

การประยุกต์ใช้ดีสเพลย์กับระบบคิว

DISPLAY APPLICATION FOR QUEUING SYSTEM



นายนิธิพัฒน์ สิงหาวิวัฒน์  
นายพงษ์รัช แก้วกล้า  
นายสมฤทัย เกิดพุ่ม

เลขหมู่.....  
เลขทะเบียน...50220  
วัน,เดือน,ปี 28 เม.ย. 2547

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

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักหอสมุดกลางพระจอมเกล้าลาดกระบัง ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงข้อมูลและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2545

# DISPLAY APPLICATION FOR QUEUING SYSTEM



A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำไปใช้ในเชิงพาณิชย์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
2002



หัวข้อปริญญานิพนธ์	การประยุกต์ดีสเพลย์กับระบบคิว
	DISPLAY APPLICATION FOR QUEUING SYSTEM
นักศึกษาผู้จัดทำ	นายนิธิพัฒน์ สิงหานุวัฒน์
	นายพงษ์รัช แก้วกล้า
	นายสมฤทัย เกิดพุ่ม
อาจารย์ที่ปรึกษา	รศ.สุพรรณ กุลพณิชย์
ปีการศึกษา	2545

### บทคัดย่อ

การประยุกต์ดีสเพลย์กับระบบคิว ประกอบด้วยส่วนต่าง ๆ หลายส่วนดังนั้นจึงจำเป็นที่จะต้องศึกษาค้นคว้า และ ทำการพัฒนาส่วนต่าง ๆ ทีละส่วน ที่ต้องเกี่ยวข้องกับไมโครคอนโทรลเลอร์ เริ่มด้วยส่วนแรกคือส่วนของบอร์ดควบคุมหลัก ทำหน้าที่แสดงหมายเลขคิว และควบคุมการพิมพ์บัตรคิว ประกอบด้วยจอแสดงผลจำนวนคิว เครื่องพิมพ์บัตรคิวและสวิตช์กดพิมพ์บัตรคิว ส่วนที่สองคือบอร์ดควบคุมตัวลูก มีจำนวนหลายชุดทำหน้าที่แสดงหมายเลขคิวในแต่ละช่องบริการ ซึ่งประกอบด้วยจอแสดงผลหมายเลขคิว และ สวิตช์กดเปลี่ยนคิว เมื่อสร้างและทดสอบส่วนสำคัญทั้งสองส่วนแล้ว จึงนำส่วนประกอบทั้งหมดมาประกอบรวมกัน โดยทุกส่วนจะเชื่อมต่อกันด้วยระบบการส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Thesis Title</b>	Display Application for Queuing System
<b>Authors</b>	Mr.Nithipat Singhanuwat Mr.Pongtawat Klaewkla Mr.Somruethai kerdpoom
<b>Thesis Advisor</b>	Assoc.Prof.Supan Kulpanich
<b>Year</b>	2002

### ABSTRACT

Display Application for Queuing System is consisted of various parts so, it is necessary to learn, research and develop is used to part by part. The system is concerned microcontroller. The first part is master board. It is used to show the number of queue and about type queue-card. The second part, is slave board. It has many subsystems for showing the number of queues in each service window. It consists of the display system and changing queue switch. These two parts are integrated to build up queuing display system, all subsystems can be communicate in the serial form.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความเมตตาจาก รองศาสตราจารย์ สุพรรณ กุลพานิชย์ และ ผู้ช่วยศาสตราจารย์ประสิทธิ์ จุลเสวีวงศ์ ที่ได้ให้คำแนะนำแก่ผู้วิจัย ตลอดจน อีกทั้งยังเอื้อเฟื้ออุปการะและเครื่องมือต่าง ๆ ในการทำปริญญาบัตรนี้ ผู้วิจัยรู้สึกซาบซึ้ง และขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ได้ให้คำแนะนำอันเป็น ประโยชน์ต่อการทำปริญญาบัตรฉบับนี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็น แรงบันดาลใจในการทำปริญญาบัตรฉบับนี้

คุณค่า และประโยชน์อันพึงมีจากปริญญาบัตรฉบับนี้ ผู้วิจัยขอบแต่ผู้มีพระคุณ ทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII

บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....	1
1.2 วัตถุประสงค์ของปริญญาานิพนธ์.....	2
1.3 ขอบเขตของปริญญาานิพนธ์.....	2
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่ได้รับจากปริญญาานิพนธ์.....	3

บทที่ 2 ความรู้เบื้องต้นและทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้สเปคัลกับระบบคิว.....	4
2.1 MCS-8051.....	4
2.2 การใช้งานพอร์ตอนุกรม(Serial Port).....	8
2.3 การสื่อสารระหว่าง ซีพียูหลายตัว(Multi Processor Communication).....	11
2.4 การสื่อสารแบบอนุกรม.....	14
2.4.1 การส่งข้อมูลแบบ ซิมเพล็กซ์และดูเพล็กซ์.....	14
2.4.2 โปรโตคอลของการสื่อสารแบบอนุกรม.....	15
2.4.3 มาตรฐานสัญญาณอนุกรมแบบ RS 232C.....	17
2.4.4 มาตรฐานสัญญาณอนุกรมแบบ RS 422.....	18
2.4.5 มาตรฐานสัญญาณอนุกรมแบบ RS 485.....	19
2.4.6 การสื่อสารข้อมูลอนุกรมด้วย RS 485.....	20
2.4.7 มาตรฐานการสื่อสารข้อมูล EIA RS 485.....	21
2.5 แผงแสดงผลแบบ 7 ส่วน(Seven Segment).....	23
2.6 พอร์ตเครื่องพิมพ์หรือพอร์ตขนาน(Centronics Port).....	25

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
2.7 ส่วนประกอบการทำงานของโปรแกรม Delphi.....	26
2.7.1 การใช้งานคอมโปเนนต์พื้นฐานของ Delphi.....	28
2.7.2 คอมโปเนนต์พื้นฐานในกลุ่ม Standard.....	30
2.7.3 การเรียกใช้งาน Template จาก Object Repository.....	32
2.7.4 การสร้างไฟล์ .EXE เพื่อนำไปใช้งานจริง.....	33
2.8 ข้อตกลงในการสื่อสาร(FAC-TALK PROTOCOL).....	34
<b>บทที่ 3 การออกแบบ.....</b>	<b>37</b>
3.1 การออกแบบในส่วนของ MASTER BOARD.....	37
3.1.1 ส่วนประกอบของ MASTER BOARD.....	37
3.2 การออกแบบในส่วนของ SLAVE BOARD.....	39
<b>บทที่ 4 สรุปการทำงาน.....</b>	<b>40</b>
4.1 แผนภาพแสดงการทำงาน.....	40
4.2 ลำดับขั้นตอนการทำงาน.....	41
<b>บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....</b>	<b>42</b>
<b>บรรณานุกรม.....</b>	<b>43</b>
<b>ภาคผนวก.....</b>	<b>44</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงการเลือกโหมดการทำงานของพอร์ตอนุกรม.....	10
2.2 แสดงการเข้ารหัสตามตัวเลขที่แสดง.....	25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

ภาพที่	หน้า
1.1 แสดงบล็อกไดอะแกรมแสดงการทำงานของการทำงานของการประยุกต์ดีสเพลย์กับระบบบิว.....	1
2.1 แสดงโครงสร้างภายในไมโครคอนโทรลเลอร์ (MCS-51).....	4
2.2 แสดงขาต่างๆของ 8051.....	5
2.3 แสดงการต่อออสซิลเลเตอร์(Oscillator Connection).....	7
2.4 แสดงการต่อแบบ External Clock Drive Configuration.....	7
2.5 แสดงบิตต่างๆ ของรีจิสเตอร์ SCON.....	9
2.6 แสดงการเชื่อมต่อไมโครคอนโทรลเลอร์หลายๆ ตัว.....	11
2.7 แสดงข้อมูลกำหนดตำแหน่ง (Address Byte ).....	12
2.8 แสดงข้อมูลข่าวสาร (Data Byte ).....	12
2.9 แสดงการสื่อสารข้อมูลอนุกรมแบบต่างๆ.....	15
2.10 แสดงแบบมาตรฐานการสื่อสารข้อมูลและอะซิงโครนัส.....	16
2.11 แสดง ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232C.....	17
2.12 แสดงลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS422.....	18
2.13 แสดงลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS485.....	19
2.14 แสดงลักษณะขาของ SN75176.....	20
2.15 แสดงการต่อแบบ Common Anode.....	23
2.16 แสดงการต่อแบบ Common Cathode.....	24
2.17 แสดงการจัดเรียงแอสีดีเป็นแผงผล 7 ส่วน.....	24
2.18 แสดง Menu Bar.....	26
2.19 แสดง Form Designer.....	26
2.20 แสดง Object Inspector.....	27
2.21 แสดง Standard.....	28
2.22 แสดง Additional.....	28
2.23 แสดง Win 32.....	28
2.24 แสดง System.....	29
2.25 แสดง Data Access.....	29
2.26 แสดง Data Controls.....	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.27 แสดงคอมโพเนนต์พื้นฐานในกลุ่ม Standard.....	30
2.28 แสดงการเลือก Template Object ของ Repository.....	32
2.29 แสดงแท็บต่างๆ ของ Object Repository.....	32
2.30 แสดงการสั่ง Build Project.....	33
3.1 แสดงวงจรการต่อ MASTER BOARD.....	38
3.2 แสดงวงจรการต่อ SLAVE BOARD.....	39
4.1 แสดงแผนภาพแสดงการทำงาน.....	40



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

ระบบคิวอัตโนมัตินี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ 89S8252 เป็นตัวควบคุมระบบทั้งหมด โดยจะมีตัว Master และ Slave คอยติดต่อกันซึ่งแยกกันทำงานคนละส่วนมีการติดต่อสื่อสารกันแบบอนุกรมทั้งหมด โดยในการติดต่อสื่อสารกันระหว่าง CPU จะใช้ติดต่อกันทาง RS 485 ซึ่งรายละเอียดที่เกี่ยวข้องจะอยู่ในส่วนของทฤษฎี การทำงานของระบบคิวอัตโนมัติจะมีตัว Master เป็นตัวที่ใช้ในการรับคิวเข้ามาเก็บไว้ในตัวของ CPU จากนั้นเมื่อตัว Slave มีการติดต่อเข้ามา ตัว Master ก็จะจ่ายคิวให้ตัว Slave ซึ่งตัว Slave ก็คือส่วนของพนักงานเพื่อใช้ในการเรียกคิว แต่ละตัวจะมีการตั้งช่องเอาไว้สามารถใช้ได้ 8 ช่อง เมื่อมีการเรียกคิวเกิดคิวขึ้นก็จะมีการแสดงที่ตัวของ Slave เพื่อบอกให้พนักงานรู้ว่า เป็นคิวอะไร จากนั้นก็จะมีการติดต่อไปยังภาคแสดงผลของคิวที่ใช้ไมโครคอนโทรลเลอร์เบอร์ 89S8252 เป็นตัวคอยรับข้อมูลเข้ามาจากตัว Master และ Slave ซึ่งเป็นส่วนในการควบคุมการแสดงผลของ LED Display เพื่อบอกคิวให้ผู้มาติดต่อทราบว่าตอนนี้กำลังเรียกคิวหมายเลขใดและบอกช่องที่จะเข้าไปใช้บริการแก่ผู้มาใช้บริการ และจะมีส่วนของภาคสังเคราะห์เสียงพูดทำให้เราได้ยินเสียงว่าตอนนี้กำลังเรียกคิวที่เท่าไร โดยใช้โปรแกรม Delphi โดยใช้ไมโครคอนโทรลเลอร์เบอร์ 89S8252 เป็นตัวดึงข้อมูลที่เรารสร้างไว้มาใช้งานซึ่งจะสัมพันธ์กันกับการแสดงผลของหมายเลขคิวเพื่อให้ผู้มาใช้บริการได้ยินอีกครั้งหนึ่ง การแสดงผลออกทาง LED Display หลักการทำงานของระบบคิวอัตโนมัติ จะแสดงยังบล็อกไดอะแกรมนี้



ภาพที่ 1.1 บล็อกไดอะแกรมแสดงการทำงานของระบบคิวอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 วัตถุประสงค์ของปริญญาานิพนธ์

การทำโครงงานชิ้นนี้ก็เพื่อใช้ความรู้ที่เรียนมามาประยุกต์ใช้ในการทำโครงงานชิ้นนี้ โดยเริ่มตั้งแต่การออกแบบของส่วนต่างๆ เพราะอะไรถึงนำอุปกรณ์เหล่านั้นมาใช้ทำ ซึ่งต้องมีการศึกษาคุณสมบัติของอุปกรณ์เหล่านั้นว่ามีคุณสมบัติอย่างไรมีการทำงานเป็นเช่นไรและตรงตามที่เรากออกแบบไว้หรือไม่ ทำให้เราสามารถสร้างชิ้นงานขึ้นมาได้ตามที่เราศึกษาและออกแบบไว้ โดยสามารถสรุปได้ดังนี้คือ

- เพื่อสร้างระบบการสื่อสาร โดยใช้ Microcontroller MCS-51 เป็นตัวประมวลผล
- เพื่อพัฒนาโปรแกรมสำหรับใช้ในการสื่อสารกับระบบที่สร้างขึ้นเพื่อให้สามารถดูการทำงานของตัว Master และ Slave โดยแสดงผลทาง Display
- เพื่อสามารถเชื่อมต่อการทำงานของตัว Master และ Slave โดยใช้ระบบการสื่อสารแบบ RS232 และ RS485

## 1.3 ขอบเขตของปริญญาานิพนธ์

- สามารถพัฒนาโปรแกรมที่ใช้กับ Microcontroller MCS-51 เพื่อใช้ในการสื่อสารกับระบบที่สร้างขึ้น
- สามารถสร้างการสื่อสารแบบอนุกรมและการสื่อสารแบบขนานโดยใช้ Microcontroller MCS-51 เป็นตัวควบคุม
- สามารถเชื่อมโยงข้อมูลจากเครื่อง PC ไปยังตัว Master และ Slave โดยผ่านทาง RS232 ได้
- สามารถสร้างส่วนต่างๆของระบบคิวตามที่ออกแบบได้
- สามารถทราบว่าส่วนต่างๆของ Master และ Slave มีการติดต่อกันได้อย่างไร
- สามารถแสดงบิตรหามยเลขคิวตั้งแต่หมายเลข 001 ถึง 999 ได้
- ออกแบบให้แสดงผลได้ 1 แถว
- ส่งเสียงเรียกคิวได้ตามหมายเลขคิวที่เรียก
- การติดต่อระหว่างตัวบอร์ดควบคุมจะใช้สายสัญญาณเพียง 2 เส้นเท่านั้น
- การทำโครงงานนี้จะทำในส่วนของภาคแสดงผลและภาคเสียงเท่านั้น

## 1.4 ขั้นตอนการศึกษา

- ศึกษาระบบการสื่อสารแบบอนุกรมกับ Microcontroller MCS-51
- ศึกษาการทำงานในการติดต่อสื่อสารข้อมูลแบบ Block Command ซึ่งจะต้องแปลงชุดคำสั่ง

เหล่านั้นให้เป็น ASCII Code ก่อน ที่จะส่งออกไปพร้อมกับ BCS หรือ FCS เพื่อตรวจสอบ

เอกสารนี้เป็นเอกสารที่รวบรวมไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ความผิดพลาดของข้อมูล ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ศึกษารายละเอียดของ Hardware และ Microcontroller MCS-51
- ศึกษาโปรแกรมที่ใช้ในการเขียนใน Microcontroller MCS-51
- สร้างระบบการสื่อสารแบบอนุกรมโดยใช้ RS232 เป็นตัวกลางในการเชื่อมต่อ
- สร้างระบบการสื่อสารแบบอนุกรมโดยใช้ RS485 เป็นตัวกลางในการเชื่อมต่อ
- ทดสอบการทำงานของระบบ
- ปรับปรุงแก้ไข

### 1.5 ประโยชน์ที่คาดว่าจะได้รับจากปริญญาโท

- ได้รับความรู้เรื่องการสื่อสารแบบอนุกรม และแบบขนาน
- เข้าใจการติดต่อสื่อสารของ Microcontroller MCS-51
- ได้รับความรู้เพิ่มเติมในการใช้งาน Microcontroller MCS-51
- ได้รับความรู้เรื่องการพัฒนาโปรแกรมโดยใช้ โปรแกรมภาษาแอสเซมบลี
- ได้รับความรู้ในการติดต่อสื่อสารข้อมูลจากเครื่อง PC ไปยังชุด Master และ Slave โดยใช้โปรแกรม Hyper Terminal เป็นตัวเขียนข้อมูลคำสั่งต่างๆ และตรวจสอบดูค่าสถานะในการรับส่งข้อมูลแบบ Block Command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

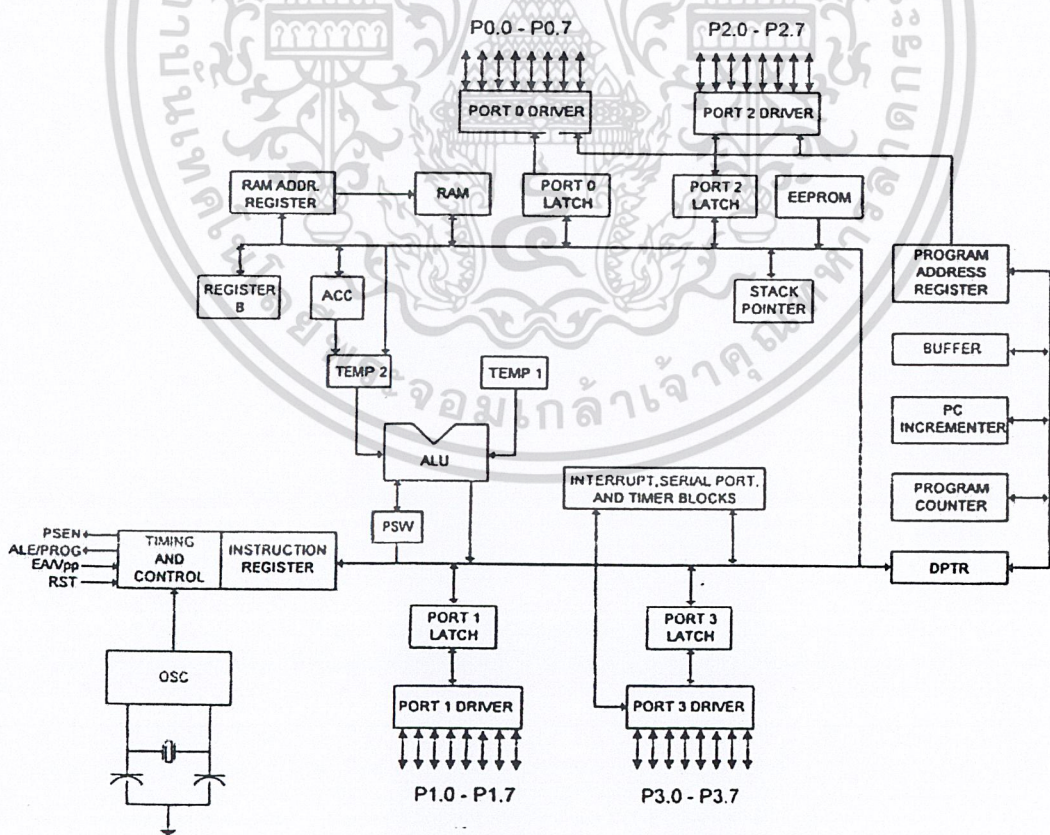
## บทที่ 2

# ความรู้เบื้องต้น และทฤษฎีที่เกี่ยวกับการประยุกต์ใช้กับระบบคิว

### 2.1 MCS-8051

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตที่มีอุปกรณ์สนับสนุนประกอบอยู่ภายในหลายอย่างได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์ มีอุปกรณ์สนับสนุนอยู่ภายในนี้เอง ทำให้การใช้งานง่ายขึ้นและมีประสิทธิภาพมากขึ้น โดยไม่ต้องมีอุปกรณ์เชื่อมต่อกับอุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับตัวไมโครโปรเซสเซอร์ (Microprocessor) ทั่วไป

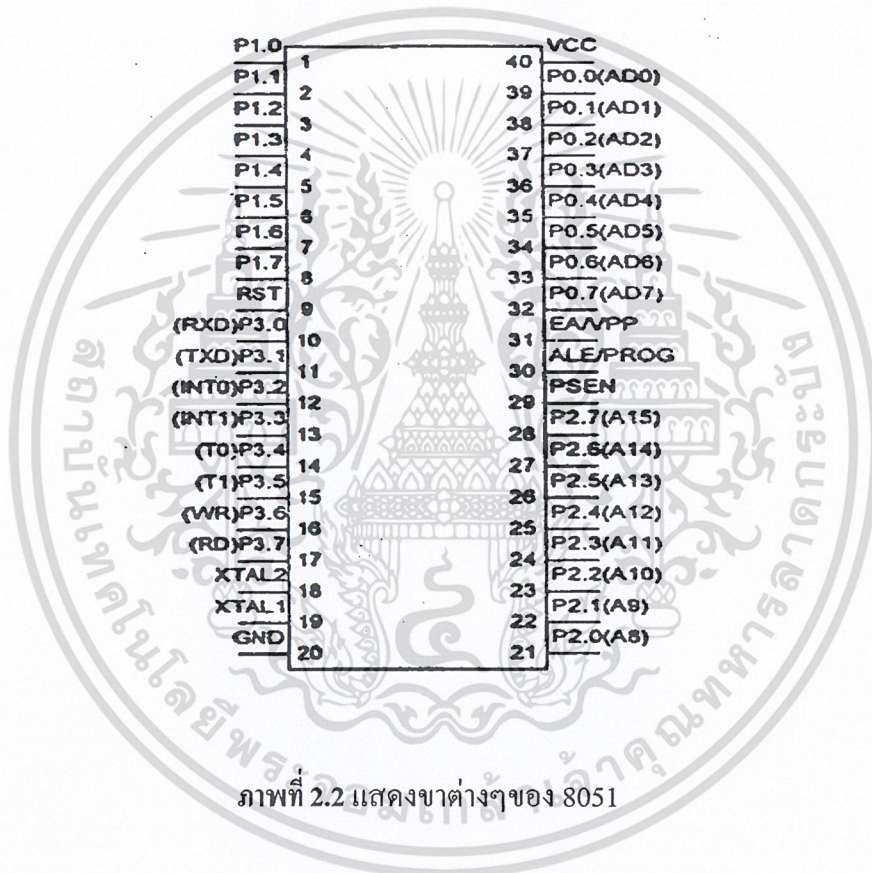
### โครงสร้างภายในไมโครคอนโทรลเลอร์ (MCS-51)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ภาพที่ 2.1 โครงสร้างภายในไมโครคอนโทรลเลอร์ (MCS-51)  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่แบบสงวนเนื้อหาและต้องอ้างอิงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8051 ประกอบด้วย อุปกรณ์ต่างๆ ดังนี้ RAM ภายใน 128 ไบต์ หน่วยความจำโปรแกรมภายในขนาด 4 Kbyte พอร์ตขนาน (Parallel Port) จำนวน 4 พอร์ตๆละ 8 บิต ไทม์เมอร์/เคานเตอร์ (Timer/Counter) ขนาด 16 บิต 2 ตัว อุปกรณ์การควบคุมการอินเทอร์รัพต์ (Interrupt) พอร์ตอนุกรม (Serial Port) แบบ Full Duplex และวงจรรักษาไฟ

## โครงสร้างภายนอกของไมโครคอนโทรลเลอร์ (MCS-8051)



ภาพที่ 2.2 แสดงขาต่างๆของ 8051

### พอร์ต 0 (ขา 32-39)

ขาของพอร์ต 0 สามารถเป็นอินพุตหรือเอาต์พุต หรือทั้งสองอย่าง ซึ่งเป็นแอดเดรสอันดับต่ำสองทางและบัสข้อมูลสำหรับความจำภายนอก สามารถใช้เป็นอิมพีแดนซ์สูง (High Impedance) ตัวอย่างเช่น เมื่อใช้ขาอินพุตเป็น “ 1 “ ต้องเขียนให้ถูกคั้งกับพอร์ต 0 Latch โดยโปรแกรม ดังนั้นการปิดทรานซิสเตอร์ด้านเอาต์พุตทั้ง 2 ตัวทำให้ขาลอย อยู่ในภาวะอิมพีแดนซ์สูง และขาต่อกับอินพุตบัฟเฟอร์ (Input Buffer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใช้เอาต์พุต ขา Latch ต้องโปรแกรมเป็น 0 ซึ่งจะเปิดเฟต (FET) ตัวล่างทำให้ขาต่อลงกราวด์ Latch ทุกขาที่โปรแกรมเป็น 1 จะยังคงลอยอยู่ ดังนั้นตัวต้านทานพูลอัพ (Pull up) จะต้องจ่ายสถานะลอจิก 1 เมื่อใช้พอร์ต 0 เป็นเอาต์พุต

### พอร์ต 1 ( ขา 1-8)

พอร์ต 1 เป็นพอร์ตที่สามารถกำหนดให้ทำงานได้แบบ 2 ทิศทาง คือเป็นได้ทั้งอินพุตหรือเอาต์พุตอย่างใดอย่างหนึ่งแล้วแต่โปรแกรมที่เขียนขึ้นควบคุม สำหรับพอร์ต 1 ที่ใช้งานบนบอร์ด MCS-51 นี้ถูกกำหนดให้ทำงานเป็นพอร์ตเอาต์พุต มีโครงสร้างคล้ายพอร์ต 0 แต่จะใช้เวลาต้านทานภายในพูลอัพแทน (Internal Pull Up Resister)

### พอร์ต 2 ( ขา 21-28 )

มีทั้งหมด 8 บิต คือขา ( ขา 1.0-1.7 ) มีโครงสร้างคล้ายพอร์ต 0 โดยมีเฟตตัวล่างตัวเดียว ส่วนด้านบนใช้เวลาต้านทานพูลอัพแทน (Internal Pull Up) พอร์ตนี้ทำงาน 2 หน้าที่คือสามารถใช้เป็นแอดเดรสบัส ขนาด 8 บิต ( A8-A15 ) และกลายเป็นไอโอพอร์ต (I/O Port ) ใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุตพอร์ตต้องส่งลอจิก” 1 “ มาที่พอร์ตนี้ก่อนเพื่อบังคับให้เฟตอยู่ในสถานะ off

### ขารีเซท (RST)

ใช้สำหรับการรีเซทการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซทการทำงานของไมโครคอนโทรลเลอร์โดยการรีเซทต้องคงสถานะเห็น 1 อย่างน้อยนาน 2 แมชชีนไซเคิล ( Machinecycle ) ในขณะที่ออสซิลเลเตอร์ (Oscillator) ยังคงทำงานอยู่

### ขา ALE/PROG

เป็นขาสัญญาณที่ทำหน้าที่ควบคุมการ Latch ค่าตำแหน่งของแอดเดรสไบต์ต่ำ (Address Latch Enable ) เมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากนี้ยังทำหน้าที่เป็นอินพุตพัลส์ในการโปรแกรม ( Program Pulse Input ) ในส่วนของหน่วยความจำ EEPROM สำหรับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ที่มีหน่วยความจำโปรแกรมภายในเป็น EEPROM

### ขา PSEN ( Program Stoe Enable )

ทำหน้าที่เป็นสัญญาณสโตรบ ( Strobe ) เพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตรบ ( Strobe ) จำนวน 2 ครั้งในแต่ละแมชชีนไซเคิล แต่ในขณะที่ติดต่อกับหน่วยความจำข้อมูลภายนอก จะไม่มีการส่งสัญญาณสโตรบแต่อย่างใด

### ขา EA ( External Access Enable )

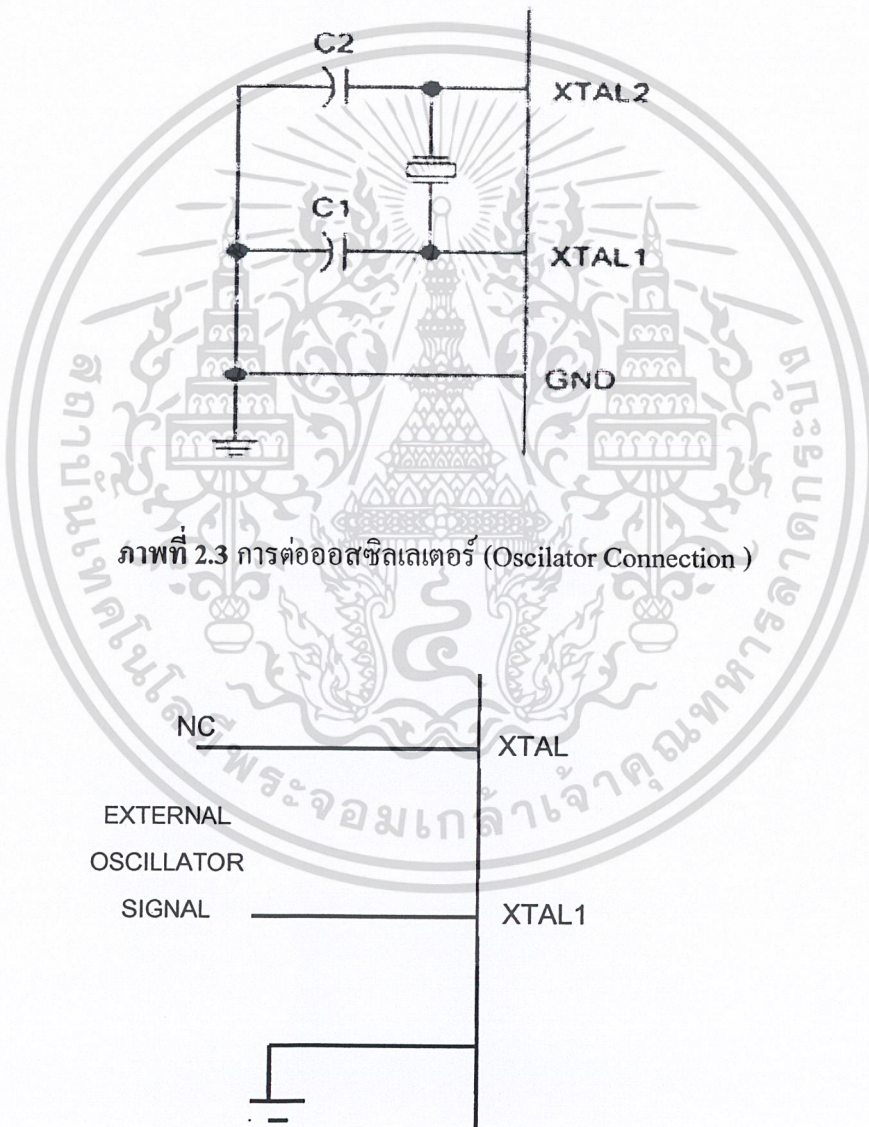
เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือจากภายนอก โดยถ้ามี

