จะต้องสามารถแสดงผลลัพธ์ทางจอภาพและพิมพ์ออกทางเครื่องพิมพ์

1.2 วัตถุประสงค์

1. เพื่อศึกษาการทำงานของ MCS-51 (Microcontroller)
2. เพื่อศึกษาวงจรตรวจจับสัญญาณเสียงโทรศัพท์
3. เพื่อศึกษาวงจรถอดรหัส DTMF (Dual Tone Muti Frequency dailing)
4. เพื่อศึกษาการส่งสัญญาณโทรศัพท์แบบต่าง ๆ
5. เพื่อศึกษาการส่งข้อมูลระหว่าง MCS-51 (Microcontroller) มายังคอมพิวเตอร์
6. เพื่อศึกษาการส่งผ่านข้อมูลทางพอร์ตอนุกรม RS-232
7. เพื่อศึกษาการทำงานของโปรแกรม Visual Basic ซึ่งเป็นโปรแกรมภาษาที่นำมาใช้ในการแสดงผล

ผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า - ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

เพื่อทำการกำจัดเสียงพูดของผู้โทรออกหรือเสียงอื่น ๆ ที่อยู่ใกล้โทรศัพท์ขณะที่ยังไม่มีการรับสายจากผู้รับ ข้อมูลการใช้โทรศัพท์ได้แก่ วัน เดือน ปี เวลา หมายเลขที่โทรติดต่อ และรวมยอดจำนวนครั้งที่โทรออกในช่วงเวลาที่กำหนด โดยที่สามารถแสดงผลทางเครื่องคอมพิวเตอร์ และพิมพ์ออกทางเครื่องพิมพ์



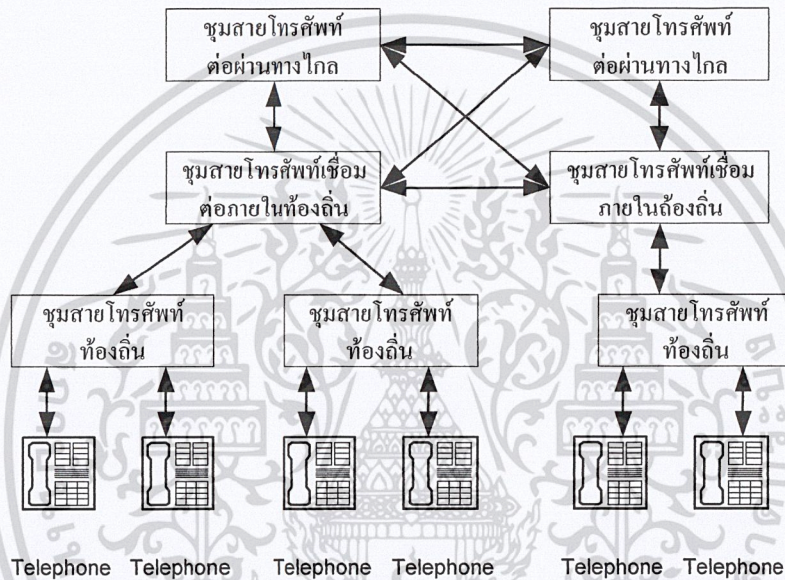
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ระบบชุมสายโทรศัพท์

ชุมสายโทรศัพท์

ในการติดต่อกันระหว่างเครื่องโทรศัพท์ ผู้ใช้จะกระทำผ่านชุมสายโทรศัพท์ต่าง ๆ ที่องค์การโทรศัพท์ให้บริการ รายละเอียดดังรูปที่ 2.1



รูปที่ 2.1 การเชื่อมต่อระหว่างชุมสายโทรศัพท์ที่ใช้ในการติดต่อระหว่างเครื่องโทรศัพท์ผู้ใช้ 2 ประเภท

โดยทั่วไปสามารถแบ่งชุมสายโทรศัพท์ออกเป็น 2 ส่วนใหญ่ ๆ ได้แก่

1. ชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ผู้ใช้ต่อเข้าโดยตรง

ชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ผู้ใช้ต่อเข้าโดยตรง ได้แก่ ชุมสายโทรศัพท์ท้องถิ่น ชุมสายปลายทางอัตโนมัติ

1.1 ชุมสายโทรศัพท์ท้องถิ่น (Local Exchange)

เป็นชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ของผู้เช่าต่อโดยตรง ชุมสายแบบนี้มีขนาดตั้งแต่เป็นร้อย ๆ หมายเลขจนถึงหมื่นหมายเลขหรือมากกว่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ขุมสายปลายทางอัตโนมัติ (PABX)

ขุมสายปลายทางอัตโนมัติ (PABX) ได้ถูกพัฒนามาจากขุมสายโทรศัพท์ปลายทาง (PBX) ซึ่งเป็นอุปกรณ์ที่ให้บริการสำหรับผู้เช่า (Customer Premise Equipment : CPE) ที่ทำการติดตั้งในอาคาร สำนักงาน บริษัท เป็นต้น เพื่อที่จะให้บริการเรียกสำหรับคู่สายสมาชิกได้ทั้งภายในและภายนอกของสำนักงานนั้น ๆ (ภายในขุมสาย) โดยไม่ต้องผ่านขุมสายท้องถิ่น

2. ขุมสายโทรศัพท์ที่ไม่มีเครื่องโทรศัพท์ผู้เช่าต่อเข้าโดยตรง

ขุมสายโทรศัพท์ที่ไม่มีเครื่องโทรศัพท์ผู้เช่าต่อเข้าโดยตรง จะเป็นขุมสายต่อผ่านที่ให้บริการเรียกระหว่างขุมสายโทรศัพท์ท้องถิ่นกับขุมสายโทรศัพท์ท้องถิ่นด้วยกัน ดังนั้นการเรียกระหว่างผู้เช่าโทรศัพท์สองเครื่องอาจเรียกผ่านไปยังขุมสายต่อผ่านหลาย ๆ ขุมสายได้ ซึ่งขุมสายต่อผ่านยังแบ่งออกเป็น 2 แบบ คือ

2.1 ขุมสายโทรศัพท์เชื่อมต่อภายในท้องถิ่น (Tandem Exchange)

เป็นขุมสายโทรศัพท์ต่อผ่านที่ให้บริการเรียกระหว่างขุมสายโทรศัพท์กับขุมสายโทรศัพท์ท้องถิ่นภายในท้องถิ่น

2.2 ขุมสายโทรศัพท์ต่อผ่านทางไกล (Transit Exchange)

เป็นขุมสายโทรศัพท์ที่ให้บริการระหว่างขุมสายโทรศัพท์เชื่อมต่อภายในท้องถิ่นกับโทรศัพท์เชื่อมต่อภายในท้องถิ่น

โครงสร้างพื้นฐานเครื่องโทรศัพท์ผู้ใช้ (Basic structure of subscriber)

1. โครงสร้างบล็อกโคอะแกรมของเครื่องโทรศัพท์

เครื่องโทรศัพท์ทั่ว ๆ ไปจะมีรายละเอียดโครงสร้างตามบล็อกโคอะแกรมดังรูปที่ 2.2 จากรูปสามารถแบ่งโครงสร้างเครื่องโทรศัพท์ออกเป็น 3 ส่วนใหญ่ ๆ ด้วยกัน คือ

ส่วนวงจรกระดิ่ง (Ringer)

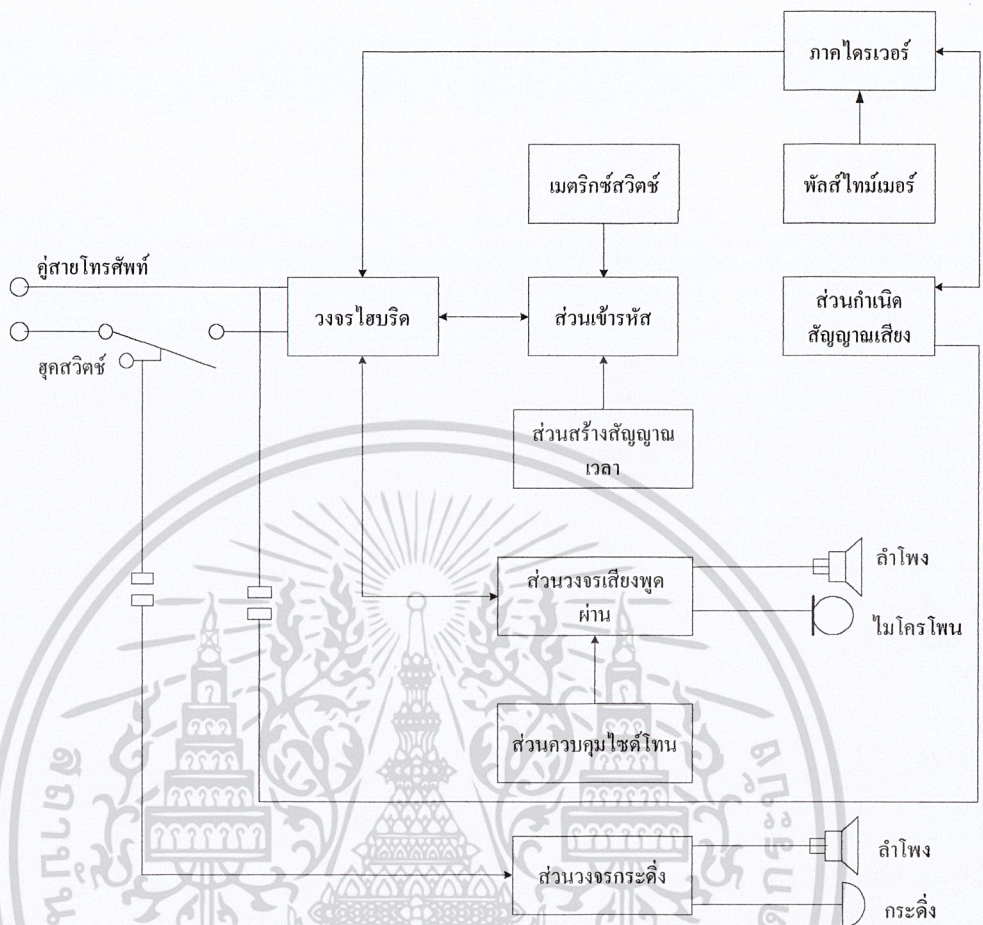
ส่วนวงจรกระดิ่งนี้จะทำหน้าที่เรียกเป็นเสียงกระดิ่งทุกครั้งที่มีการเรียกเข้ามาเพื่อให้เจ้าของทราบว่ามีกรเรียก จะเป็นวงจรกระดิ่งหรือไอซีโทนริงเกอร์ก็ได้

หากใช้กระดิ่ง เมื่อมีสัญญาณเรียกเข้ามาเป็นกระแสไฟสลับประมาณ 100-120 โวลต์ ส่งเข้ามายังโซลินอยด์จะทำให้เกิดสนามแม่เหล็กขึ้นเหมือนกระดิ่งหรือออก (Bell) ทั่ว ๆ ไป

หากเป็นระบบที่เป็นไอซีจะมีการเอาพัลส์ดังกล่าวเข้าสู่ระบบเร็คติไฟเพื่อจ่ายให้หน่วยผลิตความถี่ส่งความถี่ออกมาเป็นสัญญาณเรียกที่ใช้วิธีการสังเคราะห์เสียงในรูปแบบอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงบล็อกไดอะแกรมของเครื่องรับโทรศัพท์

ส่วนวงจรปากพูดและหูฟัง (Speech Network)

ส่วนของวงจรปากพูดและหูฟังนี้จะทำหน้าที่เสมือนวงจรเครื่องรับและเครื่องส่งที่ใช้ในการติดต่อกันระหว่างเครื่องโทรศัพท์ผู้ใช้ เมื่อมีการเรียกจะถูกส่งผ่านระบบชุมสาย จากชุมสายจะมีการเรียกด้วยระบบมัลติเพล็กซ์เพื่อแยกคู่สายไปยังเลขหมายที่ผู้เรียกต้องการติดต่อ ชุมสายก็จะส่งสัญญาณเรียกเป็นแรงดันไฟฟ้ากระแสสลับ 100-120 โวลต์มายังเครื่องผู้รับ ถ้าเครื่องรับที่ออกแบบอย่างถูกต้องจะมีกระแสไหลในวงจรด้วยระบบเคอร์เรนท์เรกูเรเตอร์ (Current Regulator) รักษากระแสคงที่ไว้ที่ 23 มิลลิแอมป์ เมื่อผู้ถูกเรียกยกหูรับสาย ทั้งผู้เรียกและผู้ถูกเรียกสามารถพูดคุยสนทนาด้วยวงจรอินเตอร์คอมหรือวงจรเสียงพูดผ่าน เพียงแต่อยู่ที่วงจรเสียงพูดผ่านต้องมีไซด์โทน (Side Tone) ไม่มากเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรถ่ายรหัสตัวเลขก็ยักด (Encode)

เมื่อต้องการเรียกสายคู่สนทนาต้องขงูโทรศัพท์แล้วกดหมายเลขที่ต้องการจากแป้นกด (Key) สัญญาณจากการกดหมายเลขจะผ่านวงจรถ่ายรหัสข้อมูลความถี่ผสมและจะทำการส่งความถี่ (สูง - ต่ำ) ไปยังระบบชุมสาย แล้วชุมสายจะส่งข้อมูลไปตามเครือข่าย หากส่วนนี้ไม่ทำงาน เครื่องโทรศัพท์ยังกรรับได้แต่ไม่สามารถโทรออกได้

ระบบของเครื่องโทรศัพท์

เครื่องโทรศัพท์ที่ใช้ติดต่อกันโดยผ่านชุมสายโทรศัพท์ มีอยู่ด้วยกัน 2 ระบบ คือ ระบบโทน (Tone) หรือระบบดีทีเอ็มเอฟ (DTMF) กับระบบพัลส์ (Pulse) ซึ่งเป็นระบบดั้งเดิม โดยตัวเครื่องจะมีลักษณะหน้าปัดแบบหมุน จะใช้งานร่วมกับชุมสายท้องถิ่นระบบเก่าได้ หากบริเวณใดเอาเครื่องระบบหมุนไปต่อแล้วไม่สามารถติดต่อกันได้ แสดงว่าชุมสายดังกล่าวเป็นระบบใหม่ ซึ่งจะต้องใช้เครื่องโทรศัพท์แบบกดปุ่มเท่านั้น แต่มีใช้ว่าบริเวณที่ใช้เครื่องแบบกดปุ่มจะต้องเป็นชุมสายระบบใหม่ทั้งหมดเสมอไป เพราะเมื่อหลายปีก่อนที่ผ่านมามีบริษัทเอทีแอนด์ทีได้คิดค้นเครื่องโทรศัพท์ระบบกดปุ่มที่สามารถให้กำเนิดสัญญาณอิมพัลส์ เพื่อให้สามารถใช้กับชุมสายท้องถิ่นระบบเก่าได้ โดยสามารถที่จะใช้งานกับชุมสายโทรศัพท์ทั้งสองชนิดได้ โดยจะมีสวิทช์เพื่อเลือกให้เครื่องทำงานแบบทัชพัลส์ (Touch Pulse) หรือแบบกดปุ่ม (Touch Tone)

1. แบบหมุน (Rotary dial)

เครื่องโทรศัพท์ชนิดนี้จะเป็นระบบพัลส์ โดยจะทำการสร้างสัญญาณจากกระแสดูบ โดยต่อเข้ากับอุปกรณ์สวิทช์และอุปกรณ์สวิทช์นี้จะทำหน้าที่ " เปิด-ปิด " เข้ากับกลไกการหมุนเลขหมายในเครื่อง ทำให้กระแสพัลส์ตอบสนองกับหมายเลขที่หมุน โดยจำนวนพัลส์ที่เกิดขึ้นก็จะมีจำนวนเท่ากับหมายเลขที่หมุน

2. แบบกดปุ่ม (Touch tone)

เครื่องโทรศัพท์แบบนี้คือระบบ โทนหรือระบบดีทีเอ็มเอฟที่มีการส่งความถี่สองความถี่ที่มอดูเลตกันของหมายเลขที่กดออกไปความถี่ที่ส่งออกไปจะอยู่ในย่านของความถี่เสียงพูด โดยมีค่าประมาณ 0 - 4 KHz โดยค่าความถี่ต่าง ๆ ในการกดหมายเลขจะแสดงได้ดังรูปที่ 2.3

ค่าความถี่ที่ใช้ในแต่ละแถวแต่ละคอลัมน์จะมีความถี่ต่างกัน ความถี่ของแต่ละแถวจะเป็นกลุ่มความถี่ต่ำ และความถี่ของแต่ละคอลัมน์จะเป็นกลุ่มความถี่สูง ความถี่ที่ได้จากการกดเลขหมายนี้ก็จะได้จากมอดูเลตของความถี่สูงและความถี่ต่ำของแต่ละหมายเลข เช่น การกดเลข 1

เอกสารจะได้ค่าของกัรรมอดูเลตที่ความถี่ 597 เฮิรต์ และ 1209 เฮิรต์ เป็นต้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของเครื่องโทรศัพท์ที่ใช้ดีทีเอ็มเอฟ

1. สามารถลดเวลาในการหมุนหมายเลขลงได้ ทำให้ติดต่อกับชุมสายได้เร็วขึ้น
2. มีความแม่นยำในการส่งเลขหมาย
3. ประหยัดอุปกรณ์หน่วยความจำที่ใช้ภายในชุมสาย
4. สามารถนำไปเชื่อมต่อกับอุปกรณ์ชุมสายได้อย่างมีประสิทธิภาพ



รูปที่ 2.3 แสดงเป็นกคหมายเลขของเครื่องโทรศัพท์แบบกดปุ่ม และค่าความถี่ที่ใช้

มาตรฐานสัญญาณโทรศัพท์พื้นฐานขององค์การโทรศัพท์

1. สัญญาณให้หมุน (Dial Tond)

เป็นสัญญาณที่ชุมสายโทรศัพท์ส่งไปให้เครื่องโทรศัพท์ จะมีความถี่ต่อเนื่องประมาณ 400 เฮิรต์ เพื่อแสดงให้ผู้อื่นรู้ว่าพร้อมรับหมายเลขโทรศัพท์ปลายทางแล้ว

2. สัญญาณหมายเลข

เป็นสัญญาณหมายเลขที่ต้องการติดต่อ โดยเครื่องโทรศัพท์จะส่งสัญญาณนี้ไปชุมสายซึ่งมีอยู่สองแบบคือ สัญญาณพัลส์ (สำหรับเครื่องรับโทรศัพท์แบบหมุน) และสัญญาณความถี่ผสม (สำหรับเครื่องโทรศัพท์แบบกดปุ่ม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องโทรศัพท์แบบพัลส์ (Pulse)

เครื่องโทรศัพท์แบบหมุนจะผลิตสัญญาณพัลส์เป็นช่วง ๆ ตามจำนวนหมายเลขที่ต้องการเรียก โดยมีการผลิตความถี่ด้วยออสซิลเลเตอร์ ในกรณีที่เป็นเครื่องแบบกดปุ่ม

สัญญาณความถี่ผสม (Dual Tone Multi Frequency dialing : DTMF)

เครื่องโทรศัพท์แบบกดปุ่มจะผลิตความถี่แบบผสม หรือในบางครั้งเราเรียกว่าระบบทัชโทน (Touch Tone) โดยการส่งหมายเลขใดหมายเลขหนึ่งจะประกอบด้วยโทนเสียง 2 ความถี่ผสมกัน โดยเราจะพิจารณาปุ่มที่กดหรือคีย์แพค การแบ่งความถี่ตามแถวหรือหลัก หากกดหมายเลขหนึ่งจะประกอบด้วยโทนเสียง 2 ความถี่ผสมกัน โดยเราจะพิจารณาปุ่มที่กดคียบอร์ด การแบ่งกลุ่มความถี่ตามแถวหรือหลัก หากกดหมายเลข 9 ซึ่งอยู่แถวที่ 3 ความถี่ 825 เฮิร์ต และหลักที่ 3 ความถี่ 1447 เฮิร์ต วงจรภายในเครื่องจะทำการเข้ารหัสข้อมูลที่ได้จากนั้นจะส่งความถี่เพื่อให้วงจรผลิตสัญญาณความถี่ผสม ระหว่างความถี่แถวและหลัก แล้วส่งออกไปยังคู่สายโทรศัพท์

3. สัญญาณไม่ว่าง (busy Tone)

เป็นสัญญาณที่ชุมสายส่งต่อไปยังเครื่องโทรศัพท์เพื่อบอกให้ผู้เรียกทราบว่า สายไม่ว่าง ผู้เรียกควรวางหูสักรุ่นแล้วค่อยโทรเรียกใหม่ สัญญาณไม่ว่างจะมีความถี่ 400 เฮิร์ต โดยจะเป็นเสียงดังเป็นเวลาประมาณ 0.5 วินาทีและเงียบ 0.5 วินาที

4. สัญญาณกริ่งเรียกหรือสัญญาณกระดิ่ง (Ringing Tone)

เป็นสัญญาณที่ชุมสายแจ้งให้ผู้ถูกเรียกทราบว่า มีคนโทรเรียกเข้ามา ให้ไปรับสาย สัญญาณกริ่งเรียกนี้เป็นสัญญาณ 25 เฮิร์ต 100 โวลต์พีคทูพีค เสียงนี้จะดังประมาณ 1 วินาที และเสียงเงียบจะประมาณ 4 วินาที โดยชุมสายโทรศัพท์ทั่วไปจะส่งสัญญาณกริ่งนี้ออกไปประมาณ 15 ครั้ง

5. สัญญาณเรียกกลับ (Ring Back Tone)

เป็นสัญญาณที่ชุมสายแจ้งให้ผู้เรียกทราบว่า การเรียกเป็นผลสำเร็จ ให้รอรับสายของผู้รับได้ จะเป็นสัญญาณ 400 เฮิร์ต ส่งเสียงดัง 1 วินาที เสียงเงียบ 4 วินาที รายละเอียดการติดต่อระหว่างเครื่องโทรศัพท์ผู้ใช้ผ่านชุมสายโทรศัพท์

ในการติดต่อกันระหว่างเครื่องโทรศัพท์ผู้ใช้จะมีอยู่ 2 กรณีด้วยกัน คือ การติดต่อกันเมื่อเครื่องโทรศัพท์ผู้ใช้ที่อยู่ภายในชุมสายโทรศัพท์เดียวกันและการติดต่อเมื่อเครื่องรับโทรศัพท์ผู้ใช้ที่อยู่ต่างชุมสายกัน

1. การติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ที่อยู่ภายในชุมสายโทรศัพท์เดียวกัน

กรณีที่ในการติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ที่อยู่ภายในชุมสายโทรศัพท์เดียวกัน ด้านของผู้เรียก (Calling Subscriber) เมื่อยังไม่ยกหูจะมีแรงไฟตรงที่คู่สายประมาณ 48 โวลต์ เมื่อผู้เรียกยกหู (off-hook) เพื่อทำการโทรเรียกไปยังผู้รับ จะทำให้ระดับแรงไฟตรงที่คู่สายโทรศัพท์เปลี่ยน

แปลงระดับไปจาก 48 โวลต์ไปเป็น 10 โวลต์ ชุมสายโทรศัพท์จะรับรู้ว่าเป็นการเริ่มต้นการเรียก ฉะนั้นชุมสายโทรศัพท์ก็จะส่งสัญญาณให้หมุนไปยังผู้เรียก (Calling Subscriber) จากนั้นวงจรภายใน เครื่องโทรศัพท์ก็จะสร้างสัญญาณรหัสหมายเลขส่งไปยังชุมสายโทรศัพท์ ชุมสายโทรศัพท์จะตัด สัญญาณให้หมุนออกทันทีที่ได้รับหมายเลขตัวแรก

เมื่อชุมสายโทรศัพท์รับหมายเลขแล้ว ผู้ถูกเรียกก็จะทำการจองทางผ่านระหว่างผู้เรียกและผู้ ถูกเรียก รวมทั้งส่งสัญญาณกริ่งไปยังผู้ถูกเรียกพร้อมทั้งส่งสัญญาณเรียกกลับไปยังผู้เรียก หากคู่ สายทางด้านผู้ถูกเรียกไม่ว่าง ชุมสายจะส่งสัญญาณไม่ว่างไปให้ผู้เรียกทราบ หากคู่สายด้านผู้รับ ว่างชุมสายจะส่งสัญญาณกริ่งเรียกไปยังผู้ถูกเรียกและสัญญาณเรียกกลับไปยังด้านผู้เรียก เมื่อผู้เรียก ได้ยินเสียงกริ่งแล้วยกหูรับสาย จะทำให้ระดับไฟตรงที่คู่สายของผู้ถูกเรียกเปลี่ยนจาก 48 โวลต์ เป็น 10 โวลต์ และจะเกิดสัญญาณตอบรับส่งไปยังชุมสาย ชุมสายก็จะตัดสัญญาณกริ่งเรียกด้านผู้ถูก เรียก และยกเลิกสัญญาณเรียกกลับด้านผู้เรียก ทำให้ทางผ่านระหว่างทรงขาออก (Outing Trunk) ของผู้เรียกและทรงขาเข้า (Incoming Trunk) ของผู้รับว่าง การสนทนาจึงจะสามารถเริ่มต้นได้

เมื่อการสนทนาสิ้นสุดลง คู่สนทนาจึงทำการวางหู ทำให้ชุมสายรับรู้ว่าเลิกการติดต่อ ด้าน ผู้รับเมื่อมีการเรียกจากภายนอกชุมสายซึ่งจะเป็นการส่งสัญญาณกริ่งเรียก สัญญาณกริ่งเรียกนี้จะดัง อยู่ประมาณ 15 ครั้ง จากนั้นชุมสายก็จะทำการตัดสัญญาณเองหากไม่มีการตอบรับ หรือถ้าด้านผู้ รับไม่ว่างก็จะมี การส่งสัญญาณไม่ว่างไปยังผู้เรียกเพื่อให้ทำการเรียกใหม่

2. การติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้อยู่ต่างชุมสายกัน

ในการติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้อยู่ต่างชุมสายกันนั้น การติดต่อด้านผู้เรียกจากชุมสาย ท้องถิ่นไปยังชุมสายเชื่อมต่อภายในท้องถิ่น ด้านผู้เรียกจะแตกต่างกับการติดต่อภายในชุมสายเดียว กัน คือ เมื่อชุมสายท้องถิ่นได้รับสัญญาณเลขหมายปลายทางก็จะส่งสัญญาณยกหูไปยังชุมสายเชื่อม ต่อภายในท้องถิ่น จากนั้นชุมสายเชื่อมต่อภายในท้องถิ่นจะส่งสัญญาณให้หมุน เพื่อให้ชุมสายท้อง ถิ่นส่งเลขหมายปลายทางมาให้ ส่วนขั้นตอนอื่น ๆ จะเหมือนการติดต่อภายในชุมสายเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระบบควบคุม

3.1 ไมโครคอนโทรลเลอร์ตระกูล MCS – 51 (Microcontroller)

ในโครงการที่จัดทำขึ้นนี้จะใช้ไมโครคอนโทรลเลอร์ 8052 แต่เนื่องจากไมโครคอนโทรลเลอร์เบอร์ 8052 และไมโครคอนโทรลเลอร์ 8051 ต่างก็เป็นไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ฉะนั้นจึงมี คุณสมบัติ โครงสร้าง จำนวนพอร์ต และรายละเอียดการใช้งานของพอร์ตเหมือนกัน ดังนั้นในการนำเสนอรายละเอียดของไมโครคอนโทรลเลอร์ 8052 จะนำเสนอรายละเอียดของไมโครคอนโทรลเลอร์ 8051 แทน

ไมโครคอนโทรลเลอร์ 8051 เป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัทอินเทล ถูกผลิตขึ้นมาให้มีคุณสมบัติเหมาะสมกับงานควบคุมระบบต่าง ๆ ในลักษณะที่เรียกว่า ไมโครคอนโทรลเลอร์ชิปเดี่ยว (Single Chip Microcontroller)

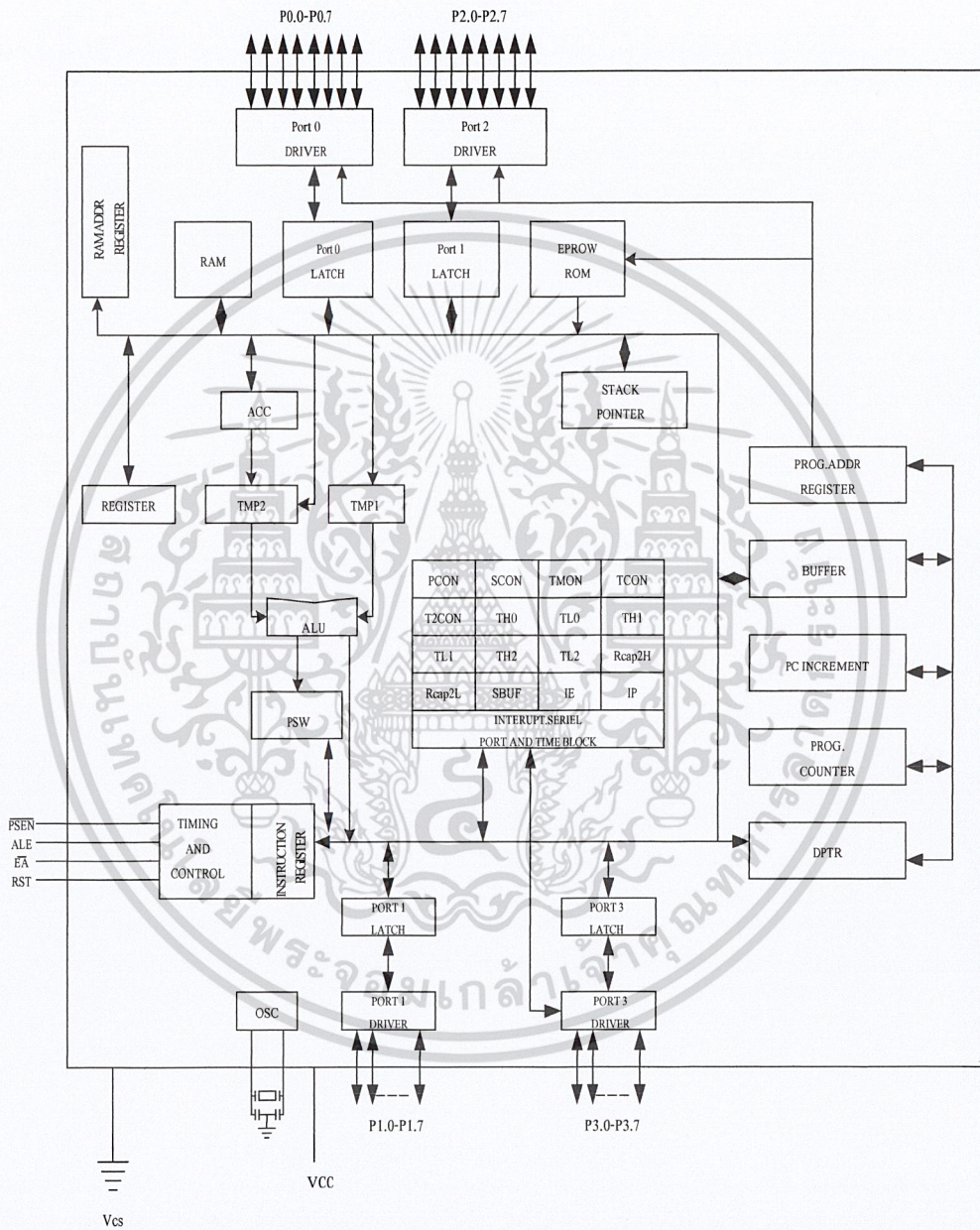
คุณสมบัติของไมโครคอนโทรลเลอร์ 8051

- เป็นไมโครโปรเซสเซอร์ขนาด 8 บิต สำหรับงานควบคุมระบบต่าง ๆ
- มีความสามารถประมวลผลของลอจิกระดับบิต
- มีขนาดหน่วยความจำสำหรับโปรแกรมทำงานได้ถึง 64 กิโลไบต์ (Program Memory)
- มีขนาดหน่วยความจำสำหรับเก็บข้อมูลทำงานได้ถึง 64 กิโลไบต์ (Data Memory)
- มีหน่วยความจำสำหรับโปรแกรมภายในขนาด 4 กิโลไบต์
- มีหน่วยความจำข้อมูลภายในขนาด 128 กิโลไบต์
- มีพอร์ตสำหรับควบคุม 4 พอร์ต สามารถอ้างอิงพอร์ตได้ระดับบิตต่อบิต
- มีชุดไทม์เมอร์ (Timer) ตัวนับ (Counter) ขนาด 16 บิต 2 ชุด
- มีวงจรรีจิสเตอร์อนุกรมแบบฟูลดูเพล็กซ์ (full duplex UART)
- มีโครงสร้างอินเตอร์รัพท์จาก 6 แหล่งกำเนิดสัญญาณและ 5 ตำแหน่งโปรแกรม
- มีวงจรรอสซิงเคลเตอร์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างพื้นฐานไมโครคอนโทรลเลอร์ 8051

โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051 แสดงดังรูปที่ 3.1

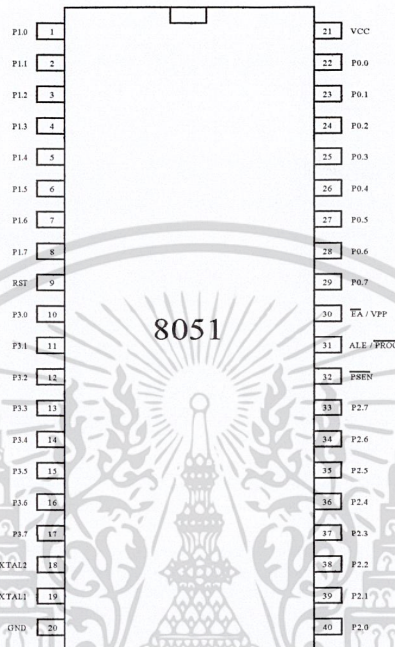


รูปที่ 3.1 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของขาสัญญาณไมโครคอนโทรลเลอร์ตระกูล MCS-51 (Microcontroller)

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกับดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างขาของ 8051

อธิบายได้ดังนี้

VCC (20) ขาแหล่งจ่ายไฟ (+5V)

VSS (40) ขากราวนด

P0 (32-39) เป็นขาพอร์ต 0 ของ 8051 ที่มีขนาด 8 บิต ชนิดสองทิศทางโดยแต่ละสัญญาณสามารถต่อพ่วงอุปกรณ์ TTL แบบ LS ได้ 8 ตัว และเป็นขาให้สัญญาณมัลติเพล็กซ์ระหว่างสัญญาณข้อมูลกับ 8 บิตล่างของสัญญาณตำแหน่ง ในกรณีที่ใช้หน่วยความจำภายนอก

P1 (1-8) เป็นขาพอร์ต 1 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasi-bidirectional โดยถ้าต้องการให้พอร์ตเส้นใดเป็นอินพุท จะให้ค่า "1" ที่บิตนั้น และสามารถต่อพ่วงกับอุปกรณ์ LS TTL ได้ 4 ตัว

P2 (21-28) เป็นขาพอร์ต 2 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasi-bidirectional

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยไม่หวังผลตอบแทนใดๆ หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง และขอสงวนสิทธิ์ในเนื้อหาเอกสารทุกครั้งที่มีการนำไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเดียวกับพอร์ต 1 นอกจากนี้พอร์ต 2 ยังทำหน้าที่ให้สัญญาณตำแหน่ง 8 บิต ในกรณีที่ใช้หน่วยความจำภายนอกในการอ้างตำแหน่งหน่วยความจำ 16 บิต ดังนั้นขณะที่ใช้หน่วยความจำภายนอก จะต้องไม่มีการเขียนข้อมูลใด ๆ ไปยังพอร์ต 2 เพราะจะทำให้เกิดความผิดพลาดในการทำงานได้

P3 (10-17) เป็นขาพอร์ต 3 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasi - bidirectional เช่นเดียวกับพอร์ต 1 และพอร์ต 2 แต่พอร์ต 3 นี้จะมีหน้าที่พิเศษดังตารางข้างล่างนี้ ดังนั้นเมื่อมีการใช้สัญญาณดังกล่าว จึงไม่ควรเขียนข้อมูลไปที่พอร์ต 3 เพราะจะทำให้การทำงานของ 8051 ผิดพลาดได้

ตารางที่ 1 แสดงหน้าที่พิเศษของพอร์ต 3

ขาพอร์ต	หน้าที่พิเศษ
P3.0	RxD (สำหรับรับข้อมูลแบบอนุกรม)
P3.1	TxD (สำหรับส่งข้อมูลแบบอนุกรม)
P3.2	INT0 (ขาอินเทอร์รัพท์ภายนอก 0)
P3.3	INT1 (ขาอินเทอร์รัพท์ภายนอก 1)
P3.4	TO (ขาอินพุตของ Timer 0)
P3.5	T1 (ขาอินพุตของ Timer 1)
P3.6	WR (สำหรับสัญญาณเขียนหน่วยความจำภายนอก)
P3.7	RD (สำหรับสัญญาณอ่านหน่วยความจำ)

RST (9) ขาสำหรับรีเซ็ตการทำงานของ 8051 โดยการให้ลอจิก "1" อย่างน้อย 2 ช่วงแมชชีนไซเคิล

ALE/PORG (30) ขาสัญญาณออกของ Address Latch Enable สำหรับแลทช์ (Latch) ค่าตำแหน่ง 8 บิตที่ได้จากพอร์ต 0 โดยจะมีความถี่ออกมาที่ 1/6 ของความถี่อ้างอิงของ 8051

PSEN (29) ขาสัญญาณ Program store Enable ใช้สำหรับหน่วยความจำภายนอก

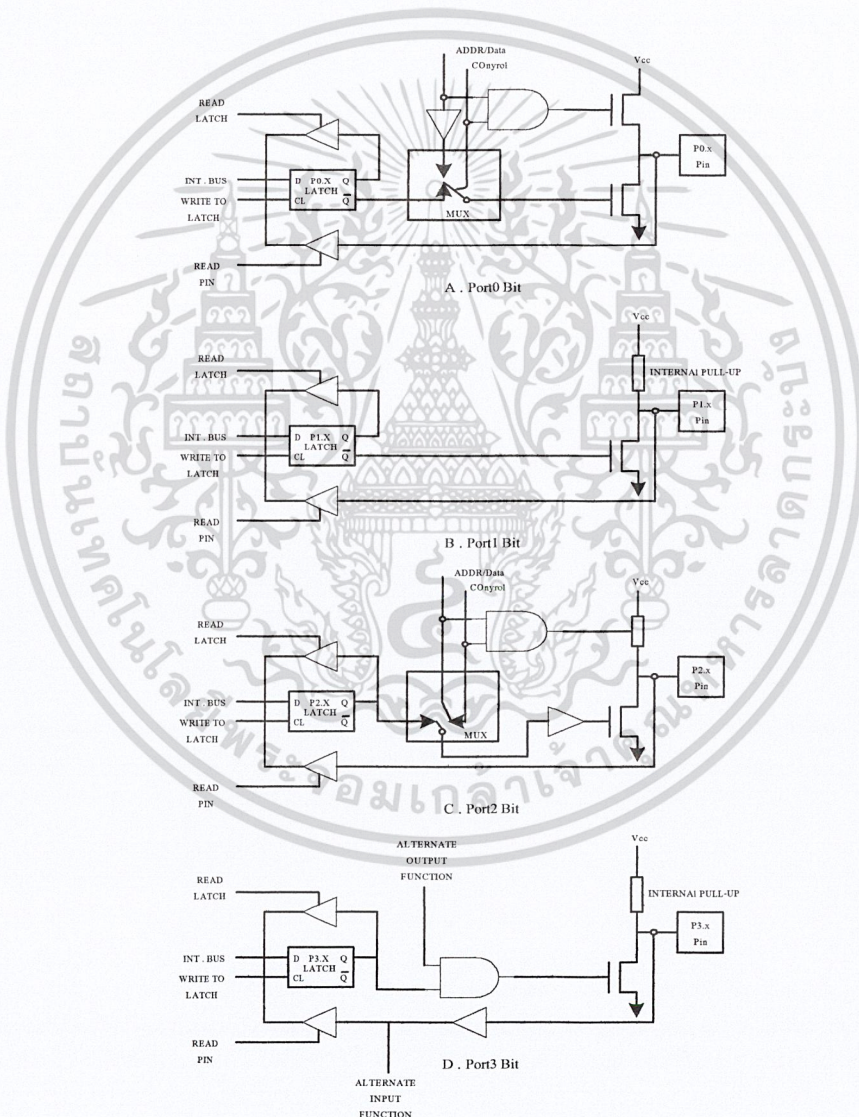
EA/ VPP (31) ขาสัญญาณ External Access Enable สำหรับกำหนดให้ 8051 อ่านหน่วยความจำโปรแกรมภายในหรือภายนอก โดยให้ลอจิก "1" เป็นการอ่านหน่วยความจำ

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้เพื่อใช้สำหรับเรียนการสอนเท่านั้น การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

XTAL1 (19) จำภายใน ลอจิก "0" เป็นการอ่านหน่วยความจำภายนอก
 XTAL2 (18) ขาเข้าของวงจรถ่ายความถี่อ้างอิงภายใน 8051
 ขาออกของวงจรถ่ายความถี่อ้างอิงภายใน 8051

โครงสร้างภายในพอร์ตของ 8051

โครงสร้างภายในพอร์ตของ 8051 แต่ละพอร์ตแสดงได้ดังรูปที่ 3.3



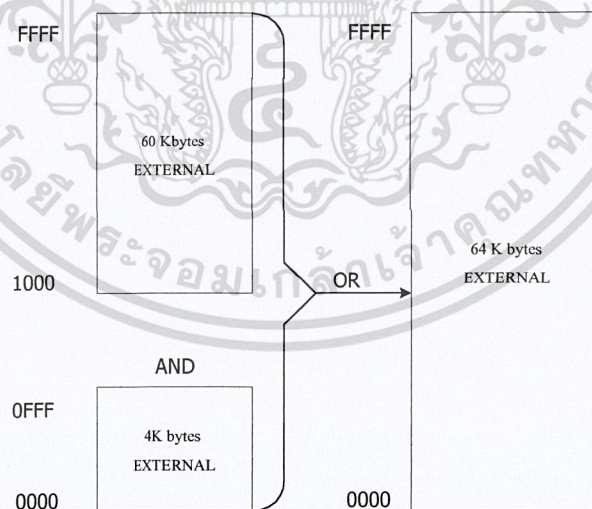
รูปที่ 3.3 แสดง โครงสร้างภายในพอร์ตอินพุตเอาต์พุตของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นได้ว่า พอร์ต 0 จะเป็นพอร์ตแบบ 2 ทิศทางอย่างแท้จริงโดยข้อมูลที่ส่งออกและอ่านเข้าจะไม่รบกวนกัน แต่โครงสร้างของพอร์ต 1 พอร์ต 2 และพอร์ต 3 จะเป็นเอาพุทให้เฟทเพียงตัวเดียว ดังนั้นข้อมูลที่ส่งออกไปโดยเฉพาะลอจิก "0" จะรบกวนข้อมูลที่ออกได้อย่างแท้จริง โดยถ้าต้องการจะใช้บิต 1 ของพอร์ตดังกล่าวเป็นอินพุท จะต้องเขียนลอจิก "1" ไปที่บิตดังกล่าว

