

การแสดงผลญาณไฟฟ้าผ่านทางเครือข่ายอินเทอร์เน็ต
DISPLAYING OF ELECTRIC SIGNAL BY INTERNET



นายสุกฤษ

ลี้ตระกูลพิชิตชัย

นายสิทธิพงษ์

อุดม

เลขหมู่.....
เลขทะเบียน 45847
วัน, เดือน, ปี 19 ก.พ. 2546

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

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DISPLAYING OF ELECTRIC SIGNAL BY INTERNET

SUPPALERK LEETRAKOOLPHICHITTHAI
SITTIPONG AUDCHUM

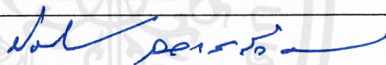
**A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ การแสดงสัญญาณไฟฟ้าผ่านทางเครือข่ายอินเทอร์เน็ต
DISPLAYING OF ELECTRIC SIGNAL BY INTERNET
นักศึกษาผู้จัดทำ นายศุภฤกษ์ ลีตระกูลพิชิตชัย รหัสประจำตัว 42015404
 นายสิทธิพงษ์ อุดชุม รหัสประจำตัว 42015410
ปริญญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2544

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ. พิพัฒน์ เลาหสงคราม	

วัน/เดือน/ปี ที่สอบ วันที่ 27 มีนาคม พ.ศ. 2545
สถานที่สอบ ณ. ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว



(ผศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การแสดงผลสัญญาณไฟฟ้าผ่านทางเครือข่ายอินเทอร์เน็ต
	DISPLAYING OF ELECTRIC SIGNAL BY INTERNET
นักศึกษาผู้จัดทำ	นายศุภฤกษ์ ถีระกุลพิชิตชัย
	นายสิทธิพงษ์ อุดชุม
อาจารย์ที่ปรึกษา	รศ.พิพัฒน์ เลหาสงคราม
ปีการศึกษา	2544

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการนำเสนอการแสดงผลสัญญาณไฟฟ้าผ่านทางเครือข่ายอินเทอร์เน็ต โดยใช้คอมพิวเตอร์เชื่อมต่อกับชุดอ่านสัญญาณ เพื่อนำสัญญาณที่อ่านได้ไปแสดงผ่านทางเครือข่ายอินเทอร์เน็ตโดยใช้โปรแกรมที่สร้างขึ้นเฉพาะ โดยที่สัญญาณอินพุตเป็นสัญญาณอนาล็อก แล้วทำการแปลงสัญญาณอนาล็อกนั้นเป็นสัญญาณดิจิทัล โดยใช้ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล และการเรียกดูสัญญาณไฟฟ้านี้ก็สามารถดูได้จากเครื่องคอมพิวเตอร์ใดก็ได้ที่เชื่อมต่อกับระบบเน็ตเวิร์ค

Thesis Title	Displaying of Electric signal by Internet
Authors	Mr. Suparurk Leetrakulphichitchai Mr. Sittipong Audchum
Thesis Advisor	Assoc.Prof.Phiphat Laohasongkram
Year	2001

ABSTRACT

This project presents the Displaying of Electric signal by Internet system with used computer link with reader signal set for tranlate signal to show signal via internet. The Displaying of Electric Signal by Internet program is specific program. Input signal are analog signals. After that converted analog signal to digital signal with Analog Digital Convector. The signal can be monitored in other computer terminals by connecting the server into the network.

กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณอาจารย์ประจำภาควิชาวิศวกรรมการวัดคุมทุกท่าน โดยเฉพาะอย่างยิ่ง รศ.พิพัฒน์ เลาหสงคราม ซึ่งเป็นอาจารย์ที่ปรึกษาที่ให้คำปรึกษาและคำแนะนำต่างๆ จนช่วยให้การทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

และขอกราบขอบพระคุณบิดา มารดา ที่เป็นที่เคารพยิ่งที่เป็นกำลังใจที่ดีตลอดมา รวมทั้งส่งเสียจนได้ร่ำเรียนถึงทุกวันนี้ และขอขอบคุณเพื่อนพ้องน้องพี่ที่ให้กำลังใจ ให้ความช่วยเหลือ ให้ความรัก ความเมตตาปรานี จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จดังสมประสงค์

สุดท้ายนี้ขอขอบคุณ ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้โอกาสได้ทำปริญญานิพนธ์เรื่องนี้ รวมทั้งมอบสิ่งดีๆ ตลอดระยะเวลาสามปีที่ผ่านมาศึกษาอยู่ในสถาบันแห่งนี้ ขอขอบคุณหอสมุดกลางและห้องสมุดคณะวิศวกรรมศาสตร์ที่เป็นแหล่งค้นคว้าข้อมูล ขอขอบคุณผู้แต่งหนังสือทุกท่านที่แต่งหนังสือดีๆ ให้เราอ่าน คุณความดีใดที่ปรากฏในการทำปริญญานิพนธ์ฉบับนี้ ผู้จัดทำขอมอบแด่ บิดา-มารดา ครู-อาจารย์ และผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	2
1.3 ขอบเขตของปริญญานิพนธ์.....	2
บทที่ 2 อินเทอร์เน็ต.....	3
2.1 อินเทอร์เน็ต คืออะไร.....	3
2.2 World Wide Web (WWW).....	3
2.3 Hypertext Transfer Protocol (HTTP).....	4
2.4 File Transfer Protocol (FTP).....	4
2.5 Uniform Resource Locator (URL).....	4
2.6 TCP/ IP.....	4
2.7 IP Address.....	5
2.8 DNS.....	5
2.9 เว็บเซิร์ฟเวอร์ (Web Server).....	6
2.10 PWS.....	7
2.11 HTML.....	7
2.12 Internet Client และ Server.....	7

สารบัญ (ต่อ)

	หน้า
บทที่ 3 Analog to Digital Converter	8
3.1 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล	8
3.2 หลักการทำงานของ A/D คอนเวอร์เตอร์	9
3.3 ความผิดพลาดของ A/D Converter	10
3.4 วงจรของ A/D Converter	11
3.5 แรมป์เอาต์พุตของ A/D Converter	12
3.6 ซีตเซสซีฟแอปพลิเคชัน	13
 บทที่ 4 ไมโครคอนโทรลเลอร์.....	 16
4.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	16
4.2 สถาปัตยกรรม 15 ไมโครคอนโทรลเลอร์ MCS-51	16
4.3 ไทม์เมอร์ / เคาน์เตอร์	20
4.4 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51	22
4.5 รีจิสเตอร์ฟังก์ชันพิเศษใน MCS-51	22
4.6 วงจรปรับระดับสัญญาณ	23
4.7 บิตพาร์ตี	24
 บทที่ 5 พอร์ตอนุกรม RS232.....	 26
5.1 พอร์ตอนุกรม.....	26
5.2 รูปแบบของเฟรมข้อมูล.....	29
 บทที่ 6 Microsoft Visual Basic	 31
6.1 สาเหตุที่ต้องใช้ Visual Basic.....	31
6.2 คอมพิวเตอร์ที่เหมาะสมสำหรับใช้งาน VB 6.....	32
6.3 คอนโทรลที่ใช้ในโครงการ	32
6.3.1 Label.....	32
6.3.2 TextBox.....	33
6.3.3 CommandButton.....	33

สารบัญ (ต่อ)

	หน้า
6.3.4 Timer	34
6.4.5 Internet Comm Control	34
6.4.6 iSevenSegmentClockX	38
6.3.7 iPlotX	39
6.3.8 Microsoft Winsock Control	39
บทที่ 7 การติดตั้งและการใช้งานโปรแกรมด้านเซิร์ฟเวอร์	40
7.1 เว็บเซิร์ฟเวอร์	40
7.2 โปรแกรมทำงานที่เซิร์ฟเวอร์	41
บทที่ 8 การติดตั้งและการใช้งานโปรแกรมด้านไคลเอนท์	42
บทที่ 9 ผลการทดลอง	43
บทที่ 10 สรุปและวิจารณ์	45
บรรณานุกรม	46
ภาคผนวก	47

สารบัญตาราง

ตารางที่	หน้า
4.1 Serial Port Control Register.....	19
4.2 แสดงโหมดต่าง ๆ ของการรับส่งแบบอนุกรม.....	20
4.3 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Timer 0	21
4.4 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ กับ Counter	21
4.5 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ กับ Timer 1	21
4.6 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ กับ Counter 1	22
4.7 รีจิสเตอร์ใช้งานเฉพาะ PSW (Program Status Word) เข้าถึงข้อมูลได้ในระดับบิต	22
6.1 Property ที่สำคัญของ Label.....	32
6.2 Property ที่สำคัญของ Textbox	33
6.3 Property ที่สำคัญของ CommandButton	33

สารบัญภาพ

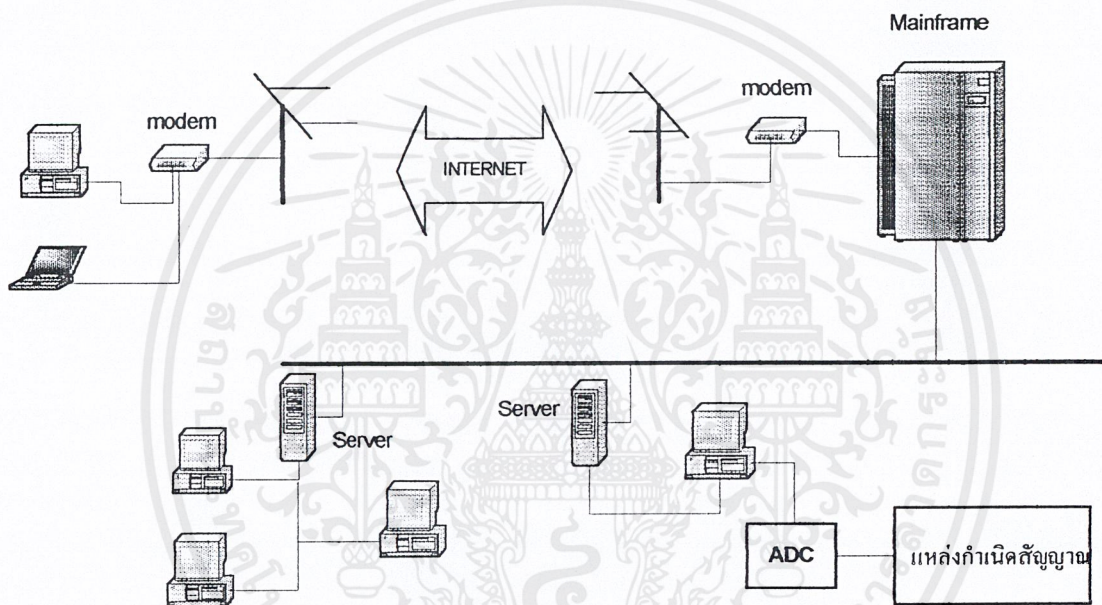
ภาพที่	หน้า
1.1 แสดงการเชื่อมต่อเซิร์ฟเวอร์กับเครือข่ายอินเทอร์เน็ต	1
1.2 บล็อกไดอะแกรมการทำงานด้านไคลเอนท์.....	2
1.3 บล็อกไดอะแกรมการทำงานด้านเซิร์ฟเวอร์	2
2.1 เครือข่ายอินเทอร์เน็ตที่เชื่อมโยงเครือข่ายขนาดเล็กไว้ด้วยกัน	3
2.2 แสดงลักษณะของโปรแกรม Personal Web Server	6
2.3 แสดงลักษณะการทำงานของ Client & Server	7
3.1 แสดงสัญญาณของ A/D คอนเวอร์เตอร์	8
3.2 คุณสมบัติของ A/D Converter แบบ 4 บิต	8
3.3 แสดงการตอบสนองของเวลาแปลงสัญญาณของ A/D Converter.....	9
3.4 ขบวนการแปลงสัญญาณดิจิทัลด้วย A/D Converter.....	10
3.5 ข้อผิดพลาดของ A/D Converter	11
3.6 A/D คอนเวอร์เตอร์	12
3.7 กราฟเอาต์พุตของส่วน D/A ของ A/D Converter.....	13
3.8 คอนเวอร์เตอร์ซึ่งใช้เทคนิค SAR.....	14
3.9 กราฟของส่วน D/A ระหว่างการแปลงสัญญาณ	14
4.1 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์ 8051	17
4.2 แสดงคุณสมบัติและลักษณะของสัญญาณแรงดันรูปไซน์ที่อินพุตและเอาต์พุตของ วงจรปรับระดับแรงดัน.....	23
4.3 วงจรที่ต่อใช้งานจริง.....	24
5.1 แสดงการเชื่อมต่อสายสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม.....	27
6.1 การเพิ่ม Component Microsoft Comm Control	34
6.2 รูปแสดงหลังการเพิ่ม Component MSComm.....	35
7.1 เว็บบ์เซิร์ฟเวอร์ที่ติดตั้งเรียบร้อยแล้ว.....	40
7.2 แสดงโปรแกรมที่ทำงานฝั่งเซิร์ฟเวอร์	41
8.1 โปรแกรมที่ทำงานที่ Client.....	42
9.1 ผลของรูปสัญญาณที่ได้ที่ฝั่งเซิร์ฟเวอร์	43
9.2 ผลของรูปสัญญาณที่ได้ที่ฝั่งไคลเอนท์.....	44

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญญานิพนธ์

ปัจจุบันเทคโนโลยีการสื่อสารผ่านทางเครือข่ายอินเทอร์เน็ตได้พัฒนาไปมาก การรับ-ส่งข้อมูลระยะไกลสามารถทำได้สะดวกและรวดเร็วขึ้น การพัฒนาในด้านต่างๆ จึงเจริญเติบโตขึ้นตามมาโดยลำดับ



ภาพที่ 1.1 แสดงการเชื่อมต่อเซิร์ฟเวอร์กับเครือข่ายอินเทอร์เน็ต

โครงการการแสดงผลสัญญาณไฟฟ้าผ่านทางเครือข่ายอินเทอร์เน็ตนี้เป็นการรับส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ต โดยค่าของสัญญาณไฟฟ้าจะถูกอ่านค่าโดย ADC (Analog to Digital Converter) และส่งให้กับไมโครคอนโทรลเลอร์รวบรวมเป็นชุดในหนึ่งช่วงเวลา แล้วจึงส่งให้กับคอมพิวเตอร์ผ่านทางพอร์ทอนุกรมมาตรฐาน RS-232 และที่เครื่องคอมพิวเตอร์จะมีโปรแกรมเฉพาะที่ทำงานอยู่ คอยรับค่าและนำมาเขียนเป็นไฟล์ข้อมูล เพื่อให้เครื่องคอมพิวเตอร์อื่นๆ สามารถเรียกไปดูและแสดงผลเป็นรูปกราฟสัญญาณได้ โดยอาจใช้โปรแกรมเฉพาะที่มีอยู่บนเครื่องคอมพิวเตอร์ไคลเอนท์ หรือเรียกใช้โปรแกรมผ่านทางระบบอินเทอร์เน็ตก็ได้

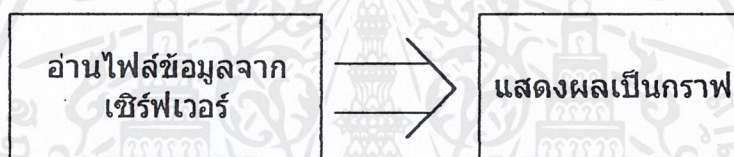
1.2 วัตถุประสงค์ของปริญญานิพนธ์

เพื่อที่จะสามารถดูการเปลี่ยนแปลงของสัญญาณไฟฟ้า ซึ่งเป็นสัญญาณอนาลอก ผ่านทางเครือข่ายอินเทอร์เน็ตและอินเทอร์เน็ต เพื่อนำค่าที่อ่านได้นั้นไปประยุกต์ใช้ประโยชน์ในการทำงานอื่นๆ ต่อไป

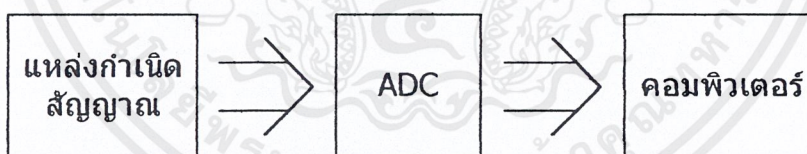
1.3 ขอบเขตของปริญญานิพนธ์

ทำให้เราเซอร์สามารถดูการเปลี่ยนแปลงของสัญญาณไฟฟ้าผ่านทางเครือข่ายอินเทอร์เน็ต โดยนำค่าของสัญญาณนั้นมาจากเครื่องคอมพิวเตอร์ที่หน้าที่เป็นเซิร์ฟเวอร์

บล็อกไดอะแกรมการทำงาน



ภาพที่ 1.2 บล็อกไดอะแกรมการทำงานด้านไคลเอนท์



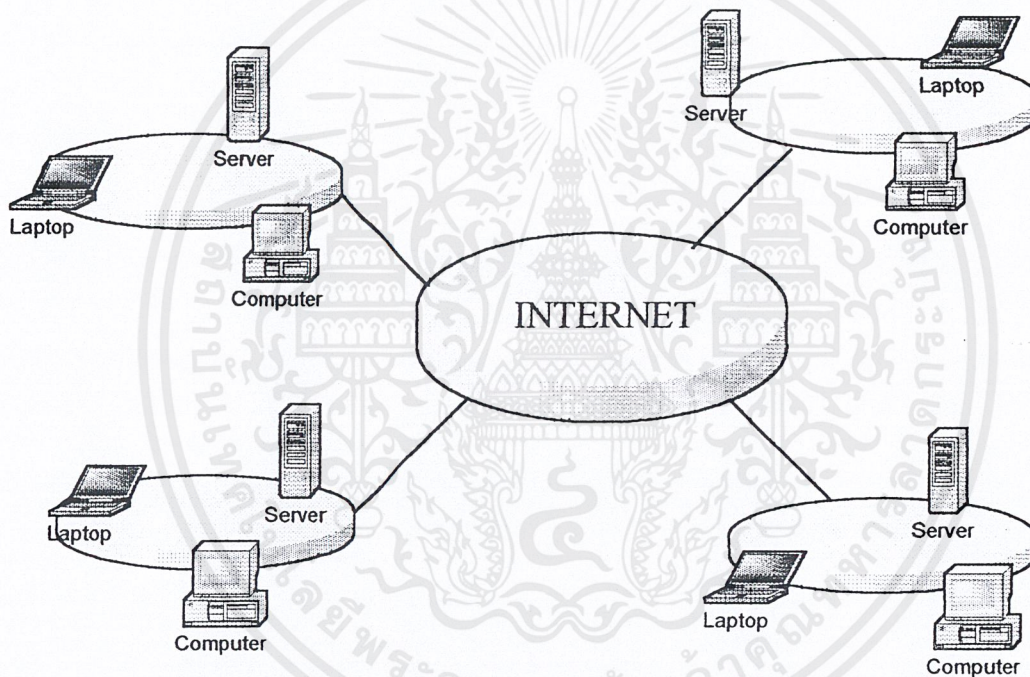
ภาพที่ 1.3 บล็อกไดอะแกรมการทำงานด้านเซิร์ฟเวอร์

บทที่ 2

อินเทอร์เน็ต (INTERNET)

2.1 อินเทอร์เน็ต คืออะไร

อินเทอร์เน็ต (INTERNET) คือ เครือข่ายคอมพิวเตอร์ขนาดใหญ่ที่เชื่อมต่อกันทั่วโลก โดยมีมาตรฐานการรับส่งข้อมูลที่เหมือนกัน โดยที่ข้อมูลเหล่านั้นอาจจะเป็นตัวอักษร, ภาพนิ่ง, ภาพเคลื่อนไหว หรือจะเป็นเสียงก็ได้ รวมทั้งยังมีความสามารถในการค้นหาข้อมูลที่อยู่ในแหล่งต่างๆ ทั่วโลกได้อย่างรวดเร็วและมีประสิทธิภาพ



ภาพที่ 2.1 เครือข่ายอินเทอร์เน็ตที่เชื่อมโยงเครือข่ายขนาดเล็กไว้ด้วยกัน

2.2 World Wide Web (WWW)

เป็นรูปแบบการให้บริการที่เป็นสภาพแวดล้อมที่เป็นกราฟฟิกที่แสดงเว็บเพจจากสถานที่ต่างๆ ซึ่งสามารถอ่านข้อมูล, ดาวน์โหลดไฟล์, ดูหนังฟังเพลง, เติมข้อมูลในฟอร์ม, โต้ตอบกับแอปพลิเคชัน (ที่เรียกว่า "applet" หรือ script) และค้นหาข้อมูล โดยแต่ละเว็บเพจจะมีแอดเดรส (address) เฉพาะที่ไม่เหมือนกัน ทำให้ผู้ใช้งานสามารถกำหนดหรือดูเว็บเพจได้ด้วยเว็บเบราว์เซอร์ (Web browser) แอดเดรสที่วนนี้เรียกว่า URL (Uniform Resource Locator) ซึ่งจะเริ่มต้นด้วย <http://>

เช่น [http:// www.microsoft.com](http://www.microsoft.com) เป็นแอดเดรสของบริษัทไมโครซอฟท์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Hypertext Transfer Protocol (HTTP)

เป็น Protocol ที่ใช้ติดต่อระหว่างเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server และ Client ผ่านทาง Hypertext Markup Language (HTML) โดยการอ้างถึงที่อยู่ (Address) ของ Protocol ประเภทนี้จะขึ้นต้นด้วยคำว่า “ http:// ”

2.4 File Transfer Protocol (FTP)

เป็นรูปแบบการให้บริการที่ให้ผู้ใช้งานสามารถรับ-ส่งไฟล์ (เรียกว่า คาวนด์โหลด (Download) และอัปโหลด (Upload) จากคอมพิวเตอร์หนึ่งไปยังอีกคอมพิวเตอร์หนึ่ง ส่วนมากเซิร์ฟเวอร์ของ FTP จะยอมให้ดาวน์โหลดหรืออัปโหลดเฉพาะสมาชิกเท่านั้น หรือบางเซิร์ฟเวอร์จะให้อิสระในการเข้าไปดาวน์โหลด เช่น www.shareware.com เป็นต้น

2.5 Uniform Resource Locator (URL)

ใช้สำหรับระบุ Path ของข้อมูลที่ต้องการเข้าถึง เช่น เอกสาร ภาพกราฟฟิค หรือไฟล์ ฯลฯ เป็นต้น ที่เก็บอยู่ในอินเทอร์เน็ต หรืออินทราเน็ต รวมถึงใช้ระบุประเภทของ Protocol ที่ใช้เข้าถึงข้อมูลที่ต้องการนั้น นอกจากนี้ยังอาจใช้ระบุที่อยู่ของ E-mail ได้เช่นเดียวกัน เช่น

<http://www.someones.homepage/default.html>

<ftp://ftp.server.somewhere/ftp.file>

<gopher://server.name>

<file://Server/Share/File.doc>

สำหรับรูปแบบของ URL มีดังนี้

Protocol :// host // localinfo

Protocol หมายถึง Protocol ที่ใช้ในการติดต่อ ซึ่งอาจเป็น HTTP หรือ FTP

Host หมายถึง ที่อยู่ปลายทางที่ต้องการติดต่อ

Localinfo หมายถึง Path และชื่อของไฟล์ที่ต้องการเข้าถึง

2.6 TCP/ IP

โปรโตคอล (Protocol) เป็นระเบียบวิธีในการติดต่อสื่อสารระหว่างคอมพิวเตอร์ที่ใช้งานร่วมกันในเครือข่ายเพื่อให้คอมพิวเตอร์ทั้งสองที่ติดต่อกันสามารถคุยกันได้ รวมทั้งการรับ-ส่งข้อมูลระหว่างกันได้อย่างไม่มีปัญหา ในปัจจุบันโปรโตคอลที่นิยมใช้กันมากที่สุดคือ TCP/ IP ซึ่งมีการใช้งานอย่างกว้างขวางมากในระบบปฏิบัติการ UNIX ที่เป็นระบบปฏิบัติการที่นิยมใช้อย่างแพร่หลาย จุดกำเนิดของโปรโตคอล TCP/ IP เริ่มขึ้นในราว พ.ศ. 2512 ที่กระทรวงกลาโหมของเอกราชเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สหรัฐฯ เมื่อพบปัญหาในการเชื่อมโยงเครือข่ายคอมพิวเตอร์ในหน่วยงานต่างๆ ของตน ซึ่งจะต้องมีการส่งข้อมูลระหว่างกันและไปยังหน่วยงานภายนอกอื่นๆ เช่น มหาวิทยาลัย, ห้างทดลองต่างๆ เนื่องจากแต่ละแห่งก็จะมีระบบคอมพิวเตอร์ของตนเองที่แตกต่างกันไป การเชื่อมต่อก็เป็นไปตามลักษณะต่างคนต่างทำไม่เหมือนกัน ดังนั้นการเชื่อมโยงข้อมูลเป็นไปด้วยความยากลำบาก หน่วยงาน ARPA จึงได้กำหนดมาตรฐานในการสื่อสารข้อมูลและได้จัดตั้งเครือข่าย ARPANET ขึ้นโดยใช้โปรโตคอล TCP/ IP ต่อมาก็กลายเป็นมาตรฐานจริงจิงในราวปี พ.ศ.2525 และใช้เป็นมาตรฐานสำหรับการเชื่อมต่อทางด้านอินเทอร์เน็ตในปัจจุบัน