สถานะเป็น 0 จะหมายถึงให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0-0FFFH (0-1FFFFH ถ้าเป็นเบอร์ 8052 ) อย่างไม่จำกัดจำนวนคำสั่ง แต่ถ้าเป็น 1 จะใช้โปรแกรมที่เก็บไว้ในหน่วยความจำภายใน ( Security Bit ) ใน

หน่วยความจำ EEPROM ถูกโปรแกรมไว้ ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอกเลย นอกจากนี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (Vpp) ขนาด 12 V เพื่อใช้ในระหว่างโปรแกรม EEPROM

#### ขา XTAL1 และขา XTAL2

XTAL1 และ XTAL2 เป็นอินพุตและเอาต์ ของวงจรถ่ายกลับเฟส (Phase ) ซึ่งสามารถแสดงการต่อใช้งานบนชิพได้ดังรูป ซึ่งอาจจะใช้ผลึกควอตซ์ ( Crystal ) หรือ Ceramic Resonator เพื่อสร้างสัญญาณนาฬิกาจากภายนอกขับให้วงจรทำงาน



ภาพที่ 2.3 การต่อออสซิลเลเตอร์ (Oscillator Connection )

#### ภาพที่ 2.4 การต่อแบบ External Clock Drive Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

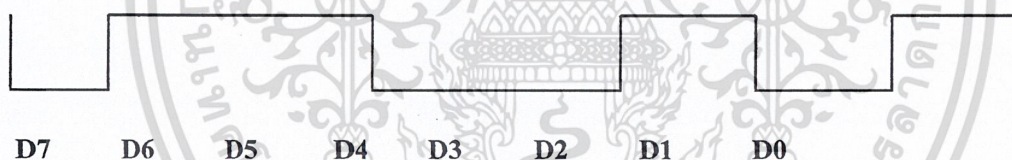
## 2.2 การใช้งานพอร์ตอนุกรม ( Serial Port )

การสื่อสารแบบอนุกรมเป็นการส่งข้อมูลครั้งละบิตแบบต่อเนื่องกันไปโดยส่งบิตค่าออกไปก่อนแล้วตามด้วยบิตสูง ซึ่งแตกต่างกับการส่งข้อมูลแบบขนานที่ส่งข้อมูลทุกบิตออกไปพร้อมกัน ตัวอย่างของการส่งข้อมูลแบบขนาน เช่นการส่งข้อมูลจากซีพียูออกไปยังหน่วยความจำ การส่งข้อมูลแบบขนานจะใช้สายสัญญาณ 1 เส้นต่อ 1 บิตแต่การส่งข้อมูลแบบอนุกรมจะใช้สายข้อมูลเพียงเส้นเดียวสำหรับข้อมูลทุกบิต

การส่งข้อมูลแบบอนุกรมมีโครงสร้างที่สำคัญหลายอย่างเช่นความเร็วในการส่งข้อมูลซึ่งเรียกว่าบอรรตเรต ( Baud Rate ) มีหน่วยเป็น บิต/วินาที เช่น ความเร็ว 1200,2400,4800, หรือ 9600 เป็นต้นรูปแบบของสัญญาณการรับส่งข้อมูลแบบอนุกรมเป็นดังรูป ข้อมูล 8 บิตที่ต้องการส่ง

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	0	1	1	0

สัญญาณที่ส่งออก



รูปแบบของสัญญาณการส่งข้อมูลอนุกรม

ภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีพอร์ตการสื่อสารแบบอนุกรม สามารถรับและส่งข้อมูลแบบ Full Duplex อยู่ 1 พอร์ต การรับส่งข้อมูลแบบ Full Duplex หมายถึงว่าพอร์ตอนุกรมสามารถรับและส่งข้อมูลได้ภายในเวลาเดียวกัน การควบคุมการทำงานของพอร์ตอนุกรมทางด้านส่งและรับข้อมูล ทำโดยการกำหนดค่าให้กับรีจิสเตอร์ (Register) ควบคุมการทำงานของพอร์ตอนุกรมชื่อ SCON

รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม SCON

รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม SCON เป็นรีจิสเตอร์เฉพาะที่ทำหน้าที่ควบคุมโหมดการทำงานของพอร์ตอนุกรม และเป็นที่ยึดข้อมูลของบิตที่ 9 ของการรับส่งข้อมูล ( บิต TB 8 และ RB8 ) และมีแฟล็ก (Flag) ของการร้องขออินเตอร์รัพของพอร์ตอนุกรมรวมกันอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 เอกสารนี้  
 ด้วย บิตต่างๆแสดงในรูป การควบคุมการทำงานเราจะกำหนดบิตต่างๆ ในรีจิสเตอร์ด้วยคำสั่งการ  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โอนย้ายข้อมูลหรือใช้คำสั่งการเซตหรือเคลียบิตก็ได้เนื่องจากรีจิสเตอร์ SCON อ้างอิงตำแหน่งแบบบิตได้

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 1
SM 0	SM 1	SM 2	REN	TB8	RB8	TI	RI

### SCON: Serial Port Control Register

ภาพที่ 2.5 รูปบิตต่างๆของรีจิสเตอร์ SCON

SM0และ SM1 เป็นบิตกำหนดโหมดการทำงานของพอร์ตอนุกรมซึ่งมี 4 โหมด

SM เป็นบิตควบคุมให้ทำงานในลักษณะของการเชื่อมต่อไมโครคอนโทรลเลอร์หลายตัวเข้าด้วยกัน REN เซตหรือรีเซตด้วยซอฟต์แวร์ (Software) เป็นตัวควบคุมการรับข้อมูลของพอร์ตอนุกรมดังนี้

1= ให้มีการรับข้อมูล

0= ไม่ให้มีการรับข้อมูล

TB8:เป็นบิตข้อมูลบิตที่ 9 ที่ต้องการส่งในโหมด 2 และ 3 สำหรับการทำงานในโหมด 1 หากกำหนดให้บิต SM2=0 บิต RB8 จะเป็นค่าของ STOP BIT ที่รับเข้ามาสำหรับโหมด 0 ไม่มีการใช้ RB8

TI:แฟลกของการอินเตอร์รัปต์ด้านส่งข้อมูล แฟลกนี้จะถูกเซตด้วยฮาร์ดแวร์เมื่อจบการส่งข้อมูลบิตที่ 8 ในโหมด 0 หรือเมื่อเริ่มต้นส่ง STOP BIT ในโหมด 1, 2, หรือ 3 เราต้องเคลียร์แฟลกนี้ด้วยซอฟต์แวร์ เมื่อจบโปรแกรมตอบสนองการอินเตอร์รัปต์ของการส่งข้อมูลแล้ว

RI:แฟลกการอินเตอร์รัปต์ด้านรับข้อมูล ถูกเซตด้วยฮาร์ดแวร์เมื่อข้อมูลบิตที่ 8 ในโหมด 0 ถูกรับเข้ามาหรือเมื่อ STOP BIT ถูกรับเข้ามาในครั้งแรกในโหมด 1, 2 หรือ 3 เราต้องเคลียร์แฟลกนี้ด้วย ซอฟต์แวร์ เมื่อจบโปรแกรมการตอบสนองการอินเตอร์รัปต์ของการส่งข้อมูลแล้ว

การส่งและรับข้อมูลของพอร์ตอนุกรมจะมีรีจิสเตอร์ที่ทำหน้าที่รับและส่งข้อมูลอยู่ 1 ตัวคือ รีจิสเตอร์ SBUF การส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ทำได้โดยการใส่ข้อมูลลงในรีจิสเตอร์ SBUF การอ่านข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะอ่านจากรีจิสเตอร์ SBUF เช่นกัน วงจรด้านรับจะมีบัฟเฟอร์ขนาด 1 ไบต์ที่ทำหน้าที่เก็บข้อมูลที่รับเข้ามาประกอบอยู่ภายใน (การมีบัฟเฟอร์รับข้อมูลทำให้ด้านรับสามารถรับข้อมูลไบต์ที่ 2 เข้ามาได้ทันทีหลังจากข้อมูลไบต์แรกเข้ามาแล้วแม้ยังไม่ถูกอ่านออกไปแต่ถ้าข้อมูลแรกยังไม่ถูกอ่านก่อนที่ข้อมูลไบต์ที่ 2 จะเข้ามาครบข้อมูลไบต์ที่ 2 จะถูกยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การเลือกโหมดการทำงานของพอร์ตอนุกรม สามารถกำหนดโหมดต่างๆ

MODE	รีจิสเตอร์ SCON	บิต SM 2 Variation
0	10H	SINGLE PROCESSOR ENVIRONMENT ( SM2=0)
1	50H	
2	90H	
3	D0H	
0	NA	MULTIPROCESSOR ENVIRONMENT (SM2=1)
1	70H	
2	B0H	
3	F0	

การทำงานของพอร์ตอนุกรมไมโครคอนโทรลเลอร์สามารถแบ่งออกได้เป็น 4 โหมดคือ 0,1,2 และ 3 แต่ละโหมดมีการทำงานต่อไปนี้

#### โหมด 0

ข้อมูลขนาด 8 บิตแบบอนุกรมรับเข้ามาทางขา RxD และข้อมูล 8 บิตส่งออกมาแบบอนุกรมทางขา TxD การรับและการส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อน อัตราการรับส่งข้อมูล (Baud Rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา

#### โหมด 1

ใช้การรับและส่งข้อมูลขนาด 10 บิตซึ่งข้อมูลอนุกรม 10 บิตเข้ามาทางขา RxD และข้อมูลขนาด 10 บิตส่งออกมาแบบอนุกรมทางขา TxD โดยข้อมูล 10 บิตประกอบด้วย 1 Start Bit ที่รับเข้ามาไปเก็บในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON อัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ

#### โหมด 2

ใช้การรับและส่งข้อมูลขนาด 11 บิต ข้อมูลแบบอนุกรมรับเข้ามาทางขา RxD และส่งออกไปทางขา TxD ซึ่งข้อมูลขนาด 11 บิตประกอบด้วย 1 Start Bit (ค่า 0), 8 บิตข้อมูล (การรับ/ส่งเริ่มจากบิตต่ำก่อน), 1 บิตข้อมูลที่กำหนดค่าได้ และ 1 Stop Bit (ค่า 1) สำหรับด้านส่งข้อมูลบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งเราสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ ในการใช้งานเราอาจใช้บิตข้อมูลที่ 9 เป็นบิตตรวจสอบก็ได้ โดยการนำค่าแฟล็ก P มากำหนดให้กับ TB8 เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล 8 ออกจากแอสคิวิมูเดเตอร์ซึ่งจะทำให้เราได้รับการตรวจสอบการส่งข้อมูลแบบพาริตีคู่ (Even Parity) ในกรณีของการรับข้อมูลจะนำข้อมูลบิตที่ 9 ที่รับเข้ามาไปเก็บในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON ส่วน Stop Bit จะไม่มีการนำมาเก็บ อัตราการรับและส่งข้อมูลในโหมดนี้จะเลือกใช้ความเร็วได้ 2 ค่าคือ 1/32 หรือ 1/64 เท่าของสัญญาณนาฬิกา

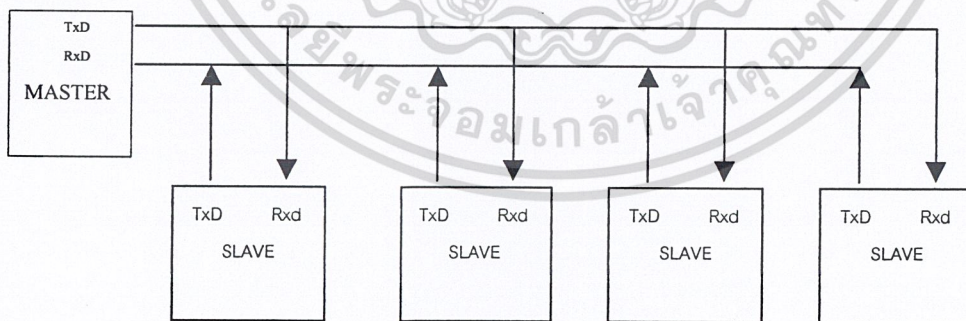
**โหมด 3**

การทำงานในโหมด 3 จะมีลักษณะคล้ายกับโหมด 2 แต่ในโหมด 3 เราสามารถกำหนดอัตราการรับและส่งข้อมูลได้ตามต้องการ

การทำงานของพอร์ตอนุกรมทั้ง 4 โหมด การส่งข้อมูลจะเริ่มต้นเมื่อมีการกำหนดข้อมูลให้กับรีจิสเตอร์ SBUF ซึ่งการกำหนดค่าให้กับรีจิสเตอร์ SBUF จะใช้คำสั่งการโอนย้ายข้อมูลเช่น MOV SBUF,#15H หรือ MOV SBUF,@R1 ก็ได้ การรับข้อมูลในโหมด 0 จะเริ่มต้นรับข้อมูลเมื่อค่าของบิต RI=0 และ REN=1 ส่วนในโหมดอื่นๆการรับข้อมูลจะเริ่มต้นเมื่อกำหนดบิต REN=1 และมี Start Bit เข้ามาที่ขา RxD

**2.3 การสื่อสารระหว่างซีพียูหลายตัว (Multi Processor Communication)**

การติดต่อสื่อสารทั่วไปจะใช้การติดต่อสื่อสารแบบจุดต่อจุดซึ่งมีตัวส่ง 1 ตัวและตัวรับ 1 ตัว แต่ในบางครั้งการควบคุม จำเป็นต้องทำการสื่อสารแบบหลายจุด ซึ่งเป็นการสื่อสารระหว่างไมโครคอนโทรลเลอร์หลายตัวที่เชื่อมต่อเข้าด้วยกันเพื่อให้เกิดการควบคุมที่มีประสิทธิภาพเพิ่มขึ้น ไมโครคอนโทรลเลอร์เบอร์ 8051 สามารถเชื่อมต่อเพื่อรับส่งข้อมูลกันได้หลายตัว ดังแสดงในภาพที่ 2.6



**ภาพที่ 2.6** การเชื่อมต่อไมโครคอนโทรลเลอร์หลายๆตัว

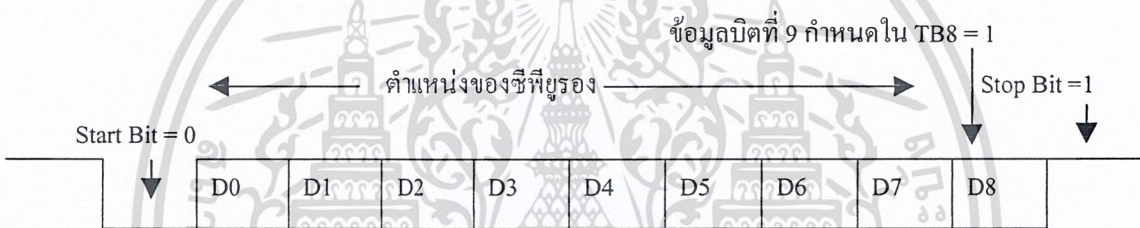
การทำงานของพอร์ตอนุกรมของ ไมโครคอนโทรลเลอร์ MCS-51 ในโหมด 2 และ 3 มีการทำงานพิเศษที่สามารถใช้กับการเชื่อมต่อสื่อสารระหว่างซีพียูหลายตัวได้การทำงานทั้ง 2 โหมดนี้ ข้อมูลบิตที่ 9 จะถูกนำไปเก็บในบิต RB8 เราสามารถโปรแกรมเพื่อกำหนดให้พอร์ตอนุกรมส่ง

สัญญาณรบกวนอินเทอร์รัพต์ได้เมื่อรับข้อมูลเข้ามาแล้วได้ค่าใน RB8 = 1 ซึ่งการควบคุมนี้กำหนดจากการเซตบิต SM2 ที่อยู่ในรีจิสเตอร์ SCON การทำงานในการเชื่อมต่อสื่อสารกับซีพียู(CPU) หลายตัว จะมีลักษณะการทำงานดังนี้

ในระบบที่เชื่อมต่อแบบจุดหลายจุดเรากำหนดให้ซีพียูตัวหนึ่งทำงานเป็นซีพียูหลัก(Master CPU) สำหรับการควบคุมการติดต่อกับซีพียูตัวอื่น ๆ ซึ่งเรียกว่าเป็นซีพียูรอง (Slave CPU) สัญญาณ TxD ของซีพียูหลักต่อเข้ากับสัญญาณ RxD ของซีพียูรองทุกตัว ทำให้ข้อมูลจากซีพียูหลักสามารถส่งไปซีพียูรองได้ทุกตัว ที่ตัวซีพียูรองจะมีตำแหน่งของตัวเองที่กำหนดไว้จากโปรแกรมหรือฮาร์ดแวร์(Hardware)โดยมีการแบ่งลักษณะของข้อมูลที่ใช้ส่งออกเป็น 2 แบบคือ

#### 1. ข้อมูลที่ใช้เป็นการกำหนดตำแหน่ง (Address Byte)

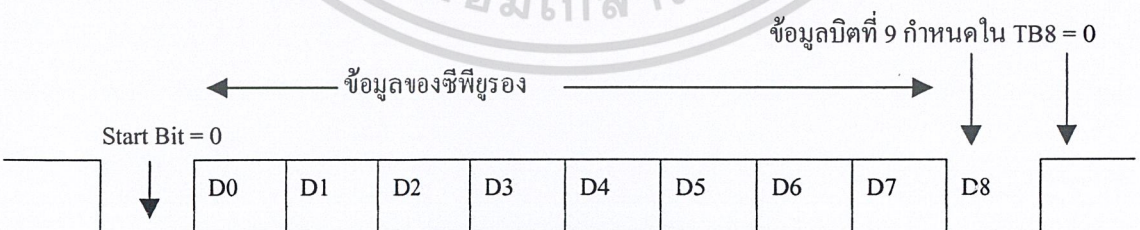
ข้อมูลที่ใช้เป็นการกำหนดตำแหน่งจะมีค่าบิตที่ 9 (DB) มีค่าเป็น 1 ใช้สำหรับกำหนดตำแหน่งของซีพียูรองที่ซีพียูหลักต้องการติดต่อกับ ลักษณะของข้อมูลที่กำหนดตำแหน่งแสดงในภาพที่ 2.7



ภาพที่ 2.7 ข้อมูลกำหนดตำแหน่ง (Address Byte)

#### 1. ข้อมูลที่ใช้เป็นข้อมูลข่าวสาร (Data Byte)

เป็นไบต์ที่ใช้แทนข้อมูลแทนการรับและส่งข้อมูล และนำไปใช้งาน โดยข้อมูลไบต์นี้จะกำหนดให้บิตที่ 9 เป็น 0 ซึ่งรูปแบบของไบต์ข้อมูลจะเป็นดังภาพที่ 2.8



ภาพที่ 2.8 ข้อมูลข่าวสาร(Data Byte)

เมื่อซีพียูหลักต้องการส่งข้อมูล 1 บล็อก(Block) ไปให้กับซีพียูตัวรองตัวใดตัวหนึ่งในหลาย ๆ ตัว ขั้นแรกซีพียูหลักต้องส่งค่า 1 ไบต์ ซึ่งถือเป็นตัวบอกตำแหน่งซีพียูรอง(Address Byte) ออกไปก่อนเพื่อเป็นการเลือกซีพียูรองที่ต้องการให้รับข้อมูล โดยข้อมูล 1 ไบต์นี้จะแตกต่างกับข้อมูลอื่น ๆ

คือบิตที่ 9 มีค่าเป็น 1 ข้อมูล ไบต์นี้ถูกส่งออกไปยังซีพียูของทุกตัว หากเรากำหนดให้ซีพียูของทุกตัว มีค่าบิต SM2=1 เมื่อซีพียูของทุกตัวได้รับข้อมูลไบต์ตำแหน่งแล้วจะเกิดการอินเตอร์รัพต์ขึ้นภายใน ตัวซีพียูของ ซีพียูจะทำการตรวจสอบว่าตำแหน่งที่รับมานั้นมีค่าตรงกับตัวเองหรือไม่ ซีพียูของ ที่มีตำแหน่งกับตำแหน่งที่รับได้จะทำการเคลียบิต SM2 และเตรียมรับข้อมูลที่เป็นข่าวสารต่อไป สำหรับซีพียูที่มีตำแหน่งไม่ตรงกับตำแหน่งที่รับมาก็จะยังคงค่า SM2=1 อยู่ต่อไป และจบโปรแกรม ตอนสนองการอินเตอร์รัพต์แล้วกลับไปทำโปรแกรมที่ค้างอยู่ และรอการอินเตอร์รัพต์ที่จะเกิดขึ้น จากการรับไบต์อีก หลังจากซีพียูหลักส่งข้อความไบต์แรกซึ่งเป็นตำแหน่งออกไปแล้วจะเริ่มส่ง ข้อมูลซึ่งถือเป็นไบต์ข้อมูลที่มีบิตที่ 9 เป็น 0 ตามออกไป ข้อมูลนี้จะมีเพียงซีพียูที่มีตำแหน่งตรงกัน เท่านั้นที่รับข้อมูลได้ การส่งข้อมูลก็จะเป็นระหว่างซีพียูหลักกับซีพียูของที่มีตำแหน่งตรงกันเท่านั้น เมื่อการส่งซีพียูหลักสิ้นสุดลงซีพียูหลักจะต้องส่งข้อมูลไบต์สุดท้ายที่เป็นตัวบอกให้ด้านรับทราบว่ าลสิ้นสุดข้อมูล ซึ่งอาจใช้รหัส ASCII ค่า 03 (ETX End of text) หรือ 04 (EOT End of transmission) ก็ได้ เมื่อซีพียูของได้รับรหัสนี้แล้วก็จะทราบว่าเป็นข้อมูลตัวสุดท้ายและจะหยุดการรับข้อมูล แล้ว เซตค่าบิต SM2 = 1 เพื่อเริ่มต้นรอรับการส่งข้อมูลที่เป็นตำแหน่งในรอบต่อไป

บิต SM2 จะไม่มีผลในโหมด 0 สำหรับการใช้งานในโหมด 1 หาก SM2 = การอินเตอร์รัพต์ ของการรับข้อมูลจะไม่เกิดขึ้นหากไม่มี Stop Bit เข้ามา

เนื่องจากในโครงการชิ้นนี้ได้มีการติดต่อสื่อสารระหว่างซีพียูหลายตัว ฉะนั้นจึงจำเป็นการ สื่อสารแบบอนุกรมซึ่งจะใช้โหมดการทำงานในโหมด 3 และในการสื่อสารแบบอนุกรมนี้เพื่อให้ สามารถส่งข้อมูลได้ในระยะไกลจำเป็นต้องใช้มาตรฐานการส่งแบบ RS - 485 โดยการนำชิพไอซี เบอร์ DS75176 ซึ่งเป็นไอซีที่ทำงานโดยเปรียบเทียบแรงดันระหว่างสายที่ใช้ในการส่งข้อมูล 2 เส้น ถ้าสายที่ใช้ในการส่งเมื่อเปรียบเทียบแล้วมีค่ามากกว่า +0.2 V เอาท์พุทที่ได้จะมีค่าเป็นลอจิก 0 ด้วย เหตุนี้เองทำให้ชิพ ไอซีเบอร์ DS75176 สามารถที่จะส่งได้ในระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

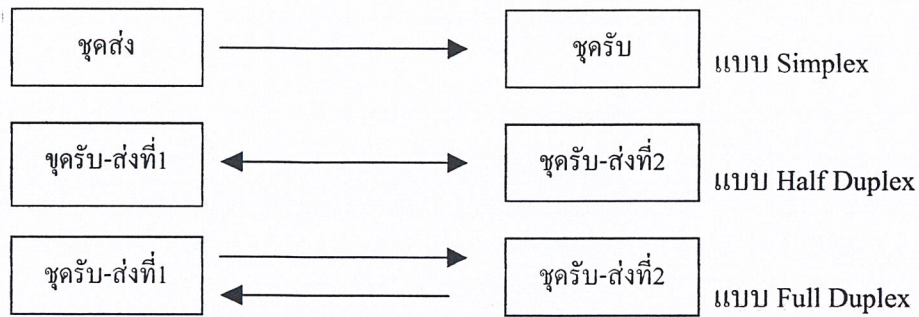
## 2.4 การสื่อสารแบบอนุกรม

ถ้ากล่าวถึงการสื่อสารแบบอนุกรมแล้วแสดงว่าจำเป็นต้องมีการสื่อสารข้อมูลแบบขนานด้วย การสื่อสารข้อมูลแบบขนานก็คือข้อมูลทุก ๆ บิตในแต่ละเวิร์ดจะถูกส่งออกไปพร้อม ๆ กันขึ้นอยู่กับเวิร์ดดังกล่าวว่ามีขนาดเท่าไร โดยทั่วไปก็คือ 1 ไบต์ หรือ 8 บิต นั่นเอง การส่งข้อมูลแบบขนานนี้จะมีข้อจำกัดทางด้านระยะทาง ด้านระยะทางโดยทั่วไปจะส่งได้ระยะไม่เกิน 3-5 ฟุต เท่านั้น ทั้งนี้ขึ้นอยู่กับอัตราความเร็วที่ใช้ในการส่งข้อมูลด้วย ยิ่งอัตราการส่งสูงก็จะได้ระยะทางที่สั้นลง การส่งข้อมูลแบบขนานนี้นิยมในระบบที่ต้องการความเร็วสูงมาก ๆ แต่อุปกรณ์อยู่ห่างกันไม่มากนัก ส่วนการส่งข้อมูลแบบอนุกรมนั้นข้อมูลจะถูกทยอยส่งออกไปทีละบิตไปครบจนครบทั้งเวิร์ดในสายสัญญาณเพียงเส้นเดียว แต่ในการใช้งานจริงจะต้องมีสายสัญญาณอีกเส้นเป็นสายสัญญาณกราวด์ (Ground) ดังนั้นเมื่อเราส่งข้อมูลในแบบอนุกรมเราจะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้น ในขณะที่การส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อยเท่ากับจำนวนบิตบวกกับสายสัญญาณระดับแรงดัน Ground อีก 1 เส้น ที่สำคัญในการส่งข้อมูลแบบอนุกรมนั้นจะสามารถส่งข้อมูลได้ไกลกว่ามากเช่น ถ้าส่งตามมาตรฐานของ RS -232 ที่จะกล่าวต่อไปในภายหลัง จะสามารถส่งไปได้ไกลถึง 30 ถึง 40 ฟุต โดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติมแต่อย่างใด อย่างไรก็ตามในการส่งข้อมูลยังมีข้อกำหนดบางประการเพื่อให้ได้รับข้อมูลที่ถูกต้อง แม่นยำ และมีความน่าเชื่อถือสูง จะต้องมีการตรวจสอบความถูกต้องของข้อมูลได้ในระหว่างที่มีการสื่อสารกันอยู่ด้วย ซึ่งจะได้กล่าวต่อไป

### 2.4.1 การส่งข้อมูลแบบซิมเพล็กซ์ (Simplex) และดูเพล็กซ์ (Duplex)

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูลหรือการสื่อสารทั่วไปนั้นย่อมจะต้องประกอบด้วยผู้รับและผู้ส่ง ผู้รับในขณะนี้อาจเป็นผู้ส่งในอนาคตได้ แต่มีบางกรณีที่เป็นผู้รับและผู้ส่งแน่นอนตายตัวอยู่ตลอดเวลา เช่น การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น การสื่อสารของอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น เราเรียกว่าการสื่อสารแบบ ซิมเพล็กซ์ กล่าวคือ การสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลา ซึ่งจะมีที่ใช้ไม่มากนัก การสื่อสารโดยทั่วไปนั้นจะเป็น แบบดูเพล็กซ์ คือมีทิศทางในการสื่อสาร 2 ทิศทางทั้งไปและกลับ การสื่อสารในลักษณะดูเพล็กซ์ นี้ยังแบ่งออกได้เป็น 2 ชนิด คือ แบบ ฮาร์ฟดูเพล็กซ์ (Half Duplex) นิยมเขียนย่อว่า HDX ซึ่งจะมีทิศทางในการสื่อสารในลักษณะที่ ผลัดกันเป็นผู้ส่งและผู้รับพร้อมกันไป และแบบ ฟูลดูเพล็กซ์ (Full Duplex) นิยมเขียนย่อว่า FDX จะมีทิศทางในการสื่อสารในลักษณะสัญญาณรับทิศทางหนึ่ง สัญญาณส่งอีกทิศทางหนึ่ง หรือกล่าวได้อีกนัยหนึ่งว่าสัญญาณรับและส่งจะมีสายตัวนำสัญญาณแยกกันจากกันโดยเด็ดขาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.9 แสดงการสื่อสารข้อมูลอนุกรมแบบต่าง ๆ

