สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบจ่ายยาในโรงพยาบาล

MEDICINE DISTRIBUTION SYSTEM IN THE HOSPITAL

นายกฤษณะ พงษ์สุวรรณ
นายฉัตรชัย เจริญโชติสังข์
นายสุทธา ปัตทวีคงคา

ปพ.
ก 2417
25AA

เลขหมู่.....
เลขทะเบียน 45861
วัน, เดือน, ปี 19 ก.พ. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
b11058043

MEDICINE DISTRIBUTION SYSTEM IN THE HOSPITAL

KRISSANA PHONGSUWAN

CHATCHAI JAROENCHOTSUNG

SUTTHA PATTAWEKONGKA

A THESIS SUBMITTED INPARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION

ENGINEERINGDEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ ระบบจ่ายยาในโรงพยาบาล

MEDICINE DISTRIBUTION SYSTEM IN THE HOSPITAL

นักศึกษาผู้จัดทำ นายกฤษณะ พงษ์สุวรรณ **รหัสนักศึกษา** 42015422

นายฉัตรชัย เจริญโชติสังข์ **รหัสนักศึกษา** 42015429

นายสุทธา ปัตทวีคงคา **รหัสนักศึกษา** 42015456

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2544

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
รศ.ดร.กิตติ ตีรเศรษฐ ผศ.ประภาษ อุคคกิมาพันธุ์ อ.วิชัย ตันติจริยางกูร	

วัน/เดือน/ปี ที่สอบ วันพุธที่ 15 พฤษภาคม พ.ศ. 2545

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ. ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบจ่ายยาในโรงพยาบาล
	MEDICINE DISTRIBUTION SYSTEM IN THE HOSPITAL
นักศึกษาผู้จัดทำ	นาย กฤษณะ พงษ์สุวรรณ
	นาย ฉัตรชัย เจริญโชติสังข์
	นาย สุทธา ปัตทวีคงคา
อาจารย์ที่ปรึกษา	รศ.ดร. กิตติ ตีรเศรษฐ
	ผศ. ประภาษ อุคคกิมาพันธุ์
	อ. วิชัย ตันติจริยางกูร
ปีการศึกษา	2544

บทคัดย่อ

จากการวิเคราะห์ระบบจ่ายยาในโรงพยาบาล พบปัญหาใน 3 ขั้นตอนได้แก่ ขั้นตอนการส่งใบสั่งยา ขั้นตอนการเรียกผู้ป่วยมารับยา และขั้นตอนการตรวจสอบการเรียกรับยา ซึ่งปริญญานิพนธ์ฉบับนี้ ได้นำเสนอวิธีการแก้ปัญหาดังกล่าว โดยจัดให้มีเครื่องจ่ายบัตรคิวในขั้นตอนการส่งใบสั่งยา เพื่อแสดงลำดับก่อน-หลัง และผู้ป่วยไม่ต้องเข้าแถวรอ อันเป็นการกีดขวางทางเดิน ในขั้นตอนการเรียกผู้ป่วยมารับยา จัดให้มีส่วนแสดงผล LED DISPLAY และ SOUND CALL เพื่อเสริมการเรียกให้มีประสิทธิผลยิ่งขึ้น และในขั้นตอนการตรวจสอบการเรียกรับยา จัดให้มีเครื่องตรวจสอบการเรียก เพื่อให้ผู้ป่วยสามารถตรวจสอบการเรียกได้ด้วยตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Medicine Distribution System in the Hospital
Authors	Mr.Krissana Phongsuwan
	Mr.Chatchai Jaroenchotsung
	Mr.Suttha Pattawekongka
Thesis Advisor	Assoc.Prof.Dr. Kitti Tirasesth
	Asst.Prof. Prapart Ukakimaparn
	Mr. Wichai Tunticharyangkul
Year	2001

ABSTRACT

The analytical results of medicine distribution system in the hospital indicate the problems 3 process sending the prescription, calling and check in the calling process. This thesis presents the solution to the problems by preparing the sequential box in sending the prescription process in order to prevent the sequential errors and not to stand waiting sequentially. In the calling process, there is a LED DISPLAY and SOUND CALL for the better efficiency. And for checking the calling process, there is a BARCODE READER for the patients to check their own calling.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความเมตตาจาก รองศาสตราจารย์ กิตติ ตีรเศรษฐ ผู้ช่วยศาสตราจารย์ประภาษ อุคคกิมาพันธุ์ และอ.วิชัย ตันติจริยางกูร ที่ได้ให้คำแนะนำ แก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆ ในการทำปริญญานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้ง และขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ประจำภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

และที่ลืมไปเสียมิได้คือ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ พี่ๆ และเพื่อนๆ ที่สนับสนุน ช่วยเหลือ และเป็นกำลังใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... **I**

บทคัดย่อภาษาอังกฤษ ..... **II**

กิตติกรรมประกาศ ..... **III**

สารบัญ ..... **IV**

สารบัญตาราง ..... **IX**

สารบัญภาพ ..... **X**

บทที่ 1 บทนำ **1**

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1

1.3 การวิเคราะห์ระบบจ่ายยา 2

1.4 คุณสมบัติของอุปกรณ์แต่ละตัว 3

บทที่ 2 ทฤษฎี **5**

2.1 ลักษณะของ MCS – 51 5

2.2 การใช้งานอินเตอร์รัพต์ 6

2.2.1 แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ 6

2.2.2 การควบคุมอินเตอร์รัพต์ 7

2.2.3 การจัดลำดับความสำคัญของการอินเตอร์รัพต์ 8

2.2.4 สัญญาณอินเตอร์รัพต์จากภายนอก 11

2.2.5 สัญญาณอินเตอร์รัพต์จาก Timer 0 และ Timer 1 14

2.2.6 สัญญาณอินเตอร์รัพต์จากพอร์ตอนุกรม 14

2.2.7 สัญญาณอินเตอร์รัพต์จาก Timer 0 และ EXF 2 14

2.2.8 การทำงานแบบ SINGLE STEP 15

2.3 การใช้งานตัวตั้งเวลา/ตัวนับ 15

2.3.1 รีจิสเตอร์ TCON 16

2.3.2 รีจิสเตอร์ TMOD 18

2.3.3 รหัสการกำหนดโหมดการทำงานของ Timer 0 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.3.4 รหัสการกำหนดโหมดการทำงานของ Timer 1 20
2.3.5 การทำงานของ Timer ในโหมด 0 21
2.3.6 การทำงานของ Timer ในโหมด 1 22
2.3.7 การทำงานของ Timer ในโหมด 2 23
2.3.8 การทำงานของ Timer ในโหมด 3 25
2.3.9 การใช้งาน Timer 2 26
2.3.10 การทำงานของ Timer 2 ในโหมด แคปเจอร์ 28
2.3.11 การทำงานของ Timer 2 ในโหมด โหลดใหม่ 29
2.3.12 การทำงานในโหมดสร้างอัตรารับส่งข้อมูล 30
2.4 การใช้งานพอร์ตสื่อสารอนุกรมแบบ SINGLE PROCESSOR 30
2.5 การสื่อสารระหว่าง CPU หลายตัว 35
2.6 การเชื่อมต่อกับเครื่องพิมพ์ 37
2.6.1 สัญญาณต่างๆของเครื่องพิมพ์ 37
2.6.2 รหัสข้อมูลที่ใช้กับเครื่องพิมพ์ 40
2.6.3 การเชื่อมต่อไมโครคอนโทรเลอร์ 8051 กับเครื่องพิมพ์ 41
2.7 การขับโมดูลแสดงผลแบบผลึกเหลว (LCD Module) 43
2.7.1 โครงสร้างภายในของตัวควบคุมโมดูล LCD 44
2.7.2 โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16x1) 45
2.7.3 คำสั่งควบคุมโมดูล LCD 46
2.7.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD 49
2.7.5 จังหวะการทำงานของ LCD โมดูล 50
2.8 การใช้ DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC) 50
2.8.1 รายละเอียดขาต่อใช้งานของ DS1307 51
2.8.2 การทำงานของ DS1307 52
2.8.3 การจัดสรรหน่วยความจำใน DS1307 53
2.8.4 รีจิสเตอร์ควบคุม 54
2.8.5 โหมดการทำงานของ DS1307 55
2.8.6 โหมดการเขียนข้อมูล 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.8.7 โหมดการอ่านข้อมูล 56

2.8.8 การเชื่อมต่อ DS1307 กับไมโครคอนโทรลเลอร์ MCS-51 56

2.9 ความรู้เบื้องต้นของพอร์ตอนุกรม 57

2.9.1 มาตรฐานพอร์ตอนุกรมแบบ RS-232 57

2.9.2 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ 58

2.9.3 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232 60

2.9.4 แอดเดรสของพอร์ตอนุกรม 60

2.9.5 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม 61

2.9.6 คอนโทรล MSComm 62

2.9.7 การใช้ MSComm เพื่อการติดต่อฮาร์แวร์ 73

2.10 ความรู้เบื้องต้นของพอร์ตขนาน 74

2.10.1 ลักษณะทางกายภาพของพอร์ตขนาน 74

2.10.2 พอร์ตดาต้า (Data Port) 77

2.10.3 พอร์ต Control 77

2.10.4 พอร์ตแสดงสถานะหรือพอร์ต Status 79

2.10.5 การนำพอร์ตขนานไปใช้งาน 79

2.10.6 การเขียนโปรแกรมติดต่อกับพอร์ตขนาน 80

2.11 การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual BASIC 81

2.11.1 สำหรับระบบปฏิบัติการ 32 บิต 81

2.11.2 การติดต่ออุปกรณ์อินพุต-เอาต์พุตอย่างง่าย 81

2.12 การเชื่อมต่อพอร์ตอนุกรมกับระบบบัส I^2C 82

2.12.1 คุณสมบัติโดยทั่วไปของบัส I^2C 83

2.12.2 สภาวะที่เกิดขึ้นบนบัส I^2C 85

2.12.3 การทำงานบนบัส I^2C 86

2.12.4 การอ้างถึงแบบ 7 บิต (7-bit addressing) 87

2.12.5 วงจรแปลงสัญญาณพอร์ตอนุกรมเพื่อเชื่อมต่อกับระบบบัส I^2C 88

2.12.6 อุปกรณ์ที่ใช้การเชื่อมต่อแบบบัส I^2C 89

2.12.7 การเขียนโปรแกรมติดต่อบัส I^2C ด้วย Visual BASIC 89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.12.8 โปรแกรมย่อยสร้างสัญญาณเริ่มต้นหรือ START 89
2.12.9 โปรแกรมย่อยการสร้างสัญญาณหยุดหรือ Stop 90
2.12.10 โปรแกรมย่อยรอรับสัญญาณรับรู้หรือ Acknowledge 90
2.12.11 โปรแกรมย่อยส่งข้อมูล 91
2.12.12 โปรแกรมย่อยอ่านและส่งข้อมูลขนาด 8 บิต 91
2.12.13 พอร์ตอินพุตเอาต์พุตให้แก่พอร์ตอนุกรมด้วยระบบบัส I^2C 93
2.13 ข้อมูลเบื้องต้นของ PCF8574/8574A 93
2.13.1 การเขียนโปรแกรมควบคุม PCF8574 ด้วย Visual BASIC 95
2.13.2 การส่งค่าไปยังเอาต์พุตของ PCF8574A 96
2.13.3 การอ่านค่าข้อมูลจาก PCF8574A 96

บทที่ 3 การออกแบบและการควบคุม 98
3.1 วงจรการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ 98
3.2 วงจรส่วนการแสดงผล 102
3.2.1 ส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มาชำระเงิน 102
3.2.2 ส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มารับยา 104
3.3 ส่วนการใช้เสียงเรียก 109
3.4 ส่วนการตรวจสอบหมายเลขที่ถูกเรียกโดยตัวผู้ใช้บริการเอง 109

บทที่ 4 การใช้งาน 110
4.1 ส่วนการพิมพ์บัตรคิว 110
4.2 การใช้งานในส่วนของโปรแกรมการเรียกและการแสดงผล 114
4.2.1 การใช้งานโปรแกรม Pay_Channel 114
4.2.2 การใช้งานโปรแกรม Medic_Channel 117

บทที่ 5 สรุปและข้อเสนอแนะ 120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บรรณานุกรม 121

ภาคผนวก.....122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แหล่งกำเนิดสัญญาณอินเตอร์รัพท์จากแต่ละแหล่งที่มาจากอุปกรณ์ต่างๆ 7
2.2 ลำดับความสำคัญของการอินเตอร์รัพท์ 8
2.3 Vector Address ของการตอบสนองอินเตอร์รัพท์ 10
2.4 การเลือกโหมดของ Timer 19
2.5 ตารางกำหนดค่าการทำงานของ Timer ในโหมด 1 23
2.6 ตารางกำหนดค่าการทำงานของ Timer ในโหมด 2 24
2.7 ตารางกำหนดโหมดการทำงานของ Timer 2 28
2.8 ตารางการใช้ Timer 1 กำหนดบอดเรต 34
2.9 สัญญาณต่างๆของเครื่องพิมพ์ 38
2.10 รหัส ASCII ที่ใช้กับเครื่องพิมพ์ 40
2.11 แสดงการควบคุมการเลื่อนเคอร์เซอร์ 48
2.12 การเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SQW / OUT 54
2.13 ข้อมูลในแอดเดรส 0000:0411 H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม 61
2.14 สัญญาณสำคัญของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์ 75
2.15 สัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน 78
2.16 แอดเดรสของพอร์ตขนาน 80
2.17 แอดเดรสรีจิสเตอร์ของพอร์ตขนาน 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

1.1 Flow Chart แสดงขั้นตอนการรับจ่ายยา 2

2.1 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์ MCS-51 5

2.2 แหล่งกำเนิด สัญญาณอินเตอร์รัพท์ของ MCS-51 6

2.3 รีจิสเตอร์อินเตอร์รัพต์ 7

2.4 รีจิสเตอร์บอกลำดับความสำคัญการอินเตอร์รัพท์ 9

2.5 ระบบการควบคุมการอินเตอร์รัพท์ 10

2.6 บิตของรีจิสเตอร์ TCON 17

2.7 รีจิสเตอร์ TMOD 18

2.8 การทำงานของ Timer/Counter 1 ในโหมด 0 21

2.9 การทำงานของ Timer/Counter 1 ในโหมด 1 22

2.10 การทำงานของ Timer/Counter 1 ในโหมด 2 24

2.11 การใช้งาน Timer o ในโหมด 3 25

2.12 รีจิสเตอร์ T2 con 26

2.13 การทำงาน Timer 2 ในโหมด แคปเจอร์ 29

2.14 การทำงาน Timer 2 ในโหมด Auto Reload 29

2.15 Timing Diagram ของการสื่อสารอนุกรม 30

2.16 Timing Diagram ของการสื่อสารอนุกรมในโหมด 1 31

2.17 Timing Diagram ของการสื่อสารอนุกรมในโหมด 2 31

2.18 Timing Diagram ของการสื่อสารอนุกรมในโหมด 3 32

2.19 การเชื่อมต่อไมโครคอนโทรลเลอร์ จุดต่อจุด และหลายจุด 35

2.20 ข้อมูลกำหนดตำแหน่ง(Address Byte) 36

2.21 ข้อมูลข่าวสาร(Data Byte) 36

2.22 ไดอะแกรมการรับข้อมูลของเครื่องพิมพ์ 39

2.23 สัญญาณการเชื่อมต่อคอมพิวเตอร์กับเครื่องพิมพ์ 42

2.24 วงจรเชื่อมต่อ 8051 กับเครื่องพิมพ์ 43

2.25 ไดอะแกรมการทำงานของโมดูล LCD MODULE แบบอักขระ 44

2.26 รูปร่างและการจัดหาโมดูล LCD แบบอักขระ 45

2.27 การจัดขาไอซี DS 1307 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ หน้า

2.28 โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS 1307 53

2.29 (ก) การจัดสรรหน่วยความจำภายใน DS 1307 54

(ข) รายละเอียดของรีจิสเตอร์เก็บค่าเวลาและรีจิสเตอร์ควบคุมของ DS 1307 54

2.30 รูปแบบของข้อมูลสำหรับติดต่อกับ DS 1307 ในโหมดการเขียนข้อมูล 55

2.31 รูปแบบของข้อมูลสำหรับติดต่อกับ DS 1307 ในโหมดการอ่านข้อมูล 55

2.32 แสดงการเชื่อมต่อไมโครลคอนโทรลเลอร์ MCS-51
กับไอซีรีลไทม์คล็อก DS 1307 57

2.33 การจัดหาของคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232
ทั้งแบบ DB-9 และ DB-25 58

2.34 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ 59

2.35 แสดงไดอะแกรมเวลาของข้อมูล 74

2.36 แสดงระบบบัสภายในของพอร์ตขนาน 76

2.37 โครงสร้างวงจรเอาท์พุตของอุปกรณ์ที่ใช้เชื่อมต่อบนระบบของบัส I^2C 83

2.38 แสดงการเชื่อมต่ออุปกรณ์ระบบบัส I^2C ที่ใช้ไฟเลี้ยงไม่เท่ากัน 84

2.39 การต่อตัวต้านทานเพื่อป้องกันแรงดันกระชากที่อาจปะปนเข้ามาใน
ไฟเลี้ยงของอุปกรณ์ ในระบบบัส I^2C 84

2.40 ไดอะแกรมเวลาแสดงสภาวะต่างๆที่เกิดขึ้นในระบบบัส I^2C 86

2.41 รูปแบบของข้อมูลที่ใช้อ้างถึงอุปกรณ์บนระบบบัส I^2C 87

2.42 รูปแบบข้อมูลอนุกรมที่ใช้ในการติดต่อกับอุปกรณ์บนระบบบัส I^2C แบบ 7 บิต 87

2.43 วงจรสร้างสายสัญญาณเพื่อเชื่อมต่อกับระบบบัส I^2C ของพอร์ตอนุกรม 88

2.44 การจัดหาของ PCF 8574/8574A 93

2.45 วงจรภายในของ พอร์ต ของไอซี PCF 8574/8574A 94

3.1 การส่งข้อมูลแบบ Haft Duplex 98

3.2 การเชื่อมต่อไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก 99

3.3 วงจรของแต่ละ Segment 102

3.4 Block Diagram ของส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มาชำระเงิน 103

3.5 บอร์ดแปลงสัญญาณแบบ I^2C จาก Serial Port (RS-232) 103

3.6 บอร์ดวงจรเอาท์พุต 104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ หน้า

3.7 แสดงรูปแบบวงจรแสดงผล 3 หลักวงจรเดียว 104

3.8 Block Diagram ของส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มารับยา 105

3.9 บอร์ดแปลงสัญญาณแบบ I^2C จาก Parallel Port (RS-232) 105

3.10 บอร์ดวงจรเอาท์พุต 106

3.11 วงจรถอดรหัส 107

3.12 บอร์ด Display ที่แยกกันตามช่อง 107

3.13 บอร์ด Display ที่มีการแสดงผลรวมกัน 108

3.14 วงจร Power Supply 108

3.15 การแสดงผล การตรวจสอบว่า หมายเลขที่นำมาตรวจสอบนี้ยังไม่ถูกเรียก 109

3.16 การแสดงผล การตรวจสอบว่า หมายเลขที่นำมาตรวจสอบนี้ถูกเรียกแล้ว 109

4.1 แสดงลักษณะด้านหน้าของเครื่องพิมพ์บัตรคิว 110

4.2 แสดงลักษณะด้านบนของเครื่องพิมพ์บัตรคิว 110

4.3 การเชื่อมต่อของเครื่องพิมพ์บัตรคิว 111

4.4 ลักษณะของบัตรคิวที่ได้ 111

4.5 แสดงหน้าจอที่แสดงเมื่อมีการ SET UP เครื่องพิมพ์ 111

4.6 แสดงหน้าจอที่แสดงเมื่อมีการตั้งวันที่ 112

4.7 แสดงหน้าจอที่แสดงเมื่อมีการตั้งเวลา 112

4.8 แสดงหน้าจอที่แสดงเมื่อมีต้องการดูค่าหมายเลขที่พิมพ์ล่าสุด 113

4.9 แสดงหน้าจอที่แสดงเมื่อมีต้องการตั้งหมายเลขเครื่อง 113

4.10 แสดงหน้าจอที่แสดงเมื่อมีต้องการตั้งค่าจำนวนกระดาษที่จะพิมพ์ 113

4.11 แสดงหน้าจอที่แสดงเมื่อมีต้องการตั้งค่าจำนวนกระดาษที่จะพิมพ์ 114

4.12 แสดงหน้าจอของโปรแกรม Pay_Channel 115

4.13 กล่องข้อความเตือนเมื่อมีการกดปุ่ม เรียก หรือ จัดเก็บ โดยไม่มีตัวเลขใน ช่องแสดงหมายเลขบัตรคิว 116

4.14 กล่องข้อความเตือนเมื่อมีการกดปุ่ม มารับแล้ว โดยไม่มีตัวเลขใน ช่องแสดงหมายเลขตกค้าง 116

4.15 กล่องข้อความเมื่อมีการกดปุ่ม ลบทั้งหมด 116

4.16 แสดงหน้าจอของโปรแกรม Medic_Channel 117

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

4.17 กล่องข้อความเตือนเมื่อมีการกดปุ่ม เรียก หรือ รับแล้ว หรือ จัดเก็บ โดย ไม่มีตัวเลขในช่องแสดงหมายเลขช่อง 118
4.18 กล่องข้อความเตือนเมื่อมีการกดปุ่ม เรียก หรือ รับแล้ว หรือ จัดเก็บ โดย ไม่มีตัวเลขในช่องแสดงหมายเลขบัตรคิว 119
4.19 กล่องข้อความเตือนเมื่อมีการกดปุ่ม มารับแล้ว โดยไม่มีตัวเลขในช่องแสดง หมายเลขตกค้าง 119
4.20 กล่องข้อความเมื่อมีการกดปุ่ม ลบทั้งหมด 119

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

การมีสุขภาพแข็งแรงเป็นสิ่งที่พึงปรารถนาของมนุษย์ทุกคน เพื่อประกอบกิจการงานต่างๆ ในชีวิตประจำวันอย่างมีความสุข แต่ไม่มีใครสามารถหลีกเลี่ยงการเจ็บป่วยได้ ดังนั้น เมื่อเกิดการเจ็บป่วย จึงต้องได้รับการรักษาพยาบาล โดยสถานพยาบาลมีให้เลือกทั้งรัฐบาลและเอกชน

สถานพยาบาลของเอกชนจะให้บริการที่สะดวกและรวดเร็ว แต่จะคิดค่าบริการและราคาที่ค่อนข้างสูงกว่าสถานพยาบาลของรัฐ ด้วยเหตุนี้ผู้ป่วยจำนวนมากจึงเลือกใช้บริการจากสถานพยาบาลของรัฐ เพราะเสียค่าใช้จ่ายในการรักษาพยาบาลน้อยกว่า ถึงแม้จะไม่ได้รับความสะดวกและรวดเร็ว ซึ่งรัฐควรหาทางแก้ไขการให้บริการแก่ผู้ป่วยให้สะดวกและรวดเร็วขึ้น

คณะผู้จัดทำมีความเห็นว่า จุดสำคัญของการให้บริการแก่ผู้ป่วยเพื่อให้ได้รับความสะดวกและรวดเร็วนั้นอยู่ที่กระบวนการจัดลำดับการรับ-จ่ายยา ซึ่งในจุดการรับ-จ่ายยานี้ ถ้ามีการจัดการอย่างเป็นระบบ จะทำให้ผู้ป่วยได้รับความสะดวกและรวดเร็วยิ่งขึ้น

ซึ่งหวังว่า ถ้าสถานพยาบาลของรัฐหันมาใช้ระบบรับ-จ่ายยาดังกล่าว จะทำให้ผู้ป่วยที่มาใช้บริการ ได้รับความสะดวก รวดเร็ว และเกิดความประทับใจมากยิ่งขึ้น

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1.เพื่ออำนวยความสะดวกแก่ผู้ป่วยในการติดต่อรับยา โดยไม่ต้องเข้าแถวรอ

2.เพื่อความเป็นระเบียบเรียบร้อย และความยุติธรรม

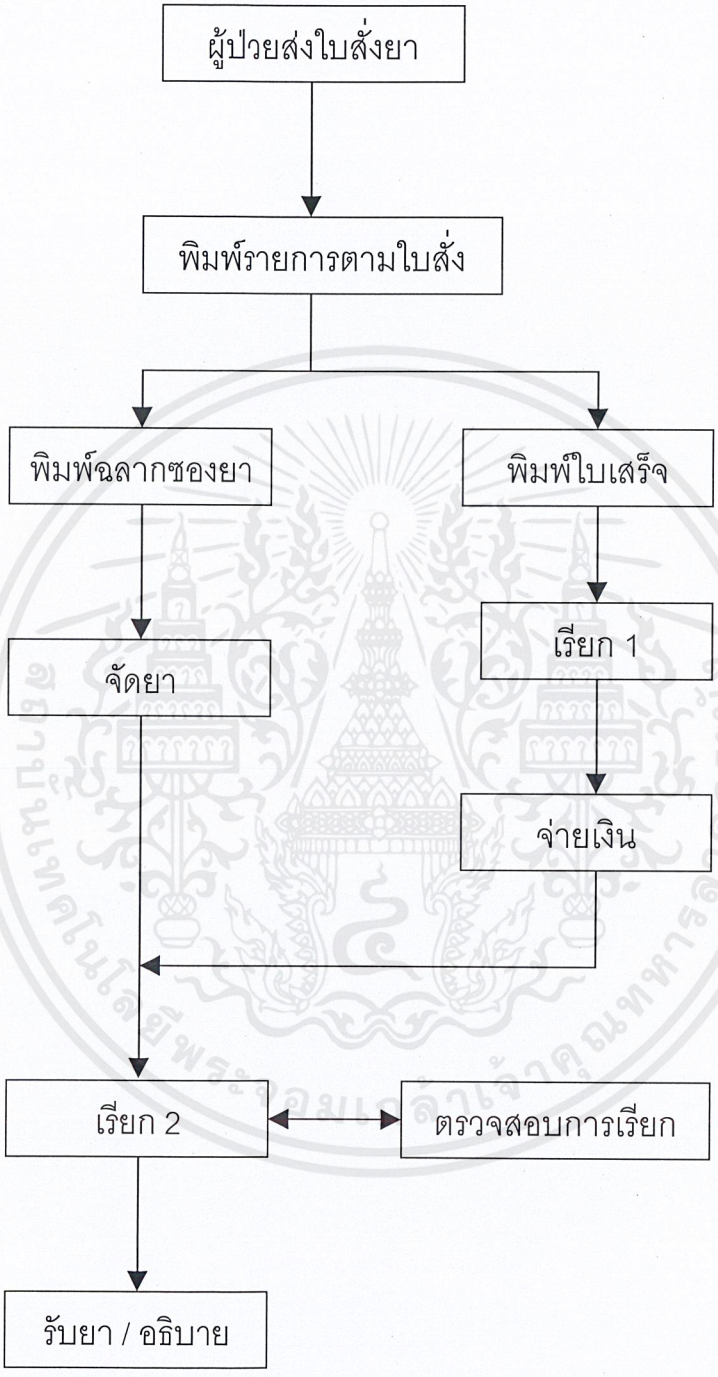
3.ใช้ในการเรียกหมายเลขบัตรคิว เพื่อลดความสับสนในชื่อของผู้ป่วย

4.จัดให้มีการแสดงผลการเรียกให้มีหลายสื่อ เพื่อเสริมการเรียกให้มีประสิทธิผลยิ่งขึ้น

5.ผู้ป่วยสามารถตรวจสอบการเรียกได้ด้วยตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 วิเคราะห์ระบบจ่ายยาในโรงพยาบาล



ภาพที่ 1.1 FLOW CHART แสดงขั้นตอนการรับ - จ่ายยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่พบ

1. ในขั้นตอนการส่งใบสั่งยา
   - ลำดับก่อน/ หลังอาจผิดพลาด
   - ต้องเข้าแถวส่งใบสั่งยา
2. ในขั้นตอนการเรียกผู้ป่วยมารับยา
   - อาจเกิดความสับสนในขั้นตอนการเรียกชื่อผู้ป่วยเช่น ชื่อซ้ำกัน หรือชื่อคล้ายกัน
   - ความผิดพลาดจากเสียงผู้เรียกเช่น ไม่ดังพอ ไม่ชัดเจน
3. ในขั้นตอนการตรวจสอบการเรียกรับยา
   - ต้องค้นหา ซึ่งไม่สะดวกในการตรวจสอบ
   -เจ้าหน้าที่ต้องหยุดงานที่ทำประจำชั่วคราว

แก้ปัญหาโดย

1. ในขั้นตอนการส่งใบสั่งยา จัดให้มีเครื่องจ่ายบัตรคิว
   - ให้หมายเลขบัตรคิว แสดงลำดับ ก่อน / หลัง เป็นการป้องกันความผิดพลาด
   - ไม่ต้องเข้าแถวรอ อันเป็นการกีดขวางทางเดิน
2. ในขั้นตอนการแยกผู้ป่วยมารับยา จัดให้มีส่วนแสดงผล LED DISPLAY & SOUND CALL
   - ใช้การเรียกด้วยหมายเลขบัตรคิวจาก BARCODE บนฉลากซองยา เพื่อลดความสับสน
   - มีการแสดงผลด้วย LED DISPLAY และ SOUND CALL เสริมการเรียกให้มีประสิทธิผลยิ่งขึ้น
3. ในขั้นตอนการตรวจสอบการเรียกรับยา จัดให้มีการตรวจสอบการเรียก
   - คนไข้สามารถตรวจสอบการเรียกได้ด้วยตนเอง โดยการรูดรหัส BARCODE บนบัตรคิวกับเครื่องตรวจสอบ

1.4 คุณสมบัติของอุปกรณ์แต่ละตัว

เครื่องจ่ายบัตรคิว
   - ต้องแสดงข้อมูล วันที่/เดือน/ปี , เวลา ชั่วโมง/นาที
   - ลำดับหมายเลข 1 - 999
   - BARCODE ต้องมีข้อมูลที่กล่าวมาแล้วด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแสดงผล LED DISPLAY

- LED DISPLAY ต้องมีขนาดที่มองเห็นได้ชัดเจน
- ต้องแสดงหมายเลขบัตรคิว และหมายเลขช่องรับยา
- การแสดงผลแต่ละครั้งจะใช้ BARCODE READER เป็นตัวอ่านข้อมูล

ส่วนแสดงผล SOUND CALL

- เสียงเรียกต้องดังพอเหมาะ และชัดเจน
- การบันทึกเสียงล่วงหน้า
- มีการเรียก 2 ครั้ง
- ในการแสดงผลแต่ละครั้งจะใช้ BARCODE READER เป็นตัวอ่านข้อมูล

เครื่องตรวจสอบการเรียก

- สามารถตรวจสอบได้ว่าหมายเลขนั้นเรียกไปแล้วหรือยัง

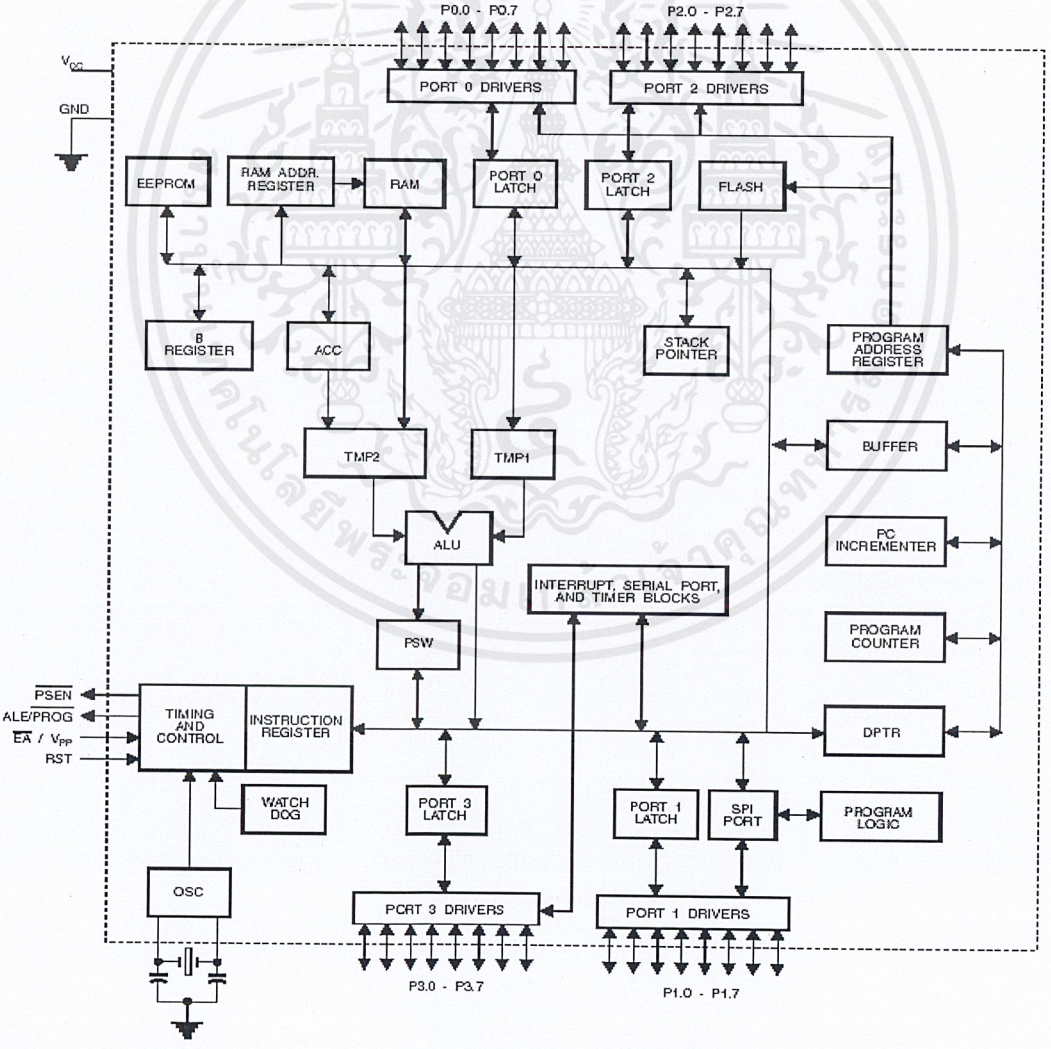
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ลักษณะของ MCS-51

โครงสร้างภายในของชิปไมโครคอนโทรลเลอร์ MCS-51 มีดังภาพที่ 2.1 และเนื่องจาก MCS-51 แต่ละเบอร์มีลักษณะโครงสร้างไม่เหมือนกันทีเดียว

8051s จะประกอบด้วยเบอร์ 8051,8051AH และ 8051BH และ ROMLESS VERSION คือเบอร์ 8031 (เป็น MCS-51 เบอร์ที่ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป) และEPROM VERSION คือเบอร์ 8751 (เป็น MCS-51 เบอร์ที่มีหน่วยความจำสำหรับเก็บโปรแกรม) 8052 ประกอบด้วย 8052AH, 8032 และ 8752BH



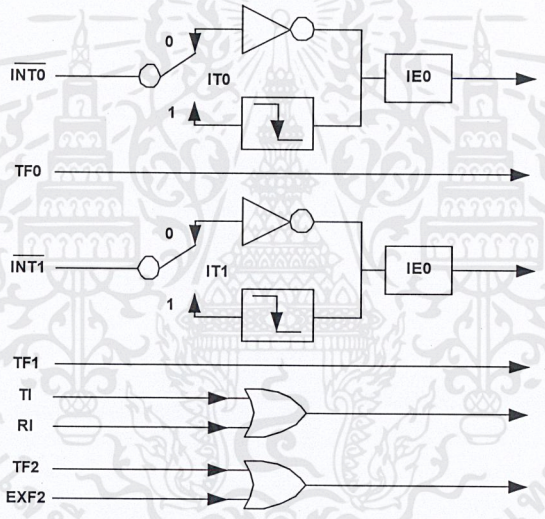
ภาพที่ 2.1 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การใช้งานอินเตอร์รัพต์

การติดต่อกันระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก สามารถทำได้ 2 ลักษณะ คือใช้วิธีการโพลลิ่ง(Polling)ซึ่งไมโครคอนโทรลเลอร์ต้องคอยตรวจสอบอุปกรณ์ภายนอกอยู่ตลอดเวลาว่ามีข้อมูลที่ต้องการติดต่อกับไมโครคอนโทรลเลอร์หรือไม่ซึ่งวิธีนี้จะทำให้ไมโครคอนโทรลเลอร์ต้องเสียเวลาไปกับการตรวจสอบนานมาก หากมีอุปกรณ์จำนวนมากต่ออยู่กับไมโครคอนโทรลเลอรการติดต่อแบบที่สองเป็นการใช้การอินเตอร์รัพต์ซึ่งเป็นวิธีการที่ไมโครคอนโทรลเลอร์ไม่ต้องคอยตรวจสอบอุปกรณ์ภายนอกอยู่ตลอด แต่จะให้อุปกรณ์ภายนอกส่งสัญญาณอินเตอร์รัพต์เข้ามาให้ไมโครคอนโทรลเลอร์เมื่ออุปกรณ์ภายนอกต้องการติดต่อกับไมโครคอนโทรลเลอร์ เมื่อไมโครคอนโทรลเลอร์ได้รับสัญญาณอินเตอร์รัพต์แล้วจึงจะทำการรับส่งข้อมูลกับอุปกรณ์ภายนอกนั้นๆ

2.2.1 แหล่งกำเนิดสัญญาณอินเตอร์รัพต์



ภาพที่ 2.2 แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ของ MCS-51

แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ของ 8061 มี 5 แหล่ง และ 8051 มี 6 แหล่ง (มี Timer 2 ที่เพิ่มขึ้น) สัญญาณร้องขออินเตอร์รัพต์ต่างๆ ของ 8051 แสดงในภาพที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แหล่งกำเนิดสัญญาณอินเตอร์รัพต์จากแต่ละแหล่งที่มาจากอุปกรณ์ต่างๆ

สัญญาณร้องขออินเตอร์รัพต์	ความหมาย
$\overline{\text{INT0}}$	สัญญาณร้องขออินเตอร์รัพต์จากภายนอกทางขา P3.2
$\overline{\text{INT1}}$	สัญญาณร้องขออินเตอร์รัพต์จากภายนอกทางขา P3.3
TF0	สัญญาณร้องขออินเตอร์รัพต์จาก Timer0 ที่อยู่ในตัวไมโครคอนโทรลเลอร์
TF1	สัญญาณร้องขออินเตอร์รัพต์จาก Timer 1 ที่อยู่ในตัวไมโครคอนโทรลเลอร์
RI, TI	สัญญาณร้องขออินเตอร์รัพต์จากพอร์ตอนุกรมที่อยู่ในตัวไมโครคอนโทรลเลอร์
TF2, EXF2	สัญญาณร้องขออินเตอร์รัพต์จาก Timer 2

2.2.2 การควบคุมอินเตอร์รัพต์

สัญญาณต่างๆ ของการร้องขออินเตอร์รัพต์ทั้งหมด เราสามารถเซตหรือเคลียได้ด้วยซอฟต์แวร์เช่นเดียวกับฮาร์ดแวร์ ดังนั้น การร้องขออินเตอร์รัพต์จึงสามารถสร้างหรือยกเลิกได้ด้วยซอฟต์แวร์ แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ทั้งหมด เราสามารถควบคุมให้ทำการร้องขออินเตอร์-รัพต์หรือไม่ก็ได้ โดยการเซตหรือเคลียบิตต่างๆ ที่อยู่ในรีจิสเตอร์ IE (Interrupt Enable) ภายในรีจิส-เตอร์ IE มีบิต EA ที่ทำหน้าที่ควบคุมให้ไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเตอร์รัพต์หรือไม่ตอบรับทั้งหมด หากเรากำหนดให้บิต EA เป็น 0 ไมโครคอนโทรลเลอร์จะไม่ตอบรับการร้องขออินเตอร์รัพต์ทั้งหมด

IE: INTERRUPT ENABLE REGISTER (Bit addressable)

IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

ภาพที่ 2.3 รีจิสเตอร์อินเตอร์รัพต์

EA IE7 (Disable all interrupt) หากกำหนดให้บิตนี้มีค่าเป็น 0 จะไม่มีการตอบรับอินเตอร์-รัพต์ทั้งหมด หากต้องการให้มีการตอบรับอินเตอร์รัพต์ จะต้องเซตให้ EA เป็น 1 และเซตบิตควบคุมแต่ละอินเตอร์รัพต์ที่ต้องการใช้อินเตอร์รัพต์ให้เป็น 1 ด้วย โดยใช้คำสั่งเซตหรือเคลียบิต

IE6 ไม่ใช้งาน

ET2 IE5 ควบคุมให้ Timer2 หรือ Capture ส่งอินเตอร์รัพต์หรือไม่ (1 = ส่งอินเตอร์รัพต์)

ES IE4 ควบคุมให้ พอร์ตอนุกรมส่งอินเตอร์รัพต์หรือไม่ (1 = ส่งอินเตอร์รัพต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET1 IE3 ควบคุมให้ Timer1 ส่งอินเตอร์รัพต์เมื่อถึงค่าที่ตั้งไว้หรือไม่ (1 = ส่งอินเตอร์รัพต์)

EX1 IE2 ควบคุมให้ตอบรับอินเตอร์รัพต์จากภายนอก $\overline{INT1}$ หรือไม่ (1 = ตอบรับอินเตอร์รัพต์)

ET0 IE1 ควบคุมให้ Timer0 ส่งอินเตอร์รัพต์เมื่อถึงค่าที่ตั้งไว้หรือไม่ (1 = ส่งอินเตอร์รัพต์)

EX0 IE0 ควบคุมให้ตอบรับอินเตอร์รัพต์จากภายนอก $\overline{INT0}$ หรือไม่ (1 = ตอบรับอินเตอร์รัพต์)

หากต้องการใช้สัญญาณอินเตอร์รัพต์จากแหล่งกำเนิดใด จะต้องเซตให้บิตควบคุมเป็น 1 และต้องเซตให้ EA เป็น 1 ด้วย

2.2.3 การจัดลำดับความสำคัญของการอินเตอร์รัพต์

การจัดลำดับความสำคัญของการอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ จัดได้ 2 ระดับแตกต่างกัน โดยการเซตค่าในรีจิสเตอร์ควบคุมการจัดลำดับความสำคัญของการอินเตอร์รัพต์ IP: Interrupt Priority Register) ในกรณีที่ไม่มีการจัดลำดับความสำคัญของการอินเตอร์รัพต์หรือจัดให้มีความสำคัญในระดับเดียวกัน ไมโครคอนโทรลเลอร์จะจัดให้มีลำดับความสำคัญของสัญญาณอินเตอร์รัพต์ที่อยู่ในระดับเดียวกันให้ (เพื่อแก้ปัญหาการขออินเตอร์รัพต์ในระดับเดียวกันที่แกิดขึ้นพร้อมกัน) ตามลำดับความสำคัญจากสูงไปต่ำดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 ลำดับความสำคัญของการอินเตอร์รัพต์

IE0	External Interrupt 0
TF0	Timer 0
IE1	External Interrupt 1
TF1	Timer 1
R1 หรือ T1	Serial Port
TF2 หรือ EXF2	Timer 2

หากต้องการกำหนดลำดับความสำคัญของการอินเตอร์รัพต์กำหนดได้จากบิตต่างๆ ของรีจิสเตอร์ IP โดยบิตที่มีค่าเป็น 1 จะมีลำดับความสำคัญสูงกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP: INTERRUPT PRIORITY REGISTER

IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0
-	-	PT2	PS	PT1	PX1	PT0	PX0

ภาพที่ 2.4 รีจิสเตอร์บอกลำดับความสำคัญการอินเตอร์รัพต์

-	IP7	ไม่ใช้
-	IP6	ไม่ใช้
PT2	IP5	กำหนดลำดับความสำคัญของ Timer 2
PS	IP4	กำหนดลำดับความสำคัญของพอร์ตอนุกรม
PT1	IP3	กำหนดลำดับความสำคัญของ Timer 1
PX1	IP2	กำหนดลำดับความสำคัญของสัญญาณ $\overline{\text{INT1}}$ (External Interrupt 1)
PT0	IP1	กำหนดลำดับความสำคัญของ Timer 0
PX0	IP0	กำหนดลำดับความสำคัญของสัญญาณ $\overline{\text{INT0}}$ (External Interrupt 0)

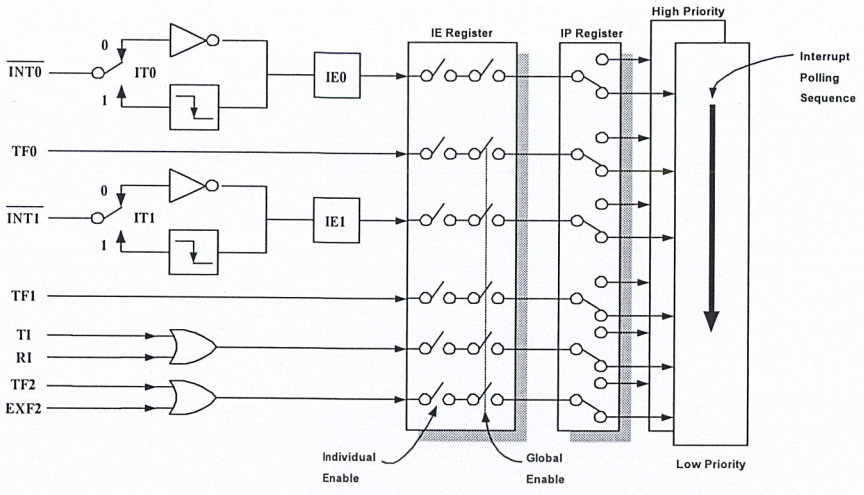
* หากบิตใดมีค่า 1 จะมีลำดับความสำคัญสูง

เมื่อมีการกำหนดลำดับความสำคัญของการอินเตอร์รัพต์แตกต่างกันแล้วหากมีการร้องขออินเตอร์รัพต์เกิดขึ้นพร้อมกันจากแหล่งกำเนิด 2 แหล่ง ที่มีระดับความสำคัญแตกต่างกัน ไมโครคอนโทรลเลอร์จะตอบรับการร้องขออินเตอร์รัพต์จากแหล่งกำนิดสัญญาณอินเตอร์รัพต์ที่มีระดับความสำคัญสูงกว่าสัญญาณการร้องขออินเตอร์รัพต์ที่มีระดับความสำคัญสูงกว่าสามารถอินเตอร์รัพต์ซ้อนในขณะที่ไมโครคอนโทรลเลอร์กำลังทำโปรแกรมตอบสนองการอินเตอร์รัพต์ของสัญญาณที่มีความสำคัญต่ำกว่าได้

หมายเหตุ ในขณะที่ไมโครคอนโทรลเลอร์กำลังทำโปรแกรมตอบสนองการอินเตอร์รัพต์อยู่ จะไม่สามารถตอบรับการร้องขออินเตอร์รัพต์จากแหล่งกำเนิดระดับที่ต่ำกว่าหรือระดับเดียวกันได้อีก จนกว่าจะจบโปรแกรมตอบสนองการอินเตอร์รัพต์ก่อน

วงจรการควบคุมการอินเตอร์รัพต์ของ MCS-51 ทำโดยการควบคุมการตัดต่อสวิตช์ต่างๆ ดังแสดงในรูป2.5 ซึ่งควบคุมการตัดต่อสวิตช์ด้วยซอฟต์แวร์จากการเซตหรือเคลียบิตของรีจิสเตอร์ IE,IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.5 ระบบควบคุมการอินเตอร์รัพต์

หากต้องการให้ไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเตอร์รัพต์ จะต้องมีการกำหนดการทำงานตามขั้นตอนต่อไปนี้

1. เซตบิต EA (Enable all) ในรีจิสเตอร์ IE7 ให้เป็น 1
2. เซตบิตควบคุมอินเตอร์รัพต์ของแต่ละแหล่งกำเนิดที่ต้องการใช้งานอินเตอร์รัพต์ในรีจิสเตอร์ IE ให้มีค่าเป็น 1
3. กำหนดลำดับความสำคัญของการอินเตอร์รัพต์ (หากต้องการ) โดยกำหนดในรีจิสเตอร์ IP
4. เขียนโปรแกรมตอบสนองการร้องขออินเตอร์รัพต์ของแต่ละแหล่งกำเนิดลงในตำแหน่งต่างๆ ตาม Vector Address ในตารางที่ 2.3

ตารางที่ 2.3 Vector Address ของการตอบสนองอินเตอร์รัพต์

Interrupt Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
R1 & T1	0023H
TF2 & EXF2	002BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงตำแหน่งต่างๆ ของโปรแกรมตอบสนองอินเตอร์รัพต์ของแต่ละแหล่งกำเนิดซึ่งจะต้องนำคำสั่งเริ่มต้นของโปรแกรมตอบสนองอินเตอร์รัพต์ไปบรรจุไว้ หากมีการใช้อินเตอร์รัพต์ทั้งหมดจะพบว่าความยาวของโปรแกรมตอบสนองอินเตอร์รัพต์ของแต่ละโปรแกรมจะมีความยาวเพียง 8 ไบต์ เท่านั้น หากโปรแกรมมีความยาวมากกว่านี้ ต้องนำไปเขียนเป็นโปรแกรมย่อยไว้ภายนอก แล้วใช้วิธีการเรียกโปรแกรมย่อยมาทำงาน