2.7 IP Address

ในการติดต่อสื่อสารในระบบเครือข่ายอินเทอร์เน็ตจำเป็นต้องใช้โปรโตคอลมาตรฐานที่เรียกว่า TCP/ IP ทำให้จำเป็นต้องมีการกำหนดเลขประจำตัวให้กับคอมพิวเตอร์ที่เชื่อมต่อกันนั้นให้แตกต่างกันไป หมายเลขประจำตัวที่ว่่านี้เรียกว่า IP Address หรือหมายเลข IP ซึ่งจะมี 4 ชุด แต่ละชุดคั่นด้วยจุด ในลักษณะ xxx.xxx.xxx.xxx โดยที่ตัวเลขแต่ละชุดจะเป็นตัวเลข 8 บิต ดังนั้นในแต่ละชุดจะมีค่าตั้งแต่ 0 จนถึง $(2^8) - 1 = 255$ ซึ่งทำให้สามารถกำหนดหมายเลขให้กับเครื่องคอมพิวเตอร์ที่เชื่อมต่อกันได้ถึง 4,294,967,296 หมายเลข

หมายเลข IP จะกำหนดโดยหน่วยงานที่เรียกว่า InterNIC (Internet Network Information Center) ขององค์กร NSI (Network Solution Incorporation) ดังนั้นเมื่อมีการตั้งโฮสต์คอมพิวเตอร์เพื่อเชื่อมต่ออินเทอร์เน็ตจำเป็นต้องแจ้งไปยังหน่วยงานที่ว่่านี้ เพื่อจดทะเบียนและกำหนดหมายเลข IP ก็สามารถทำได้โดยตรงจาก ISP (Internet Service Provider) ที่ให้บริการอินเทอร์เน็ต โดย ISP จะกำหนดหมายเลขให้คุณเอง

2.8 DNS

เนื่องจากการกำหนดหมายเลขเครื่องคอมพิวเตอร์ในอินเทอร์เน็ตด้วยหมายเลข IP มีข้อเสียคือ จำได้ยากและทำให้สับสนได้ง่าย ดังนั้น จึงได้มีการพัฒนาวิธีการอ้างอิงหรือแทนหมายเลข IP เพื่อให้ใช้งานได้ง่ายและจำได้ง่ายไม่สับสน โดยการใช้ DNS (Domain Name Server) โดยที่เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Domain Name Server จะแปลงจากชื่อโดเมนเป็นหมายเลข IP อีกทีหนึ่ง สำหรับรูปแบบของโดเมนจะเป็นดังนี้

ชื่อโฮสต์คอมพิวเตอร์ .ชื่อโดเมนย่อย. [ชื่อโดเมนย่อย] .ชื่อโดเมนบนสุด

เช่น หมายเลข IP ของ kmitl.ac.th คือ 161.246.10.21 เป็นต้น

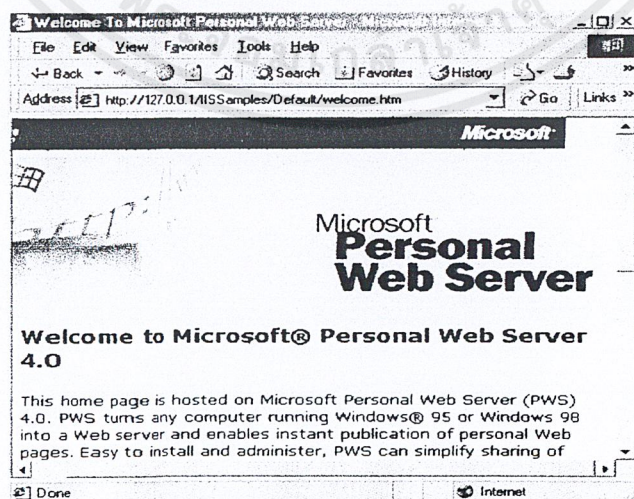
2.9 เว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์(Web Server) คือ แอปพลิเคชันที่ทำหน้าที่รับและประมวลผลข้อมูลที่ร้องขอจากผู้ใช้บริการอินเทอร์เน็ต โดยผ่านทางเว็บเบราว์เซอร์ หลังจากเว็บเบราว์เซอร์ รับคำสั่งและประมวลผลแล้ว (การประมวลผลอาจจะเป็นการคำนวณ ค้นหา หรือวิเคราะห์ข้อมูลก็ได้) ผลลัพธ์จะถูกส่งกลับไปยังผู้ใช้ โดยแสดงผลในเว็บเบราว์เซอร์นั่นเอง นอกจากเว็บเบราว์เซอร์ จะให้บริการในอินเทอร์เน็ตแล้ว ยังสามารถนำมาประยุกต์ใช้ในเครือข่ายภายในองค์กร หรืออินทราเน็ตได้อีกด้วย

เว็บเซิร์ฟเวอร์สามารถติดตั้งบนคอมพิวเตอร์ที่มีระบบปฏิบัติการต่างๆ เช่น คอมพิวเตอร์ระบบ UNIX คอมพิวเตอร์ภายใต้ Window 95/ Window 98 , Window NT หรือแม้แต่ Window 2000 เป็นต้น ซอฟต์แวร์ที่เป็นเว็บเซิร์ฟเวอร์ ได้แก่ โปรแกรมเน็ตสเคปเซิร์ฟเวอร์ (Netcape Server) ,โปรแกรมอินเทอร์เน็ตอินฟอร์เมชัน (IIS : Internet Information Server) , โปรแกรม PWS (Personal Web Server) เป็นต้น

2.10 PWS

PWS (Personal Web Server) เป็นเว็บเซิร์ฟเวอร์ที่สนับสนุนการใช้งานบนระบบปฏิบัติการ Window 95/ 98 ซึ่งอาจจะมีประสิทธิภาพไม่สูงนักแต่ก็สะดวกในการพัฒนางานเพื่อใช้ในการทดสอบก่อนการส่งไฟล์เอกสารขึ้นไปยังเว็บไซต์ และเนื่องจาก PWS สามารถทำงานได้บน Window 95/ 98 ทำให้สามารถจำลอง Window ใช้งานอยู่เป็นเซิร์ฟเวอร์ได้ทันที สำหรับซอฟต์แวร์ที่ใช้ในการติดตั้ง PWS จะอยู่ในแผ่นติดตั้ง Window 98 ในโฟลเดอร์ \add-ons\pws หรือดาวน์โหลดได้ฟรีจากเว็บไซต์ของไมโครซอฟต์



ภาพที่ 2.2 แสดงลักษณะของโปรแกรม Personal Web Server

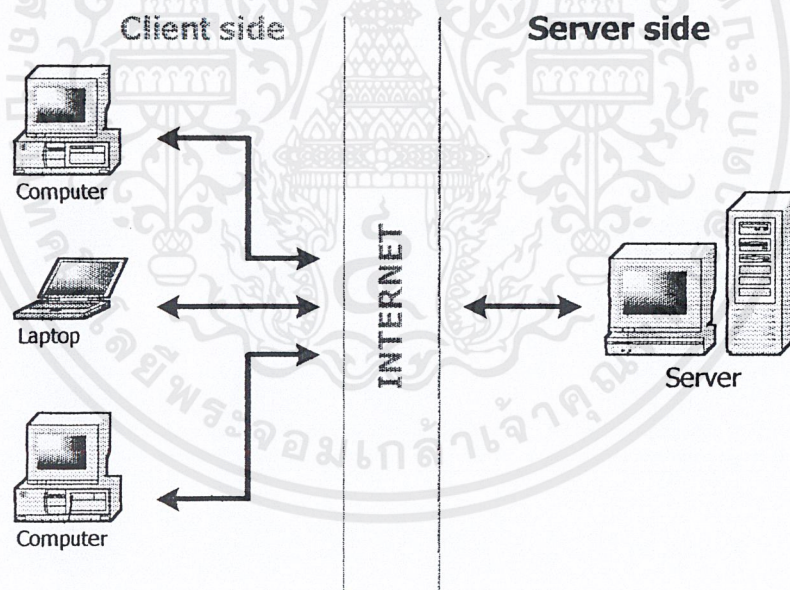
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 HTML

HTML มาจากคำว่า Hypertext Markup Language เป็นรูปแบบของภาษาที่ใช้เขียนโปรแกรมในเว็บเพจเพื่อแสดงผลบนเว็บเบราว์เซอร์ ลักษณะของเอกสาร HTML จะเป็นเท็กซ์ไฟล์ธรรมดาที่ต้องอาศัยการแปลความจากเว็บเบราว์เซอร์ ปัจจุบันได้มีการพัฒนารูปแบบการใช้ HTML ให้กว้างขวางมากยิ่งขึ้น โดยการแทรกสคริปต์หรือ applet เพื่อให้เว็บเพจมีชีวิตชีวาในลักษณะไดนามิกมากขึ้น

2.12 Internet Client และ Server

เป็นแนวคิดในการสื่อสารข้อมูลที่ใช้บนอินเทอร์เน็ต โดยแบ่งคอมพิวเตอร์ออกเป็น 2 ส่วน คือ ส่วนที่ทำหน้าที่เป็น Client และส่วนที่ทำหน้าที่เป็น Server คอมพิวเตอร์ที่ทำหน้าที่เป็น Client จะใช้โปรแกรม Browser ที่คอยส่งความต้องการ หรือที่เรียกว่า Request ไปยังคอมพิวเตอร์ที่ทำหน้าที่เป็น Server ซึ่งจะคอยตอบรับ (Response) Request ที่ส่งมา เช่น การส่ง HTML Page เพื่อกลับไปแสดงยังโปรแกรม Browser ของคอมพิวเตอร์ที่ทำหน้าที่เป็น Client



ภาพที่ 2.3 แสดงลักษณะการทำงานของ Client & Server

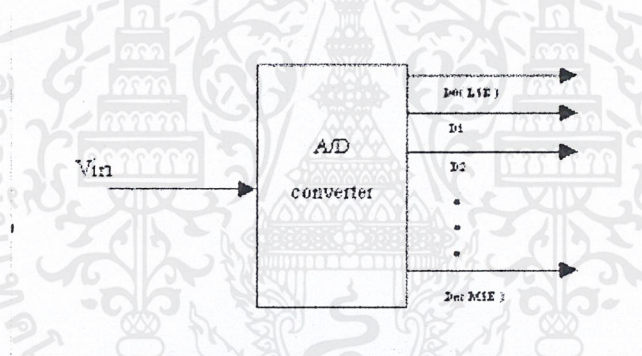
บทที่ 3

Analog to Digital Converter(ADC)

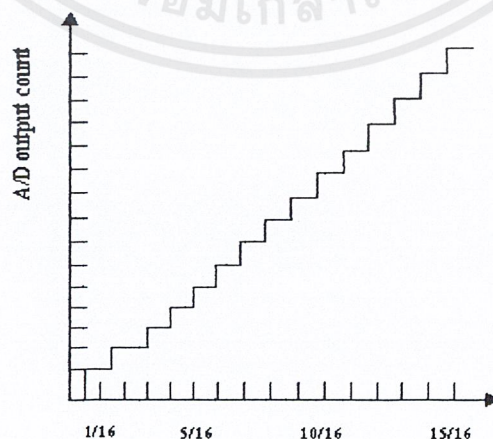
3.1 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

ในการวัดสัญญาณต่างๆ ด้วยคอมพิวเตอร์พีซีนั้น ปกติแล้วจะทำการประมวลผลข้อมูลด้วยซอฟต์แวร์จากพีซี ซึ่งการรับข้อมูลของพีซีเป็นแบบดิจิทัล ดังนั้นในการติดต่อกับสัญญาณอนาล็อกที่ได้จากอุปกรณ์ภายนอกหรือจากตัวแปลงสัญญาณ จึงจำเป็นต้องมีการแปลงสัญญาณอนาล็อกดังกล่าวให้เป็นสัญญาณดิจิทัลเสียก่อนแล้วจึงนำมาประมวลผล

A/D Converter นั้นก็คืออุปกรณ์แปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล ภาพที่ 3.1 แสดงบล็อกไดอะแกรมของ A/D Converter วงจรนี้จะผลิตเอาต์พุตแบบไบนารี n บิตซึ่งจะเป็นสัดส่วนกับแรงดันอินพุต ภาพที่ 3.2 แสดงถึงคุณสมบัติของ A/D Converter



ภาพที่ 3.1 แสดงสัญญาณของ A/D คอนเวอร์เตอร์

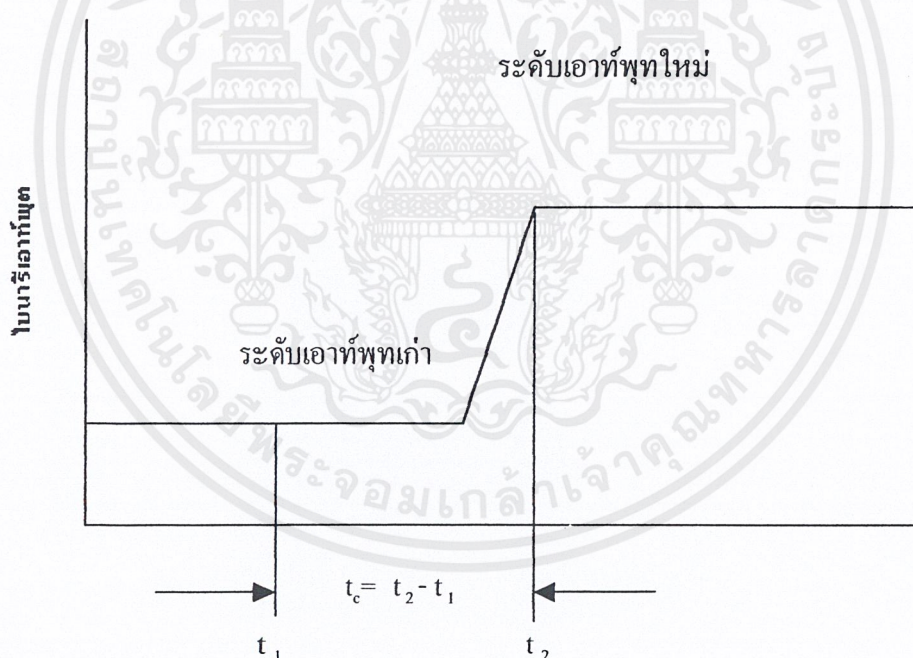


ภาพที่ 3.2 คุณสมบัติของ A/D Converter แบบ 4 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 หลักการทำงานของ A/D คอนเวอร์เตอร์

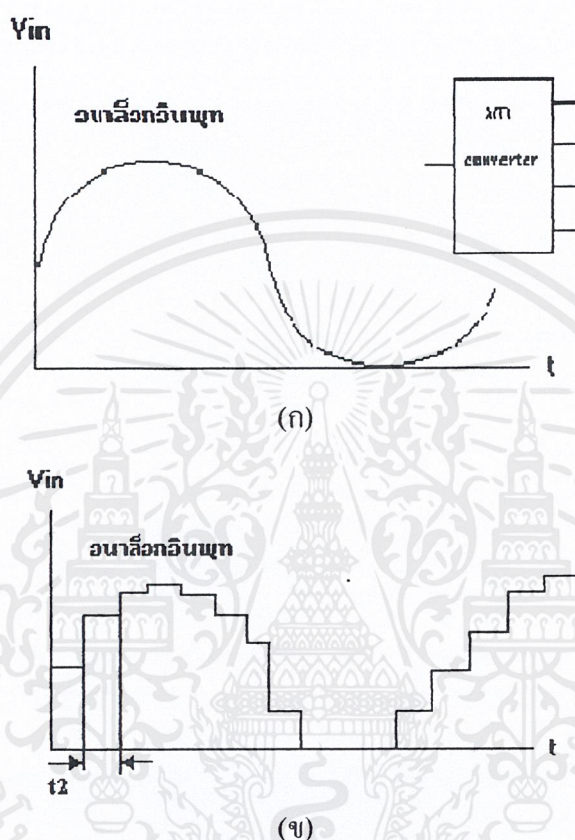
A/D Converter อินพุตเป็นสัญญาณอนาล็อก และเอาต์พุตเป็นสัญญาณดิจิทัลจำนวน n บิต อินพุตของ A/D Converter เป็นสัญญาณอนาล็อกที่ผ่านออกมาจากวงจรแอมป์และโฮลด์ (Sample And Hold) ซึ่งส่วนของวงจรนี้ทำหน้าที่รับสัญญาณอนาล็อกที่ต้องการแปลงเป็นสัญญาณดิจิทัลเข้ามาในวงจรแล้วนำเอาต์พุตไปต่อเชื่อมกับอินพุตของ A/D Converter โดยการแปลงสัญญาณจาก A/D Converter แต่ละครั้ง วงจรนี้จะรับสัญญาณอนาล็อกแล้วส่งเข้าที่ขาอินพุตของ A/D Converter ในช่วงเวลาที่เท่ากับ Sample Time ที่กำหนดจาก Sample Frequency ที่กำหนดในวงจร ดังนั้นในการควบคุมการทำงานของ A/D Converter การแปลงสัญญาณจะมีค่าจำนวนข้อมูลเท่าใดพิจารณาได้จาก Sample Time ภายในชุดวงจรแอมป์และโฮลด์นี้ อย่างไรก็ตามการพิจารณาเวลาการทำงานของ A/D Converter ยังขึ้นอยู่กับผลของเวลาในการแปลงสัญญาณ (Conversion Time, t_c) ภายใน A/D Converter คือเวลาที่ใช้ในช่วงที่อินพุตเข้ามาจนถึงการแสดงค่าระดับเอาต์พุตใหม่ ตามภาพที่ 3.3 เกิดผลการหน่วงเวลา (Time Delay) ขึ้น



ภาพที่ 3.3 แสดงการตอบสนองของเวลาแปลงสัญญาณของ A/D Converter

พิจารณาตามภาพที่ 3.3 อินพุตจากสัญญาณอนาล็อกที่เข้าใน A/D Converter จะอยู่ที่เวลา t_1 และสัญญาณตอบสนองของอินพุตอนาล็อกนั้นเกิดขึ้น โดยผลต่างของเวลาทั้งสองคือเวลาในการแปลงสัญญาณ t_c ซึ่งเวลาดังกล่าวนี้เป็นเวลาที่ใช้จริงของ A/D Converter ในการแปลงสัญญาณอนาล็อกเป็นดิจิทัล ดังนั้นการกำหนดเวลาแอมป์จากความถี่แอมป์ควรมีเวลาเอกสารนี้เป็นเอกสารทงส่วนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากกว่าเวลาในการแปลงสัญญาณจาก A/D Converter เพราะสัญญาณอนาล็อกที่เป็นอินพุตต้องคงค่าอยู่ตลอดเวลาในการแปลงสัญญาณ จึงจะได้ค่าดิจิทัลเอาต์พุตที่ถูกต้องและเพื่อให้ทราบถึงผลของการแซมเปิลงานสัญญาณอินพุตอนาล็อกไปเป็นเอาต์พุตพิจารณาจากกราฟไซน์ดังภาพที่ 3.4 ก. โบนารีเอาต์พุต

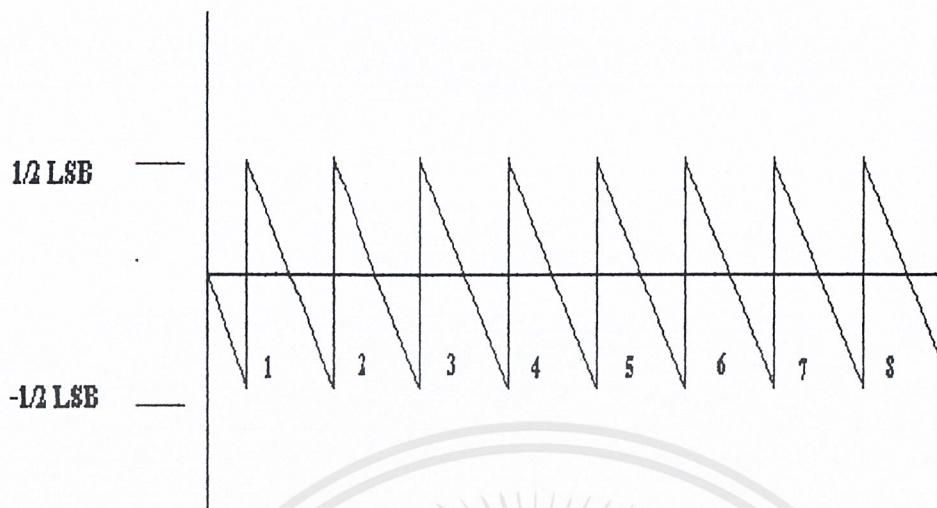


ภาพที่ 3.4 ขบวนการแปลงสัญญาณดิจิทัลด้วย A/D Converter

ถ้าให้เวลาแปลงสัญญาณน้อยมากๆ จนตัดทิ้งได้เวลาในการแซมปลิง $1/10$ ของสัญญาณอินพุตจะได้กราฟเป็นลักษณะตามภาพที่ 3.4 ข. ถ้าลดเวลาแซมเปิลให้น้อยลงจะเพิ่มเรโซลูชัน (จำนวนเอาต์พุตบิต) ให้มากขึ้นก็จะได้ผลของสัญญาณอนาล็อกอินพุตมากขึ้น

3.3 ความผิดพลาดของ A/D Converter

หลักสำคัญที่ใช้ในการพิจารณาคุณสมบัติของการเปลี่ยนแปลงสัญญาณอนาล็อกเป็นดิจิทัลนั้นสามารถสังเกตได้จากภาพที่ 3.2 จะเห็นจำนวนขั้นบันไดทั้งหมดมี 15 ขั้นเอาต์พุตของ A/D Converter นั้น จะประมาณว่าเป็นสัญญาณอินพุตแบบดิจิทัล กราฟแสดงข้อผิดพลาดในเอาต์พุตที่จุดต่างๆ ดูได้จากภาพที่ 3.5



ภาพที่ 3.5 ข้อผิดพลาดของ A/D Converter

ในกรณีของ A/D Converter ที่มีเรโซลูชันสูงๆ (เอาต์พุตบิตมากๆ) ความกว้างสูงสุดของข้อผิดพลาด ($-1/2 \text{ LSB}$ ถึง $1/2 \text{ LSB}$) จะลดลงซึ่งตามทฤษฎีแล้วถ้าเป็น A/D Converter ในอุดมคติจะต้องมีจำนวนเอาต์พุตมากนับไม่ได้ (ถึงอนันต์) ดังนั้นจะมีค่าเรโซลูชันถึงอนันต์ทำให้กราฟภาพที่ 3.5 ขนาดแกนตั้งเท่ากับ 0 หรือความแตกต่างระหว่าง $-1/2 \text{ LSB}$ ถึง $1/2 \text{ LSB}$ มีค่าประมาณ 0 การสร้าง A/D Converter ซึ่งเรียกข้อผิดพลาดนี้ว่าข้อผิดพลาดควอนไทไรซ์เป็นข้อผิดพลาดที่ไม่สามารถกำจัดทิ้ง

ข้อผิดพลาดอีกอย่างหนึ่งคือผลออฟเซตที่เกิดขึ้นในการแปลงสัญญาณของตัว A/D Converter ทำให้ผลการแปลงผิดพลาดไปได้ แต่กรณีออฟเซตนี้สามารถแก้ไขได้โดยการปรับความดันทานอินพุตเข้า A/D Converter เอาต์พุตของ A/D Converter เกิดออลิอาซิ่ง(aliasing) ขึ้นในการใช้งานที่มีการเปลี่ยนแปลงสัญญาณอินพุตเร็วๆ ควรที่จะใช้การแซมเปิลแบบความเร็วสูงเพื่อจะให้ได้ความถูกต้องมากขึ้น ในปัจจุบันการควบคุมระบบจะใช้การแซมเปิลที่มีอัตราสูง อัตราการแซมเปิลมากที่สุดถูกจำกัดโดยความเร็วในการแปลงสัญญาณของ A/D Converter เช่น ถ้าแซมเปิลทุก 10 nS ก็จะไม่ดี เมื่อความเร็วในการแปลงสัญญาณเป็น 500 nS จะทำให้เอาต์พุตออกมามาไปใช้งานไม่ได้

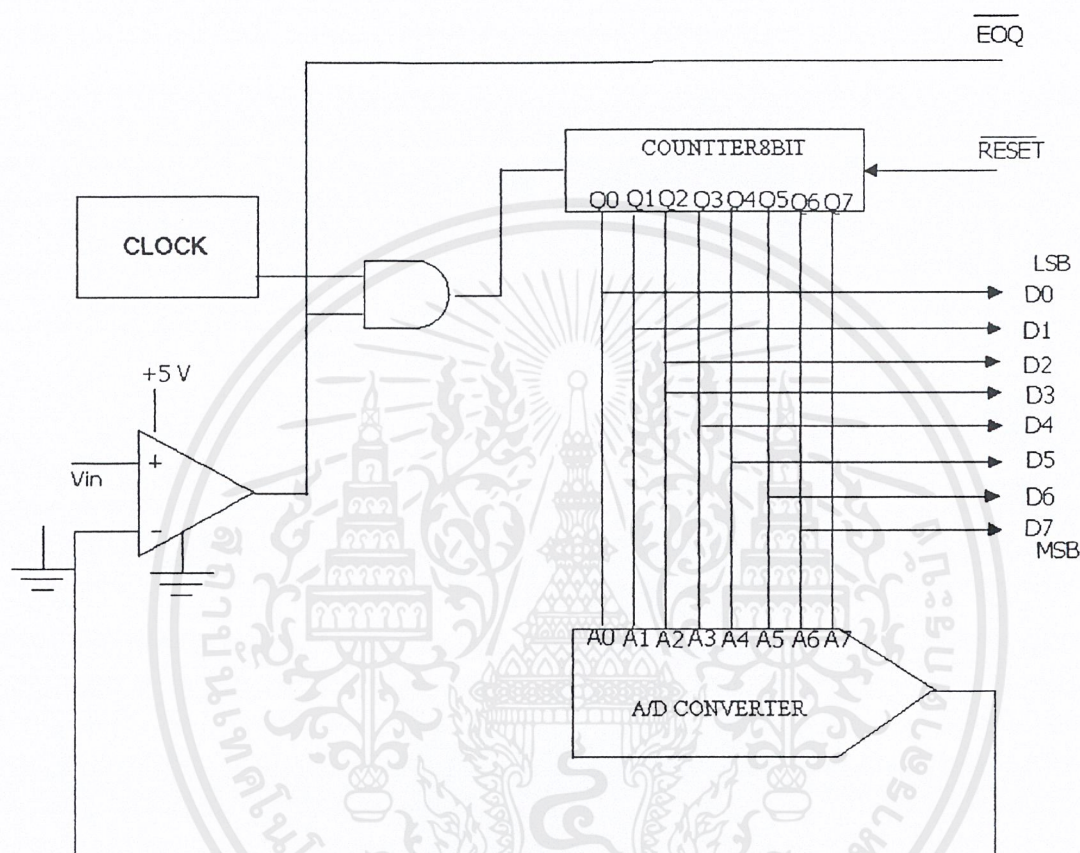
3.4 วงจรของ A/D Converter

ในปัจจุบันมีวิธีการและเทคนิคมากมายเกี่ยวกับการแปลงสัญญาณ แต่วิธีการหลัก ก็จะเป็นพวกที่จะกล่าวถึงต่อไปนี้ เช่น แรมป์ A/D Converter (Ramp A/D Converter) , ซัสเซสซีฟ แอปพร็อกซิเมชัน A/D Converter (Successive Approximation A/D Converter, SAR)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 แรมป์เอาต์พุตของ A/D Converter

