

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจัดการที่จอดรถ

CARS PARKING MANAGEMENT



นายสุนทร เศษแสนวงศ์

นายมนัส สีหานาม

นางสาวระพีพันธ์ เทียนชัย

เลขที่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

83321

11 ธ.ค. 2551

b.11d.4b.21x...
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CARS PARKING MANAGEMENT



MR. SUNET SETSANWONG

MR. MANAT SEEHANAM

MS. RAPEEPUNT THAINCHAI

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

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

การจัดการที่จอดรถ

นักศึกษา

นายสุนทร เศษแสนวงศ์ รหัส 48015642

นายมนัส สีหนาม รหัส 48015630

นางสาวระพีพันธ์ เทียนชัย รหัส 48015632

อาจารย์ที่ปรึกษา

ผศ. สมภพ แก้วมีชัย

ระดับการศึกษา

ปริญญาตรีวิศวกรรมศาสตร์ สาขาวิศวกรรมสารสนเทศ

ภาควิชา

วิศวกรรมสารสนเทศ

ปีการศึกษา

2550

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง



(ผศ. สมภพ แก้วมีชัย)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การจัดการที่จอดรถ
นักศึกษา	นายสุเนตร เศษแสนวงศ์ รหัส 48015642 นายมนัส สีหานาม รหัส 48015630 นางสาวระพีพันธ์ เทียนชัย รหัส 48015632
อาจารย์ที่ปรึกษา	ผศ. สมภพ แก้วมีชัย
ระดับการศึกษา	ปริญญาตรีวิศวกรรมศาสตร์ สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2550

บทคัดย่อ

เนื่องจากในปัจจุบันการนำรถเข้าไปจอดในลานจอดรถตามอาคารสถานที่ต่าง ๆ ต้องใช้เวลาในการหาตำแหน่งที่ว่างสำหรับจอดรถ ซึ่งจากจำนวนผู้ใช้รถมีมากขึ้นทำให้ต้องใช้เวลามากขึ้นในการหาที่จอดรถในแต่ละครั้ง ซึ่งก่อให้เกิดปัญหาการจราจรติดขัด ภายในอาคารจอดรถ ซึ่งทำให้เสียเวลาและยังมีผลทำให้สิ้นเปลืองน้ำมันและเกิดมลภาวะตามมา ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่ต้องทราบตำแหน่งที่จอดรถที่แน่นอน เพื่อสามารถนำรถเข้าไปจอดตามตำแหน่งได้อย่างสะดวกและรวดเร็ว

โครงการปริญญานิพนธ์นี้เป็นโครงการที่ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 นำมาประยุกต์ใช้งานเพื่อแสดงตำแหน่งที่ว่างของลานจอดรถแต่ละชั้น และจำนวนของรถที่มีอยู่ภายในอาคารทั้งหมด โดยวงจรตรวจนับจะใช้แสงอินฟราเรดตรวจสอบตำแหน่งของที่จอดรถแต่ละตำแหน่งซึ่งจะมีตัวตรวจนับ แล้วนำสัญญาณที่ได้ส่งไปยังไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งจะเก็บค่าของข้อมูลต่าง ๆ เอาไว้ โดยค่าทั้งหมดที่ได้จะมีการแสดงผลไปยังจอคอมพิวเตอร์โดยที่มีโปรแกรม Visual Basic คำนวณค่าต่าง ๆ ที่รับมาจากไมโครคอนโทรลเลอร์แล้วจึงแสดงผลออกมาทางจอมอนิเตอร์

Thesis Title	Cars Parking Manadement	
Student	Mr. Sunet Setsanwong	ID. 48015642
	Mr. Manat seehanam	ID. 48015630
	Ms. Rapeepunt Thainchai	ID. 48015632
Advisor	Asst. Prof. Sompoph Kewmechai	
Graduate Level	Bachelor Degree of Information Engineering	
Department	Information Engineering	
Academic Year	2007	

ABSTRACT

At present, people have to spend a long time to look for available space to park their cars in a car park building or a public car park which leads to traffic jams, time and fuel loss antipollution. An important thing to know is if space is available to park their car as quickly and conveniently as possible. This study project, uses a MCS-51 microcontroller application to show available spaces in a car park on each floor and the total number of cars in a whole building. Twenty two sets of detector circuits using infrared rays scan each car park position and send obtained signals to the MCS-51 microcontroller. This collects values of data and all of these values express their results on a computer monitor by Visual Basic Program, which then calculates these received values from the microcontroller to express out puts.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดีในทุก ๆ ด้าน ก็ด้วยความร่วมมือและการช่วยเหลือเป็นอย่างดีจากท่านอาจารย์เจ้าหน้าที่และเพื่อนๆ ทุกคน ในภาควิชาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยเฉพาะอย่างยิ่งจากท่านผู้ช่วยศาสตราจารย์ สมภพ แก้วมีชัย ซึ่งท่านเป็นที่ปรึกษาโครงการ และยังให้คำปรึกษาคำแนะนำ ต่างๆ ที่มีประโยชน์อย่างยิ่งต่อการทำโครงการนี้ให้เป็นผลสำเร็จลุล่วงไปได้ด้วยดี ตลอดมา และต้องขอขอบคุณทุก ๆ ท่าน ที่ได้ให้ความช่วยเหลือคำแนะนำช่วยแก้ไขปัญหามา ทำให้โครงการนี้เสร็จสมบูรณ์ โดยทางผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์แก่ผู้สนใจไม่มากนักน้อย หากในการจัดทำโครงการนี้มีข้อผิดพลาดประการใดผู้จัดทำก็ขออภัยไว้ ณ ที่นี้ด้วย

สุดท้ายนี้ต้องขอกราบขอบพระคุณ คุณพ่อคุณแม่ที่ได้ให้การสนับสนุนทางด้านคุณทรัพย์ทางการศึกษาและเป็นกำลังใจให้ด้วยดีตลอดมา

ผู้จัดทำ

นายสุเนตร เศษแสนวงศ์

นายมนัส สีหามาม

นางสาวระพีพันธ์ เทียนชัย

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญรูปประกอบ	ง
สารบัญตาราง	ช
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการปริญญานิพนธ์	1
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีที่สำคัญและงานวิจัยที่เกี่ยวข้อง	
2.1 ทฤษฎีเกี่ยวกับเซ็นเซอร์	3
2.2 วงจรอปแอมป์	10
2.3 ไมโครคอนโทรลเลอร์ MCS-51	11
2.4 การสื่อสารแบบอนุกรม	35
2.5 โปรแกรม Visual Basic	40
บทที่ 3 วิธีการดำเนินงาน	
3.1 โครงสร้างของระบบในส่วนต่างๆของโครงการ	57
3.2 การออกแบบโปรแกรม Visual basic	63
3.3 แบบจำลองที่จอผล	67
บทที่ 4 ผลการดำเนินงาน	
4.1 ผลการทดลองวงจรจริง	69
4.2 ทดสอบการทำงานของระบบ	71
บทที่ 5 สรุปผลการดำเนินงาน	
5.1 สรุปผล	77
5.2 ปัญหาและแนวทางการแก้ไข	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง

หน้า

5.3 แนวทางการพัฒนาต่อ

78

บรรณานุกรม

ภาคผนวก ก.

ภาคผนวก ข.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2-1 แสดงหลักการของตัวส่งและรับแสง	6
รูปที่ 2-2 ตัวอย่างแอลอีดีภาคแสดงต่าง ๆ	6
รูปที่ 2-3 ลักษณะทั่วไปของแอลอีดีอาร์และสัญลักษณ์	8
รูปที่ 2-4 แสดงการทำงานและลักษณะสมบัติของโฟโต้ไดโอด	9
รูปที่ 2-5 แสดงวงจรเปรียบเทียบแรงดัน	10
รูปที่ 2-6 แสดงวงจรแรงดันตาม	10
รูปที่ 2-7 แสดงโครงสร้างภายในของ MCS-51	16
รูปที่ 2-8 แสดงขาต่างๆของ 89C52	17
รูปที่ 2-9 ขาของ MCS-51 ที่ใช้ต่อกับ XTAL	19
รูปที่ 2-10 โครงสร้างพอร์ตทั้ง 4 ของ MCS-51	20
รูปที่ 2-11 การต่อพอร์ตเข้ากับระบบบัสภายในของ MCS-51	21
รูปที่ 2-12 การจัดหน่วยความจำของ MCS-51	21
รูปที่ 2-13 ตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต	22
รูปที่ 2-14 ไคอะแกรมกลุ่มสัญญาณที่ใช้อ่านข้อมูล	23
รูปที่ 2-15 ไคอะแกรมการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก	24
รูปที่ 2-16 การต่อ MCS - 51 กับหน่วยความจำโปรแกรมภายนอก	25
รูปที่ 2-17 ไคอะแกรมเวลาการอ่านและเขียนข้อมูลกับหน่วยความจำภายนอก	26
รูปที่ 2-18 การต่อหน่วยความจำโปรแกรมกับ MCS-51	27
รูปที่ 2-19 การรีเซ็ต MCS-51	27
รูปที่ 2-20 ผังการทำงานโหมด 0	31
รูปที่ 2-21 ผังการทำงานโหมด 1	32
รูปที่ 2-22 ผังการทำงานโหมด 2	33
รูปที่ 2-23 ผังการทำงานโหมด 3	34

สารบัญรูป(ต่อ)

หน้า

รูปที่ 2-24 แผนผังคอนเน็กเตอร์ของ RS-232	36
รูปที่ 2-25 ไดอะแกรมโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม	38
รูปที่ 2-26 แสดงสภาพแวดล้อมของการพัฒนาโปรแกรม Visual Basic	41
รูปที่ 2-27 แสดงการใช้งานของทูลบาร์	41
รูปที่ 2-28 แสดงการใช้งานของทูลบาร์	43
รูปที่ 2-29 แสดงวินโดว์ Form	44
รูปที่ 2-30 แสดงวินโดว์ Project Explorer	45
รูปที่ 2-31 ภาพแสดงโครงสร้างแฟ้มข้อมูลส่วนต่าง ๆ ที่ประกอบขึ้นมาเป็น โปรเจกใน Visual Basic	46
รูปที่ 2-32 แสดงวินโดว์ Properties	47
รูปที่ 2-33 แสดงวินโดว์ Form Layout	48
รูปที่ 2-34 แสดงวินโดว์ Code Editor	49
รูปที่ 2-35 แสดงโปรแกรมย่อย	52
รูปที่ 2-36 แสดง Event Procedure	53
รูปที่ 2-37 แสดงขอบเขตของตัวแปร	54
รูปที่ 3-1ก. แสดงไฟล์ชาร์ตขั้นตอนการดำเนินงานของโครงการ	55
รูปที่ 3-1ข. แสดงไฟล์ชาร์ตขั้นตอนการดำเนินงานของโครงการ	56
รูปที่ 3-2 แสดงบล็อกไดอะแกรมของแบบจำลองระบบบอกตำแหน่งพื้นที่ว่างในอาคารจอดรถ	57
รูปที่ 3-3 วงจรภาคส่งแสงอินฟราเรด	57
รูปที่ 3-4 วงจรภาครับของเซนเซอร์อินฟราเรด	58
รูปที่ 3-5 แสดงลายวงจรด้านล่างวงจรตรวจนับรวม 8 ตัว	58

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3-6 แสดงแผงวงจรตรวจจับโดยใช้แสงอินฟราเรดที่ใช้งานจริง	59
รูปที่ 3-7 แสดงขาต่างๆของ 89C52	60
รูปที่ 3-8 บอร์ดไมโครคอนโทรลเลอร์ MCS-51	60
รูปที่ 3-9 การทดลองให้สวิตช์แทนเซนเซอร์ในโปรแกรม Proteus	61
รูปที่ 3-10 โฟล์ชาร์ตแสดงขั้นตอนการทำงานของโปรแกรมควบคุม ไมโครคอนโทรลเลอร์	63
รูปที่3-11 แสดงแผนผังของที่จอตลอดใช้ โปรแกรม Visual basic	64
รูปที่ 3-12 แสดงตำแหน่งที่มีรถจอดและจำนวนรถทั้งหมด	65
รูปที่ 3-13 โฟล์ชาร์ตแสดงขั้นตอนการทำงานของโปรแกรม วิชวลเบสิก	66
รูปที่ 3-14 แบบจำลองที่จอตลอดที่ติดตั้งเซ็นเซอร์แล้ว	67
รูปที่ 3-15 แบบจำลองที่จอตลอดขณะมีรถจอดพร้อมชุดเซ็นเซอร์	68
รูปที่4-1 แสดงการทดลองวงจรตรวจจับ	69
รูปที่4-2 แสดงโครงสร้างถานจอตลอด	72
รูปที่4-3 แสดงตำแหน่งต่างๆ ของถานจอตลอดที่แสดงทางหน้าจอคอมพิวเตอร์	72
รูปที่4-4 แสดงโครงสร้างถานจอตลอดขณะมีรถจอด	73
รูปที่4-5 แสดงตำแหน่งที่มีรถจอตลอดออกจากหน้าจอคอมพิวเตอร์	73
รูปที่4-6 แสดงโครงสร้างถานจอตลอดขณะมีรถจอด	74
รูปที่4-7 แสดงตำแหน่งที่มีรถจอตลอดและมีการจองออกจากหน้าจอคอมพิวเตอร์	74

สารบัญตาราง

หน้า

ตารางที่ 2-1 แสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ต่างๆ	15
ตารางที่ 2-2 แสดงบิตและหน้าที่ต่างๆ ของพอร์ต 3	18
ตารางที่ 2-3 โหมดการทำงานของพอร์ตสื่อสารอนุกรม	28
ตารางที่ 2-4 แสดงฟังก์ชันของ RS – 232	37
ตารางที่ 2-5 แสดงข้อมูลในแอดเดรส 0000:041H ที่ใช้ในพอร์ตอนุกรม	39
ตารางที่ 2-6 ตารางแสดงประเภทไฟล์ต่าง ๆ	47
ตารางที่ 2-7 ชนิดตัวแปรใน Visual Basic	50
ตารางที่ 3-1 รายละเอียดขาอินพุต/เอาต์พุต ของไมโครคอนโทรลเลอร์ MCS-51	62
ตารางที่ 4-1 ก. การทดลองวัดค่าของเซ็นเซอร์	70
ตารางที่ 4-1 ข. การทดลองวัดค่าของเซ็นเซอร์	71
ตารางที่ 4-2 ก. ผลการทดลองระบบบอกตำแหน่งที่ว่างของลานจอดรถ	75
ตารางที่ 4-2 ข. ผลการทดลองระบบบอกตำแหน่งที่ว่างของลานจอดรถ	76

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในปัจจุบันนี้ตามอาคารสถานที่ต่างๆ ที่จำเป็นจะต้องไปติดต่อหรือใช้บริการ เช่น สถานที่ราชการ ห้างสรรพสินค้า หรือ สถานที่ประกอบธุรกิจต่างๆ โดยที่จำเป็นต้องใช้ยานพาหนะในการเดินทางคือรถยนต์ ทางสถานที่ต่างๆ นั้นก็จะจัดเตรียมสถานที่เพื่ออำนวยความสะดวกในการจอดรถยนต์ไว้ให้ แต่ในบางสถานที่นั้นมีพื้นที่จำกัดในการนำรถยนต์เข้าไปจอดเป็นเหตุทำให้ต้องเสียเวลาในการหาที่ที่ใช้ในการจอดรถยนต์ อาจเป็นเหตุให้เกิดความล่าช้าในการติดต่อหรือใช้บริการในสถานที่นั้นๆ จากปัญหาดังกล่าวข้างต้นนั้น ทำให้ทางผู้จัดทำได้คิดค้นหาวิธีทางในการแก้ปัญหา โดยการจัดทำโครงการแบบจำลองระบบบอกตำแหน่งพื้นที่ว่างในอาคารจอดรถอัตโนมัติ(Cars Parking Management) ซึ่งจะสามารถช่วยลดปัญหาการเสีย เวลาในการค้นหาตำแหน่งสำหรับจอดรถยนต์ ในบริเวณที่จอดรถของอาคารสถานที่นั้นๆ ได้ในระดับหนึ่ง ในโครงการนี้ ยังสามารถให้ผู้ให้บริการที่มาใช้บริการเป็นประจำจองพื้นที่จอดรถก่อนนำมาจอดได้ จึงเป็นการลดความล่าช้าสำหรับผู้ที่มาติดต่อหรือใช้บริการในสถานที่นั้นๆ ด้วย

1.2 วัตถุประสงค์ของโครงการปริญญานิพนธ์

- 1.2.1 เพื่อศึกษาหลักการและการทำงานของอุปกรณ์ตรวจจับสัญญาณทางแสง(Light Sensor)
- 1.2.2 เพื่อศึกษาการเขียน โปรแกรมภาษาวิซวลเบสิก (Visual Basic)
- 1.2.3 เพื่อศึกษาการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกโดยโปรแกรมวิซวลเบสิก (Visual Basic)
- 1.2.4 เพื่อศึกษาหลักการในการรับ-ส่งข้อมูลแบบดิจิทัล
- 1.2.5 เพื่อศึกษาการทำงานและการประยุกต์ใช้งานของไมโครคอนโทรลเลอร์ MCS AT89C52
- 1.2.6 เพื่อสร้างแบบจำลองระบบบอกตำแหน่งพื้นที่ว่างในอาคารจอดรถอัตโนมัติ (Cars Parking Management)

1.3 ขอบเขตของโครงการ

1.3.1 แบบจำลองของอาคารจอร์จรยนต์สามารถจอร์จรยนต์ได้ 30 คัน โดยที่ทางเข้าและออกของอาคารให้ผ่านเข้า-ออกได้เฉพาะรถยนต์เท่านั้น

1.3.2 ระบบสามารถแสดงจำนวนที่ว่าง-ไม่ว่างทั้งหมด สำหรับที่จอร์จรยนต์ภายในอาคารจอร์จรยนต์โดยผ่านทางจอภาพได้

1.3.4 ระบบจะมีการแสดงแผนผังของอาคารจอร์จรไปที่จอภาพ เพื่อแสดงตำแหน่งที่ว่างไม่ว่าง ซึ่งจะมีการติดตั้งจอภาพไว้ใกล้กับชุด โมเดลของอาคารจอร์จร

1.3.5 ระบบสามารถจองตำแหน่งที่จอร์จรได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 เพื่อเป็นประโยชน์ในการอำนวยความสะดวกแก่ผู้ใช้บริการของ อาคารสถานที่นั้นๆ จัดทำเพื่อผู้ที่ใช้รถยนต์เป็นพาหนะในการเดินทางเป็นหลัก

1.4.2 หากว่าผู้ที่เข้ามาหาที่จอร์จรสามารถทราบว่าที่จอร์จรเหลืออยู่หรือที่จอร์จรเต็มแล้ว โดยดูได้จากจอภาพที่ทางเข้า ถ้าหากว่าที่จอร์จรเต็มแล้วก็ไม่จำเป็นต้องเข้าไปจอดโดยสามารถที่จะไปจอดที่จอร์จอื่นๆ ได้ ซึ่งจะทำให้ลดการเสียเวลาโดยเปล่าประโยชน์จากการนำรถเข้าไปจอดในที่ที่มีรถจอดเต็มอยู่แล้ว

1.4.3 แบบจำลองของอาคารจอร์จรยนต์ที่จัดทำขึ้นนี้ อาจนำไปใช้เป็นแนวทางในการพัฒนาในการขยายพื้นที่ และนำไปใช้กับอาคารจอร์จรที่ใช้งานจริงต่อไปในอนาคตได้

1.4.4 เป็นการนำความรู้ที่มีอยู่เดิม จากการศึกษาในชั้นเรียน มาประยุกต์ใช้ให้ได้ สิ่งประดิษฐ์ใหม่ๆ ที่ทำให้เกิดประโยชน์ต่อไป

บทที่ 2

ทฤษฎีที่สำคัญและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะเป็นส่วนของทฤษฎีทั้งหมดของปริญญาโทซึ่งเนื้อหาจะประกอบด้วยหัวข้อต่าง ๆ ดังนี้

- 2.1 ทฤษฎีเกี่ยวกับเซ็นเซอร์ (Sensor)
- 2.2 ไมโครคอนโทรลเลอร์
- 2.3 การสื่อสารแบบอนุกรม
- 2.4 การสื่อสารแบบพอร์ตนาน
- 2.5 โปรแกรม Visual Basic

2.1 ทฤษฎีเกี่ยวกับเซ็นเซอร์ (Sensor)

เซ็นเซอร์เป็นอุปกรณ์ชนิดหนึ่งซึ่งออกแบบมาให้สามารถใช้งานได้หลายรูปแบบ และยังสามารถใช้ในการใช้งานอีกด้วย โดยสามารถประกอบเป็นวงจรหรือใช้งานร่วมกับอุปกรณ์ตัวอื่นได้ เซ็นเซอร์นั้นเปรียบเสมือนกับดวงตาซึ่งเป็นประสาทสัมผัสของคนที่มีความสำคัญยิ่งในโครงการนี้เราจะมองว่าการที่เราใช้เซ็นเซอร์เป็นสิ่งที่ตรวจจับสถานะใดๆเช่น อุณหภูมิ สี แสง หรือวัตถุต่างๆ โดยอาศัยหลักการที่แตกต่างกันไปตามแต่ละชนิดเช่น เซ็นเซอร์สีขาวดำ โดยอาศัยหลักการสะท้อนแสงของสีขาวและดำ ทางฟิสิกส์แล้วจะเห็นว่าสีขาวมีอัตราการสะท้อนแสงมากกว่าสีดำ เราจึงสามารถนำแสงสะท้อนมาเปรียบเทียบได้ โดยใช้ตัวเซ็นเซอร์คือ อุปกรณ์จำพวกโฟโต้ไดโอด (Photo diode) โฟโตทรานซิสเตอร์ (Photo transistor) เป็นต้น ซึ่งมีความไวต่อแสงมาก โดยส่วนใหญ่ผลลัพธ์จะแสดงในรูปของความต้านทานที่เปลี่ยนไปตามสถานะของตัวเซ็นเซอร์นั้นๆ เราสามารถแบ่งตัวเซ็นเซอร์ออกตามชนิดดังต่อไปนี้ ซึ่งในโครงการนี้เราจะเลือกใช้เซ็นเซอร์ ชนิดอินฟราเรดเซ็นเซอร์ในการตรวจจับ

2.1.1 โฟโต้ไดโอดอินฟราเรด ไดโอดทำงานด้วยแสงหรืออุปกรณ์ที่เปลี่ยนแปลงค่าความต้านทานตามแสงโดยจะเปลี่ยนแปลงความต้านทานตามความเข้มของแสงที่มาตกกระทบถ้าแสงมาตกกระทบน้อยจะมีความต้านทานสูง ถ้าแสงตกกระทบบ่อยค่าความต้านทานจะต่ำ โฟโต้ไดโอดจะประกอบด้วยสารกึ่งตัวนำ P และ N ต่อชนกัน มีขาต่อออกมาใช้งาน 2 ขาคือขาเอโนด (A) และขาแคโทด (K) ส่วนวัสดุที่นำมาผลิตมีอยู่หลายชนิด เช่น ซิลิคอน เจอร์เมเนียม เซเลเนียม เป็นต้น

2.1.2 โฟโตทรานซิสเตอร์อินฟราเรด เป็นอุปกรณ์สารกึ่งตัวนำทางแสงอีกชนิดหนึ่งที่ถูกสร้างและพัฒนามาใช้งานแทนโฟโตไดโอดเพราะมีข้อดีกว่าโฟโตไดโอดตรงที่ในการใช้งานสามารถทำการขยายสัญญาณที่ส่งออกไปได้แสงเพียงเล็กน้อยก็สามารถทำงานได้ดีการทำงานของโฟโตทรานซิสเตอร์ เมื่อจ่ายไบอัสตรงที่ขา C และขา E ของโฟโตทรานซิสเตอร์ และไม่มีแสงส่องมากระทบรอยต่อที่โฟโตทรานซิสเตอร์ตรงรอยต่อสาร PN ระหว่างขา C และขา B แล้วโฟโตทรานซิสเตอร์จะไม่นำกระแส เมื่อมีแสงส่องมาตกกระทบรอยต่อสาร PN ระหว่างขา C และขา B จะทำให้เกิดกระแส I_b ไหลส่งผลให้โฟโตทรานซิสเตอร์ทำงาน เราสามารถหากระแสคอลเลกเตอร์ได้จากสมการดังนี้

$$I_c = h_{FE} I_A \quad (2.1)$$

เมื่อ I_c = กระแสคอลเลกเตอร์
 I_A = กระแสเบสที่เกิดขึ้นจากแสงอินฟราเรดมาตกกระทบ
 h_{FE} = อัตราขยายกระแสของโฟโตทรานซิสเตอร์

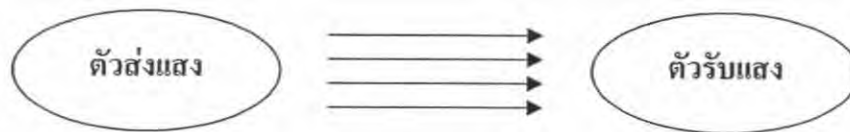
ค่ากระแส IC จะไหลเปลี่ยนแปลงตามการรับแสงอินฟราเรดที่ขาเบส

2.1.3 ออปโตไอโซเลเตอร์ (OPTO ISOLATOR) เป็นอุปกรณ์เชื่อมต่อทางแสง ด้วยอุปกรณ์อิเล็กทรอนิกส์ส่วนๆการส่งผ่านแสง และการรับแสง จะอยู่ในอุปกรณ์เดียวกันทั้งหมดโดยใช้หลักการเปลี่ยนสัญญาณทางไฟฟ้าไปเป็นสัญญาณแสง และเปลี่ยนกลับจากสัญญาณแสงเป็นสัญญาณทางไฟฟ้าเหมือนเดิม จุดประสงค์จุดประสงค์ในการเชื่อมต่อวิธีนี้ก็เพื่อป้องกันการรบกวนของสัญญาณซึ่งกันและกัน และต้องการให้เกิดการแยกกันของวงจรโดยเด็ดขาด แรงดันระหว่างอุปกรณ์ทั้งสองด้าน คือ อุปกรณ์ทางด้านปล่อยแสงและอุปกรณ์ทางด้านรับแสงจะใช้แรงดันมากหรือน้อยขึ้นอยู่กับระยะห่างระหว่างตัวปล่อยแสงและตัวรับแสง ระยะห่างของอุปกรณ์ทั้งสองยิ่งมาก แรงดันที่ป้อนให้อุปกรณ์ทั้งสองจะต้องใช้มากขึ้น แต่มีขีดตรงค่าทนแรงดันของอุปกรณ์ส่วนมากจะสร้างมาในรูปของตัว IC อุปกรณ์ตัวกำเนิดแสงจะเป็นพวก ไลต์อิมิตติงไดโอด (Light-Emitting Diode) ส่วนมากจะเป็นชนิดอินฟราเรด อุปกรณ์รับแสงพวกโฟโตดีเทคเตอร์ (Photo detector) จะนำมาใช้งานหลายชนิดด้วยกันคือ โฟโตทรานซิสเตอร์ทั้งแบบธรรมดาและแบบคาร์ลิงตัน หรืออาจเป็นโฟโตเอสซีอาร์ (Photo Silicon Control Rectifier) เป็นต้น

2.1.4 อินฟราเรดเซ็นเซอร์ แสงอินฟราเรด คือแสงที่มีความยาวคลื่นต่ำกว่าแสงสีแดงลงไป อินฟราเรดเป็นแสงที่มีความยาวคลื่นอยู่ในช่วงประมาณ 780-3000 นาโนเมตร ดังนั้นจึงไม่สามารถมองเห็นได้ด้วยสายตา ของมนุษย์ ซึ่งคุณสมบัตินี้เอง จึงทำให้ เซ็นเซอร์ชนิดที่ใช้แสงอินฟราเรด เป็นที่นิยมนำมาใช้กันมาก โดยจะอาศัยหลักการของการสะท้อนของแสง กล่าวคือ ใช้อุปกรณ์ส่งแสง เป็นแหล่งกำเนิด ปล่อยแสงออกไป และเมื่อแสงกระทบกับวัตถุด้านหน้า มันก็จะสะท้อนแสงกลับมา เข้าที่ตัวรับแสง ส่วนอัตราของการสะท้อนกลับนั้น ขึ้นอยู่กับสี และสภาพความมัน ของวัตถุที่สะท้อน เช่น สีดำ จะมีอัตราการสะท้อนกลับ น้อยกว่าสีขาว , หรือสภาพพื้นผิวที่มีความราบเรียบเป็นมันวาว จะสามารถสะท้อนแสงได้ดีกว่า พื้นผิวที่มีลักษณะด้าน และขรุขระ เป็นต้น แสงอินฟราเรดเป็นแสงที่ไม่สามารถมองเห็นได้ด้วยตาเปล่า จึงเป็นที่นิยมที่จะนำมาใช้ในการสื่อสาร หรือตรวจจับสิ่งของต่างๆ เพราะปัญหาการรบกวนของสัญญาณของแสงอื่นๆมีน้อย อีกทั้งการสร้างวงจรที่ใช้ในระบบอินฟราเรดก็ง่ายไม่มีความซับซ้อนมากนัก และความน่าเชื่อถือของสัญญาณที่ส่งก็มีความเชื่อถือสูง โดยในระบบอินฟราเรดจะต้องมีภาคส่ง และภาครับ ซึ่งการสร้างภาคส่งนั้นก็เพียงให้มีการส่งแสงออกมาในช่วงความถี่ที่สูงกว่าความถี่ทั่วไปของแสงธรรมชาติ คือต้องมากกว่า 20 kHz โดยจะใช้อินฟราเรดอีมิตติงไดโอด(Infrared Emitting Diode) เป็นตัวขับแสงอินฟราเรด ส่วนการสร้างภาครับนั้นเราก็จะใช้ โฟโตทรานซิสเตอร์เป็นตัวรับแสง โดยที่ทั้งภาครับและภาคส่งจะต้องมีระยะ ความสูงและทิศทางเท่ากัน เพราะถ้าไม่เท่ากันจะทำให้การตรวจจับสัญญาณได้ไม่ตรง แต่สำหรับโครงงานนี้จะใช้คุณสมบัติของตัวโฟโตทรานซิสเตอร์ ที่เมื่อมีแสงมาตกกระทบมันแล้วจะทำให้ปริมาณของกระแสที่วิ่งผ่านโฟโตทรานซิสเตอร์มีค่าเพิ่มขึ้นตามปริมาณความเข้มของแสงทำให้ความต่างศักย์ที่ตกคร่อมโฟโตทรานซิสเตอร์มีค่ามากตามไปด้วย และใช้ในการตรวจจับหาแหล่งที่มาของแสงเพื่อที่จะใช้บอกว่าการเคลื่อนไหว

2.1.5 ทฤษฎีที่เกี่ยวกับแสงอินฟราเรด รังสีอินฟราเรดมีลักษณะเช่นเดียวกับแสงธรรมชาติ รวมถึงคลื่นวิทยุ และรังสีเอกซ์ ซึ่งมีการแผ่รังสีออกมาในรูปคลื่นความถี่คุณสมบัติต่างๆ ของแสงอินฟราเรด มีดังนี้ คือ

1. เป็นแสงที่ไม่สามารถมองเห็นได้ด้วยตาเปล่า
2. เดินทางในอากาศด้วยความเร็ว 3×10^8 เมตรต่อวินาที
3. เกิดการสะท้อน และหักเหของแสงได้
4. มีความยาวคลื่นที่ใช้ประมาณ 780-1200 นาโนเมตร



รูปที่ 2-1 แสดงหลักการของตัวส่งและรับแสง

ในการส่งแสงอินฟราเรดจากตัวส่งแสงออกมายังตัวรับแสงจะต้องให้ตัวรับแสงวางอยู่ตรงกับตัวส่งแสงเพราะถ้าหากวางไม่ตรงกันตัวรับแสงจะไม่สามารถรับแสงได้เนื่องจากแสงเดินทางเป็นเส้นตรง ซึ่งจะแสดงหลักการของตัวส่งและรับแสง ดังรูปที่ 2-1

ตัวส่งแสงอินฟราเรด

แอลอีดี

แอลอีดีที่ให้แสงสีแดงนั้นเป็น ไดโอดที่ทำมาจากสารกึ่งตัวนำที่เรียกว่า แกลเลียมอะเซไนด์ (GaAs) ซึ่งมีประสิทธิภาพการใช้งานและความเชื่อถือสูงจะปล่อยแสงออกมาอยู่ในย่านความยาวคลื่นประมาณ 940 นาโนเมตร คุณสมบัติทั่วไปของแอลอีดีก็เหมือนกับของไดโอดโดยในขณะไบแอสตรง มันจะให้กระแสไหลผ่านได้และจะเกิดแรงดันตกคร่อมตัวมัน มีค่าประมาณ 0.6-1 โวลต์ และในขณะที่ไบแอสกลับแอลอีดีจะมีค่าแรงดันทั้งที่ต่ำกว่าไดโอดธรรมดา ดังภาพที่ 2-2 นอกจากแอลอีดีที่เปล่งแสงสีแดงแล้ว เรายังพบแอลอีดีที่ให้แสงสีอื่นอีกมากมาย เช่น แสงสีเขียว สีเหลือง สีแสด เป็นต้น การที่แอลอีดีให้แสงสีแตกต่างกันนี้ขึ้นอยู่กับชนิดของเนื้อสารกึ่งตัวนำที่ใช้หรือขึ้นอยู่กับแถบพลังงานช่องว่างของมันเอง เพราะความยาวคลื่นของแอลอีดีจะขึ้นอยู่กับแถบพลังงานช่องว่างของสารโดยตรง ประโยชน์ของแอลอีดีในปัจจุบันมีมากมาย โดยสิ่งที่เห็นได้ชัดก็คือการใช้แอลอีดีเป็นอุปกรณ์จำพวกภาคแสดงผลของเครื่องคิดเลข เครื่องมือวัด ฯลฯ



รูปที่ 2-2 ตัวอย่างแอลอีดีภาคแสดงต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสงที่เปล่งออกมาจากตัวไดโอดเปล่งแสง (LED) แบ่งออกได้เป็น 2 ชนิด คือ แสงที่ตาคนมองเห็น และแสงที่ตามองไม่เห็น

ก) แสงที่ตาคนมองเห็น ไดโอดเปล่งแสง (LED) ที่เปล่งแสงชนิดที่ตาคนมองเห็น 3 สี คือ สีแดง สีเขียว และสีเหลือง บางครั้งอาจออกสีส้ม ซึ่งก็คือสีเหลืองเข้ม ไดโอดเปล่งแสงชนิดนี้เรียกว่า ไดโอดเปล่งแสง (Light Emitting Diode) หรือ LED เหมือนชื่อที่กล่าวมา

ข) แสงที่ตาคนมองไม่เห็น ไดโอดเปล่งแสง (LED) ที่เปล่งแสงที่ตาคนมองไม่เห็น เพราะเป็นแสงที่อยู่ในย่านอินฟราเรด (Infrared Light) ซึ่งขณะที่เปล่งแสงออกมาตาคนจะมองไม่เห็น ไดโอดเปล่งแสงชนิดนี้เรียกว่า อินฟราเรดเปล่งแสง (Infrared Emitting Diode) หรือ IRED แต่สามารถเรียกชื่อแบบรวมๆกันว่า ไดโอดเปล่งแสงหรือ LED แสงที่ถูกเปล่งออกมาจาก ไดโอดเปล่งแสง (LED) ที่มีสีต่างกันนั้นขึ้นอยู่กับเนื้อสารกึ่งตัวนำที่นำมาใช้ผลิต เมื่อใช้เนื้อสารกึ่งตัวนำต่างกัน เนื้อที่ใส่และสีที่ได้มีความแตกต่างกันดังนี้

1. ใช้สารแกลเลียมอาร์ซีไนด์ (Gallium Arsenide) ใช้ตัวย่อ GaAs ไดโอดเปล่งแสงอินฟราเรดออกมา
2. ใช้สารแกลเลียมอาร์ซีไนด์ ฟอสไฟด์ (Gallium Arsenide Phosphide) ใช้ตัวย่อ GaAsP ไดโอดเปล่งแสงสีแดง หรือสีเหลืองออกมา
3. ใช้สารแกลเลียมฟอสไฟด์ (Gallium Phosphide) ใช้ตัวย่อ GaP ไดโอดเปล่งแสง (LED) จะเปล่งแสงสีแดงหรือสีเขียวออกมา

ลักษณะไดโอดเปล่งแสงแบบอินฟราเรด

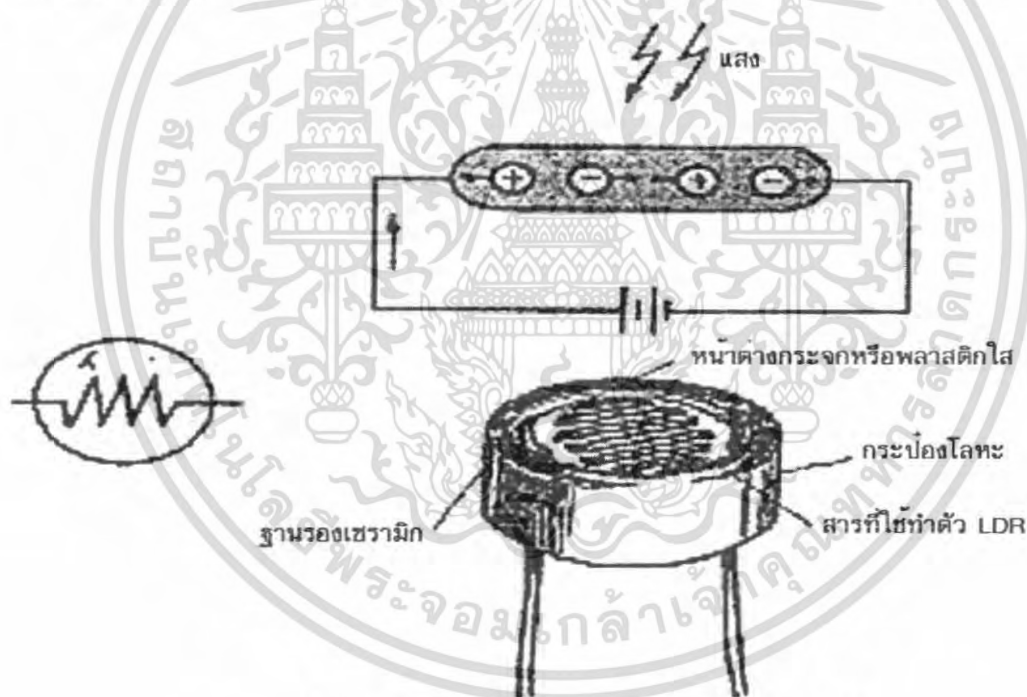
แสงที่เปล่งออกมาจากตัวไดโอดเปล่งแสง (LED) เกิดขึ้นตรงบริเวณรอยต่อของสารชนิด P และสารชนิด N เกิดการกระจายออกรอบตัว ทำให้ความเข้มของแสงลดลงการใช้ ไดโอดเปล่งแสง (LED) จำเป็นต้องมีความเข้มของแสงรวมไปในทิศทางใดทิศทางหนึ่งเฉพาะยิ่งถ้าเป็นไดโอดเปล่งแสง (LED) ที่เปล่งแสงอินฟราเรดออกมา ยังมีความจำเป็นเพราะนิยมนำไปใช้กับระบบควบคุมระยะไกล (Remote Control) ต้องให้แสงเคลื่อนที่ไปในทิศทางเดียว ดังนั้นในส่วน ของตัวถังไดโอดเปล่งแสง (LED) จึงต้องมีส่วนสะท้อนแสง และเลนส์รวมแสงเพิ่มเข้าไป

ตัวรับแสงอินฟราเรด

อุปกรณ์ตัวรับแสง หมายถึง อุปกรณ์ที่สามารถเปลี่ยนพลังงานแสงให้แปรค่ากับค่าของพลังงานทางไฟฟ้าได้ โดยตัวอุปกรณ์จะต้องประกอบด้วยแสงกึ่งตัวนำซึ่งจะนำมาต่อเชื่อมให้เกิดเป็นรอบต่อหรือเป็นเนื้อสารกึ่งตัวนำอย่างเดียวก็น่าจะได้

แอลดีอาร์

ตัวต้านทานไวแสงหรือแอลดีอาร์ แอลดีอาร์ (LDR) เป็นคำย่อของคำว่า light dependent resistor หรือค่าตัวต้านทานที่แปรค่าได้กับแสงนั่นเอง โดยเมื่อป้อนพลังงานอย่างพอเพียงให้กับ วาเลนซ์อิเล็กตรอน ของสารกึ่งตัวนำจะทำให้เกิดการแตกตัวของโฮลและอิเล็กตรอนเกิดขึ้นมากมาย พลังงานที่จะทำให้โฮล และอิเล็กตรอนแตกตัวนี้จะต้องเป็นพลังงานที่มาจากภายนอก เช่น แสง ความร้อน ปริมาณของพลังงานที่ต้องการทำให้เกิดการแตกตัวของอิเล็กตรอน-โฮล จะขึ้นอยู่กับชนิดของสารนั้น ๆ