โครงสร้างหน่วยความจำของ 8051

ไมโครคอนโทรลเลอร์ 8051 จะมีการแบ่งหน่วยความจำออกเป็น 2 ส่วน ได้แก่ หน่วยความจำสำหรับโปรแกรม และหน่วยความจำสำหรับเก็บข้อมูล โดยที่ขนาดของแต่ละส่วนจะเท่ากับ 64 กิโลไบต์ ในส่วนของหน่วยความจำสำหรับโปรแกรมจะเป็นหน่วยความจำสำหรับอ่านเพียงอย่างเดียว โดยที่ 8051 จะใช้สัญญาณ PSEN (Prom State Enable) ในการอ่านเท่านั้น แต่หน่วยความจำข้อมูลของ 8051 จะสามารถอ่านและเขียน โดยใช้สัญญาณ RD (Read) และ WR (Write) ตามลำดับแต่อย่างไรก็ตามผู้ใช้สามารถรวมหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลเข้าด้วยกันได้ โดยนำสัญญาณ RD (Read) และ PSEN (Program State Enable) มา AND กัน เพื่อสร้างสัญญาณอ่านหน่วยความจำ นอกจากนี้ที่หน่วยความจำดังกล่าวยังแบ่งออกเป็นหน่วยความจำภายในและหน่วยความจำภายในแสดงดังรูปที่ 3.4 และรูปที่ 3.5

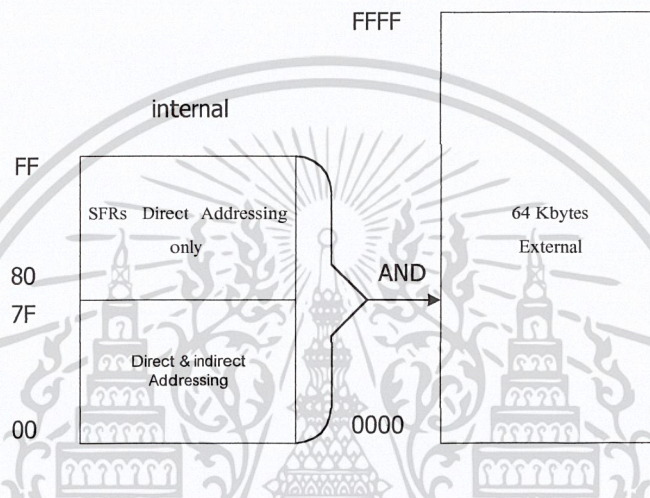


รูปที่ 3.4 แสดงหน่วยความจำโปรแกรมของ MCS-51

จากรูปที่ 3.4 แสดงหน่วยความจำโปรแกรมในกรณีที่เลือกใช้หน่วยความจำภายในหรือภายนอก ด้านซ้ายมือเป็นส่วนของหน่วยความจำโปรแกรมภายในที่มีขนาด 4 กิโลไบต์ ของ 8051 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่เหลือจะเป็นหน่วยความจำภายนอก ส่วนด้านขวาแสดงหน่วยความจำโปรแกรมเมื่อเลือกให้ติดต่อหน่วยความจำภายนอกทั้งหมด

สำหรับหน่วยความจำข้อมูลของ 8051 สามารถแบ่งออกเป็นภายนอกและภายใน โดยหน่วยความจำภายนอกแสดงไว้ด้านขวามือของรูปที่ 3.5 ซึ่งมีขนาด 64 กิโลไบต์ ส่วนหน่วยความจำข้อมูลภายในแสดงไว้ด้านซ้ายมือของรูปที่ 3.5



รูปที่ 3.5 แสดงหน่วยความจำข้อมูลของ MCS- 51

โดยหน่วยความจำของ 8051 จะแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วนของหน่วยความจำข้อมูลที่สามารถอ้างอิงแบบโดยตรงเท่านั้น เรียกอีกแบบหนึ่งว่ารีจิสเตอร์หน้าที่พิเศษ (Special Function register)

ในส่วนของหน่วยความจำข้อมูลภายในที่อ้างอิงโดยตรงและโดยอ้อม นั้นจะสามารถแบ่งออกได้เป็น 3 ส่วน แสดงดังรูปที่ 3.6 โดยมีรายละเอียดดังนี้

- **Register Bank 0-3** เป็นส่วนที่อยู่ในตำแหน่งหน่วยความจำข้อมูลภายในตั้งแต่ 00H -1FH จำนวน 32 ไบต์ โดยจะแบ่งออกเป็นชุด ชุดละ 8 ไบต์จำนวน 4 ชุด ซึ่งแต่ละชุดจะมีชื่อเรียกเป็น R0 ถึง R7 เป็นรีจิสเตอร์ที่ใช้งานร่วมในแต่ละคำสั่ง โดยเมื่อ 8051 ถูกรีเซ็ต Register Bank 0 จะถูกเลือก

- **Bit Address Area** เป็นส่วนที่มีขนาด 16 ไบต์ ที่ตำแหน่งหน่วยความจำข้อมูล 20H ถึง 2FH ในส่วนนี้สามารถที่จะอ้างอิงถึงข้อมูลได้เป็นระดับบิตถึง 128 บิต โดยการอ้างอิงตำแหน่งโดยตรงในลักษณะบิตตั้งแต่ตำแหน่ง 00H ถึง 7FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

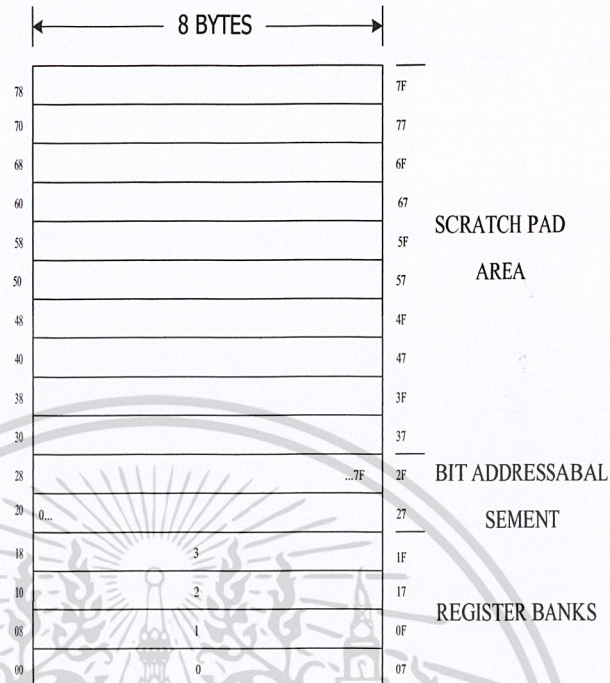
- Scratch Pad Area เป็นส่วนที่อยู่ตำแหน่งตั้งแต่ 30H ถึง 7FH ซึ่งเป็นบริเวณหน่วยความจำข้อมูลภายในเอนกประสงค์ผู้ใช้สามารถใช้ได้โดยตรง นอกจากนี้ยังสามารถใช้หน่วยความจำข้อมูลบริเวณนี้สำหรับเก็บข้อมูลแบบสแต็ก (Stack) ได้ด้วย

B							
ACC							
PSW							
T2CON		RCAP2H	RCAP2H	TL2	TH2		
IP							
P3							
IE							
P2							
SCON	SBUF						
P1							
TCON	TMOD	TL0	TL1	TH0	TH1		
P0	SP	DPL	DPH				PCON

รูปที่ 3.6 แสดงหน่วยความจำข้อมูลภายใน

ในส่วนของหน่วยความจำข้อมูลภายในที่ใช้อ้างอิงแบบโดยตรงเพียงอย่างเดียวหรือที่เรียกว่ารีจิสเตอร์หน้าที่พิเศษ ซึ่งเป็นส่วนสำหรับเก็บหรือกำหนดการทำงานภายในของ MCS-51 แสดงได้ดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงรายละเอียดของรีจิสเตอร์ฟิงชันพิเศษ

แฟลชเมมโมรีไมโครคอนโทรลเลอร์ (False Memory Controller)

AT89C51 เป็นอุปกรณ์ไมโครคอนโทรลเลอร์ที่มีหน่วยความจำแบบแฟลชอยู่ภายในขนาด 4 กิโลไบต์ ที่สามารถใช้เครื่องคอมพิวเตอร์ทำการลบหรือเขียนโปรแกรมได้โดยตรงโดยไม่ต้องถอด MCU (Microcontroller Unit) ออกจากการ์ดหรือแผงวงจรในลักษณะที่เรียกว่า In system programming หรือจะใช้เครื่องโปรแกรม (Universal Programmer) โดยตรงได้ ซึ่งสามารถโปรแกรมได้เป็นพันครั้ง ไมโครคอนโทรลเลอร์ตระกูล AT89C51 สามารถใช้งานร่วมและแทนกันได้กับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ได้แก่เบอร์ 80C51 80C51/52 87C51/52 เป็นต้น ซึ่ง AT89C51 เหมือนกับตระกูลเหล่านี้ของอินเทล ทำให้ด้านชุดคำสั่งและการจัดเรียงขา นั้นเราสามารถนำ AT89C51 มาใช้แทน MCS-51 ได้เลย

3.2 การขยายพอร์ตอินพุตหรือเอาต์พุตของไมโครคอนโทรลเลอร์ 8051 เพิ่มเติมโดยใช้ IC 8255

การขยายพอร์ตอินพุตหรือเอาต์พุตของไมโครคอนโทรลเลอร์ 8051 เพิ่มเติมนั้น นอกจากจะใช้ชิปประเภทแลตซ์และบัฟเฟอร์ประกอบกันเข้ากับระบบบัสแล้ว ยังสามารถใช้ไอซีที่เป็นวงจรรวมความจุสูงเบอร์ 8255 ได้ ซึ่งสามารถทำหน้าที่ได้ทั้งพอร์ตอินพุตและเอาต์พุตด้วยโปรแกรม ทำให้มีการอ่อนตัวในการนำไปใช้งานตามความต้องการมากขึ้น IC 8255 ได้รับการออกแบบมาเพื่อทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ หงสน อีกทั้งห้ามมีเหตุดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่เป็นพอร์ตสำหรับการ รับ-ส่ง ข้อมูลแบบขนานระหว่างอุปกรณ์ภายนอกกับ อุปกรณ์ไมโครคอนโทรลเลอร์ และสามารถเปลี่ยนแปลงลักษณะการทำงานของพอร์ตให้เป็นอินพุตหรือเอาต์พุตได้สะดวก

ลักษณะพื้นฐานของไอซี 8255

ไอซี 8255 ประกอบด้วยส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกโดยตรงผ่านสายสัญญาณที่ระบุชื่อว่า PA0 - PA7 กลุ่ม PB0 - PB7 และ PC0 - PC7 กลุ่มสัญญาณเหล่านี้จำแนกออกได้เป็น 3 กลุ่ม คือ พอร์ต A (PA) พอร์ต B (PB) และพอร์ต C (PC) นอกจากนี้ Group A control และ Group B control ยังทำหน้าที่กำหนดการทำงานของพอร์ตทั้งสาม สองกลุ่มสัญญาณนี้จะเชื่อมต่อกับส่วนอื่น ๆ ผ่านทางบัสข้อมูลภายใน (internal bus) ของ 8255 สำหรับส่วนการทำงานที่มีชื่อเรียกว่า Data bus buffer และ Read/Write Control logic ทำหน้าที่เชื่อมต่อระหว่างระบบบัสของไมโครคอนโทรลเลอร์กับ 8255 เพื่อรับส่งข้อมูลระหว่างกัน ตามลอจิกของขาสัญญาณ RD (Read) และ WR (Write) ตามลำดับ

การจำแนกกลุ่มของพอร์ต 8255

จากรูปที่ 3.8 จะเห็นว่า 8255 มีพอร์ตจำนวน 3 พอร์ต คือ พอร์ต A พอร์ต B และพอร์ต C โดยพื้นฐานแล้วทั้ง 3 พอร์ตเป็นพอร์ตแบบขนานที่ประกอบด้วยสัญญาณ 8 เส้น ซึ่งแต่ละเส้นจะแทนบิตของข้อมูลพอร์ต ซึ่งอาจกล่าวได้อีกลักษณะว่าเป็นพอร์ตแบบ 8 บิต นอกจากนี้ยังสามารถอ้างอิงถึงแต่ละบิตของเส้นสัญญาณของพอร์ตนี้ได้โดยอิสระ อย่งไรก็ตาม 8255 นี้ยังจัดกลุ่มของพอร์ตได้เป็นสองกลุ่มคือกลุ่ม A และกลุ่ม B เพื่อประโยชน์ในการกำหนดรูปแบบการทำงานของพอร์ต ดังตารางที่ 2

ตารางที่ 2 การกำหนดรูปแบบการทำงานของพอร์ต 8255

ชื่อกลุ่ม	ลักษณะ
GROUP A	พอร์ต A จำนวน 8 บิต (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตบนของพอร์ต)
GROUP B	พอร์ต B จำนวน 8 บิต (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตล่างของพอร์ต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

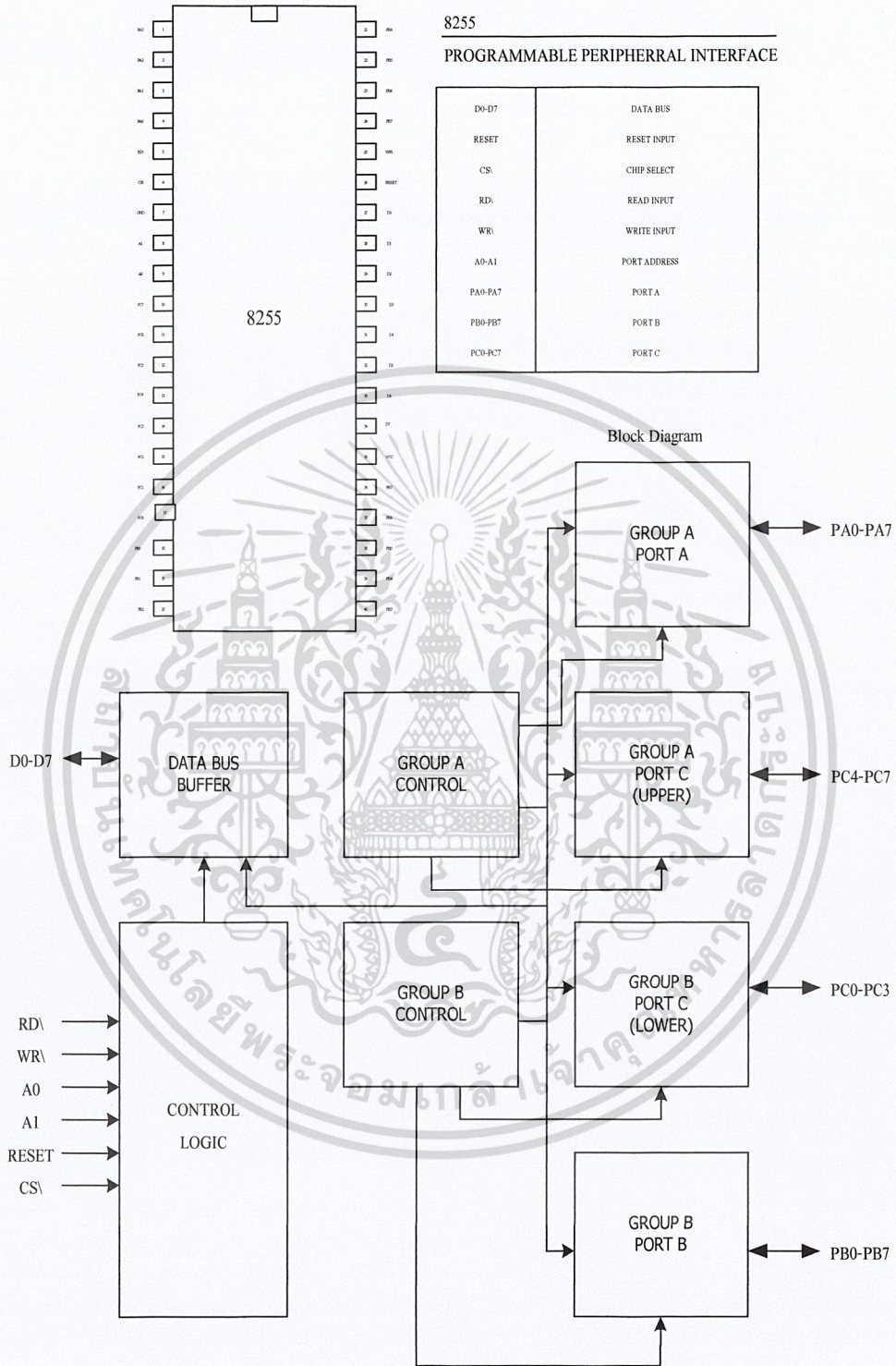
จากตารางที่ 2 จะเห็นได้ว่าจำนวนเส้นสัญญาณทั้งหมดของพอร์ต C (PC0-PC7) ได้ถูกแยกออกเป็นกลุ่มคือ กลุ่มของ 4 บิตล่าง (lower nibble) จาก PC0-PC3 และกลุ่มของ 4 บิตบน (upper nibble) จาก PC4-PC7 ดังนั้นกลุ่ม A และกลุ่ม B ของ 8255 จึงมีจำนวนบิตในแต่ละกลุ่มเป็น 12 บิต

ส่วนหน้าที่การทำงานต่าง ๆ แสดงได้ดังตารางที่ 3

ตารางที่ 3 แสดงหน้าที่การทำงานของขาสัญญาณไอซี 8255

ขาสัญญาณ	ความหมาย
D0-D7	กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียนหรืออ่าน
CS	กลุ่มของสัญญาณเลือกอุปกรณ์ เมื่อขาของสัญญาณนี้เป็นระดับลอจิกต่ำ แสดงว่าสามารถเขียนหรืออ่านข้อมูลจาก 8255 ได้
RD	สัญญาณบอกสถานะว่าต้องการอ่านข้อมูลจากรีจิสเตอร์ของ 8255
WR	สัญญาณบอกสถานะว่าต้องการเขียนข้อมูลจากรีจิสเตอร์ของ 8255
A0-A1	สัญญาณระบุตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ต้องการ
RESET	สัญญาณรีเซ็ตวงจรทำงานภายใน 8255 เพื่อเริ่มต้นใหม่
PA0-PA7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต A ของ 8255
PB0-PB7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต B ของ 8255
PC0-PC7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต C ของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงขาสัญญาณต่าง ๆ และโครงสร้างภายในของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเชื่อมต่อ 8255 กับ MCS-51

เมื่อพิจารณาจากรูปที่ 3.8 จะเห็นว่ามิชชาสัญญาณระบุตำแหน่งจำนวน 2 เส้น คือ A0 และ A1 ทำให้มีจำนวนตำแหน่งที่จะอ้างถึงได้ มีค่าเป็น 2 ตำแหน่ง หรือเท่ากับ 4 ตำแหน่งนั่นเอง ซึ่งแต่ละตำแหน่งมีความหมายถึงระบบรีจิสเตอร์หรือพอร์ตภายใน 8255 ดังตารางที่ 4

ตารางที่ 4 ระบบรีจิสเตอร์ของ 8255

A1	A0	ชื่อรีจิสเตอร์
0	0	พอร์ต A
0	1	พอร์ต B
1	0	พอร์ต C
0	1	รีจิสเตอร์ควบคุม

เมื่อพิจารณาค่าตำแหน่งเหล่านี้ร่วมกับระดับลอจิกของขาสัญญาณ RD/WR จะเป็นการอ่านหรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับรีจิสเตอร์นั้นตามลำดับ ดังตารางที่ 5

ตารางที่ 5 แสดงข้อมูลทางขาสัญญาณ

RD	WR	A1	A2	ความหมาย
0	1	0	0	ส่ง (เขียน) ข้อมูลให้กับพอร์ต A
1	0	0	0	ส่ง (อ่าน) ข้อมูลให้กับพอร์ต A
0	1	0	1	ส่ง (เขียน) ข้อมูลให้กับพอร์ต B
1	0	0	1	ส่ง (อ่าน) ข้อมูลให้กับพอร์ต B
0	1	1	0	ส่ง (เขียน) ข้อมูลให้กับพอร์ต C
1	0	1	0	ส่ง (อ่าน) ข้อมูลให้กับพอร์ต C
0	1	1	1	ส่ง (เขียน) ข้อมูลให้กับพอร์ตควบคุม
1	0	1	1	เป็นสถานะที่ไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นโดยทั่วไปจึงมักกำหนดให้ตำแหน่งของ IC 8255 ทั้ง 4 ตำแหน่งนี้ อยู่ในตำแหน่งช่วงใดช่วงหนึ่งของระบบ เช่น 10H 11H 12H และ 13H โดยขาสัญญาณระบุตำแหน่งที่นอกเหนือไปจาก A0 และ A1 นำมาเข้ายังตัวลอจิกตำแหน่งเพื่อสร้างสัญญาณเลือกอุปกรณ์ CS นี้ จะเป็นสถานะลอจิก "low" ก็ต่อเมื่อค่าบัสตำแหน่ง A0 - A7 มีค่าเท่ากับ 0000100XX (ตัวอักษร XX ระบุถึงรีจิสเตอร์ภายใน 8255 เพื่อทำการอ่านหรือเขียนข้อมูล) ดังนั้นตำแหน่งของรีจิสเตอร์ภายใน 8255 จะมีค่าตามตารางที่ 6

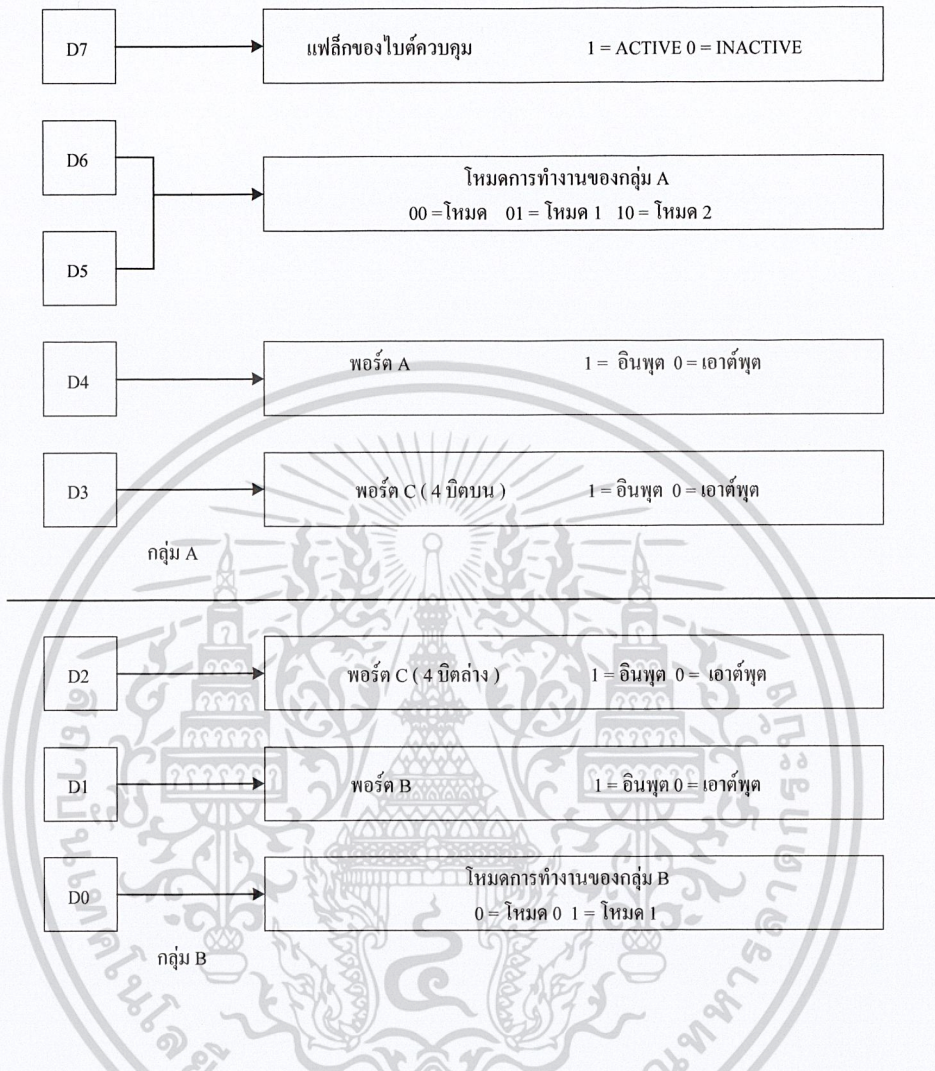
ตารางที่ 6 ตำแหน่งของรีจิสเตอร์

ตำแหน่ง	ความหมาย
10h	พอร์ต A
11h	พอร์ต B
12h	พอร์ต C
13h	รีจิสเตอร์ควบคุม

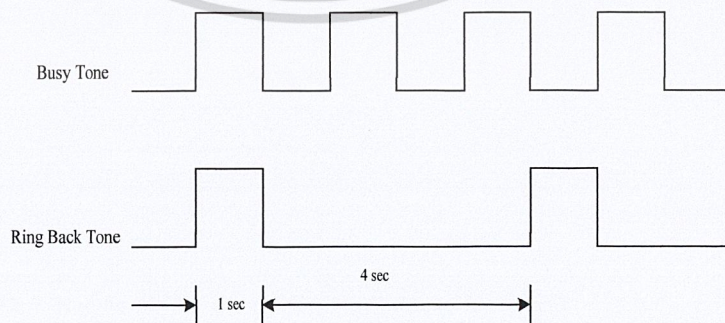
ขาสัญญาณอื่น ๆ เช่น RD (Read) และ WR (Write) มักจะเชื่อมต่อเข้ากับขาสัญญาณข้อเดียวกับของ 8051 โดยตรง ทำให้ตำแหน่งของ 8255 ซึ่งมีผลทำให้เกิดสถานะการทำงานใหม่เมื่อระดับขาของสัญญาณเป็นลอจิก "High" ดังนั้นหากว่าขาสัญญาณการรีเซ็ตเดียวกับของ IC 8051 หากต้องการจะรีเซ็ต IC 8255 ด้วยก็สามารถต่อได้ตรงส่วนขาสัญญาณ D0-D7 ก็สามารถนำไปเชื่อมต่อโดยตรงเข้ากับสายสัญญาณของ 8051 ได้เช่นกันความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ 8255 แสดงได้ดังรูปที่ 3.9

3.4 ภาคตรวจจับสัญญาณเสียง

ภาคตรวจจับสัญญาณเสียงนี้ มีหน้าที่ตรวจจับสัญญาณเสียงต่าง ๆ ที่รับมาจากสายโทรศัพท์ สัญญาณ ที่รับเข้ามาจากสายโทรศัพท์จะประกอบไปด้วยสัญญาณเสียงพูด สัญญาณเสียงเรียก (Ring Back Tone) และสัญญาณเสียงสายไม่ว่าง (Busy Tone) สัญญาณเสียงที่ถูกกำหนดโดย CCITT จะอยู่ระหว่างความถี่ 300Hz - 3400Hz และเสียง Ring Back Tone กับ Busy Tone จะมีความถี่ 400Hz - 450 Hz โดยที่ใช้เวลาในการส่งต่างกัน ในประเทศไทยโดยองค์การโทรศัพท์แห่งประเทศไทยเป็นผู้ให้บริการ ได้กำหนดให้สัญญาณ Ring Back Tone มีช่วงดัง 1 วินาที ช่วงหยุด 4 วินาที และสัญญาณ Busy Tone มีช่วงดัง 0.5 วินาที และเงียบ 0.5 วินาที ดังรูปที่ 3.10



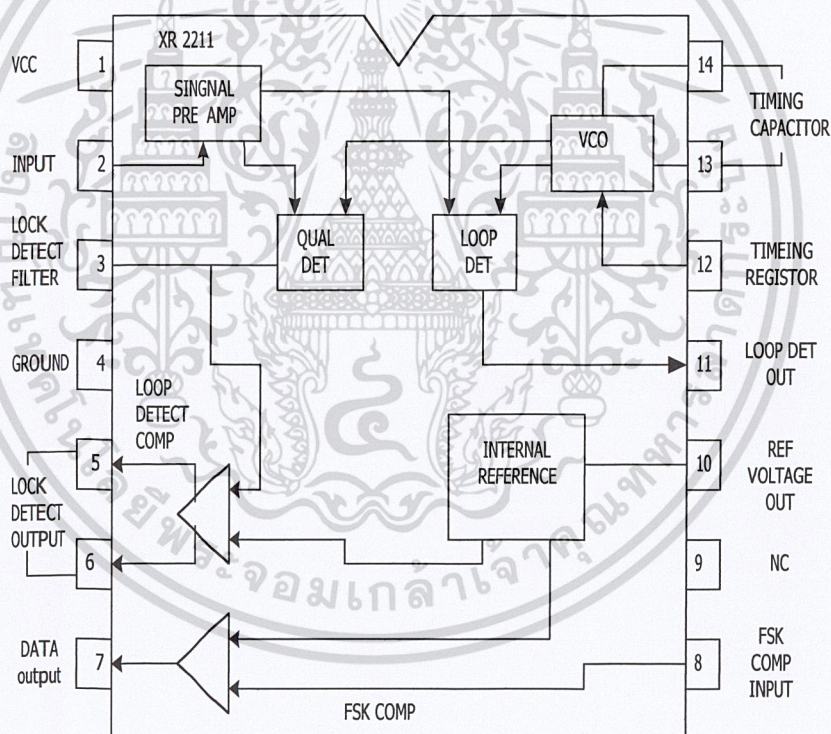
รูปที่ 3.9 แสดงความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ IC 8255



รูปที่ 3.10 ลักษณะของสัญญาณโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในภาคตรวจจับสัญญาณเสียงนี้เราจะใช้ IC XR-2211 เป็นตัวตรวจจับ เราสามารถกำหนดความถี่ในการตรวจจับได้โดยการกำหนดค่าตัวต้านทานและตัวเก็บประจุ โดยการคำนวณจากสูตรใน DATA SHEET โครงสร้างภายในของไอซีเบอร์ XR-2211 ดังแสดงตามรูปที่ 3.11 ไอซีเบอร์นี้ถูกออกแบบให้ใช้ทำหน้าที่เป็น FSK (Frequency Shift Keying) Demodulator และ tone decoder โครงสร้างของไอซีประกอบด้วย Input Preamplifier ทำหน้าที่ขยายสัญญาณความถี่เสียงที่รับเข้ามาให้มีขนาดสัญญาณแรงขึ้น วงจรเฟสดีเทคเตอร์ (Phase detector) ทำหน้าที่เปรียบเทียบเฟสของสัญญาณ และวงจรโวลต์เจกคอนโทรลเลอร์ออสซิลเลเตอร์ (Voltage Controlled oscillator : VCO) ทำหน้าที่ควบคุมการผลิตความถี่ด้วยแรงเคลื่อน โดยการกำหนดค่าความถี่ที่ตรวจจับแล้วนำมาไปคำนวณหาค่า R_0 , R_1 , C_0 , C_1 ไอซีเบอร์ XR-2211 จะให้เอาพุตที่ได้จากขา 5 จะเป็นลอจิก "0" และเอาต์พุตที่ได้จากขา 6 จะเป็นลอจิก "1"

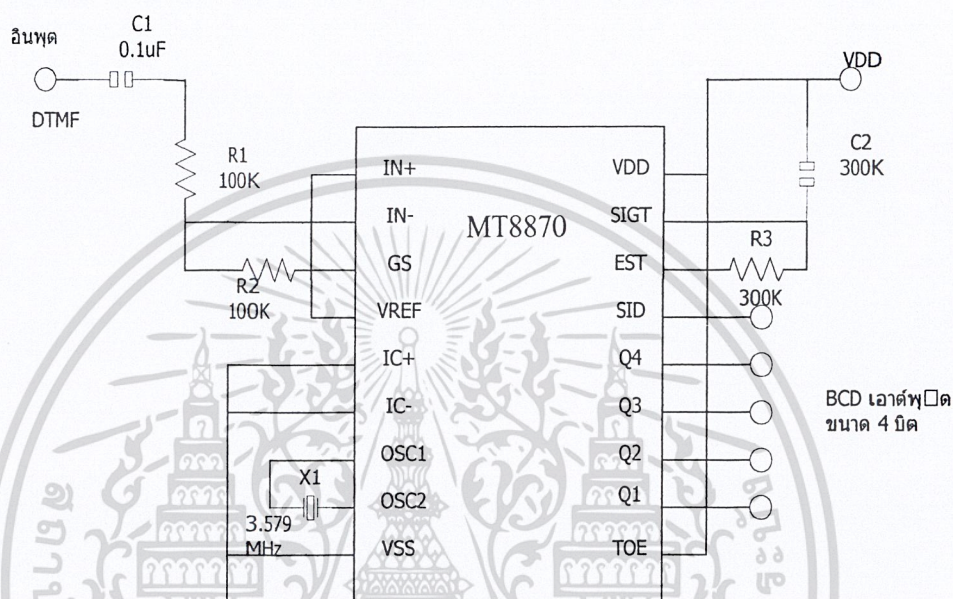


รูปที่ 3.11 โครงสร้างภายใน IC XR-2211

3.5 วงจรถอดรหัสเลขหมาย (DTMF Decoder)

โดยใช้ไอซีเฟสล็อกลูป (Phase lock loop) ซึ่งสร้างปัญหามากมายเช่น ความถี่ที่เปลี่ยนแปลงไป การปรับแต่งวงจรขนาดของวงจรเพราะต้องการใช้ ไอซีจำนวนมาก จากสารพัดปัญหาเหล่านี้ทำให้ไอซีเบอร์ MT-8870 มีคุณสมบัติคืออยู่หลายประการ โดยการถอดรหัสจะทำเมื่อมีสัญญาณ

ความถี่ DTMF เข้าสู่วงจร DTMF decode MT8870 จะทำการถอดรหัสเลขหมายซึ่งผลที่ออกมาจะเป็นสัญญาณดิจิทัล 4 บิต แล้วส่งให้ส่วนควบคุมสัญญาณต่อไปซึ่ง รายละเอียดของตัวอย่างวงจรถอดรหัสเลขหมายแสดงไว้ดังรูปที่ 3.12



รูปที่ 3.12 วงจรถอดรหัสเลขหมาย

ข้อกำหนดของวงจรถอดรหัสเลขหมายแบบดีทีเอ็มเอฟ

1. วงจรจะยังคงถอดรหัสได้อย่างถูกต้อง ถึงแม้สัญญาณที่รับเข้ามามีความเบี่ยงเบนไปจากค่ามาตรฐาน แต่ต้องไม่เกินบวกลบ 2 เปอร์เซ็นต์
2. วงจรถอดรหัสจะทำการถอดรหัสได้ก็ต่อเมื่อได้รับสัญญาณเข้ามามีระยะเวลาอย่างน้อย 40 มิลลิวินาที
3. วงจรถอดรหัสจะทำการถอดรหัสได้ถูกต้อง ก็ต่อเมื่อสัญญาณดีทีเอ็มเอฟที่รับเข้ามาในวงจรจะต้องมีช่วงเวลาเท่ากับสัญญาณดีทีเอ็มเอฟที่รับเข้ามาก่อนหน้านี้เป็นเวลาอย่างน้อย 35 มิลลิวินาที
4. วงจรถอดรหัสจะทำการถอดรหัสที่มีไดนามิกเรนจ์สูงกว่า 27.5 เดซิเบลได้โดยไม่เกิดความผิดพลาด และยังสามารถทำงานได้ในกรณีที่สัญญาณ 2 ความถี่ที่มีแอมป์ลิจูดแตกต่างกันมากกว่า 6 เดซิเบล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. วงจรถอดรหัสยังทำงานได้ตลอดเวลา ไม่ว่าจะขณะนั้นจะปรากฏเสียงพูด หรือมีสัญญาณรบกวนภายนอกเข้ามายังวงจรถอดรหัสก็จะไม่ทำให้การถอดรหัสผิดพลาด

คุณสมบัติของไอซีเบอร์ MT-8870

- เป็นตัวรับและถอดรหัสความถี่ (DTMF Receiver)
- ใช้กำลังไฟฟ้าน้อย และใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถปรับการ์ดไทม์ (Guard time) ได้
- สามารถปรับอัตราขยายในตัวไอซีได้
- เป็น ไอซีคุณภาพสูง

โครงสร้างของ ไอซีเบอร์ MT-8870

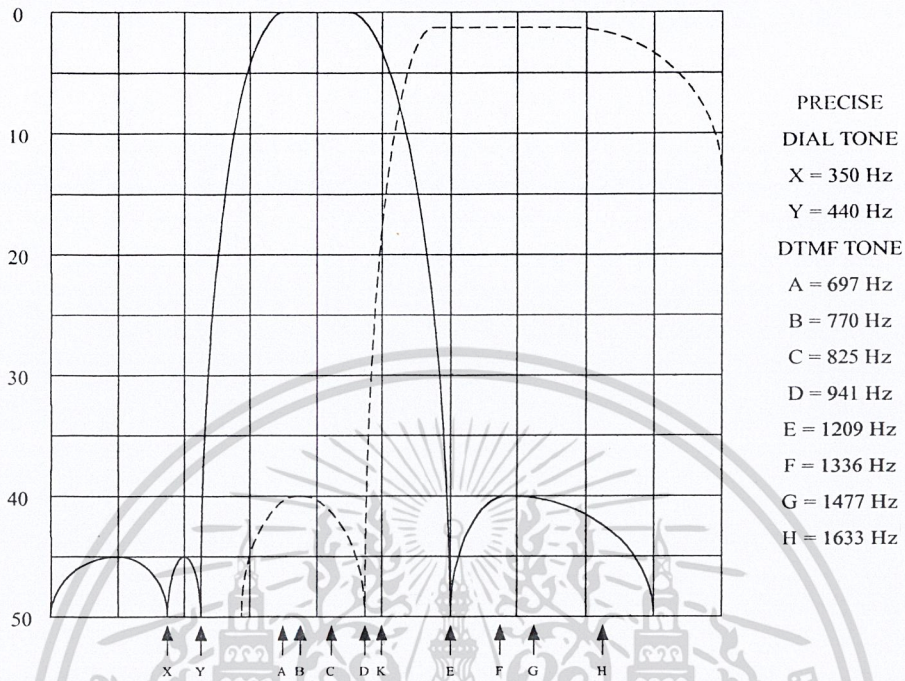
โครงสร้างภายในของไอซี MT-8870 ประกอบด้วยวงจรกรองความถี่ และวงจรถอดรหัส เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO2-CMOS ในส่วนวงจรกรองความถี่ใช้เทคนิคสวิตซ์คาปาซิเตอร์สำหรับกรองความถี่สูง และความถี่ต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิทัล เพื่อตรวจและถอดรหัสทั้ง 16 ความถี่ ออกเป็นเลขฐานสองขนาด 4 บิต และเช็ควงเวลาที่สัญญาณเข้า ส่วนภาคอินพุตเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้ โดยต่ออุปกรณ์ภายนอกเอาต์พุตเป็น วงจรแลตซ์ 3 สถานะ ภายในโครงสร้างของ MT-8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน

- ภาคกรองความถี่ (Filter Section)
- ภาคถอดรหัส (Decoder Section)
- ภาคควบคุมตรวจสอบสัญญาณ (Steering Circuit)
- ภาคขยายสัญญาณความแตกต่าง (Differential Amplifier)
- ภาคกำเนิดความถี่ (Oscillator)

ภาคกรองความถี่ (Filter section)

ในส่วนนี้จะแยกสัญญาณดีทีเอ็มเอฟที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่ผ่านอันดับ 6 ชนิดใช้ สวิตซ์ตัวเก็บประจุ (Sixth order Capaciter Band Filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วงความถี่สูงและช่วงความถี่ต่ำ แสดงได้ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงความถี่ที่ได้จากภาคกรองความถี่

ภาคถอดรหัส (Decoder Section)

ความถี่ที่เอ็มเอฟที่ถูกกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคของวงจรมับแบบดิจิทัล และมีการตรวจสอบความถี่นั้นถูกต้อง สัญญาณที่ขา EST (Early Steering) หรือขา 16 ก็จะทำงาน

สำหรับค่าที่ถอดรหัสได้จากความถี่ต่างนั้นๆ แสดงไว้ดังตารางที่ 7

ตารางที่ 7 แสดงค่ารหัสที่ได้จากความถี่ต่างๆ

Flow	Fhigh	On	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1447	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
Flow	Fhigh	On	TOE	Q4	Q3	Q2	Q1
770	1336	5	H	0	1	0	1

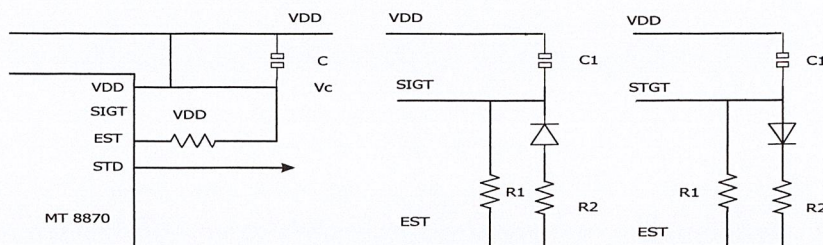
770	1447	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1447	9	H	1	0	0	1
941	1633	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1447	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	-	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

ภาคควบคุมตรวจสอบสัญญาณ (Steering Circuit)

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากรยะเวลาการกดปุ่มโทรศัพท์ที่ให้ความถี่ออกมาเป็นช่วงเวลานั้นเป็นช่วงเวลานานพอสมควร มิฉะนั้นวงจรส่วนจะไม่รับ และถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดนั้นสามารถตั้งได้โดยใช้ ตัวต้านทานและตัวเก็บประจุ (R,C) ต่อจากภายนอกตามรูปที่ 3.14 สัญญาณที่ขา EST จะเป็นระดับสูงนานใกล้เคียงกับระยะเวลาที่ตีทีเอ็มเอฟเข้ามาทำให้ V_c สูงขึ้นตัวเก็บประจุ V_c จะคายประจุทำให้แรงดัน V_c สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสจึงจะถอดรหัสออกเป็นตัวเลข 4 บิต รายละเอียดการทำงานดูได้จากแผนภูมิเวลา (Timing Diagram) จะเข้าใจได้ง่าย

สำหรับการ์ดไทม์ (Guardtime) หมายถึง ช่วงเวลาของความถี่ที่เข้ามาซึ่งจะต้องนานกว่าหรือเวลาที่เรที่ตั้งไว้ จึงจะได้รับการยอมรับสัญญาณความถี่นั้นถูกต้อง หรือพูดได้ว่าเวลาที่เรที่ตั้งไว้โดยตัวต้านทาน และตัวเก็บประจุ คือ การ์ดไทม์นั่นเอง ถ้าสัญญาณความถี่เข้ามานานเท่ากับ หรือมากกว่าเวลาที่ตั้งไว้จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่เข้ามาสั้นกว่า หรือมากกว่าเวลาที่ตั้งไว้จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสสัญญาณตัวเลขออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$t_{GTA} = (RC)\ln(VDD/V_{tst})$$

$$t_{GTP} = (R_p C_1)\ln[VDD / (VDD - V_{tst})]$$

$$t_{GTP} = (R_1 C_1)\ln [VDD / (VDD - V_{tst})]$$

$$t_{GTP} = (RC)\ln[VDD / (VDD - V_{tst})]$$

$$t_{GTA} = (RLC_1)\ln(VDD / V_{tst})$$

$$t_{GTA} = (R_p C_1)\ln(VDD / V_{tst})$$

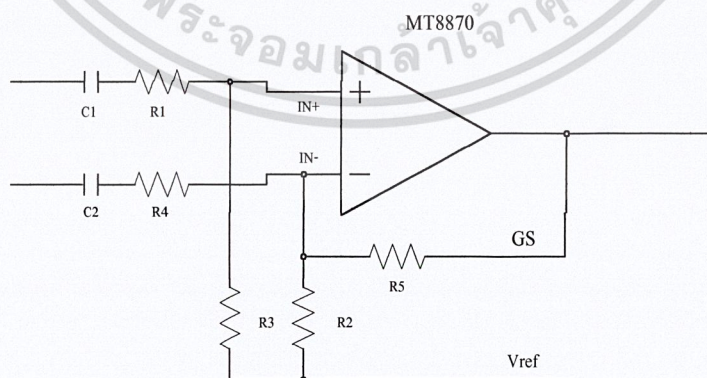
$$R_p = R_1 / R_2$$

$$R_p = R_1 / R_2$$

รูปที่ 3.14 แสดงการต่อวงจรควบคุมตรวจสอบสัญญาณเบื้องต้นของไอซี MT-8870

ภาคขยายสัญญาณความแตกต่าง (Differential Amplifier)