เป็น A/D Converter ที่เข้าใจการทำงานได้ง่ายที่สุดโดยแสดงโครงสร้างการทำงานดังภาพที่ 3.6 ซึ่งจากรูปสามารถอธิบายการทำงานตามหลักการได้ดังนี้

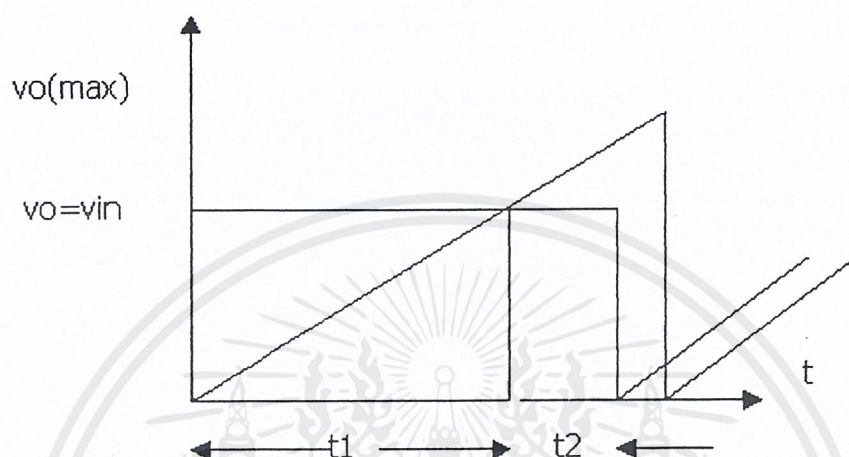


ภาพที่3.6 A/D คอนเวอร์เตอร์

1. คอมพิวเตอร์ส่งสัญญาณรีเซ็ตไปที่เคาเตอร์เพื่อให้สามารถรับสัญญาณอินพุตได้ โดยขณะนี้ผลจากเอาต์พุตของ A/D Converter (V_o) มีค่าอยู่ที่ระดับต่ำสุด
2. การทำงานของวงจรมีการเปลี่ยนแปลงความถี่ของสัญญาณนาฬิกาซึ่งสัญญาณเอาต์พุตถูกนำมา AND กับสัญญาณนาฬิกานี้แล้วจึงส่งไปยังเคาเตอร์เป็นผลให้ V_o แสดงระดับแรงดันที่สูงขึ้นทีละ 1 LSB
3. ณ จุดเดียวกันนั้นตัวเคาเตอร์จะนับเพิ่มขึ้นจนกระทั่งเอาต์พุตของ A/D Converter สูงกว่าค่าแรงดันอินพุต V_{in} เมื่อถึงจุดนี้ของคอมพิวเตอร์จะแสดงค่าไปจนกระทั่งเป็น 0 โวลต์ เมื่อถึง 0 โวลต์ที่จะหยุดสัญญาณนาฬิกาและหยุดการนับที่จุดซึ่ง V_o เริ่มมากกว่า V_{in} ษา EOC จะ

ลดระดับเป็นระดับต่ำและส่งสัญญาณไปให้คอมพิวเตอร์ว่าข้อมูลขณะนี้พร้อมที่จะอ่านได้แล้วซึ่งข้อมูลที่ส่งไปยังคอมพิวเตอร์อ่านข้อมูลโดยผ่านไปในทางอินพุตพอร์ท

4. หลังจากคอมพิวเตอร์อ่านข้อมูลเสร็จ คอมพิวเตอร์ก็จะส่งสัญญาณรีเซ็ตมาที่ A/D Converter หลังจากนั้นก็เริ่มทำกระบวนการแบบเดิมอีกครั้ง

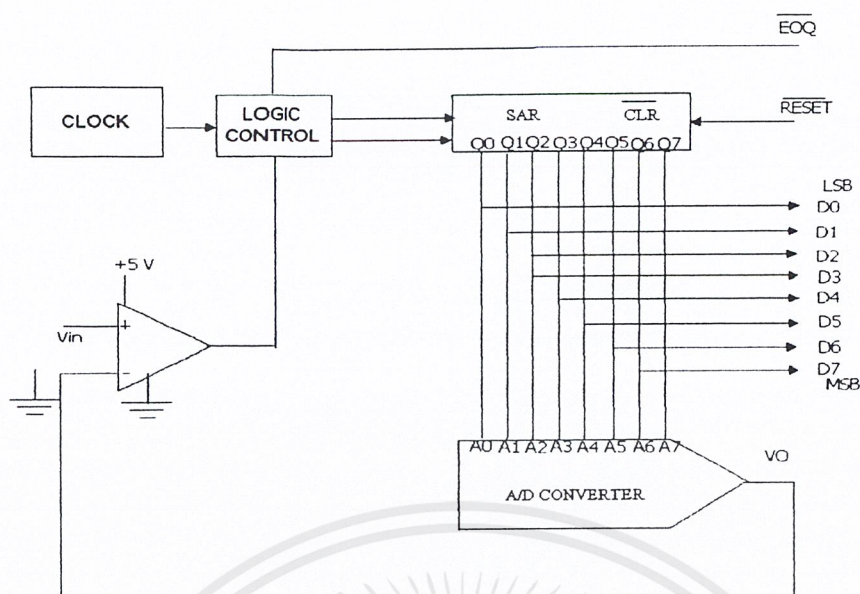


ภาพที่ 3.7 กราฟเอาต์พุตของส่วน D/A ของ A/D Converter

กราฟรูปนี้แสดงถึงการทำงานของ แรมป์ โดย t_1 แทนเวลาที่นับใช้เพื่อจะแรมป์เอาต์พุตของ D/A converter ที่เริ่มจะเลย V_{in} และ t_2 แทนเวลาระหว่างที่คอมพิวเตอร์อ่านข้อมูลและส่งสัญญาณรีเซ็ตกลับมาให้ A/D Converter เริ่มต้นกระบวนการเดิมอีกครั้งหนึ่ง ดังนั้นในการเก็บข้อมูล 1 ค่าจะต้องใช้เวลาในการแปลงสัญญาณ 2 ช่วงด้วยกันคือ เวลา t_1 ถูกกำหนดโดยค่าของ V_{in} และความถี่ของสัญญาณนาฬิกา ส่วนเวลา t_2 ถูกกำหนดโดยเวลาของคอมพิวเตอร์อ่านข้อมูลและส่งสัญญาณรีเซ็ตกลับ

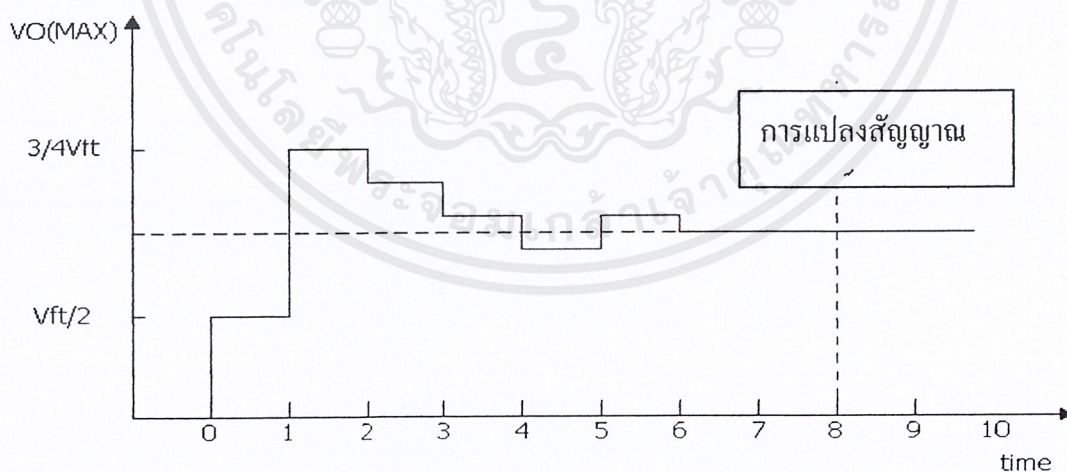
3.6 ชั้สเสตซีฟแอปพลิเคชัน A/D Converter

SAR เป็นตัวหลักของ A/D Converter ซึ่งมีลักษณะใกล้เคียงกับแบบแรมป์ โดยโครงสร้างการทำงานแสดงดังภาพที่ 3.8 ซึ่งเป็นการแสดงบล็อกไดอะแกรม A/D Converter แบบ SAR



ภาพที่ 3.8 คอนเวอร์เตอร์ซึ่งใช้เทคนิค SAR

SAR คอนเวอร์เตอร์ต้องใช้ D/A converter และคอมพาราเตอร์ในการทำงานเหมือนกัน ส่วนที่แตกต่างกันจะเห็นได้แก่ ส่วนเคาทช์เตอร์ วิธีที่แรมป์ทำงาน โดยอาศัยการนับขึ้นในลำดับ ไบนารีจนกระทั่งเกิน V_i ทำงานของ SAR คอนเวอร์เตอร์จะอธิบายต่อไปและใช้ภาพที่ 3.8 และ 3.9 ในการบรรยาย



ภาพที่ 3.9 กราฟของส่วน D/A ระหว่างการแปลงสัญญาณ

1. การเริ่มต้นของการเปลี่ยนแปลงจากสัญญาณอนาล็อกเป็นดิจิทัล พิจารณาจากสัญญาณนาฬิกาที่ส่งไปยังระบบควบคุมลอจิก โดยเอาต์พุตของระบบควบคุมลอจิกจะส่งพัลส์หนึ่งลูกให้กับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SAR เพื่อเซตค่าของ MSB และลบค่าเอาต์พุตที่ยังเหลืออยู่ ซึ่งจะทำให้ค่าของ V_o เท่ากับค่า $V_{fs}/2$ ในภาพที่ 3.9 ซึ่งจะน้อยกว่าค่า V_{in} และเอาต์พุตของคอมพาราเตอร์จะยังคงอยู่ในระดับ “1” ระบบควบคุมลอจิกจะตรวจสอบด้วยว่าคอมพาราเตอร์ยังคงเป็น “1” อยู่ที่จะส่งสัญญาณไปที่ SAR ซึ่งจะค้ำค่าของ MSB เป็นลอจิก “1” ที่ Q7

2. เมื่อมีสัญญาณนาฬิกาถูกตัดไป MSB ตัวใหม่ซึ่งอยู่ที่ Q6 ของ SAR จะถูกเซตค่าเป็น “1” ดังภาพที่ 3.9 เอาต์พุตของ D/A จะเกินค่าของ V_{in} ทำให้เอาต์พุตของคอมพาราเตอร์เปลี่ยนระดับ ระบบควบคุมลอจิกตรวจสอบทราบก็จะไม่ทำการค้ำค่า Q6 นั่นคือค่าที่ค้ำที่ตำแหน่ง Q6 เท่ากับ “0”

3. สัญญาณนาฬิกาถูกถัดมา Q6 จะถูกเคลียร์และ Q5 จะถูกเซต จากการเปรียบเทียบได้ เอาต์พุตของคอมพาราเตอร์เป็น “1” ส่วนควบคุมลอจิกก็จะไม่ทำการค้ำค่า Q5 ไว้เพราะยังสูงกว่าค่าของ V_{in}

โดยปกติแล้ววงจรจะทดสอบเอาต์พุตโดยเริ่มต้นที่ MSB ของ SAR ถ้าเอาต์พุตของ D/A Converter เกินค่า V_{in} แล้วบิตนั้นจะค้ำค่าลอจิก “1” ไว้ หลังจากตรวจสอบหมดแล้วไบนารีเอาต์พุตจะเป็นอัตราส่วนกับ V_{in} จะเห็นว่า A/D แบบนี้ดีกว่าแบบแร่มปี คือเวลาในการแปลงสัญญาณเป็นสัดส่วนโดยตรงกับจำนวนบิตของเคาท์เตอร์ SAR

SAR A/D Converter ก็มีข้อผิดพลาดแบบเดียวกับแบบแร่มปี คือข้อผิดพลาดทางอัตราขยายออฟเซต และความไม่เป็นเชิงเส้นซึ่งเกิดขึ้นในส่วน D/A

บทที่ 4

ไมโครคอนโทรลเลอร์

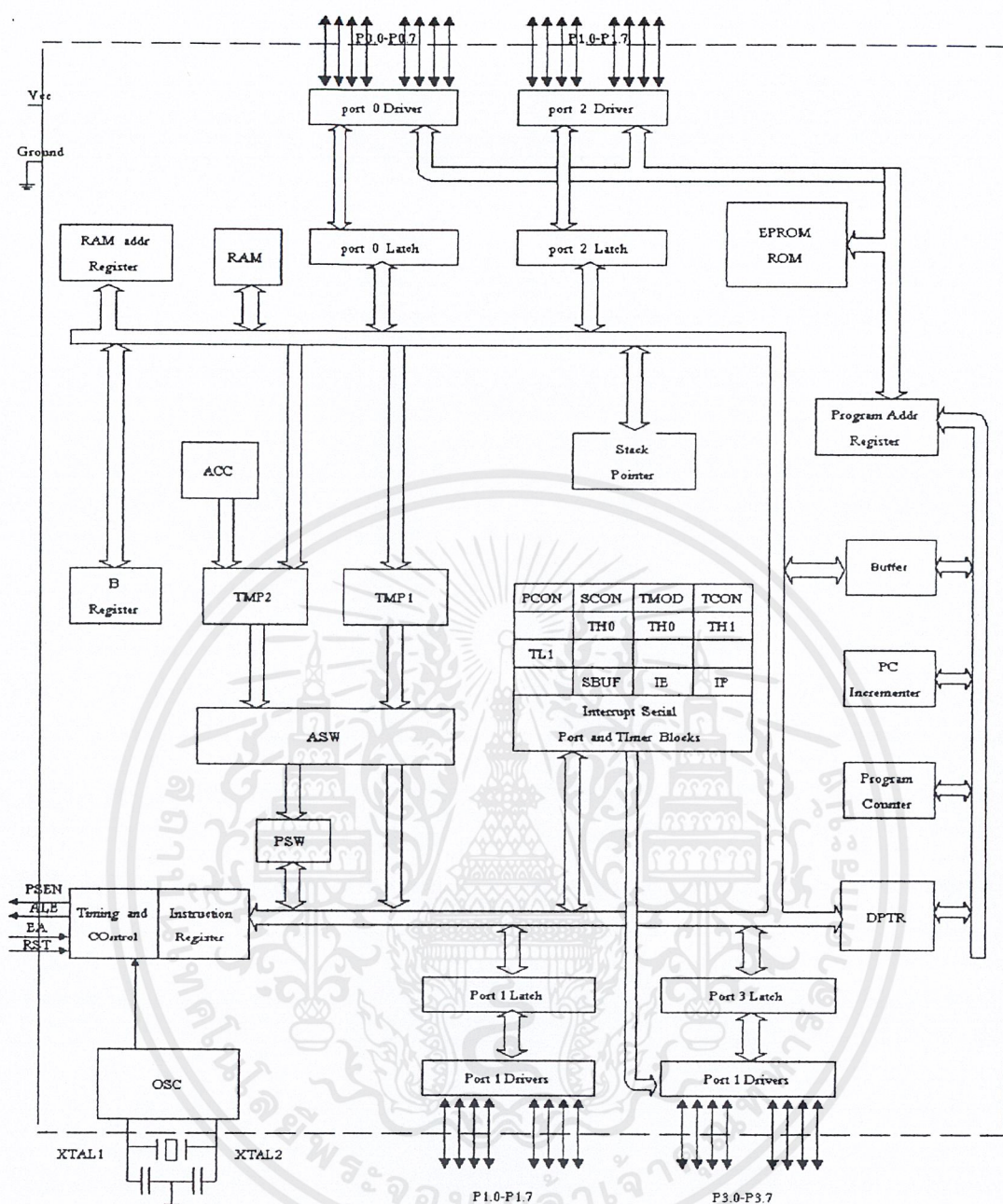
4.1 ไมโครคอนโทรลเลอร์ MCS – 51

MCS-51เป็นไมโครคอนโทรลเลอร์แบบชิปเดี่ยวที่มีข้อดีเมื่อเทียบกับไมโครโปรเซสเซอร์ขนาด 8 บิต ตระกูลอื่น ดังนี้

- มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป (RAM) บรรจุไว้ใน 128 – 256 ไบต์
 - มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในจำนวน 4 กิโลไบต์
 - มีวงจรตั้งเวลาวงจรนับขนาด 16 บิต 2 ตัว อยู่ภายใน
 - มีวงจรรับส่งข้อมูลอนุกรมได้ 2 ทิศทาง
 - มีสัญญาณนาฬิกาภายในตัว
 - มีพอร์ทที่สามารถรับหรือส่งข้อมูลได้ 2 ทิศทาง จำนวน 4 พอร์ท ๆ ละ 8 บิต
- นอกจากนี้ MCS – 51 ยังมีคุณสมบัติอื่น ๆ ที่น่าสนใจ คือ
- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
 - มีหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้อย่างละ 64 กิโลไบต์
 - มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
 - จัดลำดับความสำคัญของสัญญาณอินเทอร์รัปต์ได้ 2 ระดับ
 - รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราเร็วในการรับและส่งข้อมูลได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที
 - สามารถประมวลผลแบบบูทีนเพื่อใช้งานควบคุมโดยเฉพาะ
 - มีรีจิสเตอร์สำหรับใช้งานเป็นไทม์เมอร์หรือเคาน์เตอร์ เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิป หรือ นับการเปลี่ยนสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา
 - หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบงานทำได้ง่ายขึ้น

4.2 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ 8051

โครงสร้างภายในชิปไมโครคอนโทรลเลอร์ 8051 ชิปเดี่ยวแสดงดังภาพที่ 4.1 ซึ่งอธิบายถึงส่วนย่อย ๆ ภายใน 8051



ภาพที่ 4.1 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์ 8051

การใช้งานต่าง ๆ ของไมโครคอนโทรลเลอร์ มีดังนี้

- Vcc ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์เข้าไปเพื่อให้วงจรสามารถทำงานได้
- Vss ขา 20 เป็นขาที่ต้องต่อกับกราวด์ของแหล่งจ่ายไฟ
- RST ขา 9 ขา รีเซตนี้ จะรีเซ็ตการทำงานของ 8051 ถ้าป้อนสัญญาณที่มีสถานะลอจิก 1 ที่ ขานี้จะเป็นการรีเซ็ตการทำงาน กลับไปเริ่มการทำงานจากค่าตั้งที่อยู่ในหน่วยความจำตำแหน่ง 0000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ALE ขา 30 ใช้เป็นคำสั่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตช์ค่าตำแหน่งไบต์ค่าจากพอร์ท 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บ โปรแกรมหรือข้อมูลภายนอก
- PSEN ขา 29 ใช้ส่งสัญญาณเพื่ออ่านคำสั่งจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิป
- XTAL1 ขา 19 ใช้ต่อคริสตัลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์
- XTAL2 ขา 18 ใช้ต่อคริสตัลภายนอก โดยเป็นเอาต์พุตออกจากวงจรออสซิลเลเตอร์
- PORT 0 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึง 7 ตามลำดับ พอร์ท 0 นี้ ใช้ได้ทั้งการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้รับส่งข้อมูลก็ได้ นอกจากนี้ยังใช้งานได้หลายอย่างดังนี้
 1. ใช้สำหรับส่งตำแหน่งหน่วยความจำที่ต้องการติดต่อด้วย โดย 8 บิตล่างถูกส่งออกไปทางพอร์ท 0 และ 8 บิตบนถูกส่งออกไปทางพอร์ท 2
 2. ใช้รับส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory
 3. ใช้รับส่งข้อมูลออกจากพอร์ทโดยตรง
- PORT 1 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 1 ถึง 8 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับ ใช้ทำหน้าที่ เป็นตัวรับส่งข้อมูลเท่านั้น ไม่สามารถส่งตำแหน่งได้
- PORT 2 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 21 ถึง 28 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับ ใช้งานเพียง 2 ลักษณะ คือ
 1. ใช้ส่งตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อทำงานร่วมกับพอร์ท 0
 2. ใช้เป็นพอร์ทรับส่งข้อมูลกับภายนอก
- PORT 3 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 10 ถึง 17 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ นอกจากจะใช้งานเหมือนพอร์ทอื่น ๆ แล้วยังใช้งานอื่นโดยใช้คำสั่งควบคุม ดังนี้

P3.0 (RxD)	เป็นขาที่ใช้รับข้อมูลแบบอนุกรม
P3.1 (TxD)	เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม
P3.2 (INT0)	ใช้รับสัญญาณขัดจังหวะจากภายนอก
P3.3 (INT1)	ใช้รับสัญญาณขัดจังหวะภายใน
P3.4 (T0)	ใช้เป็นขาจับสัญญาณให้เคาน์เตอร์ของไทม์เมอร์ 0
P3.5 (T1)	ใช้เป็นขาจับสัญญาณให้เคาน์เตอร์ของไทม์เมอร์ 1
P3.6 (WR)	ใช้เป็นขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก
P3.7 (RD)	ใช้เป็นขาสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก

พอร์ท 3 ของ MCS-51 ถูกใช้เป็นพอร์ทอนุกรม จะใช้ขา TxD และ RxD ในการรับส่งข้อมูลโดยขา ทั้ง 2 จะอยู่ในพอร์ท3 คือ P3.1 หรือ ขา 11 เป็น TxD และ P3.0 หรือขา 10 เป็น RxD พอร์ท อนุกรม ของ MCS – 51 สามารถทำงานเป็นแบบ Full Duplex ได้ คือสามารถรับและส่งข้อมูลในเวลา เดียวกันได้ โดยมีการรับและส่งข้อมูลจะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ได้

รีจิสเตอร์ที่สำคัญ ในการรับส่งข้อมูลคือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน Special Function Register โดยรีจิสเตอร์ Serial Port Buffer (SBUF) จะอยู่ในตำแหน่ง 99H ถ้าเขียน ข้อมูลลงไปในตำแหน่งนี้จะเป็นการส่งข้อมูลออกจากพอร์ทอนุกรม และถ้าอ่านข้อมูลจากตำแหน่ง นี้จะเป็นการรับข้อมูลจากพอร์ทอนุกรม โดยใน SBUF จะประกอบด้วยบัฟเฟอร์ 2 ตัวสำหรับ ส่งและรับข้อมูล

สำหรับ Serial Port Control Register (SCON) ซึ่งอยู่ในตำแหน่ง 98H จะเป็นรีจิสเตอร์ที่ สามารถเข้าถึงข้อมูลระดับบิตได้รีจิสเตอร์นี้จะทำหน้าที่ควบคุม และบอกสถานะต่าง ๆ ของการ รับส่งข้อมูลแบบอนุกรม

ตารางที่ 4.1 Serial Port Control Register

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.7	SM0	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON.6	SM1	9EH	บิตเลือกโหมดการทำงานบิต 1
SCON.5	SM2	9DH	บิตเลือกโหมดการทำงานบิต 2
SCON.4	REN	9CH	บิตแฟล็กกำหนดยอมให้มีการรับข้อมูล
SCON.3	TB8	9BH	ค่าของบิต 9 สำหรับการส่งข้อมูลในโหมด 2 และ 3 สามารถเซต และเคลียร์ได้โดยซอฟต์แวร์
SCON.2	RB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา
SCON.1	T1	99H	บิตแฟล็กแสดงการอินเทอร์รัพท์ภายหลังจากส่ง ข้อมูลออกไป โดยจะเซตเมื่อส่งข้อมูลออกไป หมดแล้ว และสามารถเคลียร์ได้โดยซอฟต์แวร์
SCON.0	R1	98H	แฟล็กแสดงการอินเทอร์รัพท์ภายหลังรับข้อมูล เข้ามาสามารถเคลียร์ได้โดยซอฟต์แวร์

ตารางที่ 4.2 แสดงโหมดต่าง ๆ ของการรับส่งแบบอนุกรม

SM0	SM1	MODE	ความหมาย	Baud Rate
0	0	0	Shift Register	เปลี่ยนแปลงไม่ได้ (Oscillator Frequency / 12)
0	1	1	8 – bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจากไทม์เมอร์
1	0	2	9 – bit UART	เปลี่ยนแปลงไม่ได้ (Oscillator Frequency / 12 หรือ /64)
1	1	3	9 – bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจากไทม์เมอร์

4.3 ไทม์เมอร์ / เคาน์เตอร์

ไมโครคอนโทรลเลอร์ในตระกูล มินิรีจิสเตอร์พิเศษที่สามารถเลือกใช้งานเป็นไทม์เมอร์ หรือ เคาน์เตอร์อย่างใดอย่างหนึ่ง รีจิสเตอร์ประเภทนี้มีอยู่ด้วยกัน 2 ตัว แต่ละตัวขนาด 16 บิต เรียก ไทม์เมอร์ 0 และไทม์เมอร์ 1 ตามลำดับ