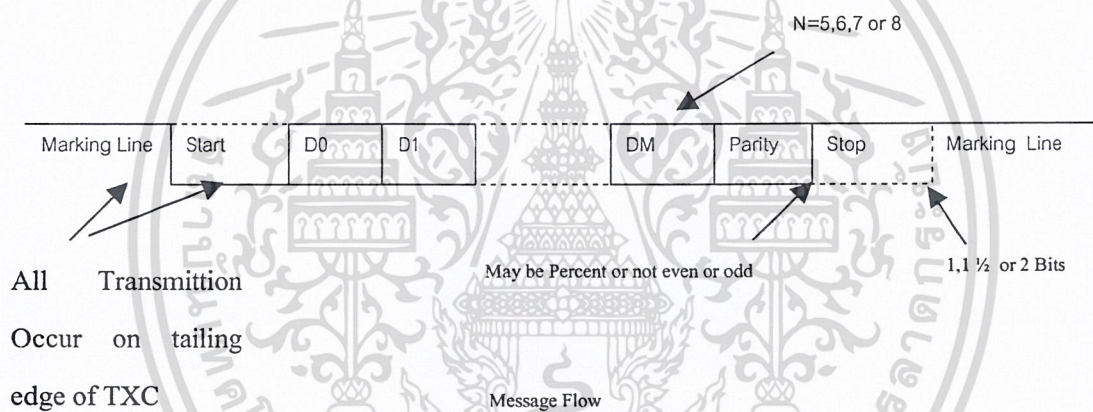
#### 2.4.2 โพรโทคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลในแบบอนุกรมให้คิดจะพบว่า ปัญหาหนึ่งที่จะเกิดขึ้นอยู่เสมอ ก็คือการตัดสินใจว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้นจึงมีการกำหนดข้อตกลงในการสื่อสารเพื่อแก้ปัญหานี้ ข้อตกลงดังกล่าวนี้เรียกว่า โพรโทคอล (Protocol) ของการสื่อสารข้อมูลแบบอนุกรม สามารถแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ โพรโทคอลสำหรับการสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) การสื่อสารแบบซิงโครนัสนั้นข้อมูลจะถูกส่งออกไปอย่างสม่ำเสมอ ช่วงเวลาระหว่างบิตและระหว่างเวิร์ดจะมีค่าเท่ากันเสมอ ดังนั้นการสื่อสารอนุกรมในแบบซิงโครนัสจึงต้องมีสายสัญญาณเพิ่มเติมเพื่อกำกับการส่งว่าควรจะส่งเมื่อใด และหยุดเมื่อใด ระบบที่เป็นซิงโครนัส จะเป็นระบบที่มีความเร็วสูง แต่ก็ยังต่ำกว่าการสื่อสารแบบขนาน

การสื่อสารแบบอะซิงโครนัสเป็นหัวใจการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ ในปัจจุบันการสื่อสารแบบนี้ช่วงระยะเวลาว่างบิต จะมีค่าเท่ากันเช่นเดียวกับซิงโครนัส แต่จะมีระยะห่างระหว่างเวิร์ดนั้นแตกต่างกันออกไปเป็นกี่วินาที นาที ชั่วโมง หรือวันเป็นต้น ได้ทั้งสิ้นขึ้นอยู่กับทางฝ่ายรับสามารถรอได้หรือไม่เท่านั้น เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาว่างเวิร์ดแล้ว ทางผู้ส่งและผู้รับจะเข้าใจตรงกันได้อย่างไรว่าที่ใดคือจุดเริ่มต้นและจุดสิ้นสุดของแต่ละเวิร์ด เพื่อแก้ปัญหาหนึ่งจึงมีการกำหนดข้อตกลงเกี่ยวกับรูปแบบของข้อมูลที่จะส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดเป็น จุดเริ่มต้นของเวิร์ด ข้อกำหนดดังกล่าวกำหนดให้แต่ละเวิร์ดจะต้องขึ้นต้นด้วยบิตที่เรียกว่า บิตเริ่มต้น (Start Bit) ซึ่งจะต้องมีข้อมูลเป็นลอจิก 0 เสมอ จากนั้นตามด้วยบิตข้อมูลที่ต้องการส่งซึ่งมีความยาว 5 ถึง 8 บิต ถัดจากบิตข้อมูลก็จะเป็นพริตบิต ซึ่งจะทำหน้าที่เป็นบิตสำหรับตรวจสอบความถูกต้องของข้อมูลที่ได้รับว่ามีความถูกต้องหรือไม่ บิตพริตมี 2 ประเภท คือ อีเวนพริตติ (Even Parity) ซึ่งจะกำหนดจำนวนบิตที่เป็นลอจิก 1 ในบิตที่เป็นข้อมูลมีจำนวนเป็นจำนวนคู่ และ

ออกจันวนนี้คือ ในการส่งข้อมูลบางครั้งจะไม่มีการใช้บิตพริตก็ได้ถ้าหากการสื่อสารในครั้งนั้นมีความการค้ำ  
ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำเชือถือสูงคือ มีสัญญาณรบกวนต่ำเป็นการเพิ่มความเร็วในการสื่อสารได้ด้วย บิตสุดท้ายในรูปแบบก็คือ บิตสุดท้าย (Stop Bit) ทำหน้าที่บอกทางผู้รับว่าขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบเวิร์ดแล้วขอให้เตรียมชุดรับเวิร์ดต่อไป บิตสุดท้าย นี้ถูกกำหนดให้เป็นลอจิก 1 เสมอทั้งนี้เพื่อให้ระบบสามารถตรวจสอบบิตเริ่มต้นได้ บิตสุดท้ายนี้อาจมีความยาวเป็น 1 บิตหรือ 2 บิตก็ได้จากรูปแบบดังกล่าว จะเห็นว่าเรามีรูปแบบสำหรับการสื่อสารมากมายไปหมด เช่น 5E1 (5Data Bit, Even Parity, 1 Stop Bit) และ 8N1 (8Data Bit, No Parity, 1 Stop Bit) เป็นต้น ในการใช้งานทั่วไปปรานิยมใช้กันอยู่เพียง 2 รูปแบบคือ 7E1 และ 8N1 จะเลือกใช้รูปแบบใดขึ้นอยู่กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด ถ้าหากสายส่งมีสัญญาณรบกวนมากก็ควรจะใช้ 7E1 แต่สายส่งสัญญาณมีสภาพดีสัญญาณรบกวนต่ำการใช้ 8N1 จะเร็วกว่า เป็นต้น ทั้งนี้จะต้องมีการตกลงกันล่วงหน้าระหว่างผู้รับและผู้ส่งว่าจะใช้รูปแบบใดในการสื่อสาร ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูป

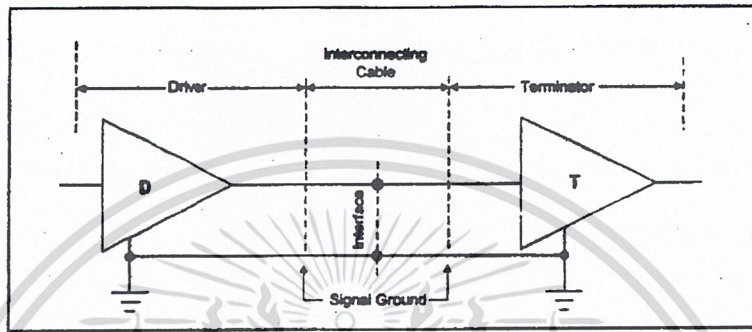


ภาพที่ 2.10 แสดงแบบมาตรฐานการสื่อสารข้อมูลและอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C

มาตรฐานการสื่อสารข้อมูลแบบอนุกรมที่กำหนดโดย EIA (Electronics Industries Association) มาตรฐาน RS232C ได้ถูกตีพิมพ์ในปี ค.ศ. 1969 ตัวอักษร RS แทน "Recomm Standard" แทนหมายเลขมาตรฐาน ส่วนอักษร C แสดงให้รู้ว่ามาตรฐานได้รับการแก้ไขที่ครั้ง



ภาพที่ 2.11 ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232C

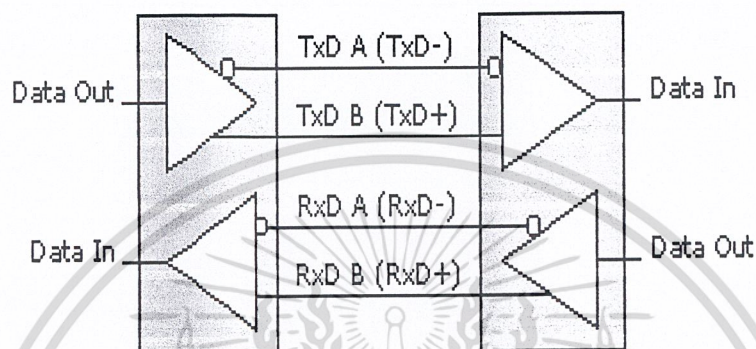
ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ Discrete

- ถูกออกแบบให้ใช้กับอุปกรณ์พวกลักษณะ Discrete
- ใช้การอินเทอร์เฟสแบบ Unbalanced
- ในแต่ละวงจรใช้ลวดนำในการนำสัญญาณ 1 เส้น และมีสายกราวด์รวมของทุกวงจรอีกหนึ่งเส้น
- อัตราเร็วในการส่งข้อมูลมีค่า < 20 กิโลบิตต่อวินาที (Kbps)
- ระยะทางสูงสุดที่ใช้ในการส่งข้อมูลมีค่า < 15 เมตร
- ทำให้เกิด Crosstalk ที่มีค่ามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.4.4 มาตรฐานสัญญาณอนุกรมแบบ RS422

มาตรฐาน RS422 เป็นชุดขับแบบขยายความแตกต่างในรูปของกระแส ทำให้อัตราเร็วในการส่งข้อมูลมีค่าสูงขึ้น และระยะทางที่ใช้ส่งข้อมูลระหว่างชุดส่งและชุดรับมีระยะไกลขึ้น เพราะเป็นการส่งชนิด พูลคูเพล็กซ์



ภาพที่ 2.12 แสดงลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟซแบบ RS422

ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟซแบบ RS422

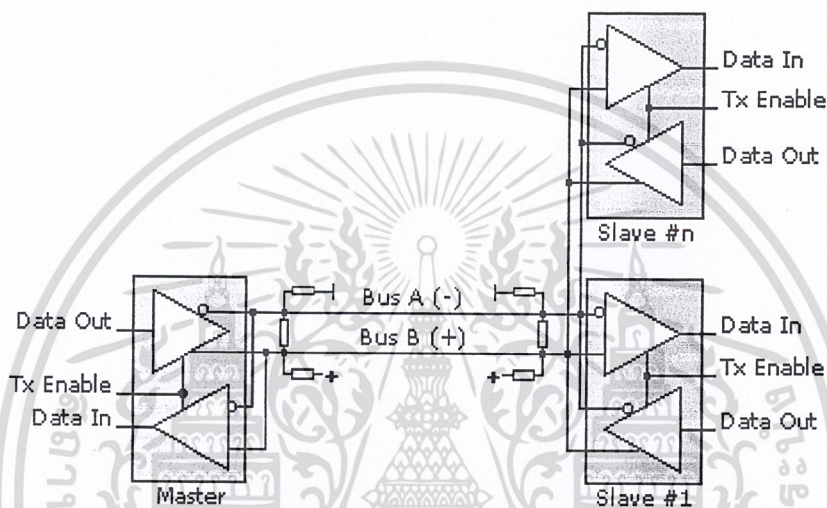
- ชุดส่งสัญญาณแบบ Balanced
- ชุดรับข้อมูลเป็นแบบขยายความแตกต่าง
- ในแต่ละวงจรใช้ลวดตัวนำในการส่งสัญญาณจำนวน 2 เส้น
- อัตราเร็วในการส่งข้อมูลสูงถึง 10 เมกกะบิตต่อวินาที (mbps)
- ระยะทางที่ใช้ในการส่งข้อมูลได้ไกลถึง 4000 ฟุต

มาตรฐาน RS422 ใช้การส่งข้อมูลในลักษณะของ One-Way Balanced-Line ซึ่งมีชุดส่งข้อมูลบนระบบอยู่ 1 ชุดและชุดรับข้อมูล 32 ชุด โดยอัตราเร็วในการส่งข้อมูลมีค่าสูงถึง 10 mbps ที่ระยะทางเท่ากับ 40 ฟุต ในกรณีที่ส่งข้อมูลในอัตราเร็วที่ต่ำกว่า 10 mbps ระยะทางที่ใช้ในการส่งข้อมูลสามารถขยายได้ถึง 4,000 ฟุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4.5 มาตรฐานสัญญาณอนุกรมแบบ RS 485

มาตรฐานสัญญาณอนุกรมแบบ RS485 นี้ก็พัฒนามาจาก RS422 คือผู้ผลิตบางบริษัทได้ทำวงจรสัญญาณ (ชุดส่ง) เป็นแบบ Tri-State ทำให้เราสามารถส่งข้อมูลได้สองทิศทางบนสายคู่เดียว (Single Pair) คุณสมบัติข้อนี้จึงทำให้ระบบส่งข้อมูลมีโครงสร้างเป็นแบบ Multidrop ซึ่งอุปกรณ์หลาย ๆ ประเภทสามารถรับและส่งข้อมูลแบบ ฮาร์ฟดูเพล็กซ์ บนสายคู่เดียวได้



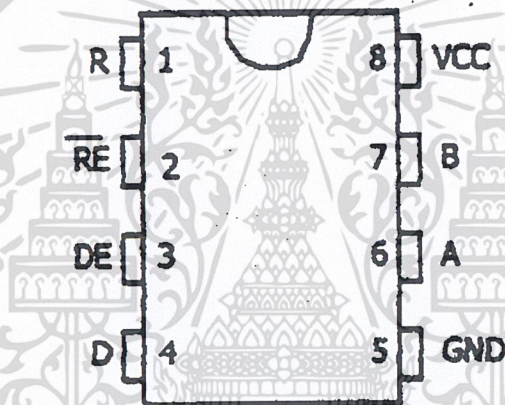
ภาพที่ 2.13 แสดงลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟซแบบ RS485

มาตรฐาน RS485 นี้ ทางบริษัท ผู้ผลิตได้ออกแบบชุดส่งและชุดรับให้สามารถต่อรวมอยู่บนบัสได้ถึงอย่างละ 32 แต่การส่งข้อมูลในขณะเวลาหนึ่ง ๆ นั้นจะส่งได้ทีละชุด ฉะนั้นจะต้องมีกรรมวิธีตรวจสอบบัสก่อนว่าว่าง และพร้อมที่จะให้ข้อมูลผ่านไปได้ ทั้งนี้เพื่อป้องกันการสับสนของข้อมูลนั่นเองการรับข้อมูลนั้นสามารถรับส่งได้ทั้งหมดรวมกันถึง 32 ชุด แต่ละชุดต้องมีการตรวจหัวข้อมูล (Header) หรือ Identify Code ก่อนเพื่อเป็นรหัสให้รับทราบที่ต้องการสื่อสารกับชุดใด ส่วนประกอบของมาตรฐาน RS485 ประกอบด้วย ชุดส่งรับ สายเคเบิล และ Resister Terminating ซึ่งค่าความต้านทาน RT มีค่าประมาณ 220 โอห์ม โดยทั่วไปจะต้องศึกษาข้อมูลจาก Data sheet ของ IC เบอร์นั้น ๆ ประกอบคู่ไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.6 การสื่อสารข้อมูลอนุกรมด้วย RS-485

การรับส่งข้อมูลด้วยมาตรฐาน RS-485 เป็นมาตรฐานที่ถูกพัฒนามาจาก RS-422A เพื่อให้ตัวรับและตัวส่งจำนวนมากคู่สามารถใช้คู่สายสัญญาณรับร่วมกันได้ซึ่งในกรณีของคู่สายสัญญาณรับส่งหนึ่งคู่จะมีตัวรับไม่เกิน 10 ชุด และจะมีตัวส่งเพียง 1 ชุดแต่ในกรณีของ RS-485 สามารถใช้ตัวรับ 32 ชุด และตัวส่ง 32 ตัว ร่วมกันได้ภายในคู่สายเดียวกัน โดยทั่วไป RS-485 มีคุณสมบัติเฉพาะทางไฟฟ้าของตัวรับและตัวส่ง และยังมีราคาไม่แพงทำให้ RS-485 ถูกนำมาใช้งานในระบบสื่อสารข้อมูลอนุกรมแบบโครงข่ายอย่างแพร่หลายสำหรับลักษณะขาอุปกรณ์เบอร์ SN75176 ถูกนำมาใช้งานในระบบสื่อสารข้อมูลแบบอนุกรมในระบบมาตรฐานของ RS-485 ซึ่งในตัวอุปกรณ์นี้ ได้มีตัวรับและตัวส่งอยู่ในชิปเดียวกันดังรูป



ภาพที่ 2.14 แสดงลักษณะขาของ SN75176

การใช้งานขาต่างๆของ SN75176

รายละเอียดของแต่ละขา

- ขา 1 RO ( Receiver Output ) เป็นขาที่จะนำข้อมูลด้านรับเข้าส่งให้ CPU
- ขา 2 /RE ( Receiver Enable ) เป็นขาที่เป็นตัวกำหนดการรับ ถ้าจะรับจะให้ '0'
- ขา 3 DE ( Driver Enable ) เป็นขาที่เป็นตัวกำหนดการส่ง ถ้าจะส่งต้องป้อน '1'
- ขา 4 DI ( Driver Input ) ขาที่จะนำข้อมูลจากชิปผู้ส่งออกไปภายนอก
- ขา 5 GND ต่อกราวด์
- ขา 6 A ทำหน้าที่ส่งและรับข้อมูลที่ติดต่อจากภายนอก
- ขา 7 B ทำหน้าที่ส่งและรับข้อมูลที่ติดต่อจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา 8 VCC ไฟเลี้ยงไอซี 5 โวลต์กลาดเคลื่อนได้ 5 %

การทำงานของขา RO นั้นมันจะเปรียบเทียบระหว่างอินพุต A และ B ถ้า A มากกว่า B มีค่า 200 mV หรือ 0.2 โวลต์มันจะให้ลอจิก '1' แต่ถ้า A น้อยกว่า B มีค่า 200 mV ขา RO จะให้เอาท์พุท เป็น '0'

การทำงานของขา /RE จะเป็นขาควบคุมการทำงานของขา RO อีกทีหนึ่งถ้าขา /RE นี้มีค่า '0' ขา RO จะทำหน้าที่รับข้อมูลจากคู่ A และ B เพื่อนำไปส่งออกให้ซีพียู แต่ถ้าขา /RE นี้มีค่า '1' ขา RO จะไม่มีข้อมูลที่ เป็น '0' หรือ '1' ออกมา

การทำงานของขา DI เราจะนำไปต่อกับขาที่ใช้ส่งข้อมูลเพื่อจะส่งข้อมูล

การทำงานของขา DE มีค่าเป็น '1' 75176 จะสามารถส่งผ่านข้อมูลผ่านออกทางขา A และ B เพื่อส่งข้อมูล แต่ถ้าขา DE มีค่าเป็น '0' จะไม่มีสัญญาณข้อมูลออกไปที่ขา A และ B

การทำงานของขา A รับอินพุทเป็น Noninverting และขับข้อมูลที่ เป็น Noninverting

การทำงานของขา B เป็นขาขับข้อมูลที่ เป็น Inverting

#### 2.4.7 มาตรฐานการสื่อสารข้อมูล EIA RS-485

RS-485 เป็นมาตรฐานการสื่อสารข้อมูลแบบสมดุล ที่ได้มีการพัฒนามาจาก มาตรฐาน RS-422 เพื่อที่จะให้ตัวรับและตัวส่งมีจำนวนมากคู่สายขึ้นสามารถใช้คู่สายสัญญาณรับและส่งรวมกัน ได้ (Multipoint Multiple Driver Receiver) ซึ่งในกรณีของ RS-422 มีคู่สาย 1 ชุด แต่ในกรณีของ RS-485 มีคุณลักษณะเฉพาะของสายไฟฟ้าตัวรับและตัวส่งคล้ายตัวรับของ RS-422 และไม่จำกัดรูปแบบข้อที่จะนำมาใช้งานที่ระบบจะพัฒนาขึ้น โดยขึ้นอยู่กับผู้พัฒนาว่าจะเลือกใช้ข้อมูลแบบไหน นอกจากตัวรับและตัวส่งที่มีราคาถูกทำให้ RS-485 ถูกนำมาประยุกต์ใช้งานอย่างแพร่หลายจากลำดับการจตุรวมของ โวลท์แดง (VCM Voltage Common Mode) ทำให้การส่งและการรับที่ทนแรงดันถึง +12 ถึง -7 โวลท์ ซึ่งการส่งอาจไม่เกิดการต่อเนื่องขึ้นได้จากสัญญาณ มันต้องทนแรงดันที่จตุรวม ในขณะที่เกิดสภาวะของ " Tristate " บางส่วนของการส่ง RS-422 ถ้าเกิดจำนวนคู่สาย "Tristate" อาจจะไม่เกิดการต่อต้านเมื่อแรงดันอยู่ในลักษณะที่สมบูรณ์คือ +12 ถึง -7 โวลท์

#### คุณสมบัติเฉพาะของคู่ตัวรับและตัวส่ง

คู่ตัวรับและตัวส่งเป็นอุปกรณ์ที่มีตัวรับและตัวส่งอยู่ในชิปเดียวกัน เพื่อให้เกิดความสะดวกในการใช้งาน และให้ระบบมีขนาดเล็ก และมีตัวรับส่งที่มีมาตรฐานเป็น RS-485,RS-422,CCITT และเอาท์พุทตัวส่งสามารถขับกระแสได้สูงถึง 60mA และก็มีระบบ Thermal Shutdown Protect และค่าความต้านทานน้อยที่สุดของอินพุทตัวรับ 20 กิโลโอห์ม และในการรับข้อมูลมีความไวบวก เอกสลบ 200mV คู่สายสัญญาณทั้งตัวรับและตัวส่งควรพันสลับเป็นเกลียวเพื่อลดทอนสัญญาณรบกวนที่ไม่อาจเกิดขึ้นในขณะที่มีการรับส่งข้อมูลที่มีระยะทางไกลๆ และทำการรับส่งข้อมูลที่มีตัวรับหลายตัว

ไดโอดเปล่งแสง (Light Emitting Diode) หรือ แอลอีดี (LED)

แอลอีดีมีการทำงานที่เหมือนไดโอดทั่วไปต่างกันที่แอลอีดีนั้นเมื่อกระแสไหลผ่านจะสามารถส่องแสงออกมาได้ และทนต่อการไบอัสกลับ (Reverse Bias) ได้น้อยกว่าไดโอดทั่วไป แอลอีดีเป็นตัวกำเนิดแสงทางอิเล็กทรอนิกส์ จะทำให้เกิดแสงเมื่อมีการไบอัสตรง (Forward Bias) โดยบริเวณรอยต่อของ p-n ซึ่งเป็นผลให้พลังงานอิเล็กทรอนิกส์เปลี่ยนแปลง และจะเปลี่ยนเป็นพลังงานออกมาเป็นรูปคลื่นแสง แสงแอลอีดีที่ให้ออกมานั้นจะมีหลายสีแตกต่างกันออกไปซึ่งขึ้นอยู่กับสารที่ใช้ทำและจะให้ความสว่างไม่เท่ากัน แต่ที่เรานิยมใช้กันคือสีแดง

อธิบายหลักการทำงานของแอลอีดี (LED) ได้คือเมื่อมีแรงดันไบอัสตรงด้วยไฟตรงค่าๆ อิเล็กตรอนจะไหลผ่านรอยต่อ n ไปยังรอยต่อ p และโฮลจะวิ่งไปยัง n และรวมกับพาหะข้างน้อยเป็นผลทำให้เกิดการรีคอมไบเนชัน จะปล่อยพลังงานออกมาเป็นแสงสว่าง และความร้อน ความถี่แสงที่ให้ออกมานั้นจะขึ้นอยู่กับแถบพลังงานช่องว่างของวัสดุที่ใช้ทำแอลอีดีนั้นซึ่งมีอยู่ 3 ชนิดคือ

1. แกลเลียมอาเซนไนด์ จะให้แสงอินฟราเรด และมีประสิทธิภาพในการเปลี่ยนพลังงานให้เป็นความถี่แสงอินฟราเรด ได้สูงมากถึง 71 เปอร์เซ็นต์ ประมาณ 1.2 ถึง 1.3
2. แกลเลียมฟอสไฟด์ นิยมใช้ทำแอลอีดีแสงสีเขียวและเหลือง
3. แกลเลียมอาร์เซไนด์ฟอสไฟด์ นิยมใช้ทำ แอลอีดีสีแดง

ดังที่กล่าวมาแล้วข้างต้นว่าแอลอีดีนั้นทนต่อแรงดันไบอัสกลับได้น้อยกว่าไดโอดธรรมดา ซึ่งจะทนได้ประมาณ 3-10 โวลต์ แต่ไดโอดธรรมดานั้นจะทนได้ประมาณ 5 โวลต์ขึ้นไป ซึ่งถ้าเราให้ไบอัสกลับเกินกว่าที่แอลอีดีทนได้และยอมให้กระแสไหลผ่านกลับทาง จะทำให้แอลอีดีเสียหายได้ ในการใช้งานทางดิจิตอลนั้นมักจะใช้ทรานซิสเตอร์เข้ามาช่วยขับแอลอีดี ก็ใช้ทรานซิสเตอร์ทำเป็นสวิตช์ในการเปิดปิดนั่นเอง

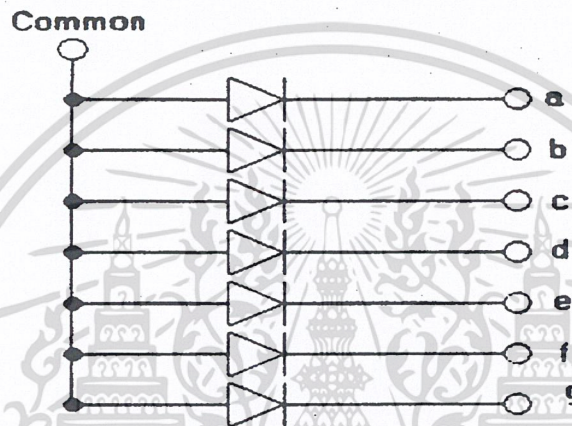
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 แผงแสดงผลแบบ 7 ส่วน (Seven Segment)

แผงแสดงผลแบบ 7 ส่วน คือการนำเอาแอลอีดี(Light Emitting Diode)มาต่อเป็นส่วนต่างๆ ของตัวแสดงผลแบบตัวเลขทั้ง 7 ส่วนนั้นจะแยกอิสระจากกัน คือสามารถเปิดหรือปิดส่วนใดก็ได้ ก็จะทำให้แอลอีดีแสดงตัวเลขได้ทุกตัว ตั้งแต่ 0-9 หรืออาจจะแสดงเป็นตัวอักษรอื่นๆ ก็ได้ขึ้นอยู่กับ การเปิดแต่ละส่วน แต่ในที่นี้เราจะกล่าวเฉพาะตัวเลขเท่านั้น

การนำแอลอีดีมาต่อเป็นแผงแสดงผลแบบ 7 ส่วนนั้น มีการต่ออยู่ 2 แบบคือ

1. แบบฮาโนคร่วม(Common Anode)ดังจะแสดงดังภาพที่ 2.15

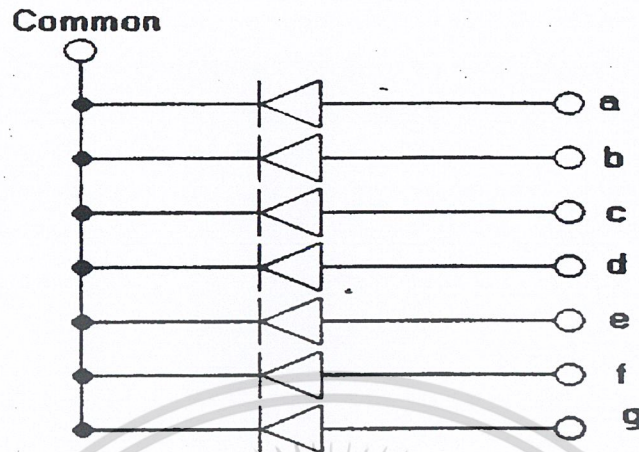


ภาพที่ 2.15 แสดงการต่อแบบ Common Anode

จากภาพที่ 2.15 แต่ละส่วนนั้นจะติดก็ต่อเมื่อ ป้อนไฟบวกประมาณ 3 โวลต์ เข้าที่ขาร่วม และขั้ว a b c d e f g นั้นต่อกับกราวด์(Ground) ก็จะสามารถบังคับให้ส่วนใดๆ ติดก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แบบคาโทดร่วม(Common Cathode) ดังจะแสดงดังภาพที่ 2.16