วงจรส่วนอินพุตของ MT-8870 เป็นภาคขยายออปแอมป์ที่สามารถคำนวณอัตราขยายได้ โดยต่อวงจรภายนอกเพิ่มเข้าดังภาพที่ ซึ่งสามารถคำนวณอัตราขยายความแตกต่างอินพุต และอิมพีแดนซ์ได้ดังนี้จากรูปที่ 3.15 สามารถคำนวณอัตราขยายความแตกต่างของอินพุตและอิมพีแดนซ์ได้ดังนี้ อัตราขยาย ($A_v \text{ diff}$) = R_5/R_1 อินพุตอิมพีแดนซ์ (Z_{indiff}) = $2(R_1 + [1/wc])$

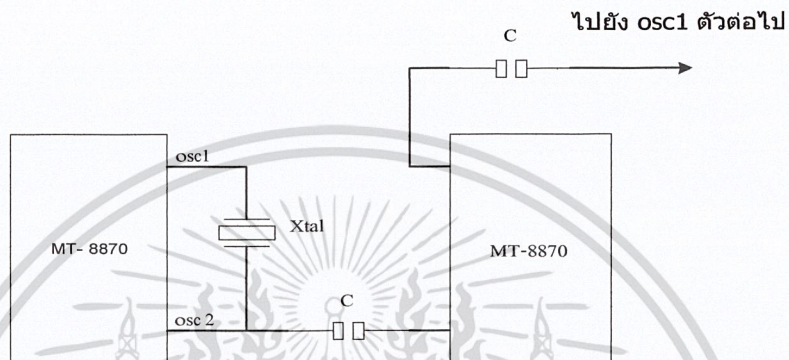


รูปที่ 3.15 การต่อวงจรภาคอินพุตของไอซี MT-8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคกำเนิดความถี่ (Oscillator)

ในภาคนี้ภายในไอซี MT-8870 จะมีวงจรเวลาอยู่ภายใน เพียงต่อแค่วจรขนาด 3.579 MHz ก็จะสามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่แสดงในรูปที่ 3.16



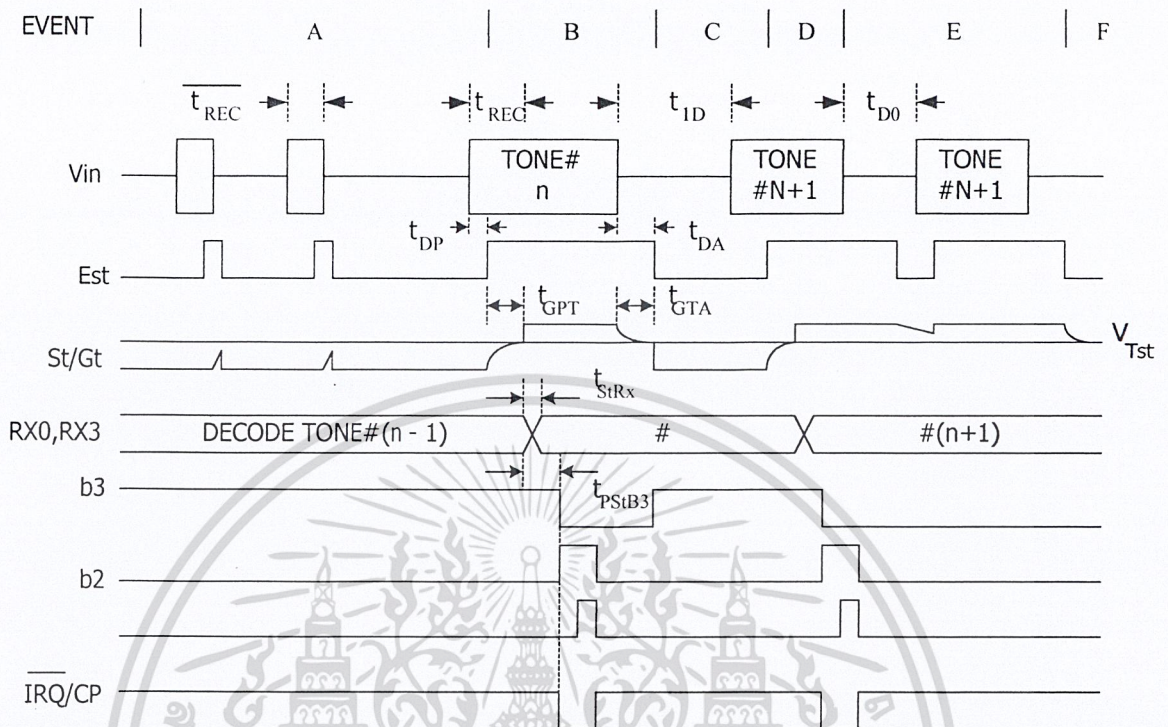
รูปที่ 3.16 การต่อวงจรผลิตความถี่

เพื่อให้ทราบถึงรายละเอียดต่าง ๆ ในการทำงาน และหน้าที่ต่าง ๆ ของ MT-8870 ในการต่อเข้าเป็นระบบเชื่อมโยงกับส่วนต่าง ๆ สามารถที่จะอธิบายด้วยไดอะแกรมสัญญาณ ดังรูปที่ 3.9

อธิบายศัพท์

Vin	สัญญาณความถี่ที่เอ็มเอฟ
Est	Early Steering Output ใช้แสดงความถี่ที่ถูกต้อง
St/GT	Steering input / Guard Time Output สำหรับต่อวงจรกับ RC ภายนอก
Q1-Q4	เอาต์พุต BCD ขนาด 4 บิต
STD	Delay Steering Output ใช้แสดงค่าความถี่ที่ได้รับหรือหายไปมีคาบความถี่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ
TOE	Tone Output Enable (input) ใช้ควบคุม Q1-Q4 ให้เป็นอิมพีแดนซ์สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 ไคอะแกรมสัญญาณต่างๆ ของไอซี MT-8870

3.6 วงจรถอดรหัสคีย์แพด (16 Keypad decoder) โดยใช้ไอซี 74C922

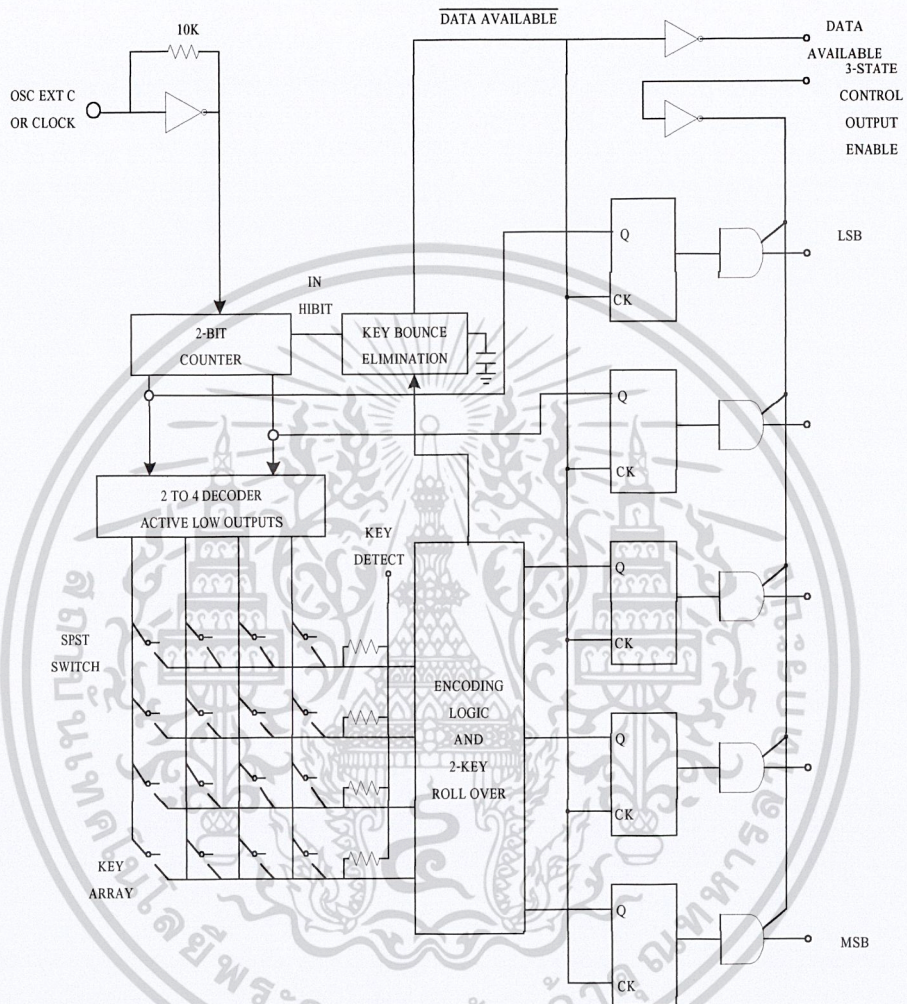
ในการ Scankey เราสามารถทำได้ในทางด้านซอฟต์แวร์ และทางด้านฮาร์ดแวร์ คือ การตรวจโดยการเขียนโปรแกรม ทางด้านฮาร์ดแวร์นี้เราสามารถใช้อิซีตระกูล TTL เบอร์ 74C922 ซึ่งมีคุณสมบัติในการถอดรหัส key matrix 4*4 หรือ 16 keypad ซึ่งเอาท์พุทที่ได้จากไอซีจะเป็นสัญญาณดิจิทัล 4 Bit

คุณสมบัติของไอซีเบอร์ 74C922

- เป็นตัวถอดรหัส keypad 4*4
- ใช้กำลังไฟน้อย ใช้ไฟเลี้ยงได้ตั้งแต่ 3-15 โวลต์
- มี pull-up ที่แถว
- สามารถแก้ไข Key bounce ได้โดยตัวเก็บประจุตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างภายในไอซี 74C922



รูปที่ 3.18 โครงสร้างภายในของไอซี 74C922

แบ่งเป็น 3 ส่วนใหญ่ ๆ คือ

- วงจรถอดรหัสคีย์
- วงจรภาคเอาต์พุต
- วงจรภาค key bounce

วงจรถอดรหัสคีย์

วงจรถอดรหัสคีย์จะประกอบด้วยภาค row scan ซึ่งจะมีค่า internal pull up Encoding logic

และ 2 key roll over เพื่อจะได้เอาต์พุตทางด้าน row ไปให้แก่วงจรภาคเอาต์พุต ส่วนในภาค Scan

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

coloum จะถูกควบคุมโดย OSC (Oscilater) ทางอินพุตจะเป็นแบบ Schmitt trigger osc ประกอบด้วย Cext ที่ต่ออยู่กับขา 5 ของไอซีและวงจร 2 - 4 bit decode

วงจรรภาคเอาต์พุต

วงจรรภาคเอาต์พุตจะประกอบด้วย D-Flip Flop ทำหน้าที่ latch data output ให้มีสภาวะค้างสถานะเอาต์พุตเก่าจะอยู่จนกว่าจะมีการกดคีย์ เอาต์พุตจึงเปลี่ยนไปตามคีย์ที่กด

วงจรรภาค keybounce

วงจรรภาค keybounce จะมีการต่อ Cext ที่ขา 6 เพื่อใช้ตั้งค่าคาบเวลาในการแก้ปัญหา bounce

3.7 การขับโมดูลแสดงผลแบบผลึกเหลว LCD module

ในปัจจุบันแอลซีดี (LCD : liquid Crstal Display) เป็นที่นิยมใช้งานเพื่อเป็นส่วนแสดงผลในอุปกรณ์และเครื่องใช้งานต่าง ๆ เป็นอย่างมาก ทั้งนี้เนื่องจากมีความเหมาะสมหลาย ๆ ด้าน เช่น ใช้กระแสไฟต่ำ สามารถแสดงผลเป็นตัวอักษร แสดงผลเป็นตัวเลข หรือ แสดงผลเป็นกราฟฟิก (graphic) ได้ ส่วนที่ยังมีปัญหาลูกค้างบ้างก็ คือ เรื่องวงจร เพราะแอลซีดีมีระบบการทำงานที่ซับซ้อนและหาอุปกรณ์ในวงจรได้ค่อนข้างจะยาก แต่ในขณะนี้ผู้ผลิตแอลซีดีรุ่นที่เป็น โมดูลที่มีตัวแอลซีดีและวงจรควบคุมมาให้พร้อมซึ่งสามารถทำให้ผู้ใช้งานด้าน ไมโครคอนโทรลเลอร์หันมาใช้แผงแสดงผลด้วยแอลซีดีโมดูลกันมากขึ้น

แอลซีดีโมดูลมีอยู่หลายรุ่นและมีคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลัก คือ

1. แบบดอตแมทริกซ์ (Dot Matrix) จะแสดงผลเป็นตัวอักษรขนาด 5*8 จุดมีจำนวนอักษรและจำนวนบรรทัดแตกต่างกันในแต่ละรุ่น
2. แบบกราฟฟิก (Graphic) จะสามารถแสดงผลในแบบบิตแมพ (Bit map) คือจะสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ

แนวทางในการใช้งานของทั้งสองแบบจะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบดอตแมทริกซ์มากกว่า เนื่องจากราคาถูกและเพียงพอต่อการใช้งานส่วนใหญ่

คุณสมบัติของดอตแมทริกซ์

คุณสมบัติของดอตแมทริกซ์แบบแอลซีดีโมดูล สามารถสรุปเป็นข้อ ๆ ได้ดังนี้

1. มีให้เลือกหลายรุ่นตามความต้องการใช้งาน โดยมีจำนวนตัวอักษร และบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วยดอตแมทริกซ์ขนาด 5*8 จุด
3. สามารถต่อไมโครคอนโทรลเลอร์เข้ากับระบบได้ 2 ลักษณะคือต่อแบบ memory map

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของ บริษัท อีอีอี เทคโนโลยี จำกัด ขอสงวนสิทธิ์ในสิ่งที่ปรากฏ ไม่สามารถรับผิดชอบต่อการใช้งาน

ไม่ว่ากรณีใด ๆ ที่ใช้งานง่ายและสะดวก เพียงให้ระบบไมโครคอนโทรลเลอร์ส่งข้อมูลให้กับแอลซีดีโมดูล

เท่านั้น ข้อความก็จะปรากฏบนแผงแสดงผล และค้างไว้ตลอด

5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวก เช่น CLEADISPLAY, HOME , CHARACTER
6. สามารถแสดงผลเป็นตัวอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสไฟน้อย มีน้ำหนักรวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์เท่านั้น

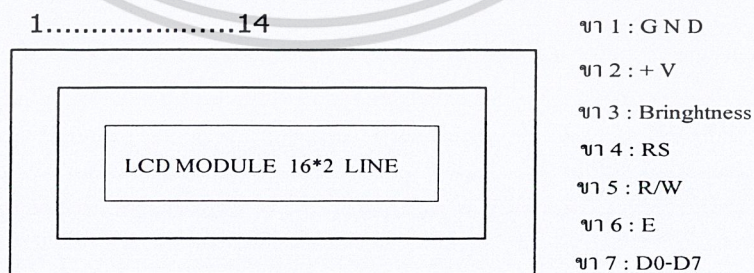
ในแอลซีดีโมดูลจะประกอบด้วยส่วนหลัก ๆ 3 ส่วน ดังนี้

1. ตัวแสดงผล (display) ภายในจะเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอแอลซีดี
2. ตัวควบคุม (control) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของแอลซีดีโมดูล เช่น ลบจอภาพ แสดงตัวอักษรหรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้คือเบอร์ HD44780 สำหรับควบคุมแอลซีดีแบบอักษร และ HD-6180 ใช้ควบคุมแอลซีดีแบบกราฟฟิก
3. ตัวขับ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับได้แก่เบอร์ HD-44100H และ MSM5259 เป็นต้น

ขาสัญญาณของแอลซีดีโมดูล

แอลซีดีโมดูลที่เลือกใช้ในโครงงานเป็นแบบ 16 ตัวอักษร 2 บรรทัด เพราะเป็นแอลซีดีโมดูลที่มีโครงสร้างเป็นมาตรฐาน มีผู้ผลิตหลายราย และมีการระบุเบอร์แตกต่างกันออกไปตามผู้ผลิต เช่น LM 020L ของฮิตาชิ DMC-16117A ของคอปแทร์รี่ เป็นต้น แต่อย่างไรก็ตามคอนโทรลเลอร์ที่ใช้จะเป็นเบอร์เดียวกัน คือ HD44780 ของฮิตาชิ

รูปร่างและการจัดขาสัญญาณของแอลซีดีโมดูลแสดงได้ดังรูปที่ 3.19



รูปที่ 3.19 รูปร่างและการจัดขาสัญญาณของแอลซีดีโมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอลซีดีโมดูลขนาด 16*2 มีขาสัญญาณที่ต่อใช้งานทั้งสิ้น 14 ขา รายละเอียดของสัญญาณของแอลซีดีโมดูล แสดงได้ดังตารางที่ 8

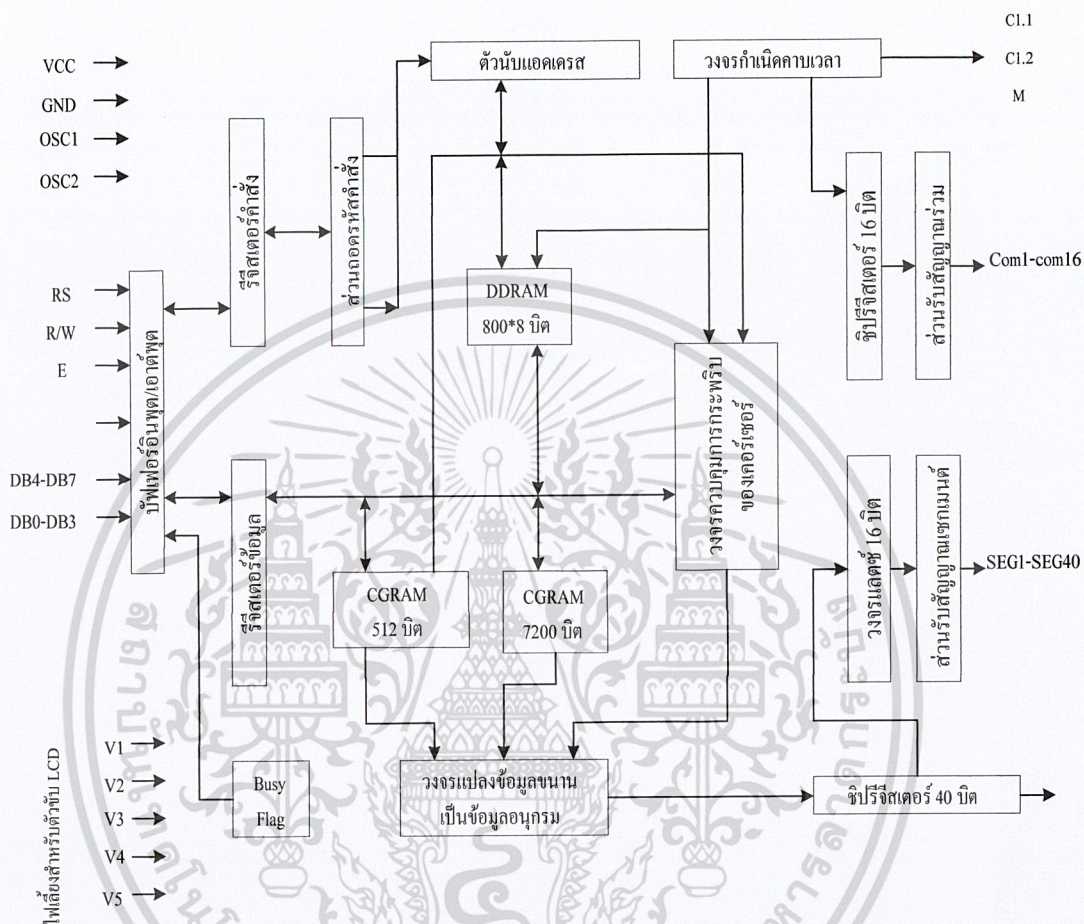
โครงสร้างภายในตัวควบคุมแอลซีดีโมดูล

ในการใช้งานแอลซีดีโมดูล จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมก่อน โดยแอลซีดีโมดูลแบบอักษระสามารถเข้าใจได้ง่าย โครงสร้างภายในแอลซีดีโมดูลแบบอักษระแสดงได้ดังรูปที่ 3.20 เป็นบล็อกไดอะแกรมภายในชิปควบคุมเบอร์ HD-44780 ซึ่งใช้ในแอลซีดีโมดูลแบบอักษระ

ตารางที่ 8 แสดงขาสัญญาณของแอลซีดีโมดูล

ขา	สัญลักษณ์	ระดับ	หน้าที่
1	VSS	-	0 V, GND
2	Vcc	-	+5V POWERSUPPLY
3	Vee	-	+V FOR LIQUID CRYSTAL DRIVE
4	RS	H/L	REGISTER SELECT H: data input L: introduction input H: data read L: data write
5	R/W	H/L	Enable singnal
6	E	L	Data Bit bit 1
7	DB0	H/L	Data Bit bit 2
8	DB1	H/L	Data Bit bit 3
9	DB2	H/L	Data Bit bit 4
10	DB3	H/L	Data Bit bit 5
11	DB4	H/L	Data Bit bit 6
12	DB5	H/L	Data Bit bit 7
13	DB6	H/L	Data Bit bit 8
14	DB7	H/L	Data Bit bit 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 ไดอะแกรมการทำงานของแอลซีดีโมดูลแบบอักขระ

โครงสร้างภายในแอลซีดีโมดูลแบบอักขระประกอบไปด้วย

1. บัฟเฟอร์อินพุต/เอาต์พุต (I/O Buffer) เป็นที่ใช้ในการติดต่อรับส่งกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม
2. รีจิสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจิสเตอร์ที่ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล
3. รีจิสเตอร์ข้อมูล (Data Register:DR) เป็นรีจิสเตอร์ที่ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดไปยังหน่วยความจำ ที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้าง

เอกสารนี้เป็นเอกสารตัวอักษรเพิ่มเติมในเล่มเก็บตัวอักษรศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แรมเก็บข้อมูลแสดงผล (Display Data RAM:DDRAM) เป็นหน่วยความจำที่ทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ข้อมูล ตัวควบคุมจะนำข้อมูลในแรมเก็บข้อมูลแสดงผลนี้ไปเปิดตารางของตัวอักษรที่เก็บไว้ในหน่วยความจำรวมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล
5. รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรวมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงผลได้ มีขนาด 7200 บิต โดยจะอ่านด้วยค่าของข้อมูลแสดงผล
6. แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บตัวอักษรที่มีการสร้างเพิ่มเติมเข้าไปใหม่ในกรณีที่ตัวอักษรในรอมเก็บตัวอักษรไม่เพียงพอมีขนาด 512 บิต การเขียนและการอ่านค่าไปใช้นั้นจะทำงานเช่นเดียวกับรอมเก็บตัวอักษร คือ เขียนข้อมูลลงในแรมเก็บข้อมูลแสดงผล แล้วตัวควบคุมจะมาอ่านข้อมูลจากแรมเก็บตัวอักษรเอง
7. แฟล็ก Busy เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งได้หรือไม่ ดังนั้นก่อนส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟล็ก busy นี้เสียก่อน

การต่อแอลซีดีโมดูลเข้ากับไมโครคอนโทรลเลอร์

แอลซีดีโมดูลสามารถต่อไมโครคอนโทรลเลอร์เข้ากับระบบได้ 2 ลักษณะ คือ ต่อแบบเมโมรีแมพ (memory map) และแบบผ่านพอร์ตของไอซี 8255 ในโครงงานนี้จะใช้การต่อแอลซีดีโมดูลแบบอินพุต/เอาต์พุตพอร์ต ซึ่งมีลักษณะดังนี้

1. สามารถต่อเข้ากับอินพุต/เอาต์พุตพอร์ตใด ๆ ก็ได้โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมาให้ตรงกับแอลซีดีโมดูล
2. ผู้ใช้จะเขียนข้อมูลให้แอลซีดีโมดูลได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนดหน่วยความจำส่วนหนึ่งให้เสมือนว่าเป็นบัฟเฟอร์ให้กับแอลซีดีโมดูล
3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบเองเพื่อรอให้แอลซีดีโมดูลกระทำกระบวนการต่าง ๆ
4. ใช้ได้กับแผงวงจรทั่วไป ๆ ไปที่มีพอร์ต
5. ไม่เปลืองหน่วยความจำในการใช้งาน

6. การจัดการสัญญาณกระทำได้อย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 9 แสดงชุดคำสั่ง

INSTRUCTION	RS	R/W									EXECUTE TIME	
			7	6	5	4	3	2	1	0		
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MONE SET	0	0	0	0	0	0	0	1	I/D	S		40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B		40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*		40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*		40
SET CGRAM ADD	0	0	0	1								40
SET DDRAM ADD	0	0	1									40
BUSY,ADD, READ	0	1	BF									40
CGRAM, DDRAM WR	1	0										40
CGRAM, DDRAM RD	1	1										40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งควบคุมและการแสดงข้อความ

ขาสัญญา VEE มีไว้สำหรับกำหนดความเข้มของตัวอักษรโดยถ้าต่อกับกราวด์จะมีความเข้มของตัวอักษรสูงสุด แต่ถ้าต่อกับไฟจะมีความเข้มต่ำสุด ปกติแล้วแอลซีดีรุ่นธรรมดาอาจต่อกับกราวด์ไว้เลยก็ได้ไม่จำเป็นต้องใส่ตัวต้านทานปรับค่าได้ แต่ถ้าเป็นรุ่น STN (มูมมอกร้าง) ให้ใช้ตัวต้านทาน 2 กิโลโอห์มต่อลงกราวด์อีกทีเพื่อให้ความเข้มมีความเหมาะสม การเขียนหรือการอ่านข้อมูลกับแอลซีดีโมดูลจะเป็นการกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของแอลซีดีโมดูลตามชุดคำสั่งควบคุมและรวมไปถึงการเขียนข้อมูลที่เป็นข้อความเพื่อให้ปรากฏบนแผงแสดงด้วย

ความเข้าใจพื้นฐานเกี่ยวกับแอลซีดีโมดูล

1. การเขียนข้อมูลให้กับแอลซีดีโมดูลจะแบ่งเป็น 2 ลักษณะ คือ ส่วนเริ่มต้น และส่วนข้อมูล โดยจะกำหนดขาสัญญา RS คือถ้า $RS=0$ จะหมายถึง ส่งสัญญาณควบคุม ส่วนเริ่มต้นของแอลซีดีโมดูลหรือค่าเฟล็กของสภาพการทำงานของแอลซีดีโมดูล และถ้า $RS=1$ จะหมายถึงการเขียน และอ่านข้อมูลกับแอลซีดีโมดูล
2. หลักในการเขียนข้อมูลให้กับแอลซีดีโมดูลนี้ คือ เมื่อมีการเขียนข้อมูลลงไปแล้วตัวแอลซีดีโมดูลจะต้องใช้เวลาในการส่งระดับหนึ่ง (ตามค่า excute time ในตารางที่ 9) ซึ่งระบบไมโครคอนโทรลเลอร์จะสามารถตรวจสอบได้จากเฟล็ก BF (Busy Flag) และถ้าเรียบร้อยแล้วจึงสามารถเขียนข้อมูลอันต่อไปได้ในกรณีที่การต่อวงจรเป็นแบบอินพุต/เอาต์พุตพอร์ต จะไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครคอนโทรลเลอร์ต้องใช้เวลาหน่วงเวลาแทน
3. การเขียนข้อมูลให้กับแอลซีดีโมดูลนี้สามารถทำได้ทั้งแบบ 4 บิต และแบบ 8 บิต โดยในกรณีแบบ 4 บิตจะมีสายสัญญาณข้อมูล 4 เส้น คือ DB4-DB7 (ใช้สำหรับระบบของไมโครคอนโทรลเลอร์แบบ 4 บิต เพื่อประหยัดสาย) การเขียนข้อมูลแบบ 8 บิตสามารถกระทำเหมือนการเขียนข้อมูลแบบ 4 บิตเพียงแต่ให้เขียน 2 ครั้งคือ DB4-DB7 แล้วตามด้วย DB0-DB3 และต้องกำหนดคุณสมบัติตามค่า DL ตามคำสั่งในชุดหน้าที่ของคำสั่งด้วย
4. DDRAM (display Data Ram) คือ หน่วยความจำภายในตัวแอลซีดีโมดูลที่เป็นบัฟเฟอร์ (buffer) ของข้อมูลโดยถ้าเขียนรหัสแอสกีใด ๆ ลงไปในหน่วยความจำนี้จะปรากฏเป็นตัวอักษรที่แผงแสดงทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. CGRAM (Character Generator Ram) คือ หน่วยความจำภายในตัวแอลซีดีโมดูลสำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างเองได้ (8 ตัวอักษร) โดยจะอ้างตำแหน่งได้ทั้งหมด 64 ไบต์ (ได้มาจากจำนวน 8 ตัวอักษรคูณกับจำนวน 8 แถว)

รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

สำหรับการ clear display จะทำให้ได้โดยการเขียนตัวอักษร space ลงไปใน DDRAM ทั้งหมดและทำการกำหนดค่าตำแหน่ง DDRAM ให้เป็น 0 พร้อมทั้ง cursor จะกลับไปตำแหน่งซ้ายบนสุดของจอภาพ

ตารางที่ 10 แสดงการกำหนดค่า clear display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

2. CURSOR AT HOME สำหรับกำหนดค่าตำแหน่ง DDRAM ให้เป็น 0 พร้อมทั้ง CURSER จะไปอยู่ที่ตำแหน่งซ้ายสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

ตารางที่ 11 แสดงการกำหนดค่าตำแหน่ง DDRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

3. ENTRY MODE SET

ตารางที่ 12 แสดงการกำหนดค่าตำแหน่ง mode set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	0	0

I/D = 0 กำหนดทิศทางของ cursor และ DDRAM ให้เป็นแบบลดค่า (decrement)

I/D = 1 กำหนดทิศทางของ cursor และ DDRAM ให้เป็นแบบเพิ่มค่า (increment)

S = 0 เมื่อเขียนข้อมูลแล้ว cursor จะถูกเลื่อนไปตามทิศทางของค่า I/D

S = 1 เมื่อเขียนข้อมูล cursor จะอยู่กับที่ และตัวอักษรจะถูกดันไปตามทิศทางของค่า I/D

การกำหนดค่า I/D และ S นี้ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้วจะต้องไม่ใช่คำสั่ง clear display อีก

4. DISPLAY ON/OFF

ตารางที่ 13 แสดงการกำหนดค่า ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	1	0	0

D = 0 กำหนดให้ปิดการแสดงผล

D = 1 กำหนดให้เปิดการแสดงผล

C = 0 กำหนดให้ปิด cursor

C = 1 กำหนดให้ cursor เป็นเส้นใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่ง cursor (กระพริบเป็นรูปสี่เหลี่ยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. DISPLAY SHIFT

ตารางที่ 14 แสดงการกำหนดค่าการเลื่อน

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

S/C = 0 กำหนดให้ cursor R/L

S/C = 1 กำหนดให้ R/L

R/L = 0 กำหนดให้มีทิศทางการเลื่อนไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางการเลื่อนไปทางขวา

6. FUNCTION SET (ในกรณีที่แอลซีดีโมดูลเป็นแบบ 5*7 อยู่แล้วจะไม่มีผลอะไร)

ตารางที่ 15 แสดงการกำหนดค่าการเซตฟังก์ชัน

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้การติดต่อกับแอลซีดีโมดูลเป็นแบบ 4 บิต

DL = 1 กำหนดให้การติดต่อกับแอลซีดีโมดูลเป็นแบบ 8 บิต จะสังเกตว่าการกำหนดค่า D/L นี้สามารถกระทำได้ที่ DB4-DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 บิตตั้งแต่แรกหลัง

จากจ่ายไฟเลี้ยงก็จะทำให้แอลซีดีโมดูลมีการรับข้อมูลเป็นแบบ 4 บิตอีกที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 duty และ 1/6 duty

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ลิขสิทธิ์ทั้งหมดเป็นของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 duty

F = 0 กำหนดจำนวนอักษรเป็นแบบ 5*7 จุด

F = 1 กำหนดจำนวนตัวอักษรเป็นแบบ 5*10 จุด

7. SET CGRAM ADDRESS

สำหรับการกำหนดตำแหน่งของ CGRAM เมื่อได้ทำการกำหนดไว้แล้วการอ่าน และเขียน ข้อมูลที่ต่อกรนี้จะ เป็นไปตามตำแหน่งที่กำหนดทันที

ตารางที่ 16 แสดงการทำตำแหน่งของ CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

8. SET DDRAM ADDRESS

ตารางที่ 17 แสดงการกำหนดค่าตำแหน่งของ DDRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

สำหรับการกำหนดค่าตำแหน่งของ DDRAM เมื่อได้ทำการกำหนดไว้แล้วการอ่าน และการเขียนข้อมูลที่จะเป็นไปตามตำแหน่งที่กำหนดทันที ตำแหน่งในแต่ละรุ่นมีความแตกต่างกันบ้าง เพราะจำนวนตัวอักษรต่อบรรทัดไม่เท่ากัน ซึ่งแสดงตามตารางที่ 15

(ตารางที่ 17 นี้จะกำหนดให้บิตที่ 7 เป็นลอจิก "1" เสมอเพื่อความสะดวกในการเลือกใช้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 18 แสดงการกำหนดตำแหน่งของ DDRAM ที่ใช้งาน

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

9. BUSY FLAG AND ADDRESS READ

ตารางที่ 19 แสดงการกำหนดค่า

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	ADDRESS							

สำหรับการอ่านแฟล็ก BF ซึ่งบอกถึงความพร้อมของแอสซีดีโมดูลในการรับข้อมูล

BE = 0 หมายความว่าพร้อมที่จะรับข้อมูลต่อไป

BF = 1 หมายความว่ายังไม่พร้อม

นอกจากนี้ยังเป็นการอ่านค่าตำแหน่งของ CGRAM หรือ DDRAM ด้วย

การอ่านและการเขียนข้อมูลกับ DDRAM/CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

ตารางที่ 20 แสดงการเขียนข้อมูลลง DDRAM หรือ CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	DATA							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเขียนข้อมูลลงในหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้วตำแหน่งจะถูกเพิ่มขึ้นหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง entry mode set และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้านี้นี้คำสั่งนี้มีการกำหนดตำแหน่งที่ตำแหน่งใด

2. READ DATA TO DDRAM OR CGRAM

ตารางที่ 21 แสดงการอ่านข้อมูลลง DDRAM หรือ CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	DATA							

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการอ่านแล้วตำแหน่งจะถูกเพิ่มขึ้นหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง entry mode set และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้านี้นี้คำสั่งนี้มีการกำหนดตำแหน่งที่ตำแหน่งใด

แนวทางการเขียนโปรแกรมควบคุม

- เมื่อจ่ายไฟเลี้ยงให้กับแอลซีดีโมดูลครั้งแรก ภายในจะมีการ reset ระบบโดยอัตโนมัติ ซึ่งจะใช้เวลาประมาณ 10 มิลลิวินาทีหลังจากที่ระดับแรงดันไฟขึ้นถึง 4.5 โวลต์แล้ว ทั้งนี้ระบบรีเซ็ตดังกล่าวจะกระทำสิ่งต่าง ๆ
- ทำการ clear จอภาพทั้งหมด
- กำหนดคุณสมบัติด้วยคำสั่ง function set คือ DL = 1 (ติดต่อกับระบบไมโครคอนโทรเลอร์ในแบบ 8 บิต) N = 0 (แสดงข้อมูล 1 บรรทัด) F = 0 (กำหนดตัวอักษรแบบ 5*7)
- กำหนดคุณสมบัติด้วยคำสั่ง Display ON/OFF คือ D = 0 (ไม่แสดงข้อมูล) C = 0 (cursor OFF) B = 0 (blank OFF)

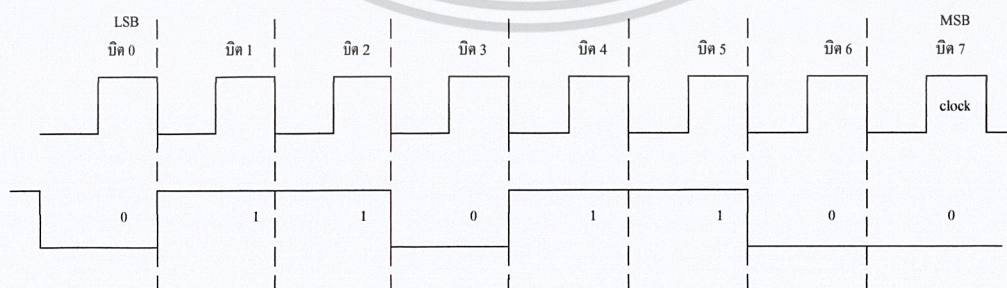
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ที่ละเมิดลิขสิทธิ์ทางปัญญาและสงวนลิขสิทธิ์ไว้ทุกประการนำไปใช้

- การใช้งานแอลซีดีโมดูลต้องรอให้ขบวนการรีเซ็ตภายในทำงานให้เรียบร้อยก่อน ซึ่งจะตรวจสอบได้ด้วยแฟล็ก BF หรืออาจจะใช้การหน่วงเวลาก็ได้
- การใช้งานแอลซีดีโมดูลจะต้องเกี่ยวข้องกับทางด้านโปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่าง ๆ รวมทั้งการอ่าน และเขียนข้อมูลนั้นจะต้องถูกกำหนดด้วยขาสัญญาณทั้งหมดที่มีอยู่ โดยปกติโปรแกรมจะต้องกำหนดคุณสมบัติต่าง ๆ ที่ต้องการไว้ ที่ส่วนร่วมเริ่มต้น จากนั้นจะเป็นการเขียนข้อมูลลงในส่วนข้อมูล ซึ่งก็คือข้อความที่จะให้แสดงผลนั่นเอง

3.8 การสื่อสารแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลได้คราวละหลาย ๆ บิตได้ หากแต่ต้องมีการตกลงกันระหว่างตัวส่ง และตัวรับว่าจะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล และส่งผลให้การสื่อสารข้อมูลอนุกรมอาจมีความเร็วต่ำกว่าแบบขนาน ในด้านจำนวนสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายที่น้อยกว่ามาก อย่างไรก็ตามการรับส่งข้อมูลแบบอนุกรมสามารถใช้สายสัญญาณที่มีความยาวมากกว่าแบบขนาน ทำให้ระยะทางในการสื่อสารข้อมูลแบบอนุกรมสามารถทำได้มากกว่า

การสื่อสารแบบอนุกรมนี้จะแบ่งออกได้เป็น 2 แบบ คือ การสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัสการสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาาร่วมอยู่กับการรับและส่งสัญญาณด้วยตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือ บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา ข้อมูลและกราวด์ รูปที่ 3.21 แสดงให้เห็นถึงไทม์มิงไคอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 3.21 รูปแบบอย่างง่ายที่สุดของข้อมูลแบบอนุกรม

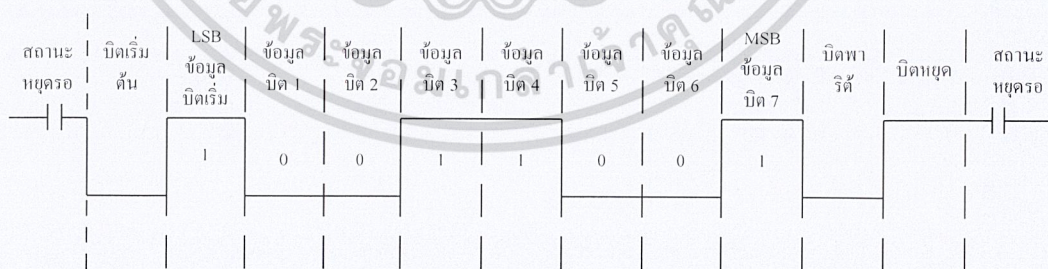
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือ การรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะทำการกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากันซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรตมีหน่วยเป็นบิตต่อวินาที (bitpersecone : bps) รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5 6 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1 1.5 หรือ 2 บิต

รูปที่ 3.22 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัสซึ่งเมื่อไม่มีข้อมูลที่จะส่งหา DATA จะมีสถานะลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้หา DATA มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อนซึ่งข้อมูลในไบต์ที่จะส่งอาจมีจำนวนบิต 5 , 6 , 7 หรือ 8 บิตก็ได้จากนั้นจะตามด้วยบิตพาริตี ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูลบิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งก็จะทำให้ที่ขาดามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต 1.5 บิตหรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



รูปที่ 3.22 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับ และ ส่งข้อมูลแบบอะซิงโครนัสเรียก

ว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับ และ ส่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส คือ ค่าบอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดเบลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110 150 300 600 1200 2400 4800 9600 และ 19200 บิตต่อวินาทีและมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาทีเนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่างข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิตและบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (odd) แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตีก็ได้ การตรวจสอบพาริตีเป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก “1” ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตีเข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก “1” จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตีเป็นคู่ค่าในบิตพาริตี จะต้องมิลอจิกเป็น “0” แต่ถ้าพาริตีเป็นคี่ ค่าที่บิตพาริตีจะต้องเป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตีมีจำนวนบิตที่เป็นลอจิก “1” มีจำนวนรวมกันเป็นเลขคี่ ในตารางที่ 22 แสดงตัวอย่างของบิตพาริตีในการรับส่งข้อมูลอนุกรม

ตารางที่ 22 แสดงบิตพาริตีของข้อมูล

ข้อมูลบิต	บิตพาริตีคู่	บิตพาริตีคี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตีให้ตรงกันว่าจะตรวจสอบพาริตีคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือคี่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งค่าบิตพาริตี

ด้วย ถ้ากำหนดพาริตีไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ทางภาครับจะแสดงค่าผิดพลาดออกมาให้ผู้ใช้งาน นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น NONE นั้นทั้งภาครับ และภาคส่งจะไม่มี การตรวจสอบพาริตี

คอมพิวเตอร์รุ่น AT เกือบทั้งหมดจะใช้ UART เบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ UART เบอร์ 8250 UART ชิปเหล่านี้มีระดับแรงดันเป็นแบบ TTL (0 และ +5V) แต่เพื่อให้แรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ในระยะทางไกลมากขึ้น ระดับแรงดัน TTL จะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดยลอจิก "0" มีระดับแรงดัน +3 ถึง +12V ในขณะที่ลอจิก "1" มีระดับแรงดัน -3 จนถึง -12V

มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดหนึ่งซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ ได้วางมาตรฐานที่มีชื่อเรียกว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะเป็นคอนเนคเตอร์แบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12 V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12 แสดงว่าเป็นช่องว่าง (Space)