- ไทม์เมอร์นั้น ค่าในรีจิสเตอร์ที่ใช้เป็นไทม์เมอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มขึ้นทุกแมกซีนไซเคิล
- เคาน์เตอร์นั้นค่าในรีจิสเตอร์ที่ใช้เป็นเคาน์เตอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มค่าทีละ 1 เมื่อมีการเปลี่ยนสถานะ

ไทม์เมอร์ 0 และ ไทม์เมอร์ 1

สามารถเลือกการทำงานให้เป็นไทม์เมอร์หรือเคาน์เตอร์ได้โดยการกำหนดค่าบิตในรีจิสเตอร์ใช้งานเฉพาะ โดยหากบิตนี้มีค่าเป็น 0 หมายถึงเลือกใช้งานเป็นไทม์เมอร์ถ้าบิตนี้มีค่าเป็น 1 หมายถึงเลือกใช้งานเป็นเคาน์เตอร์

นอกจากนี้จะเลือกการทำงานของรีจิสเตอร์ให้เป็นไทม์เมอร์หรือเคาน์เตอร์ได้แล้วในแต่ละการทำงานยังมีการทำงานย่อยอยู่อีก 4 แบบ ตามความเหมาะสมของการใช้งาน

โหมด 0 จะใช้รีจิสเตอร์ขนาด 8 บิต เป็นตัวนับ โดยมีการเพิ่มค่าครั้งละ 1 ทุกครั้งนับสัญญาณได้ครบ 32 ครั้ง โดยในโหมดนี้รีจิสเตอร์ที่ใช้ นับเพียง 13 บิต (8 บิตในรีจิสเตอร์ TLx รวมกับ 5 บิต ใน THx)

โหมด 1 การทำงานเหมือน โหมด 0 เว้นแต่ค่าในรีจิสเตอร์ถูกใช้งานครบทั้ง 16 บิตนั่นเอง คือ ไทม์เมอร์หรือเคาน์เตอร์ในโหมดนี้มีขนาด 16 บิต

โหมด 2 ในโหมดนี้จะกำหนดรีจิสเตอร์ใช้งานในการนับเพียง 8 บิต ที่มีการโหลดค่าด้วยค่าในรีจิสเตอร์ THx การใช้งานโหมดนี้มีไว้เพื่อสร้างสัญญาณอินเตอร์รัปต์ที่มีความยาวเวลาที่

โหมด 3 ในโหมดนี้ไทม์เมอร์ 1 จะไม่มีการนับแต่ไทม์เมอร์จะบังคับให้รีจิสเตอร์ TLO ของไทม์เมอร์ 0 ถูกใช้เป็นไทม์เมอร์เพียงอย่างเดียว การทำงานโหมด 3 มีไว้เพื่อใช้งานที่ต้องการไทม์เมอร์หรือเคาน์เตอร์ขนาด 8 บิตเพิ่มขึ้น

ตารางที่ 4.3 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Timer 0

โหมด	ฟังก์ชันไทม์เมอร์ 0	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	00H	08H
1	16 bit Timer	01H	09H
2	8 bit Auto Reload	02H	0AH
3	two 8 bit Timer	03H	0BH

ตารางที่ 4.4 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Counter 0

โหมด	ฟังก์ชันไทม์เมอร์ 0	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	04H	0CH
1	16 bit Timer	05H	0CH
2	8 bit Auto Reload	06H	0EH
3	two 8 bit Counter	07H	0FH

ตารางที่ 4.5 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Timer 1

โหมด	ฟังก์ชันไทม์เมอร์ 0	ควบคุมการโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	00H	80H
1	16 bit Timer	10H	90H
2	8 bit Auto Reload	20H	A0H
3	Does not run	30H	B0H

ตารางที่ 4.6 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่าง ๆ สำหรับ Counter 1

โหมด	ฟังก์ชันไทมเมอร์ 0	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	40H	C0H
1	16 bit Timer	50H	D0H
2	8 bit Auto Reload	60H	E0H
3	not available	--	--

4.4 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS – 51

รีจิสเตอร์ A,B และรีจิสเตอร์ใช้งานทั่วไป R0-R7 ซึ่งอยู่ในหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายในชิปบริเวณ 128 ไบต์แรก รีจิสเตอร์ใช้งานทั่วไป R0-R7 ใน MCS -51 มีอยู่ด้วยกันทั้งหมด 4 กลุ่ม แต่ละกลุ่มประกอบด้วย รีจิสเตอร์ จำนวน 8 ตัว (R0-R7) ซึ่งมีชื่อเรียกเหมือนกัน ดังนั้นรีจิสเตอร์ใช้งานทั่วไป R0-R7 ใน MCS – 51 จึงมีทั้งหมด 32 ตัว ในการทำงานขณะใด ๆ รีจิสเตอร์ทั้ง 4 กลุ่ม (R0-R7) จะถูกเลือกใช้งานเพียงกลุ่มเดียวเท่านั้น การเลือกใช้งานรีจิสเตอร์ R0-R7 กลุ่มใดกลุ่มหนึ่งใน 4 กลุ่ม กระทำโดยการเซตหรือเคลียร์บิต RS0,RS1 ในรีจิสเตอร์ใช้งานเฉพาะ PSW

4.5 รีจิสเตอร์ฟังก์ชันพิเศษใน MCS – 51

ใน MCS – 51 รีจิสเตอร์จะใช้หน่วยความจำ RAM ภายในชิป โดยส่วนหนึ่งเป็นรีจิสเตอร์พิเศษ (Special Function Register : SFR) ซึ่งมีทั้งหมด 21 ตัว โดยมีรีจิสเตอร์พิเศษต่างๆ จะเริ่มที่หน่วยความจำตั้งแต่ 80H ถึง FFH ซึ่งมีทั้งหมด 128 ตัวตำแหน่ง แต่จะเป็นรีจิสเตอร์ฟังก์ชันพิเศษเพียง 21 ตำแหน่ง แต่ถ้าเป็น 8032/8051 จะใช้ 26 ตำแหน่ง หรือมี SFR 26 ตัว

ตารางที่ 4.7 รีจิสเตอร์ใช้งานเฉพาะ PSW (Program Status Word) เข้าถึงข้อมูลได้ในระดับบิต

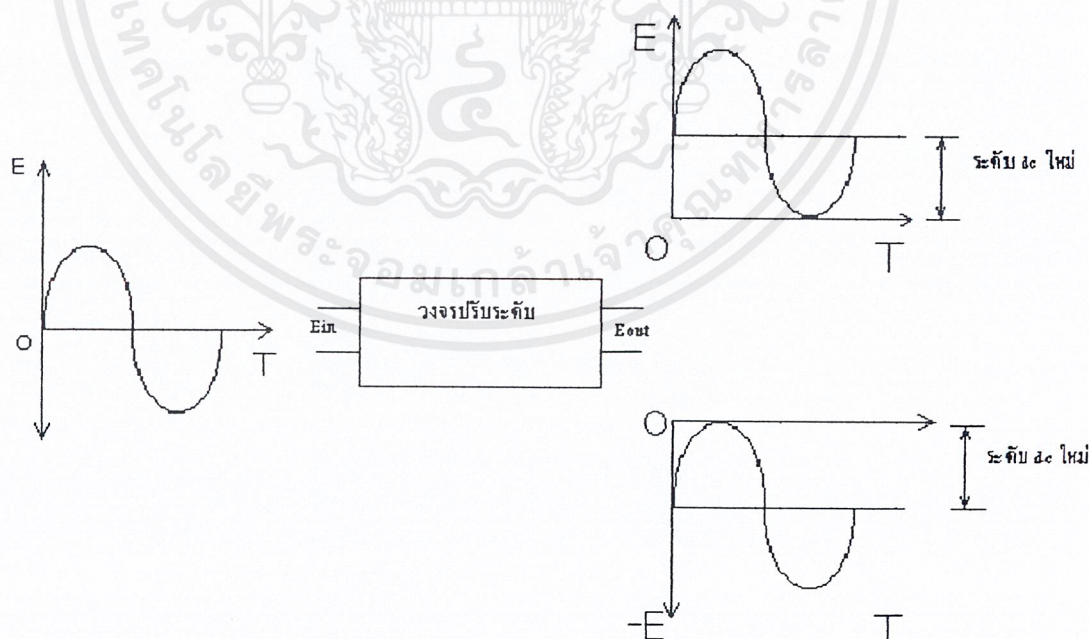
บิต	ชื่อบิต	ตำแหน่ง	ความหมาย
PSW.7	CY	D7H	Carry Flag
PSW.6	AC	D6H	Auxiliary Carry Flag
PSW.5	F0	D5H	Flag 0
PSW.4	RS1	D4H	บิตสำหรับเลือกรีจิสเตอร์ แบงค์ 1
PSW.3	RS0	D3H	บิตสำหรับเลือกรีจิสเตอร์ แบงค์ 0

ตารางที่ 4.7 (ต่อ) รีจิสเตอร์ใช้งานเฉพาะ PSW (Program Status Word) เข้าถึงข้อมูลได้ในระดับบิต

			00 = Bank 0 ; Address 00H – 0
			01 = Bank 1 ; Address 08H – 0FH
			10 = Bank 2 ; Address 10H – 17H
			11 = Bank 3 ; Address 18H – 1FH
PSW.2	OV	D2H	Overflow Flag
PSW.1	-	D1H	Reserved
PSW.0	P	D0H	Even Parity Flag7.20

4.6 วงจรปรับระดับสัญญาณ (clamper circuit)

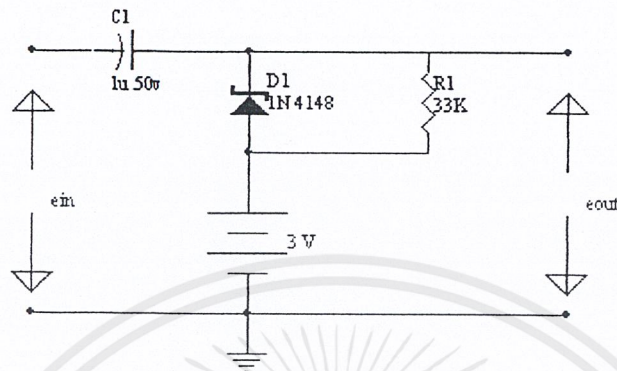
วงจรปรับระดับสัญญาณ (clamper circuit) เป็นวงจรไฟฟ้าซึ่งสามารถปรับหรือเปลี่ยนระดับของสัญญาณคลื่นไฟฟ้าได้โดยทั่วไปมักหมายถึงการเปลี่ยนแปลงหรือปรับระดับขนาดของสัญญาณกระแสสลับให้มีระดับที่ต้องการ โดยมีลักษณะของคลื่นสัญญาณกระแสตรงใดๆก็ได้ วงจรดังกล่าวนี้บางทีถูกเรียกว่า “วงจรเพิ่มกระแสตรง” (dc restorer) หรือ “วงจรเติมกระแสตรง” (DC inserter circuit) สัญญาณที่เอาต์พุตจะมีลักษณะเหมือนกับสัญญาณอินพุตทุกประการ นอกจากระดับกระแสตรงซึ่งเป็นระดับเปรียบเทียบเท่านั้นที่มีการเปลี่ยนแปลง ดังภาพที่ 4.2



ภาพที่ 4.2 แสดงคุณสมบัติและลักษณะของสัญญาณแรงดันรูปไซน์ที่อินพุตและเอาต์พุตของวงจรปรับระดับแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในปริณิธานิพนธ์นี้จะใช้วงจรปรับระดับแรงดันด้านเอาต์พุตให้อยู่ในระดับแรงดันค่าหนึ่งค่าใดที่ไม่ศูนย์ ซึ่งจากภาพที่ 4.2 เป็นวงจรที่ทำการเพิ่มแบตเตอรี่ ซึ่งมีแรงดันคงที่ E เข้าไปในวงจร แรงดันเอาต์พุตจะมีแรงดันค่าสุดท้ายอยู่ที่ระดับ E โวลต์ เท่ากับของแบตเตอรี่



ภาพที่ 4.3 วงจรที่ต่อใช้งานจริง

ซึ่งจากวงจรจะมีหลักการทำงาน คือ ขณะที่แรงดันอินพุตมีค่า -5 โวลต์ แรงดันนี้จะช่วยเสริมกับแรงดันของแบตเตอรี่ในวงจรและทำให้ไดโอดได้รับไบอัสตรง ไดโอดจึงทำงานคล้ายกับเป็นสวิตช์ปิด ดังนั้นแรงดันที่เอาต์พุตจะมีค่าเท่ากับแรงดันของแบตเตอรี่ คือ $+3$ โวลต์ ตัวเก็บประจุจะสะสมประจุทำให้มีแรงดันตกคร่อม 8 โวลต์ และทันทีที่แรงดันอินพุตเปลี่ยนจาก -5 โวลต์ เป็น $+5$ โวลต์ แรงดันนี้จะไปหักล้างกับแรงดันจากแบตเตอรี่ แต่จะเสริมกับแรงดันที่ตกคร่อมตัวเก็บประจุผลรวมแรงดันอินพุตและแรงดันที่ตกคร่อมตัวเก็บประจุ $(+5 + (+8))$ จะมีค่ามากกว่าแรงดันที่เอาต์พุตจึงมีค่าเท่ากับผลรวมของแรงดันอินพุตและแรงดันที่ตกคร่อมตัวเก็บประจุ $(+13$ โวลต์) นั่นคือที่ t_0 แรงดันเอาต์พุตเท่ากับ $+3$ โวลต์, ที่ t_{+0} และ t_{-1} แรงดันเอาต์พุตเท่ากับ $+13$ โวลต์, ที่ t_{+1} แรงดันอินพุตมีค่าเป็น -5 โวลต์ ไดโอดจะกลายเป็นสวิตช์ปิด แรงดันเอาต์พุตจึงเป็น $+3$ โวลต์

4.7 บิตพาริตี (Parity bit)

เนื่องจากการส่งผ่านข้อมูลทางสายโทรศัพท์นั้น สามารถเกิดสัญญาณรบกวนได้ง่าย ด้วยเหตุนี้จึงเป็นไปได้มากที่สถานะของแต่ละบิตของข้อมูลที่ถูกส่งจะมีการเปลี่ยนแปลง เช่น จากบิต 0 เป็นบิต 1 เป็นต้น ในระหว่างการส่งผ่านข้อมูล ด้วยเหตุนี้จึงมีการคิดค้นวิธีที่ง่ายที่สุดที่ช่วยในการตรวจสอบความถูกต้องของสถานะของบิตข้อมูล ซึ่งเรียกกันว่า การตรวจสอบค่าพาริตี (Parity Check) โดยในการส่งผ่านข้อมูลด้วยโปรโตคอล START/STOP นั้น ในการส่งตัวอักษรข้อความทั่วไปจะใช้เพียง 7 บิตข้อมูลเท่านั้น ดังนั้นจึงมีการเพิ่มพาริตีต่อสายเฟรมข้อมูลที่ถูกเอกสารเป็นเอกสารที่ส่งวนไปสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับว่าได้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างขึ้นมา ตามที่กล่าวมาแล้วในส่วนของบิตที่สิ้นสุดข้อมูล เพื่อตรวจสอบสถานะของผลบวกของบิตที่เป็น 1 ของสายบิตข้อมูลของแต่ละตัวอักษร สำหรับหลักการในการคำนวณค่าของบิตพาริตีมีดังนี้

1. ถ้าหากจำนวนของบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคู่ (Even number) บิต พาริตี จะมีค่าเท่ากับ 0
2. ถ้าหากจำนวนของบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคี่ (Odd number) พาริตีจะมีค่าเท่ากับ 1



บทที่ 5

พอร์ตอนุกรม RS232

คำจำกัดความต่าง ๆ ที่ควรรู้เกี่ยวกับการสื่อสารผ่านพอร์ตอนุกรมมีดังนี้

- Baud Rate คือ อัตราความเร็วในการส่งถ่ายข้อมูลของพอร์ท ในอดีตพอร์ทจะมีอัตราความเร็วในการส่งถ่ายข้อมูล 1 บิต ต่อ อัตรา 1 Baud แต่ในปัจจุบันหลังจากที่พอร์ทได้รับการพัฒนาไปอย่างมาก อัตราความเร็วในการส่งถ่ายข้อมูล 1 บิต จะไม่เท่ากับอัตรา 1 Baud อีกต่อไป เช่น ตามมาตรฐาน V.22bis นั้นอัตรา 1 baud จะเท่ากับ 4 บิต ดังนั้นพอร์ทที่มีอัตราความเร็ว 9,000 Baud จะเท่ากับ อัตราความเร็วในการส่งถ่ายข้อมูล 38,400 บิตต่อวินาที เป็นต้น
- Bit Per Second คืออัตราความเร็วในการส่งถ่ายจำนวนบิตของพอร์ท ในปัจจุบันพอร์ทจะมีค่าอัตราความเร็วในการส่งถ่ายจำนวนบิตมาตรฐานอยู่ที่ 9,600 BPS
- Through Put จำนวนของบิตที่สามารถส่งผ่านพอร์ทได้จริง ๆ ภายในช่วงเวลา 1 วินาที ในทางปฏิบัติแม้ว่าพอร์ทจะมีความเร็วมากน้อยเพียงใดก็ตาม อัตราการส่งผ่านข้อมูลผ่านทางพอร์ทในแต่ละช่วงเวลาอาจจะมีค่าไม่เท่ากันตลอดก็ได้ ทั้งนี้ขึ้นกับปัญหาต่าง ๆ เช่น การเกิดสัญญาณรบกวนหรือความไม่พร้อมของโฮสต์ปลายทาง เป็นต้น ดังนั้น อัตราการส่งผ่านข้อมูลอาจจะต่ำกว่าความสามารถในการส่งผ่านข้อมูลของพอร์ทก็ได้

5.1 พอร์ตอนุกรม (Serial Port)

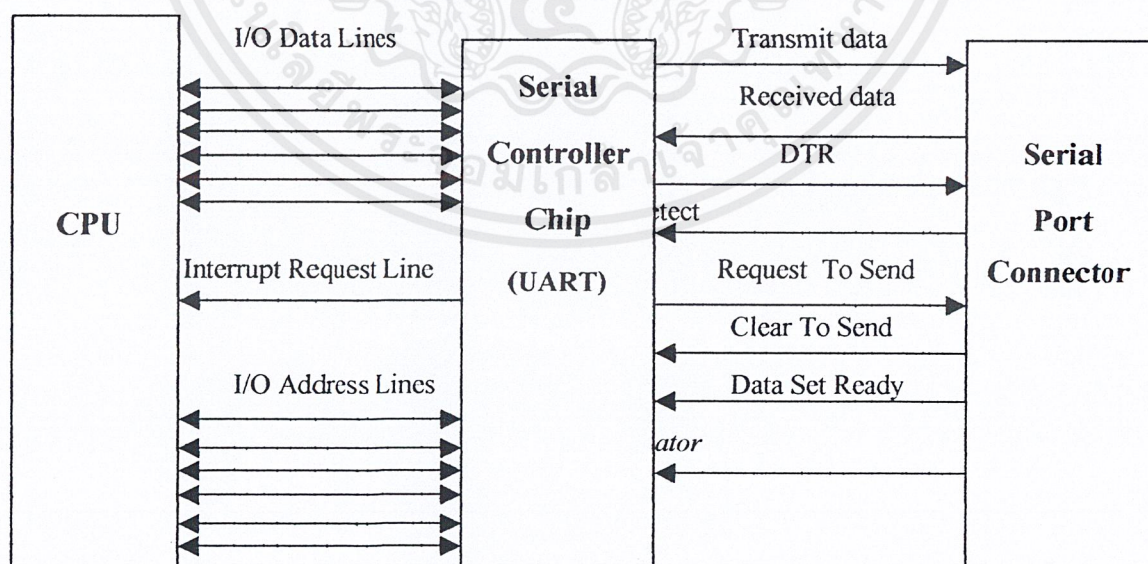
เนื่องจากในปัจจุบันมีการใช้งานตามมาตรฐานการเชื่อมต่อแบบ RS-232-C กันอย่างแพร่หลาย ดังนั้นจึงขอกล่าวรายละเอียดเฉพาะมาตรฐานการเชื่อมต่อแบบ RS-232-C เท่านั้น ซึ่งเป็นมาตรฐานที่ถูกกำหนดโดย EIA ซึ่งเป็นองค์กรอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกา โดยแบ่งการเชื่อมต่อเป็น 2 ลักษณะ คือ DTE (Data Terminal Equipment) และ DCE (Data Communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เสมอ เช่น การต่อเครื่องคอมพิวเตอร์ (อุปกรณ์ DTE) เข้ากับพอร์ท (อุปกรณ์ DCE) เป็นต้น

พอร์ตอนุกรม RS-232-C จะเป็นพอร์ทของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 และ 25 ขา และเราเรียกกันว่า พอร์ท COM1: และ COM2: นั่นเอง ในความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิป UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่ง ซึ่งในปัจจุบันเบอร์ที่ใช้กันมากที่สุดคือเบอร์ 16550C ซึ่งเป็นเวอร์ชันที่ได้รับการแก้ไขข้อผิดพลาดแล้ว ซึ่งชิป UART นี้จะทำหน้าที่ในการรับและส่งข้อมูลดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การส่งข้อมูล (Data Transmission)
 1. รับตัวอักษรจากเครื่องคอมพิวเตอร์
 2. แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่าขบวนการ serialization)
 3. สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต START, STOP และ PARITY เป็นต้น
 4. ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมา ด้วยความเร็วของ พอร์ตอนุกรม (baud rate)
 5. แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์
- การรับข้อมูล (Data Receiver)
 1. รับตัวอักษรจากอินเตอร์เฟซ
 2. ตรวจสอบความถูกต้องของเฟรมข้อมูล ตามมาตรฐานเฟรมที่กำหนด โดยถ้าหากเฟรมข้อมูลมีรูปแบบที่ไม่ถูกต้องก็จะมีอาการแจ้งข้อผิดพลาดทันที
 3. ตรวจสอบความถูกต้องของพาริตี
 4. แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร
 5. ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
 6. แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟซ

สำหรับการเชื่อมต่อสายสัญญาณต่าง ๆ ระหว่าง CPU ของเมนบอร์ดของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรมนั้น จะต้องกระทำผ่านทางชิป UART ซึ่งจะมีวิธีการเชื่อมต่อดังในรูปข้างล่าง



ภาพที่5.1 แสดงการเชื่อมต่อสายสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่าการเชื่อมต่อระหว่าง UART กับพอร์ทอนุกรมนั้น มีสายสัญญาณมากมายที่ช่วยให้การสื่อสารมีความถูกต้องมากขึ้น โดยที่สายสัญญาณแต่ละเส้นมีความหมายดังต่อไปนี้

- Transmit Data (TD)
เป็นวงจรที่สร้างสัญญาณ Transmitted Data ซึ่งถูกส่งจาก DTE ไปยัง DCE ซึ่งสัญญาณที่ส่งออกมาอาจจะเป็น โคลด์คำสั่งของพอร์ทหรือข้อมูลก็ได้
- Received Data (RD)
เป็นวงจรที่สร้างสัญญาณ Received data ซึ่งถูกส่งจาก DCE ไปยัง DTE ซึ่งสัญญาณที่ส่งออกมาอาจจะเป็น โคลด์คำสั่งของพอร์ทหรือข้อมูลก็ได้ ซึ่งเป็นสัญญาณที่มีทิศทางการส่งตรงข้ามกับสัญญาณ Transmitted Data
- Data Terminal Ready (DTR)
สัญญาณ DTR จะถูกส่งจาก DTE ไปยัง DCE เพื่อเป็นการแจ้งความพร้อมในการสื่อสารให้อุปกรณ์ได้ทราบ โดยถ้าหากพอร์ทนั้น (อุปกรณ์ DCE) มีความสามารถในการตอบรับแบบอัตโนมัติ (Automatically answer) อุปกรณ์นั้นก็จะสามารถตอบรับได้เฉพาะเมื่อสัญญาณ DTR อยู่ในสถานะ on เท่านั้น
- Carrier Detect (CD)
สัญญาณ CD จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งว่าพอร์ทอยู่ในสถานะกำลังติดต่อกับพอร์ทตัวอื่น หรือพอร์ทกำลังได้รับสัญญาณที่พร้อมสำหรับการติดต่อสื่อสาร สำหรับสัญญาณ Carrier Detect นี้ สามารถเรียกอีกชื่อได้ว่า Received Line Signal Detector
- Request to Send (RTS)
สัญญาณ RTS จะถูกส่งจาก DTE ไปยัง DCE โดยเมื่อสัญญาณ RTS อยู่ในสถานะ on ก็หมายถึงเครื่องคอมพิวเตอร์พร้อมที่จะรับข้อมูลจากพอร์ท และในทางกลับกันถ้าหากสัญญาณ RTS อยู่ในสถานะ off ก็หมายถึง เครื่องคอมพิวเตอร์ไม่พร้อมที่จะรับข้อมูลจากพอร์ท
- Clear to Send (CTS)
สัญญาณ CTS จะถูกส่งจาก DCE ไปยัง DTE ซึ่งเป็นสัญญาณที่ทำหน้าที่ตรงกันข้ามกับสัญญาณ RTS โดยเมื่อสัญญาณ CTS อยู่ในสถานะ on ก็หมายถึงพอร์ทพร้อมที่จะรับข้อมูลจากเครื่องคอมพิวเตอร์และในทางกลับกันถ้าหากสัญญาณ CTS อยู่ในสถานะ off ก็หมายถึง พอร์ทไม่พร้อมที่จะรับข้อมูลจากเครื่องคอมพิวเตอร์
- Data Set Ready (DSR)
สัญญาณ DSR จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งความพร้อมในการสื่อสารจากพอร์ทให้กับเครื่องคอมพิวเตอร์ได้ทราบ โดยสัญญาณ DSR จะอยู่ในสถานะ on ก็ต่อเมื่อพอร์ทได้รับสัญญาณ DTR เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Ring Indicator (RI)