ภาพที่ 2.16 แสดงการต่อแบบ Common Cathode

จากภาพที่ 2.16 แต่ละส่วนจะติดได้เมื่อนำขาร่วมไปต่อกับกราวด์ และต่อขั้ว a b c d f g เข้ากับไฟเลี้ยงประมาณ 3 ถึง 5 โวลต์ จากการศึกษาสามารถบังคับให้ส่วนใดติดก็ได้ และเรานำเอาแอลอีดีมาจัดเรียงให้เป็นรูปร่างที่เหมาะสม ก็จะทำให้แผงแสดงผลเป็นตัวเลขออกมาได้ดังจะแสดงในภาพที่ 2.17



ภาพที่ 2.17 แสดงการจัดเรียงแอลอีดีเป็นแผงผล 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงการเข้ารหัสตามตัวเลขที่แสดง

ตัวเลข	h	g	f	e	d	c	b	a	HEX
0	H	L	H	H	H	H	H	H	B0
1	H	L	L	L	L	H	H	L	86
2	H	H	L	H	H	L	H	H	DB
3	H	H	L	L	H	H	H	H	CF
4	H	H	H	L	L	H	H	L	E6
5	H	H	H	L	H	H	L	H	ED
6	H	H	H	H	H	H	L	H	FD
7	H	L	L	L	L	H	H	H	87
8	H	H	H	H	H	H	H	H	FF
9	H	H	H	L	H	H	H	H	EF

จากภาพที่ 2.17 คงจะมองออกแล้วว่าจะทำให้แผงแสดงผล 7 ส่วนแสดงออกมาเป็นตัวเลข 0-9 ได้อย่างไร ต่อไปนี้จะกล่าวถึงแผงแสดงผล 7 ส่วนแบบที่ต่อใช้ขั้วคาโทดร่วม(Common Cathode)เท่านั้น เพื่อให้แต่ละส่วนติดเมื่อได้รับลอจิกสูง(LogicHI) และให้สอดคล้องกับการเขียนโปรแกรม ดังนั้นการป้อนไฟเข้าที่ใดเพื่อให้แผงแสดงผลแบบ 7 ส่วนติดเป็นตัวเลขต่างๆตามตารางที่ 2.1

จากตารางที่ 2.2 จะเห็นว่าที่ช่อง h นั้นให้แผงแสดงผล 7 ส่วน ไม่มีแสดงไว้เพื่อสะดวกในการกำหนดค่าออกพอร์ต และบิตนี้จะไม่ใช้งาน H หมายถึงลอจิกสูงหรือค่าไฟบวก(Logic HI) หมายถึงลอจิกต่ำ หรือกราวด์(Logic LOW) a เป็นบิตนัยสำคัญต่ำ(LSB) และ h เป็นบิตที่มีนัยสำคัญสูง(MSB)

## 2.6 พอร์ตเครื่องพิมพ์ หรือ พอร์ตขนาน (Centronics Port)

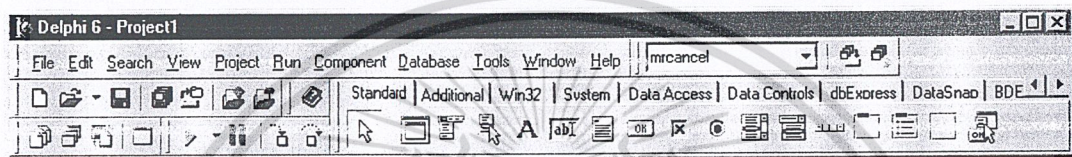
เนื่องจากเครื่องเอกซเรย์คอมพิวเตอร์โนมคิมมีการพิมพ์หมายเลข ดังนั้นจึงต้องใช้เครื่องพิมพ์เข้ามาเกี่ยวข้อง การที่จะพิมพ์ข้อความได้นั้นจะต้องมีการส่งแอสกี(ASCII) ทางพอร์ต พอร์ตขนานนั้นเป็นพอร์ตขนานที่สามารถรับส่งข้อมูล ได้คราวละ 8 บิต หรือ 1 ไบต์ โดยส่งผ่านสายสัญญาณ 8 เส้น แต่ส่วนใหญ่จะรู้จักกันว่าพอร์ตเครื่องพิมพ์ เพราะเราจะใช้ต่อกับเครื่องพิมพ์เป็นส่วนใหญ่ เอกสหรือ เรียกว่า พอร์ตเซ็นทรอนิกส์ (Centronic) คือเรียกตามชื่อบริษัทที่เป็นคนกำหนดมาตรฐานของการค้าไม่พอร์ตชนิดนี้ขึ้นมา อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 ส่วนประกอบการทำงานของโปรแกรม Delphi

การทำงานเพื่อสร้างแอปพลิเคชันกับ Delphi นั้นเราจะอยู่ภายใต้สภาวะแวดล้อมการพัฒนาแบบครบครัน นั่นคือ ประกอบด้วยเครื่องมือชนิดต่างๆที่นำมาสร้าง และทดสอบแอปพลิเคชัน

### Menu Bar

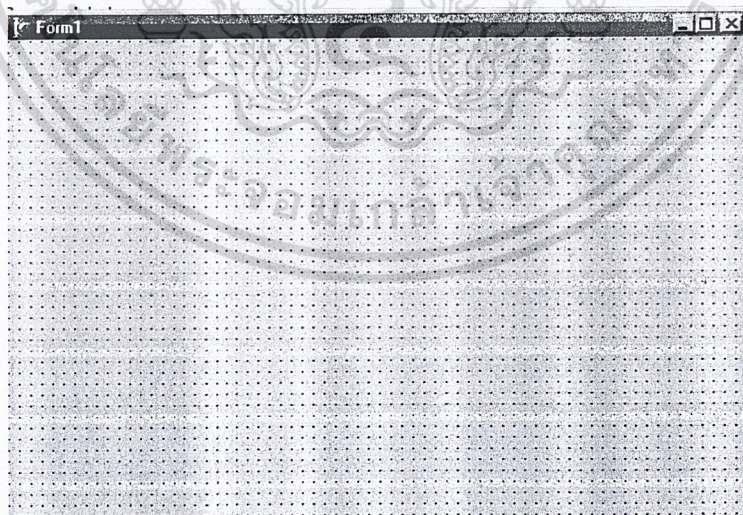
Menu Bar เป็นส่วนที่เก็บคำสั่งเพื่อเรียกใช้งาน Delphi ในรูปแบบเมนูซึ่งอยู่บนสุดของการแสดงผล



ภาพที่ 2.18 Menu Bar

### Form Designer

Form Designer เป็นส่วนที่เราออกแบบหน้าตาของฟอร์ม ซึ่งถือเป็นพื้นที่แสดงผลหลักที่เราสามารถนำเอาออบเจกต์ต่างๆ มาวางไว้ภายในให้เป็นแอปพลิเคชันตามที่เราต้องการได้

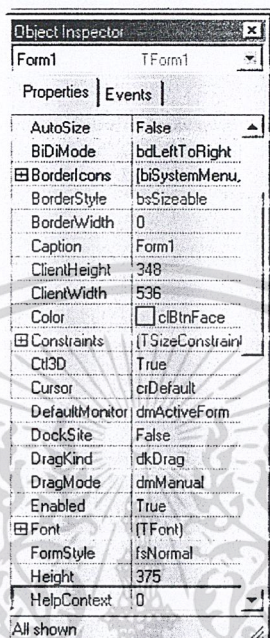


ภาพที่ 2.19 Form Designer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Object Inspector

เราสามารถกำหนดค่าพรีอเพอร์ตี้ของออบเจกต์ต่างๆ ผ่าน Object Inspector นอกจากนี้ เรายังเริ่มต้นจัดการกับอีเวนต์ต่างๆ ของออบเจกต์ผ่านทาง Object Inspector เช่นเดียวกัน



ภาพที่ 2.20 Object Inspector

## Component Palette

Object Inspector คือส่วนที่เก็บคอมโพเนนต์ต่างๆ แยกเป็นหมวดหมู่เอาไว้ พร้อมให้เรานำไปประกอบในแอปพลิเคชันที่สร้างขึ้นมา และไม่ใช่จะมีเพียงแค่คอมโพเนนต์ที่ Delphi เตรียมไว้ให้เท่านั้น เรายังสามารถสร้างคอมโพเนนต์ไว้ใช้งานเอง แล้วบรรจุไว้ใน Component Palette ได้ด้วย

## Code Editor

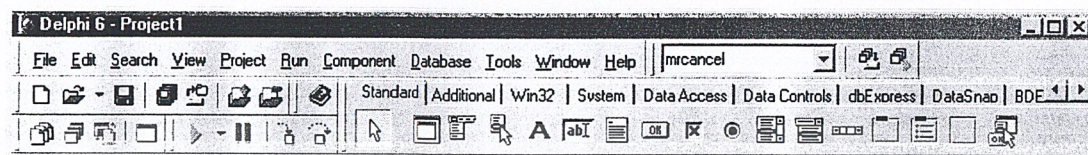
Code Editor เป็นส่วนที่เราใช้ในการเขียนคำสั่งต่างๆ เข้าไปเพื่อกำกับการทำงานของแอปพลิเคชัน ให้สอดคล้องกันซึ่งในคำสั่งต่างๆ ในภาษาปาสคาลที่อยู่ใน Code Editor จะประกอบไปด้วยส่วนที่ Delphi เตรียมไว้ให้ และส่วนที่เราต้องเขียนเพิ่มเติมเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7.1 การใช้งานคอมโพเนนต์พื้นฐานของ Delphi

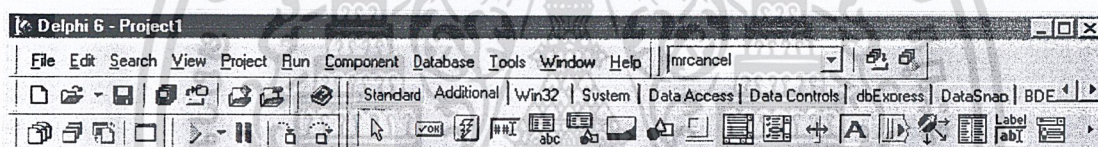
สำหรับ Delphi 5.0 นั้นแบ่งคอมโพเนนต์ได้ทั้งสิ้น 19 กลุ่มแต่ที่ใช้งานหลักๆ นั้นมีดังนี้

1. **Standard** - เป็นกลุ่มคอมโพเนนต์มาตรฐาน ซึ่งมีหน้าตาเหมือนกับการใช้งานทั่วไปของ Windows



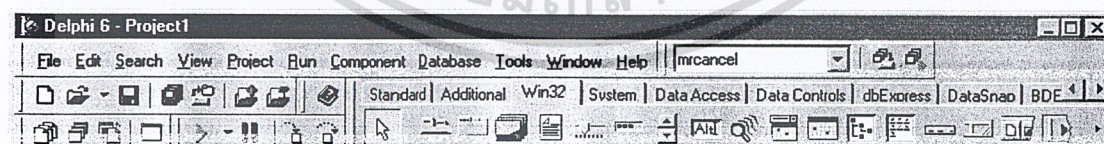
ภาพที่ 2.21 Standard

2. **Additional** – เป็นกลุ่มคอมโพเนนต์ที่พิเศษ นำมาเสริมการทำงานของคอมโพเนนต์ Standard ทำให้แอปพลิเคชันที่สร้างขึ้นนั้นใช้งานได้อย่างสวยงาม และสะดวกสบาย



ภาพที่ 2.22. Additional

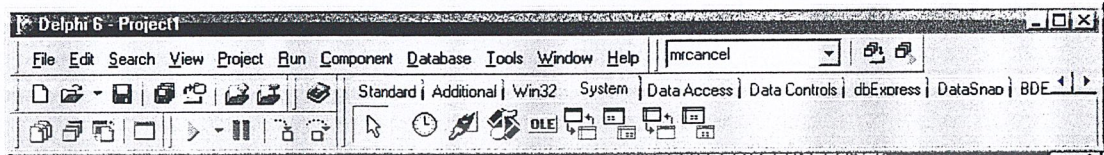
3. **Win32** – เป็นกลุ่มคอมโพเนนต์ที่เกิดขึ้นมาพร้อมกับระบบปฏิบัติการ 32 บิตของไมโครซอฟท์ (Windows 95, Windows 98, Windows NT 4.0 และ Windows 2000)



ภาพที่ 2.23 Win32

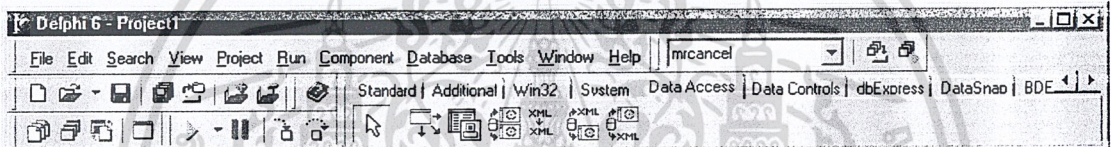
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. System – เป็นกลุ่มคอมโพเนนต์ที่สร้างขึ้นเพื่อติดต่อกับฮาร์ดแวร์ และระบบการทำงานภายในของคอมพิวเตอร์



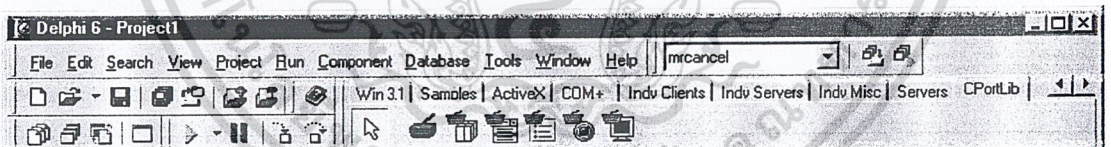
ภาพที่ 2.24 System

5. Data Access - เป็นกลุ่มคอมโพเนนต์ที่สร้างขึ้นมาเพื่อติดต่อกับฐานข้อมูล



ภาพที่ 2.25 Data Access

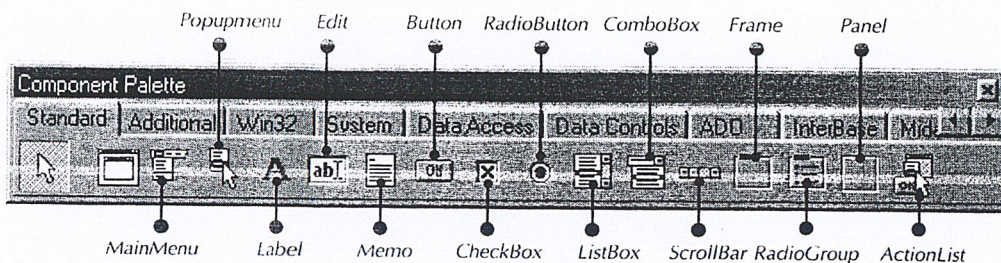
6. Data Controls – เป็นกลุ่มคอมโพเนนต์ที่สร้างมาเพื่อติดต่อกับผู้ใช้งานในการสร้างแอปพลิเคชันฐานข้อมูล ซึ่งจะใช้งานในการรับค่า/แสดงข้อมูลจากฐานข้อมูล



ภาพที่ 2.27 . Data Controls

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7.2 คอมโพเนนต์พื้นฐานในกลุ่ม Standard



ภาพที่ 2.27 คอมโพเนนต์พื้นฐานในกลุ่ม Standard

### การใช้งานคอมโพเนนต์ Label

คอมโพเนนต์ Label จะทำหน้าที่แสดงข้อความตามที่เราต้องการ ซึ่งข้อความดังกล่าวมักจะเป็นข้อความที่ไม่มีการเปลี่ยนแปลงแล้ว หรือไม่มีการรับค่าจากผู้ใช้งาน บางคนจึงมักจะเรียก Label ว่า “ป้ายข้อความ”

### การใช้งานคอมโพเนนต์ Button

คอมโพเนนต์ Button เป็นคอมโพเนนต์ที่มีการนำมาใช้งานมากที่สุด ทำหน้าที่รับคำสั่งจากผู้ใช้งานในลักษณะการคลิกที่ปุ่มนั้น หรือกด Enter เมื่อโฟกัสอยู่ที่ปุ่มนั้น

### การใช้งานคอมโพเนนต์ Edit

คอมโพเนนต์ Edit ทำหน้าที่รับข้อความจากผู้ใช้งานที่ป้อนเข้ามา นอกจากนี้ยังสามารถแสดงข้อความได้อีกด้วย

### การใช้งานคอมโพเนนต์ Memo

คอมโพเนนต์ Memo นั้นมีความสามารถเช่นเดียวกับคอมโพเนนต์ Edit แต่ต่างกันที่ Memo สามารถแสดงข้อความได้มากกว่า 1 บรรทัด

### การใช้งานคอมโพเนนต์ CheckBox

คอมโพเนนต์ CheckBox มีไว้เพื่อสร้างตัวเลือกในแอปพลิเคชัน ซึ่งตัวเลือกนี้สามารถเลือกได้พร้อมๆ กันหลายๆ ตัวเลือก

### การใช้งานคอมโพเนนต์ RadioButton

คอมโพเนนต์ RadioButton นั้นมีไว้เพื่อสร้างตัวเลือกเช่นเดียวกันกับคอมโพเนนต์ CheckBox แต่เราสามารถเลือกได้เพียงตัวเลือกเดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การใช้งานคอมโพเนนต์ ListBox

คอมโพเนนต์ ListBox ก็เป็นอีกวิธีที่ใช้ในการให้ผู้ใช้เลือกจากตัวเลือกที่มีให้ โดยภายใน ListBox จะมีตัวเลือกต่างๆ ให้เลือก ถ้ามีจำนวนมากก็จะแสดง ScrollBar ให้เราเลื่อนขึ้นลงเพื่อเลือกได้

### การใช้งานคอมโพเนนต์ ComboBox

คอมโพเนนต์ ComboBox นั้นเป็นรายการตัวเลือกได้เหมือนกับ ListBox แต่มีข้อพิเศษกว่าคือ นอกจากจะเลือกจากตัวเลือกในลักษณะ DropDown เช่นเดียวกับ ListBox แล้วยังสามารถเลือกโดยการคลิกรายละเอียดเข้าไปในลักษณะเดียวกันกับคอมโพเนนต์ Edit, พูดย่างๆ ก็คือ ComboBox เป็นเหมือนการผสมกันระหว่างคอมโพเนนต์ Edit และ ListBox

### การใช้งานคอมโพเนนต์ ScrollBar

คอมโพเนนต์ ScrollBar ก็คือ แถบเลื่อน (ซึ่งเราพบกันบ่อยใน Windows) ซึ่ง ScrollBar จะเลื่อนตำแหน่งภายในวินโดว์

### การใช้งานคอมโพเนนต์ GroupBox

คอมโพเนนต์ GroupBox ใช้ในการจัดกลุ่มคอมโพเนนต์ที่วางอยู่บนฟอร์ม โดยเราจะเลือกคอมโพเนนต์ที่สัมพันธ์กันมาวางไว้ใน GroupBox เดียวกัน โดยที่ GroupBox จะมีคุณสมบัติเป็นเหมือน Container คือ สามารถบรรจุเอาคอมโพเนนต์ต่างๆ ไว้ภายใน ซึ่งจะไม่เกี่ยวข้องกับคอมโพเนนต์อื่นใด ที่อยู่นอก Container

### การใช้งานคอมโพเนนต์ RadioGroup

คอมโพเนนต์ RaduoGroup ทำหน้าที่คล้ายกับ GroupBox แต่มีความพิเศษคือ ใช้แยกกลุ่มของ RadioButton เข้าด้วยกัน ซึ่งจะให้ผลเหมือนกับการทำงานของ GroupBox แต่มีวิธีการใช้งานที่แตกต่างกัน

### การใช้งานคอมโพเนนต์ Panel

คอมโพเนนต์ Panel เป็นคอมโพเนนต์อีกประเภทที่มีคุณสมบัติเป็น Container คือ สามารถบรรจุเอาคอมโพเนนต์อื่นๆ เข้าไปได้ แต่จะช่วยให้รูปร่างหน้าต่างที่สวยงามไปอีกแบบ

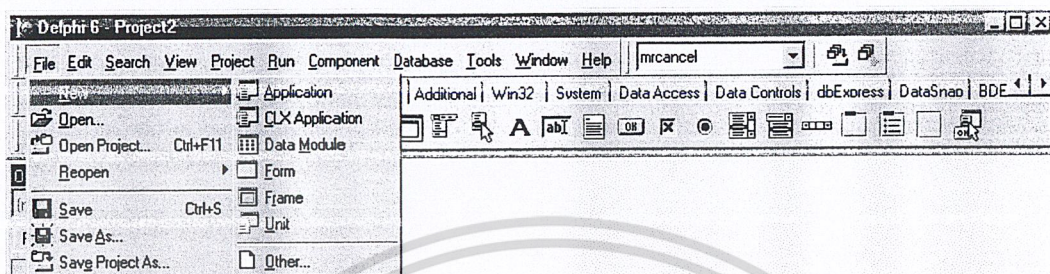
### การใช้งานคอมโพเนนต์ ActionList

คอมโพเนนต์ ActionList ทำหน้าที่รวบรวมเอาคอมโพเนนต์ต่างๆ ที่เราเห็นว่าน่าจะมีผลการทำงานอย่างเดียวกัน ให้นำมาอยู่ในที่เดียวกัน ซึ่งจะง่ายต่อการจัดการ, แก้ไข และเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

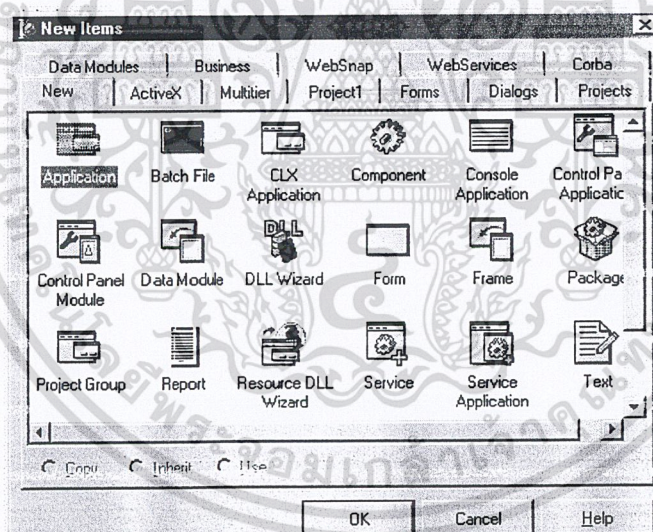
### 2.7.3 การเรียกใช้งาน template จาก Object Repository

เราสามารถเรียกใช้งาน Template ต่างๆ ที่อยู่ใน Object Repository โดยได้เลือกเมนู File > New ... ซึ่งจะปรากฏไอคอนที่ชื่อ New ให้เราเลือก Template ที่ต้องการ



ภาพที่ 2.28 การเลือก TemplateObject ของ Repository

จะเห็นว่า Template ต่างๆ จะถูกจัดอยู่เป็นกลุ่ม ซึ่งเราสามารถเลือกได้ตามความเหมาะสมของการสร้างแอปพลิเคชันแต่ละแบบ

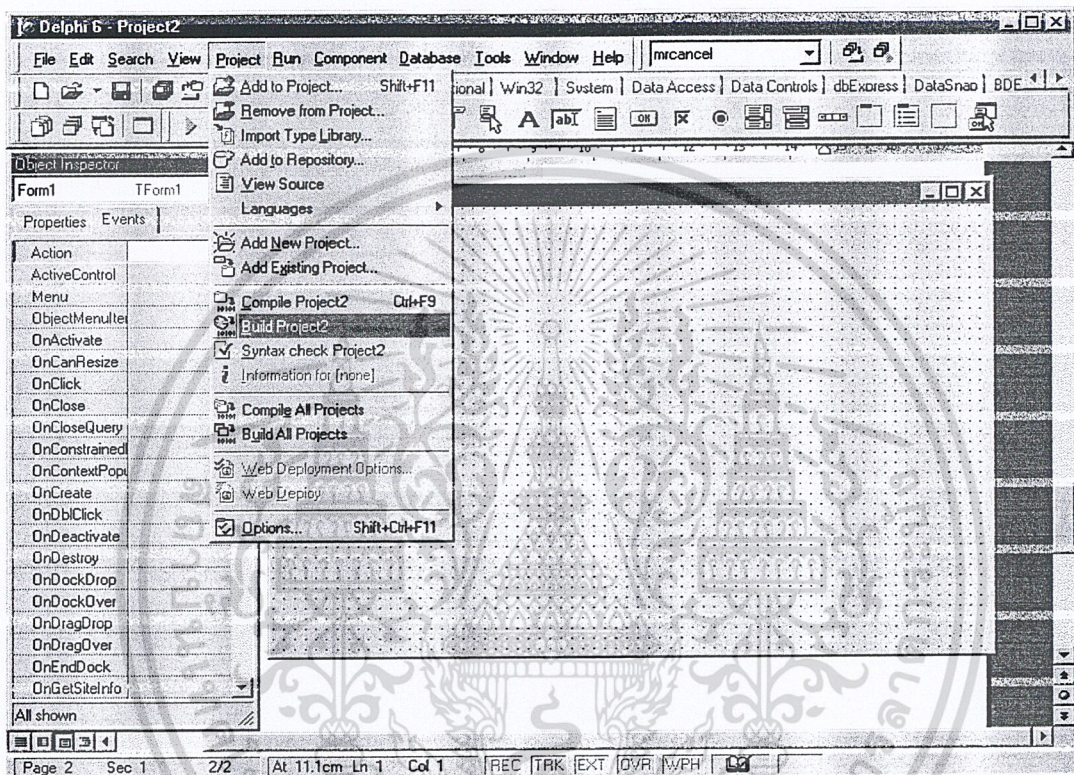


ภาพที่ 2.29 แท็บต่างๆ ของ Object Repository

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7.4 การสร้างไฟล์ .EXE เพื่อนำไปใช้งานจริง

แอปพลิเคชันที่ผ่านการทดสอบแล้ว เราสามารถนำไปรันใน PC เครื่องใดๆ ก็ได้ แต่ก่อนจะทำได้เช่นนั้น เราต้องแปลงให้เป็นไฟล์ที่มีนามสกุลเป็น .EXE ซึ่งทำได้โดยเรียกเมนู Project> Build ชื่อโปรเจกต์ ซึ่งจะทำให้ Delphi ทำการคอมไพล์โปรเจกต์ แล้วสร้างไฟล์นามสกุล .EXE ให้เรา

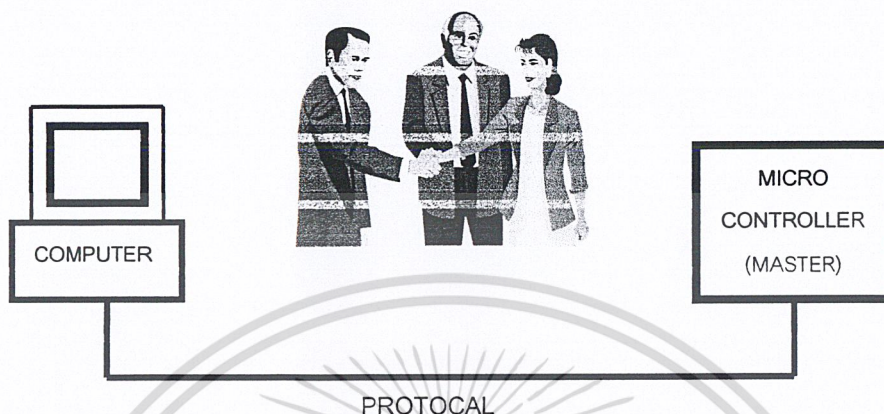


ภาพที่ 2.30 ตั้ง Build Project

ถ้าไม่พบข้อผิดพลาดจากการคอมไพล์ เราสามารถตรวจสอบที่โฟลเดอร์ที่ได้บันทึกโปรเจกต์ เราจะพบไฟล์ .EXE ที่ต้องการ และเราสามารถนำไฟล์ดังกล่าวเพียงไฟล์เดียว ไปรันที่ PC เครื่องใดก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8 ข้อตกลงในการสื่อสาร(FAC-TALK PROTOCOL)



ข้อตกลงในการสื่อสารถูกออกแบบให้อยู่ในรูป ASCII จัดเป็นบล็อกโดยแบ่งเป็น

- บล็อกคำสั่ง (Command Block)
- บล็อกตอบสนอง (Response Block)

ในแต่ละบล็อกจะประกอบด้วยสาระสำคัญดังนี้

@	Unit Number	Header	Data	*	BCS หรือ XX	(CR)
---	-------------	--------	------	---	-------------	------

### รูปแบบบล็อกคำสั่ง

@	เป็นอักขระเริ่มต้นของบล็อก ค่าฐานสิบหกเท่ากับ 40 (Hex)
UnitNumber	ตำแหน่งของโมดูลที่คอมพิวเตอร์อ้างอิง (0-8) เลือกจากการทำงานที่ DIP Switch No.4,5,6,7,8
Header	คำสั่ง/รหัส ให้ปฏิบัติ เป็น ASCII 2 อักขระ อันได้แก่ RD: หมายถึงคำสั่งในการอ่านข้อมูลจาก โมดูล WR: หมายถึงคำสั่งในการเขียนข้อมูลจาก โมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data	ข้อมูลประกอบคำสั่ง/รหัส เป็น ASCII 2-4 อักขระ RD: ไม่ต้องมีข้อมูลประกอบ WR: ข้อมูลที่เป็นรหัส ASCII แทนค่าฐานสิบหก เช่น ข้อมูล 12AB(Hex) ชุดข้อมูล ASCII จะเป็น 31324142
*	เป็นอักขระปิดท้ายข้อมูล ค่าฐานสิบหกเท่ากับ(Hex)
Bcs หรือ XX	รหัสดำกับบล็อก เป็น ASCII 2 อักขระ ได้จากการ XOR ข้อมูล ASCII ทุกๆ อักขระ ในแต่ละบล็อก ถ้าเป็น XX โมคูลจะรู้ว่า ไม่มีการตรวจสอบ BCS
(CR)	รหัส ASCII (Return Code) 1 อักขระ