รูปที่ 2-3 ลักษณะทั่วไปของแอลดีอาร์และสัญลักษณ์

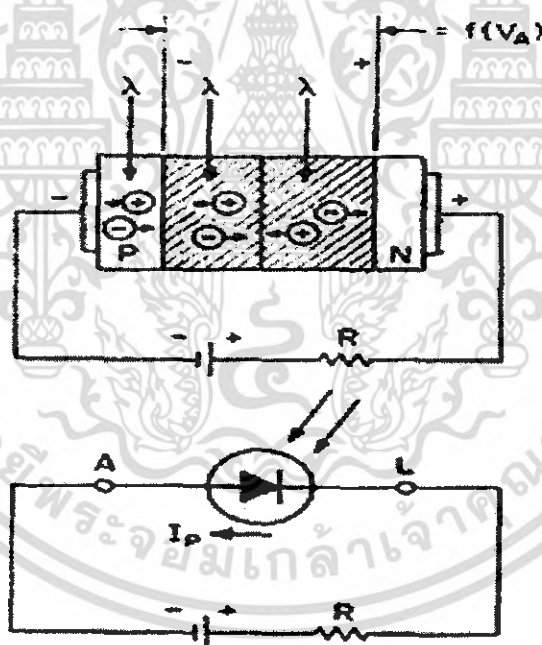
ลักษณะการทำงานของแอลดีอาร์จะเห็นได้ว่าเมื่อแสงมีความเข้มมาก ๆ มาตกกระทบ ก็จะทำให้ตัวมันนำกระแสไฟฟ้าได้ดีขึ้นเป็นผลทำให้เกิดค่าความต้านทานในตัวมันลดลง ข้อดีในการใช้งานของแอลดีอาร์ คือ มีความไวต่อแสงสูง ราคาถูก และมีขนาดเล็ก มีการเปลี่ยนแปลงค่าได้กว้าง แต่ก็ยังมีข้อเสียเมื่อเทียบกับอุปกรณ์รับแสงอื่น ๆ คือมีผลตอบสนองต่อความถี่ของแสงได้แคบดัง

รูปที่ 2-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟโตไดโอด

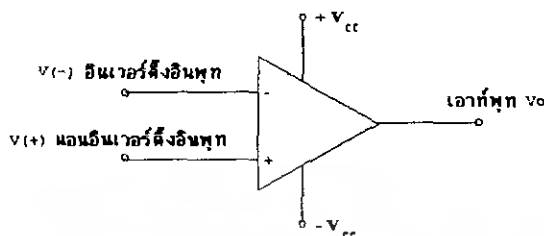
โฟโตไดโอดเป็นอุปกรณ์ไวแสงชนิดหนึ่งซึ่งประกอบด้วยรอยต่อ P-N การใช้งานตัวโฟโตไดโอดจะให้กระแสไหลผ่านตัวมันที่ขึ้นอยู่กับแสงเมื่อโฟโตไดโอดชนิดซิลิกอน ถูกไบแอสกลับด้วยแรงดันค่าหนึ่งและมีแสงส่องไปที่บริเวณรอยต่อ ถ้าแสงที่ส่องมีความยาวคลื่นที่พอเหมาะจะมีกระแสไหลในวงจร โดยเป็นสัดส่วนกับความเข้มของแสงที่ส่องผ่านบนอุปกรณ์นั้น ลักษณะทั่วไปของไบแอสตรงจะยังคงเหมือนกับไดโอดธรรมดา คือให้กระแสผ่านตัวมันเองได้ เมื่อแสงตกกระทบบริเวณรอยต่อจะทำให้เกิดการแตกตัวของโฮลและอิเล็กตรอนอิสระ โฮลและอิเล็กตรอนอิสระจะถูกแรงดันไบแอสกลับเหนี่ยวนำให้เกิดการไหลข้ามรอยต่อเป็นกระแสเนื่องจากแสงได้ โดยปกติค่ากระแสเนื่องจากแสงนี้มีค่าค่อนข้างต่ำมาก คืออยู่ในช่วง 1-10 μA เท่านั้น เนื่องจากไดโอดนี้ให้ค่าการเปลี่ยนแปลงของกระแสต่อแสงค่า ดังนั้นการใช้งานของโฟโตไดโอดจึงต้องมีตัวขยายกระแสเพิ่มเติม เช่น โฟโตทรานซิสเตอร์ ดังรูปที่ 2-4



รูปที่ 2-4 แสดงการทำงานและลักษณะสมบัติของโฟโตไดโอด

2.2 วงจรออปแอมป์

2.2.1 วงจรเปรียบเทียบแรงดัน วงจรเปรียบเทียบแรงดันเป็นวงจรที่ใช้ในการเปรียบเทียบแรงดันระหว่างขาอินเวอร์ตติ้ง(Inverting) และขาอนอินเวอร์ตติ้ง (Non-Inverting) ของออปแอมป์



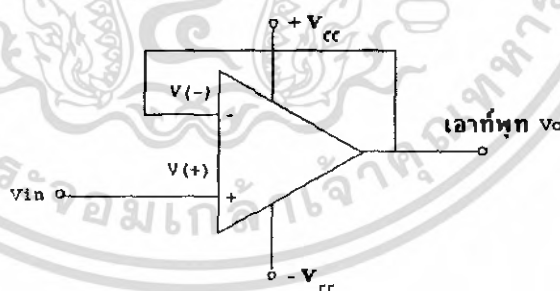
รูปที่ 2.5 แสดงวงจรเปรียบเทียบแรงดัน

$$V_o = +V_{sat} \quad ; \quad V(+)> V(-) \quad (2.2)$$

$$= -V_{sat} \quad ; \quad V(+)< V(-) \quad (2.3)$$

เมื่อ V_o เป็นแรงดันเอาต์พุตของออปแอมป์
 $V(+)$ เป็นแรงดันขาเข้าไม่กลับเฟส (+) ของออปแอมป์
 $V(-)$ เป็นแรงดันขาเข้ากลับเฟส (-) ของออปแอมป์

2.2.2 วงจรแรงดันตาม



รูปที่ 2.6 แสดงวงจรแรงดันตาม

$$V_o = V_{in} \quad (2.4)$$

เมื่อ V_o เป็นแรงดันเอาต์พุตของออปแอมป์
 V_{in} เป็นแรงดันขาเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ไมโครคอนโทรลเลอร์ MCS-51

2.3.1 ไมโครคอมพิวเตอร์ เมื่อเราเริ่มต้นเรียนรู้เรื่องคอมพิวเตอร์พิจารณาได้อย่างไรว่า ชิ้นงานที่เห็นอยู่นั้นเป็นไมโครคอมพิวเตอร์ได้หรือไม่ ให้เราพิจารณาได้จาก องค์ประกอบ ของ ชิ้นงาน ซึ่งมีส่วนประกอบที่สำคัญดังนี้

1. หน่วยประมวลผลกลาง (CPU: Central Processing Unit) มีคุณสมบัติหลัก คือการ ประมวลผลข้อมูลการคำนวณทางคณิตศาสตร์และลอจิก เรารู้จักกันดีในชื่อของไอซีไมโคร โพรเซสเซอร์ เช่น 8080, 80286, 80486 Pentium ฯลฯ ซึ่งเป็นของบริษัท Intel หรืออาจเป็น CPU รุ่น เก่าที่มีขนาด 8 บิต เช่นเบอร์ Z80 ที่เป็น CPUของบริษัท ZILOG เป็นต้น

2. หน่วยความจำ (Memory)

2.1 หน่วยความจำรอม (ROM: Read Only Memory) เป็นหน่วยความจำแบบถาวรที่มี การบันทึกข้อมูลไว้ ล่วงหน้าก่อนแล้วไม่สามารถเปลี่ยนแปลงข้อมูลหรือคำสั่งใดๆ ได้อีก ตัวอย่างเช่น ไอซีที่เป็นไบออส (BIOS) ของคอมพิวเตอร์ ในขณะที่เริ่มเปิดเครื่อง ไมโครคอมพิวเตอร์ครั้งแรก สังเกตได้ว่าจะมีการแสดงชื่อผู้ผลิตของบริษัทหรือ คุณสมบัติของ เครื่องบนหน้าจอโมนิเตอร์ ถ้าหากเป็นเครื่องเล่นวีดีโอ จะเป็นตัวอักษรที่ทำหน้าที่แสดงผลเพื่อ บอกให้ตั้งค่าข้อมูลต่างๆ ซึ่งไม่ว่าจะปิดแล้วเปิดก็ครั้งตัวอักษรเดิมนั้นจะยังคงอยู่

2.2 หน่วยความจำแรม (RAM: Random Access Memory) เป็นหน่วยความจำที่ใช้เก็บ ข้อมูลไว้เพียงชั่วคราวอาจเป็นข้อมูลที่ CPU ต้องการประมวลผลในขณะนั้น และเมื่อ CPU ประมวลผลเรียบร้อยแล้วอาจลบหรือเปลี่ยนข้อมูลได้ บางครั้งเมื่อหยุดการจ่ายไฟให้กับวงจรจะทำให้ ข้อมูลสูญหายไปได้ในทันที ตัวอย่าง เช่น ขณะที่เรากำลังพิมพ์งานแต่ยังไม่ได้บันทึกข้อมูลไว้ใน ส่วนใด ข้อมูลนี้จะถูกเก็บไว้ที่หน่วยความจำแรมก่อนหากเกิดเหตุการณ์ไฟฟ้าดับ จะทำให้ข้อมูลสูญ หายไปได้ หรือการเก็บค่าของเวลาและอุณหภูมิของเครื่องไมโครเวฟ ที่สามารถเปลี่ยนแปลงค่าได้ ตลอด หน่วยความจำแบบแรมนี้จะแตกต่างกับหน่วยความจำแบบรอม โดยหน่วยความจำแบบรอม จะไม่สามารถแก้ไข ข้อมูลได้ในขณะนั้น ในขณะที่หน่วยความจำแบบแรมไม่สามารถเก็บค่าข้อมูล ไว้ได้ตลอดดังนั้นหากต้องการให้ข้อมูลคงอยู่ต้องใช้แบตเตอรี่สำรองไฟฟ้าไว้

3. หน่วยอินพุต (Input Unit) เป็นหน่วยที่ใช้สำหรับการรับสัญญาณข้อมูลจากภายนอกเช่น คีย์บอร์ด สแกนเนอร์ หรือที่รับสัญญาณมาจากอุปกรณ์เซนเซอร์ (Sensor) ซึ่งอาจเป็นค่าแรงเสียดทานของล้อรถยนต์ขณะเบรกการกดปุ่มสวิทช์ตั้งเวลาของวีดีโอเทป ฯลฯ กล่าวได้ว่าส่วนที่เป็นอินพุต คือส่วนที่ทำหน้าที่ป้อนข้อมูล

4. หน่วยเอาต์พุต (Output Unit) เป็นหน่วยที่ใช้สำหรับการแสดงผลของข้อมูลเช่น จอมอนิเตอร์ พรินเตอร์ เครื่องตัดสติ๊กเกอร์หรืออุปกรณ์ประเภทแอลอีดี (LED) ลำโพง มอเตอร์ รีเลย์ หลอดไฟ ฯลฯ

2.3.2 ไมโครคอนโทรลเลอร์ ปัจจุบันการพัฒนาและการแข่งขันทางด้านเทคโนโลยีผลิตชิ้นส่วนสารกึ่งตัวนำ ที่นำไปสร้างเป็นไอซีมี ประสิทธิภาพสูงมากขึ้นและมีเทคโนโลยีที่เกิดจากการผลิตของบริษัทต่างๆ ซึ่ง ส่งผลให้การผลิตชิพไอซีมีขนาดที่เล็กลง แต่มีประสิทธิภาพและคุณสมบัติต่าง ๆ มากขึ้น ไอซีที่ถูกสร้างเป็นแบบ LSI (Large Scale Integrate Circuit) เป็นเทคโนโลยีการสร้างโดยการนำเอาทรานซิสเตอร์จำนวนมากมาสร้างเป็นไอซีดิจิทัลที่ซับซ้อน โดยทำขึ้นเพื่อหน้าที่เป็นหน่วยประมวลผลข้อมูลหรือ เรียกว่าไมโครโพรเซสเซอร์ (Microprocessor) ที่มีคุณสมบัติหลัก คือ การประมวลผลข้อมูลการคำนวณทางคณิตศาสตร์ และลอจิก ถ้าหากมีการติดต่อกับหน่วยความจำที่เป็นแบบแรมแบบรอมหรืออุปกรณ์ภายนอกที่เป็นอินพุต-เอาต์พุตต้องมีการต่ออุปกรณ์อื่น ๆ ร่วมด้วย เพื่อทำหน้าที่เลือกอุปกรณ์ในการติดต่อหรือวงจรถอดรหัส (Decoder) ซึ่งสามารถทำงานได้ภายใต้การควบคุมของโปรแกรมและ ในการที่เรานำไมโครโพรเซสเซอร์มาเป็นตัวประมวลผลกลางมีหน่วยความจำแบบแรมพอร์ตอินพุตและเราเรียกว่า ไมโครคอมพิวเตอร์ เป็นสิ่งไม่คุ้มกับการลงทุนหากนำมาใช้ในงานควบคุมขนาดเล็ก และอาจต้องใช้เนื้อที่มากในการออกแบบ ดังนั้นการพัฒนาชิ้นส่วนเทคโนโลยีในการสร้างชิพจึงมีการรวบรวมคุณสมบัติที่ต้องการใช้งานมาอยู่ในตัวเดียวกันคือ มีองค์ประกอบเกือบทุกอย่างของคอมพิวเตอร์อยู่ภายในตัวไอซีที่เราเรียกว่า ไมโครคอมพิวเตอร์แบบชิพเดี่ยวประกอบด้วยอุปกรณ์พื้นฐานเหมือนไมโครคอมพิวเตอร์ เช่น หน่วยประมวลผลกลางขนาดเล็ก (8 บิต - 16 บิต) และหน่วยประมวลผล ที่สามารถเข้าข้อมูลแบบบิต หน่วยความจำข้อมูลพื้นฐานแบบแรมขนาด 128 ไบต์ และบรรจุหน่วยความจำโปรแกรมประเภทรอม(บางเบอร์) สามารถใช้งานให้เป็นได้ทั้งอินพุตและเอาต์พุตมีวงจรสื่อสารอนุกรมแบบ พูลดูเพล็กซ์ วงจรCounter/Timer ที่อยู่ภายใน สามารถต่ออุปกรณ์ที่ใช้ในการสร้างวงจรถ่ายสัญญาณนาฬิกาเช่น คริสตัล(Crystal) และตัวเก็บประจุก็สามารถใช้งานได้เป็นต้น เราเรียกกันทั่วไปว่า ไอซีไมโครคอนโทรลเลอร์ (Microcontroller) ดังนั้นเมื่อเราต้องการใช้งานควบคุมขนาดเล็ก เช่น เตาไมโครเวฟ เครื่องซักผ้า เครื่องเล่นวีดีโอเทปและเครื่องใช้ไฟฟ้าอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 เหตุผลที่ต้องเลือกไอซีไมโครคอนโทรลเลอร์ MCS-51 จากความก้าวหน้าของเทคโนโลยีจึงพบว่าอุปกรณ์บางส่วนที่นำมาควบคุมเครื่องใช้ไฟฟ้าระบบควบคุมทางอิเล็กทรอนิกส์ หรือ โครงงานในวารสารต่าง ๆ จะนำเสนอในรูปแบบของไอซีไมโครคอนโทรลเลอร์ ซึ่งเทคโนโลยีดังกล่าวมีบริษัทต่างๆ ที่ผลิตออกมาแข่งขันด้านคุณสมบัติ และการใช้งานกันในตลาด อาทิเช่น Intel, Atmel, Microchip, Motorola ส่วนทางด้านการศึกษาและการเรียนรู้นั้น มีการพัฒนาทางด้านการเรียนการสอนวิชาไมโครคอนโทรลเลอร์ โดยบริษัทต่างๆ แข่งขันกันเพื่อสร้างชุดฝึกตอบสนองต่อความต้องการเรียนรู้ ความสะดวกในการสร้างชุดทดลองของนักศึกษาและผู้สนใจ ในขณะเดียวกันกับเรา ซึ่งเป็นผู้ตามเพื่อศึกษาเทคโนโลยีที่ทั้งห่างทุกขณะ จึงต้องใช้ชุดทดลองที่ถูกกำหนดรูปแบบและราคา ทั้งที่อุปกรณ์ ที่เป็นอุปกรณ์ร่วมกับเทคโนโลยีบางอย่าง เรากลับทิ้งไปและไม่ได้นำมาใช้งาน ยกตัวอย่าง เช่น Main Board ,CD-ROM หรือ Hard Disk ที่ใช้งานไม่ได้แล้ว เรา นำอุปกรณ์ภายในมาใช้งานได้ก็ขึ้นบ้าง ตลอดจนตัวต้านทานและตัวเก็บประจุ ที่เป็นอุปกรณ์ติดบนผิวหน้าแบบ SMD (Surface Mounted Device) หน่วยความจำชนิดแรม และรอม แผ่นระบายความร้อนสเตปปีงมอเตอร์ (Stepping Motor) คอนเนคเตอร์ (Connector) ฯลฯ ซึ่งพร้อมที่จะให้เราใช้งานมากกว่าที่จะได้เพิ่มมาจากการซื้อชุดทดลองใหม่

ดังนั้น ไมโครคอนโทรลเลอร์จึงถูกเลือกใช้ในการทดลองด้วยเหตุผลในการใช้อุปกรณ์ที่ค่อนข้างน้อยชิ้น ราคาถูก หาแหล่งข้อมูลได้ง่ายมีการพัฒนาประสิทธิภาพ ในการประมวลผลเทียบเท่าไมโครโพรเซสเซอร์ขนาด 8 บิต-16บิต และ ใช้ในการพัฒนากับงานที่ไม่ต้องการความซับซ้อนซึ่งจะทำให้มีความสะดวกมากขึ้นปัจจุบันมีไอซีของสองบริษัทที่เรานิยมใช้งานออกแบบและสำหรับในการเรียนการสอนในสถาบันหลายแห่ง คือ ไอซีไมโครคอนโทรลเลอร์ของบริษัท Microchip ตระกูล MCS-51และไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51 จากหลายบริษัทที่ผลิตออกมามากมายหลายเบอร์ แต่คำสั่งจะใกล้เคียงกัน ดังนั้น การทดลองในหนังสือเล่มนี้จึงเลือกใช้งานไอซี ไมโครคอนโทรลเลอร์ของตระกูล MCS-51 ด้วยเหตุผลเพราะมีข้อมูล และตัวอย่างที่หลากหลายหาใช้งานได้ง่าย แต่ก็ไม่ได้ตัดสินใจดีกว่าไอซีในตระกูลอื่นๆ ดังนั้น ในการออกแบบชุดทดลองที่ใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ของ MCS-51 ที่จะแนะนำให้สร้างขึ้นนี้สามารถที่จะนำมาใช้งานร่วมกับไอซีไมโครคอนโทรลเลอร์ได้ทุกเบอร์ และทุกตระกูล

2.3.4 ความหมายของไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่บรรจุความสามารถมากมายไม่ว่าจะเป็นหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณออกทางเอาต์พุต หน่วยความจำ และวงจรกำเนิดสัญญาณนาฬิกา ทำให้ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี โดยช่วยลดจำนวนของอุปกรณ์และขนาดของระบบลงในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกัน คือ “ไมโคร (Micro)” ซึ่งหมายถึงไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วยหน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรเชื่อมต่อสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์ (Controller)” หมายถึงอุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์ จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.3.5 โครงสร้างของไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีจำนวนมาก ขึ้นกับโครงสร้างภายใน บางเบอร์มีหน่วยความจำภายในเป็นแบบรวม บางเบอร์เป็นแบบอีพรอม บางเบอร์มีแรมภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น คุณสมบัติที่สำคัญของ MCS-51 มีดังนี้

- หน่วยความจำรวม 4 กิโลไบต์
- หน่วยความจำแรม 128 ไบต์
- พอร์ตอินพุตและเอาต์พุต ขนาด 8 บิต 4 พอร์ต
- Timer 16 บิต 2 ตัว
- สามารถอินเทอร์รัพท์ได้ 5 แหล่ง
- วงจรออสซิลเลเตอร์และวงจรมานาฬิกาบนชิพ
- พอร์ตอนุกรมสามารถรับส่งข้อมูลแบบ Full Duplex ความเร็วสูง
- อ้างหน่วยความจำโปรแกรมภายนอกได้ 64 กิโลไบต์
- อ้างหน่วยความจำข้อมูลภายนอกได้ 64 กิโลไบต์
- สามารถประมวลผลทีละบิต
- อ้างหน่วยความจำแบบบิตได้ 210 ตำแหน่ง

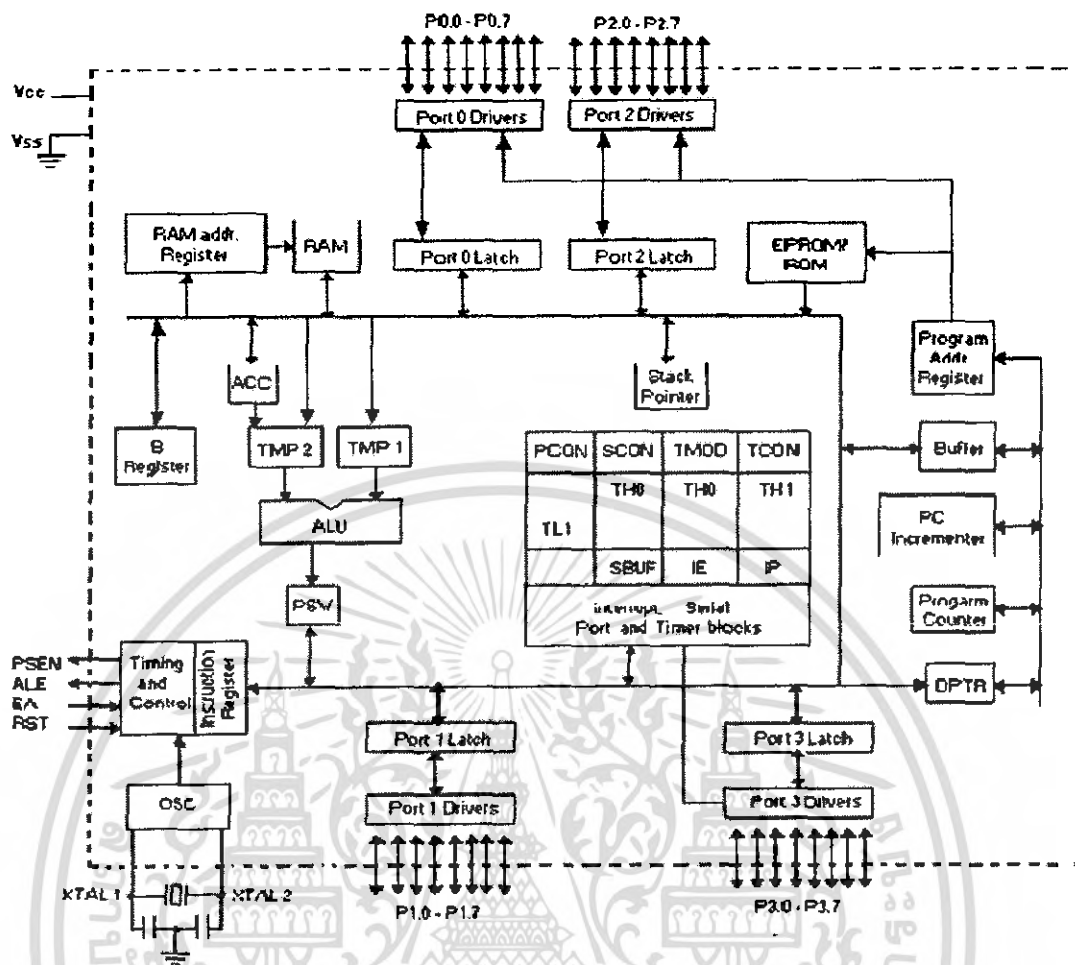
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หนึ่งวัฏจักรคำสั่งกินเวลาประมาณ 1 ไมโครวินาทีขณะทำงานด้วย Clock 12 เมกะเฮิร์ต
- ติดต่อกับหน่วยความจำรวมได้สูงถึง 32 กิโลไบต์
- มีหน่วยความจำแรมภายในขนาด 128 ไบต์และ 256 ไบต์สำหรับเบอร์ 8051 และ 8052
- มีรีจิสเตอร์ 8 บิต 19 ตัว, 16 บิต 5 ตัว - มีพอร์ตอนุกรมในตัว
- สามารถติดต่อกับแหล่งกำเนิดอินเตอร์รัปต์ได้ 5 ชุดและ 9 ชุดสำหรับเบอร์ 8052
- มีพอร์ตอินพุตเอาต์พุต 32 บิต - มีตัวตั้งเวลา ขนาด 16 บิต 2 ตัว และ 3 ตัว ในเบอร์ 8052
- มีคำสั่ง 111 คำสั่งสามารถอ้างแอดเดรสได้ 2 โหมด

ตัวอย่าง ไมโครคอนโทรลเลอร์ตระกูล MCS-51 และลักษณะต่างๆ สามารถแสดงได้ ดังตารางที่ 1 สำหรับไดอะแกรมโครงสร้างภายในแสดงดังรูปที่ 12

เบอร์	หน่วยความจำโปรแกรมบนชิพ	หน่วยความจำข้อมูลบนชิพ	TIMERS
8051	4K ROM	128 bytes	2
8031	-	128 bytes	2
8751	4K EPROM	128 bytes	2
8052	8K ROM	256 bytes	3
8032	-	256 bytes	3
8752	8K EPROM	256 bytes	3

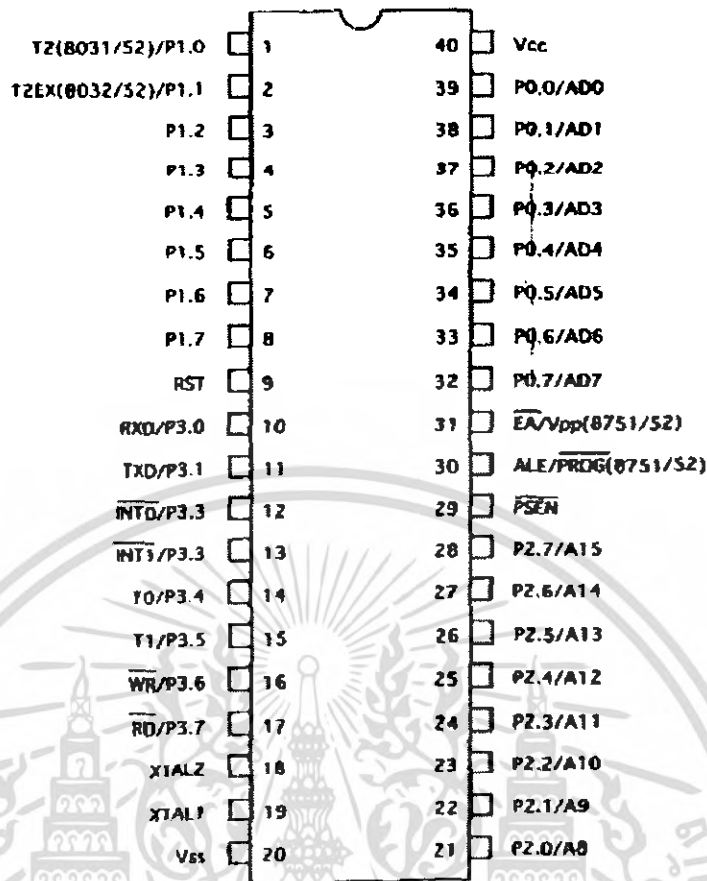
ตารางที่ 2-1 แสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ต่างๆ



รูปที่ 2-7 แสดงโครงสร้างภายในของ MCS-51

2.3.6 การจัดขาต่าง ๆ ของ MCS-51 ไอซี ไมโครคอนโทรลเลอร์ 8051 โครงสร้าง ไอซี เป็นแบบ DIP มีขาทั้งหมด 40 ขา โดยขาต่างๆ เป็นขาพอร์ทอินพุต, ขาพอร์ทเอาต์พุต, ขาสัญญาณควบคุม, ขาด้านแหล่งหน่วยความจำข้อมูลดังรูปที่ 2-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 แสดงขาต่างๆของ 89C52

ความหมายขาต่างๆ มีดังนี้

1. พอร์ต 0 (Port 0)

พอร์ต 0 ได้แก่ขา 32-39 สามารถเป็นพอร์ตอินพุตและเอาต์พุตได้ ซึ่งในการติดต่อกับหน่วยความจำภายนอกสามารถเป็นขาบัสดำแหน่ง และบัสดข้อมูลได้

2. พอร์ต 1 (Port 1)

พอร์ต 1 ได้แก่ขา 1-8 เป็นพอร์ต 8 บิต สามารถอ้างทีละบิตได้ เป็น P1.0, P1.1,...etc

3. พอร์ต 2 (Port 2)

พอร์ต 2 ได้แก่ขา 21-28 สามารถทำงานได้ 2 หน้าที่ เป็นพอร์ต 8 บิต และเป็นขาแอดเดรส 8 บิต ในการอ้างหน่วยความจำภายนอก

4. พอร์ต 3 (Port 3)

พอร์ต 3 ได้แก่ขา 10-17 สามารถทำงานได้ 2 หน้าที่ เป็นพอร์ตอินพุตและเอาต์พุต และเป็นขาควบคุมต่างๆ ดังตารางที่ 2

บิต	ชื่อ	หน้าที่พิเศษ
P3.0	RXD	ใช้รับข้อมูลทางพอร์ตอนุกรม
P3.1	TXD	ใช้ส่งข้อมูลทางพอร์ตอนุกรม
P3.2	$\overline{\text{INT0}}$	อินเทอร์รัพท์ภายนอกหมายเลข 0
P3.3	$\overline{\text{INT1}}$	อินเทอร์รัพท์ภายนอกหมายเลข 1
P3.4	T0	ตัวจับเวลา / ตัวนับ คำที่ 0
P3.5	T1	ตัวจับเวลา / ตัวนับ คำที่ 1
P3.6	$\overline{\text{WR}}$	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก
P3.7	$\overline{\text{RD}}$	สัญญาณอ่านข้อมูลหน่วยความจำภายนอก

ตารางที่ 2-2 แสดงบิตและหน้าที่ต่างๆ ของพอร์ต 3

5. $\overline{\text{PSEN}}$ (Program Store Enable)

ขา $\overline{\text{PSEN}}$ ได้แก่ขา 29 เป็นขาที่ส่งสัญญาณออก ขานี้ทำการแอกทีฟเมื่อ MCS-51 ต้องการอ่าน Code โปรแกรมภายนอก โดยปกติถ้าหน่วยความจำภายนอกเป็นอีพรอมขา $\overline{\text{PSEN}}$ คู่กับขา Output Enable ($\overline{\text{OE}}$) ของอีพรอม

6. ALE (Address Latch Enable)

เนื่องจากพอร์ต 0 สามารถเป็นขาอ้างตำแหน่ง และขาข้อมูล MCS-51 ซึ่งมีขา ALE ได้แก่ขา 30 ขานี้ใช้ผสมสัญญาณบัสตำแหน่งของพอร์ต 0 ในการใช้งานระบบ MCS-51 นั้น ต้องมีอุปกรณ์มาต่อกับพอร์ต 0 ที่ทำหน้าที่ Latch สัญญาณบัสตำแหน่ง เมื่อ MCS-51 ติดต่อกับ หน่วยความจำภายนอก MCS-51 ส่งสัญญาณบัสตำแหน่งออกมาก่อนทาง พอร์ต 0 จากนั้นส่งสัญญาณ ALE มา Latch อุปกรณ์ภายนอก ให้เก็บค่าบัสตำแหน่งของพอร์ต 0 ไว้เพื่อใช้พอร์ต 0 เป็นบัสข้อมูลต่อไป

7. $\overline{\text{EA}}$ (External Access)

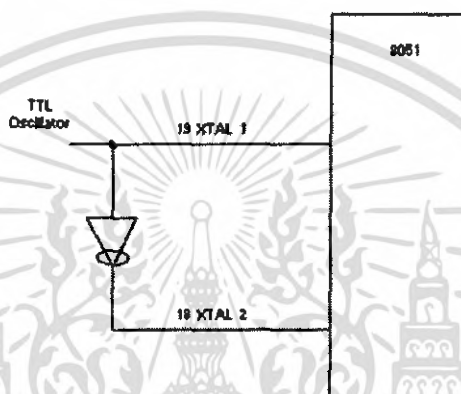
ขา $\overline{\text{EA}}$ ได้แก่ขา 31 ในสถานะเป็นลอจิก "1" ใช้กับเบอร์ 8051/8052 เพื่อให้อ่านโปรแกรมจากหน่วยความจำภายใน และในสถานะเป็นลอจิก "0" ทำให้ MCS-51 ทำโปรแกรมโดยอ่านจากหน่วยความจำโปรแกรมภายนอก (เมื่อ $\overline{\text{EA}}$ เป็น "0" ขา $\overline{\text{PSEN}}$ ทำการแอกทีฟ) หากเป็นเบอร์ 8031 หรือ 8032 ขา $\overline{\text{EA}}$ เป็น "0" เสมอ เพราะไม่มีโปรแกรมหน่วยความจำภายใน หากใช้เบอร์ 8051/8052 ซึ่งมีหน่วยความจำภายในและให้ขา $\overline{\text{EA}}$ เป็น " 0 " ซึ่งหยุดการทำงานของรอมภายในและอ่านโปรแกรมจากอีพรอมภายนอกแทน

8. RST (Reset)

ขา RST ได้แก่ขา 9 เป็นขารีเซ็ต MCS-51 โดยให้เป็นสถานะลอจิก "1" อย่างน้อย 2 Machine Cycles จึงสามารถรีเซ็ตระบบได้

ความถี่สัญญาณนาฬิกาบนชิพ (ON-Chip Oscillator Inputs)

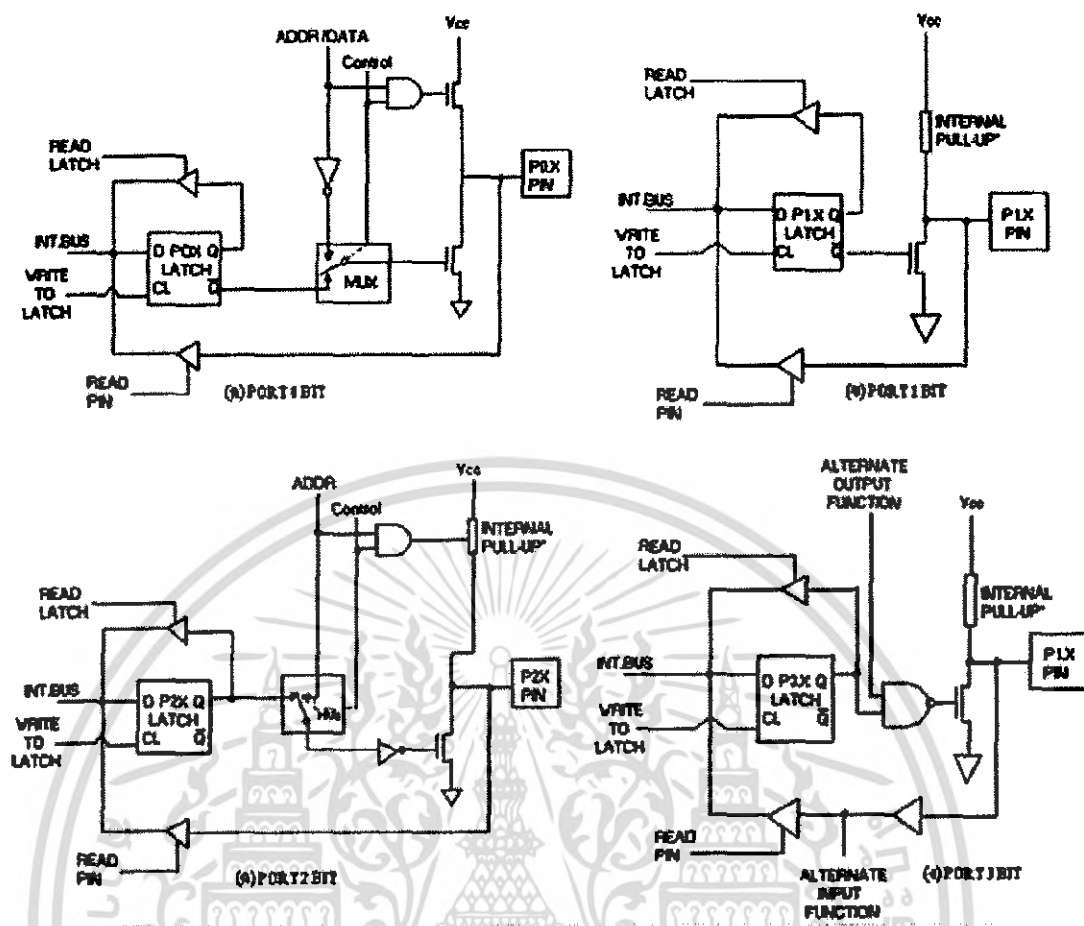
ได้แก่ขา 18-19 เป็นวงจรออสซิลเลเตอร์บนชิพ ซึ่งปกติทำการต่อคริสตัล ความถี่ 12 เมกกะเฮิร์ต กับตัวเก็บประจุ หรืออาจใช้สัญญาณนาฬิกาจากต้นกำเนิดชนิด ที่ที่แอลต์กับ XTAL1 และ XTAL2 ดังรูปที่ 2-9



รูปที่ 2-9 ขาของ MCS-51 ที่ใช้ต่อกับ XTAL

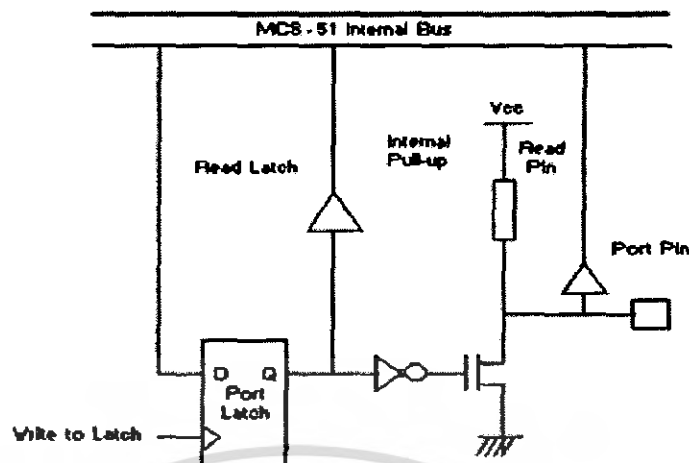
Power Connections ใน MCS - 51 นำแหล่งจ่ายไฟ 5 โวลต์ ต่อเข้ากับ Vcc (ขา 40) ส่วนขา Vss (ขา 20) ต่อลง กราวด์

2.3.7 โครงสร้างของพอร์ตอินพุตและเอาต์พุต (I/O Port Structure) ขาของพอร์ตแสดงโครงสร้างภายในได้ ดังรูปที่ 2-10 โดยมีโครงสร้างเป็น Field-Effect Transistor ต่ออยู่กับภายนอก และมีความต้านทานค่อ Pull-Up อยู่สำหรับพอร์ต 1, 2, 3 เมื่อเป็นพอร์ต 0 ไม่มีตัวต้านทาน Pull-Up ภายใน เพราะต้องใช้เป็นขาบั๊ตตำแหน่ง และบั๊ตข้อมูล



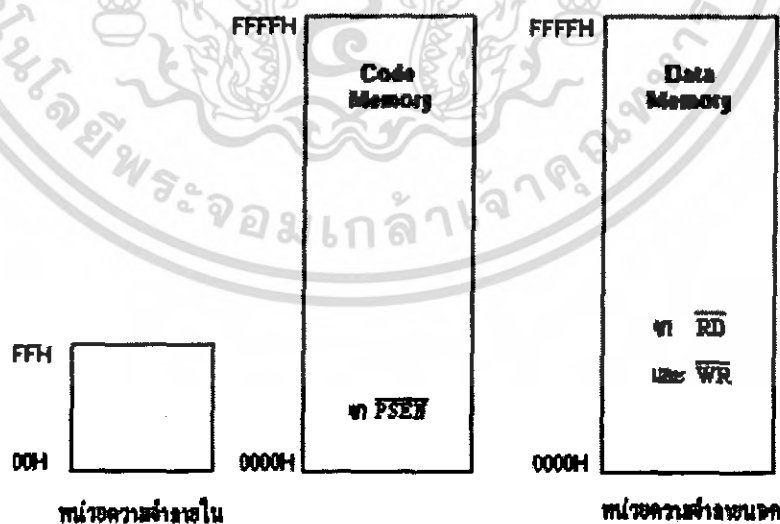
รูปที่ 2-10 โครงสร้างพอร์ตทั้ง 4 ของ MCS-51