สัญญาณ RI จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งให้เครื่องคอมพิวเตอร์ทราบว่า พอร์ตกำลังได้รับสัญญาณกระดิ่ง (ring signal) จากอุปกรณ์ตัวอื่น โดยที่สัญญาณ RI จะอยู่ในสถานะ on พอร์ตได้รับสัญญาณกระดิ่ง และ off เมื่อพอร์ตรับไม่ได้รับสัญญาณกระดิ่ง เนื่องจากอุปกรณ์รุ่นใหม่ ๆ ในปัจจุบันมักจะสามารถในการตอบรับแบบอัตโนมัติ (Automatically Answer) ดังนั้นจึงไม่มีความจำเป็นต้องใช้สัญญาณ RI อีกต่อไป

5.2 รูปแบบของเฟรมข้อมูล (Data Format)

ในการสื่อสารระหว่างอุปกรณ์ 2 ตัวนั้น อุปกรณ์ทั้งสองจะต้องมีความเข้าใจถึงรูปแบบของเฟรมข้อมูลที่ตรงกัน ซึ่งการส่งข้อมูลแบบ Asynchronous นั้น นอกจากข้อมูลที่จะต้องส่งผ่านแล้ว อุปกรณ์นั้นยังต้องจัดการช่วงจังหวะในการส่งหรือรับข้อมูลระหว่างกันอีกด้วย โดยที่เฟรมข้อมูลในการส่งหรือรับจะประกอบด้วยบิตข้อมูลที่มีความหมายดังต่อไปนี้

- บิตข้อมูล (Data Bit)

เมื่อชิป UART ได้รับตัวอักษรที่ส่งมาจากเครื่องคอมพิวเตอร์แล้ว ก็ต้องทำการแปลงตัวอักษรดังกล่าวให้เป็นสายข้อมูลชนิดบิตที่มีความยาวตั้งแต่ 5 ถึง 8 บิต ซึ่งเราเรียกขบวนการเปลี่ยนแปลงตัวอักษรให้เป็นสายข้อมูลชนิดบิตนี้ว่า ขบวนการ Serialization จากนั้นพอร์ตรก็จะทำการส่งแต่ละบิตไปยังอุปกรณ์ปลายทาง โดยจะเริ่มต้นส่งจากบิตที่มีนัยสำคัญต่ำสุด (least significant bit) ไปยังบิตที่มีนัยสำคัญสูงสุด (most significant bit)

- บิตเริ่มต้นข้อมูล (START bit)

ในการส่งข้อมูลแบบ Asynchronous นั้น เราจะต้องมีวิธีการบอกพอร์ตรให้ทราบถึงจุดเริ่มต้นของข้อมูลที่ต้องการส่ง ดังนั้นก่อนหน้าข้อมูลในทุก ๆ เฟรมจะต้องถูกกำหนดด้วยบิตเริ่มต้นข้อมูล START bit เสมอ

- บิตสิ้นสุดข้อมูล (STOP bit)

ในการส่งข้อมูลแบบ Asynchronous นั้น ในกรณีที่บิตเริ่มต้นข้อมูลเกิดการสูญหายในระหว่างการส่งอุปกรณ์ปลายทางก็จะไม่สามารถทราบถึงจุดสิ้นสุดของสายข้อมูลบิตได้เลย นอกเสียจากอุปกรณ์นั้นจะตรวจพบบิตเริ่มต้นข้อมูลใหม่อีกครั้งเท่านั้น ดังนั้น จึงมีการเพิ่มบิตสิ้นสุดข้อมูลต่อท้ายทุก ๆ ข้อมูลแต่ละตัวอักษรเพื่อแจ้งการสิ้นสุดของสายข้อมูลบิต โดยที่การเรียงกันของบิตเริ่มต้นข้อมูล สายข้อมูล และบิตสิ้นสุดข้อมูลเรียกว่า เฟรมของข้อมูล (data frame)

- บิตพาริตี (Parity bit)

เนื่องจากการส่งผ่านข้อมูลทางสายโทรศัพท์นั้น สามารถเกิดสัญญาณรบกวนได้ง่าย ด้วยเหตุนี้จึงเป็นไปได้มากที่สถานะของแต่ละบิตของข้อมูลที่ถูกส่งจะมีการเปลี่ยนแปลง เช่น จากบิต 0 เป็นบิต 1 เป็นต้น ในระหว่างการส่งผ่านข้อมูล ด้วยเหตุนี้จึงมีการคิดค้นวิธีที่ง่ายที่สุดที่ช่วยในการตรวจสอบความถูกต้องของสถานะของบิตข้อมูล ซึ่งเรียกกันว่า การตรวจสอบค่าพริตี้ (Parity Check) โดยในการส่งผ่านข้อมูลด้วยโปรโตคอล START/STOP นั้น ในการส่งตัวอักษรข้อความทั่วไปจะใช้เพียง 7 บิตข้อมูลเท่านั้น ดังนั้นจึงมีการเพิ่มพริตี้คือสายเฟรมข้อมูลที่ถูกสร้างขึ้นมา ตามที่กล่าวมาแล้วในส่วนของบิตที่สิ้นสุดข้อมูล เพื่อตรวจสอบสถานะของผลบวกของบิตที่เป็น 1 ของสายบิตข้อมูลของแต่ละตัวอักษร สำหรับหลักการในการคำนวณค่าของบิตพริตี้มีดังนี้

1. ถ้าหากจำนวนของบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคู่ (Even number) บิตพริตี้ จะมีค่าเท่ากับ 0
2. ถ้าหากจำนวนของบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคี่ (Odd number) พริตี้จะมีค่าเท่ากับ 1

บทที่ 6

Microsoft Visual Basic

ในปัจจุบันระบบปฏิบัติการในลักษณะของ Windows ได้เข้ามาแทนที่ระบบปฏิบัติการในลักษณะเดิม ซึ่งส่วนใหญ่ที่นิยมใช้กันอยู่คือ MS-DOS เนื่องจากรูปแบบของจอภาพที่ใช้ติดต่อระหว่างคอมพิวเตอร์และผู้ใช้ อยู่ในรูปแบบของ Graphic User Interface (GUI) ที่ใช้รูปภาพแทนคำสั่งต่างๆ แทน ซึ่งต่างจาก MS-DOS ที่รูปแบบของคำสั่งจะอยู่ในรูปแบบของตัวอักษร และเป็นแบบป้อนทีละบรรทัด หรือที่เรียกว่า “Command Line” ซึ่งผู้ใช้จะต้องเรียนรู้ และจดจำรูปแบบของแต่ละคำสั่งให้ถูกต้องและแม่นยำ จึงจะใช้งานโปรแกรมต่างๆ ได้เป็นอย่างมีประสิทธิภาพและด้วยเหตุนี้ ได้ส่งผลต่อการพัฒนาโปรแกรมเช่นเดียวกัน เนื่องจากโปรแกรมเมอร์ ซึ่งแต่เดิมการพัฒนาโปรแกรมอยู่บน MS-DOS ต้องเปลี่ยนแปลงรูปแบบและแนวความคิด และหันมาพัฒนาโปรแกรมบน Windows แทน

การพัฒนาโปรแกรมบน Windows ในปัจจุบัน กระทำได้ง่าย และสะดวกขึ้น เนื่องจากมีการใช้เทคโนโลยีทางด้าน Visualize เข้ามาประกอบในการออกแบบจอภาพ ซึ่งต่างจากในยุคแรกๆ ที่การพัฒนาโปรแกรมบน Windows นั้นค่อนข้างจะทำได้ยาก เนื่องจากการพัฒนาโปรแกรมหนึ่งๆ ให้แล้วเสร็จ โปรแกรมเมอร์จะต้องเขียน Routine ต่างๆ ขึ้นเป็นจำนวนมาก ซึ่ง Visual Basic ก็จัดเป็นภาษาหนึ่งที่ได้รับคามนิยม และถูกนำมาใช้ในการพัฒนาโปรแกรมเพื่อใช้งานบน Windows

6.1 สาเหตุที่ต้องใช้ Visual Basic

Visual Basic เป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมนำมาใช้ในการเขียนโปรแกรมบน Windows เนื่องจาก เป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize ซึ่งเพียงแค่เลือก Control ที่เหมาะสม แล้ววางลงบน Form ก็สามารถสร้างจอภาพที่ใช้สำหรับการติดต่อกับผู้ใช้ รวมทั้งการใช้เทคนิคการเขียนโปรแกรมแบบ Event-driven ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนในการทำงานให้กับ Control ต่างๆ ที่สร้างขึ้นตามเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น การเลื่อนเมาส์ หรือการรับข้อมูลจากคีย์บอร์ด ฯลฯ เป็นต้น ประกอบกับภาษาที่ใช้เขียนโปรแกรม เป็นภาษา Basic ซึ่งเป็นภาษาคอมพิวเตอร์ ที่ผู้ใช้งานคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่คุ้นเคย จึงส่งผลให้ การพัฒนาโปรแกรมบน Windows ด้วย Visual Basic มีขั้นตอนน้อย กระทำได้ง่าย และสะดวกต่อการใช้งาน จึงทำให้ผู้ใช้สามารถเรียนรู้ได้ในระยะเวลาอันสั้น ก็สามารถที่จะพัฒนาโปรแกรมบน Windows ขึ้นเป็นโปรแกรมแรกได้

Visual Basic นี้เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมขึ้นใช้งาน ที่ใช้ได้ตั้งแต่ผู้ใช้ระดับต้น เพื่อใช้สร้างโปรแกรมง่ายๆ บน Windows หรือโปรแกรมเมอร์ระดับกลาง ที่จะเรียกใช้เอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันการทำงานต่างๆ ของ Visual Basic ได้อย่างมีประสิทธิภาพ ตลอดจนโปรแกรมเมอร์ในระดับมืออาชีพ ที่จะพัฒนาโปรแกรมในระดับสูง โดยใช้ Object Linking and Embedding (OLE) และ Application Programming Interface (API) ของ Windows มาประกอบในการเขียนโปรแกรม

6.2 คอมพิวเตอร์ที่เหมาะสมสำหรับใช้งาน VB 6

สำหรับเครื่องคอมพิวเตอร์ที่ใช้งาน VB 6 จะต้องเป็นเครื่องคอมพิวเตอร์ที่มีรายละเอียดดังต่อไปนี้

- ต้องมีระบบปฏิบัติการ Windows 95 หรือสูงกว่า หรือระบบปฏิบัติการ Windows NT 4.0 หรือสูงกว่า
- หน่วยประมวลผลกลางรุ่น Pentium 90 MHz หรือสูงกว่า
- เนื้อที่ในฮาร์ดดิสก์อย่างน้อย 50 MB
- ต้องการ CD-ROM ไดรฟ์
- การ์ดจอ VGA 640 X 480 หรือสูงกว่าที่ Windows สนับสนุน
- RAM 24 MB สำหรับ Windows 95, 32 MB สำหรับ Windows NT 4.0

6.3 คอนโทรลที่ใช้ในโครงการ

6.3.1 Label

Label1

เป็นคอนโทรลที่ใช้สำหรับเขียนข้อความบนจอภาพ ซึ่งมี Property ต่างๆ ที่น่าสนใจดังนี้

ตารางที่ 6.1 Property ที่สำคัญของ Label

ชื่อ Property	หน้าที่
Caption	ใช้สำหรับกำหนดข้อความให้แสดงบน Object
Font	ใช้สำหรับกำหนดรูปแบบของตัวอักษรของ Object
ForeColor	ใช้สำหรับกำหนดสีของตัวอักษรของ Object
Alignment	ใช้สำหรับกำหนดรูปแบบในการแสดงผลของข้อความที่กำหนดใน Property “Caption”
BackColor	ใช้สำหรับกำหนดสีฉากหลังของ Object

6.4.2 TextBox

TextBox

เป็นคอนโทรลที่ใช้สำหรับรับข้อมูลจากคีย์บอร์ด ซึ่งมี Property ต่างๆ ที่น่าสนใจดังนี้

ตารางที่ 6.2 Property ที่สำคัญของ TextBox

Property	Description
Text	ใช้สำหรับรับข้อมูลจากคีย์บอร์ดและแสดงผล
Width	ใช้สำหรับกำหนดความยาวของ Object
Height	ใช้สำหรับกำหนดความสูงของ Object
Left	ใช้สำหรับกำหนดตำแหน่งเริ่มต้นของ Object ในแนวนอน X
Top	ใช้สำหรับกำหนดตำแหน่งเริ่มต้นของ Object ในแนวแกน Y
MaxLength	ใช้สำหรับกำหนดจำนวนตัวอักษรสูงสุดที่จะรับได้ใน Object
Multiline	เป็นข้อมูลชนิดตรรกะ ซึ่งจะมีค่าเป็นจริง (True) หรือ (False) โดยเมื่อกำหนดให้มีค่าเป็นจริง จะทำให้สามารถรับข้อมูลได้หลายบรรทัดในกรณีที่ใช้กับข้อมูลชนิด Memo

6.4.3 CommandButton

Command1

ใช้เป็นปุ่มต่างๆ บน Form ซึ่งมี Property ต่างๆ ที่น่าสนใจดังนี้

ตารางที่ 6.3 Property ที่สำคัญของ CommandButton

Property	Description	
Enabled	ใช้กำหนดให้ Object นั้นทำงาน (กำหนดให้มีค่าเป็น True) หรือ ไม่ทำงาน (กำหนดให้มีค่าเป็น False)	
Picture	ใช้แสดงรูปภาพบนปุ่ม	
	0 - Standard	เป็นปุ่มที่มีเฉพาะข้อความบนปุ่มเท่านั้น
	1 – Graphical	เป็นปุ่มที่มีทั้งรูปภาพและข้อความบนปุ่ม โดยจะต้องใช้คู่กับ Property “Picture” ในการแสดงภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.4 Timer



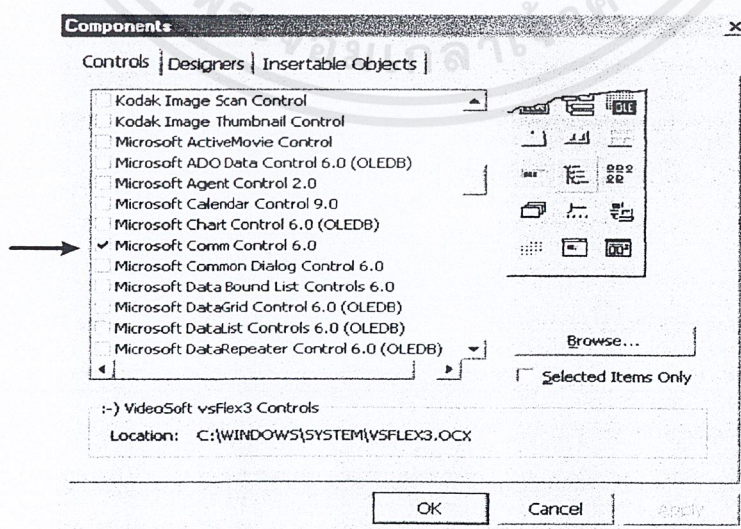
เป็นคอนโทรลที่จะไม่ปรากฏบนจอภาพเมื่อทำการ Run โดยทำหน้าที่เป็นตัวจับเวลาเพื่อกำหนดการทำงานของโปรแกรมตามเวลาที่กำหนด ซึ่ง Event ที่มักใช้ได้แก่ Event “Timer” ในการทำงานของ Timer จะใช้ช่วงเวลาหรือที่เรียกว่า Interval เป็นตัวกำหนดการทำงาน (Interval คือ ช่วงเวลาในหน่วย มิลลิวินาที)

6.3.5 Microsoft Comm Control



คอนโทรล MSComm (Communication) เป็นคอนโทรลตัวหนึ่งที่จะช่วยในการติดต่อกับพอร์ทอนุกรม (Serial Port) ซึ่งสามารถทำการส่งข้อมูลผ่านพอร์ทอนุกรมได้ด้วยคอนโทรลนี้ เช่น การติดต่อโดยตรงกับบอร์ดอิเล็กทรอนิกส์ ซึ่งคอนโทรล MSComm ที่มากับ Visual Basic จะเป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบ Even-driven นั่นก็คือคอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือการร้องขอให้เกิดเหตุการณ์ต่างๆ กับพอร์ทอนุกรมโดยอัตโนมัติ และจะมีการแจ้งเตือนให้ทราบทางโพรซีเยอร์เหตุการณ์เช่นเดียวกับคอนโทรลทั่วไปของ Visual Basic นั่นเอง ดังนั้นในการเขียนโค้ดจึงไม่จำเป็นต้องสร้างโพรซีเยอร์ที่ทำหน้าที่คอยตรวจสอบเหตุการณ์ต่างๆ ของพอร์ทอนุกรม ซึ่งจะทำให้โค้ดการใช้งานเป็นอย่างมาก

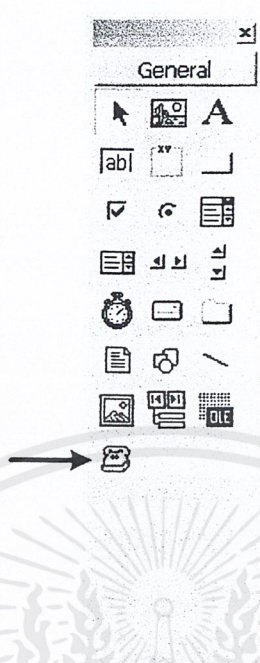
การเพิ่มคอนโทรล Microsoft Comm Control ทำได้โดยการเรียกหน้าต่าง Components แล้วเลือกที่ Microsoft Comm Control 6.0 ดังภาพที่ 6.1



ภาพที่ 6.1 การเพิ่ม Component Microsoft Comm Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะได้ MSComm เพิ่มเข้ามาใน Toolbox ดังรูป



ภาพที่ 6.2 รูปแสดงหลังการเพิ่ม Component MSComm

คอนโทรล MSComm จะมีหน้าที่มาตรฐานหลักๆ สำหรับการสื่อสารผ่านพอร์ตอนุกรม 3 ประการดังต่อไปนี้

- หมุนหมายเลขติดต่อกับโทรศัพท์ปลายทางที่กำหนด
- ตรวจสอบการเข้ามาของข้อมูลยังพอร์ตอนุกรมโดยอัตโนมัติ
- ส่งข้อมูลตามที่กำหนดจากโปรแกรมไปยังพอร์ตอนุกรม

เหตุการณ์ที่สำคัญของคอนโทรล MSComm คือ CommEvent จะรายงานเหตุการณ์ทุกครั้งที่เกิดข้อผิดพลาดหรือมีการสื่อสาร ซึ่งสามารถอ่านค่าได้เฉพาะในขณะที่ทำงานเท่านั้น รูปแบบคำสั่งคือ

Object. CommEvent

คอนโทรล MSComm จะมีการเรียกโปรซีเยอร์เหตุการณ์ OnComm ทุกครั้งที่เกิดข้อผิดพลาดหรือมีการสื่อสารเกิดขึ้น ซึ่งค่าตัวเลขที่เป็นจำนวนเต็ม แสดงถึงข้อผิดพลาดหรือเหตุการณ์ที่มีการสื่อสารดังกล่าว ก็จะถูกจัดเก็บเอาไว้ในคุณสมบัติ CommEvent เสมอ ดังนั้นถ้าต้องการตรวจสอบข้อผิดพลาดหรือเหตุการณ์ที่มีการสื่อสารภายในโปรซีเยอร์เหตุการณ์ OnComm ก็ควรที่จะใช้ค่าตัวเลขจากคุณสมบัติ CommEvent ในการตรวจสอบเสมอ

Property ที่สำคัญมีดังนี้

CommPort

เป็นตัวรายงานหรือกำหนดหมายเลขของพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ที่ต้องการ

ติดต่อ สำหรับหมายเลขของพอร์ตอนุกรมสามารถมีค่าได้ตั้งแต่ 1 ถึง 16 (ค่าปกติจะเท่ากับ 1) ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะเปิดพอร์ตด้วยคุณสมบัติ Port Open ต้องกำหนดหมายเลขของพอร์ตอนุกรมให้กับคุณสมบัติ CommPort เสียก่อน โดยถ้าหากหมายเลขของพอร์ตอนุกรมที่กำหนดให้กับคุณสมบัติ CommPort ไม่เป็นความจริง ก็จะเกิดข้อผิดพลาด 68 (Device unavailable) ทันที ซึ่งในกรณีนี้สามารถแก้ไขได้โดยการกำหนดหมายเลขของพอร์ตอนุกรมที่ถูกต้องเสียใหม่ แล้วจึงทำการเปิดพอร์ตอนุกรมอีกครั้งด้วยคุณสมบัติ Port Open

InBufferSize

เป็นตัวรายงานหรือกำหนดขนาดของบัฟเฟอร์ด้านรับเข้า ซึ่งมีหน่วยเป็นไบต์ (โดยปกติ 1 ไบต์จะเท่ากับ 1 ตัวอักษร) ในการเลือกขนาดของบัฟเฟอร์ด้านรับเข้าที่เหมาะสมนั้น ในทางปฏิบัติเป็นสิ่งที่ยากมาก ทั้งนี้ขึ้นอยู่กับความหนาแน่นของข้อมูลที่มีการส่งมาในแต่ละครั้งและความเร็วของการสื่อสาร (Transmission rate) ของพอร์ต ซึ่งปกติโปรแกรมเมอร์โดยทั่วไปจะกำหนดให้มีค่าเท่ากับ 1024 ไบต์ (1 KB) โดยถ้าหากเกิดข้อผิดพลาด overflow ในขณะรันแอปพลิเคชัน ก็จะทำให้การเพิ่มขนาดของบัฟเฟอร์ด้านรับเข้าต่อไป

InputMode

คือ การกำหนดชนิดของข้อมูลที่จะถูกอ่านโดยคุณสมบัติ Input จากบัฟเฟอร์ด้านรับเข้า โดย comInputModeText เป็นข้อมูลชนิดข้อความทั่วไป (default) และ comInputModeBinary เป็นข้อมูลชนิดไบนารี

การกำหนดชนิดของข้อมูลที่จะถูกอ่านโดยคุณสมบัติ Input ก็ขึ้นกับลักษณะของงานที่กำลังควบคุม โดยปกติถ้าหากข้อมูลชนิดตัวอักษร ANSI ทั่วไป ก็จะกำหนดให้เป็นข้อมูลชนิดข้อความ (comInputModeText) แต่ถ้าหากเป็นข้อมูลที่ประกอบด้วยตัวอักษรควบคุม (แอสกีตั้งแต่ 0 ถึง 31) ก็จะกำหนดให้เป็นข้อมูลชนิด (comInputModeBinary)

OutBufferSize

เป็นการกำหนดขนาดของบัฟเฟอร์ด้านส่งออกซึ่งมีหน่วยเป็นไบต์ (โดยปกติ 1 ไบต์จะเท่ากับ 1 ตัวอักษร) ปกติซอฟต์แวร์จะกำหนดให้มีขนาดเท่ากับ 512 ไบต์

ในการเลือกขนาดของบัฟเฟอร์ด้านส่งออกที่เหมาะสมนั้น ในทางปฏิบัติเป็นสิ่งที่ยากมากเช่นกัน ทั้งนี้ก็ขึ้นอยู่กับข้อมูลที่ส่งไปมาในแต่ละครั้งและความเร็วของการสื่อสารของพอร์ต ซึ่งโดยปกติโปรแกรมเมอร์โดยทั่วไปจะกำหนดให้มีค่าเท่ากับ 512 ไบต์ โดยถ้าหากเกิดข้อผิดพลาด overflow ในขณะรันแอปพลิเคชัน ก็จะทำให้การเพิ่มขนาดของบัฟเฟอร์ด้านรับเข้าต่อไป

Rthreshold

คือ การกำหนดจำนวนตัวอักษรที่จะรับเข้าก่อนที่คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเท่ากับ comEvReceive และมีการเรียกใช้โพธิ์เซอร์เหตุการณ์ OnComm

ถ้าหากคุณสมบัติ Rthreshold มีค่าเท่ากับ 0 (default) ก็เป็นการเรียกโปรซีเยอร์เหตุการณ์ OnComm เมื่อมีการรับตัวอักษรเข้ามายังบัฟเฟอร์ด้านรับเข้า และในทางกลับกัน ถ้าหากคุณสมบัติ Rthreshold มีค่าเท่ากับ 1 ก็จะมีการเรียกโปรซีเยอร์ เหตุการณ์ OnComm ทุกครั้งที่มีการรับตัวอักษรมายังบัฟเฟอร์ด้านรับเข้า

Setting

คือ การกำหนดพารามิเตอร์ในการสื่อสารผ่านพอร์ทอนุกรม การกำหนดพารามิเตอร์ในการสื่อสารผ่านพอร์ทอนุกรมมีดังต่อไปนี้

รูปแบบของการกำหนดลำดับของพารามิเตอร์ในการสื่อสารผ่านพอร์ทอนุกรม สำหรับคุณสมบัติ Setting จะต้องเรียงลำดับดังนี้ “BBBB, P,D,S” (โดยปกติทุกๆ ไปจะมีค่าเท่ากับ “9600,N,8,1”) เพราะถ้าหากลำดับไม่ถูกต้องจะทำให้เกิดข้อผิดพลาดหมายเลข 380 (Invalid property value) ทันที ซึ่งสัญลักษณ์แต่ละตัวมีความหมายดังนี้