รูปแบบของบล็อกคอบสนอง

@	เป็นอักขระเริ่มต้นของบล็อก ค่าฐานสิบหกเท่ากับ 40(Hex)
Unit Number	ตำแหน่งของโมคูลที่คอมพิวเตอร์อ้างอิง (0-8)
Header	คำสั่ง/รหัส ให้ปฏิบัติ เป็น ASCII 2 อักขระ อัน ได้แก่ RD: หมายถึงคำสั่งให้ โมคูลตอบกลับเพื่อส่งข้อมูลให้ PC หรือ คอมพิวเตอร์ จาก MICROCONTROLLER ER: เมื่อการปฏิบัติคำสั่งใดๆ เกิดข้อผิดพลาด โมคูลจะตอบกลับ
Data	ข้อมูลประกอบคำสั่ง/รหัส เป็น ASCII 2-4 อักขระ RD: ข้อมูลที่ตอบกลับมาจะเป็นค่าสถานะปัจจุบันขณะอ่าน เป็นข้อมูลที่เป็นรหัส ASCII แทนค่าฐานสิบหก เช่น ข้อมูล 12AB(Hex) ชุดข้อมูลของ ASCII จะเป็น 31324142 เป็นต้น ER: 00: Over Range อักขระในบล็อกไม่อยู่ในพิกัด 01: Format Error รูปแบบผิด 02: Block Check Sum Error รหัสกำกับบล็อกไม่ถูกต้อง 03: Unknow Command ไม่รู้จักคำสั่งนี้ 04: Execute Error ปฏิบัติการไม่ได้ หรือมีข้อผิดพลาดอื่นๆ
*	เป็นอักขระปิดท้ายข้อมูล ค่าฐานสิบหกเท่ากับ 2A(Hex)

เอกสารนี้เป็นเอกสารหนึ่งของกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ หากมีข้อผิดพลาดประการใด  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รหัสกำกับบล็อก เป็น ASCII 2 อักขระ ได้จากการ XOR ข้อมูล ASCII ทุกๆ อักขระในแต่ละบล็อก
(CR)	รหัส ASCII (Return Code) 1 อักขระ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบ

#### การออกแบบในส่วนของระบบคิวอัตโนมัติดังนี้

##### 3.1 การออกแบบในส่วนของ MASTER BOARD

การออกแบบในส่วนของ MASTER BOARD นี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ 89S8252 เหตุผลที่ใช้เบอร์นี้ก็เพราะว่าเป็น MCU ตัวใหม่ของ ATMEL ที่มีคุณสมบัติพิเศษคือ มี SPI-PORT ที่ประกอบขึ้นจากขาสัญญาณ 3 ขา คือ P 1.5 (MOSI ),P1.6 (MISO )และ P1.7 (SCK ) ทำหน้าที่เพื่อการอ่านและการเขียนข้อมูลลงในชิป โดยส่วนของ FLASH MEMORY สามารถเขียนซ้ำได้ถึง 1,000 ครั้ง ในขณะที่ส่วน EEPROM สามารถเขียนซ้ำได้ถึง 10,000 ครั้ง ซึ่ง SPI เป็นคุณสมบัติที่มีอยู่แล้วภายในตัวของ CPU ซึ่งถูกนำมาทำเป็น PORT สำหรับต่อใช้งานกับ SPI-LOAD (ตัวช่วย Load ) เพื่อการพัฒนาโปรแกรมที่มีรูปแบบใหม่ที่ทำให้ความสะดวกรวดเร็วขึ้น โดยสามารถนำโปรแกรมจากเครื่อง PC ผ่านตัวช่วย LOAD เก็บลงตัว CPU ได้เลยโดยไม่ต้องใช้เครื่องโปรแกรม MCU ทำให้สะดวกมากยิ่งขึ้นในการแก้ไขในส่วนของโปรแกรม

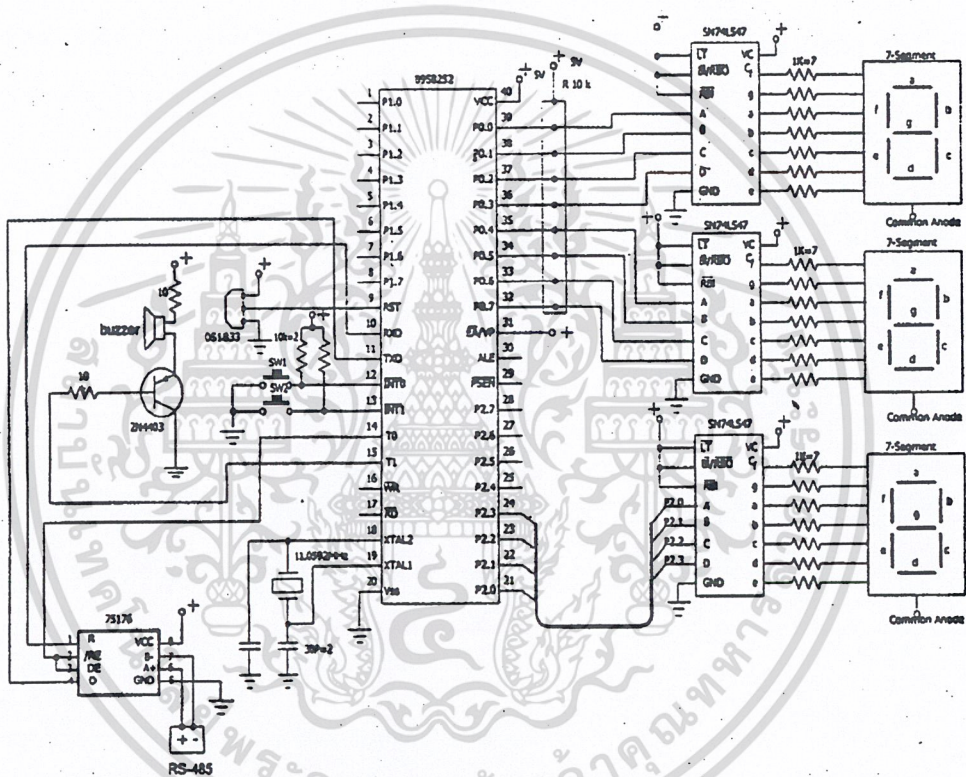
MASTER BOARD นี้ใช้ในการควบคุมการทำงานของระบบทั้งหมดในส่วนของ การติดต่อกันของบอร์ดต่างๆ การทำงานของ MASTER BOARD จะมีการแอสกนไปยังตัว Slave ว่ามีตัวไหนว่างอยู่บ้าง หากว่าตัว Slave ตัวไหนว่างอยู่ไมโครคอนโทรลเลอร์จะส่งหมายเลขคิวไปยัง Slave ตัวนั้น แต่ถ้าหากว่า Slave ว่างก็จะแอสกนหา Slave ตัวต่อไป MASTER BOARD จะใช้ในส่วนของการรับค่าคิวเข้ามาเก็บไว้เมื่อมีผู้เข้ามาใช้บริการมากคีย์สวิตซ์จากนั้นไมโครคอนโทรลเลอร์ก็จะนำค่าคิวที่ได้ไปเก็บไว้ในหน่วยความจำ คิวที่สามารถเก็บได้สามารถเก็บได้ทั้งหมด 16 คิวเมื่อยังไม่มีการจ่ายคิวออกไปแต่ถ้าเกิน 16 คิวเมื่อไหร่ไมโครคอนโทรลเลอร์ก็จะเริ่มเก็บคิวที่ 17 ส่วนคิวที่อยู่ก่อนหน้านี้อาจจะหายไป การติดต่อกันระหว่าง CPU แต่ละตัวจะใช้การติดต่อแบบอนุกรมด้วย RS-485 และมีการแสดงผลคิวด้วย 7-segment เพื่อบอกให้รู้ว่าตอนนี้มีคิวทั้งหมดกี่คิวและในส่วนของการพิมพ์หมายเลขคิวสามารถออกแบบให้ไมโครคอนโทรลเลอร์สั่งให้พิมพ์คิวได้แต่ส่วนนี้ยังไม่ได้ทำ จึงทำเฉพาะในส่วนของการแสดงผลและภาคเสียงเท่านั้น

##### 3.1.1 ส่วนประกอบของ MASTER BOARD มีดังนี้

- MCU 89S8252 FLASH MEMORY 8 KBYTE,EEPROM 2 KBYTE INTERNAL 256 BYTE

เอกสารนี้เป็นเอกสารของบริษัทเอกชนสงวนลิขสิทธิ์ไว้ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SOUND ใช้ Buzzer ตัวเล็กติดอยู่บนบอร์ดควบคุมเสียงด้วย BIT T1 เพื่อบอกสถานะการทำงานของบอร์ด
- RS-485 ใช้ชิป 75176 ใช้การติดต่อสื่อสารแบบอนุกรม ควบคุมทิศทางด้วย T0
- KEY 1 และ KEY 2 ติดอยู่กับ INTO และ INT1 สำหรับการใช้งานในการเช็คค่า เพื่อรับค่าคิวเข้ามา
- ส่วนของภาคแสดงผลด้วย 7-segment แสดงผลได้ 3 หลัก 7-segment ใช้เป็นแบบคอมมอนแอโนดรวม และมีชิป 74LS47 เพื่อ decode สัญญาณจาก P0 และ P2

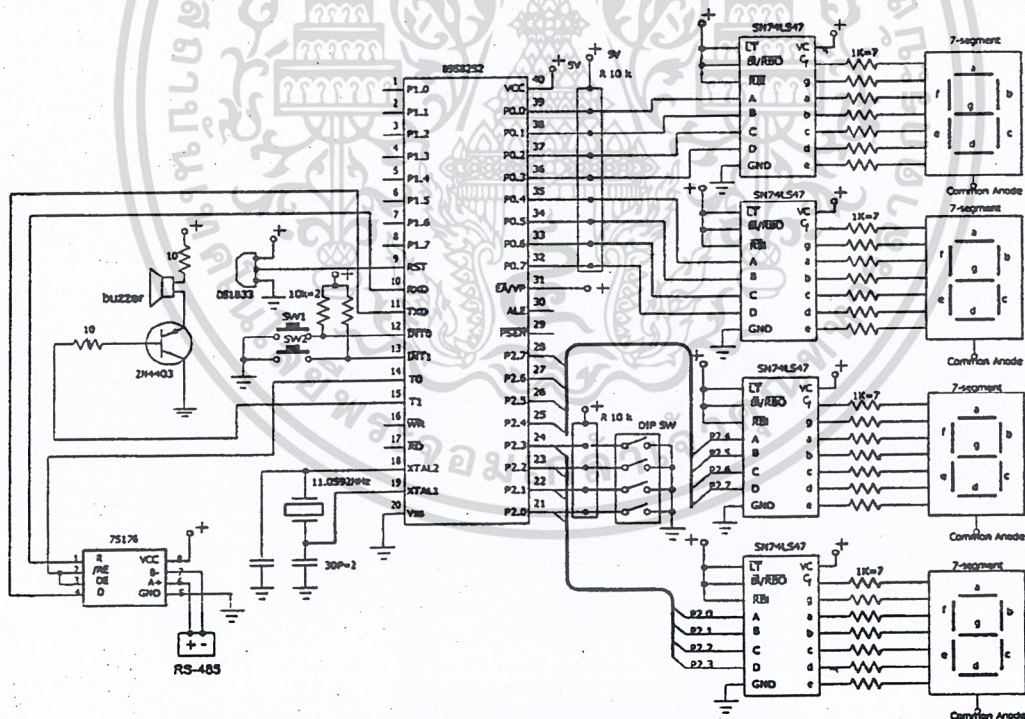


ภาพที่ 3.1 วงจรการต่อ MASTER BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การออกแบบในส่วนของ SLAVE BOARD

การออกแบบในส่วนของ SLAVE BOARD จะคล้ายกับ MASTER BOARD แต่จะเพิ่มในการกำหนดของช่องที่ต้องการให้ SLAVE BOARD ตัวนี้เป็นช่องไหนเพียงแต่กำหนดที่ DIPSWITCH ว่าต้องการให้เป็นช่องอะไร แต่ต้องขึ้นอยู่กับโปรแกรมว่าต้องการให้มีการกำหนดได้ที่ช่อง ซึ่งต้องสอดคล้องกัน การทำงานของ SLAVE BOARD จะรอรับหมายเลขคิวจาก MASTER BOARD หากตัว SLAVE BOARD ตัวไหนว่างอยู่ก็จะรับหมายเลขคิวเข้ามาซึ่งก็คือมีการกดเรียกคิว และจากนั้นก็จะมีการแสดงผลของหมายเลขคิวที่รับเข้ามา แสดงออกทาง 7-segment ของบอร์ดทำให้พนักงานทราบว่าคิวที่เท่าไร และในเวลาเดียวกันนั้น ข้อมูลก็จะถูกส่งออกทาง RS-485 ซึ่งใช้สายสัญญาณเพียง 2 เส้น ซึ่งจะต่อกับในส่วนของภาคแสดงผลและภาคเสียง ใช้ในการแสดงผลการเรียกคิวให้แก่ผู้มาติดต่อทราบว่าตอนนี้เป็นคิวที่เท่าไร ในส่วนของภาคเสียงก็จะพูดคิวที่กำลังแสดงอยู่นั้นซึ่งจะสอดคล้องกัน ซึ่งก็จะทำงานตามขั้นตอนที่จะกล่าวมาไปอย่างนี้เมื่อมีการรับค่าคิวเข้ามาใหม่ และใน SLAVE BOARD นอกจากจะมีส่วนของการเรียกคิวแล้ว ยังสามารถทำการเรียกซ้ำได้อีกเมื่อมีคิวที่เรียกไปไม่มีการมาติดต่อ



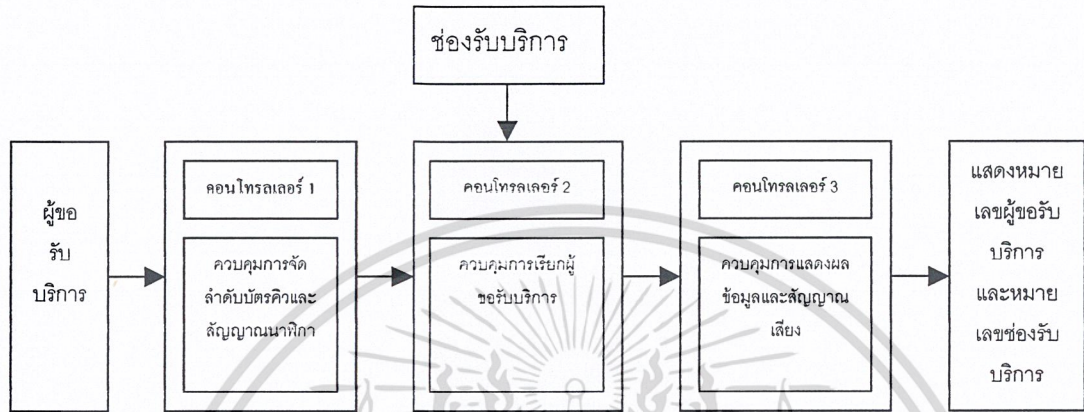
ภาพที่ 3.2 วงจรการต่อ SLAVE BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

# สรุปการทำงาน

### 4.1 แผนภาพแสดงการทำงาน



ภาพที่ 4.1 แผนภาพแสดงการทำงาน

#### 1. ไมโครคอนโทรลเลอร์ตัวที่ 1

ควบคุมการนับจำนวนผู้มาใช้บริการ และส่งข้อมูลนั้นให้กับไมโครคอนโทรลเลอร์ตัวที่ 2 เพื่อใช้ในการเปรียบเทียบค่าต่อไป

นอกจากนั้นไมโครคอนโทรลเลอร์ตัวที่ 1 ยังประกอบด้วยสัญญาณนาฬิกา ซึ่งจะเป็ข้อมูลอ้างอิงให้กับสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ ตัวที่ 3 ซึ่งใช้ข้อมูลในการแสดงผลต่อไป

#### 2. ไมโครคอนโทรลเลอร์ตัวที่ 2

ใช้ข้อมูลของผู้ใช้บริการจากไมโครคอนโทรลเลอร์ ตัวที่ 1 เพื่อนำมาเปรียบเทียบกับการเรียกผู้ให้บริการของพนักงานที่เคาน์เตอร์เพื่อให้ได้มาซึ่งข้อมูลหมายเลขของผู้รับบริการและช่องบริการที่จะให้กรบริการ เพื่อส่งไปแสดงผลซึ่งควบคุมโดยไมโครคอนโทรลเลอร์ ตัวที่ 3 ต่อไป

นอกจากนั้นยังควบคุมการเรียกซ้ำในกรณีที่ผู้รับบริการไม่มาขอรับบริการโดยเคาเตอร์บริการจะทำการเรียกซ้ำโดยกดปุ่มเรียกซ้ำ โดยที่จะนำข้อมูลที่ส่งไปที่ไมโครคอนโทรลเลอร์ ตัวที่ 3 เพื่อแสดงการเรียกซ้ำต่อไป

#### 3. ไมโครคอนโทรลเลอร์ ตัวที่ 3

เมื่อได้ข้อมูลจากไมโครคอนโทรลเลอร์ ตัวที่ 2 แล้วไมโครคอนโทรลเลอร์ ตัวที่ 3 จะนำข้อมูลที่ได้นำมาทำการประมวลผลเพื่อนำไปแสดงผลของหมายเลขของผู้รับบริการและช่องบริการที่จะให้บริการ และนำไปจัดลำดับข้อมูลเพื่อแสดงผลทางเสียงต่อไป

เอกสารนี้เป็นเอกสารในหนังสือพิมพ์ที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่เกิดการเรียกซ้ำ จะนำข้อมูลในส่วนของเคาน์เตอร์ที่มีการเรียกซ้ำมาแสดงผลทางหน้าจอต่อไป

## 4.2 ลำดับขั้นของการทำงาน

จากระบบที่กล่าวมาแบ่งการทำงานได้ 3 ส่วนหลักๆ คือส่วนของการจัดลำดับบัตรคิวของผู้ขอรับบริการ ส่วนของการจัดสรรผู้รับบริการของแต่ละช่องบริการ และ ส่วนของการแสดงผลทั้งทางหน้าจอและทางเสียง โดยใช้การทำงานของไมโครคอนโทรลเลอร์ 3 ตัวซึ่งแต่ละตัวจะควบคุมการทำงานในแต่ละส่วน โดยที่แต่ละตัวสื่อสารร่วมกันผ่านทางพอร์ตอนุกรม ส่วนของการจัดลำดับมีขั้นตอนการทำงานดังนี้

1. เริ่มต้นบริการที่ช่องรับบริการรับคำสั่งจากพนักงานแสดงความต้องการให้บริการ
2. ระบบรอคำสั่งจากลูกค้าแสดงความต้องการรับบริการเพื่อนำมาประมวลผลร่วมกับส่วนช่องรับบริการผ่านทางพอร์ตอนุกรม
3. ส่วนจัดลำดับบัตรคิวรับคำสั่งจากลูกค้าเพื่อจัดลำดับและส่งข้อมูลให้ส่วนช่องรับบริการ
4. หน่วยประมวลผลที่ควบคุมส่วนจัดลำดับการรับบริการรับข้อมูลจากส่วนจัดลำดับบัตรคิวนำมาเปรียบเทียบเพื่อแสดงลำดับที่ให้บริการออกทางหน้าจอแสดงผล
5. เมื่อส่วนแสดงผลรับข้อมูลจากส่วนจัดลำดับการรับบริการแล้วจะนำมาแยกส่วนที่เป็นหมายเลขบัตรคิว และหมายเลขเคาน์เตอร์ออกจากกันเพื่อนำเป็นข้อมูลแสดงผลทางหน้าจอ และจัดลำดับในการแสดงข้อมูลเสียงเพื่อเรียกผู้รับบริการต่อไป
6. ในกรณีที่ผู้รับบริการไม่ปรากฏตัวที่ช่องเคาน์เตอร์บริการผู้ให้บริการจะขอให้แสดงสัญญาณที่หน้าจอเพื่อเรียกผู้รับบริการอีกครั้ง โดยส่วนการจัดลำดับการรับบริการจะส่งข้อมูลเดิมของเคาน์เตอร์บริการนั้น ไปที่ส่วนแสดงผลเพื่อแสดงผลทางหน้าจอต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

จากผลการทดลองในบทที่ 4 โดยรวมพบว่าระบบสามารถทำงานได้ตามความต้องการที่ออกแบบไว้ แต่ในบางส่วนของการทำงาน ยังมีข้อที่ควรปรับปรุงอยู่ดังนี้

ในส่วนของการแสดงผล ถ้าต้องการแสดงข้อมูลที่มากกว่านี้ เช่น จำนวนคนที่ยังรอรับบริการอยู่ จำเป็นจะต้องเพิ่มตัวเลข(Digit) บนจอแสดงผล และเขียนโปรแกรมเพิ่มเติม

ในส่วนของการ Simulink การทำงานของระบบทั้งหมด การทำงานยังไม่เสถียรภาพดีพอ คือยังมีปัญหาทางด้าน Software ที่ใช้ทำการ Simulate ซึ่งอาจทำการปรับปรุงให้ดีขึ้น โดยการเขียนโปรแกรมรองรับในส่วนที่ยังไม่เสถียรภาพดีพอ

ในส่วนของเครื่องพิมพ์บัตรคิว ยังไม่สามารถทำการ Simulate ได้ และอีกทั้งเครื่องพิมพ์ยังมีราคาแพง ซึ่งจะต้องทำการหา Software ที่รองรับในจุดนี้มาทำการ Simulate



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. การใช้งานพีซีลิงค์ รศ. สุพรรณ กุลพานิชย์
2. FACTORY AUTOMATION SYSTEM ผศ. ทวีพล ชื้อสตัย
3. คู่มือการใช้งาน RTI16 & RTO16 บริษัทแฟคตอรีคอนซัลแตนซ์ จำกัด
4. ภาควิชาวิศวกรรมการวัดคุม “OMRON FA2 ADVANCED PLC WITH C200HS”  
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
5. Microsoft Visual Basic Version 5.0 ชาริน สิริธรรมชารี  
ชัญญชัช จำนงค์ภักดี
6. เริ่มต้นอย่างมืออาชีพด้วย Delphi 5 สัจจะ จรัสรุ่งรวีร์  
จักรพงษ์ สุขประเสริฐ
7. เรียนรู้และปฏิบัติการ ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช  
วรพจน์ กรแก้ววัฒนกุล  
ชัยวัฒน์ ลิ้มพรจิตวิไล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory  
Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

## Description

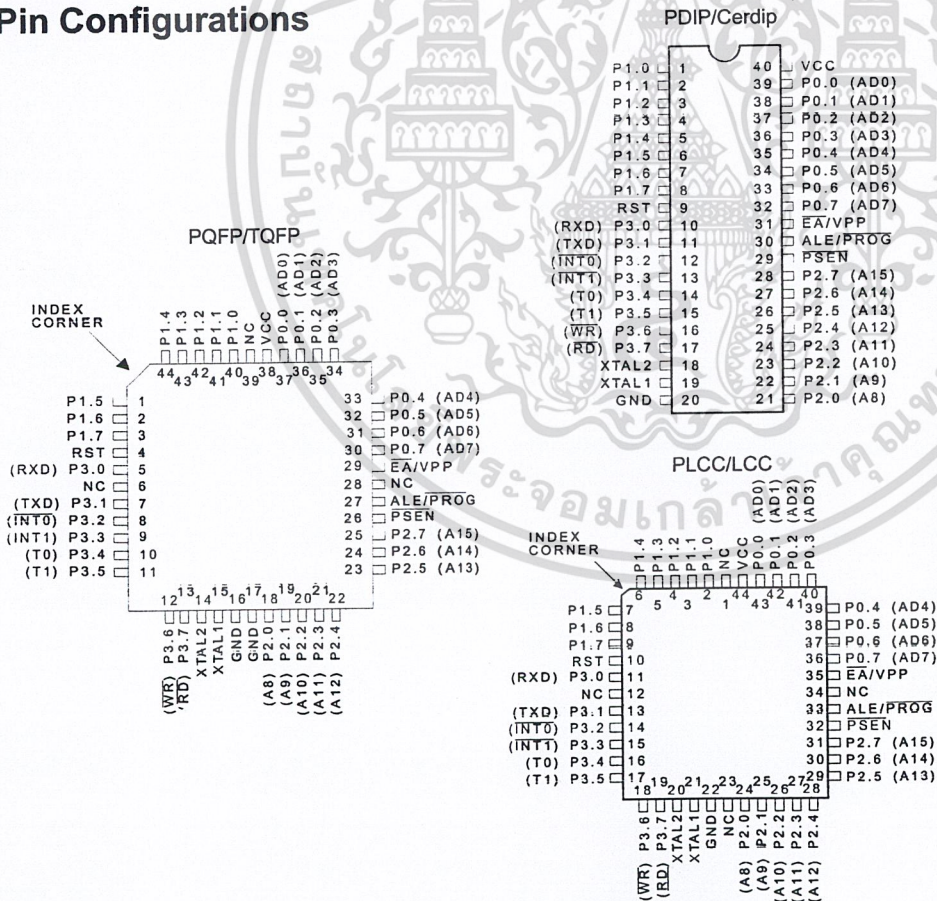
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

## 8-Bit Microcontroller with 4 Kbytes Flash

### AT89C51

## Pin Configurations

(continued)

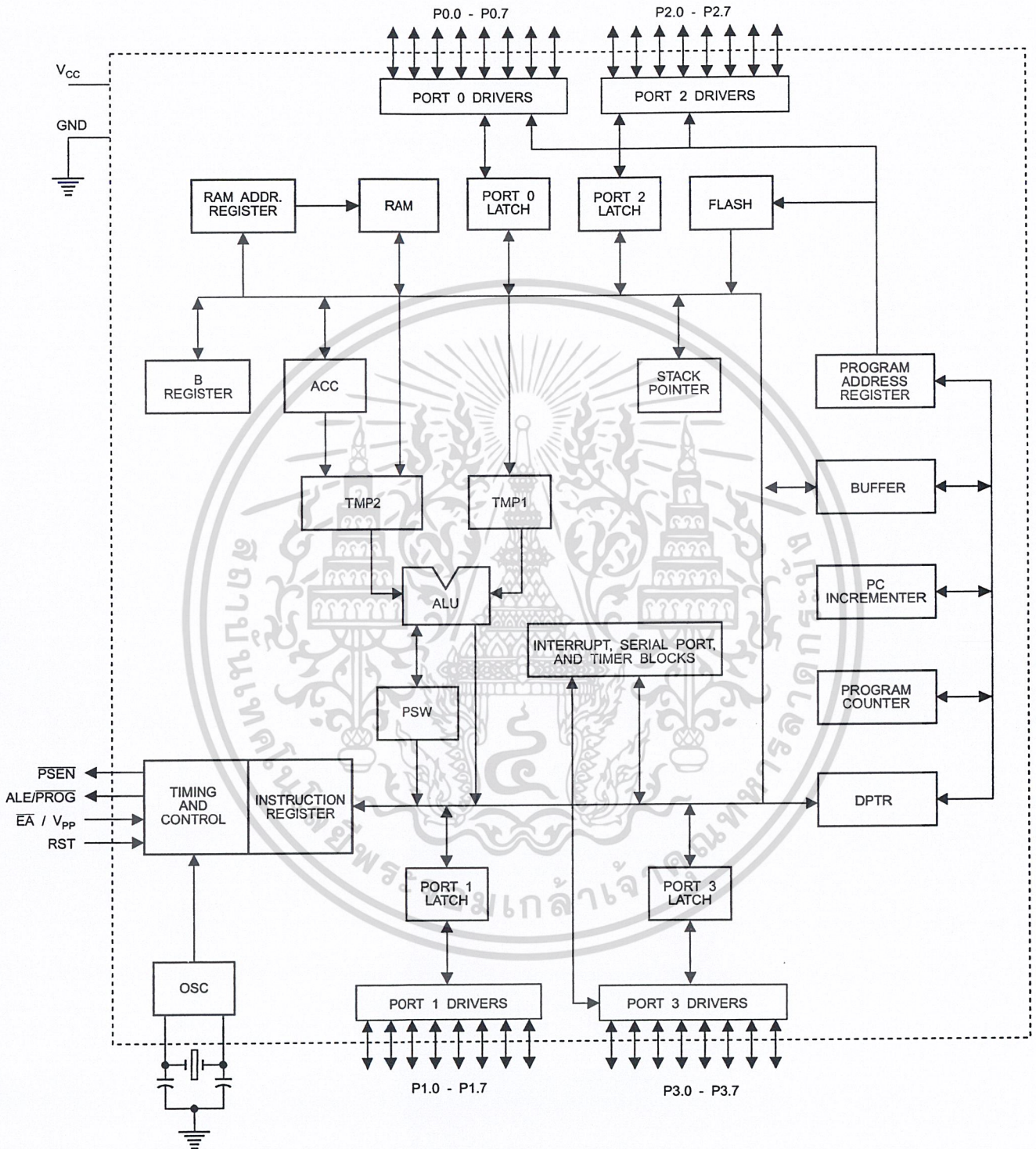


0265E



Atmel หน่วยงานนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Block Diagram



2 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นว่าเหมาะสมจะขอคืนเอกสารนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Description (Continued)

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

## Pin Description

V<sub>CC</sub>

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I<sub>IL</sub>) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I<sub>IL</sub>) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX

@ DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification. Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I<sub>IL</sub>) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/ $\overline{\text{PROG}}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ( $\overline{\text{PROG}}$ ) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

(continued)



## Pin Description (Continued)

When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

### $\overline{EA}/V_{PP}$

External Access Enable.  $\overline{EA}$  must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed,  $\overline{EA}$  will be internally latched on reset.