พอร์ตนี้สามารถเป็นอินพุตและเอาต์พุตกับอุปกรณ์ภายนอกได้ ในการอ่านข้อมูลจากพอร์ตอ่านได้ 2 แบบ ได้แก่ Read Latch และ Read Pin โดย Read Latch หมายถึงการอ่านข้อมูลที่ถูกลatch เอาไว้เข้าสู่บัสภายในของ MCS-51 เช่นการทำคำสั่ง CPL P1.5 ถ้าเป็นการ Read Pin ซึ่งเป็นการใช้พอร์ตเป็นอินพุต โดยอ่านค่าจากขาของไอซีเข้าสู่บัสภายใน โดยการอ่านแบบ Read Latch และ Read Pin มีสัญญาณควบคุมที่บัพเฟอร์ดังรูปที่ 2-11



รูปที่ 2-11 การต่อพอร์ทเข้ากับระบบบัสภายในของ MCS-51

2.3.8 โครงสร้างหน่วยความจำ หน่วยความจำสำหรับ MCS-51 มี 2 ชนิดได้แก่ หน่วยความจำที่ใช้เก็บโปรแกรมรอม กับหน่วยความจำที่ใช้เก็บข้อมูล ในการประมวลผลรวม MCS-51 บางเบอร์ เช่น 8051, 8052 มีหน่วยความจำภายในชิพ และ MCS-51 ทุกเบอร์สามารถอ้างหน่วยความจำ โปรแกรมภายนอกได้มากที่สุด 64 กิโลไบต์ สำหรับหน่วยความจำแรมภายในประกอบไปด้วยพื้นที่ใช้งานทั่วไป, รีจิสเตอร์ชุด, พื้นที่ใช้งานระดับบิต และรีจิสเตอร์ฟังก์ชันพิเศษ ไดอะแกรมของหน่วยความจำของ 8031 แสดงได้ดังรูปที่ 2-12



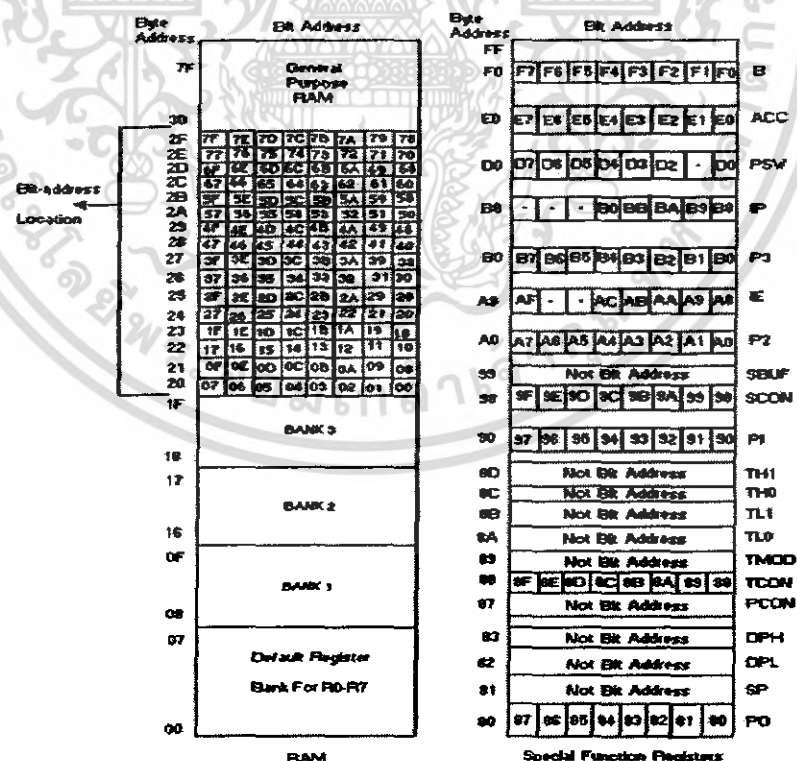
รูปที่ 2-12 การจัดหน่วยความจำของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน 8031 มีหน่วยความจำภายใน ตั้งแต่ตำแหน่ง 00H ถึง FFH และสามารถอ้างหน่วยความจำโปรแกรมภายนอกได้ 64 กิโลไบต์ ตำแหน่งที่อ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก $\overline{PSEN}$ ทำการเอกทึฟ ซึ่ง 8031 สามารถอ้างหน่วยความจำภายนอกได้ 64 กิโลไบต์ ตำแหน่งโดยการติดต่อกับหน่วยความจำนี้ $\overline{RD}$ และ $\overline{WR}$ ทำการเอกทึฟ สำหรับหน่วยความจำข้อมูลภายในสามารถแบ่งได้ดังนี้

1. ชุดรีจิสเตอร์ 4 ชุดแต่ละชุดเรียกว่ารีจิสเตอร์ชุด ที่ตำแหน่ง 00H ถึง 1FH โดย แต่ละชุดประกอบด้วยรีจิสเตอร์ R0 ถึง R7
2. หน่วยความจำที่สามารถเข้าถึงข้อมูลระดับบิตได้ ตำแหน่ง 20H ถึง 2FH
3. หน่วยความจำใช้งานทั่วไปตำแหน่ง 30H ถึง 7FH
4. รีจิสเตอร์ฟังก์ชันพิเศษ ตำแหน่ง 80H ถึง FFH

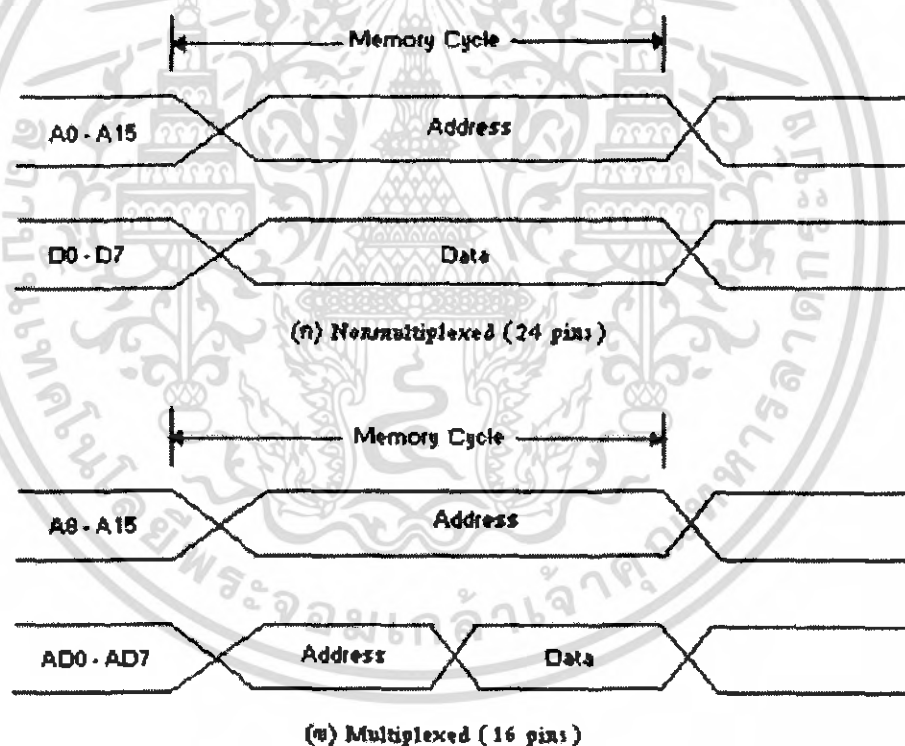
แผนผังการจัดการหน่วยความจำข้อมูลภายใน แสดงได้ดังรูปที่ 2-13 จากแผนผังแสดงถึงการอ้างตำแหน่ง หน่วยความจำภายในซึ่งอ้างได้ 2 แบบ ได้แก่ การอ้างไปที่ตำแหน่งของไบต์ (เขียนหมายเลขด้านนอก) หรือการอ้างไปที่ตำแหน่งของบิต (เขียนหมายเลขตำแหน่งด้านใน) โดยตำแหน่งของหน่วยความจำ ที่อ้างเป็นแบบบิตที่แน่นอน



รูปที่ 2-13 ตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต

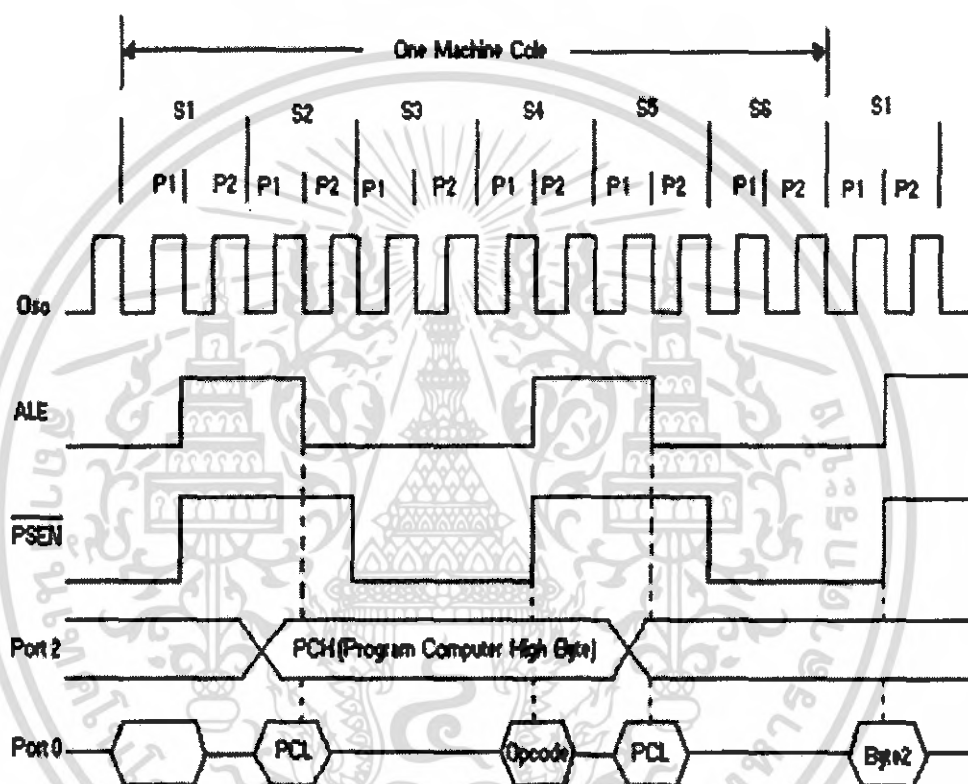
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.9 หน่วยความจำภายนอก MCS-51 สามารถอ้างหน่วยความจำข้อมูลภายนอกได้ 64 กิโลไบต์ MCS-51 ใช้พอร์ท 0 ในการอ้างตำแหน่งหน่วยความจำ 8 บิตล่าง และใช้พอร์ท 2 เป็นขาอ้างตำแหน่ง 8 บิตบน (รวมขาอ้างตำแหน่ง 16 เส้น ซึ่งอ้างได้ 64 กิโลไบต์) นอกจากพอร์ท 0 ใช้งาน 2 หน้า ที่ในการติดต่อกับหน่วยความจำ ใช้วิธีมัลติเพล็กซ์ระหว่างตำแหน่งกับข้อมูล ถ้าต้องการติดต่อกับหน่วยความจำที่เก็บข้อมูล 8 บิต และเก็บได้ 64 กิโลไบต์ ต้องใช้สายสัญญาณ 24 เส้นเป็นขาแอดเดรส 16 เส้น และขาข้อมูล 8 เส้น ดังรูปที่ 19 ถ้าใช้วิธีมัลติเพล็กซ์ ใช้ขา A0 - A7 เป็นขาข้อมูล D0 - D7 ใช้สายสัญญาณเพียง 16 เส้น รูปที่ 2-14 เมื่อต้องการติดต่อกับหน่วยความจำ ทำการส่งสัญญาณแอดเดรส A0 - A15 ออก 16 เส้น จากนั้นขา A0-A7 เปลี่ยนเป็น D0 - D7 ในการติดต่อกับหน่วยความจำภายนอก MCS - 51 ใช้วิธีนี้



รูปที่ 2-14 ไคอะแกรมกลุ่มสัญญาณที่ใช้อ่านข้อมูล

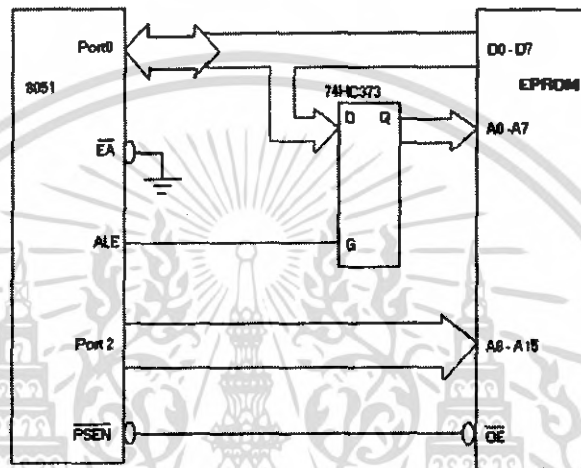
2.3.10 การติดต่อกับหน่วยความจำโปรแกรมภายนอก การอ่านข้อมูลทีหน่วยความจำโปรแกรมภายนอก MCS - 51 ส่งค่าตำแหน่งหน่วยความจำออกไป ซึ่งค่าตำแหน่งเก็บอยู่ใน PC ส่งออกทางพอร์ท 0 และพอร์ท 2 จากนั้นส่งขา ALE เป็นลอจิก "0" เพื่อ Latch ขาแอดเดรส ของ 8 บิตต่ำคือพอร์ท 0 และส่งสัญญาณทางขา PSEN เป็นลอจิก " 0 " เพื่ออ่านข้อมูลซึ่งได้ Opcode เข้าไปทางขาบั๊สข้อมูล คือพอร์ท 0 ไคอะแกรมเวลาการอ่านข้อมูลจากหน่วยความจำภายนอก แสดงได้ดังรูปที่ 2-15



รูปที่ 2-15 ไคอะแกรมการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก

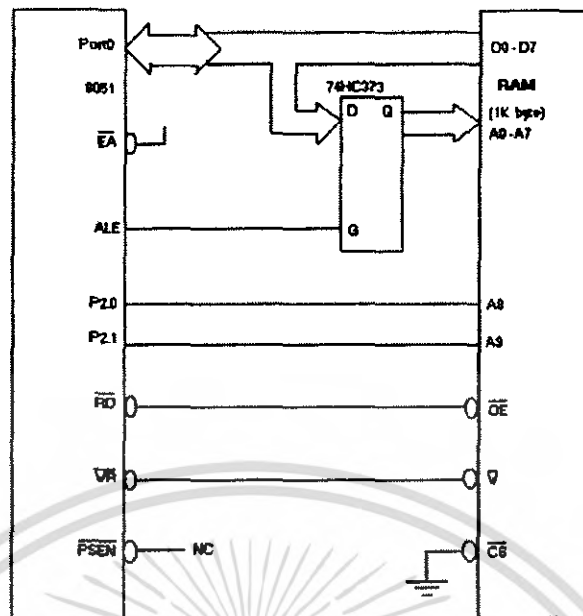
รูปที่ 20 ช่วงเวลาการทำงานของ MCS - 51 ใน S2 MCS - 51 ส่งค่าตำแหน่งหน่วยความจำโปรแกรม (ค่า PC) ทางพอร์ต 0 และ พอร์ท 2 จากนั้นส่งขา ALE เป็น " 0 " เพื่อ Latch อุปกรณ์ภายนอกค่าตำแหน่งไบต์ต่ำไว้ (พอร์ต 0) เพื่อใช้พอร์ต 0 เป็นขาข้อมูลต่อไป จากนั้นส่งขา PSEN เป็น " 0 " เพื่ออ่าน Opcode เข้าทางพอร์ต 0

การต่อหน่วยความจำกับ MCS - 51 แสดงได้ดังรูปที่ 21 โดยขา $\overline{EA}$ ต่อเป็น "0" เพื่อให้ MCS - 51 อ่านหน่วยความจำโปรแกรมภายนอกสำหรับการ มัลติเพล็กซ์ ใช้ฟลิปฟล็อป 8 ตัวเบอร์ 74373 เก็บค่าตำแหน่ง 8 บิตค่าไว้ เมื่อ MCS - 51 ส่งค่าตำแหน่งพอร์ตออกไปจากนั้นส่งขา ALE ให้เป็น "0" ซึ่งใช้ขานี้ต่อกับ 74373 เพื่อ Latch ข้อมูลสำหรับขา $\overline{PSEN}$ ต่อกับขา Output Enable ($\overline{OE}$) ของหน่วยความจำดังรูปที่ 2-16



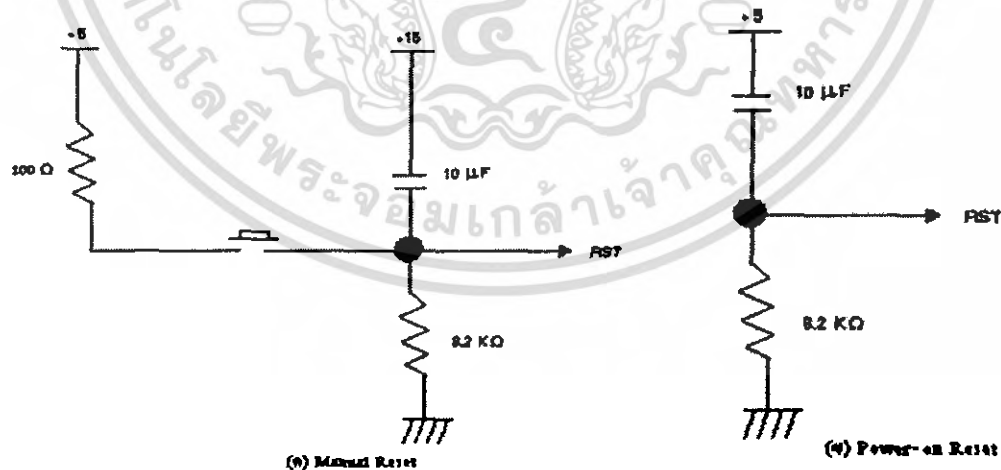
รูปที่ 2-16 การต่อ MCS - 51 กับหน่วยความจำโปรแกรมภายนอก

2.3.11 การติดต่อกับหน่วยความจำข้อมูลภายนอก หน่วยความจำข้อมูลภายนอก MCS - 51 สามารถอ่าน และ เขียนได้ ในการติดต่อกับหน่วยความจำข้อมูลภายนอก MCS - 51 ส่งขาแอดเดรสออกทางพอร์ต 0 และ พอร์ต 2 จากนั้นส่งขา ALE เพื่อไป Latch Address 8 บิตค่า โดยการอ่านเขียนข้อมูลนั้นใช้ขา $\overline{RD}$ หรือ P3.7 และขา $\overline{WR}$ หรือ P3.6 ตามลำดับ ไคอะแกรมเวลาการอ่านและเขียนข้อมูลกับหน่วยความจำภายนอกแสดงดังรูปที่ 2-17



รูปที่ 2-18 การต่อหน่วยความจำโปรแกรมกับ MCS-51

Reset Operation การรีเซ็ตหรือเริ่มต้นทำงานใหม่ของ MCS-51 ให้ลอจิก "1" ที่ขา RST เป็นเวลา 2 Machine Cycles (1 Machine Cycle เท่ากับ 12 Clock) จากนั้นให้กลับเป็นลอจิก "0" การรีเซ็ตอาจทำได้โดยใช้สวิทช์กด ดังรูปที่ 24 (ก) หรือใช้วิธี Power-up โดยใช้ตัว R-C ต่อเป็นวงจรดังรูปที่ 2-19



รูปที่ 2-19 การรีเซ็ต MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.12 การใช้งานพอร์ตสื่อสารอนุกรม พอร์ตสื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่าฟูลดูเพล็กซ์ (Full Duplex) สามารถรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน Serial Port Buffer (SBUF) ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรมโดยมีอยู่ 2 ตัวพอร์ตสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่ละบิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานของทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรม มีดังตารางที่ 2-3

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift Register ความเร็วในการรับหรือส่งข้อมูลเท่ากับ (1/2) ของ CPU Osc
0	1	1	8 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดได้จาก Timer 1,2
1	0	2	9 Bit UART ความเร็วในการรับหรือส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU Osc โดยขึ้นกับบิต SMOD ใน PCON
1	1	3	9 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดที่ Timer 1,2

ตารางที่ 2-3 โหมดการทำงานของพอร์ตสื่อสารอนุกรม

โหมด 0 : พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งข้อมูลเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อน ทางขา RxD เนื่องจากไม่มีการส่งบิตเริ่มต้น แต่จะส่ง Shift Clock ทางขา TxD (ความเร็ว 1/12 เท่าของ CPU Osc)

โหมด 1 : พอร์ตสื่อสารข้อมูลอนุกรม 10 บิต ข้อมูล 8 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โวลท์ของ ไทมเมอร์ 1,2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{\text{SMOD}} \times \text{CPU Osc}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ ไทเมอร์ 1} \quad (1)$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPU Osc}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ไทเมอร์ 2} \quad (2)$$

โหมด 2 : พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด (TB8 นิยม

นำมาใช้ ส่งพาริตีบิต) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 หรือ 1/64 เท่าของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU Osc โดยขึ้นกับบิต SMOD ใน PCON

$$\text{บอดเรท (โหมด 2)} = \frac{(2^{\text{SMOD}}) \text{CPU Osc}}{64} \quad (3)$$

- บอดเรท (โหมด 2) = 1/32 CPU Osc เมื่อ SMOD = 1
- บอดเรท (โหมด 2) = 1/64 CPU Osc เมื่อ SMOD = 0

โหมด 3 : พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ ไทเมอร์ 2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{\text{SMOD}} \times \text{CPU Osc}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ ไทเมอร์ 1} \quad (4)$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPU Osc}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ ไทเมอร์ 2} \quad (5)$$

2.3.13 กระบวนการรับและส่งข้อมูลอนุกรมของ MCS-51 การส่งข้อมูลออกทางพอร์ตอนุกรมของ MCS-51 จะเริ่มต้นขึ้นภายหลังเมื่อมีการเขียนข้อมูลลงใน SBUF ข้อมูลนี้ จะถูกเลื่อนทีละบิต และส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ส่งออกไปครบถ้วนแล้วจะทำให้ค่าแฟล็กซ์ T1 เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้ SBUF ว่างและพร้อมที่จะส่งข้อมูลไปต่อไปแล้ว ในกรณีที่ผู้ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟล็กซ์ T1 มีค่าเป็น 1 ก่อน จะมีผลทำให้ข้อมูลที่ส่งไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่า REN (Receive Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีข้อมูลภายนอกถูกส่งเข้ามายัง 8051 ทีละบิตจนครบ และเมื่อบิตสุดท้ายเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายมาเก็บไว้ยังรีจิสเตอร์ SBUF และแฟล็กซ์ RI ก็จะมีค่าเป็น 1 (ถูกเซต)

1. พอร์ตอนุกรม (โหมด 0)

การทำงานของพอร์ตอนุกรม (โหมด 0) เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิต โดยใช้เพียงขาสัญญาณ RxD เท่านั้น (ขาที่ใช้งาน 2 หน้าที่ใช้ส่งและรับข้อมูล) ส่วนขาสัญญาณ TxD จะนำไปใช้เพื่อเป็นขาสัญญาณนาฬิกาในการให้จังหวะ การเลื่อนข้อมูลกับวงจร เลื่อนบิตภายนอก สำหรับอัตราเร็วจะถูกกำหนดไว้คงที่ที่ค่า $1/12$ เท่าของ CPU Osc จากรูปที่ 25 แสดงให้เห็นถึงแผนภาพเวลาสัญญาณต่างๆในโหมด 0 เมื่อมีการรับหรือส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดภายในตัว 8051 เอง เนื่องจากโหมดนี้ไม่มีการส่งบิตเริ่มต้นและบิตสิ้นสุด ดังนั้นจึงจำเป็นที่จะต้องส่งสัญญาณ Shift clock ออกไป เพื่อใช้ Synchronize ระหว่างฝ่ายรับและฝ่ายส่ง โดยจะใช้ขา TxD ส่วนการรับข้อมูลจะรับข้อมูลเข้าทางขา RxD และรับ Shift clock เข้าทางขา TxD ถ้า CPU Osc มีค่าเท่ากับ 12 MHz ก็จะสามารถส่งได้ถึง 1 ล้านบิตต่อวินาที ซึ่งโหมด 0 เป็นโหมดที่ส่งข้อมูลได้เร็วที่สุด รายละเอียดผังเวลาในการรับส่งดังแสดงในรูปที่ 2-20

2. พอร์ตอนุกรม (โหมด 1)

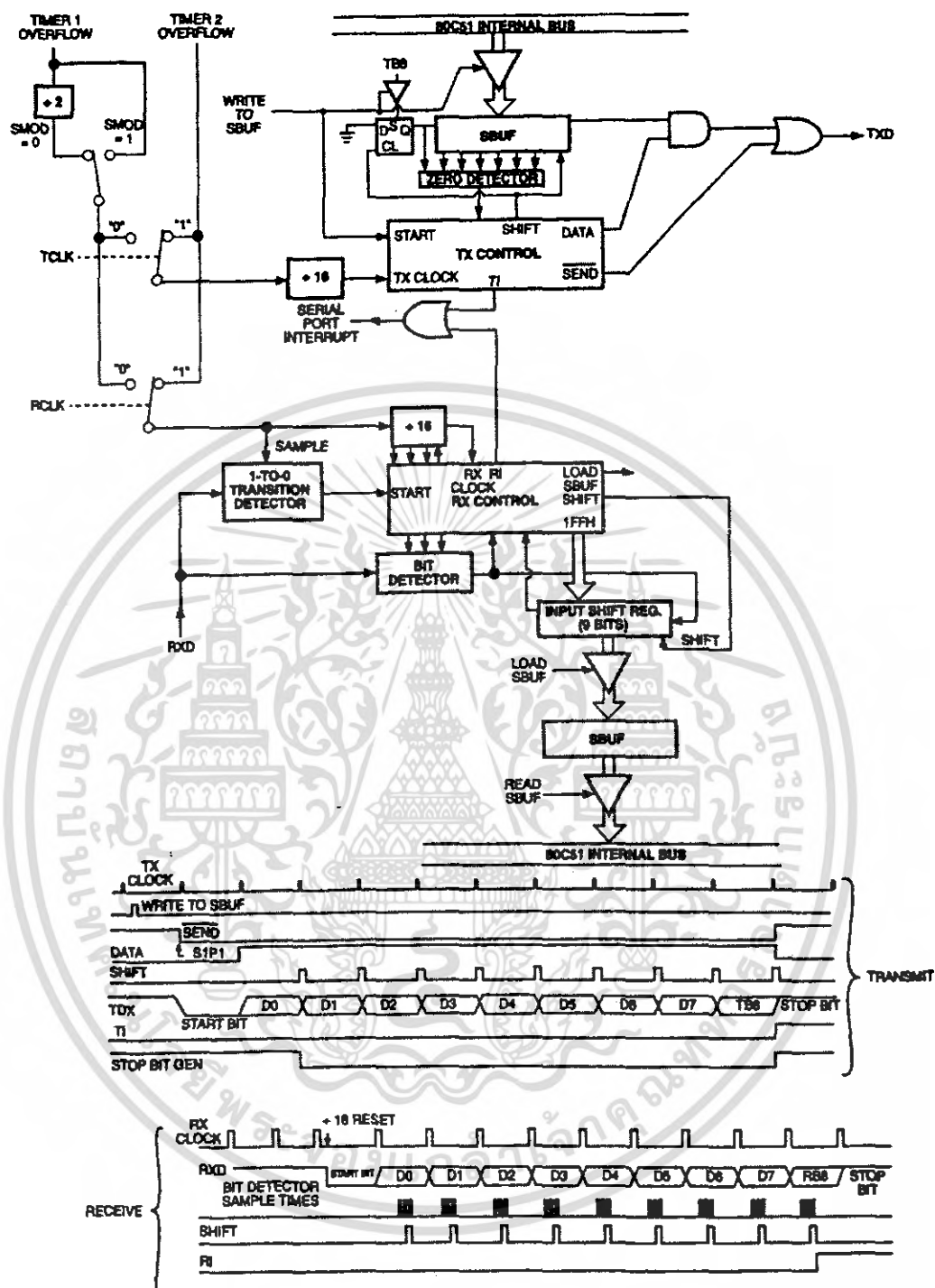
การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต ข้อมูลจำนวน 8 บิต และบิตสิ้นสุดอีก 1 บิต ดังแสดงในรูปที่ 26 โดยข้อมูลจะถูกส่งออกจาก TxD และรับเข้ามาทางขาสัญญาณ RxD ในส่วนของข้อมูล 8 บิต ที่ได้รับหรือทำการส่งออก จะเป็นบิตนัยสำคัญต่ำเป็นลำดับแรก ทางฝ่ายรับค่าของบิตสิ้นสุดจะส่งเข้ามาจัดเก็บไว้ในบิต RB8 ภายในรีจิสเตอร์ SCON สำหรับอัตราเร็วในการส่งข้อมูลของโหมด 1 นั้น สามารถเลือกได้จากไทมเมอร์ 1 ผังเวลาการทำงานแสดงดังในรูปที่ 2-21

3. พอร์ตอนุกรม (โหมด 2)

โหมด 2 ใช้ทั้งหมด 11 บิต โดยแบ่งเป็น บิตเริ่มต้น 9 บิตข้อมูล และบิตสิ้นสุด โดยบิตที่ 9 ผู้ใช้สามารถกำหนดค่าเองได้ว่าจะส่งค่าอะไรออกไป โดยจะต้องนำไปใส่ไว้ในบิต TB8 ในรีจิสเตอร์ SCON ส่วนมากผู้ใช้นักจะนำบิตนี้มาใช้เป็นพาริตีบิต โดยรับค่ามาจากพาริตีแฟล็กซ์ใน PSW ส่วนทางด้านรับ บิตที่ 9 จะถูกนำไปเก็บไว้ใน RB8 อัตราเร็วในการส่ง/รับข้อมูลกับ CPU Osc และค่า SMODซึ่งอยู่ในบิต 7 ใน SCON ผังเวลาการทำงานแสดงดังในรูปที่ 2-22

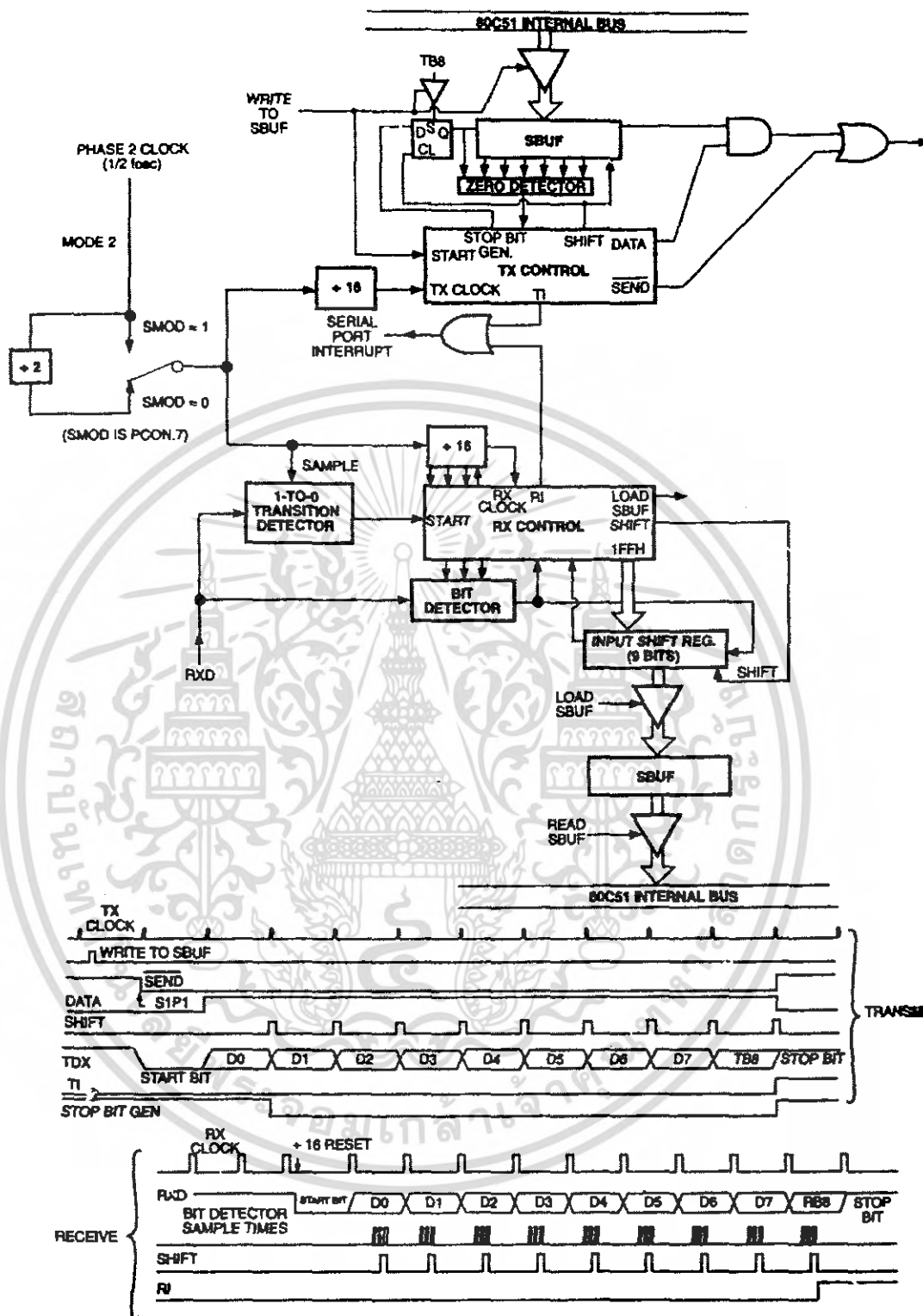
4. พอร์ตอนุกรม (โหมด 3)

การทำงานเหมือนกับโหมด 2 ทุกอย่างจะแตกต่างกันที่ความเร็วในการรับส่งจะขึ้นอยู่กับอัตราโอเวอร์โพล์ของไทมเมอร์ 1 หรือ ไทมเมอร์ 2 โดยมีผังการทำงานแสดงดังในรูปที่ 2-23

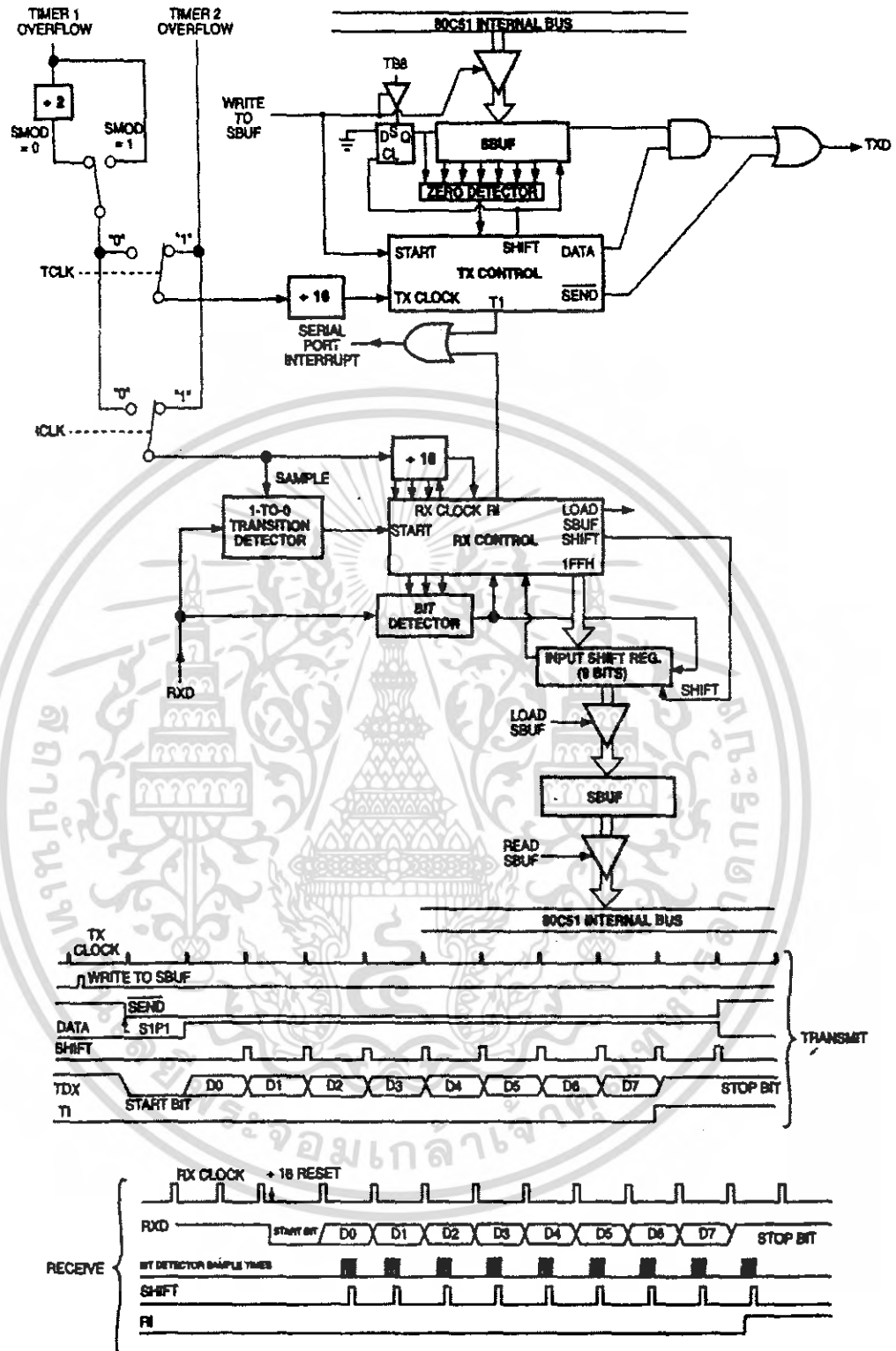


รูปที่ 2-21 ฝั่งการทำงานโมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-22 ผังการทำงาน โหมด 2



รูปที่ 2-23 ฟังก์การทำงาน โมด 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสื่อสารแบบอนุกรม

ถึงแม้ว่าการสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้ก็เพราะว่าการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตนานนั้นสามารถส่งข้อมูลได้ครั้งละหลายๆบิตพร้อมกันส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน แต่การส่งข้อมูลแบบอนุกรมนั้นมีข้อที่เหนือกว่าการส่งข้อมูลแบบขนานคือ การสามารถส่งข้อมูลได้ในระยะทางที่ไกลกว่าแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนาน การสื่อสารแบบอนุกรมสามารถแบ่งออกเป็น 3 รูปแบบ ดังนี้

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

นอกจากนี้แล้วยังสามารถแบ่งประเภทของการสื่อสารแบบอนุกรมตามลักษณะสัญญาณในการส่งได้อีก 2 แบบ

1. การสื่อสารแบบซิงโครนัส สำหรับการสื่อสารแบบซิงโครนัสนี้จะใช้สัญญาณนาฬิกาควบคุมการรับส่งสัญญาณ เช่น สายเคเบิลคอมพิวเตอร์ ส่วนอีกเส้นหนึ่งเป็นสายของข้อมูล (และมักจะมีสายกราวด์) สำหรับการสื่อสารแบบซิงโครนัสนี้เหมาะสำหรับการทำงานในระยะใกล้ ข้อมูลที่จะส่งมีไม่มากนัก เพราะถ้าระยะทางไกลขึ้นจะทำให้สัญญาณนาฬิกามีปัญหา อีกทั้งต้องมีสายหลายเส้นทำให้สิ้นเปลืองมาก

2. การสื่อสารแบบอะซิงโครนัส สำหรับการสื่อสารแบบอะซิงโครนัสนั้นจะใช้สายข้อมูลเพียงตัวเดียว แต่จะใช้รูปแบบการส่งข้อมูล หรือ bit Pattern เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นส่วนตรวจสอบความถูกต้องข้อมูล และส่วนไหนเป็นส่วนปิดท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาทำงานทั้งภาคส่งและภาครับ

2.4.1 มาตรฐาน RS-232C มาตรฐาน RS-232C เป็นมาตรฐานที่ได้รับการออกแบบมาเพื่อที่จะทำให้อุปกรณ์ต่อพ่วงจากผู้ผลิตต่างกันสามารถทำงานร่วมกันได้ มาตรฐานหลายชนิดได้รับการออกแบบขึ้นมา แต่มาตรฐานที่ได้รับความนิยมและใช้กันกว้างขวางมากที่สุดคือ มาตรฐาน RS-232C ซึ่งถูกประกาศใช้ในปี 1969 โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ ในยุคแรกๆ การอินเตอร์เฟสแบบ RS-232C ถูกออกแบบสำหรับเชื่อมต่อเทอร์มินอล (DTE) กับ โมเด็ม (DCE) ทั้งนี้ก็เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลบนสายเส้นเดียวกัน

