สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รถบังคับด้วยแผนที่จากคอมพิวเตอร์

CAR CONTROL USE MAP IN COMPUTER

นายเฉลิมชัย ปัสกุล
นายธวัชชัย หัตถวิจิตรกุล
นายประพันธ์ บุญเฉิด

เลขหมู่.....
เลขทะเบียน 50432
วัน,เดือน,ปี 13 พ.ค. 2547

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

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CAR CONTROL USE MAP IN COMPUTER

CHALERMCHAI PUSSAKUL

TAWATCHAI HATTAVIJITKUL

PRAPAN BOONCHERD

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KINGMONGKUT 'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2002

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ รถบังคับด้วยแผนที่จากคอมพิวเตอร์

CAR CONTROL USE MAP IN COMPUTER

นักศึกษาผู้จัดทำ

นายเฉลิมชัย	ปัสกุล	รหัสประจำตัว	43015505
นายธวัชชัย	หัตถวิจิตรกุล	รหัสประจำตัว	43015512
นายประพันธ์	บุญเฉิด	รหัสประจำตัว	43015516

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2545

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 22 เมษายน พ.ศ.2546

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	รถบังคับด้วยแผนที่จากคอมพิวเตอร์
	CAR CONTROL USE MAP IN COMPUTER
นักศึกษาผู้จัดทำ	นายเฉลิมชัย ปัสกุล
	นายธวัชชัย หัตถวิจิตรกุล
	นายประพันธ์ บุญเฉิด
อาจารย์ที่ปรึกษา	รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์
ปีการศึกษา	2545

บทคัดย่อ

การควบคุมจากเครื่อง PC ได้ถูกนำมาใช้ในงานอุตสาหกรรม เนื่องจากสามารถแสดงผลหรือควบคุมอุปกรณ์หรือเครื่องจักรได้อย่างมีประสิทธิภาพ โดยจะนำ PC ไปควบคุมอุปกรณ์ต่างๆ ร่วมกับ Board Controller มาเป็นตัวรับคำสั่งเพื่อทำการประมวลผลและทำตามคำสั่งที่ได้รับต่อไป โดยปริญญานิพนธ์ฉบับนี้เป็นการนำเครื่อง PC และMicrocontroller มาทำงานร่วมกัน จำลองการควบคุมรถเพื่อให้เคลื่อนที่ไปตามเส้นทางที่กำหนดโดยผู้ควบคุม ซึ่งจะสามารถกำหนดเป้าหมายที่จะทำให้รถเคลื่อนที่ไปได้ แสดงให้เห็นการทำงานร่วมกันระหว่าง PC และ Microcontroller ควบคุมรถได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Car Control Use Map In Computer
Authors	Mr. Chalremchai Pussakul
	Mr. Tawatchai Hattavajitkul
	Mr. Prapan Booncherd
Thesis Advisor	Assoc.Prof.Dr.Fusak Shevasuwit
Year	2002

ABSTRACT

PC Controlled had used in industrial factory because it could control a lot of device. PC used control with microcontroller board which recived data to processed and activated up to data from PC.Destination is car control by map in computer. User can order the car to destination on map in computer which can show in computer screen.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุลวงได้ด้วยดีเพราะได้รับความเมตตาและการช่วยเหลือ จาก รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์ ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่าง ๆ ในการทำปริญญานิพนธ์นี้ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวัดคุมทุกท่านที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณพ่อและคุณแม่ อันเป็นที่รักยิ่ง ที่ให้การสนับสนุนและเป็นกำลังใจและแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้ คุณค่าและประโยชน์อันพึงมีการปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญภาพ.....VII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 ขั้นตอนการศึกษา.....1

บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับพอร์ตขนาน.....3

2.1 ความรู้เบื้องต้นเกี่ยวกับพอร์ตขนาน.....3

2.2 ทำไมจึงเลือกใช้พอร์ตขนาน.....3

2.3 ความรู้เบื้องต้นของพอร์ตขนาน.....4

2.4 ลักษณะทางกายภาพของพอร์ตขนาน.....4

2.5 พอร์ตข้อมูล.....8

2.6 พอร์ตควบคุม.....10

2.7 พอร์ตแสดงสถานะ.....11

2.8 การนำพอร์ตขนานไปใช้งาน.....12

2.9 การเขียนโปรแกรมติดต่อกับพอร์ตขนาน.....12

2.10 การเขยีนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual Basic.....13

บทที่ 3 หลักการและทฤษฎีที่ใช้ในการออกแบบ.....14

3.1 โปรแกรม Visual Basic.....14

3.2 ไมโครคอนโทรเลอร์ MCS – 51.....29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.1 การจัดขาของไมโครคอนโทรเลอร์....32
3.2.2 โครงสร้างและการทำงานของพอร์ต....35
3.2.3 การใช้งานเป็นพอร์ตอินพุต....38
3.2.4 การใช้งานเป็นพอร์ตเอาต์พุต....38
3.2.5 การอ่านค่าลอจิกจากพอร์ต....39

บทที่ 4 ผลการทดลอง....40
4.1 Flow Chart ส่วนของ Main Program....40
4.2 Flow Chart ส่วนของ Sub Program....41
4.3 Flow Chart ส่วนของการควบคุมรถ....42
4.4 การแสดงหน้าจอของการเรียก Program....43
4.5 การแสดงหน้าจอของการเรียก Main Program....44
4.6 การแสดงหน้าจอของการเรียก Main File....45
4.7 การแสดงหน้าจอของการเรียก Main Map....46
4.8 การแสดงหน้าจอของการเรียก Main About....47
4.9 การแสดงหน้าจอของการเรียก Map....48
4.10 การแสดงหน้าจอของการเรียก Map 1....49
4.11 การแสดงหน้าจอของการเรียก Map 2....50
4.12 ภาพชิ้นงาน....51

บทที่ 5 สรุปผลการทดลองและปัญหา....52

บรรณานุกรม....53

ภาคผนวก....54
ภาคผนวก ก. 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงสัญญาณสำคัญต่าง ๆ ของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์ 6
2.2 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน 9
2.3 แสดงแอดเดรสของพอร์ตขนาน 13
3.1 แสดงอักขระพิเศษที่ใช้กำหนดภาพแบบข้อมูล 18
3.2 แสดงตัวดำเนินการ AND 23
3.3 แสดงตัวดำเนินการ OR 24
3.4 แสดงตัวดำเนินการ XOR 24
3.5 แสดงตัวดำเนินการ IMP 24
3.6 แสดงตัวดำเนินการ EQV 24
3.7 แสดงตัวดำเนินการ NOT 25
3.8 แสดงตัวดำเนินการเปรียบเทียบ 26
3.9 แสดงลำดับของการคำนวณตัวดำเนินการ 27
3.10 แสดงทูลบ๊อกซ์ใน Visual Basic 27
3.11 แสดงรายละเอียดโดนสรุปบางส่วนของไมโครคอนโทรเลอร์ MCS – 51 แบบแฟลชที่ Atmel ผลิตขึ้น 32
3.12 แสดงหน้าที่พิเศษของพอร์ต 1 ในตัวไมโครคอนโทรเลอร์ MCS – 51 แบบแฟลช 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์ 5
2.2 แสดงระบบภายในของพอร์ตขนาน.....7
2.3 แสดงวงจรภายในของพอร์ตข้อมูล.....10
2.4 แสดงวงจรภายในของพอร์ตควบคุม.....11
2.5 แสดงวงจรภายในของพอร์ตแสดงสถานะ.....12
3.1 แสดงส่วนประกอบต่าง ๆ ของโปรแกรม Visual Basic.....14
3.2 แสดงการกำหนดตัวแปร.....17
3.3 แสดงการกำหนดค่าคงที่.....18
3.4 แสดงการกำหนดตัวแปรขึ้นเอง.....19
3.5 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรเลอร์ MCS – 51แบบแฟลชในอนุกรม AT89Cxx 30
3.6 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรเลอร์ MCS – 51 แบบแฟลชในอนุกรม AT89Sxx 31
3.7 แสดงราบละเอียดโครงสร้างหลักของไมโครคอนโทรเลอร์ MCS – 51 แบบแฟลช.....33
3.8 แสดงการจัดขามาตรฐานของไมโครคอนโทรเลอร์ MCS – 51 34
3.9 แสดงวงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรเลอร์ MCS – 51 แบบแฟลช.....37
3.10 แสดงวงจรพูลอัปภายในพอร์ตของไมโครคอนโทรเลอร์ MCS – 51 แบบแฟลช.....37
4.1 แสดง Flow chart แสดงการทำงานของ Main Program 40
4.2 แสดง Flow Chart แสดงส่วนของ Choose Map ใน Sub Program 41
4.3 แสดง Flow Chart แสดงส่วนของการควบคุมรถ 42
4.4 แสดงหน้าจอต้อนรับ 43
4.5 แสดงหน้าจอของ Main Program 44
4.6 แสดงหน้าจอของ Main File 45
4.7 แสดงหน้าจอของ Main Map 46
4.8 แสดงหน้าจอของ Main About 47
4.9 แสดงหน้าจอของการเรียก Map 48
4.10 แสดงหน้าจอของ Map 1 49
4.11 แสดงหน้าจอของ Map 2 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

4.12 แสดงชิ้นงาน..... 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

การควบคุมอุปกรณ์ทางไฟฟ้าและเครื่องจักรต่างๆ มีหลายรูปแบบ ในปัจจุบันได้มีการนำเครื่อง PC มาใช้กันอย่างแพร่หลาย การนำเครื่อง PC มาใช้ในการควบคุมก็เป็นอีกทางเลือกหนึ่งที่นำมาใช้ในการควบคุม ซึ่งสามารถแสดงผลทางมอนิเตอร์ ทำให้ง่ายต่อการควบคุมและการสังเกตผลที่ได้ โดยส่วนมากแล้ว การควบคุมอุปกรณ์และเครื่องจักรที่อยู่กับที่จะมีสายต่อระหว่างเครื่อง PC กับ แผงควบคุม (Board Control) แต่ถ้าหากเป็นอุปกรณ์หรือเครื่องจักรที่เคลื่อน ถ้าหากเครื่อง PC และ แผงควบคุม (Board Control) สามารถติดต่อกันโดยไม่ใช้สายก็จะทำให้เพิ่มความยืดหยุ่นของการเคลื่อนที่ของอุปกรณ์หรือเครื่องจักรที่ทำการควบคุม ทำให้สามารถเคลื่อนที่ได้อย่างอิสระ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะเป็นออกแบบและสร้าง รถควบคุมด้วยคอมพิวเตอร์ โดยใช้โปรแกรม Visual Basic ทำให้สามารถเข้าใจถึงโปรแกรมที่ใช้และการทำงานของ แผงควบคุม (Board Control) ซึ่งจะต้องทำงานร่วมกันได้อย่างมีประสิทธิภาพ

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์เล่มนี้จะกล่าวถึงการออกแบบและการสร้างรถควบคุมด้วยคอมพิวเตอร์ ซึ่งขั้นตอนสำคัญของ รถควบคุมด้วยคอมพิวเตอร์ ก็คือ การสั่งงานจากเครื่อง PC เพื่อควบคุมรถให้เคลื่อนที่ไปตามเป้าหมายที่กำหนดจากผู้ควบคุม โดยผู้ควบคุมจะสามารถกำหนดได้จากแผนที่ที่สร้างขึ้นทาง มอนิเตอร์ โดยเครื่อง PC จะทำการส่งข้อมูลมาทางเครื่องส่งและทำการส่งออกผ่านทางสายอากาศมายังเครื่องรับเพื่อส่งต่อให้ แผงควบคุมทำการประมวลผลและควบคุมการเคลื่อนที่ของรถต่อไป

1.4 ขั้นตอนการศึกษา

การทำโครงงานวิจัยในปริญญานิพนธ์ฉบับนี้ มีขั้นตอนการศึกษาเริ่มจากการทำงานของพอร์ตขนานของคอมพิวเตอร์ การทำงานของไมโครคอนโทรลเลอร์และการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของโปรแกรม Visual Basic ที่จะใช้ในการควบคุมการเคลื่อนที่ของรถบังคับให้เคลื่อนที่ไปในทิศทางที่ต้องการได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้เบื้องต้นเกี่ยวกับพอร์ตขนาน