2.2.4 สัญญาณอินเตอร์รัพต์จากภายนอก

แหล่งกำเนิดสัญญาณอินเตอร์รัพต์จากภายนอกมี 2 สัญญาณคือ $\overline{\text{INT0}}$ และ $\overline{\text{INT1}}$ ซึ่งอยู่ที่ขา P3.2 และ P3.3 เราสามารถเลือกลักษณะสัญญาณร้องขออินเตอร์รัพต์ได้สองลักษณะคือ เลือกใช้สัญญาณแอคทีฟที่ระดับลอจิก 0 หรือเลือกใช้สัญญาณแอคทีฟที่การเปลี่ยนแปลงของสัญญาณจาก 1 ไป 0 ซึ่งการเลือกทำโดยการกำหนดค่าบิตของ IT0 และ IT1 ที่อยู่ในรีจิสเตอร์ TCON หากรีเซตให้ค่าของบิต IT0 เป็น 0 จะเป็นการกำหนดการเลือกใช้สัญญาณแอคทีฟที่ระดับลอจิก 0 หากเซตให้ IT0 = 1 เป็นการเลือกใช้สัญญาณแอคทีฟที่การเปลี่ยนระดับสัญญาณจาก 1 ไป 0 การตรวจสอบการร้องขอการอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์จะกระทำในทุกๆแมชชีนไซเกิลภายในตัวไมโครคอนโทรลเลอร์ มีแฟลกการร้องขออินเตอร์รัพต์จากสัญญาณภายนอกทั้งสองอยู่ที่บิต IE0 และ IE1 ในรีจิสเตอร์ TCON โดยแฟลกการร้องขออินเตอร์รัพต์จะถูกเซตเมื่อมีสัญญาณอินเตอร์รัพต์จากภายนอกเข้ามาตามเงื่อนไขที่กำหนด

ในกรณีที่เรากำหนดให้สัญญาณอินเตอร์รัพต์แอคทีฟ เมื่อมีการเปลี่ยนค่าจาก 1 ไป 0 แฟลกการร้องขออินเตอร์รัพต์จะถูกเซตเมื่อสัญญาณที่ขา INT0 หรือ INT1 เปลี่ยนจาก 1 ไป 0 (ค่าของลอจิก 1 หรือลอจิก 0 จะต้องนานกว่า 12 คาบเวลาของสัญญาณนาฬิกาเพื่อให้ไมโครคอนโทรลเลอร์สามารถตรวจสอบได้ทัน) คาของแฟลกการร้องขออินเตอร์รัพต์ IE0 หรือ IE1 จะถูกเคลียจากฮาร์ดแวร์อัตโนมัติเมื่อไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเตอร์รัพต์ โดยการเรียกโปรแกรมย่อยของการตอบสนองการอินเตอร์รัพต์มาทำงาน

หากเรากำหนดให้สัญญาณอินเตอร์แอคทีฟที่ลอจิก 0 สัญญาณการร้องของอินเตอร์รัพต์ที่ส่งเข้ามาจะต้องค้างเป็น 0 ไว้จนกว่าไมโครคอนโทรลเลอร์จะทำการตอบรับการร้องขออินเตอร์รัพต์ การตอบรับการร้องขออินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ โดยการเรียกโปรแกรมย่อยการตอบสนองอินเตอร์รัพต์มาทำงาน เมื่อไมโครคอนโทรลเลอร์เรียกโปรแกรมการตอบสนองอินเตอร์รัพต์มาทำงานแล้ว ต้องเลิกส่งสัญญาณการร้องขออินเตอร์รัพต์ก่อนที่โปรแกรมการตอบสนองจะจบลง หรือก่อนที่การร้องขออินเตอร์จากแหล่งกำเนิดอื่นจะเข้ามาเพื่อป้องกันไม่ให้เกิดการอินเตอร์รัพต์ซ้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ การใช้งานสัญญาณร้องขออินเตอร์รัพต์จากภายนอก $\overline{\text{INT0}}$ และ $\overline{\text{INT1}}$ (ขา P3.2 และ P3.3) จะต้องเซตบิต P3.2 และ P3.3 ให้เป็น 1 ในตอนเริ่มต้นก่อนเพื่อทำให้สูงโดยใช้คำสั่ง SETB P3.2 และ SETB P3.3

2.2.4.1 การใช้งานสัญญาณ $\overline{\text{INT0}}$

เนื่องจากตำแหน่งของโปรแกรมตอบสนองการอินเตอร์รัพต์ของสัญญาณ $\overline{\text{INT0}}$ เริ่มต้นที่ตำแหน่ง 0003 และโปรแกรมหลักหลังการรีเซตไมโครคอนโทรลเลอร์เริ่มต้นทำงานที่ตำแหน่ง 0000 ดังนั้น คำสั่งแรกของโปรแกรมหลักที่อยู่ในตำแหน่ง 0000 ต้องเป็นคำสั่งการกระโดดข้ามส่วนของโปรแกรมตอบสนองอินเตอร์รัพต์ไป เพื่อไม่ให้มีการทำโปรแกรมตอบสนองอินเตอร์รัพต์ก่อนที่จะมีการอินเตอร์รัพต์ ในส่วนของโปรแกรมหลักจะต้องมีการกำหนดให้ไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเตอร์รัพต์จากสัญญาณ $\overline{\text{INT0}}$ โดยกำหนดให้บิตที่ 7 (EA) ของรีจิสเตอร์ IE = 1 เพื่อให้ไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเตอร์รัพต์ และกำหนดให้ EX0 อีนาเบิล โดยกำหนดบิต 0 ของรีจิสเตอร์ IE = 1 และต้องเซตให้บิตที่พอร์ต P3.21 = 1 ด้วย

โปรแกรมตอบสนองอินเตอร์รัพต์ของสัญญาณ INT0 จะมีลักษณะเป็นโปรแกรมย่อย แต่คำสั่งสุดท้ายจะเป็น RFT1 เพื่อแสดงการจบโปรแกรมตอบสนองอินเตอร์รัพต์

ตัวอย่าง การใช้งาน INT0 เมื่อมีการร้องขออินเตอร์รัพต์ให้ทำการเพิ่มค่าในหน่วยความจำข้อมูลภายในที่ตำแหน่ง 18H ขึ้นไป 1 ค่า เลือกใช้การแอคทีฟของสัญญาณอินเตอร์รัพต์ที่ขอบขาลง (เปลี่ยนจาก 1 ไป 0)

```
        ORG     0000
        LIMP    MAIN        : กระโดดข้ามโปรแกรมตอบสนองอินเตอร์รัพต์
        ORG     0003H       : ตำแหน่งเริ่มต้นของโปรแกรมตอบสนอง INT0
INT0_SERV:  PUSH    PSW     : โปรแกรมตอบสนองอินเตอร์รัพต์ของ INT0
        INC     18H         : ส่วนโปรแกรม
        POP     PSW
        RET1                : จบโปรแกรมตอบสนองการอินเตอร์รัพต์
MAIN    MOV     IE,#10000001B   :เริ่มต้นโปรแกรมหลักกำหนดการตอบรับ INT0
        SETB    P3.2        : กำหนด บิตที่ P3.2 = 1
        SETB    TCON.0      : เลือกสัญญาณแอคทีฟ INT0 ที่ขอบขาลง
        …                   : ส่วนการทำงานของโปรแกรมหลัก
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างหลังจากโปรแกรมทำงานแล้วจะกระโดดข้ามโปรแกรมตอบสนองอินเตอร์รัพต์ไปทำงานที่โปรแกรมหลักซึ่งมีการเซตบิตต่างๆ เมื่อมีสัญญาณอินเตอร์รัพต์เข้ามาจะกระโดดไปทำโปรแกรมตอบสนองอินเตอร์รัพต์ที่ตำแหน่ง 0003 ในส่วนของโปรแกรมตอบสนองอินเตอร์รัพต์จะมีการเก็บค่าของ PSW ลงในสแตกก่อนและดึงกลับมาเมื่อจบการทำงานของโปรแกรมตอบสนองอินเตอร์รัพต์

ในกรณีที่มีการใช้งานอินเตอร์รัพต์จากแหล่งอื่นด้วยเช่น TF0 ซึ่งมีตำแหน่งของโปรแกรมตอบสนองการอินเตอร์รัพต์เริ่มต้นที่ 000BH ทำให้ความยาวของโปรแกรมตอบสนองอินเตอร์รัพต์ของ $\overline{\text{INT0}}$ ถูกจำกัดอยู่ในช่วง 0003H - 000AH ซึ่งไม่สามารถเก็บโปรแกรมที่มีความยาวมากกว่า 8 ไบต์ แยกโปรแกรมตอบสนองอินเตอร์รัพต์ของ INT0 ไปเขียนเป็นโปรแกรมย่อยไว้ภายนอกแล้วเรียกมาใช้งานดังนี้

```
                ORG     0000
                LIMP    MAIN        :กระโดดข้ามโปรแกรมตอบสนองอินเตอร์รัพต์
                ORG     0003H       :ตำแหน่งเริ่มต้นของโปรแกรมตอบสนอง INT0
INT0_SERV:      LCALL   SUB_INT0    :เรียกโปรแกรมตอบสนองอินเตอร์รัพต์ของ INT0
                RET1                :จบโปรแกรมตอบสนองการอินเตอร์รัพต์
                ORG     000BH       :ตำแหน่งเริ่มต้นของโปรแกรมตอบสนอง TIMER0
TIMERO-SERV:    PUSH    PSW
                …                   :ส่วนโปรแกรมตอบสนองของ TIMER0
                …
                POP     PSW
                RET1
MAIN            MOV     IE,#100000011B: ส่วนเริ่มต้นโปรแกรมหลัก
                SETB    P3.2    :   กำหนด บิตที่ P3.2 = 1
                                :   ส่วนโปรแกรมหลัก
SUB_INT0 :      PUSH    PSW     :   โปรแกรมย่อยตอบสนองอินเตอร์รัพต์
                …               :   โปรแกรมย่อยตอบสนองอินเตอร์รัพต์ของ INT0
                …
                POP     PSW
                RET
                END
```

ในส่วนของโปรแกรมสนองอินเตอร์รัพต์ทั้งหมดจะต้องมีการเก็บค่าของรีจิสเตอร์ที่ใช้งานในโปรแกรมตอบสนองอินเตอร์รัพต์ทั้งหมด ลงสแตกและดึงกลับออกมาก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อไม่ให้ข้อมูลของโปรแกรมหลักเปลี่ยนแปลงไป เนื่องจากสัญญาณอินเตอร์รัพต์จะเกิดขึ้นในเวลาใดก็ได้ ดังนั้นค่าของรีจิสเตอร์ต่างๆ ที่โปรแกรมตอบสนองการอินเตอร์รัพต์ใช้จะเปลี่ยนค่าไปจากเดิมเมื่อจบโปรแกรมตอบการอินเตอร์รัพต์ หากไม่มีการเก็บค่าเดิมไว้จะทำให้ค่าเดิมที่มาจากโปรแกรมหลักเปลี่ยนไป ในตัวอย่างนี้เราเก็บเพียงรีจิสเตอร์ PSW เท่านั้น

การใช้งานสัญญาณอินเตอร์รัพต์ของ INT1 จะมีลักษณะเช่นเดียวกับการใช้อินเตอร์-รัพต์ของ INT0 ซึ่งตำแหน่งของโปรแกรมตอบสนองอินเตอร์รัพต์ของ INT1 จะอยู่ที่ 0013H แทน 0003H

2.2.5 สัญญาณอินเตอร์รัพต์จาก Timer 0 และ Timer 1

แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ TF0 และ TF1 เป็นแหล่งกำเนิดที่มาจากตัวตั้งเวลาตัวนับ Timer 0 และ Timer 1 ตามลำดับ TF0 และ TF1 จะเกิดขึ้นเมื่อค่าเวลาของ Timer0 และ Timer 1 ที่ตั้งไว้ครบกำหนดแล้วหรือนับได้ครบจำนวนที่ตั้งแล้วในกรณีที่ทำหน้าที่เป็นตัวนับ โดย TF0 และ TF1 จะถูกเซตโดยฮาร์ดแวร์ภายในตัวไมโครคอนโทรลเลอร์ และค่าของแฟลก TF0 และ TF1 จะถูกเคลียอัตโนมัติโดยฮาร์ดแวร์เมื่อไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเตอร์รัพต์โดยการเรียกโปรแกรมย่อยที่อยู่ในตำแหน่งเริ่มต้นที่ 0BH (สำหรับ Timer0) หรือ 1BH (สำหรับ Timer 1) มาทำงาน สำหรับการใช้งานจะขอนำไปกล่าวในเรื่องการใช้งาน Timer

2.2.6 สัญญาณอินเตอร์รัพต์จากพอร์ตอนุกรม

แหล่งกำเนิดสัญญาณอินเตอร์รัพต์จากพอร์ตอนุกรมมีสัญญาณอินเตอร์รัพต์เกิดจาก 2 แหล่ง คือ จาก RI (ด้านรับ) และ TI (ด้านส่ง) เป็นแหล่งกำเนิดที่มาจากบัฟเฟอร์ทางด้านรับและส่งข้อมูลของพอร์ตอนุกรมที่อยู่ในไมโครคอนโทรลเลอร์ สัญญาณทั้งสองจะถูกนำมา OR กันเพื่อรวมเป็นสัญญาณเดียว ทำให้มีสัญญาณร้องขออินเตอร์รัพต์เกิดขึ้นได้ทั้งในกรณีของการรับและการส่งข้อมูลของพอร์ตอนุกรม โปรแกรมย่อยของการตอบสนองอินเตอร์รัพต์ของพอร์ตอนุกรมอยู่ที่ตำแหน่ง 23H ตำแหน่งเดียว ดังนั้นในส่วนของโปรแกรมตอบสนองการอินเตอร์รัพต์ของพอร์ตอนุกรม จะต้องมีการตรวจสอบก่อนว่าการขออินเตอร์รัพต์เกิดขึ้นจาก RI หรือ TI โดยใช้คำสั่งการตรวจสอบบิต RI หรือ TI ที่อยู่ในรีจิสเตอร์ SCON เมื่อทราบที่มาของการร้องขออินเตอร์รัพต์แล้วจึงไปทำโปรแกรมสำหรับการรับหรือการส่งข้อมูลต่อไป หลังจากตรวจสอบแหล่งกำเนิดสัญญาณร้องขออินเตอร์รัพต์แล้วจะต้องทำการเคลียร์บิต RI หรือ TI ด้วยซอฟต์แวร์ก่อนที่จะจบโปรแกรมตอบสนองการอินเตอร์รัพต์ ตัวอย่างการใช้งานจะอยู่ในเรื่องการรับส่งข้อมูลของพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 สัญญาณอินเตอร์รัพต์จาก Timer2 และ EXF2

แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ของ Timer2 มีอยู่ 2 แหล่งคือ TF2 และ EXF2 ซึ่งจะมีเฉพาะในไอซี 8052 และ 8032 สัญญาณ TF2 และ EXF2 จะถูกนำมารวมกันด้วยวงจร OR เมื่อมีการตอบรับการร้องขออินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ต่อ TF2 และ EXF2 ไมโครคอนโทรลเลอร์จะเรียกโปรแกรมย่อยของการตอบสนองอินเตอร์รัพต์ที่อยู่ในตำแหน่ง 002BH มาทำงาน สัญญาณ TF2 และ EXF2 จะไม่ถูกเคลียจากฮาร์ดแวร์ในขณะที่มีการตอบสนองการอินเตอร์รัพต์ ในโปรแกรมตอบสนองการอินเตอร์รัพต์ต้องมีการตรวจสอบก่อนว่าแหล่งกำเนิดมาจาก TF2 หรือ EXF2 เพื่อที่จะได้เลือกทำโปรแกรมตอบสนองได้ถูกต้องแล้วจึงทำคำสั่งเคลียร์บิต TF2 หรือ EXF2 ก่อนที่จะจบโปรแกรมตอบสนองการอินเตอร์รัพต์

2.2.8 การทำงานแบบ SINGLE STEP

โครงสร้างอินเตอร์รัพต์ของ 8051 ช่วยสนับสนุนให้มีการทำงานแบบ Single Step ได้ การเขียนคำสั่งควบคุมเพิ่มขึ้นเพียงไม่กี่คำสั่งจากที่ผ่านมาเราทราบว่าการตอบสนองอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์จะไม่เกิดขึ้นขณะที่ไมโครคอนโทรลเลอร์กำลังทำโปรแกรมตอบสนองอินเตอร์รัพต์ของสัญญาณร้องขออินเตอร์รัพต์ที่มีระดับความสำคัญเท่ากันอยู่ และเช่นกันไมโครคอนโทรลเลอร์จะไม่ตอบรับอินเตอร์รัพต์ก่อนที่จะทำคำสั่งอื่น 1 คำสั่งหลังจากคำสั่ง RET1 ดังนั้นเมื่อไมโครคอนโทรลเลอร์ไปทำโปรแกรมตอบสนองอินเตอร์รัพต์อยู่ ไมโครคอนโทรลเลอร์ไม่สามารถกระโดดไปทำโปรแกรมตอบสนองอินเตอร์รัพต์เดียวกันได้อีกจนกว่าจะมีการทำคำสั่งในโปรแกรมหลักจบไป 1 คำสั่ง การนำลักษณะดังกล่าวมาใช้งาน ในการสร้างการทำงานแบบ Single Step ทำได้โดยการเขียนโปรแกรมให้มีการอินเตอร์รัพต์จากภายนอก (ใช้ INT0) และเขียนโปรแกรมตอบสนองอินเตอร์รัพต์ที่จบด้วยคำสั่งต่อไปนี้