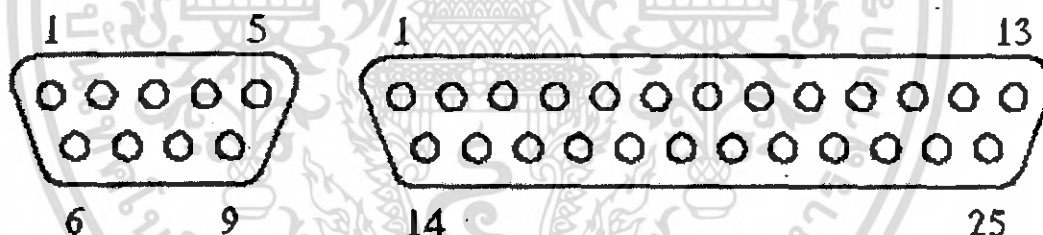
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน RS-232C ได้แบ่งอุปกรณ์ออกเป็น 2 ประเภท ซึ่งอุปกรณ์ทั้งสองประเภทนี้ คือ

1. อุปกรณ์ DTE (Data Terminal Equipment) เป็นอุปกรณ์สำหรับส่งข้อมูล (เอาต์พุต)
2. อุปกรณ์ DCE (Data Communication Equipment) เป็นอุปกรณ์สำหรับรับข้อมูล (อินพุต)

ตามมาตรฐาน RS-232C แล้วคอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งคอนเน็กเตอร์ที่นิยมใช้จะเป็นชนิด D-type แบบ 9 ขา และแบบ 25 ขา สัญญาณไฟฟ้าของ RS-232C กำหนดว่าถ้าเป็นลบมากกว่า -3 จะเป็น 1 และถ้าเป็น +4V ขึ้นไปจะเป็น 0 ความเร็วสูงสุดของข้อมูลสูงสุดได้ถึง 20 Kbps สำหรับลอจิก 1 และลอจิก 0 จะมีระดับแรงดันตามที่กล่าวมาจะสามารถรับส่งข้อมูลได้ด้วยความยาวของสายสัญญาณสูงสุด 50 ฟุต หรือ 150 เมตร แต่ถ้าเราต้องการสื่อสารกับอุปกรณ์อื่นที่อยู่ห่างกันมากๆ จึงจำเป็นต้องใช้อุปกรณ์อื่นๆ เข้าช่วย เช่น การใช้โมเด็ม เป็นต้น

2.4.2 ลักษณะของคอนเน็กเตอร์แบบ D-Type หัวต่อแบบ D-Type ที่ใช้ในการสื่อสารแบบอนุกรมของเครื่องคอมพิวเตอร์นั้น จะมีอยู่ 2 ลักษณะ คือ แบบ 9 ขา และแบบ 25 ขา บางครั้งเราจะเรียกว่า DB9 และ DB25 ซึ่งหัวต่อทั้งสองชนิดจะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน ดังภาพที่ 2-24 และตารางที่ 2-4



ภาพที่ 2-24 แผนผังคอนเน็กเตอร์ของ RS-232

D-Type 25 Pin	D-Type 9 Pin	สัญลักษณ์	ชื่อสัญญาณ
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

ตารางที่ 2-4 แสดงฟังก์ชันของ RS – 232

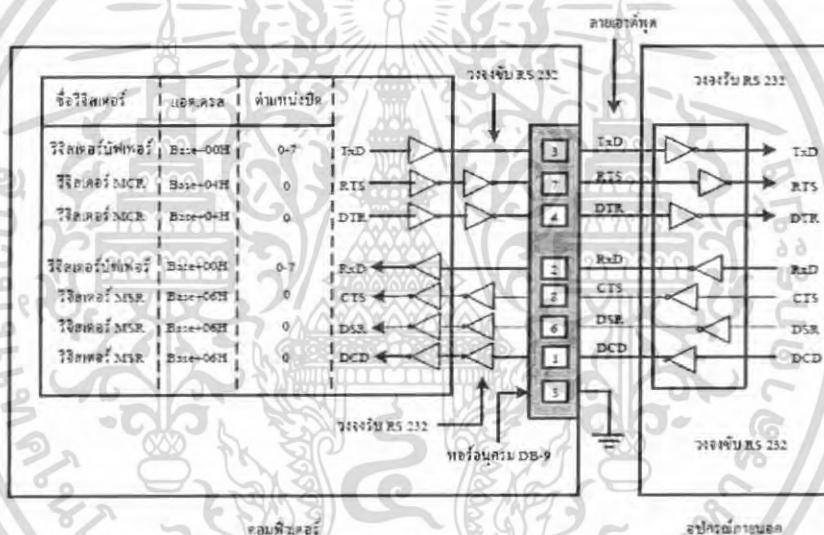
2.4.3 รายละเอียดของสัญญาณสำหรับรายละเอียดของสัญญาณนั้นประกอบไปด้วย

1. Transmit Data: TD ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์
2. Receive Data: RD ใช้สำหรับรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์
3. Request To Send: RTS ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทางเพื่อร้องขอให้
อุปกรณ์ปลายทางส่งข้อมูลกลับมา
4. Clear To Send: CTS ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อด้วยพร้อมที่จะรับข้อมูลหรือไม่ โดยจะคอยรับสัญญาณ RTS เมื่อทุกขาพร้อมก็จะทำการส่งข้อมูลออกทางขา TD
5. Data Set Ready: DSR ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับ
อุปกรณ์ปลายทาง จะใช้คู่กับขา DTR
6. Signal Ground: SG เป็นกราวด์ของระบบ
7. Carrier Detect: CD ขานี้จะ Active เมื่อมีการส่งสัญญาณ Carrier จากโมเด็ม
8. Data Terminal Ready: DRT ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าการติดต่อ
ด้วยโดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง
9. Ring Indicator: RI ขานี้จะ Active เมื่อ โมเด็ม ได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่จะใช้ควบคุม (RTS และ DTR) และจะมีสัญญาณแสดงสถานะทางอินพุต (CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่ง และรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณที่เอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกันเพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะดังแสดงเป็นบล็อกไดอะแกรมในภาพที่ 2-11



รูปที่ 2-25 ไดอะแกรมโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรม มี 4 ตำแหน่งดังนี้ คือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไปออสภายในคอมพิวเตอร์ จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไปออสตรวจพบแอดเดรสของพอร์ตที่อนุกรมไปออสจะนำแอดเดรสที่ตรวจพบ ไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1 จะเก็บไว้แอดเดรส 0000:0400H และ 0000:0401H ส่วนที่ตำแหน่งอื่นๆมีรายละเอียดดังตารางที่ 2-3

COM2 = 0000:0402H - 0000:0403H

COM3 = 0000:0404H - 0000:0405H

COM4 = 0000:0406H - 0000:0407H

นอกจากนี้ ที่หน่วยความจำแอดเดรส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วย โดยมีรายละเอียด ดังแสดงในตารางที่ 2-5

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

ตารางที่ 2-5 แสดงข้อมูลในแอดเดรส 0000:0411H ที่ใช้ในพอร์ตอนุกรม

2.4.5 อัตราเร็วในการรับส่งข้อมูลแบบอนุกรม การที่อุปกรณ์ 2 อย่างจะติดต่อสื่อสารกันได้นั้น จะต้องทำงานด้วยอัตราเร็วเท่ากันซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสคือ ค่าบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที ซึ่งอัตราเร็วในการสื่อสารแบบอนุกรมสำหรับมาตรฐาน RS-232 นั้นมีใช้ดังนี้ 110 , 150 , 300 , 600 , 1200 , 2400 , 4800 , 9600 และ 1920

2.5 โปรแกรม Visual Basic

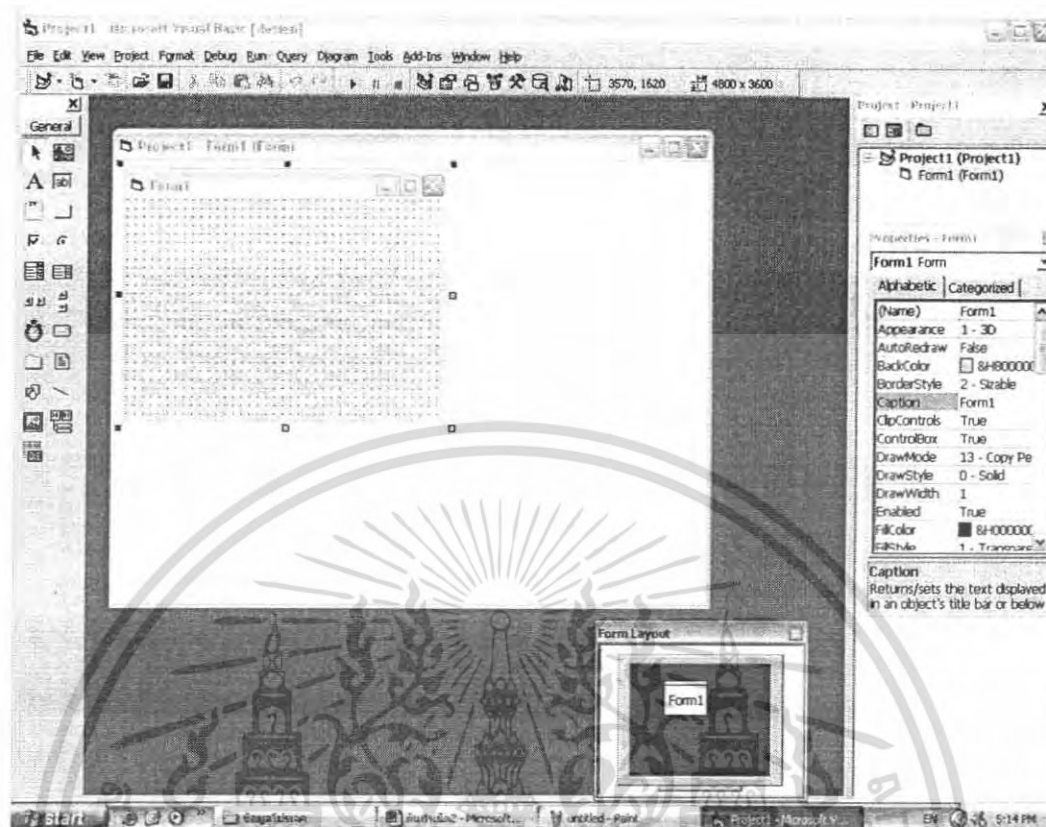
Visual Basic เป็นภาษาที่เหมาะสมสำหรับการเรียนรู้ในการเขียน โปรแกรมนั้นเนื่องจาก Visual Basic มีข้อดีหลายประการคือ

1. ง่ายต่อการเรียนรู้เหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องไวยากรณ์ของภาษาเองและเครื่องมือการใช้งาน ดังชื่อที่บอกอยู่แล้วว่า Basic ซึ่งเหมาะสำหรับผู้เริ่มต้น
2. ความนิยมของตัวภาษา โดยอาจกล่าวได้ว่าภาษา Basic นั้นเป็นภาษาที่มีคนเรียนรู้และใช้งานมากที่สุดในประวัติศาสตร์ของเครื่องคอมพิวเตอร์
3. การพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วของการประมวลผล และในเรื่องของความสามารถใหม่ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต
4. ผู้พัฒนาสำคัญของ Visual Basic คือบริษัทไมโครซอฟท์ซึ่งจัดว่าเป็นยักษ์ใหญ่ ของวงการคอมพิวเตอร์ในปัจจุบัน เราจึงสามารถมั่นใจได้ว่า Visual Basic จะยังมีการพัฒนา ปรับปรุง และคงอยู่ไปอีกนาน

2.5.1 IDE คืออะไรและส่วนประกอบต่างๆ ของ IDE คำว่า IDE หรือ Integrated Development Environment หมายถึง สภาพแวดล้อมการทำงานในการพัฒนาโปรแกรมโดยใช้ Visual Basic ดังภาพที่ 2-12 หรือจะแปลอีกอย่างก็คือ อุปกรณ์เครื่องมือต่างๆแบบเพียบพร้อมที่ไมโครซอฟท์เตรียมมาให้ใช้ในการพัฒนาโปรแกรมด้วย Visual Basic เมื่อเปิดโปรแกรม Visual Basic ครั้งแรก โปรแกรมจะปรากฏหน้าจอ IDE ซึ่งมี

2.5.1.1 ส่วนประกอบหลักของหน้าจอ Visual Basic IDE มีดังนี้

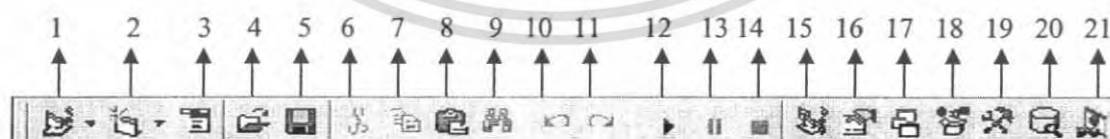
1. ทูลบาร์ (Toolbar)
2. Toolbox
3. วินโดว์ Form
4. วินโดว์ Project Explorer
5. วินโดว์ Properties
6. วินโดว์ Form Layout



รูปที่ 2-26 แสดงสภาพแวดล้อมของการพัฒนาโปรแกรม Visual Basic

2.5.1.2 ทูลบาร์ (Toolbar)

เมื่อพิจารณาจากภาพหน้าจอ IDE จะเห็นปุ่มต่าง ๆ ที่วางเรียงเป็นแถวแนวนอน ช่วยให้สามารถเรียกใช้งานคำสั่งได้อย่างสะดวกรวดเร็ว โดยเพียงแต่คลิกเมาส์ที่ปุ่มเท่านั้น รายละเอียดของปุ่มต่าง ๆ มีดังภาพที่ 2-27



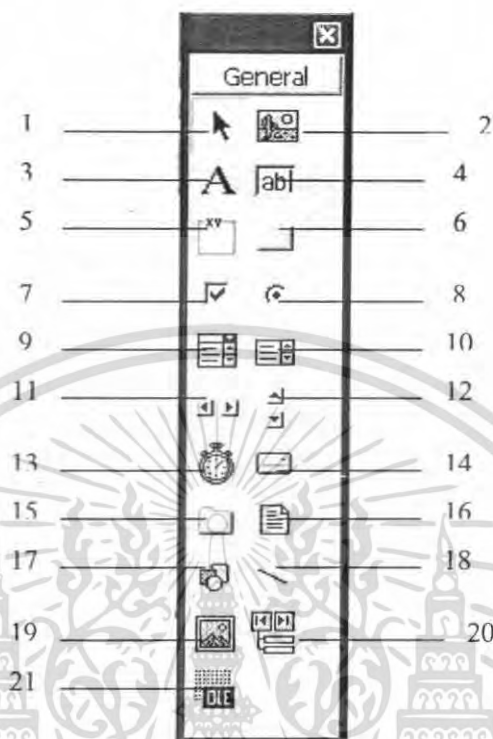
รูปที่ 2-27 แสดงการใช้งานของทูลบาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เปิดโปรเจกต์ใหม่ขึ้นมา เราสามารถพัฒนาได้หลายโปรเจกต์ไปพร้อม ๆ กัน
2. เพิ่มฟอร์ม โมดูล หรือออบเจกต์ประเภทต่าง ๆ เข้าไปในโปรเจกต์หรือโปรแกรม เปิดวินโดว์ Menu Editor ซึ่งเป็นเครื่องมือช่วยในการสร้างเมนูของโปรแกรม
3. เปิดไฟล์โปรเจกต์ (Open)
4. ตัด (Cut)
5. คัด (Copy)
6. วาง (Paste)
7. ค้นหา (Find)
8. ยกเลิกการกระทำหรือการพิมพ์ (Undo)
9. เรียกคืนกลับสิ่งที่ Undo ไป (Redo)
10. สั่งให้โปรแกรมทำงาน (Run)
11. ให้โปรแกรมหยุดการทำงานชั่วคราว (Pause)
12. ให้โปรแกรมหยุดทำงาน (Stop)
13. เปิดวินโดว์ Project Explorer ที่แสดงฟอร์ม โมดูล และส่วนประกอบต่าง ๆ ของโปรเจกต์
14. เปิดวินโดว์ Properties เพื่อดูและกำหนดคุณสมบัติต่าง ๆ ของออบเจกต์
15. เปิดวินโดว์ Form Layout เพื่อจัดตำแหน่งวินโดว์ของโปรแกรมบนจอภาพเมื่อโปรแกรมทำงาน
16. Object Browser เป็นเครื่องมือช่วยค้นหาข้อมูลรายละเอียดของออบเจกต์ต่าง ๆ
17. Toolbox เป็นที่รวมของออบเจกต์ต่าง ๆ ที่จะนำมาประกอบในโปรแกรมหรือแอปพลิเคชัน
18. เปิดวินโดว์ Data View เพื่อดูการติดต่อกับฐานข้อมูลต่าง ๆ รวมทั้งดูโครงสร้างของฐานข้อมูลที่เรา กำลังคิดอยู่ด้วย

19. Visual Component Manager ใช้ในการช่วยค้นหา เรียบเรียง ดูแล และจัดการส่วนประกอบต่าง ๆ ที่จะนำมาใช้ในการพัฒนาโปรเจกต์

2.5.1.3 ทูลบ็อกซ์ (Toolbox) เป็นที่รวมออบเจกต์ต่าง ๆ ที่จะนำมาประกอบกันเป็นโปรแกรมหรือแอปพลิเคชัน เมื่อใช้ ออบเจกต์เหล่านี้ประกอบกับจะได้เป็นหน้าต่างของโปรแกรม จึงอาจเรียกให้ชัดเจนได้ว่า Control object ซึ่งมีออบเจกต์หลักดังภาพต่อไปนี้ นอกจากนี้เราสามารถเพิ่มออบเจกต์ต่าง ๆ เข้าไปใน Toolbox ได้มากมายซึ่งจะกล่าวถึงอย่างละเอียดต่อไปภายในเล่ม สำหรับรายละเอียดคร่าว ๆ ของออบเจกต์หลักจะมีดังนี้ดังภาพที่ 2-28



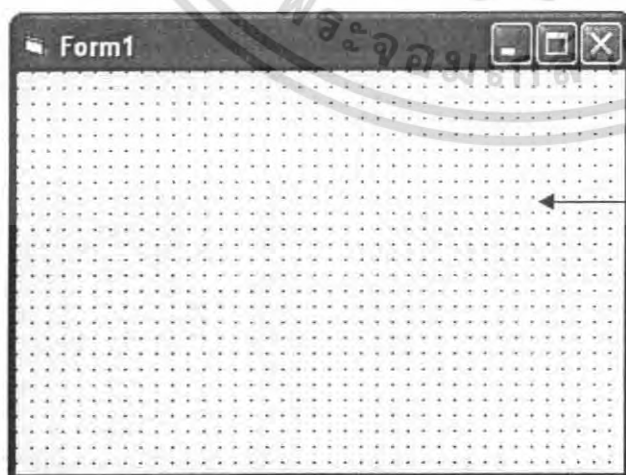
รูปที่ 2-28 แสดงการใช้งานของทูลบาร์

1. Pointer ใช้ในการจัดขนาด เคลื่อนย้าย และวางตำแหน่งออบเจ็กต์ต่าง ๆ ในฟอร์ม
2. Picture ใช้ควบคุมและแสดงข้อมูลภาพต่าง ๆ บนฟอร์ม
3. Label ใช้แสดงข้อความต่าง ๆ บนฟอร์มเหมือนกับเป็นป้ายลาเบลหรือข้อความกำกับ
4. Text Box เป็นออบเจ็กต์สำหรับรับข้อความที่ผู้ใช้ป้อนเข้ามา
5. Frame ใช้จัดกลุ่มและรวบรวมออบเจ็กต์ต่าง ๆ เข้าไว้ด้วยกัน เพื่อให้สะดวกในการควบคุมและเคลื่อนย้ายตำแหน่ง หรือจัดหน้าจอให้เป็นระเบียบเรียบร้อยและสะดวกแก่การใช้งาน
6. Command Button หรือ ปุ่มคำสั่ง เป็นออบเจ็กต์ที่เป็นปุ่มกด เพื่อให้ผู้ใช้สั่งทำงาน ซึ่งเป็นออบเจ็กต์ที่ใช้อย่างมากที่สุดอันหนึ่ง
7. Check Box เป็นปุ่มที่ใช้เลือกว่าต้องการหรือไม่
8. Option Button บางครั้งเรียกว่า Radio Button ใช้สำหรับเลือกค่าใดค่าหนึ่งจากหลาย ๆ ค่า คล้ายกับปุ่มกดเลือกกระดับความแรงของพัดลมหรือปุ่มกดในวิทยุเทป ที่เลือกได้ครั้งละ 1 ปุ่มเท่านั้น
9. Combo Box ผู้ใช้สามารถเลือกตัวเลือกได้จากการกดปุ่ม Drop down เพื่อแสดงทางเลือกต่างขึ้นมาให้มีความสามารถเหมือนกับ List Box และ Text Box ผสมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. List Box ใช้แสดงตัวเลือกต่าง ๆ ในลักษณะของบรรทัดรายการ โดยผู้ใช้สามารถเลือกรายการใดรายการหนึ่ง หรือหลาย ๆ รายการจากลิสต์รายการที่มีอยู่ก็ได้
11. Horizontal Scroll Bar เป็นแถบเลื่อนทางแนวนอน ใช้เลื่อนปรับค่าโดยค่าจะเปลี่ยนไปตามตำแหน่งที่อยู่ของแถบเลื่อน (ตำแหน่งซ้ายสุดค่าจะน้อยที่สุด ตำแหน่งขวาสุดค่าจะมากที่สุด)
12. Vertical Scroll Bar เป็นแถบเลื่อนในแนวตั้ง ใช้เลื่อนปรับค่าโดยค่าจะเปลี่ยนไปตามตำแหน่ง (ตำแหน่งบนสุดค่าจะน้อยที่สุด ตำแหน่งล่างสุดค่าจะมากที่สุด)
13. Timer ใช้ในการควบคุมเวลา และการทำงานของโปรแกรมเมื่อมีเรื่องเวลาเข้ามาเกี่ยวข้อง
- 14-16. Drive List Box, Directory List Box, File List Box ใช้ในการติดต่อกับระบบแฟ้มข้อมูลของเครื่องคอมพิวเตอร์
17. Shape ใช้สร้างภาพรูปทรงต่าง ๆ ลงในฟอร์ม
18. Line ใช้วาดเส้นต่าง ๆ ลงในฟอร์ม
19. Image Picture เป็นคอนโทรลที่ใช้ควบคุมข้อมูลภาพเหมือนกับ Picture เพียงแต่มีความสามารถน้อยกว่าแต่ก็ใช้หน่วยความจำน้อยตามลงไปด้วย
20. Data Control ใช้ในการเชื่อมต่อกับฐานข้อมูล
21. OLE (Object Linked and Embedded) เป็นคอนโทรลที่นำเอาโปรแกรมสำเร็จรูปต่าง ๆ ที่มีความสามารถ OLE เข้ามาใช้เป็นออบเจกต์ในโปรเจกต์

2.5.1.4 วินโดว์ Form เป็นวินโดว์เปล่า ๆ หรือ ตัวฟอร์มเปล่าสำหรับสร้างองค์ประกอบของแอปพลิเคชัน โดยการนำออบเจกต์ต่าง ๆ มาใส่ในฟอร์ม หรือพูดอีกนัยหนึ่งก็คือเป็นหน้าจอของโปรแกรมที่ผู้ใช้จะเห็นเมื่อเรียกใช้งานโปรแกรมนั่นเอง เมื่อเริ่มเข้าสู่ Visual Basic จะปรากฏฟอร์มเปล่าขึ้นมาให้เสมอ ดังภาพที่ 2-29



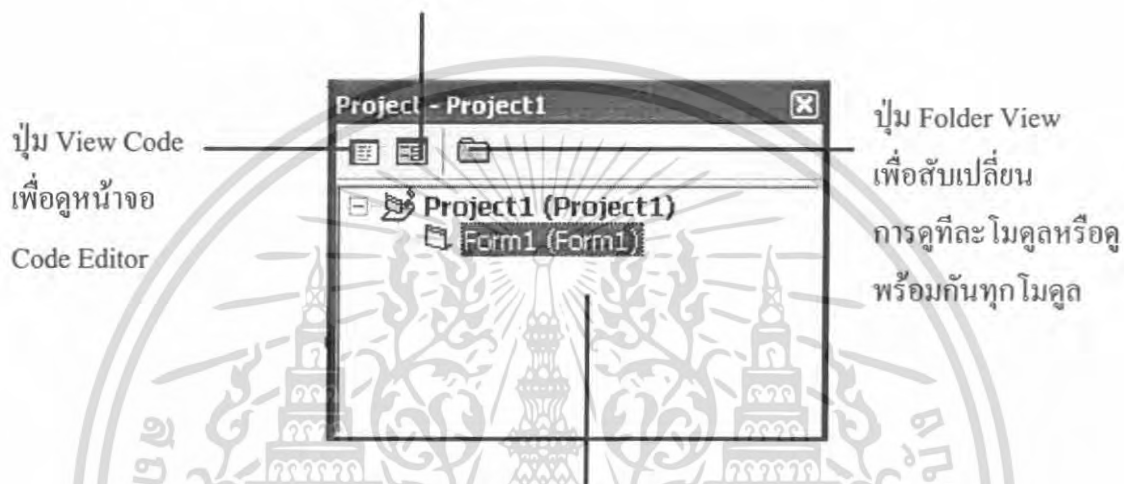
พื้นที่ว่างสำหรับออกแบบหน้าต่างโปรแกรม โดยนำออบเจกต์ต่าง ๆ จาก Toolbox เข้ามาวางในฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2-29 แสดงวินโดว์ Form1

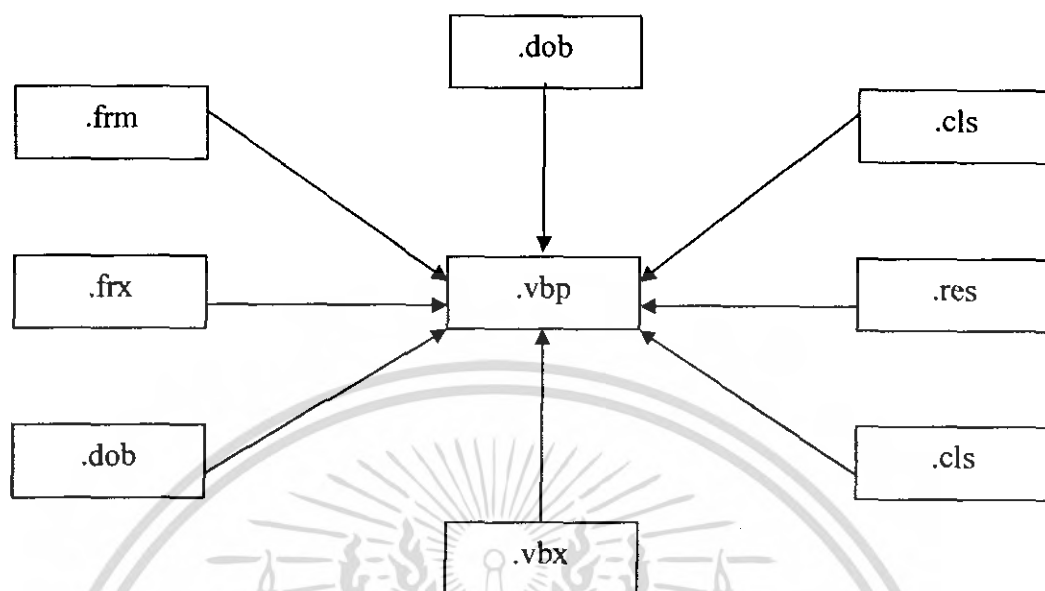
2.5.1.5 วินโดว์ Project Explorer โปรแกรมต่าง ๆ ที่เราพัฒนาเขียนโปรแกรมขึ้นมาจะเรียกว่าเป็นโปรแกรมประยุกต์ หรือ แอปพลิเคชัน (Application) ซึ่งใน Visual Basic จะเรียกโปรแกรมที่เรากำลังสร้างว่าเป็น โครงการหรือ โปรเจค ดังภาพที่ 2-30

ปุ่ม Object View เพื่อกลับไปดูที่
ฟอร์มกลับไป



รูปที่ 2-30 แสดงวินโดว์ Project Explorer

Project explorer จะใช้ควบคุมส่วนประกอบและเพิ่มข้อมูลต่าง ๆ ที่อยู่ในโปรเจค เพื่อความสะดวกในการควบคุมและเปลี่ยนการทำงานระหว่างส่วนประกอบต่าง ๆ โดยแต่ละโปรเจคจะประกอบด้วยแฟ้มข้อมูลมากมายหลายประเภท ซึ่งแฟ้มข้อมูลหลัก ๆ จะแสดงดังภาพที่ 2-31



รูปที่ 2-31 ภาพแสดงโครงสร้างแฟ้มข้อมูลส่วนต่าง ๆ ที่ประกอบขึ้นมาเป็นโปรเจกใน Visual Basic

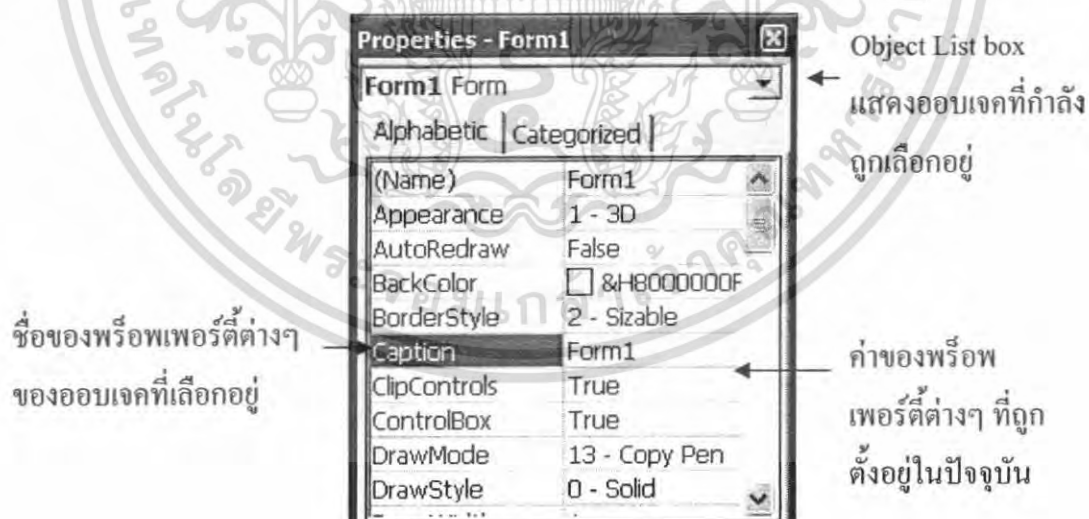
ประเภทไฟล์	รายละเอียด	นามสกุล
ไฟล์โปรเจก (Project file)	เก็บข้อมูลต่าง ๆ ของโปรเจก รวมทั้งรายชื่อแฟ้มที่ประกอบขึ้นมาเป็นโปรเจก	.vbp
ไฟล์ฟอร์ม (Form file)	เก็บฟอร์มที่เราได้ออกแบบไว้โดยในไฟล์นี้จะรวมคำสั่งต่าง ๆ ที่ เขียนโปรแกรมไว้ให้กับแต่ละออบเจกที่อยู่ในฟอร์มด้วย	.frm
ไฟล์ไบนารีฟอร์ม	จะเก็บข้อมูลที่เป็นแฟ้มไบนารีของฟอร์ม เช่น รูปภาพ หรือ ไอคอน เป็นต้น	.frx
ไฟล์โมดูลแบบปกติ (standard module)	เพื่อให้ฟอร์มหรือโมดูลอื่นสามารถเรียกใช้งานได้เก็บโปรแกรมย่อยและตัวแปลต่าง ๆ ที่เราเขียนแยกออกจากฟอร์ม	.bas
ไฟล์ Object Control	นามสกุลลงท้ายด้วย .ocx (activeX Control) หรือ .vbx เป็นพื้นฐาน ได้แก่ Internet Control Object, Database Grid Control Object เป็นต้น ออบเจกที่เราเพิ่มเข้าไปในโปรเจก นอกเหนือจากคอนโทรล	.ocx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์เอกสาร ActiveX	เว็บเบราว์เซอร์ เช่น Internet Explorer เหมือนกับฟอร์ม เพียงแต่ต้องเรียกดูผ่าน โปรแกรม	.dob
ไฟล์คลาสโมดูล (Class Module)	เก็บออกเจตต่าง ๆ ที่เราสร้างขึ้น เมื่อมีการเรียกใช้ Class Module โปรแกรมก็จะสร้างออบเจกต์นั้นขึ้นมาใหม่ (เรียกว่า Instance) แทนที่จะใช้จาก โมดูลหรือออบเจกต์นั้น โดยตรง อาจกล่าวได้ว่า Class module เปรียบเสมือนที่เก็บแผนผัง หรือ Template ของออบเจกต์ที่เราจะสร้างขึ้นมานั้นเอง	.cls
ไฟล์ทรัพยากรอื่นๆ (Resource File)	เก็บภาพ Bitmap (BMP), ข้อความ (Text string) หรือข้อมูล ใด ๆ หรือฟอร์มต่าง ๆ ในโปรเจกต์ที่เราสามารถแก้ไขได้ โดยไม่ต้องไปยังเกี่ยวกับโปรแกรมในโมดูล	.res

ตารางที่ 2-6 ตารางแสดงประเภทไฟล์ต่าง ๆ

2.5.1.6 วินโดว์ Properties

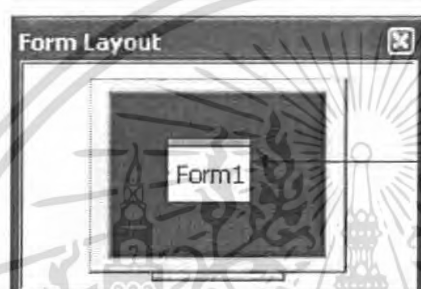


รูปที่ 2-32 แสดงวินโดว์ Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วินโดว์นี้จะแสดงคุณสมบัติทั้งหมดของออบเจกต์ที่ถูกเลือกอยู่ การคลิกเลือกที่ออบเจกต์ใดในฟอร์มจะทำให้คุณสมบัติที่แสดงในวินโดว์ Properties เปลี่ยนไปตามออบเจกต์ที่เลือก ซึ่งการแก้ไขหรือตั้งค่าคุณสมบัติสามารถทำได้โดยตรงที่คุณสมบัติแต่ละค่า สำหรับแท็บ Alphabetic และ Categorized มีไว้เพื่อช่วยให้เราหาพร็อพเพอร์ตี้ได้ง่ายขึ้นเท่านั้น โดยแท็บ Alphabetic จะแสดงคุณสมบัติเรียงตามชื่อตัวอักษร ส่วนแท็บ Categorized จะแสดงคุณสมบัติเรียงตามลักษณะการใช้งาน ดังภาพที่ 2-3

2.5.1.7 วินโดว์ Form Layout



ใช้เมาส์ลากเพื่อกำหนดตำแหน่ง
ของฟอร์มเมื่อรันโปรแกรม

รูปที่ 2-33 แสดงวินโดว์ Form Layout

จะแสดงตำแหน่งฟอร์มของโปรแกรมที่กำลังสร้างให้ดูบนจอภาพ เพื่อกำหนดตำแหน่งสำหรับตอนที่โปรแกรมทำงานจริง ๆ การย้ายตำแหน่งทำได้โดยใช้เมาส์ลาก

2.5.1.8 วินโดว์ Code Editor Code Editor เป็นเนื้อที่สำหรับเขียนโปรแกรม เรียกขึ้นมาแสดงได้โดยใช้เมนู view Code หรือดับเบิลคลิกที่ออบเจกต์ใด ๆ ในฟอร์ม ซึ่งวินโดว์ Code Editor จะแสดงขึ้นมาพร้อมสำหรับการป้อนโปรแกรมให้กับเหตุการณ์หลักของออบเจกต์นั้น ส่วนที่สำคัญของวินโดว์คือ คอมโบบ็อกซ์ (Combo box) ทั้งสองช่องที่อยู่ตรงส่วนบนของวินโดว์ ซึ่งจะเป็นควบคุมการเลือกออบเจกต์และเหตุการณ์ (Event) ที่จะเกิดขึ้นกับออบเจกต์นั้น โดยโค้ดที่ปรากฏจะเป็นโปรแกรมหรือคำสั่งที่จะถูกเรียกใช้งานเมื่อมีเหตุการณ์นั้นเกิดขึ้นกับออบเจกต์ดังภาพที่ 2-34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Object list box

แสดงชื่อออบเจกต์

ของโปรแกรมและ

ใช้เลือกออบเจกต์

View คู่มือโปรแกรมของ

ออบเจกต์และอีเวนต์ทุก

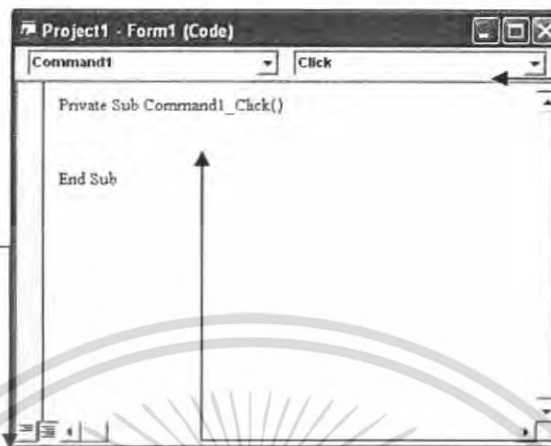
อันที่เราเขียนโปรแกรม

View คู่มือเฉพาะ

โปรแกรมของ

ออบเจกต์และอีเวนต์

ปัจจุบันอย่างเดียว



Event list box แสดง

ชื่ออีเวนต์ของออบเจกต์

ที่กำลังเขียนโปรแกรม

อยู่และใช้เลือก

เหตุการณ์ต่างๆของ

ออบเจกต์นั้น ๆ

ชื่อโปรซีเจอร์

ของเหตุการณ์

รูปที่ 2-34 แสดงวินโดว Code Editor

Object list box จะแสดงชื่อออบเจกต์ว่าส่วนของโปรแกรมที่กำลังแสดงอยู่ใน Code editor เป็นของออบเจกต์ใด ถ้าคลิกที่กรอบคำวนจะปรากฏลิสต์รายการของออบเจกต์ต่าง ๆ ที่อยู่ในฟอร์ม (โดยเป็นชื่อของออบเจกต์ที่ตั้งไว้ในพร็อพเพอร์ตี้ Name) เราสามารถทราบออบเจกต์และเหตุการณ์ได้จากชื่อของโปรซีเจอร์ เช่น Command1_Click จะหมายถึงส่วนของโปรแกรมที่จะทำงาน เมื่อออบเจกต์ ชื่อ Command1 ถูกคลิก เป็นต้น

2.5.1.9 ชนิดตัวแปรใน Visual Basic

ชนิด	รายละเอียด	หน่วยความจำ
Byte	มีค่าตั้งแต่ 0 – 255	1 ไบต์
Integer	ใช้เก็บเลขจำนวนเต็ม มีค่าตั้งแต่ –32,768 ถึง 32,767	2 ไบต์
Long	ใช้เก็บเลขจำนวนเต็ม มีค่าตั้งแต่ –2,147,483,648 ถึง 2,147,483,647	4 ไบต์
Single	ใช้เก็บเลขจำนวนจริงมีจุดทศนิยม โดยมีค่าลบตั้งแต่ -3.402823×10^{38} ถึง $-1.401298 \times 10^{-45}$ และค่าบวกตั้งแต่	4 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1.401298×10^{-45} ถึง 3.402823×10^{38}	
Double	ใช้เก็บเลขจำนวนจริงมีจุดทศนิยม โดยมีค่าลบตั้งแต่ $-1.79769313486232 \times 10^{308}$ ถึง $4.94065645841247 \times 10^{-324}$ และค่าบวกตั้งแต่ $4.94065645841247 \times 10^{-324}$ ถึง $1.79769313486232 \times 10^{308}$	8 ไบต์
Currency	ใช้เก็บเลขทศนิยมตำแหน่งจำกัด(Fixed Decimal Point) มีค่าตั้งแต่ -922,337,203,685,477.5808 ถึง 922,337,203,685,477.5808	8 ไบต์
Boolean	ใช้เก็บค่าตรรกะ(logic)คือ True และ false	2 ไบต์
String	ใช้เก็บกลุ่มของตัวอักษรหรือชุดของข้อความ(Character string)	ขึ้นกับความยาวของข้อความ
Date	ใช้เก็บตัวแปรที่เป็นเวลา หรือวันที่	8 ไบต์
Object	ใช้เก็บตัวแปรที่อ้างถึงออบเจกต์ใดๆ	4 ไบต์
Variant	เป็นตัวแปรพิเศษของ Visual basic สามารถใช้เก็บตัวแปรได้ทุกแบบตามข้างต้น	22 ไบต์