2.1 ความรู้เบื้องต้นเกี่ยวกับพอร์ตขนาน

การประมวลผลข้อมูลเพื่องานควบคุมนั้น สิ่งแรกจะต้องมีส่วนของสัญญาณอินพุต ซึ่งอาจมาจากตัวตรวจจับต่างๆ ผ่านวงจรภาคหน้าเพื่อเปลี่ยนรูปแบบสัญญาณอินพุตให้เหมาะสมกับการเชื่อมต่อกับคอมพิวเตอร์ เมื่อข้อมูลอินพุตถูกส่งเข้าคอมพิวเตอร์แล้วคอมพิวเตอร์จะทำการประมวลผลของมูลที่ได้มาเหล่านั้น ให้อยู่ในรูปแบบที่เหมาะสมก่อนที่จะส่งไปยังอุปกรณ์ภายนอกผ่านอุปกรณ์เอาต์พุต ซึ่งอาจจะเป็นจอภาพ หรือส่งไปยังจุดเชื่อมต่ออื่นๆ เพื่อควบคุมอุปกรณ์เอาต์พุตต่อไป

การเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกทั้งส่วนของภาคอินพุต ภาคเอาต์พุต สามารถทำได้หลายวิธีดังนี้

1. เชื่อมต่อผ่านทางการ์ดอินพุตเอาต์พุตซึ่งใช้วิธีการเสียบหรือติดตั้งการ์ดลงในสล็อตภายในเครื่องคอมพิวเตอร์
2. เชื่อมต่อผ่านทางพอร์ตอนุกรม
3. เชื่อมต่อพอร์ตขนาน
4. เชื่อมต่อผ่านระบบมาตรฐานอื่น ๆ

2.2 ทำไมถึงเลือกใช้งานพอร์ตขนาน

เมื่อเทียบกับการใช้งานการ์ดอินพุตเอาต์พุตที่ต้องติดตั้งอยู่ภายในเครื่องคอมพิวเตอร์แล้ว พอร์ตขนานมีข้อได้เปรียบอยู่หลายประการดังนี้

ในด้านความปลอดภัย การที่ต้องถอดฝาเครื่องคอมพิวเตอร์ออกมาเพื่อเสียบการ์ดเชื่อมต่อลงในสล็อตของคอมพิวเตอร์ อาจทำให้เกิดการเสียหายกับส่วนอื่นๆ ของคอมพิวเตอร์ได้ ถ้าผู้ที่ใช้งานไม่มีความชำนาญหรือเกิดการต่อวงจรที่ผิดพลาด

ในด้นการเข้ากันได้กับคอมพิวเตอร์ส่วนใหญ่ การเชื่อมต่อโดยใช้การ์ดที่เสียบลงไปในสล็อตไม่สามารถใช้กับคอมพิวเตอร์ได้ในปัจจุบันได้ทุกรุ่น ยกตัวอย่าง คอมพิวเตอร์โน๊ตบุ๊กจะไม่มีที่เสียบแต่จะมีที่เสียบการ์ด PCMCIA แทน ในขณะที่พอร์ตขานจะมีติดตั้งอยู่ในคอมพิวเตอร์ทุกเครื่องทั้งนี้เพื่อใช้ในการติดต่อกับเครื่องพิมพ์

ข้อจำกัดทางด้านพื้นที่ คอมพิวเตอร์บางเครื่องมีที่เสียบการ์ดเชื่อมต่อตัวอื่นๆ อยู่แล้ว อาทิ การ์ดเสียง การ์ดโมเด็ม เป็นต้น จนไม่มีสล็อตเหลือพอสำหรับการเสียบการ์ดเพิ่มเติม

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสะดวกในการใช้งาน การเชื่อมต่อพอร์ตขนานสามารถทำได้ง่ายๆ เพียงต่อสายสำหรับเชื่อมต่อคอนเน็กเตอร์ DB – 25 ของพอร์ตขนาน

จำนวนช่องสัญญาณอินพุตเอาต์พุต พอร์ตขนานมีจำนวนช่องสัญญาณอินพุตเอาต์พุตมากพอที่จะนำไปใช้งานต่างๆ และยังสามารถขยายให้มีจำนวนพอร์ตเพิ่มขึ้นได้ โดยพอร์ตขนานปกติมีจำนวนขาเอาต์พุต 12 ขา และขาอินพุต 5 ขา

ความเร็วในการสื่อสารข้อมูลกับพอร์ตขนาน มีความเร็วเท่ากับการติดต่อระบบบัสโดยตรง และมีความเร็วมากกว่าการติดต่อผ่านพอร์ตอนุกรม

อะไหล่และชิ้นส่วนประกอบ คอนเน็กเตอร์และสายเชื่อมต่อต่าง ๆ ของการเชื่อมต่อผ่านพอร์ตขนาน หาได้ง่ายและราคาไม่แพง หรือสร้างขึ้นเองก็สามารถทำได้อย่างง่ายดาย

จากคุณสมบัติดังที่ได้กล่าวมาแล้วนั้น ทำให้พอร์ตขนานเหมาะสมอย่างยิ่งที่จะนำมาใช้ในการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ควบคุมภายนอกเพื่อควบคุมหรือรับสัญญาณข้อมูล นอกจากนั้น หากนำคุณสมบัติการเขียนโปรแกรมง่ายๆ ผ่านระบบปฏิบัติการวินโดวส์ด้วยโปรแกรม Visual Baisc ก็จะสามารถสร้างระบบการเชื่อมต่อที่สมบูรณ์และใช้งานง่ายได้ไม่ยาก

2.3 ความรู้เบื้องต้นของพอร์ตขนาน

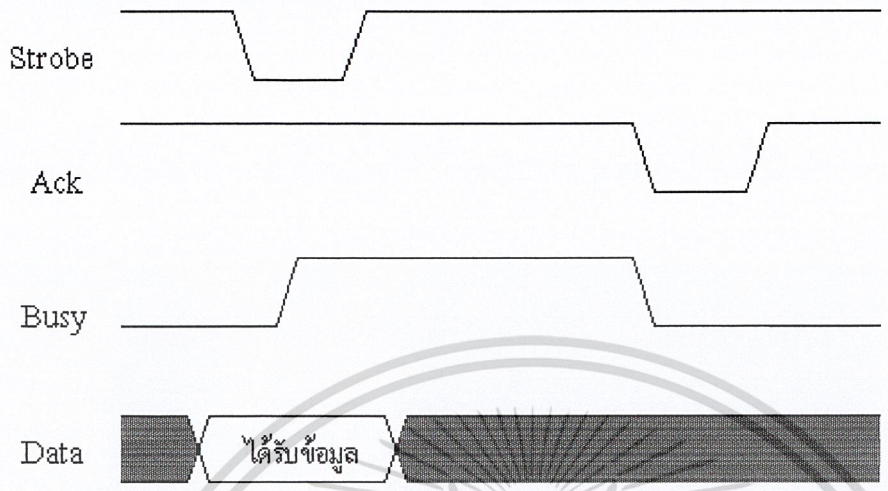
พอร์ตขนาน (Parallel port) สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายทอดข้อมูลของพอร์ตนี้เป็นแบบขนานสำหรับอีกชื่อเรียกหนึ่งของพอร์ตนี้คือ พอร์ตเครื่องพิมพ์ (Printer port) เนื่องจากพอร์ตนี้ใช้สำหรับต่อกับเครื่องพิมพ์นั้นเอง

ด้วยการถ่ายทอดข้อมูลแบบขนานนี้เอง ทำให้พอร์ตขนานมีอัตราการถ่ายทอดข้อมูลสูงกว่าการถ่ายทอดข้อมูลแบบอนุกรม 8 – 10 เท่า และการประมวลผลข้อมูลส่วนใหญ่มีขนาด 8 บิต ดังนั้นพอร์ตขนานจึงสามารถรองรับการถ่ายทอดข้อมูล 8 บิตได้ โดยไม่ต้องต่อส่วนเพิ่มเติมใดๆ

2.4 ลักษณะทางกายภาพของพอร์ตขนาน

เพื่อให้เข้าใจถึงการนำเอาพอร์ตขนานไปใช้งาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่าปกตินั้นการส่งพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตขนานนั้นมีรูปแบบการทำงานภายในอย่างไร ในภาพที่ 2 - 1 แสดงไดอะแกรมเวลาของการติดต่อระหว่างพอร์ตขนานกับเครื่องพิมพ์ ซึ่งจะเห็นได้ว่ามีสัญญาณที่ใช้งานจริงๆ มีไม่มาก เริ่มจากสัญญาณพอร์ต Data ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่งสัญญาณ Strobe ออกไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่ามีการส่งข้อมูลมาใหม่มาที่ขา Data แล้วจากนั้นคอมพิวเตอร์จะต้องรอการตอบรับจากเครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์

สัญญาณข้อมูลขนาด 8 บิต, สัญญาณ Strobe และสัญญาณ ACK (acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้วส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่น ๆ ร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ 3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนพอร์ต เป็นต้น บางครั้งอาจเกิดเหตุการณ์ไม่ปกติ เช่น บัฟเฟอร์สำหรับข้อมูลเต็มเครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลได้มากกว่านี้ สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์ คือ สัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่น กระดาษติด เครื่องจะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่า สัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซตเครื่องพิมพ์ คอมพิวเตอร์จะต้องส่งสัญญาณ Reset ไปยังเครื่องพิมพ์เพื่อรีเซตเครื่องพิมพ์ด้วย สามารถสรุปขาสัญญาณที่จำเป็นสำหรับการติดต่อดังในตารางที่ 2.1

จากตารางที่ 2.1 จะเห็นได้ว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกออกเป็นอีก 3 พอร์ต ได้แก่ พอร์ตเอาต์พุตทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณ Strobe และ Reset พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledeg, Busy และสัญญาณ Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 . 1 สัญญาณสำคัญต่าง ๆ ของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์

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

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณทั้งหมด 17 เส้น สายสัญญาณเหล่านั้นจะมีรีจิสเตอร์ 3 ตัว ควบคุมการทำงานดังนี้

1. พอร์ตเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ DATA ควบคุม
2. พอร์ตอินพุตสำหรับการอ่านค่าสถานะต่าง ๆ จากภายนอกมีอยู่ด้วยกัน 5 เส้น ใช้รีจิสเตอร์ Status ในการควบคุม
3. พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้น ใช้รีจิสเตอร์ Control ในการควบคุม

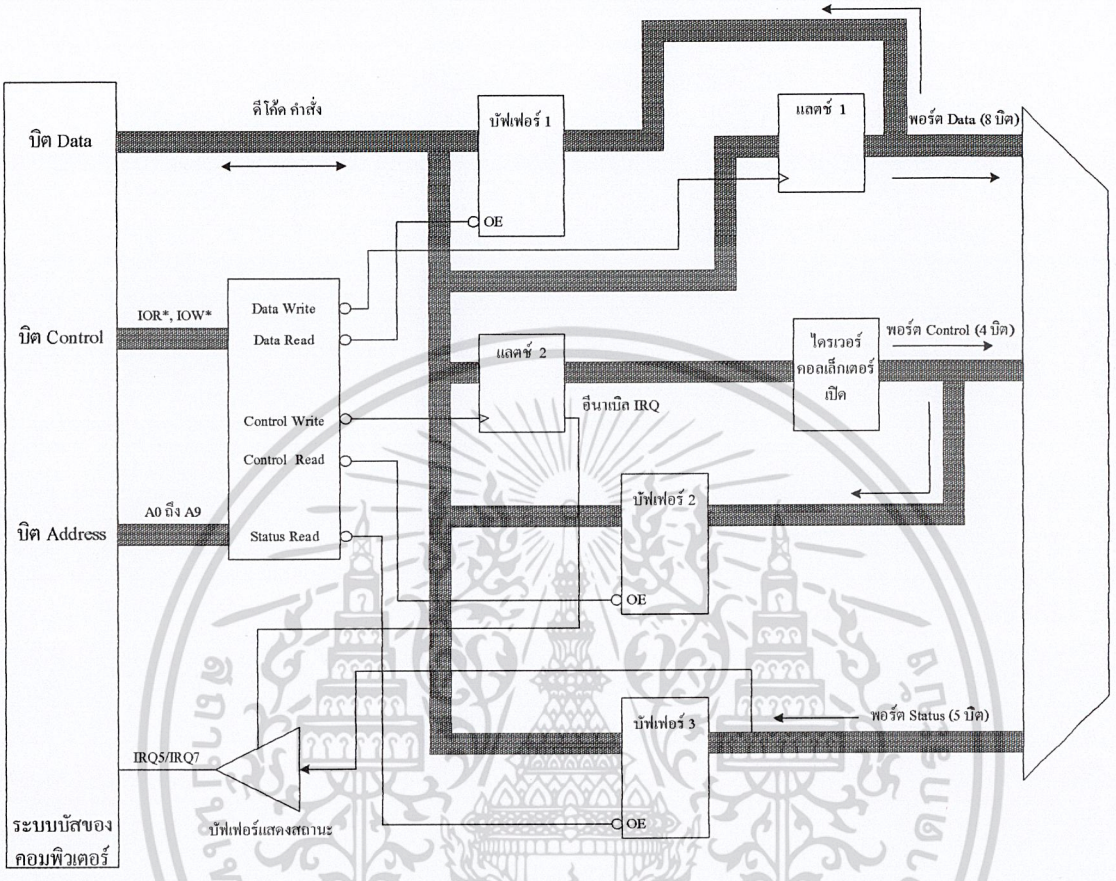
บล็อกไดอะแกรมในรูปที่ 1 - 2 แสดงระบบบัสของคอมพิวเตอร์สำหรับการติดต่อพอร์ตขนาน สัญญาณเอาต์พุตจากพอร์ตขนานจะถูกส่งไปยังคอนเน็กเตอร์แบบ DB-25 สำหรับคอมพิวเตอร์ส่วนใหญ่ในปัจจุบันพอร์ตขนานจะมากับเมนบอร์ด ไม่จำเป็นต้องเสียบการ์ดเหมือนในอดีต พร้อมทั้งมีฟังก์ชั่นการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ตขนานในรูปแบบมาตรฐานอยู่