```
JNB     P3.2, $   :  รอสัญญาณ INT0 เป็น 1
JB      P3.2, $   :  รอสัญญาณ INT0 เป็น 0
RET1              :  กลับไปจากการอินเตอร์รัพต์และทำคำสั่ง 1 คำสั่ง
```

เมื่อสัญญาณ INT0 (อยู่ที่ขา P3.2) ถูกต่อกับลอจิก 0 จะทำให้ไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองอินเตอร์รัพต์ของ INT0 และวนอยู่จนกระทั่งสัญญาณ INT0 เป็นพัลส์ (จาก 0 ไป 1 ไป 0) แล้วจึงทำคำสั่ง RET1 แล้วกลับไปทำโปรแกรมเดิมต่ออีก 1 คำสั่ง และกลับไปทำงานในส่วนของโปรแกรมตอบสนองอินเตอร์รัพต์ใหม่ และรอสัญญาณพัลส์ที่ขา INT0 อีก ซึ่งการทำงานจะทำ 1 Step ต่อสัญญาณ 1 พัลส์ที่ขา INT0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การใช้งานตัวตั้งเวลา/ตัวนับ

โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ 8051 มีตัวตั้งเวลา/ตัวนับ (Timer/Counter) ประกอบอยู่ภายใน 2 ตัว คือ Timer0 และ Timer1 (ใน 8032 และ 8052 มี Timer2 เพิ่มขึ้นอีก 1 ตัว) การใช้งานของ Timer เราสามารถกำหนดการใช้งานในลักษณะของตัวตั้งเวลา หรือในลักษณะของตัวนับก็ได้ ภายในตัว Timer แต่ละตัวมีวงจรนับขนาด 8 บิต แบบนับขึ้น (Count up Counter) อยู่ 2 ตัว คือ TH และ TL ซึ่งสามารถใช้ร่วมกันเป็นรีจิสเตอร์ขนาด 16 บิตได้ ค่าเริ่มต้นของการนับจะกำหนดได้จากโปรแกรม การกำหนดการทำงานของ Timer ทำโดยการกำหนดค่าให้กับรีจิสเตอร์ TMOD และ TCON การทำงานของ timer แต่ละตัวจะทำงานเป็นอิสระจากกัน

การทำงานในโหมดของตัวจับเวลา จะใช้สัญญาณนาฬิกาที่อยู่ภายในตัวไมโครคอนโทรลเลอร์ ซึ่งทราบความถี่ ค่าของเวลาเริ่มต้นที่อยู่ในรีจิสเตอร์ TH และ TL จะเพิ่มขึ้น 1 ในทุกแมชชีนไซเกิล (ความถี่สัญญาณนาฬิกาหารด้วย 12) เมื่อค่าของรีจิสเตอร์ TH และ TL เพิ่มขึ้นเป็น 1 ทุกบิตแล้วจะวนกลับไปเป็น 0 ใหม่อีก ช่วงนี้เรียกว่าเป็นโอเวอร์โฟลว (Overflow) ซึ่งเป็นช่วงของการครบกำหนดเวลาที่ตั้งไว้ ค่าของเวลาสามารถคำนวณได้จากค่าเริ่มต้นใน TH และ TI ไปจนถึงค่าสูงสุดคือ 0000H (หรือ 65536) ตัวอย่างเช่น ไมโครคอนโทรลเลอร์ใช้สัญญาณนาฬิกา 12 MHz ซึ่งมีค่าแมชชีนไซเกิลเท่ากับ 1 μSec หากต้องการให้ Timer ตั้งเวลา 50 μSec ค่าเริ่มต้นของรีจิสเตอร์ TH และ TL จะคำนวณได้จาก

TH.TL = 65536 – 50 = 65486

= FFCEH

หมายเหตุ ค่าเริ่มต้นสามารถหาได้จาก ค่า 2's complement ของจำนวนที่ต้องการนับก็ได้เช่นกัน ดังในตัวอย่างเลย 2's complement ของ 50 (32H) คือ FFCEH

การทำงานในโหมดของการนับ ค่าของการนับที่อยู่ในรีจิสเตอร์ TH และ TL จะเพิ่มขึ้นเมื่อสัญญาณจากภายนอกเข้ามาที่ขา T0 (หรือ T1) มีการเปลี่ยนแปลงจาก 1 ไป 0 ความสามารถในการนับของตัวนับจะนับได้กับความถี่ไม่สูงกว่า 1/24 เท่าของสัญญาณนาฬิกา โดยสัญญาณพัลส์ที่เข้ามาจะมีช่วงห่างเท่าใดก็ได้ การกำหนดค่าเริ่มต้นของการนับกำหนดในรีจิสเตอร์ TH และ TL ดังนั้นจำนวนของการนับ จะได้จากค่าที่เหลืออยู่ในรีจิสเตอร์ TH และ TL ดังนั้น จำนวนของการนับจะได้จากค่าที่เหลืออยู่ในรีจิสเตอร์ TH และ TL เช่นเดียวกับในโหมดตั้งเวลา

การทำงานของ Timer0 และ Timer1 ในโหมดตั้งเวลาหรือโหมดตัวนับแบ่งการทำงานได้เป็น 4 โหมด คือ โหมด 0, 1, 2 และ 3 ส่วนการทำงานของ Timer2 มี 3 โหมด คือ Capture, Auto reload และ Buad rate generator รีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของตัวตั้งเวลา/ตัวนับ มี 2 ตัว คือ รีจิสเตอร์ TCON และรีจิสเตอร์ TMOD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3.1 รีจิสเตอร์ TCON (Timer/Counter Control Register)

เป็นรีจิสเตอร์ที่ทำหน้าที่เฉพาะ (อยู่ในตำแหน่ง 88H สามารถอ้างตำแหน่งแบบบิตได้) สำหรับควบคุมการทำงานของ Timer และเลือกลักษณะสัญญาณอินเตอร์รัพต์จากภายนอกโดยรีจิส-เตอร์นี้จะทำหน้าที่เป็นแฟลกแสดงสถานะการทำงาน 4 บิตที่เซตและเคลียด้วยการทำงานของฮาร์ดแวร์ ส่วนที่เหลืออีก 4 บิตเป็นบิตควบคุมการทำของ Timer ซึ่งเราเซตและเคลียด้วยคำสั่งทางซอฟต์แวร์จากการใช้คำสั่งการเซตบิตหรือการโอนย้ายข้อมูลก็ได้ บิตต่างๆ ของรีจิสเตอร์ TCON แสดงในรูป 2.6

TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Time/counter control register: TCON (Bit addressable)

ภาพที่ 2.6 บิตของรีจิสเตอร์ TCON

TF1: TCON.7 (Timer1 Overflow flag) ทำหน้าที่เป็นแฟลก ซึ่งถูกเซตด้วยฮาร์ดแวร์ให้เป็น 1 เมื่อค่าการนับของ Timer 1 เกิดโอเวอร์โฟลว (นับขึ้นไปถึง 0000) และบิตนี้จะถูกเคลียร์ด้วยฮาร์ดแวร์อัตโนมัติ เมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัพต์ของ Timer1 ซึ่งอยู่ในตำแหน่งเริ่มต้นที่ 001BH

TR1: TCON.6 (Timer1 run control bit) เป็นบิตที่ควบคุมการทำงานของ Timer1 เขตจากโปรแกรม

1 = ควบคุมให้ Timer1 ทำงาน

0 = ควบคุมให้ Timer1 หยุดทำงาน

TF0: TCON.5 (Timer0 Overflow flag) ทำหน้าที่เป็นแฟลก ซึ่งถูกเซตด้วยฮาร์ดแวร์ให้เป็น 1 เมื่อทำการนับของ Timer0 เกิดโอเวอร์โฟลว (นับขึ้นไปถึง 0000) และบิตนี้จะถูกเคลียร์ด้วยฮาร์ดแวร์ เมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัพต์ของ Timer0 ซึ่งอยู่ในตำแหน่งเริ่มต้นที่000BH

TR0: TCON.4 (Timer0 run control bit) เป็นบิตที่ควบคุมการทำงานของ Timer0 เซตจากโปรแกรม

1 = ควบคุมให้ Timer0 ทำงาน

0 = ควบคุมให้ Timer0 หยุดทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

45861

IE1: TCON.3 (External interrupt 1 edge flag) เป็นแฟลกการร้องขออินเตอร์รัพต์ภายนอกของสัญญาณ $\overline{INT1}$ บิตนี้จะถูกเซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณอินเตอร์รัพต์เข้ามาที่ขา $\overline{INT1}$ และบิตนี้จะถูกเคลียอัตโนมัติเมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัพต์ของสัญญาณ $\overline{INT1}$ ซึ่งอยู่ในตำแหน่งเริ่มต้นที่ 0013H

IT1: TCON.2 (Interrupt 1 type control bit) เป็นบิตควบคุมการเลือกรูปแบบการแอคทีฟของสัญญาณอินเตอร์รัพต์ INT1 ซึ่งเราสามารถเซตหรือเคลียด้วยซอฟต์แวร์ ดังนี้

1 = เลือกใช้สัญญาณอินเตอร์รัพต์ INT1 แอคทีฟที่การเปลี่ยนจาก 1 ไป 0

2 = เลือกใช้สัญญาณอินเตอร์รัพต์ INT1 แอคทีฟที่ลอจิก 0

IE0: TCON.1 (External Interrupt 0 edge flag) เป็นแฟลกการร้องขออินเตอร์รัพต์จากสัญญาณ $\overline{INT0}$ บิตนี้จะถูกเซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณอินเตอร์รัพต์เข้ามาที่ขา $\overline{INT0}$ และบิตนี้จะถูกเคลียอัตโนมัติเมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัพต์ของสัญญาณ $\overline{INT0}$ ซึ่งอยู่ในตำแหน่งเริ่มต้นที่ 0003H

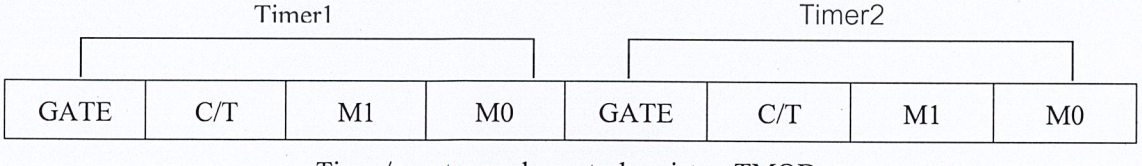
IT0: TCON.0 (Interrupt 0 type control bit) เป็นบิตควบคุมการเลือกรูปแบบการแอคทีฟของสัญญาณอินเตอร์รัพต์ INT0 ซึ่งเราสามารถเซตหรือเคลียด้วยซอฟต์แวร์

1 = เลือกใช้สัญญาณอินเตอร์รัพต์ INT0 แอคทีฟที่การเปลี่ยนจาก 1 ไป 0

2 = เลือกใช้สัญญาณอินเตอร์รัพต์ INT0 แอคทีฟที่ลอจิก 0

2.3.2 รีจิสเตอร์ TMOD (Timer/Counter Mode Control Register)

TMOD (อยู่ในตำแหน่ง 89H) เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมโหมดการทำงานและสัญญาณควบคุมการทำงานของ Timer0 และ Timer1 รีจิสเตอร์ตัวนี้ไม่สามารถอ้างตำแหน่งแบบบิต การกำหนดค่าทำได้โดยใช้คำสั่งโอนย้ายข้อมูล 8 บิต เช่น MOV 89H,#02 บิตต่างๆ ของรีจิสเตอร์นี้แสดงในรูปที



Timer/counter mode control register: TMOD

ภาพที่ 2.7 รีจิสเตอร์ TMOD

GATE: (Gating control when set) เป็นบิตสำหรับเลือกการควบคุมการทำงานของ Timer 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 = การทำงานของ Timer0 (หรือ Timer1) จะถูกควบคุมจากสัญญาณที่ขา INT0 (หรือ INT1 หากสัญญาณที่ขา INT0 (หรือ INT1) เป็น 1 Timer จะทำงาน (ค่าของบิต TR0 หรือ TR1 ในรีจิสเตอร์ TCON จะต้องกำหนดให้เป็น 1 ด้วย)

0 = การทำงานของ Timer0 (หรือ Timer) จะถูกควบคุมจากบิต TR0 (หรือ TR1)

C/T: เป็นบิตสำหรับเลือกโหมดการทำงานของ Timer ให้ทำงานในโหมดตัวตั้งเวลาหรือโหมดการนับ

1 = เลือกใช้งาน Timer ในลักษณะของการนับโดยสัญญาณอินพุตของการนับจะรับมาจากขา T0 (หรือ T1)

0 = เลือกใช้งาน Timer ในลักษณะของตัวตั้งเวลา โดยสัญญาณเวลาอินพุตจะมาจากสัญญาณนาฬิกาภายในตัวไมโครคอนโทรลเลอร์

M1,M0: เป็นบิตสำหรับเลือกโหมดการทำงานของ Timer ซึ่งกำหนดได้ 4 โหมด คือ 0, 1, 2 และ 3 การเลือกโหมดกำหนดได้ดังนี้

ตารางที่ 2.4 การเลือกโหมดของ Timer

M1	M0	โหมด	การทำงาน
0	0	0	8 Bit timer/counter with TL as 5-bit prescaler
0	1	1	16 Bit timer/counter
1	0	2	8 Bit auto-reload timer/counter
1	1	3	(Timer0) TL is 8 Bit timer/counter. TH0 is 8 Bit timer only controlled by timer1 control bit.
1	1	3	(Timer1) Timer/counter 1 stooped

2.3.3 รหัสการกำหนดโหมดการทำงานของ Timer0

รหัสสำหรับควบคุมการทำงานของ Timer0 และ Timer1 กำหนดได้จากตารางต่อไปนี้ หากต้องการกำหนดการทำงานพร้อมกันให้นำค่าในตารางของ Timer0 และ Timer1 มา OR กัน ตัวอย่างเช่น ต้องการให้ timer 0 เป็นตัวตั้งเวลาในโหมด 1 GATE (ควบคุมการทำงานด้วยสัญญาณภายนอกที่ขา INT0) และ Timer 1 ทำงานเป็นตัวนับในโ หมด 2 ควบคุมการทำงานด้วยซอฟต์แวร์ ค่าที่ต้องกำหนดให้กับรีจิสเตอร์ TMOD คือ 09H OR 60H ได้เท่ากับ 69H หลังจากกำหนดค่าให้กับ TMOD แล้ว Timer จะยังไม่ทำงานจนกว่าจะมีการเซตบิต TR (ที่อยู่ในรีจิสเตอร์ TCON) ให้เป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timer0 ทำงานในโหมดตัวจับเวลา:

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL[1]	EXTERNAL CONTROL[2]
0	13-bit timer	00H	08H
1	16-bit timer	01H	09H
2	8 bit auto reload	02H	0AH
3	two 8 bit timers	03H	0BH

Timer0 ทำงานในโหมดตัวนับ:

MODE	COUNTER 0 FUNCTION	TMOD	
		INTERNAL CONTROL[1]	EXTERNAL CONTROL[2]
0	13-bit timer	04H	0CH
1	16-bit timer	05H	0DH
2	8 bit auto reload	06H	0EH
3	one 8 bit timers	07H	0FH

1 ควบคุมด้วยซอฟต์แวร์ หมายถึงการ ON/OFF ตัว Timer ทำได้โดยการเซต หรือเคลียร์บิต TR0 ที่อยู่ในรีจิสเตอร์ TCON

2. ควบคุมด้วยฮาร์ดแวร์ หมายถึง การ ON/OFF ตัว Timer ทำได้โดยการเปลี่ยนสัญญาณที่ขา $\overline{\text{INT0}}$ หากเป็น 1 = ON เมื่อบิต TR0 = 1

2.3.4 รหัสการกำหนดโหมดการทำงานของ Timer 1

รหัสกำหนดโหมดการทำงานของ Timer1 มีลักษณะเช่นเดียวกับรหัสกำหนดโหมดการทำงานของ Timer0

Timer1 ทำงานในโหมดตัวจับเวลา:

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL[1]	EXTERNAL CONTROL[2]
0	13-bit timer	00H	80H
1	16-bit timer	01H	90H
2	8 bit auto reload	02H	A0H
3	does not run	03H	B0H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

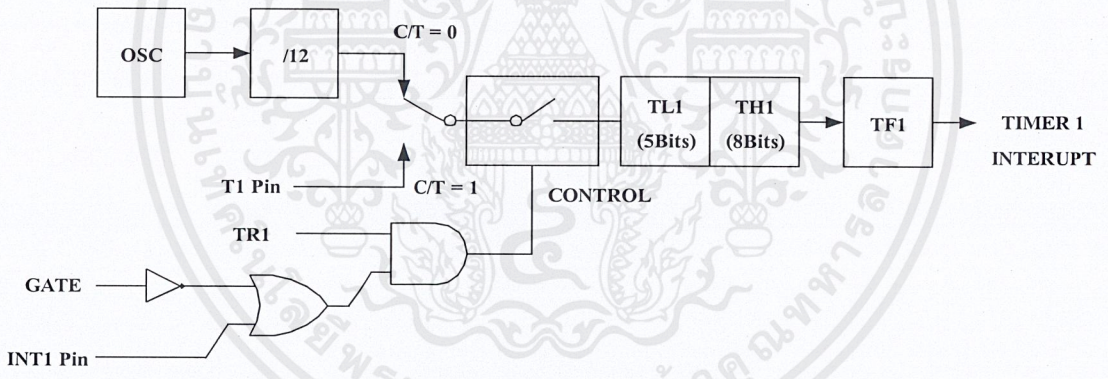
Timer1 ทำงานในโหมดตัวนับ:

MODE	COUNTER 0 FUNCTION	TMOD	
		INTERNAL CONTROL[1]	EXTERNAL CONTROL[2]
0	13-bit counter	40H	C0H
1	16-bit counter	50H	D0H
2	8 bit auto reload	60H	E0H
3	not available	--	--

1. ควบคุมด้วยซอฟต์แวร์ หมายถึง การ ON/OFF ตัว Timer ทำโดยการเซตหรือเคลียบิต TR1 ที่อยู่ในรีจิสเตอร์ TCON

2. ควบคุมด้วยฮาร์ดแวร์ หมายถึง การ ON/OFF ตัว Timer ทำโดยเปลี่ยนสัญญาณที่ขา $\overline{INT1}$ หากเป็น 1 = ON เมื่อ GR1 = 1

2.3.5 การทำงานของ Timer ในโหมด 0



ภาพที่ 2.8 การทำงานของ Timer/Counter1 ในโหมด 0 (13 บิต)

การทำงานในโหมด 0 จะทำงานในลักษณะของตัวนับหรือตัวจับเวลาก็ได้ โดยค่าของการนับสูงสุดคือ 13 บิต ที่มาจาก 5 บิต ต่ำของรีจิสเตอร์ TL และ 8 บิตของรีจิสเตอร์ TH ค่าของการนับสูงสุดคือ 2^{13} = 8192 หรือ 2000H (เมื่อกำหนดค่าเริ่มต้นของรีจิสเตอร์ TL = 0 และ TH = 0) การทำงานในโหมด 0 ของ Timer1 แสดงในรูป 2.8

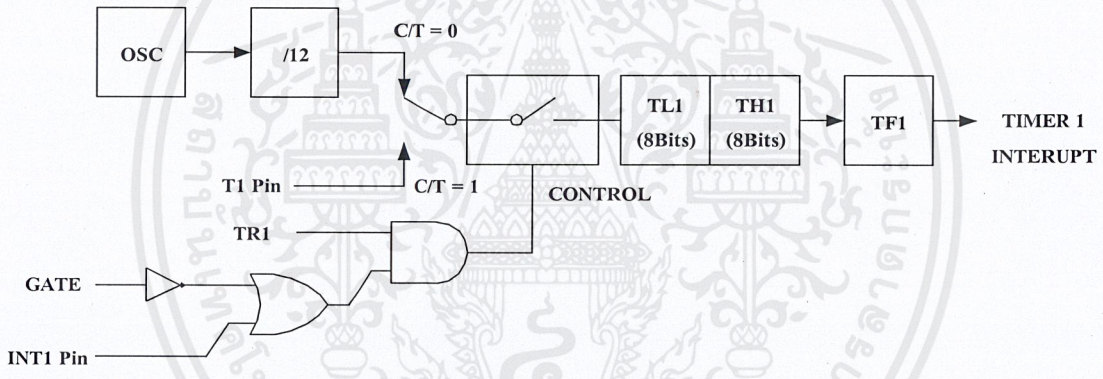
จากรูป 2.8 Timer1 ทำงานได้เมื่อเรากำหนดค่าของ TR1 = 1 (TR เป็นบิตอยู่ในรีจิสเตอร์ TCON) และ GATE = 0 หรือ INT1 = 1 (การกำหนด GATE = 1 เป็นการเลือกการควบคุมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงาน Timer1 ด้วยสัญญาณ INT1) สัญญาณ GATE กำหนดจากการกำหนดค่าให้กับรีจิสเตอร์ TMOD ค่าของการนับจะเริ่มต้นจากค่าที่อยู่ในรีจิสเตอร์ TH1 และ TL1 ซึ่งเรากำหนดจากคำสั่งทางซอฟต์แวร์ ค่าในรีจิสเตอร์ TL1 ใช้เพียง 5 บิตต่ำเท่านั้น อีก 3 บิตที่เหลือไม่ใช้ การนับจะนับขึ้นทีละ 1 เมื่อค่าในรีจิสเตอร์ TH1 และ TL1 เป็น 1 ทุกบิตแล้วจะวนกลับไปเป็น 0 ทุกบิตเมื่อมีสัญญาณนับเข้ามาอีก 1 ลูก ซึ่งช่วงเวลาของการเปลี่ยนจาก 1 ทุบบิตไปเป็น 0 ทุกบิตจะทำให้ค่าของแฟลก TF1 (Timer Overflow) ถูกเซตให้เป็น 1 แฟลก TF1 นี้จะเป็นสัญญาณร้องขออินเตอร์รัพต์ของ Timer1 ต่อไมโครคอนโทรลเลอร์ และแฟลก TF1 จะถูกเคลียเมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการร้องขออินเตอร์รัพต์ของ Timer1 ซึ่งอยู่ในตำแหน่งเริ่มต้นที่ 001BH

การทำงานของ Timer0 ในโหมด 0 จะมีลักษณะคล้ายกันแต่การควบคุมการทำงานจากสัญญาณภายนอกจะใช้ INT0 แทน INT1 และค่าของการนับจะอยู่ในรีจิสเตอร์ TH0 และ TL0

2.3.6 การทำงานของ Timer ในโหมด 1



ภาพที่ 2.9 การทำงานของ Timer/Counter1 ในโหมด 1 (16 บิต)

การทำงานของ Timer ในโหมด 1 จะมีลักษณะคล้ายกับการทำงานในโหมด 0 แตกต่างกันที่ค่าของการนับสูงสุดคือ 16 บิต ใช้รีจิสเตอร์ TH และ TL ตัวละ 8 บิต การกำหนดค่าการนับกำหนดค่าเริ่มต้นในรีจิสเตอร์ TH และ TL ซึ่งอยู่ในช่วง 0000H-FFFFH หากกำหนดให้ค่าในรีจิส-เตอร์ TH และ TL เป็น 0000 จะได้ค่าของการนับจะเป็น 65536 และหากกำหนดค่าเริ่มต้นในรีจิส-เตอร์ TH และ TL เป็น FFFFH ค่าของการนับคือ 1 ค่าเริ่มต้นของ TH และ TL คิดจากค่า 2's complement ของค่าที่ต้องการนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง การใช้ Timer1 เป็นตัวกำหนดคาบเวลาของสัญญาณพัลส์ความกว้าง 100 ไมโครวินาที ออกที่ขา P1.0 เมื่อไมโครคอนโทรลเลอร์ใช้สัญญาณนาฬิกาความถี่ 12 MHz

หลักการ ใช้ Timer1 ในโหมด 1 ทำงานในลักษณะของการตั้งเวลา มีค่า 100 ไมโครวินาที และควบคุมการทำงานด้วยซอฟต์แวร์ และให้มีการส่งสัญญาณอินเตอร์รัพต์เมื่อครบเวลาแล้วซึ่งจะได้ค่าของการกำหนดในรีจิสเตอร์ TMOD และ TCON ดังนี้

ตารางที่ 2.5 ตารางกำหนดค่าการทำงานของ Timer ในโหมด1

ค่าของบิต	รีจิสเตอร์	การทำงาน
C/T = 0 GATE = 0 M1, M0 = 0 1	TMOD	ทำงานเป็นตัวตั้งเวลา ควบคุมจากบิต TR1 เลือกการทำงานโหมด 1 ค่านับ 16 บิต
TR1 = 1	TCON	ให้ Timer1 ทำงาน
EA = 1 ET1 = 1	IE	ให้ไมโครคอนโทรลเลอร์ตอบรับอินเตอร์รัพต์ให้ Timer1 ส่งการร้องขออินเตอร์รัพต์

สำหรับค่าเริ่มต้นของการนับกำหนดจาก 100 = 64H

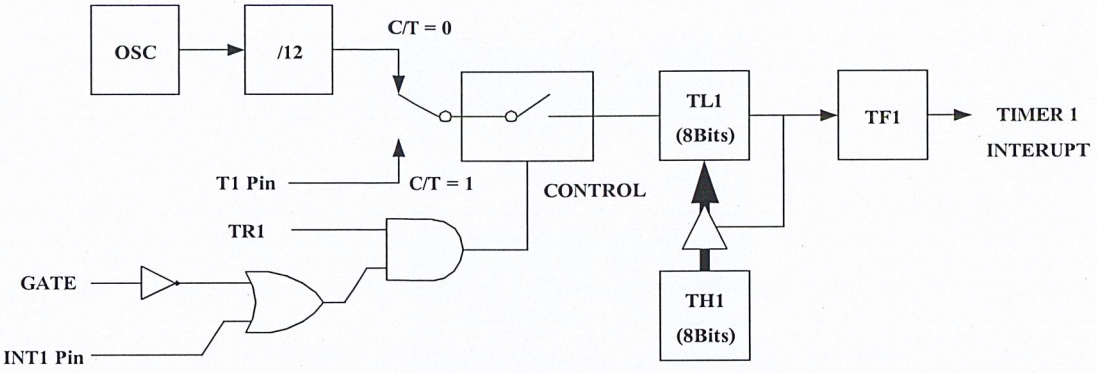
2's complement ของ 64H = FF9CH หรือ 1111 1111 1001 1100

ได้ค่าของรีจิสเตอร์ TH = FFH และรีจิสเตอร์ TL – 9CH

2.3.7 การทำงานของ Timer ในโหมด 2

ในโหมดนี้เป็นแบบโหลดใหม่อัตโนมัติ จำนวนของการนับจะกำหนดในรีจิสเตอร์ TH ค่าของการนับสูงสุดคือ 256 (TH=00) การนับจะใช้ได้เพียง 8 บิต ที่อยู่ในรีจิสเตอร์ TL ซึ่งนำค่าเริ่มต้นมาจากค่าในรีจิสเตอร์ TH เมื่อ Timer ทำงานค่าที่อยู่ใน TL จะถูกเพิ่มขึ้นจนวนกลับไปถึง 00 อีก จะทำให้แฟลก TF0 (หรือ TF1) ถูกเซตเป็น 1 และค่าในรีจิสเตอร์ TH จะถูกนำไปกำหนดให้กับรีจิสเตอร์ TL เพื่อเป็นค่าเริ่มต้นของการนับในรอบต่อไปอีก ซึ่งการทำงานจะเป็นเช่นนี้ตลอดไป แฟลก TF จะถูกเคลียร์อัตโนมัติ เมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองอินเตอร์รัพต์ของ Timer วงจรการทำงานของ Timer1 ในโหมด 2 แสดงอยู่ในรูป 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.10 การทำงานของ Timer/counter1 ในโหมด 2 (8 bit Auto reload)

ตัวอย่าง กำหนดใช้ Timer0 เพื่อสร้างฐานเวลาของการเพิ่มค่าทุกๆ 1 วินาที เมื่อไมโครคอนโทรลเลอร์ใช้สัญญาณนาฬิกาความถี่ 12 MHz

หลักการ ใช้ Timer0 ในโหมด 2 ทำงานในลักษณะของการตั้งเวลาแบบ Auto reload มีค่าการนับ 200 ไมโครวินาที และควบคุมการทำงานด้วยซอฟต์แวร์ และให้มีการส่งสัญญาณอินเตอร์รัพต์ทุกครั้ง เมื่อครบเวลาแล้ว ซึ่งจะได้ค่าที่กำหนดในรีจิสเตอร์ TMOD และ TCON ดังนี้

ตารางที่ 2.6 ตารางกำหนดค่าการทำงานของ Timer ในโหมด2

ค่าของบิต	รีจิสเตอร์	การทำงาน
C/T = 0 GATE = 0 M1,M0 = 1 0	TMOD	ทำงานเป็นตัวตั้งเวลา ควบคุมการทำงานจาก TRO เลือกโหมด 2 Auto reload 8 บิต
TR0 = 1	TCON	ให้ Timer0 ทำงาน
EA = 1 ET0 = 1	IE	ให้ไมโครคอนโทรลเลอร์ตอบรับอินเตอร์รัพต์ ให้ Timer0 ส่งการร้องขออินเตอร์รัพต์

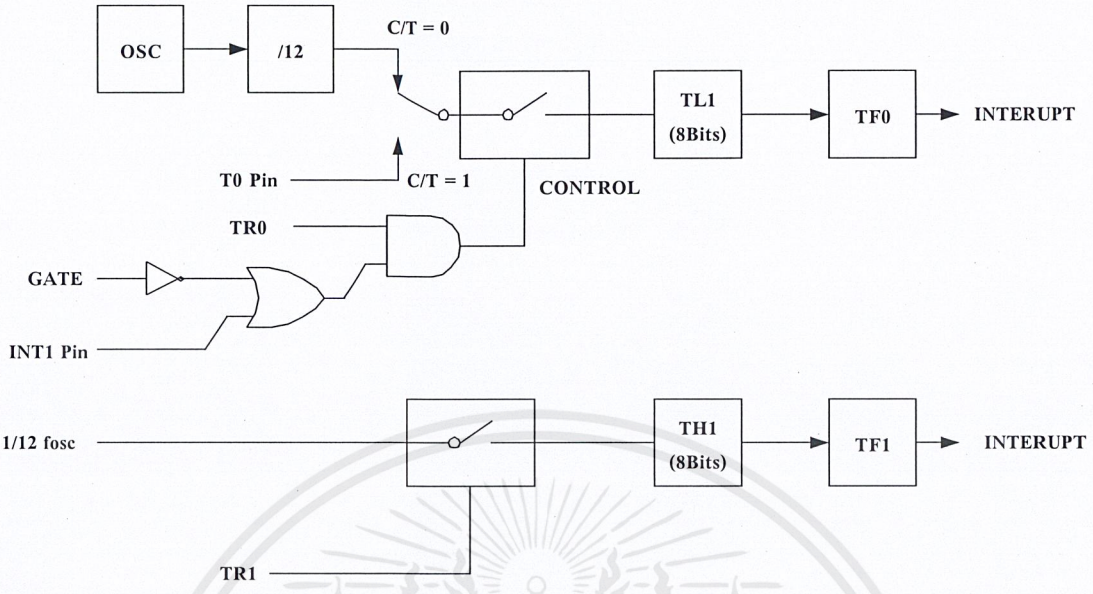
สำหรับค่าเริ่มต้นของการนับ 200 กำหนดจาก 200 = C8H

2's complement ของ C8H = 38H หรือ 00111000B

ได้ค่าเริ่มต้นของรีจิสเตอร์ TH และ TL = 38H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.8 การทำงานของ Timer ในโหมด 3



ภาพที่ 2.11 การใช้งาน Timer0 ในโหมด 3

ในโหมด 3 จะใช้ได้กับ Timer0 เท่านั้นสำหรับ Timer1 ไม่มีการทำงานในโหมด 3 หากกำหนดให้ Timer 1 อยู่ในโหมด 3 จะเป็นลักษณะเดียวกับการกำหนดให้ Timer 1 หยุดทำงาน การทำงานของ Timer 0 ในโหมด 3 จะแยกตัวนับ TH0 และ TL0 ออกจากกัน ทำให้เกิดเป็นตัวนับ 2 ตัวทำงานอิสระต่อกัน ดังแสดงในรูป 2.11

จากรูป 2.11 ตัวนับ TL0 จะใช้บิตควบคุมการทำงานของ Timer0 ดังนี้ C/T, GATE, TR0, INT0 และแฟลก TF0 เป็นตัวควบคุมการทำงาน ส่วน TH0 จะถูกควบคุมให้ทำงานในลักษณะของตัวตั้งเวลาเท่านั้น (นับค่าของแมชชีนไซเกิล) ซึ่งใช้บิต TR1 และ TF1 ของ Timer 1 มาเป็นตัวควบคุมการทำงานเมื่อ TR1 = 1 ตัวนับ TH0 จะทำงาน และเมื่อค่าของ TH0 เกิดโอเวอร์โฟลว์ (TH0 = 00) จะทำให้แฟลก TF = 1 ซึ่งเป็นการร้องขออินเตอร์รัพต์ต่อไมโครคอนโทรลเลอร์ การใช้งานในโหมด 3 จะทำให้ 8051 เหมือนกับมี Timer/counter อยู่ 3 ชุด และใน 8052 เหมือนกับมี Timer/Counter อยู่ 4 ชุด

เมื่อเซตให้ Timer0 ทำงานในโหมด 3 เราสามารถควบคุม Timer1 ให้ ON หรือ OFF ได้ด้วยการเซตให้ Timer1 อยู่ในโหมด 3 เพื่อ OFF และเซตให้อยู่โหมดปกติเมื่อต้องการให้ Timer 1 ON แต่จะไม่มีการอินเตอร์รัพต์ของ Timer 1 หรือใช้ Timer 1 ทำงานในโหมดของการสร้าง Baud rate ให้กับพอร์ตอนุกรมได้ซึ่งจะกล่าวต่อไปในเรื่องการรับส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.9 การใช้งาน Timer2

Timer2 เป็นตัวตั้งเวลา/ตัวนับขนาด 16 บิต ที่มีอยู่เฉพาะในเบอร์ 8052 หรือ 8032 นั้น เราสามารถใช้ Timer2 เป็นตัวจับเวลาหรือตัวนับได้เช่นเดียวกับ Timer0 หรือ Timer1 ภายใน Timer2 มีรีจิสเตอร์ขนาด 8 บิต เพิ่มขึ้นจาก Timer0 และ Timer 1 อยู่ 2 ตัว คือ RCAP2L และ RCAP2H การกำหนดการทำงานของ Timer2 จะกำหนดได้จากรีจิสเตอร์ T2CON (อยู่ในตำแหน่ง C8H) ซึ่งมีบิตต่างๆ ดังนี้

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bot0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$

T2CON: Timer/Counter 2 Control Register

ภาพที่ 2.12 รีจิสเตอร์ T2CON

TF2 (Timer2 overflow flag) เป็นแฟลกของการเกิดโอเวอร์โฟลว โดยแฟลกนี้จะถูกเซตด้วยฮาร์ดแวร์เมื่อเกิดโอเวอร์โฟลวและจะต้องเคลียด้วยซอฟต์แวร์ ซึ่งแฟลกนี้เป็นการร้องขออินเตอร์รัพต์ของ Timer2 หากกำหนดบิต RCLK = 1 หรือ TCLK = 1 ซึ่งมีการทำงานเป็นฐานเวล RCLK (Receive clock flag) เป็นบิตควบคุมเซตหรือเคลียด้วยซอฟต์แวร์ ซึ่งการทำงานเป็นดังนี้

1 = เป็นการนำพัลส์ที่เกิดจากโอเวอร์โฟลว์ของ Timer 2 ไปเป็นสัญญาณนาฬิกา ของการรับข้อมูลของพอร์ตอนุกรมในโหมด 1 และโหมด 3

0 = เป็นการเลือกใช้สัญญาณนาฬิกาของการรับข้อมูลของพอร์ตอนุกรมจากโอเวอร์โฟลวของ Timer1

TCLK (Transmit clock flag) เป็นบิตควบคุม เซตหรือเคลียร์ด้วยซอฟต์แวร์ ซึ่งการทำงานเป็นดังนี้

1 = เป็นการนำพัลส์ที่เกิดจากโอเวอร์โฟลว์ของ Timer 2 ไปเป็นสัญญาณนาฬิกาของการส่งข้อมูลของพอร์ตอนุกรมในโหมด 1 และโหมด 3

0=เป็นการเลือกใช้สัญญาณนาฬิกาของการส่งข้อมูลของพอร์ตอนุกรมจากการเกิดโอเวอร์โฟลว์ของ Timer 1 ของพอร์ตอนุกรม บิตนี้จะม่มีการเซต

EXF2 (External flag set) เมื่อ Timer2 ทำงาน Capture หรือ Auto reload แฟลกนี้จะถูกเซตเมื่อกำหนด EXEN2 = 1 และสัญญาณที่ขา T2EX มีการเปลี่ยนจาก 1 ไป 0 หากกำหนดให้ Timer 2 สามารถร้องขออินเตอร์รัพต์ได้ Timer 2 จะร้องขออินเตอร์รัพต์เมื่อแฟลก EXF2 = 1 ซึ่งจะทำให้ไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองอินเตอร์รัพต์ที่ตำแหน่ง 002BH หลังจากทำโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบสนองอินเตอร์รัพต์แล้วต้องเคลียแฟลก EXF2 ด้วยซอฟต์แวร์ก่อนที่จะออกจากการตอบสนองอินเตอร์รัพต์

RCLK (Receive clock flag) เป็นบิตควบคุม เซตหรือเคลียด้วยซอฟต์แวร์ ซึ่งการทำงานเป็นดังนี้

1 = เป็นการนำพัลส์ที่เกิดจากโอเวอร์โฟลว์ของ Timer 2 ไปเป็นสัญญาณนาฬิกา ของการรับข้อมูลของพอร์ตอนุกรมในโหมด 1 และโหมด 3

0 = เป็นการเลือกใช้สัญญาณนาฬิกาของการรับข้อมูลของพอร์ตอนุกรมจากโอเวอร์โฟลว์ของ Timer 1

TCLK (Transmit clock flag) เป็นบิตควบคุม เซตหรือเคลียร์ด้วยซอฟต์แวร์ ซึ่งการทำงานเป็นดังนี้

1 = เป็นการนำพัลส์ที่เกิดจากโอเวอร์โฟลว์ของ Timer 2 ไปเป็นสัญญาณนาฬิกาของการส่งข้อมูลของพอร์ตอนุกรมในโหมด 1 และโหมด 3

0 = เป็นการเลือกใช้สัญญาณนาฬิกาของการส่งข้อมูลของพอร์ตอนุกรมจากการเกิดโอเวอร์โฟลว์ของ Timer 1

EXEN2 (Timer2 external enable flag) หากกำหนด EXEN2 = 1 จะทำให้การโหลดใหม่หรือแคปเจอร์ เกิดขึ้นเมื่อมีการเปลี่ยนแปลงสัญญาณที่ขา T2EX จาก 1 ไป 0 และไม่มีการใช้ Timer2 เป็นตัวกำเนิดสัญญาณนาฬิกาของการรับ-ส่งข้อมูลขอพอร์ตอนุกรม หากกำหนด EXEN2 = 0 จะมีการทำงานคล้ายกับการกำหนด EXEN2 = 1 แต่ไม่มีการใช้งานที่ขา T2EX

TR2 เป็นบิตควบคุมการทำงานหรือหยุดทำงานของ Timer2

1 = กำหนดให้ Timer2 ทำงาน

0 = กำหนดให้ Timer2 หยุดทำงาน

C/$\overline{T2}$ เป็นบิตเลือกโหมดการทำงานของ Timer2 ในโหมดตัวจับเวลาหรือตัวนับ

0 = ทำงานในโหมดตัวจับเวลา (OSC/12)

1 = ทำงานในโหมดตัวนับ (นับขอบขาลง)

CP/$\overline{RL2}$ (Capture/Reload flag) เป็นตัวเลือกโหมดการทำงานกำหนดจากโปรแกรม

1 = ทำงานในโหมดแคปเจอร์ ซึ่งการแคปเจอร์จะเกิดขึ้นเมื่อบิต EXEN2 = 1 และสัญญาณที่ขา T2EX เปลี่ยนจาก 1 เป็น 0

0 = ทำงานในโหมดการโหลดใหม่ ซึ่งการโหลดใหม่จะเกิดขึ้นเมื่อเกิดโอเวอร์โฟลวของ Timer2 หรือเมื่อกำหนดบิต EXEN2 = 1 แล้วมีสัญญาณอินพุตที่ขา T2EX เปลี่ยนจาก 1 เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามีการกำหนดให้บิต RCLK = 1 หรือ TCLK = 1 บิตนี้จะไม่มีการใช้งาน และการโหลดใหม่ของ Timer2 จะเกิดขึ้นเมื่อเกิดโอเวอร์โฟลวของ Timer2 อัตโนมัติ การทำงานของ Timer2 แบ่งได้เป็น 3 โหมด คือ Capture, Auto Reload และ Baud rate generator การกำหนดโหมดการทำงานจะกำหนดจากบิตต่างๆ ในรีจิสเตอร์ T2CON ดังแสดงในตารางที่ 2.7

ตารางที่ 2.7 ตารางกำหนดโหมดการทำงานของ Timer 2

RCLK+TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto Reload
0	1	1	16-bit Capture
1	x	1	Baud Rate Generator
x	x	0	Off

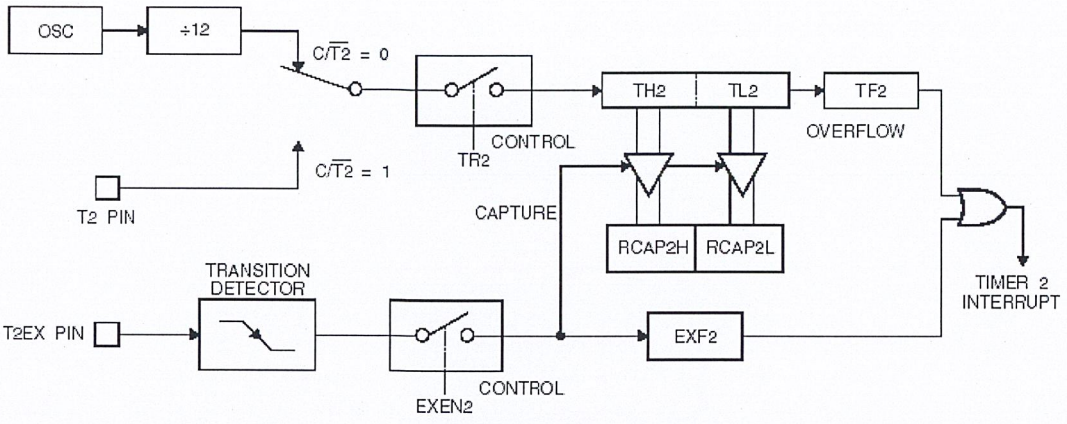
2.3.10 การทำงานของ Timer2 ในโหมดแคปเจอร์ (Capture)

การทำงานในโหมดแคปเจอร์ สามารถเลือกการทำงานได้ 2 ลักษณะจากการกำหนดค่าบิต EXEN2 ที่อยู่ในรีจิสเตอร์ T2CON ดังนี้

ถ้ากำหนดบิต EXEN2 = 0 การทำงานของ Timer2 จะเป็นตัวจับเวลาหรือตัวนับแบบ 16 บิต โดยแฟลก TF2 จะถูกเซตเมื่อเกิดโอเวอร์โฟลว ซึ่งแฟลก TF2 นี้จะเป็นแฟลกการร้องขออินเตอร์รัพต์ของ Timer2

ถ้ากำหนดบิต EXEN2 = 1 การทำงานจะคล้ายกันแต่มีส่วนเพิ่มเติมคือ เมื่อสัญญาณที่ขา T2EX ซึ่งเป็นสัญญาณอินพุตจากภายนอกเปลี่ยนจาก 1 ไป 0 จะทำให้ค่าที่อยู่ในรีจิสเตอร์ของการนับ TL2 และ TH2 ถูกโหลดไปเก็บในรีจิสเตอร์ RCAP2L และ RCAP2H ตามลำดับ (RCAP2L และ RCAP2H เป็นรีจิสเตอร์ที่ทำงานเฉพาะซึ่งมีอยู่ในเบอร์ 8052 และ 8032) นอกจากนี้การเปลี่ยนจาก 1 ไป 0 ของสัญญาณที่ขา T2EX ยังทำให้แฟลก EXF2 ในรีจิสเตอร์ T2CON ถูกเซตด้วย ซึ่งแฟลก EXF2 ก็ทำให้เกิดการร้องขออินเตอร์รัพต์ได้เช่นเดียวกับแฟลก TF2 แต่แฟลก EXF2 ต้องเคลียร์ด้วยซอฟต์แวร์ การทำงานของ Timer2 ในโหมดแคปเจอร์แสดงอยู่ในภาพที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



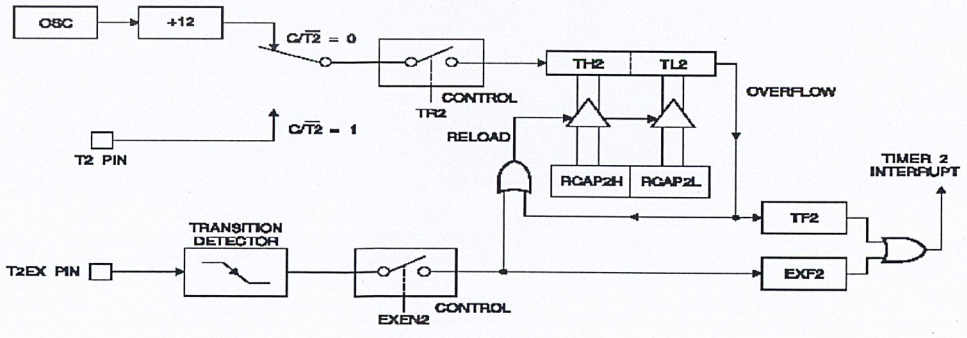
ภาพที่ 2.13 การทำงาน Timer2 ในโหมดแคปเจอร์

2.3.11 การทำงาน Timer2 ในโหมดโหลดใหม่ (Auto Reload)

การทำงานในโหมดนี้เราสามารถเลือกการทำงานได้ 2 แบบ เช่นกัน ซึ่งการเลือกกำหนดจากบิต EXEN2 ที่อยู่ในรีจิสเตอร์ T2CON ดังนี้

หากกำหนดบิต EXEN2 = 0 เมื่อค่าของการนับ (หรือค่าเวลา) เกิดโอเวอร์โฟลว์ขึ้น นอกจากจะเซตแฟลก TF2 แล้วยังทำให้ค่าในรีจิสเตอร์ RCAP2L และ RCAP2H ถูกนำไปกำหนดให้กับรีจิสเตอร์ TL2 และ TH2 เพื่อเป็นค่าเริ่มต้นของการนับครั้งต่อไปด้วย

หากกำหนดบิต EXEN2 = 1 เมื่อค่าของการนับ (หรือค่าเวลา) เกิดโอเวอร์โฟลว์ขึ้น นอกจากจะเซตแฟลก TF2 และนำค่าในรีจิสเตอร์ RCAP2L และ RCAP2H ไปกำหนดให้กับรีจิสเตอร์ TL2 และ TH2 การโหลดค่าจากรีจิสเตอร์ RCAP2L และ RCAP2H ไปกำหนดให้กับรีจิสเตอร์ TL2 และ TH2 ยังสามารถควบคุมได้จากสัญญาณที่ขา T2EX อีกด้วย โดยการโหลดจะกระทำเมื่อสัญญาณที่ขา T2EX เปลี่ยนจาก 1 ไป 0 และจะทำให้แฟลก EXF2 ถูกเซตด้วย ซึ่งแฟลก EXF2 เป็นสัญญาณร้องขอการอินเตอร์รัพต์ต่อไป



ภาพที่ 2.14 การทำงาน Timer2 ในโหมด Auto Reload

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.12 การทำงานในโหมดสร้างอัตรารับส่งข้อมูล (Baud rate generator)

การทำงานในโหมดนี้ใช้ Timer2 เป็นตัวสร้างอัตรารับส่งข้อมูลให้กับพอร์ตอนุกรม การกำหนดการทำงานจะกำหนดโดยเซตบิต RCLK = 1 และ/หรือ TCLK = 1 ที่อยู่ในรีจิสเตอร์ T2CON สำหรับการใช้งาน Timer2 ในโหมดนี้ จะขอนำไปกล่าวในเรื่องการรับส่งข้อมูลแบบอนุกรมซึ่งอยู่ในบทต่อไป

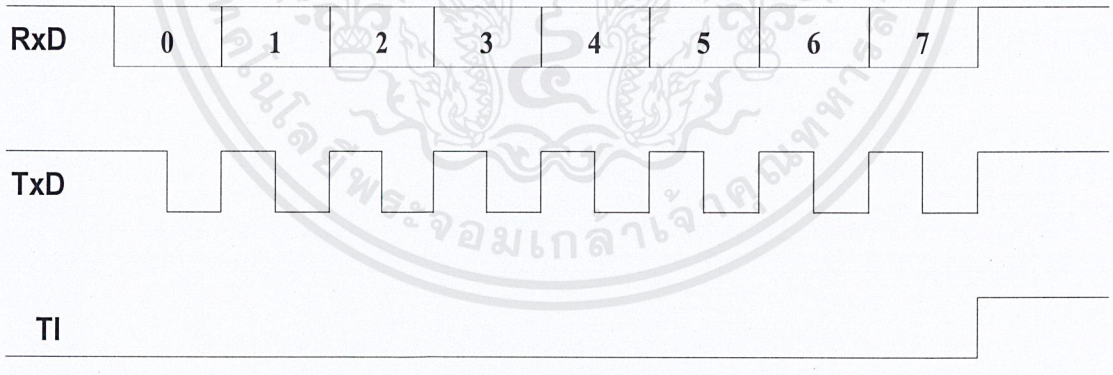
2.4 การใช้งานพอร์ตสื่อสารอนุกรม แบบ Single Processor

หลักการของพอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรมมีโครงสร้างการทำงานเป็นแบบที่เรียกว่าฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านส่งใช้ขา TxD (พอร์ต 3.1) ทางด้านรับใช้ขา RxD (พอร์ต 3.0) SBUF ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรม

พอร์ตสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่บิต SM0 และ SM1 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรมมีดังนี้

โหมด 0 : พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งจะเลื่อนออกทีละบิต โดยส่งบิต D0 ออกไปก่อนทางขา RxD และไม่มีการส่ง Start bit แต่จะส่ง Shift clock ทางขา TxD (ความเร็ว 1/12 เท่าของ CPU Clock)



ภาพที่ 2.15 Timing Diagram ของการสื่อสารอนุกรม

โหมด 1 : พอร์ตสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 Start bit และ 1 Stop clock และสามารถเปลี่ยนแปลง (ความเร็วในการส่งข้อมูลได้โดยขึ้นกับบิต SMOD ใน PCON และ อัตราโอเวอร์โฟลว์ของ Timer 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

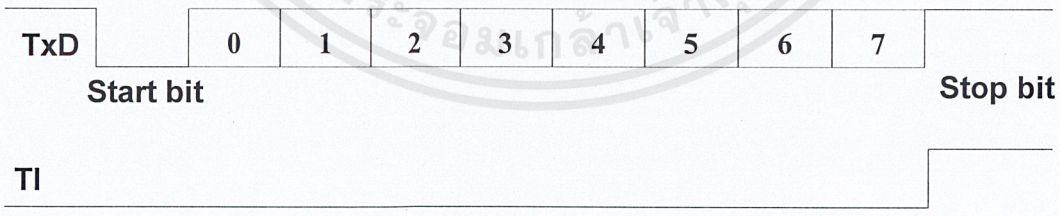
$$\text{Baud Rate Mode1,3} = \frac{2^{\text{smod}} \text{ x Oscillator}}{32\text{x}12\text{x}[256-(\text{TH1})]}$$ โดยใช้ Timer 1

$$\text{Baud Rate Mode 1,3} = \frac{\text{Oscillator Freq.}}{32\text{x}[65536-\text{P2H,RCAP3L})]}$$ โดยใช้ Timer 2

ภาพที่ 2.16 Timing Diagram ของการสื่อสารอนุกรมในโหมด 1

โหมด 2 : พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 Start bit และ 1 Stop clock (TB8 นิยมนำมาใช้ส่ง Parity bit) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU Clock โดยขึ้นกับบิต SMOD ใน PCON

- Baud Rate Mode 2 = (1/32) (Osc Freq) เมื่อ SMOD = 1
- Baud Rate Mode 2 = (1/64) (Osc Freq) เมื่อ SMOD = 0



ภาพที่ 2.17 Timing Diagram ของการสื่อสารอนุกรมในโหมด 2

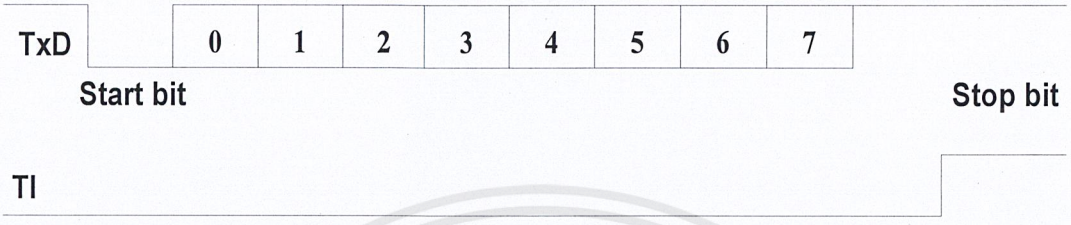
โหมด 3 : พอร์ตสื่อสารอนุกรม 11 บิต UART ข้อมูล 9 บิต 1 Start bit และ 1 Stop clock เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของ Timer 1,2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ 8051 หรืออัตราโอเวอร์โฟลว์ของ Timer 2 (สำหรับ 80C154D)

$$\text{Baud Rate Mode 3} = \frac{2^{smod} \times \text{Oscillator Freq.}}{32 \times 12 \times [256-(TH1)]}$$ โดยใช้ Timer 1

$$\text{Baud Rate Mode 3} = \frac{\text{Oscillator Freq}}{32 \times [65536-(RCAP2H,RCAP3L)]}$$ โดยใช้ Timer 2



ภาพที่ 2.18 Timing Diagram ของการสื่อสารอนุกรมในโหมด 3

การเชื่อมต่อไมโครโปรเซสเซอร์เพื่อรับส่งข้อมูลอนุกรมมี 2 โหมด คือ

- Single Processor Mode
- Multiprocessor Mode

Single Processor Mode : ในโหมดนี้เราจะใช้ไมโครคอนโทรลเลอร์ 2 ตัวเชื่อมเข้าหากัน

Multiprocessor Mode : ในโหมดนี้เราจะใช้ไมโครคอนโทรลเลอร์ 1 ตัว เป็นตัวแม่ (Master) และอีก 0-256 ตัวลูก (Slave) รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลอนุกรม Serial Control Port Register (SCON) อยู่ใน SFR ตำแหน่ง (98H)

SCON : Serial Control Port Register (Bit Addressable)

SM0	SM1	SM2	REN	TN8	RB8	TI	RI

SM0 SCON.7 Serial Port mode specifier (Note 1)

SM1 SCON.6 Serial Port mode specifier (Note 1)

SM2 SCON.5 Enables the multiprocessor communication feature in mode 2 & 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit is 0. In mode 1 ,if SM2=1 then RI will not be activated if a valid stop bit was not received. In mode 0 ,SM2 should be 0.

REN SCON.4 Set/Cleared by software to Enable/Disable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TB8 SCON.3 The 9th bit that will be transmitted in modes 2 & 3.

RB8 SCON.2 In modes 2 & 3 is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is a stop bit that was received. In mode 0, RB8 is not used.

TI SCON.1 Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0 ,or at the beginning of the stop bit in the other modes. Must be cleared by software.

RI SCON.0 Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0 ,or haft way through the stop bit time in the other modes. Must be cleared by software.

Note1 :

SM0	SM1	SM2	DESCRIPTION	BAUD RATE
0	0	0	SHIFT REGISTER	Fosc/12
0	1	1	8 BIT UART	Variable
1	0	2	9 BIT UART	Fosc/64 OR Fosc/32
1	1	3	9 BIT UART	Variable

Serial Port SET-UP

MODE	SCON	SM2 VARIATION
0	10H	
1	50H	Single Processor
2	90H	Environment
3	D0H	(SM2=0)
0	NA	
1	NA	Multiprocessor
2	BOH	Environment
3	FOH	(SM2=0)

SM0,SM1 บิตเลือกโหมดการทำงาน

SM0	SM1	MODE	การทำงาน
0	0	0	Shift register อัตราเร็วในการรับส่งข้อมูลเท่ากับ 1/12 ของ CPU osc
0	1	1	8 bit UART อัตราเร็วในการรับส่งข้อมูลกำหนดจาก Timer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			1,2
1	0	2	9 bit UART อัตราเร็วในการรับส่งข้อมูล=1/32 หรือ 1/64 ของความถี่ ออสซิลเลเตอร์ ขึ้นกับบิต SMOD ใน PCON
1	1	3	9 bit UART อัตราเร็วในการรับส่งข้อมูลกำหนดจาก Timer 1,2

SM2 บิตเลือกการทำงานแบบ Single Processor Mode หรือ Multiprocessor Mode

1 : เลือก Multiprocessor Mode ใช้ได้กับโหมด 2,3

0 : เลือก Single Processor Mode ใช้ได้กับทุกโหมด

เมื่อเลือกการทำงานรับข้อมูลแบบ แล้ว

ถ้าข้อมูลบิตที่ 9 ที่รับได้มีค่าเป็น 1 RI จะเซ็ท

ถ้าข้อมูลบิตที่ 9 ที่รับได้มีค่าเป็น 0 RI จะไม่เซ็ท

REN บิตควบคุมให้รับหรือไม่รับข้อมูล

1 : ให้รับข้อมูลได้

0 : ห้ามรับข้อมูล

TB8 ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2,3 ให้ใช้ในบิตนี้ได้เลย

RB8 ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมาเก็บในบิตนี้

TI แฟลกซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI แฟลกซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์

ตารางที่ 2.8 ตารางการใช้ Timer 1 กำหนด บอดเรท

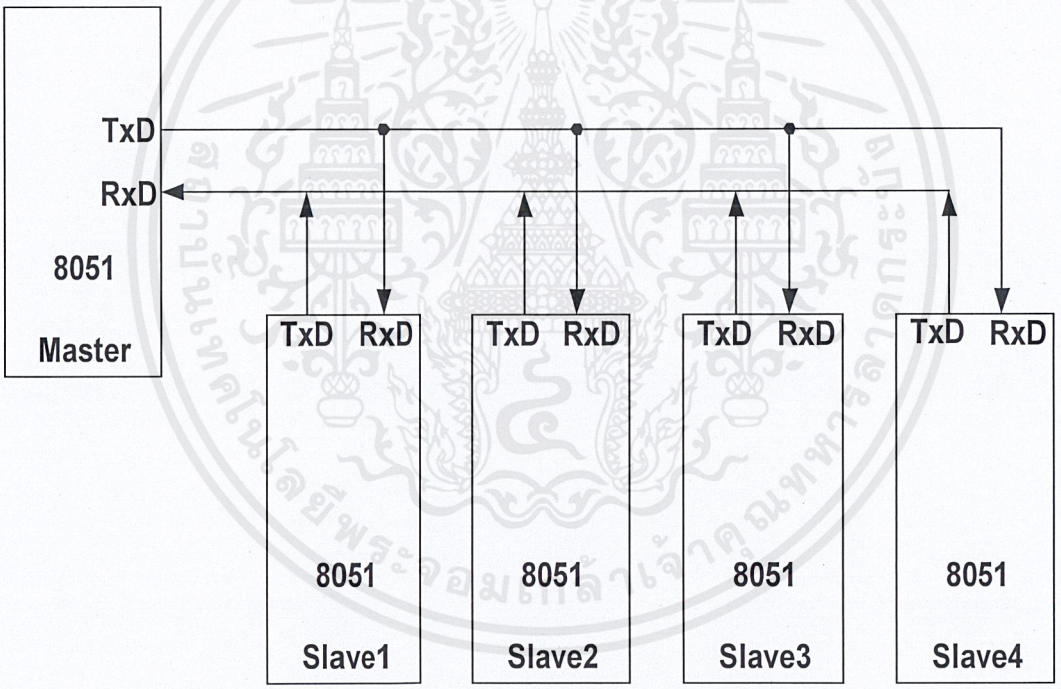
BAUD RATE	FOSC	SMOD	Timer 1		
		ใน PCON	C/T	MODE	Reload Value
(Mode 0) Max : 1 MHz	12 MHz	X	X	X	X
(Mode 2) Max : 375 kHz	12 MHz	1	X	X	X
(Mode 2) Min : 187.5 kHz	12 MHz	0	X	X	X
Mode 1 , 3 : 62.5 k	12 MHz	1	0	2	FFH
19.2 k	11.059 MHz	1	0	2	FDH
9.6 k	11.059 MHz	0	0	2	FDH
4.8 k	11.059 MHz	0	0	2	FAH
2.4 k	11.059 MHz	0	0	2	F4H
1.2 k	11.059 MHz	0	0	2	E8H
137.5	11.059 MHz	0	0	2	1DH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

2.5 การสื่อสารระหว่าง CPU หลายตัว (Multi Processor Communication)

การติดต่อสื่อสารทั่วไป จะใช้การติดต่อสื่อสารแบบจุดต่อจุดซึ่งมีตัวรับ 1 ตัว และตัวส่ง 1 ตัวแต่ในบางครั้งการควบคุมจำเป็นต้องทำการสื่อสารแบบหลายจุด ซึ่งเป็นการสื่อสารระหว่างไมโครคอนโทรลเลอร์หลายตัวที่ต่อเชื่อมเข้าด้วยกัน เพื่อให้เกดการควบคุมที่มีประสิทธิภาพยิ่งขึ้น ไมโครคอนโทรลเลอร์ เบอร์ 8051 สามารถเชื่อมต่อเพื่อรับส่งข้อมูลกันได้หลายตัว ดังแสดงในภาพที่ 2.19



ภาพที่ 2.19 การเชื่อมต่อไมโครคอนโทรลเลอร์จุดต่อจุดและหลายจุด

การทำงานของพอร์ตอนุกรม ไมโครคอนโทรลเลอร์ MSC-51 ในโหมด 2 และ โหมด 3 มีการทำงานพิเศษ ที่สามารถใช้สำหรับการเชื่อมต่อสื่อสารระหว่างซีพียูหลายตัวได้ การทำงานใน 2 โหมดนี้ ข้อมูลบิตที่ 9 จะถูกนำไปเก็บในบิต RB8 เราสามารถโปรแกรมเพื่อกำหนดให้พอร์ตอนุกรมส่งสัญญาณร้องขออินเตอร์รัพต์ได้เมื่อรับข้อมูลเข้ามาแล้วได้ค่าใน RB8 = 1 ซึ่งการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้กำหนดจากการเซตบิต SM2 ที่อยู่ในรีจิสเตอร์ SCON การทำงานในการเชื่อมต่อสื่อสารกับซีพียูหลายตัวจะมีลักษณะการทำงานดังนี้

ในระบบที่เชื่อมต่อแบบหลายจุด เรากำหนดให้ซีพียูตัวหนึ่งเป็น ซีพียูหลัก (Master) สำหรับควบคุมการติดต่อกับซีพียูตัวอื่นๆ ซึ่งเรียกว่าเป็นซีพียูรอง (SLAVE)

สัญญาณของซีพียูหลักต่อเข้ากับสัญญาณของซีพียูรองทุกตัว ทำให้ข้อมูลจากซีพียูหลักสามารถส่งไปยังซีพียูรองได้ทุกตัว ที่ซีพียูรองทุกตัวจะมีจะมีตำแหน่งเป็นของตัวเองที่กำหนดไว้จากโปรแกรมหรือฮาร์ดแวร์ โดยมีการแบ่งลักษณะของข้อมูลที่ใช้ส่งออกเป็น 2 แบบ คือ

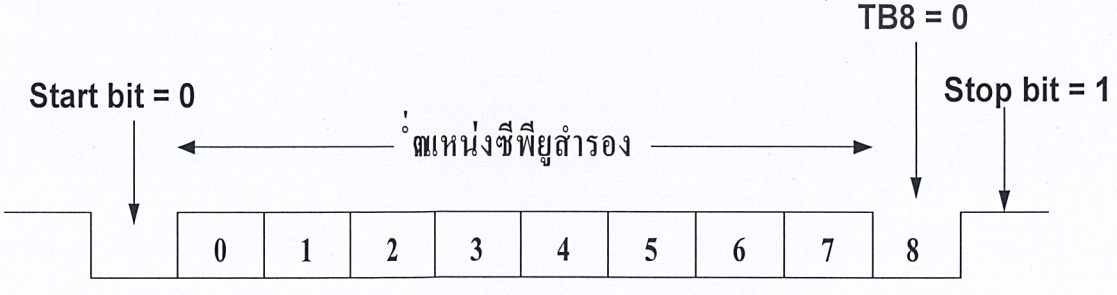
1. ข้อมูลที่เป็นตัวกำหนดตำแหน่ง (Address Byte)

ข้อมูลที่เป็นตัวกำหนดตำแหน่งจะมีค่าข้อมูลบิตที่ 9 มีค่าเป็น 1 ใช้สำหรับการกำหนดตำแหน่งของซีพียูรอง ที่หลักต้องการติดต่อด้วย ตำแหน่ง ข้อมูลแสดงในภาพที่ 2.20

ภาพที่ 2.20 ข้อมูลกำหนดตำแหน่ง (Address Byte)

2. ข้อมูลที่ใช้เป็นข้อมูลข่าวสาร (Data Byte)

เป็นไบต์ที่ใช้เป็นข้อมูลสำหรับการรับและส่งข้อมูล และนำไปใช้งานโดยข้อมูลไบต์นี้กำหนดให้บิตที่ 9 เป็น 0 ซึ่งรูปแบบของไบต์ข้อมูลจะเป็นดังภาพที่ 2.21



ภาพที่ 2.21 ข้อมูลข่าวสาร (Data Byte)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อซีพียูหลักต้องการส่งข้อมูล 1 บล็อก ไปให้กับซีพียูรองตัวใดตัวแหน่งในหลายๆตัว

2.6 การเชื่อมต่อกับเครื่องพิมพ์

เครื่องพิมพ์ที่ใช้ต่อร่วมกับเครื่องคอมพิวเตอร์ในระดับไมโครคอมพิวเตอร์และคอมพิวเตอร์ส่วนใหญ่จะเป็นเครื่องพิมพ์แบบ DOT MATRIX คือลักษณะของตัวอักษรจะเกิดจากการสร้างจุดหลายๆจุดต่อเนื่องกันจำนวนจุดที่พิมพ์ในหนึ่งตัวอักษรจะมีจำนวนแตกต่างกันออกไปขึ้นอยู่กับชนิดของเครื่องพิมพ์ ถ้าหากเครื่องพิมพ์เป็นแบบ NLQ (Near letter quality) หรือ LQ (Letter quality) จำนวนจุดหนึ่งในตัวอักษรจะมีจำนวนมากขึ้นซึ่งทำให้ตัวอักษรที่เกิดจากการพิมพ์มีความสวยงามมากกว่าเครื่องพิมพ์ธรรมดาความเร็วในการพิมพ์ของเครื่องพิมพ์มีขนาดไม่เท่ากันหน่วยความเร็วในการพิมพ์จะเป็นจำนวนตัวอักษรต่อวินาที CPS (Character per second) โดยทั่วไปเครื่องพิมพ์จะมีความเร็วประมาณ 80-250 CPS ในโหมดธรรมดา (Draft Mode) เครื่องพิมพ์ที่เป็นแบบ LQ และ NLQ จะมีการพิมพ์ในหมดของ LQ และ NLQได้ด้วยซึ่งในโหมดนี้จะได้ตัวอักษรที่สวยงามขึ้น แต่จะได้ความเร็วลดลงไป

2.6.1 สัญญาณต่างๆของเครื่องพิมพ์

การเชื่อมต่อของสัญญาณระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ สามารถทำได้ 2 แบบ คือ แบบขนานและแบบอนุกรมขึ้นอยู่กับเครื่องพิมพ์ในที่นี้จะขอกล่าวถึงการส่งข้อมูลแบบขนาน ซึ่งจะเป็นที่นิยมมากกว่าแบบอนุกรม

การส่งข้อมูลแบบขนานระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์มีมาตรฐานที่นิยมมากที่สุดคือมาตรฐานของ Centronics ซึ่งมีการใช้กันทั่วไปในเครื่องพิมพ์เกือบทุกชนิดที่มีการรับข้อมูลแบบขนานสัญญาณต่างของเครื่องพิมพ์ซึ่งอยู่ใน Connector ขนาด 36 pin.

หน้าที่ของสัญญาณต่างๆ

STROBE เป็นสัญญาณควบคุมการส่งข้อมูลที่อยู่บนบัสข้อมูลให้เครื่องพิมพ์สัญญาณนี้จะต้องมีความกว้างของพัลซ์ไม่ต่ำกว่า 1 ไมโครวินาที

DATA1-DATA8 เป็นสัญญาณข้อมูลที่ส่งให้กับเครื่องพิมพ์

ACK เป็นสัญญาณพัลซ์ที่ส่งออกจากเครื่องพิมพ์เพื่อบอกให้ทราบว่าขณะนี้ได้รับข้อมูล 1 ไบต์ส่งมาเรียบร้อยแล้ว

BUSY เป็นสัญญาณออกจากเครื่องพิมพ์เมื่อสัญญาณนี้เป็นลอจิก 1 แสดงว่าเครื่องพิมพ์ไม่พร้อมที่จะรับข้อมูล ซึ่งสัญญาณจะมีค่าเป็นลอจิก 1 เมื่อเกิดกรณีเหล่านี้

1. มีการรับข้อมูลเข้ามาเต็มบัฟเฟอร์
2. เครื่องพิมพ์ไม่ได้อยู่ในสภาวะของการติดต่อกับคอมพิวเตอร์ (DESECLECT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เกิดความผิดพลาดกับเครื่องพิมพ์

ตารางที่ 2.9 สัญญาณต่างๆของเครื่องพิมพ์

Pin No (D-Type 25)	Pin No (Centronics)	SPP Signal	Direction In/out	Register	Hardware Inverted
1	1	nStrobe	In/Out	Control	Yes
2	2	Data 0	Out	Data	
3	3	Data 1	Out	Data	
4	4	Data 2	Out	Data	
5	5	Data 3	Out	Data	
6	6	Data 4	Out	Data	
7	7	Data 5	Out	Data	
8	8	Data 6	Out	Data	
9	9	Data 7	Out	Data	
10	10	nAck	In	Status	
11	11	Busy	In	Status	Yes
12	12	Paper-Out PaperEnd	In	Status	
13	13	Select	In	Status	
14	14	nAuto-Linefeed	In/Out	Control	Yes
15	32	nError / nFault	In	Status	
16	31	nInitialize	In/Out	Control	
17	36	nSelect-Printer nSelect-In	In/Out	Control	Yes
18 - 25	19-30	Ground	Gnd		

PE เป็นสัญญาณออกจากเครื่องพิมพ์ ทำหน้าที่แสดงสภาวะของเครื่องพิมพ์สัญญาณนี้จะมีค่าลอจิก 1 เมื่อไม่มีกระดาษพิมพ์ในเครื่องพิมพ์

SELECTเป็นสัญญาณออกจากเครื่องพิมพ์ทำหน้าที่แสดงสภาวะของเครื่องพิมพ์ว่าพร้อมจะติดต่อกับคอมพิวเตอร์หรือไม่ หากมีลอจิก 1 แสดงว่าเครื่องพิมพ์พร้อมที่จะติดต่อกับคอมพิวเตอร์ (SELECT) หากมีลอจิก 1 แสดงว่าเครื่องพิมพ์ไม่พร้อมที่จะติดต่อกับคอมพิวเตอร์ (DESELECT) สัญญาณนี้จะมีค่าลอจิก 1 เมื่อกดปุ่ม SELECT ที่เครื่องพิมพ์หรือเมื่อเครื่องพิมพ์ได้รับรหัส DC1

CHASSIS GND เป็นกราวด์ของแทนเครื่อง

TWISTED PAIR GND เป็นกราวด์ของสัญญาณที่ใช้ร่วมกับสายข้อมูลและสัญญาณควบคุมต่างๆ

ERROR เป็นสัญญาณออกจากเครื่องพิมพ์ เมื่อสัญญาณนี้มีค่าลอจิก 0 แสดงว่าเครื่องพิมพ์ไม่พร้อมที่จะติดต่อสื่อสารกับคอมพิวเตอร์ ซึ่งสัญญาณนี้จะเป็น 0 เมื่อมีความผิดพลาดต่อไปนี้

1. ไม่มีกระดาษในเครื่องพิมพ์
2. เมื่อมีการกดปุ่ม PAUSE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

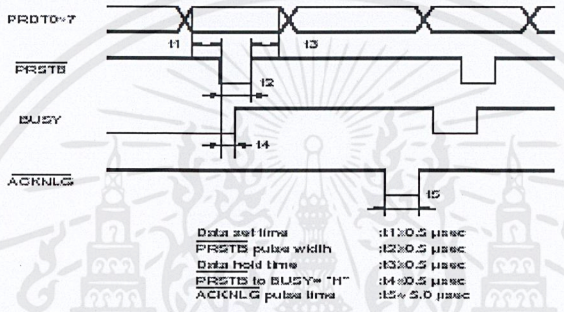
3. เครื่องพิมพ์ทำงานผิดพลาด

INIT เป็นสัญญาณออกจากคอมพิวเตอร์ส่งมาให้เครื่องพิมพ์ เมื่อสัญญาณนี้มีพัลซ์ 0 จะทำให้เครื่องพิมพ์ถูกรีเซตและกลับไปอยู่ในสภาวะเริ่มต้น

SLCT IN เป็นสัญญาณออกจากคอมพิวเตอร์ส่งมาให้เครื่องพิมพ์ เมื่อสัญญาณนี้มีค่าลอจิก 1 เราสามารถควบคุมการติดต่อกับเครื่องพิมพ์ได้ด้วยรหัส DC1 และ DC3 สัญญาณที่ขานี้ปกติจะเป็น 0

AUTO FEED XT เมื่อสัญญาณนี้มีค่าลอจิก 0 เครื่องพิมพ์จะเพิ่ม Line feed หลังจากพิมพ์ในแต่ละบรรทัด

การส่งข้อมูลจากคอมพิวเตอร์ ไปยังเครื่องพิมพ์จะเป็นการตรวจสอบความพร้อม ซึ่งสามารถแสดงไดอะแกรมเวลาได้ดังรูป



ภาพที่ 2.22 ไดอะแกรมการรับข้อมูลของเครื่องพิมพ์

การส่งข้อมูลจากเครื่องคอมพิวเตอร์ไปให้กับเครื่องพิมพ์ไม่พร้อมจะส่งสัญญาณ BUSY ออกมาเป็นลอจิก 1 ดังนั้นก่อนที่คอมพิวเตอร์จะส่งข้อมูลให้กับเครื่องพิมพ์ คอมพิวเตอร์จะตรวจสอบสัญญาณ BUSY ของเครื่องพิมพ์ก่อนว่ามีค่าเป็น 1 หรือไม่หากสัญญาณ BUSY มีค่าเป็น 1 แสดงว่าเครื่องพิมพ์ไม่พร้อมที่จะรับข้อมูล (หากไม่มีการต่อเครื่องพิมพ์สัญญาณจะมีค่าเป็น 1 เช่นกัน) เครื่องคอมพิวเตอร์จะต้องรอจนกว่า BUSY มีค่าลอจิก 0 จึงจะส่งข้อมูลไปให้กับคอมพิวเตอร์เครื่องพิมพ์ได้ การส่งข้อมูลให้กับเครื่องพิมพ์จะส่งไปทาง DATA1-DATA8 แล้วส่งสัญญาณ STROBE ดังแสดงในภาพที่ 2.22 เราจะเริ่มจากการส่งข้อมูลบนสาย DATA1-DATA8 ก่อนแล้วรอเวลาระยะหนึ่งแล้วจึงทำการส่งสัญญาณ STROBE ซึ่งมีความกว้างของสัญญาณไม่ต่ำกว่า 1 ไมโครวินาที เมื่อเครื่องพิมพ์ได้รับสัญญาณ STROBE แล้วจะส่งสัญญาณ BUSY ออกมามีค่าลอจิก 1 เมื่อเครื่องพิมพ์ได้อ่านข้อมูลไปแล้วจะส่งสัญญาณ ACKNLG ออกมาเพื่อเป็นการบอกให้เครื่องคอมพิวเตอร์ทราบว่าได้อ่านข้อมูลไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 รหัสข้อมูลที่ใช้กับเครื่องพิมพ์

การพิมพ์ของเครื่องแบ่งออกได้เป็น 2 ลักษณะคือในโหมดของกราฟฟิกและในโหมดของตัวอักษรASCIIในส่วนนี้จะขอกล่าวเฉพาะในโหมดของตัวอักษรเพียงอย่างเดียวตัวอักษรที่ส่งไปให้กับเครื่องพิมพ์จะใช้รหัส ซึ่งเป็นรหัส 8 บิตที่ใช้แทนตัวอักษรต่างๆดังแสดงในตารางที่ 2.10

ตารางที่ 2.10 รหัส ASCII ที่ใช้กับเครื่องพิมพ์

CODE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL		SP	0	@	P		p								
1		DC1	!	1	A	Q	a	q								
2		DC2	"	2	B	R	b	r								
3		DC3	#	3	C	S	c	s								
4		DC4	$	4	D	T	d	t								
5			%	5	E	U	e	u								
6			&	6	F	V	f	v								
7			'	7	G	W	g	w								
8			(	8	H	X	h	x								
9	HT	EM	)	9	I	Y	i	y								
A	LF		*	:	J	Z	j	z								
B	VT	ESC	+	;	K	[	k	{								
C	FF		,	<	L	\	l	|							
D	CR		-	=	M	]	m	}								
E	SO		.	>	N	^	n	~								
F	SI		/	?	O	_	o	DEL								

จากตารางรหัส ASCII ในตารางที่ 2.10 เราอ่านรหัสจากทางด้านคอลัมน์ (หลักสูง) รวมกับทางด้านโรว์ (หลักต่ำ) เช่นตัวเลข 9 = 39H อักษร A=41H เราสามารถแบ่งรหัส ASCII ออกเป็น 2 ประเภทคือ

1. รหัสของข้อมูลแทนตัวอักษรและอักขระตั้งแต่ A-Z และตัวเลข 0-9 และสัญญาณต่างๆ ซึ่งสามารถพิมพ์ออกที่เครื่องพิมพ์ได้ เช่นหากต้องการให้พิมพ์ตัวอักษร B จะต้องส่งค่า 42 ไปให้เครื่องพิมพ์
2. รหัสควบคุมการทำงานและการควบคุมเครื่องพิมพ์ในลักษณะต่างๆมีดังนี้

CR (Carriage return) รหัส 0DH ใช้สำหรับให้เครื่องพิมพ์พิมพ์ข้อมูลที่อยู่ในบัฟเฟอร์แล้วเลื่อนหัวพิมพ์ไปที่จุดเริ่มต้นทางซ้ายมือสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LF (Line feed) รหัส 0AH สำหรับให้เครื่องพิมพ์ พิมพ์ข้อมูลในบัฟเฟอร์และเลื่อนกระดาษเพิ่มไปอีก 1 บรรทัด

VT (Vertical tab) รหัส 0BH สำหรับการควบคุมการเลื่อนกระดาษไปตามที่กำหนดไว้

FF (Form feed) รหัส 0CH สำหรับควบคุมเครื่องพิมพ์ให้เลื่อนกระดาษไปให้ครบหน้าเพื่อขึ้นหน้าใหม่

CAN (Cancel) รหัส 18H สำหรับยกเลิกข้อมูลที่เข้ามาก่อนหน้านี้

SO (Shift out) รหัส 0EH สำหรับควบคุมให้เครื่องพิมพ์ขยายตัวพิพม์เป็น 2 เท่า

SI (Shift in) รหัส 0FH สำหรับควบคุมให้เครื่องพิมพ์แบบตัวบีบ

DC1 (Device control1) รหัส 11H สำหรับควบคุมให้เครื่องพิมพ์อยู่ในสภาวะ SELECT

DC2 (Device control2) รหัส 12H ยกเลิกการพิมพ์ตัวบีบ

DC3 (Device control3) 13H สำหรับควบคุมให้เครื่องพิมพ์อยู่ในสภาวะ DESELECT

DC4 (Device control4) 14H ยกเลิกการพิมพ์ตัวขยาย

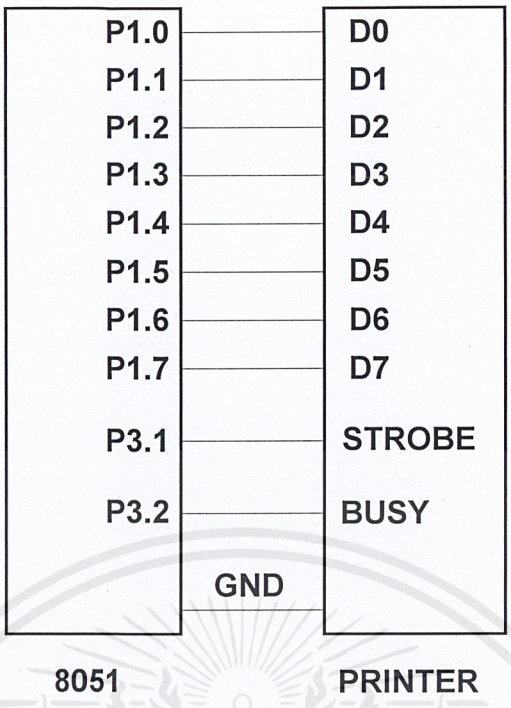
BS (Back space) 08H ควบคุมให้เครื่องพิมพ์ พิมพ์ข้อมูลในบัฟเฟอร์ แล้วเลื่อนหัวพิมพ์ไปทางซ้ายมือหนึ่งตำแหน่ง หากมีข้อมูลเข้ามาก็จะพิมพ์ทับของเดิม

HT (Horizontal tab) 09H เคลื่อนหัวพิมพ์ไปทางแนวนอนในตำแหน่งที่กำหนดไว้

2.6.3 การเชื่อมต่อไมโครคอนโทรเลอร์ 8051 กับเครื่องพิมพ์

การต่อไมโครคอนโทรเลอร์ 8051 กับเครื่องพิมพ์เราใช้วงจรเชื่อมต่อที่ใช้สำหรับส่งข้อมูลแบบขนานที่มีการตรวจสอบความพร้อม โดยจะใช้สัญญาณตรวจสอบต่างๆดังแสดงในภาพที่ 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

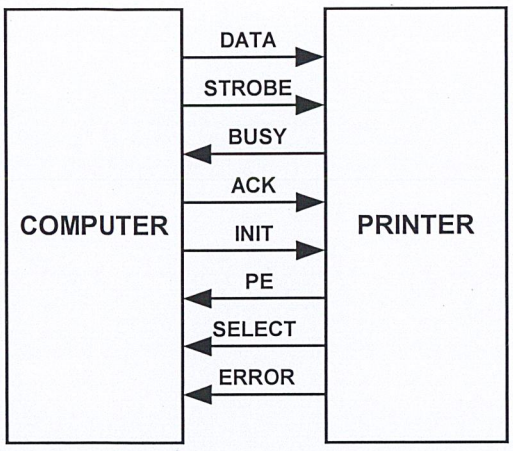


ภาพที่ 2.23 สัญญาณการเชื่อมต่อคอมพิวเตอร์กับเครื่องพิมพ์

วงจรเชื่อมต่อไมโครคอนโทรลเลอร์กับเครื่องพิมพ์ จะใช้สัญญาณข้อมูล และตรวจสอบเท่าที่จำเป็นคือ DATA1-DATA8 , STROBE และ BUSY ซึ่งเพียงพอกับการใช้ส่งข้อมูลแบบมีการตรวจสอบความพร้อมพอร์ตที่ใช้จะมีพอร์ตเอาต์พุตที่ส่งข้อมูล 8 บิต พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุม (STROBE) 1บิต และมีพอร์ตอินพุตสำหรับตรวจสอบสภาวะของเครื่องพิมพ์ (BUSY) อีก 1 บิต วงจรที่เราใช้เชื่อมต่ออาจใช้ 8255 ตัวเดียวก็ได้ หรือใช้อุปกรณ์ทีทีแอลที่เป็นเกตและบัฟเฟอร์ต่างๆมาต่อร่วมกันเป็นพอร์ต หรือจะต่อเครื่องพิมพ์กับพอร์ตของ 8051 โดยตรงก็ได้ในที่นี้จะขอยกตัวอย่างวงจรที่เชื่อมต่อกับพอร์ตของ 8051 และการเชื่อมต่อผ่าน 8255

การเชื่อมต่อเครื่องพิมพ์เข้ากับพอร์ตของ 8051 โดยตรงสามารถทำได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.24 วงจรเชื่อมต่อ 8051 กับเครื่องพิมพ์

จากภาพ 2.24 เราใช้พอร์ต P1 สำหรับการส่งข้อมูล DATA1-DATA8 และสัญญาณ BUSY ของเครื่องพิมพ์ต่อกับ P3.2 สัญญาณ STROBE ต่อกับ P3.1 การทำงานของพอร์ตต่างๆของ 8051 จะเป็นดังนี้

P1.0-1.7 เป็นเอาท์พุต

P3.1 เป็นเอาท์พุต

P3.2 เป็นอินพุต

2.7 การขับโมดูลแสดงผลแบบผลึกเหลว (LCD Module)

รายละเอียดเกี่ยวกับโมดูล LCD

ในโมดูล LCD จะมีส่วนประกอบหลัก 3 ส่วน ดังนี้

ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้น จึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

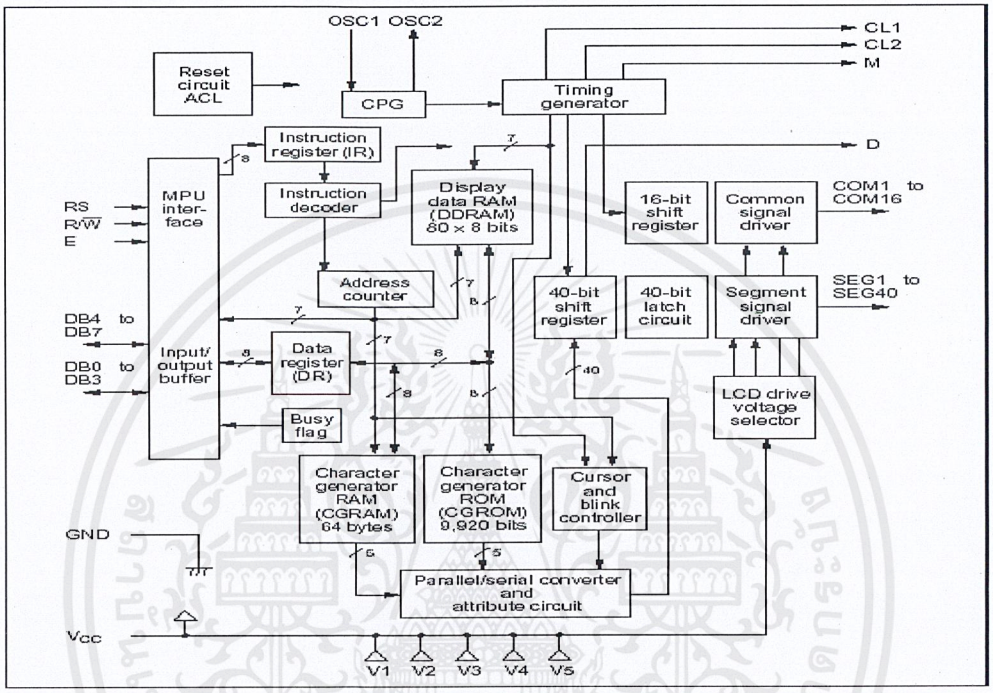
ตัวควบคุม(Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะชิปที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักขระ ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟิก

ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ ได้แก่ เบอร์ HD44100H เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อนในหนังสือนี้ขอยกตัวอย่างโมดูล LCD แบบอักขระ เพราะสามารถเข้าใจได้ง่าย ให้ภาพที่ 2.25 เป็นบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780 ซึ่งใช้ในโมดูล LCD แบบอักขระ ประกอบด้วย



ภาพที่ 2.25 ไดอะแกรมการทำงานของโมดูล LCD MODULE แบบอักขระ

บัฟเฟอร์อินพุตเอาต์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอกเพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register: IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล

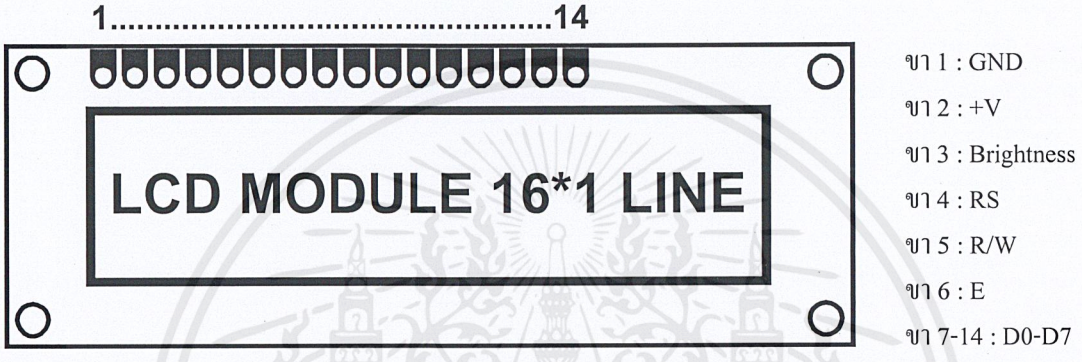
รีจิสเตอร์ข้อมูล (Data Register: DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data RAM: DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up lable) ของตัวอักษรที่เก็บในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอมเก็บตัวอักษร (Character Generator ROM: CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษร หรือสัญลักษณ์ที่สามารถอ่านออก ไปแสดงที่ตัวแสดงผลได้ ขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generator RAM: CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกัน CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง



ภาพที่ 2.26 รูปร่างและการจัดขาโมดูล LCD แบบอักขระ

แฟลก Busy เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุม ให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้น ก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก Busy นี้เสียก่อน

2.7.2 โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16x1)

สำหรับโมดูล LCD ที่ยกมาใช้ในการเรียนรู้ในการทดลอง เป็นขนาด 16 ตัวอักษร 1 บรรทัด เนื่องจากราคาถูก หาง่าย และเป็นโมดูล LCD ที่มีโครงสร้างเป็นมาตรฐาน มีผู้ผลิตหลายราย และมีการระบุเบอร์แตกต่างกันออกไปตามผู้ผลิต อาทิ LM020L ของฮิตาชิ, DMC16117A ของคอปเทร็กซ์ (Optrex) เป็นต้น แต่อย่างไรก็ตามคอนโทรลเลอร์ที่ใช้คือเบอร์เดียวกันนั้นคือเบอร์ HD44750 ของฮิตาชิ

โมดูล LCD ขนาด 16x1 มีขาต่อใช้งานทั้งสิ้น 14 ขา มีการจัดขาดังในภาพที่ 2.26 สำหรับรายละเอียดการทำงานของแต่ละขามีดังนี้

V_{ss} (ขา 1): ต่อกราวด์

V_{DD} (ขา 2): ต่อไฟเลี้ยง + 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V_0 (ขา 3): เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4): เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/$\overline{W}$ (ขา 5): เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6): เป็นขาอีนาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7-14): เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

อนึ่ง ขา RS, R/W และ E จะใช้งานร่วมกัน โดยมีความสัมพันธ์กัน

2.7.3 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของซิปควบคุม HD44780 ที่สำคัญมี 10 คำสั่ง ดังนี้

1.คำสั่งเคลียร์ตัวแสดงผล (Clear Display)

มีข้อมูลคำสั่ง เป็น 10H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ Space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ๊กซิคิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวถึงภายหลัง) ให้เป็น “1”

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

2.คำสั่ง Return home

ต้องกำหนดให้บิด 1 ของข้อมูล เป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลงนั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

3.คำสั่งเลือกโหมดการป้อนข้อมูล (Entry Mode Set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น “1” เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น “0” เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น “0” แอดเดรสจะลดลง

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ ได้แก่ 04H-07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4.คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น “1” จะเป็นการเปิดจอแสดงผล ถ้าเป็น “0” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้เป็น “0” จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกะพริบ

ดังนั้น จะมีข้อมูลคำสั่งได้ตั้งแต่ 08H-0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์และสั่งให้เคอร์เซอร์กะพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ตามตารางที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.11 แสดงการควบคุมการเลื่อนเคอร์เซอร์

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1C-1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น “0” จะแสดงผล 1 บรรทัด ถ้าเป็น “1” จะแสดง 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น “1”

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น “0” จะเป็นการแสดงผลแบบ 5x7 จุด และถ้าเป็น “1” จะแสดงผลเป็นแบบ 5x10 จุด

ข้อมูลคำสั่งที่ใช้บ่อย คือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต เป็นการแสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5x7 จุด

จุดที่น่าสังเกตคือ โมดูล LCD แบบ 16 ตัวอักษร 1 บรรทัด แม้จะมีบรรทัดการแสดงผลเพียง 1 บรรทัด แต่จะต้องกำหนด N ให้เป็น “1” เนื่องจากแอดเดรสของ DDRAM แบ่งเป็น 2 ช่อง คือ 00H และ 40H

7. คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิต ที่เหลือแทนด้วยค่าแอดเดรสของ CGRAM ต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A	A	A	A	A	A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น “1” และข้อมูลอีก 7 บิต ที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น “0” แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น “1” แอดเดรสของ DDRAM จะมี 2 ช่วง คือ 8CH-87H และ 0C0H-0C7H

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A	A	A	A	A	A	A

9. คำสั่งอ่านแฟลก Busy และแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านแฟลก Busy (BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลหรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่า ขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/$\overline{W}$ เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่ เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0-บิต 6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.7.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ จะต้องส่งคำสั่ง (Instruction) แล้วกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อน จากนั้นจึงค่อยส่งข้อมูล (Data) ที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูล จึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก “0” หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่ง ในทางตรงข้าม หากขา RS ได้รับลอจิก “1” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น “1” เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/$\overline{W}$ เป็น “1” ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก “1” ให้ขา RS แล้วต้องกำหนดให้ขา R/W เป็น “0” เป็นข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.7.5 จังหวะการทำงานของ LCD โมดูล

ในการติดต่อกับโมดูล LCD จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูลเนื่องจากต้องรอให้คอนโทรลเลอร์ภายใน LCD โมดูล แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

ดังนั้น ในการใช้งานโมดูล LCD ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้โมดูล LCD พร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่โมดูล LCD ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้โมดูล LCD ทำการเตรียมความพร้อมหรืออินิเชียล (Initial) หลังจากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของโมดูล LCD แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาที เพื่อให้คอนโทรลเลอร์ใน LCD โมดูลแปลความหมายของลอจิกที่ขา RS ว่า ข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมด 8 บิต) ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิลโมดูล LCD ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยป้อนเข้าที่ขา E ของโมดูล LCD ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

2.8 การใช้ DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)

ผู้ผลิตคือ ดัลลัสเซมิคอนดักเตอร์ (Dallas Semiconductor) มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์ โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที, นาที, ชั่วโมง, วันที่ (Date), วันในสัปดาห์ (Day), เดือน และ ปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีอธิกสุรทินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

- เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทินด้วย สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

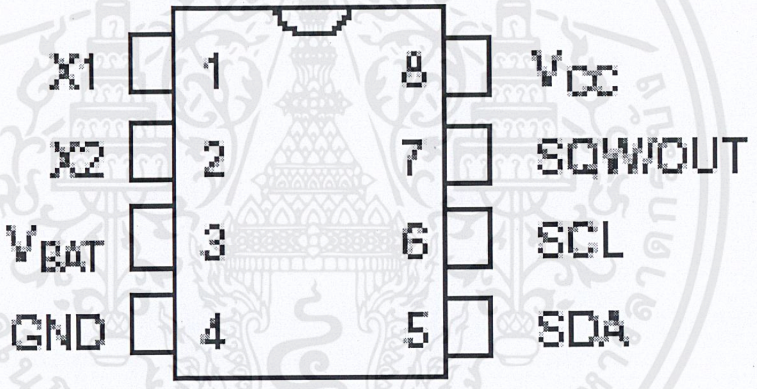
- มีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์ อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส I^2C
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

2.8.1 รายละเอียดขาต่อใช้งานของ DS1307

ในภาพที่ 2.27 แสดงการจัดขาของ DS1307 แต่ละขามีหน้าที่และการใช้งานดังนี้

V_{CC}, GND (ขา 8, 4) ต่อกับไฟเลี้ยง+5V

VB_{AT} (ขา 3) ใช้ต่อกับแบตเตอรี่ 3V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS 1307 ให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสม คือ แบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40mAhr หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปี ที่อุณหภูมิ 25 องศาเซลเซียส



ภาพที่ 2.27 การจัดขาของไอซี DS1307

SDA, SCL (ขา 5 และ 6) เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัส I^2C

SQW/OUT (ขา 7) ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1Hz, 4.096kHz, 8.192kHz และ 32kHz ในการใช้งานต้องต่อตัวต้านทาน 1k พูลอัปที่ขานี้ด้วย

X1, X2 (ขา 1 และ 2) ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริงในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่ำๆ ประมาณ 15pF คร่อมกับขากราวด์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

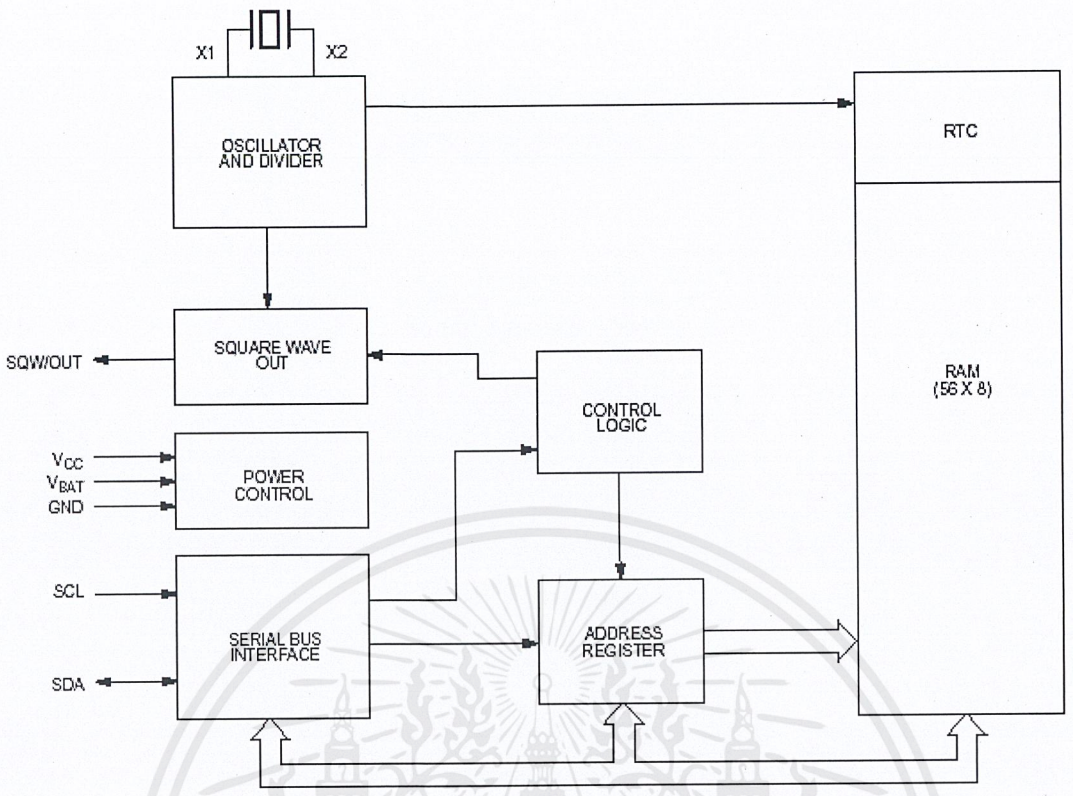
2.8.2 การทำงานของ DS1307

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I^2C ในภาพที่ 2.28 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT มีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการอีนาเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่า คือ 1Hz, 4.096kHz, 8.192kHz และ 32kHz พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่วยความจำแรมซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

สัญญาณ GATE กำหนดจากการกำหนดค่าให้กับรีจิสเ ตอร์ TMOD ค่าของการนับจะเริ่มต้นจากค่าที่อยู่ในรีจิสเตอร์ TH1 และ TL1 ซึ่งเรากำหนดจากคำสั่งทางซอฟต์แวร์ ค่าในรีจิสเตอร์ TL1 ใช้เพียง 5 บิตต่ำเท่านั้น อีก 3 บิตที่เหลือไม่ใช้ การนับจะนับขึ้นทีละ 1 เมื่อค่าในรีจิสเตอร์ TH1 และ TL1 เป็น 1 ทุกบิตแล้วจะวนกลับไปเป็น 0 ทุกบิตเมื่อมีสัญญาณนับเข้ามาอีก 1 ลูก ซึ่งช่วงเวลาของการเปลี่ยนจาก 1 ทุบบิตไปเป็น 0 ทุกบิตจะทำให้ค่าของแฟลก TF1 (Timer Overflow) ถูกเซตให้เป็น 1 แฟลก TF1 นี้จะเป็นสัญญาณร้องขออินเคอร์รัพต์ของ Timer1 ต่อไมโครคอนโทรลเลอร์ และแฟลก TF1 จะถูกเคลียเมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการร้องขออินเตอร์รัพต์ของ Timer1 ซึ่งอยู่ในตำแหน่งเริ่มต้นที่ 001BH

การตรวจสอบสภาวะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า $1.25xV_{BAT}$ ก็จะควบคุมให้ DS1307 หยุดการทำงานรีเซตค่าตัวนับแอดเดรสภายในทำให้ ไม่สามารถติดต่อกับ DS1307 ดังนั้นในการใช้งาน DS1307 ควรระมัดระวังอย่าให้ไฟเลี้ยงต่ำกว่า $1.25xV_{BAT}$ หรือประมาณ 3.75V ในกรณีที่ใช้ V_{BAT} เท่ากับ 3V หากไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมี ไฟเลี้ยง ปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.28 โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส I^2C

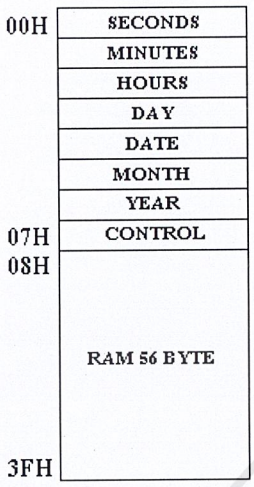
2.8.3 การจัดสรรหน่วยความจำใน DS1307

ในภาพที่ 2.29 (ก) แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์ แรกตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจิสเตอร์ค่าเวลาใช้ในเก็บข้อมูลเกี่ยวกับเวลาไบต์ต่อมาที่แอดเดรส 07H เป็นพื้นที่ของรีจิสเตอร์ควบคุมการทำงานของ DS1307 ในภาพที่ 2.29 (ข) แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

ด้วยการจัดสรรพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามที่ต้องการ โดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐานสิบ สำหรับการแสดงเวลาในรูปของชั่วโมง สามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24 ชั่วโมง โดยกำหนดที่บิต 6 ของแอดเดรส 02H และเมื่อเลือกแบบ 12 ชั่วโมง ที่บิต 5 ในแอดเดรสเดียวกันจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแสดงค่า AM/PM โดยถ้าบิตนี้เป็น “1” หมายถึง ค่าชั่วโมงในขณะนี้เป็นช่วงเวลาหลังเที่ยงวัน ในกรณีที่เป็นแบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง



(ก)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
CH	10 SECONDS				SECONDS				00-59
X	10 MINUTES				MINUTES				00-59
X	12/24	10 HR A/P		10 HR	HOURS				01-12 / 00-23
X	X	X	X		X	DAY			1-7
X	X	10 DATE			DATE				01-28/29 01-30 01-31
X	X	10 MOUTH			MONTH				01-12
10 YEAR					YEAR				00-99
OUT	X	X	SQWE		X	X	RS1	RS0	

(ข)

ภาพที่ 2.29 (ก) การจัดสรรหน่วยความจำแรมภายใน DS1307
(ข) รายละเอียดของรีจิสเตอร์เก็บค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

2.8.4 รีจิสเตอร์ควบคุม

มีแอดเดรสอยู่ที่ 07H มีรายละเอียดของแต่ละบิตดังนี้

OUT (Output control) : ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่ดิสเอเบิลการกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น “1” ที่ขา SQW/OUT ก็จะเป็น “1” ถ้าบิตนี้เป็น “0” ที่ขา SQW/OUT ก็จะเป็น “0”

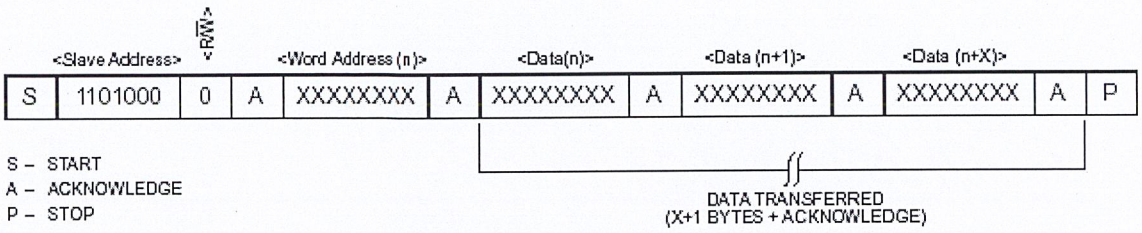
SQWE (Square Wave Enable): ใช้ในการอีนาเบิลวงจรกำเนิดสัญญาณสี่เหลี่ยมที่ขา SQW/OUT ถ้าต้องการให้มีสัญญาณสี่เหลี่ยมออกให้กำหนดบิตนี้เป็น “1”

RS1, RS0 (Rate Select): ใช้ในการเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SQW/OUT

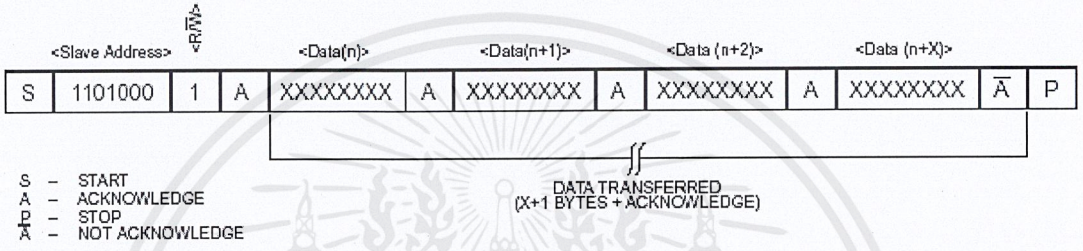
ตารางที่ 2.12 การเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SQW/OUT

RS1	RS0	ค่าความถี่ของสัญญาณสี่เหลี่ยม
0	0	1Hz
0	1	4,096kHz
1	0	8,192kHz
1	1	32.,768 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.30 รูปแบบของข้อมูลสำหรับติดต่อกับ DS 1307 ในโหมดการเขียนข้อมูล



ภาพที่ 2.31 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

2.8.5 โหมดการทำงานของ DS1307

มีด้วยกัน 2 โหมดคือ โหมดเขียนข้อมูลและโหมดอ่านข้อมูลในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งานโหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโปมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

2.8.6 โหมดการเขียนข้อมูล

มีรูปแบบดังในภาพที่ 2.30 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ ทำการกำหนดสภาวะเริ่มต้น (Start: S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดรอการตอบรับจาก DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกครั้ง จึงจะสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสภาวะหยุด (Stop: P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล

2.8.7 โหมดการอ่านข้อมูล

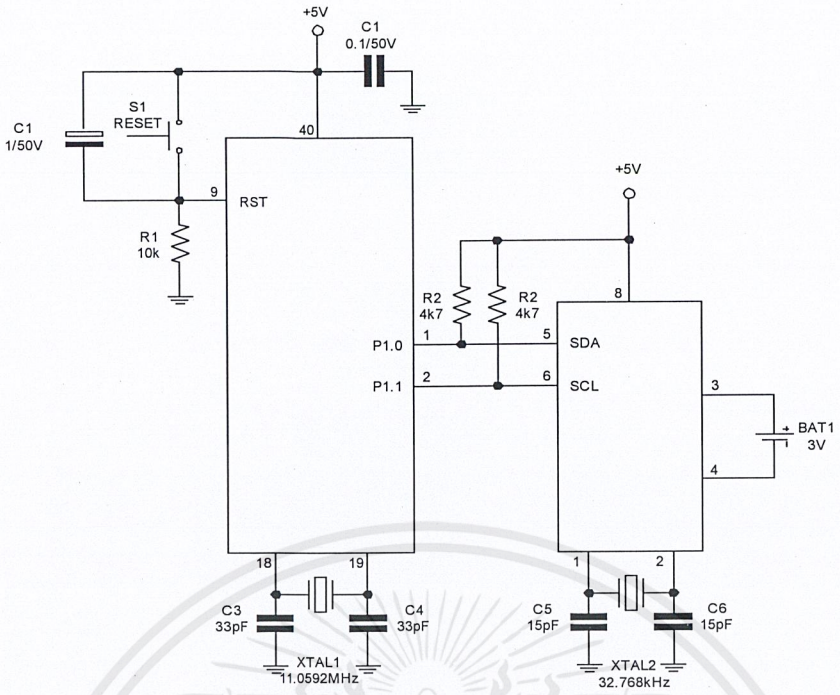
มีรูปแบบแสดงในภาพที่ 2.31 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูล คือไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้น แล้วส่งข้อมูลกำหนดแอเดรสตามด้วยข้อมูลเลือกการอ่านซึ่งเท่ากับ 1 จากนั้น รอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อย DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรส หรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆ คือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูล ให้ทำการสร้างสภาวะเริ่มต้น และส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้

2.8.8 การเชื่อมต่อ DS1307 กับไมโครคอนโทรลเลอร์ MCS-51

มีตัวอย่างวงจรแสดงในภาพที่ 2.32 จะเห็นได้ว่ามีลักษณะการต่อเหมือนกับอุปกรณ์บนระบบ บัส I^2C ตัวอื่นๆ ทุกประการ และสามารถที่จะต่อไอซีทั้งหมดร่วมกันบนสาย SDA และ SCL ได้ เป็นการย้ำให้เห็นถึงความสามารถพิเศษของระบบบัส I^2C ที่ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ที่มีความต่างกันในหน้าที่การทำงานบนสายสัญญาณเดียวกันได้ ถึงการทดลองนี้ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ระบบบัส I^2C ได้ถึง 3 ตัว 3 ลักษณะการทำงานโดยใช้สายสัญญาณเพียง 2 เส้น

จากวงจรในภาพที่ 2.32 ไอซี DS1307 จำเป็นจะต้องต่อแบตเตอรี่ไว้ตลอดเวลาไม่ว่าจะใช้งานหรือไม่ ทั้งนี้เพื่อรักษาการทำงานของวงจรภายใน DS1307 ให้ยังคงทำงานต่อเนื่องไป เมื่อใดที่ไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูล ก็จะได้ข้อมูลเวลาที่เป็นจริงตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.32 แสดงการเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับไอซีรีลไทม์คล็อก DS1307

2.9 ความรู้เบื้องต้นของพอร์ตอนุกรม

ในขณะที่การรับส่งข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลได้คราวละหลายๆ บิตได้ หากแต่จะต้องมีการตกลงกันระหว่างตัวส่งและตัวรับว่าจะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลอนุกรมอาจมีความเร็วต่ำกว่าแบบขนาน ในด้านจำนวนสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายที่น้อยกว่ามาก อย่างน้อยที่สุดใช้เพียง 2-3 เส้นเท่านั้น แต่อัตราเร็วในการรับส่งข้อมูลอาจต่ำกว่าแบบขนาน อย่างไรก็ตามการรับส่งข้อมูลแบบอนุกรมสามารถใช้สายสัญญาณที่มีความยาวมากกว่าแบบขนาน ทำให้ระยะทางในการสื่อสารข้อมูลแบบอนุกรมสามารถทำได้มากกว่า

2.9.1 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่าสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association: EIA) ได้กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ –3 ถึง –12V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12V แสดงว่าเป็นช่องว่าง (Space)

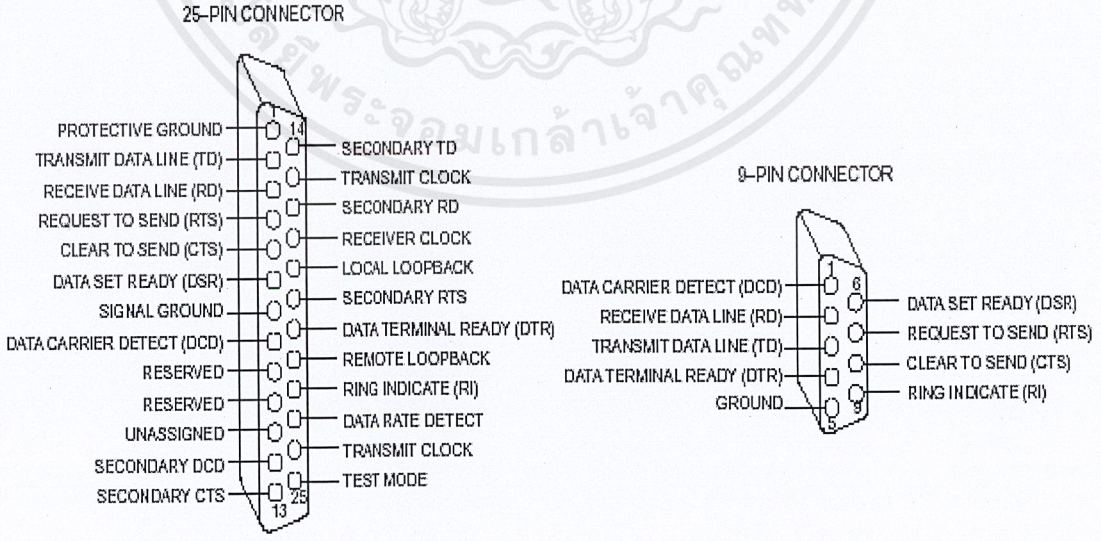
มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE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

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

2.9.2 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

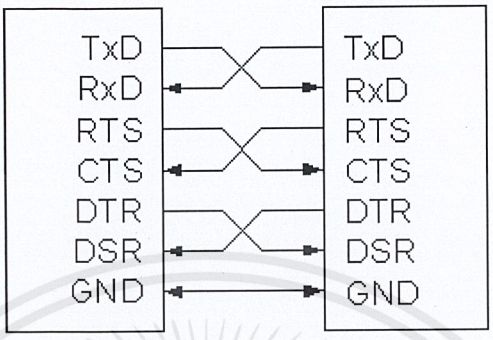
มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีตปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป



ภาพที่ 2.33 การจัดขาของพอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบDB-9 และ DB-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังในภาพที่ 2.33 ลูกศรในรูปแสดงถึงทิศทางของข้อมูลในภาพที่ 2.34 เป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้



ภาพที่ 2.34 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

- Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect: CD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ถูกใช้งานมากนัก

- Receive Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

- Transmitted Data: TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

- Data Terminal Ready: DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วย ในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

- Signal Ground: GND ขากราวด์ของระบบ

- Data Set Ready: DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Request to Send: RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

- Clear to Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้น ขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมรับข้อมูลหรือไม่

- Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.9.3 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้น เมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์ และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ

2.9.4 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่ง ดังนี้คือ

COM1: 3F8H

COM2: 2F8H

COM3: 3E8H

COM4: 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำ ขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1 จะเก็บไว้ที่แอดเดรส 0000:0400H และ 0000:0401H ส่วนตำแหน่งอื่นๆ มีรายละเอียดดังนี้

COM2 = 0000:0402H - 0000:0403H

COM3 = 0000:0404H - 0000:0405H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM4 = 0000:0406H - 0000:0407H

นอกจากนี้ ที่หน่วยความจำแอดเดรส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วย ตามที่แสดงรายละเอียดดังกล่าวมาแล้ว

ตารางที่ 2.13 แสดงข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

บิต 3	บิต 2	บิต 1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

2.9.5 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม

การรับและส่งข้อมูลแบบอนุกรม มีหลากหลายวิธีในการรับและส่งข้อมูลแบบอนุกรมผ่านพอร์ตอนุกรม RS-232 เช่น ใช้คำสั่งพิมพ์ออกทางเครื่องพิมพ์ เรียกอินเตอร์รัปต์ของไบออสหรือของดอสการเขียนหรืออ่านไปยังแอดเดรสของพอร์ตโดยตรงวิธีสุดท้ายเป็นวิธีที่มีความยืดหยุ่นในการใช้งานที่สุด ยกตัวอย่าง ถ้าต้องการส่งข้อมูลไปยังพอร์ตอนุกรม COM1 สามารถเขียนข้อมูลโดยตรงไปที่รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (แอดเดรส 3F8H) โดยใช้คำสั่งภาษา QBASIC ง่ายๆ ดังนี้

OUT &H3F8, X

ค่า X ในที่นี้หมายถึงข้อมูลที่ต้องการส่ง มีขนาด 8 บิต

สำหรับการอ่านข้อมูลจากพอร์ตอนุกรม จะเป็นการอ่านข้อมูลมาจากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (แอดเดรส 3F8H เช่นเดียวกัน) ซึ่งสามารถเขียนโปรแกรมง่ายๆ ได้ดังนี้

Y = INP (&H3F8)

ค่า Y ในที่นี้คือค่าที่อ่านได้จากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล โดยมีขนาด 8 บิต

สำหรับการเขียนโปรแกรมด้วย TURBO PASCAL ก็สามารถใช้คำสั่ง

PORT (S3F8) = X

สำหรับการอ่านข้อมูลจากพอร์ตอนุกรม

แต่เมื่อใช้คำสั่งนี้ในขณะที่โปรแกรมทำงานผ่านระบบปฏิบัติการวินโดวส์ จะไม่สามารถใช้งานได้เนื่องจากระบบปฏิบัติการวินโดว์ได้เข้าฝังตัวพอร์ตอนุกรมเข้าเป็นส่วนหนึ่งของระบบปฏิบัติการแล้ว ดังนั้น การเรียกใช้งานจึงจำเป็นต้องเรียกผ่านเครื่องมือที่ติดต่อผ่านระบบปฏิบัติการ เช่นการใช้คอนโทรล MSCOMM32.OCX ของโปรแกรม Visual BASIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.6 คอนโทรล MSComm

สำหรับการใช้งาน Visual BASIC ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual BASIC จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยใน Visual BASIC เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MSCOMM32.OCX สำหรับการทำงานกับระบบปฏิบัติการ 32 บิต สำหรับใน Visual BASIC เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้น เพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทาง เพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (Event-driven Communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request to Send (RTS) เหตุการณ์ ONComm ของ MSComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEvent ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ต จะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัว เพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสของพอร์ตอนุกรมและแอดเดรสของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติ (Property) มากมาย แต่สามารถทำความเข้าใจได้ไม่ยากดังนี้

CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่ออยู่ (COM1, COM2, COM3 COM4)

รูปแบบการใช้งาน

Object. CommPort [= value]

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้ว ทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Error68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

Setting

ใช้ในการกำหนดและอ่านค่าอัตรา บอดเรต, พาริตี้, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย

รูปแบบการใช้งาน

Object. Settings [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB, P.D.S” BBBB เป็นค่าอัตราบอด, P เป็นค่าพาริตี้, D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600, N, 8, 1”

ค่าบอดเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

110 บิต ต่อวินาที

300 บิต ต่อวินาที

600 บิต ต่อวินาที

1,200 บิต ต่อวินาที

2,400 บิต ต่อวินาที

9,600 บิต ต่อวินาที (ค่าปกติ)

14,400 บิต ต่อวินาที

19,200 บิต ต่อวินาที

28,800 บิต ต่อวินาที

38,400 บิต ต่อวินาที (สงวน)

56,000 บิต ต่อวินาที (สงวน)

128,000 บิต ต่อวินาที (สงวน)

256,000 บิต ต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตี้มีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตี้คู่ (Even)
M	ลอจิก “1” (Mark)
N	ไม่ใช้ (ค่าปกติ)
O	พาริตี้คี่ (Odd)
S	ลอจิก “0” (Space)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่า คือ 4,5,6,7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่า คือ 1 (เป็นค่าปกติ), 1.5 และ 2

ตัวอย่างการใช้งานคำสั่ง Setting โดยเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี้ จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

MSComm1.Settings = “9600, N, 8, 1”

หมายเหตุ สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด “” เนื่องจากค่าที่กำหนดนี้อยู่ในรูปตัวแปร String

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

Object. PortOpen [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึง การเปิดพอร์ตอนุกรม และ False หมายถึง การปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดบอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรมก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

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

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9,600 บิต ต่อวินาที ไม่มีพาริตี้ จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

MSComm1.Settings = “960, N, 8, 1”

MSComm1.ComPort = 1

MSComm1.PortOpen = True

Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

Object. Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมุลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น CominputMode Text คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น ComInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปของไบนารี่และชนิดข้อมูลเป็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีในการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Commandl _ Click ( )
Dim InString as String
   MsComml.InputLen = 0                ‘ Retrieve all available data.
   If MSComml. InBufferCount Then      ‘ Check for data.
   End If                              ‘ Read data.
End Sub
```

InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object. InBufferCount [= value]

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอก และยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

หมายเหตุ อย่าสับสนระหว่างคำสั่ง InBufferSize และ InBufferCount คำสั่ง InBufferSize นั้นใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ

InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

Object. InBufferSize [= value]

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆ จะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โพลว์หรือข้อมูลล้นบัฟเฟอร์เว้นแต่จะมีการใช้แฮนด์เช็กดังนั้นค่าปานกลางที่เหมาะสมก็คือ ค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลว์แล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีค่ามากขึ้น

InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object. InputLen [= value]

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

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน InputLen คำสั่ง Input จะส่งค่าว่าง (“”) กลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ InBufferCount โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

ตัวอย่างโปรแกรมการอ่านค่าตัวอักษรออกมา 10 ตัวอักษร

```
Private Commandl_Click ( )
Dim CommData as String
MSComml. InputLen = 10          ‘ Specify a 10 character block of data.
CommData = MSComm1.Input        ‘ Read data.
End Sub
```

InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับโดยคำสั่ง Input

รูปแบบการใช้งานคำสั่ง

Object. InputMode [= value]

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภท คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ComInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น “0” และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

ComInputModeBinary สำหรับข้อมูลอื่นๆจะเก็บในรูปไบนารี่รวมกันอยู่เป็นไบต์ข้อมูล

ตัวอย่างการใช้งาน InputMode ต่อไปนี้จะทำการอ่านค่าข้อมูล 10 ไบต์ จากพอร์ตอนุกรม และเก็บข้อมูลไว้ในตัวแปรแบบอาเรย์ ชนิดข้อมูลเป็นแบบไบต์

```
Private Sub Commandl_Click ()
Dim Buffer as Variant
Dim Arr () as Byte
MSComm1. CommPort = 1 ‘ Set and open port.
MSComm1. PortOpen = True
MSComm1. InputMode = ComInputModeBinary ‘ Set InputMode to read binary data.
Do Until MSComm1. InBufferCount < 10 ‘ Wait until 10 bytes are in the input buffer
   DoEvents
Loop
Buffer  =  MSComm1. Input          ‘ Store binary data in buffer
Arr = Buffer                       ‘ Assign to byte array for processing.
End Sub
```

Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

Object. Output [= value]

ค่า Value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารี่ก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารี่จะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ Byte

ตัวอย่างโปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ดไปยังพอร์ตอนุกรม โดยใช้คุณสมบัติ Output

```
Private Sub Form_KeyPress (KeyAscii As Integer)
Dim Buffer as Variant
MSComm1. CommPort = 1        ‘ Use COM1
MSComm1. CommPort = True     ‘ Open port
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Buffer = ChrS (KeyAscii)
MSComm1. Output = Buffer      'Send DATA
End Sub
```

OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถใช้คำสั่งนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

Object, OutBufferCount [= value)

ผู้ใช้งานสามารถเคลียร์บัฟเฟอร์ภาคส่งได้โดยการกำหนดค่า OutBufferCount เท่ากับ "0"

หมายเหตุ ระวังการสับสนระหว่างคุณสมบัติ OutBufferCount กับ OutBufferSize ซึ่ง OutBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาคส่ง

OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

Object. OutBufferSize [= object]

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไปจะทำให้ มีหน่วยความจำเหลือให้ใช้งานน้อย แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้เกิดข้อมูลล้นบัฟเฟอร์ขึ้นได้ ยกเว้นจะมีการใช้แฮนด์เช็ค วิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 512 ไบต์ดูก่อน ถ้าโปรแกรมทำงานแล้วเกิดการล้นของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น

ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี้

รูปแบบการใช้งานคำสั่ง

Object. ParityReplace [= value]

บิตพาริตี้ เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี้ คอนโทรล MSCOM จะทำการบวกบิตทุกบิตที่มีค่าลอจิก "1" ในแต่ละไบต์ และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเลขคู่หรือคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตี้ไม่ตรง แสดงว่าการรับส่งข้อมูลผิดพลาด

การกำหนดค่าเริ่มต้นให้กับ ParityReplace นั้นกำหนดให้ใช้เครื่องหมาย (?) ไปวางไว้ที่ตำแหน่งที่เกิดพาริตี้ผิดพลาด ถ้ากำหนดค่า ParityReplace ให้เป็นค่าว่าง ("") จะเป็นการยกเลิกการใช้งาน ParityReplace และไม่มีการป้อนข้อมูลแทนเมื่อตรวจพบข้อผิดพลาด

ParityReplace ใช้ชนิดข้อมูลเป็นแบบสตริง แต่จะการกำหนด จะกำหนดได้เพียงไบต์เดียวเท่านั้น ซึ่งจะสามารถใช้ค่าใดๆ ก็ได้ที่เป็นโค้ด ANSI มีค่าอยู่ระหว่าง 0-255

DTREnable

ใช้ในการกำหนดสถานะลอจิกของขา Data Terminal Ready (DTR) โดยสัญญาณของขา DTR จะส่งจากคอมพิวเตอร์ไปยังโมเด็มเพื่อแสดงว่าคอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งาน

Object. DTREnable [= value]

ค่า Value เป็นค่าสถานะ 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" เมื่อมีการปิดพอร์ต เมื่อขา DTR ถูกกำหนดสถานะเป็น False ท่ขา DTR จะมีสถานะลอจิก "0" ตลอดเวลาไม่ว่าจะใช้คำสั่งเปิดพอร์ตหรือปิดพอร์ต

สำหรับการใช้งานกับโมเด็ม การทำให้ขา DTR เป็นลอจิก "0" จะเป็นการวางหูโทรศัพท์หรือยกเลิกการติดต่อ

RTSEnable

ใช้เพื่อกำหนดสถานะลอจิกให้ขา Request 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"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

False หมายถึง ให้ขา RTS มีลอจิก “0” (เป็นค่าปกติ)

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

EOFEnable

เป็นการกำหนดให้ MSComm รอสัญลักษณ์แสดงส่วนท้ายสุดของไฟล์ (End of file: EOF) ระหว่างการรับอินพุตเข้ามา ถ้าพบสัญลักษณ์ EOF ภาคอินพุตจะหยุดรับข้อมูล และเหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน คุณสมบัติ CommEvent จะมีค่าเท่ากับ 7 หรือ ComEvEOF

รูปแบบการใช้งาน

Object. EOFEnable [= value]

โดย Value เป็นค่าสถานะ True หรือ False เพื่ออีนาเบิลหรือดิสเอเบิลการทำงานของเหตุการณ์ OnComm เมื่อตรวจพบสัญลักษณ์ EOF โดย

True หมายถึง เหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงานด้วย EOF

False หมายถึง เหตุการณ์ OnComm จะไม่ถูกกระตุ้นให้ทำงานด้วย EOF (เป็นค่าปกติ)

เมื่อ EOFEnable กำหนดให้เป็น False ส่วนควบคุมจะไม่มีการตรวจสอบสัญลักษณ์ EOF

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 comEvenCTSTO (Clear to send Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

CDHolding

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อขา DCD มีลอจิก “1” (CDHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCDTO (Carrier 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 comEvenDSRTO (Data Set Ready Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

Handshaking

กำหนดคุณสมบัติและคืนค่ารูปแบบแฮนด์เช็กทางฮาร์ดแวร์

รูปแบบการใช้งานคำสั่ง

Object. Handshaking [= value]

ค่าตัวแปร Value ที่ใช้กำหนดค่ากำหนดได้ 4 รูปแบบ ด้วยกันคือ

1. comNone ค่าที่กำหนดคือ 0 เป็นการกำหนดให้ไม่มีการแฮนด์เช็ก (เป็นค่าเริ่มต้น)
2. comXOnXOff ค่าที่กำหนดคือ 1 เป็นการกำหนดให้ใช้แฮนด์เช็กแบบ XON/XOFF
3. comRTS ค่าที่กำหนดคือ 2 เป็นการกำหนดให้ใช้ขา RTS/CTS (Request to Send และ XON/XOFF

คุณสมบัติ Handshaking ใช้เพื่อกำหนดรูปแบบการสื่อสารภายในระหว่างที่ข้อมูลถูกส่งไปยังบัฟเฟอร์ภาครับ เมื่อข้อมูลตัวอักษรถูกส่งมาถึงพอร์ตอนุกรม อุปกรณ์สื่อสารข้อมูลจะทำการย้ายข้อมูลไปยังบัฟเฟอร์ภาครับ เพื่อที่จะให้โปรแกรมสามารถอ่านค่าไปใช้งานได้ ถ้าไม่มีบัฟเฟอร์ภาครับโปรแกรมที่ใช้งานจะต้องทำการอ่านค่าข้อมูลโดยตรงจากฮาร์ดแวร์ของพอร์ตอนุกรม ซึ่งผู้ใช้งานจะเกิดปัญหาข้อมูลสูญหายได้ เนื่องจากว่าการเปลี่ยนแปลงของข้อมูลที่ส่งเข้ามามีการเปลี่ยนแปลงอย่างรวดเร็ว

คุณสมบัติ Handshaking ช่วยให้ผู้ใช้งานแน่ใจได้ว่าข้อมูลที่ได้รับมานั้นไม่มีการสูญหาย เมื่อบัฟเฟอร์ภาครับที่รับข้อมูลนั้นเกิดข้อมูลล้นหรือโอเวอร์โฟลว (Overflow) โดยใช้วิธีการตรวจสอบความพร้อมของบัฟเฟอร์ว่าพร้อมรับข้อมูลหรือไม่ก่อนที่จะส่งข้อมูลมาให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Break

ใช้ในการเซตและเคลียร์ค่าสัญญาณ Break ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

Object. Break [= value]

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

เมื่อกำหนดให้สัญญาณ Break เป็น True จะเป็นการหยุดการส่งข้อมูลชั่วคราวจนกว่าจะมีการสั่งให้สัญญาณ Break เป็น False

ตัวอย่าง เป็นวิธีการส่งสัญญาณ Break ออกไปเป็นช่วงเวลาสั้นๆ ที่ 1/10 ของวินาที

```
MSComm1. Break = True          ' Set the Break condition.
Duration! = Timer + .1         ' Set duration to 1210 second.
Do Until Timer > Duration!     ' Wait for the duration to pass.
Dummy = DoEvents ()
Loop
MSComm1. Break – False         ' Clear the Break condition.
```