BBBB คือ ความเร็วของการส่งถ่ายข้อมูลในหน่วยของbaudrateซึ่งในทางปฏิบัติ 1baud rate อาจจะมีค่าเท่ากับ 1 bps (Bit per second) หรือมากกว่าก็ได้ สำหรับค่าของ baud rate ที่คอนโทรล MSComm สามารถรับได้มีค่าดังต่อไปนี้ 110, 300, 600, 1200, 2400, 9600(default), 14400, 19200, 28800, 38400(reserved), 56000(reserved), 128000(reserved) , 256000(reserved)

P คือ บิตพาริตี (parity bit) สำหรับใช้ในการตรวจสอบความถูกต้องของข้อมูล

โดยกำหนด

E คือ Even parity

M คือ Mark parity

N คือ None(default)

O คือ Odd parity

S คือ Space

D คือ ขนาดของบิตข้อมูล ซึ่งสามารถกำหนดได้เป็น 4, 5, 6, 7 หรือ 8(default)

S คือ ขนาดของบิตหยุด (stop bit) ซึ่งสามารถกำหนดได้เป็น 1(default),1.5,2

SThreshold

เป็นการกำหนดจำนวนตัวอักษรที่น้อยที่สุดที่ถูกจัดเก็บในบัฟเฟอร์ด้านส่งออก ก่อนที่คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเท่ากับ comEvSend และมีการเรียกโปรซีเยอร์เหตุการณ์ OnComm

ถ้าหากคุณสมบัติ SThreshold มีค่าเท่ากับ 0(default) ก็จะเป็นการยกเลิกการเรียกโปรซีเยอร์เหตุการณ์ OnComm เมื่อมีการส่งตัวอักษรไปยังบัฟเฟอร์ด้านส่งออก และในทางกลับกันถ้าหากคุณสมบัติ SThreshold มีค่าเท่ากับ 1 ก็จะมีการเรียกโปรซีเยอร์เหตุการณ์ OnComm เมื่อบัฟเฟอร์ด้านส่งออกว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยถ้าหากจำนวนของตัวอักษรในบัฟเฟอร์ด้านส่งออกน้อยกว่าค่าตัวเลขที่กำหนด คุณสมบัติ CommEvent ก็จะมีค่าเท่ากับ comEvSend และพร้อมทั้งเกิดการเรียกโปรซีเยอร์เหตุการณ์ OnComm ทันที เช่น สมมติให้คุณสมบัติ Sthreshold มีค่าเท่ากับ 10 และถ้าจำนวนตัวอักษรในบัฟเฟอร์ด้านส่งออกลดลงจาก 10 เป็น 9 ก็จะเกิดเหตุการณ์ OnComm ทันที เป็นต้น

คำสั่งที่สำคัญ คือ

PortOpen คือ การกำหนดสถานะการเปิด (open) หรือปิด(close) ของพอร์ทอนุกรมของเครื่องคอมพิวเตอร์ มีรูปแบบการใช้งานคือ

Object.PortOpen[=Boolean]

Boolean หมายถึงข้อมูลชนิดบูลีน ที่กำหนดสถานะของพอร์ทอนุกรม ดังนี้

True หมายถึง พอร์ทอนุกรมถูกเปิด

False หมายถึง พอร์ทอนุกรมถูกปิด (default)

พอร์ทอนุกรมถูกปิดโดยอัตโนมัติโดยคอนโทรล MSComm เมื่อแอปพลิเคชันสิ้นสุดการทำงาน โดยถ้าหากหมายเลขของพอร์ทอนุกรมที่กำหนดให้เปิดไม่มีการติดตั้งอยู่จริง ก็จะเกิดข้อผิดพลาดหมายเลข 68 (Device unavailable) ทันที

Input เป็นการอ่านข้อมูลจากบัฟเฟอร์ พร้อมทั้งลบข้อมูลในบัฟเฟอร์ด้านรับเข้าทิ้ง ซึ่งสามารถอ่านค่าได้เฉพาะในขณะที่ทำงานเท่านั้น รูปแบบการใช้งาน คือ

Object.Input

ทุกครั้งที่มีการใช้คุณสมบัติ Input ในการอ่านข้อมูลจากบัฟเฟอร์ด้านรับเข้านั้น จำนวนของตัวอักษรที่อ่านได้จะถูกกำหนดลงในคุณสมบัติ InputLen ทันที ซึ่งถ้ากำหนดให้คุณสมบัติ InputLen มีค่าเท่ากับ 0 ก็จะหมายถึงการกำหนดให้คุณสมบัติ อ่านข้อมูลทั้งหมดจากบัฟเฟอร์ด้านรับเข้านั่นเอง ซึ่งชนิดของข้อมูลที่อ่านโดยคุณสมบัติ Input จะเป็นข้อมูลแบบข้อความหรือไบนารี ก็ขึ้นกับการกำหนดค่าของคุณสมบัติ InputMode

Output คือ การส่งข้อมูลไปยังบัฟเฟอร์ด้านส่งออก ซึ่งสามารถกำหนดค่าได้เฉพาะในขณะที่ทำงานเท่านั้น รูปแบบการใช้งาน คือ

Object.Output [=value]

สำหรับชนิดของข้อมูลที่ถูกส่งโดยคุณสมบัติ Output จะเป็นข้อมูลแบบข้อความหรือไบนารี ก็ขึ้นกับการกำหนดค่าของคุณสมบัติ InputMode ดังที่กล่าวมาข้างต้น

6.3.6 iSevenSegmentClockX

00:00:00

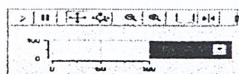
เป็นคอมโพเนนต์สำหรับการแสดงค่าเวลา เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

iSevenSegmentClock.Time = Now

เป็นคำสั่งสำหรับแสดงค่าเวลาในปัจจุบัน

6.3.7 iPlotX



เป็นคอมโพเนนต์สำหรับพล็อตกราฟ ซึ่งในคอมโพเนนต์นี้จะฟังก์ชันต่างๆ ให้ใช้เรียบร้อย เช่น Save จะบันทึกรูปภาพเป็นไฟล์ตามที่ผู้ใช้งานต้องการ สามารถขยายกราฟได้ สามารถเลื่อนค่าสเกลได้ เป็นต้น

รูปแบบคำสั่งของคอมโพเนนต์นี้คือ

iPlotX1.Channel (0).AddXY X, Y

โดยที่ Channel คือ ค่าของเส้นกราฟที่ต้องการแสดง

6.3.8 Microsoft Winsock Control 6.0



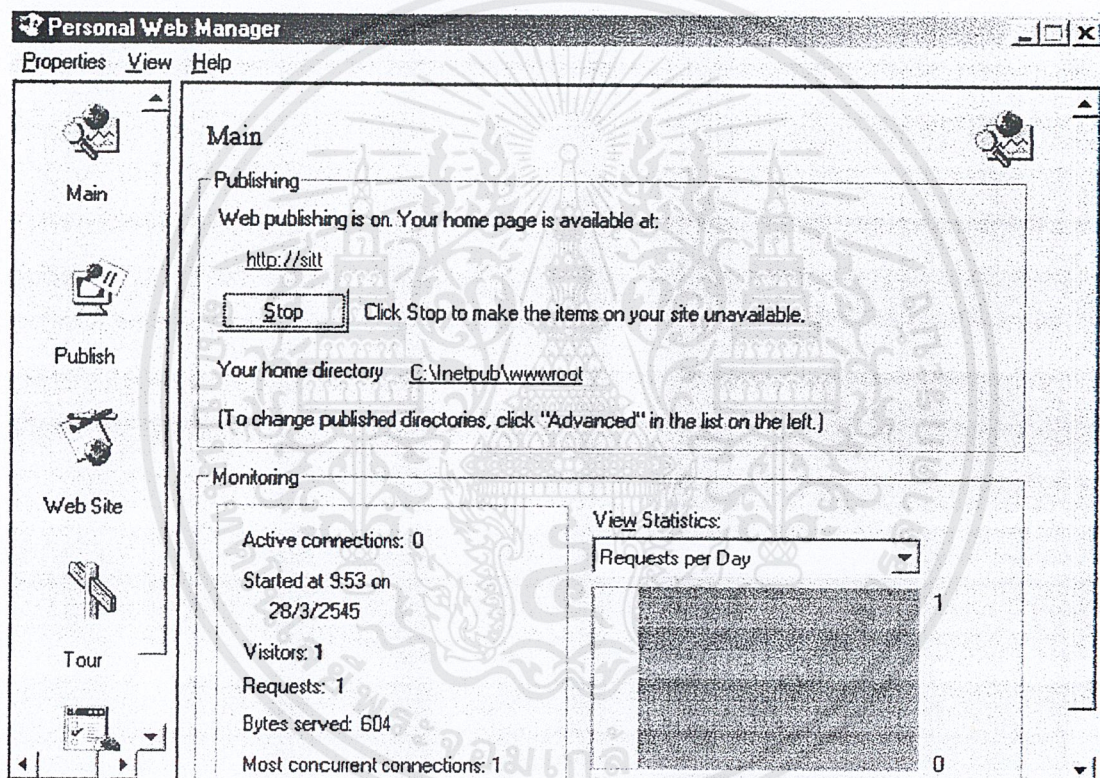
เนื่องจาก Microsoft Winsock Control 6.0 ไม่ได้เป็นคอนโทรลมาตรฐานที่อยู่ใน Toolbox ดังนั้นก่อนใช้งานจะต้องทำการเพิ่มคอนโทรลนี้ ซึ่งการเพิ่มคอนโทรลนี้ก็ทำเหมือนกับการเพิ่มคอนโทรล Microsoft Comm Control

บทที่ 7

การติดตั้งและใช้งานโปรแกรมด้านเซิร์ฟเวอร์

7.1 เว็บเซิร์ฟเวอร์

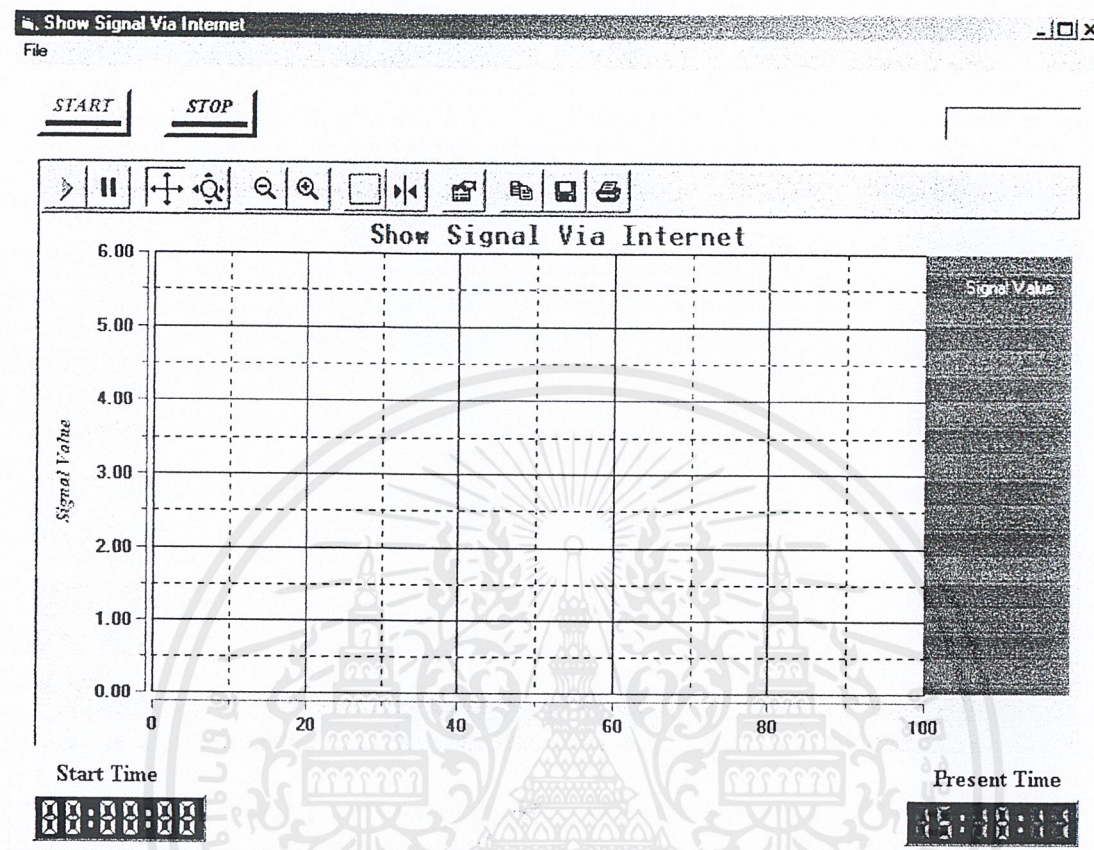
ก่อนที่จะติดตั้งและใช้งาน โปรแกรมด้านเซิร์ฟเวอร์จะต้องทำการติดตั้งโปรแกรมที่ทำหน้าที่เป็นเซิร์ฟเวอร์ก่อน ซึ่งในโครงงานนี้ใช้โปรแกรม PWS (Personal Web Server) ซึ่งเมื่อติดตั้งเรียบร้อยแล้วเครื่องคอมพิวเตอร์ของเราก็พร้อมที่จะทำหน้าที่เป็นเว็บเซิร์ฟเวอร์



ภาพที่ 7.1 เว็บเซิร์ฟเวอร์ที่ติดตั้งเรียบร้อยแล้ว

เมื่อติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์เรียบร้อยแล้ว โปรแกรมต่างๆ จะใช้เป็นโปรแกรมที่ทำงานที่เซิร์ฟเวอร์ก็นำไปติดตั้งที่ไดเรกทอรีที่โปรแกรมเว็บเซิร์ฟเวอร์ติดตั้งอยู่ เช่น ในปริญญานิพนธ์นี้โปรแกรมเว็บเซิร์ฟเวอร์ติดตั้งอยู่ที่ C:\inetpub\wwwroot ซึ่งเมื่อติดตั้งแล้วก็จะทำให้คอมพิวเตอร์อื่นที่ต่อกับระบบเน็ตเวิร์คสามารถเข้าถึงข้อมูลที่อยู่ในเว็บเซิร์ฟเวอร์ได้

7.2 โปรแกรมการทำงานที่เซิร์ฟเวอร์



ภาพที่ 7.2 แสดงโปรแกรมที่ทำงานฝั่งเซิร์ฟเวอร์

START เป็นปุ่มสั่งให้โปรแกรมทำงาน

STOP เป็นปุ่มสั่งให้โปรแกรมหยุดทำงาน

Start Time แสดงเวลาที่โปรแกรมเริ่มทำงาน

Present Time แสดงเวลาปัจจุบันที่โปรแกรมทำงานอยู่

ปุ่มคำสั่งที่กราฟประกอบด้วยปุ่มคำสั่ง ย่อ ขยาย ดูค่าที่ตำแหน่งใด บันทึก และพิมพ์

การทำงานของโปรแกรม

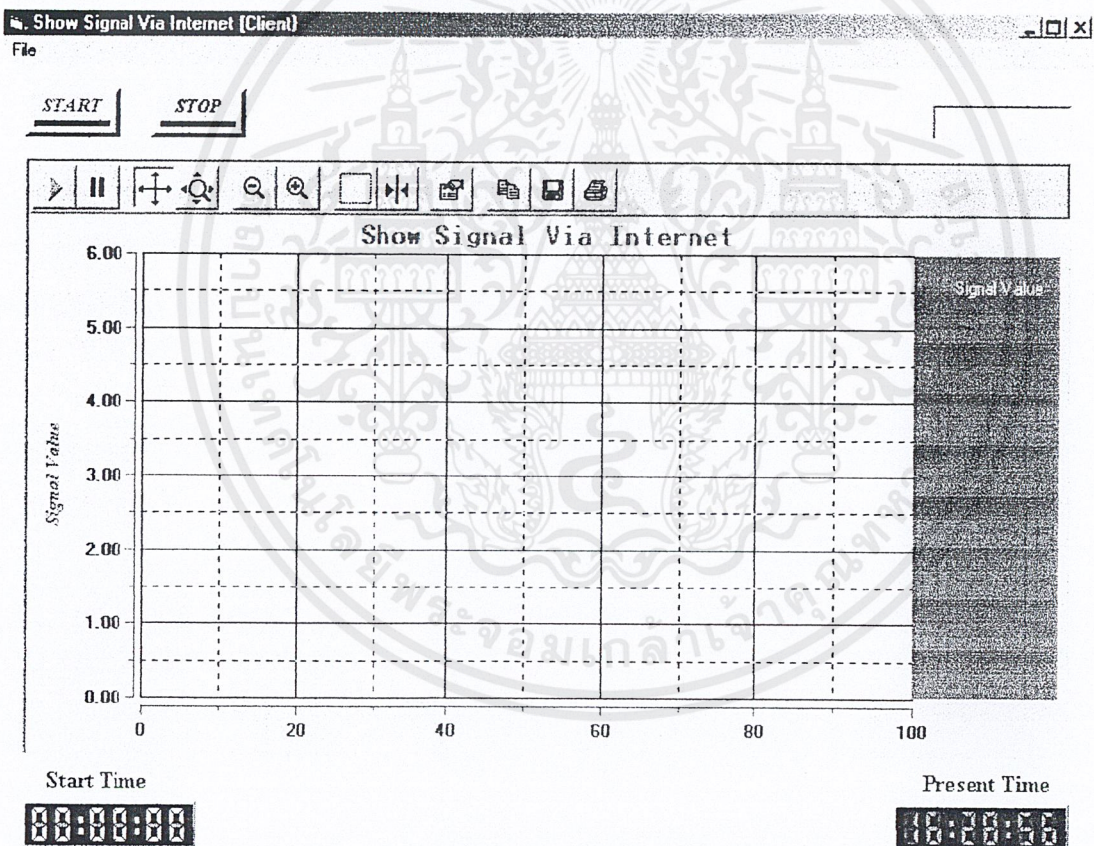
เมื่อกดปุ่ม START โปรแกรมก็จะเริ่มอ่านค่าจากชุดอ่านสัญญาณ เพื่อนำค่าที่ได้นั้น มาพล็อตกราฟ โดยที่โปรแกรมจะทำการพล็อตกราฟทุกๆ 1 วินาที

บทที่ 8

การติดตั้งและการใช้โปรแกรมด้านงานด้านไคลเอนท์

การใช้งานโปรแกรมด้านไคลเอนท์สามารถทำได้โดย 2 วิธี คือ

1. นำโปรแกรม Client.exe ไปติดตั้งที่เครื่องคอมพิวเตอร์ที่ต้องการจะเข้าไปดูค่าของกราฟที่เปลี่ยนแปลง
2. เข้าไปที่ www.kmitl.ac.th/~s2015410 เพื่อที่จะดาวน์โหลดโปรแกรม Client.exe แล้วสั่งให้โปรแกรมทำงานก็จะสามารถดูการเปลี่ยนแปลงของกราฟได้ หรือจะให้โปรแกรมทำงานเลยโดยไม่ต้องเซฟลงเครื่องก็ได้



ภาพที่ 8.1 โปรแกรมที่ทำงานที่ Client

การทำงานของโปรแกรมทางฝั่งไคลเอนท์ก็จะเหมือนกับทางฝั่งเซิร์ฟเวอร์ทุกอย่างประการ แต่ค่าที่ได้จะถูกหน่วยออกไปเล็กน้อย

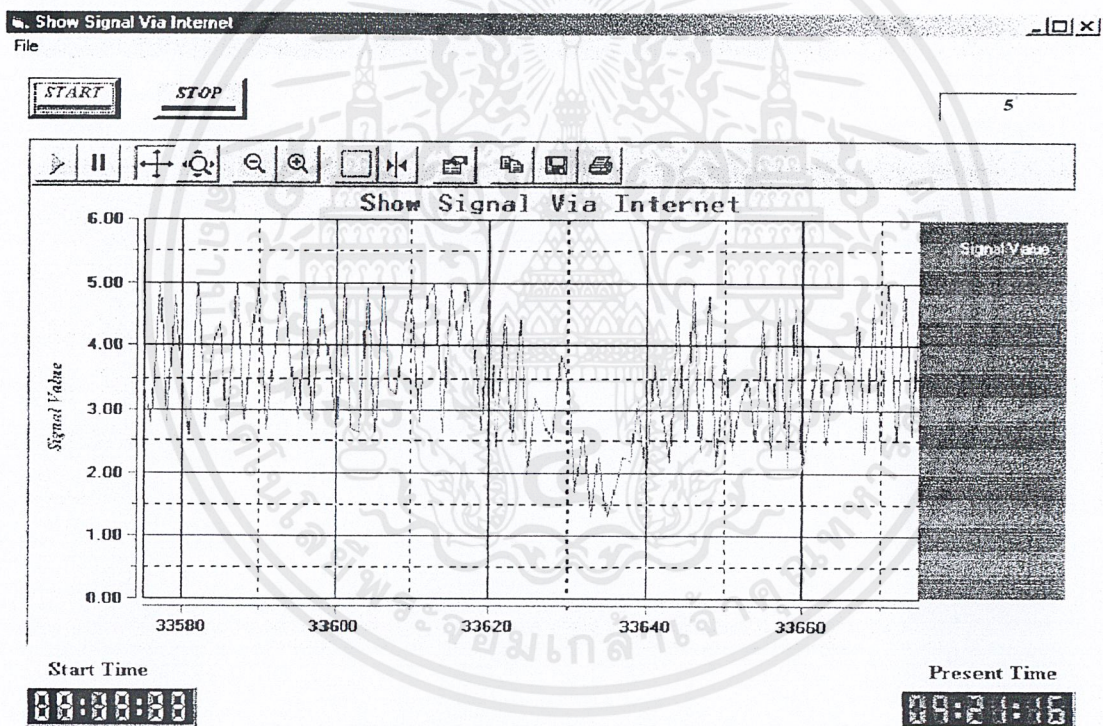
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

ผลการทดลอง

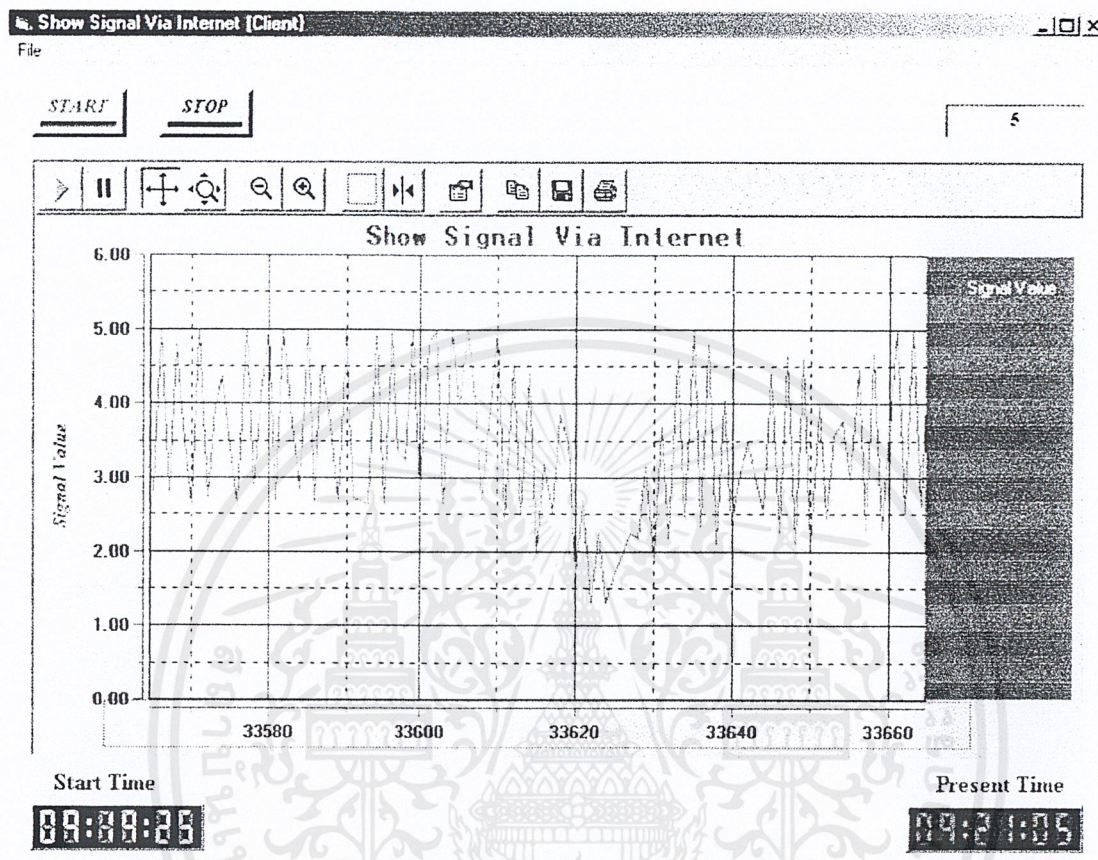
ผลการทดลองที่ได้ในโครงงานนี้ จะเป็นรูปคลื่นที่ได้จาก Signal Generator โดยสัญญาณของรูปคลื่นจะเป็นอินพุท จากนั้นชุดอ่านสัญญาณจะอ่านข้อมูลขนาด 8 บิต แล้วส่งไปยังคอมพิวเตอร์ เพื่อใช้โปรแกรมที่พัฒนาจากโปรแกรม Visual Basic นำค่าไปพล็อตกราฟ โดยผลของกราฟที่ได้ทางฝั่งไคลเอนต์จะต้องมีค่าเหมือนกับทางฝั่งเซิร์ฟเวอร์ทุกประการ