ตารางที่ 2-7 ชนิดตัวแปรใน Visual Basic

ตัวแปรในภาษาทางคอมพิวเตอร์นั้นมีอยู่มากมายหลายชนิดตามลักษณะการใช้งาน โดยตัวแปรแต่ละชนิดก็จะมีช่วงเก็บค่า หรือประเภทของค่าที่เก็บ รวมถึงเนื้อที่ขนาดต่างๆ จะสังเกตเห็นว่าตัวแปร Variant สามารถจัดเก็บตัวแปรได้ทุกประเภท แล้วทำไมต้องมีตัวแปรชนิดอื่น ให้อนุวยด้วยแต่ถ้าสังเกตดีๆจะพบว่าตัวแปรแบบ Variant นั้นต้องการเนื้อที่จัดเก็บต่อหนึ่งตัวแปร ถึง 22 ไบต์ตัวอย่าง เช่น ถ้าคอมพิวเตอร์ของเราคำนวณสต็อกของสินค้า 1,000 ชนิด โดยจำนวนสินค้าแต่ละชนิดเป็นเลขจำนวนเต็ม เราต้องการใช้ตัวแปรแบบจำนวนเต็ม Integer 1,000 ตัว ซึ่งต้องการเนื้อที่ในหน่วยความจำ $1,000 \text{ ตัวแปร} \times 2 \text{ ไบต์} = 2,000 \text{ ไบต์}$ แต่ถ้าใช้ตัวแปรแบบ Variant ในการเก็บค่าจะต้องใช้เนื้อที่ถึง $1,000 \text{ ตัวแปร} \times 22 \text{ ไบต์} = 22,000 \text{ ไบต์}$ ต่างกันถึง 10 เท่าเลยทีเดียว นอกจากเรื่องการใช้นี้เนื้อที่แล้ว การตั้งตัวแปรไม่ตรงชนิดยังมีผลต่อความเร็วของการประมวลผลที่ลงอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.10 การประกาศตัวแปรและกฎการตั้งชื่อตัวแปรใน Visual basic การประกาศตัวแปร คือการที่เราบอกเครื่องคอมพิวเตอร์ให้เตรียมเนื้อที่ในหน่วยความจำสำหรับเก็บตัวแปรที่เราจะใช้ในการประมวลผลในโปรแกรม โดยการประกาศตัวแปรจะใช้คำสั่งว่า Dim (Dimension) ซึ่งมีไวยากรณ์ดังนี้

รูปแบบ Dim ชื่อตัวแปร [as ชนิดของตัวแปร]

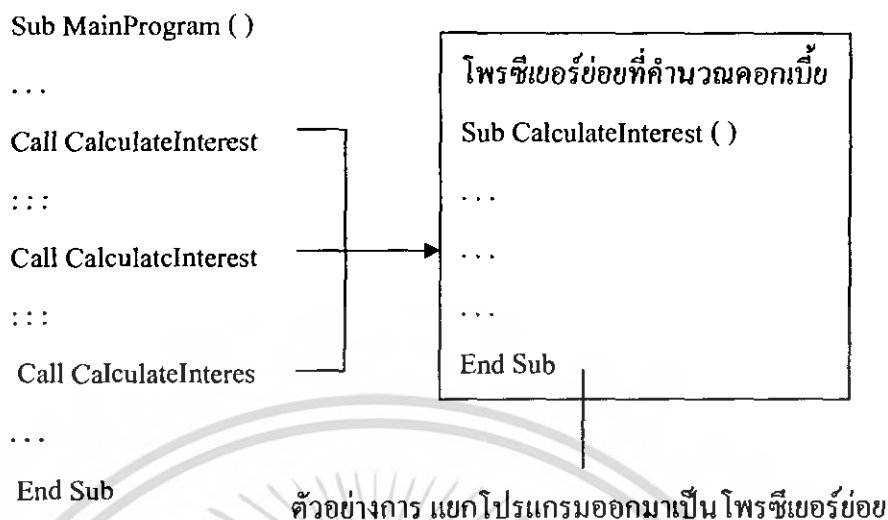
ค่าที่อยู่ภายใต้เครื่องหมาย [] จะเป็น Optional คือใส่หรือไม่ก็ได้ ถ้าไม่ได้ใส่จะเป็นค่าดีฟอลต์ (Default) ที่กำหนดไว้แล้ว สำหรับชนิดของตัวแปรก็คือชนิด variant

ก) ส่วนการตั้งชื่อตัวแปรใน Visual Basic นั้นต้องมีกฎเกณฑ์ดังต่อไปนี้

1. ชื่อตัวแปรที่ใช้ต้องขึ้นต้นด้วยตัวอักษร
2. ชื่อตัวแปรห้ามยาวเกิน 255 ตัวอักษร
3. สามารถใช้ตัวอักษรตัวเลขหรือเครื่องหมายขีดล่าง (_) ผสมเป็นชื่อ แต่ห้ามมีจุด, ขีดหรือเครื่องหมายอื่นภายในชื่อ
4. ห้ามซ้ำกับคำที่สงวนไว้ในโปรแกรม เช่น Dim เป็นต้น
5. ชื่อตัวแปรห้ามซ้ำกันถ้าอยู่ในขอบเขต (Scope) เดียวกัน

2.5.1.11 ตัวแปรแบบค่าคงที่ รูปแบบ Constant ชื่อตัวแปร = ค่าที่ต้องการ ตัวแปรแบบ Constant จะใช้ในการเก็บค่าคงที่ที่เราคาดว่าจะใช้บ่อยในโปรแกรม โดยการประกาศตัวแปรที่จะใช้ คำสั่ง Const ชื่อตัวแปร = ค่าที่ต้องการ

2.5.1.12 โปรแกรมย่อย ในการเขียนโปรแกรมนั้น มักมีการแบ่งโปรแกรมออกเป็น ส่วนย่อยๆ เพื่อสะดวกต่อการ ควบคุมการทำงานและจัดการ โปรแกรมให้ง่ายขึ้น โปรแกรมย่อยๆ เหล่านั้นเรียกว่า Sub Procedure หรือ โพรซีเจอร์ย่อย ลองพิจารณาจากโปรแกรมด้านล่าง สมมติว่า ในโปรแกรมมีส่วนที่ต้องคำนวณดอกเบี้ยซึ่งเป็นโค้ดที่เหมือนกันเลยซ้ำกันอยู่สามจุด ทำให้เปลืองเนื้อที่ และถ้าเกิดมีการแก้ไข โปรแกรมในส่วนนี้ เราต้องแก้ไขซ้ำๆ ถึงสามที่ด้วยกัน ซึ่งไม่สะดวกอย่างยิ่ง วิธีการที่ดีกว่านั้นคือแยกโปรแกรมที่ซ้ำกันนี้ออกมาเป็น โพรซีเจอร์ย่อยดังภาพ จะเห็นว่า โปรแกรมของเราจะสั้นลงอย่างมาก นอกจากนั้นยังสะดวกต่อการจัดการ การแก้ไข และดูแลโปรแกรม เข้าใจได้ง่ายขึ้นกว่าเดิมจะแสดงดังภาพที่ 2-35



รูปที่ 2-35 แสดงโปรแกรมย่อย

2.5.1.12 ฟังก์ชัน ฟังก์ชัน (Function) เป็นโปรซีเยอร์ย่อยที่ฟังก์ชันมีการคืนค่ากลับไปให้โปรแกรมที่เรียกใช้ด้วย วิธีใช้ก็เพียงแต่ใช้คำสั่ง Function แทนคำว่า Sub และมีการกำหนดชนิดของค่าที่คืนกลับมาคือท้ายฟังก์ชันเหมือนกันกับตัวแปร เช่น

Function CalculateInterest () As Currency

ในฟังก์ชันจะส่งค่ากลับไปโดยการกำหนดค่านั้นให้กับชื่อฟังก์ชันเหมือนกับว่าชื่อฟังก์ชันเป็นตัวแปรตัวหนึ่ง และในการเรียกใช้ฟังก์ชันก็สามารถใช้ในสูตรคำนวณหรือนำไปกำหนดค่าให้กับตัวแปรอื่นๆก็ได้

2.5.1.13 Event Procedure Event Procedure คือโปรแกรมย่อยชนิดหนึ่ง ที่จะทำงานก็ต่อเมื่อมีเหตุการณ์เกิดขึ้นกับออบเจกต์โดยใน Event Procedure จะเก็บโค้ดของแต่ละเหตุการณ์สำหรับแต่ละออบเจกต์ตามหลักการเขียนโปรแกรมแบบ Event-driven ตัวอย่าง เช่น โปรซีเยอร์ย่อยที่ชื่อ Command1_Click () เป็น Event Procedure ซึ่งเก็บโค้ดที่จะทำงานก็ต่อเมื่อมีการคลิกปุ่ม Command1 หรือ โปรซีเยอร์ ชื่อ Form_Load () จะเป็น Event Procedure ซึ่งเก็บโค้ดที่จะทำงานก็ต่อเมื่อมีการเปิดฟอร์มครั้งแรก เป็นต้น

Object list box

จะแสดงชื่อออบเจกต์

ต่างๆ ที่มีอยู่ใน

ฟอร์มขณะนี้

Event procedure



Event list box
จะมีเหตุการณ์
ไว้ให้ใช้

รูปที่ 2-36 แสดง Event Procedure

การเข้าถึง Event Procedure นั้นทำได้โดยการเลือก Object list box และ Event list box ของ Code Editor เมื่อเราเลือกออบเจกต์และอีเวนต์ตามต้องการแล้ว หน้าจอของ Code Editor ก็จะเป็น Sub procedure ของเหตุการณ์และออบเจกต์นั้นดังภาพที่ 2-36

2.5.1.14 ขอบเขตของตัวแปร (Scope of variable)

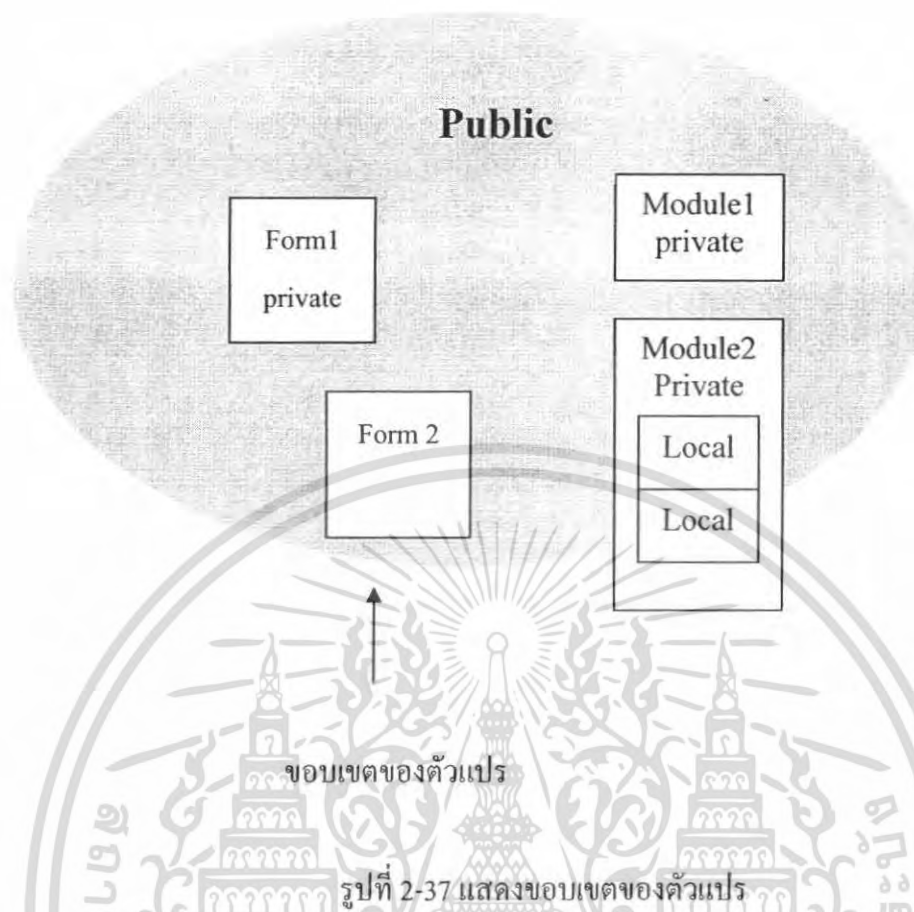
ก) ตัวแปร Public มองเห็นและเรียกใช้ได้ในทุกฟอร์ม, โมดูล หรือทุกโปรซีเยอร์ การประกาศทำได้โดยใช้คำว่า Public แทนคำว่า Dim ในบรรทัดต่อจาก Option Explicit นอกโปรซีเยอร์ใดๆ (เป็นของสาธารณะทุกคนเห็นและใครจะใช้ได้)

ข) ตัวแปร private มองเห็นและเรียกใช้ได้เฉพาะทุกโปรซีเยอร์ในโมดูล หรือ ฟอร์มที่ประกาศ

เท่านั้น อาจเรียกได้ว่าเป็นตัวแปร Module level variable (เป็นของส่วนตัวของฟอร์ม หรือโมดูลฟอร์มอื่น โมดูลอื่น อย่างใด) การประกาศทำได้โดยใช้คำว่า Private แทนคำว่า Dim ในบรรทัดต่อจาก Option Explicit)

ค) ตัวแปร Local มองเห็นและเรียกใช้ได้เฉพาะโปรซีเยอร์ที่ประกาศเท่านั้น โดยโปรซีเยอร์อื่นห้ามแตะ อาจเรียกได้ว่าเป็น Local variable ดังนั้นตัวแปรชื่อเดียวกันที่อยู่คนละโปรซีเยอร์จะเป็นตัวแปรคนละตัวกัน ตัวแปรเหล่านี้จะถูกสร้างขึ้นเฉพาะเมื่อเข้ามาทำงานในโปรซีเยอร์นี้เท่านั้น และเมื่อออกไปจากโปรซีเยอร์นี้ก็จะถูกทำลายทิ้งไปจะแสดงดังภาพที่ 2-37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

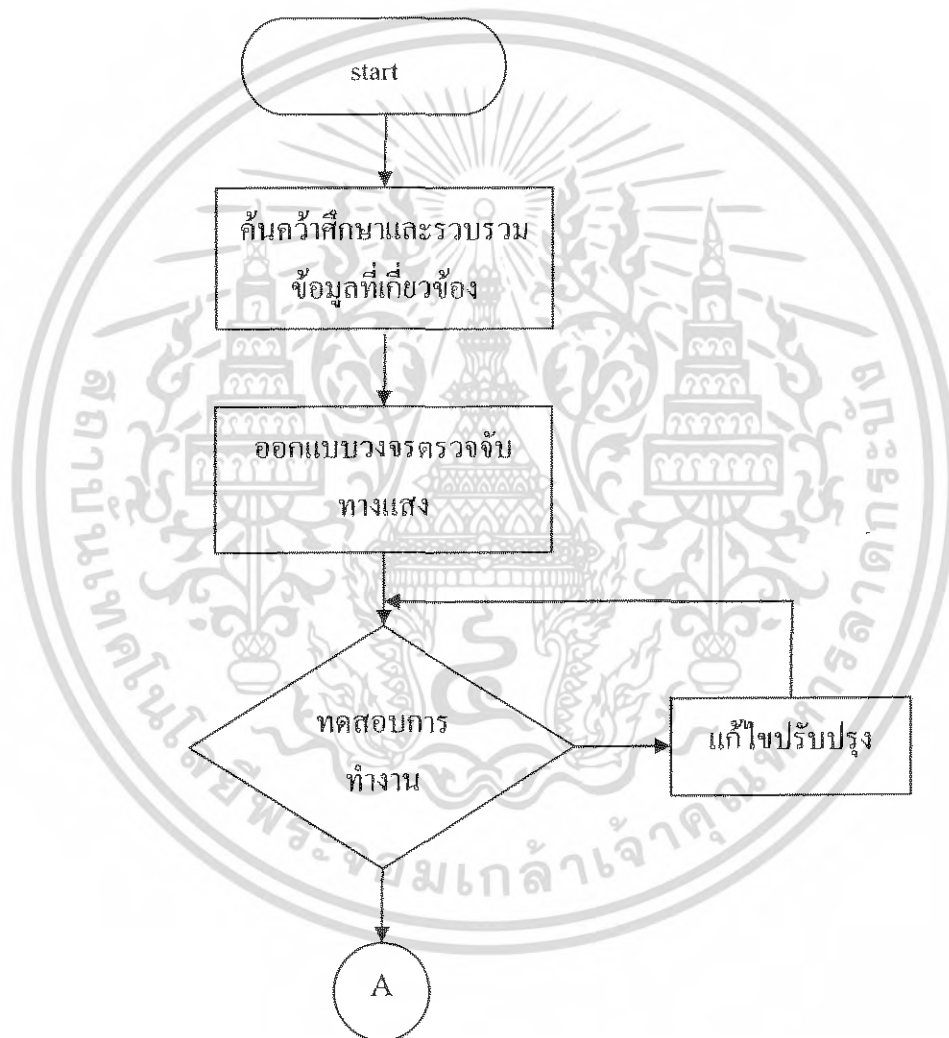


2.5.1.15 ตัวแปรแบบ Static ในบางครั้งหากเราต้องการให้ตัวแปรของโปรซีเยอร์ มีค่าคงที่ อยู่แม้ว่าโปรแกรมจะทำงานไปจนจบโปรแกรมย่อยนั้นแล้ว สามารถทำได้โดยประกาศตัวแปรนั้น ให้เป็นตัวแปรแบบ Static ซึ่งตัวแปรแบบนี้จะต่างจากตัวแปรทั่วไปคือ มักจะเก็บค่าที่อยู่ในตัวแปร ไว้ตราบใดที่โปรแกรมยังทำงานอยู่ โดยไม่ทำลายทิ้งหรือสร้างขึ้นใหม่ทุกครั้ง ที่จบโปรซีเยอร์ หรือฟังก์ชัน แต่ยังคงใช้เฉพาะในขอบเขตของโปรซีเยอร์หรือฟังก์ชันเท่านั้น กล่าวอีกนัยหนึ่งคือ ตัวแปรแบบ Static ก็คือตัวแปรแบบ Local ในโปรซีเยอร์แต่ทำงานเหมือนกับตัวแปร Private ที่อยู่ ภายนอกซึ่งจะไม่ลืมหาค่าเมื่อถึงคำสั่ง End Sub นั่นเองและยังคงใช้ได้เฉพาะในโปรซีเยอร์ที่ประกาศ เท่านั้น

บทที่ 3

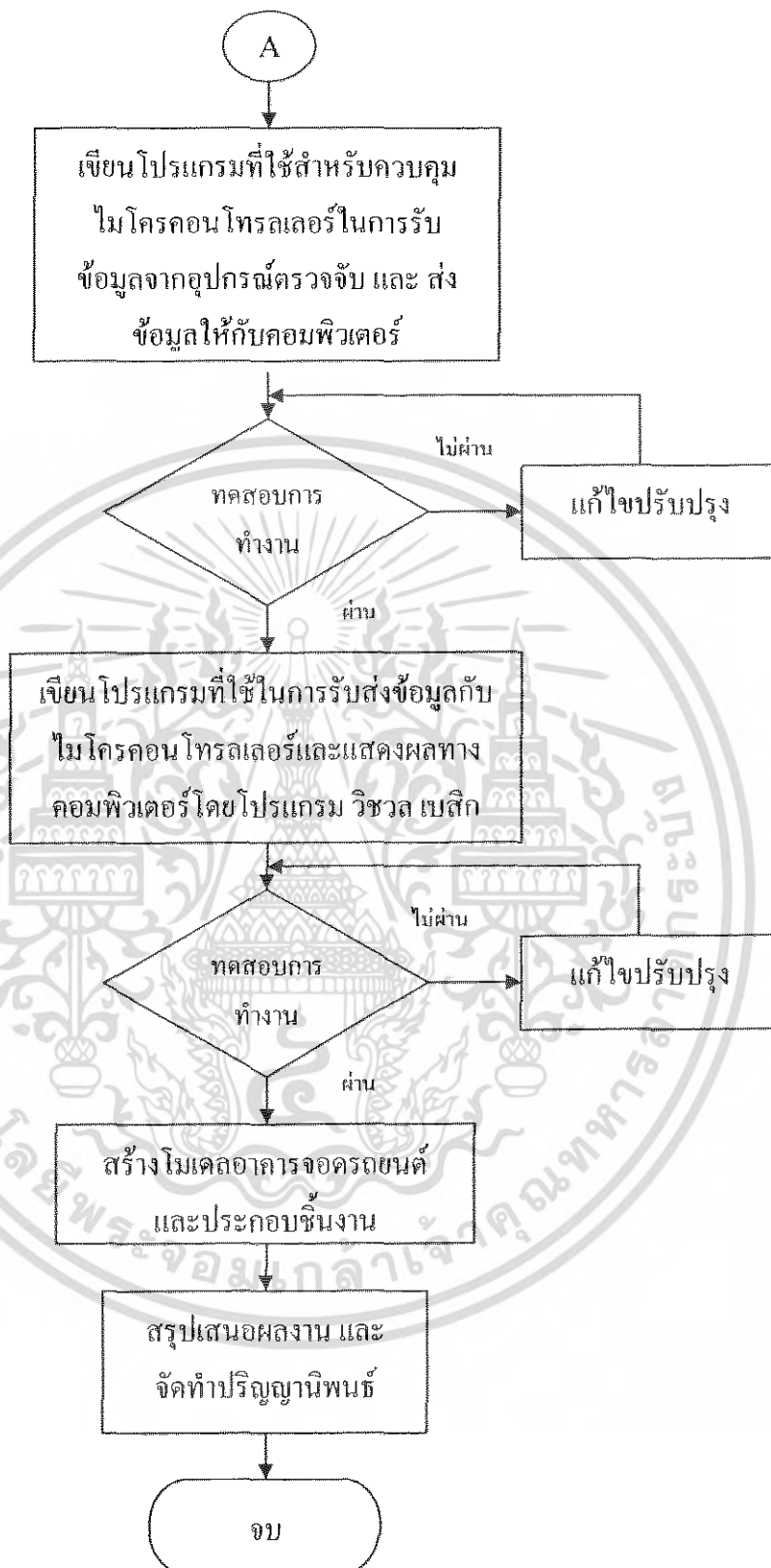
วิธีการดำเนินงาน

ทฤษฎีของระบบที่ทำการศึกษามา ซึ่งได้นำมาประกอบ การออกแบบและสร้างเป็น
โครงงานปริญญานิพนธ์ซึ่งสามารถแยกออกเป็นส่วนประกอบหลัก ได้ดังนี้



รูปที่ 3-1ก. แสดงฟโลว์ชาร์ตขั้นตอนการดำเนินงานของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

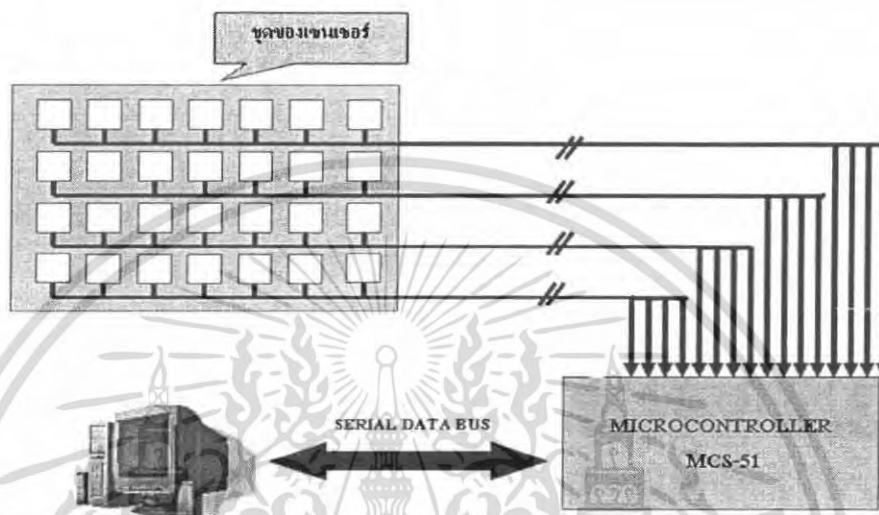


รูปที่ 3-1ข. แสดงโฟลว์ชาร์ตขั้นตอนการดำเนินงานของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

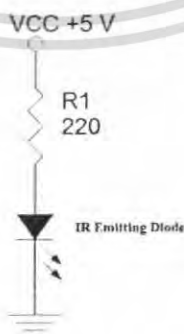
3.1 โครงสร้างของระบบในส่วนต่างๆของโครงการ

- 1 ชุดวงจรเซ็นเซอร์
- 2 ไมโครคอนโทรลเลอร์ MCS-51
- 3 แบบจำลองอาคารจอดรถ



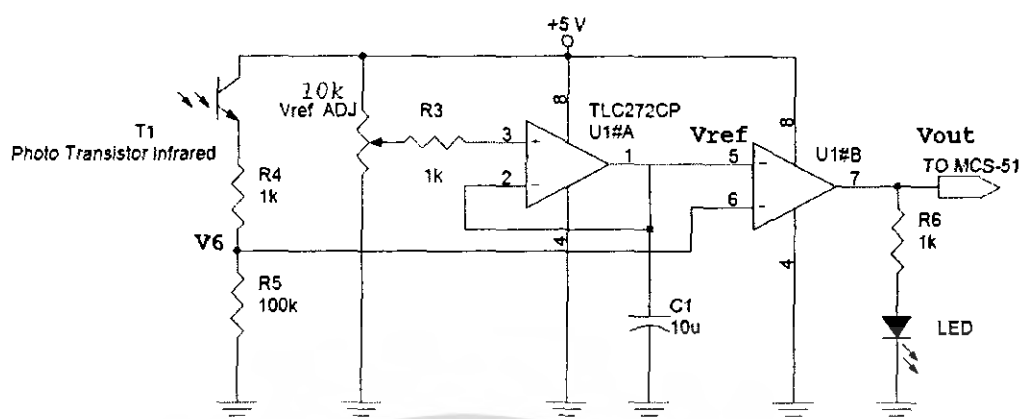
รูปที่ 3-2 แสดงบล็อกไดอะแกรมของแบบจำลองระบบบอกตำแหน่งพื้นที่ว่างในอาคารจอดรถ

3.1.1 วงจรตรวจจับเซ็นเซอร์ ศึกษาและออกแบบสร้างวงจรตรวจจับ โดยให้สามารถใช้ได้ในอาคารจำลองโดยจะมีระยะห่างกันประมาณ 8-10 เซนติเมตรตรวจจับด้วยแสงอินฟราเรดจะใช้โฟโตไดโอดเป็นตัวตรวจจับโดยติดตั้งตัวตรวจจับนี้ไว้ตรงตำแหน่งรถทุกตำแหน่ง เมื่อมีรถเข้ามาจอดจะตัดลำแสงอินฟราเรดระหว่างตัวส่งและตัวรับทำให้ได้สัญญาณลอจิกมีสถานะเป็น “1” แต่ถ้าไม่มีรถเข้ามาจอดก็จะได้สัญญาณที่มีสถานะเป็นลอจิก “0”

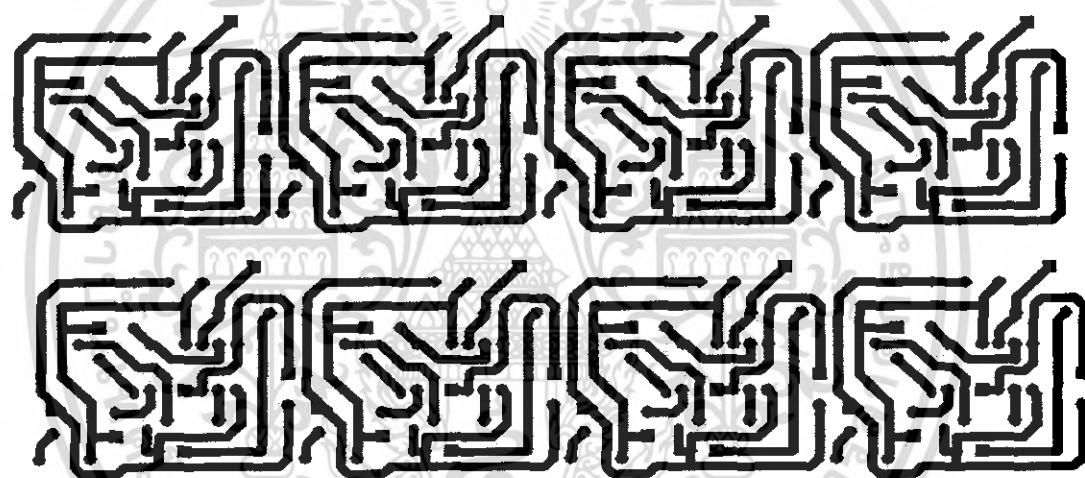


รูปที่ 3-3 วงจรภาคส่งแสงอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

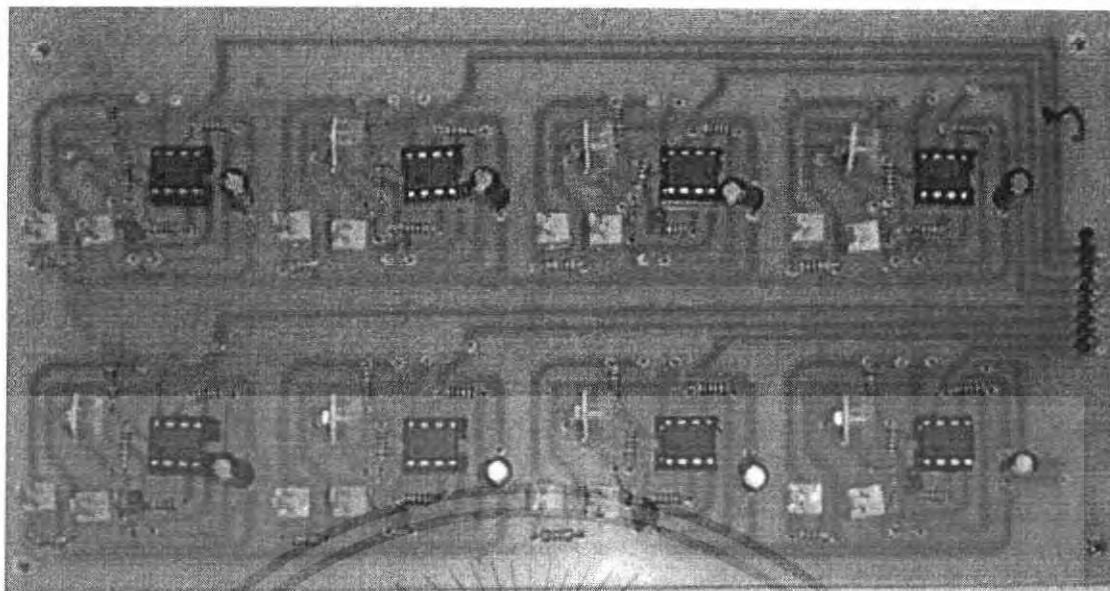


รูปที่ 3-4 วงจรภาครับของเซนเซอร์อินฟราเรด



รูปที่ 3-5 แสดงลายวงจรด้านล่างวงจรตรวจจับรวม 8 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-6 แสดงแผงวงจรตรวจจับโดยใช้แสงอินฟราเรดที่ใช้งานจริง

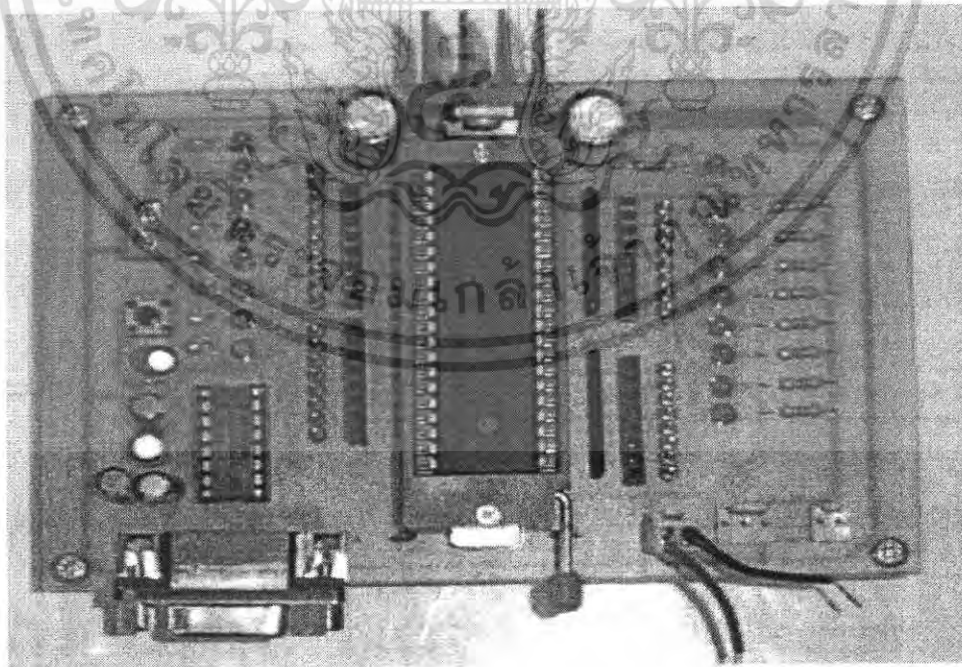
3.1.2 ไมโครคอนโทรลเลอร์ MCS-51 ไมโครคอนโทรลเลอร์ที่นำมาใช้งานเป็นบอร์ดสำเร็จรูปของ MCS-51 ซึ่งมีขนาดของหน่วยความจำโปรแกรมภายใน 128 Byte นำมาเพื่อใช้ในการรับและ ส่งข้อมูลจากอุปกรณ์ตรวจจับให้กับ เครื่องคอมพิวเตอร์ วงจรของบอร์ดแสดงดังภาพที่ 3-7 โดยการทำงานจะให้ไมโครคอนโทรลเลอร์ทำการเช็คสถานะของเซนเซอร์ทั้งหมด และส่งข้อมูลสถานะทั้งหมดไปยังคอมพิวเตอร์โดยการเชื่อมต่อแบบอนุกรม ผ่าน Serial Port และจะทำการรับการประมวลผลจำนวนของจำนวนข้อมูลที่ได้จากคอมพิวเตอร์ และ ส่งลอจิกออกมาเป็นเอาต์พุตเพื่อใช้เป็นสัญญาณไฟแสดงผล โดยขั้นตอนการทำงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์

การจัดขาต่าง ๆ ของ MCS-51

ไอซี ไมโครคอนโทรลเลอร์ 8051 โครงสร้าง ไอซี เป็นแบบ DIP มีขาทั้งหมด 40 ขา โดยขาต่างๆ เป็นขาพอร์ทอินพุต, ขาพอร์ทเอาต์พุต, ขาสัญญาณควบคุม, ขาดำเนินหน่วยความจำ ข้อมูลดังรูปที่ 3-7

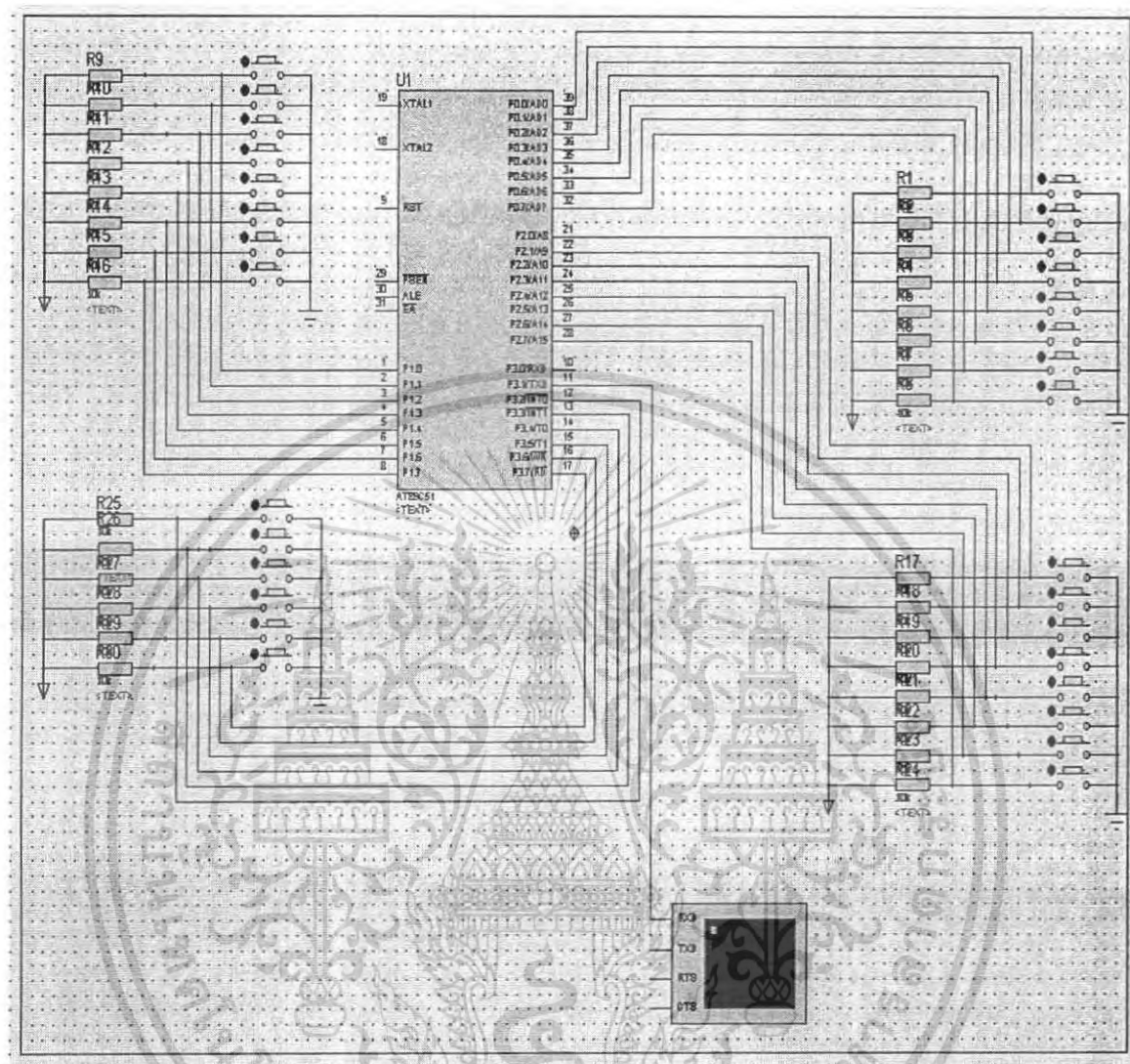


รูปที่ 3-7 แสดงขาต่างๆของ 89C52



รูปที่ 3-8 บอร์ดไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

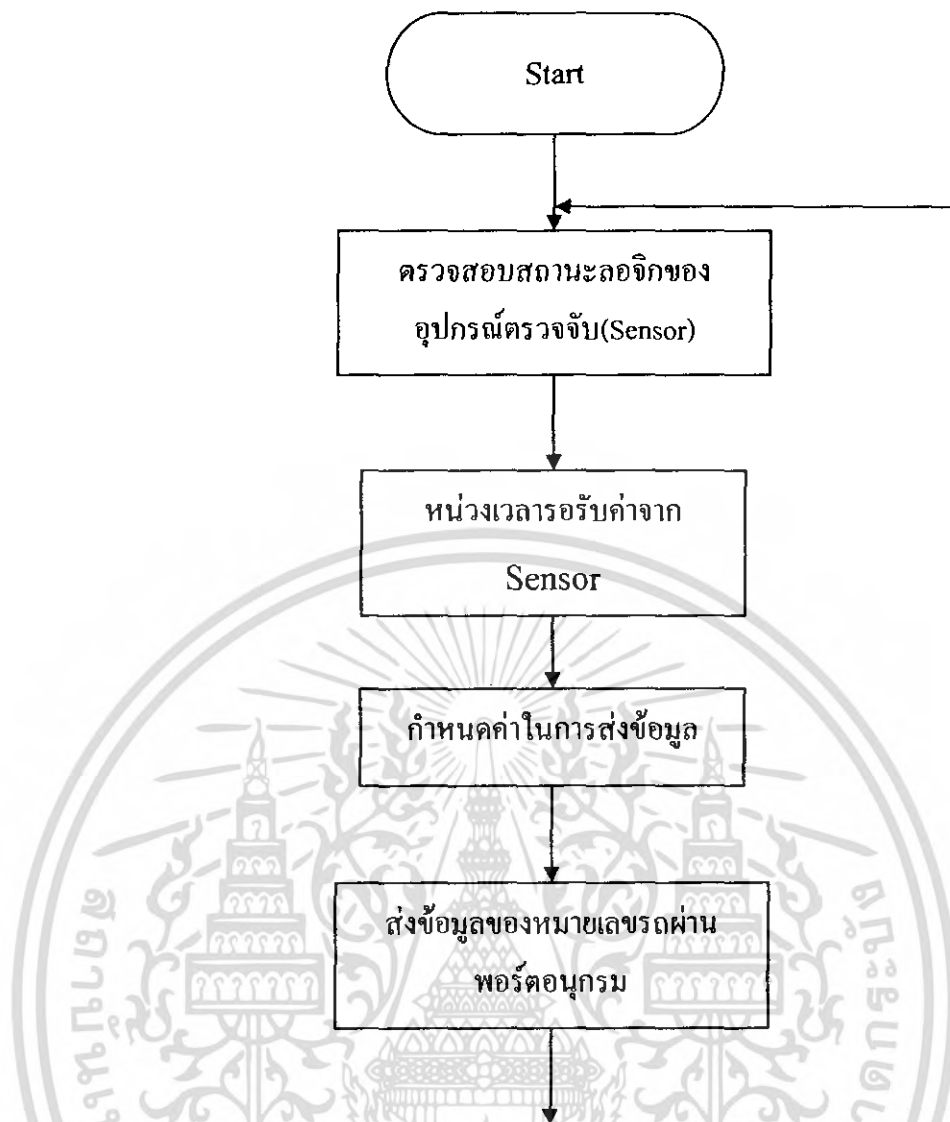


รูปที่ 3-9 การทดลองให้สวิตช์แทนเซนเซอร์ใน โปรแกรม Proteus

ตำแหน่งขาของ MCS-51	อินพุตที่รับ	ตำแหน่งที่โซลวบน VB
1	Sensor 9	B2
2	Sensor 10	B3
3	Sensor 11	B4
4	Sensor 12	B5
5	Sensor 13	B6
6	Sensor 14	B7
7	Sensor 15	C1
8	Sensor 16	C2
12	Sensor 25	C3
13	Sensor 26	C4
14	Sensor 27	C5
15	Sensor 28	C6
21	Sensor 18	C7
22	Sensor 19	D1
23	Sensor 20	D2
24	Sensor 21	D3
25	Sensor 22	D4
26	Sensor 23	D5
27	Sensor 24	D6
28	Sensor 25	D7
32	Sensor 8	B1
33	Sensor 7	A7
34	Sensor 6	A6
35	Sensor 5	A5
36	Sensor 4	A4
37	Sensor 3	A3
38	Sensor 2	A2
39	Sensor 1	A1