เหตุการณ์ OnComm

เหตุการณ์ OnComm จะถูกสร้างขึ้นเมื่อค่าของคุณสมบัติ CommEvent มีการเปลี่ยนแปลง เพื่อแสดงผลการเปลี่ยนแปลงเหล่านั้นแบบทันทีทันใดหรือแสดงข้อผิดพลาดที่เกิดขึ้น ตัวอย่างโปรแกรมย่อย OnComm เพื่อนำเหตุการณ์ commEvent มาแสดง

```
Private Sub MSComm_OnComm ()
    Select Case MSComm1. CommEvent
    ' Handle each event or error by placing
    ' Code below each case statement
    ' Errors
            Case comEventBreak      ' A break was received.
            Case comEventCDTO       ' CD (RLSD) Timeout.
            Case comEventCTSTO      ' CTS Timeout.
            Case comEventDSRTO      ' DSR Timeout.
            Case comEventFrame      ' Framing Error
            Case comEventOverrun    ' Data Lost.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Case comEventRxOver        ' Receive buffer overflow.
        Case comEventRxParity      ' Parity Error.
        Case comEventTxFull        ' Events
        Case comEvCD               ' Change in the CD line.
        Case comEvCTS              ' Change in the CTS line.
        Case comEvDSR              ' Change in the DSR line.
        Case comEvRing             ' Change in the Ring Indicator.
        Case comEvReceive          ' Received Rthreshold #  of chars.
        Case comEvSend             ' Sthreshold number in the 'transmit buffer.
        Case comEvEof
    End Select
End Sub
```

2.9.7 การใช้ MSComm เพื่อการติดต่อฮาร์แวร์

จากรายละเอียดของ MSComm ที่กล่าวไปในตอนต้นนั้น จะเห็นได้ว่าวิธีการที่จะอ่านค่าหรือเขียนค่าไปยังขาสถานะและขาควบคุมของพอร์ตอนุกรมสามารถทำได้ง่ายดายมาก 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” หนทอ “1”

Break สำหรับการสั่งให้ขา Txd มีลอจิก “0” หรือ “1”

```
MSComm1. Break = True          ' Set the Break condition.
Duration! = Timer + .1         ' Set duration to 1210 second.
Do Until Timer > Duration!     ' Wait for the duration to pass.
Dummy = DoEvents ()
Loop
MSComm1. Break – False         ' Clear the Break condition.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

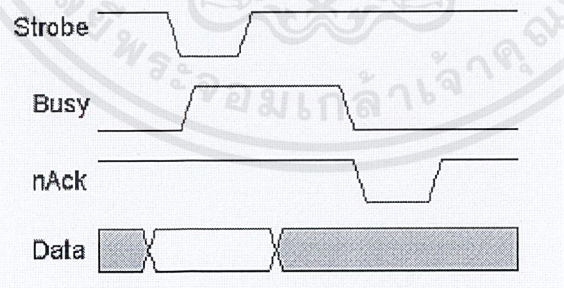
2.10 ความรู้เบื้องต้นของพอร์ตขนาน

พอร์ตขนาน (Parallel Port) สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายทอดข้อมูลของพอร์ตนี้เป็นแบบขนาน สำหรับชื่อเรียกอีกชื่อของพอร์ตขนานคือ พอร์ตเครื่องพิมพ์ (Printer Port) เนื่องจากพอร์ตนี้ใช้สำหรับต่อเครื่องพิมพ์นั่นเอง

ด้วยการถ่ายทอดข้อมูลแบบขนานนี้เองทำให้พอร์ตขนานมีอัตราการถ่ายทอดข้อมูลสูงกว่าการถ่ายทอดข้อมูลแบบอนุกรมประมาณ 8-10เท่า และการประมวลผลข้อมูลส่วนใหญ่จะมีขนาด 8 บิต ดังนั้น พอร์ตขนานจึงสามารถรองรับการถ่ายทอดข้อมูล 8 บิต ได้โดยไม่ต้องต่อส่วนเพิ่มเติมใดๆ

2.10.1 ลักษณะทางกายภาพของพอร์ตขนาน

เพื่อให้เข้าใจถึงการนำเอาพอร์ตขนานไปใช้งาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่า ปกตินั้นการสั่งพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตขนานนั้น มีรูปแบบการทำงานภายในอย่างไร ในภาพที่ 2.35 แสดงไดอะแกรมเวลาของติดต่อระหว่างพอร์ตขนานกับเครื่องพิมพ์ ซึ่งจะเห็นได้ว่ามีสัญญาณที่ใช้งานจริงๆมีไม่มาก เริ่มจากสัญญาณพอร์ต Data ถูกส่งออกไปยังเครื่องพิมพ์พร้อมทั้งส่งสัญญาณ Strobe ออกไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่ามีการส่งข้อมูลใหม่มาที่ขา Data จากนั้นคอมพิวเตอร์จะต้องรอการตอบกลับจากเครื่องพิมพ์ นั่นคือ เครื่องพิมพ์จะสร้างสัญญาณ Busy หรือเพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่องพิมพ์พร้อม เครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว



ภาพที่ 2.35 แสดงไดอะแกรมเวลาของการส่งข้อมูล

สัญญาณข้อมูล ขนาด 8 บิต, สัญญาณ Strobe และสัญญาณ ACK (acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้วส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่นๆร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนฟอนต์ เป็นต้น บางครั้งอาจเกิดเหตุการณ์ที่ไม่ปกติ เช่น บัฟเฟอร์สำหรับข้อมูลเต็ม (เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกลย่อมทำงานได้ช้ากว่าการส่งข้อมูลของคอมพิวเตอร์) เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลมากกว่านี้ได้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์คือ สัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่น กระดาษติด เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่าสัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซตเครื่องพิมพ์ คอมพิวเตอร์จะต้องส่งสัญญาณ Reset ไปยังเครื่องพิมพ์เพื่อรีเซตเครื่องพิมพ์ด้วย สามารถสรุปขาสัญญาณที่จำเป็นสำหรับการติดต่อดังในตาราง 2.13

จากตารางที่ 2.14 จะเห็นได้ว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ต ได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณ Strobe และ Reset พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, Busy และสัญญาณ Error จากเครื่องพิมพ์

ตารางที่ 2.14 สัญญาณสำคัญๆ ของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซตเครื่องพิมพ์	คอมพิวเตอร์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านั้นจะมีรีจิสเตอร์ 3 ตัว ควบคุมการทำงาน ดังนี้

1. พอร์ตเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ Data ควบคุม
2. พอร์ตอินพุตสำหรับการอ่านค่าสถานะต่างๆ จากภายนอก มีอยู่ด้วยกัน 5 เส้น ใช้รีจิสเตอร์ Status ในการควบคุม
3. พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้น ใช้รีจิสเตอร์ Control ในการควบคุม

บล็อกไดอะแกรมในรูปแสดงระบบบัสของคอมพิวเตอร์ สำหรับการติดต่อกับพอร์ตขนาน สัญญาณเอาต์พุตจากพอร์ตขนานจะถูกส่งไปยังคอนเน็กเตอร์แบบ DB-25 สำหรับ

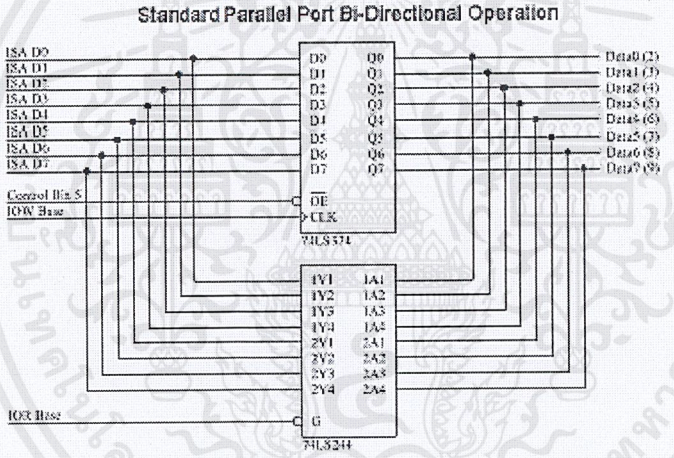
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ส่วนใหญ่ในปัจจุบันพอร์ตขนานจะมีมาพร้อมกับเมนบอร์ด ไม่จำเป็นต้องใช้การ์ดเสียบเพิ่มเติมเหมือนในอดีต พร้อมทั้งมีฟังก์ชันการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ตขนานในรูปแบบมาตรฐาน (SPP) อยู่

เมื่อดูจากภาพที่ 2.35 เทียบการทำงานโดยทั่วไปกับการเชื่อมต่อผ่านการ์ดที่เสียบลงในสล็อตของอ้างแอดเดรส ตำแหน่งแอดเดรสที่ใช้อ้างถึงจะเป็นตำแหน่ง A0-A9 และใช้ขา IOR และ IOW สำหรับเป็นตัวเลือกว่าต้องการอ่านหรือเขียนรีจิสเตอร์ตัวใด จากการดีโค้ดแอดเดรส A0-A9 นี้เองทำให้ได้สัญญาณออกมาเพื่อไปควบคุมหรืออีนาเบิลวงจรบัฟเฟอร์ต่างๆ ดังนี้

Data Write สัญญาณอีนาเบิล สำหรับนำข้อมูลที่อยู่ในบัส Data ไปออกที่ขา Data ของพอร์ตขนาน

Data Read สัญญาณอีนาเบิล สำหรับอ่านข้อมูลจากขา Data ของพอร์ตขนานมาเก็บไว้ในบัส Data



ภาพที่ 2.36 แสดงระบบบัสภายในของพอร์ตขนาน

Control Write สัญญาณอีนาเบิลสำหรับข้อมูลที่อยู่ในบัส Data ไปออกที่ขา Control ของพอร์ตขนาน สำหรับพอร์ตนี้นอกจากจะส่งข้อมูลออกไปยังพอร์ตขนานแล้วยังทำหน้าที่อีนาเบิลการอินเตอร์รัปต์ของการเปลี่ยนแปลงสัญญาณที่พอร์ต Status อีกด้วย

Control Read สัญญาณอีนาเบิลสำหรับอ่านค่าข้อมูลจากขา Control มาเก็บไว้ในบัส Data

Status Readสัญญาณอีนาเบิลสำหรับอ่านค่าข้อมูลจากขาพอร์ต Status มาเก็บไว้ในบัส

ตารางที่ 2.14 แสดงชื่อและหน้าที่การทำงานของตำแหน่งขาต่างๆ บนพอร์ตขนานส่วนในตารางแสดงแอดเดรสของพอร์ตขนาน ซึ่งกำหนดไว้ 3 ตำแหน่ง คือ LPT1, LPT2 และ LPT3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.2 พอร์ตดาต้า (Data Port)

จากภาพที่ 2.36 แสดงให้เห็นว่าพอร์ต Data ประกอบไปด้วยบัฟเฟอร์ 1 ตัว และไอซีแลตช์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตซ์ 1 ทั้ง 8 บิต เอาต์พุตของไอซีแลตช์ 1 คือ DO-D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึงขา 9 และที่ขาเอาต์พุตนี้สัญญาณ Data จะส่งกลับไปเป็นอินพุตของบัฟเฟอร์ 1 ด้วย ทำให้คอมพิวเตอร์สามารถอ่านค่าสถานะปัจจุบันที่เกิดขึ้นกับพอร์ต Data ได้

เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมาจากบัสข้อมูลของคอมพิวเตอร์ผ่านไปให้กับไอซี 74LS374 ซึ่งเป็นไอซีแลตช์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุต คอมพิวเตอร์จะส่งสัญญาณ Data Write ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกรองด้วยวงจร RC ซึ่งประกอบด้วยตัวต้านทานคา 27 □ และตัวเก็บประจุ 0.0022 μ F เพื่อให้ช่วงเวลาที่เปลี่ยนจากลอจิก “0” เป็นลอจิก “1” หรือจากลอจิก ”1” เป็นลอจิก “0” เป็นไปอย่างช้าๆ เนื่องจากการเปลี่ยนแรงดันที่รวดเร็วทำให้เกิดสัญญาณรบกวนเหนี่ยวนำข้ามไปยังข้อมูลบิตอื่นๆ ได้ ทำให้ข้อมูลที่ส่งออกไปมีข้อผิดพลาด จากค่าตัวต้านทานและตัวเก็บประจุในวงจรทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที จากวงจรในภาพที่ 2.37 ทำให้เอาต์พุตของพอร์ต Data มีคุณสมบัติดังนี้

- กระแสซิงก์สูงสุด 24 mA
- กระแสซอร์สสูงสุด 2.6 mA
- ระดับแรงดันของลอจิก “1” ต่ำสุดเท่ากับ 2.4V
- ระดับแรงดันสูงสุดสำหรับลอจิก “0” เท่ากับ 0.5V

สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับ ได้แก่ เบอร์ 74LS244 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ Data Read ออกมาเพื่ออีนาเบิลไอซี 74LS244 สำหรับพอร์ตขนานแบบมาตรฐาน (Standard Parallel Port: SPP) พอร์ต Data จะต้องใช้เพื่อการส่งค่าวอกเอาต์พุตเท่านั้น แต่สำหรับพอร์ตขนานที่มีการสื่อสารสองทิศทาง (Bidirectional Parallel Port) สามารถอ่านค่าจากพอร์ต Data ได้ด้วย แต่ก่อนที่จะอ่านค่าต้องจำไว้เสมอว่าจะต้องป้อนค่าเอาต์พุตให้มีค่าลอจิก “1” ทั้งหมดก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.15 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน

DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
1	Control	Out	$\overline{C0}$	$\overline{STROBE}$	แอกทีฟ “0” ส่งค่าออกไปเพื่อบอกว่าที่ซาดาต้า มีข้อมูลแล้ว
2-9	Data	Out	D1-D8	DATA1-DATAB	สำหรับพอร์ตขนานมาตรฐานเดิมขานี้ทำหน้าที่เป็นขาส่งข้อมูลเอาต์พุตเท่านั้นสำหรับในปัจจุบันขานี้รับข้อมูลอินพุตได้ด้วย
10	Status	In	S6	nACK	เป็นพัลส์ลอจิก “0” ที่ส่งมาจากเครื่องพิมพ์เพื่อบอกว่าได้รับข้อมูลที่ส่งไปแล้ว
11	Status	In	$\overline{S7}$	$\overline{BUSY}$	เป็นสัญญาณแจ้งมาจากเครื่องพิมพ์ว่ายังไม่พร้อมรับข้อมูล
12	Status	In	SS	PE	แจ้งกระดาษหมด
13	Status	In	S4	SELECT	แจ้งว่าเครื่องพิมพ์ต่ออยู่
14	Control	Out	$\overline{C1}$	$\overline{AUTO FEED}$	สั่งเครื่องพิมพ์ให้เลื่อนบรรทัด
15	Status	In	$\overline{S3}$	$\overline{ERROR}$	สัญญาณจากเครื่องพิมพ์มายังคอมพิวเตอร์ เพื่อแสดงข้อผิดพลาดจากการพิมพ์
16	Control	Out	C2	$\overline{INIT}$	รีเซตเครื่องพิมพ์โดยให้ลอจิก “0”
17	Control	Out	$\overline{C3}$	$\overline{SELECT-IN}$	ส่งสัญญาณไปยังเครื่องพิมพ์เพื่อแจ้งว่าต้องการเลือกเครื่องพิมพ์เครื่องนี้
18-25		-		GND	กราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.3 พอร์ต Control

พอร์ต Control ใช้สำหรับคอมพิวเตอร์ควบคุมเครื่องพิมพ์ จากตารางที่ 2.15 จะเห็นว่าพอร์ต Control ประกอบไปด้วยบิตเอาต์พุต 4 บิต ที่ต่อออกไปยังเครื่องพิมพ์ ส่วนบิตอินาเบิลอินเตอร์รัพท์ไม่ได้ถูกต่อออกไป บล็อกไดอะแกรมของพอร์ต Control เอาต์พุตของพอร์ต Control มีอินเวอร์เตอร์แบบคอลเล็กเตอร์เปิดต่อร่วมอยู่ โดยเอาต์พุตเหล่านี้จะถูกพูลอัปไว้ด้วยตัวต้านทานค่า 4.7 kΩ สำหรับบิต C2 จะผ่านอินเวอร์เตอร์ถึงสองตัวทำให้ที่เอาต์พุตของบิต C2 ไม่มีการกลับสถานะลอจิก

สถานะของพอร์ต Control สามารถอ่านกลับได้โดยการใช้บัฟเฟอร์เบอร์ 74LS240 ซึ่งเอาต์พุตของ 74LS240 มีอินเวอร์เตอร์อยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่านและเขียนข้อมูลกับพอร์ต Control คอมพิวเตอร์จะส่งข้อมลมาที่ขา Control Write และ Control Read

เนื่องจากเอาต์พุตของพอร์ต Control เป็นแบบคอลเล็กเตอร์เปิด ดังนั้น ผู้ใช้งานสามารถใช้พอร์ตนี้ในการอ่านค่าสัญญาณอินฟุตจากภายนอกได้โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านค่ามีลอจิก “1” เสียก่อน

2.10.4 พอร์ตแสดงสถานะหรือพอร์ต Status

พอร์ต Status เป็นพอร์ตที่ใช้สำหรับการอ่านค่าสถานะของเครื่องพิมพ์ รายละเอียดภายในของพอร์ต Status จะสังเกตได้ว่ามีขาสัญญาณอยู่ทั้งหมด 5 สัญญาณด้วยกัน และจะเรียกชื่อเป็น S3, S4, S5, S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจิสเตอร์ Status นั่นเอง สำหรับบิต S7 จะมีข้อแตกต่างจากบิตอื่นๆ ที่เมื่อสัญญาณจากภายนอกส่งเข้ามาแล้วจะไม่ผ่านอินเวอร์เตอร์ ในขณะที่ขาอื่นๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้น เมื่อข้อมูลผ่านจากขาอินพุตไปยัง 74LS240 ซึ่งเอาต์พุตมีการกลับสถานะทำให้บิต S7 เป็นบิตเดียวที่มีการกลับสถานะ นอกจากนี้ในการใช้งานถ้าต้องการให้มีการสร้างสัญญาณอินเตอร์รัปต์จากขอบขาขึ้นของขา S6 สามารถกำหนดค่าได้จากพอร์ต Control บิต 4

2.10.5 การนำพอร์ตขนานไปใช้งาน

สำหรับพอร์ตขนานแบบมาตรฐาน ผู้ใช้งานสามารถนำพอร์ตอินพุต 5 บิต (พอร์ต Status) พอร์ตเอาต์พุท 4 บิต (พอร์ต Control) และพอร์ตเอาต์พุตอีก 8 บิต (พอร์ต Data) ไปใช้งานได้โดยตรง โดยที่ 4 บิตของพอร์ตเอาต์พุตหรือพอร์ต Control นั้นสามารถดัดแปลงให้ใช้งานเป็นพอร์ตอินพุตขนาด 4 บิต ได้ด้วย ดังนั้น ผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ตขนานที่มีมากถึง 17 เส้นไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.6 การเขียนโปรแกรมติดต่อกับพอร์ตขนาน

พอร์ตขนานของคอมพิวเตอร์จะมีลักษณะเช่นเดียวกับอุปกรณ์อินพุตเอาต์พุตตัวอื่นๆ คือ เมื่อต้องการติดต่อจะต้องกำหนดแอดเดรสที่ต้องการติดต่อด้วย ตารางที่ 2.16 แสดงแอดเดรสของพอร์ตขนาน โดยแบ่งออกเป็น 3 ตำแหน่ง คือ แอดเดรสของรีจิสเตอร์ Data, รีจิสเตอร์ Status และ รีจิสเตอร์ Control โดยแอดเดรสนี้จะมีอยู่ทั้งหมด 3 ชุด คือ LPT1, LPT2 และ LPT 3

ตารางที่ 2.16 แสดงแอดเดรสของพอร์ตขนาน

ชื่อพอร์ต	LPT1:		LPT2:		LPT3:	
	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก
DATA	888	378H	956	3BCH	632	278H
STATUS	889	379H	957	3BDH	633	279H
CONTROL	890	37AH	958	3BEH	634	27AH

เมื่อต้องการติดต่อกับพอร์ตขนานในตำแหน่งใด ก็ให้ส่งค่าข้อมูลออกไปที่พอร์ตขนานในตำแหน่งนั้นๆ ยกตัวอย่างการเขียนโปรแกรมด้วย QBASIC เพื่อส่งค่าลอจิก “1” ออกไปทุกบิตของพอร์ต Data ของ LPT1 จะต้องเขียนโปรแกรมดังนี้

OUT & H378, &HFF

โดยที่

เครื่องหมาย &H ที่แสดงนั้นหมายถึงตัวเลขฐานสิบหก

คำสั่ง OUT เป็นการส่งค่าข้อมูลออกเอาต์พุตของพอร์ตอินพุตเอาต์พุต

ค่า 378 เป็นแอดเดรสของรีจิสเตอร์ Data สำหรับ LPT1

ค่าข้อมูล FF เป็นข้อมูลเลขฐานสิบหก ซึ่งหมายถึง การให้บิตทุกบิตของรีจิสเตอร์ Data มีลอจิกเป็น “1” นั่นเอง

ส่วนการอ่านค่าจากพอร์ตขนานมายังคอมพิวเตอร์ผ่านทางพอร์ต Status ของ LPT1 สามารถเขียนโปรแกรมด้วย QBASIC ได้ดังนี้

Temp = INP (&H379)

โดยที่

คำสั่ง INP(_) เป็นคำสั่งสำหรับการอ่านค่าข้อมูล

ค่า 379 เป็นตำแหน่งแอดเดรสของรีจิสเตอร์ Status สำหรับ LPT1 ในตัวเลขฐานสิบหก

ตัวแปร Temp เป็นตัวแปรที่ใช้เก็บข้อมูลที่อ่านได้จากพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual BASIC

การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual BASIC ชุดคำสั่งส่วนใหญ่จะมีรูปแบบใกล้เคียงกับ QBASIC แต่ Visual BASIC จะไม่มีคำสั่งสำหรับการติดต่อกับพอร์ตโดยตรงคือ คำสั่ง INP( ) และคำสั่ง OUT เหมือนกับ QBASIC ดังนั้น เพื่อให้สามารถติดต่อกับพอร์ตขนานได้จึงจำเป็นต้องเพิ่มโปรแกรมบางตัวเข้าไป โดยโปรแกรมที่เพิ่มเข้าไปนี้ จะอยู่ในรูปของ DLL (Dynamic Linked Library)

ไฟล์ DLL นี้จะมีอยู่ 2 ไฟล์ คือ inpout.dll และ inpout32.dll โดย inpout.dll นั้นใช้สำหรับระบบปฏิบัติการ 16 บิต หรือวินโดวส์ 3.1 นั่นเอง ส่วน inpout32.dll จะใช้สำหรับระบบปฏิบัติการที่เป็น 32 บิตซึ่งก็คือวินโดวส์ 95 หรือวินโดวส์ 98

2.11.1 สำหรับระบบปฏิบัติการ 32 บิต

Public Declare Function INP Lip Winpout32.dll"

Alias "Inp32" (ByVal PortAddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32.dll"

Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

2.11.2 การติดต่ออุปกรณ์อินพุต-เอาต์พุตอย่างง่าย

จากทฤษฎี สามารถแบ่งรีจิสเตอร์ของพอร์ตขนานออกได้เป็น 3 รีจิสเตอร์ คือ

1. รีจิสเตอร์ Data ทำหน้าที่เป็นเอาต์พุต
2. รีจิสเตอร์ Status ทำหน้าที่เป็นอินพุต
3. รีจิสเตอร์ Control ทำหน้าที่เป็นเอาต์พุต

ดังนั้น ถ้าผู้ใช้งานต้องการส่งค่าออกเอาต์พุตก็จะต้องใช้รีจิสเตอร์ Data หรือรีจิสเตอร์ Control ส่วนถ้าต้องการรับค่าจากอินพุต ผู้ใช้งานต้องใช้รีจิสเตอร์ Status ในการอ่านค่าอินพุต การจะอ้างถึงรีจิสเตอร์แต่ละตัวนั้น ผู้ใช้งานจะต้องใช้ตำแหน่งแอดเดรสเป็นตัวอ้าง ดังมีรายละเอียดแสดงในตารางที่ 2.16

โดยปกติแล้ว Visual BASIC ไม่มีคำสั่ง OUT เหมือนกับ QBASIC ดังนั้นเพื่อให้ Visual BASIC สามารถใช้คำสั่งนี้ได้จำเป็นต้องเพิ่มไฟล์ Inpout32.bas เข้าไปใน Project ของ Visual BASIC ที่กำลังใช้งานอยู่ โดยต้องทำตามขั้นตอนดังนี้

1. ไปที่เมนู Project เรียกคำสั่ง Add File แล้วเพิ่มไฟล์ Inpout32.bas ลงไปใน Project ที่หน้าต่าง Project จะปรากฏไฟล์ Inpout32.bas

2. เมื่อเลือกชี้ที่ไฟล์ Inpourt32.bas แล้วใช้คำสั่ง View Code เพื่อดูรายละเอียดภายในของไฟล์ Inpout32.bas

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.17 แสดงแอดเดรสรีจิสเตอร์ของพอร์ตขนาน

รีจิสเตอร์	LPT1	LPT2	LPT3
DATA	378H	3BCH	278H
STATUS	379H	3BDH	279H
CONTROL	37AH	3BEH	27AH

3. สำหรับไฟล์ INPOUT32.BAS นั้นจะไปกำหนดคำสั่ง INP และ OUT ให้กับ Visual BASIC โดยจะต้องมีไฟล์ Inpout32.dll บรรจุอยู่ในไดเรกเตอรี่ System อยู่ก่อนแล้ว โดยโปรแกรม INPOUT32.BAS มี คำสั่ง ดังนี้

```
#If Win32 Then
'Declare Inp and Out for port I/O
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
#Else
Declare Function Inp Lib "InpOut.DLL" (ByVal Port%) As Integer
eclare Sub Out Lib "InpOut.DLL" (ByVal Port%, ByVal Value%)
End If
```

4. เมื่อถึงขั้นตอนนี้ผู้ใช้งานสามารถใช้คำสั่ง INP และคำสั่ง OUT ในโปรแกรม เพื่อรับและส่งค่ากับพอร์ตขนานได้แล้ว

2.12 การเชื่อมต่อพอร์ตอนุกรมกับระบบบัส I^2C

I^2C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซี โดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูลอีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

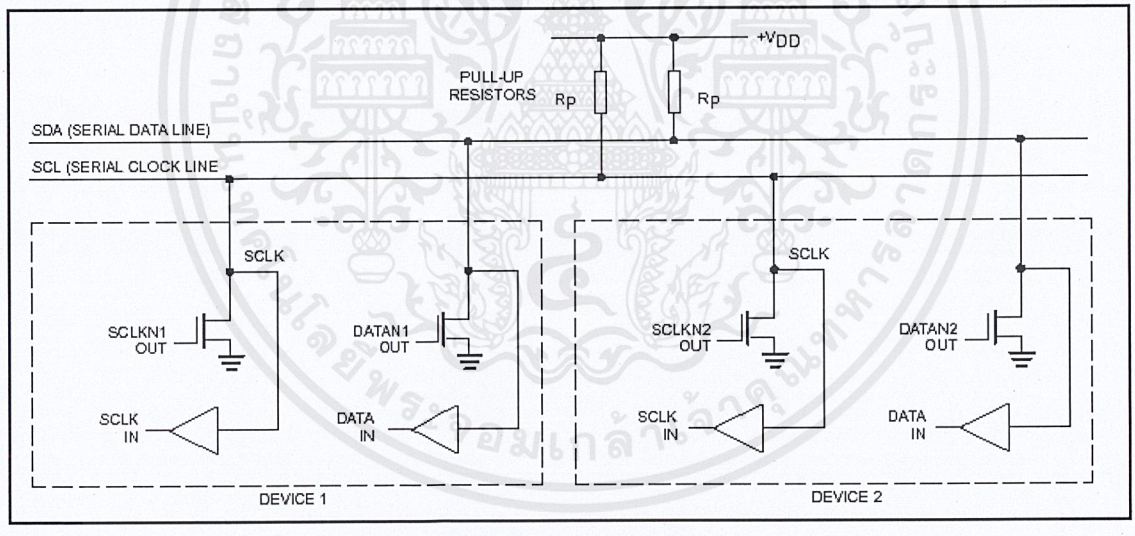
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock Line) ในการอธิบายต่อไปนี้จะเรียกสายสัญญาณทั้งสองว่า สาย SDA และ SCL

อุปกรณ์ที่ใช้การเชื่อมต่อด้วยบัส I^2C มีหลากหลาย ไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O Expander) , ไอซีแปลงสัญญาณอะนาลอกเป็นดิจิตอล (ADC) และแปลงสัญญาณดิจิตอลเป็นอะนาลอก (DAC), ไอซีรีลไทม์คล็อก (RTC), ไอซีขับโมดูล LCD, หน่วยความจำอีพรอม และไมโครคอนโทรลเลอร์

2.12.1 คุณสมบัติโดยทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-directional line) ต้องมีการต่อตัวต้านทานพูลอัปกับแรงดัน +5V ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสองวงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (Open-drain) หรือคอนเล็คเตอร์เปิด (Open-collector) ดังแสดงรายละเอียดในภาพที่ 2.37

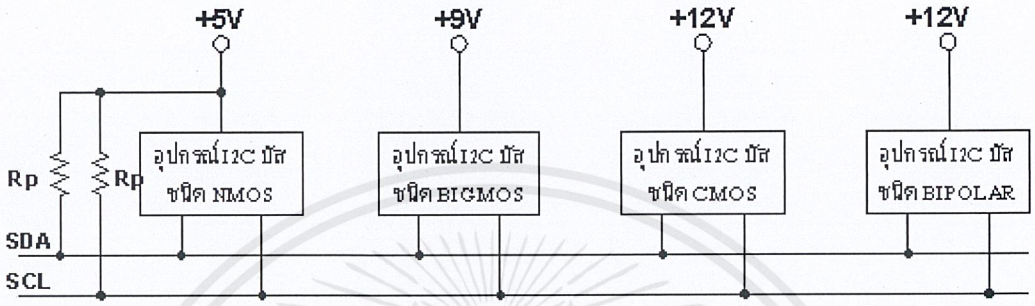


ภาพที่ 2.37 โครงสร้างวงจรเอาต์พุตของอุปกรณ์ที่ใช้การเชื่อมต่อบนระบบบัส I^2C

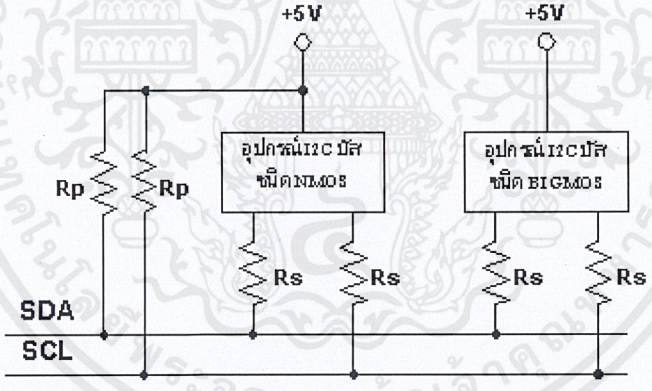
อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิต ต่อวินาที ในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิต ต่อวินาที ในโหมดความเร็วสูง (Fast Mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400pF การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่า คือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเด่นอีกประการหนึ่งของบัส I^2C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยอุปกรณ์บนบัส I^2C ตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวใช้ไฟเลี้ยง +12V การต่อร่วมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัป (Rp) เข้ากับแรงดัน +5V ไว้ด้วยเสมอ ดังแสดงในภาพที่ 2.38



ภาพที่ 2.38 แสดงการเชื่อมต่ออุปกรณ์ระบบบัส I^2C ที่ใช้ไฟเลี้ยงไม่เท่ากัน



ภาพที่ 2.39 การต่อตัวต้านทานเพื่อป้องกันแรงดันกระชากที่อาจปะปนเข้ามาในไฟเลี้ยงของอุปกรณ์ในระบบบัส I^2C

ในกรณีที่อาจมีแรงดันไฟกระชากขนาดใหญ่ปะปนเข้ามาในบัส I^2C ที่ขา SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้น จึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า โปรโตคอล (Protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I^2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I^2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (Transmitter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (Receiver)

ในอุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว

อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการทำงานหรือการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์ (Master)

อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ (Slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

2.12.2 สภาวะที่เกิดขึ้นบนบัส I^2C

มีด้วยกัน 5 สภาวะ ดังนี้

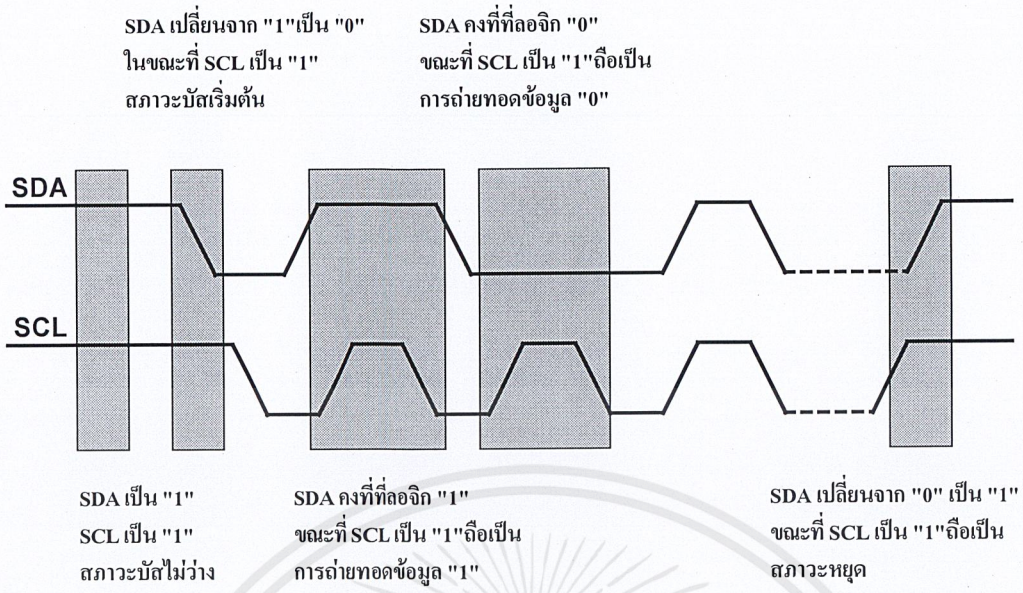
1) บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

2) เริ่มต้นการถ่ายทอดข้อมูล (Start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น (START)

3) ข้อมูลดำรงอยู่บนบัส(Data Valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้นโดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือ ข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง เกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้นอุปกรณ์มาสเตอร์ที่ ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

4) รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิต เรียกว่า บิตรับรู้ (Acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้ที่มีสถานะลอจิกต่ำเพื่อตอบสนองให้ทราบว่าได้รับข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.40 ไดอะแกรมเวลาแสดงสภาวะต่างๆ ที่เกิดขึ้นบนระบบบัส I^2C

5) หยุดการถ่ายทอดข้อมูล (Stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (Stop)

ในภาพที่ 2.40 เป็นไดอะแกรมเวลาที่แสดงถึงการเกิดสภาวะต่างๆ บนบัส I^2C ไม่ว่าจะเป็นสภาวะบัสว่าง, เริ่มต้น, ถ่ายทอดข้อมูล, รับรู้และหยุดการถ่ายทอดข้อมูล

2.12.3 การทำงานบนบัส I^2C

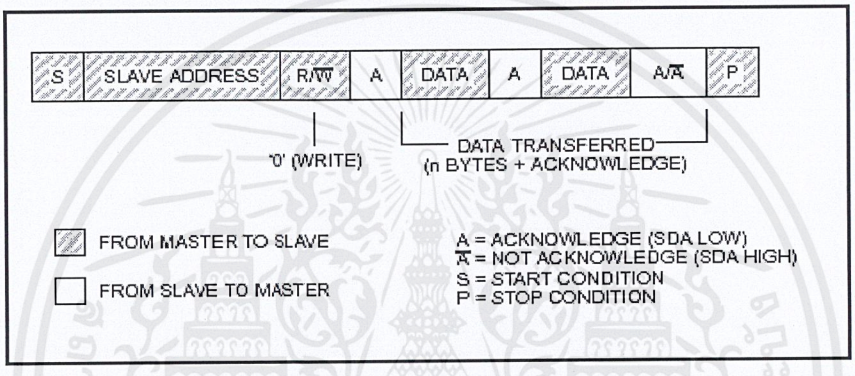
ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่างๆที่ต่อยู่บนบัส ต้องมีการอ้างถึงอุปกรณ์เสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I^2C นั้นจะใช้การอ้างถึงแบบ 7 บิต หรือ 10 บิตในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มาก ก็ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิด หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

ดังนั้นหัวใจสำคัญในอันดับแรกของการทำงานบนบัส I^2C คือการอ้างถึงอุปกรณ์แต่ละตัว ซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงแบบ 7 บิต เพียงรูปแบบเดียว

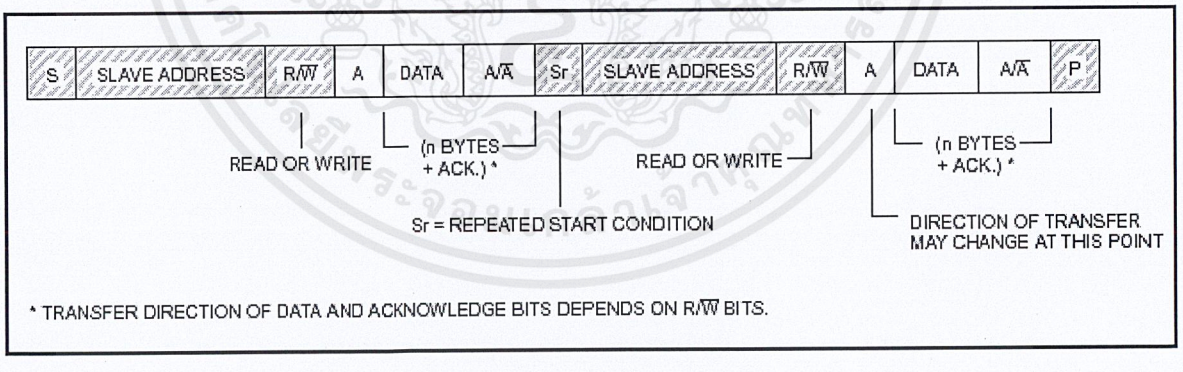
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.4 การอ้างถึงแบบ 7 บิต (7-bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ โดยมีรูปแบบแสดงในภาพที่ 2.41 ใน 7 บิต บนรวมทั้งบิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อโดยแบ่งเป็นบิตกำหนดแอดเดรสคงที่ (Fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิต เป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบัส I^2C ส่วนในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB เป็น “0” หมายถึง ต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น “1” จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ



ภาพที่ 2.41 รูปแบบของข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์บนระบบบัส I^2C



ภาพที่ 2.42 รูปแบบข้อมูลอนุกรมที่ใช้ในการติดต่อกับอุปกรณ์บนระบบบัส I^2C แบบ 7 บิต

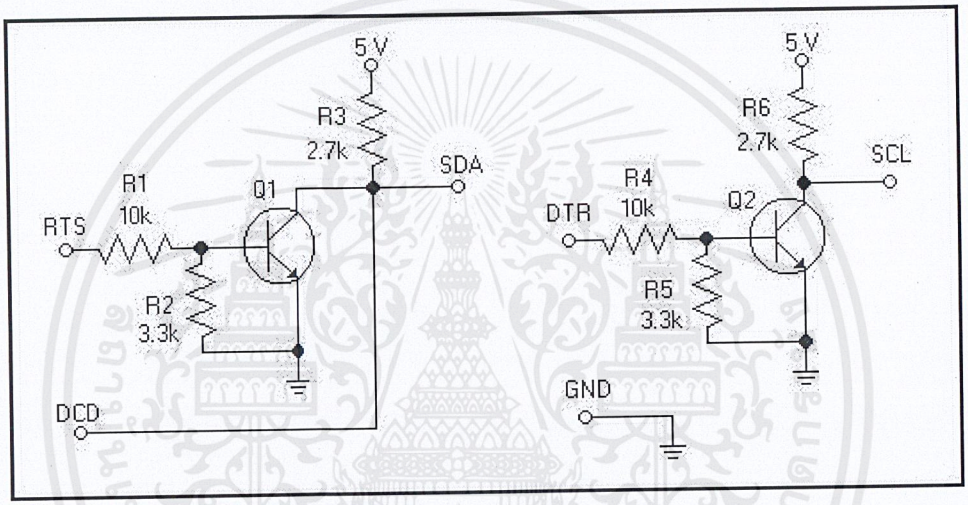
ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (Control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่าง ไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่าบิตใดเป็นอินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้งเพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในภาพที่ 2.42 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I^2C ของการอ้างถึงแบบ 7 บิต

2.12.5 วงจรแปลงสัญญาณพอร์ตอนุกรมเพื่อเชื่อมต่อกับระบบบัส I^2C

เพื่อให้สายสัญญาณของพอร์ตอนุกรมจากคอมพิวเตอร์สามารถติดต่อกับอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I^2C ต้องมีวงจรเพื่อทำหน้าที่สร้างสายสัญญาณ SDA และ SCL ขึ้นจากสัญญาณของพอร์ตอนุกรม ดังมีตัวอย่างวงจรตามภาพที่ 2.43 โดยในการเชื่อมต่อคอมพิวเตอร์เข้าสู่บัส I^2C นี้ต้องกำหนดให้คอมพิวเตอร์เป็นอุปกรณ์มาสเตอร์เท่านั้น



ภาพที่ 2.43 วงจรสร้างสายสัญญาณเพื่อเชื่อมต่อกับระบบบัส I^2C ของพอร์ตอนุกรม

ทรานซิสเตอร์ Q1 และ Q2 ได้รับการจัดวงจรให้มีลักษณะเป็นวงจรบัฟเฟอร์แบบคอลเล็กเตอร์เปิดตามข้อกำหนดของวงจรเอาต์พุตของบัส I^2C โดย Q1 ใช้สำหรับถ่ายทอดสัญญาณของสาย SDA ในขณะที่ Q2 ทำหน้าที่ถ่ายทอดสัญญาณของสาย SCL ไฟเลี้ยงของวงจรคือ +5V จึงสามารถใช้ทรานซิสเตอร์ชนิดเอ็นพีเอ็นเบอร์มาตรฐานเบอร์ใดก็ได้ที่สามารถตอบสนองความถี่ได้สูงถึง 100 kHz ข้อมูลจากคอมพิวเตอร์จะถ่ายทอดสู่สาย SDA ของบัส I^2C ผ่านทางขา RTS และรับข้อมูลเข้าทางขา DCD ส่วนสายสัญญาณนาฬิกาที่ออกจากพอร์ตอนุกรมจะถูกส่งออกมาทางขา DTR เพื่อใช้เป็นสาย SCL สำหรับบัส I^2C

2.12.6 อุปกรณ์ที่ใช้การเชื่อมต่อแบบบัส I^2C

ในปัจจุบัน บัส I^2C ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ด้วยข้อดีที่ชัดเจน คือ ใช้สายสัญญาณเพียง 2 เส้นเท่านั้น และการขยายระบบไมโครคอนโทรลเลอร์ที่มีจำนวนอินพุตเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และหน่วยความจำ มีจำกัดสามารถทำได้ง่ายขึ้นด้วยระบบบัส I^2C เมื่อเป็นเช่นนี้จึงมีอุปกรณ์เพอริเฟอรัลที่ใช้การเชื่อมต่อแบบ I^2C มากมายจากหลายผู้ผลิตออกมาให้ได้งานกัน ดังตัวอย่าง

ไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O Expander): PCF8574, PCF8582, PCF8584

ไอซีหน่วยความจำอีอีพรอมอนุกรม (Serial EEPROM): 24Cxx, PCF8570, PCF8572/73, PCF8582

ไอซี ADC/DAC: PCF8591

ไอซีรีลไทม์คล็อก (Real-time clock: RTC): PCF8583, PCF8593, PCF8598, 41T56C

ไอซีขับ LCD โมดูล (LCD driver): PCF8466, PCF8576, PCF8577/78, PCF8579, SAA1064

ไอซีกำเนิดสัญญาณ DTMF (DTMF generator): PCD3311/12

2.12.7 การเขียนโปรแกรมติดต่อบัส I^2C ด้วย Visual BASIC

จากส่วนของฮาร์ดแวร์จะเห็นว่า ขาที่ใช้งานในการติดต่ออุปกรณ์บนระบบบัส I^2C นั้นใช้ขาของพอร์ตอนุกรมทั้งหมด 3 ขา คือ DTR, RTS และ DCD ขาทั้งสามนี้สามารถควบคุมด้วยคอนโทรล MSComm1 ของ Visual BASIC ซึ่งมีรูปแบบการใช้งานดังนี้

การกำหนดให้ขา SDA เป็น “1” ต้องเขียนคำสั่งเป็น MSComm1. RTSEnable = True

การกำหนดให้ขา SDA เป็น “0” ต้องเขียนคำสั่งเป็น MSComm1. RTSEnable = False

การกำหนดให้ขา SCL เป็น “1” ต้องเขียนคำสั่งเป็น MSComm1. DTREnable = True

การกำหนดให้ขา SCL เป็น “0” ต้องเขียนคำสั่งเป็น MSComm1. DTREnable = False

การอ่านค่าอินพุตของระบบบัส I^2C นั้น จะใช้คำสั่งเพื่อตรวจสอบค่าจาก CDHolding โดยจะต้องตรวจสอบว่าค่าที่ได้เป็น True หรือ False

2.12.8 โปรแกรมย่อยสร้างสัญญาณเริ่มต้นหรือ START

จากทฤษฎีในตอนต้นกำหนดไว้ว่า เมื่อต้องการติดต่อกับอุปกรณ์บนระบบบัส I^2C ในครั้งแรกจะต้องส่งสัญญาณเริ่มต้นหรือ Start ออกไปก่อน โดยมีขั้นตอนการสร้างสัญญาณ Start ดังนี้

1. กำหนดให้ขา SCL และ SDA มีลอจิกเป็น “1” ก่อนเพื่อให้อยู่ในสภาวะบัสว่าง
2. กำหนดให้ขา SDA มีลอจิก “0”
3. กำหนดให้ขา SCL มีลอจิก “0” ตาม

ผู้ใช้งานสามารถเปรียบเทียบการเขียนโปรแกรมนี้กับไดอะแกรมเวลาได้ เพื่อให้เกิดความเข้าใจมากขึ้น โดยลำดับขั้นต่างๆ ที่กล่าวมาสามารถเขียนเป็นโปรแกรมด้วย Visual Basic ได้ดังนี

```
Private Sub I2Cstart ()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm1. RTSEnable = True      ‘ SDA = 1
MSComm1. DTREnable = True      ‘ SCL = 1
MSComm1. RTSEnable  = False    ‘ SDA = 0
MSComm1. DTREnable = False     ‘ SCL = 0
End Sub
```

2.12.9 โปรแกรมย่อยการสร้างสัญญาณหยุดหรือ Stop

การทำให้เกิดสถานะหยุดหรือสิ้นสุดการถ่ายทอดข้อมูลบนบัส I^2C หรือสถานะหยุด (Stop) ก็คือ การทำให้บัสของ I^2C เข้าสู่สภาวะบัสว่าง มีขั้นตอนดังนี้

1. กำหนดให้ขา SDA มีลอจิก “0” ก่อน
2. จากนั้นกำหนดให้ขา SCL มีลอจิกเป็น “1”
3. ทำให้ขา SDA มีลอจิกเป็น “1”

สามารถนำขั้นตอนทั้งหมดมาเขียนเป็นโปรแกรมย่อยด้วย Visual BASIC ได้ดังนี้

```
Private Sub I2Cstop ()
MSComm1. RTSEnable = False     ‘ SDA = 0
MSComm1. DTREnable = True      ‘ SCL = 1
MSComm1. RTSEnable  = True     ‘ SDA = 1
End Sub
```

2.12.10 โปรแกรมย่อยรอรับสัญญาณรับรู้หรือ Acknowledge

เมื่อมีการส่งข้อมูลจากอุปกรณ์มาสเตอร์ไปยังอุปกรณ์ สเลฟ อุปกรณ์สเลฟจะตอบกลับสัญญาณที่ส่งออกไปนี้ด้วยสัญญาณ รัลรู้หรือ Acknowledge (ACK) เพื่อเป็นการบอกให้รู้ว่าได้รับข้อมูลเรียบร้อยแล้ว แต่เมื่อมาเขียนโปรแกรมบน Visual BASIC การทำงานที่มีลักษณะเป็นมัลติทาสกิ้ง (Multi-tasking) การเขียนโปรแกรมเพื่อให้รอรับสัญญาณ ACK นั้นจะทำให้โปรแกรมหยุดชะงัก จึงต้องเขียนโปรแกรมเสมือนว่าได้รับสัญญาณ ACK แทน โดยทำให้บัสว่างแล้วส่ง SCL เป็น “0” ออกไปเพื่อแสดงว่าได้รับ ACK แล้ว สามารถเขียนโปรแกรมได้ดังนี้

```
Private Sub Ack ()
MSComm1. RTSEnable = True      ‘ SDA = 1
MSComm1. DTREnable = True      ‘ SCL = 1
MSComm1. DTREnable = False     ‘ SCL = 0
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.11 โปรแกรมย่อยส่งข้อมูล

การส่งข้อมูลลอจิก “0” ก็คือ การทำให้ขา SDA มีลอจิกเป็น “0” แล้วทำการส่งสัญญาณนาฬิกาออกไป โดยมีขั้นตอนดังนี้

1. ทำให้ขา SDA เป็น “0” สำหรับการส่งข้อมูลลอจิก “0”
2. ให้ขา SCL เป็น “1” สำหรับการป้อนสัญญาณนาฬิกาในขณะที่ SDA ยังคงเป็น “0”
3. ให้ขา SCL กลับสู่สถานะเดิมคือลอจิก “0”

สามารถเขียนโปรแกรมได้ดังนี้

```
Private Sub Send0 ()
MSComm1. RTSEnable = False      ‘ SDA = 0
MSComm1. DTREnable = True       ‘ SCL = 1
MSComm1. DTREnable = False      ‘ SCL = 0
End Sub
```