เมื่อดูจากภาพที่ 2.1 เทียบการทำงานไปกับการเชื่อมต่อผ่านการ์ดที่เสียบลงไปในสล็อตของคอมพิวเตอร์แล้ว พอร์ตขนานมีลักษณะใกล้เคียงกัน โดยการติดต่อกับพอร์ตขนานจะต้อง มีการอ้างแอดเดรส ตำแหน่งแอดเดรสที่ใช้อ้างถึงจะเป็นตำแหน่ง A0 - A9 และขาที่ใช้ IOR และ IOW สำหรับเป็นตัวเลือกว่าต้องการอ่านหรือเขียนรีจิสเตอร์ตัวใด จากการดีโค้ดแอดเดรส A0 - A9 นี้เองทำให้ได้สัญญาณออกมาเพื่อไปควบคุมหรือเอ็นเอเบิลวงจรบัฟเฟอร์ต่าง ๆ ดังนี้

Data Write สัญญาณเอ็นเอเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ไปที่ขาออก Data ของพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Read สัญญาณเอ็นเอเบิลสำหรับนำข้อมูลจากขา Data ของพอร์ตขนานมาเก็บไว้ ในพอร์ต Data



ภาพที่ 2. 2 แสดงระบบบัสภายในของพอร์ตขนาน

Control Write สัญญาณเอ็นเอเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ออกไปที่ขา Control ของพอร์ตขนาน สำหรับพอร์ตนี้นอกจากจะส่งข้อมูลออกไปยังพอร์ตขนานแล้ว ยังทำหน้าที่การเอ็นเอเบิลการอินเตอร์รัปต์ของการเปลี่ยนแปลงสัญญาณที่พอร์ต Status อีกด้วย

Control Read สัญญาณเอ็นเอเบิล สำหรับอ่านข้อมูลจากขา Control มาเก็บไว้ในบัส Data

StatusRead สัญญาณเอ็นเอเบิล สำหรับอ่านค่าข้อมูลจากขาพอร์ต Status มาเก็บไว้ในบัส Data

ตารางที่ 2.2 แสดงชื่อและการทำงานของตำแหน่งขาต่าง ๆ บนพอร์ตขนาน ส่วนในตารางที่ 2.3 แสดงแอดเดรสของพอร์ตขนาน ซึ่งกำหนดไว้ 3 ตำแหน่ง LPT1, LPT2, และ LPT3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 พอร์ตข้อมูล

จากภาพที่ 2.3 แสดงให้เห็นว่าพอร์ต Data ประกอบไปด้วยบัฟเฟอร์ 1 ตัวและไอซีแลตช์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยัง ไอซีแลตช์ ทั้ง 8 บิต เอาต์พุตของไอซีแลตช์ 1 คือ D0 - D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานตำแหน่งขา 2 ถึงขา 9 และที่ขาเอาต์พุตสัญญาณนี้ Data จะส่งกลับเป็นอินพุตของบัฟเฟอร์ 1 ด้วย ทำให้คอมพิวเตอร์สามารถอ่านค่าสถานะปัจจุบันที่เกิดขึ้นกับพอร์ต Data ได้

เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมาจากข้อมูลของคอมพิวเตอร์ผ่านไปให้กับไอซี 74LS374 ซึ่งเป็นไอซีแลตช์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุต คอมพิวเตอร์จะส่งสัญญาณ Data Write ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกกรองด้วยวงจร RC ซึ่งประกอบด้วยค่าความต้านทาน 27 โอห์ม และตัวเก็บประจุ 0.0022 ไมโครฟารัด เพื่อให้ช่วงเวลาที่เปลี่ยนจากลอจิก “ 0 ” เป็นลอจิก “ 1” หรือจากลอจิก “ 1 ” เป็นลอจิก “ 0 ” เป็นไปอย่างช้าๆ เนื่องจากแรงดันที่รวดเร็วทำให้เกิดสัญญาณกวนเหนี่ยวนำข้ามไปยังข้อมูลบิตอื่นๆ ได้ ทำให้ข้อมูลที่ส่งออกไปมีข้อผิดพลาด จากค่าตัวต้านทานและตัวเก็บประจุในวงจรทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที จากวงจรในรูปที่ 1 - 3 ทำให้พอร์ตเอาต์พุต Data มีคุณสมบัติดังนี้

1. กระแสซิงก์สูงสุด 24 มิลลิแอมป์
2. กระแสซอร์สสูงสุด 2.6 มิลลิแอมป์
3. ระดับแรงดันของลอจิก “ 1” ต่ำสุดเท่ากับ 2.4 โวลท์
4. ระดับแรงดันสูงสุดสำหรับลอจิก “0” เท่ากับ 0.5 โวลท์

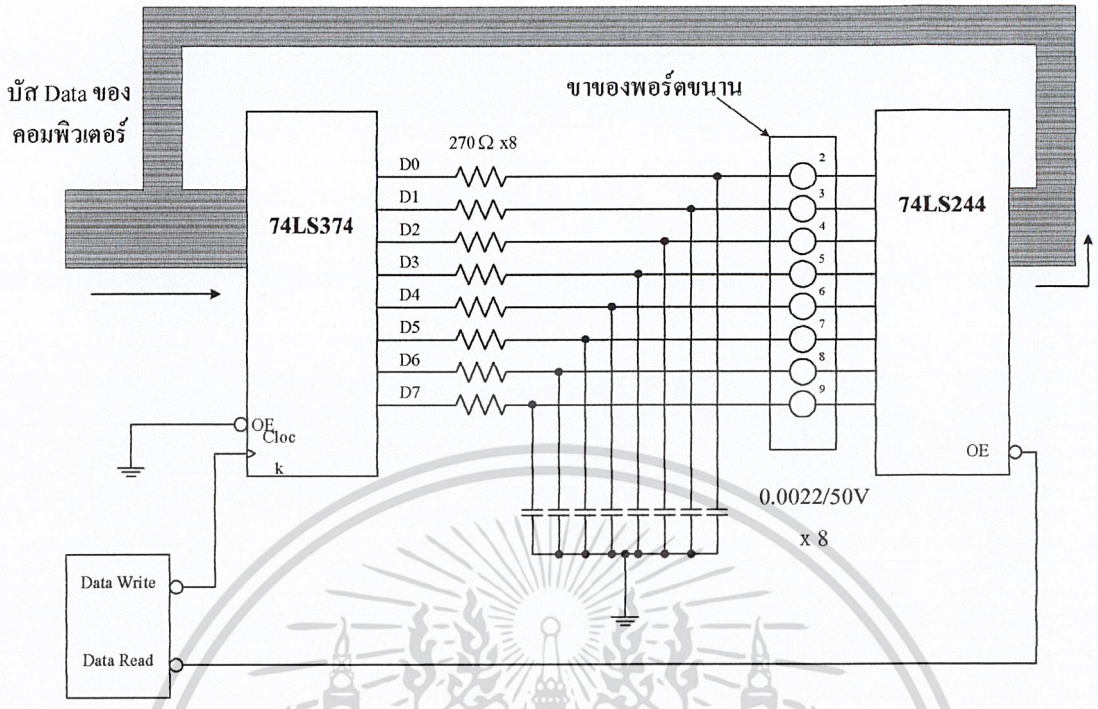
สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับได้แก่เบอร์ 74LS244 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ Data Read ออกมาเพื่อเป็นเอ็นเอเบิลไอซี 74LS244 สำหรับพอร์ตขนานแบบมาตรฐาน พอร์ต Data จะต้องให้เพื่อการส่งออกเอาต์พุตเท่านั้น แต่สำหรับพอร์ตขนานที่มีการสื่อสารสองทิศทาง สามารถอ่านค่าจากพอร์ต Data ได้ด้วย แต่ก่อนที่จะอ่านค่าต้องไว้เสมอว่าจะต้องป้อนค่าเอาต์พุตให้มีค่าลอจิก “1” ทั้งหมดก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2. 2 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน

ขาของพอร์ตขนาน	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อสัญญาณ	หน้าที่การทำงาน
1	Control	Out	C0	STROBE	แอกตีฟ “0” ส่งค่าออกไปเพื่อบอกว่าที่ขาดดาต้ามีข้อมูล
2-9	Data	Out	D0-D7	DATA0-DATA7	สำหรับพอร์ตขนานมาตรฐานเดิมขานี้ทำหน้าที่เป็นขาส่งข้อมูลเอาต์พุตเท่านั้นสำหรับในปัจจุบันขานี้รับข้อมูลอินพุตได้ด้วย
10	Status	In	S6	nACK	เป็นพัลส์ลอจิก “0” ที่ส่งมาจากเครื่องพิมพ์เพื่อบอกว่าได้รับข้อมูลที่ส่งไปแล้ว
11	Status	In	S7	BUSY	เป็นสัญญาณแจ้งมาจากเครื่องพิมพ์ว่ายังไม่พร้อมรับข้อมูล
12	Status	In	S5	PE	แจ้งกระดาษหมด
13	Status	In	S4	SELECT	แจ้งว่าเครื่องพิมพ์ต่ออยู่
14	Control	Out	C1	AUTO FEED	สั่งเครื่องพิมพ์ให้เลื่อนบรรทัด
15	Status	In	S3	ERROR	สัญญาณจากเครื่องพิมพ์มายังคอมพิวเตอร์เพื่อแสดงข้อพลาดจากการพิมพ์
16	Control	Out	C2	INT1	รีเซตเครื่องพิมพ์โดยให้ลอจิก “0”
17	Control	Out	C3	SELECT -IN	ส่งสัญญาณไปยังเครื่องพิมพ์เพื่อแจ้งว่าต้องการเลือกเครื่องพิมพ์เครื่องนี้
18-25				GND	กราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 วงจรภายในของพอร์ตข้อมูล

2.6 พอร์ต ควบคุม

พอร์ต Control ใช้สำหรับคอมพิวเตอร์ควบคุมเครื่องพิมพ์ จากตารางที่ 2.2 จะเห็นว่าพอร์ต Control ประกอบไปด้วยบิตเอาต์พุต4บิตที่ต่อ ไปยังเครื่องพิมพ์ ส่วนบิตเอ็นเอเบิล อินเตอร์รัปต์ ไม่ได้ถูกต่อออกไป ภาพที่ 2.4 แสดงบล็อกไดอะแกรมของพอร์ต Control จะเห็นว่าเอาต์พุตของพอร์ต Control มีอินเวอร์เตอร์แบบคอลเล็กเตอร์เปิดต่ออ่วมอยู่ โดยเอาต์พุตเหล่านี้ จะถูกพูลอัปไว้ด้วยตัวต้านทานค่า 4.7 กิโลโอห์ม สำหรับบิต C2 จะผ่านอินเวอร์เตอร์ถึงสองตัว ทำที่เอาต์พุตของบิต C2 ไม่มีการกลับสถานะลอจิก