ผลของกราฟทางฝั่งเซิร์ฟเวอร์



ภาพที่ 9.1 ผลของรูปสัญญาณที่ได้ที่ฝั่งเซิร์ฟเวอร์

ผลของกราฟทางไคลเอนท์



ภาพที่ 9.2 ผลของรูปสัญญาณที่ได้ที่ฝั่งไคลเอนท์

จากผลการทดลองค่าที่ได้มีค่าเหมือนกัน แต่ช่วงเวลาที่อ่านได้ไม่เท่ากัน

บทที่ 10

สรุปและข้อเสนอแนะ

การแสดงผลสัญญาณไฟฟ้าผ่านทางเครือข่ายอินเทอร์เน็ตโดยอาศัยเบรเซอร์จะทำให้สามารถดูการเปลี่ยนแปลงของรูปคลื่นสัญญาณไฟฟ้าที่เราต้องการว่ามีการเปลี่ยนแปลงอย่างไร เพื่อที่ว่าจะนำค่าที่ได้นั้นไปใช้ประโยชน์ต่อไป

จากการทำงานในโครงงานนี้สามารถที่จะสรุปผลของการทำงานได้ดังนี้

1. รูปคลื่นสัญญาณที่ได้ไม่เป็นรูปคลื่นมาตรฐานเท่าที่ควร ทั้งนี้อาจจะเนื่องจากตัว Signal Generator เองที่มีค่าผิดพลาดมากหรือจากชุดอ่านสัญญาณ
2. เนื่องจากหลักการการทำงานของชุดอ่านสัญญาณเป็นการสุ่มค่าสัญญาณเข้ามาเป็นช่วงๆ แล้วนำค่าที่ได้มาพล็อตเป็นกราฟ และเมื่อมีการเปลี่ยนแปลงแรงดันอย่างรวดเร็วก็จะทำให้สัญญาณเพี้ยนไป
3. เมื่อความถี่มีค่าสูงขึ้น ค่าสัญญาณที่ชุดอ่านสัญญาณอ่านได้จะมีความผิดเพี้ยนมากขึ้นด้วย

เนื่องจากลักษณะการทำงานของโครงงานนี้ยังเป็นเพียงแค่ระดับพื้นฐานเท่านั้น ดังนั้นการทำงานของโครงงานนี้จึงไม่ยุ่งยากมากนัก ฉะนั้นในการที่จะนำไปใช้งานจริงก็จะต้องมีการพัฒนาในส่วนของโปรแกรมและชุดอ่านสัญญาณให้มีประสิทธิภาพมากยิ่งขึ้น และอีกประการหนึ่งที่สำคัญสำหรับการสื่อสารบนอินเทอร์เน็ตคือ ต้องคำนึงถึงความปลอดภัยของข้อมูลและระบบที่ให้บริการและควบคุมผ่านอินเทอร์เน็ตว่าสามารถป้องกันผู้ที่อาจเข้ามาก่อวินาศกรรมหรือทำลายระบบได้ ดังนั้นโปรแกรมจึงต้องมีการพัฒนา โดยคำนึงถึงความปลอดภัยของข้อมูลและระบบเครือข่ายที่ให้บริการมากขึ้น

บรรณานุกรม

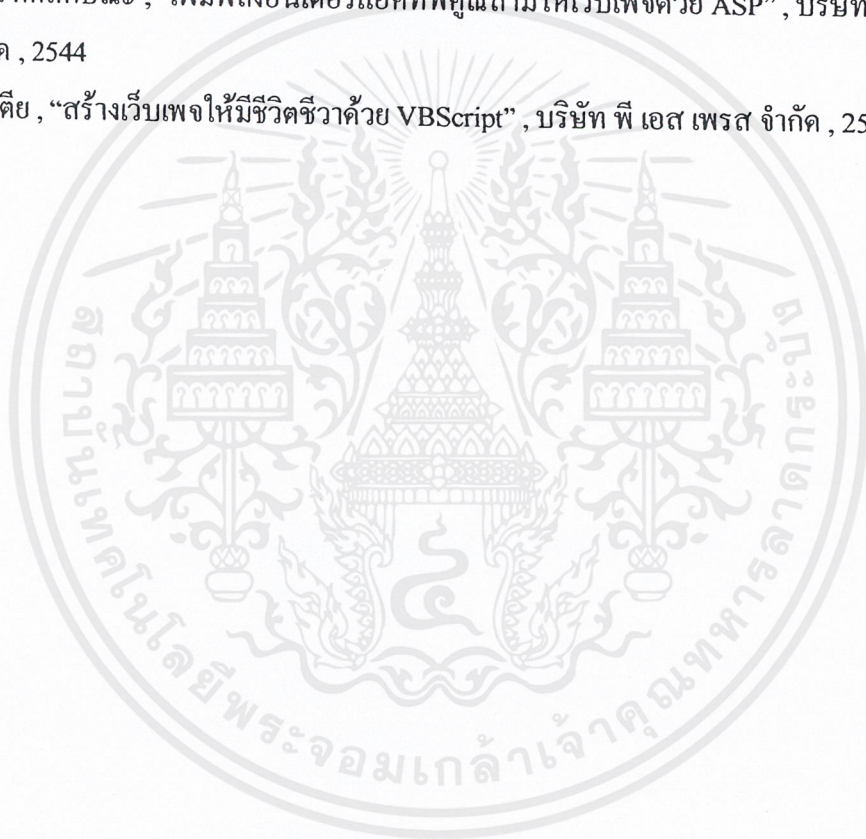
กิตติ ภัคดีวัฒนกุล , จำลอง ทรูอดุทธาหะ, “Visual Basic 6 ฉบับโปรแกรมเมอร์” , หจก.ไทยเจริญ
การพิมพ์ , 2543

ธาริน สิทธิธรรมชารี , “คู่มือการเขียน โปรแกรม Microsoft Visual Version 6.0” , บริษัท ชัคเซส
มีเดีย จำกัด

ไพศาล โมลิตกุลมงคล , “พัฒนา Web Database ด้วย ASP” , หจก.ไทยเจริญการพิมพ์

สุรัตน์ บัณฑิตลักษณ์ะ , “เพิ่มพลังอินเทอร์เน็ตแอปพลิเคชันตามให้เว็บเพจด้วย ASP” , บริษัท วิตตี้ กรุ๊ป
จำกัด , 2544

วันชัย แซ่เตีย , “สร้างเว็บเพจให้มีชีวิตชีวาด้วย VBScript” , บริษัท พี เอส เพรส จำกัด , 2544





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

89C51/89C52/89C54/89C58
80C51 8-bit microcontroller family
4K/8K/16K/32K Flash

Product data

Supersedes data of 1999 Oct 27
IC28 Data Handbook

2002 Jan 15

Philips
Semiconductors



PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

80C51 8-bit microcontroller family

4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

DESCRIPTION

The 89C51/89C52/89C54/89C58 contain a non-volatile FLASH program memory that is parallel programmable. For devices that are serial programmable (In-System Programmable (ISP) and In-Application Programmable (IAP) with a boot loader), see the 89C51Rx2 or 89C66x datasheets.

All three families are Single-Chip 8-bit Microcontrollers manufactured in advanced CMOS process and are derivatives of the 80C51 microcontroller family. All the devices have the same instruction set as the 80C51.

SELECTION TABLE FOR FLASH DEVICES

	MTP devices (this data sheet)		ISP/IAP devices (see separate data sheets)	
	89C51	89C52/54/58	89C51Rx2	89C66x
ROM/EPROM memory size	4K	8K/16K/32K	16K–64K	16K–64K
RAM size (byte)	128	256	512–1K	512–8K
Parallel programming	yes	yes	yes	yes
In-System Programming (ISP)	no	no	yes	yes
In-Application Programming (IAP)	no	no	yes	yes
PWM	no	no	yes	yes
Programmable Timer/Counter (PCA)	no	no	yes	yes
Hardware Watchdog Timer	no	no	yes	yes
Serial Channels	UART	UART	yes	yes
			UART	UART + I ² C

MTP = Multi-Time Programming (via parallel programmer)

ISP = In-System Programming (via serial interface)

IAP = In-Application Programming

Please note that the FLASH programming algorithm for these parts has been modified. Please see the Device Comparison table for details.

DEVICE COMPARISON TABLE

Item	Old devices	New devices	Reason for change
Type description	P89C5xUBxx / P89C5xUFxx	P89C5xBx	Letter U dropped for shorter type descriptions (formerly designated speed (0–33 MHz))
Programming algorithm	When using parallel programmer, be sure to select P89C5xUxxx devices	When using a parallel programmer, be sure to select P89C5xBx devices (no more letter U). IF DEVICES ARE NOT YET SELECTABLE, ASK YOUR VENDOR FOR A SOFTWARE UPDATE.	Programming algorithm modification required by process change!
Quad Flat Package type	PQFP package (P89C5xUxBB)	PQFP package replaced by LQFP package (P89C5xBBD). SEE NEW DIMENSIONS AT THE END OF THIS DATA SHEET.	Reduction in package height
Package identifiers	PLCC = AA PQFP = BB PDIP = PN	PLCC = A LQFP = BD PDIP = P	Shorter type descriptions
Flash memory program and erase cycles	100 program and erase cycles	10,000 program and erase cycles	Process change allows more program and erase cycles
Power consumption	Active mode: $I_{CC(MAX)} = (0.9 \times \text{FREQ.} + 20)\text{mA}$ Idle mode: $I_{CC(MAX)} = (0.37 \times \text{FREQ.} + 1.0)\text{mA}$	Active mode: $I_{CC(MAX)} = (0.55 \times \text{FREQ.} + 8.0)\text{mA}$ Idle mode: $I_{CC(MAX)} = (0.3 \times \text{FREQ.} + 2.0)\text{mA}$	Process change allows lower power consumption

80C51 8-bit microcontroller family

4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

FEATURES

- 80C51 Central Processing Unit
 - On-chip FLASH Program Memory
 - Speed up to 33 MHz
 - Fully static operation
 - RAM expandable externally up to 64 kbytes
 - 4 interrupt priority levels
 - 6 interrupt sources
 - Four 8-bit I/O ports
 - Full-duplex enhanced UART
 - Framing error detection
 - Automatic address recognition
- Three 16-bit timers/counters T0, T1 (standard 80C51) and additional T2 (capture and compare)
 - Power control modes
 - Clock can be stopped and resumed
 - Idle mode
 - Power down mode
 - Programmable clock out
 - Second DPTR register
 - Asynchronous port reset
 - Low EMI (inhibit ALE)
 - Wake up from power down by an external interrupt

ORDERING INFORMATION

Type number				Package			Temperature Range ³ (°C)	Voltage Range (V)	Frequency (MHz)
4K Flash version	8K Flash version	16K Flash version	32K Flash version	Name	Description	Version			
P89C51BA	P89C52BA	P89C54BA	P89C58BA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2	0 to +70	5	0 to 33
P89C51BP ¹	P89C52BP ¹	P89C54BP ¹	P89C58BP ¹	DIP40	plastic dual in-line package; 40 leads	SOT129-1	0 to +70	5	0 to 33
P89C51BN ²	P89C52BN ²	P89C54BN ²	P89C58BN ²	LQFP44	plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm	SOT389-1	0 to +70	5	0 to 33

- NOTES:
1. Philips (except North America) Part Order Number

2. Philips North America Part Order Number. Note that parts will be marked "P89C5xBP", respectively (x = 1, 2, 4, 8)

3. Industrial temperatures will be released with P89C5xX2 devices (see separate data sheet)

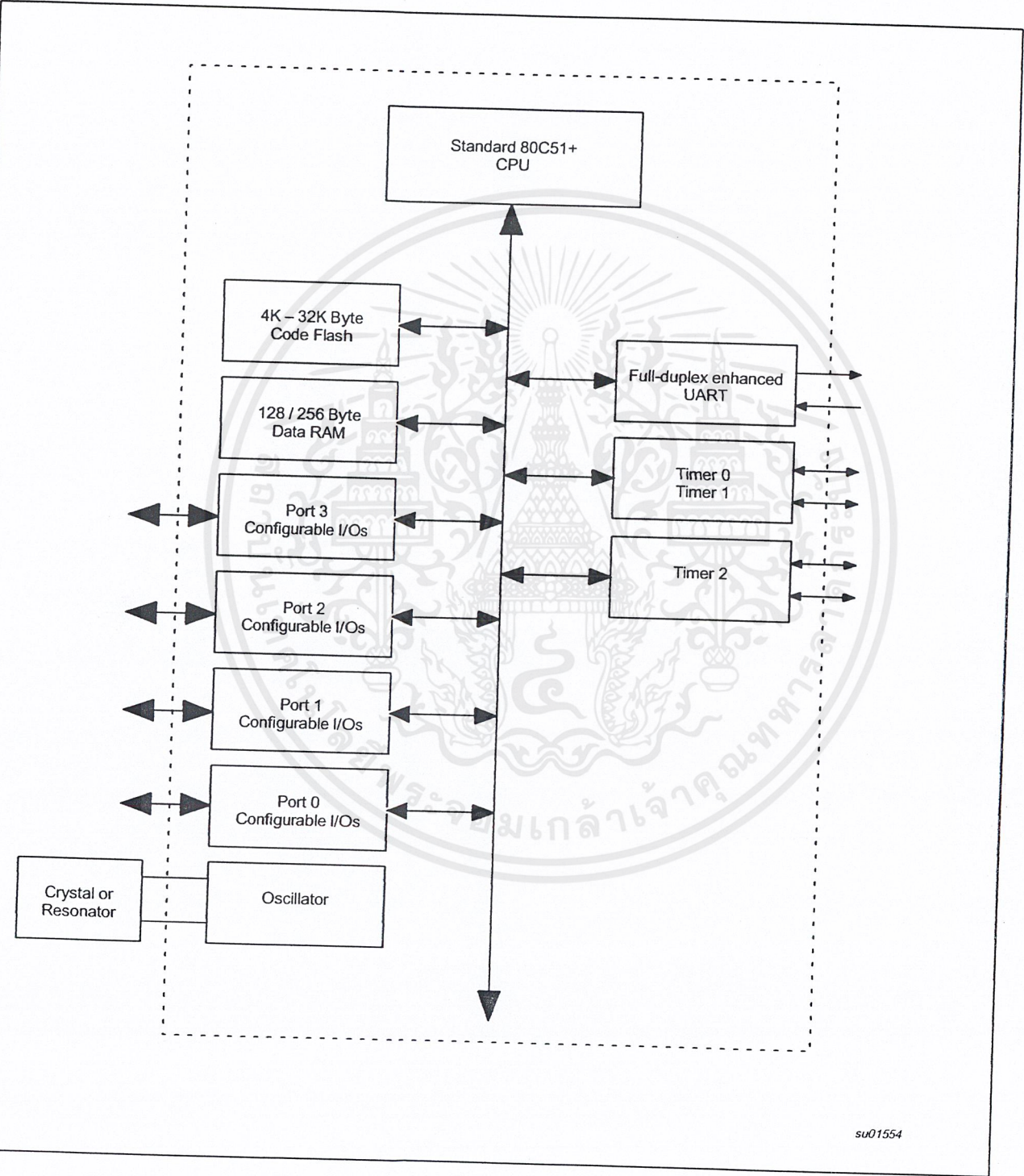
PART NUMBER DERIVATION

Device number (P89C5x)	Temperature range	Package
P89C51	B = 0 °C to 70 °C	BD = LQFP
P89C52		A = PLCC
P89C54		P = PDIP
P89C58		

80C51 8-bit microcontroller family
4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

BLOCK DIAGRAM 1



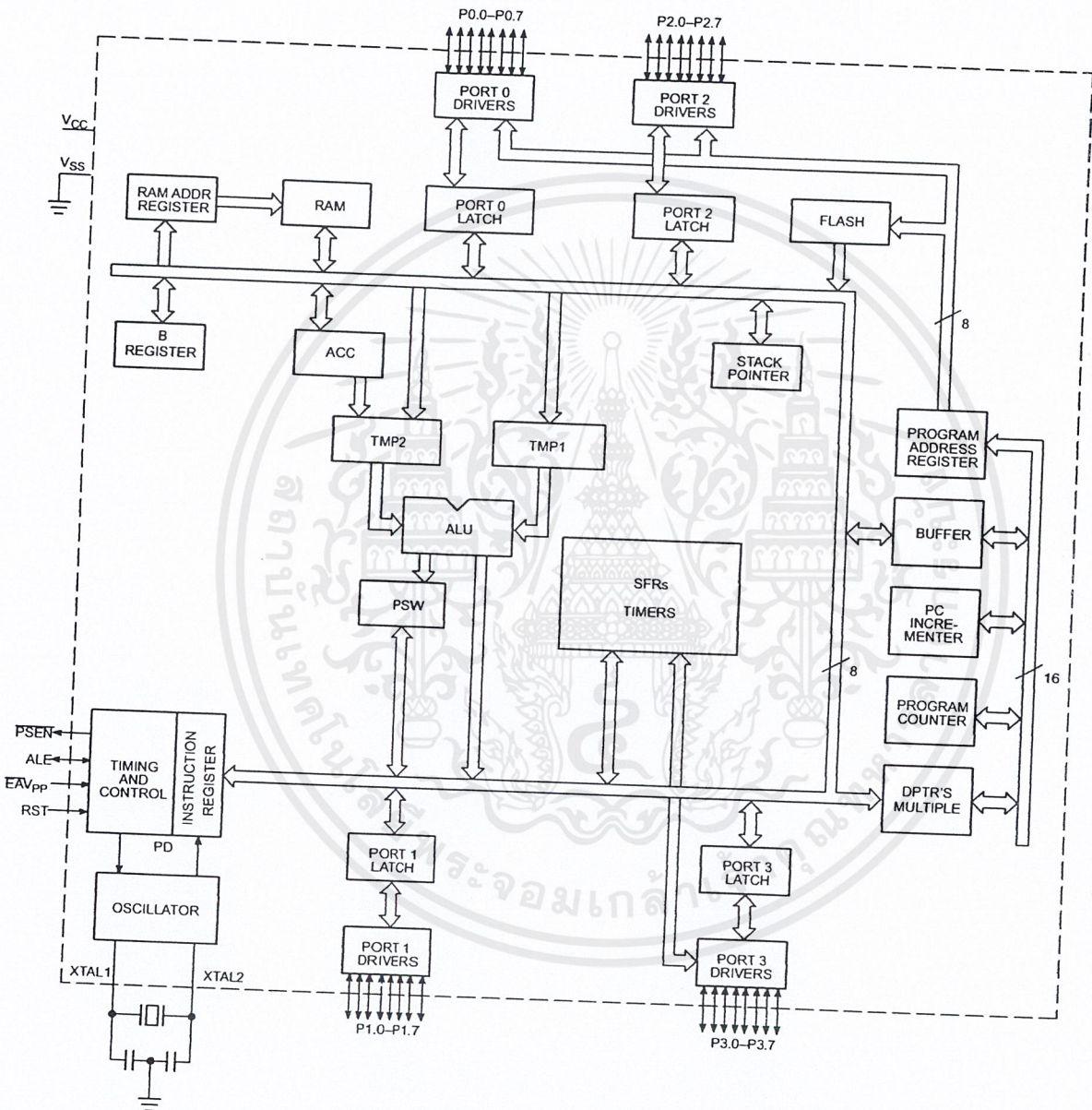
su01554

80C51 8-bit microcontroller family

4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

BLOCK DIAGRAM 2 (CPU ORIENTED)

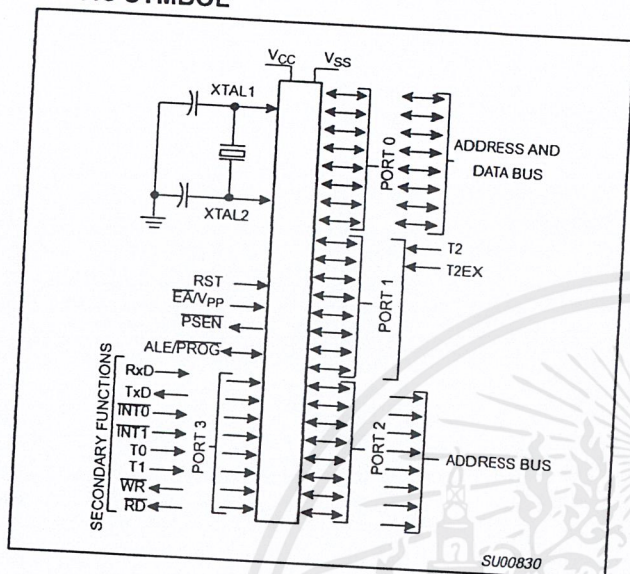


SU01066

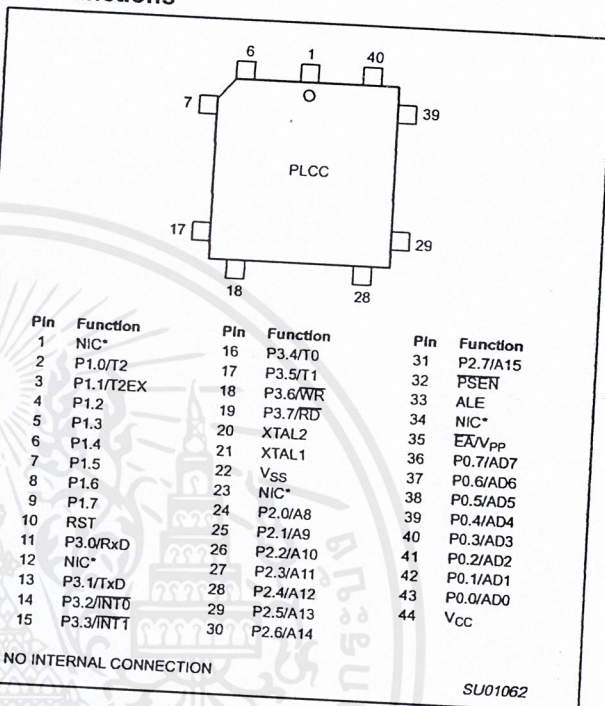
80C51 8-bit microcontroller family 4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

LOGIC SYMBOL

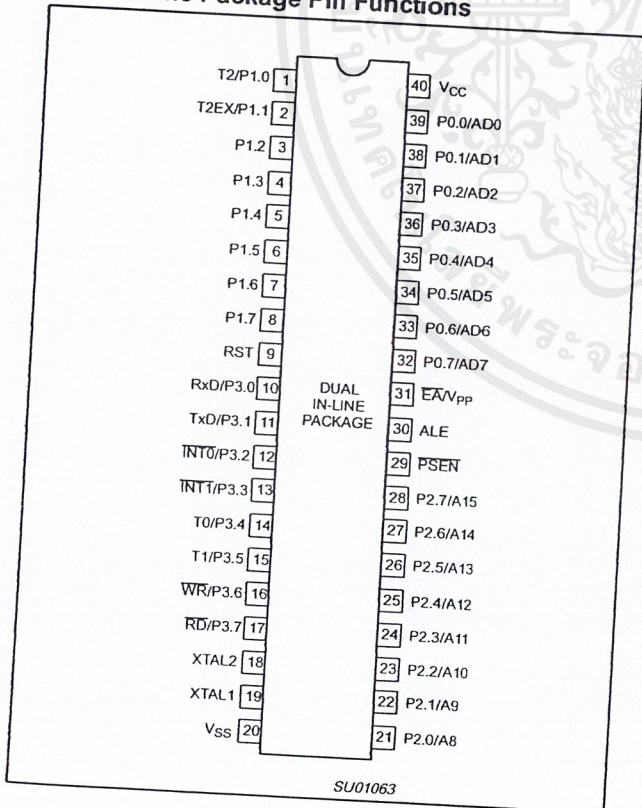


Ceramic and Plastic Leaded Chip Carrier Pin Functions

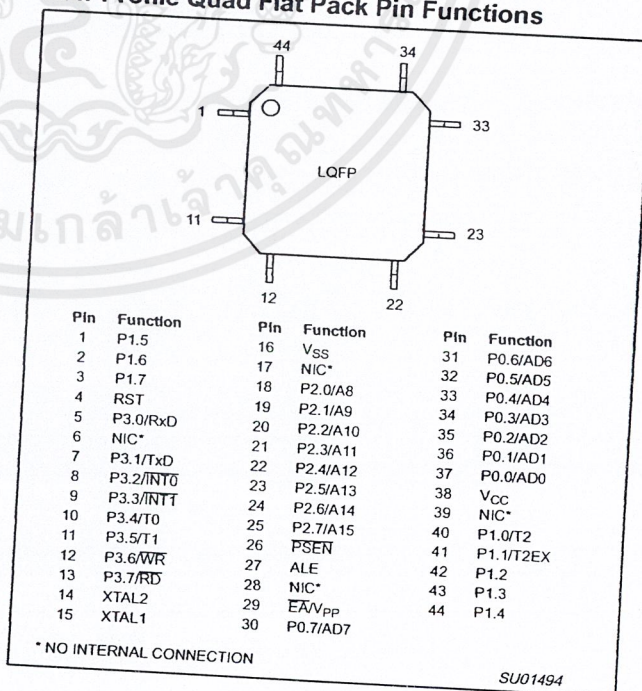


PIN CONFIGURATIONS

Dual In-Line Package Pin Functions



Low Profile Quad Flat Pack Pin Functions



80C51 8-bit microcontroller family

4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V _{SS}	20	22	16	I	Ground: 0 V reference.
V _{CC}	40	44	38	I	Power Supply: This is the power supply voltage for normal, idle, and power-down operation.
P0.0–P0.7	39–32	43–36	37–30	I/O	Port 0: Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I _{IL}). Alternate function for Port 1: T2 (P1.0): Timer/Counter2 external count input/clockout (see Programmable Clock-Out). T2EX (P1.1): Timer/Counter2 reload/capture/direction control.
P2.0–P2.7	1 2 21–28	2 3 24–31	40 41 18–25	I/O I I/O	Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I _{IL}). 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 pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I _{IL}). Port 3 also serves the special features of the 89C51/89C52/89C54/89C58, as listed below: RxD (P3.0): Serial input port TxD (P3.1): Serial output port INT0 (P3.2): External interrupt INT1 (P3.3): External interrupt T0 (P3.4): Timer 0 external input T1 (P3.5): Timer 1 external input WR (P3.6): External data memory write strobe RD (P3.7): External data memory read strobe
RST	9	10	4	I	Reset: A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V _{SS} permits a power-on reset using only an external capacitor to V _{CC} .
ALE	30	33	27	O	Address Latch Enable: Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. ALE can be disabled by setting SFR auxiliary.0. With this bit set, ALE will be active only during a MOVX instruction.
PSEN	29	32	26	O	Program Store Enable: The read strobe to external program memory. When executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V _{PP}	31	35	29	I	External Access Enable/Programming Supply Voltage: EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to the maximum internal memory boundary. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 0FFFH for 4 k devices, 1FFFH for 8 k devices, 3FFFH for 16 k devices, and 7FFFH for 32 k devices. The value on the EA pin is latched when RST is released and any subsequent changes have no effect. This pin also receives the 5V/12V (±10%) programming supply voltage (V _{PP}) during FLASH programming.
XTAL1	19	21	15	I	Crystal 1: Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	Crystal 2: Output from the inverting oscillator amplifier.