สำหรับการส่งข้อมูลลอจิก “1” มีขั้นตอนดังนี้

1. ทำให้ขา SDA มีลอจิก “1” สำหรับการส่งข้อมูลลอจิก “1”
2. ทำให้ขา SCL เป็น “1” ในขณะ SDA ยังคงเป็น “1”
3. ทำให้ขา SCL มีลอจิก “0” เหมือนเดิม

สามารถเขียนโปรแกรมได้ดังนี้

```
Private Sub Send1 ()
MSComm1. RTSEnable = True       ‘ SDA = 1
MSComm1. DTREnable = True       ‘ SCL = 1
MSComm1. DTREnable = False      ‘ SCL = 0
End Sub
```

2.12.12 โปรแกรมย่อยอ่านและส่งข้อมูลขนาด 8 บิต

ในการอ่านข้อมูลจากอุปกรณ์ในระบบบัส I^2C ส่วนใหญ่ข้อมูลที่อ่านได้จะมีขนาด 8 บิต โดยขั้นตอนการอ่านค่าข้อมูลมีดังนี้

1. กำหนดลูปในการรับค่าอินพุตไว้ 8 รอบ สำหรับการอ่านค่าอินพุต 8 บิต
2. ให้บัสเป็นบัสว่างเพื่อรอรับสัญญาณจากอุปกรณ์ สเลฟ โดยกำหนดให้ SDA และ SCL มีลอจิก “1”
3. ทำการอ่านค่าข้อมูลจากขา SDA ผ่านคำสั่ง CDHolding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ถ้าอ่านค่าบิตเข้ามาแล้วมีค่าเป็น “1” ให้นำค่าบิตนั้น OR กับตัวแปรที่ทำการเก็บค่า แต่ถ้าค่าบิตที่อ่านได้เป็น “0” ไม่ต้องเปลี่ยนแปลงใดๆ

5. กำหนดให้ขา SCL มีลอจิก “0” เพื่อเป็นการตอบรับการรับข้อมูล

6. ถ้าข้อมูลที่ทำการอ่านค่ายังไม่ครบ 8 บิต ให้ย้อนกลับไปอ่านค่าในบิตต่อๆ ไปจนครบ

7. นำค่าตัวแปรคืนค่าให้กับฟังก์ชัน

จากขั้นตอนที่กล่าวมานี้สามารถนำมาเขียนเป็นโปรแกรมย่อยได้ดังนี้

```
Private Function dat ()
Dim DAT1 AS Byte
For i – 7 To 0 Step –1
MSComm1.RTSEnable = True            'SDA = 1
MSComm1.DTREnable = True            'SCL = 1
If (MSComm1.CDHolding And &H80) < > &H80 Then
DAT1 = 2 ^ i Or DAT1
End If
MSComm1.DTREnable = False           ' SCL = 0
Next i
Dat = DAT1
End Function
```

การส่งค่าข้อมูลไปยังอุปกรณ์บนระบบบัส I^2C ปกติจะเป็นการส่งข้อมูลขนาด 8 บิต หรือ 1 ไบต์ โปรแกรมย่อยต่อไปนี้จะเป็นการนำเอาข้อมูล 8 บิต มาทำการตรวจสอบว่าเป็นลอจิก “0” หรือ “1” แล้วทำการส่งข้อมูลเหล่านั้นออกไป

```
Private Sub Send8BIT (Add As Byte)
For i – 7 To 0 Step –1                  ' Loop 8 Cycle
If (Add And 2 ^ i) = 2 ^ i Then         ' Test Bit 0 OR 1
Call Send1
Else
Call Send0
End If
End Sub
```

ในทนี้จะกล่าวถึงการทำงานและการเขียนโปรแกรมขั้นต้นเพื่อติดต่อกับระบบบัส I^2C ขอให้ผู้เรียนจดจำหลักการและนำโปรแกรมย่อยที่ใช้สร้างสถานะการทำงานบนระบบบัส I^2C มาใช้ประโยชน์เพื่อติดต่อกับอุปกรณ์บนระบบบัส I^2C ซึ่งจะได้ทำการทดลองต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.13 พอร์ตอินพุตเอาต์พุตให้แก่พอร์ตอนุกรมด้วยระบบบัส I^2C

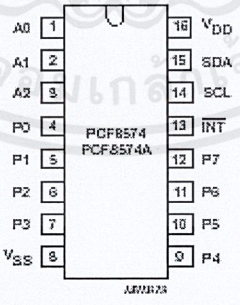
ด้วยข้อจำกัดของจำนวนขาสัญญาณของพอร์ตอนุกรม หากต้องการให้สามารถตอบสนองระบบหรืองานที่ต้องการพอร์ตอินพุตเอาต์พุตจำนวนมากได้นั้น จำเป็นต้องต่ออุปกรณ์ภายนอกเพิ่มเติม เพื่อขยายจำนวนพอร์ตอินพุตเอาต์พุต จากการเรียนรู้และทดลอง ในการขยายพอร์ตเอาต์พุตสามารถใช้ไอซีชิฟต์รีจิสเตอร์ช่วยได้ และที่จะเรียนรู้ต่อไปนี้เป็นการขยายพอร์ตด้วยอุปกรณ์บนระบบบัส I^2C เพื่อช่วยให้พอร์ตอนุกรมมีจำนวนพอร์ตใช้งานเพิ่มมากขึ้น โดยใช้สายสัญญาณของพอร์ตอนุกรมเพียง 3 เส้น แต่สามารถขยายพอร์ตอินพุตเอาต์พุตได้มากถึง 128 บิต

อุปกรณ์ที่แนะนำคือไอซีขยายพอร์ตเบอร์ PCF8574 และ PCF8574A โดยไอซีแต่ละเบอร์สามารถขยายพอร์ตอินพุตเอาต์พุตได้ 8 ช่อง หรือ 8 บิต สามารถต่อพ่วงกันได้มากถึง 8 ตัว ทำให้สามารถต่อพอร์ตอินพุตเอาต์พุตได้มากถึง 64 ช่องต่อไอซี 1 เบอร์ ถ้าหากนำ PCF8574 และ PCF8574A มาต่อพ่วงกัน จะเพิ่มจำนวนพอร์ตได้อีก 64 บิต รวมเป็น 128 บิต โดยการเชื่อมต่อนั้นต้องกระทำบนบัส I^2C เท่านั้น

2.13 ข้อมูลเบื้องต้นของ PCF8574/8574A

ไอซี PCF8574 และ PCF8574A มีคุณสมบัติดังนี้

- ทำงานที่ระดับแรงดันตั้งแต่ 2.5V ถึง 6V
- กินกระแสในสภาวะสแตนต์บายต่ำเพียง 10 μA
- ใช้การเชื่อมต่อแบบบัส I^2C
- มีเอาต์พุตอินเตอร์รัปต์แบบเดรนเปิด



ภาพที่ 2.44 การจัดขาของ PCF8574/8574A

- เอาต์พุตสามารถแลตช์หรือคงค่าไว้ได้ และขับ LED ได้ โดยต่อวงจรในลักษณะกระแสซิงก์ หรือทำงานด้วยลอจิก “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถกำหนดแอดเดรสของไอซีแต่ละตัวได้ทางฮาร์ดแวร์ ด้วยขา A0-A2 ทำให้สามารถต่อพ่วงกันได้ถึง 8 ตัว ต่อเบอร์
- สามารถต่อพ่วงทั้ง PCF8574 และ PCF8574A ได้ ทำให้สามารถขยายพอร์ตได้มากถึง 128 บิต

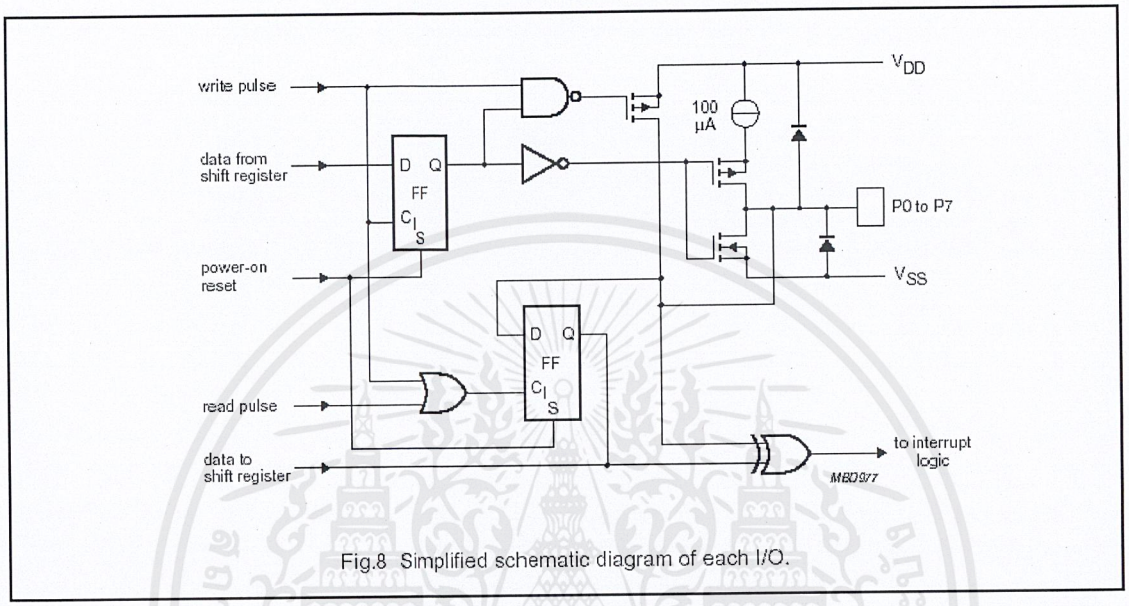


Fig.8 Simplified schematic diagram of each I/O.

ภาพที่ 2.45 วงจรภายในของขาพอร์ตของไอซี PCF8574/8574A

การจัดขาและหน้าที่การทำงานของแต่ละขาของไอซี PCF8574 แสดงในภาพที่ 2.44

ขาพอร์ตทั้ง 8 ขา ของ PCF8574 สามารถกำหนดให้เป็นอินพุตหรือเอาต์พุตได้โดยอิสระ โดยไม่จำเป็นต้องใช้คำสั่งควบคุมเพื่อเลือกให้เป็นขาเอาต์พุตหรือขาอินพุต ลักษณะวงจรภายในของพอร์ตอินพุตเอาต์พุตแสดงในภาพที่ 2.45 เมื่อจ่ายไฟให้กับ PCF8574 ครั้งแรก ขาพอร์ตทั้ง 8 ขา จะมีลอจิกเป็น “1” ซึ่งจะเป็นการจ่ายกระแสมาจากแหล่งกระแสคงที่ภายในตัวไอซี ทำให้มีกระแสในขณะลอจิก “1” นี้เพียง 100 µA เท่านั้น จึงไม่เหมาะที่จะใช้ไอซี PCF8574 ในการขับกระแสซอร์ส แต่จะเหมาะกับการนำ PCF8574 ไปขับกระแสซิงก์มากกว่า

เมื่อต้องการให้ขาพอร์ตเหล่านี้ทำหน้าที่เป็นอินพุตจะต้องส่งสัญญาณให้ขาเหล่านี้มีลอจิก “1” เสียก่อน เมื่อขาอินพุตได้รับสัญญาณจากภายนอกป้อนเข้ามา ไอซี PCF8574 จะสร้างสัญญาณอินเตอร์รัพท์ป้อนให้ไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์รับรู้แทนการตรวจสอบขาอินพุตตลอดเวลา สัญญาณอินเตอร์รัปต์นี้จะถูกรีเซตเมื่อมีการอ่านค่าข้อมูลหรือมีการเปลี่ยนค่าของอินพุตไปสู่ค่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ PCF8574 และ PCF8574A มีการทำงานเหมือนกันทุกประการ แตกต่างกันเพียงค่าแอดเดรสที่ใช้ในการติดต่อเท่านั้น ด้วยเหตุนี้จึงสามารถนำไอซีทั้งสองเบอร์นี้มาต่อพ่วงกันเพื่อขยายพอร์ตได้มากถึงเบอร์ละ 8 ตัว รวมทั้งสิ้น 165 ตัว ส่งผลให้สามารถขยายพอร์ตได้สูงสุด 128 บิต

2.13.1 การเขียนโปรแกรมควบคุม PCF8574 ด้วย Visual BASIC

จากโปรแกรมย่อยต่างๆ เพื่อสร้างสัญญาณบนระบบบัส การเขียนโปรแกรมเพื่อควบคุม PCF8574 สามารถทำได้ง่ายขึ้น เพียงนำโปรแกรมย่อยต่างๆ เหล่านั้นมาเรียงต่อกันตามรูปแบบของ PCF8574 เท่านั้น เริ่มจากการติดต่อกับ PCF8574 จะต้องมีการกำหนดแอดเดรสและระบุโหมดที่จะใช้ว่าต้องการอ่านหรือเขียนข้อมูลไปยังตัวมัน โดยไบต์แรกที่ใช้เก็บออดเดรสมีรายละเอียดดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	1	0	0	A2	A1	A0	$R/\overline{W}$

บิต A0, A1, A2 ใช้ในการระบุ PCF8574 ที่ใช้บนบอร์ดในกรณีที่มีการต่อ PCF8574 มากกว่า 1 ตัว โดยค่าของ A0-A2 จะมีความแตกต่างกันไปในแต่ละตัว ซึ่งถ้าเทียบกับบอร์ด EX-07 ก็คือ การกำหนดค่าของจั๊มเปอร์นั่นเอง ส่วนบิต $\overline{R/W}$ ใช้กำหนดว่าต้องการอ่านหรือเขียนข้อมูลกับไอซี PCF8574

ไอซี PCF8574 ยังมีอีกเบอร์หนึ่งในอนุกรมเดียวกัน นั่นคือ PCF8574A ซึ่งก็มีข้อมูลกำหนดแอดเดรสที่แตกต่างกับ PCF8574A แต่ฟังก์ชันการทำงานเหมือนกันทุกประการ โดยข้อมูลกำหนดแอดเดรสของ PCF8574A มีดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	1	1	1	A2	A1	A0	$R/\overline{W}$

ดังนั้น จึงสามารถต่อพ่วงไอซีในอนุกรม PCF8574x ได้สูงถึง 16 ตัว

2.13.2 การส่งค่าไปยังเอาต์พุตของ PCF8574A

มีลำดับขั้นตอนดังนี้

1. ส่งสัญญาณ START

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่งข้อมูลกำหนดแอดเดรส โดยในที่นี้จะกำหนดแอดเดรสของ PCF8574A ไว้ที่ 000 (กำหนดจั๊มเปอร์บนบอร์ด Ex-07 ตำแหน่ง A0, A1, A2 เป็น “0” ทั้งหมด) และให้ทำงานในโหมดเขียนข้อมูล โดยการป้อนลอจิก “0” ให้แก่บิต $\overline{R/W}$

3. รอรับสัญญาณ ACK หรือรอรับการตอบกลับจาก PCF8574

4. ส่งข้อมูลไปยังเอาต์พุตของ PCF8574

5. รอรับสัญญาณ ACK อีกครั้ง

6. ส่งสัญญาณ STOP

จากขั้นตอนดังกล่าวสามารถนำมาเขียนเป็นโปรแกรมย่อยได้ดังนี้

```
Private Sub Sendout (B As Byte)
    Call I2Cstart                ‘ Start
    Call Send8BIT (&H40)         ‘ Send Control Word
    Call Ack                     ‘ Acknowledge
    Call Send8BIT(B)             ‘ Send Data
    Call Ack                     ‘ Acknowledge
    Call I2Cstop                 ‘ Stop
End Sub
```

ค่าตัวเลข &H40 ซึ่งอยู่ที่บรรทัด Call Send8BIT (&H40) นั้นก็คือ ค่าแอดเดรสของ PCF8574A ที่ต้องการติดต่อด้วย โดยในโปรแกรมย่อย Send8BIT จะทำการส่งค่าแอดเดรสและบิตควบคุมการเขียนข้อมูลของ PCF8574A ดังนั้น ในบิตสุดท้ายจึงต้องมีค่าเป็น “0”

สำหรับโปรแกรมย่อย Send8BIT (B) จะเป็นการนำค่าตัวแปร B ซึ่งเรียกใช้งานโดยป้อนจากคำสั่ง SendOut ส่งออกพอร์ตของ PCF8574

2.13.3 การอ่านค่าข้อมูลจาก PCF8574A

มีขั้นตอนดังนี้

1. ส่งสัญญาณ START

2. ส่งข้อมูลกำหนดแอดเดรส โดยในที่นี้จะกำหนดแอดเดรสของตัว PCF8574A ไว้ที่ 001 (จั๊มเปอร์บนบอร์ด Ex-07 ตำแหน่ง A1, A2 เป็น “0” ส่วน A0 เป็นลอจิก “1”) และให้ทำงานในโหมดอ่านข้อมูล โดยการป้อนลอจิก “1)