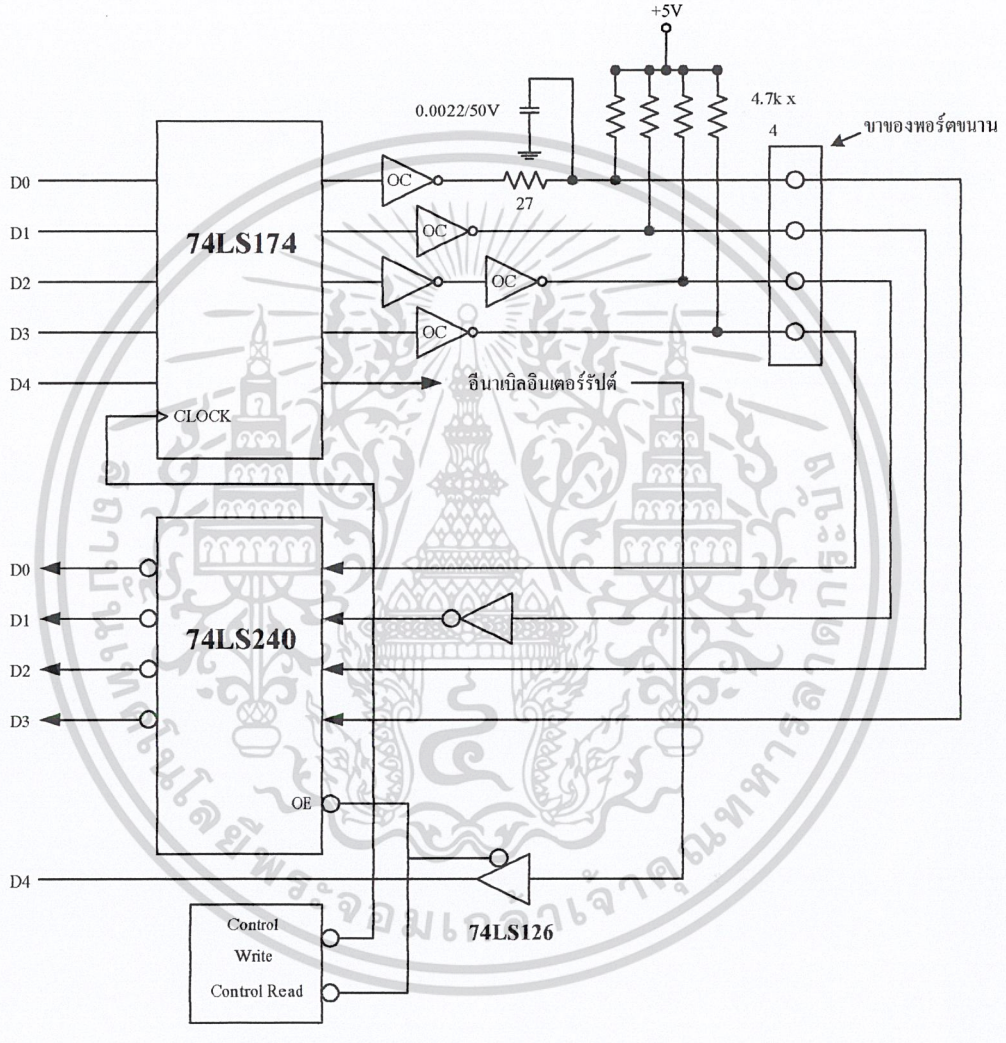
สถานะของพอร์ต Control สามารถอ่านกลับได้ด้วยการใช้บัฟเฟอร์ 74LS240 ซึ่งมีเอาต์พุตอินเวอร์เตอร์ต่ออยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่านและการเขียนข้อมูลกับพอร์ต Control คอมพิวเตอร์จะส่งข้อมูลมาที่ขา Control Write และ Control Read

เนื่องจากเอาต์พุตของพอร์ต Control เป็นแบบคอลเล็กเตอร์เปิด ดังนั้นผู้ใช้งานสามารถใช้พอร์ตนี้ ในการอ่านค่าสัญญาณอินพุตจากภายนอกได้โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านค่ามีลอจิก “1” เสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

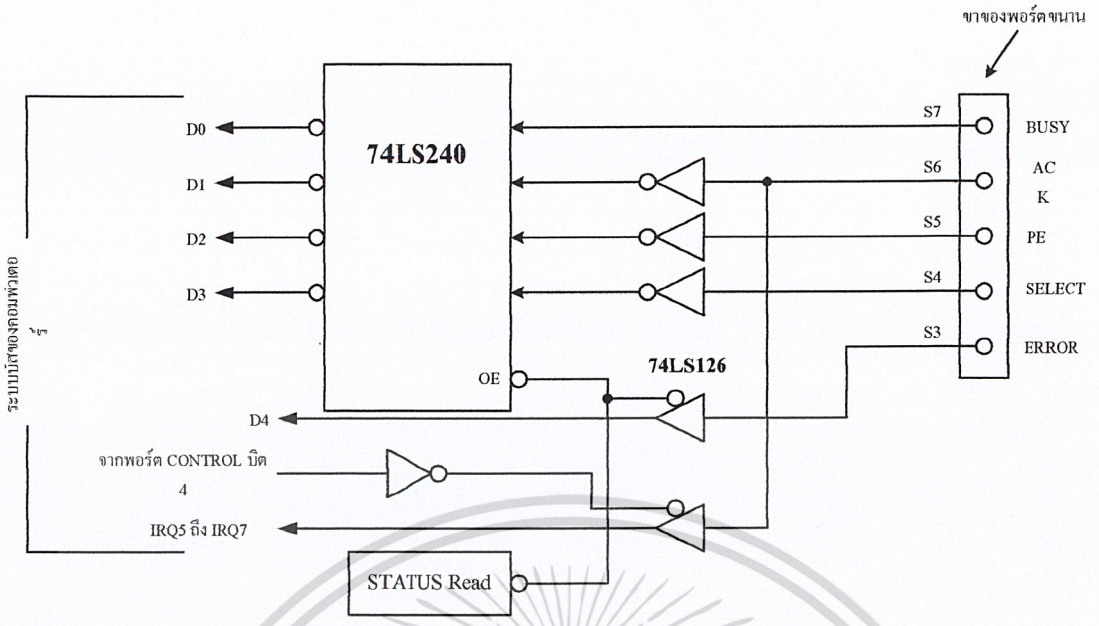
2.7 พอร์ตแสดงสถานะ

พอร์ต Status เป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะจากเครื่องพิมพ์ภาพที่ 2.5 แสดงรายละเอียดภายในของพอร์ต Status จะสังเกตได้ว่ามีขาสัญญาณอยู่ทั้งหมด 5 สัญญาณด้วยกันและจะเรียกชื่อเป็น S3,S4,S5,S6 และ S7 ซึ่งตังเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้



ภาพที่ 2.4 วงจรภายในของพอร์ตควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2. 5 แสดงวงจรภายในของพอร์ตแสดงสถานะ

ภายในรีจิสเตอร์ Status นั่นเอง สำหรับบิต S7 จะมีข้อแตกต่างจากบิตอื่นๆ ที่เมื่อสัญญาณจากภายนอกส่งเข้ามาแล้วจะไม่ผ่านจากขาอินพุตไปยัง 74LS240 ซึ่งเอาต์พุตมีการกลับสถานะ ทำให้บิต S7 เป็นบิตเดียวที่มีการกลับสถานะ นอกจากนี้การใช้งานถ้าต้องการให้มีการสร้างสัญญาณอินเตอร์รัปต์จากขอบขาขึ้นของขา S6 สามารถกำหนดค่าได้จากพอร์ต Control บิต 4

2.8 การนำพอร์ตขนานไปใช้งาน

สำหรับพอร์ตขนานแบบมาตรฐาน ผู้ใช้งานสามารถนำอินพุต 5 บิต (พอร์ต Status) พอร์ตเอาต์พุต 4 บิต(พอร์ต Control) และพอร์ตเอาต์พุตอีก 8 บิต(พอร์ต Data) ไปใช้งานได้โดยตรง โดยที่ 4 บิตของพอร์ตเอาต์พุตหรือพอร์ต Control นั้นสามารถดัดแปลงให้ใช้งานเป็นพอร์ตอินพุตขนาด 4 บิตได้ด้วยดังนั้นผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ตขนานที่มีมากถึง 17 เส้นไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TTL

2.9 การเขียนโปรแกรมติดต่อกับพอร์ตขนาน

พอร์ตขนานของคอมพิวเตอร์จะมีลักษณะเช่นเดียวกับอุปกรณ์อินพุตเอาต์พุตตัวอื่นๆ คือเมื่อต้องการติดต่อจะต้องกำหนดแอดเดรสที่ต้องการติดต่อด้วย ตารางที่ 2.1 แสดงแอดเดรส ของพอร์ตขนาน โดยแบ่งออกเป็น 3 ตำแหน่งคือ แอดเดรสของรีจิสเตอร์ Data รีจิสเตอร์ Status และรีจิสเตอร์ Control โดยแอดเดรสนี้จะอยู่ทั้งหมด 3 ชุด สำหรับพอร์ตขนาน 3 ชุดคือ LPT1, LPT2, และ LPT3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ หาประโยชน์ในเชิงการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

OUT &H378 , &HFF

โดยที่ เครื่องหมาย &H ที่แสดงนั้นหมายถึงตัวเลขของฐานสิบหก

คำสั่ง OUT เป็นการส่งค่าข้อมูลออกเอาต์พุตของพอร์ตอินพุตเอาต์พุต

ค่า 378 เป็นแอดเดรสของรีจิสเตอร์ Data สำหรับ LPT1

ค่าข้อมูล FF เป็นข้อมูลเลขฐานสิบหก ซึ่งหมายถึงการให้บิตทุกบิตของรีจิสเตอร์ Data มีลอจิกเป็น “1” นั่นเอง

ส่วนการอ่านค่าจากพอร์ตขนานมายังคอมพิวเตอร์ผ่านทางพอร์ต Status ของ LPT1 สามารถเขียนโปรแกรมด้วย QBASIC ได้ดังนี้

ตารางที่ 2.3 แสดงแอดเดรสของพอร์ตขนาน

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

Temp = INS (&H379)

โดยที่ คำสั่ง INS (_) เป็นคำสั่งสำหรับการอ่านข้อมูล

ค่า 379 เป็นตำแหน่งแอดเดรสของรีจิสเตอร์ STATUS สำหรับ LPT1 ในเลขฐานสิบหก

ตัวแปร Temp เป็นตัวแปรที่ใช้เก็บข้อมูลที่อ่านได้จากพอร์ตขนาน

2.10 การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual BASIC

การเขียนโปรแกรมด้วย Visual Basic ชุดคำสั่งส่วนใหญ่จะมีรูปแบบใกล้เคียงกับ QBasic แต่ Visual Basic จะมีคำสั่งสำหรับการติดต่อกับพอร์ตโดยตรงคือ คำสั่ง Ins (_) และคำสั่ง OUT เหมือน QBasic ดังนั้นเพื่อให้สามารถติดต่อกับพอร์ตขนานได้จึงจำเป็นต้องเพิ่มโปรแกรมบางตัวเข้าไป โดยโปรแกรมที่เพิ่มเข้าไปนี้จะอยู่ในรูปของ DLL (DYNAMIC LINKED LIBRARY)

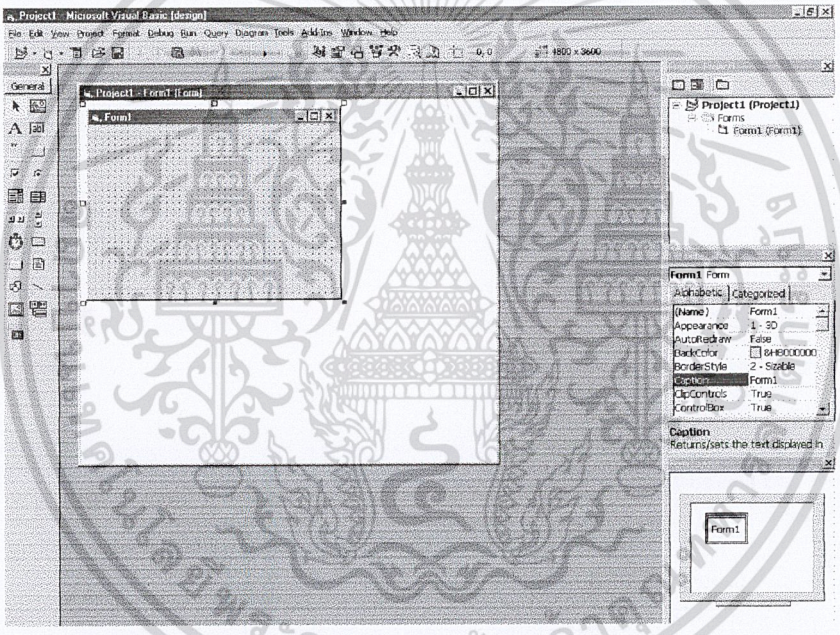
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการและทฤษฎีที่ใช้ในการออกแบบ