มาตรฐานของ RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่ได้เห็นได้ชัดคือ คอนเนคเตอร์เอกสารนี้ของ DTE จะเป็นตัวผู้ ส่วนคอนเนคเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ไม่ว่าที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเนคเตอร์ที่อยู่ทีโมเด็มจะเป็นแบบ DCE ที่มีการนำไปใช้

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็ม หรือ เมาส์ โดยสามารถรับส่งข้อมูลได้ถึงความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงดังรูปร่าง และตำแหน่งขาในรูปที่ 3.23

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready:DTR	เอาต์พุต
5	7	Signal Ground:GND	-
6	6	Data Set Ready:DSR	อินพุต
7	4	Request To Send:RTS	เอาต์พุต
8	5	Clear To Send:CTS	อินพุต
9	22	Ring Indicator:RI	อินพุต

รูปที่ 3.23 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาและเรียนรู้เท่านั้น ไม่สามารถนำไปใช้ในเชิงพาณิชย์ได้
 ไม่ว่าลิขสิทธิ์ในสิ่งที่มีอยู่ในรูปที่ 3.24 (ก) เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดย

ตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนรูปที่ 3.24 (ข) Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นหนึ่งสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232

- Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกติฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก
- Receive Data : RD หรือ RxD ขานี้เพื่อใช้รับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- Transmitted Data : TD หรือ TxD ขานี้เพื่อใช้ส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วยโดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และต้องต่อกับขา DCD ด้วยในกรณีที่ใช้โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์
- Signal Ground : GND ขากราวด์ของระบบ
- Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
- Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS และ CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อกับขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายใน UART มีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable baudrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงกำหนดตัวหารอยู่ในช่วง 1-65,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (Full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับ และส่งข้อมูลได้ในคราวเดียวกัน

ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์ คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมียูนิฟิเคชันสำหรับรับ และส่งข้อมูลตำแหน่งเดียวกันทำให้การรับ และส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 5706 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ถือถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุก ๆ รุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

UART อีกเบอร์หนึ่ง คือ 16450 มีความสามารถรับส่งข้อมูลได้ด้วยความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนี้ยังเพิ่มส่วนของชิพรีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไปทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่าเช่น เบอร์ TL16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีโหมดประหยัดพลังงานสามารถรับส่งข้อมูลได้ด้วยความเร็ว 1 เมกกะบิตต่อวินาที เมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โค่นทั่วไปสามารถต่อพอร์ตอนุกรม RS-232 สูงสุดได้ 4 พอร์ต ซึ่งจะมีชื่อเรียกเป็น COM1, COM2, COM3, COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน

ในรูปที่ 3.25 แสดงไดอะแกรมการทำงานภายในพอร์ตอนุกรม ซึ่งประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของภายในพอร์ตอนุกรมสามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM1 มีแอดเดรสอยู่ที่

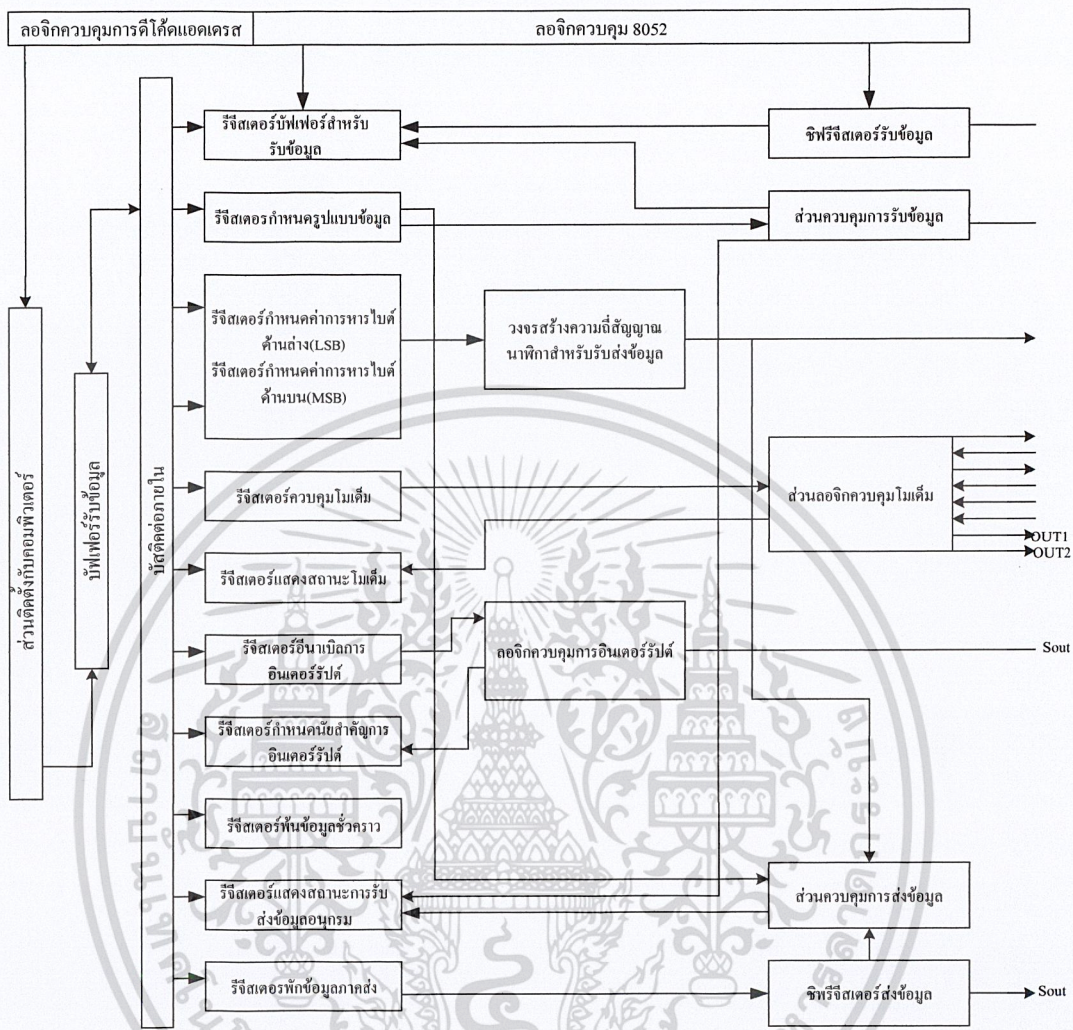
3F8H ค่าแห่งของรีจิสเตอร์ต่าง ๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้ทำงานกับพอร์ตอนุกรมมีดังนี้

ที่ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลที่รับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออก
- 01H รีจิสเตอร์อินาเบิลการอินเตอร์รัปต์ในการเซตโหมดอินเตอร์รัปต์ของพอร์ตอนุกรม
- 02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ใช้เพื่อตรวจสอบโหมดของการอินเตอร์รัฟเมื่อมีการอินเตอร์รัฟเกิดขึ้น
- 03H รีจิสเตอร์กำหนดรูปแบบข้อมูล
- 04H รีจิสเตอร์ควบคุมโมเด็ม ตรวจสอบบิตสำหรับติดต่อกับโมเด็ม คือ RTS หรือ DTR
- 05H รีจิสเตอร์แสดงสถานะการรับส่งข้อมูลแบบอนุกรม
- 06H รีจิสเตอร์แสดงสถานะของโมเด็ม จะแสดงสถานะของขา DCD, RI,DSR และ CTR
- 07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 ไคอะแกรมการทำงานภายในของพอร์ตอนุกรมของเครื่องคอมพิวเตอร์

รีจิสเตอร์ตำแหน่ง 00H : รีจิสเตอร์บัพเฟอร์

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่รับเข้ามาและข้อมูลที่ส่งออกไปโดยการติดต่อกับรีจิสเตอร์นี้เพื่อเก็บข้อมูลที่ต้องการจะส่งจะต้องกำหนดให้บิต DLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล (03H) จะต้องมีสถานะเป็น 0 ซึ่งการเขียนข้อมูลมายังแอดเดรสนี้ เป็นการส่งข้อมูลไปยังรีจิสเตอร์ส่งข้อมูลและข้อมูลถูกส่งออกไปแบบอนุกรม สำหรับการรับข้อมูลเมื่อข้อมูลที่รับเข้ามาเรียบร้อยแล้วจะแปลงเป็นแบบขนานแล้ว ข้อมูลจะถูกส่งมายังรีจิสเตอร์เก็บข้อมูล หลังจากมีการอ่านรีจิสเตอร์นี้ออกไปรีจิสเตอร์นี้จะถูกเคลียร์และเตรียมพร้อมสำหรับการรับข้อมูลในไปต่อไป

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปบนเว็บไซต์ การค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ตำแหน่ง 01H : รีจิสเตอร์อีนามิเตอร์อินเทอร์รัปต์

เป็นรีจิสเตอร์สำหรับการอีนามิเตอร์อินเทอร์รัปต์ ซึ่งเป็นการกำหนดให้ UART สร้างสัญญาณอินเทอร์รัปต์ขึ้นมา ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์มีดังนี้ ตารางที่ 23 แสดงรีจิสเตอร์อีนามิเตอร์อินเทอร์รัปต์

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	SINP	ERBK	TBE	RxRD

บิตที่ 4-7 บิตเหล่านี้ไม่ถูกใช้งาน กำหนดให้เท่ากับ “0”

SINP อีนามิเตอร์อินเทอร์รัปต์เนื่องจากเกิดการเปลี่ยนสถานะที่ขาอินพุต

CTS, DSR, DCD หรือ XRI

“1” อีนามิเตอร์อินเทอร์รัปต์

“0” ไม่มีการใช้อินเทอร์รัปต์รูปแบบนี้หรือดีสเอเบิล

ERBK อีนามิเตอร์อินเทอร์รัปต์เนื่องจากเกิดความผิดพลาดขึ้นด้วยสาเหตุจากพาริตี โอเวอร์รัน เพรมข้อมูล หรือ การเบกข้อมูล

“1” อีนามิเตอร์อินเทอร์รัปต์

“0” ไม่มีการใช้อินเทอร์รัปต์รูปแบบนี้หรือดีสเอเบิล

TBE อีนามิเตอร์อินเทอร์รัปต์เมื่อรีจิสเตอร์บัพเฟอร์สำหรับส่งข้อมูลว่าง

“1” อีนามิเตอร์อินเทอร์รัปต์

“0” ไม่มีการใช้อินเทอร์รัปต์รูปแบบนี้หรือดีสเอเบิล

RxRD อีนามิเตอร์อินเทอร์รัปต์เนื่องจากรีจิสเตอร์บัพเฟอร์ได้รับข้อมูลเรียบร้อยแล้ว

“1” อีนามิเตอร์อินเทอร์รัปต์

“0” ไม่มีการใช้อินเทอร์รัปต์รูปแบบนี้หรือดีสเอเบิล

รีจิสเตอร์ตำแหน่ง 02H : รีจิสเตอร์แสดงโหมดและสถานะการอินเทอร์รัปต์

มีรายละเอียดของแต่ละบิตดังนี้

ตารางที่ 24 แสดงรีจิสเตอร์โหมดและสถานะการอินเทอร์รัปต์

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	ID1	ID0	PND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต3-7 ไม่ได้ใช้งาน อ่านค่าได้เท่ากับ “0”

ID1 . ID0 ใช้งานร่วมกันเพื่อแจ้งสาเหตุของการเกิดอินเทอร์รัปต์

“00” เกิดการอินเทอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตขึ้นการเกิดอินเทอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 4

“01” เกิดการอินเทอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่างขึ้นกสนอินเทอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 3

“10” เกิดการอินเทอร์รัปต์จากข้อมูลเก็บลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อยแล้ว การอินเทอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 2

“11” เกิดการอินเทอร์รัปต์เนื่องจากความผิดพลาดในการถ่ายทอดข้อมูลหรือเกิดการเบรก (break : เกิดการหยุดถ่ายทอดข้อมูลกะทันหัน) การอินเทอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 1 หรือมีนัยสำคัญต่ำสุดPND ใช้แสดงสถานะของการเกิดอินเทอร์รัปต์

“1” แสดงว่าไม่มีการอินเทอร์รัปต์

“0” แสดงว่ามีการเกิดอินเทอร์รัปต์ขึ้น

เมื่อมีการสร้างสัญญาณอินเทอร์รัปต์ขึ้น จะต้องมีการเคลียร์ค่าก่อนที่จะเกิดอินเทอร์รัปต์ครั้งต่อไปทำได้ดังนี้ คือ

- ถ้าเกิดอินเทอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตจะต้องอ่านค่าจากรีจิสเตอร์แสดงสถานะของโมเด็ม (รีจิสเตอร์ตำแหน่ง 06H) เพื่อเคลียร์ค่าการอินเทอร์รัปต์
- ถ้าเกิดการอินเทอร์รัปต์เนื่องจากบัฟเฟอร์ส่งข้อมูลว่าง จะต้องเขียนข้อมูลไปยังรีจิสเตอร์บัฟเฟอร์ส่งข้อมูล (รีจิสเตอร์ตำแหน่ง 00H) หรืออ่านค่ารีจิสเตอร์แสดงสถานะอินเทอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อเคลียร์ค่าการอินเทอร์รัปต์
- ถ้าเกิดการอินเทอร์รัปต์เนื่องจากการเก็บข้อมูลลงในรีจิสเตอร์บัฟเฟอร์สำหรับข้อมูลเรียบร้อยแล้วจะต้องเคลียร์ค่าอินเทอร์รัปต์โดยการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์
- ถ้าเกิดการอินเทอร์รัปต์เนื่องจากความผิดพลาดในการรับส่งข้อมูลหรือเกิดการเบรกต้องเคลียร์ค่าอินเทอร์รัปต์โดยการอ่านค่ารีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลแบบอนุกรม

รีจิสเตอร์ตำแหน่ง 03H : รีจิสเตอร์กำหนดรูปแบบของข้อมูล

มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 25 แสดงรีจิสเตอร์กำหนดรูปแบบของข้อมูล

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
DLAB	BRK	PAR2	PAR1	PAR0	STOP	DAB1	DAB0

DLAB ใช้ในการกำหนดหน้าที่การทำงานของรีจิสเตอร์บัพเฟอร์ (00H)

“1” เป็นการเข้าสู่โหมดการหารค่าบอกรेट

“2” เป็นการเข้าถึงค่ารีจิสเตอร์บัพเฟอร์ (รีจิสเตอร์ตำแหน่ง 00H) และรีจิสเตอร์สำหรับอินทิเกรตอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 01H)

เมื่อบิต DLAB เป็น “1” รีจิสเตอร์บัพเฟอร์ (00H) และรีจิสเตอร์อินทิเกรตอินเตอร์รัปต์ (01H) จะใช้สำหรับโหลดค่าการหารความถี่สำหรับกำหนดค่าบอกรेट โดยมีรีจิสเตอร์ 00H เก็บค่าตัวหารไบต์ต่ำ ส่วนรีจิสเตอร์ 01H ใช้เก็บค่าตัวหารไบต์สูง การหาค่าบอกรेटสามารถเขียนเป็นสมการได้ดังนี้

$$\text{บอกรेट} = 115200 / \text{ค่าตัวหาร 16 บิต}$$

ค่าตัวเลข 115200 มาจากความถี่ของคริสตอลในวงจร UART ภายในเครื่องคอมพิวเตอร์ โดยคริสตอลที่ใช้มีความถี่ 1.8432 MHz วงจรภายใน UART จะทำการหาค่าความถี่นี้ด้วย 16 ทำให้มีค่าความถี่ 115200 Hz ออกมา

$$\text{ค่าตัวหาร 16 บิต} = \text{ข้อมูลในรีจิสเตอร์ 00H} + (256 * \text{ข้อมูลในรีจิสเตอร์ 01H})$$

สมมติว่าต้องการค่าบอกรेटเท่ากับ 9600 ค่าตัวหารที่ใช้ต้องเท่ากับ 12 ซึ่งค่านี้จะต้องถูกโหลดลงในรีจิสเตอร์ 00H และโหลดค่า 0 ลงไปในรีจิสเตอร์ 01H ค่าตัวหารที่ทำให้เกิดค่าบอกรेटสูงสุดที่ 115200 บิตต่อวินาที คือค่า 0001 นั่นคือรีจิสเตอร์ 00H มีค่าเท่ากับ 1 และรีจิสเตอร์ 01H มีค่าเท่ากับ “0”

BRK ใช้ควบคุมการหยุดถ่ายข้อมูล

“1” สามารถหยุดหรือเบรกได้

“0” ไม่มีการหยุดหรือเบรกได้

PAR2 , PAR1 , PAR0 ใช้เพื่อกำหนดบิตพาริตี

“000” ไม่ใช่บิตพาริตี

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ “01H” กำหนดพาริตีคู่ แต่แปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“101” มาร์ก (mark)

“111” ช่องว่าง (space)

STOP ใช้กำหนดจำนวนบิตปิดท้าย

“1” มีบิตปิดท้าย 2 บิต

“0” มีบิตปิดท้าย 1 บิต

DAB1 , DAB0 ใช้ร่วมกันในการรวมจำนวนบิตของข้อมูลที่ต้องการถ่ายทอด

“00” จำนวนบิตข้อมูลเท่ากับ 5 บิต

“01” จำนวนบิตข้อมูลเท่ากับ 6 บิต

“10” จำนวนบิตข้อมูลเท่ากับ 7 บิต

“11” จำนวนบิตข้อมูลเท่ากับ 8 บิต

รีจิสเตอร์ตำแหน่ง 04H : รีจิสเตอร์ควบคุมโมเด็ม

มีรายละเอียดของแต่ละบิตดังนี้

ตารางที่ 26 แสดงรีจิสเตอร์ควบคุมโมเด็ม

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	LOOP	OUT2	OUT1	RTS	DTR

บิต 5-7 ไม่มีค่าใช้งาน อ่านค่าได้เท่ากับ 0

LOOP

“1” อินาเบิ้ลการส่งค่ากลับ

“0” ดิสเอเบิ้ล

OUT1 , OUT2

“1” อินาเบิ้ลการใช้งานภายใน

“0” ดิสเอเบิ้ล

RTS ใช้ควบคุมการทำงานของขา RTS (Ready To Send)

“1” อินาเบิ้ล

“0” ดิสเอเบิ้ล

DTR ใช้ควบคุมการทำงานของขา DTR (Data Terminal Ready)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“1” อีนาเบิล

“0” ดิสเอเบิล

รีจิสเตอร์ตำแหน่ง 05H : รีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลของ UART

ใช้งานร่วมกับรีจิสเตอร์แสดงโหมดและสถานะของการอินเทอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อแสดงสาเหตุของการเกิดอินเทอร์รัปต์ มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้ ตารางที่ 27 แสดงรีจิสเตอร์สถานะการรับและการส่งข้อมูล

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	TXE	TBE	BREK	FRME	PARE	OVFE	RxRD

TXE (Transmitter Empty)

“1” แสดงว่ารีจิสเตอร์สำหรับส่งข้อมูลว่าง

“0” แสดงว่ายังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

TBE (Transmitter Buffer Empty)

“1” รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

“0” ยังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

BREK (Break)

“1” UART ตรวจพบการเบรก

“0” ไม่มีการเบรก

FRME (Frame Error)

“1” UART ตรวจพบความผิดพลาดด้านเฟรมข้อมูล

“0” ไม่พบความผิดพลาดด้านเฟรมข้อมูล

PARE (Parity Error)

“1” UART ตรวจพบความผิดพลาดด้านพาริตี

“0” ไม่พบความผิดพลาดทางด้านพาริตี

OVRE (Overrun Error)

“1” UART ตรวจพบความผิดพลาดด้านโอเวอร์รัน

“0” ไม่พบความผิดพลาดแบบโอเวอร์รัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

RxRD (Received Data Ready)

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“1” มีการรับข้อมูลเข้ามาเก็บไว้ในข้อมูลบัฟเฟอร์

“0” ไม่มีข้อมูล

รีจิสเตอร์ 06H : รีจิสเตอร์แสดงสถานะของโมเด็ม

ใช้เพื่อกำหนดสถานะสัญญาณอินพุต ของพอร์ตอนุกรม RS-232 ซึ่งได้แก่สัญญาณ DCD , DSR , CTS และ RI สำหรับการเชื่อมต่อใช้งานแบบอนุกรมประสงค์มีหน้าที่ของแต่ละบิตดังนี้ ตารางที่ 28 แสดงรีจิสเตอร์สถานะของโมเด็ม

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

DCD ใช้แสดงสถานะของขา DCD

“1” แสดงว่าที่ขา DCD เป็นลอจิก “1”

“0” แสดงว่าที่ขา DCD เป็นลอจิก “0”

RI ใช้แสดงสถานะของขา RI

“1” แสดงว่าที่ขา RI เป็นลอจิก “1”

“0” แสดงว่าที่ขา RI เป็นลอจิก “0”

DSR ใช้แสดงสถานะของขา DSR

“1” แสดงว่าที่ขา DSR เป็นลอจิก “1”

“0” แสดงว่าที่ขา DSR เป็นลอจิก “0”

DCTS (Delta Clear To Send) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต CTS

“1” แสดงว่าที่บิต CTS (Clear To Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านครั้งที่แล้ว

“0” แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับครั้งที่แล้ว

DDSR (Delta Data Set Read) ใช้แจ้งการเปลี่ยนแปลงเมื่อเทียบจากการอ่านครั้งที่แล้ว

“1” แสดงว่าบิต DSR (Data Set Ready) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

“0” แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DRI (Delta Ring Indicator) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต RI

“1” แสดงว่าบิต RI (Ringing Indicator) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปยังเว็บไซต์อื่น การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“0” แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DDCD (Delta Data Carrier Detect) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DDCD

“1” แสดงว่าบิต CTS (Clear To Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

“0” แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DCTS (Delta Clear To Send) ใช้แสดงสถานะของขา CTS

“1” แสดงว่าที่ขา CTS เป็นลอจิก “1”

“0” แสดงว่าที่ขา CTS เป็นลอจิก “0”

รีจิสเตอร์ตำแหน่ง 07H : รีจิสเตอร์สำหรับเก็บข้อมูลชั่วคราว

ทำหน้าที่เป็นหน่วยความจำแรมขนาด 1 ไบต์ การอ่าน และเขียนข้อมูลที่รีจิสเตอร์ตัวนี้ไม่ส่งผลใด ๆ ต่อการใช้งาน UART

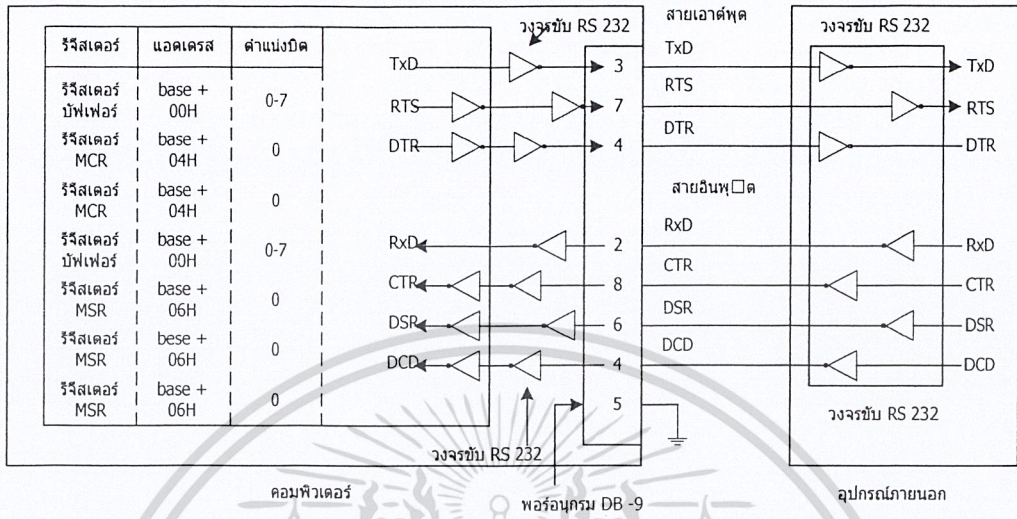
ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณอินพุตและเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และแสดงสถานะอินพุต (CTS , DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่ง และภาครับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตเป็นแบบ TTL เท่านั้น ดังนั้นเมื่อระดับสัญญาณถูกส่งออกจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็ต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อที่จะให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้น จะถูกกลับสถานะ ดังแสดงเป็นบล็อกไดอะแกรมรูปที่ 3.26

แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

- COM1:3F8H
- COM2:2F8H
- COM3:3F8H
- COM4:2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1 จะเก็บไว้ที่แอดเดรส 0000H:0400H และ 0000H:0401H ส่วนตำแหน่งอื่น ๆ มีรายละเอียดดังนี้

- COM2 = 0000H : 0402H – 0000H : 0403H
- COM3 = 0000H : 0404H – 0000H : 0405H
- COM4 = 0000H : 0406H – 0000H : 0407H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000H : 0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีอยู่ในคอมพิวเตอร์อีกด้วย โดยมีรายละเอียดดังแสดงในตารางที่ 29

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 29 แสดงข้อมูลในแอดเดรส 0000H : 0411H ที่ใช้แจ้งจำนวนพอร์ต

บิต 3	บิต 2	บิต 1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรมหนึ่งพอร์ต
0	1	0	มีพอร์ตอนุกรมสองพอร์ต
0	1	1	มีพอร์ตอนุกรมสามพอร์ต
1	0	0	มีพอร์ตอนุกรมสี่พอร์ต

การเขียนโปรแกรมเพื่องานพอร์ตอนุกรม

การหาค่าตำแหน่งแอดเดรสของพอร์ตอนุกรม

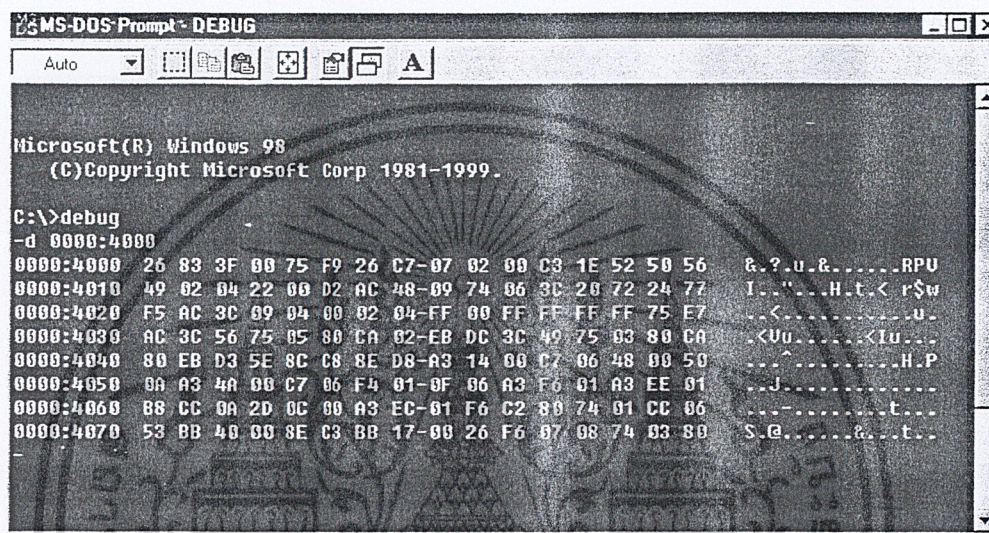
การหาค่าตำแหน่งแอดเดรสของพอร์ตอนุกรมสามารถทำได้หลายวิธี

วิธีที่ 1 โดยการ ใช้โปรแกรม Debug ไปดูค่าตำแหน่งแอดเดรสที่ตำแหน่ง 0000H:0400H โดยใช้พิมพ์คำสั่งที่คอสมพร้อมดังแสดงในรูปที่ 3.27 ค่าที่เห็นในรูปแสดงว่า พอร์ตอนุกรมของคอมพิวเตอร์เครื่องนี้มีถึง 4 พอร์ต มีตำแหน่งแอดเดรสไล่เรียงกันตั้งแต่หน่วยความจำตำแหน่งที่ 0000H : 0400H - 000H - 0407H ตัวอย่างเช่น ที่ตำแหน่งหน่วยความจำ 0000H : 0400H - 0000H : 0401H แสดงตัวเลข F8 03 ซึ่งหมายความว่าแอดเดรสของพอร์ต COM 1 คือ 03F8H นั่นเอง สำหรับจำนวนของพอร์ตอนุกรมที่ระบุนอยู่ที่หน่วยความจำตำแหน่ง 0000H : 0411H มีค่าเท่ากับ D8H ซึ่งเมื่อแปลงเป็นเลขฐาน 2 จะได้ค่าเป็น 11011000 บิตที่ 1-3 มีค่าเท่ากับ 100 หมายความว่า มีจำนวนพอร์ตอนุกรมทั้งหมด 4 พอร์ตดังที่ได้กล่าวไปแล้วข้างต้น

วิธีที่ 2 สามารถดูได้จากวินโดวส์ 95 โดยไปที่ Control Panel เรียก System => Device Manager => Port (COM&LPT) จากนั้นเลือกพอร์ตอนุกรมที่ต้องการดูค่าแล้วดังแสดงในรูปที่ 3.29 แล้วเลือก Properties => Resources ดังแสดงในรูปที่ 3.28 ซึ่งในหน้าต่างนี้จะแสดงทั้งตำแหน่งแอดเดรสของพอร์ตอนุกรมนั้น ๆ รวมถึงตำแหน่งของอินเตอร์รัปต์ที่ใช้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีที่ 3 การเขียนโปรแกรมเพื่ออ่านค่า โดยสามารถใช้โปรแกรมภาษาใด ๆ ก็ได้เพื่ออ่านค่าแต่ในตัวอย่างนี้จะใช้ QBASIC ในการแสดงค่าแอดเดรสของพอร์ตอนุกรม โดยจะใช้คำสั่ง PEEK ซึ่งเป็นคำสั่งที่ใช้อ่านค่าจากหน่วยความจำ นอกจากนี้ยังอ่านค่าจำนวนของพอร์ตอนุกรมออกมาโดยใช้คำสั่งเดียวกัน แต่จะใช้คำสั่ง AND เข้าเพื่อช่วยเลือกเก้าบิตที่ต้องการอ่านค่าเท่านั้น จากนั้น



```

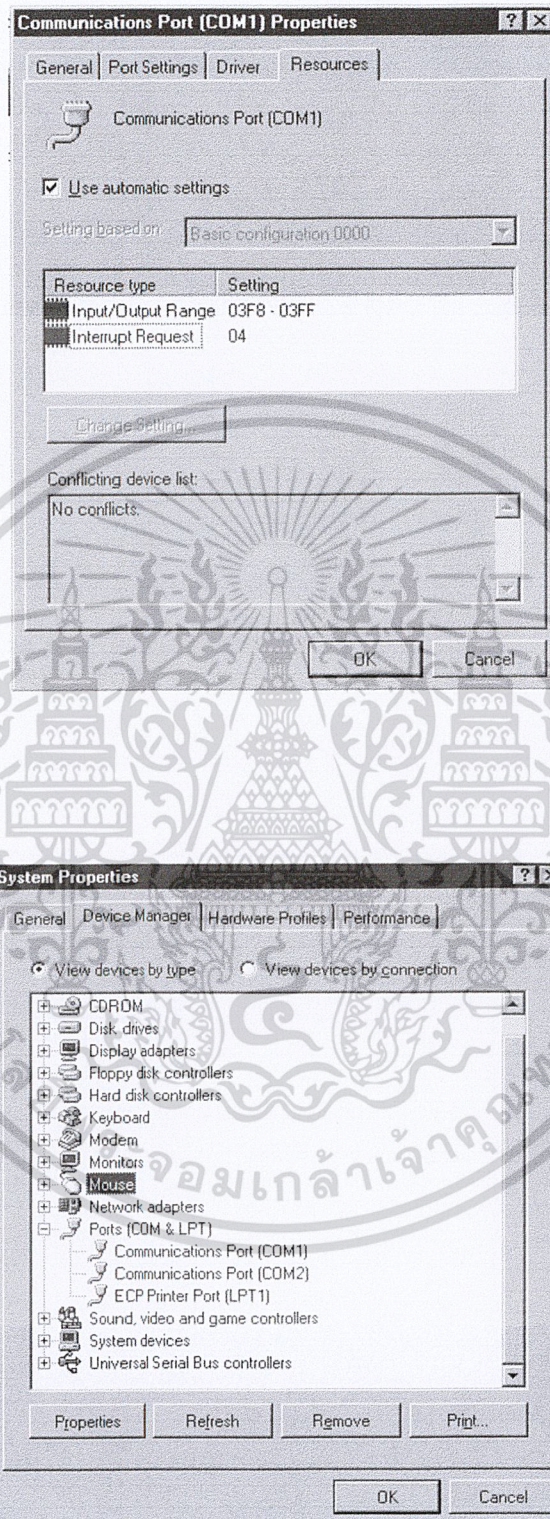
MS-DOS Prompt - DEBUG
Auto
Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1999.
C:\>debug
-d 0000:4000
0000:4000 26 83 3F 00 75 F9 26 C7-07 02 00 C3 1E 52 50 56   &.?.u.&.....RPU
0000:4010 49 02 04 22 00 D2 AC 48-09 74 06 3C 20 72 24 77   I..".H.t.< r$w
0000:4020 F5 AC 3C 09 04 00 02 04-FF 00 FF FF FF FF 75 E7   <.....u.
0000:4030 AC 3C 56 75 05 80 CA 02-E8 DC 3C 49 75 03 80 CA   <Uu.....<Iu..
0000:4040 80 EB D3 5E 8C C8 8E D8-A3 14 00 C7 06 48 00 50   ^.....H.P
0000:4050 0A A3 4A 00 C7 06 F4 01-0F 06 A3 F0 01 A3 EE 01   ..J.....
0000:4060 B8 CC 0A 2D 0C 00 A3 EC-01 F6 C2 80 74 01 CC 06   .....t..
0000:4070 53 BB 40 00 8E C3 BB 17-00 26 F6 07 08 74 03 80   S.@.....&...t..

```

รูปที่ 3.27 หน้าต่างของโปรแกรม DEBUG แสดงตำแหน่งของพอร์ตอนุกรม

ก็ทำการเลื่อนบิตไปทางขวา 1 บิต โดยใช้วิธีการหารด้วย 2 เพื่อให้บิตที่ต้องการไปอยู่ด้านขวามือสุด และแสดงค่าจำนวนพอร์ตที่แท้จริงออกมา คำสั่ง HEX\$ ช่วยให้การไปอยู่ด้านขวามือสุด และแสดงผลตำแหน่งแอดเดรสของพอร์ตอนุกรมที่ออกมาเป็นค่าเลขฐานสิบหก ส่วนการคูณค่าด้วย 100H นั้นก็เพื่อที่จะเลื่อนบิตไปด้านหน้า 1 ไบต์ทำให้เมื่อเวลานำมาบวกแล้วจะได้ค่าเป็น 2 ไบต์พอดี มีโปรแกรมตัวอย่างแสดงดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 หน้าต่างแสดงรายละเอียดของพอร์ตอนุกรมบนวินโดวส์ 95/98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่าเริ่มต้นให้กับพอร์ตอนุกรม

ก่อนการใช้งานพอร์ตอนุกรมนั้นจะต้องมีการกำหนดค่าเริ่มต้นให้กับตัวมันก่อน ซึ่งก็คือการกำหนดจำนวนบิตข้อมูลที่ต้องการส่ง จำนวนบิตท้าย ชนิดของพาริตีที่ใช้ และบอดเรต

การกำหนดสามารถทำได้หลายวิธี วิธีแรกเป็นการกำหนดจากคอสพรีอัมพ์ โดยใช้คำสั่ง MODE ซึ่งมีวิธีการใช้งานดังนี้

MODE COMm : baud = b , parity = p , data = d , stop = s , retry = r หรือ

MODE COMm : b , p , d , s , r

ตัวอย่าง MODE COM1 : 9600 , n , 8 , 1 จะเป็นการกำหนดให้พอร์ตอนุกรม COM1 มีบอดเรตเท่ากับ 9600 บิตต่อวินาที ไม่มีการตรวจสอบพาริตี รับส่งข้อมูลแบบ 8 บิต และมีบิตปิดท้าย 1 บิต

วิธีที่ 2 เป็นการกำหนดโดยใช้อินเทอร์รัปต์ของคอส ตำแหน่งที่ 14H ซึ่งการใช้งานจะต้องกำหนดค่าต่างๆ ลงในรีจิสเตอร์ด้วย โดยจะต้องกำหนดให้รีจิสเตอร์ AH มีค่าเท่ากับ 0 รีจิสเตอร์ DX เก็บค่าพอร์ตอนุกรมที่ต้องการกำหนดค่าเริ่มต้นโดย

DX = 0 จะกำหนดให้พอร์ตอนุกรม COM1

DX = 1 จะกำหนดให้พอร์ตอนุกรม COM2

DX = 2 จะกำหนดให้พอร์ตอนุกรม COM3

DX = 3 จะกำหนดให้พอร์ตอนุกรม COM4

รีจิสเตอร์ AL ซึ่งมีขนาด 8 บิตใช้เก็บค่าเริ่มต้นต่างๆ มีรายละเอียดดังนี้

ตารางที่ 30 แสดงรายละเอียดของรีจิสเตอร์ AL

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
BD2	BD1	BD0	PAR1	PAR0	STOP	DA1	DA0

BD2 , BD1 , BD0 ใช้สำหรับกำหนดค่าบอดเรต

“111” บอดเรตเท่ากับ 9,600 บิตต่อวินาที

“110” บอดเรตเท่ากับ 4,800 บิตต่อวินาที

“101” บอดเรตเท่ากับ 2,400 บิตต่อวินาที

“100” บอดเรตเท่ากับ 1,200 บิตต่อวินาที

“011” บอดเรตเท่ากับ 600 บิตต่อวินาที

“010” บอดเรตเท่ากับ 300 บิตต่อวินาที

“001” บอดเรตเท่ากับ 150 บิตต่อวินาที

“000” บอดเรตเท่ากับ 110 บิตต่อวินาที

PAR1, PAR0 ใช้กำหนดค่าพาริตีโดย

“00” หรือ “10” ไม่มีการตรวจสอบพาริตี

“01” พาริตีคู่

“11” พาริตีคู่

STOP ใช้กำหนดจำนวนของบิตปิดท้าย

“1” มีบิตปิดท้ายเท่ากับ 2 บิต

“0” มีบิตปิดท้ายเท่ากับ 1 บิต

DA1, DA0 ใช้กำหนดความยาวของข้อมูลโดย

“10” ความยาวข้อมูลเท่ากับ 7 บิต

“11” ความยาวข้อมูลเท่ากับ 8 บิต

3.9 การรับส่งข้อมูลแบบอนุกรม

มีหลายหลายวิธีในการรับ และส่งข้อมูลแบบอนุกรมผ่านพอร์ตอนุกรม RS-232 เช่น ใช้คำสั่งพิมพ์พ้ออกทางเครื่องพิมพ์ เรียกอินเตอร์รัปต์ของไบออส หรือของคอสการเขียนหรืออ่านไปยังแอดเดรสของพอร์ตโดยตรง วิธีสุดท้ายเป็นวิธีที่มีความยืดหยุ่นในการใช้งานที่สุด ยกตัวอย่าง ถ้าต้องการส่งข้อมูลไปยังพอร์ตอนุกรม COM1 สามารถเขียนข้อมูลโดยตรงไปที่รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (แอดเดรส 3F8) โดยใช้คำสั่งภาษา QBASIC ง่าย ๆ ดังนี้

```
OUT &H3F8,X
```

ค่า X ในที่นี้หมายถึงข้อมูลที่ต้องการส่ง มีขนาด 8 บิต

สำหรับการอ่านข้อมูลจากพอร์ตอนุกรม จะเป็นการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (แอดเดรส 3F8H เช่นเดียวกัน) ซึ่งสามารถเขียนโปรแกรมง่าย ๆ ได้ดังนี้

```
Y = INP (&H3F8)
```

ค่า Y ในที่นี้คือค่าให้อ่านได้จากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล โดยมีขนาด 8 บิต

สำหรับการเขียนโปรแกรมด้วย TURBO PASCAL ก็สามารถใช้คำสั่ง

```
PORT[$3F8] = X
```

สำหรับการเขียนข้อมูลไปยังพอร์ตอนุกรมและ

```
Y = PORT[$3F8]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปะลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการใช้งาน

Object.CommPort[= value]

โดย Value เป็นค่าของพอร์ตอนุกรมชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error68 ขึ้นมาซึ่งหมายถึงอุปกรณ์ตัวนี้ไม่มีอยู่ในระบบดังนั้นการเขียน โปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนใช้คำสั่ง Openport

Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด พาริตี จำนวนของบิตข้อมูล จำนวนของบิตปิดท้าย รูปแบบการใช้งาน

Object.Setting [= value]

โดย Value มีชนิดข้อมูลเป็นแบบ Sting มีรูปแบบเป็น “ BBBB . P , D , S ” โดย BBBB เป็นค่าอัตราบอด P เป็นค่าพาริตี D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “ 9600 , n , 8 , 1 ”

ค่าบอดเรตมาตรฐานที่ใช้กับ Mscomm มีดังนี้

110	บิต ต่อ วินาที
300	บิต ต่อ วินาที
600	บิต ต่อ วินาที
1,200	บิต ต่อ วินาที
2,400	บิต ต่อ วินาที
9,600	บิต ต่อ วินาที (ค่าปกติ)
14,400	บิต ต่อ วินาที
19,200	บิต ต่อ วินาที
28,800	บิต ต่อ วินาที
38,400	บิต ต่อ วินาที (สงวน)
56,000	บิต ต่อ วินาที (สงวน)
128,000	บิต ต่อ วินาที (สงวน)
256,000	บิต ต่อ วินาที (สงวน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 สำหรับค่ามาตรฐานในการกำหนดค่าพาริตีมีดังนี้
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 31 แสดงมาตรฐานการกำหนดบิตพาริตี RS-232

สัญลักษณ์	รายละเอียด
E	พาริตีคู่ (Even)
M	ลอจิก "1" (MARK)
N	ไม่ใช่ (ค่าปกติ)
O	พาริตีคี่ (Odd)
S	ลอจิก "0" (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4, 5, 6, 7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ) 1.5 และ 2

ตัวอย่างการใช้งานคำสั่ง Settings โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

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

หมายเหตุ ค่าที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด “ ” เนื่องจากค่าที่กำหนดนี้อยู่ในรูปตัวแปร Sting

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิด และปิดพอร์ตอนุกรมรูปแบบใช้งาน

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

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึง การปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากเปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารผ่านพอร์ตอนุกรม COM1 และมีบอดเรต 9,600 บอดต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และบิต 1 บิต มีดังนี้

```
MSComm.Setting = "9600 , n , 8 , 1 "
```

```
MSComm.CommPort = 1
```

```
MSComm.PortOpen = True
```

INPUT

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

Object.Input

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น ComInputMondeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น ComInputMondeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปแบบของไบนารีและชนิดข้อมูลเป็นแบบ Variant

ตัวอย่าง โปรแกรมแสดงให้เห็นถึงวิธีการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Commmand1_Click ()
```

```
Dim InString as String
```

```
MSComm1.InputLen = 0 ; Retrieve all available data.
```

```
If MSComm1.InBufferCount Then ; Check for data
```

```
InString = MSComm.Input ; Read data
```

```
End Sub
```

InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

```
Object.InBufferCount [= value]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง `InBufferCount` จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับกาเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ `InBufferCount` มีค่าเป็น 0

หมายเหตุ อย่าสับสนระหว่างคำสั่ง `InBufferSize` และ `InBufferCount` คำสั่ง `InBufferSize` นั้นเพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ

InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์
รูปแบบการใช้งานคำสั่ง

`Object.InBufferSize [= value]`

คำสั่ง `InBufferSize` ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้ หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่น ๆ จะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โฟลวหรือทอข้อมูลสั้นบัฟเฟอร์ เว้นแต่จะมีการใช้แฮนเช็ก ดังนั้นค่าปานกลางก็คือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิด โอเวอร์โฟลแล้วจึงค่อยปรับเพิ่มขนาดบัฟเฟอร์ให้มีค่ามากขึ้น

InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

`Object.Inputlen [= value]`

ค่าเริ่มต้นของคุณสมบัติ `InputLen` มีค่าเท่ากับ “0” การกำหนดค่าเท่ากับ “0” จะทำให้คำสั่ง `Input` ของ `MSComm` อ่านค่าข้อมูลที่อยู่ภายในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน `InputLen` คำสั่ง `Input` จะส่งค่าว่าง (“_”) กลับออกมาผู้ใช้งานสามารถตรวจสอบข้อมูลในบัฟเฟอร์ภาครับได้โดยคุณสมบัติ `InBufferCount` โดยกำหนดให้ข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับคุณสมบัตินี้มักใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับ โดยคำสั่ง Input

รูปแบบการใช้งานคำสั่ง

Object.InputMode [= value]

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยที่ข้อมูลจะเลือกได้ 2 ประเภท

ComInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

ComInputModeBinary สำหรับข้อมูลอื่น ๆ ซึ่งเก็บอยู่ในรูปไบนารีรวมกันอยู่เป็นไบต์ข้อมูล

Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัพเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

Object.Output [= value]

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัพเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลเป็นแบบ Byte

OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัพเฟอร์ภาคส่ง และสามารถใช้คำสั่งนี้เพื่อเคลียร์บัพเฟอร์ภาคส่งด้วย

รูปแบบการใช้งานภาคส่ง

Object.OutBufferCount [= value]

ผู้ใช้งานสามารถเคลียร์บัพเฟอร์ภาคส่งได้โดยการกำหนดค่า OutBufferCount เท่ากับ "0"

หมายเหตุ ระวังการสับสนระหว่างคุณสมบัติ OutBufferCount กับ OutBufferSize ซึ่ง OutBufferSize ใช้เพื่อกำหนดขนาดของบัพเฟอร์ภาคส่ง

OutbufferSize

กำหนดค่าและคืนค่าขนาดของบัพเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Object.OutBufferSize [= object]

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่งานจะมีค่าเท่ากับ 512 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไปจะทำให้มีหน่วยความจำเหลือให้ใช้งานน้อยลง แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้เกิดข้อมูลสั้นบัฟเฟอร์ ยกเว้นจะมีการใช้แฮนด์เช็ก วิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 512 ไบต์ดูก่อนที่โปรแกรมทำงานแล้วเกิดการสั้นของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น

ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี

รูปแบบการใช้งานคำสั่ง

Object.ParityReplace [= value]

บิตพาริตี เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี คอมโทรล MSCOM จะทำการบวกบิตที่มีค่าลอจิก “1” ในแต่ละไบต์ และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก “1” เป็นเลขคู่หรือเลขคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตีไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

การกำหนดค่า เริ่มต้นให้กับ ParityReplace นั้นกำหนดให้ใช้เครื่องหมาย (?) ไปวางไว้ที่ตำแหน่งที่เกิดพาริตีผิดพลาด ถ้ากำหนดค่า ParityReplace ให้เป็นค่าว่าง (“ ”) จะเป็นการยกเลิกการใช้งาน Parity Replace และ ไม่มีการป้อนข้อมูลอื่นแทนเมื่อตรวจพบข้อผิดพลาด

ParityReplace ใช้ชนิดข้อมูลเป็นแบบสตริง แต่จะการกำหนด จะกำหนดได้เพียงไบต์เดียวเท่านั้นซึ่งจะสามารถใช้ค่าใด ๆ ก็ได้ที่เป็นโค้ด ANSI มีค่าอยู่ระหว่าง 0 - 255

DTREnable

ใช้ในการกำหนดสถานะลอจิกของขา Data Terminal Ready (DTR) โดยสัญญาณของขา DTR จะส่งออกจากคอมพิวเตอร์ไปยังโมเด็ม เพื่อแสดงว่าคอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งาน

Object.DTREnable[= value]

ค่า value เป็นสถาน True หรือ False เพื่อกำหนดลอจิกของขา DTR ให้เป็น “0” หรือ “1” โดย

True หมายถึง ให้ขา DTR มีลอจิก “1”

False หมายถึง ให้ขา DTR มีลอจิก “0” (เป็นค่าปกติ)

หมายเหตุ เมื่อขา DTR ถูกกำหนดสถานะให้เป็น True ที่ขา DTR จะมีสถานะลอจิก “1” เมื่อทำการเปิดพอร์ตและจะมีสถานะเป็น “0” เมื่อมีการเปิดพอร์ต เมื่อขา DTR ถูกกำหนดสถานะเป็น False ที่ขา DTR จะมีสถานะลอจิกเป็น “0” ตลอดเวลาไม่ว่าจะใช้คำสั่งเปิดพอร์ตหรือปิดพอร์ต

สำหรับการใช้งานกับโมเด็ม การทำให้ขา DTR เป็นลอจิก “0” จะเป็นการวางหูโทรศัพท์หรือยกเลิกการติดต่อ

RTSEnable

ใช้เพื่อกำหนดสถานะลอจิกให้ขา Reques To Send (RTS) โดยขา RTS จะเป็นสัญญาณที่ส่งจากคอมพิวเตอร์ไปยังโมเด็มเพื่อร้องขอส่งข้อมูล ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

Object.RTSEnable [= value]

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิก “0” หรือ “1” ให้ขา RTS โดย

True หมายถึง ให้ขา RTS มีลอจิก “1”

False หมายถึง ให้ขา RTS มีลอจิก “0” (เป็นค่าปกติ)

หมายเหตุ เมื่อขา RTSEnable ถูกกำหนดให้เป็น True ขา RTS จะมีสถานะลอจิก “1” เมื่อเปิดพอร์ตและมีสถานะลอจิก “0” เมื่อปิดพอร์ต

EOFEnable

เป็นการกำหนดให้ MSComm รอสัญญาณแสดงส่วนท้ายสุดของไฟล์ (End of file : EOF) ระหว่างการรับอินพุตเข้ามา ถ้าพบสัญญาณ EOF ภาคอินพุตจะหยุดรับข้อมูล และเหตุการณ์ Oncommm จะถูกกระตุ้นให้ทำงาน คุณสมบัติ CommEvent จะมีค่าเท่ากับ 7 หรือ ComEvEOF

รูปแบบการใช้งาน

Object.EOFEnable [= value]

โดย Value เป็นค่าสถานะ True หรือ False เพื่ออีนาเบิลหรือดิสเอเบิลการทำงานของเหตุการณ์

Oncommm เมื่อตรวจพบสัญญาณ EOF โดย

True หมายถึง เหตุการณ์ Oncommm จะถูกกระตุ้นให้ทำงานด้วย EOF

False หมายถึง เหตุการณ์ Oncommm จะไม่ถูกกระตุ้นการทำงานด้วย EOF (เป็นค่าปกติ) เมื่อ

EOFEnable กำหนดให้เป็น False ส่วนควบคุมจะไม่มีการตรวจสอบสัญญาณ EOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CTSHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Clear To Send (CTS) ได้ว่ามีสถานะลอจิก “0” หรือ “1” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CTSHolding เป็น True ขา CTS จะมีสถานะลอจิกเป็น “1” ถ้าค่า CTSHolding เป็น False ขา CTS จะมีสถานะลอจิกเป็น “0”

รูปแบบการใช้งาน

Object.CTSHolding

เมื่อขา CTS เป็นลอจิก “0” (CTSHolding = False) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น ComEventCTSTO (Clear To Send Timeout) และกระตุ้นเหตุการณ์ OnComm

CD Holding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Data Carrier Detect (DCD) ได้ว่ามีสถานะลอจิกเป็น “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CDHolding เป็น True ขา DCD จะมีสถานะลอจิก “1” ถ้าค่า CDHolding เป็น False ขา DCD จะมีสถานะลอจิก “0”

รูปแบบการใช้งาน

Object.CDHolding

เมื่อขา DCD มีลอจิก “1” (CDHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEventCDTO (Carrie Detect Timeout Error) และกระตุ้นให้เกิดเหตุการณ์ Oncomm

DSRHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา DSR (DSR) ได้ว่ามีสถานะลอจิก “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า DSRHolding เป็น True ขา DSR จะมีสถานะลอจิก “1” ถ้าค่า DSRHolding เป็น False ขา DSR จะมีสถานะลอจิก “0”

รูปแบบการใช้งาน

Object.DSRHolding

เมื่อขา DSR เป็นลอจิก “1” (DSRHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น ComEventDSRTO (Data Set Ready Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Handshaking

กำหนดคุณสมบัติและค่านำรูปแบบแฮนด์เช็กทางฮาร์ดแวร์

รูปแบบการใช้งาน

Object. Handshaking [= value]

ค่าตัวแปร Value ที่ใช้กำหนดค่ากำหนดได้ 4 รูปแบบด้วยกันคือ

1. ComNone ค่าที่กำหนดคือ 0 เป็นการกำหนดให้ไม่มีการแฮนด์เช็ก (เป็นค่าเริ่มต้น)
2. ComXOnXoff ค่าที่กำหนดคือ 1 เป็นการกำหนดให้ใช้แฮนด์เช็กแบบ XON/XOFF
3. ComRTS ค่าที่กำหนดคือ 2 เป็นการกำหนดให้ใช้ขา RTS/CTS (Request To Send/Clear To Send)
4. ComRTSXonXoff ค่าที่กำหนดคือ 3 เป็นการกำหนดให้ใช้ทั้งแบบ Request To Send และ XON/XOFF

คุณสมบัติ Handshaking ใช้เพื่อกำหนดรูปแบบการสื่อสารภายในระหว่างที่ข้อมูลถูกส่งไปยังบัพเฟอร์ภาครับ เมื่อข้อมูลตัวอักษรถูกส่งมาถึงพอร์ตอนุกรม อุปกรณ์สื่อสารข้อมูลจะทำการย้ายข้อมูลไปยังบัพเฟอร์ภาครับ เพื่อที่จะให้โปรแกรมสามารถอ่านค่าไปใช้งานได้ ถ้าไม่มีบัพเฟอร์ภาครับ โปรแกรมที่ใช้งานจะต้องทำการอ่านค่าข้อมูลโดยตรงจากฮาร์ดแวร์ของพอร์ตอนุกรม ซึ่งผู้ใช้งานจะเกิดปัญหาข้อมูลสูญหายได้ เนื่องจากว่าการเปลี่ยนแปลงของข้อมูลที่ส่งเข้ามามีการเปลี่ยนแปลงอย่างรวดเร็ว

คุณสมบัติ handshaking ช่วยให้ผู้ใช้งานแน่ใจได้ว่าข้อมูลที่รับเข้ามานั้นไม่มีการสูญหายเมื่อบัพเฟอร์ภาครับที่รับข้อมูลนั้นเกิดข้อมูลล้นหรือโอเวอร์โฟลว (overflow) โดยวิธีการตรวจสอบความพร้อมของบัพเฟอร์ว่าพร้อมรับข้อมูล หรือไม่ก่อนที่จะส่งข้อมูลมาให้

Break

ใช้ในการเซตและเคลียร์ค่าสัญญาณ Break ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

โดย Value เป็นค่าบูลีน ถ้า Value = True หมายถึง การส่งสัญญาณ Break ออกไป ถ้า Value = False หมายถึงการเคลียร์ค่าสัญญาณ Break

เมื่อกำหนดให้สัญญาณ Break เป็น True จะเป็นการหยุดการส่งข้อมูลชั่วคราวจนกว่าจะมีการส่งสัญญาณ Break เป็น False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์ Oncomm

เหตุการณ์ Oncomm จะถูกสร้างขึ้นเมื่อค่าของคุณสมบัติ CommEvent มีการเปลี่ยนแปลง เพื่อแสดงผลการเปลี่ยนแปลงเหล่านั้นแบบทันทีทันใดหรือแสดงข้อผิดพลาดที่เกิดขึ้น

ค่าคงที่คุณสมบัติของคอนโทรล MComM ค่าคงที่สำหรับคุณสมบัติ Handshake

ตารางที่ 32 แสดงค่าคงที่คุณสมบัติ Handshake

ค่าคงที่	ค่า	รายละเอียด
ComNone	0	ไม่ใช้การตรวจสอบแฮนด์เชก
ComXonXoff	1	ใช้การตรวจสอบแฮนด์เชกแบบ Xon/Xoff
ComRTS	2	ใช้ในการตรวจสอบแฮนด์เชกผ่านทางขา RTS และ CTS
ComRTSXOnXOff	3	กำหนดการตรวจสอบแบบแฮนด์เชกทั้งแบบ RTS,CTS และ Xon/Xoff

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่สำหรับคุณสมบัติ OnComm

ตารางที่ 33 แสดงค่าคงที่สำหรับคุณสมบัติ OnComm

ค่าคงที่	ค่า	รายละเอียด
ComEvSend	1	ส่งค่าเหตุการณ์ (Send event)
ComEvReceive	2	รับค่าเหตุการณ์ (receive event)
ComEvCTS	3	มีการเปลี่ยนแปลงที่ขา CTS
ComEvDSR	4	มีการเปลี่ยนแปลงที่ขา DSR
ComEvCD	5	มีการเปลี่ยนแปลงที่ขา DCD
ComEvRing	6	ตรวจจับสัญญาณกระดิ่งของโทรศัพท์
ComEvEOF	7	ตรวจพบตำแหน่งท้ายสุดของไฟล์ (End of file)

ค่าคงที่สำหรับคุณสมบัติ Error

ตารางที่ 34 ค่าคงที่สำหรับคุณสมบัติ Error

ค่าคงที่	ค่า	รายละเอียด
ComEventBreak	1001	ได้รับสัญญาณ Break
ComEventCTSTO	1002	ขา CTS เกิดไทม์เอาต์
ComEventDSRTO	1003	ขา DSR เกิดไทม์เอาต์
ComEventFrame	1004	เกิดข้อผิดพลาดที่เฟรมข้อมูล (Framing error)
ComEventOverrun	1006	พอร์ตอนุกรมเกิดโอเวอร์รัน (Port Overrun)
ComEventCDTO	1007	ขา DCD เกิดไทม์เอาต์
ComEventRxOver	1008	บัฟเฟอร์รับข้อมูลเกิดโอเวอร์โฟลว
ComEventRxParity	1009	เกิดข้อผิดพลาดที่พาริตี (Parity Error)
ComEventTxFull	1010	บัฟเฟอร์ส่งข้อมูลเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่สำหรับคุณสมบัติ InputMode

ตารางที่ 35 แสดงค่าคงที่สำหรับคุณสมบัติ InputMode

ค่าคงที่	ค่า	รายละเอียด
ComInputModeText	0	ข้อมูลที่รับมีคุณสมบัติเป็นข้อความ (ค่าปกติ)
ComInputModeBinary	1	ข้อมูลที่รับเข้ามาเป็นไบนารี

การใช้ MSComm เพื่อการติดต่อฮาร์ดแวร์

จากรายละเอียดของ MSComm ที่กล่าวไปในตอนต้นนั้น จะเห็นได้ว่าวิธีการที่จะอ่านค่าหรือเขียนค่าไปยังสถานะและควบคุมของพอร์ตอนุกรมสามารถทำได้ง่ายคายนมาก โดยใช้คำสั่งเหล่านี้

DTREnable	สำหรับสั่งให้ขา DTR มีลอจิก “0” หรือ “1”
RTSEnable	สำหรับสั่งให้ขา RTS มีลอจิก “0” หรือ “1”
CTSHolding	สำหรับอ่านค่าสถานะจากขา CTS ว่ามีลอจิก “0” หรือ “1”
CDHolding	สำหรับอ่านค่าสถานะจากขา DCD ว่ามีลอจิก “0” หรือ “1”
DSRHolding	สำหรับอ่านค่าสถานะจากขา DSR ว่ามีลอจิก “0” หรือ “1”
Break	สำหรับการสั่งให้ขา Txd มีลอจิก “0” หรือ “1”

การส่งและการอ่านข้อมูลผ่านพอร์ตอนุกรม RS-232

พอร์ตอนุกรมนอกจากจะมีขาสำหรับรับและส่งข้อมูลปกติแล้วยังมีขาที่ออกแบบไว้สำหรับควบคุมการไหลของข้อมูลอีกหลายตำแหน่งด้วยกัน โดยแยกเป็น 2 ประเภท คือ ขาที่ทำหน้าที่เป็นเอาต์พุต และขาที่ทำหน้าที่เป็นอินพุต

ขาสัญญาณเอาต์พุตของพอร์ตอนุกรม

ขาที่ทำหน้าที่เป็นขาเอาต์พุต ได้แก่ขา DTR , RTS และ TxD โดยรีจิสเตอร์ที่ทำหน้าที่ควบคุมขาเหล่านี้คือรีจิสเตอร์ควบคุมโมเด็ม (MCR) โดยมีแอดเดรสอยู่ถัดจากรีจิสเตอร์หลักของพอร์ตอนุกรม 4 ตำแหน่ง รีจิสเตอร์ควบคุมการทำงานของขา DTR จะอยู่ที่ตำแหน่งบิต 0 ส่วนขา RTS จะอยู่ที่ตำแหน่งบิต 1 ส่วนขา TxD เป็นขานอกปกติในการส่งข้อมูล ดังนั้นจึงมีแอดเดรสอยู่ที่แอดเดรสของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้ไปใช้ประโยชน์ด้านการค้า รีจิสเตอร์หลัก

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS-232

มาตรฐานการสื่อสารข้อมูลของพอร์ตอนุกรม ได้ระบุช่วงระดับแรงดันสำหรับการทำงานของพอร์ตอนุกรมไว้ว่า ที่ลอจิก “0” จะมีระดับสัญญาณ +3 ถึง +15V ส่วนลอจิก “1” จะมีระดับสัญญาณ -3 ถึง -15 V ระดับสัญญาณนี้ทำให้ไม่สามารถที่จะนำเอาต์พุตใด ๆ ต่อเข้ากับลอจิกเกตเพื่อใช้งานได้ โดยตรงจะต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อน โดยปกติจะใช้ไอซีจําพวก RS-232 trasceiver ที่นิยมมากคือ MAX 232 หรือ ICL 232 ไอซีเหล่านี้ในกลุ่มนี้จะทำหน้าที่แปลงระดับแรงดันเป็น 0V ส่วนลอจิก “1” ซึ่งมีระดับสัญญาณ -3 ถึง -15V จะแปลงเป็น +5V ทั้งนี้เพื่อให้สามารถเชื่อมต่อกับอุปกรณ์ดิจิทัลอื่นที่ใช้ระดับแรงดันที่ทีแอลได้

การเขียนซอฟต์แวร์เพื่อควบคุมขาเอาต์พุต

การติดต่อกับพอร์ตอนุกรมบนระบบปฏิบัติการวินโดวส์จะต้องเพิ่มอุปกรณ์ทางซอฟต์แวร์สำหรับการติดต่อกับพอร์ตอนุกรม นั่นคือ Mscomm

ผู้ใช้งานสามารถเขียนโปรแกรมด้วย Visual Basic เพื่อส่งค่าออกไปยังขาเอาต์พุตต่าง ๆ ของพอร์ตอนุกรมได้ โดยใช้คำสั่งดังนี้

MSComm1.DTREnable = True สำหรับการกำหนดให้ขา DTR มีลอจิก “1”

MSComm1.DTREnable = False สำหรับการกำหนดให้ขา DTR มีลอจิก “0”

MSComm1.RTSEnable = True สำหรับการกำหนดให้ขา RTS มีลอจิก “1”

MSComm1.RTSEnable = False สำหรับการกำหนดให้ขา RTS มีลอจิก “0”

MSComm1.Break = True สำหรับการกำหนดให้ขา TxD มีลอจิก “1”

MSComm1.Break = False สำหรับการกำหนดให้ขา TxD มีลอจิก “0”

หมายเหตุ ก่อนที่จะใช้งานคำสั่งของคอนโทรล MSComm จะต้องทำการเปิดพอร์ตก่อน โดยเขียนโปรแกรมดังนี้

```
Private Sub Form_Load()
```

```
MSComm.PortOpen = True
```

```
End Sub
```

พร้อมกันนั้นจะต้องตรวจสอบพอร์ตที่ใช้งานให้คิดว่าพอร์ตอนุกรมที่ใช้นั้นถูกต้องหรือไม่ มิฉะนั้นโปรแกรมจะแสดงข้อผิดพลาดขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านค่าลอจิกจากพอร์ตอนุกรม RS-232

พอร์ตอนุกรมมีขาที่ทำหน้าที่อินพุตได้แก่ DSR , CTS , RI และ DCD โดยรีจิสเตอร์ที่ทำหน้าที่ควบคุมขาเหล่านี้คือรีจิสเตอร์แสดงสถานะโมเด็ม (MSR) มีแอดเดรสจากรีจิสเตอร์หลักของพอร์ตอนุกรม 6 ตำแหน่ง สำหรับบิตต่าง ๆ บนรีจิสเตอร์มีรายละเอียดดังนี้

ตารางที่ 36 แสดงการอ่านค่าลอจิกจากพอร์ตอนุกรม RS-232

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

4 บิตที่กล่าวบนรีจิสเตอร์จะแสดงสถานะการเปลี่ยนแปลงที่ขาอินพุตทั้ง 4 ขาของพอร์ตอนุกรมโดยตรง ส่วน 4 บิตล่าง จะมีสถานะเป็น “1” ก็ต่อเมื่อ 4 บิตบนมีการเปลี่ยนแปลงสถานะเมื่อเทียบกับการอ่านค่าครั้งก่อนหน้า

สำหรับการอ่านค่าจากขา DCD,CTS และ DSR โดยโปรแกรม Visual Basic จะใช้ MComm ร่วมกับคำสั่ง CDHolding , CTSHolding และ DSRHolding ตามลำดับ ซึ่งค่าที่อ่านได้นั้นจะเป็นบูลีน มีค่าเป็น True หรือ False ผู้ใช้งานสามารถตรวจสอบผลจากขาอินพุตเหล่านี้ได้โดยใช้คำสั่ง IF THEN

เนื่องจากสัญญาณของ RS-232 ต้องมีระดับแรงดัน +3 , -3 ถึง +12 , -12 แต่สัญญาณอินพุตที่เกิดขึ้นเป็นระดับทีทีแอล ดังนั้นเมื่อเกิดสัญญาณอินพุตขึ้น จะต้องส่งผ่านวงจรขับ เพื่อปรับระดับแรงดันให้เหมาะสมเสียก่อน

3.10 อัตราค่าบริการโทรศัพท์ทางไกลต่างจังหวัด

สำหรับอัตรานี้ถูกกำหนดไว้เป็นมาตรฐานโดยองค์การโทรศัพท์แห่งประเทศไทย โดยแสดงอยู่ในอัตราของราคาต่อ 1 นาที โดยการคิดเศษของนาทีปัดขึ้นเป็น 1 นาที ตารางที่ 37 แสดงอัตราค่าโทรศัพท์ทางไกลต่างจังหวัดเมื่อคิดจุดโทรศัพท์ที่ออกจากพื้นที่กรุงเทพฯ โดยแยกเป็นเขตภูมิภาคเขตรหัสทางไกล แล้วแยกย่อยออกเป็นแต่ละจังหวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสทางไกล 3 ตัวแรก	เลขนำหน้าตัว แรก	จังหวัด	อัตราค่าบริการ (บาท/นาที)
032	2,3	ราชบุรี	6
	4	เพชรบุรี	6
	5,6	ประจวบคีรีขันธ์	12
034	2,3	นครปฐม	3
	4	สมุทรสาคร	3
	5,6	กาญจนบุรี	9
035	2,3	อยุธยา	6
	5	สุพรรณบุรี	6
	6	อ่างทอง	6
036	2,3	สระบุรี	6
	4	ลพบุรี	9
	5	สิงห์บุรี	9
037	2	ปราจีนบุรี, สระแก้ว	6
	3	นครนายก	6
038	2,3,4	ชลบุรี	6
	5	ฉะเชิงเทรา	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างชื่อของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

039	3,4	จันทบุรี	12
	5	ตราด	12
042	2,3	อุดรธานี,หนอง บัวลำภู	15
	4	หนองคาย	18
	5	นครพนม	18
	7	เพชรบูรณ์	12
073	2	ยะลา	18
	3,4	ปัตตานี	18
	5,6	นราธิวาส	18
074	2,3,4,5	สงขลา	18
	6	พัทลุง	18
	7	สตูล	18
075	2	ตรัง	18
	3,4,5	นครศรีธรรมราช	18
	6	กระบี่	18
076	2,3	ภูเก็ต	18
	4	พังงา	18
		สุราษฎร์ธานี	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานทางการศึกษาเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	5	ชุมพร	15
	8	ระนอง	15
02	-	กรุงเทพ	3
01	-	มือถือ	3

หมายเหตุ : ค่าบริการที่แสดงในตารางข้างบนนี้คิดเมื่อต้นทางอยู่ในเขตกรุงเทพฯ

: อัตราค่าบริการที่แสดงในตารางเป็นอัตราภาคกลางวัน

3.11 อัตราค่าบริการภายในท้องถิ่นหรือภายในจังหวัด

การคิดอัตราค่าบริการในลักษณะนี้จะเป็นแบบราคาเดียว คืออัตราค่าบริการไม่ได้ขึ้นกับเวลาในการใช้งาน และสำหรับประเทศไทยนั้นได้กำหนดให้อัตรานี้ คือ 3 บาท ต่อการใช้โทรศัพท์ภายในจังหวัด 1 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

โปรแกรมใช้งาน

4.1 ส่วนแสดงผลทาง Visual Basic

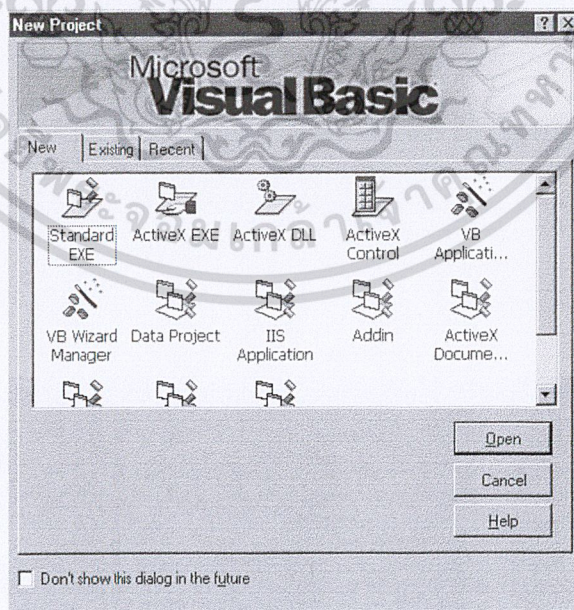
Visual Basic เบื้องต้น

เราพบว่าในปัจจุบันนี้ Windows เป็นระบบปฏิบัติการที่นิยมมาก และการใช้ Windows ก็แพร่หลายไปทั่วโลก ไม่ว่าใครก็สามารถใช้ Windows อย่างง่ายดายเพราะ Windows สามารถใช้งาน และเรียนรู้ได้ง่าย และจุดเด่นที่สำคัญอีกอย่างนั้นก็คือ มันสามารถที่จะทำงานได้หลายอย่างพร้อม ๆ กันได้ดังนั้น Software ที่เป็นที่นิยมปัจจุบันนี้จึงเป็น โปรแกรมที่สามารถทำงานบน Windows ได้

Visual Basic เป็นโปรแกรมภาษาของคอมพิวเตอร์ที่พัฒนาให้สามารถใช้งานได้ง่าย และสามารถพัฒนาโปรแกรมบน Windows แต่ก่อนจะพบว่า การเขียนโปรแกรมบน Windows การที่จะเขียนหน้าต่างของ Windows ขึ้นมานั้นจะต้องใช้คำสั่งต่าง ๆ มากมาย แต่ถ้าเป็น Visual Basic แล้วสามารถสร้างได้ง่ายดาย ต่อไปจะกล่าวถึงการใช้งาน Visual Basic

การใช้งาน Visual Basic เบื้องต้น

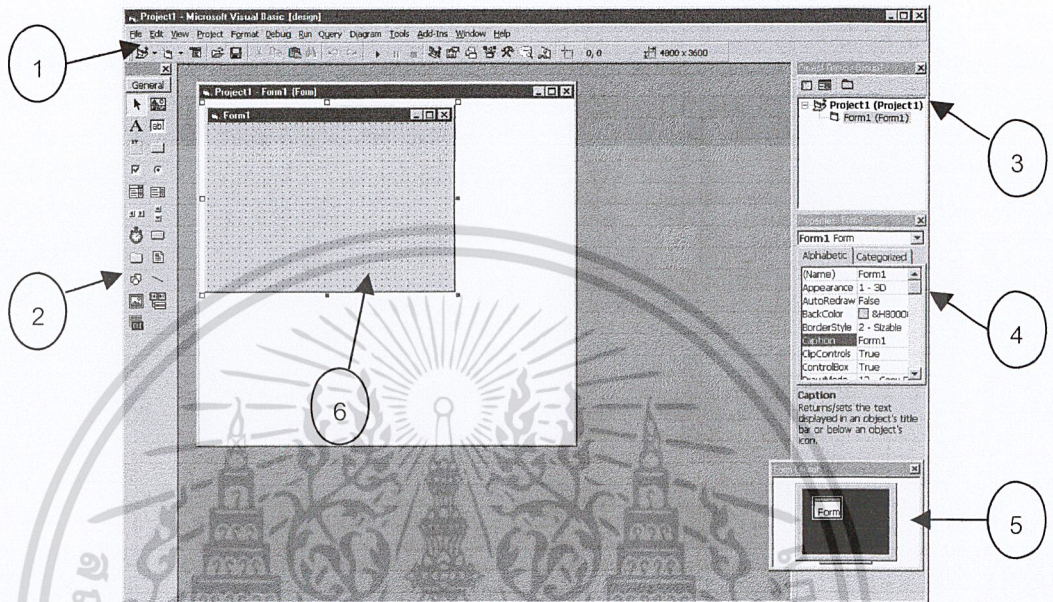
สิ่งแรกที่พบเมื่อเข้าสู่โปรแกรมก็คือหน้าจอสำหรับเปิด project ดังรูปที่ 4.1



รูปที่ 4.1 หน้าจอสำหรับเปิดโปรเจกต์ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในนี้จะเลือก “Standard EXE” เพื่อเข้าสู่การใช้ Visual Basic ที่ใช้ในการเขียนโปรแกรมต่อไป
 ดังรูปที่ 4.2



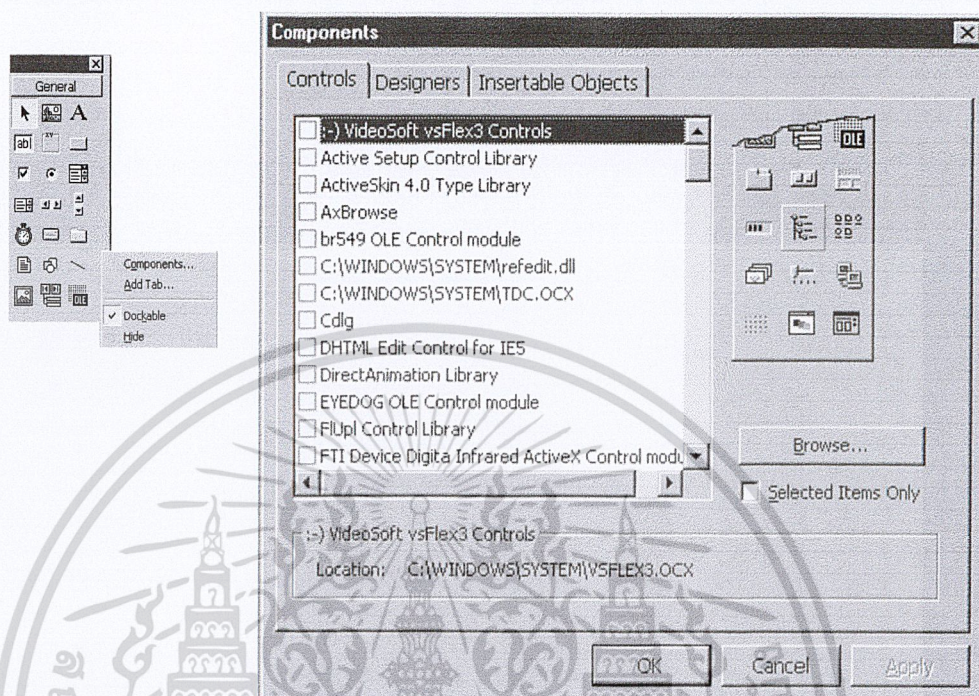
รูปที่ 4.2 รูปแสดงจอภาพของ Visual Basic

ส่วนประกอบต่างๆ ของ Visual Basic

1. Toolbar จะประกอบด้วย icon ที่ใช้ในการพัฒนาโปรแกรมเหมือนกับโปรแกรมอื่น ๆ
2. Toolbox จะประกอบด้วย icon ต่าง ๆ ซึ่งใน Visual Basic เราเรียกว่า control
3. Project Explorer Windows จะแสดงรายชื่อ Form ที่พร้อมจะถูกแก้ไข
4. Properties Windows ใช้สำหรับกำหนดคุณสมบัติ (Property) ให้กับ Form และ Object ต่าง ๆ ที่อยู่บน Form
5. Form Layout Windows จะแสดงตำแหน่งของ Form บนจอคอมพิวเตอร์เมื่อทำการรัน
6. Form เป็นส่วนที่ใช้ในการสร้างจอเมื่อทำการรันแล้วจะทำให้มีหน้าต่างอย่างไรซึ่งเราสามารถที่จะตกแต่งให้สวยงามได้ และเป็นส่วนที่เราสามารถเลือกเอา Control ต่าง ๆ มาวางไว้ และสามารถจัดตำแหน่งได้

ส่วนของ Control ต่าง ๆ ที่สามารถดึงมาใช้ได้นอกจากนี้ยังมีอีกส่วน โดยการคลิกขวาที่บริเวณสีเทาแล้วเลือก Component แล้วทำการเลือก Control ที่ต้องการ ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงการเลือก Control ต่าง ๆ ที่ต้องการใช้งาน

ส่วนของ Toolbar

เป็นแถบเครื่องมือที่ประกอบด้วย Icon ต่าง ๆ ดังรูปที่ 4.4 จะมีลักษณะที่คล้ายกับ Toolbar ของ MS-Word และมีหน้าที่ที่คล้าย ๆ กัน



รูปที่ 4.4 แถบเครื่องมือที่ประกอบด้วย Icon ต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 38 แสดงรายละเอียดการใช้งาน Toolbar ต่าง ๆ

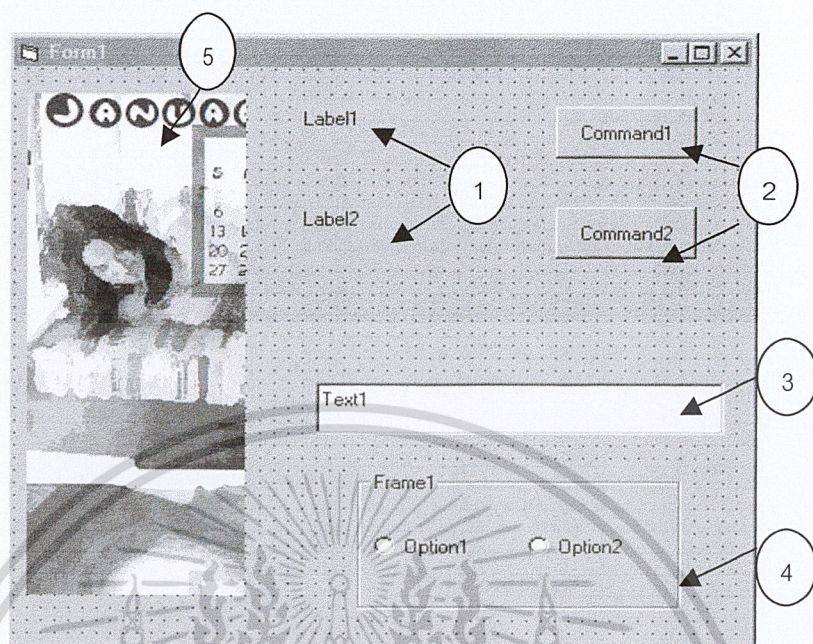
รูป Icon	ชื่อเรียก	รายละเอียดการใช้งาน
	Add Standard EXE Project	ใช้สำหรับเปิด Project ใหม่ ในกรณีที่มีหลาย Project ได้ โดยเข้าเลือกในเมนู Windows
	Add From	ใช้ในการเพิ่ม From ให้กับ Project ซึ่งเรียกจากเมนู Project และ Add From ตามลำดับ
	Manu Editor	ใช้เรียกเมนู Editor ซึ่งเป็น Tool สำหรับสร้างเมนูให้กับ From ซึ่งอาจใช้การกดปุ่ม Ctrl+E หรือเลือกจากเมนู Tool และ Editer ตาม
	Open Project	ใช้สำหรับเปิด Project ซึ่งอาจใช้ Hot Key Ctrl+O หรือเรียกจากเมนู File และ Open Project
	Save Project	ใช้สำหรับบันทึก Project ซึ่งอาจใช้ Hot Key Ctrl+S หรือเรียกจากเมนู File และ Open Project ตามลำดับ
	Cut	ใช้สำหรับตัด Object ต่างๆ บน From ซึ่งอาจใช้ Hot Key Ctrl+x หรือเลือกจากเมนู Edit และ Cut
	Coppy	ใช้สำหรับ Coppy Object บน From ซึ่งอาจใช้ Hot Keys Ctrl+C หรือเลือกจากเมนู Edit และ Coppy ตามลำดับสำหรับ Object ที่ถูก Copy จะเรียกว่า Control Array ซึ่งกล่าวถึงในส่วนต่อไป
	Paste	ใช้สำหรับ Paste Object ที่ Cut หรือ Copy ไว้ ซึ่งอาจใช้ Hot Keys Ctrl+V หรือเลือกจาก Edit และ Copy ตามลำดับ สำหรับ Object ที่ถูก Copy จะเรียกว่า Control Array ซึ่งกล่าวถึงในบทต่อไป
	Find	ใช้สำหรับค้นหาคำใน Editor ซึ่งเป็น Tool ที่ใช้ในการเขียนโปรแกรม โดยอาจใช้ Hot Key Ctrl+F หรือเลือกจากเมนู Edit และ Find ตามลำดับ
	Undo Typing	ใช้สำหรับยกเลิกคำที่พิมพ์ใน Editor ซึ่งอาจใช้ Hot Keys Ctrl+Z หรือเลือกจากเมนู Edit และ Undo Typing ตามลำดับในกรณีที่ไม่มีอยู่ในจอภาพ Editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ใด ๆ ในทางค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงชื่อของเจ้าของลิขสิทธิ์ที่ปรากฏในเอกสาร

		ปรากฏข้อความ Can't Redo แทน
	Start	ใช้สำหรับ Run Project ที่จัดทำขึ้น ซึ่งอาจใช้ Hot Keys F5 หรือเลือกจากเมนู Run และ Start
	Break	ใช้สำหรับหยุดการทำงานของ Project ซึ่งอาจเลือกจากเมนู Run และ Break ตามลำดับ
	End	ใช้สำหรับหยุดการทำงานของ Project ซึ่งอาจเลือกจากเมนู Run และ End ตามลำดับ
	Project Explorer	ใช้แสดงว่า project นั้นประกอบด้วย Form และ Module ใดบ้าง ซึ่งอาจใช้ Hot Keys Ctrl-R หรือเลือกจากเมนู View และ Project Explorer ตามลำดับ
	Properties Windows	ใช้สำหรับกำหนดคุณสมบัติ (Property) ของ Object และ Form ซึ่งอาจใช้ Hot Keys F4 หรือเลือกจากเมนู View และ Properties Windows ตามลำดับ
	Form layout Windows	ใช้สำหรับเรียก Form Layout ซึ่งใช้แสดงตำแหน่งของ Form บนจอภาพโดยอาจเลือกจากเมนู View และ Form Layout Windows ตามลำดับ
	Object Browser	ใช้สำหรับเรียกจอภาพ Object ซึ่งใช้แสดงถึง Class และสมาชิกของแต่ละ Class อาจใช้ Hot Keys F2 หรือเลือกจากเมนู View และ Object Browser
	Toolbox	ใช้สำหรับเรียก Toolbox ขึ้นมาบนจอภาพ ซึ่งอาจเลือกจากเมนู View และ Toolbar ตามลำดับ
	ตำแหน่งของ Form	ใช้บอกตำแหน่งในแกน X และ Y ของ Form
	ขนาดของ Form	ใช้บอกขนาดของ Form ตามแนวแกน X และ Y

Visual Basic เป็นโปรแกรมที่ง่ายต่อการเรียนรู้ และมีขั้นตอนที่ง่ายด้วยอาทิ เช่น เมื่อเราต้องการที่จะให้หน้าจอของเราเป็นอย่างไรก็เพียงแต่ดึงเอา Control นั้น ๆ มาวางไว้บน Form จึงทำให้เราไม่ต้องเขียน คำสั่งต่าง ๆ มากมายจึงทำให้ผู้ใช้สะดวกมาก บนหน้าจอของ Form ก็คือส่วนที่เราจะแสดงเมื่อมีการ Run เราสามารถที่จะตกแต่งให้ดูสวยงามได้ซึ่งจะยกตัวอย่างการเขียน Form และการดึง Control มาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษา ในเพื่อการศึกษาเท่านั้น มิใช่เพื่อเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นหากมีเหตุที่แสดงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงตัวอย่างการเขียน Form และการดึง Control มาใช้งาน

หมายเลข 1 เป็นการใส่ Control ชื่อ Label เมื่อเราดึงมาวางบน Form เราก็จะสามารถพิมพ์ข้อความใส่รูปแบบตัวอักษรหรือใส่สีให้ข้อความได้

หมายเลข 2 เป็นการใส่ Control ชื่อ CommandButton เป็นลักษณะของปุ่มให้เราคลิกซึ่งจะพบว่าเราจะคุ้นเคยกับ Control ตัวนี้มากในการใช้งาน Windows ซึ่งเราสามารถที่จะเขียนโปรแกรมให้กับตัวมัน โดยมองลักษณะที่ว่าเมื่อเกิดการคลิกแล้วให้ทำงานใด ๆ ก็ได้

หมายเลข 3 เป็นการใส่ Control ชื่อ TextBox เป็นลักษณะกล่องข้อความใช้แสดงข้อความ

หมายเลข 4 เป็นการใส่ Control Frame และ Control OptionButton ร่วมกันลักษณะเช่นนี้ก็จะพบบ่อยคือเราอาจจะเขียนโปรแกรมให้ผู้ใช้ผู้ใช้เลือกอย่างใดอย่างหนึ่งจึงเอามารวมเป็น Frame เดียวกัน

หมายเลข 5 เป็นการใส่ Control PictureBox ก็เป็นการใส่รูปภาพเพื่อเป็นการจัดให้โปรแกรมของเรามี Presentation

Form ตัวอย่างข้างบนทำให้มองเห็นภาพการใช้งานของ Visual Basic ได้ชัดเจนมากขึ้นจะพบว่าเราสามารถที่จะจัดรูปแบบตามที่เรต้องการดังนี้ Visual Basic จะเป็น Presentation ที่ใช้ได้ทีเดียวแถมยังสามารถที่จะเขียนคำนวณ และพัฒนาโปรแกรมที่ซับซ้อน อาทิเช่น พวกโปรแกรม

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวชิรราชบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ติดต่อกับอุปกรณ์ Input/output งานทางด้านฐานข้อมูล Application ทางด้าน Internet และยังมีด้านต่าง ๆ มากมายดังนั้นก็จึงพบว่า Visual Basic เป็นโปรแกรมภาษาที่น่าสนใจ ที่สามารถรวมเอา Presentation ที่สวยงามแสนง่าย และรวดเร็วกับความสามารถที่สูงในการคำนวณ และพัฒนาโปรแกรม

ที่ยกตัวอย่างเป็นการใช้งานของ Control บาง Control เท่านั้นเองยังมี Control ที่ไม่ได้กล่าวถึงอีกมากมาย และแต่ละ Control ก็มีเหตุการณ์ที่เกิดขึ้นกับตัวมัน ได้หลายเหตุการณ์เช่นเดียวกับที่กล่าวไว้เป็นเพียงการแนะนำเท่านั้น

4.2 โปรแกรมในการคอนโทรล Visual Basic (From 1)

From 1 ใช้ในการแสดงผลการคิดจำนวนเงินทั้งหมดที่โทรออก

หมายเลขโทรศัพท์	จังหวัด	เวลาที่โทรออก	วันที่โทรออก	จำนวนนาทีที่ใช้	จำนวนเงิน
▶ 027374477	กรุงเทพ	00.49	27/10/2003	1	3
067526959	มอญลือ	00.50	27/10/2003	1	3
029972222	กรุงเทพ	00.51	27/10/2003	29	3

รวมเป็นเงิน 9 บาท

รูปที่ 4.6 แสดงรูปแบบ From 1

From1

Option Explicit

Dim Conn As New ADODB.Connection

Dim Rs1 As New ADODB.Recordset

Dim Rs2 As New ADODB.Recordset

Dim temp As Variant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Cmd1_Click()
Dim temp_str, st, str1, str2, str3, str4, num_reg As Variant
Dim time_b, time_em, time_u, time_e, time_eh, rg_code, rg_sub_code, check As String

temp_str = temp

If Rs1.State = adStateOpen Then
    Rs1.Close
End If

Rs1.CursorLocation = adUseClient

Rs1.Open "delete from record", Conn, adOpenKeyset, , adCmdText

While Len(temp_str) >= 33
Minute (CDate("14:50") - CDate("12:30"))
st = Mid(temp_str, 1, 33)
str1 = Mid(st, 4, 9)
str2 = Mid(st, 14, 5)
str3 = Mid(st, 20, 5)
str4 = Mid(st, 26, 8)
rg_sub_code = Mid(str1, 4, 1)
temp_str = Mid(temp_str, 34, Len(temp_str))
*****

Debug.Print "st = " & st
Debug.Print "str1 = " & str1
Debug.Print "str2 = " & str2
Debug.Print "str3 = " & str3
Debug.Print "str4 = " & str4

Debug.Print "rg_sub_code = " & rg_sub_code

Debug.Print "temp_str = " & temp_str

*****

If Rs1.State = adStateOpen Then
    Rs1.Close
End If

Rs1.CursorLocation = adUseClient

Rs1.Open "record", Conn, adOpenKeyset, adLockOptimistic, adCmdTable
With Rs1
AddNew

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

num_reg = Mid(str1, 1, 2)
Debug.Print "num_reg = " & num_reg
'If num_reg = "01" Or num_reg = "02" Or num_reg = "09" Or num_reg = "06" Or num_reg =
"07" Then
    If num_reg = "02" Then
        !num_region = num_reg
        rg_code = num_reg
    Else
        If num_reg = "01" Or num_reg = "09" Or num_reg = "06" Then
            rg_code = num_reg
            Debug.Print rg_code
        Else
            If num_reg = "07" Then
                check = Trim(Str(Mid(str1, 3, 1)))
                Debug.Print check
                If check = "3" Or check = "4" Or check = "5" Or check = "6" Or check = "7" Then
                    !num_region = Trim(Str(Mid(str1, 1, 3)))
                    rg_code = Trim(Str(Mid(str1, 1, 3)))
                Else
                    rg_code = "07"
                End If
            Else
                !num_region = Trim(Str(Mid(str1, 1, 3)))
                rg_code = Trim(Str(Mid(str1, 1, 3)))
                Debug.Print "rg_code = " & rg_code
            End If
        End If
    End If

!num_tele = str1
time_b = Trim(Mid(str2, 1, 2)) & "." & Trim(Mid(str2, 4, 2))
!time_begin = time_b
time_u = Mid(str4, 1, 2) * 60 + Int(Val(Mid(str4, 4, 5)) * -1) * -1
time_em = Val(Mid(time_b, 4, 2)) + time_u
If Val(time_em) > 59 Then

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายวิชาการ

```

time_em = time_em Mod 60
End If
If Val(time_em) < 10 Then
time_em = "0" & Trim(Str(time_em))
End If
If Val(time_u) > 59 Then
time_eh = Val(Mid(time_b, 1, 2)) + (time_u / 60)
If Val(time_eh) > 23 Then
time_eh = Val(time_eh) Mod 24
If time_eh <= 10 Then
time_eh = "0" & Trim(time_eh)
End If
End If
time_e = Str(time_eh) & ":" & Trim(time_em)
Else
time_e = Str(Val(Mid(time_b, 1, 2))) & ":" & Trim(time_em)
End If
!date_use = str3 & "/" & Trim(Str(Year(Date)))
!time_use = time_u
If Rs2.State = adStateOpen Then
Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select price from tele_price where (region_code = '0' & Trim(rg_code) & ") &
"And (sub_code=" & Trim(rg_sub_code) & ")", Conn, adOpenKeyset, , adCmdText
If Rs2.RecordCount <> 0 Then
!price_use = Val(time_u) * Rs2!price
Else
If rg_code = "02" Then
!price_use = 3
Else
!price_use = Val(time_u) * 3 'mobile
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ผู้ใช้มีหน้าที่ห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Rs2.Close 'add
Rs2.CursorLocation = adUseClient 'add
If rg_code = "02" Then
    !city = "กรุงเทพ"
Else
    If rg_code = "01" Or rg_code = "06" Or rg_code = "09" Or rg_code = "07" Then
        !city = "มืองี้อ"
    Else

        Debug.Print rg_code
        Rs2.Open "select region_name from tele_price where (region_code = '0' & Trim(rg_code) &
        '') & "And (sub_code=' " & Trim(rg_sub_code) & "')"; Conn, adOpenKeyset, , adCmdText
        If Rs2.RecordCount <> 0 Then
            !city = Rs2!region_name
        Else
        End If
        End If
        End If
        .Update
        End With
Wend

If Rs1.State = adStateOpen Then
    Rs1.Close
End If

Rs1.Open "select num_tele,city,time_begin,date_use,time_use,price_use from record", Conn,
adOpenKeyset, , adCmdText

Set DataGrid1.DataSource = Rs1
DataGrid1.Columns.Item(0).Width = 1500
DataGrid1.Columns.Item(3).Width = 1500
DataGrid1.Columns.Item(4).Width = 1500
DataGrid1.Columns.Item(0).Caption = "หมายเลขโทรศัพท์"
DataGrid1.Columns.Item(1).Caption = "จังหวัด" 'add
DataGrid1.Columns.Item(2).Caption = "เวลาที่โทรออก"
DataGrid1.Columns.Item(3).Caption = "วันที่โทรออก"

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม กรุณาแจ้งไปยังเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DataGrid1.Columns.Item(4).Caption = "จำนวนนาที่ที่ใช้"
DataGrid1.Columns.Item(5).Caption = "จำนวนเงิน"
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select sum(price_use) as sp from record". Conn, adOpenKeyset, , adCmdText
Label3 = Str(Rs2!sp) & " บาท"
cmdprint.Enabled = True
cmdrec.Enabled = True
cmdexit.Enabled = True
End Sub

```

```

Private Sub cmdrec_Click()
Dim lfile As Integer
cmdexit.Enabled = False
cmdprint.Enabled = False
cmdrec.Enabled = False
lfile = FreeFile
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select num_tele,time_begin,date_use,time_use,price_use,city from record", Conn,
adOpenKeyset, , adCmdText
On Error GoTo Error1
If Rs2.RecordCount <> 0 And Trim(Label3) <> "0 บาท" Then
Rs2.MoveFirst

```

```

Open "A:\tele.txt" For Output As #lfile

```

```

Print #lfile, "

```

```

รายการข้อมูลการใช้โทรศัพท์ "

```

```

Print #lfile,

```

```

Print #lfile, "เบอร์ที่โทร"; Tab; " เวลาที่โทร "; Tab; "วันที่ใช้"; Tab; " นาที่"; Tab; "จำนวนเงิน"; Tab;

```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Do
    Print #lfile, Rs2!num_tele; Tab; Rs2!time_begin; Tab; Rs2!date_use; Tab; Rs2!time_use;
Tab; Rs2!price_use; Tab; Rs2!city
    Rs2.MoveNext
Loop Until Rs2.EOF = True
Print #lfile,
Print #lfile, "                                รวมเป็นเงิน " & Label3
Close #lfile
MsgBox "บันทึกลงแผ่นเรียบร้อยแล้ว", vbInformation, "Information"
cmdexit.Enabled = True
cmdrec.Enabled = True
cmdprint.Enabled = True
Else
MsgBox "ไม่มีรายการให้บันทึกลงแผ่น", vbInformation, "Information"
'Call Form_Activate
'Call re
cmdexit.Enabled = True
cmdrec.Enabled = True
cmdprint.Enabled = True
End If
Exit Sub
Error1:
MsgBox "กรุณาตรวจสอบ Disket", vbExclamation, "Error!"
Call Form_Activate
Call re
cmdexit.Enabled = True
cmdrec.Enabled = True
cmdprint.Enabled = True
End Sub

Private Sub cmdprint_Click()
cmdexit.Enabled = False
cmdprint.Enabled = False

```

```

cmdrec.Enabled = False
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select num_tele,time_begin,date_use,time_use,price_use from record", Conn,
adOpenKeyset, , adCmdText
If Rs2.RecordCount <> 0 And Trim(Label3) <> "0 บาท" Then
    DataReport1.Show
Else
    MsgBox "ไม่มีรายการให้พิมพ์", vbInformation, "Information"
    'Call Form_Activate
    'Call re
    cmdexit.Enabled = True
    cmdrec.Enabled = True
    cmdprint.Enabled = True
End If

cmdexit.Enabled = True
cmdrec.Enabled = True
cmdprint.Enabled = True
End Sub

Private Sub cmdexit_Click()
    Unload Me
    Form3.Show
End Sub

Private Sub Form_Activate()
    DataGrid1.Columns.Add (1)
    DataGrid1.Columns.Add (1)
    DataGrid1.Columns.Add (1)
    DataGrid1.Columns.Add (1)
    DataGrid1.Columns.Item(0).Caption = "หมายเลขโทรศัพท์"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อสงสัยหรือข้อผิดพลาดประการใด กรุณาแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DataGrid1.Columns.Item(1).Caption = "จังหวัด"           'add
DataGrid1.Columns.Item(2).Caption = "เวลาที่โทรออก"
DataGrid1.Columns.Item(3).Caption = "วันที่โทรออก"
DataGrid1.Columns.Item(4).Caption = "จำนวนนาทีที่ใช้"
DataGrid1.Columns.Item(5).Caption = "จำนวนเงิน"
Label3 = "0 บาท"

```

```
End Sub
```

```

Private Sub Form_Load()
If Conn.State = adStateOpen Then
    Conn.Close
End If
Conn.Open "DRIVER=Microsoft Access Driver (*.mdb);UID=admin;DBQ=" & App.Path &
"\tele_count.mdb"
MSComm1.PortOpen = True
cmdprint.Enabled = False
cmdrec.Enabled = False
'cmdexit.Enabled = False
End Sub
Private Sub Timer1_Timer()

temp = (MSComm1.Input)
'temp = "$003_~~073473161 00:49 27/10 00.10.02_~~042426956 00:50 27/10
00.05.02_~~042372222 12:51 27/10 00.28.30_~~078646732 10:00 27/10 00.20.00_~~098646732
00:51 28/10 00.10.00_~~075246732 00:51 30/10 01.00.30_~~077346732 01:51 30/10
00.15.00_~~077546732 00:51 27/10 00.28.30_~~077846732 00:51 27/10 00.28.30_~~073246732
00:51 27/10 00.28.30_~~028646732 00:51 27/10 00.28.30_~~074646732 00:51 27/10
00.28.30_~~075646732 00:51 27/10 00.28.30_~~028646732 00:51 27/10 00.28.30_~~042746732
00:51 27/10 00.28.30_~~039346732 00:51 27/10 00.28.30_~~039546732 00:51 27/10
00.28.30_~~032446732 00:51 27/10 00.28.30_~~"

```

เอกสารนี้จัดทำขึ้นเพื่อสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อผิดพลาดประการใดขออภัยและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Call Cmd1_Click
Timer1.Enabled = False
End If
End Sub
Private Sub re()
If Rs1.State = adStateOpen Then
    Rs1.Close
End If
Rs1.Open "select num_tele,city,time_begin,date_use,time_use,price_use from record", Conn,
adOpenKeyset, , adCmdText
Set DataGrid1.DataSource = Rs1
DataGrid1.Columns.Item(0).Width = 1500
DataGrid1.Columns.Item(3).Width = 1500
DataGrid1.Columns.Item(4).Width = 1500
DataGrid1.Columns.Item(0).Caption = "หมายเลขโทรศัพท์"
DataGrid1.Columns.Item(1).Caption = "จังหวัด" 'add
DataGrid1.Columns.Item(2).Caption = "เวลาที่โทรออก"
DataGrid1.Columns.Item(3).Caption = "วันที่โทรออก"
DataGrid1.Columns.Item(4).Caption = "จำนวนนาที่ที่ใช้"
DataGrid1.Columns.Item(5).Caption = "จำนวนเงิน"
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select sum(price_use) as sp from record", Conn, adOpenKeyset, , adCmdText
Label3 = Str(Rs2!sp) & " บาท"
cmdprint.Enabled = True
cmdrec.Enabled = True
cmdexit.Enabled = True
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
End Sub
```

```
Private Sub Command2_Click()
```

```
    If Text2 <> "" Then
```

```
        Adodc1.Recordset.MoveFirst
```

```
        Adodc1.Recordset.Find "region_name = " & Text2 & "'", , adSearchForward
```

```
        If Adodc1.Recordset.EOF Then
```

```
            MsgBox "ไม่พบข้อมูลที่ต้องการค้นหา", vbInformation, "Information"
```

```
            Adodc1.Recordset.MoveFirst
```

```
            Text2.SetFocus
```

```
            Text2 = ""
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    Unload Me
```

```
    Form3.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Text1 = ""
```

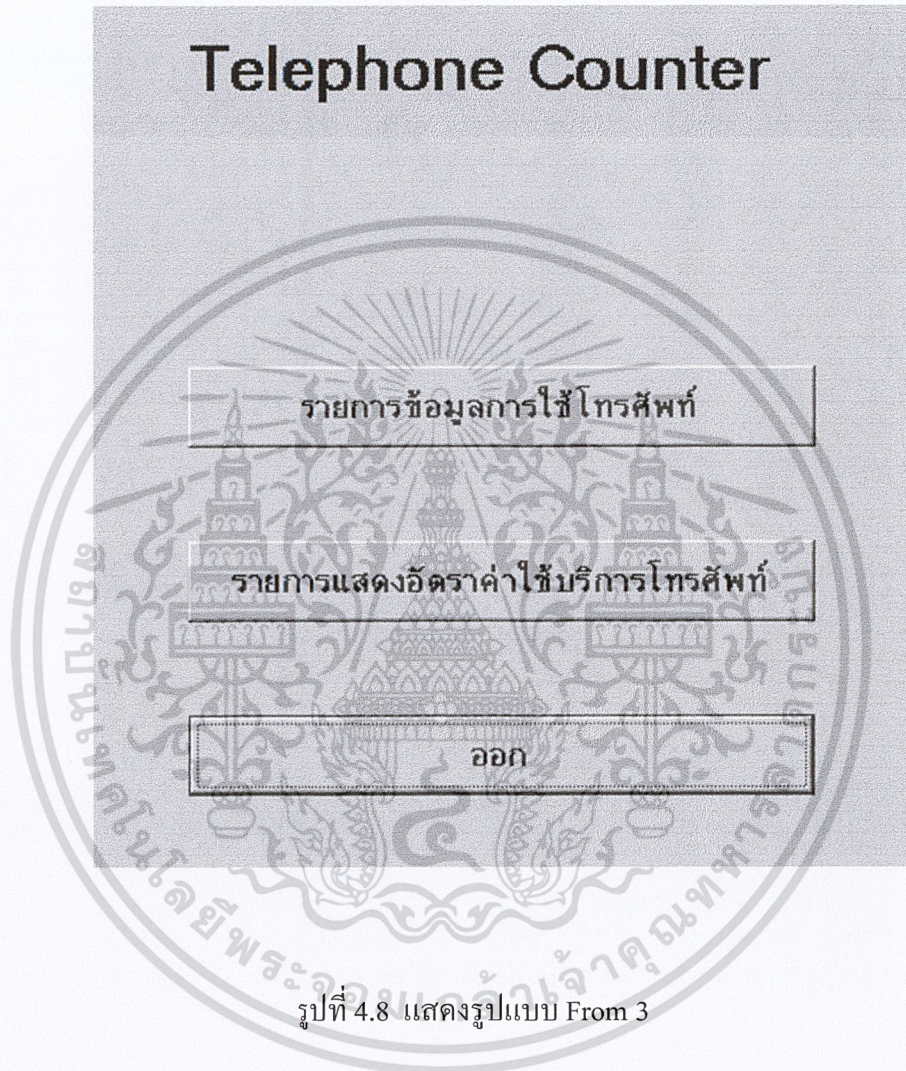
```
    Text2 = ""
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 โปรแกรมที่ใช้ในการคอนโทรล Visual Basic (From 3)

From 3 ใช้เป็นเมนูในการเลือกที่จะเข้าสู่การแสดงผลต่าง ๆ



รูปที่ 4.8 แสดงรูปแบบ From 3

From3

Private Sub Command1_Click()

Form1.Show

Unload Me

End Sub

Private Sub Command2_Click()

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub Command3_Click()

End

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบและการสร้าง

ส่วนของ วงจรตีเทคเตอร์

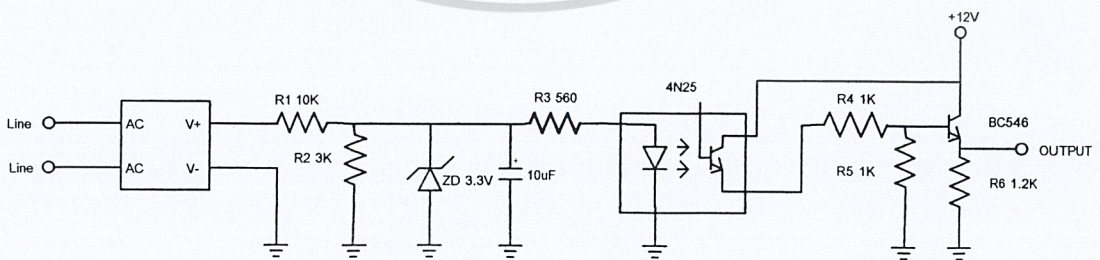


รูปที่ 5.1 บล็อกไดอะแกรมแสดงส่วนประกอบของโครงการ

5.1 ส่วนวงจรตีเทคเตอร์

วงจรส่วนนี้ประกอบด้วยวงจรย่อย 3 วงจรด้วยกัน ได้แก่ วงจรตรวจสอบการยกหูโทรศัพท์ วงจรถอดรหัสเลขหมาย วงจรถอดรหัสคีย์แพค และวงจรตรวจสอบจับสัญญาณเสียง วงจรเหล่านี้จะต่ออยู่กับคู่สายโทรศัพท์ของเครื่องที่ต้องการจะแสดงข้อมูลการใช้งาน โทรศัพท์ที่รองรับสัญญาณอินพุตจากคู่สายโทรศัพท์นั้นและเอาต์พุตของวงจรจะส่งไปยังส่วนประมวลผลควบคุมการทำงาน

5.1.1 วงจรตรวจสอบการยกหูโทรศัพท์



รูปที่ 5.2 วงจรตรวจสอบการยกหูโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อป้อนสัญญาณคู่สายโทรศัพท์เข้าวงจร เราจะใช้วงจรเปรียบเทียบสัญญาณโทรศัพท์ในการยกหูและวางหู การทำงานของโทรศัพท์นั้นในสภาวะปกติ (วางหู) จะมีแรงดันตกคร่อมคู่สายโทรศัพท์ประมาณ 48-50 โวลต์ เมื่อยกหูจะมีแรงดันลดลงเหลือประมาณ 6-10 โวลต์ ขึ้นอยู่กับชนิดของโทรศัพท์แต่ละเครื่องวงจรทำงานจะใช้วงจรเปรียบเทียบแรงดันและใช้ออปโตไดโอดเป็นตัวยกหรือลดระดับแรงดันจาก 12 โวลต์ เป็น 5 โวลต์

ขณะที่วางหูแรงดันไฟตรง 48 โวลต์ จะผ่านวงจรบริดจ์ซึ่งจะทำให้ขั้วของแรงดันแน่นอนหลังจากนั้นจะคร่อมตัวต้านทานและซีเนอร์ไดโอด 12 โวลต์ เป็นผลทำให้เกิดการนำกระแสของ LED ในออปโตไดโอด และไปกระตุ้นให้ทรานซิสเตอร์นำกระแสตามไปด้วย เกิดแรงดันตกคร่อมตัวต้านทาน 1.2 กิโลโอห์มที่เอาต์พุตเป็น 5 โวลต์ จะแทนด้วย ลอจิก “1”

ขณะที่ยกหู แรงดันไฟฟาลลดลงเหลือประมาณ 6-9 โวลต์ ตกคร่อมทำให้มีแรงดันเข้ามาน้อยกว่าแรงดันอ้างอิง ทำให้เอาต์พุตของออปแอมป์มีค่าประมาณ 0 โวลต์ เป็นผลทำให้ LED และทรานซิสเตอร์ในออปโตไดโอดไม่นำกระแส แรงดันเอาต์พุตจึงตกลงเป็น 0 โวลต์ ซึ่งจะแทนด้วยระดับลอจิก “0”

5.2 การออกแบบวงจรตรวจจับสัญญาณเสียง

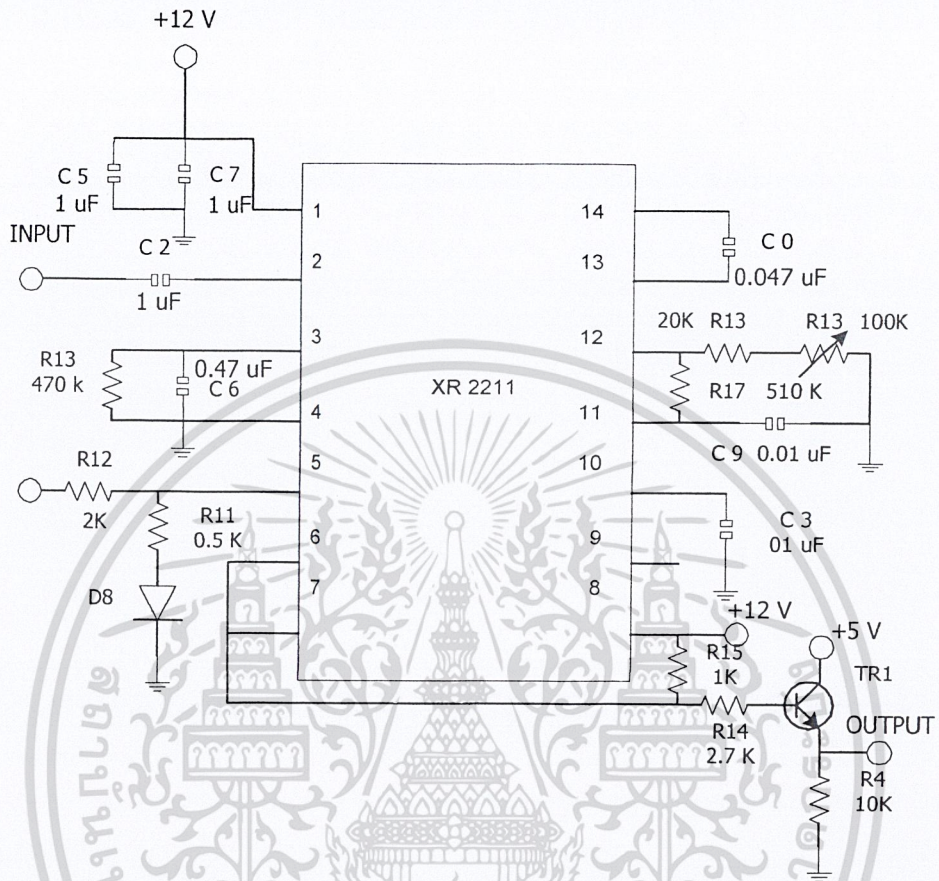
การออกแบบวงจรตรวจจับสัญญาณเสียงนั้น ก่อนอื่นเราจะต้องกำหนดความถี่ของสัญญาณโทนต่าง ๆ จะอยู่ระหว่าง 400-500 เฮิร์ต ดังนั้นเราจะกำหนดให้ความถี่ตั้งแต่ 500-4000 เฮิร์ต เพื่อให้รับกวนกับความถี่สัญญาณ โทน โดยกำหนดให้สัญญาณ โทนอยู่ระหว่าง 400-500 เฮิร์ต

การคำนวณวงจรตรวจสอบสัญญาณ

1. เลือก $C_0 = 0.047\mu\text{F}$
2. เลือก $R_x = 5\text{K}$
3. หา $R_0 = 1/C_0 \cdot F_0$

$$= 1/0.047 \cdot 10^{-6} \cdot 400 = 53\text{K}$$
4. หา $C_1 = 0.25C_0 = 0.25 \cdot 0.047 = 0.01\mu\text{F}$
5. หา $R_1 = R_0 \cdot (f_0/\Delta f) = 53 \cdot 10^3 \cdot (400/40) = 530\text{K}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 วงจรตรวจจับสัญญาณเสียง

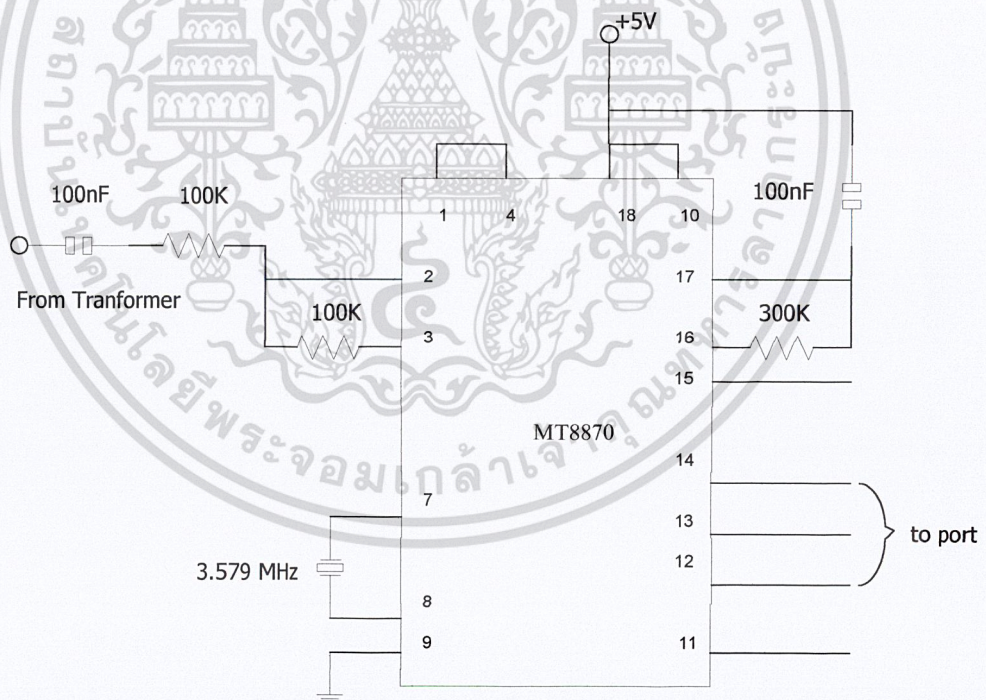
การทำงานและการปรับแต่ง

ก่อนที่จะนำวงจรไปใช้งานจะต้องมีการปรับแต่งวงจรเสียก่อน โดยการป้อนความถี่ 400 เฮิร์ตเข้าไปที่อินพุตของวงจรตรวจสอบสัญญาณ สังเกตว่าก่อนป้อนอินพุต LED จะติด และเมื่อป้อนอินพุตเข้าไปแล้ว LED จะดับ ถ้า LED ยังติดอยู่ให้ปรับ Rx จนกระทั่ง LED ดับ จากนั้นให้เพิ่มความถี่อินพุตไปเรื่อย ๆ จนถึงค่าประมาณ 480 เฮิร์ต LED จะติด ถ้า LED ยังดับอยู่ให้ปรับ Rx จน LED ติด เอาต์พุตที่ขา 6 ของไอซี XR-2211 จะเป็นลอจิก “0” เมื่อ LED ติด และจะเป็นลอจิก “1” เมื่อ LED ดับเมื่อปรับแต่งวงจรเสร็จแล้วจึงจะนำวงจรไปใช้งานได้โดยที่ความถี่ 400-450 เฮิร์ต วงจร detect tone จะให้เอาต์พุตเป็นลอจิก “1” นำสัญญาณเอาต์พุตที่ขา 6 ไปขับทรานซิสเตอร์เพื่อให้ส่งลอจิก “1” เป็นบวก 5 โวลต์ ไปที่ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การออกแบบวงจรถอดรหัสสัญญาณ DTMF

ในการออกแบบวงจร decode สัญญาณ DTMF นั้นเราจะใช้ IC MT 8870 ซึ่งจะทำหน้าที่ในการถอดรหัสความถี่ที่เอ็มเอฟออกมาเป็นรหัสไบนารี 4 บิต ภายใน MT 8870 ซึ่งประกอบไปด้วย ภาคกรองความถี่ ภาคถอดรหัส ภาคตรวจสอบสัญญาณ ภาคขยายสัญญาณความแตกต่าง และภาคกำเนิดความถี่ โดยใช้คริสตอล 3.57954 เมกกะเฮิร์ตซ์เป็นฐานเวลา จากวงจรจะใช้เมฆซึ่งมาเป็นตัวคัปปลิ่งสัญญาณจากคู่สายโทรศัพท์มายังวงจรเฉพาะสัญญาณไฟกระแสดับเท่านั้น และยังทำหน้าที่ในการแยกกราวด์ของวงจรอีกด้วย เพราะบนคู่สายโทรศัพท์จะมีสัญญาณรบกวนต่าง ๆ มากมาย ซึ่งอาจจะมีผลต่อการตีโค้ดสัญญาณได้ สัญญาณจากการกดหมายเลขหนึ่งตัว จะถูกถอดรหัสออกเป็นเลขไบนารี 4 บิต ทางขา 11 , 12 , 13 และ 14 (Q1-Q4) ส่วนสัญญาณที่ขา 15 จะเป็น “1” ขณะกดปุ่มโทรศัพท์ เมื่อไม่กดจะเป็นสัญญาณ “0” สัญญาณนี้จะบอกให้ส่วนควบคุมทราบว่ามีการกดปุ่มโทรศัพท์ จากนั้นที่ไอซีจะทำการแลตซ์ค่าสัญญาณนั้นเอาไว้จนกว่าจะได้รับสัญญาณใหม่เข้ามาแล้วจึงค่อยเปลี่ยนแปลงเป็นรหัสตัวใหม่

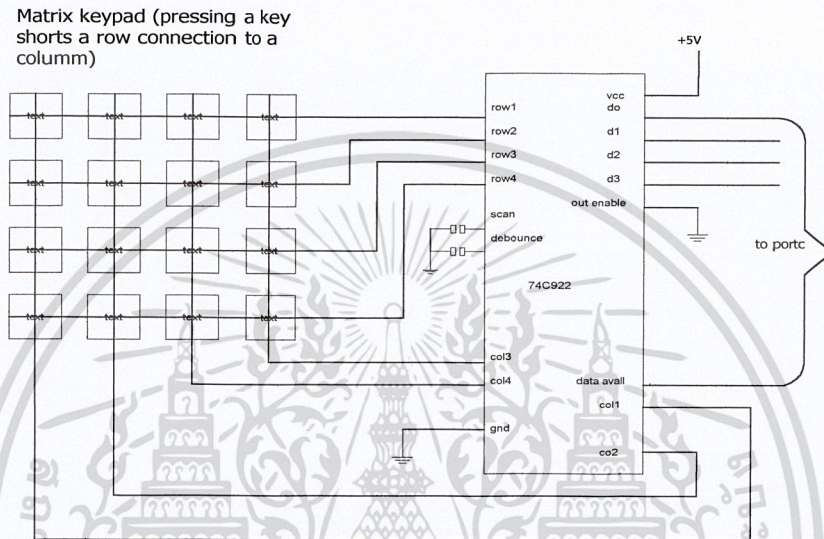


รูปที่ 5.4 แสดงวงจร Decode DTMF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 การออกแบบวงจรถอดรหัส keypad

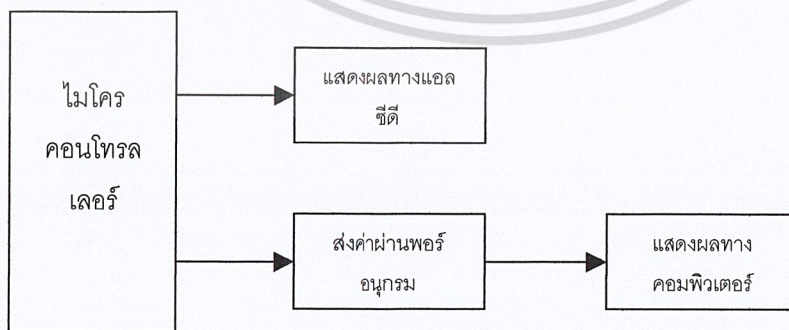
ในการออกแบบ decode สัญญาณ DTMF นั้น จาก Application ที่ได้ศึกษาสามารถใช้ค่า Cext ในการออกแบบเพื่อแก้ปัญหา Bounce โดยใช้ค่า 1 นาโนฟารัด และความถี่ ออสซิลเลเตอร์ที่ใช้ในการสแกนก็ใช้ตัวเก็บประจุค่า 1 นาโนฟารัด เช่นกัน แสดงดังรูปที่ 5.5



รูปที่ 5.5 แสดงวงจร scan keypads

5.5 ส่วนควบคุมการทำงาน

ในโครงงานนี้จะใช้ Microcontroler ในการเขียนโปรแกรมควบคุมสัญญาณต่าง ๆ รวมถึงการแสดงผลทาง LCD และส่งค่าออก พอร์ต RS-232 ไปเข้าโปรแกรม Visual Basic



รูปที่ 5.6 แสดงบล็อกไดอะแกรมการทำงานของ ไมโครคอนโทรลเลอร์

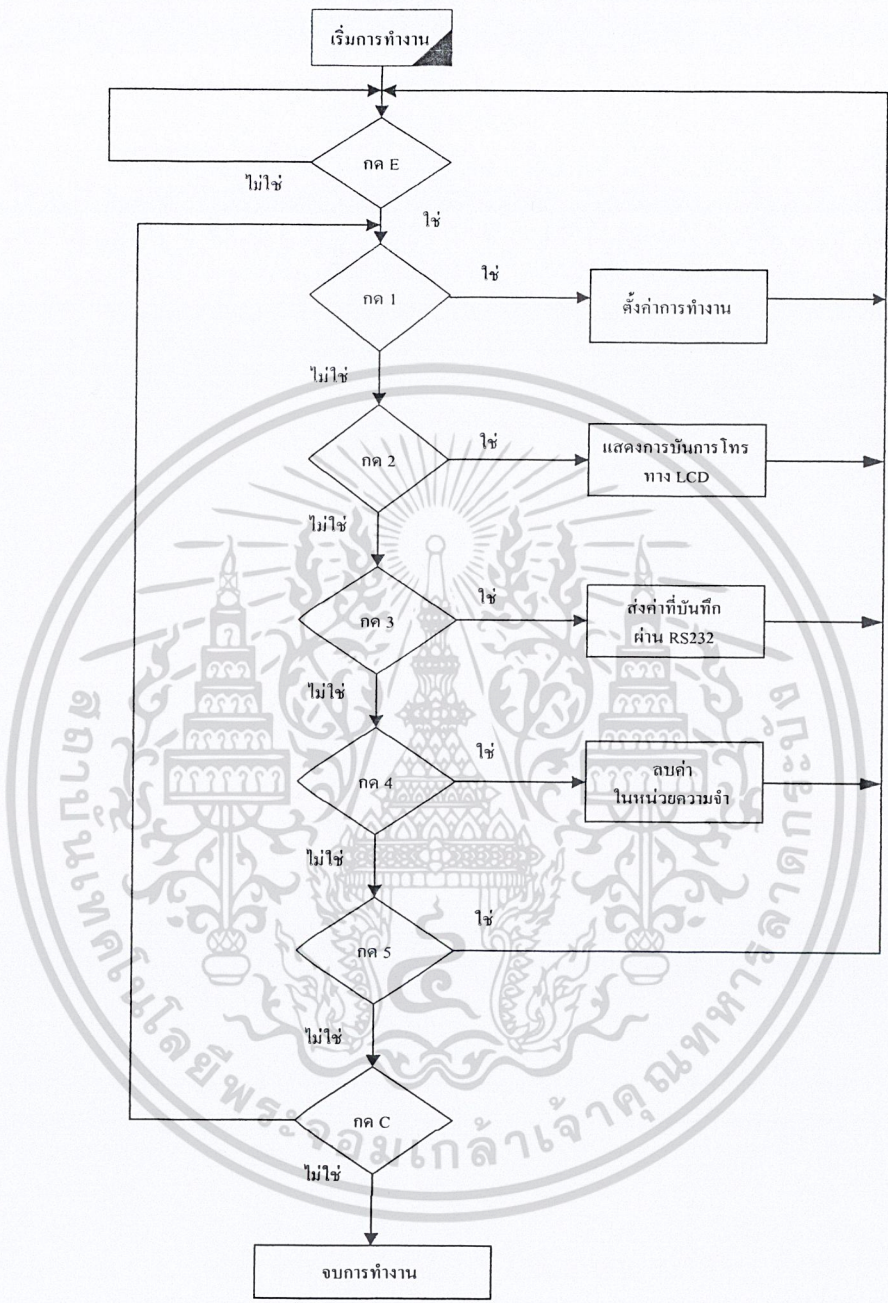
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 การออกแบบส่วน Soft ware

โปรแกรมที่อยู่ในส่วนควบคุมโดยใช้ไมโครคอนโทรลเลอร์ เขียนเป็นโพลีชาร์ต ดังที่ได้แสดงไว้ในรูปที่ 5.7

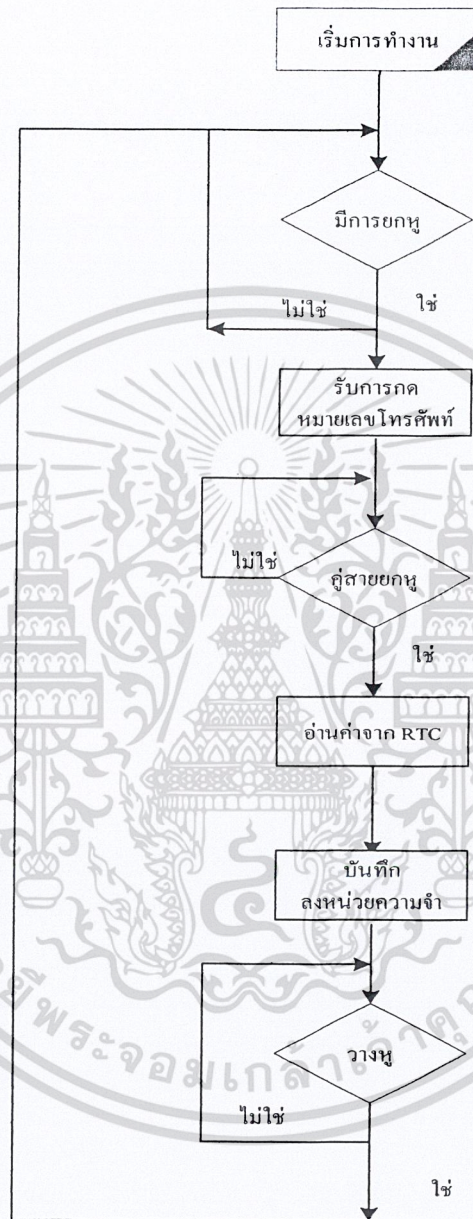
การออกแบบส่วนของโปรแกรม จะใช้การตรวจจับสัญญาณเพื่อใช้ในการนับจำนวนการใช้งานโทรศัพท์ เมื่อเริ่มต้นโปรแกรมจะทำการเคลียร์รีจิสเตอร์ต่าง ๆ และเข้าสู่โหมดการตั้งวันที่และเวลาเมื่อยกหูโทรศัพท์โปรแกรมจะเช็คสัญญาณให้หมุน และหน้าจอจะแสดงคำว่า “Tel Out” และจะรอรับการกดหมายเลข เมื่อได้มีการกดเลขหมายแล้วโปรแกรมจะเช็คสัญญาณที่ส่งมาจากชุมสายโทรศัพท์ ถ้าสถานะของเครื่องทางผู้รับมีสถานะเป็น “ไม่ว่าง” ทางชุมสายโทรศัพท์จะส่งสัญญาณ “สายไม่ว่าง” กลับมาโปรแกรมก็จะทำการเช็ค แต่ถ้าสถานะของเครื่องโทรศัพท์ทางผู้รับมีสถานะ “ว่าง” ทางชุมสายโทรศัพท์ก็จะส่ง “สัญญาณเรียก” กลับมาโปรแกรมก็จะทำการเช็คสัญญาณเมื่อผู้รับทำการรับสายโปรแกรมก็จะทำการนับการใช้โทรศัพท์และจะนำรายละเอียดของเบอร์โทรศัพท์ วัน เดือน ปี ที่เรียกออก เวลาเริ่มต้น และ เวลาสิ้นสุดในการเรียกออกไปเก็บไว้ในหน่วยความจำภายในเครื่อง นอกจากนี้เรายังสามารถนำข้อมูลต่าง ๆ เหล่านี้ส่งออกทางพอร์ตอนุกรมเพื่อส่งข้อมูลให้กับเครื่องคอมพิวเตอร์ได้ โดยโปรแกรมจะทำการติดต่อกับเครื่องคอมพิวเตอร์ให้ทำการแสดงข้อมูลและทำการพิมพ์ออกทางเครื่องพิมพ์ ส่วนซอร์สโค้ดของโปรแกรมได้แสดงไว้ในส่วนของภาคผนวก ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 รูปแสดงการควบคุมโดยใช้ไมโครคอนโทรลเลอร์ในส่วนเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 รูปแสดงการควบคุมโดยใช้ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7 ส่วนของโปรแกรมประยุกต์

การออกแบบ Software ในส่วนของ Visual Basic เราจะใช้โปรแกรม Microsoft Access มาช่วยในการเก็บข้อมูลและได้แสดงผลแต่ละ Form ที่จัดทำรวมทั้งโปรแกรมไว้ในข้างต้นในบทที่ 2 ส่วนซอร์สโค้ดของโปรแกรมแสดงดังภาคผนวก ก