3. รอรับสัญญาณ ACK หรือรอรับการตอบกลับจาก PCF8574A

4. อ่านข้อมูลจาก PCF8574A

5. รอรับสัญญาณ ACK อีกครั้ง

6. ส่งสัญญาณ STOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมย่อย Timer_Timer เป็นโปรแกรมย่อยสำหรับอ่านข้อมูลจาก PCF8574 อย่างต่อเนื่อง โดยการกำหนดค่า Interval ของไทเมอร์ ให้มีค่าเท่ากับ 100 ซึ่งจะทำให้การทำงานของโปรแกรมย่อยไทเมอร์เกิดขึ้นทุกๆ 100 มิลลิวินาที การอ่านค่าจาก PCF8574 ก็จะทำงานอย่างต่อเนื่องโดยมีรายละเอียดการเขียนโปรแกรมดังนี้

```
Private Sub Timer_Timer ()
    Call I2Cstart                ' Start
    Call Send8BIT (&H43)         ' Control Word Read
    Call Ack                     ' Acknowledge
    Text2.Text = DAT             ' Read Data
    Call Ack                     ' Acknowledge
    Call I2Cstop                 ' Stop
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการควบคุม

3.1 วงจรการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์

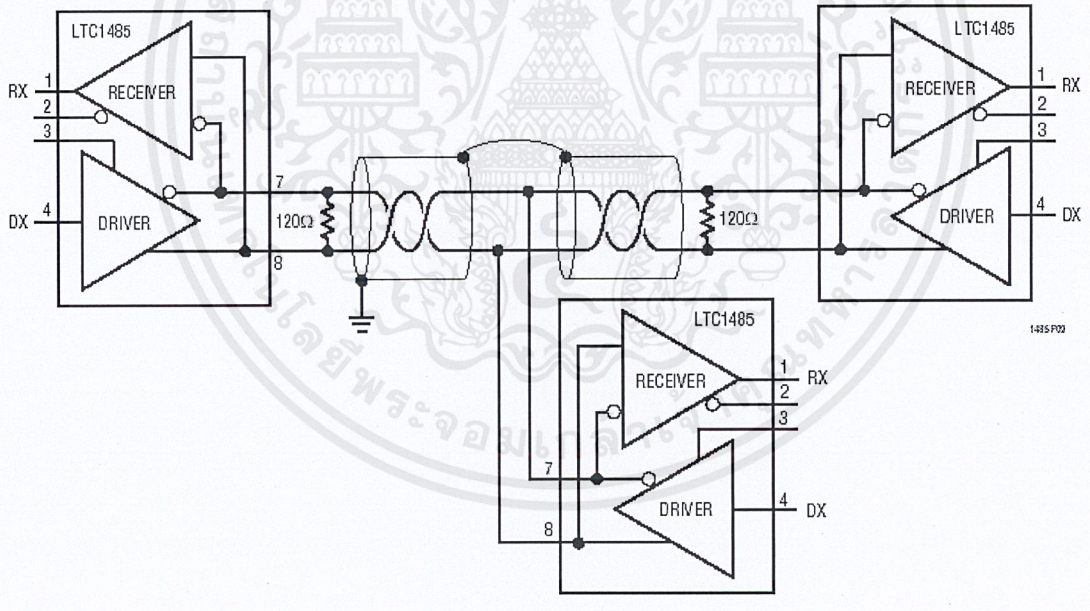
การออกแบบใช้ไมโครคอนโทรลเลอร์เป็นตัวประมวลผลกลางติดต่อสื่อสารกับอุปกรณ์ต่างๆดังนี้

DS1307 Real Time Clock เป็นตัวสร้างฐานเวลา วัน เดือน ปี ชั่วโมง นาที วินาที สามารถอ่านและเขียนข้อมูลใน ROM ของไอซีได้ ติดต่อสื่อสารแบบ I^2C

LCD ใช้แสดงผลเวลา

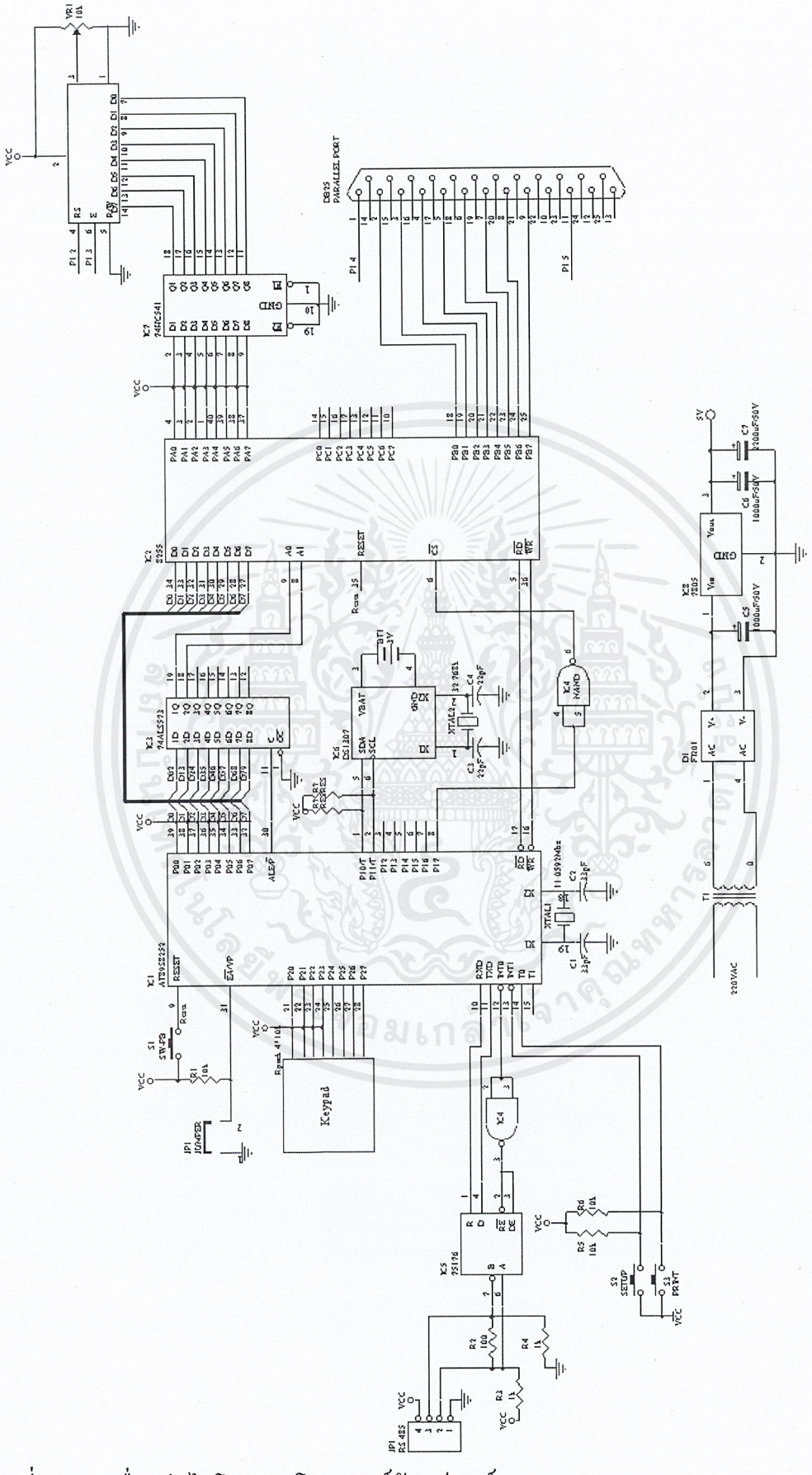
Printer แสดงผลบัตรคิว

SN75176BP ใช้ติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ตัวอื่น เป็นการติดต่อสื่อสารชนิด RS-485 แบบ Half Duplex



ภาพที่ 3.1 การส่งข้อมูลแบบ Half Duplex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 การเชื่อมต่อไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

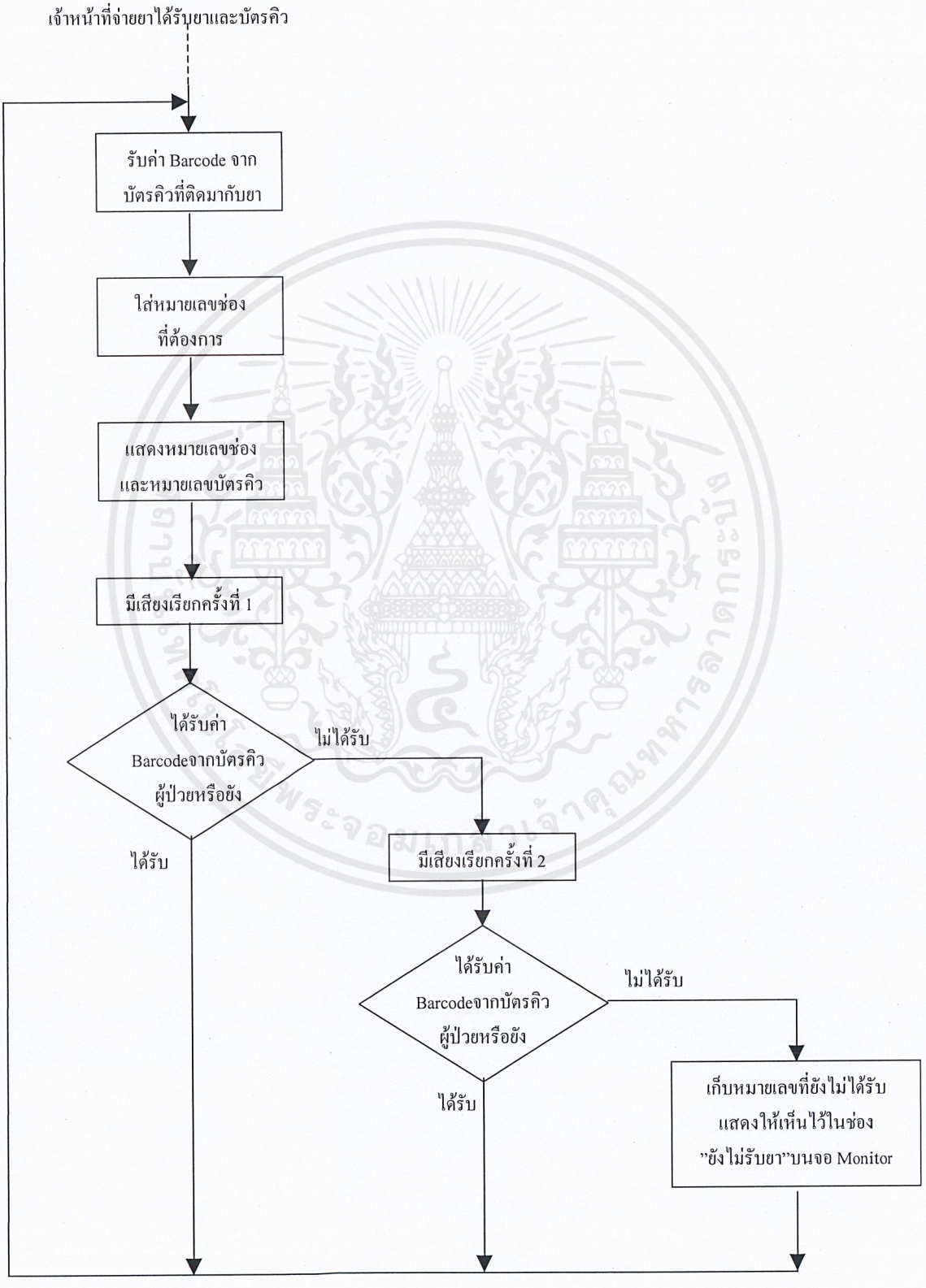
อุปกรณ์ส่วนฮาร์ดแวร์ประกอบด้วย

1. AT89S8252
2. KEYPAD 4*4
3. LCD 16*1
4. 75176
5. 74HC00
6. 74HC541
7. 74HC373
8. DS1307
9. SWITCH
10. PARALLEL PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของโปรแกรม

ในการทำงานของโปรแกรมที่ใช้ดังที่ได้กล่าวไปแล้วข้างต้นว่าเราจะใช้โปรแกรม Visual Basic ในการทำงานซึ่งจะมีลำดับขั้นตอนการทำงานดังนี้



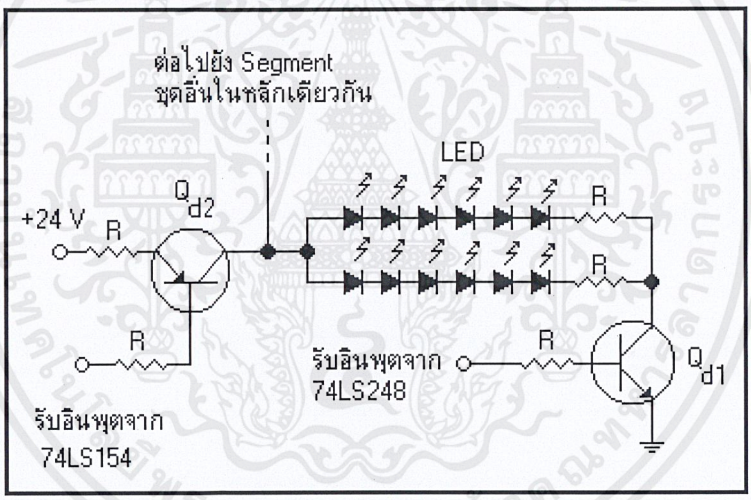
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรส่วนการแสดงผล ซึ่งจะแยกเป็น 2 ส่วน คือ

1 ส่วนการแสดงผลสำหรับช่องการเรียกผู้ป่วย มาชำระเงิน

2 ส่วนการแสดงผลสำหรับช่องการเรียกผู้ป่วยให้มารับยา

ซึ่งในส่วนการแสดงผลทั้งสองแบบนี้ จะมีวงจรของการประกอบกันของ LED ที่เหมือนกัน เราจะใช้ LED มาสร้างเป็นตัวเลขแสดงผลขนาดใหญ่ โดยจะนำมาเรียงเป็น 7-Segment จำนวน 3 หลักต่อ 1 บรรทัด มีอยู่ 1 ชุด และมีเป็น 7-Segment จำนวน 4 หลักต่อ 3 บรรทัด มี 1 ชุด รวมแล้วจะต้องสร้าง 7-Segment จำนวน 3 หลักขึ้น 1 ชุด และ 7-Segment จำนวน 12 หลักอีก 1 ชุด และเนื่องจากใช้ LED จำนวนมาก จึงต้องมีการใช้วงจรขับ (Drive) ให้กับวงจร DISPLAY ด้วย โดยจะใช้ทรานซิสเตอร์เป็นอุปกรณ์ในการขับให้แต่ละ Segment โดย อินพุตจะได้จากสัญญาณวงจร Decoder และจากวงจรเอาท์พุท ที่ได้จากการแปลงสัญญาณ I^2C ซึ่งจะได้อธิบายให้ทราบต่อไป วงจรในแต่ละ Segment มีวงจร ดังภาพที่ 3.3

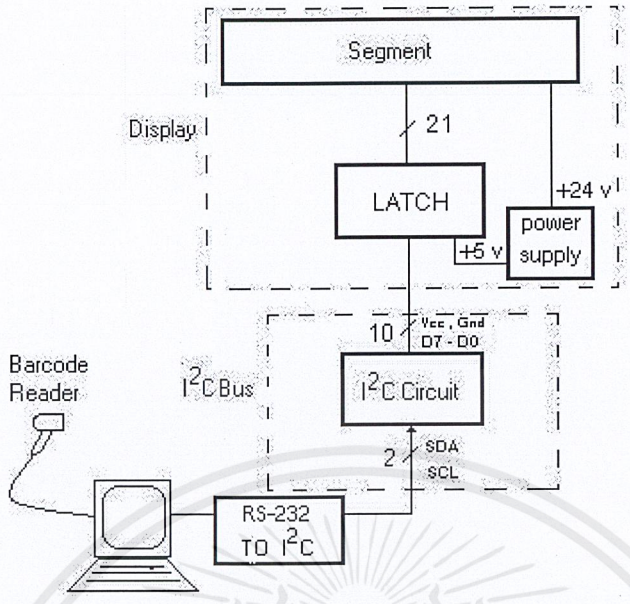


ภาพที่ 3.3 วงจรของแต่ละ Segment

3.2.1 ส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มาชำระเงิน

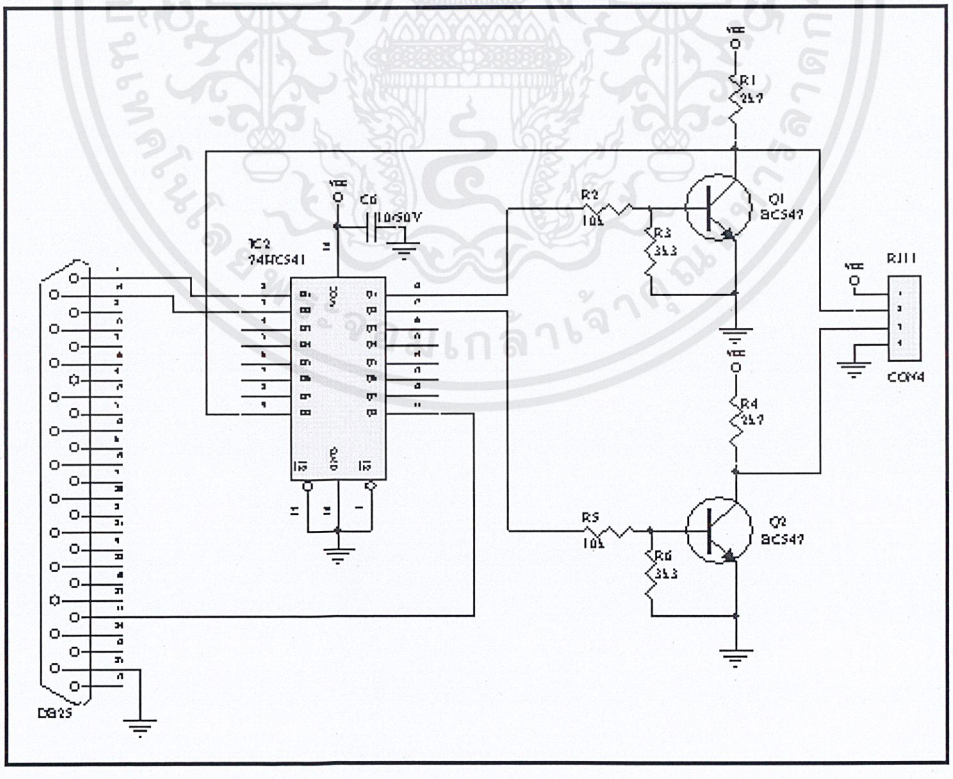
ส่วนนี้เป็นส่วนที่ จะใช้การติดต่อกับวงจรการแสดงผลโดยใช้ SERIAL PORT เนื่องจาก COMPUTER ที่ใช้ในการพิมพ์ใบเสร็จรับเงินนั้น จะต่อเข้ากับ PRALLEL PORT ดังนั้น จึงเหลือ SERIAL PORT เอาไว้ เราจึงใช้งานจาก PORT นี้ได้ โดยจะมีส่วนประกอบของวงจรต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4 Block diagram ของส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มาชำระเงิน

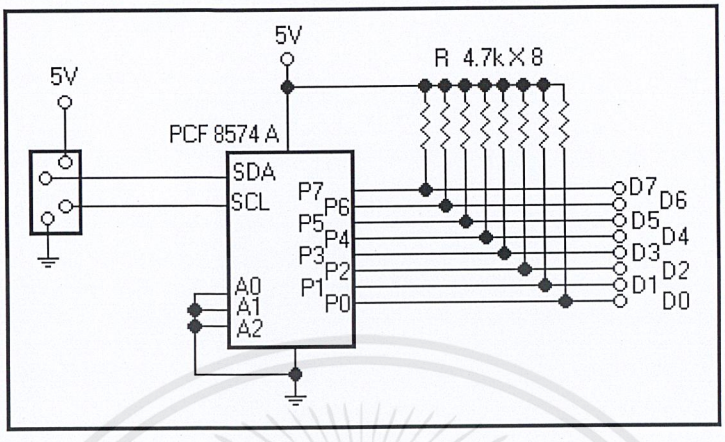
วงจรบอร์ดแปลงสัญญาณแบบ I^2C จาก SERIAL PORT (RS-232) มีวงจรตามรูป



ภาพที่ 3.5 บอร์ดแปลงสัญญาณแบบ I^2C จาก SERIAL PORT (RS-232)

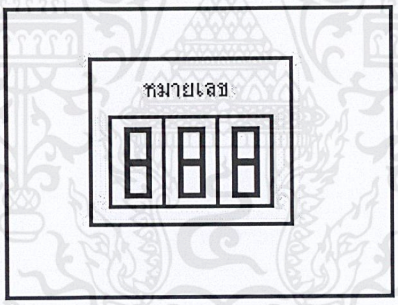
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดวงจรเอาท์พุทเป็นวงจรที่ใช้สำหรับการเปลี่ยนสัญญาณ I^2C ออกมาเป็นสัญญาณ TTL ขนาด 8 bit เพื่อนำส่งไปยังวงจรแสดงผลอีกทีหนึ่ง



ภาพที่ 3.6 บอร์ดวงจรเอาท์พุท

วงจรแสดงผลที่ใช้ในส่วนนี้จะเป็น 7-SEGMENT 3 หลัก วงจรเดียว ดังรูป

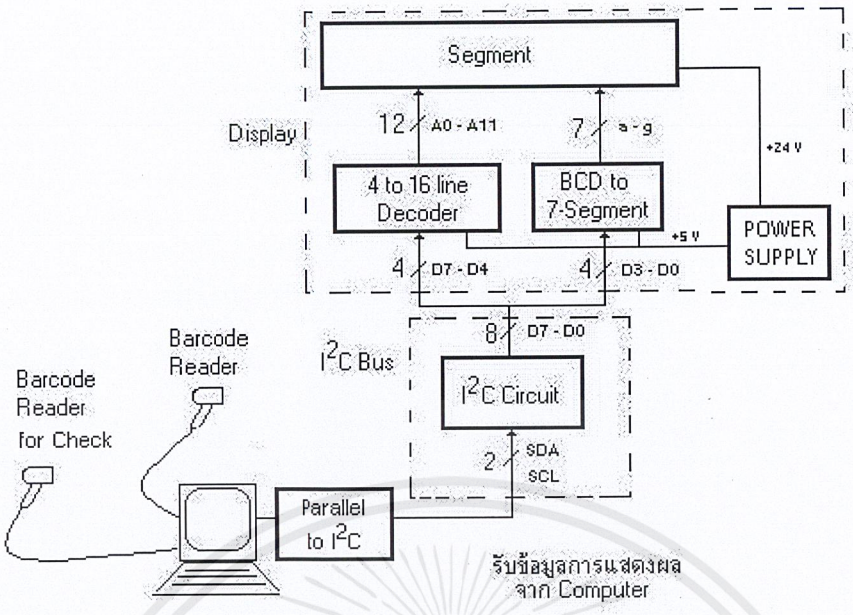


ภาพที่ 3.7 แสดงรูปแบบวงจรแสดงผล 3 หลัก วงจรเดียว

3.2.2 ส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มารับยา

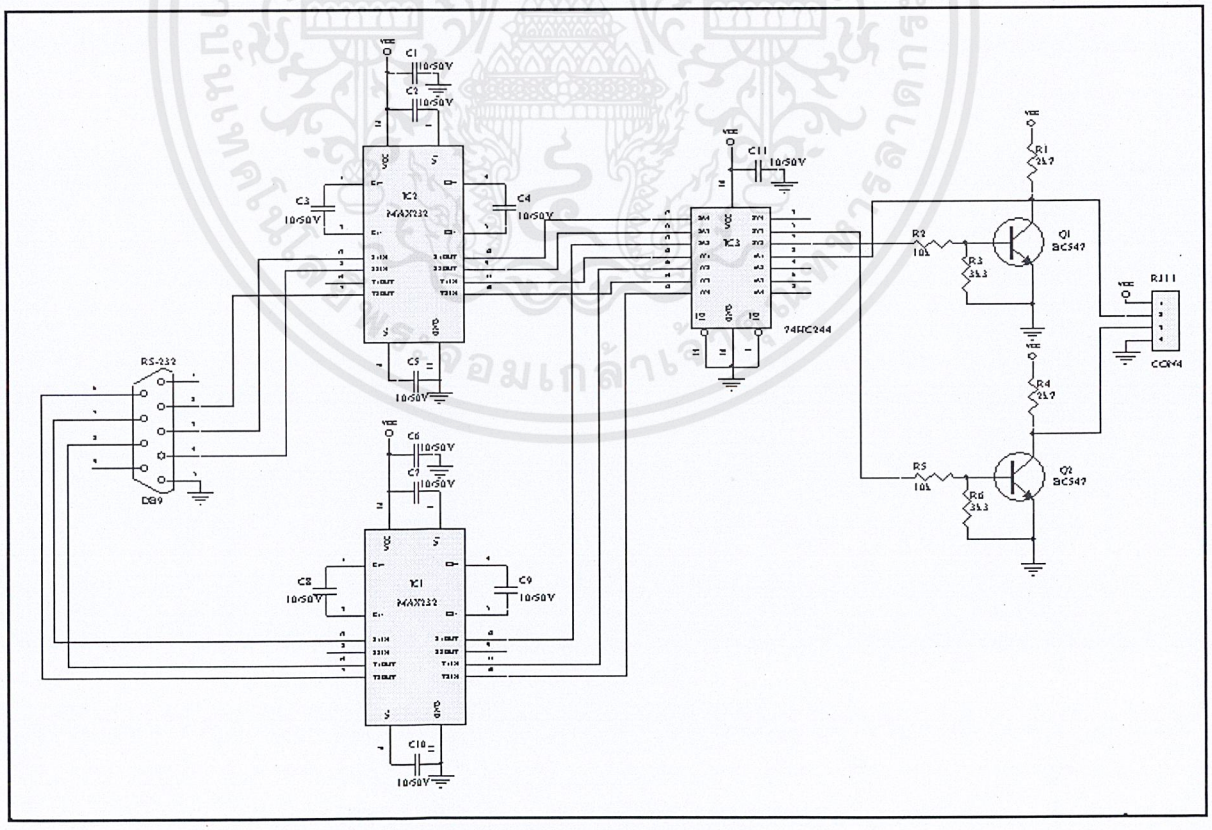
ส่วนนี้เป็นส่วนที่ จะใช้การติดต่อกับวงจรการแสดงผลโดยใช้ PARALLEL PORT เราใช้งานจาก PORT นี้ได้ โดยจะมีส่วนประกอบของวงจรต่างๆดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 Block diagram ของส่วนการแสดงผล สำหรับช่องการเรียกผู้ป่วย มารับยา

วงจรบอร์ดแปลงสัญญาณแบบ I^2C จาก PARALLEL PORT มีวงจรตามรูป

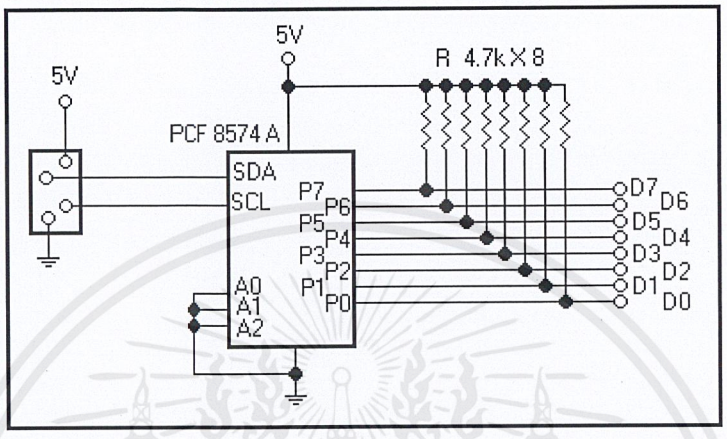


ภาพที่ 3.9 บอร์ดแปลงสัญญาณแบบ I^2C จาก PARALLEL PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดวงจรเอาท์พุท

เป็นวงจรที่ใช้สำหรับการเปลี่ยนสัญญาณ I^2C ออกมาเป็นสัญญาณ TTL ขนาด 8bit เพื่อนำส่งไปยังวงจรแสดงผลอีกทีหนึ่ง

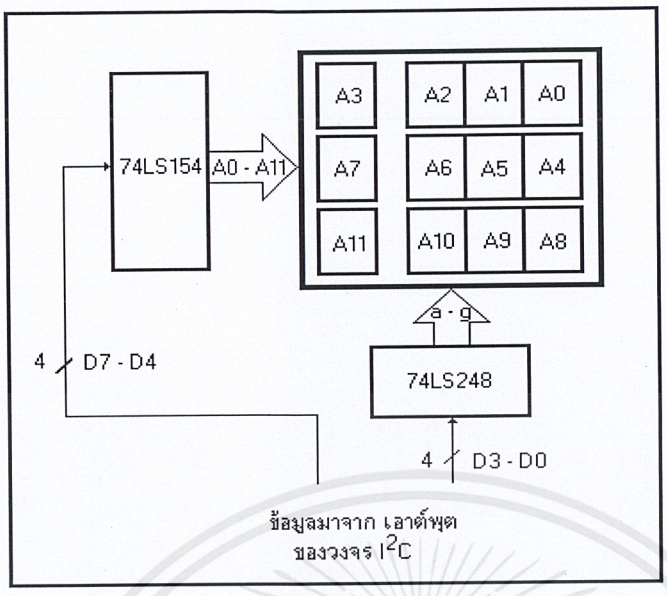


ภาพที่ 3.10 บอร์ดวงจรเอาท์พุท

วงจรแสดงผล

การควบคุมตัวเลขแสดงผล 7-Segment จำนวน 12 หลักโดยใช้ข้อมูลขนาด 8 bit ทำได้โดยวงจรถอดรหัส ซึ่งประกอบด้วย IC TTL 74LS154 (4 to 16 line decoder) และ IC TTL 74LS248 (BCD to 7-Segment) เราจะใช้ ข้อมูลที่ได้จาก วงจรเอาต์พุตของ I^2C ทาง 4 bit ต่ำ คือ D_3-D_0 เป็นอินพุตให้กับ 74LS248 ในการแปลงข้อมูลออกมาเป็นตัวเลข และใช้ด้าน 4 bit สูง คือ D_7-D_4 เป็นอินพุตให้กับ 74LS154 ในการแปลงข้อมูลในการเลือกว่าจะให้ติดที่หลักใดซึ่งเราใช้ 12 หลักวงจรจะทำการ Scan หลักต่างๆให้สอดคล้องกันกับการส่งตัวเลขออกไปซึ่งจะทำให้เราประหยัด พลังงานในการจ่ายให้กับตัว LED ดังวงจรแสดงในภาพที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.11 วงจรถอดรหัส

รูปแบบการแสดงผล

สำหรับวงจร Display ได้มีการปรับเปลี่ยนจากการแยกการแสดงผลของแต่ละช่องกันมาเป็นแบบรวมกันอยู่ในบอร์ดใหญ่เพียงบอร์ดเดียวเนื่องจากพบข้อเสียหลายประการดังนี้

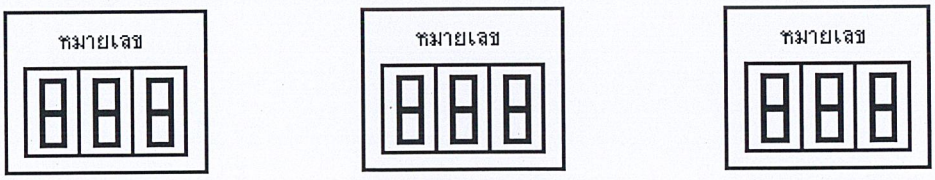
1.ติดตั้งยาก ต้องทำหลายจุด

2.ต้องใช้แหล่งจ่ายให้กับแต่ละบอร์ดซึ่งต้องทำจำนวนชุดเท่ากับจำนวนบอร์ดแสดงผล

3.หากต้องการขยายเพิ่มให้ใช้ได้กับหลายช่องมากขึ้น ก็จะต้องทำบอร์ดเพิ่มอีก

4.การเคลื่อนย้ายทำได้ด้วยความยุ่งยาก

5.มุมมองการมองผู้ใช้บริการต้องมองหลายตำแหน่ง



ภาพที่ 3.12 บอร์ด Display ที่แยกกันตามช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใช้บอร์ดรวมจะสามารถแก้ไขปัญหาเหล่านี้ได้

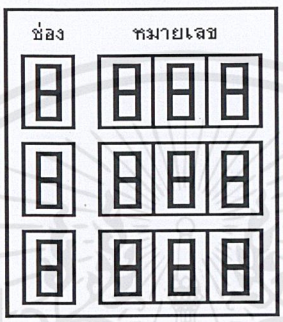
1.ติดตั้งง่ายกว่าเพราะทำเพียงจุดเดียว

2.ใช้แหล่งจ่ายเพียงชุดเดียว

3.หากต้องการขยายช่องการทำงานก็เพียงการตั้งค่าใหม่ให้กับโปรแกรมเท่านั้น

4.เคลื่อนย้ายง่ายกว่า

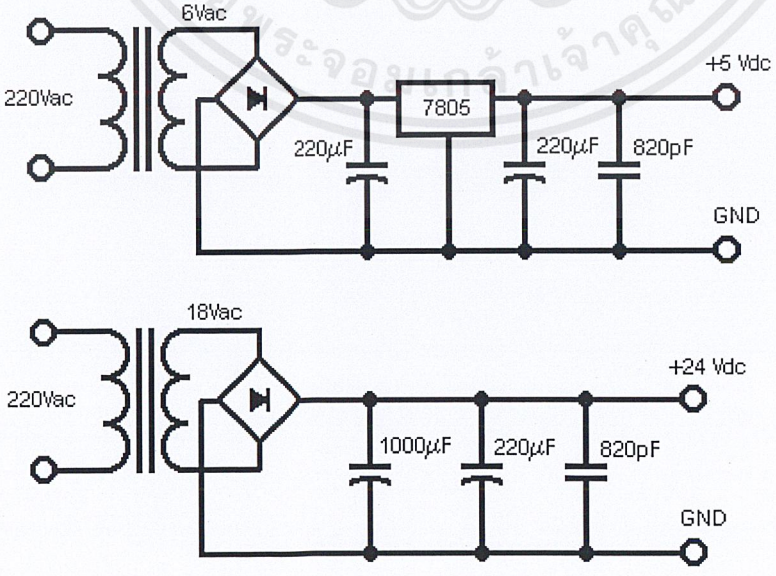
5.มีศูนย์กลางการมองเพียงจุดเดียว



ภาพที่ 3.13 บอร์ด Display ที่มีการแสดงผลรวมกัน

วงจร POWER SUPPLY

เป็นแหล่งจ่ายไฟเลี้ยงในวงจรแปลงสัญญาณ ซึ่งจะใช้แรงดัน 5 โวลต์ และอีกส่วนหนึ่งเป็นแหล่งจ่ายไฟเลี้ยงให้กับ แผงวงจร 7-SEGMENT ใช้แรงดัน 24 โวลต์



ภาพที่ 3.14 วงจร POWER SUPPLY

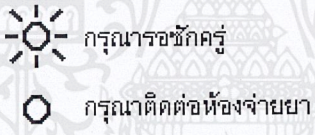
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนการใช้เสียงเรียก

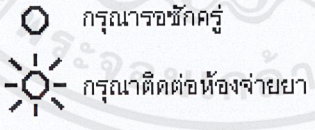
ในส่วนการเรียกหมายเลขเราจะใช้การอัดเสียงเข้าไปเป็นคำสั้นๆที่เราจำเป็นจะต้องใช้ เช่นคำว่า “หมายเลข” , “ติดต่อ” , “ช่องยา” , “ครับ” , “หนึ่ง” , ”สาม” , ”สิบ” , “ยี่” , “ร้อย” เป็นต้น หากมีการเรียกตัวเลขเราก็จะรวมคำกันเช่น 853 จะได้การเรียกว่า “ แปด ร้อย ห้า สิบ สาม “ โดยการเรียกในส่วนที่เป็นช่องการจ่ายยาจะมีเสียงพูดว่า ” หมายเลข XXX มารับยาที่ช่องรับยา X ครับ(ค่ะ) ” ถ้าเป็นการเรียกในส่วนที่เป็นช่องให้ผู้ป่วยมาชำระเงินจะมีเสียงพูดว่า “ หมายเลข XXX ติดต่อที่ช่องการเงินด้วยครับ(ค่ะ) ”

3.4 ส่วนการตรวจสอบหมายเลขที่ถูกเรียกโดยตัวผู้ใช้บริการเอง

ที่บริเวณด้านนอกของห้องจัดยาจะมีวงจรในส่วนการตรวจสอบการเรียกโดย ผู้ใช้บริการจะนำบัตรคิวที่มี Barcode มาที่เครื่องอ่าน Barcode ที่ได้จัดเตรียมไว้แล้ว ให้เครื่องอ่านค่าบัตรคิวที่ได้แล้วจะมีไฟแสดงผลการตรวจสอบให้เห็น ถ้าหมายเลขที่นำมาตรวจสอบนี้ถูกเรียกไปแล้วจะมีหลอดไฟสีเขียวขึ้นที่ กรุณาติดต่อห้องจ่ายยา แต่ถ้าหมายเลขที่นำมาตรวจสอบนี้ยังไม่ถูกเรียกจะมีหลอดไฟสีแดงขึ้นที่ กรุณารอซักครู่



ภาพที่ 3.15 การแสดงผล ผลการตรวจสอบว่า หมายเลขที่นำมาตรวจสอบนี้ยังไม่ถูกเรียก



ภาพที่ 3.16 การแสดงผล ผลการตรวจสอบว่า หมายเลขที่นำมาตรวจสอบนี้ถูกเรียกแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การใช้งาน

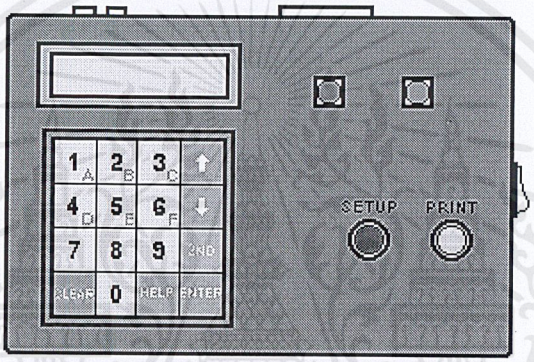
การใช้งานมี 2 ส่วน คือ

1.ส่วนการพิมพ์บัตรคิว

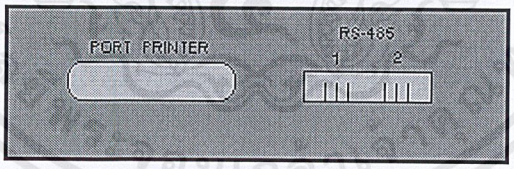
2.ส่วนของโปรแกรมการเรียกและการแสดงผล

4.1 ส่วนการพิมพ์บัตรคิว

วิธีใช้ติดตั้งเครื่องพิมพ์ BARCODE



ภาพที่ 4.1 แสดงลักษณะด้านหน้าของเครื่องพิมพ์บัตรคิว



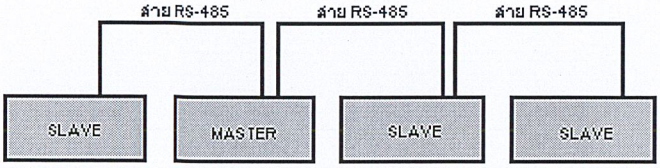
ภาพที่ 4.2 แสดงลักษณะด้านบนของเครื่องพิมพ์บัตรคิว

วิธีใช้ติดตั้งเครื่องพิมพ์ BARCODE

1. ต่อสายเครื่องพิมพ์เข้าทางด้านหลังเครื่องพิมพ์ BARCODE ทาง PORT PRINTER

2.ต่อสาย RS-485 เข้าทางด้านหลังเครื่องพิมพ์ BARCODE (เมื่อต้องการใช้เครื่องพิมพ์หลายเครื่อง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.3 การเชื่อมต่อของเครื่องพิมพ์บัตรคิว

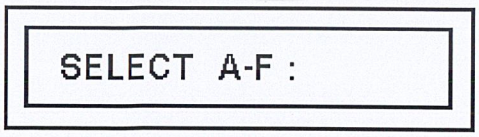
วิธีใช้งานเครื่องพิมพ์ Barcode

การพิมพ์บัตรคิว ให้กดปุ่มสีเหลืองทางด้านขวามือก็จะได้บัตรคิวออกมา ในบัตรคิวจะประกอบด้วย ชื่อสถานประกอบการ,วัน/เดือน/ปี,เวลา,หมายเลขบัตรคิว,Barcode

ภาพที่ 4.4 ลักษณะของบัตรคิวที่ได้

การ SETUP เครื่องพิมพ์ BARCODE

กดปุ่ม SETUP หน้าจอ LCD จะแสดง SELECT A-F :



ภาพที่ 4.5 แสดงหน้าจอที่แสดงเมื่อมีการ SET UP เครื่องพิมพ์

A (ลูกศรขึ้นและซ้าย) = การตั้งวันที่

B (ลูกศรลงและขวา) = การตั้งเวลา

C (2ND) = การดูค่าหมายเลขที่พิมพ์แล้วครั้งล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D (CLEAR) = การตั้งค่าหมายเลขเครื่อง

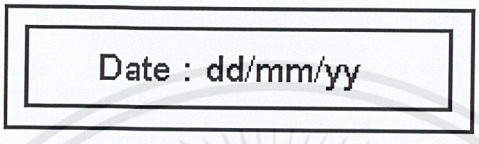
E (HELP) = การตั้งค่าจำนวนกระดาษที่จะพิมพ์

F (ENTER) = การตั้งค่าหมายเลขบัตรคิว

ไฟสีแดงคือไฟ POWER

ไฟสีเขียวคือ MASTER ถ้าไม่ติดจะเป็น SLAVE

การตั้งวันที่



ภาพที่ 4.6 แสดงหน้าจอที่แสดงเมื่อมีการตั้งวันที่

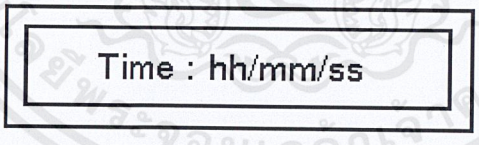
กดปุ่ม SETUP แล้วกดปุ่มลูกศรขึ้นและซ้าย หน้าจอ LCD จะแสดง Date : dd/mm/yy

dd หมายถึง วันที่ ให้ใส่เลข 01-31

mm หมายถึง เดือน ให้ใส่เลข 01-12

yy หมายถึง ปี ให้ใส่เลข 00-99

การตั้งเวลา



ภาพที่ 4.7 แสดงหน้าจอที่แสดงเมื่อมีการตั้งเวลา

กดปุ่ม SETUP แล้วกดปุ่มลูกศรลงและขวา หน้าจอ LCD จะแสดง Time : hh/mm/ss

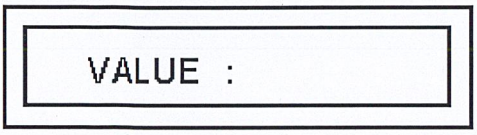
hh หมายถึง ชั่วโมงให้ใส่เลข 01-31

mm หมายถึง นาที ให้ใส่เลข 01-12

ss หมายถึง วินาที ให้ใส่เลข 00-99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

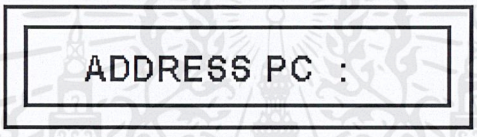
การดูค่าหมายเลขที่พิมพ์แล้วครั้งล่าสุด



ภาพที่ 4.8 แสดงหน้าจอที่แสดงเมื่อมีต้องการดูค่าหมายเลขที่พิมพ์ล่าสุด

กดปุ่ม SETUP แล้วกดปุ่ม 2ND หน้าจอ LCD จะแสดง VALUE : เมื่อกดแล้วจะแสดงตัวเลขที่พิมพ์ครั้งล่าสุดประมาณ 2 วินาที

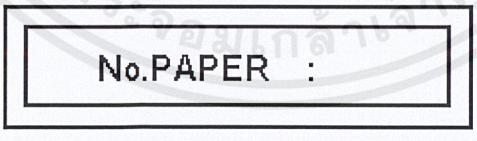
การตั้งหมายเลขเครื่อง



ภาพที่ 4.9 แสดงหน้าจอที่แสดงเมื่อมีต้องการตั้งหมายเลขเครื่อง

กดปุ่ม SETUP แล้วกดปุ่ม CLEAR หน้าจอ LCD จะแสดง ADDRESS PC : สามารถใส่หมายเลขเครื่องตั้งแต่ 01-FF ห้ามซ้ำกัน

การตั้งค่าจำนวนกระดาษที่จะพิมพ์

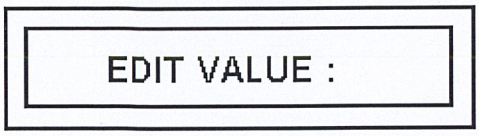


ภาพที่ 4.10 แสดงหน้าจอที่แสดงเมื่อมีต้องการตั้งค่าจำนวนกระดาษที่จะพิมพ์

กดปุ่ม SETUP แล้วกดปุ่ม HELP หน้าจอ LCD จะแสดง No.PAPER : สามารถใส่เลขตั้งแต่ 1-9 เวลากด PRINT ก็จะพิมพ์กระดาษออกมาตามจำนวนที่ตั้งไว้ ปกติจะอยู่ที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตั้งค่าหมายเลขบัตรคิว



ภาพที่ 4.11 แสดงหน้าจอที่แสดงเมื่อมีต้องการตั้งค่าจำนวนกระดาษที่จะพิมพ์

กดปุ่ม SETUP แล้วกดปุ่ม ENTER หน้าจอ LCD จะแสดง EDIT VALUE : สามารถใส่เลขตั้งแต่ 000-999

4.2 การใช้งานในส่วนของโปรแกรมการเรียกและการแสดงผล แบ่งออกเป็น 2 ส่วน

1.ส่วนการเรียกที่ช่องการเงินจะใช้ โปรแกรม Pay_Channel

2.ส่วนการเรียกที่ช่องจ่ายยา จะใช้ โปรแกรม Medic_Channel

4.2.1 การใช้งานโปรแกรม Pay_Channel

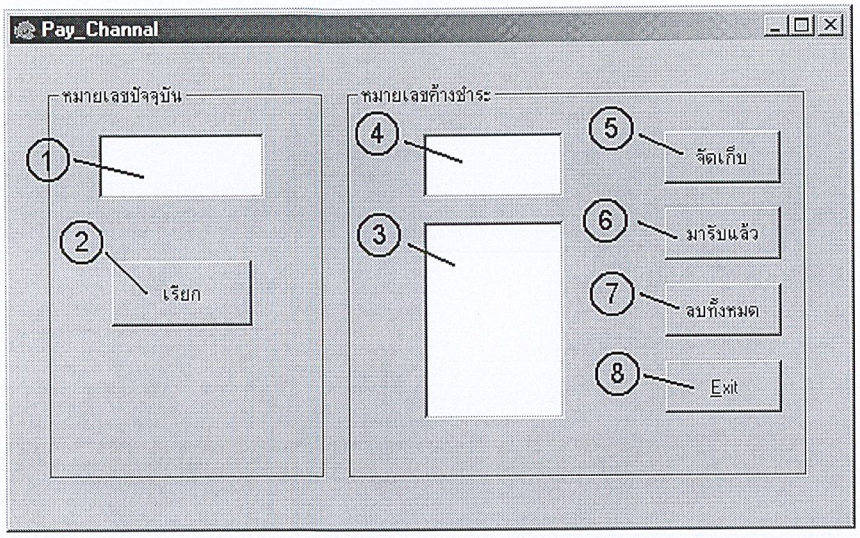
การติดต่อกับอุปกรณ์แสดงผลและเครื่องอ่านบาร์โค้ดสามารถต่อได้โดย

1. ต่อเครื่องอ่านบาร์โค้ดเข้ากับพอร์ตอนุกรม 1 (Comm. Port 1)
2. ต่อวงจรแปลงสัญญาณ Serial และ I^2C เข้ากับพอร์ตอนุกรม 2 (Comm. Port 2)

หมายเหตุ

เนื่องจากการเข้าไปสังเกตตามสถานพยาบาลจริง ในการพิมพ์ใบเสร็จค่ายา การติดต่อระหว่างเครื่อง Computer กับเครื่องพิมพ์ จะใช้ Parallel Port หรือเป็น USB Port อยู่แล้ว เราจึงออกแบบระบบการเรียกในส่วนนี้ให้ใช้การติดต่อกับพอร์ตอนุกรม เพื่อเป็นการไม่ซ้ำกับ พอร์ตของเครื่องพิมพ์โปรแกรม Pay_Channel เป็นโปรแกรมที่ใช้ในการอ่านค่าที่ได้การอ่านบาร์โค้ดที่บัตรคิวแล้วนำไปแสดงผล ที่บอร์ด Display โปรแกรมจะทำการเรียกให้ผู้ใช้บริการให้มารับใบเสร็จและชำระค่ายา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.12 แสดงหน้าจอของโปรแกรม Pay_Channel

จากโปรแกรมจะมีส่วนประกอบและรายละเอียดต่างๆดังนี้

หมายเลข 1 เป็นช่องแสดงหมายเลขบัตรคิวจะแสดงหมายเลขของบัตรคิวที่เจ้าหน้าที่การเงินทำการเรียกจากการอ่านค่าของบัตรคิวโดยใช้เครื่องอ่าน Barcode ซึ่งถ้าระบบจะทำการเรียกและแสดงผลให้เห็นได้โดยทันที และถ้าหากต้องการเรียกซ้ำ ก็สามารถทำได้โดยการคลิกที่ปุ่ม "เรียก" ระบบ จะทำการเรียกหมายเลขดังกล่าวอีกครั้ง

หมายเลข 2 เป็นปุ่ม "เรียก" ซึ่งจะใช้เมื่อต้องการจะเรียกหมายเลขที่เรียกไปแล้วอีกครั้งหนึ่ง ซึ่งถ้าหากมีการกดปุ่มนี้ก่อนการใส่ค่าตัวเลข ระบบจะแสดงกล่องข้อความเตือนให้เห็นซึ่งจะได้กล่าวถึงในหัวข้อต่อไป

หมายเลข 3 เป็น List แสดงหมายเลขที่เราทำการจัดเก็บแล้วโดยสามารถนำหมายเลขบัตรคิวที่ถูกเรียกไปแล้วแต่ยังไม่มาติดต่อกับช่องการเงิน เจ้าหน้าที่การเงินจะนำหมายเลขบัตรคิวดังกล่าวมาเก็บไว้ในนี้ได้

หมายเลข 4 เป็นช่องแสดงหมายเลขบัตรคิวที่ตกค้างการติดต่อ ที่ถูกเลือกขึ้นมาจาก List

หมายเลข 5 เป็นปุ่ม "จัดเก็บ" จะใช้ในการจัดเก็บหมายเลขบัตรคิวจากช่องแสดงหมายเลขบัตรคิวที่มีการเรียกแล้ว แต่หมายเลขดังกล่าวยังไม่มาติดต่อ และต้องการเก็บหมายเลขดังกล่าวเพื่อใช้ในการตรวจสอบหรือรอหมายเลขดังล่าวมาติดต่อ

หมายเลข 6 เป็นปุ่ม "มารับแล้ว" ใช้ในการลบหมายเลขที่อยู่ใน List โดยเจ้าหน้าที่จะเลือกหมายเลขจาก List และหมายเลขนั้นจะปรากฏที่ช่องแสดงหมายเลขตกค้าง กดปุ่ม มารับแล้วหมายเลขที่อยู่ใน List จะหายไป จะใช้เมื่อหมายเลขที่ถูกจัดเก็บไว้มาทำการติดต่อกับเจ้าหน้าที่การเงินแล้ว

หมายเลข 7 ปุ่ม "ลบทั้งหมด" จะใช้เมื่อต้องการลบหมายเลขที่อยู่ใน List ทั้งหมด

หมายเลข 8 ปุ่ม Exit เป็นปุ่มที่ใช้เมื่อต้องการออกจากโปรแกรม Pay_Channel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

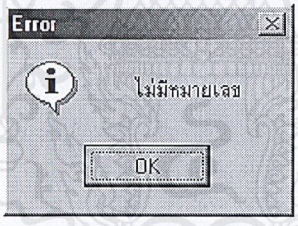
การแสดงข้อผิดพลาดต่างๆ

หากมีการผิดพลาดในการใช้งานจะมี ข้อความฟ้องขึ้นมาคอยเตือน ในลักษณะต่างๆ

1. ถ้ามีการกดปุ่ม เรียก หรือ จัดเก็บ โดยที่ไม่มีตัวเลขในช่องแสดงหมายเลขบัตรคิว จะมีข้อความเตือน การปฏิบัติ ให้ใส่หมายเลขบัตรคิวโดยการอ่านค่าจากเครื่องอ่านบัตรคิว

ภาพที่ 4.13 กล่องข้อความเตือนเมื่อมีการกดปุ่ม เรียก หรือ จัดเก็บ โดยไม่มีตัวเลขในช่องแสดงหมายเลขบัตรคิว

2. ถ้ามีการกดปุ่ม มารับแล้ว โดยไม่มีตัวเลขในช่องแสดงหมายเลขตกค้าง หรือยังไม่มีการเลือกหมายเลขจาก List การปฏิบัติ ให้เลือกหมายเลขที่อยู่ใน List



ภาพที่ 4.14 กล่องข้อความเตือนเมื่อมีการกดปุ่ม มารับแล้ว โดยไม่มีตัวเลขในช่องแสดงหมายเลขตกค้าง

3. ถ้ามีการกดปุ่ม ลบทั้งหมด จะมีข้อความบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

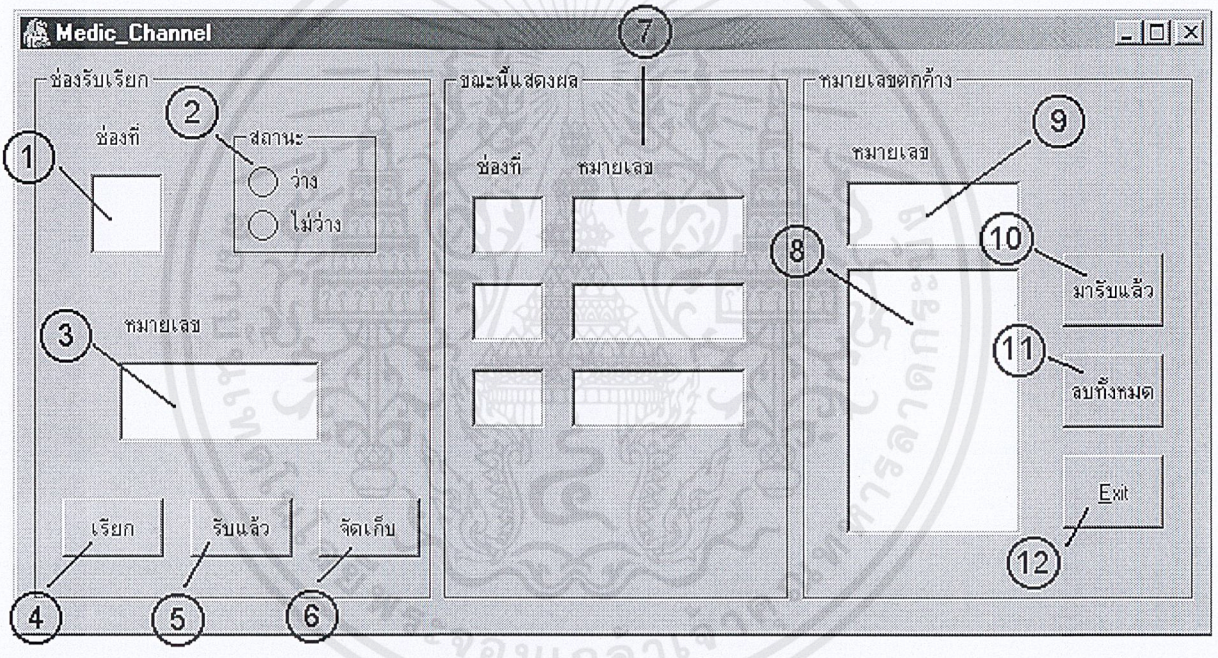
ภาพที่ 4.15 กล่องข้อความเมื่อมีการกดปุ่ม ลบทั้งหมด

4.2.2 การใช้งานโปรแกรม Medic_Channel

โปรแกรม Medic_Channel เป็นโปรแกรมที่ใช้ในการอ่านค่าที่ได้การอ่านบาร์โค้ดจากบัตรคิวแล้วนำไปแสดงผล ที่บอร์ด Display โปรแกรมจะทำการเรียกให้ผู้ใช้บริการให้มารับยาที่ช่องจ่ายยาที่ได้แสดงไว้ได้ถูกต้องตามที่ได้แสดงผลไว้บนบอร์ด Display

การติดต่อกับอุปกรณ์แสดงผลและเครื่องอ่านบาร์โค้ดสามารถต่อได้โดย

1. ต่อเครื่องอ่านบาร์โค้ด 1 เข้ากับพอร์ตอนุกรม 1 (Comm. Port 1)
2. ต่อเครื่องอ่านบาร์โค้ด 2 เข้ากับพอร์ตอนุกรม 2 (Comm. Port 2)
3. ต่อวงจรแปลงสัญญาณ Parallel และ I^2C เข้ากับพอร์ตขนาน (Parallel Port)



ภาพที่ 4.16 แสดงหน้าจอของโปรแกรม Medic_Channel

จากโปรแกรมจะมีส่วนประกอบและรายละเอียดต่างๆดังนี้

<u>หมายเลข 1</u> เป็นช่องแสดงหมายเลขช่องที่เจ้าหน้าที่จ่ายยาจะมาใส่เลขช่องจ่ายยาที่ตนเองรับผิดชอบว่าจะให้หมายเลขบัตรคิวที่ตนจะนำส่งนั้นสามารถติดต่อได้ที่ช่องรับยาใด

<u>หมายเลข 2</u> เป็นส่วนการแสดงสถานะของช่องจ่ายยาที่เจ้าหน้าที่จ่ายยาใส่ค่าลงไป ถ้าช่องที่เจ้าหน้าที่จ่ายยาใส่ลงไปนั้นยังไม่มีการใช้งาน จะมีสีเขียวปรากฏที่หน้าคำว่า “ว่าง” แต่ถ้าช่องที่เจ้าหน้าที่จ่ายยาใส่ลงไปนั้นยังมีการใช้งานอยู่ จะมีสีแดงปรากฏที่หน้าคำว่า “ไม่ว่าง” พร้อมทั้งปรากฏหมายเลขบัตรคิวที่กำลังติดต่ออยู่กับช่องจ่ายยานั้นๆอยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข 3 เป็นช่องแสดงหมายเลขของบัตรคิวที่อ่านได้จากเครื่องอ่าน Barcode

หมายเลข 4 เป็นปุ่ม “เรียก” จะใช้ในการเรียกหมายเลขบัตรคิวทีอ่านได้ โดยหมายเลขดังกล่าวจะถูกเรียกโดยเสียงและปรากฏที่ชุด Display หมายเลขที่เรียกจะปรากฏอยู่บนสุดของชุด Display และเมื่อมีหมายเลขใหม่เข้ามา หมายเลขเดิมจะเลื่อนลงมาในช่องถัดลงมา

หมายเลข 5 เป็นปุ่ม “รับแล้ว” เป็นปุ่มที่ใช้เมื่อหมายเลขที่ถูกเรียกไปนั้นมารับยาแล้ว โดยเจ้าหน้าที่ประจำช่องนั้นจะมาใส่เลขช่องของตน จะปรากฏหมายเลขของบัตรคิวที่ตนได้จ่ายยาไป สังเกตว่าปุ่มรับแล้วนี้จะพร้อมที่จะกดอยู่แล้วให้กด Enter ได้ทันที

หมายเลข 6 เป็นปุ่ม “จัดเก็บ” เป็นปุ่มที่ใช้เมื่อหมายเลขที่ถูกเรียกไปนั้นไม่มารับยา โดยจะเก็บหมายเลขดังกล่าวไว้ก่อนเพื่อว่าจะมีการมารับได้ในภายหลัง เจ้าหน้าที่จ่ายยาประจำช่องใส่เลขช่องของตนเองลงไป จะปรากฏหมายเลขของบัตรคิวที่ตนได้เรียกไป สังเกตว่าปุ่มรับแล้วจะพร้อมใช้อยู่ให้กดมาทางขวา แล้วจึงกด Enter หมายเลขดังกล่าวจะถูกเก็บไว้ใน List

หมายเลข 7 เป็นช่องที่ใช้ในการแสดงผลของการเรียกซึ่งจะตรงกับการแสดงผลของชุด Display โดยสามารถแสดงผลให้เห็นได้ครั้งละ 3 ช่อง โดยหมายเลขที่ถูกเรียกล่าสุดจะอยู่บนสุดเสมอ และเมื่อมีหมายเลขใหม่เข้ามา หมายเลขเดิมจะถูกเลื่อนให้ลงมาแสดงในช่องถัดลงมา

หมายเลข 8 เป็น List ที่ใช้เก็บหมายเลขที่ถูกเรียกไปแล้วแต่ยังไม่มีการมาติดต่อรับยาที่ช่องจ่ายยา เจ้าหน้าที่จ่ายยาจะนำหมายเลขดังกล่าวนั้นมาเก็บไว้

หมายเลข 9 เป็นช่องแสดงหมายเลขที่ถูกเลือกขึ้นมาจาก List

หมายเลข 10 เป็นปุ่ม “มารับแล้ว” เป็นปุ่มที่ใช้เมื่อหมายเลขที่ถูกจัดเก็บไว้นั้นกลับมารับยาในภายหลัง โดยจะลบหมายเลขที่ถูกเรียกขึ้นมาจาก List

หมายเลข 11 ปุ่ม “ลบทั้งหมด” จะใช้เมื่อต้องการลบหมายเลขที่อยู่ใน List ทั้งหมด

หมายเลข 12 ปุ่ม Exit เป็นปุ่มที่ใช้เมื่อต้องการออกจากโปรแกรม Medic_Channel

การแสดงข้อผิดพลาดต่างๆ

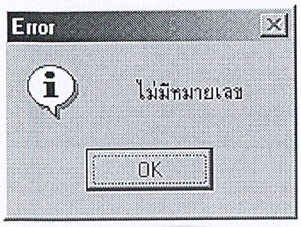
หากมีการผิดพลาดในการใช้งานจะมี ข้อความฟ้องขึ้นมาคอยเตือน ในลักษณะต่างๆ

1. ถ้ามีการกดปุ่ม เรียก หรือ รับแล้ว หรือ จัดเก็บโดยที่ไม่มีตัวเลขในช่องแสดงหมายเลขช่องจ่ายยา จะมีข้อความเตือน การปฏิบัติ ให้ใส่หมายเลขช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

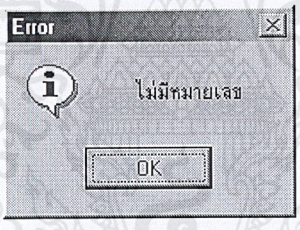
ภาพที่ 4.17 กล่องข้อความเตือนเมื่อมีการกดปุ่มเรียก หรือ รับแล้ว หรือ จัดเก็บ โดยไม่มีตัวเลขช่องแสดงหมายเลขช่อง

2. ถ้ามีการกดปุ่ม เรียก หรือ รับแล้ว หรือ จัดเก็บโดยที่ไม่มีตัวเลขในช่องแสดงหมายเลขบัตรคิว จะมีข้อความเตือน การปฏิบัติ ให้ใส่หมายเลขบัตรคิว โดยการอ่านค่าจากเครื่องอ่านบัตรคิว



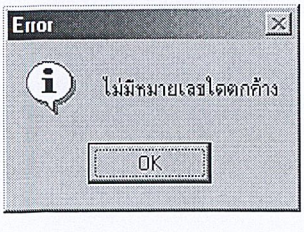
ภาพที่ 4.18 กล่องข้อความเตือนเมื่อมีการกดปุ่ม เรียก หรือ รับแล้ว หรือ จัดเก็บ โดยไม่มีตัวเลขในช่องแสดงหมายเลขบัตรคิว

3. ถ้ามีการกดปุ่ม มารับแล้ว โดยไม่มีตัวเลขในช่องแสดงหมายเลขตกค้างหรือยังไม่มีการเลือกหมายเลขจาก List การปฏิบัติ ให้เลือกหมายเลขที่อยู่ใน List



ภาพที่ 4.19 กล่องข้อความเตือนเมื่อมีการกดปุ่ม มารับแล้ว โดยไม่มีตัวเลขในช่องแสดงหมายเลขตกค้าง

4. ถ้ามีการกดปุ่ม ลบทั้งหมด จะมีข้อความบอก



ภาพที่ 4.20 กล่องข้อความเมื่อมีการกดปุ่ม ลบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์ในการศึกษาและแก้ไขปัญหาในระบบการรับจ่ายยาในโรงพยาบาล เพื่อลดความผิดพลาด และอำนวยความสะดวกแก่ผู้ใช้บริการ และเจ้าหน้าที่ประจำห้องจ่ายยา จากการวิเคราะห์ระบบการรับจ่ายยาในโรงพยาบาล พบปัญหาใน 3 ขั้นตอน ได้แก่ขั้นตอนการส่งใบสั่งยาแก้ไข โดยจัดให้มีเครื่องจ่ายบัตรคิว เพื่อแสดงลำดับก่อนหลัง และไม่ต้องให้ผู้ป่วยต้องเข้าแถวรอ อันเป็นการกีดขวางทางเดิน ในขั้นตอนการเรียกผู้ป่วยมารับยา และชำระเงินค่าบริการจัดให้มี ส่วนแสดงผล LED DISPLAY ร่วมกับ SOUND CALL เพื่อเสริมให้ การเรียกมีประสิทธิผลยิ่งขึ้น และในขั้นตอนการตรวจสอบการเรียกรับยา จัดให้มีเครื่องตรวจสอบการเรียกเพื่อให้ผู้ใช้บริการสามารถตรวจสอบการเรียกได้ด้วยตนเอง

จากการทดลองเครื่องจ่ายบัตรคิวจะต้องพิมพ์รหัสแถบ (BARCODE) 13 หลักซึ่งจะมีข้อมูลวันที่/เดือน/ปี,เวลา ชั่วโมง/นาที และหมายเลขบัตรคิวทั้ง 3 หลัก จากนั้นเจ้าหน้าที่ประจำห้องจ่ายยาจะทำการบันทึกข้อมูลการสั่งยาตามใบสั่งแพทย์ เพื่อจัดทำเป็นใบเสร็จรับเงินและฉลากหน้าซองยาในแต่ละรายการ ซึ่งจะมีรหัสแถบเช่นเดียวกับบัตรคิว และจะใช้เครื่องอ่าน BARCODE เป็นตัวอ่านข้อมูลหมายเลขจากบัตรคิว 3 หลักจากรหัสแถบ เพื่อเรียกให้ผู้ใช้บริการมาชำระเงินค่าบริการ หรือรับยาตามช่องทำการที่ส่วนแสดงผลได้แสดงไว้ให้เห็นแล้ว อีกทั้งผู้ใช้บริการยังสามารถตรวจสอบการเรียกได้ด้วยตนเองจากเครื่องอ่าน BARCODE ที่ได้จัดเตรียมไว้ให้ ซึ่งจะเป็นการลดความผิดพลาดและอำนวยความสะดวกแก่ผู้ใช้บรการ และเจ้าหน้าที่ประจำห้องจ่ายยา

ปัญหาที่พบในขั้นตอนการทดลองนั้นได้แก่ความผิดพลาดในการอ่านข้อมูลจากบัตรคิว 3 หลักจากรหัสแถบเพื่อเรียกผู้ใช้บริการด้วยเครื่องอ่าน BARCODE ซึ่งมีสาเหตุจากการพิมพ์ ที่ไม่ชัดเจนเนื่องมาจากหมึกจาง ทำให้รหัสแถบที่พิมพ์ออกมาเลือนลางไป และก็เป็นผลทำให้การอ่านค่าที่ได้คลาดเคลื่อนไป

ข้อเสนอแนะสำหรับการขยายขนาดของระบบ หรือการเพิ่มช่องบริการนั้นสามารถแยกพิจารณาได้ 2 ส่วน ได้แก่ ส่วน SOFTWARE ต้องมีการแก้ไขโปรแกรมคำสั่งให้เหมาะสมกับจำนวนช่องบริการ ส่วน HARDWARE ในด้านของ LED DISPLAY สามารถต่อขยายให้มีหลายชุดมากขึ้นได้ และยังสามารถใช้ได้กับระยะทางที่ไกลขึ้นอีก เนื่องจากใช้ระบบ I^2C ซึ่งใช้สายส่งสัญญาณเพียง 2 เส้นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ชัยวัฒน์ ลิ้มพรจิตรวิไล,วรพจน์ กรแก้ววัฒนกุล,"เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 (MSC-51 Microcontroller Theory & Practical Approach)" บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด , 476 หน้า

2. อุดม จีนประดับ , "ไมโครคอนโทรลเลอร์ MCS-51" , ศูนย์ผลิตตำรา สถาบันเทคโนโลยี-พระจอมเกล้าพระนครเหนือ , 260 หน้า , กุมภาพันธ์ 2541

3. รองศาสตราจารย์ สมยศ จุณณะปิยะ ," การประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ Microcontroller Applications MCS-51" , ศูนย์ผลิตตำรา สถาบันเทคโนโลยีพระจอมเกล้า-พระนครเหนือ , 358 หน้า , 2543

4. กฤษฎา ใจเย็น , อรรถพล บุญยะโภคา , ชัยวัฒน์ ลิ้มพรจิตรวิไล ," เรียนรู้และปฏิบัติการการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม",บริษัทอินโนเวทีฟเอ็กเพอริเมนต์ จำกัด , 163 หน้า

5. กฤษฎา ใจเย็น , ชัยวัฒน์ ลิ้มพรจิตรวิไล ," เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ต", บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด , 122 หน้า

6. ฉันทวุฒิ พืชผล , พิชิต สันติกุลานนท์ ," คู่มือเรียน Visual Basic 6 " ,บริษัทโปรวิชั่นจำกัด , 476 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

แสดง DATA SHEET ของ IC หลักๆ ได้แก่ AT89S8252, DS1307, PCF8574 และ ระบบ บัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

PCF8574
Remote 8-bit I/O expander for I²C-bus

Product specification
Supersedes data of September 1994
File under Integrated Circuits, IC12

1997 Apr 02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

Philips Semiconductors

PHILIPS

CONTENTS

1 FEATURES
2 GENERAL DESCRIPTION
3 ORDERING INFORMATION
4 BLOCK DIAGRAM
5 PINNING
6 CHARACTERISTICS OF THE I²C-BUS
6.1 Bit transfer
6.2 Start and stop conditions
6.3 System configuration
6.4 Acknowledge
7 FUNCTIONAL DESCRIPTION
7.1 Addressing
7.2 Interrupt
7.3 Quasi-bidirectional I/Os
8 LIMITING VALUES
9 HANDLING
10 DC CHARACTERISTICS
11 I²C-BUS TIMING CHARACTERISTICS
12 PACKAGE OUTLINES
13 SOLDERING
13.1 Introduction
13.2 DIP
13.2.1 Soldering by dipping or by wave
13.2.2 Repairing soldered joints
13.3 SO and SSOP
13.3.1 Reflow soldering
13.3.2 Wave soldering
13.3.3 Repairing soldered joints
14 DEFINITIONS
15 LIFE SUPPORT APPLICATIONS
16 PURCHASE OF PHILIPS I²C COMPONENTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus PCF8574

1 FEATURES

- Operating supply voltage 2.5 to 6 V
- Low standby current consumption of 10 • A maximum
- I²C to parallel port expander
- Open-drain interrupt output
- 8-bit remote I/O port for the I²C-bus
- Compatible with most microcontrollers
- Latched outputs with high current drive capability for directly driving LEDs
- Address by 3 hardware address pins for use of up to 8 devices (up to 16 with PCF8574A)
- DIP16, or space-saving SO16 or SSOP20 packages.

2 GENERAL DESCRIPTION

The PCF8574 is a silicon CMOS circuit. It provides general purpose remote I/O expansion for most microcontroller families via the two-line bidirectional bus (I²C).

The device consists of an 8-bit quasi-bidirectional port and an I²C-bus interface. The PCF8574 has a low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. It also possesses an interrupt line ($\overline{INT}$) which can be connected to the interrupt logic of the microcontroller. By sending an interrupt signal on this line, the remote I/O can inform the microcontroller if there is incoming data on its ports without having to communicate via the I²C-bus. This means that the PCF8574 can remain a simple slave device.

The PCF8574 and PCF8574A versions differ only in their slave address as shown in Fig.9.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF8574P; PCF8574AP	DIP16	plastic dual in-line package; 16 leads (300 mil)	SOT38-1
PCF8574T; PCF8574AT	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1
PCF8574TS	SSOP20	plastic shrink small outline package; 20 leads; body width 4.4 mm	SOT266-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 BLOCK DIAGRAM

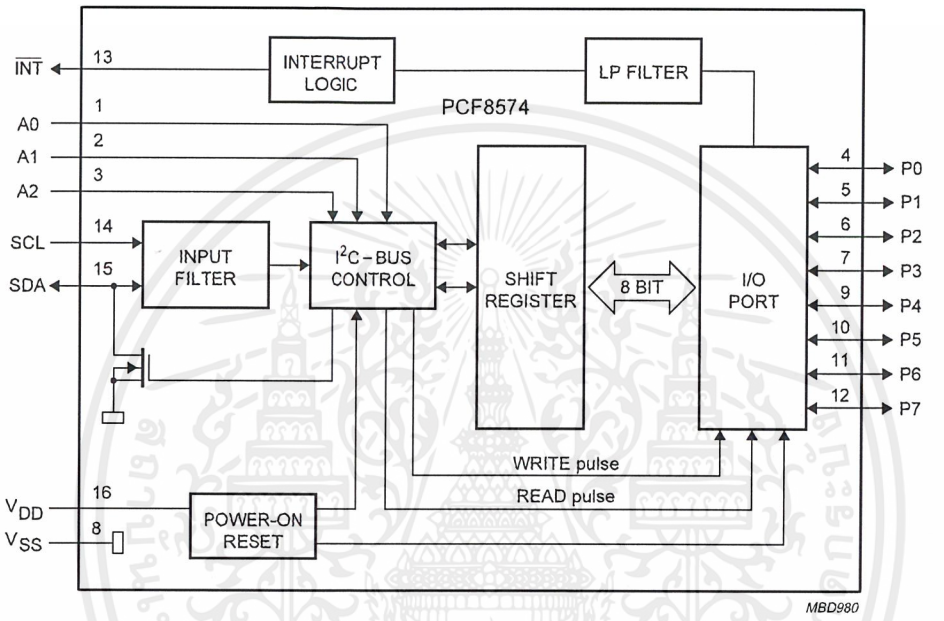


Fig.1 Block diagram (SOT38-1 and SOT162-1).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 PINNING

SYMBOL	PIN		DESCRIPTION
	DIP16; SO16	SSOP20	
A0	1	6	address input 0
A1	2	7	address input 1
A2	3	9	address input 2
P0	4	10	quasi-bidirectional I/O 0
P1	5	11	quasi-bidirectional I/O 1
P2	6	12	quasi-bidirectional I/O 2
P3	7	14	quasi-bidirectional I/O 3
V_{SS}	8	15	supply ground
P4	9	16	quasi-bidirectional I/O 4
P5	10	17	quasi-bidirectional I/O 5
P6	11	19	quasi-bidirectional I/O 6
P7	12	20	quasi-bidirectional I/O 7
$\overline{INT}$	13	1	interrupt output (active LOW)
SCL	14	2	serial clock line
SDA	15	4	serial data line
V_{DD}	16	5	supply voltage
n.c.	•	3	not connected
n.c.	•	8	not connected
n.c.	•	13	not connected
n.c.	•	18	not connected

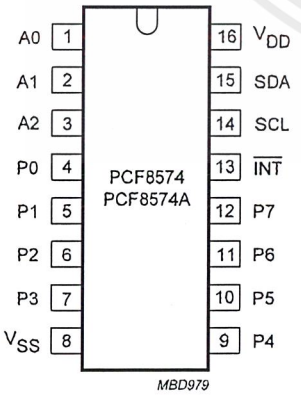


Fig.2 Pin configuration (DIP16; SO16).

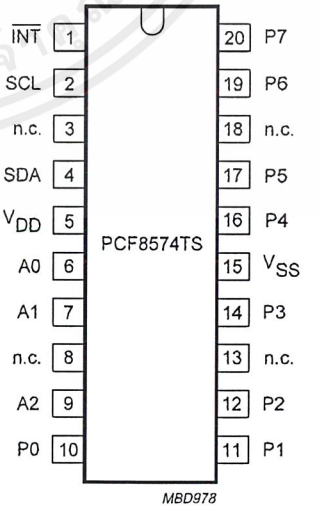


Fig.3 Pin configuration (SSOP20).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

6.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Fig.4).

6.2 Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S).
A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P) (see Fig.5).

6.3 System configuration

A device generating a message is a 'transmitter', a device receiving is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves' (see Fig.6).

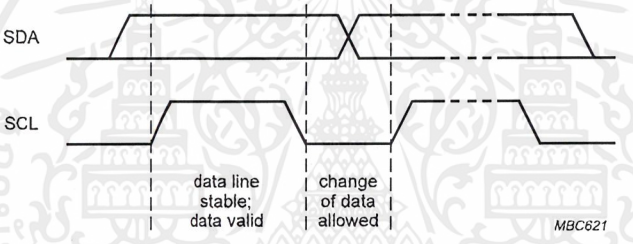


Fig.4 Bit transfer.

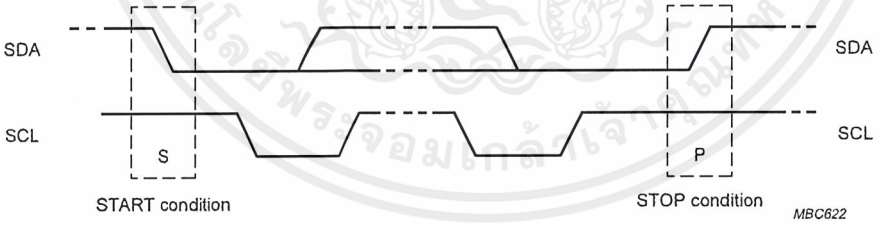


Fig.5 Definition of start and stop conditions.

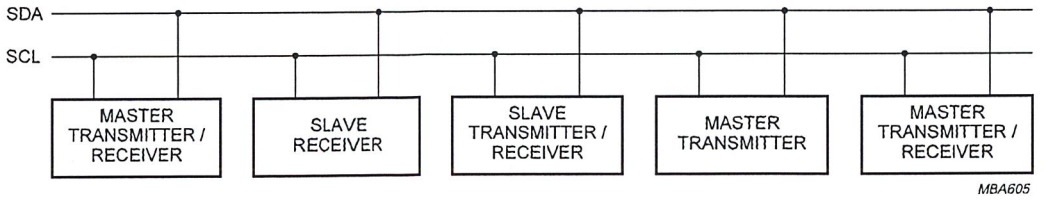


Fig.6 System configuration.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 Acknowledge

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set-up and hold times must be taken into account.

A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

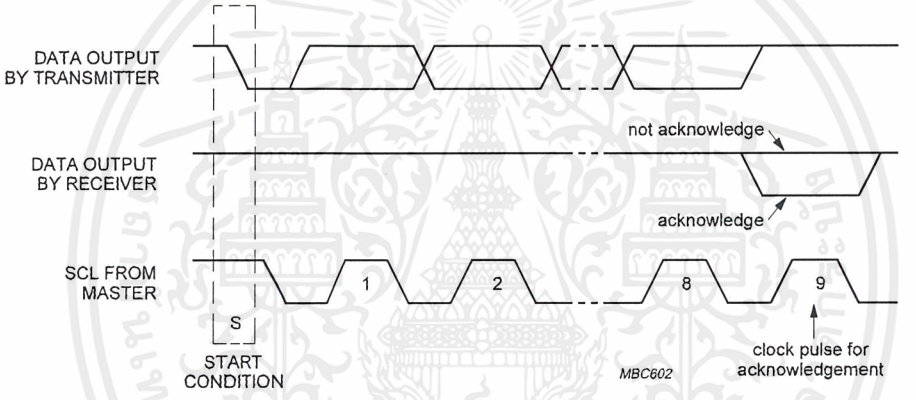


Fig.7 Acknowledgment on the I²C-bus.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 FUNCTIONAL DESCRIPTION