ตารางที่ 3-1 รายละเอียดขาอินพุต/เอาต์พุต ของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-10 ฟลัวร์ชาร์ตแสดงขั้นตอนการทำงานของ โปรแกรมควบคุมไมโครคอนโทรลเลอร์

3.2 การออกแบบโปรแกรม Visual basic

ในการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MCS- 51 จะใช้ภาษาซี เขียนเพื่อควบคุมการทำงานของเซนเซอร์ และส่งข้อมูลต่าง ๆ ค่อไปให้กับโปรแกรม Visual basicโครงสร้างของระบบในส่วนของการโปรแกรมจะแยกได้ 3 ส่วนคือ

1. โปรแกรม Visual basic จะแสดงแผนผังของที่จอดรถ ในการแสดงผลผังของอาคารจะใช้โปรแกรม Visual basic วาดฟอร์มของอาคารจอดรถยนต์ขึ้นมาโดยใช้เครื่องมือต่างๆ ที่ใช้ในการวาดเส้น ของโปรแกรม

รูปที่3-11 แสดงแผนผังของที่จอดครดโคใช้ โปรแกรม Visual basic

2. โปรแกรม Visual basic จะแสดงจำนวนของจำนวนรถที่จอดอยู่ภายในอาคาร โดยการให้โปรแกรม คำนวณจากจำนวนรถยนต์ที่อยู่ภายในอาคาร จำนวนรถยนต์ที่เข้ามา และจำนวนรถออกไปจากอาคารจอดรถยนต์เพื่อคำนวณให้ได้ผลลัพธ์ของจำนวนรถยนต์ทั้งหมดที่อยู่ภายในอาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form1

Text4

วัน/เดือน/ปี
16/2/2551

เวลา
0:19:34

จำนวนขบวน
8 คัน

เที่ยวที่จอด
20 คัน

จอง ยกเลิก

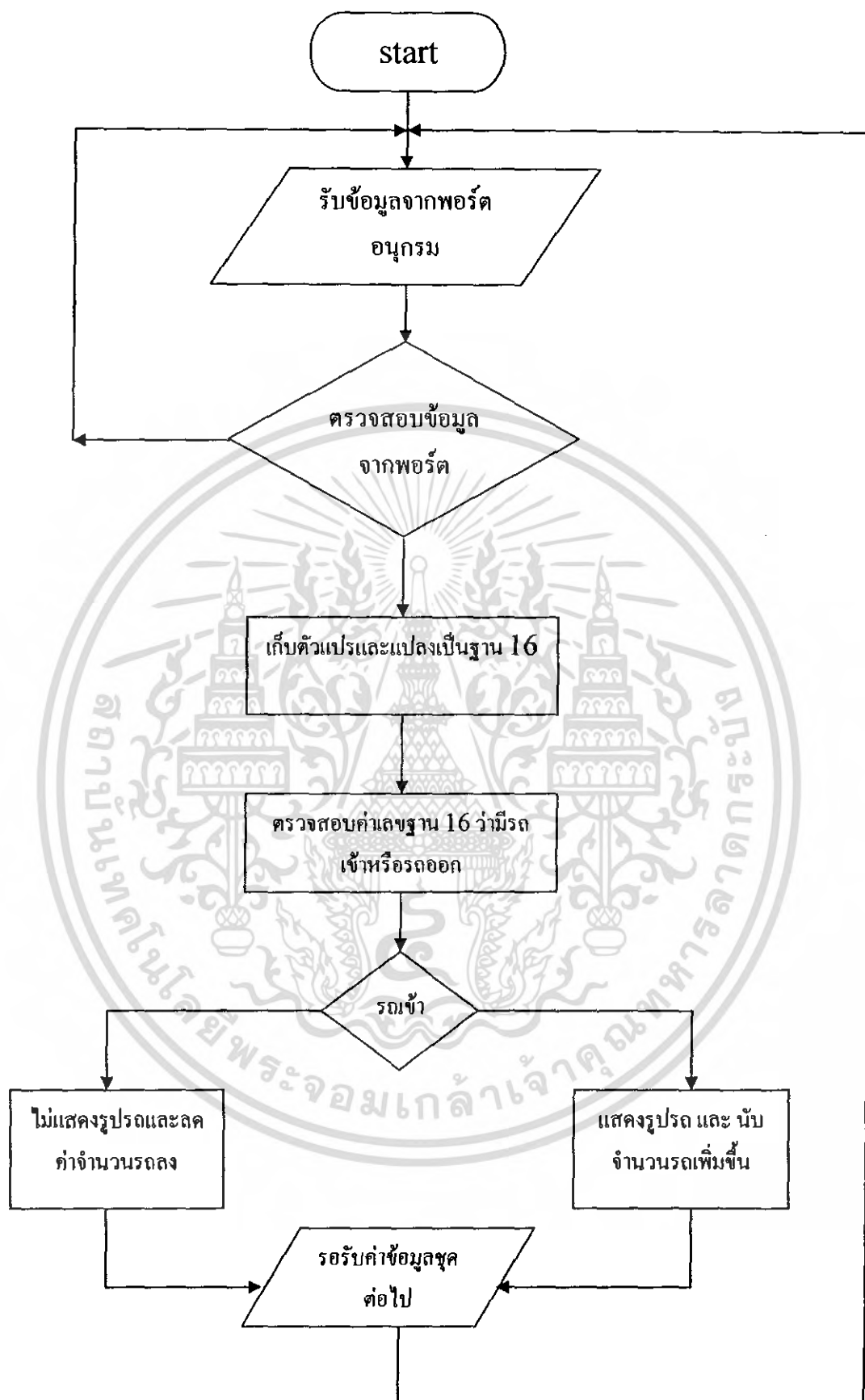
D1 D1
D2 D2
D3 D3
D4 D4
D5 D5
D6 D6
D7 D7

เข้า ออก

รูปที่ 3-12 แสดงตำแหน่งที่มีรถจอดและจำนวนรถทั้งหมด

3. โปรแกรม Visual basic จะแสดงจำนวนของรถว่าเหลือที่ว่างสำหรับจอดครรถเท่าไร โดยการให้โปรแกรมประมวลผลจากการคำนวณภาพของรถยนต์ที่แสดงขึ้นมาเพื่อหาผลรวมของจำนวนรถยนต์

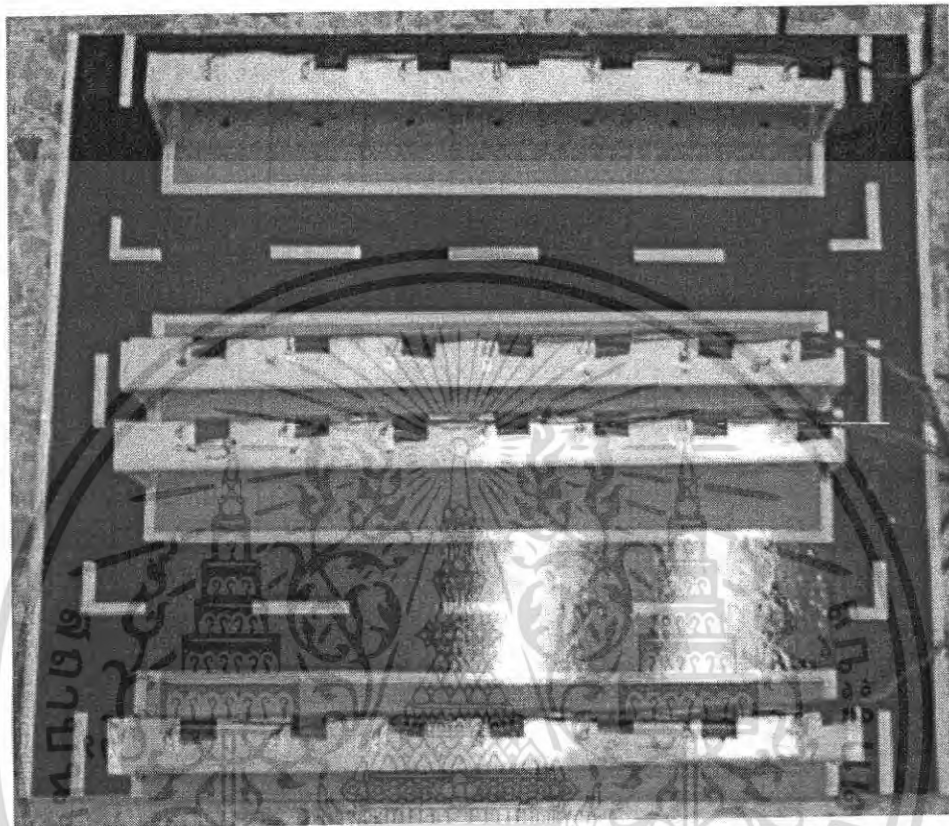
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3-13 โฟลว์ชาร์ตแสดงขั้นตอนการทำงานของโปรแกรม วิชาพลเบสิก

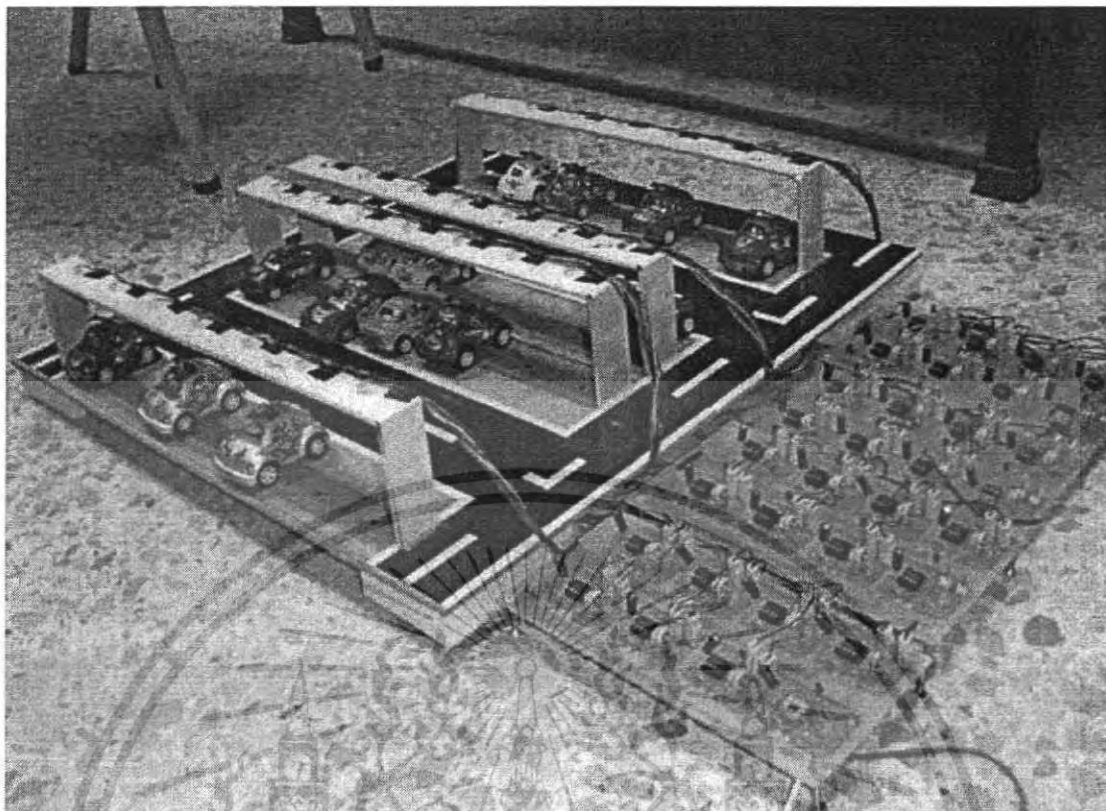
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3แบบจำลองที่จอตรง



รูปที่ 3-14 แบบจำลองที่จอตรงที่ติดตั้งเซ็นเซอร์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-15 แบบจำลองที่จอขณะมีรถจอดพร้อมชุดเซ็นเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

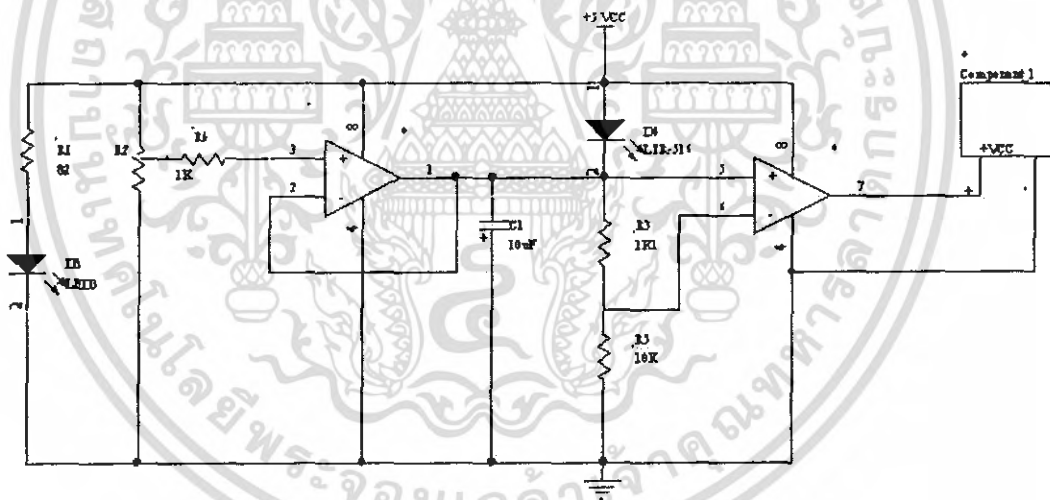
ผลการดำเนินงาน

ผลที่ได้จากการทำโครงงานปริญญานิพนธ์นี้จะได้แบบจำลองระบบบอกตำแหน่งพื้นที่ว่าง ในอาคารจอดรถอัตโนมัติโดยแสดงผลไปยังจอคอมพิวเตอร์และสามารถแสดงตำแหน่งที่ว่างของที่จอดรถและคำนวณจำนวนรถที่เข้า-ออก และจำนวนรถที่จอดในแต่ละชั้นได้ซึ่งประกอบด้วย

4.1 ผลการทดลองวงจรตรวจจับ

4.1.1 วงจรตรวจจับ

1. ตรวจสอบอุปกรณ์ทุกตัวเพื่อไม่ทำให้การทำงานของวงจรผิดพลาด
2. ต่อยังวงจรดังภาพที่ 4-1 และจ่ายไฟให้กับวงจร
3. ปรับระยะของตัวรับแสงและตัวส่งแสงให้ตรงกัน



รูปที่4-1 แสดงการทดลองวงจรตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่ง	วัดค่าแรงดันที่วงจรเซนเซอร์ เมื่อมีรบกวน(v)	วัดค่าแรงดันที่วงจรเซนเซอร์ เมื่อไม่มีรบกวน(v)
A1	3.8	0
A2	3.8	0
A3	3.8	0
A4	3.8	0
A5	3.8	0
A6	3.8	0
A7	3.8	0
B1	3.8	0
B2	3.8	0
B3	3.8	0
B4	3.8	0
B5	3.8	0
B6	3.8	0
B7	3.8	0
C1	3.8	0
C2	3.8	0
C3	3.8	0
C4	3.8	0
C5	3.8	0
C6	3.8	0
C7	3.8	0
D1	3.8	0
D2	3.8	0

ตาราง4-1 ก. การทดลองวัดค่าของเซ็นเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

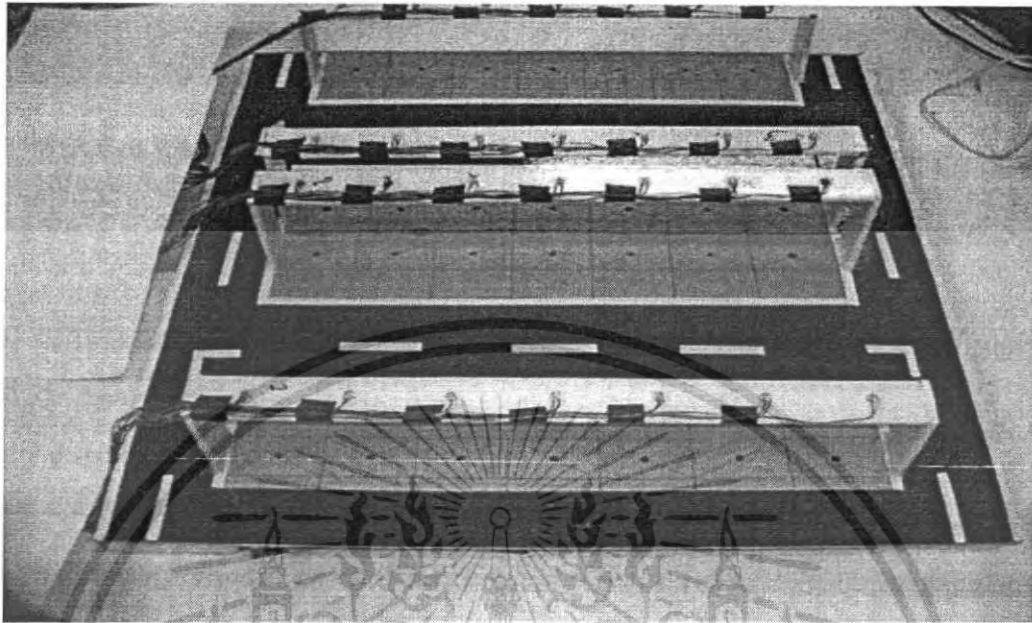
ตำแหน่ง	วัดค่าแรงดันที่วงจรเซนเซอร์ เมื่อมีรถเข้าจอด(v)	วัดค่าแรงดันที่วงจรเซนเซอร์ เมื่อไม่มีรถเข้าจอด(v)
D3	3.8	0
D4	3.8	0
D5	3.8	0
D6	3.8	0
D7	3.8	0

ตาราง4-1 ข. การทดลองวัดค่าของเซนเซอร์

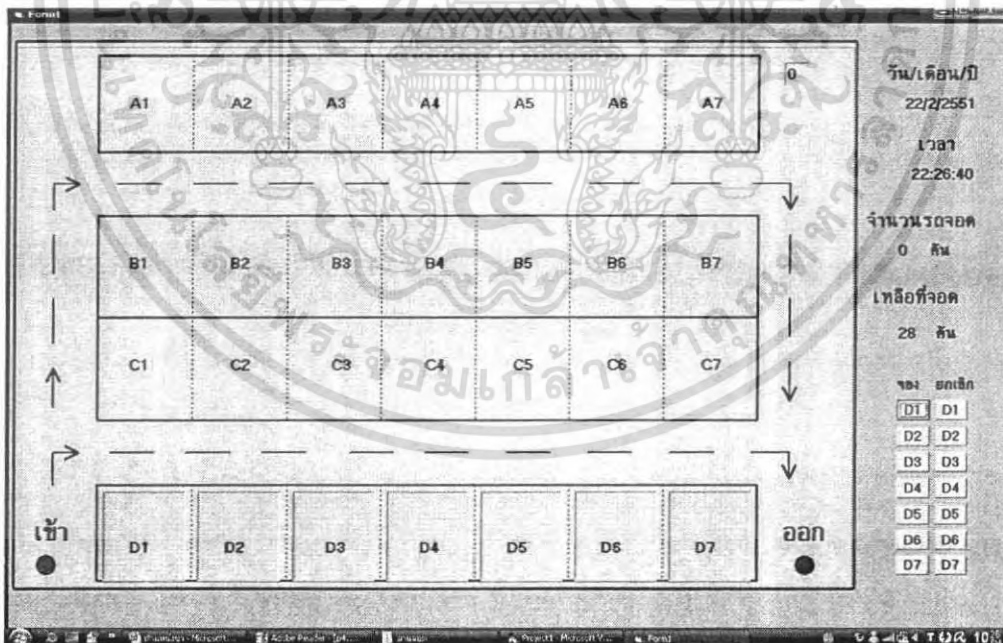
4.2 ทดสอบการทำงานของระบบ

1. ต่อชุดเซ็นเซอร์กับชุดไมโครคอนโทรลเลอร์และส่งค่าไปยังระบบ
2. ทดสอบนำรถเข้าไปจอดในลานจอดจำลอง
3. ตรวจสอบตำแหน่งที่เข้ามาจอดในลานจอดจริงจำลองกับหน้าจอมอนิเตอร์ตรงกันหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

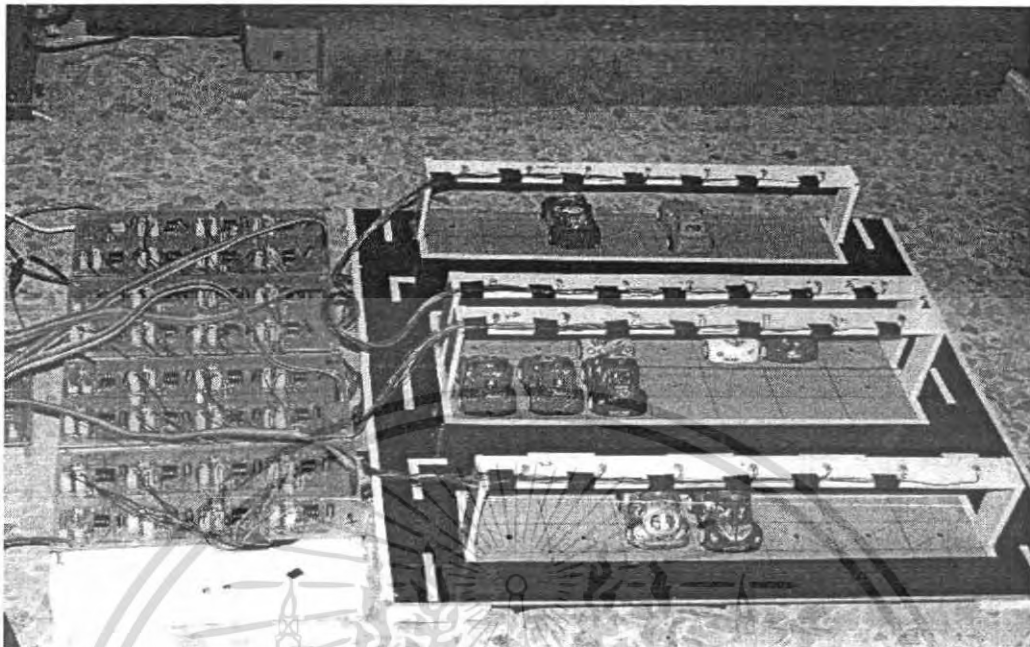


รูปที่4-2 แสดงโครงสร้างลานจอดรถ

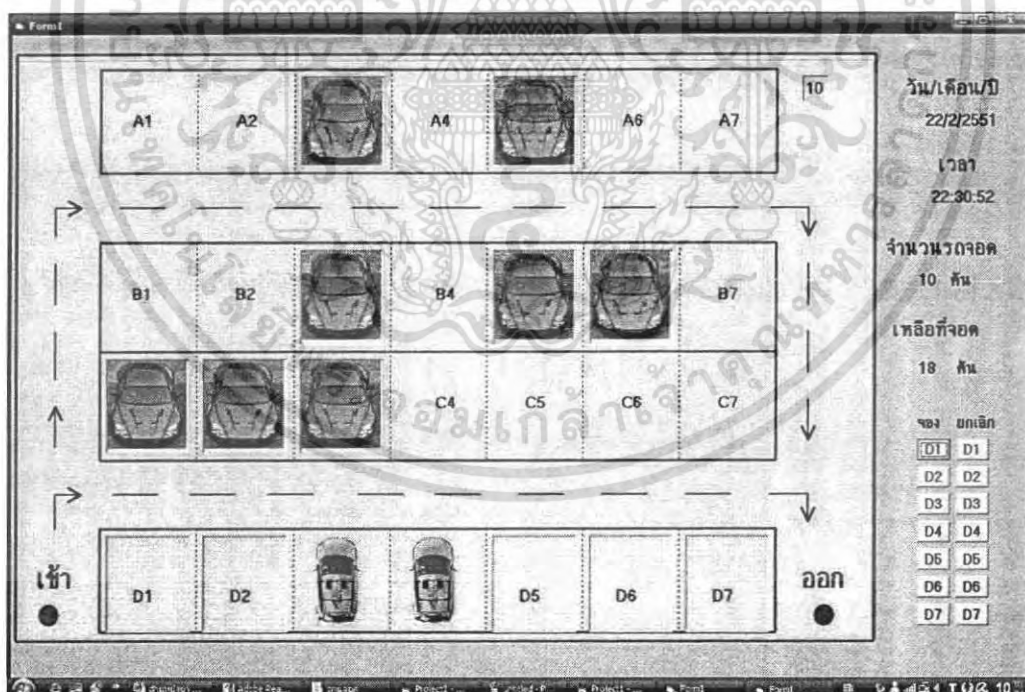


รูปที่4-3 แสดงตำแหน่งต่างๆ ของลานจอดรถที่แสดงทางหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

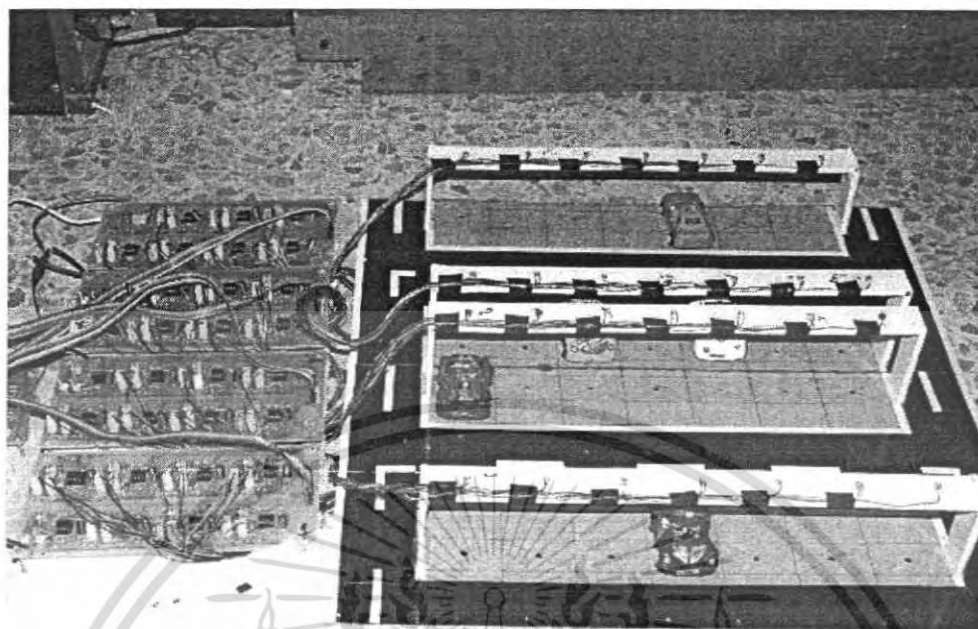


รูปที่4-4 แสดงโครงสร้างถานจอตระขณะมีรถจอด

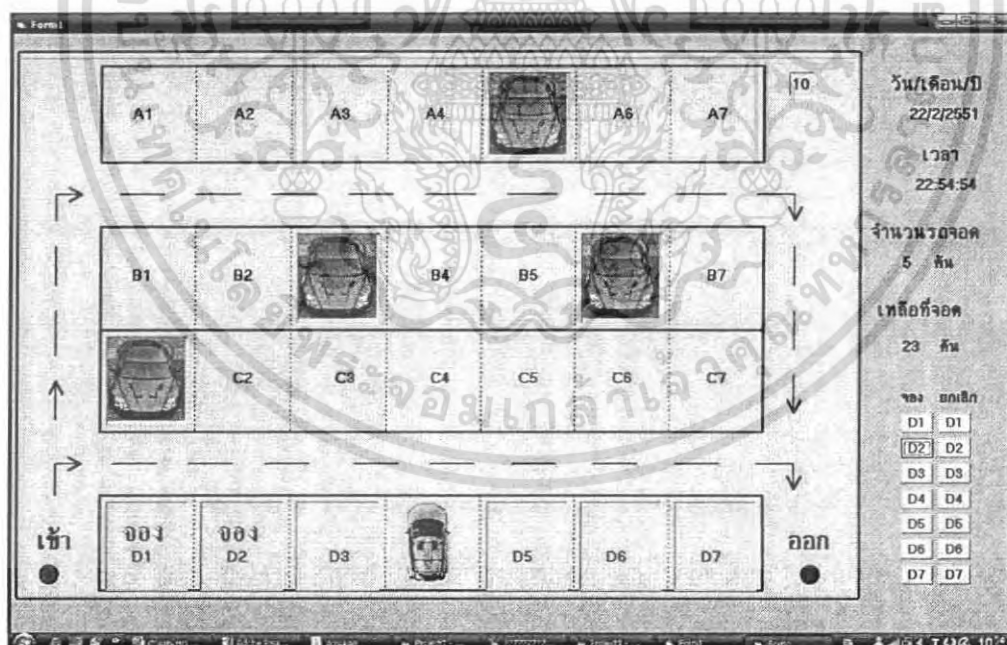


รูปที่4-5 แสดงตำแหน่งที่มีรถจอดตรงออกทางหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 แสดงโครงสร้างลานจอดรถขณะมีรถจอด



รูปที่ 4-7 แสดงตำแหน่งที่มีรถจอดครรถและมีการจองออกทางหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่ง	จำนวนครั้งที่ทดสอบ	เปอร์เซ็นต์ความผิดพลาด(%)
A1	5	0
A2	5	0
A3	5	0
A4	5	0
A5	5	0
A6	5	0
A7	5	0
B1	5	0
B2	5	0
B3	5	0
B4	5	0
B5	5	0
B6	5	0
B7	5	0
C1	5	0
C2	5	0
C3	5	0
C4	5	0
C5	5	0
C6	5	0
C7	5	0
D1	5	0
D2	5	0
D3	5	0
D4	5	0

ตารางที่ 4-2 ก. ผลการทดลองระบบบอกตำแหน่งที่ว่างของลานจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่ง	จำนวนครั้งที่ทดสอบ	เปอร์เซ็นต์ความผิดพลาด(%)
D5	5	0
D6	5	0
D7	5	0

ตารางที่ 4-2 ข. ผลการทดลองระบบบอกตำแหน่งที่ว่างของลานจอดรถ

จากการทดลองอุปกรณ์ตรวจจับเมื่อติดตั้งแล้วในแต่ละตำแหน่ง จากตารางที่ 4-2 พบว่าเมื่อทำการทดลองทุกๆตำแหน่งในลานจอดรถ ตั้งแต่ A1-A7, B1-B7, C1-C7 และ D1-D7 เป็นจำนวน 5 ครั้ง เปอร์เซ็นต์ค่าผิดพลาดของการทำงานมีค่าเป็น 0% แสดงว่าการทำงานของระบบเป็นไปอย่างถูกต้อง

บทที่ 5

สรุปผลการดำเนินงาน

ในการทำโครงงานปริญญานิพนธ์ แบบจำลองระบบบอกตำแหน่งพื้นที่ว่างในอาคารจอดรถอัตโนมัติ เพื่อศึกษาในด้านต่างๆที่เกี่ยวข้องกับโครงงาน เช่น การทำงานของอุปกรณ์ตรวจจับสัญญาณทางแสง การเขียนโปรแกรมด้วยวิซวลเบสิก การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกและการทำงานของไมโครคอนโทรลเลอร์ เป็นต้น โดยขอบเขตของโครงงานนี้คือ สามารถแสดงตำแหน่งของที่จอดรถยนต์แต่ละตำแหน่งภายในสถานจอดรถยนต์และแสดงจำนวนรถยนต์ที่จอดอยู่ในแบบจำลองนี้สามารถจอดรถยนต์ได้ 28 คันและทางเข้าจะต้องแสดงว่าอาคารนี้มีจำนวนรถยนต์เข้าไปเต็มหรือไม่ เราสามารถที่จะสรุปการทำงาน และปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการทำโครงงานปริญญานิพนธ์ รวมถึงแนวทางแก้ไข และข้อเสนอแนะเพื่อที่จะได้นำโครงงานปริญญานิพนธ์นี้ไปพัฒนาต่อไปซึ่งจะแบ่งเป็นข้อๆ ได้คือ

5.1 สรุปผล

แบบจำลองสามารถนำรถเข้าไปจอดได้จำนวน 28 คัน ซึ่งวงจรตรวจจับสามารถตรวจจับรถที่เข้ามาจอดได้โดยการควบคุมจากไมโครคอนโทรลเลอร์และส่งค่าผลจากวงจรตรวจจับไปยังคอมพิวเตอร์โดยผ่านพอร์ตอนุกรม บอกจำนวนรถที่เข้ามาจอด จำนวนที่จอดรถที่เหลือ และบอกจำนวนรถทั้งหมดที่เข้ามาจอดในแต่ละวัน เพื่อนำไปใช้เก็บสถิติทั้งหมดของรถที่เข้ามาจอดโดยจะเก็บจำนวนรถแยกตามวันที่

จากการทำโครงงานปริญญานิพนธ์นี้ทำให้ได้รับความรู้ความเข้าใจแล้วจะนำความสามารถในการทำงานของไมโครคอนโทรลเลอร์มาประยุกต์ใช้งานในด้านต่างๆและการเขียนโปรแกรม Visual Basic ในการคำนวณและการทำผังจอดรถที่แสดงเป็นภาพต่าง ๆ ได้ หรือกราฟฟิคต่างๆ ที่ทำให้มีภาพที่แสดงออกมามีความสวยงามมากขึ้น

5.2 ปัญหาและแนวทางการแก้ไข

ปัญหาและแนวทางแก้ไขที่เกิดขึ้นของการทำโครงการปริญญานิพนธ์แบบจำลองระบบบอกตำแหน่งพื้นที่ว่างในอาคารจอดรถอัตโนมัติ นี้คือ

1. ปัญหา การส่งแสงอินฟราเรดจากตัวส่งไปยังตัวรับระยะห่างของอุปกรณ์ทั้งสองทำให้ลำแสงที่ส่งไปเกิดการลดทอน และการรบกวนจากภายนอก แนวทางการแก้ไข เลือกตัวส่งที่มีกำลังส่ง (เปลี่ยนวงจรให้มีกำลังมากดังมากขึ้น) หรือความเข้มแสงที่มาก หรืออาจเลือกใช้เซ็นเซอร์ชนิดอื่นๆ ตามความเหมาะสมของสถานที่นั้นๆ
2. ปัญหา สถานะเอาต์พุตของวงจรตรวจจับไม่แน่นอนคือ ถ้าเอาต์พุตมีแรงดันระหว่าง 0.8-2.5 โวลต์ทำให้การผิดพลาดของข้อมูลเกิดขึ้นได้ แนวทางการแก้ไข โดยต่อเอาต์พุตผ่านบัฟเฟอร์ เพื่อแยกสถานะลอจิกให้ถูกต้อง
3. ปัญหา ความผิดพลาดที่เกิดจากอุปกรณ์ไม่ได้มาตรฐาน ทำให้การทำงานของวงจรผิดพลาดไม่ได้ตามที่ต้องการ แนวทางการแก้ไข ต้องมีการทดสอบอุปกรณ์ก่อนที่จะนำไปใช้งานเสียก่อน เพื่อความสะดวกและรวดเร็วต่อการติดตั้ง

5.3 แนวทางการพัฒนา

1. ใช้วงจรตรวจจับชนิดอื่น มาใช้แทนวงจรตรวจจับอินฟราเรด เช่น อัลตราโซนิก เพื่อให้ทำงานได้ดี และเหมาะสมขึ้น เพราะอินฟราเรดมีการรบกวนของแสง
2. ควรมีการขยายวงจรเพื่อให้สามารถรองรับชุดวงจรตรวจจับ ที่มีจำนวนมากขึ้นในสถานที่ที่ต่างกันออกไปพัฒนาจำนวนรถในระบบมากขึ้นด้วยการ ได้มากขึ้นด้วยการเพิ่มวงจรเซ็นเซอร์และทำการขยายพอร์ต MCS-51 การขยายพอร์ต MCS-51 ทำได้หลายวิธียกตัวอย่างเช่น การต่อสายบััส I2C หรือการต่อชุดคอนโทรลเพิ่ม
3. สามารถใช้เป็นแนวทางในการพัฒนา ให้ใช้กับสถานที่จริงได้ในอนาคต หรือสามารถนำไปพัฒนาให้ดีกว่าที่เป็นอยู่จากเดิม อาจมีการติดตั้งที่กันประตูทางเข้า-ออก ที่ควบคุมด้วยสเปคเตอร์
4. พัฒนาระบบให้มีการเก็บข้อมูลจำนวนรถเป็นแบบฐานข้อมูล และพัฒนาให้มีการจองที่จอดรถผ่านเครือข่ายอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ฉัททวุฒิ พิษผล. 2542. **Visual Basic 6**. กรุงเทพมหานคร: บริษัท โปรวิชั่น จำกัด
นคร ภัคดีชาติ และ ชัยวัฒน์ ลัมพรวิจิตร. 2521. **การทดลองและใช้งานไมโครคอลโทรลเลอร์
MCS-51 ด้วยโปรแกรมภาษาซี**. กรุงเทพฯ: บริษัท อินโนเวติฟ เอ็กเพอร์เมนต์ จำกัด
พจนางกูร สุวรรณมณี. **เซ็นเซอร์และทรานสดิวเซอร์เบื้องต้น**. กรุงเทพฯ :
สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2545.