NOTE: To avoid "latch-up" effect at power-on, the voltage on any pin (other than V_{PP}) at any time must not be higher than V_{CC} + 0.5 V or V_{SS} – 0.5 V, respectively.

80C51 8-bit microcontroller family

4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

Table 1. 89C51/89C52/89C54/89C58 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	—	—	—	—	—	—	—	AO	xxxxxxx0B
AUXR1#	Auxiliary 1	A2H	—	—	—	—	GF2	0	—	DPS	xxxx00x0B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
IE*	Interrupt Enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	
			EA	—	ET2	ES	ET1	EX1	ET0	EX0	0x000000B
IP*	Interrupt Priority	B8H	BF	BE	BD	BC	BB	BA	B9	B8	
			—	—	PT2	PS	PT1	PX1	PT0	PX0	xx000000B
IPH#	Interrupt Priority High	B7H	B7	B6	B5	B4	B3	B2	B1	B0	
			—	—	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	xx000000B
P0*	Port 0	80H	87	86	85	84	83	82	81	80	
			AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
P1*	Port 1	90H	97	96	95	94	93	92	91	90	
			—	—	—	—	—	—	T2EX	T2	FFH
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	
			AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	
			RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON#1	Power Control	87H	SMOD1	SMOD0	—	POF ²	GF1	GF0	PD	IDL	00xxx000B
PSW*	Program Status Word	D0H	D7	D6	D5	D4	D3	D2	D1	D0	
RACAP2H#	Timer 2 Capture High	CBH	CY	AC	F0	RS1	RS0	OV	—	P	000000x0B
RACAP2L#	Timer 2 Capture Low	CAH									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									00H
			9F	9E	9D	9C	9B	9A	99	98	xxxxxxxxB
SCON*	Serial Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack Pointer	81H									07H
TCON*	Timer Control	88H	8F	8E	8D	8C	8B	8A	89	88	
			TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
T2CON*	Timer 2 Control	C8H	CF	CE	CD	CC	CB	CA	C9	C8	
T2MOD#	Timer 2 Mode Control	C9H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
TH0	Timer High 0	8CH	—	—	—	—	—	—	T2OE	DCEN	xxxxxx00B
TH1	Timer High 1	8DH									00H
TH2#	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2#	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

* SFRs are bit addressable.

SFRs are modified from or added to the 80C51 SFRs.

— Reserved bits.

1. Reset value depends on reset source.

2. Bit will not be affected by reset.

80C51 8-bit microcontroller family 4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

FLASH EPROM MEMORY

General Description

The 89C51/89C52/89C54/89C58 FLASH reliably stores memory contents even after 10,000 erase and program cycles. The cell is designed to optimize the erase and programming mechanisms. In addition, the combination of advanced tunnel oxide processing and low internal electric fields for erase and programming operations produces reliable cycling.

Features

- FLASH EPROM internal program memory with Chip Erase
- Up to 64 k byte external program memory if the internal program memory is disabled ($\overline{EA} = 0$)
- Programmable security bits
- 10,000 minimum erase/program cycles for each byte
- 10 year minimum data retention
- Programming support available from many popular vendors

OSCILLATOR CHARACTERISTICS

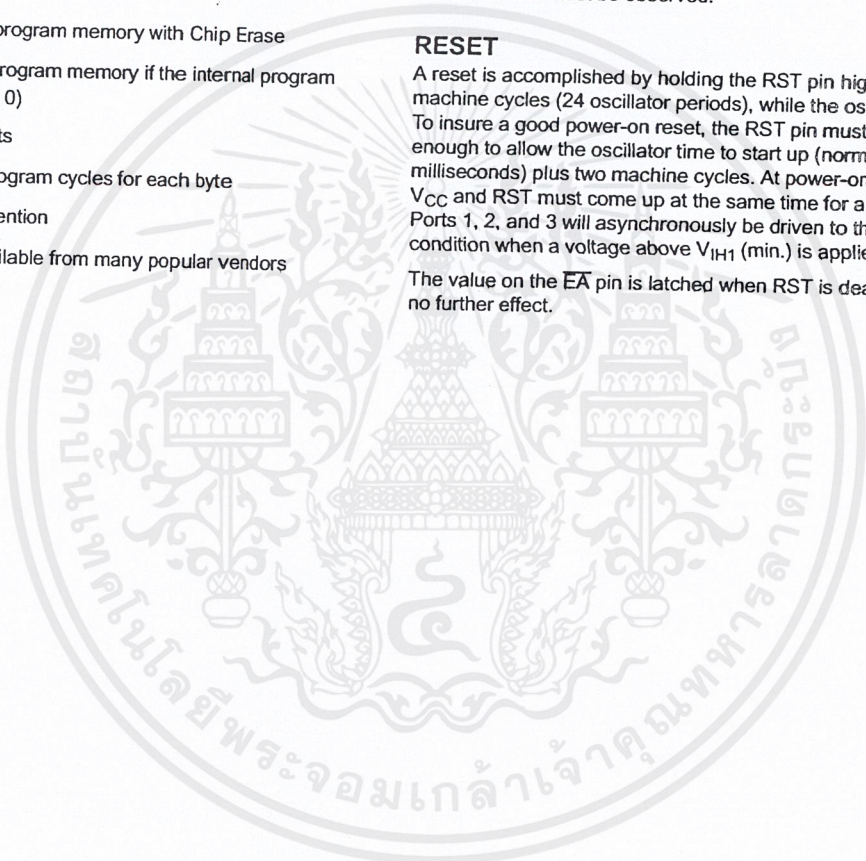
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up. Ports 1, 2, and 3 will asynchronously be driven to their reset condition when a voltage above V_{IH1} (min.) is applied to RST.

The value on the $\overline{EA}$ pin is latched when RST is deasserted and has no further effect.



ADC0801/ADC0802/ADC0803/ADC0804/ADC0805

8-Bit μ P Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Features

- Compatible with 8080 μ P derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"

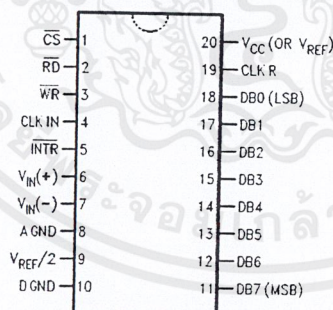
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC} , 2.5 V_{DC} , or analog span adjusted voltage reference

Key Specifications

- Resolution 8 bits
- Total error $\pm 1/4$ LSB, $\pm 1/2$ LSB and ± 1 LSB
- Conversion time 100 μ s

Connection Diagram

ADC080X
Dual-In-Line and Small Outline (SO) Packages



DS005671-30

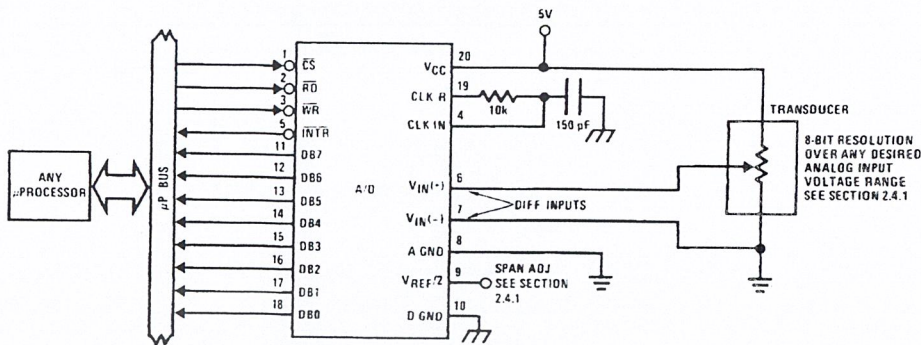
See Ordering Information

Ordering Information

TEMP RANGE		0°C TO 70°C	0°C TO 70°C	-40°C TO +85°C
ERROR	±¼ Bit Adjusted	ADC0802LCWM	ADC0804LCN	ADC0801LCN
	±½ Bit Unadjusted			ADC0802LCN
	±½ Bit Adjusted	ADC0804LCWM		ADC0803LCN
	±1Bit Unadjusted			ADC0805LCN/ADC0804LCJ
PACKAGE OUTLINE		M20B — Small Outline	N20A — Molded DIP	

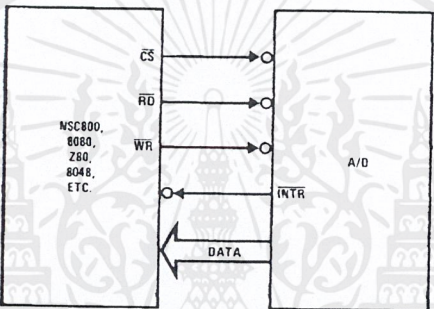
Z-80 is a registered trademark of Zilog Corp.

Typical Applications



DS005671-1

8080 Interface



DS005671-31

Error Specification (Includes Full-Scale, Zero Error, and Non-Linearity)			
Part Number	Full-Scale Adjusted	$V_{REF}/2=2.500 V_{DC}$ (No Adjustments)	$V_{REF}/2=$ No Connection (No Adjustments)
ADC0801	$\pm 1/4$ LSB		
ADC0802		$\pm 1/2$ LSB	
ADC0803	$\pm 1/2$ LSB		
ADC0804		± 1 LSB	
ADC0805			± 1 LSB

Absolute Maximum Ratings (Notes 1, 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage	
Logic Control Inputs	-0.3V to +18V
At Other Input and Outputs	-0.3V to ($V_{CC}+0.3V$)
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C

Infrared (15 seconds)	220°C
Storage Temperature Range	-65°C to +150°C
Package Dissipation at $T_A=25^\circ\text{C}$	875 mW
ESD Susceptibility (Note 10)	800V

Operating Ratings (Notes 1, 2)

Temperature Range	$T_{MIN} \leq T_A \leq T_{MAX}$
ADC0804LCJ	-40°C $\leq T_A \leq$ +85°C
ADC0801/02/03/05LCN	-40°C $\leq T_A \leq$ +85°C
ADC0804LCN	0°C $\leq T_A \leq$ +70°C
ADC0802/04LCWM	0°C $\leq T_A \leq$ +70°C
Range of V_{CC}	4.5 V_{DC} to 6.3 V_{DC}

Electrical Characteristics

The following specifications apply for $V_{CC}=5 V_{DC}$, $T_{MIN} \leq T_A \leq T_{MAX}$ and $f_{CLK}=640$ kHz unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
ADC0801: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			$\pm 1/4$	LSB
ADC0802: Total Unadjusted Error (Note 8)	$V_{REF}/2=2.500 V_{DC}$			$\pm 1/2$	LSB
ADC0803: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			$\pm 1/2$	LSB
ADC0804: Total Unadjusted Error (Note 8)	$V_{REF}/2=2.500 V_{DC}$			± 1	LSB
ADC0805: Total Unadjusted Error (Note 8)	$V_{REF}/2$ -No Connection			± 1	LSB
$V_{REF}/2$ Input Resistance (Pin 9)	ADC0801/02/03/05	2.5	8.0		k Ω
	ADC0804 (Note 9)	0.75	1.1		k Ω
Analog Input Voltage Range	(Note 4) $V(+)$ or $V(-)$	Gnd-0.05		$V_{CC}+0.05$	V_{DC}
DC Common-Mode Error	Over Analog Input Voltage Range		$\pm 1/16$	$\pm 1/8$	LSB
Power Supply Sensitivity	$V_{CC}=5 V_{DC} \pm 10\%$ Over Allowed $V_{IN}(+)$ and $V_{IN}(-)$ Voltage Range (Note 4)		$\pm 1/16$	$\pm 1/8$	LSB

AC Electrical Characteristics

The following specifications apply for $V_{CC}=5 V_{DC}$ and $T_{MIN} \leq T_A \leq T_{MAX}$ unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T_C	Conversion Time	$f_{CLK}=640$ kHz (Note 6)	103		114	μs
T_C	Conversion Time	(Notes 5, 6)	66		73	$1/f_{CLK}$
f_{CLK}	Clock Frequency	$V_{CC}=5V$, (Note 5)	100	640	1460	kHz
	Clock Duty Cycle		40		60	%
CR	Conversion Rate in Free-Running Mode	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}=0 V_{DC}$, $f_{CLK}=640$ kHz	8770		9708	conv/s
$t_{W(\overline{WR})L}$	Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS}=0 V_{DC}$ (Note 7)	100			ns
t_{ACC}	Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	$C_L=100$ pF		135	200	ns
t_{TH} , t_{OH}	TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	$C_L=10$ pF, $R_L=10k$ (See TRI-STATE Test Circuits)		125	200	ns
t_{WI} , t_{RI}	Delay from Falling Edge of $\overline{WR}$ or $\overline{RD}$ to Reset of $\overline{INTR}$			300	450	ns
C_{IN}	Input Capacitance of Logic Control Inputs			5	7.5	pF

AC Electrical Characteristics (Continued)

The following specifications apply for $V_{CC}=5 V_{DC}$ and $T_{MIN} \leq T_A \leq T_{MAX}$ unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{OUT}	TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF

CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]

$V_{IN}(1)$	Logical "1" Input Voltage (Except Pin 4 CLK IN)	$V_{CC}=5.25 V_{DC}$	2.0		15	V_{DC}
$V_{IN}(0)$	Logical "0" Input Voltage (Except Pin 4 CLK IN)	$V_{CC}=4.75 V_{DC}$			0.8	V_{DC}
$I_{IN}(1)$	Logical "1" Input Current (All Inputs)	$V_{IN}=5 V_{DC}$		0.005	1	μA_{DC}
$I_{IN}(0)$	Logical "0" Input Current (All Inputs)	$V_{IN}=0 V_{DC}$	-1	-0.005		μA_{DC}

CLOCK IN AND CLOCK R

V_{T+}	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V_{DC}
V_{T-}	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	V_{DC}
V_H	CLK IN (Pin 4) Hysteresis ($V_{T+} - V_{T-}$)		0.6	1.3	2.0	V_{DC}
$V_{OUT}(0)$	Logical "0" CLK R Output Voltage	$I_O=360 \mu A$ $V_{CC}=4.75 V_{DC}$			0.4	V_{DC}
$V_{OUT}(1)$	Logical "1" CLK R Output Voltage	$I_O=-360 \mu A$ $V_{CC}=4.75 V_{DC}$	2.4			V_{DC}

DATA OUTPUTS AND INTR

$V_{OUT}(0)$	Logical "0" Output Voltage Data Outputs INTR Output	$I_{OUT}=1.6 mA, V_{CC}=4.75 V_{DC}$ $I_{OUT}=1.0 mA, V_{CC}=4.75 V_{DC}$			0.4 0.4	V_{DC} V_{DC}
$V_{OUT}(1)$	Logical "1" Output Voltage	$I_O=-360 \mu A, V_{CC}=4.75 V_{DC}$	2.4			V_{DC}
$V_{OUT}(1)$	Logical "1" Output Voltage	$I_O=-10 \mu A, V_{CC}=4.75 V_{DC}$	4.5			V_{DC}
I_{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	$V_{OUT}=0 V_{DC}$ $V_{OUT}=5 V_{DC}$	-3			μA_{DC} μA_{DC}
I_{SOURCE}		V_{OUT} Short to Gnd, $T_A=25^\circ C$	4.5	6		mA_{DC}
I_{SINK}		V_{OUT} Short to V_{CC} , $T_A=25^\circ C$	9.0	16		mA_{DC}

POWER SUPPLY

I_{CC}	Supply Current (Includes Ladder Current)	$f_{CLK}=640 kHz$, $V_{REF}/2=NC$, $T_A=25^\circ C$ and $\overline{CS}=5V$				
	ADC0801/02/03/04LCJ/05			1.1	1.8	mA
	ADC0804LCN/LCWM			1.9	2.5	mA

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the D Gnd.

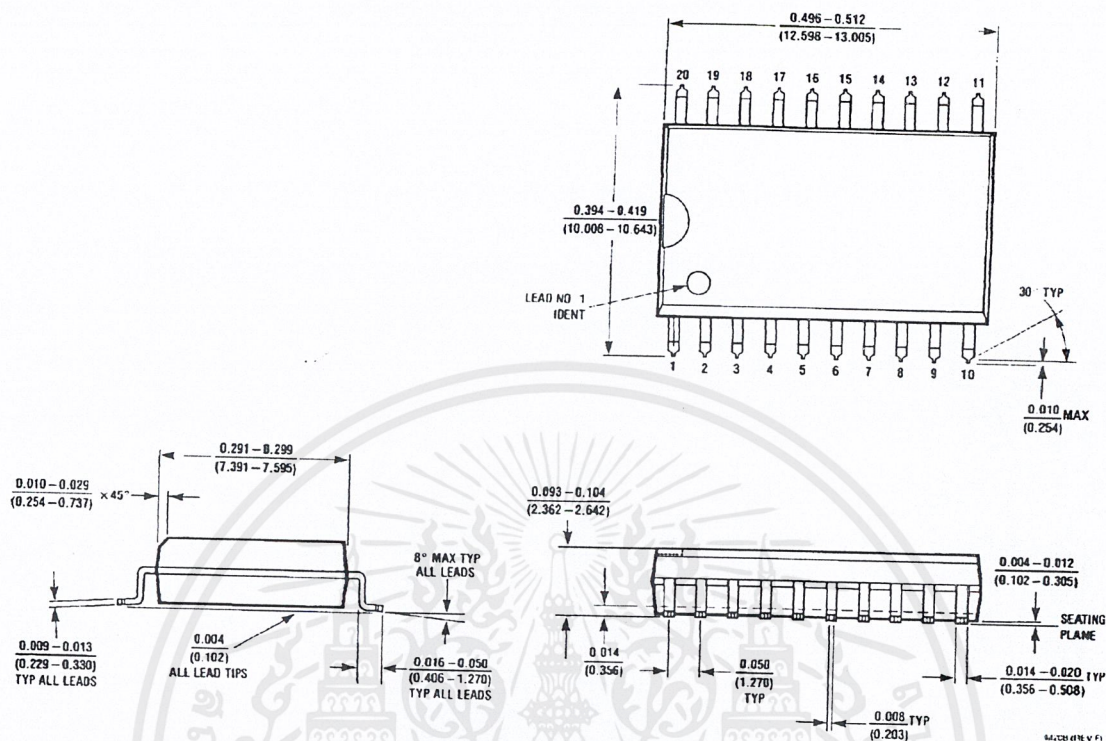
Note 3: A zener diode exists, internally, from V_{CC} to Gnd and has a typical breakdown voltage of $7 V_{DC}$.

Note 4: For $V_{IN}(-) \geq V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of $4.950 V_{DC}$ over temperature variations, initial tolerance and loading.

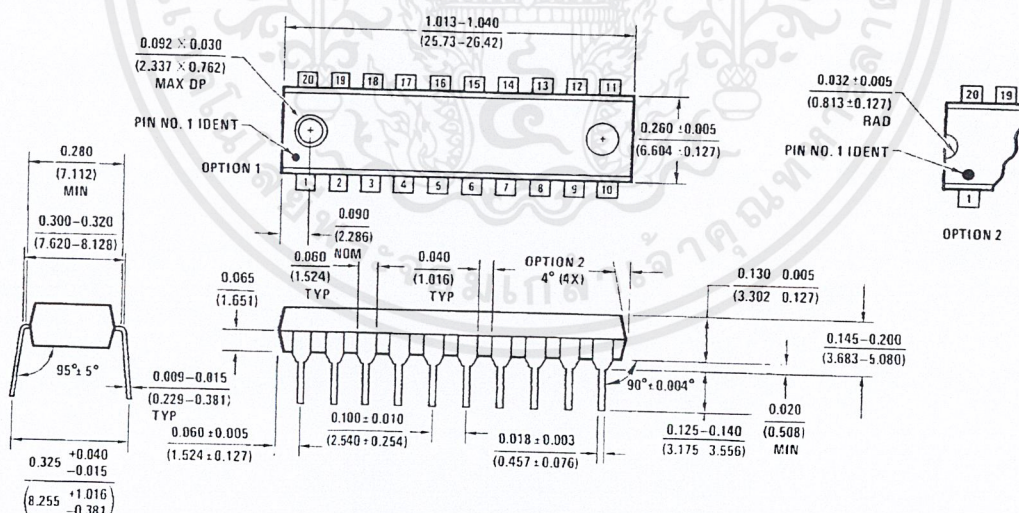
Note 5: Accuracy is guaranteed at $f_{CLK} = 640 kHz$. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

Note 6: With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The start request is internally latched, see Figure 4 and section 2.0.

Physical Dimensions inches (millimeters) unless otherwise noted



SO Package (M)
Order Number ADC0802LCWM or ADC0804LCWM
NS Package Number M20B



Molded Dual-In-Line Package (N)
Order Number ADC0801LCN, ADC0802LCN,
ADC0803LCN, ADC0804LCN or ADC0805LCN
NS Package Number N20A

AC Electrical Characteristics (Continued)

Note 7: The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input and therefore timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see timing diagrams).

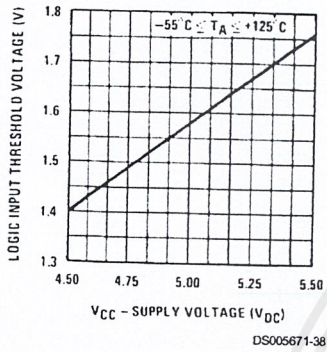
Note 8: None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and Figure 7.

Note 9: The $V_{REF}/2$ pin is the center point of a two-resistor divider connected from V_{CC} to ground. In all versions of the ADC0801, ADC0802, ADC0803, and ADC0805, and in the ADC0804LCJ, each resistor is typically 16 k Ω . In all versions of the ADC0804 except the ADC0804LCJ, each resistor is typically 2.2 k Ω .

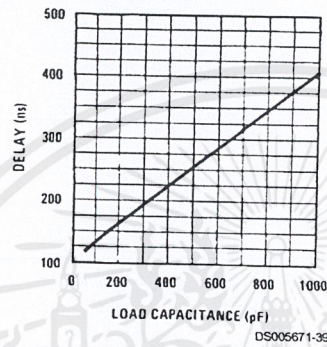
Note 10: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Typical Performance Characteristics

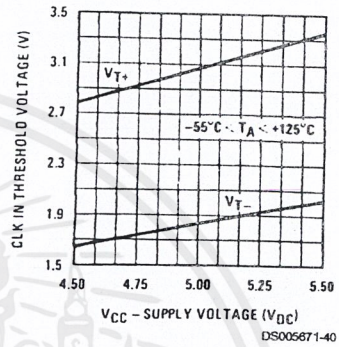
Logic Input Threshold Voltage vs. Supply Voltage



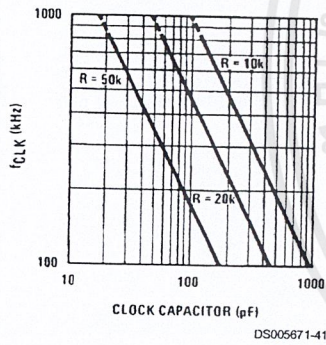
Delay From Falling Edge of $\overline{RD}$ to Output Data Valid vs. Load Capacitance



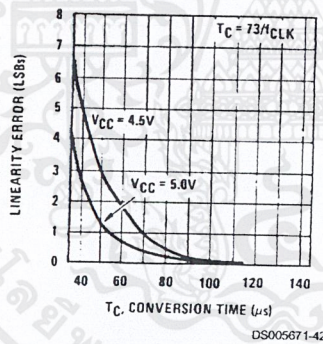
CLK IN Schmitt Trip Levels vs. Supply Voltage



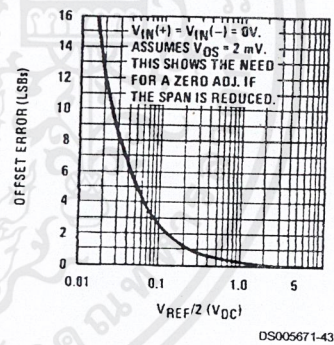
f_{CLK} vs. Clock Capacitor



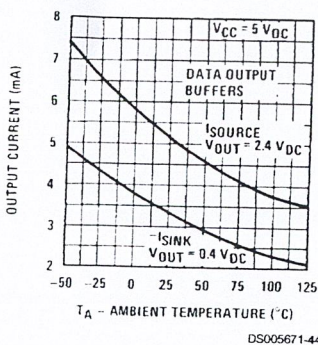
Full-Scale Error vs. Conversion Time



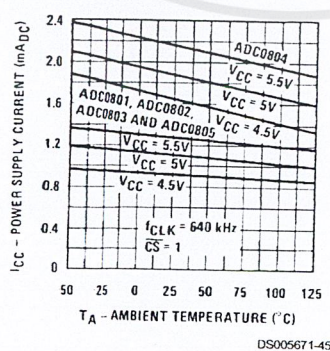
Effect of Unadjusted Offset Error vs. $V_{REF}/2$ Voltage



Output Current vs. Temperature



Power Supply Current vs. Temperature (Note 9)



Linearity Error at Low $V_{REF}/2$ Voltages