5.8 ส่วนแสดงผล

ในการติดต่อับส่วนควบคุมนั้นเราจะใช้ Visual Basic ซึ่งเป็นภาษาที่ใช้ง่ายได้รับความนิยมสูง อีกทั้งยังมี Component สำเร็จรูปต่างให้ใช้งานมากมาย ในการสื่อสารผ่านพอร์ตอนุกรมนั้นเราใช้ Component ชื่อ Microsoft Comm Control ที่สร้างมาขึ้นเพื่อการสื่อสารกับพอร์ตอนุกรมโดยตรง

5.9 การทำงานของวงจรรวม

ที่ภาคตรวจสอบสถานะขงหว วงหุ เมื่อไมม่มีการยกหุจะมีระดับแรงดันลดลงเหลือ 9-10 โวลต์ จากเดิม 49-50 โวลต์ ทำให้ค่าแรงดันมีค่าน้อยกว่าแรงดันอ้างอิงที่ตั้งไว้จากวงจรตรวจสอบภาวะขงหว วงหุที่มีค่าประมาณ 11 โวลต์ ทำให้เอาต์พุตเปลี่ยนจากลอจิก “1” เมื่อวงหุเป็นลอจิก “0” บอกให้ไมโครคอนโทรลเลอร์ทราบว่าการยกหุ ไมโครคอนโทรลเลอร์จะทำการ on delay ทำให้มีสัญญาณ dial tone , ring back tone , busy tone ไหลผ่านหม้อแปลงอิมพีแดนซ์ค้บปล้ึง 600 โอห์ม ไปสู่วงจรภาคตรวจสอบสัญญาณค่าง ๆ เพื่อตรวจจับสัญญาณที่เข้ามาตามคาบเวลาของสัญญาณที่เข้ามานั้น ๆ แล้วส่งให้ไมโครคอนโทรลเลอร์ทราบ หลังจากนั้นก็จะทำการประมวลผลต่อไปยังส่วนวงจรตรวจสอบสัญญาณ DTMF สัญญาณ DTMF จะผ่านหม้อแปลงเข้ามาเพื่อป้อนให้กับวงจรภาคตรวจสอบสัญญาณ DTMF เอาต์พุตที่ได้จะเป็นสัญญาณดิจิทัล 4 บิต จะส่งต่อไปยังพอร์ต B ของ 8255 เพื่อส่งไปให้ไมโครคอนโทรลเลอร์ประมวลผลต่อไป ส่วนในภาควงจรถอดรหัสการกดคีย์ เมื่อมีการกดคีย์ใด ๆ ไอซี 74C922 จะทำการถอดรหัสออกมาทำให้ได้เอาต์พุตเป็นดิจิทัล 4 บิต เช่นเดียวกับไอซี MT 8870 หลังจากนั้นจะส่งต่อไปยังพอร์ต C ของ 8255 เพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ทำการประมวลผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและผลการทดลอง

รายละเอียดเกี่ยวกับโครงการ

ในโครงการนี้ส่วนประกอบของโครงการประกอบด้วยสองส่วนใหญ่ ๆ คือ โครงสร้างทางฮาร์ดแวร์ และโครงสร้างทางซอฟต์แวร์ซึ่งโครงสร้างทางฮาร์ดแวร์นั้นจะประกอบด้วยวงจรต่าง ๆ และประยุกต์ใช้ไมโครคอนโทรลเลอร์ในการควบคุมและคอนโทรล สำหรับในส่วนของซอฟต์แวร์นั้นเราใช้โปรแกรม Visual Basic ประกอบกับโปรแกรม Microsoft Access ในการแสดงผลจำนวนค่าบริการและจำนวนครั้งที่โทรออก

6.1 วงจรตรวจสอบสถานะยกหู วางหู

ผลการทดลองในส่วนวงจรตรวจสอบสถานะยกหู วางหู ดังรูปที่ 6.1 คือ จะเห็นว่าขณะที่คู่สายวางหูอยู่แรงดันไฟฟ้าจะถูกเปลี่ยนให้เป็นแรงดัน 5 โวลต์ แทนด้วยระดับลอจิก “1” เพื่อให้สามารถใช้งานร่วมกับระบบไมโครคอนโทรลเลอร์ได้ส่วนในขณะการยกหูแรงดันจะลดลงเหลือ 0 โวลต์ แทนด้วยระดับลอจิก “0” รูปด้านบนเป็นสัญญาณที่วัดคร่อมคู่สายโทรศัพท์ซึ่งเป็นขณะที่ทำการวางหุระดับแรงดันไฟฟ้าจะมีค่าประมาณ 48 โวลต์ ขณะที่ทำการยกหูแรงดันไฟฟ้าลดลงเหลือประมาณ 10 โวลต์ดังรูปที่ 6.1

6.2 วงจรตรวจสอบสัญญาณการเรียกกลับ

เป็นสัญญาณที่ได้จากการตรวจสอบสัญญาณการเรียกกลับ ซึ่งแทนด้วยระดับของลอจิก “0” เมื่อมีสัญญาณเรียกกลับความถี่ประมาณ 425 Hz เข้ามา และแทนด้วยระดับลอจิก “1” เมื่อมีความถี่อื่นเข้ามาดังรูปที่ 6.2

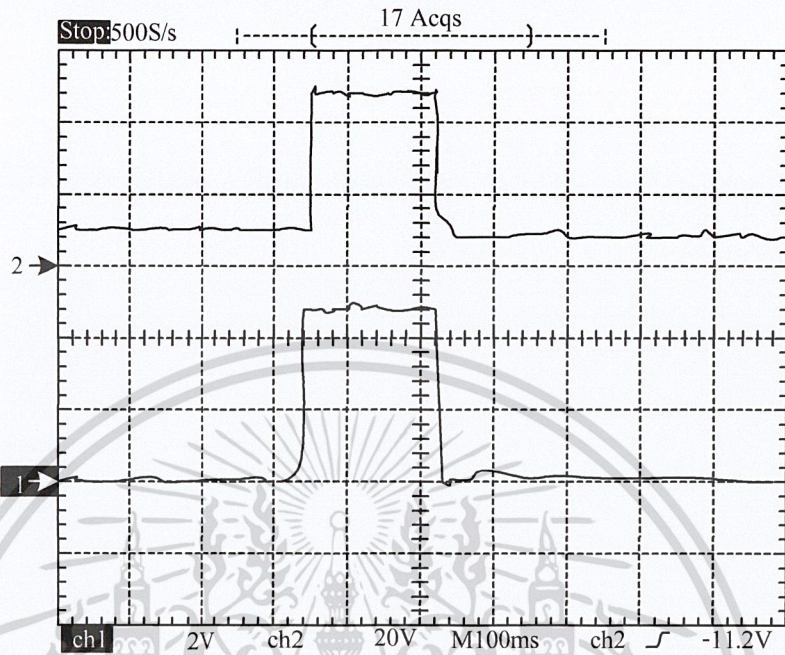
6.3 สัญญาณการตรวจสอบการกดหมายเลข

เป็นสัญญาณที่ได้จากการตรวจสอบสัญญาณให้หมุนหรือให้กดหมายเลขเพื่อเป็นข้อมูลการเรียกฝ่ายผู้รับดังรูปที่ 6.3

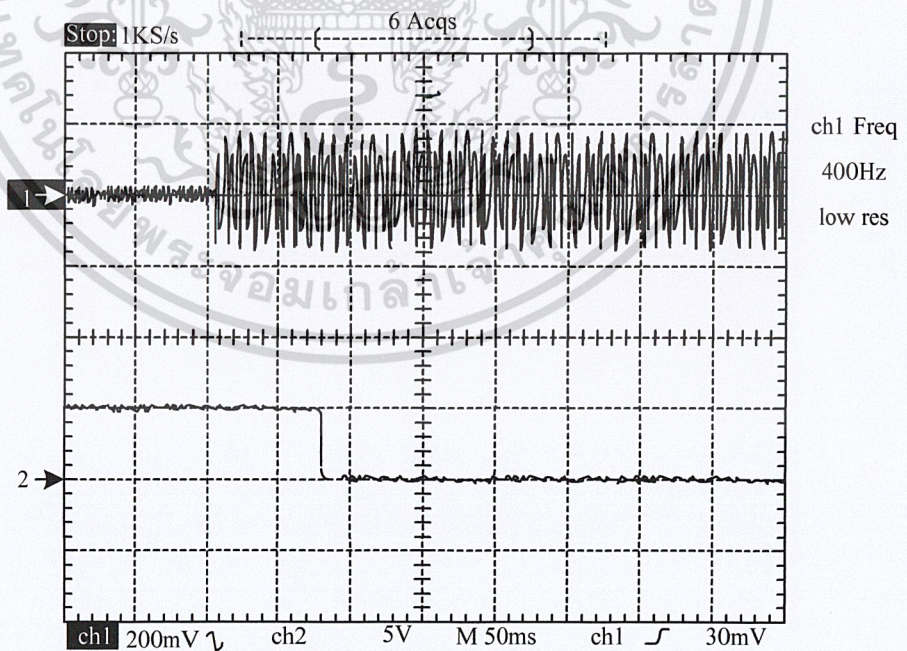
6.4 สัญญาณสายไม่ว่าง

เป็นสัญญาณที่บอกให้ทราบว่าทางด้านผู้รับขณะนี้สายยังไม่ว่างแสดงดังรูปที่ 6.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

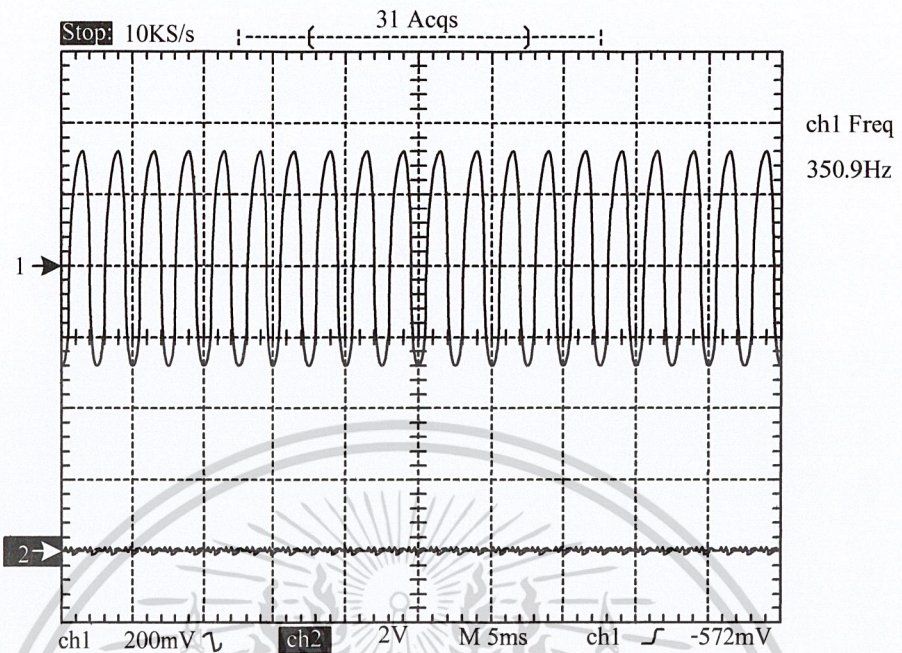


รูปที่ 6.1 แสดงรูปสัญญาณที่ได้จากวงจรตรวจสอบการยกหู

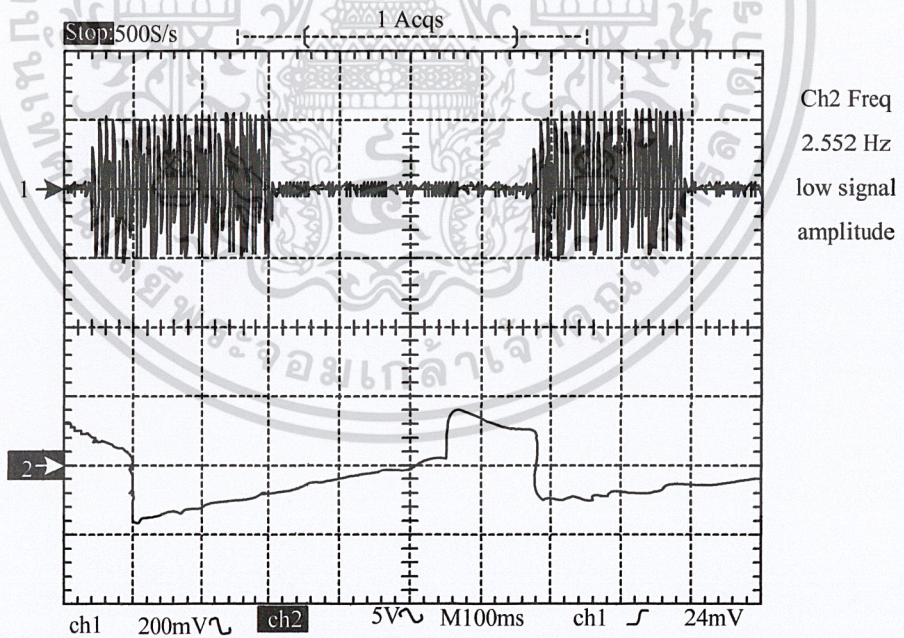


รูปที่ 6.2 รูปแสดงสัญญาณการเรียกกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 แสดงสัญญาณการให้คดหมายเลข



รูปที่ 6.4 แสดงสัญญาณสายไม่ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 วงจร Decode สัญญาณ DTMF

ตารางที่ 39 แสดงการทดสอบวงจรส่วนการถอดรหัสเลขหมายโดยใช้ไอซี MT8870

Flow	Fhigh	DIGIT	D3	D2	D1	D0
697	1219	1	0	0	0	1
697	1336	2	0	0	1	0
697	1477	3	0	0	1	1
770	1209	4	0	1	0	0
770	1336	5	0	1	0	1
770	1477	6	0	1	1	0
852	1209	7	0	1	1	1
852	1336	8	1	0	0	0
852	1477	9	1	0	0	1
941	1209	0	1	0	1	0
941	1336	*	1	0	1	1
941	1477	#	1	1	0	0
697	1633	A	1	1	0	1
770	1633	B	1	1	1	0
852	1633	C	1	1	1	1
941	1633	D	0	0	0	0

ผลการทดลองของวงจรเราได้นำเอาตารางเอาต์พุตของวงจรมาแสดง เพราะว่าจากการทดลองเขียนโปรแกรมรับข้อมูลเมื่อมีการถอดรหัสสัญญาณแล้ว ได้ข้อมูลเอาต์พุต 4 บิต ดังตารางข้างบนนี้

6.6 วงจรถอดรหัสการกดคีย์

ผลการทดลองจากวงจรมี จะได้อาต์พุตเป็นสัญญาณดิจิทัล 4 บิต ในการทดลองเขียนโปรแกรมรับข้อมูลจะได้อาต์พุตดังตารางข้างล่างนี้

เอกสารที่ 40 แสดงเอาต์พุตที่ได้จากการทดสอบวงจรถอดรหัสการกดคีย์ที่ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROW	COLUMN	D	C	B	A
Y1	X1	0	0	0	0
Y1	X2	0	0	0	1
Y1	X3	0	0	1	0
Y1	X4	0	0	1	1
Y2	X1	0	1	0	0
Y2	X2	0	1	0	1
Y2	X3	0	1	1	0
Y2	X4	0	1	1	1
Y3	X1	1	0	0	0
Y3	X2	1	0	0	1
Y3	X3	1	0	1	0
Y3	X4	1	0	1	1
Y4	X1	1	1	0	0
Y4	X2	1	1	0	1
Y4	X3	1	1	1	0
Y4	X4	1	1	1	1

6.7 การทดลองการแสดงผลของ Visual Basic

ในส่วนแสดงผลนี้จะทำการแสดงค่าที่รับได้จากพอร์ต RS-232 มาเก็บไว้ในคอมพิวเตอร์ และเมื่อทำการ Run โปรแกรม Visual Basic จะปรากฏดังรูปที่ 6.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการข้อมูลการใช้โทรศัพท์

หมายเลขโทรศัพท์	จังหวัด	เวลาที่โทรออก	วันที่โทรออก	จำนวนนาทีที่ใช้	จำนวนเงิน
▶ 027374477	กรุงเทพฯ	00.49	27/10/2003	1	3
067526959	ฉะเชิงเทรา	00.50	27/10/2003	1	3
029972222	กรุงเทพฯ	00.51	27/10/2003	29	3

รวมเป็นเงิน 9 บาท

พิมพ์รายการ

บันทึกลงแผ่น

ออก

รูปที่ 6.5 แสดงผลการทดลองของ From 1

จากรูปปุ่มคำสั่งสามารถอธิบายได้ดังนี้

ปุ่มพิมพ์รายการ คือ สามารถพิมพ์ข้อมูลออกทางเครื่องพิมพ์ได้

ปุ่มบันทึกลงแผ่น คือ สามารถบันทึกข้อมูลลงแผ่นไดร์ A

ปุ่มออก คือ สามารถถลันออกสู่หน้าจอหลักหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการแสดงอัตราค่าใช้บริการโทรศัพท์

รหัสเขต	เริ่มต้นด้วยเลข	จังหวัด	ราคา/นาที
▶ 038	2	ชลบุรี	6
038	3	ชลบุรี	6
038	4	ชลบุรี	6
038	5	ฉะเชิงเทรา	6
038	6	ระยอง	9
039	3	จันทบุรี	12
039	4	จันทบุรี	12
039	5	ตราด	12
042	2	อุดรธานี, ทลิ่งบัวลำภู	15

ค้นหารหัสเขต

038

ค้นหา

ค้นหาชื่อเขต

ค้นหา

ออก

รูปที่ 6.6 แสดงผลการทดลองของ From 2

จากรูปปุ่มคำสั่งสามารถแสดงได้ดังนี้

ค้นหารหัสเขต คือ สามารถพิมพ์รหัสเขตแล้วโปรแกรมจะค้นหาให้อัตโนมัติ

ค้นหารหัสพื้นที่ คือ สามารถพิมพ์ชื่อจังหวัดแล้วโปรแกรมสามารถค้นหาอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Telephone Counter

รายการข้อมูลการใช้โทรศัพท์

รายการแสดงอัตราค่าใช้บริการโทรศัพท์

ออก

จากรูปที่ 6.7 แสดงผลการทดลองของ From 3

จากรูปปุ่มคำสั่งสามารถแสดงได้ดังนี้

รายการข้อมูลการใช้โทรศัพท์

คือ เข้าสู่ From ข้อมูลแสดงผลการใช้โทรศัพท์ทั้งหมด

รายการแสดงอัตราค่าบริการโทรศัพท์

คือ เข้าสู่ From ข้อมูลแสดงผลรหัสเขตและจังหวัด

ออก

คือ กลับเข้ามาสู่ windows ปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานข้อมูลการใช้โทรศัพท์

027374477	กรุงเทพ	27/10/2003	00.49	1	3
067526959	มือถือ	27/10/2003	00.50	1	3
029972222	กรุงเทพ	27/10/2003	00.51	29	3

รวมเป็นเงิน 9 บาท



รูปที่ 6.8 แสดงผลการพิมพ์ของ From1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทสรุป ปัญหา และข้อเสนอแนะ

สรุปการจัดทำโครงการ

จากการศึกษาโครงการโปรเจกต์เรื่อง Development of Telephone Counter เราสามารถทำการตรวจสอบการยกหูการวางหู และการโทรออกในพื้นที่เดียวกันได้ และสามารถที่จะนำข้อมูลมาเขียนโปรแกรมเพื่อคำนวณจำนวนเงินที่โทรออกในพื้นที่เดียวกันมาเก็บไว้ที่ Drive A และสามารถแสดงผลออกทาง Printer ได้อย่างถูกต้อง ตามที่ได้ตั้งวัตถุประสงค์ไว้

ปัญหาที่เกิดขึ้นจากการทำโครงการ

1. จะสามารถใช้ได้ในเขตกรุงเทพมหานครเท่านั้นเนื่องจากข้อจำกัดในการเขียนโปรแกรมไม่เหมือนกันของแต่ละจังหวัด
2. จะต้องต่อใช้สายตรงจากองค์การเท่านั้น ไม่สามารถต่อใช้กับ (PABX) ได้

ข้อเสนอแนะ

1. ต้องออกแบบวงจรในส่วนของการตรวจสอบสัญญาณการยกหู และวางหูให้ดียิ่งขึ้นกว่าเดิม
2. เขียนโปรแกรมการตรวจเช็คเพิ่มขึ้นจากสัญญาณที่ได้จากวงจรการตรวจสอบสัญญาณการยกหู และวางหูที่สร้างขึ้นใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

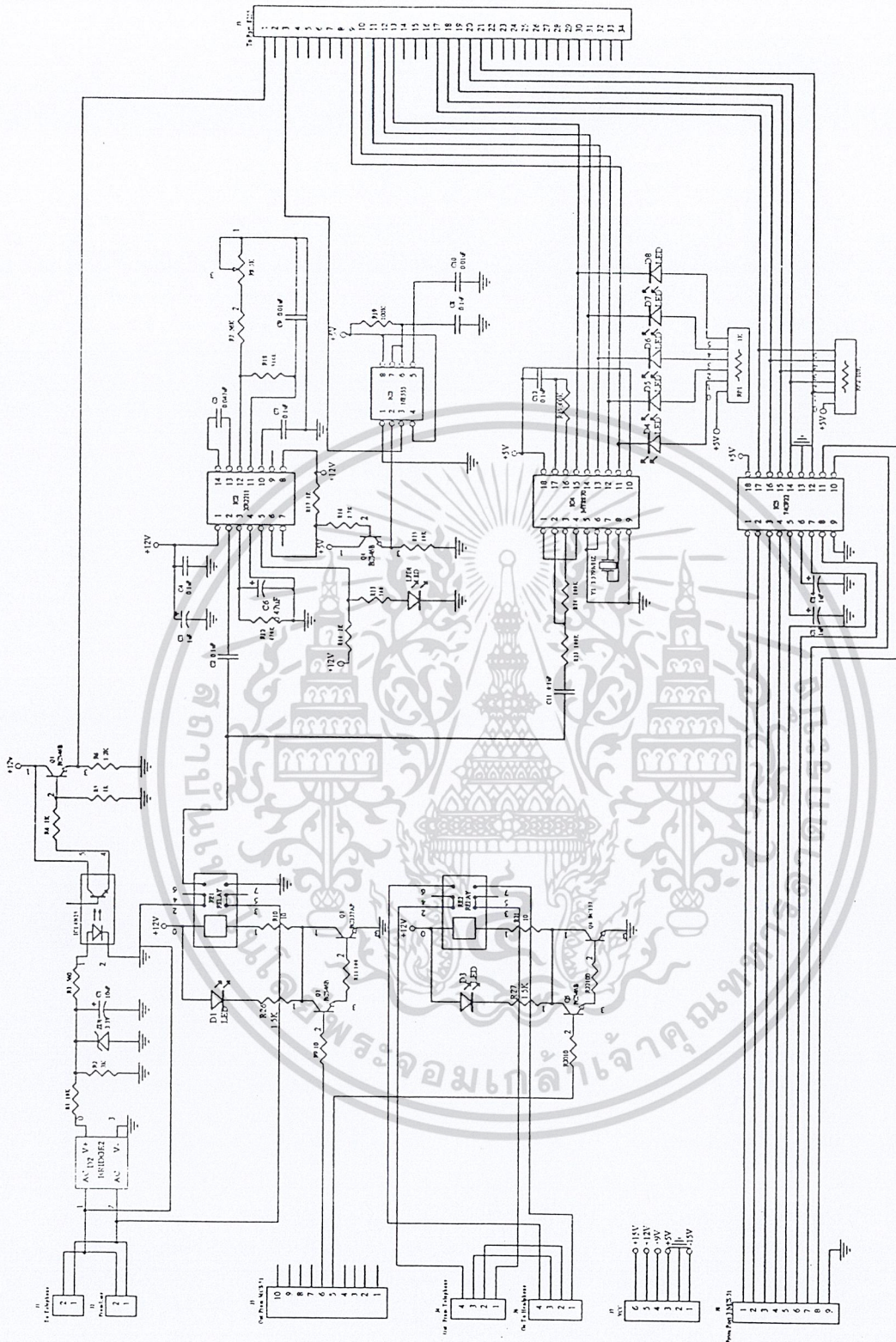
1. วรพงษ์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, **เรียนรู้ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบเฟิร์ม ฉบับ AT89C5x ของ Atmel**, บริษัทอินโนเวทีฟ เอ็กเพริเมนต์ จำกัด, กรุงเทพมหานคร
2. สัจจะ จรัสรุ่งรวีร, **คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.0**, สำนักพิมพ์อินโฟเพรส, กรุงเทพมหานคร
3. ศุภชัย สมพานิช, **Database Programming กับ Visual Basic ฉบับมืออาชีพ**, สำนักพิมพ์อินโฟเพรส, กรุงเทพมหานคร
4. สุชิน จำจด, **วิศวกรรมโทรศัพท์**, สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง, พิมพ์ครั้งที่ 3, กรุงเทพมหานคร, พ.ศ. 2526
5. [http:// www.atmel.com](http://www.atmel.com)
6. [http:// www.exar.com](http://www.exar.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

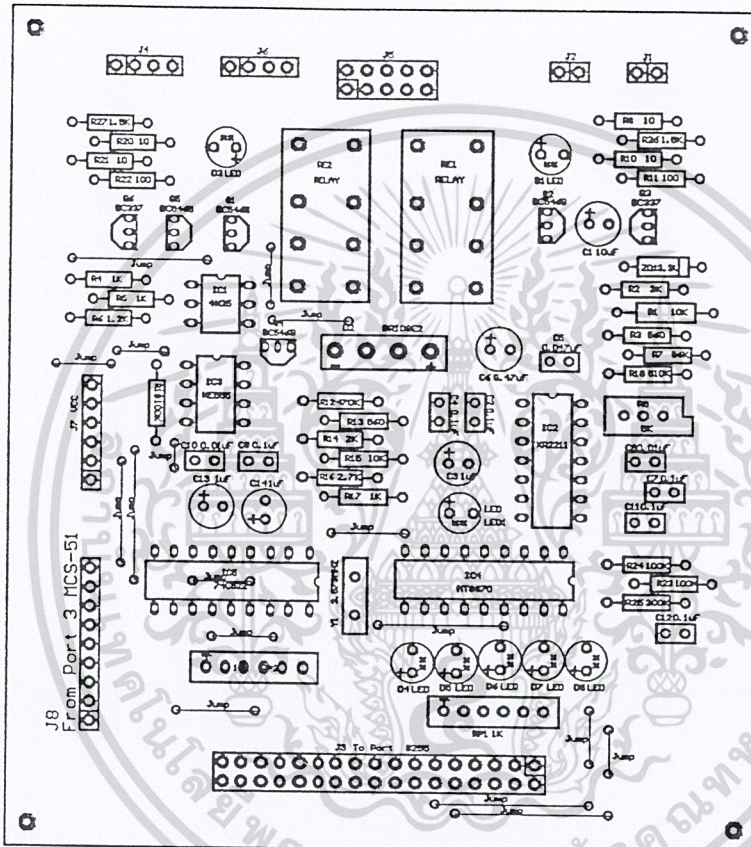


ภาคผนวก ก
วงจรรวมและแผ่นปรีน

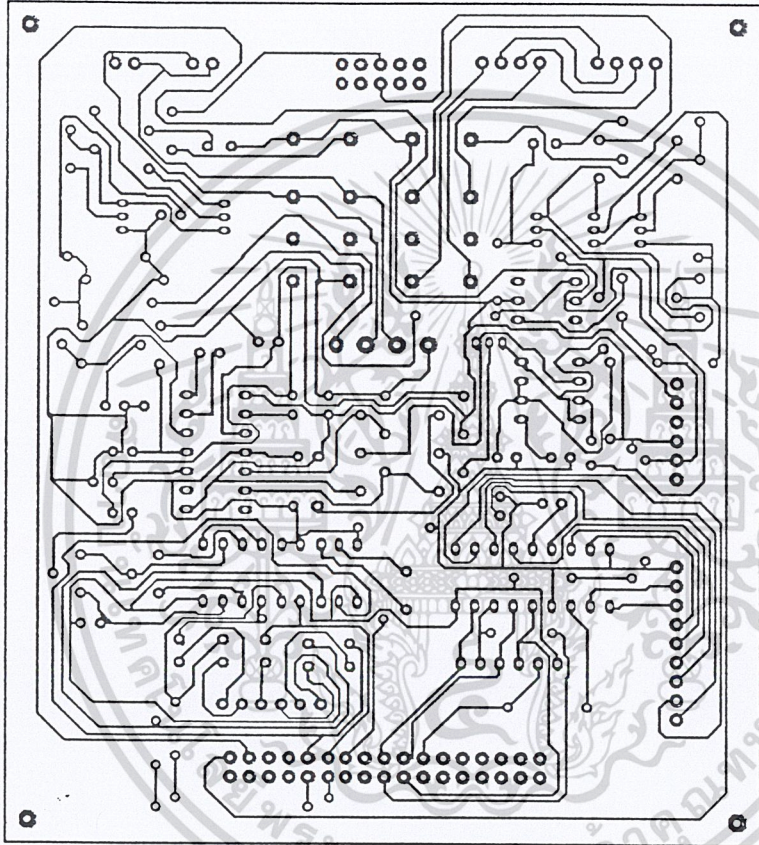
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



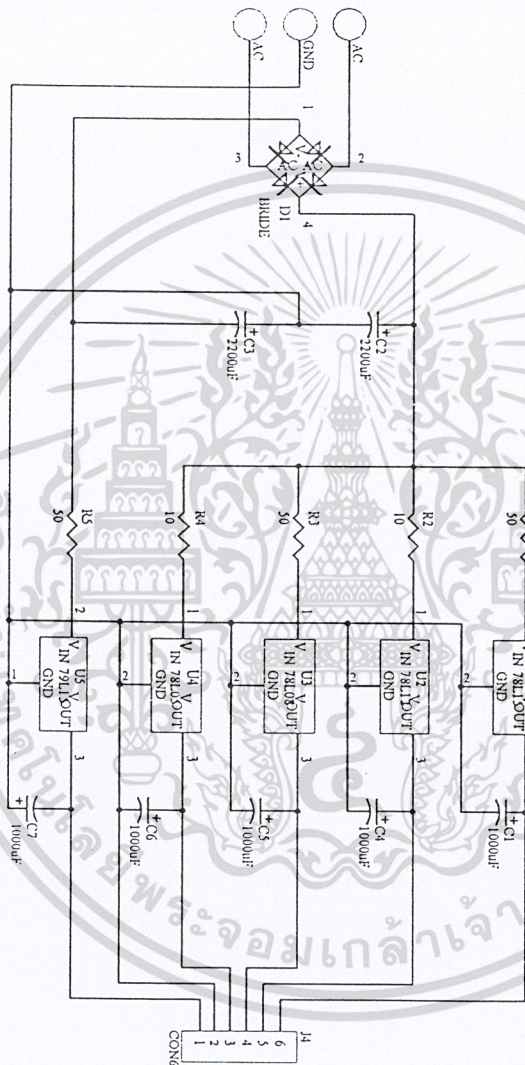
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



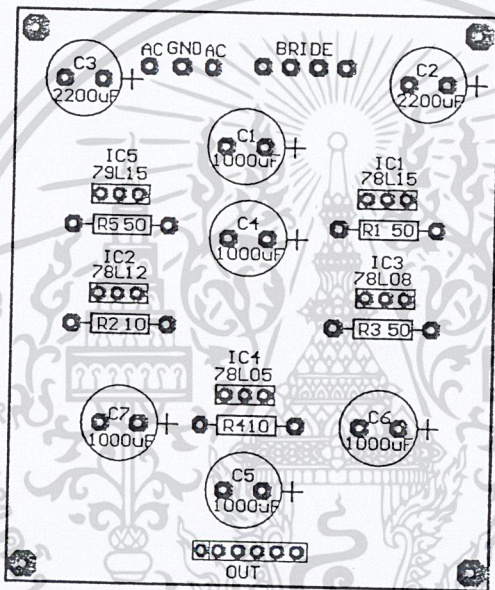
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



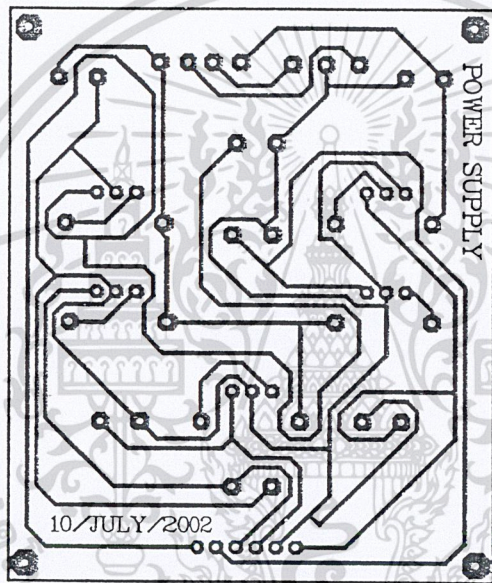
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข

COURSE CODE ของ VISUAL BASIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

From1
Option Explicit
Dim Conn As New ADODB.Connection
Dim Rs1 As New ADODB.Recordset
Dim Rs2 As New ADODB.Recordset
Dim temp As Variant
Private Sub Cmd1_Click()
Dim temp_str, st, str1, str2, str3, str4, num_reg As Variant
Dim time_b, time_em, time_u, time_e, time_eh, rg_code, rg_sub_code, check As String

temp_str = temp
If Rs1.State = adStateOpen Then
Rs1.Close
End If
Rs1.CursorLocation = adUseClient
Rs1.Open "delete from record", Conn, adOpenKeyset, , adCmdText
While Len(temp_str) >= 33
Minute (CDate("14:50") - CDate("12:30"))
st = Mid(temp_str, 1, 33)
str1 = Mid(st, 4, 9)
str2 = Mid(st, 14, 5)
str3 = Mid(st, 20, 5)
str4 = Mid(st, 26, 8)
rg_sub_code = Mid(str1, 4, 1)
temp_str = Mid(temp_str, 34, Len(temp_str))
*****
Debug.Print "st = " & st
Debug.Print "str1 = " & str1
Debug.Print "str2 = " & str2
Debug.Print "str3 = " & str3
Debug.Print "str4 = " & str4
Debug.Print "rg_sub_code = " & rg_sub_code
Debug.Print "temp_str = " & temp_str
*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If

Rs1.CursorLocation = adUseClient

Rs1.Open "record". Conn, adOpenKeyset, adLockOptimistic, adCmdTable

With Rs1
    .AddNew

    num_reg = Mid(str1, 1, 2)

    Debug.Print "num_reg = " & num_reg

    'If num_reg = "01" Or num_reg = "02" Or num_reg = "09" Or num_reg = "06" Or num_reg =
"07" Then

    If num_reg = "02" Then
        !num_region = num_reg
        rg_code = num_reg
    Else
        If num_reg = "01" Or num_reg = "09" Or num_reg = "06" Then
            rg_code = num_reg
            Debug.Print rg_code
        Else
            If num_reg = "07" Then
                check = Trim(Str(Mid(str1, 3, 1)))
                Debug.Print check

                If check = "3" Or check = "4" Or check = "5" Or check = "6" Or check = "7" Then
                    !num_region = Trim(Str(Mid(str1, 1, 3)))
                    rg_code = Trim(Str(Mid(str1, 1, 3)))
                Else
                    rg_code = "07"
                End If
            End If
        End If
    End If

    !num_region = Trim(Str(Mid(str1, 1, 3)))
    rg_code = Trim(Str(Mid(str1, 1, 3)))
    Debug.Print "rg_code = " & rg_code

End If
End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 time_b = Trim(Mid(str2, 1, 2)) & "." & Trim(Mid(str2, 4, 2))
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อแหล่งอื่นและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

!time_begin = time_b
time_u = Mid(str4, 1, 2) * 60 + Int(Val(Mid(str4, 4, 5)) * -1) * -1
time_em = Val(Mid(time_b, 4, 2)) + time_u
If Val(time_em) > 59 Then
    time_em = time_em Mod 60
End If
If Val(time_em) < 10 Then
    time_em = "0" & Trim(Str(time_em))
End If
If Val(time_u) > 59 Then
    time_eh = Val(Mid(time_b, 1, 2)) + (time_u / 60)
    If Val(time_eh) > 23 Then
        time_eh = Val(time_eh) Mod 24
        If time_eh <= 10 Then
            time_eh = "0" & Trim(Str(time_eh))
        End If
    End If
    time_e = Str(time_eh) & ":" & Trim(Str(time_em))
Else
    time_e = Str(Val(Mid(time_b, 1, 2))) & ":" & Trim(Str(time_em))
End If
!date_use = str3 & "/" & Trim(Str(Year(Date)))
!time_use = time_u
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select price from tele_price where (region_code = '0' & Trim(rg_code) & '') &
"And (sub_code=' " & Trim(rg_sub_code) & "')", Conn, adOpenKeyset, , adCmdText
If Rs2.RecordCount <> 0 Then
    !price_use = Val(time_u) * Rs2!price
Else
    If rg_code = "02" Then
        !price_use = 3
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

!price_use = Val(time_u) * 3      'mobile
End If

End If

Rs2.Close 'add

Rs2.CursorLocation = adUseClient 'add

If rg_code = "02" Then
    !city = "กรุงเทพ"
Else
    If rg_code = "01" Or rg_code = "06" Or rg_code = "09" Or rg_code = "07" Then
        !city = "มอญี่อ"
    Else

        Debug.Print rg_code

        Rs2.Open "select region_name from tele_price where (region_code = '0' & Trim(rg_code) &
        '')" & "And (sub_code=' " & Trim(rg_sub_code) & "')", Conn, adOpenKeyset, , adCmdText

        If Rs2.RecordCount <> 0 Then
            !city = Rs2!region_name
        Else

        End If
    End If
End If

.Update
End With

Wend

If Rs1.State = adStateOpen Then
    Rs1.Close

End If

Rs1.Open "select num_tele,city,time_begin,date_use,time_use,price_use from record", Conn,
adOpenKeyset, , adCmdText

Set DataGrid1.DataSource = Rs1

DataGrid1.Columns.Item(0).Width = 1500
DataGrid1.Columns.Item(3).Width = 1500
DataGrid1.Columns.Item(4).Width = 1500

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 DataGrid1.Columns.Item(1).Caption = "จังหวัด" 'add
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DataGrid1.Columns.Item(2).Caption = "เวลาที่โทรออก"
DataGrid1.Columns.Item(3).Caption = "วันที่โทรออก"
DataGrid1.Columns.Item(4).Caption = "จำนวนนาทีที่ใช้"
DataGrid1.Columns.Item(5).Caption = "จำนวนเงิน"
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select sum(price_use) as sp from record", Conn, adOpenKeyset, , adCmdText
Label3 = Str(Rs2!sp) & " บาท"
cmdprint.Enabled = True
cmdrec.Enabled = True
cmdexit.Enabled = True
End Sub

```

```

Private Sub cmdrec_Click()
Dim lfile As Integer
cmdexit.Enabled = False
cmdprint.Enabled = False
cmdrec.Enabled = False
lfile = FreeFile
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select num_tele,time_begin,date_use,time_use,price_use,city from record", Conn,
adOpenKeyset, , adCmdText
On Error GoTo Error1
If Rs2.RecordCount <> 0 And Trim(Label3) <> "0 บาท" Then
Rs2.MoveFirst
Open "A:\tele.txt" For Output As #lfile
Print #lfile, "          รายการข้อมูลการใช้โทรศัพท์"
Print #lfile,

```

```

Print #lfile, "เบอร์ที่โทร"; Tab: " เวลาที่โทร "; Tab: "วันที่ใช้"; Tab: " นาที"; Tab: "จำนวนเงิน"; Tab:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้า " จังหวัด "

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Print #lfile,
Do
    Print #lfile, Rs2!num_tele; Tab; Rs2!time_begin; Tab; Rs2!date_use; Tab; Rs2!time_use;
    Tab; Rs2!price_use; Tab; Rs2!city
    Rs2.MoveNext
Loop Until Rs2.EOF = True
Print #lfile,
Print #lfile, "รวมเป็นเงิน" & Label3
Close #lfile
MsgBox "บันทึกลงแผ่นเรียบร้อยแล้ว", vbInformation, "Information"
cmdexit.Enabled = True
cmdrec.Enabled = True
cmdprint.Enabled = True
Else
MsgBox "ไม่มีรายการให้บันทึกลงแผ่น", vbInformation, "Information"
'Call Form_Activate
'Call re
cmdexit.Enabled = True
cmdrec.Enabled = True
cmdprint.Enabled = True
End If
Exit Sub
Error1:
MsgBox "กรุณาตรวจสอบ Disket", vbExclamation, "Error!"
Call Form_Activate
Call re
cmdexit.Enabled = True
cmdrec.Enabled = True
cmdprint.Enabled = True
End Sub

```

```
Private Sub cmdprint_Click()
```

```
cmdexit.Enabled = False
```

```
cmdprint.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cmdrec.Enabled = False
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select num_tele,time_begin,date_use,time_use,price_use from record", Conn,
adOpenKeyset, , adCmdText
If Rs2.RecordCount <> 0 And Trim(Label3) <> "0 บาท" Then
    DataReport1.Show
Else
    MsgBox "ไม่มีรายการให้พิมพ์", vbInformation, "Information"
    'Call Form_Activate
    'Call re
    cmdexit.Enabled = True
    cmdrec.Enabled = True
    cmdprint.Enabled = True
End If

cmdexit.Enabled = True
    cmdrec.Enabled = True
    cmdprint.Enabled = True
End Sub

Private Sub cmdexit_Click()
Unload Me
Form3.Show
End Sub

Private Sub Form_Activate()
DataGrid1.Columns.Add (1)
DataGrid1.Columns.Add (1)
DataGrid1.Columns.Add (1)
DataGrid1.Columns.Add (1) ' add
DataGrid1.Columns.Item(0).Caption = "หมายเลขโทรศัพท์"
DataGrid1.Columns.Item(1).Caption = "จังหวัด" 'add

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DataGrid1.Columns.Item(2).Caption = "เวลาที่โทรออก"  
DataGrid1.Columns.Item(3).Caption = "วันที่โทรออก"  
DataGrid1.Columns.Item(4).Caption = "จำนวนนาที่ที่ใช้"  
DataGrid1.Columns.Item(5).Caption = "จำนวนเงิน"  
Label3 = "0 บาท"
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
If Conn.State = adStateOpen Then
```

```
Conn.Close
```

```
End If
```

```
Conn.Open "DRIVER=Microsoft Access Driver (*.mdb);UID=admin;DBQ=" & App.Path &  
"tele_count.mdb"
```

```
MSComm1.PortOpen = True
```

```
cmdprint.Enabled = False
```

```
cmdrec.Enabled = False
```

```
'cmdexit.Enabled = False
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
temp = (MSComm1.Input)
```

```
'temp = "$003_~~073473161 00:49 27/10 00.10.02_~~042426956 00:50 27/10
```

```
00.05.02_~~042372222 12:51 27/10 00.28.30_~~078646732 10:00 27/10 00.20.00_~~098646732
```

```
00:51 28/10 00.10.00_~~075246732 00:51 30/10 01.00.30_~~077346732 01:51 30/10
```

```
00.15.00_~~077546732 00:51 27/10 00.28.30_~~077846732 00:51 27/10 00.28.30_~~073246732
```

```
00:51 27/10 00.28.30_~~028646732 00:51 27/10 00.28.30_~~074646732 00:51 27/10
```