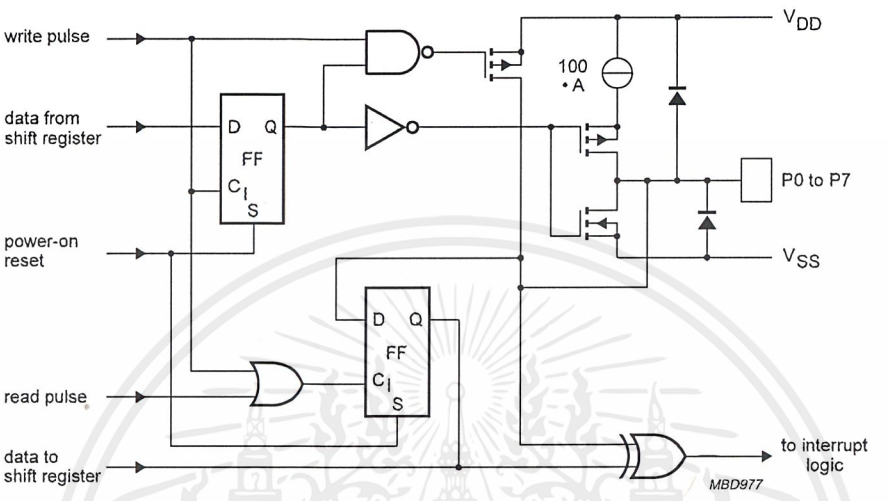
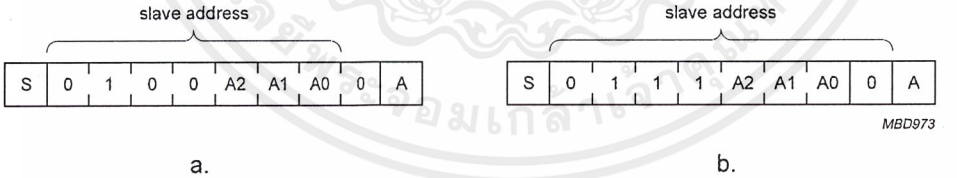


Fig.8 Simplified schematic diagram of each I/O.

7.1 Addressing

For addressing see Figs 9, 10 and 11.



(a) PCF8574.
(b) PCF8574A.

Fig.9 PCF8574 and PCF8574A slave addresses.

Each of the PCF8574's eight I/Os can be independently used as an input or output. Input data is transferred from the port to the microcontroller by the READ mode (see Fig.11). Output data is transmitted to the port by the WRITE mode (see Fig.10).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

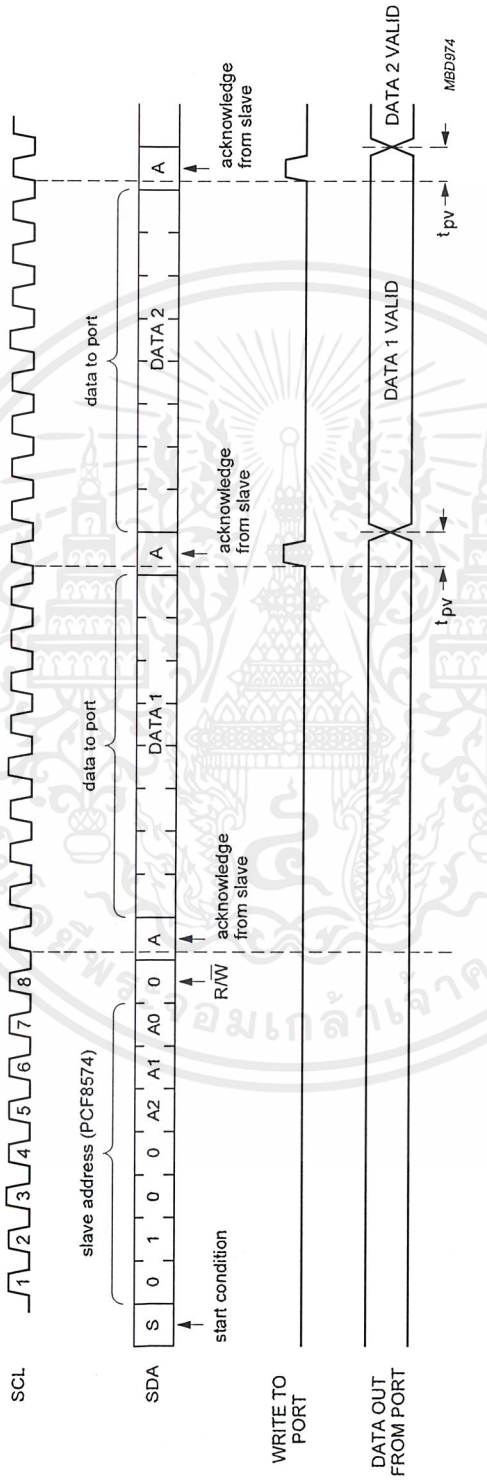
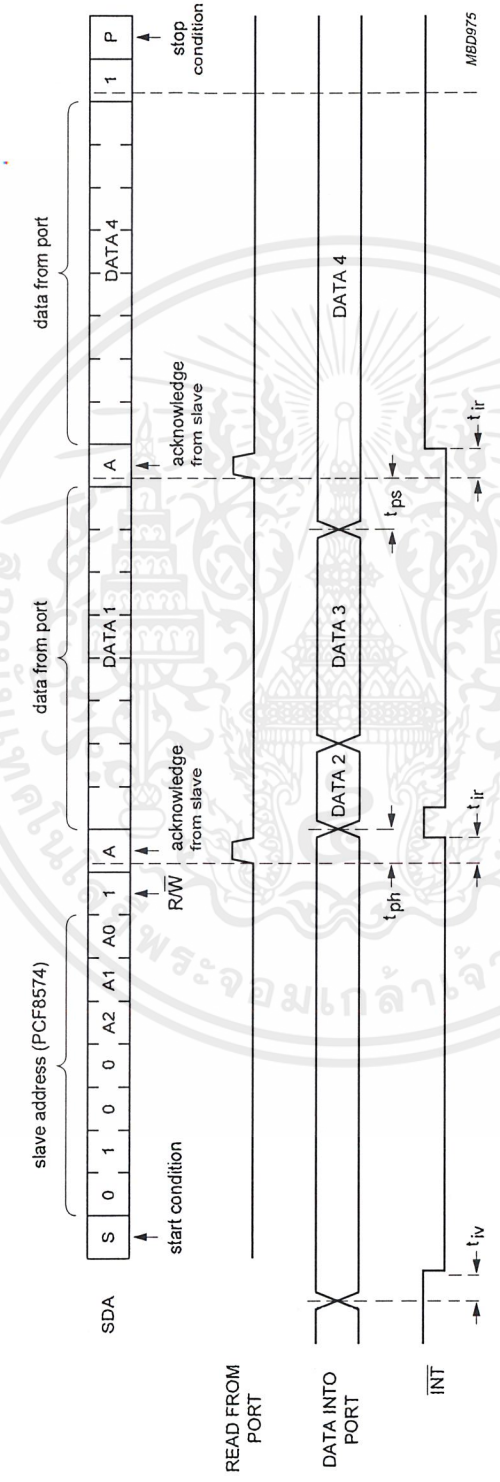


Fig.10 WRITE mode (output).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



A LOW-to-HIGH transition of SDA, while SCL is HIGH is defined as the stop condition (P). Transfer of data can be stopped at any moment by a stop condition. When this occurs, data present at the last acknowledge phase is valid (output mode). Input data is lost.

Fig.11 READ mode (input).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 Interrupt (see Figs 12 and 13)

The PCF8574 provides an open drain output ($\overline{INT}$) which can be fed to a corresponding input of the microcontroller. This gives these chips a type of master function which can initiate an action elsewhere in the system.

An interrupt is generated by any rising or falling edge of the port inputs in the input mode. After time t_{iv} the signal $\overline{INT}$ is valid.

Resetting and reactivating the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from or written to the port which has generated the interrupt.

Resetting occurs as follows:

- In the READ mode at the acknowledge bit after the rising edge of the SCL signal
- In the WRITE mode at the acknowledge bit after the HIGH-to-LOW transition of the SCL signal
- Interrupts which occur during the acknowledge clock pulse may be lost (or very short) due to the resetting of the interrupt during this pulse.

Each change of the I/Os after resetting will be detected and, after the next rising clock edge, will be transmitted as $\overline{INT}$. Reading from or writing to another device does not affect the interrupt circuit.

7.3 Quasi-bidirectional I/Os (see Fig.14)

A quasi-bidirectional I/O can be used as an input or output without the use of a control signal for data direction. At power-on the I/Os are HIGH. In this mode only a current source to V_{DD} is active. An additional strong pull-up to V_{DD} allows fast rising edges into heavily loaded outputs. These devices turn on when an output is written HIGH, and are switched off by the negative edge of SCL. The I/Os should be HIGH before being used as inputs.

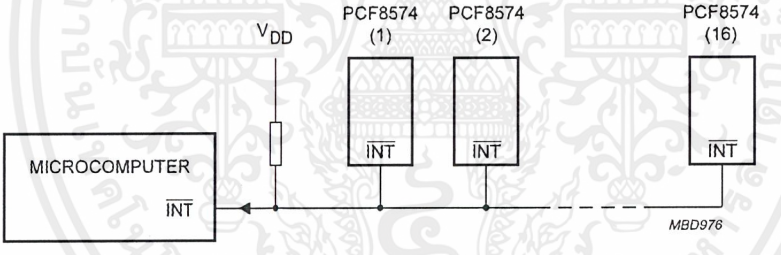


Fig.12 Application of multiple PCF8574s with interrupt.

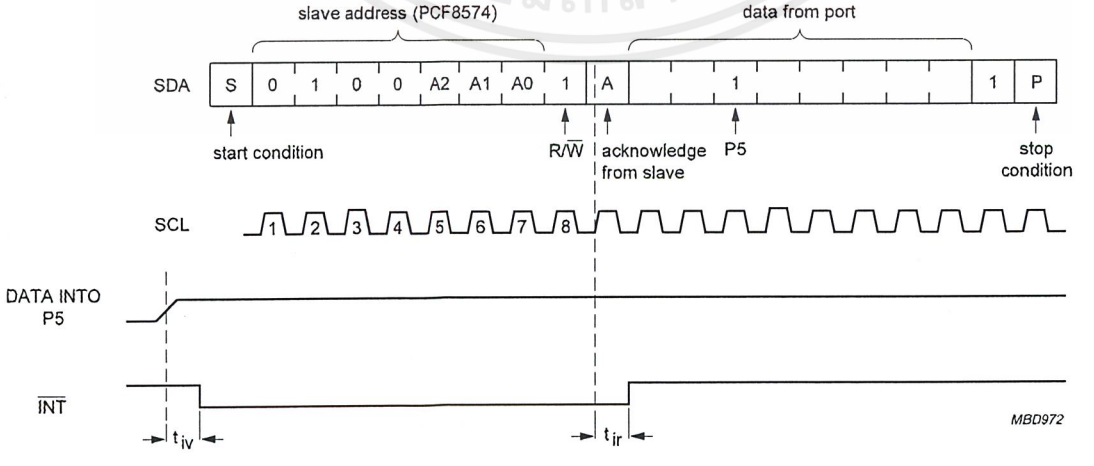


Fig.13 Interrupt generated by a change of input to I/O P5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

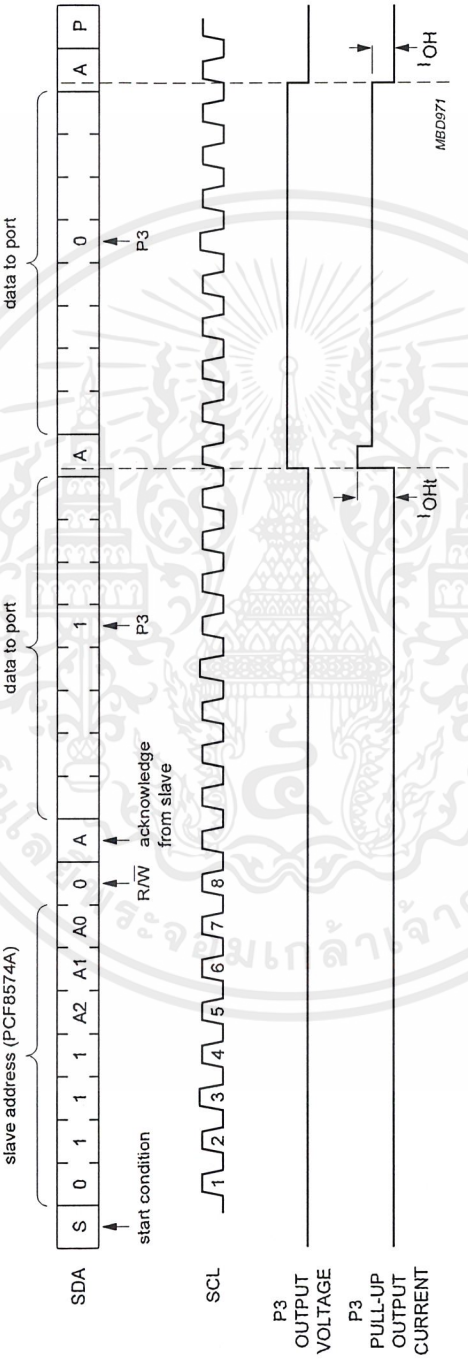


Fig.14 Transient pull-up current I_{OHt} while P3 changes from LOW-to-HIGH and back to LOW.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{DD}	supply voltage	• 0.5	+7.0	V
V_I	input voltage	V_{SS} • 0.5	V_{DD} + 0.5	V
I_I	DC input current	•	• 20	mA
I_O	DC output current	•	• 25	mA
I_{DD}	supply current	•	• 100	mA
I_{SS}	supply current	•	• 100	mA
P_{tot}	total power dissipation	•	400	mW
P_O	power dissipation per output	•	100	mW
T_{stg}	storage temperature	• 65	+150	•C
T_{amb}	operating ambient temperature	• 40	+85	•C

9 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take precautions appropriate to handling MOS devices. Advice can be found in Data Handbook IC12 under *"Handling MOS Devices"*.

10 DC CHARACTERISTICS

V_{DD} = 2.5 to 6 V; V_{SS} = 0 V; T_{amb} = • 40 to +85 •C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{DD}	supply voltage		2.5	•	6.0	V
I_{DD}	supply current	operating mode; V_{DD} = 6 V; no load; V_I = V_{DD} or V_{SS}; f_{SCL} = 100 kHz	•	40	100	• A
I_{stb}	standby current	standby mode; V_{DD} = 6 V; no load; V_I = V_{DD} or V_{SS}	•	2.5	10	• A
V_{POR}	Power-on reset voltage	V_{DD} = 6 V; no load; V_I = V_{DD} or V_{SS}; note 1	•	1.3	2.4	V
Input SCL; input/output SDA						
V_{IL}	LOW level input voltage		• 0.5	•	+0.3V_{DD}	V
V_{IH}	HIGH level input voltage		0.7V_{DD}	•	V_{DD} + 0.5	V
I_{OL}	LOW level output current	V_{OL} = 0.4 V	3	•	•	mA
I_L	leakage current	V_I = V_{DD} or V_{SS}	• 1	•	+1	• A
C_i	input capacitance	V_I = V_{SS}	•	•	7	pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I/Os						
V_{IL}	LOW level input voltage		• 0.5	•	+0.3V_{DD}	V
V_{IH}	HIGH level input voltage		0.7V_{DD}	•	V_{DD} + 0.5	V
$I_{IHL(max)}$	maximum allowed input current through protection diode	V_I • V_{DD} or V_I • V_{SS}	•	•	• 400	• A
I_{OL}	LOW level output current	V_{OL} = 1 V; V_{DD} = 5 V	10	25	•	mA
I_{OH}	HIGH level output current	V_{OH} = V_{SS}	30	•	300	• A
I_{OHt}	transient pull-up current	HIGH during acknowledge (see Fig.14); V_{OH} = V_{SS}; V_{DD} = 2.5 V	•	• 1	•	mA
C_i	input capacitance		•	•	10	pF
C_o	output capacitance		•	•	10	pF
Port timing; C_L • 100 pF (see Figs 10 and 11)						
t_{pv}	output data valid		•	•	4	• s
t_{su}	input data set-up time		0	•	•	• s
t_h	input data hold time		4	•	•	• s
Interrupt $\overline{INT}$ (see Fig.13)						
I_{OL}	LOW level output current	V_{OL} = 0.4 V	1.6	•	•	mA
I_L	leakage current	V_I = V_{DD} or V_{SS}	• 1	•	+1	• A
TIMING; C_L • 100 pF						
t_{iv}	input data valid time		•	•	4	• s
t_{ir}	reset delay time		•	•	4	• s
Select inputs A0 to A2						
V_{IL}	LOW level input voltage		• 0.5	•	+0.3V_{DD}	V
V_{IH}	HIGH level input voltage		0.7V_{DD}	•	V_{DD} + 0.5	V
I_{LI}	input leakage current	pin at V_{DD} or V_{SS}	• 250	•	+250	nA

Note

1. The Power-on reset circuit resets the I²C-bus logic with V_{DD} < V_{POR} and sets all I/Os to logic 1 (with current source to V_{DD}).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11 I²C-BUS TIMING CHARACTERISTICS

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
I²C-BUS TIMING (see Fig.15; note 1)					
f_{SCL}	SCL clock frequency	•	•	100	kHz
t_{SW}	tolerable spike width on bus	•	•	100	ns
t_{BUF}	bus free time	4.7	•	•	• s
$t_{SU;STA}$	START condition set-up time	4.7	•	•	• s
$t_{HD;STA}$	START condition hold time	4.0	•	•	• s
t_{LOW}	SCL LOW time	4.7	•	•	• s
t_{HIGH}	SCL HIGH time	4.0	•	•	• s
t_r	SCL and SDA rise time	•	•	1.0	• s
t_f	SCL and SDA fall time	•	•	0.3	• s
$t_{SU;DAT}$	data set-up time	250	•	•	ns
$t_{HD;DAT}$	data hold time	0	•	•	ns
$t_{VD;DAT}$	SCL LOW to data out valid	•	•	3.4	• s
$t_{SU;STO}$	STOP condition set-up time	4.0	•	•	• s

Note

1. All the timing values are valid within the operating supply voltage and ambient temperature range and refer to V_{IL} and V_{IH} with an input voltage swing of V_{SS} to V_{DD}.

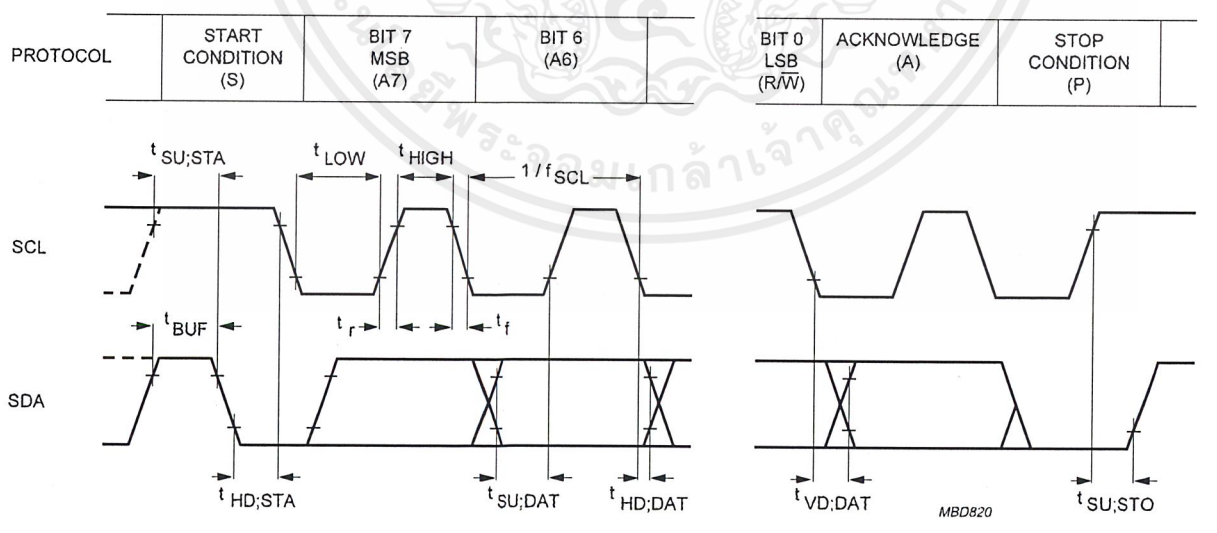
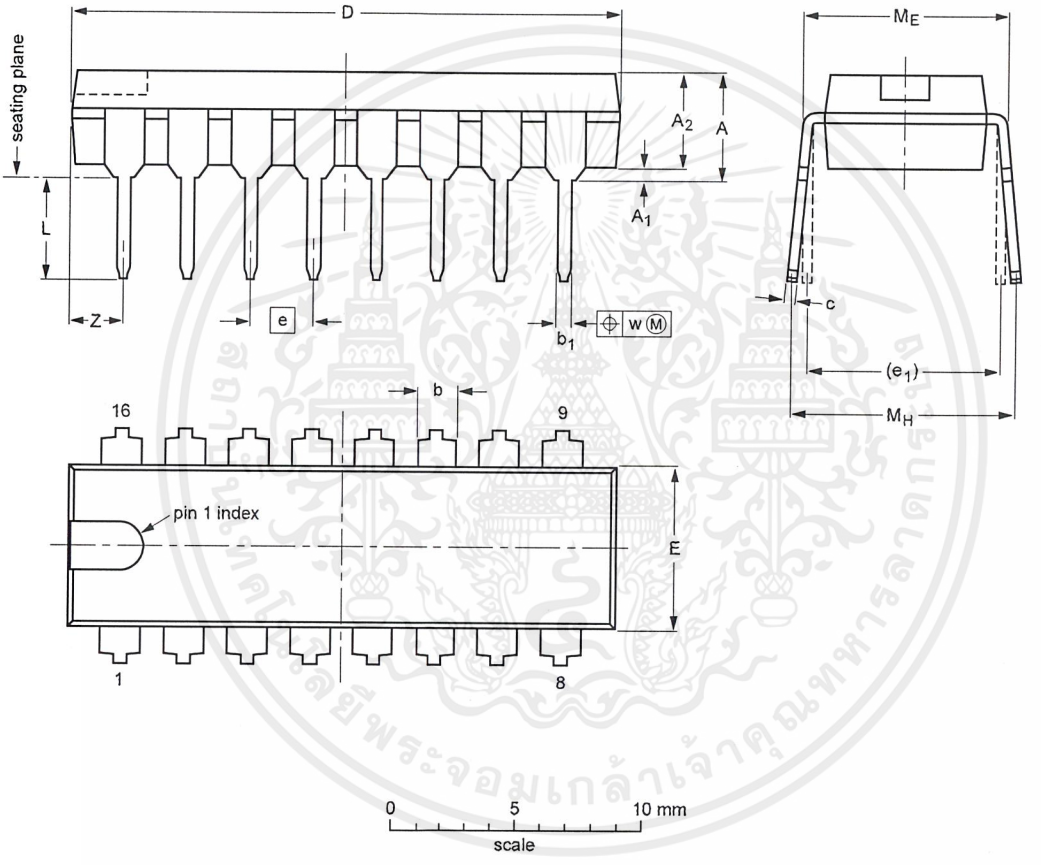


Fig.15 I²C-bus timing diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12 PACKAGE OUTLINES

DIP16: plastic dual in-line package; 16 leads (300 mil); long body — SOT38-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.7	0.51	3.7	1.40 1.14	0.53 0.38	0.32 0.23	21.8 21.4	6.48 6.20	2.54	7.62	3.9 3.4	8.25 7.80	9.5 8.3	0.254	2.2
inches	0.19	0.020	0.15	0.055 0.045	0.021 0.015	0.013 0.009	0.86 0.84	0.26 0.24	0.10	0.30	0.15 0.13	0.32 0.31	0.37 0.33	0.01	0.087

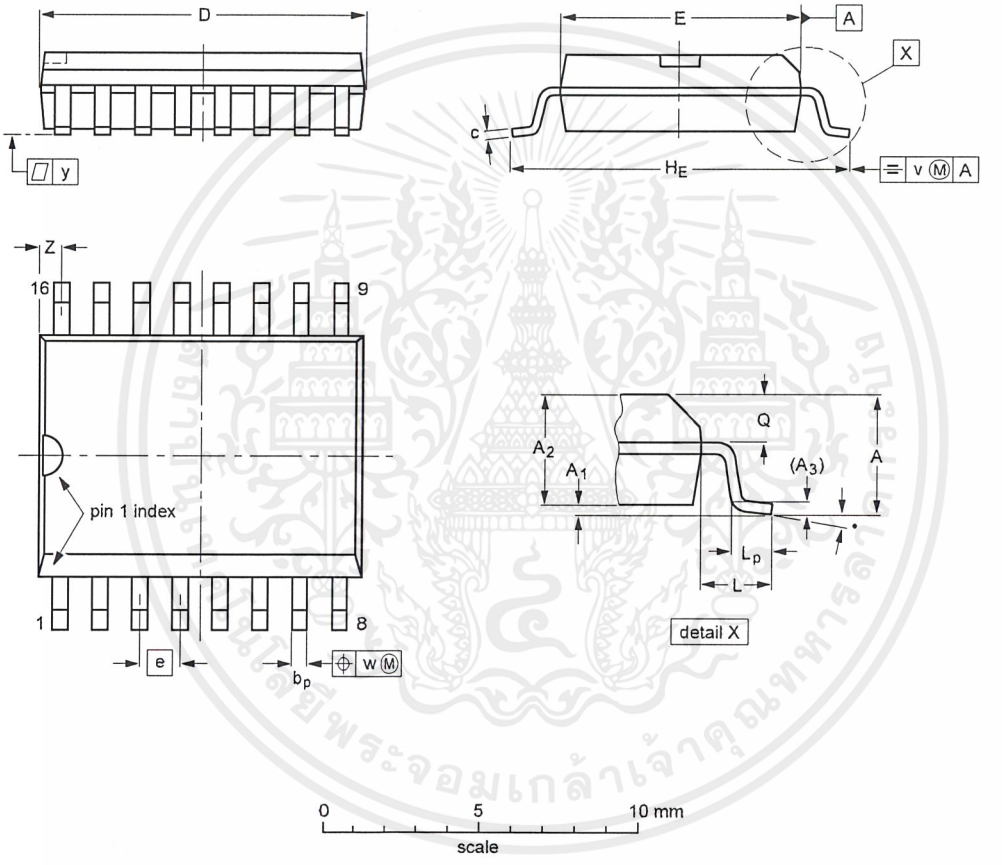
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT38-1	050G09	MO-001AE				~~92-10-02~~ 95-01-19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SO16: plastic small outline package; 16 leads; body width 7.5 mm SOT162-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	10.5 10.1	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.41 0.40	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

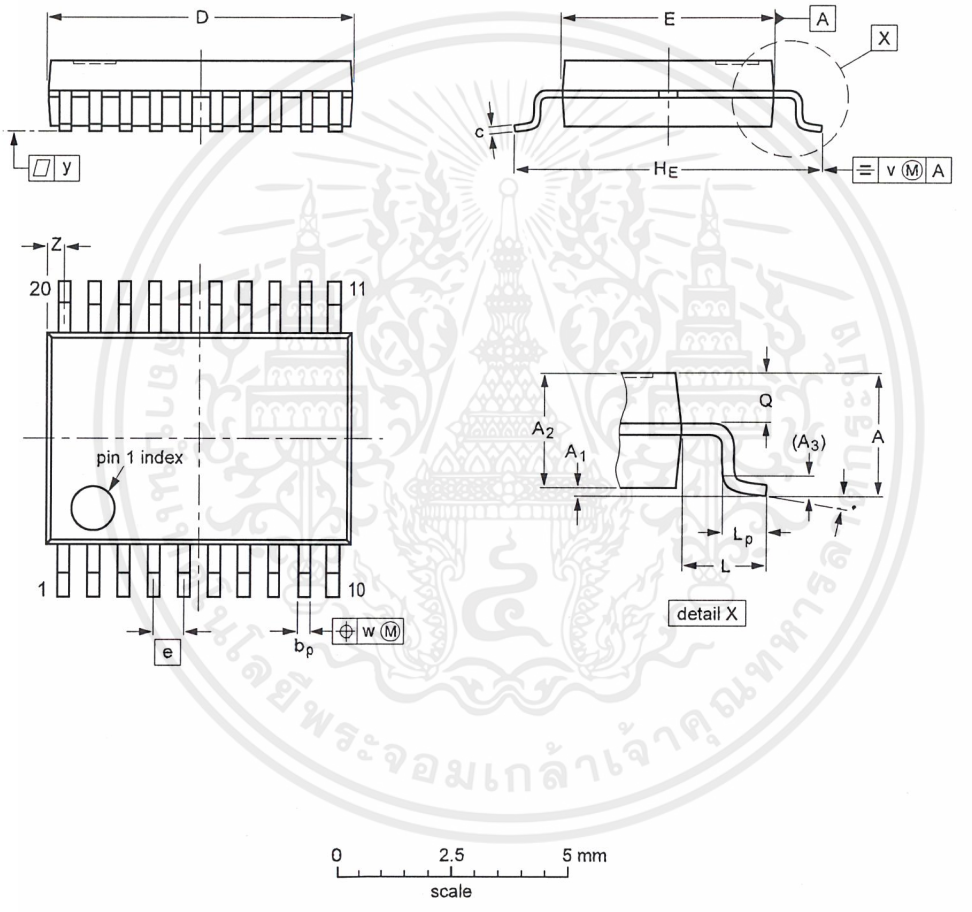
Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT162-1	075E03	MS-013AA				~~95-01-24~~ 97-05-22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SSOP20: plastic shrink small outline package; 20 leads; body width 4.4 mm SOT266-1



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	1.5	0.15 0	1.4 1.2	0.25	0.32 0.20	0.20 0.13	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1.0	0.75 0.45	0.65 0.45	0.2	0.13	0.1	0.48 0.18	10° 0°

Note

1. Plastic or metal protrusions of 0.20 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES					EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ				
SOT266-1							~~90-04-05~~ 95-02-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13 SOLDERING

13.1 Introduction

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"IC Package Databook"* (order code 9398 652 90011).

13.2 DIP

13.2.1 Soldering by dipping or by wave

The maximum permissible temperature of the solder is 260 •C; solder at this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg\,max}$). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

13.2.2 Repairing soldered joints

Apply a low voltage soldering iron (less than 24 V) to the lead(s) of the package, below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 •C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400 •C, contact may be up to 5 seconds.

13.3 SO and SSOP

13.3.1 Reflow soldering

Reflow soldering techniques are suitable for all SO and SSOP packages.

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt. Dwell times vary between 50 and 300 seconds depending on heating method. Typical reflow temperatures range from 215 to 250 •C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 minutes at 45 •C.

13.3.2 Wave soldering

Wave soldering is not recommended for SSOP packages. This is because of the likelihood of solder bridging due to closely-spaced leads and the possibility of incomplete solder penetration in multi-lead devices.

If wave soldering cannot be avoided, the following conditions must be observed:

- A double-wave (a turbulent wave with high upward pressure followed by a smooth laminar wave) soldering technique should be used.
- The longitudinal axis of the package footprint must be parallel to the solder flow and must incorporate solder thieves at the downstream end.

Even with these conditions, only consider wave soldering SSOP packages that have a body width of 4.4 mm, that is SSOP16 (SOT369-1) or SSOP20 (SOT266-1).

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Maximum permissible solder temperature is 260 •C, and maximum duration of package immersion in solder is 10 seconds, if cooled to less than 150 •C within 6 seconds. Typical dwell time is 4 seconds at 250 •C.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

13.3.3 Repairing soldered joints

Fix the component by first soldering two diagonally-opposite end leads. Use only a low voltage soldering iron (less than 24 V) applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 •C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 •C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14 DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

15 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

16 PURCHASE OF PHILIPS I²C COMPONENTS

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213,
Tel. +43 1 60 101, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 200 733, Fax. +375 172 200 773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel. +359 2 689 211, Fax. +359 2 689 102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel. +1 800 234 7381

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. +45 32 88 2636, Fax. +45 31 57 1949

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 9 615800, Fax. +358 9 61580/xxx

France: 4 Rue du Port-aux-Vins, BP317, 92156 SURESNES Cedex,
Tel. +33 1 40 99 6161, Fax. +33 1 40 99 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 23 53 60, Fax. +49 40 23 536 300

Greece: No. 15, 25th March Street, GR 17778 TAVROS/ATHENS,
Tel. +30 1 4894 339/239, Fax. +30 1 4814 240

Hungary: see Austria

India: Philips INDIA Ltd, Shivsagar Estate, A Block, Dr. Annie Besant Rd.
Worli, MUMBAI 400 018, Tel. +91 22 4938 541, Fax. +91 22 4938 722

Indonesia: see Singapore

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, TEL AVIV 61180,
Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,
20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +9-5 800 234 7381

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 82785, Fax. +31 40 27 88399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

Philippines: Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA,
Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 755 6918, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,
Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Rua do Rocio 220, 5th floor, Suite 51,
04552-903 São Paulo, SÃO PAULO - SP, Brazil,
Tel. +55 11 821 2333, Fax. +55 11 829 1849

Spain: Balmes 22, 08007 BARCELONA,
Tel. +34 3 301 6312, Fax. +34 3 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 632 2000, Fax. +46 8 632 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2686, Fax. +41 1 481 7730

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1,
TAIPEI, Taiwan Tel. +886 2 2134 2870, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7,
252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 625 344, Fax.+381 11 635 777

For all other countries apply to: Philips Semiconductors, Marketing & Sales Communications,
Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: http://www.semiconductors.philips.com

© Philips Electronics N.V. 1997

SCA53

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands 417067/1200/02/pp24 Date of release: 1997 Apr 02 Document order number: 9397 750 01758

Let's make things better.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Philips Semiconductors

PHILIPS

DALLAS SEMICONDUCTOR

DS1307
64 X 8 Serial Real Time Clock

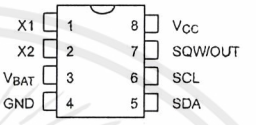
FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2–wire serial interface
- Programmable squarewave output signal
- Automatic power fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode at 25•C
- Optional industrial temperature range –40•C to +85•C (IND)
- Available in 8–pin DIP or SOIC

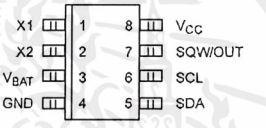
ORDERING INFORMATION

DS1307	Serial Timekeeping Chip; 8–pin DIP
DS1307Z	Serial Timekeeping Chip; 8–pin SOIC (150 mil)
DS1307N	8–pin DIP (IND)
DS1307ZN	8–pin SOIC (IND)

PIN ASSIGNMENT



DS1307 8–PIN DIP (300 MIL)



DS1307Z 8–PIN SOIC (150 MIL)

PIN DESCRIPTION

V_{CC}	–	Primary Power Supply
X1, X2	–	32.768 KHz Crystal Connection
V_{BAT}	–	+3 Volt Battery Input
GND	–	Ground
SDA	–	Serial Data
SCL	–	Serial Clock
SQW/OUT	–	Square wave/Output Driver

DESCRIPTION

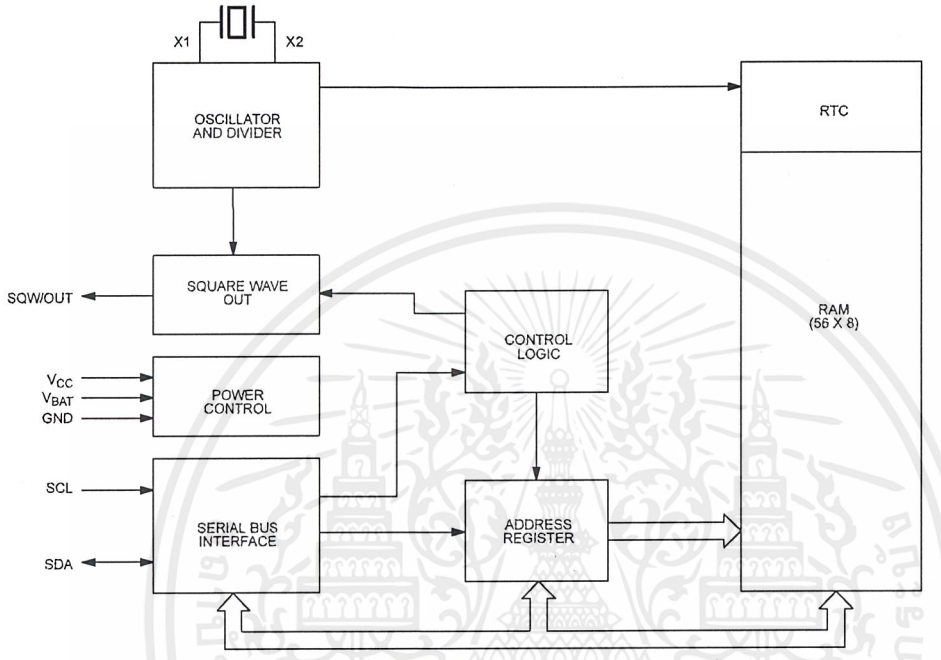
The DS1307 Serial Real Time Clock is a low power full BCD clock calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2–wire bi–directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24–hour or 12–hour format with AM/PM indicator. The DS1307 has a built–in power sense circuit which detects power failures and automatically switches to the battery supply.

OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT} the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low current battery backup mode. Upon power up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT}+0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock. The following paragraphs describe the function of each pin.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

V_{CC}, GND – DC power is provided to the device on these pins. V_{CC} is the +5 volt input. When 5 volts are applied within normal limits, the device is fully accessible and data can be written and read. When a 3 volt battery is connected to the device and V_{CC} is below 1.25 x V_{BAT}, reads and writes are inhibited. However, the Timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below V_{BAT} the RAM and timekeeper are switched over to the external 3 volt battery.

V_{BAT} – Battery input for any standard 3 volt lithium cell or other energy source. Battery voltage must be held between 2.5 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which access to the real time clock and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A Lithium battery with 35 mAh or greater will back up the DS1307 for more than 10 years in the absence of power.

SCL (Serial Clock Input) – SCL is used to synchronize data movement on the serial interface.

SDA (Serial Data Input/Output) – SDA is the input/output pin for the 2–wire serial interface. The SDA pin is open drain which requires an external pull–up resistor.

SQW/OUT (Square Wave/ Output Driver) – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1 Hz, 4 KHz, 8 KHz, 32 KHz). The SQW/OUT pin is open drain which requires an external pull–up resistor.

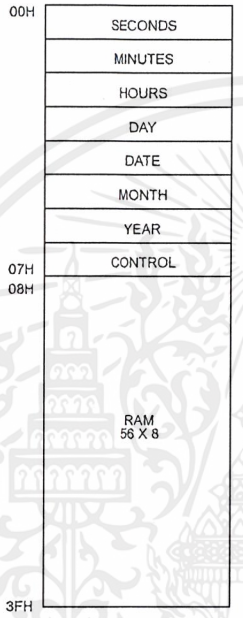
X1, X2 – Connections for a standard 32.768 KHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The real time clock registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

DS1307 ADDRESS MAP Figure 2



CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The real time clock registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the Binary–Coded Decimal (BCD) format. Bit 7 of Register 0 is the Clock Halt (CH) bit. When this bit is set to a one, the oscillator is disabled. When cleared to a zero, the oscillator is enabled.

The DS1307 can be run in either 12–hour or 24–hour mode. Bit 6 of the hours register is defined as the 12– or 24–hour mode select bit. When high, the 12–hour mode is selected. In the 12–hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24–hour mode, bit 5 is the second 10 hour bit (20–23 hours).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307 TIMEKEEPER REGISTERS Figure 3

	BIT7							BIT0	
00H	CH	10 SECONDS			SECONDS				00–59
	X	10 MINUTES			MINUTES				00–59
	X	12/24	10 HR A/P	10 HR	HOURS				01–12 / 00–23
	X	X	X	X	X	DAY			1–7
	X	X	10 DATE		DATE				01–28/29 / 01–30 / 01–31
	X	X	10 MONTH		MONTH				01–12
	10 YEAR				YEAR				00–99
07H	OUT	X	X	SQWE	X	X	RS1	RS0	

CONTROL REGISTER

The DS1307 Control Register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	X	X	SQWE	X	X	RS1	RS0

OUT (Output control): This bit controls the output level of the SQW/OUT pin when the square wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0.

SQWE (Square wave Enable): This bit when set to a logic 1 will enable the oscillator output. The frequency of the square wave output depends on the value of the RS0 and RS1 bits.

RS (Rate Select): These bits control the frequency of the square wave output when the square wave output has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

SQUAREWAVE OUTPUT FREQUENCY Table 1

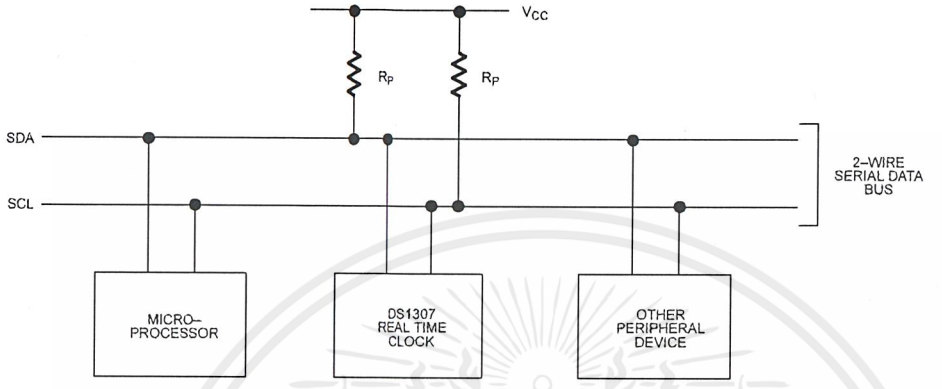
RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4 KHz
1	0	8 KHz
1	1	32 KHz

2–WIRE SERIAL DATA BUS

The DS1307 supports a bi–directional 2–wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are slaves. The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the 2–wire bus. A typical bus configuration using this 2–wire protocol is show in Figure 4.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL 2–WIRE BUS CONFIGURATION Figure 4



The following bus protocol has been defined (see Figure 5).

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

Start data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

Stop data transfer: A change in the state of the data line from low to high, while the clock line is high defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the high period of the clock signal. The data on the line must be changed during the low period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP conditions is not limited, and is determined by the master device. The information is transferred byte–wise and each receiver acknowledges with a ninth bit.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

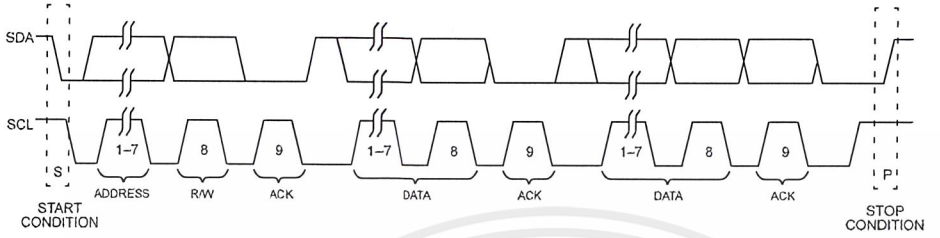
A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. When receiving data from a slave a master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line high to enable the master to generate the STOP condition.

DATA TRANSFER

Figures 5, 6, and 7 detail how data transfer is accomplished on the 2–wire bus. Depending on the state of the R/$\overline{W}$ bit in the transmission protocols as shown in Figures 6 and 7, two types of data transfer are possible:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA TRANSFER ON 2–WIRE SERIAL BUS Figure 5



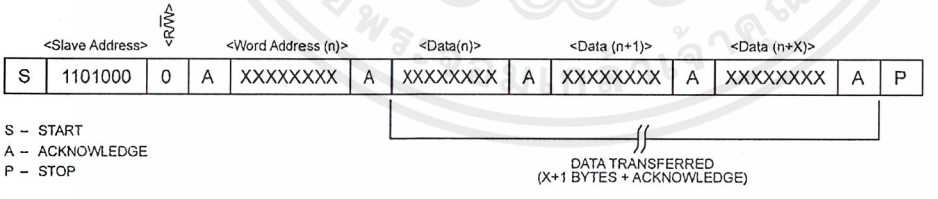
1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transfered with the most significant bit (MSB) first.
2. Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a 'not acknowledge' is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transfered with the most significant bit (MSB) first.

The DS1307 may operate in the following two modes:

1. Slave receiver mode (DS1307 write mode): Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the direction bit (R/W\) which for a write is a 0. After receiving and decoding the address byte the DS1307 outputs an acknowledge on the SDA line. After the DS1307 acknowledges the slave address + write bit, the master transmits a register address to the DS1307 This will set the register pointer on the DS1307. The master will then begin transmitting each byte of data with the DS1307 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

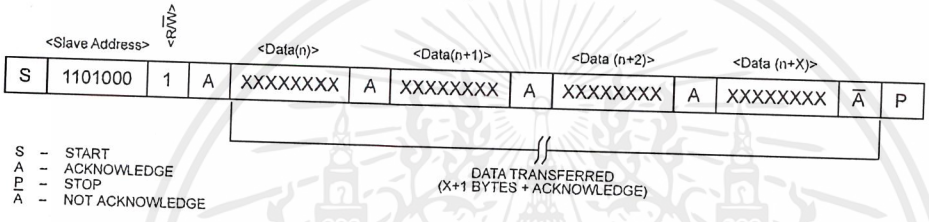
DATA WRITE – SLAVE RECEIVER MODE Figure 6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Slave transmitter mode (DS1307 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the direction bit (R/$\overline{W}$) which for a read is a 1. After receiving and decoding the address byte the DS1307 inputs an acknowledge on the SDA line. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307 must receive a Not Acknowledge to end a read.

DATA READ – SLAVE TRANSMITTER MODE Figure 7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	–0.5V to +7.0V
Operating Temperature	0•C to 70•C
Storage Temperature	–55•C to +125•C
Soldering Temperature	260•C for 10 seconds

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

The Dallas Semiconductor DS1307 is built to the highest quality standards and manufactured for long term reliability. All Dallas Semiconductor devices are made using the same quality materials and manufacturing methods. However, standard versions of the DS1307 are not exposed to environmental stresses, such as burn–in, that some industrial applications require. Products which have successfully passed through this series of environmental stresses are marked IND or N, denoting their extended operating temperature and reliability rating. For specific reliability information on this product, please contact the factory at (972) 371–4448.

RECOMMENDED DC OPERATING CONDITIONS

(0•C to 70•C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	1
Logic 1	V_{IH}	2.2		V_{CC}+0.3	V	1
Logic 0	V_{IL}	–0.3		+0.8	V	1
V_{BAT} Battery Voltage	V_{BAT}	2.5		3.5	V	1

DC ELECTRICAL CHARACTERISTICS

(0•C to 70•C; V_{CC}=4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage	I_{LI}			1	•A	10
I/O Leakage	I_{LO}			1	•A	11
Logic 0 Output	V_{OL}			0.4	V	2
Active Supply Current	I_{CCA}			1.5	mA	9
Standby Current	I_{CCS}			200	•A	3
Battery Current (OSC ON); SQW/OUT OFF	I_{BAT1}		300	500	nA	4
Battery Current (OSC ON); SQW/OUT ON (32 KHz)	I_{BAT2}		480	800	nA	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS

(0•C to 70•C; V_{CC}=4.5V to 5.5V)

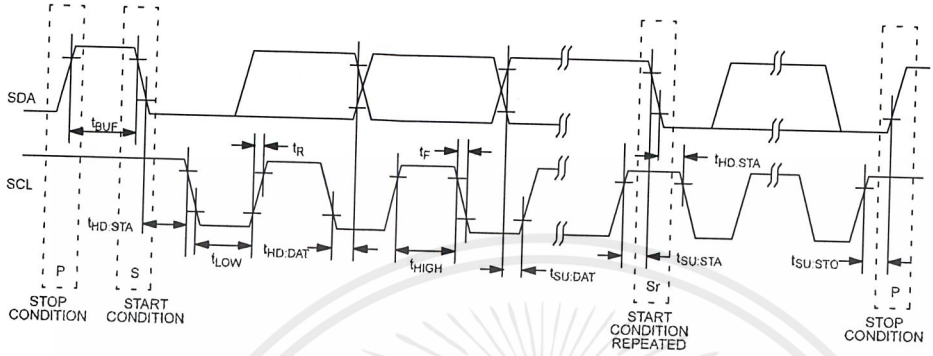
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	KHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			• s	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			• s	5
LOW Period of SCL Clock	t_{LOW}	4.7			• s	
HIGH Period of SCL Clock	t_{HIGH}	4.0			• s	
Set–up Time for a Repeated START Condition	$t_{SU:STA}$	4.7			• s	
Data Hold Time	$t_{HD:DAT}$	0			• s	6, 7
Data Set–up Time	$t_{SU:DAT}$	250			ns	
Rise Time of Both SDA and SCL Signals	t_R			1000	ns	
Fall Time of Both SDA and SCL Signals	t_F			300	ns	
Set–up Time for STOP Condition	$t_{SU:STO}$	4.7			• s	
Capacitive Load for each Bus Line	C_B			400	pF	8
I/O Capacitance	$C_{I/O}$		10		pF	
Crystal Capacitance			12.5		pF	

NOTES:

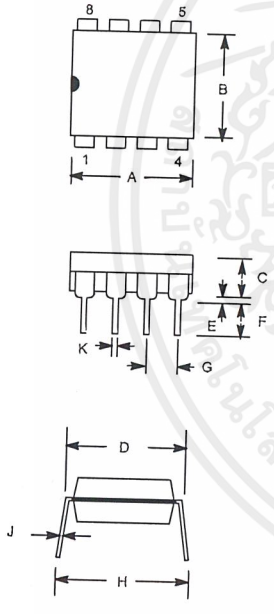
1. All voltages are referenced to ground.
2. Logic zero voltages are specified at a sink current of 5 mA at V_{CC}=4.5V, V_{OL}=GND for capacitive loads.
3. I_{CCS} specified with V_{CC}=5.0V and SDA, SCL=5.0V.
4. V_{CC}=0V, V_{BAT}=3V.
5. After this period, the first clock pulse is generated.
6. A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHMIN} of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
7. The maximum $t_{HD:DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.
8. C_B – total capacitance of one bus line in pF.
9. I_{CCA} – SCL clocking at max frequency = 100 KHz.
10. SCL only.
11. SDA and SQW/OUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING DIAGRAM



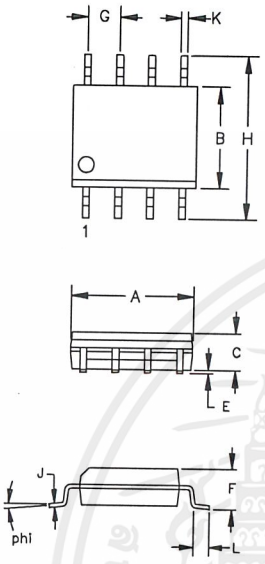
DS1307 64 X 8 SERIAL REAL TIME CLOCK 8–PIN DIP



PKG	8–PIN	
DIM	MIN	MAX
A IN. MM	0.360 9.14	0.400 10.16
B IN. MM	0.240 6.10	0.260 6.60
C IN. MM	0.120 3.05	0.140 3.56
D IN. MM	0.300 7.62	0.325 8.26
E IN. MM	0.015 0.38	0.040 1.02
F IN. MM	0.120 3.04	0.140 3.56
G IN. MM	0.090 2.29	0.110 2.79
H IN. MM	0.320 8.13	0.370 9.40
J IN. MM	0.008 0.20	0.012 0.30
K IN. MM	0.015 0.38	0.021 0.53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307Z 64 X 8 SERIAL REAL TIME CLOCK 8–PIN SOIC (150 MIL)



PKG	8–PIN (150 MIL)	
DIM	MIN	MAX
A IN. MM	0.188 4.78	0.196 4.98
B IN. MM	0.150 3.81	0.158 4.01
C IN. MM	0.048 1.22	0.062 1.57
E IN. MM	0.004 0.10	0.010 0.25
F IN. MM	0.053 1.35	0.069 1.75
G IN. MM	0.050 BSC 1.27 BSC	
H IN. MM	0.230 5.84	0.244 6.20
J IN. MM	0.007 0.18	0.011 0.28
K IN. MM	0.012 0.30	0.020 0.51
L IN. MM	0.016 0.41	0.050 1.27
phi	0•	8•

56–G2008–001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้