$\overline{EA}$  should be strapped to  $V_{CC}$  for internal program executions.

This pin also receives the 12-volt programming enable voltage ( $V_{PP}$ ) during Flash programming, for parts that require 12-volt  $V_{PP}$ .

### XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

### XTAL2

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

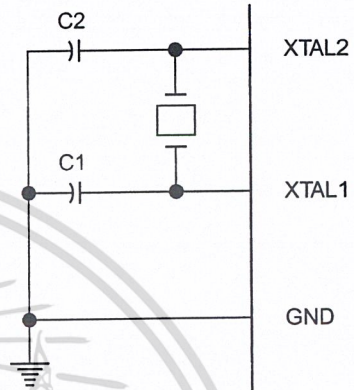
## Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this

mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

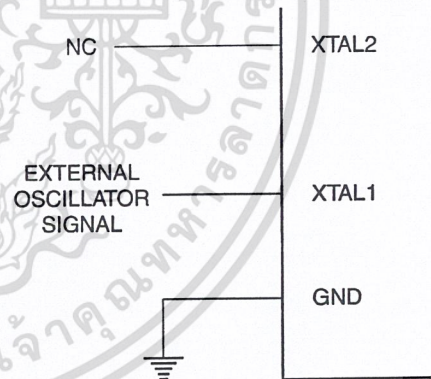
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hard-

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF  $\pm$  10 pF for Crystals  
= 40 pF  $\pm$  10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



## Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

ware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

## Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before Vcc

is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

## Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the  $\overline{EA}$  pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of EA be in agreement with the current logic level at that pin in order for the device to function properly.

## Lock Bit Protection Modes

Program Lock Bits				Protection Type
LB1	LB2	LB3		
1	U	U	U	No program lock features.
2	P	U	U	MOV <sub>C</sub> instructions executed from external program memory are disabled from fetching code bytes from internal memory, EA is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

## Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (Vcc) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V <sub>pp</sub> = 12 V	V <sub>pp</sub> = 5 V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

**Programming Algorithm:** Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise EA/V<sub>pp</sub> to 12 V for the high-voltage programming mode.
5. Pulse ALE/ $\overline{PROG}$  once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

**Data Polling:** The AT89C51 features  $\overline{Data}$  Polling to indicate the end of a write cycle. During a write cycle, an at-



ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า (continued)

## Programming the Flash (Continued)

tempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

**Chip Erase:** The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel

(031H) = 51H indicates 89C51

(032H) = FFH indicates 12 V programming

(032H) = 05H indicates 5 V programming

## Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

## Flash Programming Modes

Mode	RST	PSEN	ALE/ PROG	EA/ V <sub>PP</sub>	P2.6	P2.7	P3.6	P3.7		
Write Code Data	H	L		H/12V <sup>(1)</sup>	L	H	H	H		
Read Code Data	H	L	H	H	L	L	H	H		
Write Lock	Bit - 1	L		H/12V	H	H	H	H		
			Bit - 2	L		H/12V	H	H	L	L
					Bit - 3	L		H/12V	H	L
Chip Erase	H	L		H/12V			H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L		

Notes: 1. The signature byte at location 032H designates whether V<sub>PP</sub> = 12 V or V<sub>PP</sub> = 5 V should be used to enable programming.

2. Chip Erase requires a 10 ms  $\overline{\text{PROG}}$  pulse.

Figure 3. Programming the Flash

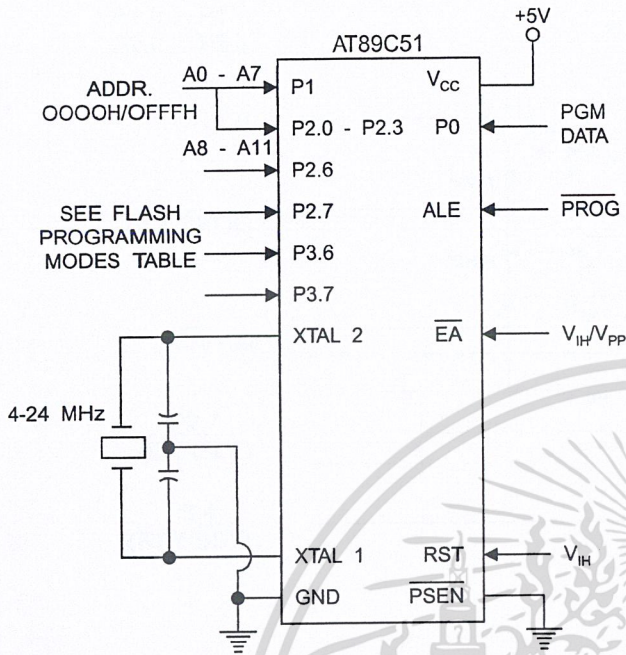
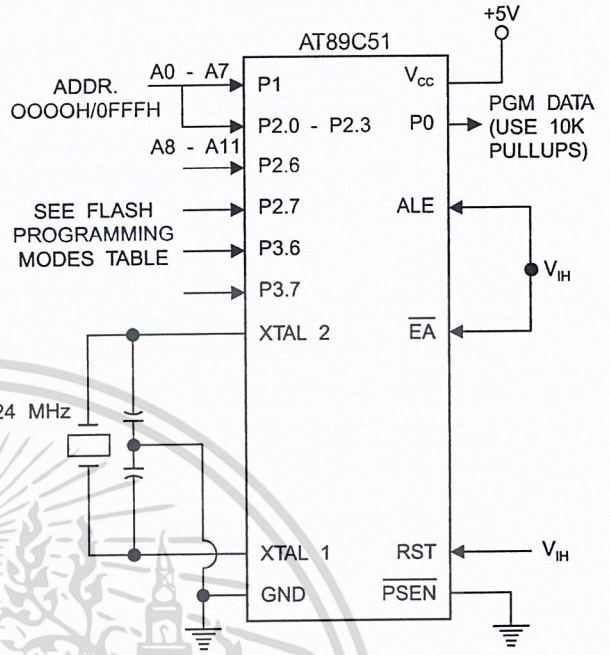


Figure 4. Verifying the Flash



## Flash Programming and Verification Characteristics

$T_A = 21^\circ\text{C to } 27^\circ\text{C}$ ,  $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
$V_{PP}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}^{(1)}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	4	24	MHz
$t_{AVGL}$	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
$t_{GHAX}$	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{DVGL}$	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
$t_{GHDX}$	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) High to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ Setup to $\overline{\text{PROG}}$ Low	10		$\mu\text{s}$
$t_{GHSL}^{(1)}$	$V_{PP}$ Hold After $\overline{\text{PROG}}$	10		$\mu\text{s}$
$t_{GLGH}$	$\overline{\text{PROG}}$ Width	1	110	$\mu\text{s}$
$t_{AVQV}$	Address to Data Valid		$48t_{CLCL}$	
$t_{ELQV}$	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
$t_{EHQV}$	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
$t_{GHBL}$	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	$\mu\text{s}$
$t_{WC}$	Byte Write Cycle Time		2.0	ms

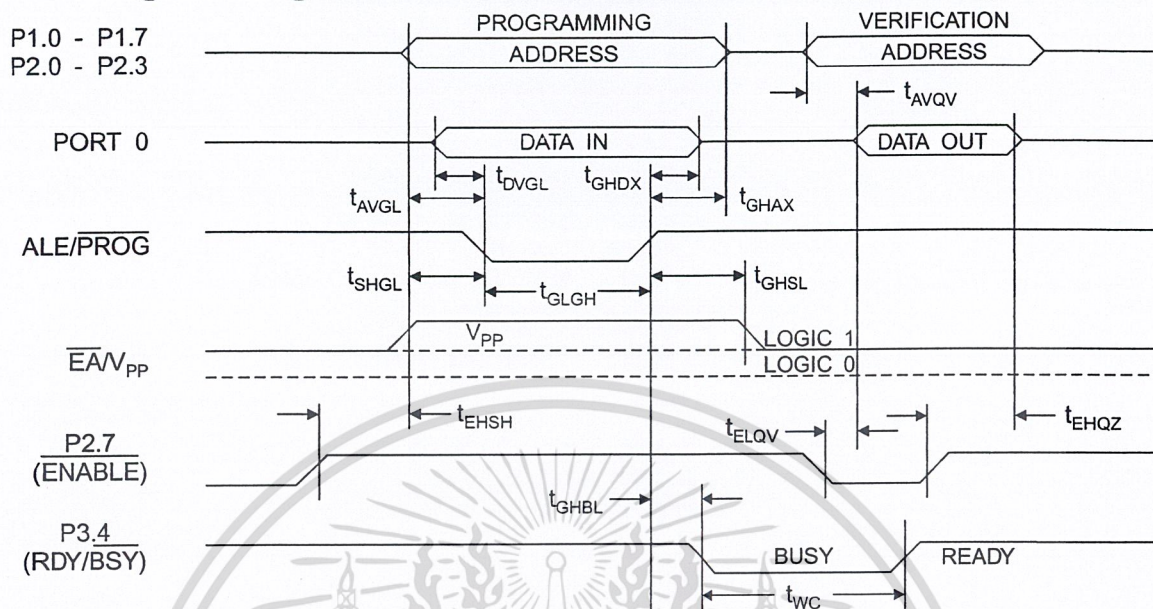
Note: 1. Only used in 12-volt programming mode.



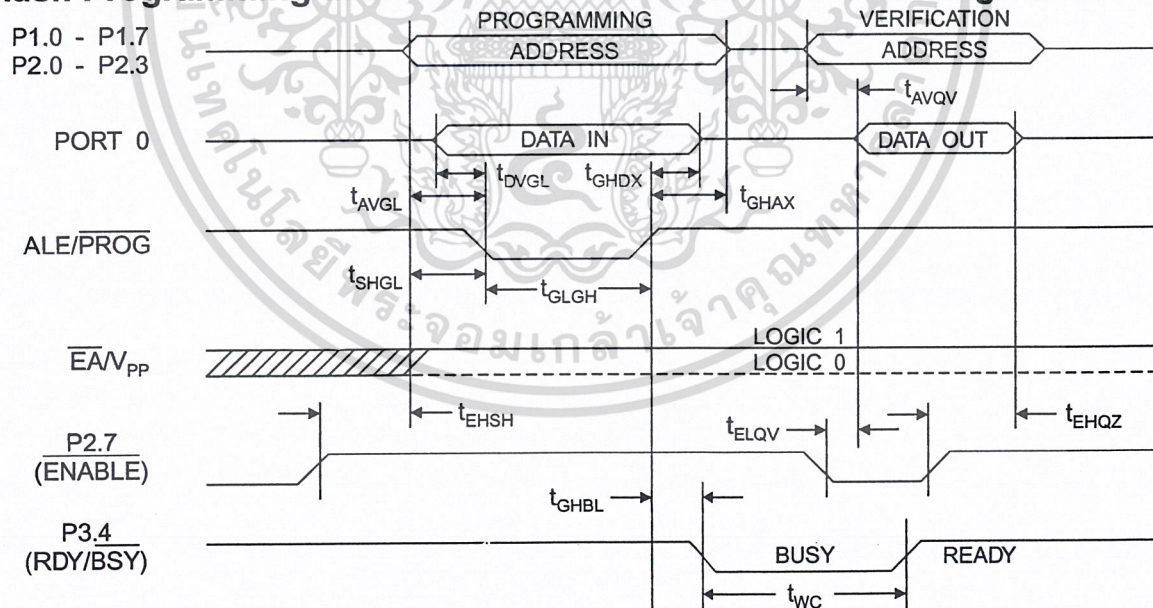
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## Flash Programming and Verification Waveforms - High Voltage Mode



## Flash Programming and Verification Waveforms - Low Voltage Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้ **AT89C51**

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Absolute Maximum Ratings\*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground .....	-1.0 V to +7.0 V
Maximum Operating Voltage .....	6.6 V
DC Output Current.....	15.0 mA

\*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## D.C. Characteristics

T<sub>A</sub> = -40°C to 85°C, V<sub>CC</sub> = 5.0 V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V <sub>IL</sub>	Input Low Voltage	(Except $\overline{EA}$ )	-0.5	0.2 V <sub>CC</sub> -0.1	V
V <sub>IL1</sub>	Input Low Voltage ( $\overline{EA}$ )		-0.5	0.2 V <sub>CC</sub> -0.3	V
V <sub>IH</sub>	Input High Voltage	(Except XTAL1, RST)	0.2 V <sub>CC</sub> +0.9	V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input High Voltage	(XTAL1, RST)	0.7 V <sub>CC</sub>	V <sub>CC</sub> +0.5	V
V <sub>OL</sub>	Output Low Voltage <sup>(1)</sup> (Ports 1,2,3)	I <sub>OL</sub> = 1.6 mA		0.45	V
V <sub>OL1</sub>	Output Low Voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	I <sub>OL</sub> = 3.2 mA		0.45	V
V <sub>OH</sub>	Output High Voltage (Ports 1,2,3, ALE, PSEN)	I <sub>OH</sub> = -60 μA, V <sub>CC</sub> = 5 V ± 10%	2.4		V
		I <sub>OH</sub> = -25 μA	0.75 V <sub>CC</sub>		V
		I <sub>OH</sub> = -10 μA	0.9 V <sub>CC</sub>		V
V <sub>OH1</sub>	Output High Voltage (Port 0 in External Bus Mode)	I <sub>OH</sub> = -800 μA, V <sub>CC</sub> = 5 V ± 10%	2.4		V
		I <sub>OH</sub> = -300 μA	0.75 V <sub>CC</sub>		V
		I <sub>OH</sub> = -80 μA	0.9 V <sub>CC</sub>		V
I <sub>IL</sub>	Logical 0 Input Current (Ports 1,2,3)	V <sub>IN</sub> = 0.45 V		-50	μA
I <sub>TL</sub>	Logical 1 to 0 Transition Current (Ports 1,2,3)	V <sub>IN</sub> = 2 V		-650	μA
I <sub>LI</sub>	Input Leakage Current (Port 0, EA)	0.45 < V <sub>IN</sub> < V <sub>CC</sub>		±10	μA
RRST	Reset Pulldown Resistor		50	300	KΩ
C <sub>IO</sub>	Pin Capacitance	Test Freq. = 1 MHz, T <sub>A</sub> = 25°C		10	pF
I <sub>CC</sub>	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode <sup>(2)</sup>	V <sub>CC</sub> = 6 V		100	μA
		V <sub>CC</sub> = 3 V		40	μA

Notes: 1. Under steady state (non-transient) conditions, I<sub>OL</sub> must be externally limited as follows:  
 Maximum I<sub>OL</sub> per port pin: 10 mA  
 Maximum I<sub>OL</sub> per 8-bit port:  
 Port 0: 26 mA  
 Ports 1, 2, 3: 15 mA

Maximum total IOL for all output pins: 71 mA  
 If IOL exceeds the test condition, V<sub>OL</sub> may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.  
 2. Minimum V<sub>CC</sub> for Power Down is 2 V.





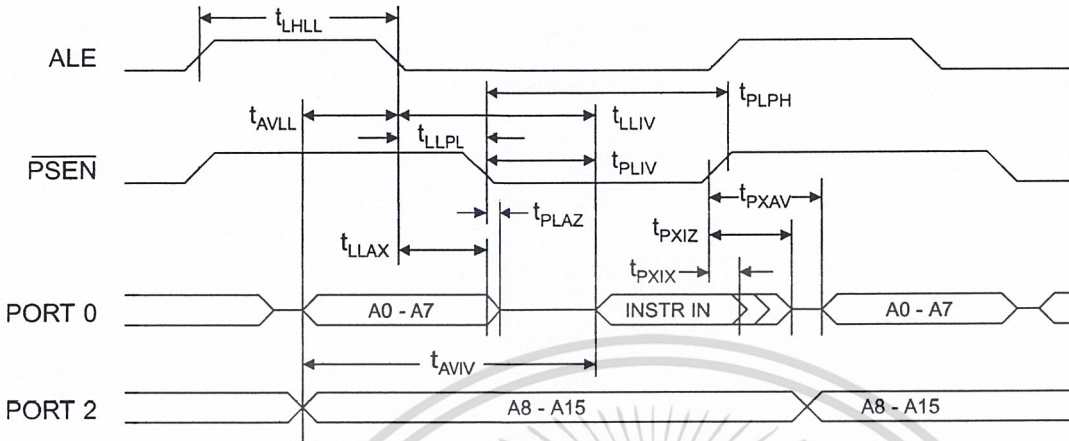
## A.C. Characteristics

(Under Operating Conditions; Load Capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; Load Capacitance for all other outputs = 80 pF)

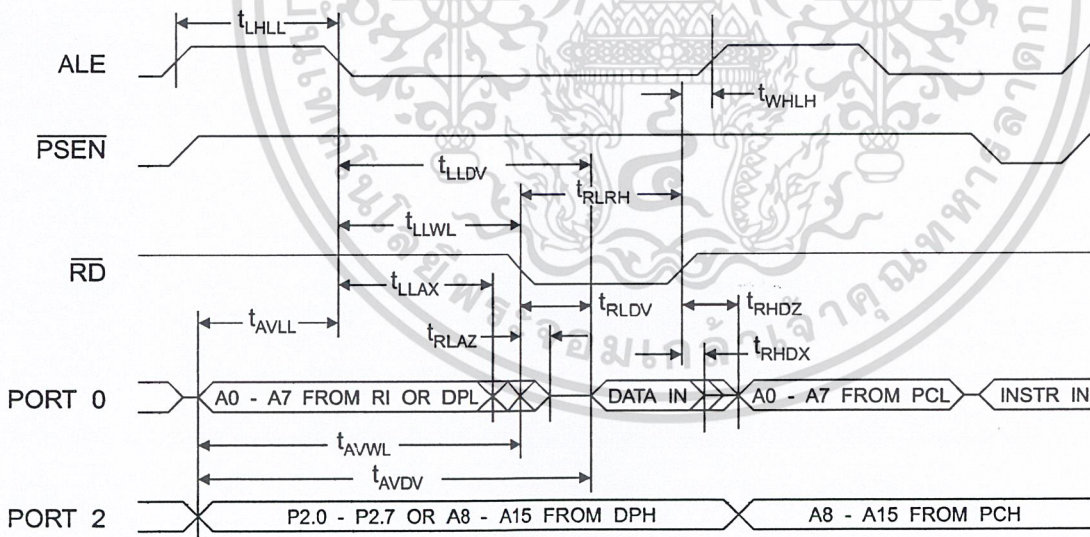
### External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
1/t <sub>CLCL</sub>	Oscillator Frequency			0	24	MHz
t <sub>LHLL</sub>	ALE Pulse Width	127		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	Address Valid to ALE Low	28		t <sub>CLCL</sub> -13		ns
t <sub>LLAX</sub>	Address Hold After ALE Low	48		t <sub>CLCL</sub> -20		ns
t <sub>LLIV</sub>	ALE Low to Valid Instruction In		233		4t <sub>CLCL</sub> -65	ns
t <sub>LLPL</sub>	ALE Low to PSEN Low	43		t <sub>CLCL</sub> -13		ns
t <sub>PLPH</sub>	PSEN Pulse Width	205		3t <sub>CLCL</sub> -20		ns
t <sub>PLIV</sub>	PSEN Low to Valid Instruction In		145		3t <sub>CLCL</sub> -45	ns
t <sub>PIX</sub>	Input Instruction Hold After PSEN	0		0		ns
t <sub>PIXZ</sub>	Input Instruction Float After PSEN		59		t <sub>CLCL</sub> -10	ns
t <sub>PIXAV</sub>	PSEN to Address Valid	75		t <sub>CLCL</sub> -8		ns
t <sub>AVIV</sub>	Address to Valid Instruction In		312		5t <sub>CLCL</sub> -55	ns
t <sub>PLAZ</sub>	PSEN Low to Address Float		10		10	ns
t <sub>RLRH</sub>	RD Pulse Width	400		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	WR Pulse Width	400		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	RD Low to Valid Data In		252		5t <sub>CLCL</sub> -90	ns
t <sub>RHDX</sub>	Data Hold After RD	0		0		ns
t <sub>RHDZ</sub>	Data Float After RD		97		2t <sub>CLCL</sub> -28	ns
t <sub>LLDV</sub>	ALE Low to Valid Data In		517		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	Address to Valid Data In		585		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	ALE Low to RD or WR Low	200	300	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	Address to RD or WR Low	203		4t <sub>CLCL</sub> -75		ns
t <sub>QVWX</sub>	Data Valid to WR Transition	23		t <sub>CLCL</sub> -20		ns
t <sub>QVWH</sub>	Data Valid to WR High	433		7t <sub>CLCL</sub> -120		ns
t <sub>WHQX</sub>	Data Hold After WR	33		t <sub>CLCL</sub> -20		ns
t <sub>RLAZ</sub>	RD Low to Address Float		0		0	ns
t <sub>WHLH</sub>	RD or WR High to ALE High	43	123	t <sub>CLCL</sub> -20	t <sub>CLCL</sub> +25	ns

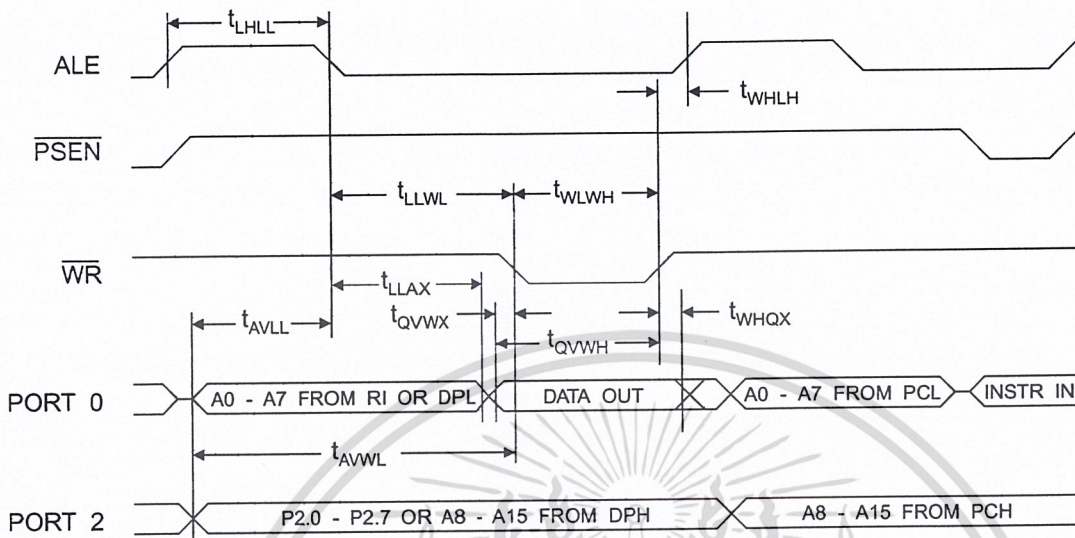
External Program Memory Read Cycle



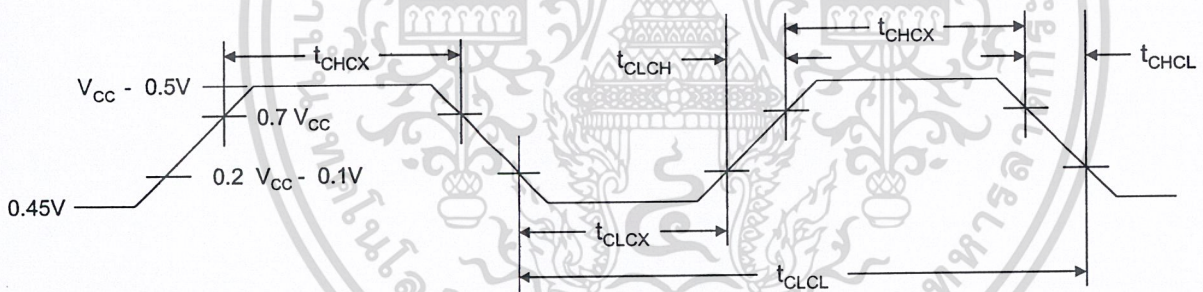
External Data Memory Read Cycle



## External Data Memory Cycle



## External Clock Drive Waveforms



## External Clock Drive

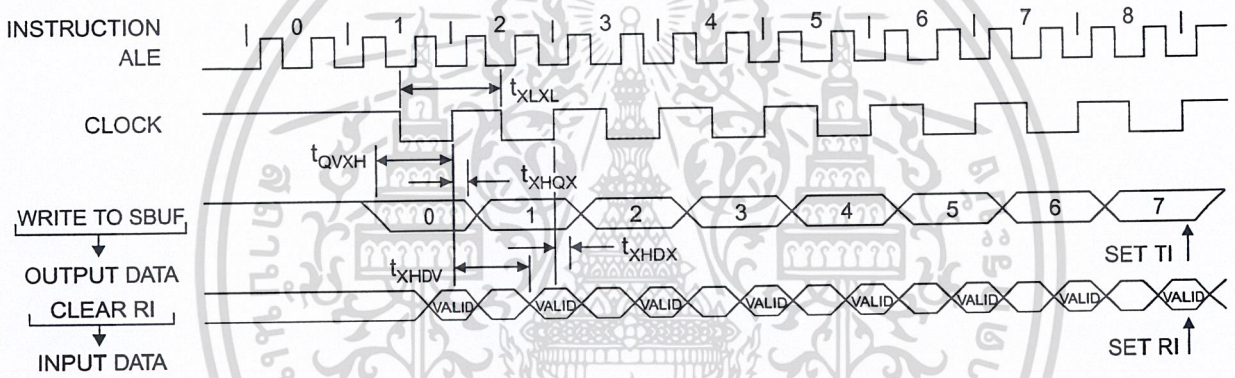
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
$t_{CLCL}$	Clock Period	41.6		ns
$t_{CHCX}$	High Time	15		ns
$t_{CLCX}$	Low Time	15		ns
$t_{CLCH}$	Rise Time		20	ns
$t_{CHCL}$	Fall Time		20	ns

**Serial Port Timing: Shift Register Mode Test Conditions**

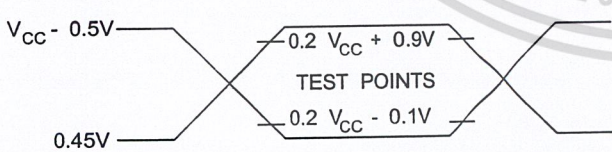
(V<sub>CC</sub> = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t <sub>XLXL</sub>	Serial Port Clock Cycle Time	1.0		12t <sub>CLCL</sub>		μs
t <sub>QVXH</sub>	Output Data Setup to Clock Rising Edge	700		10t <sub>CLCL</sub> -133		ns
t <sub>XHQX</sub>	Output Data Hold After Clock Rising Edge	50		2t <sub>CLCL</sub> -33		ns
t <sub>XHDX</sub>	Input Data Hold After Clock Rising Edge	0		0		ns
t <sub>XHDV</sub>	Clock Rising Edge to Input Data Valid		700		10t <sub>CLCL</sub> -133	ns

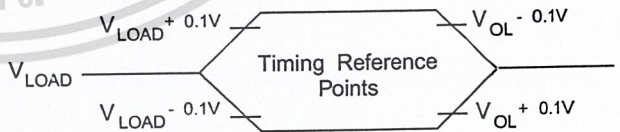
**Shift Register Mode Timing Waveforms**



**AC Testing Input/Output Waveforms <sup>(1)</sup>**



**Float Waveforms <sup>(1)</sup>**



Note: 1. AC Inputs during testing are driven at V<sub>CC</sub> - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V<sub>IH</sub> min. for a logic 1 and V<sub>IL</sub> max. for a logic 0.

Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V<sub>OH</sub>/V<sub>OL</sub> level occurs.





## Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range			
12	5 V $\pm$ 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)			
		AT89C51-12JC	44J				
		AT89C51-12PC	40P6				
		AT89C51-12QC	44Q				
		AT89C51-12AI	44A		Industrial (-40°C to 85°C)		
		AT89C51-12JI	44J				
	AT89C51-12PI	40P6					
	AT89C51-12QI	44Q					
	5 V $\pm$ 10%	AT89C51-12AA	44A	Automotive (-40°C to 125°C)			
		AT89C51-12JA	44J				
		AT89C51-12PA	40P6				
		AT89C51-12QA	44Q				
AT89C51-12DM		40D6	Military (-55°C to 125°C)				
AT89C51-12LM		44L					
16	5 V $\pm$ 20%	AT89C51-12DM/883	40D6	Military/883C Class B, Fully Compliant (-55°C to 125°C)			
		AT89C51-12LM/883	44L				
		AT89C51-16AC	44A		Commercial (0°C to 70°C)		
		AT89C51-16JC	44J				
		AT89C51-16PC	40P6				
		AT89C51-16QC	44Q				
20	5 V $\pm$ 20%	AT89C51-16AI	44A	Industrial (-40°C to 85°C)			
		AT89C51-16JI	44J				
		AT89C51-16PI	40P6				
		AT89C51-16QI	44Q				
		20	5 V $\pm$ 20%	AT89C51-16AA	44A	Automotive (-40°C to 125°C)	
				AT89C51-16JA	44J		
AT89C51-16PA	40P6						
AT89C51-16QA	44Q						
20	5 V $\pm$ 20%			AT89C51-20AC	44A		Commercial (0°C to 70°C)
				AT89C51-20JC	44J		
		AT89C51-20PC	40P6				
		AT89C51-20QC	44Q				
20	5 V $\pm$ 20%	AT89C51-20AI	44A	Industrial (-40°C to 85°C)			
		AT89C51-20JI	44J				
		AT89C51-20PI	40P6				
		AT89C51-20QI	44Q				

## Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5 V ± 20%	AT89C51-24AC AT89C51-24JC AT89C51-24PC AT89C51-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89C51-24AI AT89C51-24JI AT89C51-24PI AT89C51-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)



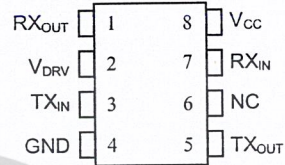
Package Type	
<b>44A</b>	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
<b>40D6</b>	40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
<b>44J</b>	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
<b>44L</b>	44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
<b>40P6</b>	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
<b>44Q</b>	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)



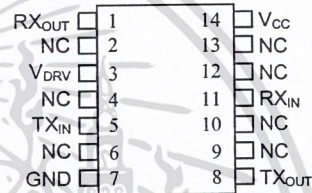
### FEATURES

- Low-power serial transmitter/receiver for battery-backed systems
- Transmitter steals power from receive signal line to save power
- Ultra-low static current, even when connected to RS-232-E port
- Variable transmitter level from +5 to +12 volts
- Compatible with RS-232-E signals
- Available in 8-pin, 150 mil wide SOIC package (DS275S)
- Low-power CMOS

### PIN ASSIGNMENT



DS275 8-Pin DIP (300-mil)  
 DS275 8-Pin SOIC (150-mil)



DS275E 14-Pin TSSOP

### ORDERING INFORMATION

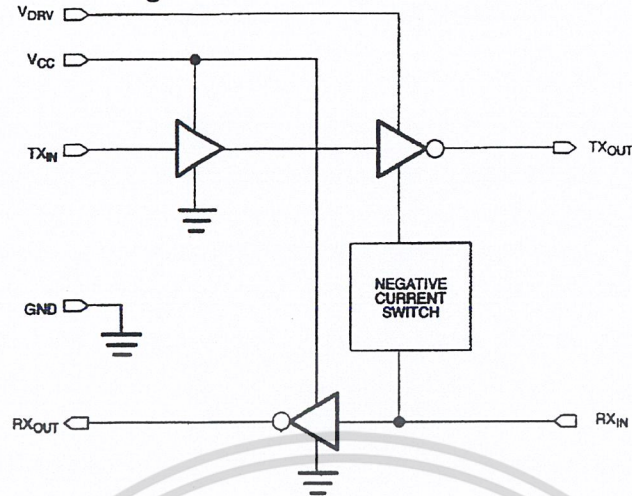
DS275	8-pin DIP
DS275S	8-pin SOIC
DS275E	14-pin TSSOP

### PIN DESCRIPTION

- RX<sub>OUT</sub> - RS-232 Receiver Output
- V<sub>DRV</sub> - Transmit driver +V
- TX<sub>IN</sub> - RS-232 Driver Input
- GND - System Ground (0V)
- TX<sub>OUT</sub> - RS-232 Driver Output
- NC - No Connection
- RX<sub>IN</sub> - RS-232 Receive Input
- V<sub>CC</sub> - System Logic Supply (+5V)

### DESCRIPTION

The DS275 Line-Powered RS-232 Transceiver Chip is a CMOS device that provides a low-cost, very low-power interface to RS-232 serial ports. The receiver input translates RS-232 signal levels to common CMOS/TTL levels. The transmitter employs a unique circuit which steals current from the receive RS-232 signal when that signal is in a negative state (marking). Since most serial communication ports remain in a negative state statically, using the receive signal for negative power greatly reduces the DS275's static power consumption. This feature is especially important for battery-powered systems such as laptop computers, remote sensors, and portable medical instruments. During an actual communication session, the DS275's transmitter will use system power (5-12 volts) for positive transitions while still employing the receive signal for negative transitions.

**DS275 BLOCK DIAGRAM Figure 1****OPERATION**

Designed for the unique requirements of battery-backed systems, the DS275 provides a low-power half-duplex interface to an RS-232 serial port. Typically, a designer must use an RS-232 device which uses system power during both negative and positive transitions of the transmit signal to the RS-232 port. If the connector to the RS-232 port is left connected for an appreciable time after the communication session has ended, power will statically flow into that port, draining the battery capacity. The DS275 eliminates this static current drain by stealing current from the receive line (RXIN) of the RS-232 port when that line is at a negative level (marking). Since most asynchronous communication over an RS-232 connection typically remains in a marking state when data is not being sent, the DS275 will not consume system power in this condition. System power would only be used when positive-going transitions are needed on the transmit RS-232 output (TXOUT) when data is sent. However, since synchronous communication sessions typically exhibit a very low duty-cycle, overall system power consumption remains low.

**RECEIVER SECTION**

The RXIN pin is the receive input for an RS-232 signal whose levels can range from  $\pm 3$  to  $\pm 15$  volts. A negative data signal is called a mark while a positive data signal is called a space. These signals are inverted and then level-shifted to normal +5-volt CMOS/TTL logic levels. The logic output associated with RXIN is RXOUT which swings from +VCC to ground. Therefore, a mark on RXIN produces a logic 1 at RXOUT; a space produces a logic 0.

The input threshold of RXIN is typically around 1.8 volts with 500 millivolts of hysteresis to improve noise rejection. Therefore, an input positive-going signal must exceed 1.8 volts to cause RXOUT to switch states. A negative-going signal must now be lower than 1.3 volts (typically) to cause RXOUT to switch again. An open on RXIN is interpreted as a mark, producing a logic 1 at RXOUT.

**TRANSMITTER SECTION**

TXIN is the CMOS/TTL-compatible input for digital data from the user system. A logic 1 at TXIN produces a mark (negative data signal) at TXOUT while a logic 0 produces a space (positive data signal). As mentioned earlier, the transmitter section employs a unique driver design that uses the RXIN line for swinging to negative levels. The RXIN line must be in a marking or idle state to take advantage of this design; if RXIN is in a spacing state, TXOUT will only swing to ground. When TXOUT needs to transition to a positive level, it uses the VDRV power pin for this level. VDRV can be a voltage supply between 5 to 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อที่ 2 of 8 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

volts, and in many situations it can be tied directly to the +5 volt  $V_{CC}$  supply. *It is important to note that  $V_{DRV}$  must be greater than or equal to  $V_{CC}$  at all times.*

The voltage range on  $V_{DRV}$  permits the use of a 9-volt battery in order to provide a higher voltage level when TXOUT is in a space state. When  $V_{CC}$  is shut off to the DS275 and  $V_{DRV}$  is still powered (as might happen in a battery-backed condition), only a small leakage current (about 50-100 nA) will be drawn. If TXOUT is loaded during such a condition,  $V_{DRV}$  will draw current only if RXIN is not in a negative state. During normal operation ( $V_{CC}=5$  volts),  $V_{DRV}$  will draw less than 2 uA when TXOUT is marking. Of course, when TXOUT is spacing,  $V_{DRV}$  will draw substantially more current—about 3 mA, depending upon its voltage and the impedance that TXOUT sees.

The TXOUT output is slew rate-limited to less than 30 volts/us in accordance with RS-232 specifications. In the event TXOUT should be inadvertently shorted to ground, internal current-limiting circuitry prevents damage, even if continuously shorted.

## RS-232 COMPATIBILITY

The intent of the DS275 is not so much to meet all the requirements of the RS-232 specification as to offer a low-power solution that will work with most RS-232 ports with a connector length of less than 10 feet. As a prime example, the DS275 will not meet the RS-232 requirement that the signal levels be at least  $\pm 5$  volts minimum when terminated by a 3 k $\Omega$  load and  $V_{DRV} = +5$  volts. Typically a voltage of 4 volts will be present at TXOUT when spacing. However, since most RS-232 receivers will correctly interpret any voltage over 2 volts as a space, there will be no problem transmitting data.

## APPLICATIONS INFORMATION

The DS275 is designed as a low-cost, RS-232-E interface expressly tailored for the unique requirements of battery-operated handheld products. As shown in the electrical specifications, the DS275 draws exceptionally low operating and static current. During normal operation when data from the handheld system is sent from the TXOUT output, the DS275 only draws significant  $V_{DRV}$  current when TXOUT transitions positively (spacing). This current flows primarily into the RS-232 receiver's 3-7 k $\Omega$  load at the other end of the attaching cable. When TXOUT is marking (a negative data signal), the  $V_{DRV}$  current falls dramatically since the negative voltage is provided by the transmit signal from the other end of the cable. This represents a large reduction in overall operating current, since typical RS-232 interface chips use charge-pump circuits to establish both positive and negative levels at the transmit driver output.

To obtain the lowest power consumption from the DS275, observe the following guidelines. First, to minimize  $V_{DRV}$  current when connected to an RS-232 port, always maintain TXIN at a logic 1 when data is not being transmitted (idle state). This will force TXOUT into the marking state, minimizing  $V_{DRV}$  current. Second,  $V_{DRV}$  current will drop to less than 100 nA when  $V_{CC}$  is grounded. Therefore, if  $V_{DRV}$  is tied directly to the system battery, the logic +5 volts can be turned off to achieve the lowest possible power state.

## FULL-DUPLEX OPERATION

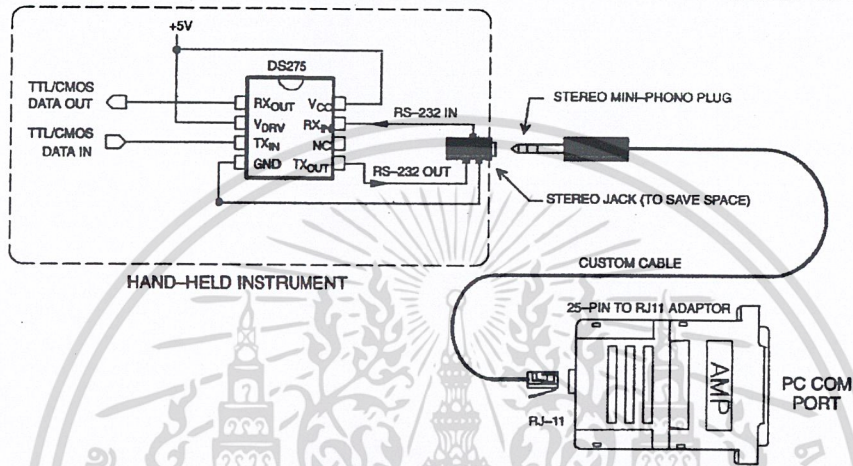
The DS275 is intended primarily for half-duplex operation; that is, RXIN should remain idle in the marking state when transmitting data out TXOUT and visa versa. However, the part can be operated full-duplex with most RS-232-E serial ports since signals swinging between 0 and +5V will usually be correctly interpreted by an RS-232-E receiver device. The 5-volt swing occurs when TXOUT attempts to swing negative while RXIN is at a positive voltage, which turns on an internal weak pulldown to ground for the TXOUT driver's negative reference. So, transmit mark signals at TXOUT may have voltage jumps from some negative value (corresponding to RXIN marking) to approximately ground. One possible

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

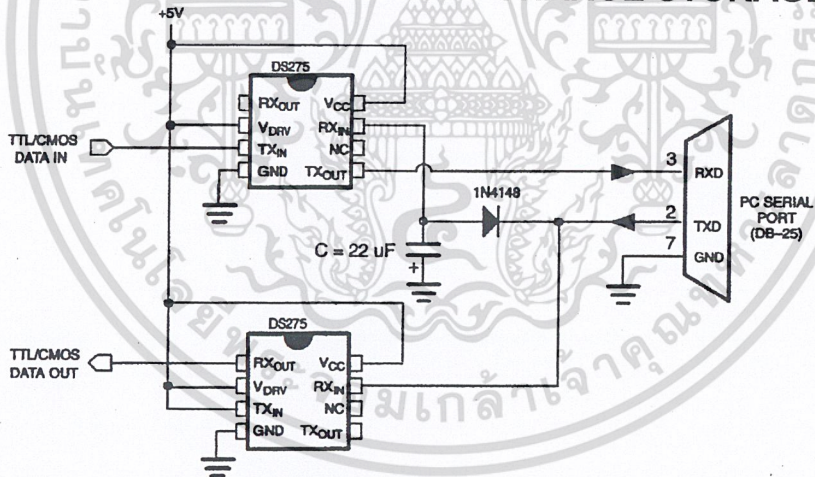
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

problem that may occur in this case is if the receiver at the other end requires a negative voltage for recognizing a mark. In this situation, the full-duplex circuit shown in Figure 3 can be used as an alternative. The 22  $\mu\text{F}$  capacitor forms a negative-charge reservoir; consequently, when the TXD line is spacing (positive), TXOUT still has a negative source available for a time period determined by the capacitor and the load resistance at the other end (3-7 k $\Omega$ ). This circuit was tested from 150-19,200 bps with error-free operation using a SN75154 Quad Line Receiver as the receiver for the TXout signal. Note that the SN75154 can have a marking input threshold below ground; hence there is the need for TXout to swing both positive and negative in full-duplex operation with this device.

**HANDHELD RS-232-C APPLICATION USING A STEREO MINI-JACK Figure 2**



**FULL-DUPLEX CIRCUIT USING NEGATIVE-CHARGE STORAGE Figure 3**



**NOTE:**

The capacitor stores negative charge whenever the TXD signal from the PC serial port is in a marking data state (a negative voltage that is typically -10 volts). The top DS275's TX<sub>OUT</sub> uses this negative charge reservoir when it is in a marking state. The capacitor will discharge to 0 volts when the TXD line is spacing (and TX<sub>OUT</sub> is still marking) at a time constant determined by its value and the value of the load resistance reflected back to TX<sub>OUT</sub>. However, when TXD is marking the capacitor will quickly charge back to -10 volts. Note that TXD remains in a marking state when idle, which improves the performance of this circuit.

**ABSOLUTE MAXIMUM RATINGS\***

- V<sub>CC</sub> -0.3 to +7.0 volts
- V<sub>DRV</sub> -0.3 to +13.0 volts

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RX <sub>IN</sub>	±15 volts
TX <sub>IN</sub>	-0.3 to V <sub>CC</sub> + 0.3 volts
TX <sub>OUT</sub>	±15 volts
RX <sub>OUT</sub>	-0.3 to V <sub>CC</sub> + 0.3 volts
Storage Temperature	-55°C to +125°C
Operating Temperature	0°C to 70°C

\* 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.

## RECOMMENDED DC OPERATING CONDITIONS

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Logic Supply	V <sub>CC</sub>	4.5	5.0	5.5	V	1
Transmit Driver Supply	V <sub>DRV</sub>	4.5	5-12	13.0	V	1
Logic 1 Input	V <sub>IH</sub>	2.0		V <sub>CC</sub> +0.3	V	2
Logic 0 Input	V <sub>IL</sub>	-0.3		+0.8	V	
RS-232 Input Range (RX <sub>IN</sub> )	V <sub>RS</sub>	-15		+15	V	
Dynamic Supply Current TX <sub>IN</sub> = V <sub>CC</sub>	I <sub>DRV1</sub>		400	800	μA	3
	I <sub>CC1</sub>		40	100	μA	
TX <sub>IN</sub> = GND	I <sub>DRV1</sub>		3.8	5.0	μA	
	I <sub>CC1</sub>		40	100	μA	
Static Supply Current TX <sub>IN</sub> = V <sub>CC</sub>	I <sub>DRV2</sub>		1.5	10.0	μA	4
	I <sub>CC2</sub>		10.0	15.0	μA	
TX <sub>IN</sub> = GND	I <sub>DRV2</sub>		3.8	5.0	mA	
	I <sub>CC2</sub>		10.0	20.0	μA	
Driver Leakage Current (V <sub>CC</sub> =0V)	I <sub>DRV3</sub>		0.05	1.0	μA	5

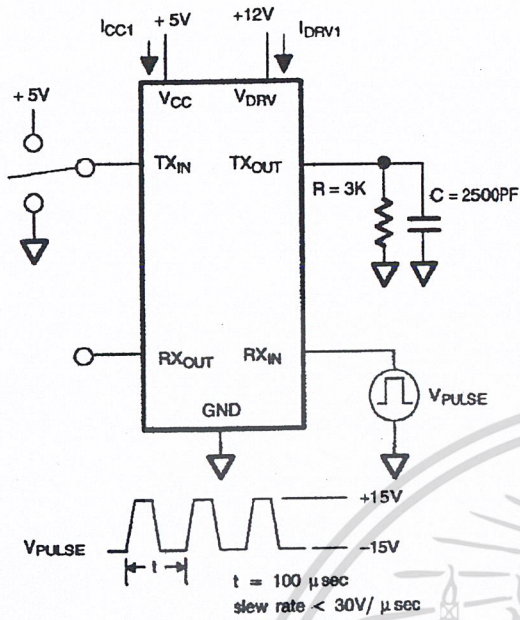
**DC ELECTRICAL CHARACTERISTICS** (0°C to 70°C;  $V_{CC} = V_{DRV} = 5V \pm 10\%$ )

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
TX <sub>OUT</sub> Level High	V <sub>OTXH</sub>	3.5	4.0	5.0	V	6
TX <sub>OUT</sub> Level Low	V <sub>OTXL</sub>	-8.5	-9.0		V	7
TX <sub>OUT</sub> Short Circuit Current	I <sub>SC</sub>		+60	+85	mA	
TX <sub>OUT</sub> Output Slew Rate	t <sub>SR</sub>			30	V/μs	
Propagation Delay	t <sub>PD</sub>		5		μs	8
RX <sub>IN</sub> Input Threshold Low	V <sub>TL</sub>	0.8	1.2	1.6	V	
RX <sub>IN</sub> Input Threshold High	V <sub>TH</sub>	1.6	2.0	2.4	V	
RX <sub>IN</sub> Threshold Hysteresis	V <sub>HYS</sub>	0.5	0.8		V	9
RX <sub>OUT</sub> Output Current @ 2.4V	I <sub>OH</sub>	-1.0			mA	
RX <sub>OUT</sub> Output Current @ 0.4V	I <sub>OL</sub>			3.2	mA	

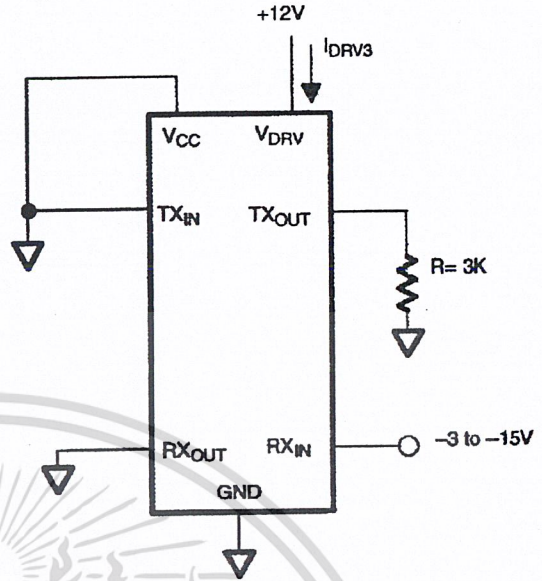
**NOTES:**

1. V<sub>DRV</sub> must be greater than or equal to V<sub>CC</sub>.
2. V<sub>CC</sub> = V<sub>DRV</sub> = 5V ± 10%.
3. See test circuit in Figure 4.
4. See test circuit in Figure 5.
5. See test circuit in Figure 6.
6. TX<sub>IN</sub> = V<sub>IL</sub> and TX<sub>OUT</sub> loaded by 3 kΩ to ground.
7. TX<sub>IN</sub> = V<sub>IH</sub>, RX<sub>IN</sub> = -10 volts and TX<sub>OUT</sub> loaded by 3 kΩ to ground.
8. TX<sub>IN</sub> to TX<sub>OUT</sub> - see Figure 7.
9. V<sub>HYS</sub> = V<sub>TH</sub> - V<sub>TL</sub>.

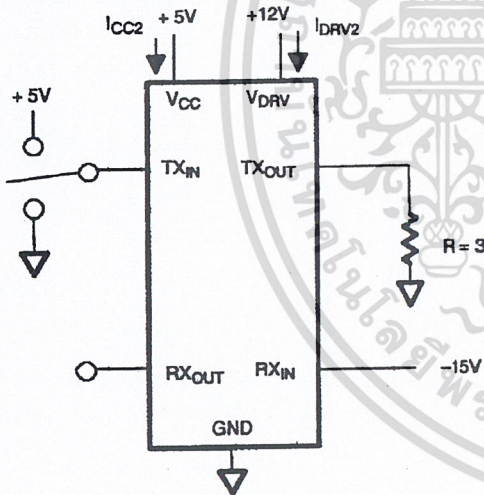
### DYNAMIC OPERATING CURRENT TEST CIRCUIT Figure 4



### DRIVER LEAKAGE TEST CIRCUIT Figure 6

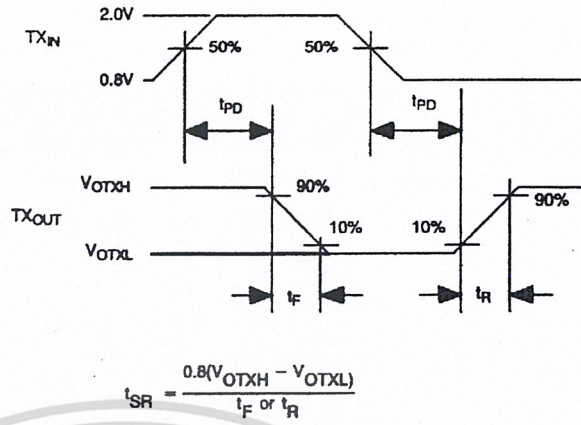
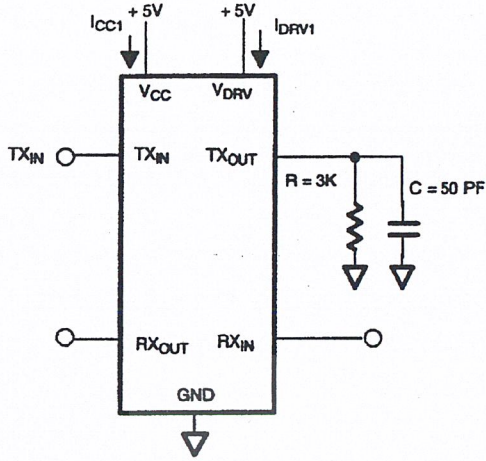


### STATIC OPERATING CURRENT TEST CIRCUIT Figure 5

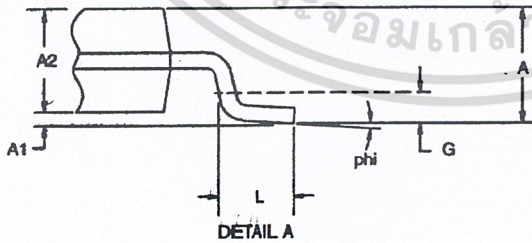
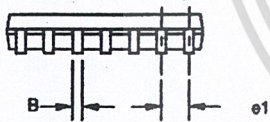
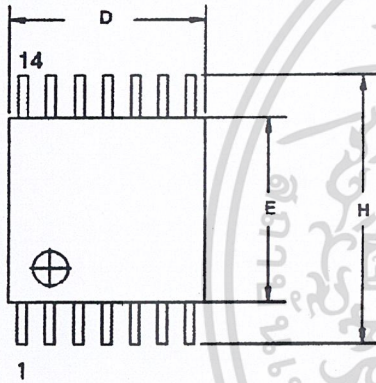


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROPAGATION DELAY TEST CIRCUIT Figure 7



DS275E 14-PIN TSSOP



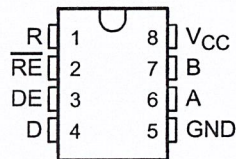
DIM	14-PIN	
	MIN	MAX
A MM	-	1.10
A1 MM	0.05	-
A2 MM	0.75	1.05
B MM	0.18	0.30
C MM	0.09	0.18
D MM	4.90	5.10
E MM	4.40 NOM	
e1 MM	0.65 BSC	
G MM	0.25 REF	
H MM	6.25	6.55
L MM	0.50	0.70
phi	0°	8°

# SN75176A DIFFERENTIAL BUS TRANSCEIVER

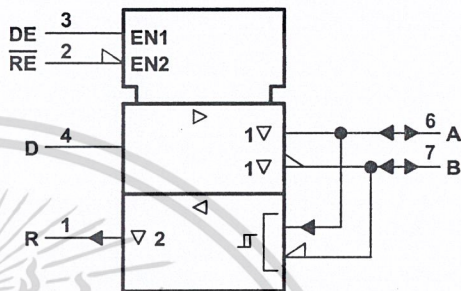
SLLS100A – JUNE 1984 – REVISED MAY 1995

- Bidirectional Transceiver
- Meets or Exceeds the Requirements of ANSI Standards EIA/TIA-422-B and ITU Recommendation V.11
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Driver and Receiver Outputs
- Individual Driver and Receiver Enables
- Wide Positive and Negative Input/Output Bus Voltage Ranges
- Driver Output Capability . . .  $\pm 60$  mA Max
- Thermal-Shutdown Protection
- Driver Positive- and Negative-Current Limiting
- Receiver Input Impedance . . . 12 k $\Omega$  Min
- Receiver Input Sensitivity . . .  $\pm 200$  mV
- Receiver Input Hysteresis . . . 50 mV Typ
- Operates From Single 5-V Supply
- Low Power Requirements

D OR P PACKAGE  
(TOP VIEW)



logic symbol†



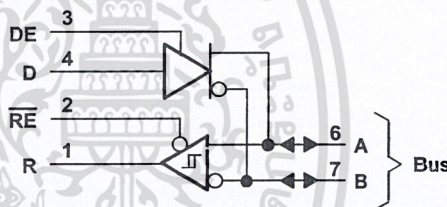
† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

## description

The SN75176A differential bus transceiver is a monolithic integrated circuit designed for bidirectional data communication on multipoint bus-transmission lines. It is designed for balanced transmission lines and meets ANSI Standard EIA/TIA-422-B and ITU Recommendation V.11.

The SN75176A combines a 3-state differential line driver and a differential input line receiver, both of which operate from a single 5-V power supply. The driver and receiver have active-high and active-low enables, respectively, that can be externally connected together to function as a direction control. The driver differential outputs and the receiver differential inputs are connected internally to form differential input/output (I/O) bus ports that are designed to offer minimum loading to the bus whenever the driver is disabled or  $V_{CC} = 0$ . These ports feature wide positive and negative common-mode voltage ranges making the device suitable for party-line applications.

logic diagram (positive logic)



## Function Tables

DRIVER				RECEIVER		
INPUT D	ENABLE DE	OUTPUTS		DIFFERENTIAL INPUTS A - B	ENABLE RE	OUTPUT R
		A	B			
H	H	H	L	$V_{ID} \geq 0.2$ V	L	H
L	H	L	H	$-0.2$ V $< V_{ID} < 0.2$ V	L	?
X	L	Z	Z	$V_{ID} \leq -0.2$ V	L	L
				X	H	Z
				Open	L	?

H = high level, L = low level, ? = indeterminate, X = irrelevant, Z = high impedance (off)

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1995, Texas Instruments Incorporated

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ข้อมูลนี้แก่บุคคลอื่นโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A – JUNE 1984 – REVISED MAY 1995

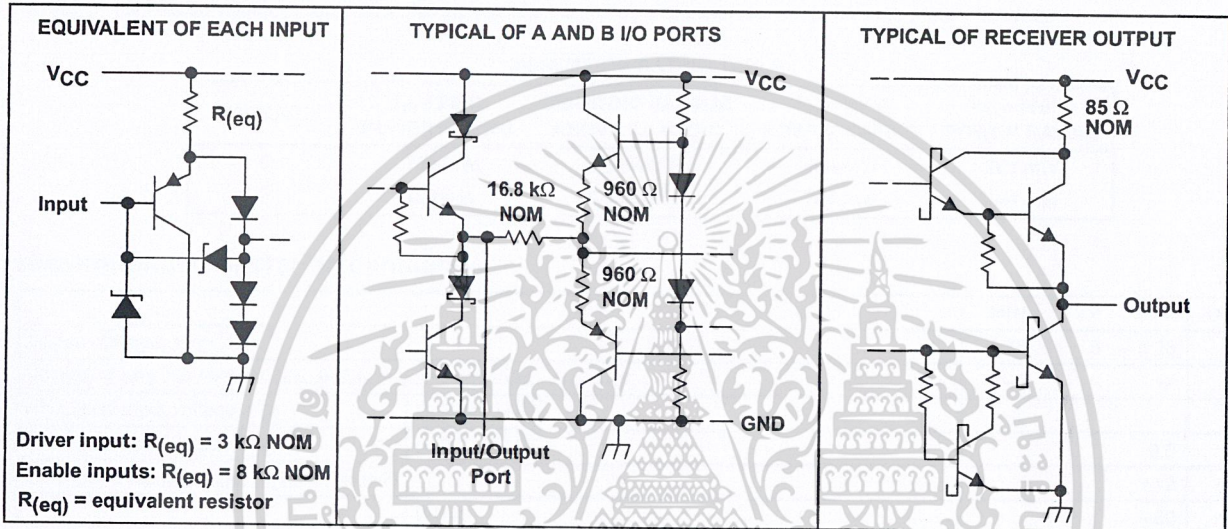
## description (continued)

The driver is designed to handle loads up to 60 mA of sink or source current. The driver features positive- and negative-current limiting and thermal shutdown for protection from line fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 150°C. The receiver features a minimum input impedance of 12 kΩ, an input sensitivity of ±200 mV, and a typical input hysteresis of 50 mV.