<http://www.sut.ac.th/Engineering/Telecom/about/student2/ST51.htm>

<http://203.170.167.149/it/elearning/mcstext/stumCS51.htm>

<http://suranat.tripod.com/u3c2.html>

http://61.19.247.211/~adisak51/page06_1.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



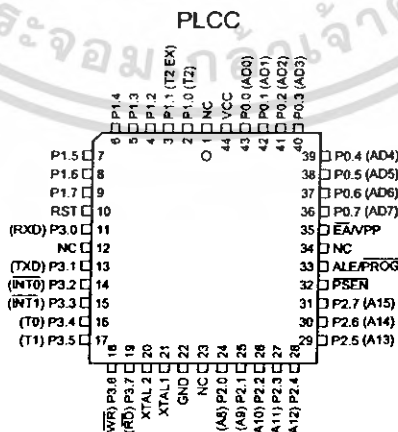
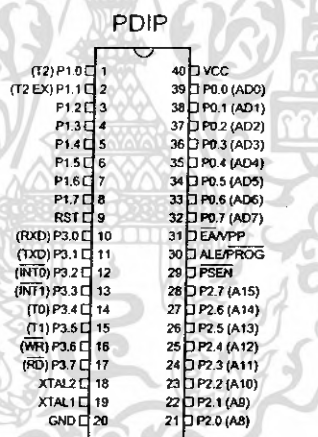
ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Compatible with MCS-51™ Products**
- **8K Bytes of In-System Reprogrammable Flash Memory**
- **Endurance: 1,000 Write/Erase Cycles**
- **Fully Static Operation: 0 Hz to 24 MHz**
- **Three-level Program Memory Lock**
- **256 x 8-bit Internal RAM**
- **32 Programmable I/O Lines**
- **Three 16-bit Timer/Counters**
- **Eight Interrupt Sources**
- **Programmable Serial Channel**
- **Low-power Idle and Power-down Modes**

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

PQFP/TQFP	
	P1.4
	P1.3
	P1.2
	P1.1 (T2 EX)
	P1.0 (T2)
	38 VCC
	39 GND
	P0.0 (A0)
	P0.1 (A01)
	P0.2 (A02)
	P0.3 (A03)
P1.5	1
P1.6	2
P1.7	3
RST	4
(RXD) P3.0	5
NC	6
(TXD) P3.1	7
(INT0) P3.2	8
(INT1) P3.3	9
(T0) P3.4	10
(T1) P3.5	11
(WR) P3.6	12
(RD) P3.7	13
XTAL1	14
XTAL2	15
GND	16
NC	17
(A8) P2.0	18
(A9) P2.1	19
(A10) P2.2	20
(A11) P2.3	21
(A12) P2.4	22
	33
	32
	31
	30
	29
	28
	27
	26
	25
	24
	23



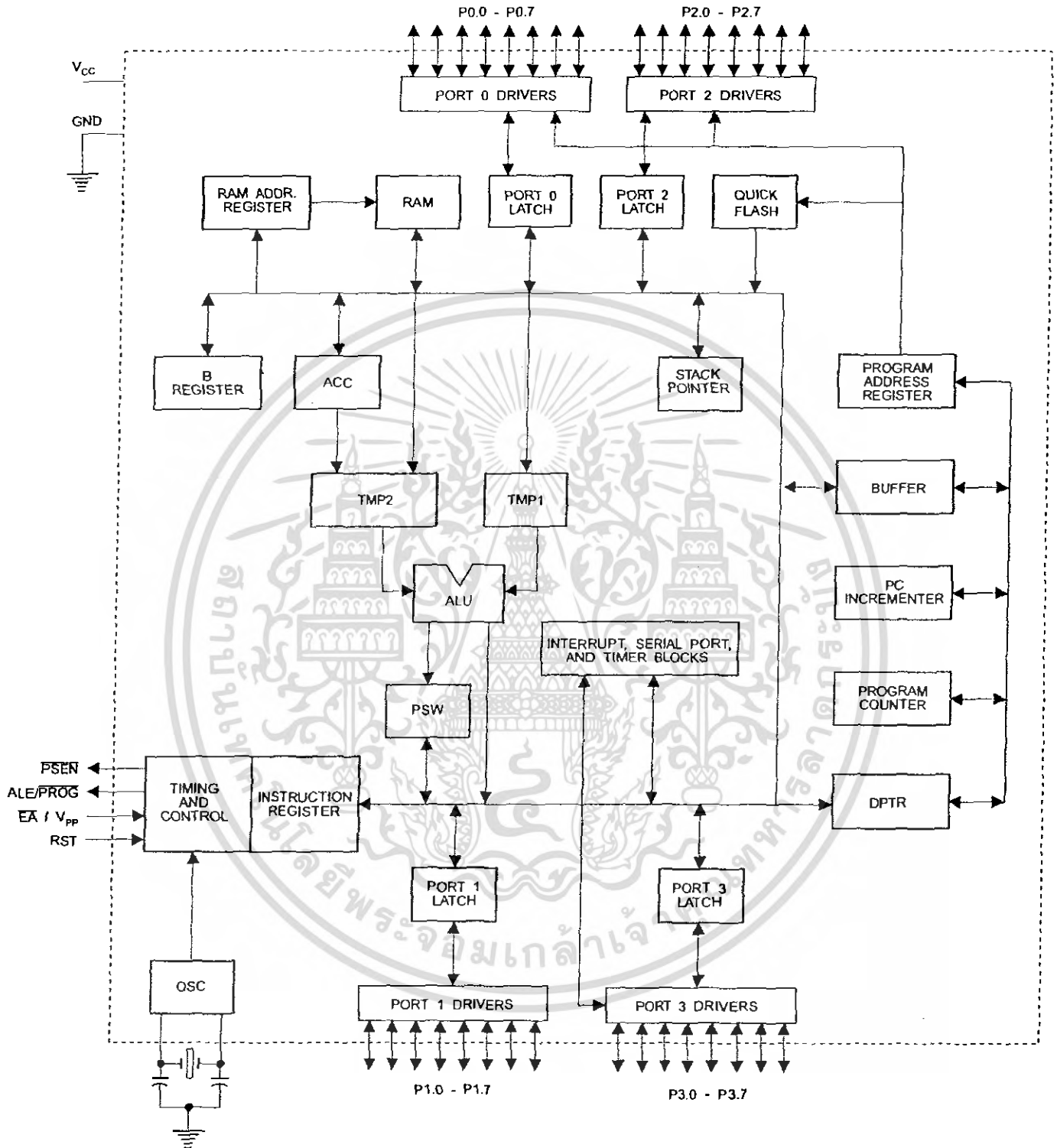
AT89C52

**Not Recommended
for New Designs.
Use AT89S52.**

Rev. 0313H-02/00



Block Diagram



The AT89C52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full-duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

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

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

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

Port 1

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

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

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

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

Port 2

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

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

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

Port 3

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

Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table.

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

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

RST

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

ALE/PROG

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

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

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

PSEN

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

When the AT89C52 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}/\text{VPP}$

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

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

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89C52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke

new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H

Bit Addressable

Reset Value = 0000 0000B

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/T2	Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered).
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Data Memory

The AT89C52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction

specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit $\overline{CT2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external

input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 1. Timer in Capture Mode

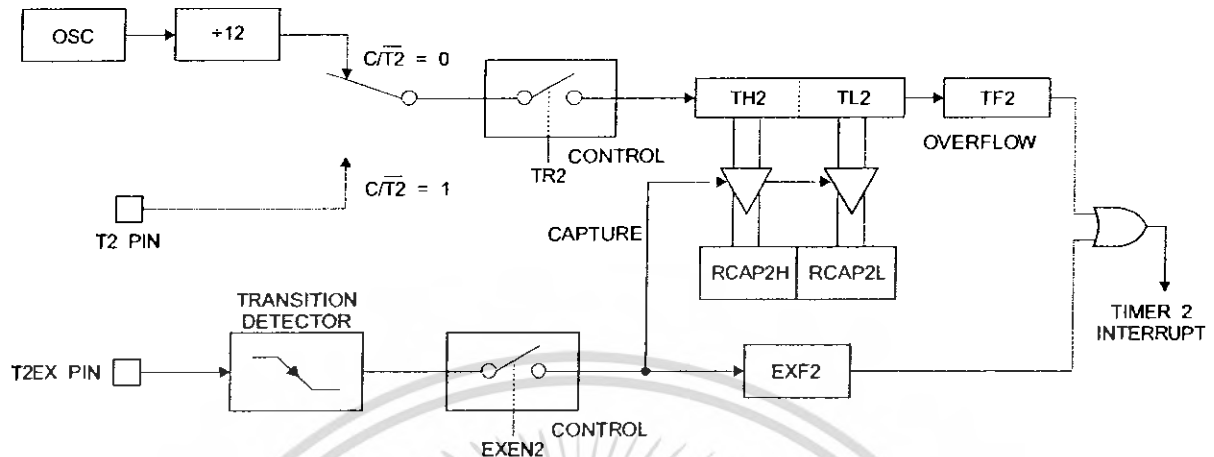


Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture Mode RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls

the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 Auto Reload Mode (DCEN = 0)

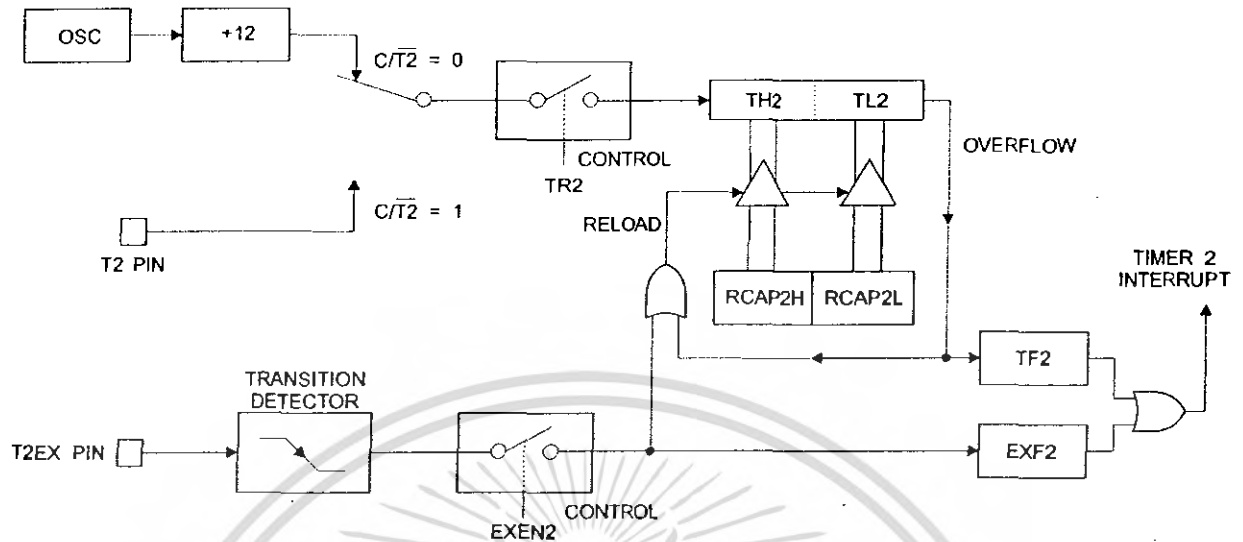


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H						Reset Value = XXXX XX00B		
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	T2OE	DCEN

Symbol	Function
—	Not implemented, reserved for future
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

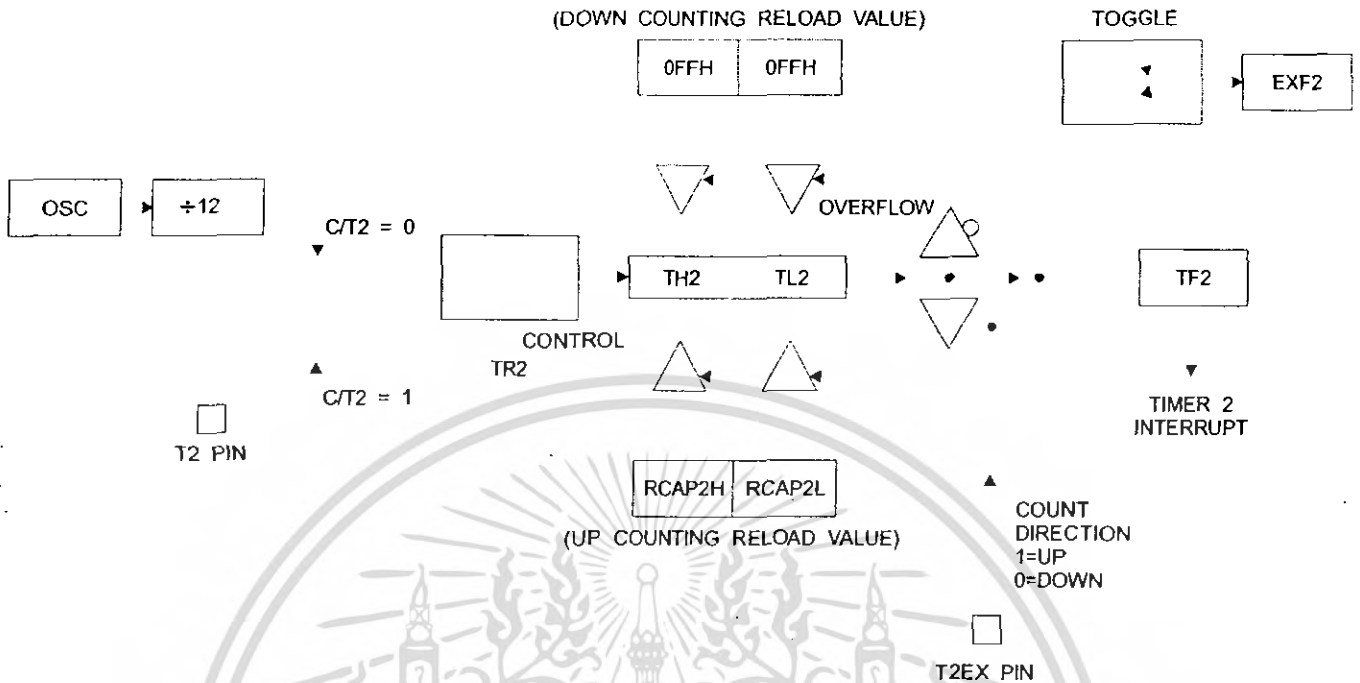
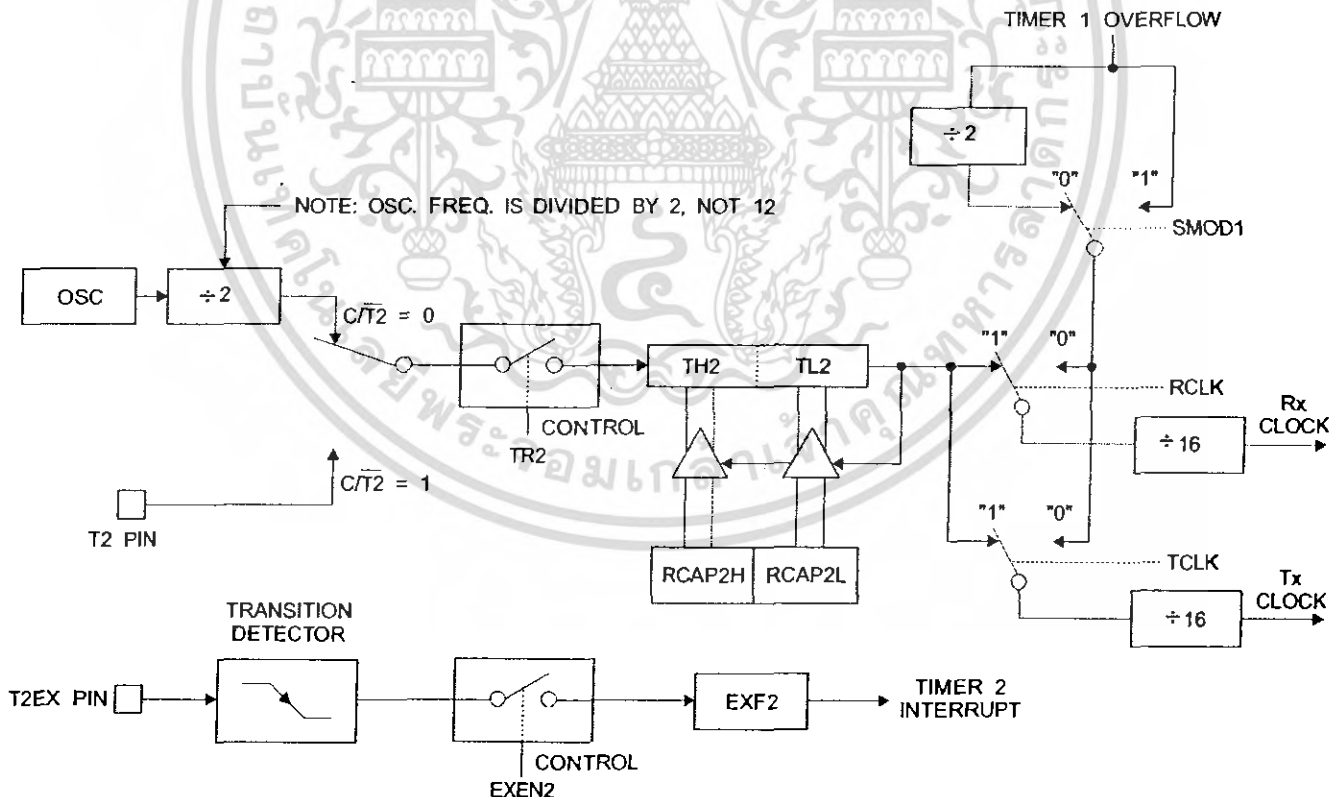


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($CP/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it

increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

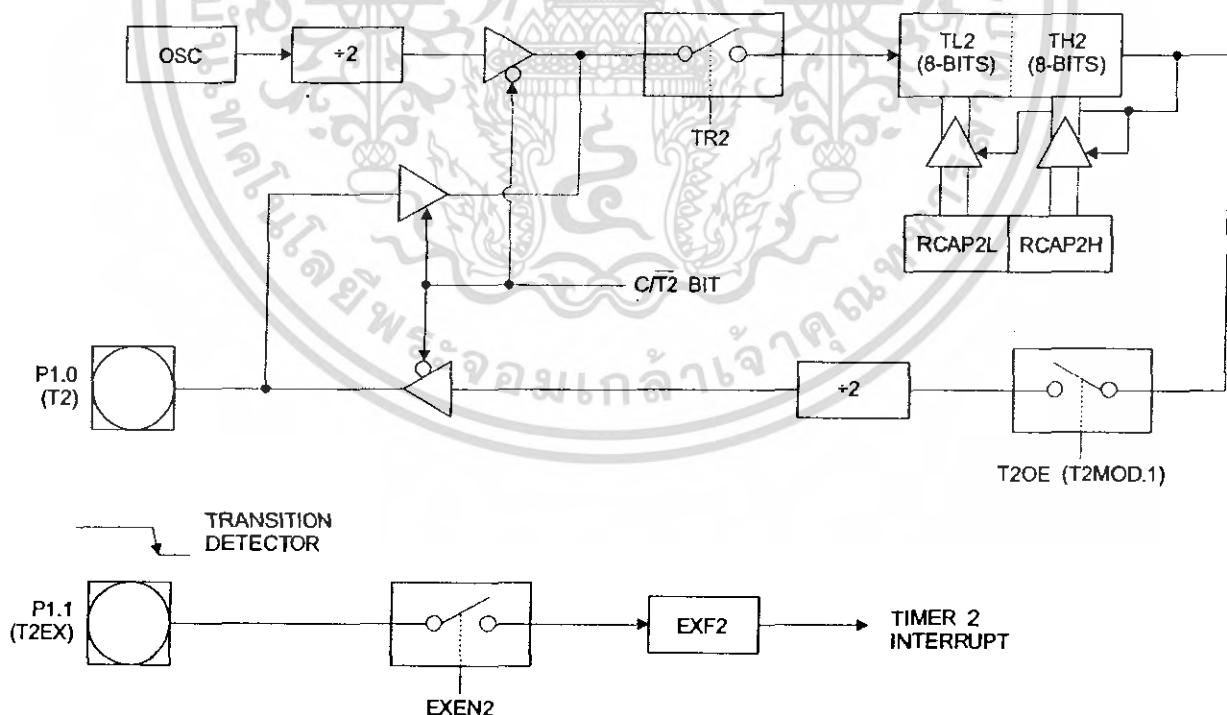
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running ($TR2 = 1$) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-out Mode



Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit $C/\overline{T}2$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

UART

The UART in the AT89C52 operates the same way as the UART in the AT89C51.

Interrupts

The AT89C52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However,

the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

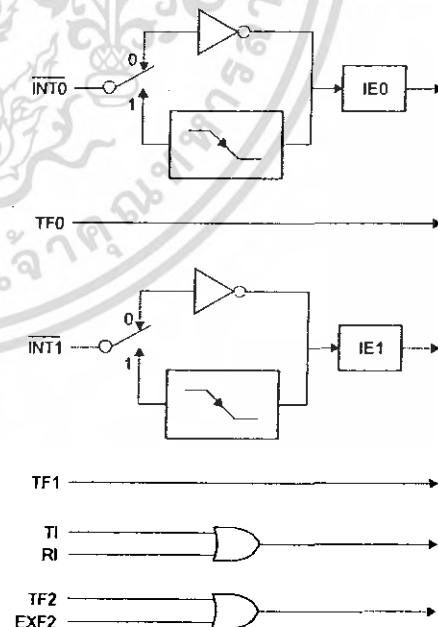
Table 5. Interrupt Enable (IE) Register

(MSB)				(LSB)			
EA	–	ET2	ES	ET1	EX1	ET0	EX0
Enable Bit = 1 enables the interrupt.							
Enable Bit = 0 disables the interrupt.							

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

Figure 6. Interrupt Sources



Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

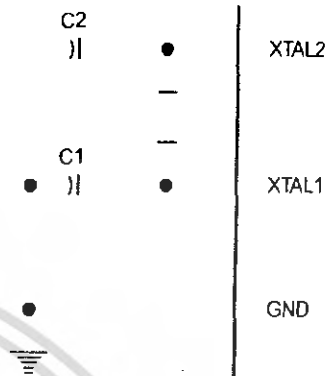
Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

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

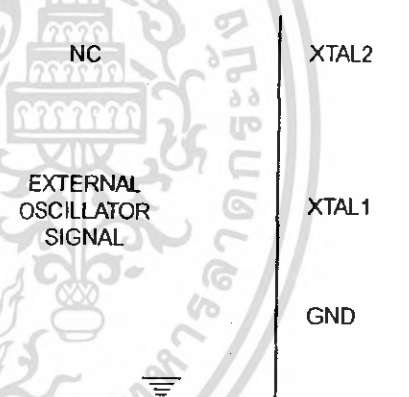
is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 7. Oscillator Connections



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

Figure 8. External Clock Drive Configuration



Status of External Pins During Idle and Power-down Modes

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

Program Memory Lock Bits

The AT89C52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Lock Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

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

Programming the Flash

The AT89C52 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The Low-voltage programming mode provides a convenient way to program the AT89C52 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C52 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	$V_{PP} = 12V$	$V_{PP} = 5V$
Top-side Mark	AT89C52 xxxx yyww	AT89C52 xxxx - 5 yyww

	$V_{PP} = 12V$	$V_{PP} = 5V$
Signature	(030H) = 1EH (031H) = 52H (032H) = FFH	(030H) = 1EH (031H) = 52H (032H) = 05H

The AT89C52 code memory array is programmed byte-by-byte in either programming mode. To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

Programming Algorithm Before programming the AT89C52, the address, data and control signals should be set up according to the Flash programming mode table and Figure 9 and Figure 10. To program the AT89C52, take the following steps.

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

Data Polling The AT89C52 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on PO.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

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

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

Chip Erase The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/ $\overline{PROG}$ low for 10 ms. The code array is written with all 1s. The chip erase operation must be executed before the code memory can be reprogrammed.

Reading the Signature Bytes The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

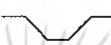
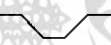
- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 52H indicates 89C52
- (032H) = FFH indicates 12V programming
- (032H) = 05H indicates 5V programming

Programming Interface

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

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

Flash Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{pp}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H
	Bit - 2	H	L		H/12V	H	L	L
	Bit - 3	H	L		H/12V	H	L	L
Chip Erase	H	L	 (1)	H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 9. Programming the Flash Memory

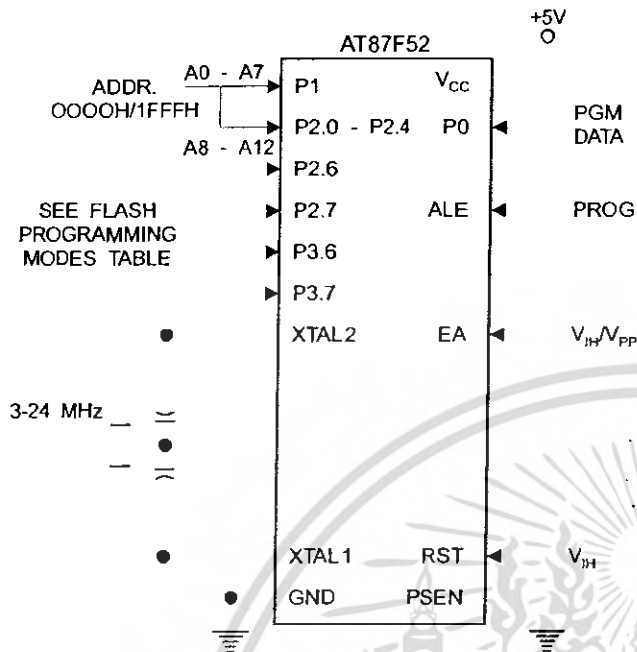
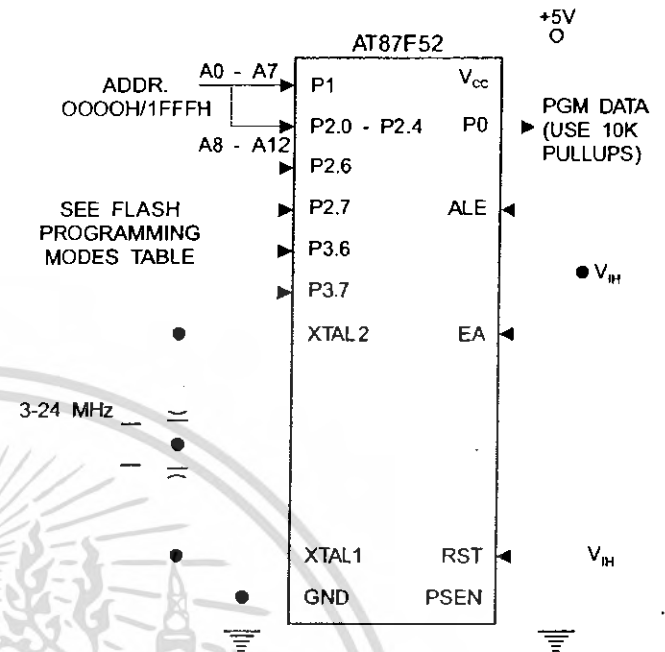


Figure 10. Verifying the Flash Memory



Flash Programming and Verification Characteristics

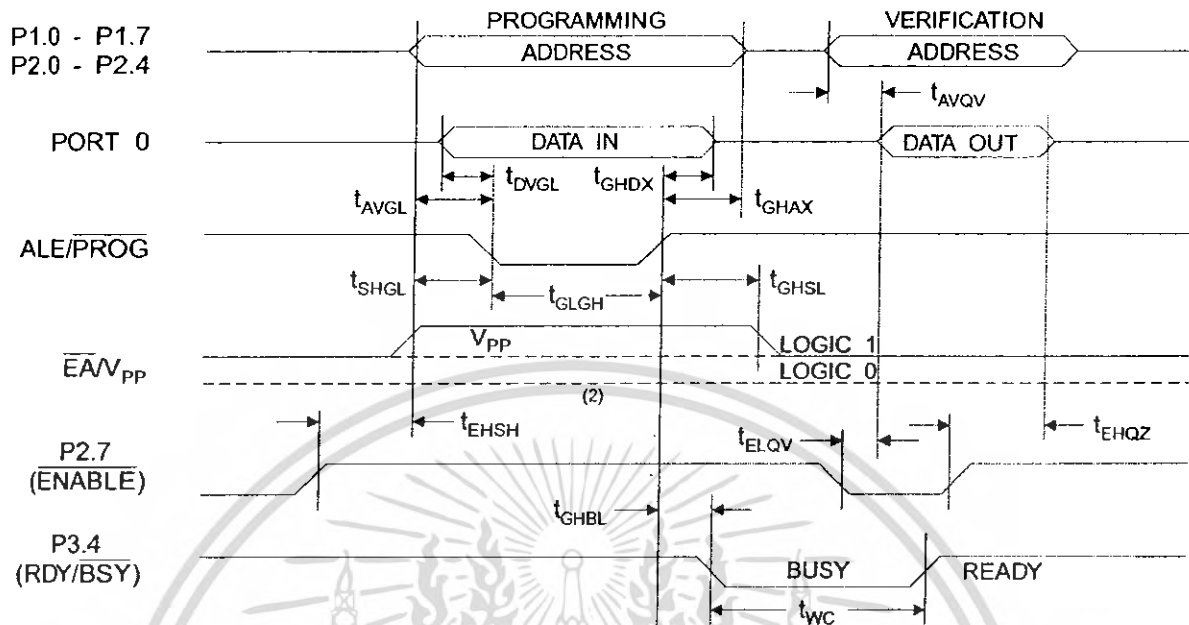
$T_A = 0^\circ\text{C to } 70^\circ\text{C}$, $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
$V_{PP}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}^{(1)}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHS}	P2.7 (ENABLE) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		μs
$t_{GHS}^{(1)}$	V_{PP} Hold after $\overline{PROG}$	10		μs
t_{GLGH}	$\overline{PROG}$ Width	1	110	μs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms

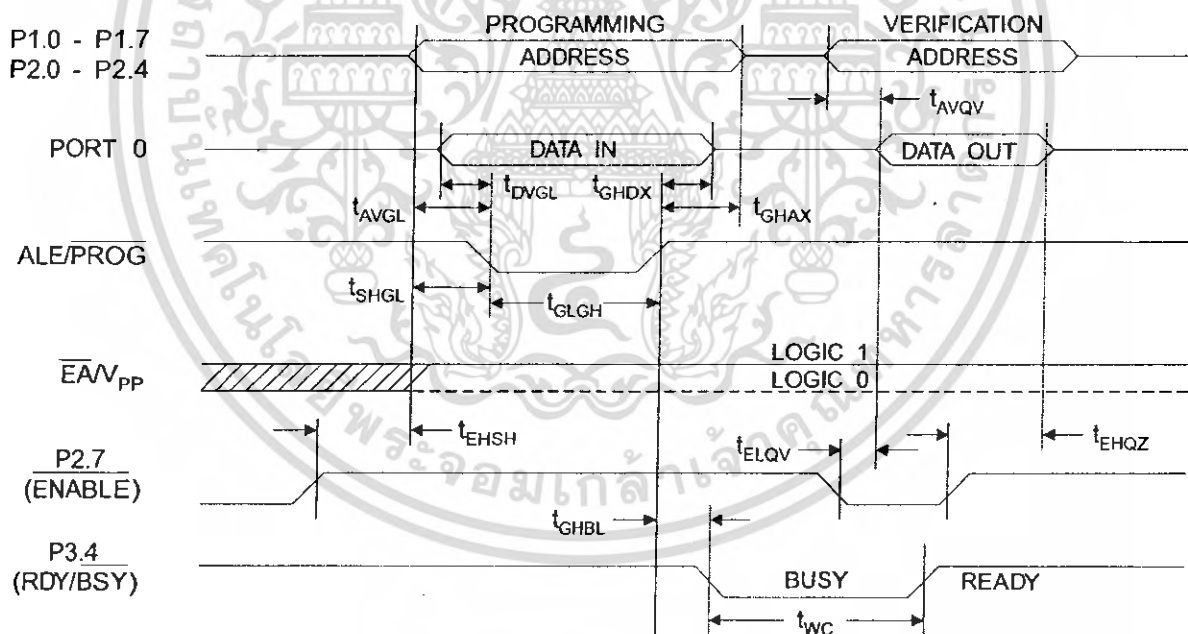
Note: 1. Only used in 12-volt programming mode.



Flash Programming and Verification Waveforms - High-voltage Mode ($V_{pp}=12V$)



Flash Programming and Verification Waveforms - Low-voltage Mode ($V_{pp}=5V$)



Absolute Maximum Ratings*

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

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

DC Characteristics

The values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 5.0\text{V} \pm 20\%$, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low-voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low-voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
V_{OL1}	Output Low-voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_L	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_U	Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽¹⁾	$V_{CC} = 6\text{V}$		100	μA
		$V_{CC} = 3\text{V}$		40	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port:

Port 0: 26 mA Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V.



AC Characteristics

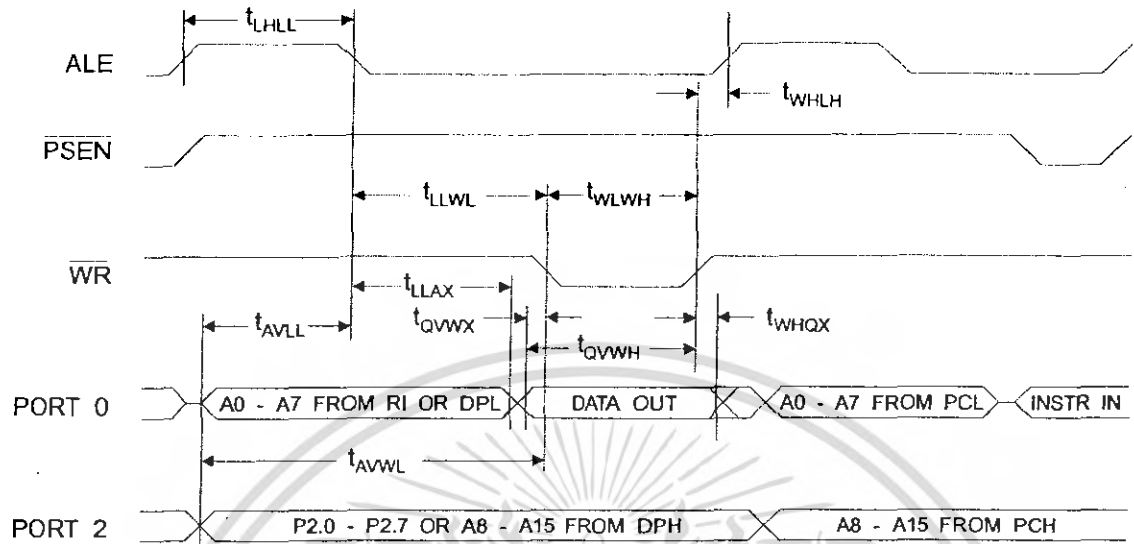
Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

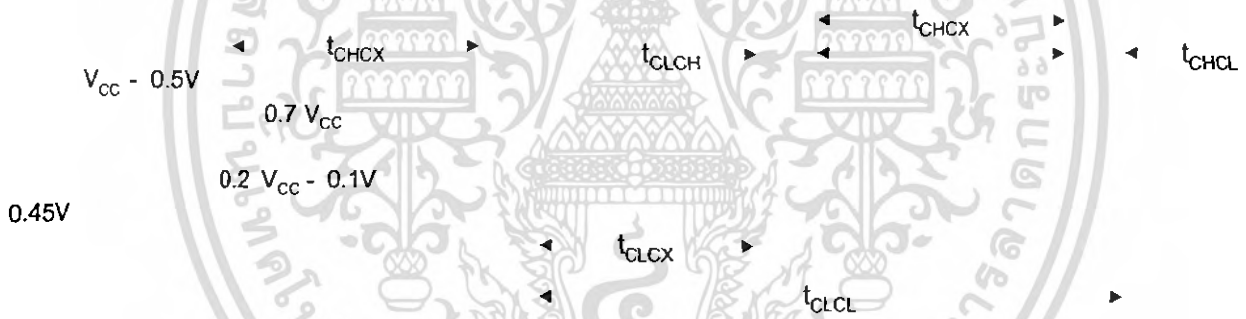
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{\text{CLCL}}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{\text{CLCL}}-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{\text{CLCL}}-20$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

19

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

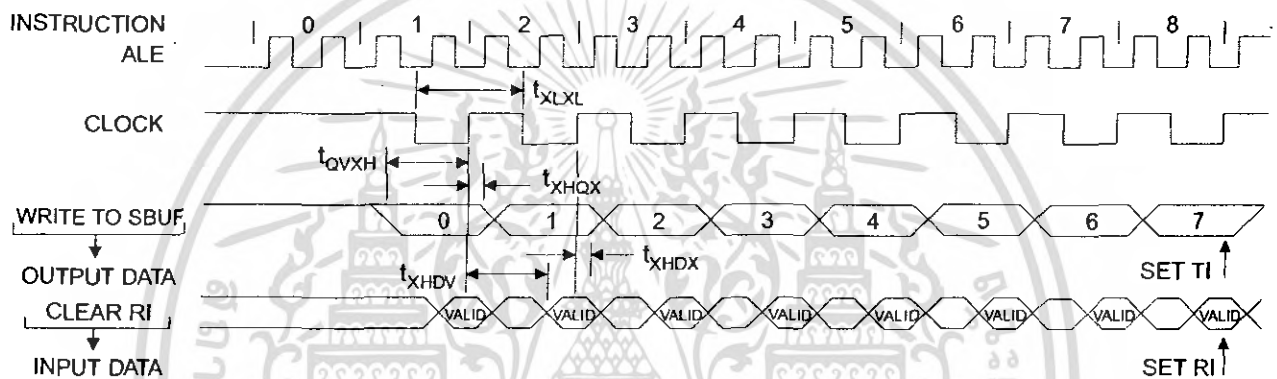
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

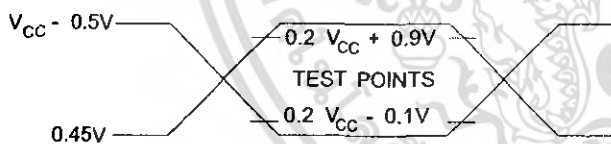
The values in this table are valid for $V_{CC} = 5.0V \pm 20\%$ and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHGX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-117$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms

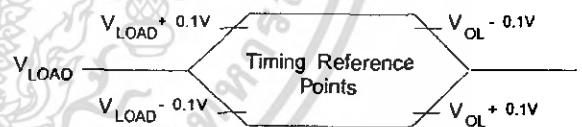


AC Testing Input/Output Waveforms⁽¹⁾



Note: 1. AC Inputs during testing are driven at $V_{CC} - 0.5V$ for a logic 1 and $0.45V$ for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms⁽¹⁾



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

Ordering Information

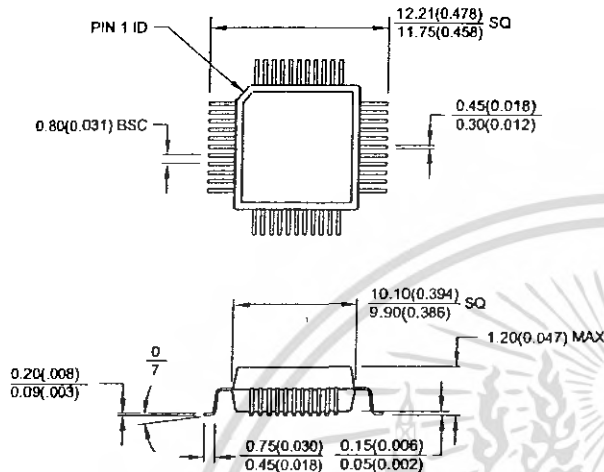
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ±20%	AT89C52-12AC	44A	Commercial (0°C to 70°C)
		AT89C52-12JC	44J	
		AT89C52-12PC	40P6	
		AT89C52-12QC	44Q	
		AT89C52-12AI	44A	Industrial (-40°C to 85°C)
		AT89C52-12JI	44J	
		AT89C52-12PI	40P6	
		AT89C52-12QI	44Q	
16	5V ±20%	AT89C52-16AC	44A	Commercial (0°C to 70°C)
		AT89C52-16JC	44J	
		AT89C52-16PC	40P6	
		AT89C52-16QC	44Q	
		AT89C52-16AI	44A	Industrial (-40°C to 85°C)
		AT89C52-16JI	44J	
		AT89C52-16PI	40P6	
		AT89C52-16QI	44Q	
20	5V ±20%	AT89C52-20AC	44A	Commercial (0°C to 70°C)
		AT89C52-20JC	44J	
		AT89C52-20PC	40P6	
		AT89C52-20QC	44Q	
		AT89C52-20AI	44A	Industrial (-40°C to 85°C)
		AT89C52-20JI	44J	
		AT89C52-20PI	40P6	
		AT89C52-20QI	44Q	
24	5V ±20%	AT89C52-24AC	44A	Commercial (0°C to 70°C)
		AT89C52-24JC	44J	
		AT89C52-24PC	40P6	
		AT89C52-24QC	44Q	
		AT89C52-24AI	44A	Industrial (-40°C to 85°C)
		AT89C52-24JI	44J	
		AT89C52-24PI	40P6	
		AT89C52-24QI	44Q	

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-headed Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

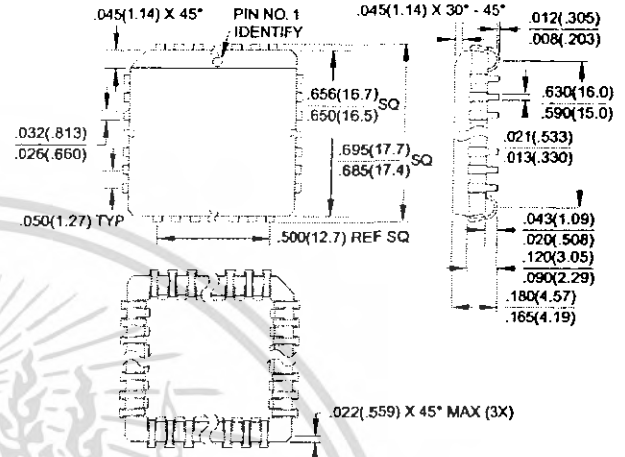
Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
 Dimensions in Millimeters and (Inches)*
 JEDEC STANDARD MS-026 ACB

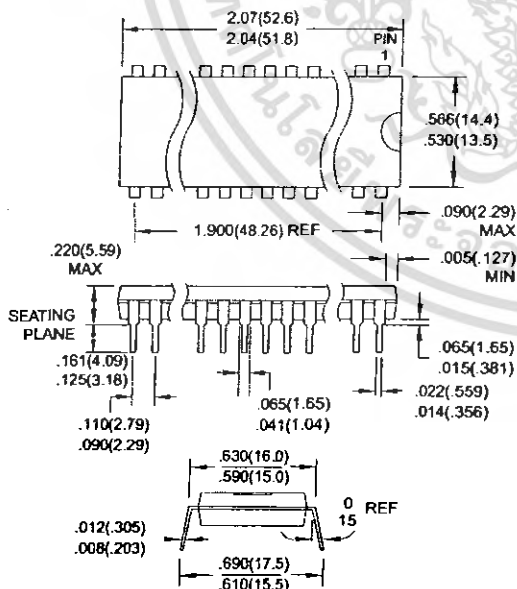


Controlling dimension: millimeters

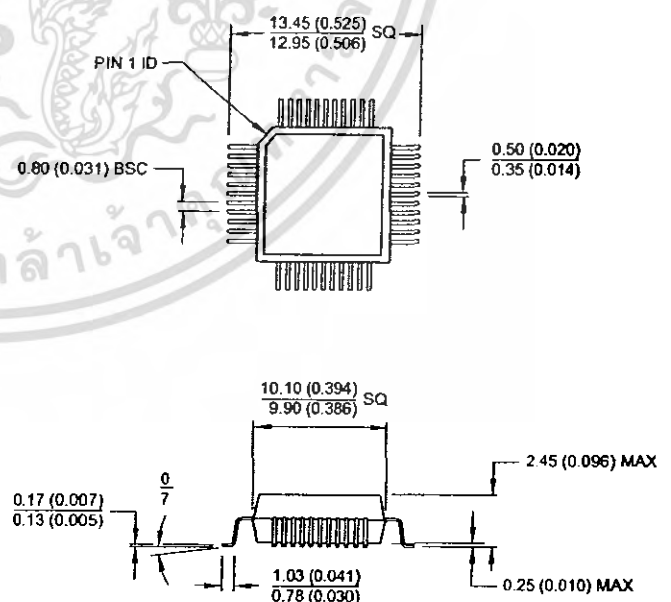
44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
 Dimensions in Inches and (Millimeters)
 JEDEC STANDARD MS-018 AC



40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
 Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
 Dimensions in Millimeters and (Inches)*
 JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters





Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs

1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

© Atmel Corporation 1999.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

0313H-02/00xM

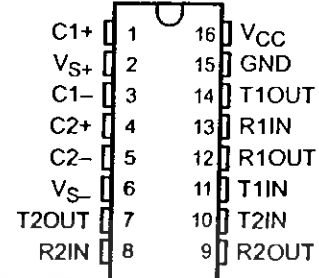
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS0471 – FEBRUARY 1989 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
 - TIA/EIA-232-F
 - Battery-Powered Systems
 - Terminals
 - Modems
 - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
–40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

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



Copyright © 2002, Texas Instruments Incorporated

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I

DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

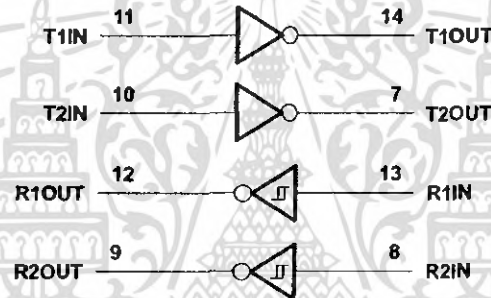
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS0471 – FEBRUARY 1989 – REVISED OCTOBER 2002

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

Input supply voltage range, V_{CC} (see Note 1)	–0.3 V to 6 V
Positive output supply voltage range, V_{S+}	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, V_{S-}	–0.3 V to –15 V
Input voltage range, V_I : Driver	–0.3 V to $V_{CC} + 0.3$ V
Receiver	±30 V
Output voltage range, V_O : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	–0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, θ_{JA} (see Note 2): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	–65°C to 150°C

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

NOTE 1: All voltage values are with respect to network ground terminal.