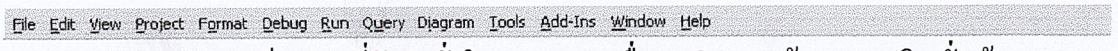
3.1 โปรแกรม Visual Basic

Microsoft Visual Basic ถือว่าเป็นเครื่องมือพัฒนาแอพพลิเคชั่นที่ถูกพัฒนามาอย่างต่อเนื่อง เพื่อเป็นเครื่องมือพัฒนาแอพพลิเคชั่นที่สามารถทำงานได้มากมายและง่ายในการใช้งาน

ลักษณะการทำงานของ Visual Basic จะทำงานในลักษณะ IDE (Integrated Deverlopment Environment) คือ รวบรวมเครื่องมือ, ข้อมูลที่ใช้งานต่าง ๆ ไว้ในหน้าจอเดียว ทำให้เรียกใช้งานได้ง่าย



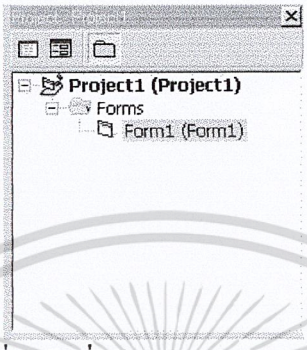
ภาพที่ 3.1 ส่วนประกอบต่าง ๆ ของโปรแกรม Visual Basic



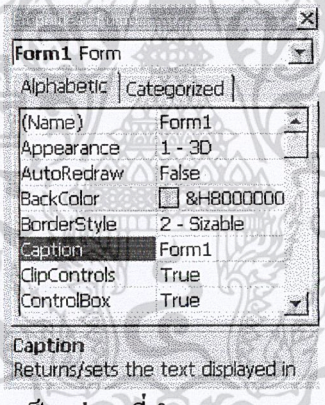
1. Menu Bar เป็นส่วนที่รับคำสั่งในแบบเมนู เมื่อเราทำการสร้างแอพพลิเคชั่นด้วย Visual Basic เป็นเหมือนศูนย์กลางที่ควบคุมการสร้างแอพพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

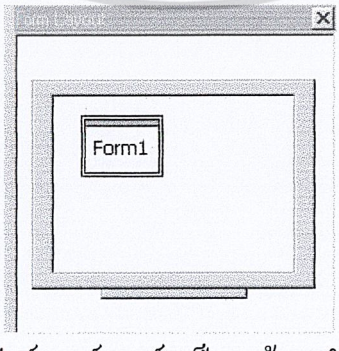
2. Tool Bar ในการใช้งานในเมนูบาร์สั่งงานอาจจะมีขั้นตอนที่ยุ่งยากเพื่อลดขั้นตอนลง จะทำให้การคลิกที่ทูลบาร์เพียงครั้งเดียวสามารถสั่งงานที่เราต้องการได้ (เป็นเหมือนคีย์ลัดในการทำงาน)



3. Project Window เป็นเครื่องมือที่ใช้ควบคุมการทำงานของโปรเจ็กต์เพราะ Visual Basic สนับสนุนการสร้างแอพพลิเคชั่นได้หลายแบบ

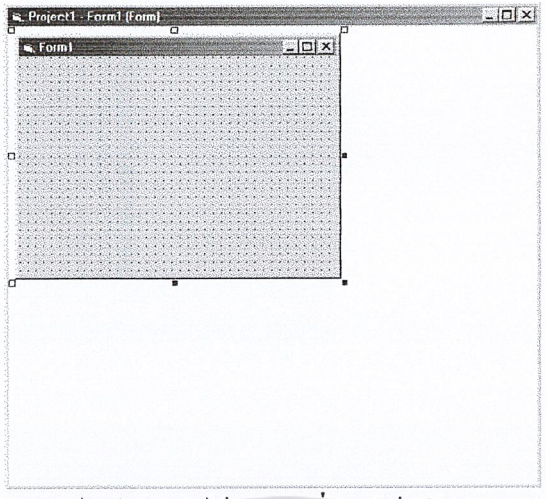


4. Properties Window เป็นส่วนที่กำหนดพรอพเพอร์ตี้ให้กับออบเจ็กต์ต่างๆ ในแอพพลิเคชั่น

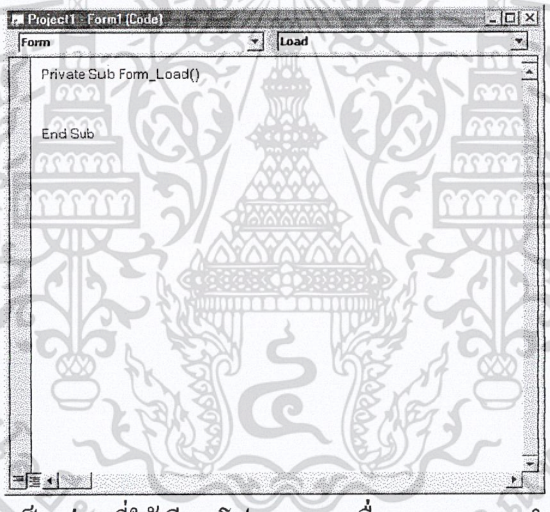


5. Form Layout ฟอร์มเลย์เอาท์ เป็นหน้าตาจำลองของฟอร์มที่ได้จากการรันแอพพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



6. Form Designer ฟอร์มดีไซเนอร์เป็นส่วนที่มองเห็นในขณะออกแบบแอพพลิเคชั่น ของ Visaul Basic เป็นส่วนที่จะใช้ติดต่อกับผู้ใช้งาน โดยการนำ ActiveX Control ต่างๆ มาวางไว้ข้างบน



7. Code Window เป็นส่วนที่ใช้เขียนโปรแกรม เพื่อควบคุมการทำงานของแอพพลิเคชั่น

8. ข้อกำหนดและคำสั่งต่างๆ ของ Visual Basic ในการเขียนโปรแกรมด้วย Visual Basic นั้นจำเป็นที่จะต้องทราบถึงข้อกำหนดต่าง ๆ และคำสั่งในการทำงานต่าง ๆ ของ Visual Basic ซึ่งจะมีความใกล้เคียงกับภาษา Basic ทั่วๆ ไป ข้อกำหนดและคำสั่งต่าง ๆ ของ Visual Basic มีดังนี้

8.1 ชนิดของข้อมูลที่ใช้ใน Visual Basic ไมโครซอฟท์ได้กำหนดมาตรฐานของข้อมูลที่ใช้ในโปรแกรมหรือโค้ดสำหรับ Visual Basic ดังนี้

8.1.1 Byte ข้อมูลตัวเลขจำนวนตั้งแต่ 0 ถึง 255 ขนาดหน่วยความจำ 1 ไบต์

8.1.2 Boolean ข้อมูลทางตรรก : จริง (True) , เท็จ (False) ขนาดหน่วยความจำ 2 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

8.1.3 Integer จำนวนเต็มระหว่าง -32,768 ถึง 32,767 ขนาดหน่วยความจำ 2 ไบต์

8.1.4 Longจำนวนเต็มระหว่าง -2,147,483,648 ถึง 2,147,483,647 ขนาดหน่วยความจำ 4 ไบต์

8.1.5 Single เลขทศนิยมระหว่าง -3.402823E38 ถึง -1.401298E-4 สำหรับ ค่าลบ และ 1.401298E-45 ถึง 3.402823E38 สำหรับค่าบวก 4 ไบต์

8.1.6 Double เลขทศนิยมระหว่าง -1.79769313486232E308 ถึง -494065645841247E-324 สำหรับค่าลบและ 4.94065645841247E-324 ถึง 1.79769313486232E308 สำหรับค่าบวก ขนาดหน่วยความจำ 8 ไบต์

8.1.7 Currency เลขที่มีค่าตั้งแต่ -922,337,203,685,477.5808 ถึง 922,337,203,685,477.5807 ขนาดหน่วยความจำ 8 ไบต์

8.1.8 Date วันที่ตั้งแต่ 1 มกราคม ค.ศ.100 ถึง 31 ธันวาคม ค.ศ.999 ขนาดหน่วยความจำ 8 ไบต์

8.1.9 Object ข้อมูลที่อ้างอิงออปเจ็กต์ ซึ่งเก็บแอดเดรสของออปเจ็กต์ไว้ ขนาดหน่วยความจำ 4 ไบต์

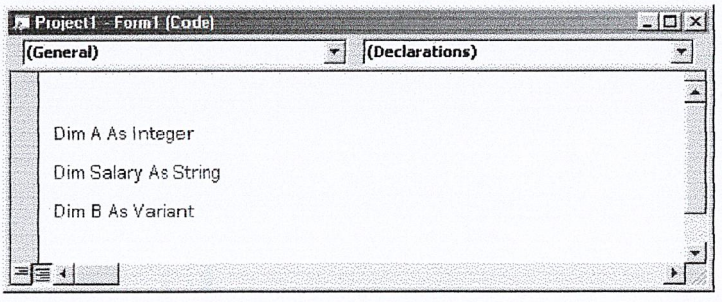
8.1.10 String เก็บสตริงหรือข้อความที่เรียงต่อกัน ขนาดหน่วยความจำ 64 KB หรือ 2 MB

8.1.11 Variant ข้อมูลชนิดพิเศษที่เก็บค่าได้ทุกแบบ (รวมไปถึงค่าพิเศษต่าง ๆ ที่มีตัวเลข) เช่น EMPTY , NULL เป็นต้น) ขนาดหน่วยความจำ 16 ไบต์

8.2 การกำหนดตัวแปรและค่าคงที่ ในการกำหนดตัวแปรให้กับโปรแกรมนั้น จะใช้คำสั่ง Dim กำหนดตัวแปรดังนี้

Dim Variable [As Type]

Variable หมายถึงชื่อของตัวแปร สำหรับ Type หมายถึงชนิดของข้อมูล ในส่วนที่อยู่ในวงเล็บนี้หากไม่มีจะเป็นข้อมูลแบบ Variant สำหรับการกำหนดตัวแปรมีตัวอย่างดังนี้



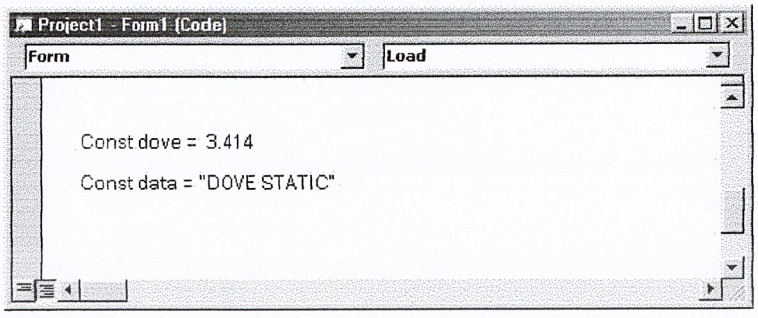
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.2 การกำหนดตัวแปร

50432

สำหรับตัวแปรที่เก็บข้อมูลแบบตัวเลขซึ่งมีค่าคงที่จะใช้คำสั่ง Const กำหนดดังนี้

Const Name = Expression



ภาพที่ 3.3 การกำหนดค่าคงที่

ถ้าต้องการเจาะจงแบบของข้อมูลสำหรับคำสั่งแบบ Const สามารถใส่อักขระพิเศษหลังตัวแปรหรือที่ค่าของตัวแปรเพื่อกำหนดแบบของข้อมูล ความหมายของอักขระพิเศษที่ใช้ในการกำหนดแบบของข้อมูลของ Visual Basic แสดงดังตารางที่ 8

ตารางที่ 3.1 อักขระพิเศษที่ใช้กำหนดภาพแบบข้อมูล

อักขระ	แบบของข้อมูล
%	Integer
&	Long
!	Single
#	Double
@	Currency
$	String

8.3 แบบข้อมูลที่ผู้ใช้กำหนดขึ้นเอง การกำหนดแบบของข้อมูลขึ้นเองนั้น สามารถทำได้โดยใช้คำสั่ง Type ซึ่งมีภาพแบบดังต่อไปนี้

Type Defined-type

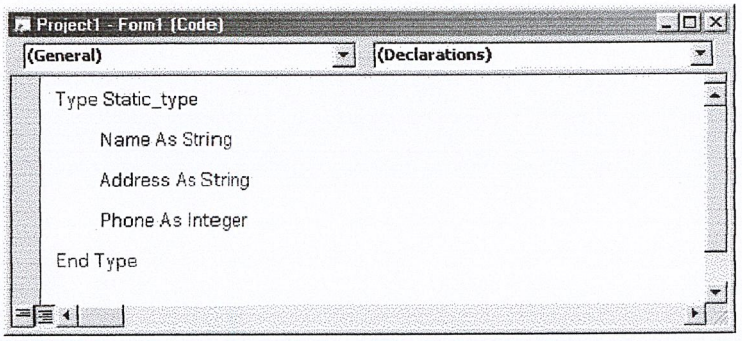
Element [As Type]

[Element [As Type]

.....

End Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4 การกำหนดตัวแปรขึ้นเอง

เมื่อกำหนดแบบของข้อมูลขึ้นใช้แล้ว ต้องกำหนดตัวแปรให้เป็นข้อมูลนี้เพื่อใช้ในโปรแกรมดังนี้

Dim Variable As Defined-type

Dim Mybook As BOOK

วิธีในการอ้างอิงข้อมูลซึ่งกำหนดขึ้นเองนั้น สามารถทำได้ดังตัวอย่าง

Mybook. Title = "Visual BASIC 4.0"

Mybook. Serial = 220

Mybook. Author = "K. Nakamura"

9. การตัดสินใจ (Decision) ในการเขียนโปรแกรม บางครั้งจำเป็นต้องตัดสินใจให้แอพพลิเคชั่นทำงานอย่างใดอย่างหนึ่งจากทางเลือกที่จะมีให้เลือกมากกว่า 1 ทางเลือก โดยการตัดสินใจในโปรแกรม จะมี 2 ประเภท คือ

9.1 ตัดสินใจเลือก จากทางเลือก 2 ทางเลือก : If…Then…Else

If <ทดสอบเงื่อนไขว่าจริง หรือเท็จ> Then

ถ้าเป็นจริงให้ทำงานหลังคำว่า Then

Else

ถ้าเป็นเท็จให้ทำงานหลังคำว่า Else

End If

ในบางครั้งจะใช้ If…Then…Else เพื่อตัดสินใจ จากทางเลือกมากกว่า 2 ทางเลือก โดยการตัดสินใจเลือกทีละ 2 ทางเลือก

If <ทดสอบเงื่อนไขว่าจริง หรือเท็จ> Then

ถ้าเป็นจริงให้ทำงานหลังคำว่า Then

Else If <ทดสอบเงื่อนไขว่าจริง หรือเท็จ> Then

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเป็นจริงให้ทำงานหลังคำว่า Then เสมอ

Else If

:

Else

ถ้าเป็นเท็จให้ทำงานหลังคำว่า Else

End If

9.2 ตัดสินใจเลือก จากทางเลือกมากกว่า 2 ทางเลือก : Select...Case

Select Case <ทดสอบเงื่อนไข>

Case เงื่อนไขแรก : <ทำงานตามเงื่อนไขแรก>

Case เงื่อนไขที่สอง : <ทำงานตามเงื่อนไขที่สอง>

:

Case สุดท้าย : <ทำงานตามเงื่อนไขสุดท้าย>

Case Else <เมื่อไม่ตรงกับเงื่อนไขใด ๆ เลย ทำงานหลังคำว่า Else>

End Select

10. การวนซ้ำ (Iteration) ในการเขียนโปรแกรม เราอาจจำเป็นต้องสั่งให้แอพพลิเคชั่นทำงานวนซ้ำตามจำนวนครั้งที่ต้องการได้ ซึ่งภาพแบบของการวนซ้ำมี 2 ภาพแบบ ได้แก่

10.1 การวนซ้ำด้วยจำนวนรอบที่แน่นอน : For...Next เป็นภาพแบบการวนซ้ำที่เราสามารถกำหนดรอบของการวนซ้ำได้แน่นอน

For ตัวแปรใช้นับจำนวนรอบ = จำนวนรอบเริ่มต้น To จำนวนรอบสุดท้าย

<ทำงานตามคำสั่ง>

Next ตัวแปรที่ใช้นับจำนวนรอบ

จะเห็นว่าต้องใช้ตัวแปร 1 ตัวทำหน้าที่เป็นตัวนับรอบในการวนซ้ำ ซึ่งตัวนับนี้สามารถนับได้ทั้งแบบเดินหน้า (ตัวนับมีค่าเพิ่มขึ้น) และแบบถอยหลัง (ตัวนับมีค่าลดลง)

10.2 การวนซ้ำด้วยจำนวนรอบที่ไม่แน่นอน จะมี 2 ภาพแบบ คือ

10.2.1 การวนซ้ำด้วยจำนวนรอบที่ไม่แน่นอน : While...Wend เป็นภาพแบบการวนซ้ำที่ไม่สามารถกำหนดเงื่อนไขการวนซ้ำได้ โดยอาศัยการตรวจสอบเงื่อนไขก่อนการวนซ้ำว่าต้องวนซ้ำอีกหรือไม่

While <ทดสอบเงื่อนไข จริงหรือเท็จ>

<ถ้าเงื่อนไขเป็นจริง ให้ทำงานตามคำสั่ง>

Wend

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเงื่อนไขการหลุดจากการวนซ้ำ นั้นจะต้องทำให้เงื่อนไขเป็นเท็จ (False) เท่านั้นจึงจะหลุด

10.2.2 การวนซ้ำด้วยจำนวนรอบที่ไม่แน่นอน : Do/While…Until/Loop

Do While <ทดสอบเงื่อนไข จริงหรือเท็จ>

<ถ้าเงื่อนไขเป็นจริง ให้ทำงานตามคำสั่ง>

Loop

Do Until <ทดสอบเงื่อนไข จริงหรือเท็จ>

<ถ้าเงื่อนไขเป็นเท็จอยู่ ให้ทำงานตามคำสั่ง>

Loop

Do

<ทำงานตามคำสั่ง>

Loop While <ทดสอบเงื่อนไข จริงหรือเท็จ ถ้าเป็นจริงให้กลับไปทำงานอีกรอบ>

Do

<ทำงานตามคำสั่ง ก่อน 1 รอบ>

Loop Until <ทดสอบเงื่อนไข จริงหรือเท็จ ถ้าเป็นเท็จให้กลับไปทำงานอีกรอบ>

จะเห็นได้ว่าสามารถตรวจสอบเงื่อนไขได้ 2 แบบ

10.2.2.1 While จะหลุดจากการวนซ้ำได้เมื่อทดสอบเงื่อนไขแล้วเป็นเท็จ (False)

10.2.2.2 Until จะหลุดจากการวนซ้ำได้เมื่อทดสองเงื่อนไขแล้วเป็นจริง(True)

นอกจากนี้ยังสามารถวางตำแหน่งการตรวจสอบเงื่อนไขได้ ว่าจะตรวจสอบก่อนการวนซ้ำ หรือหลังจากการวนซ้ำ

10.3 การกระโดดออกจากการวนซ้ำในบางครั้งเราจำเป็นต้องกระโดดออกมา จากการวนซ้ำ ด้วยเงื่อนไขบางอย่างสามารถทำได้ โดยใช้คำสั่ง Exit For, Exit Loop

11. โปรแกรมย่อย (Procedure) ในการเขียนโปรแกรมย่อยเกิดการทำงานที่ซ้ำ ๆ กัน สามารถลดการทำงานซ้ำ ๆ นั้น โดยการเขียนโปรแกรมย่อยเก็บไว้ เมื่อต้องการทำงานแบบเดิมอีกเพียงแต่เรียกโปรแกรมย่อยนั้นมาใช้งาน ทำให้ลดเวลาการเขียนโปรแกรมและลดความยุ่งยากของโปรแกรมที่เขียนด้วย

โปรแกรมย่อยแบ่งออกเป็น 2 ประเภท ตามลักษณะของการคืนค่ากลับมาหลังจากจบโปรแกรมย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.1 Sub (Sub Routine) สับรูทีนเป็นโปรแกรมย่อยที่เขียนขึ้นมา เมื่อโปรแกรมย่อยทำงานเสร็จแล้วจะไม่มีการคืนค่าใด ๆ กลับมายังผู้ที่เรียกใช้งาน ซึ่งสับรูทีนมีโครงสร้างดังนี้

```
Sub ชื่อสับรูทีน (รายการอาร์กิวเมนต์)
    :
<คำสั่งใน Visual Basic >
    :
End Sub
```

11.2 Function ฟังก์ชันเป็นโปรแกรมย่อยที่จะต้องคืนค่ากลับมาหาผู้เรียกใช้หลังจากโปรแกรมย่อยทำงานเสร็จแล้วซึ่งฟังก์ชัน มีโครงสร้างดังนี้

```
Function ชื่อฟังก์ชัน (รายการอาร์กิวเมนต์) As ชนิดข้อมูลที่คืนให้ผู้เรียกใช้
    :
<คำสั่งใน Visual Basic >
    :
End Function
```

12. ขอบเขตการใช้งานตัวแปรทั้งตัวแปรและค่าคงที่ที่ได้กล่าวมาแล้ว โดยปกติจะมีขอบเขตการใช้งานเพื่อให้สะดวก เหมาะสม และมีความเป็นเหตุเป็นผล ตัวแปรหรือค่าคงที่ใน Visual Basic จะมีขอบเขตการใช้งาน 3 ระดับคือ

12.1 ตัวแปรแบบโลคอล (Local) เป็นตัวแปรที่มีขอบเขตอยู่ในระดับโพรซีเดอร์หรือในฟังก์ชั่นที่ประกาศในตัวแปรนั้น ส่วนมากเพื่อการใช้งานเป็นส่วนย่อย ๆ เพื่อเก็บค่าเพียงชั่วคราวเมื่อโพรซีเดอร์หรือฟังก์ชั่นนั้นสิ้นสุด ค่าจะถูกยกเลิกด้วย

12.2 ตัวแปรแบบโมดูล (Module) เป็นตัวแปรที่ประกาศ เพื่อใช้ในหลาย โพรซีเดอร์หรือหลายฟังก์ชั่นในโมดูลนั้น รวมถึงโมดูลที่มีหลายรูทีนและโมดูลของฟอร์มที่มีรูทีนสำหรับการทำงานกับ ออปเจคในฟอร์มนั้นประกาศตัวแปรในโมดูล จะประกาศในส่วน Declaration

12.3 ตัวแปรแบบโกลบอล (Global) เป็นตัวแปรที่ประกาศได้ทุกส่วนของ โปรเจกต์ ตัวแปรนั้นสามารถนำไปใช้ได้ในทุกรูทีนและทุกส่วน การประกาศเช่นเดียวกับ ตัวแปรทั่วไปเพียงแต่เพิ่มคำว่า “Golbal” นำหน้าตัวแปรนั้น ภาพแบบการประกาศเป็นดังนี้