The SN75176A can be used in transmission-line applications employing the SN75172 and SN75174 quadruple differential line drivers and SN75173 and SN75175 quadruple differential line receivers.

The SN75176A is characterized for operation from 0°C to 70°C.

## schematics of inputs and outputs



# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A—JUNE 1984—REVISED MAY 1995

## absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, $V_{CC}$ (see Note 1)	7 V
Voltage range at any bus terminal	-10 V to 15 V
Enable input voltage, $V_I$	5.5 V
Continuous total power dissipation	See Dissipation Rating Table
Operating free-air temperature range, $T_A$	0°C to 70°C
Storage temperature range, $T_{stg}$	-65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values, except differential input/output bus voltage, are with respect to network ground terminal.

DISSIPATION RATING TABLE

PACKAGE	$T_A \leq 25^\circ\text{C}$ POWER RATING	DERATING FACTOR ABOVE $T_A = 25^\circ\text{C}$	$T_A = 70^\circ\text{C}$ POWER RATING	$T_A = 105^\circ\text{C}$ POWER RATING
D	725 mW	5.8 mW/°C	464 mW	261 mW
P	1100 mW	8.8 mW/°C	704 mW	396 mW

## recommended operating conditions

		MIN	TYP	MAX	UNIT
Supply voltage, $V_{CC}$		4.75	5	5.25	V
Voltage at any bus terminal (separately or common mode), $V_I$ or $V_{IC}$		-7		12	V
High-level input voltage, $V_{IH}$	D, DE, and $\overline{RE}$	2			V
Low-level input voltage, $V_{IL}$	D, DE, and $\overline{RE}$			0.8	V
Differential input voltage, $V_{ID}$ (see Note 2)				±12	V
High-level output current, $I_{OH}$	Driver			-60	mA
	Receiver			-400	µA
Low-level output current, $I_{OL}$	Driver			60	mA
	Receiver			8	
Operating free-air temperature, $T_A$		0		70	°C

NOTE 2: Differential-input/output bus voltage is measured at the noninverting terminal A with respect to the inverting terminal B.

# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A – JUNE 1984 – REVISED MAY 1995

## DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$V_{IK}$ Input clamp voltage	$I_I = -18 \text{ mA}$			-1.5	V
$V_{OH}$ High-level output voltage	$V_{IH} = 2 \text{ V}$ , $I_{OH} = -33 \text{ mA}$		3.7		V
$V_{OL}$ Low-level output voltage	$V_{IH} = 2 \text{ V}$ , $I_{OH} = 33 \text{ mA}$		1.1		V
$ V_{OD1} $ Differential output voltage	$I_O = 0$			$2V_{OD2}$	V
$ V_{OD2} $ Differential output voltage	$R_L = 100 \Omega$ , See Figure 1	2	2.7		V
	$R_L = 54 \Omega$ , See Figure 1	1.5	2.4		
$\Delta V_{OD} $ Change in magnitude of differential output voltage‡				$\pm 0.2$	V
$V_{OC}$ Common-mode output voltage§	$R_L = 54 \Omega$ or $100 \Omega$ , See Figure 1			3	V
$\Delta V_{OC} $ Change in magnitude of common-mode output voltage‡				$\pm 0.2$	V
$I_O$ Output current	Output disabled, See Note 3	$V_O = 12 \text{ V}$		1	mA
		$V_O = -7 \text{ V}$		-0.8	
$I_{IH}$ High-level input current	$V_I = 2.4 \text{ V}$			20	$\mu\text{A}$
$I_{IL}$ Low-level input current	$V_I = 0.4 \text{ V}$			-400	$\mu\text{A}$
$I_{OS}$ Short-circuit output current	$V_O = -7 \text{ V}$			-250	mA
	$V_O = V_{CC}$			250	
	$V_O = 12 \text{ V}$			500	
$I_{CC}$ Supply current (total package)	No load	Outputs enabled	35	50	mA
		Outputs disabled	26	40	

† All typical values are at  $V_{CC} = 5 \text{ V}$  and  $T_A = 25^\circ\text{C}$ .

‡  $\Delta|V_{OD}|$  and  $\Delta|V_{OC}|$  are the changes in magnitude of  $V_{OD}$  and  $V_{OC}$  respectively, that occur when the input is changed from a high level to a low level.

§ In ANSI Standard EIA/TIA-422-B,  $V_{OC}$ , which is the average of the two output voltages with respect to GND, is called output offset voltage,  $V_{OS}$ .

NOTE 3: This applies for both power on and off; refer to ANSI Standard EIA/TIA-422-B for exact conditions.

switching characteristics,  $V_{CC} = 5 \text{ V}$ ,  $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_d(\text{OD})$ Differential-output delay time	$R_L = 60 \Omega$ , See Figure 3		40	60	ns
$t_t(\text{OD})$ Differential-output transition time			65	95	
$t_{PZH}$ Output enable time to high level	$R_L = 110 \Omega$ , See Figure 4		55	90	ns
$t_{PZL}$ Output enable time to low level	$R_L = 110 \Omega$ , See Figure 5		30	50	ns
$t_{PHZ}$ Output disable time from high level	$R_L = 110 \Omega$ , See Figure 4		85	130	ns
$t_{PLZ}$ Output disable time from low level	$R_L = 110 \Omega$ , See Figure 5		20	40	ns



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A – JUNE 1984 – REVISED MAY 1995

## RECEIVER SECTION

electrical characteristics over recommended ranges of common-mode input voltage, supply voltage, and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
$V_{IT+}$	Positive-going input threshold voltage	$V_O = 2.7\text{ V}$ ,	$I_O = -0.4\text{ mA}$			0.2	V
$V_{IT-}$	Negative-going input threshold voltage	$V_O = 0.5\text{ V}$ ,	$I_O = 8\text{ mA}$	-0.2‡			V
$V_{hys}$	Input hysteresis voltage ( $V_{IT+} - V_{IT-}$ )				50		mV
$V_{IK}$	Enable clamp voltage	$I_I = -18\text{ mA}$				-1.5	V
$V_{OH}$	High-level output voltage	$V_{ID} = 200\text{ mV}$ , See Figure 2	$I_{OH} = -400\text{ }\mu\text{A}$ ,	2.7			V
$V_{OL}$	Low-level output voltage	$V_{ID} = -200\text{ mV}$ , See Figure 2	$I_{OL} = 8\text{ mA}$ ,			0.45	V
$I_{OZ}$	High-impedance-state output current	$V_O = 0.4\text{ V to }2.4\text{ V}$				$\pm 20$	$\mu\text{A}$
$I_I$	Line input current	Other input = 0 V, See Note 3	$V_I = 12\text{ V}$ $V_I = -7\text{ V}$			1 -0.8	mA
$I_{IH}$	High-level enable input current	$V_{IH} = 2.7\text{ V}$				20	$\mu\text{A}$
$I_{IL}$	Low-level enable input current	$V_{IL} = 0.4\text{ V}$				-100	$\mu\text{A}$
$r_i$	Input resistance			12			k $\Omega$
$I_{OS}$	Short-circuit output current			-15		-85	mA
$I_{CC}$	Supply current (total package)	No load	Outputs enabled Outputs disabled		35 26	50 40	mA

† All typical values are at  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$ .

‡ The algebraic convention, in which the less-positive (more-negative) limit is designated minimum, is used in this data sheet for common-mode input voltage and threshold voltage levels only.

NOTE 3: This applies for both power on and power off. Refer to ANSI Standard EIA/TIA-422-B for exact conditions.

switching characteristics,  $V_{CC} = 5\text{ V}$ ,  $C_L = 15\text{ pF}$ ,  $T_A = 25^\circ\text{C}$

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
$t_{PLH}$	Propagation delay time, low-to-high-level output	$V_{ID} = -1.5\text{ V to }1.5\text{ V}$ ,	See Figure 6		21	35	ns
$t_{PHL}$	Propagation delay time, high-to-low-level output				23	35	ns
$t_{PZH}$	Output enable time to high level	See Figure 7			10	30	ns
$t_{PZL}$	Output enable time to low level				12	30	ns
$t_{PHZ}$	Output disable time from high level	See Figure 7			20	35	ns
$t_{PLZ}$	Output disable time from low level				17	25	ns



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องส่งไปยังถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A – JUNE 1984 – REVISED MAY 1995

## PARAMETER MEASUREMENT INFORMATION

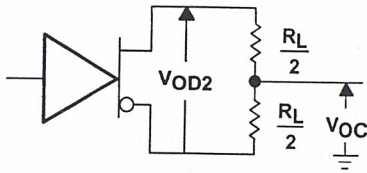


Figure 1. Driver  $V_{OD}$  and  $V_{OC}$

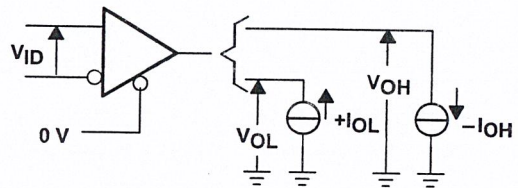


Figure 2. Receiver  $V_{OH}$  and  $V_{OL}$

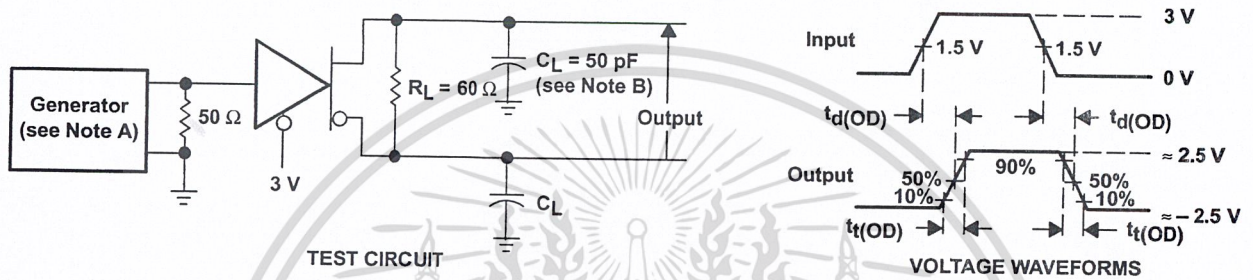


Figure 3. Driver Test Circuit and Voltage Waveforms

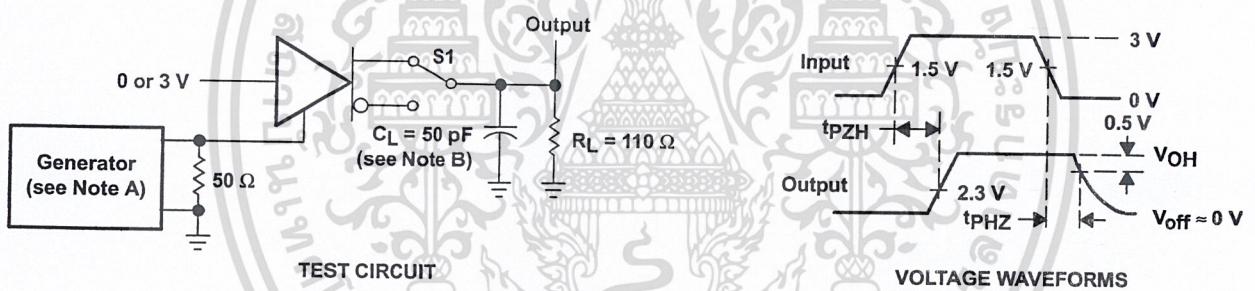


Figure 4. Driver Test Circuit and Voltage Waveforms

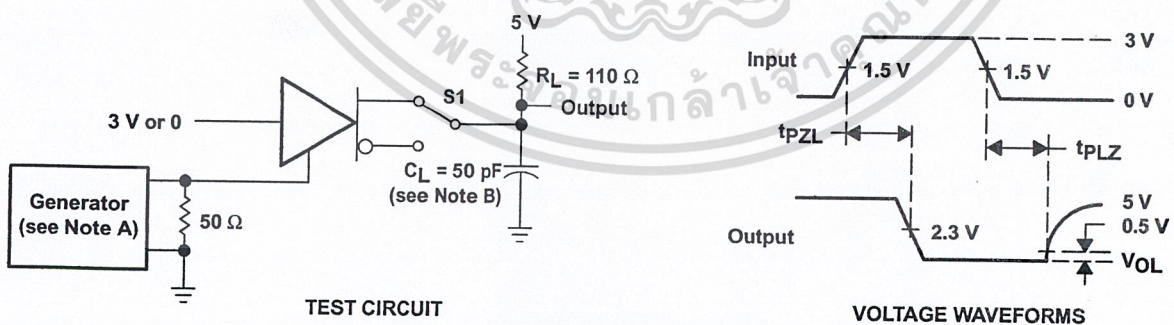


Figure 5. Driver Test Circuit and Voltage Waveforms

- NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR = 1 MHz, 50% duty cycle,  $t_r \leq 6$  ns,  $t_f \leq 6$  ns,  $Z_O = 50 \Omega$ .  
B.  $C_L$  includes probe and jig capacitance.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ข้อมูลนี้แก่บุคคลอื่นโดยไม่ได้รับอนุญาตจาก Texas Instruments

# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A—JUNE 1984—REVISED MAY 1995

## PARAMETER MEASUREMENT INFORMATION

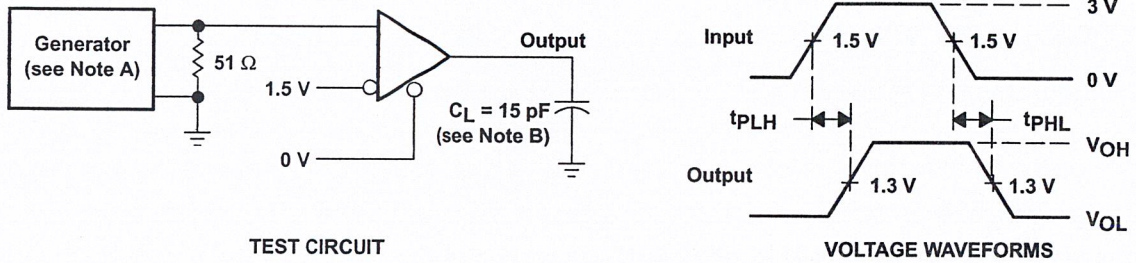


Figure 6. Receiver Test Circuit and Voltage Waveforms

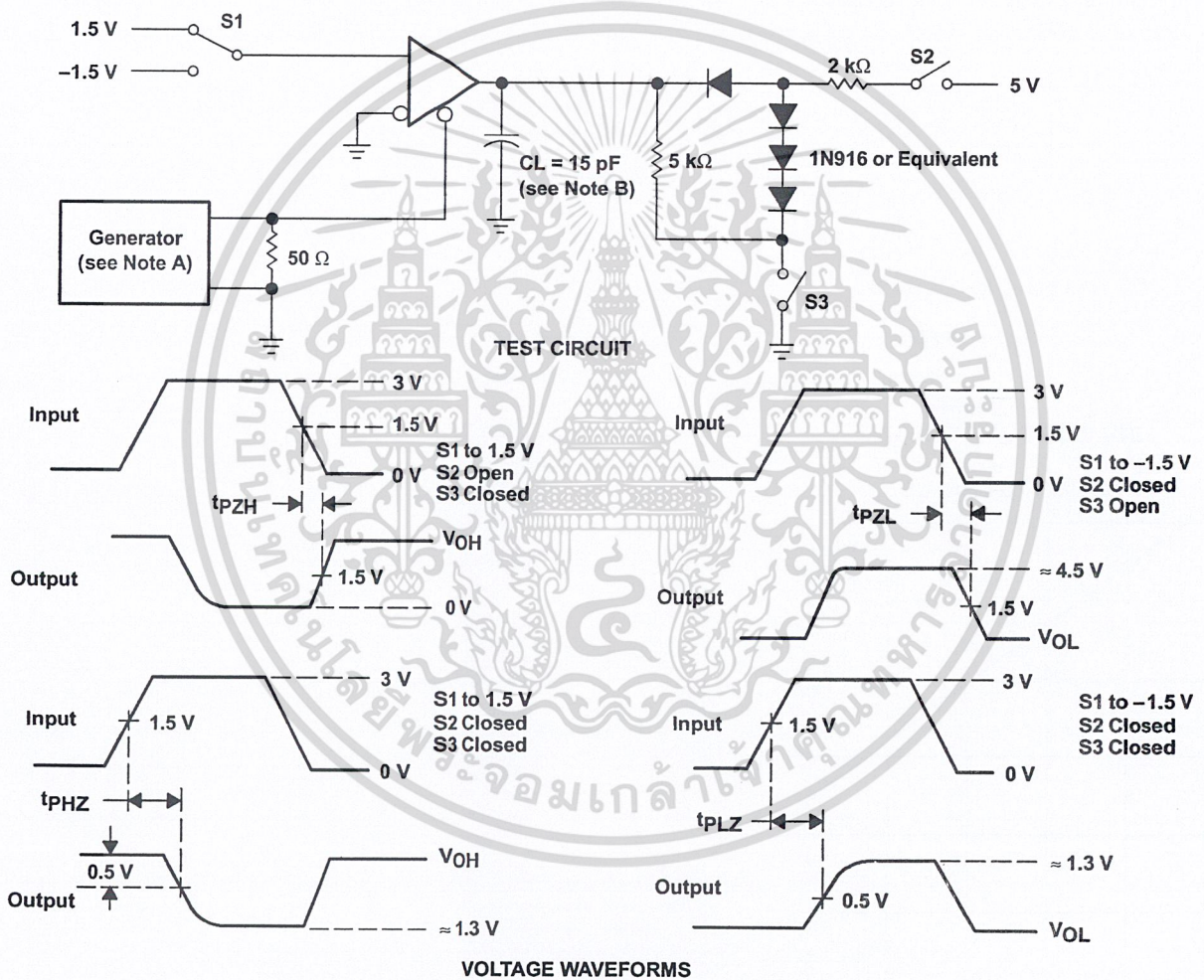


Figure 7. Receiver Test Circuit and Voltage Waveforms

- NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR = 1 MHz, 50% duty cycle,  $t_r \leq 6$  ns,  $t_f \leq 6$  ns,  $Z_0 = 50 \Omega$ .  
 B.  $C_L$  includes probe and jig capacitance.



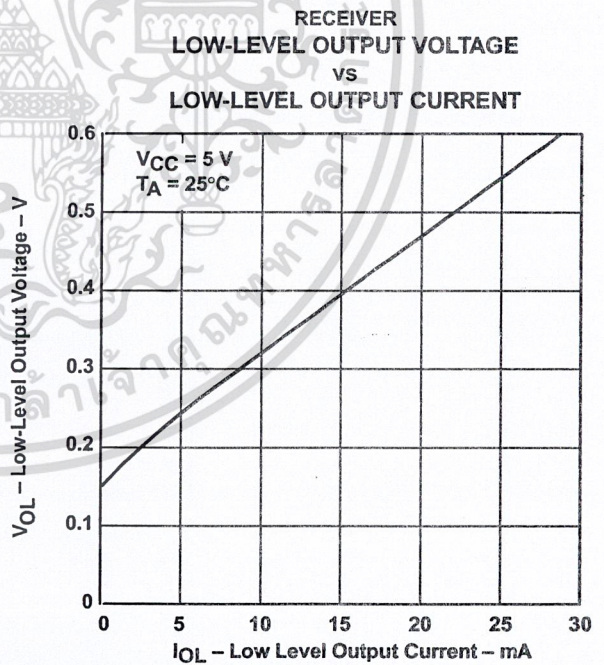
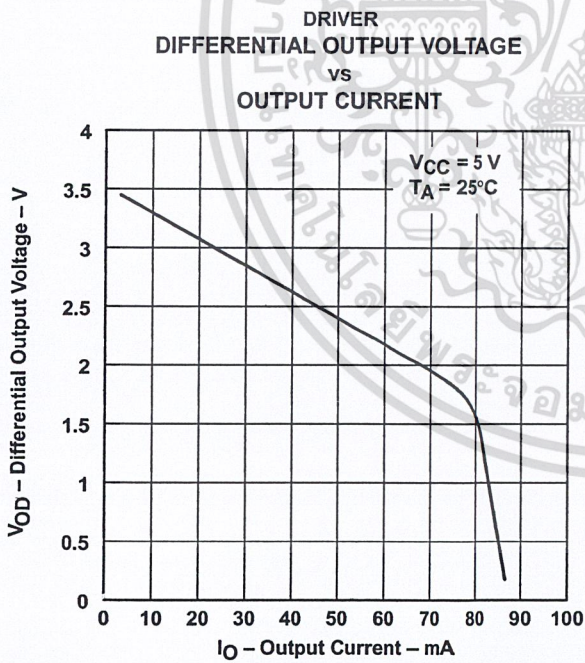
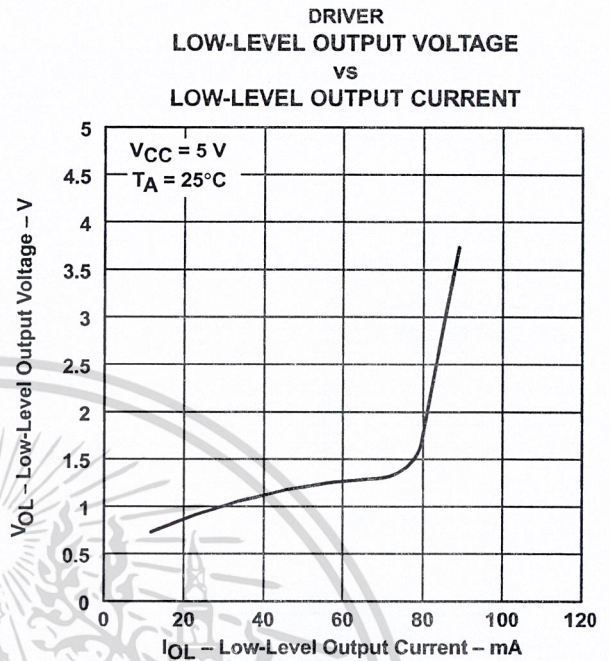
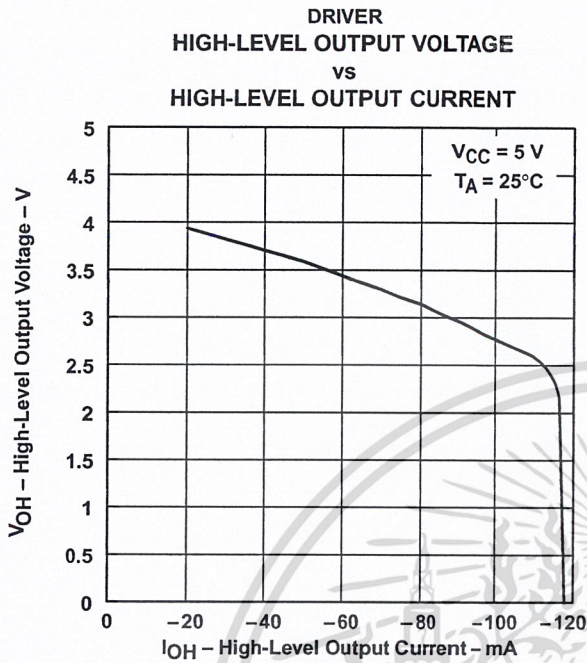
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตีแบบลงเนื้อหาและต้องขออนุญาตเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A—JUNE 1984—REVISED MAY 1995

## TYPICAL CHARACTERISTICS



TYPICAL CHARACTERISTICS

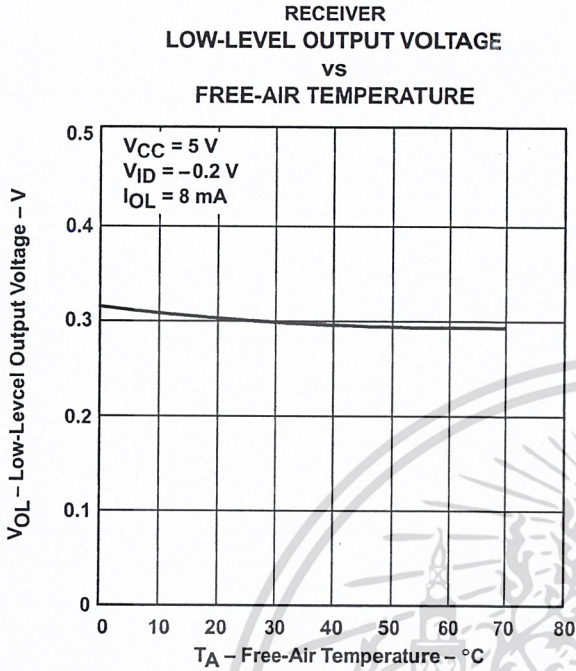


Figure 12

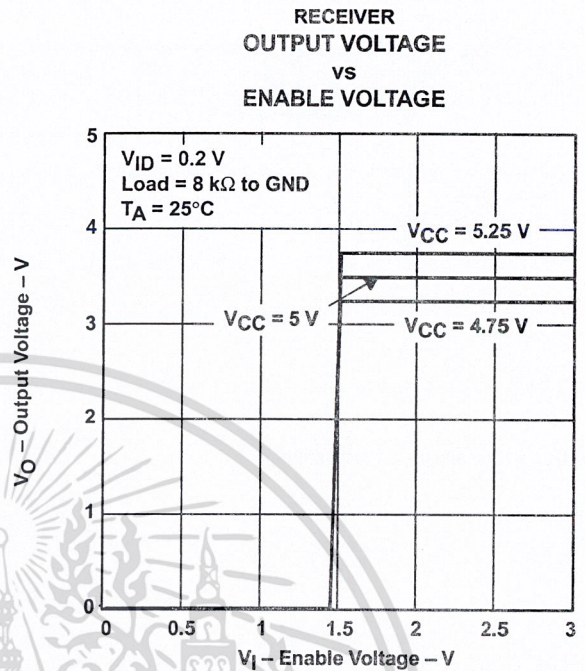


Figure 13

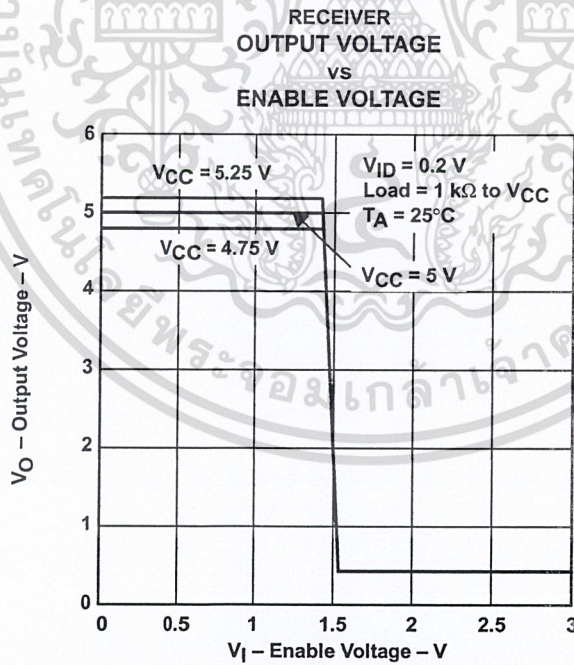


Figure 14



# SN75176A DIFFERENTIAL BUS TRANSCEIVER

SLLS100A – JUNE 1984 – REVISED MAY 1995

## APPLICATION INFORMATION

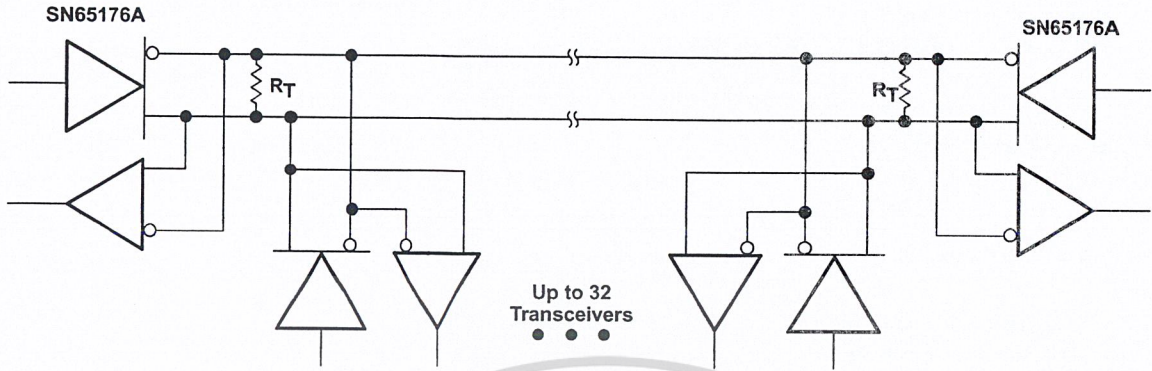


Figure 15. Typical Application Circuit

NOTE A: The line should be terminated at both ends in its characteristic impedance ( $R_T = Z_0$ ). Stub lengths off the main line should be kept as short as possible.



## IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Copyright © 1995, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้