```
00.28.30_~~075646732 00:51 27/10 00.28.30_~~028646732 00:51 27/10 00.28.30_~~042746732
```

```
00:51 27/10 00.28.30_~~039346732 00:51 27/10 00.28.30_~~039546732 00:51 27/10
```

```
00.28.30_~~032446732 00:51 27/10 00.28.30_~~"
```

```
If Trim(temp) <> "" Then
```

```
temp = Mid(temp, 5, Len(Trim(temp)))
```

```
Call Cmd1_Click
```

```
Timer1.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
End Sub
Private Sub re()
If Rs1.State = adStateOpen Then
    Rs1.Close
End If
Rs1.Open "select num_tele,city,time_begin,date_use,time_use,price_use from record", Conn,
adOpenKeyset, , adCmdText
Set DataGrid1.DataSource = Rs1
DataGrid1.Columns.Item(0).Width = 1500
DataGrid1.Columns.Item(3).Width = 1500
DataGrid1.Columns.Item(4).Width = 1500
DataGrid1.Columns.Item(0).Caption = "หมายเลขโทรศัพท์"
DataGrid1.Columns.Item(1).Caption = "จังหวัด" 'add
DataGrid1.Columns.Item(2).Caption = "เวลาที่โทรออก"
DataGrid1.Columns.Item(3).Caption = "วันที่โทรออก"
DataGrid1.Columns.Item(4).Caption = "จำนวนนาที่ที่ใช้"
DataGrid1.Columns.Item(5).Caption = "จำนวนเงิน"
If Rs2.State = adStateOpen Then
    Rs2.Close
End If
Rs2.CursorLocation = adUseClient
Rs2.Open "select sum(price_use) as sp from record", Conn, adOpenKeyset, , adCmdText
Label3 = Str(Rs2!sp) & " บาท"
cmdprint.Enabled = True
cmdrec.Enabled = True
cmdexit.Enabled = True
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

From 2

```
Private Sub Command1_Click()
```

```
If Text1 <> "" Then
```

```
    Adodc1.Recordset.MoveFirst
```

```
    Adodc1.Recordset.Find "region_code = " & Text1 & """, , adSearchForward
```

```
    If Adodc1.Recordset.EOF Then
```

```
        MsgBox "ไม่พบข้อมูลที่ต้องการค้นหา", vbInformation, "Information"
```

```
        Adodc1.Recordset.MoveFirst
```

```
        Text1.SetFocus
```

```
        Text1 = ""
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
If Text2 <> "" Then
```

```
    Adodc1.Recordset.MoveFirst
```

```
    Adodc1.Recordset.Find "region_name = " & Text2 & """, , adSearchForward
```

```
    If Adodc1.Recordset.EOF Then
```

```
        MsgBox "ไม่พบข้อมูลที่ต้องการค้นหา", vbInformation, "Information"
```

```
        Adodc1.Recordset.MoveFirst
```

```
        Text2.SetFocus
```

```
        Text2 = ""
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    Unload Me
```

```
    Form3.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text1 = ""

Text2 = ""

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

From3

Private Sub Command1_Click()

Form1.Show

Unload Me

End Sub

Private Sub Command2_Click()

Form2.Show

Unload Me

End Sub

Private Sub Command3_Click()

End

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CS_LCD: BIT P1.3 ; P1.3;E LCD (PIN INT1)

RS_LCD: BIT P1.2 ; P1.2;RS LCD (Pin T0)

SCL BIT P1.0

SDA BIT P1.1

ADD_LCD: EQU 024H ; ADDRESS LCD

CHARAC: EQU 025H ; CHARACTER

ADD_TAB: EQU 026H ; ADDRESS TABLE

ADD_SEN: EQU 027H ; ADDRESS SENTENC

R8: EQU 028H ; DEALY

R9: EQU 029H ; DELAY

RA: EQU 02AH ; DELAY

ADD_LOW: EQU 02BH ;

DATA: EQU 02CH

HOUR: EQU 02EH ; START TO ADDRESS SAVE TIME

MIN: EQU 02FH

SEC: EQU 030H

DAY: EQU 031H

DATE: EQU 032H

MONTH: EQU 033H

YEAR: EQU 034H ; 7 ADDRESS TO SAVE TIME

KEY: EQU 035H ; BACK UP KEY

KEY_PAD: EQU 036H ; KAY DATA IN ASC

SENT_CON: EQU 037H; CONTORL SELECT

HOUR_S: EQU 038H ; SAVE SETING TIME

MIN_S: EQU 039H

SEC_S: EQU 03AH

DAY_S: EQU 03BH

DATE_S: EQU 03CH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MONTH_S: EQU 03DH
YEAR_S: EQU 03EH ; 7 ADDRESS TO SETING TIME
BACK_UP: EQU 03FH ; COUNTER SETING
DTMF: EQU 040H ;
POINTER: EQU 041H ; POINTER MEMORY
USING_HOUR: EQU 042H ; KEEP USING TIME TO USE TELEPNOE
USING_MIN: EQU 043H
USING_SEC: EQU 044H
L_8870: BIT P3.5 ; P3.4 LINE CONNECT 8870 ACTIVE "1"
L_HAND: BIT P3.4 ; P3.5 HAND CONNECT CIRCUIT
PORT_A EQU 8000H ; 8255 PORT A TELEPHONE
STATUS
PORT_B EQU 8001H ; 8255 PORT B DTMF
CONNECT
PORT_C EQU 8002H ; 8255 PORT C KEY PAD
CONNECT
ADD_MEM: EQU 4000D ; BACK UP TOTAL TELE

CONT_BYTE_W EQU 0D0H
CONT_BYTE_R EQU 0D1H

; PA.0 EQU HAND CHECK , PA.0=0 HAND_UP
; PA.0=1 HAND DOWN

; PA.2 EQU RING BACK TONE CHECK
; P3.4 EQU HAND_CONNECT ; RELAY ACTIVE 1
; P3.5 EQU LINE_CONNECT ; RELAY ACTIVE 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                ORG    0000H

;Serial port set
INIT_SER:      MOV    TMOD,#00100000B    ;time1 mode2
                MOV    SCON,#01010000B    ;mode1 serial port
                MOV    TH1,#0FBH
                MOV    A,#00H
                MOV    PCON,A            ;SMOD = 0
                CLR    ET1                ;clear timer1 interrupt
                SETB   TRI                ;start timer1
                CLR    ES
                CLR    EA
CONTROL:        MOV    SP,#256-32
                MOV    P3,#0FFH
                SETB   CS_LCD            ;initial LCD
                SETB   SCL
                SETB   SDA
                ACALL  DELAY              ;initial delay
                ACALL  INIT_LCD
                ACALL  DELAY
                ACALL  CLR_LCD
                MOV    DPTR,#8003H        ;CONTROL PROT
                MOV    A,#9BH
                MOVX   @DPTR,A
                ACALL  DELAY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

START:          ACALL CLR_LCD
                CLR    L_8870                ;DISCONNECT LINE
TO 8870
                SETB   L_HAND                ;CONNECT HAND

                MOV    DPTR,#TAB_DATE
                ACALL  SHOW_L1
                MOV    DPTR,#TAB_TIME        ;TIME
                ACALL  SHOW_L2
                ACALL  KEY_UP
START_1:        MOV    ADD_LCD,#06H
                ACALL  SHOW_DATE
                MOV    ADD_LCD,#46H
                ACALL  SHOW_TIME
;*****
                MOV    DPTR,#PORT_C         ;CHECK KEY ACTIVE
                MOVX   A,@DPTR
                ANL   A,#00011111B
                CJNE  A,KEY,KEY_ACTIVE
;*****

                MOV    DPTR,#PORT_A         ;CHECK HAND UP
                MOVX   A,@DPTR              ;HAND UP ACC.0 = 0
                JNB   ACC.0,HAND_UP         ;HAND DOWN ACC.0 = 1
                AJMP  START_1
KEY_ACTIVE:    ACALL  KEY_DATA
                MOV    A,KEY
                ANL   A,#0FH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#TAB_KEY
MOVC A,@A+DPTR
CJNE A,#"E",START_1
AJMP MANU_EDIT

```

```

;***** HAND UP *****
;

```

```

HAND_UP: ACALL CLR_LCD
          SETB L_8870 ;CONNECT

```

```

LINE TO 8870

```

```

MOV R0,#POINTER
MOV R1,#09H
MOV DPTR,#TAB_USE
ACALL SHOW_L1
MOV ADD_LCD,#40H

```

```

;*****
;

```

```

DTMF_CHECK: MOV DPTR,#PORT_A
             MOVX A,@DPTR
             JB ACC.0,START ;HAND DOWN CHECK

```

```

;*****
;

```

```

DTMF_DOWN: MOV DPTR,#PORT_B
            MOVX A,@DPTR
            MOV DTMF,A
            JB ACC.4,DTMF_ACTIVE ;CHECK DTMF DOWN
            AJMP DTMF_CHECK

```

```

;*****
;

```

```

DTMF_ACTIVE: ACALL BUSY
             MOV DPTR,#PORT_B
             MOVX A,@DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      DTMF,A
ANL  A,#00FH
CJNE    A,#010D,DTMF_A1
MOV      A,#00H
DTMF_A1: MOV  @R0,A
          ORL  A,#00110000B
          ACALL SHOW_1C
          INC  ADD_LCD
          INC  R0
;*****
DTMF_UP: MOV  DPTR,#PORT_B
          MOVX A,@DPTR
          JB  ACC.4,DTMF_UP ;CHECK DTMF UP
          DJNZ R1,DTMF_CHECK ;CHECK 9 NUMBER
;***** CHECK RING BACK TONE *****
          ACALL DELAY
          ACALL CLR_LCD
          ACALL DELAY
          MOV  DPTR,#TAB_CALL
          ACALL SHOW_L1

;          AJMP MEMORY

          CLR  L_HAND
;*****
CHECK_TONE: MOV  DPTR,#8000H
LX1:      MOVX  A,@DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JB ACC.0,HAND_DOWN
        JNB ACC.2,LX1
        ACALL BUSY_R
LX2:    MOVX A,@DPTR
        JB ACC.0,HAND_DOWN
        JB ACC.2,LX2
CHECK_TONE_1: MOV R1,#200D
CHECK_TONE_2: MOV R2,#170D
CHECK_TONE_3: MOV R3,#28D
LOOP_CH1: MOVX A,@DPTR
        JB ACC.0,HAND_DOWN
        JB ACC.2,CHECK_TONE_1
        DJNZ R3,LOOP_CH1
        DJNZ R2,CHECK_TONE_3
        DJNZ R1,CHECK_TONE_2
        SETB L_HAND
ACTIVE: SETB L_HAND
        SETB P3.3
        ACALL DELAY
        AJMP MEMORY
HAND_DOWN:CLR P3.3
        ACALL DELAY
        SETB P3.3
        ACALL DELAY
        CLR P3.3
        ACALL DELAY
        SETB P3.3
        ACALL DELAY
        AJMP START

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;***** GET DATA TO MEMORY *****

```
MEMORY:      MOV    R1,#09H
              MOV    R0,#POINTER
              MOV    DPTR,#ADD_MEM
              MOVX   A,@DPTR
              MOV    B,#16D
              MUL   AB
              MOV   DPL,A
              MOV   DPH,B
MEM_1:      MOV    A,@R0
MEM_2:      MOVX   @DPTR,A
              INC   R0
              INC   DPTR
              DJNZ  R1,MEM_1
              MOV   R0,#POINTER      ;KEEP FIRST USED TIME
              ACALL RTC_READ
              MOV   A,HOUR
              MOV   @R0,HOUR
              MOVX  @DPTR,A
              INC   R0
              INC   DPTR
              MOV   A,MIN
              MOV   @R0,MIN
              MOVX  @DPTR,A
              INC   R0
              MOV   @R0,SEC
              INC   R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC DPTR
MOV A,DATE
MOV @R0,DATE
MOVX @DPTR,A
INC R0
INC DPTR
MOV A,MONTH
MOV @R0,MONTH
MOVX @DPTR,A
INC R0
INC DPTR ;KEEP ADDRESS
MOV R6,DPL
MOV R7,DPH
;CHECK TIME AND WAIT HAND DOWN
CHECK_XXX: MOV DPTR,#PORT_A ;CHECK HAND DOWN
MOVX A,@DPTR
JNB ACC.0,CHECK_XXX ;WAIT HAND DOWN TO
CHECK TIME
ACALL RTC_READ
CLR C
MOV DPL,R6
MOV DPH,R7

;CONVERT BCD TO HEX , FOR EASY TO CAL USING TIME

MOV R0,#POINTER ;KEEP USING TIME (BCD)
MOV A,@R0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL BCD_TO_HEX
MOV @R0,A
INC R0
MOV A,@R0
ACALL BCD_TO_HEX
MOV @R0,A
INC R0
MOV A,@R0
ACALL BCD_TO_HEX
MOV @R0,A
MOV A,HOUR
ACALL BCD_TO_HEX
MOV HOUR,A
MOV A,MIN
ACALL BCD_TO_HEX
MOV MIN,A
MOV A,SEC
ACALL BCD_TO_HEX
MOV SEC,A
;*** CAL USING SEC ***
CLR C
MOV A,@R0
MOV R1,A
MOV A,SEC
SUBB A,R1
JC NEW_ADD_S
MOV USING_SEC,A
AJMP NEXT_T1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NEW_ADD_S:  MOV  A,@R0          ;IF C SET
             MOV   R1,A
MOV  A,SEC   ;60 SEC = 1 MIN
       ADD   A,#60D
       CLR  C
       SUBB  A,R1
       MOV   USING_SEC,A
       DEC  MIN
;*** CAL USING MIN ***
NEXT_T1:  DEC  R0
          CLR  C
          MOV  A,@R0
          MOV  R1,A
          MOV  A,MIN
          SUBB A,R1
          JC   NEW_ADD_M
          MOV  USING_MIN,A
          AJMP NEXT_T2

```

```

NEW_ADD_M:  MOV  A,@R0          ;IF C SET
             MOV   R1,A
MOV  A,#60D   ;60 MIN = 1 HOUR
       ADD   A,MIN
       CLR  C
       SUBB  A,R1
       MOV   USING_MIN,A
       DEC  HOUR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*** CAL USING HOUR ***
NEXT_T2:  DEC    R0

          CLR    C

          MOV    A,@R0

          MOV    R1,A

          MOV    A,HOUR

          SUBB   A,R1

          JC     NEW_ADD_H

          MOV    USING_HOUR,A

          AJMP   USING_MEM

NEW_ADD_H: MOV    A,@R0      ;IF C SET

          MOV    R1,A

          MOV    A,#24D      ;24 HOUR = 1 DAY

          ADD    A,HOUR

          CLR    C

          SUBB   A,R1

          MOV    USING_HOUR,A

;*****

USING_MEM: MOV    A,USING_HOUR

          ACALL  HEX_TO_BCD

          MOVX   @DPTR,A

          INC   DPTR

          MOV    A,USING_MIN

          ACALL  HEX_TO_BCD

          MOVX   @DPTR,A

          INC   DPTR

          MOV    A,USING_SEC

          ACALL  HEX_TO_BCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX    @DPTR,A

;*****

MOV     DPTR,#ADD_MEM  ;KEEP VALUE TELE
MOVX   A,@DPTR
INC    A
MOVX   @DPTR,A
AJMP  START

;*****
;***** MAIN MANU EDIT *****
;*****
MANU_EDIT: MOV    SENT_CON,#01H
           ACALL CLR_LCD
           MOV    DPTR,#TAB_EDIT  ;1 EDIT
           ACALL SHOW_L1
           MOV    DPTR,#TAB_SHOW  ;2 CLEAR MEMORY
           ACALL SHOW_L2

;**** SELECT MANU ****

MOV    SENT_CON,#01H
MANU_0    ACALL KEY_DOWN
MANU_1:   CJNE  A,#"U",MANU_2
           AJMP  MANU_UP
MANU_2:   CJNE  A,#"D",MANU_3
           AJMP  MANU_DOWN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MANU_3:          CJNE  A,#"C",MANU_4
                  AJMP  START
MANU_4:          CJNE  A,#"E",MANU_5
                  ;AJMP  SELECT
MANU_5:          CJNE  A,#"1",MANU_6
                  AJMP  EDIT_RTC
MANU_6:          CJNE  A,#"2",MANU_7
                  AJMP  SHOW_NUMBER
MANU_7:          CJNE  A,#"3",MANU_8
                  AJMP  SENT_COM
MANU_8:          CJNE  A,#"4",MANU_9
                  AJMP  CLR_MEM
MANU_9:          CJNE  A,#"5",MANU_A
                  AJMP  START
MANU_A:          AJMP  MANU_0

```

```

;*****
;***** SHOW SELECT TO LCD *****
;*****

```

```

MANU_UP:        DEC  SENT_CON
                  MOV   A,SENT_CON
                  CJNE  A,#000H,MANU_LCD
                  MOV   SENT_CON,#005H
                  AJMP  MANU_LCD

```

```

MANU_DOWN:      INC   SENT_CON

```

```

MOV   A,SENT_CON

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                CJNE     A,#006H,MANU_LCD
                MOV      SENT_CON,#001H
                AJMP     MANU_LCD
MANU_LCD:      MOV      A,SENT_CON
                CJNE   A,#01H,M_LCD1
                AJMP     MANU_LCD1
M_LCD1:       CJNE   A,#02H,M_LCD2
                AJMP     MANU_LCD2
M_LCD2:       CJNE   A,#03H,M_LCD3
                AJMP     MANU_LCD3
M_LCD3:       CJNE   A,#04H,M_LCD4
                LJMP     MANU_LCD4
M_LCD4:       CJNE   A,#05H,M_LCD5
                LJMP     MANU_LCD5
M_LCD5:       AJMP     MANU_EDIT
MANU_LCD1:    ACALL   CLR_LCD
                MOV      DPTR,#TAB_EDIT
                ACALL   SHOW_L1
                MOV      DPTR,#TAB_SHOW
                ACALL   SHOW_L2
                ACALL   KEY_UP
                AJMP     MANU_0

```

```
MANU_LCD2:    ACALL   CLR_LCD
```

```

                MOV      DPTR,#TAB_SHOW
                ACALL   SHOW_L1
                MOV      DPTR,#TAB_COM
                ACALL   SHOW_L2
                ACALL   KEY_UP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AJMP MANU_0

MANU_LCD3: ACALL CLR_LCD

MOV DPTR,#TAB_COM

ACALL SHOW_L1

MOV DPTR,#TAB_CLR

ACALL SHOW_L2

ACALL KEY_UP

AJMP MANU_0

MANU_LCD4: ACALL CLR_LCD

MOV DPTR,#TAB_CLR

ACALL SHOW_L1

MOV DPTR,#TAB_EXIT

ACALL SHOW_L2

ACALL KEY_UP

AJMP MANU_0

MANU_LCD5: ACALL CLR_LCD

MOV DPTR,#TAB_EXIT

ACALL SHOW_L1

MOV DPTR,#TAB_EDIT

ACALL SHOW_L2

ACALL KEY_UP

AJMP MANU_0

***** CLEAR MEMORY *****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLR_MEM:    MOV     DPTR,#ADD_MEM
            MOV     A,#00H
            MOVX   @DPTR,A
            MOV     DPTR,#0000H
            MOV     A,#0FFH
            MOV     R0,#20D

```

```

CLR_M2:     MOV     R1,#200D

```

```

CLR_M1:     MOVX   @DPTR,A

```

```

            INC     DPTR
            DJNZ   R1,CLR_M1
            DJNZ   R0,CLR_M2
            ACALL  CLR_LCD
            MOV     DPTR,#TAB_COMPLE
            ACALL  SHOW_L1
            ACALL  DELAY_2
            AJMP   CONTROL

```

```

;*****
;***** SHOW DATA TO LCD *****
;*****

```

```

SHOW_NUMBER: MOV     POINTER,#00H

```

```

            AJMP   SHOW_TELE

```

```

SENT_COM1   ACALL   KEY_DOWN

```

```

            MOV   A,KEY_PAD

```

```

            CJNE A,#"U",SENT_C1

```

```

            AJMP   UP_TELE

```

```

SENT_C1:    CJNE A,#"D",SENT_C2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                AJMP      DOWN_TELE
SENT_C2:                        CJNE A,#"C",SENT_COM1
                                AJMP      START
DOWN_TELE:                       INC      POINTER
                                AJMP      SHOW_TELE
UP_TELE:                          DEC      POINTER
                                MOV      A,POINTER
                                CJNE A,#0FFH,DOWN_T1
                                MOV      POINTER,#00H
DOWN_T1:                          AJMP      SHOW_TELE
SHOW_TELE:                       ACALL     CLR_LCD
                                MOV      ADD_LCD,#00H
                                MOV      A,#"("
                                ACALL     SHOW_1C
                                INC      ADD_LCD
                                MOV      A,POINTER
                                INC      A
                                ACALL     H_B_TO_LCD      ;HEX TO BCD AND
SHOW LCD
                                INC      ADD_LCD
                                MOV      A,#")"
                                ACALL     SHOW_1C
;SHOW NUMBER
                                MOV      ADD_LCD,#07H
                                MOV      R1,#09D
                                MOV      A,POINTER
                                MOV      B,#16D
                                MUL      AB
                                MOV      DPL,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPH,B

DATA_LCD_1: MOVX    A,@DPTR
              ORL   A,#00110000B
              ACALL SHOW_1C
              INC   DPTR
              INC   ADD_LCD
              DJNZ  R1,DATA_LCD_1

;SHOW TIME
MOV ADD_LCD,#43H
MOVX  A,@DPTR ;SHOW HOUR
ACALL ASC_LCD
INC   ADD_LCD
MOV  A,#":"
ACALL SHOW_1C
INC  DPTR ;SHOW MIN
INC  ADD_LCD
MOVX  A,@DPTR
ACALL ASC_LCD
INC  ADD_LCD ;SHOW DATE
INC  ADD_LCD
INC  DPTR

MOVX  A,@DPTR
ACALL ASC_LCD
INC  ADD_LCD
MOV  A,#"/"
ACALL SHOW_1C
INC  ADD_LCD ;SHOW MONTH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC DPTR

MOVX A,@DPTR

ACALL ASC_LCD

INC DPTR ;KEEP USED HOUR

INC DPTR ;KEEP USED MIN

INC DPTR ;KEEP USED SEC

ACALL KEY_UP

AJMP SENT_COM1

;*****
;***** SENT DATA TO COMPUTER BY 9600,N,8,1 *****
;***** SERIAL PORT COMMUNICATION *****
;*****
SENT_COM: MOV A,#"$" ;START WORD
          ACALL TX_BYTE

;*****

SENT_TOTEL: MOV DPTR,#ADD_MEM ;SENT VALUE TELE
            MOVX A,@DPTR
            MOV B,#10D ; CONVERT HEX TO BCD
            DIV AB ; AND SHOW LCD
            MOV R7,B
            MOV B,#10D
            DIV AB
            MOV R6,B
            MOV R5,A
            MOV A,R5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ORL A,#00110000B
        ACALL TX_BYTE
        MOV A,R6
        ORL A,#00110000B
        ACALL TX_BYTE
        MOV A,R7
        ORL A,#00110000B
        ACALL TX_BYTE
        MOV A,#"_"
        ACALL TX_BYTE
        ;*****
        MOV POINTER,#00H
        MOV B,#16D
        MOV DPTR,#ADD_MEM
        MOVX A,@DPTR
        MOV R7,A ;COUNTER NUMBER VALUE
        ; MOV R7,#200D
SENT_COM_3: MOV R1,#09D
        MOV B,#16D
        MOV A,POINTER
        MUL AB
        MOV DPL,A
        MOV DPH,B
SENT_COM_1: MOVX A,@DPTR ;9 NUMBER
        ORL A,#00110000B
        ACALL TX_BYTE
        INC DPTR
        DJNZ R1,SENT_COM_1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#"-";CONNECT
ACALL TX_BYTE
;*****
MOVX A,@DPTR ;SENT HOUR
ACALL BCD_ASC_RS
MOV A,#":"
ACALL TX_BYTE
INC DPTR ;SENT MIN
MOVX A,@DPTR
ACALL BCD_ASC_RS
MOV A,#"-";CONNECT
ACALL TX_BYTE
;*****
INC DPTR ;SENT DATE
MOVX A,@DPTR
ACALL BCD_ASC_RS
MOV A,#"/"
ACALL TX_BYTE
INC DPTR ;SENT MONTH
MOVX A,@DPTR
ACALL BCD_ASC_RS
;*****
MOV A,#"-";CONNECT
ACALL TX_BYTE
INC DPTR ;SENT USED HOUR
MOVX A,@DPTR
ACALL BCD_ASC_RS
MOV A,#"."
ACALL TX_BYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC DPTR ;SENT USED MIN
MOVX A,@DPTR
ACALL BCD_ASC_RS
MOV A,#"."
ACALL TX_BYTE
INC DPTR ;SENT USED SEC
MOVX A,@DPTR
ACALL BCD_ASC_RS
MOV A,#"_"
ACALL TX_BYTE
INC POINTER
DJNZ R7,SENT_COM_3
MOV A,#"*" ;STOP WORD
ACALL TX_BYTE
SENT_COM_2: ACALL CLR_LCD
MOV DPTR,#TAB_COMPLE ;COMPLETE
ACALL SHOW_L1
ACALL DELAY_2
AJMP START

BCD_ASC_RS: PUSH ACC
SWAP A
ANL A,#00001111B
ORL A,#00110000B
ACALL TX_BYTE
POP ACC
ANL A,#00001111B
ORL A,#00110000B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                ACALL    TX_BYTE
                                RET
TX_BYTE:    PUSH    IE
            CLR    TI
            MOV    SBUF,A
            JNB   TI,$
            CLR    TI
            POP   IE
            RET

;*****
;***** EDIT RTC *****
;*****
EDIT_RTC:  ACALL   CLR_LCD
            ACALL   RTC_READ
            MOV    DPTR,#TAB_DATE_S ;DATE
            ACALL   SHOW_L1
            MOV    ADD_LCD,#06H
            ACALL   SHOW_DATE
            MOV    DPTR,#TAB_TIME   ;TIME
            ACALL   SHOW_L2
            MOV    ADD_LCD,#46H
            ACALL   SHOW_TIME

;***** EDIT SELECT *****

EDIT_A:    MOV    SENT_CON,#01H
EDIT_A1:   ACALL  KEY_DOWN
            MOV    A,KEY_PAD
            CJNE  A,#"U",E_R1 ;SELECT DATE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                AJMP     SELECT_UP
E_R1:          CJNE   A,#"D",E_R2   ;SELECT TIME
                AJMP     SELECT_DOWN
E_R2:          CJNE   A,#"E",E_R3   ;PRESS ENTER TO SETING
                AJMP     SETING
E_R3:          CJNE   A,#"C",EDIT_A  ;PRESS CLR TO RE MANU

```

```

                AJMP     START
SELECT_UP:     MOV     SENT_CON,#01H
                MOV     ADD_LCD,#05H
                MOV     A,#"("
                ACALL  SHOW_1C
                MOV     A,#")"
                MOV     ADD_LCD,#0EH
                ACALL  SHOW_1C
                MOV     ADD_LCD,#45H
                MOV     A,#" "
                ACALL  SHOW_1C
                MOV     A,#" "
                MOV     ADD_LCD,#4EH
                ACALL  SHOW_1C

```

```

                AJMP  EDIT_A1
SELECT_DOWN:   MOV     SENT_CON,#02H
                MOV     ADD_LCD,#45H
                MOV     A,#"("
                ACALL  SHOW_1C
                MOV     A,#")"
                MOV     ADD_LCD,#4EH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL SHOW_1C
MOV      ADD_LCD,#05H
MOV      A,#" "
ACALL SHOW_1C
MOV      A,#" "
MOV      ADD_LCD,#0EH
ACALL SHOW_1C
AJMP     EDIT_A1
SETTING: MOV      R1,#07H
MOV      R0,#HOUR_S
SET_X:   MOV      @R0,#00H
INC      R0
DJNZ     R1,SET_X
MOV      A,SENT_CON ;SELECT DATE OR TIME
CJNE     A,#01H,SET_1
AJMP     SET_DATE
SET_1:   CJNE     A,#02H,EDIT_A
AJMP     SET_TIME
;*****
SET_DATE: MOV      DPTR,#TAB_SET_D2 ;DATE (XX/XX/XX)
ACALL SHOW_L1
MOV      ADD_LCD,#06H ;SELECT DATE
MOV      R0,#DATE_S
AJMP     SET_DT
SET_TIME: MOV      DPTR,#TAB_SET_T2 ;TIME (XX:XX:XX)
ACALL SHOW_L2
MOV      ADD_LCD,#46H ;SELECT TIME
MOV      R0,#HOUR_S
SET_DT:  MOV      R1,#003H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SET_DT_1:    ACALL KEY_DOWN    ;MEMORY FIRST DIGIT
             MOV     A,KEY_PAD
             CJNE   A,#"C",SET_D11
             AJMP   EDIT_RTC

SET_D11:    CLR  C
            MOV   A,KEY_PAD
            SUBB  A,#00111010B
            JC   IN_NUMBER_1
            AJMP SET_DT_1

IN_NUMBER_1:  MOV   A,KEY_PAD
             ACALL SHOW_1C
             MOV   A,KEY_PAD
             ANL  A,#00FH
             SWAP A
             MOV   @R0,A
             INC  ADD_LCD
             ACALL KEY_UP
;***** ;MENMORY SECOND DIGIT
SET_DT_2:    ACALL KEY_DOWN
             MOV   A,KEY_PAD
             CJNE A,#"C",SET_D21
             AJMP EDIT_RTC

SET_D21:    CLR  C
            MOV   A,KEY_PAD
            SUBB  A,#00111010B
            JC   IN_NUMBER_2
            AJMP SET_DT_2

```

```
IN_NUMBER_2:  MOV   A,KEY_PAD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL SHOW_1C
MOV     A,KEY_PAD
ANL A,#00FH
ORL A,@R0
MOV     @R0,A
ACALL KEY_UP
INC  ADD_LCD

;*****
INC  ADD_LCD      ;NEXT DATA
INC  R0
DJNZ R1,SET_DT_1
;*****
WRITE_DATA:ACALL KEY_DOWN      ;PREE ENTER TO SET DATA
MOV     A,KEY_PAD      ;WRITE I2C
CJNE   A,#"C",WRITE_D11
AJMP   EDIT_RTC
WRITE_D11: CJNE   A,#"E",WRITE_DATA
AJMP   SELECT_WRITE

SELECT_WRITE: MOV   A,SENT_CON      ;SELECT DATE OR TIME
CJNE   A,#01H,SELECT_W1
AJMP   WRITE_DATE

SELECT_W1: CJNE   A,#02H,END_GET
AJMP   WRITE_TIME

WRITE_DATE: MOV   R1,#03H
MOV     ADD_LOW,#04H
MOV     R0,#DATE_S      ;START DATA

WRITE_DATE_1: MOV   DATA,@R0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL WRITE_BYTE
INC ADD_LOW
INC R0
DJNZ R1,WRITE_DATE_1
AJMP START

WRITE_TIME: MOV R1,#03H
MOV ADD_LOW,#02H
MOV R0,#HOUR_S ;START DATA
WRITE_TIME_1: MOV DATA,@R0
ACALL WRITE_BYTE
DEC ADD_LOW
INC R0
DJNZ R1,WRITE_TIME_1
END_GET: AJMP START

;***** SHOW TIME *****

SHOW_TIME: ACALL RTC_READ
MOV A,HOUR
ACALL ASC_LCD
INC ADD_LCD
MOV A,#":"
ACALL SHOW_1C
INC ADD_LCD
MOV A,MIN
ACALL ASC_LCD
INC ADD_LCD
MOV A,#":"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL SHOW_1C
INC     ADD_LCD
        MOV     A,SEC
        ACALL  ASC_LCD
RET

```

```

;***** SHOW DATE *****

```

```

SHOW_DATE: ACALL RTC_READ
            MOV     A,DATE
            ACALL  ASC_LCD
            INC     ADD_LCD
            MOV     A,#"/"
            ACALL  SHOW_1C
            INC     ADD_LCD
            MOV     A,MONTH
            ACALL  ASC_LCD
            INC     ADD_LCD
            MOV     A,#"/"
            ACALL  SHOW_1C
            INC     ADD_LCD
            MOV     A,YEAR
            ACALL  ASC_LCD
RET

```

```

;*****

```

```

;***** KEY CONNECT *****

```

```

;*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

KEY_DOWN:  ACALL KEY_DATA
           MOV     A,KEY
           ANL A,#00010000B
           JNB ACC.4,KEY_DOWN
           MOV     DPTR,#TAB_KEY
           MOV     A,KEY
           ANL A,#00FH
           MOVC   A,@A+DPTR
           MOV     KEY_PAD,A
           RET
KEY_UP:    ACALL KEY_DATA
           ANL A,#00010000B
           JB ACC.4,KEY_UP
           RET
KEY_DATA:  MOV     DPTR,#PORT_C
           MOVX   A,@DPTR
           ANL A,#00011111B
           MOV     KEY,A
           RET

```

```

;*****
;***** RTC CONNECT *****
;*****

```

```

RTC_READ: MOV  ADD_LOW,#00H
          LCALL READ_BYTE
          MOV  SEC,A
          MOV  ADD_LOW,#01H
          LCALL READ_BYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     MIN,A
MOV  ADD_LOW,#02H
LCALL READ_BYTE
MOV     HOUR,A
MOV  ADD_LOW,#03H
LCALL READ_BYTE
MOV     DAY,A
MOV  ADD_LOW,#04H
LCALL READ_BYTE
MOV     DATE,A
MOV  ADD_LOW,#05H
LCALL READ_BYTE
MOV     MONTH,A
MOV  ADD_LOW,#06H
LCALL READ_BYTE
MOV     YEAR,A
RET

```

***** DECODE AND GO TO LCD *****

```

ASC_LCD:  PUSH ACC
          SWAP A
          ANL A,#0FH
          LCALL HEXASC
          ACALL SHOW_1C
          POP ACC
          ANL A,#0FH
          LCALL HEXASC

```

```
INC ADD_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACALL SHOW_1C

RET

HEXASC: PUSH DPH

PUSH DPL

MOV DPTR,#HEXASC_TAB

MOVC A,@A+DPTR

POP DPL

POP DPH

RET

;INPUT A ; OUTPUT R5,R6,R7 = 100+10+1 ;USE SHOW VALUE TELE

H_B_TO_LCD: MOV B,#10D ; CONVERT HEX TO BCD

DIV AB ; AND SHOW LCD

MOV R7,B

MOV B,#10D

DIV AB

MOV R6,B

MOV R5,A

MOV A,R5

ORL A,#00110000B

ACALL SHOW_1C

INC ADD_LCD

MOV A,R6

ORL A,#00110000B

ACALL SHOW_1C

INC ADD_LCD

MOV A,R7

ORL A,#00110000B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACALL SHOW_1C

RET

;INPUT A ;OUT R6,R7 = 10+1 , A,R5>BCD ; USE IN SHOW USED TELE

HEX_TO_BCD: MOV B,#10D

DIV AB

MOV R7,B

MOV R6,A

MOV A,R6

SWAP A

ORL A,R7

MOV R5,A

RET

;INPUT A ; OUT PUT A

BCD_TO_HEX:MOV B,#10D

MOV R6,A

ANL A,#00FH

MOV R5,A

MOV A,R6

SWAP A

ANL A,#00FH

MUL AB

ADD A,R5

RET

;***** WRITE DATA OUTPUT A*****

WRITE BYTE: CLR SDA ;start bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLR SCL
MOV A,#CONT_BYTE_W ;send control byte
LCALL LOOP_WRITE
SETB SDA
SETB SCL
JB SDA,WRITE_BYTE ;loop until busy
CLR SCL
MOV A,ADD_LOW ;send address low
LCALL LOOP_WRITE
SETB SDA
SETB SCL
JB SDA,WRITE_BYTE ;loop until busy
CLR SCL
MOV A,DATA ;send data
LCALL LOOP_WRITE
SETB SDA
SETB SCL
JB SDA,WRITE_BYTE ;loop until busy
CLR SDA
CLR SCL
SETB SCL ;stop bit
SETB SDA
RET

```

;***** READ DATA INPUT A *****

```

READ_BYTE: CLR SDA ;start bit
CLR SCL

```

```

MOV A,#CONT_BYTE_W ;send control byte

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL LOOP_WRITE
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
MOV A,ADD_LOW ;send address low
LCALL LOOP_WRITE
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
SETB SCL
SETB SDA
CLR SDA ;start bit
CLR SCL
MOV A,#CONT_BYTE_R ;send control byte
LCALL LOOP_WRITE
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
LCALL LOOP_READ
SETB SDA
SETB SCL
CLR SCL
SETB SCL ;stop bit
SETB SDA
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;***** WRITE TO I2C *****

```
LOOP_WRITE: PUSH 02H
            MOV R2,#08H
LOOP_WRITE_1: RLC A
            MOV SDA,C
            ACALL BUSY_I2C
            SETB SCL
            ACALL BUSY_I2C
            CLR SCL
            DJNZ R2,LOOP_WRITE_1
            POP 02H
            RET
```

;***** READ TO I2C *****

```
LOOP_READ:  PUSH 02H
            MOV R2,#08H
LOOP_READ_1: SETB SCL
            ACALL BUSY_I2C
            MOV C,SDA
            ACALL BUSY_I2C
            CLR SCL
            RLC A
            DJNZ R2,LOOP_READ_1
            MOV DATA,A
            POP 02H
            RET
```

;*****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;***** LCD *****
;*****
SHOW_1C:   PUSH     ACC
           MOV  A,ADD_LCD
           SETB   ACC.7
           ACALL  WR_INS
           POP   ACC
           ACALL  WR_DATA
           RET
SHOW_L1:   MOV  ADD_LCD,#00H
           AJMP  SHOW_LCD
SHOW_L2:   MOV  ADD_LCD,#40H
SHOW_LCD:  MOV  CHARAC,#16D
           MOV  ADD_TAB,#00H
S_LCD_1:   MOV  A,ADD_LCD ;SET ADDRESS
           SETB ACC.7
           ACALL WR_INS
           MOV  A,ADD_TAB
           MOVC A,@A+DPTR
           CJNE A,#" ",S_LCD_2
           AJMP S_LCD_3 ;*****
S_LCD_2:   ACALL WR_DATA
           INC  ADD_TAB
           INC  ADD_LCD
           DJNZ CHARAC,S_LCD_1
S_LCD_3:   RET
;*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SHOW_GOING: MOV ADD_SEN,#00H+15D

SHOW_ROW_A: MOV     CHARAC,#16D
             PUSH   ADD_LCD
             MOV    ADD_LCD,#00H

S_R_B:  MOV A,ADD_LCD
        SETB  ACC.7
        ACALL WR_INS
        MOV A,#" "
        ACALL WR_DATA
        INC  ADD_LCD
        DJNZ CHARAC,S_R_B
        POP  ADD_LCD
        MOV  ADD_LCD,ADD_SEN
        MOV  ADD_TAB,#00H

S_ROW_1: MOV A,ADD_LCD ;SET ADDRESS
        SETB ACC.7
        ACALL WR_INS
        MOV A,ADD_TAB
        MOVC A,@A+DPTR
        CJNE A,#" ",S_ROW_2
        AJMP S_ROW_3 ;*****

S_ROW_2: ACALL WR_DATA
        INC  ADD_TAB
        INC  ADD_LCD
        MOV  A,ADD_LCD
        CJNE A,#16H,S_ROW_1

S_ROW_3: DEC  ADD_SEN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,ADD_SEN
CJNE A,#0FFH,SHOW_ROW_A
MOV ADD_SEN,#00H
INC DPTR
MOV A,#00H
MOVC A,@A+DPTR
CJNE A,#" ",SHOW_ROW_A
RET
;*****
WR_DATA: MOV B,A
ANL A,#0F0H
ORL A,#00001100B ;RS/CS HI
MOV R2,A
MOV A,P1
ANL A,#00000011B
ORL A,R2
MOV P1,A
ACALL EN_LCD
MOV A,B ;Low byte
SWAP A
ANL A,#0F0H
ORL A,#00001100B
MOV R2,A
MOV A,P1
ANL A,#00000011B
ORL A,R2
MOV P1,A
ACALL EN_LCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                RET
WR_INS:  MOV  B,A
          ANL  A,#0F0H
          ORL  A,#00001000B ; CS HI
          MOV  R2,A
          MOV  A,P1
          ANL  A,#00000011B ;else bit
          ORL  A,R2
          MOV  P1,A ; High byte
          ACALL EN_LCD
          MOV  A,B ; Low byte
          SWAP A
          ANL  A,#0F0H
          ORL  A,#00001000B ; CS HI
          MOV  R2,A
          MOV  A,P1
          ANL  A,#00000011B
          ORL  A,R2
          MOV  P1,A
          ACALL EN_LCD
                RET
INIT_LCD: CLR  RS_LCD
          MOV  A,#33H ; Set DL = 1 3-time
          ACALL WR_INS
          MOV  A,#32H ; Clear DL = 0 1-time
          ACALL WR_INS
          MOV  A,#28H ; Function set
          ACALL WR_INS ; DL=0 4Bit,N=1 2Line,F=0 5X7
          MOV  A,#0CH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL WR_INS      ; Entry display,cursor off,cursor not blink
MOV  A,#06H      ; Entry mode set
ACALL WR_INS      ; I/D=1 Increment,S=0 cursor shift
MOV  A,#01H      ; Clear display
ACALL WR_INS      ; Clear display,set DD RAM address=0
RET
CLR_LCD:  MOV  A,#01H
          ACALL WR_INS
          RET
EN_LCD:   CLR  CS_LCD      ; Enable LCD
          ACALL BUSY      ; Busy delay time
          SETB CS_LCD      ; Disable LCD
          RET
;***** DELAY *****
BUSY:     MOV  R8,#07H
BUSY1:    MOV  R9,#0FFH
          DJNZ R9,$
          DJNZ R8,BUSY1
          RET
BUSY_R:   MOV  R8,#050H
B_R1:    MOV  R9,#0FFH
          DJNZ R9,$
          DJNZ R8,B_R1
          RET
DELAY:    MOV  R8,#0FFH
DEL:      MOV  R9,#0FFH
          DJNZ R9,$
          DJNZ R8,DEL
          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DELAY_2:    MOV R8,#0AH
DEL2_1:    MOV R9,#0FFH
DEL2_2:    MOV RA,#0FFH
           DJNZ     RA,$
           DJNZ     R9,DEL2_2
           DJNZ     R8,DEL2_1

```

```
RET
```

```

BUSY_I2C:  MOV R8,#0FFH
           DJNZ     R8,$

```

```
RET
```

```

;*****
;***** TABLE *****
;*****

```

```

TAB_KEY:   DB "123U456D789?C0HE_"
HEXASC_TAB: DB "0123456789ABCDEF"
TABLE_1:   DB "Tele counter_"
TAB_EDIT:  DB "1,EDIT TIME_"
TAB_SHOW:  DB "2,SHOW NUMBER_"
TAB_COM:   DB "3,CONNECT COMTER_"
TAB_CLR:   DB "4,CLEAR MEMORY_"
TAB_EXIT:  DB "5, EXIT_"
TAB_NONE:  DB " NONE_"
TAB_DATE:  DB "DATE_"
TAB_TIME:  DB "TIME_"
TAB_DATE_S: DB "DATE ( )_"
TAB_SET_D2: DB "DATE (XX/XX/XX)_"
TAB_SET_T2: DB "TIME (XX:XX:XX)_"
TAB_USE:   DB " HAND UP_"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TAB_COMPLE: DB " COMPLETE_"
TAB_FREE: DB " FREE_"
TAB_BUSY: DB " BUSY_"
TAB_CALL: DB " CALLING_"
END
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้