เช่น Global variable As type

```
Global A As Integer
Global Const Pi = 3.1415
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. ตัวแปรแบบอาร์เรย์ (Array Variable) ตัวแปรแบบอาร์เรย์เปรียบเสมือนตัวแปรซ้อนตัวแปร มีประโยชน์มากสำหรับโปรแกรมที่มีตัวแปรแบบลักษณะเดียวกัน ภาพแบบการกำหนดตัวแปรแบบอาร์เรย์เป็นดังนี้

เช่น Dim variable (size) As type

Dim X(3) As Integer หรือ X%(3)

สำหรับการกำหนดค่าหรือการรับค่าจากตัวแปรอาร์เรย์ สามารถระบุได้ดังนี้

Dim X(1) = 5

Dim X(2) = 10

Dim X(3) = 15

14. ตัวดำเนินการ ในการประมวลผลข้อมูลทางคณิตศาสตร์ ทางตรรกศาสตร์ การเปรียบเทียบข้อมูลและการประมวลผลข้อมูลชนิดสตริง นั้นต้องใช้ตัวดำเนินการต่างๆ ซึ่ง Visual Basic ได้กำหนดไว้ดังนี้

14.1 ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

+ การบวก (Addition)

- การลบ (Subtraction)

* การคูณ (Multiplication)

^ การยกกำลัง (Exponentiation)

/ การหารผลลัพธ์ที่ได้เป็นเลขจำนวนจริง (Floating-Point Division)

\ การหารผลลัพธ์ที่ได้เป็นเลขจำนวนเต็ม (Integer Division)

Mod การหาเศษจากการหารเลขจำนวนเต็ม 2 จำนวน (Modulus)

14.2 ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator) ตัวดำเนินการนี้ ในบางครั้งเรียกว่าเป็นตัวดำเนินการ Boolean เพราะตัวดำเนินการเหล่านี้ได้มาจากพีชคณิต ของตรรกคณิตศาสตร์ ซึ่งพัฒนาขึ้นโดย จอร์จ บูล ตัวดำเนินการเหล่านี้มีค่า Boolean เป็นโอเปอแรนด์และตัวดำเนินการจะส่งคืนผลลัพธ์ Boolean

ตารางที่ 3.2 ตัวดำเนินการ AND

การดำเนินการ	ผลลัพธ์
False And False	False
False And True	False
True And False	False
True And True	True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ตัวดำเนินการ OR

การดำเนินการ	ผลลัพธ์
False Or False	False
False Or True	True
True Or False	True
True Or True	True

ตารางที่ 3.4 ตัวดำเนินการ XOR

การดำเนินการ	ผลลัพธ์
False Or False	False
False Or True	True
True Or False	True
True Or True	False

ตารางที่ 3.5 ตัวดำเนินการ IMP

การดำเนินการ	ผลลัพธ์
False Imp False	True
False Imp True	True
True Imp False	False
True Imp True	True

ตารางที่ 3.6 ตัวดำเนินการ EQV

การดำเนินการ	ผลลัพธ์
False Eqv False	True
False Eqv True	False
True Eqv False	False
True Eqv True	True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 ตัวดำเนินการ NOT

การดำเนินการ	ผลลัพธ์
NOT False	True
NOT True	False

14.3 ตัวดำเนินการเชื่อมสตริง (String Concatenation Operator) ตัวดำเนินการ ทางสตริงจะใช้เครื่องหมาย + เป็นตัวเชื่อมสตริง ตัวอย่างเช่น

กำหนดให้ Name$ = “Dumrong”
ทำการเชื่อมสตริงดังนี้ “My name is + Name$ +”. ”
ผลลัพธ์คือ My name is Dumrong

นอกจากนี้ยังสามารถใช้เครื่องหมาย & เป็นตัวเชื่อมข้อมูลแบบสตริง ข้อมูลแบบตัวเลข (Integer , Long , Currency , Single หรือ Double) เข้าด้วยกัน โดยที่ Visual Basic จะเปลี่ยนเป็น ข้อมูลแบบสตริงโดยอัตโนมัติซึ่งผลลัพธ์ที่ได้จะเป็นข้อมูลแบบสตริง

ตัวอย่างเช่น Varl = 76
Val2 = “trombones”
MsgBox Varl& “ ” & Var2
ผลลัพธ์ที่ปรากฏคือ 76 trombores

14.4 ตัวดำเนินการเปรียบเทียบ (Comparison Operator หรือ Relational Operator) ตัวดำเนินการเปรียบเทียบใช้กับข้อมูลได้ทุกชนิด ผลของการเปรียบเทียบที่ได้ คือ True และ False ซึ่งเป็น Built-in Constant ของ Visual Basic ค่าแท้จริงของ True และ False จะเป็นเลขจำนวนเต็ม –1 และ 0 ตามลำดับ ผลจากการเปรียบเทียบจะเป็นค่าตรรกตัวดำเนินการเปรียบเทียบที่ใช้ใน Visual Basic ดังตารางที่ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	ความหมายของตัวดำเนินการ
>	มากกว่า (Greater Than)
<	น้อยกว่า (Less Than)
>=	มากกว่าหรือเท่ากับ (Greater Than Or Equal To)
<=	น้อยกว่าหรือเท่ากับ (Less Than Or Equal To)
=	เท่ากับ (Equal To)
<>	ไม่เท่ากับ (Not Equal To)

ตัวอย่างเช่น 4 > 2^2 ให้ค่าเป็น False

a <= Sum

"Exit" = "E" + "x" + "I" + "t" ให้เป็นค่า True

True <= 12 ให้ค่าเป็น True เพราะ –1 น้อยกว่า 12

15. ลำดับการประมวลผลข้อมูล ในกรณีที่นิพจน์มีตัวดำเนินการหลายแบบรวมอยู่ด้วยกัน หากมีการใส่วงเล็บ Visual Basic จะประมวลผลข้อมูลในวงเล็บในก่อน จากนั้นจึงประมวลผลข้อมูลตามลำดับความสำคัญของตัวดำเนินการ ในกรณีที่ตัวดำเนินการมีความสำคัญเท่ากันจะประมวลผลนิพจน์จากซ้ายไปขวาและจากตารางที่ 16 แสดงรายการของชุดของตัวดำเนินการของ Visual Basic ทั้งหมด (ยกเว้นตัวดำเนินการเชื่อมต่อสตริง) เรียงลำดับความสำคัญ ของตัวดำเนินการที่ใช้ในนิพจน์ โดยเรียงลำดับตามการทำก่อนจากสูงสุดไปยังต่ำสุด เมื่อพบกับนิพจน์ย่อยนั้นเป็นนิพจน์แรก หลังจากนั้นหาว่าตัวดำเนินการใดที่มีการทำก่อนอยู่ในลำดับรองลงมาให้ประเมินผลนิพจน์ย่อยนั้น และทำเช่นนี้ต่อเนื่องไปจนกว่าจะได้ผลลัพธ์ในขั้นสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 ลำดับของการคำนวณตัวดำเนินการ

ลำดับ	ตัวดำเนินการ	สัญลักษณ์
1	ยกกำลัง	^
2	เท่ากับ	=
3	NOT	NOT
4	ไม่เท่ากับ	<>
5	AND	AND
6.	คูณหาร	* /
7	น้อยกว่า	<
8	OR	OR
9	หารเอาจำนวนเต็ม	\
10	มากกว่า	>
11	XOR	XOR
12	เศษการหาร	MOD
13	น้อยกว่าหรือเท่ากับ	<=
14	บวก ลบ	+ -
15	มากกว่าหรือเท่ากับ	>=
16	รวมข้อความ	&

16. ทูลบ็อกซ์ (Tool Box) ส่วนประกอบของแอพพลิเคชันที่ได้บรรจุเอาไว้บนแบบฟอร์ม เรียกว่า Object หรือตัวควบคุม ทูลบ็อกซ์เป็นที่รวม Object ต่าง ๆ ที่จะนำมาประกอบในแอพพลิเคชัน โดย Object จะมีดังตารางที่ 17

ตารางที่ 3.10 ทูลบ็อกซ์ใน Visual Basic

ออปเจ็กต์	คุณสมบัติ
ab|	Text Box ใช้สำหรับข้อความซึ่งอาจยาวเกินกว่าจะแสดงในกรอบหรือ Box ให้เห็นพร้อมกันหมดได้ Text เป็นคุณสมบัติที่เก็บข้อความ ปรากฏอยู่ในกรอบหรือ Box เวลาจะเอาข้อความไปใช้อ้างถึงคุณสมบัติ Text นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออปเจ็กต์	คุณสมบัติ
	Image ทำหน้าที่แสดงภาพ เช่นเดียวกับ Picture Box ทั้งภาพในแบบ Bitmap , Icon หรือ Metafile ข้อดีของ Image คือ ใช้ทรัพยากรของระบบน้อยกว่า Picture Box และยังทำการวาดภาพใหม่ได้เร็วกว่าด้วยหน้าที่หลักของ Image คือ แสดงภาพเท่านั้น
	Data Control ทำหน้าที่สร้างตัวควบคุมข้อมูล โดยอาศัยตัวควบคุมนี้ ในการเข้าถึงข้อมูลจำเพาะในฐานข้อมูล โดยคุณสมบัติและวิธีการใช้งานของตัวควบคุมข้อมูลนี้ จะแสดงอย่างละเอียดในหัวข้อ Data Control
	DBCombo มีลักษณะคล้ายกับคอมโบบ็อกซ์ที่เป็นมาตรของ Windows 95 แต่สามารถแสดงข้อมูลฟิลด์ในเรคคอร์ดเซตที่อยู่ในดาต้าคอนโทรลได้ โดยทั่วไปเราใช้ DBCombo คู่กับ DBGrid ในการตั้งเงื่อนไขในประโยคย่อย WHERE ของนิพจน์ SQL