2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN, T2IN)	2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			±30	V
T_A	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	–40	85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 3 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP [‡]	MAX	UNIT
I_{CC} Supply current	$V_{CC} = 5.5$ V, All outputs open, $T_A = 25^\circ\text{C}$		8	10	mA

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

NOTE 3: Test conditions are C1–C4 = 1 μF at $V_{CC} = 5 \text{ V} \pm 0.5 \text{ V}$.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I

DUAL EIA-232 DRIVERS/RECEIVERS

SLLS0471 – FEBRUARY 1989 – REVISED OCTOBER 2002

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	T1OUT, T2OUT R _L = 3 kΩ to GND	5	7		V
V _{OL}	Low-level output voltage‡	T1OUT, T2OUT R _L = 3 kΩ to GND		-7	-5	V
r _o	Output resistance	T1OUT, T2OUT V _{S+} = V _{S-} = 0, V _O = ±2 V	300			Ω
I _{OS} §	Short-circuit output current	T1OUT, T2OUT V _{CC} = 5.5 V, V _O = 0		±10		mA
I _{IS}	Short-circuit input current	T1IN, T2IN V _I = 0			200	μA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R _L = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/μs
	Data rate	One TOUT switching		120		kbit/s

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	R1OUT, R2OUT I _{OH} = -1 mA	3.5			V
V _{OL}	Low-level output voltage‡	R1OUT, R2OUT I _{OL} = 3.2 mA			0.4	V
V _{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C		1.7	2.4	V
V _{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C	0.8	1.2		V
V _{hys}	Input hysteresis voltage	R1IN, R2IN V _{CC} = 5 V	0.2	0.5	1	V
r _i	Receiver input resistance	R1IN, R2IN V _{CC} = 5, T _A = 25°C	3	5	7	kΩ

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3 and Figure 1)

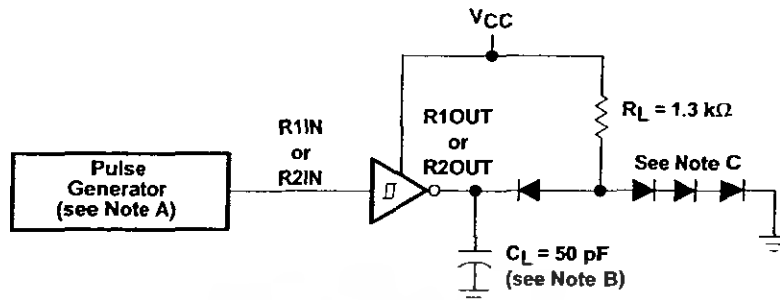
PARAMETER		TYP	UNIT
t _{PLH(R)}	Receiver propagation delay time, low- to high-level output	500	ns
t _{PHL(R)}	Receiver propagation delay time, high- to low-level output	500	ns

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

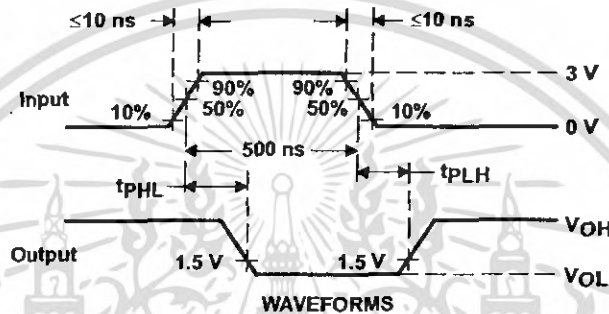


POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT



WAVEFORMS

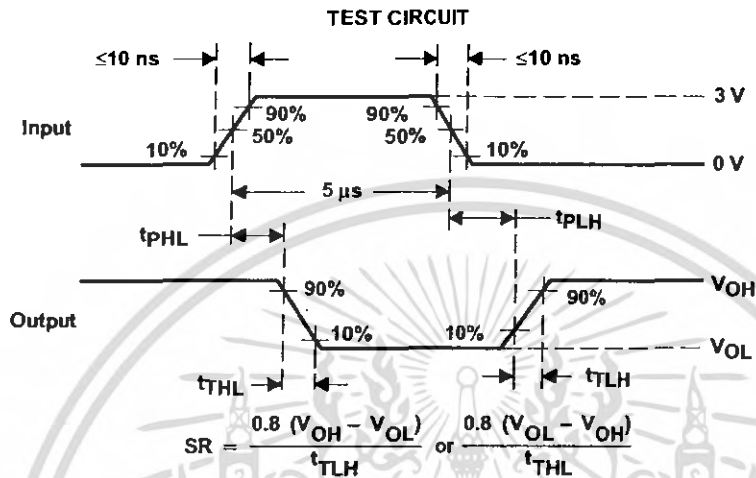
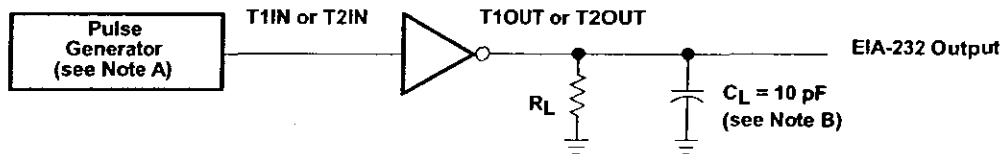
- NOTES: A. The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.
B. C_L includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

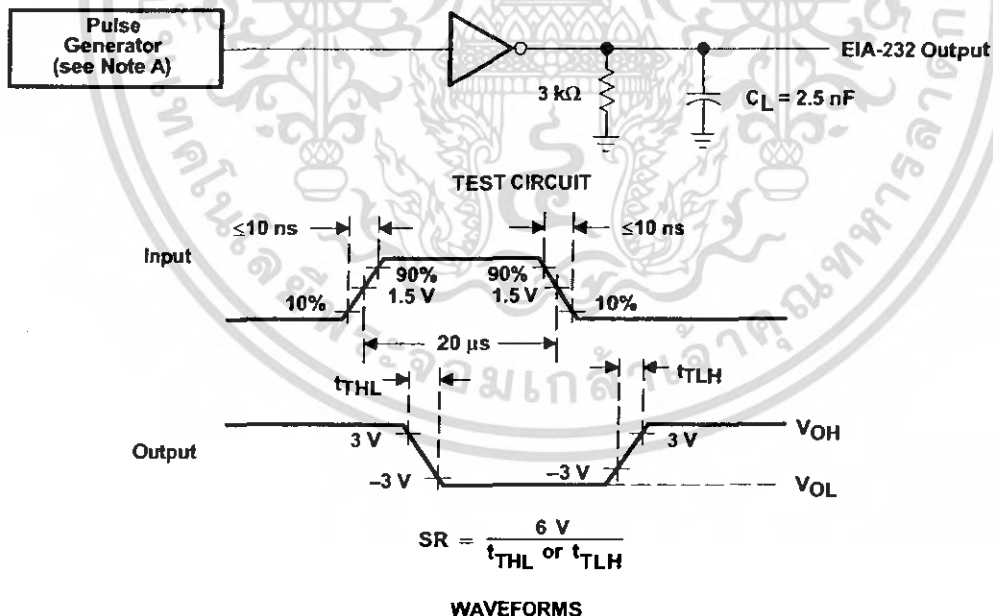
SLLS0471 - FEBRUARY 1989 - REVISED OCTOBER 2002

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.
B. C_L includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5- μ s Input)



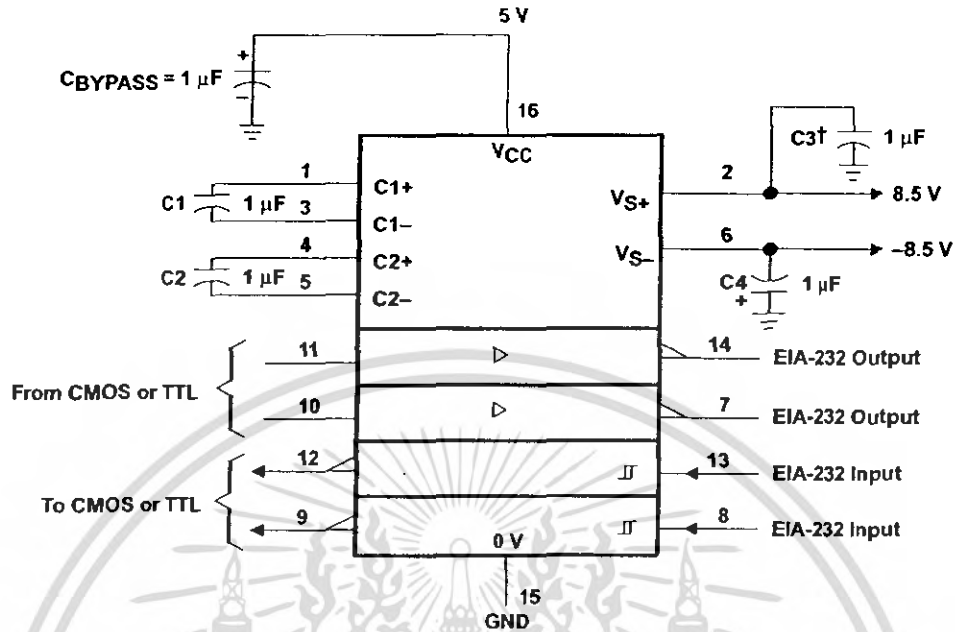
NOTE A: The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.

Figure 3. Test Circuit and Waveforms for t_{THL} and t_{TLH} Measurements (20- μ s Input)



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

APPLICATION INFORMATION



† C3 can be connected to VCC or GND.

Figure 4. Typical Operating Circuit



TEXAS
INSTRUMENTS

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TLC272, TLC272A, TLC272B, TLC272Y, TLC277 LinCMOS™ PRECISION DUAL OPERATIONAL AMPLIFIERS

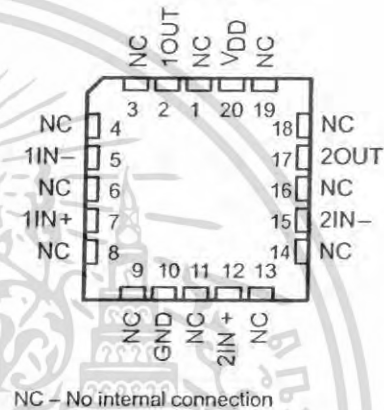
SLOS091E – OCTOBER 1987 – REVISED FEBRUARY 2002

- **Trimmed Offset Voltage:**
TLC277 . . . 500 μV Max at 25°C,
 $V_{\text{DD}} = 5\text{ V}$
- **Input Offset Voltage Drift . . . Typically**
0.1 $\mu\text{V}/\text{Month}$, Including the First 30 Days
- **Wide Range of Supply Voltages Over Specified Temperature Range:**
0°C to 70°C . . . 3 V to 16 V
–40°C to 85°C . . . 4 V to 16 V
–55°C to 125°C . . . 4 V to 16 V
- **Single-Supply Operation**
- **Common-Mode Input Voltage Range**
Extends Below the Negative Rail (C-Suffix,
I-Suffix types)
- **Low Noise . . . Typically 25 nV/ $\sqrt{\text{Hz}}$ at**
 $f = 1\text{ kHz}$
- **Output Voltage Range Includes Negative Rail**
- **High Input impedance . . . 10¹² Ω Typ**
- **ESD-Protection Circuitry**
- **Small-Outline Package Option Also Available in Tape and Reel**
- **Designed-In Latch-Up Immunity**

D, JG, P, OR PW PACKAGE
(TOP VIEW)



FK PACKAGE
(TOP VIEW)



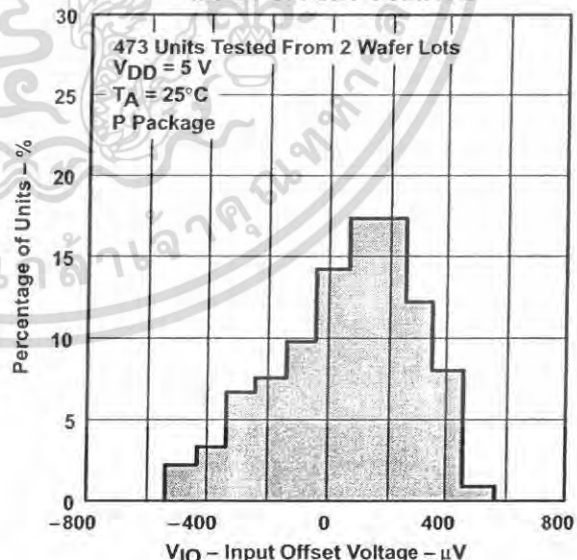
description

The TLC272 and TLC277 precision dual operational amplifiers combine a wide range of input offset voltage grades with low offset voltage drift, high input impedance, low noise, and speeds approaching those of general-purpose BiFET devices.

These devices use Texas Instruments silicon-gate LinCMOS™ technology, which provides offset voltage stability far exceeding the stability available with conventional metal-gate processes.

The extremely high input impedance, low bias currents, and high slew rates make these cost-effective devices ideal for applications previously reserved for BiFET and NFET products. Four offset voltage grades are available (C-suffix and I-suffix types), ranging from the low-cost TLC272 (10 mV) to the high-precision TLC277 (500 μV). These advantages, in combination with good common-mode rejection and supply voltage rejection, make these devices a good choice for new state-of-the-art designs as well as for upgrading existing designs.

DISTRIBUTION OF TLC277
INPUT OFFSET VOLTAGE



LinCMOS is a trademark of Texas Instruments.

Copyright © 2002, Texas Instruments Incorporated

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

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับลูกค้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องส่งคืนเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TLC272, TLC272A, TLC272B, TLC272Y, TLC277 LinCMOS™ PRECISION DUAL OPERATIONAL AMPLIFIERS

SLOS091E – OCTOBER 1987 – REVISED FEBRUARY 2002

description (continued)

AVAILABLE OPTIONS

T _A	V _{IO} max AT 25°C	PACKAGED DEVICES					CHIP FORM (Y)
		SMALL OUTLINE (D)	CHIP CARRIER (FK)	CERAMIC DIP (JG)	PLASTIC DIP (P)	TSSOP (PW)	
0°C to 70°C	500 μ V	TLC277CD	—	—	TLC277CP	—	—
	2 mV	TLC272BCD	—	—	TLC272BCP	—	—
	5 mV	TLC272ACD	—	—	TLC272ACP	—	—
	10mV	TLC272CD	—	—	TLC272CP	TLC272CPW	TLC272Y
–40°C to 85°C	500 μ V	TLC277ID	—	—	TLC277IP	—	—
	2 mV	TLC272BID	—	—	TLC272BIP	—	—
	5 mV	TLC272AID	—	—	TLC272AIP	—	—
	10 mV	TLC272ID	—	—	TLC272IP	—	—

The D package is available taped and reeled. Add R suffix to the device type (e.g., TLC277CDR).

In general, many features associated with bipolar technology are available on LinCMOS™ operational amplifiers without the power penalties of bipolar technology. General applications such as transducer interfacing, analog calculations, amplifier blocks, active filters, and signal buffering are easily designed with the TLC272 and TLC277. The devices also exhibit low voltage single-supply operation, making them ideally suited for remote and inaccessible battery-powered applications. The common-mode input voltage range includes the negative rail.

A wide range of packaging options is available, including small-outline and chip carrier versions for high-density system applications.

The device inputs and outputs are designed to withstand –100-mA surge currents without sustaining latch-up.

The TLC272 and TLC277 incorporate internal ESD-protection circuits that prevent functional failures at voltages up to 2000 V as tested under MIL-STD-883C, Method 3015.2; however, care should be exercised in handling these devices as exposure to ESD may result in the degradation of the device parametric performance.

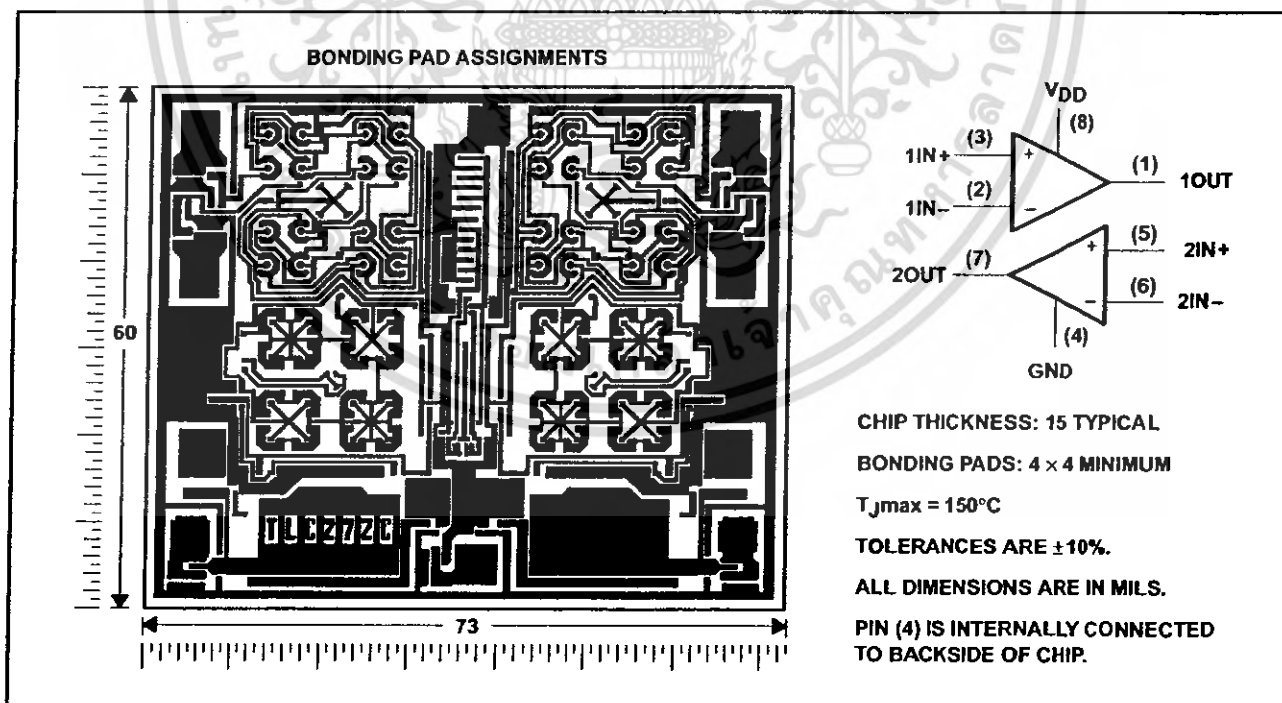
The C-suffix devices are characterized for operation from 0°C to 70°C. The I-suffix devices are characterized for operation from –40°C to 85°C. The M-suffix devices are characterized for operation over the full military temperature range of –55°C to 125°C.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

SLOS091E - OCTOBER 1987 - REVISED FEBRUARY 2002

This chip, when properly assembled, displays characteristics similar to the TLC272C. Thermal compression or ultrasonic bonding may be used on the doped-aluminum bonding pads. Chips may be mounted with conductive epoxy or a gold-silicon preform.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมไมโครคอนโทรลเลอร์

```
#include<reg51.h>

sbit a0=P0^0;
sbit a1=P0^1;
sbit a2=P0^2;
sbit a3=P0^3;
sbit a4=P0^4;
sbit a5=P0^5;
sbit a6=P0^6;
sbit a7=P0^7;
sbit b0=P1^0;
sbit b1=P1^1;
sbit b2=P1^2;
sbit b3=P1^3;
sbit b4=P1^4;
sbit b5=P1^5;
sbit b6=P1^6;
sbit b7=P1^7;
sbit c0=P2^0;
sbit c1=P2^1;
sbit c2=P2^2;
sbit c3=P2^3;
sbit c4=P2^4;
sbit c5=P2^5;
sbit c6=P2^6;
sbit c7=P2^7;
sbit d2=P3^2;

sbit d3=P3^3;
sbit d4=P3^4;
sbit d5=P3^5;
sbit d6=P3^6;
sbit d7=P3^7;

int a[35]=0;

void delay(unsigned int de)
{ unsigned int x,y;
  for(x=0;x<de;x++){
    for(y=0;y<100;y++);
  }
}

void inter4(void)interrupt 4
{ TI=0; }

void main(void){
  TMOD=0x20;
  SCON=0x50;
  TH1=0xFD;
  TL1=0xFD;
  EA=1; ES=1;
  RI=0; TI=0; TR1=1;

  while(1){
    if(a0==0){
      delay(400);
      if(a0==0){
        if(a[0]==0){
          a[0]=1;
          SBUF=0;
        }
      }
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} }

if(a1==0){
    delay(400);
    if(a1==0){
        if(a[1]==0){
            a[1]=1;
            SBUF=1;}
        }}
    if(a2==0){
        delay(400);
        if(a2==0){
            if(a[2]==0){
                a[2]=1;
                SBUF=2;}
            }}
        if(a3==0){
            delay(400);
            if(a3==0){
                if(a[3]==0){
                    a[3]=1;
                    SBUF=3;}
                }}
            if(a4==0){
                delay(400);
                if(a4==0){
                    if(a[4]==0){
                        a[4]=1;
                        SBUF=4;}
                    }}
                }}
            if(a5==0){
                delay(400);
                if(a5==0){
                    if(a[5]==0){
                        a[5]=1;
                        SBUF=5;}
                    }}
                if(a6==0){
                    delay(400);
                    if(a6==0){
                        if(a[6]==0){
                            a[6]=1;
                            SBUF=6;}
                        }}
                    if(a0==1){
                        if(a[0]==1){
                            a[0]=0;
                            SBUF=31;}
                        }
                    if(a1==1){
                        if(a[1]==1){
                            a[1]=0;
                            SBUF=32;}
                        }
                    if(a2==1){
                        if(a[2]==1){
                            a[2]=0;
                            SBUF=33;}}
                }}
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(a3==1){
    if(a[3]==1){
        a[3]=0;
        SBUF=34;}
    }
    if(a4==1){
        if(a[4]==1){
            a[4]=0;
            SBUF=35;}
        }
        if(a5==1){
            if(a[5]==1){
                a[5]=0;
                SBUF=36;}
            }
            if(a6==1){
                if(a[6]==1){
                    a[6]=0;
                    SBUF=37;}
                }
                if(b0==0){
                    delay(400);
                }
                if(b0==0){
                    if(a[8]==0){
                        a[8]=1;
                        SBUF=8;}
                    }
                    if(b1==0){
                        delay(400);
                    }
                    if(b1==0){
                        if(a[9]==0){
                            a[9]=1;
                            SBUF=9;}
                        }
                        if(b2==0){
                            delay(400);
                        }
                        if(b2==0){
                            if(a[10]==0){
                                a[10]=1;
                                SBUF=10;}
                            }
                            if(b3==0){
                                delay(400);
                            }
                            if(b3==0){
                                if(a[11]==0){
                                    a[11]=1;
                                    SBUF=11;}
                                }
                                if(b4==0){
                                    delay(400);
                                }
                                if(b4==0){
                                    if(a[12]==0){
                                        a[12]=1;
                                        SBUF=12;}
                                    }
                                    if(b5==0){
                                        delay(400);
                                    }
                                    if(b5==0){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(a[13]==0){
    SBUF=42;}
a[13]=1;
    }
    SBUF=13;}
    }
    if(b6==0){
        delay(400);
        SBUF=43;}
    if(b6==0){
    }
    if(a[14]==0){
        a[14]=1;
        SBUF=14;}
    }
    if(b0==1){
    if(a[8]==1){
        a[8]=0;
        SBUF=39;}
    }
    if(b1==1){
    if(a[9]==1){
        a[9]=0;
        SBUF=40;}
    }
    if(b2==1){
    if(a[10]==1){
        a[10]=0;
        SBUF=41;}
    }
    if(b3==1){
    if(a[11]==1){
        a[11]=0;
        SBUF=42;}
    }
    if(b4==1){
    if(a[12]==1){
        a[12]=0;
        SBUF=43;}
    }
    if(b5==1){
    if(a[13]==1){
        a[13]=0;
        SBUF=44;}
    }
    if(b6==1){
    if(a[14]==1){
        a[14]=0;
        SBUF=45;}
    }
    if(c0==0){
        delay(400);
        if(c0==0){
            if(a[16]==0){
                a[16]=1;
                SBUF=16;}
            }
            if(c1==0){
                delay(400);
                if(c1==0){
                    if(a[17]==0){
                        a[17]=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SBUF=17;}}
}
if(c2==0){
    delay(400);
if(c2==0){
if(a[18]==0){
    a[18]=1;
    SBUF=18;}}
}}
if(c3==0){
    delay(400);
if(c3==0){
if(a[19]==0){
    a[19]=1;
    SBUF=19;}}
}}
if(c4==0){
    delay(400);
if(c4==0){
if(a[20]==0){
    a[20]=1;
    SBUF=20;}}
}}
if(c5==0){
    delay(400);
if(c5==0){
if(a[21]==0){
    a[21]=1;
    SBUF=21;}}
}}
}}
if(c6==0){
    delay(400);
if(c6==0){
if(a[22]==0){
    a[22]=1;
    SBUF=22;}}
}}
if(c0==1){
if(a[16]==1){
    a[16]=0;
    SBUF=47;}}
}
if(c1==1){
if(a[17]==1){
    a[17]=0;
    SBUF=48;}}
}}
if(c2==1){
if(a[18]==1){
    a[18]=0;
    SBUF=49;}}
}
}
if(c3==1){
if(a[19]==1){
    a[19]=0;
    SBUF=50;}}
}
if(c4==1){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(a[20]==1){
    a[20]=0;
    SBUF=51;}
}
if(c5==1){
if(a[21]==1){
    a[21]=0;
    SBUF=52;}
}
if(c6==1){
if(a[22]==1){
    a[22]=0;
    SBUF=53;}
}
if(d2==0){
    delay(400);
if(d2==0){
if(a[24]==0){
    a[24]=1;
    SBUF=24;}
}}
if(d3==0){
    delay(400);
if(d3==0){
if(a[25]==0){
    a[25]=1;
    SBUF=25;}
}}
if(d4==0){
    delay(400);
if(d4==0){
if(a[26]==0){
    a[26]=1;
    SBUF=26;}
}}
if(d5==0){
if(a[27]==0){
    a[27]=1;
    SBUF=27;}
}}
if(d6==0){
    delay(400);
if(d6==0){
if(a[28]==0){
    a[28]=1;
    SBUF=28;}
}}
if(d7==0){
    delay(400);
if(d7==0){
if(a[29]==0){
    a[29]=1;
    SBUF=29;}
}}
if(d2==1){
if(a[24]==1){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SBUF=56;}

if(c7==1){
    if(a[7]==1){
        a[7]=0;
        SBUF=38;}
    }
    if(a7==0){
        if(a[15]==0){
            a[15]=1;
            SBUF=7;}
        }
        if(a7==1){
            if(a[15]==1){
                a[15]=0;
                SBUF=15;}
            }}}
}
}
}

if(b[26]==1){
    a[26]=0;
    SBUF=57;}

if(b[27]==1){
    a[27]=0;
    SBUF=58;}

if(b[28]==1){
    a[28]=0;
    SBUF=59;}

```

โปรแกรมวิชาเว็บติก

```
Dim x, y, z As Integer
Dim pic, car As Variant
Dim select_com As Integer
Dim buff As String

Private Sub Command1_Click()
    If Picture22.Picture=Then
Picture22.Picture=LoadPicture
End If
End Sub

Private Sub Command2_Click()
    Picture22.Picture =LoadPicture("")
End Sub

Private Sub Command3_Click()
    If Picture23.Picture = Then
Picture23.Picture = LoadPicture
End If
End Sub

Private Sub Command4_Click()
    Picture23.Picture = LoadPicture("")
End Sub

Private Sub Command5_Click()
    If Picture24.Picture = 0 Then
        Picture24.Picture = LoadPicture
End If
End Sub
```

```
Private Sub Command6_Click()
    Picture24.Picture = LoadPicture("")
End Sub

Private Sub Command7_Click()
    If Picture25.Picture = 0 Then
        Picture25.Picture = LoadPicture
End If
End Sub

Private Sub Command8_Click()
    Picture25.Picture = LoadPicture("")
End Sub

Private Sub Command9_Click()
    If Picture26.Picture = 0 Then
        Picture26.Picture = LoadPicture
End If
End Sub

Private Sub Command10_Click()
    Picture26.Picture = LoadPicture("")
End Sub

Private Sub Command11_Click()
    If Picture27.Picture = 0 Then
        Picture27.Picture = LoadPicture
End If
End Sub

Private Sub Command12_Click()
    Picture27.Picture = LoadPicture("")
End Sub

Private Sub Command13_Click()
    If Picture28.Picture = 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Picture28.Picture = LoadPicture	Picture14.Visible = False
End If	Picture15.Visible = False
End Sub	Picture16.Visible = False
Private Sub Command14_Click()	Picture17.Visible = False
Picture28.Picture = LoadPicture("")	Picture18.Visible = False
End Sub	Picture19.Visible = False
Private Sub Timer1_Timer()	Picture20.Visible = False
Label5.Caption = Date	Picture21.Visible = False
Label6.Caption = Time	Picture22.Visible = True
End Sub	Picture23.Visible = True
Private Sub Form_Load()	Picture24.Visible = True
MSComm1.CommPort = 4	Picture25.Visible = True
MSComm1.Settings = "9600,N,8,1"	Picture26.Visible = True
MSComm1.InputLen = 0	Picture27.Visible = True
MSComm1.RThreshold = 1	Picture28.Visible = True
MSComm1.PortOpen = True	Image1.Visible = False
Picture1.Visible = False	Image2.Visible = False
Picture2.Visible = False	Image3.Visible = False
Picture3.Visible = False	Image4.Visible = False
Picture4.Visible = False	Image5.Visible = False
Picture5.Visible = False	Image6.Visible = False
Picture6.Visible = False	Image7.Visible = False
Picture7.Visible = False	x = 0 y = 28 z = 0
Picture8.Visible = False	Label42.Caption = x
Picture9.Visible = False	Label43.Caption = y
Picture10.Visible = False	Text4.Text = z
Picture11.Visible = False	End Sub
Picture12.Visible = False	
Picture13.Visible = False	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub MSComm1_OnComm()

car = MSComm1.Input

pic = Hex\$(Asc(car))

Text3.Text = pic

Select Case pic

Case 0

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture1.Visible = False

Case 1

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture2.Visible = False

Case 2

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture3.Visible = False

Case 3

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture4.Visible = False

Case 4

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture5.Visible = False

Case 5

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture6.Visible = False

Case 6

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture7.Visible = False

Case 8

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

Picture8.Visible = False

Case 9

x = x - 1

y = y + 1

Label42.Caption = x

Label43.Caption = y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Picture9.Visible = False	Label43.Caption = y
Case "A"	Picture14.Visible = False
x = x - 1	Case 10
y = y + 1	x = x - 1
Label42.Caption = x	y = y + 1
Label43.Caption = y	Label42.Caption = x
Picture10.Visible = False	Label43.Caption = y
Case "B"	Picture15.Visible = False
x = x - 1	Case 11
y = y + 1	x = x - 1
Label42.Caption = x	y = y + 1
Label43.Caption = y	Label42.Caption = x
Picture11.Visible = False	Label43.Caption = y
Case "C"	Picture16.Visible = False
x = x - 1	Case 12
y = y + 1	x = x - 1
Label42.Caption = x	y = y + 1
Label43.Caption = y	Label42.Caption = x
Picture12.Visible = False	Label43.Caption = y
Case "D"	Picture17.Visible = False
x = x - 1	Case 13
y = y + 1	x = x - 1
Label42.Caption = x	y = y + 1
Label43.Caption = y	Label42.Caption = x
Picture13.Visible = False	Label43.Caption = y
Case "E"	Picture18.Visible = False
x = x - 1	Case 14
y = y + 1	x = x - 1
Label42.Caption = x	y = y + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label42.Caption = x	Case "1A"
Label43.Caption = y	x = x - 1
Picture19.Visible = False	y = y + 1
Case 15	Label42.Caption = x
x = x - 1	Label43.Caption = y
y = y + 1	Image1.Visible = False
Label42.Caption = x	Picture24.Visible = True
Label43.Caption = y	Case "1B"
Picture20.Visible = False	x = x - 1
Case 16	y = y + 1
x = x - 1	Label42.Caption = x
y = y + 1	Label43.Caption = y
Label42.Caption = x	Image1.Visible = False
Label43.Caption = y	Picture25.Visible = True
Picture21.Visible = False	Case "1C"
Case 18	x = x - 1
x = x - 1	y = y + 1
y = y + 1	Label42.Caption = x
Label42.Caption = x	Label43.Caption = y
Label43.Caption = y	Image1.Visible = False
Image1.Visible = False	Picture26.Visible = True
Picture22.Visible = True	Case "1D"
Case 19	x = x - 1
x = x - 1	y = y + 1
y = y + 1	Label42.Caption = x
Label42.Caption = x	Label43.Caption = y
Label43.Caption = y	Image1.Visible = False
Image1.Visible = False	Picture27.Visible = True
Picture23.Visible = True	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case "1E"

$x = x - 1$

$y = y + 1$

Label42.Caption = x

Label43.Caption = y

Image1.Visible = False

Picture28.Visible = True

Case "1F"

$x = x + 1$

$y = y - 1$

$z = z + 1$

Text4.Text = z

Label42.Caption = x

Label43.Caption = y

Picture1.Visible = True

Case 20

$x = x + 1$

$y = y - 1$

$z = z + 1$

Text4.Text = z

Label42.Caption = x

Label43.Caption = y

Picture2.Visible = True

Case 21

$x = x + 1$

$y = y - 1$

$z = z + 1$

Text4.Text = z

Label42.Caption = x

Label43.Caption = y

Picture3.Visible = True

Case 22

$x = x + 1$

$y = y - 1$

$z = z + 1$

Text4.Text = z

Label42.Caption = x

Label43.Caption = y

Picture4.Visible = True

Case 23

$x = x + 1$

$y = y - 1$

$z = z + 1$

Text4.Text = z

Label42.Caption = x

Label43.Caption = y

Picture5.Visible = True

Case 24

$x = x + 1$

$y = y - 1$

$z = z + 1$

Text4.Text = z

Label42.Caption = x

Label43.Caption = y

Picture6.Visible = True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 25**x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture7.Visible = True****Case 27****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture8.Visible = True****Case 28****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture9.Visible = True****Case 29****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture10.Visible = True****Case "2A"****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture11.Visible = True****Case "2B"****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture12.Visible = True****Case "2C"****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture13.Visible = True**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case "2D" $x = x + 1$ $y = y - 1$ $z = z + 1$ **Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture14.Visible = True****Case "2F"** $x = x + 1$ $y = y - 1$ $z = z + 1$ **Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture15.Visible = True****Case 30** $x = x + 1$ $y = y - 1$ $z = z + 1$ **Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture16.Visible = True****Case 31** $x = x + 1$ $y = y - 1$ $z = z + 1$ **Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture17.Visible = True****Case 32** $x = x + 1$ $y = y - 1$ $z = z + 1$ **Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture18.Visible = True****Case 33** $x = x + 1$ $y = y - 1$ $z = z + 1$ **Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture19.Visible = True****Case 34** $x = x + 1$ $y = y - 1$ $z = z + 1$ **Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture20.Visible = True**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 35**x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Picture21.Visible = True****Case 37****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Image1.Visible = True****Picture22.Visible = False****Case 38****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Image2.Visible = True****Picture23.Visible = False****Case 39****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Image3.Visible = True****Picture24.Visible = False****Case "3A"****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Image4.Visible = True****Picture25.Visible = False****Case "3B"****x = x + 1****y = y - 1****z = z + 1****Text4.Text = z****Label42.Caption = x****Label43.Caption = y****Image5.Visible = True****Picture26.Visible = False****Case "3C"****x = x + 1****y = y - 1****z = z + 1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text4.Text = z
Label42.Caption = x
Label43.Caption = y
Image6.Visible = True
Picture27.Visible = False

```

Case 26

```

x = x + 1
y = y - 1
z = z + 1
Text4.Text = z
Label42.Caption = x
Label43.Caption = y
Image7.Visible = True
Picture28.Visible = False
Label5.Caption = Date
Label6.Caption = Time

```

End Select

End Sub

```

Private Sub Timer2_Timer()
    If Label6.Caption = "0:00:00" Then
        Dim olerock As String
        'olerock = InputBox("Enter the path for
        a text file", "The Two
        R's", "C:\Therock.TXT")Open
        "C:\Therock.TXT" For Append As#1
        Print #1, Text4.Text, "Date Record :",
        Label5.CaptionClose #1Text4.Text = "0"
    End If
End Sub

```