	Command Button เป็นคอนโทรลพื้นฐานที่ใช้บ่อยที่สุดคือ ใช้เป็นปุ่มในการสั่งให้ทำงาน เพราะอย่างน้อยที่สุดเกือบทุกฟอร์มจะมีปุ่ม Ok และ Cancel อยู่ด้วย
	Label มีหน้าที่แสดงข้อความให้ปรากฏบนฟอร์ม โดยผู้ใช้ไม่สามารถแก้ไขได้เหมือน Text Box และเหมือนกับคอนโทรลอื่น ๆ อีกหลายตัวที่มีคุณสมบัติ Caption ซึ่งเก็บข้อความให้เห็น
	Timer ใช้เพื่อให้ทำงานในทุก ๆ ช่วงเวลาที่กำหนดได้ และสามารถทำงานในแบบฉากหลังได้ ในตอนรันโปรแกรมจะไม่เห็นตัวคอนโทรลในขณะที่รันโปรแกรมขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	OLE Container ใช้ในการเชื่อม (Linking) และฝัง (Embedding) ออปเจ็กต์จากโปรแกรมอื่น นำความสามารถที่โปรแกรมอื่นมีอยู่แล้วมาใช้ได้โดยไม่ต้องสร้างขึ้นใหม่เอง
	Picture Box เป็นออบเจ็กต์ที่แสดงภาพในภาพของ Bitmap , Icon หรือ Metfile โดยที่แสดงภาพเท่าที่แสดงได้ในขนาดของตัวในเท่านั้น นั่นคือ หากขนาดของภาพโตกว่า Picture Box จะเห็นภาพไม่หมด นอกจากนี้อาจใช้คำสั่งด้านกราฟิกเพื่อวาดภาพใน Picture Box รวมไปถึงการใช้ Method Print ได้ด้วย หรือจะใช้แสดงข้อความ (Text) ได้ซึ่ง Picture Box มีคุณสมบัติเกี่ยวกับ Text ไว้ให้ด้วย
	DBList มีลักษณะคล้ายกับลิสต์บ็อกซ์ที่เป็นมาตรฐานของ Windows 95 แต่สามารถแสดงข้อมูลของฟิลด์ในเรคคอร์ดเซตที่อยู่ในดาต้าคอนโทรลได้ DBList ทำงานร่วมกับดาต้าคอนโทรล

17. Data Control ตัวควบคุม Data มีคุณสมบัติจำนวนหนึ่งที่นิยามการติดต่อระหว่างแอพพลิเคชัน Visual Basic กับฐานข้อมูลที่ต้องการเข้าถึง แอพพลิเคชันสามารถมีตัวควบคุม Data มากกว่าหนึ่งตัวและชุดข้อมูลที่นิยามไว้แต่ละชุดต้องการวัดควบคุม Data ที่แยกต่างหากกัน ตัวควบคุม Data แต่ละตัวจะเข้าถึงได้เพียงทีละหนึ่งเรคคอร์ด ซึ่งเรียกว่า เรคคอร์ดล่าสุด (Current Record) หลังจากตั้งคุณสมบัติเหล่านี้แล้ว จะสามารถยึดเหนี่ยวตัวควบคุมอื่นๆ ของ Visual Basic เข้ากับตัวควบคุม Data เมื่อเปลี่ยนแปลงเรคคอร์ดล่าสุด ตัวควบคุมเหล่านั้น จะแสดงข้อมูลที่อยู่ในเรคคอร์ดล่าสุดออกมา Visual Basic นี้สนับสนุนการแลกเปลี่ยนข้อมูลกับ DBMS คือ Microsoft Access , Microsoft Foxpro , dBase และ Paradox

3.2 ไมโครคอนโทรลเลอร์ MCS – 51

คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ MCS – 51

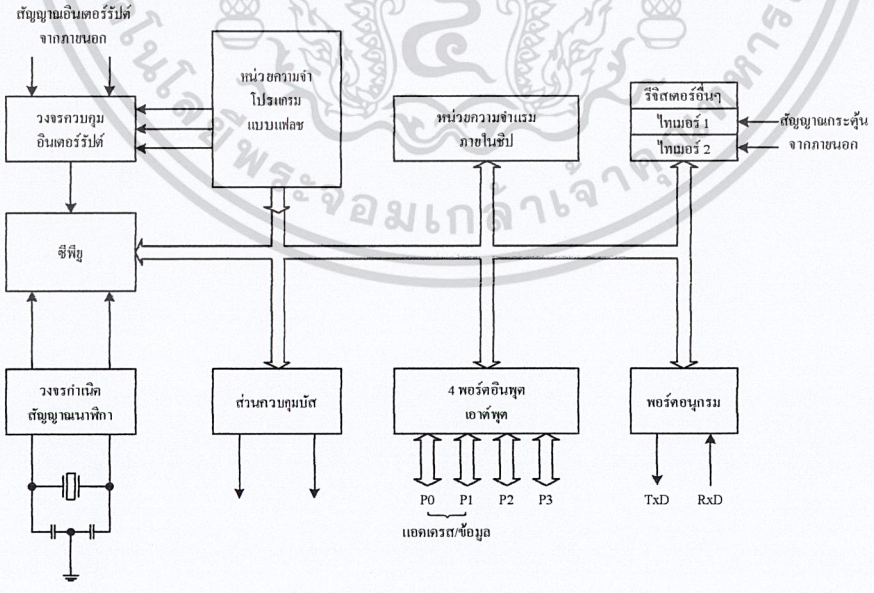
- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมแบบแฟลชสามารถลบและเขียนใหม่ได้เป็นพันครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์ / เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติม ได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI
- มีวอตช์ด็อกไทเมอร์ในตัว

ในภาพที่ 3.5 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS - 51 ในอนุกรม AT89Cxx จะเห็นได้ว่า โครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS - 51 พื้นฐานหากแตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา หากเป็นไมโครคอนโทรลเลอร์ในอนุกรม AT87Cxx หน่วยความจำภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้ครั้งเดียว

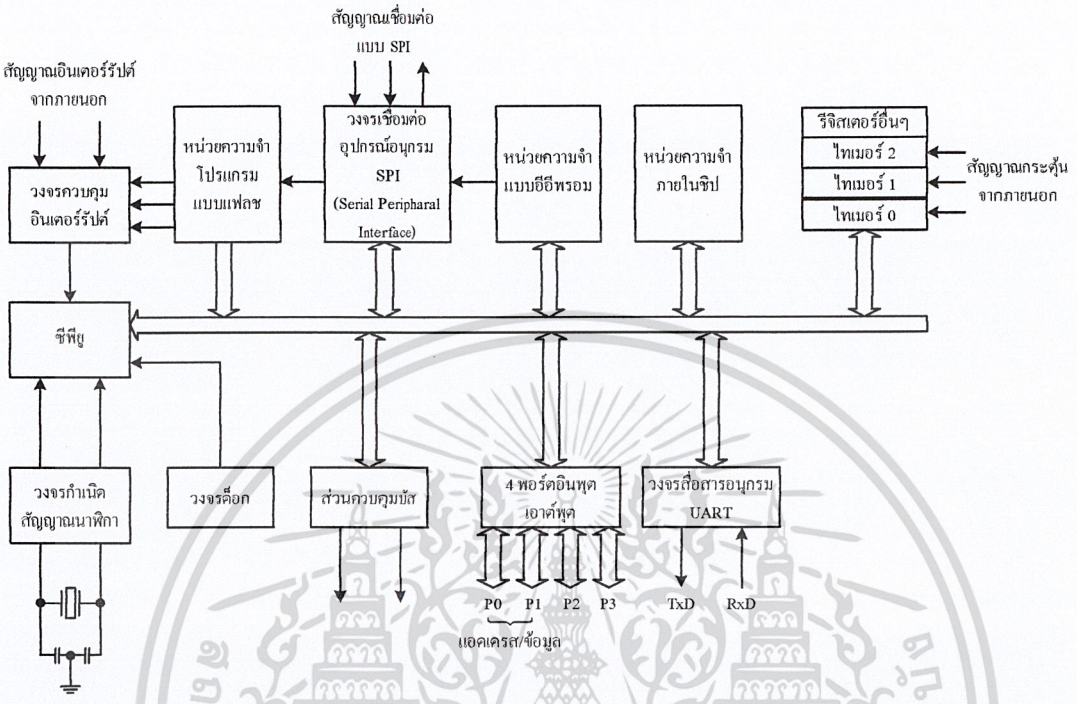
สำหรับในภาพที่ 3.6 เป็นโครงสร้างพื้นฐานของอนุกรม AT89Cxx จะเห็นได้ว่ามีส่วนประกอบที่เพิ่มเติมแตกต่างกันจาก AT89Cxx อยู่หลายส่วน อาทิ วงจรที่เชื่อมต่ออนุกรมแบบ SPI ซึ่งในไมโครคอนโทรลเลอร์



ภาพที่ 3.5 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชในอนุกรม AT89Cxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนุกรมนี้ใช้ในการเขียนข้อมูลลงในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิปออกไปจากระบบหรือเรียกว่า การโปรแกรมในวงจร ไทเมอร์ / เคาน์เตอร์ ขนาด 16 บิตที่เพิ่มเข้ามาอีกหนึ่งตัวเป็นไทเมอร์ 2 และวงจรวอตซ์ด็อกที่ใช้ในการตรวจสอบการทำงานของซีพียู



ภาพที่ 3.6 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชในอนุกรม AT89Sxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 รายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชที่ Atmel ผลิตขึ้น

เบอร์ของไมโครคอนโทรลเลอร์	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล	จำนวนไทเมอร์ / เคาน์เตอร์ 16 บิต
AT89C1051	แบบแฟลช ขนาด 1 กิโลไบต์	แรม 64 ไบต์	1
AT89C2051	แบบแฟลช ขนาด 2 กิโลไบต์	แรม 128ไบต์	2
AT89C51	แบบแฟลช ขนาด 4 กิโลไบต์	แรม 128ไบต์	2
AT89C52	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256ไบต์	3
AT89C55	แบบแฟลช ขนาด 20 กิโลไบต์	แรม 256ไบต์	3
AT89S8252	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256ไบต์ อีอีพรอม 2 กิโลไบต์	3
AT89S53	แบบแฟลช ขนาด 12 กิโลไบต์	แรม 256ไบต์	3

3.2.1 การจัดขาของไมโครคอนโทรลเลอร์ MCS – 51

ไมโครคอนโทรลเลอร์ MCS - 51 ทุกเบอร์จะมีสถาปัตยกรรมและขาที่ใช้งานพื้นฐานเหมือนกันดังแสดงในภาพที่ 2-3 และ 2-4 โดยมีรายละเอียดขั้นต้น ดังนี้

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง + 5 V

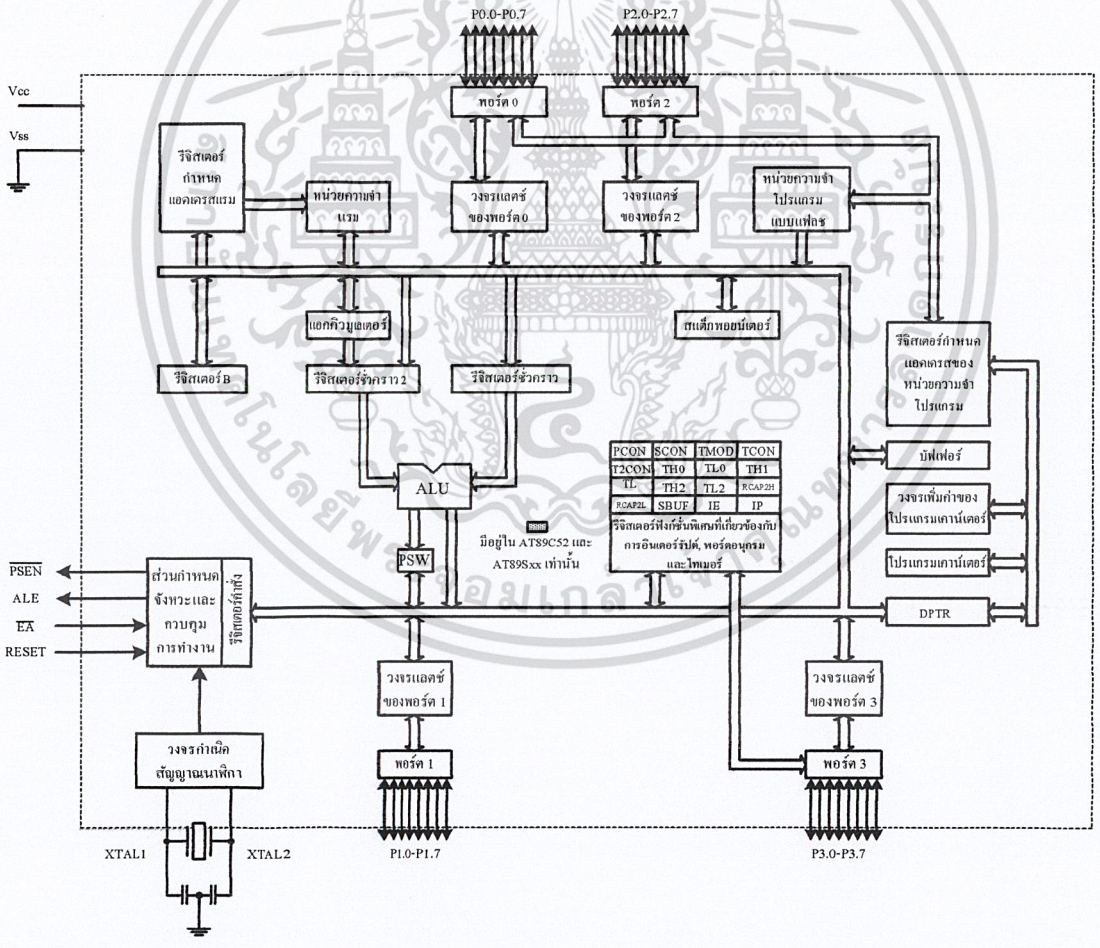
ขา GND เป็นขากราวด์ สำหรับต่อกราวด์ของระบบ

ขาพอร์ต (P 0.0 – P 0.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อ ส่งผลให้พอร์ตนั้นมีสถานะปล่อยลอยจึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของ หน่วยความจำภายนอก (A0 - A7) และขาข้อมูล (D0 - D7)โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

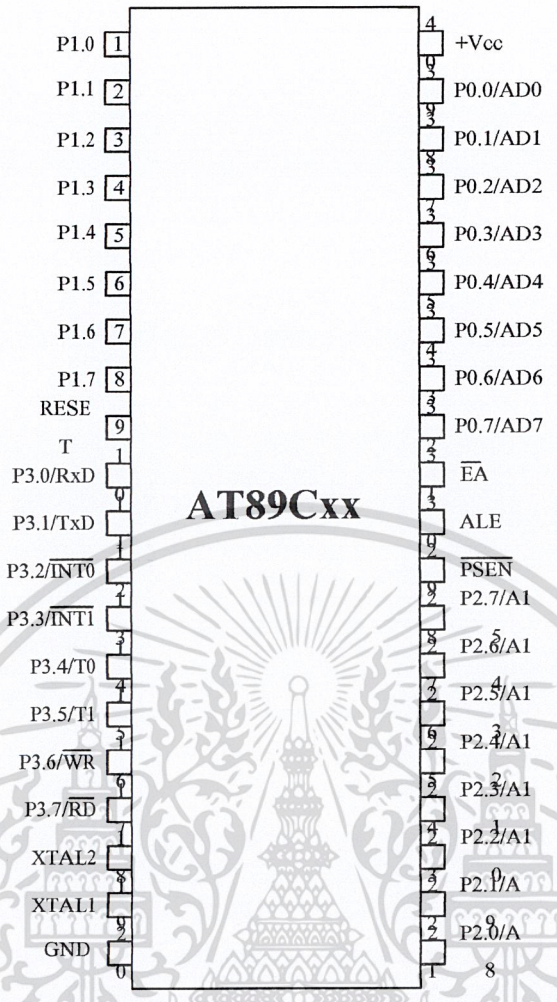
ขาพอร์ต 1 (P1.0 - P1.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

ขาพอร์ต 2 (P2.0 – P2.7)) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อ ส่งผลให้พอร์ตนั้นมีสถานะปล่อยลอยจึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8 - A15)



ภาพที่ 3.7 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS – 51

ขาพอร์ต 3 (P3.0 - P3.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อ ส่งผลให้พอร์ตนั้นมีสถานะปล่อยลอยจึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 0 หรือขา INT0

P3.3 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

เอกสารนี้เป็นสิ่งที่ได้รับการคุ้มครองลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ ต่อสาธารณชน ครั้งที่มีการนำไปใช้

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซต สถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมตชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา ALE / PROG (Address Latch Enable / Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลในไมโครคอนโทรลเลอร์ MCS – 51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อเรียกร้องขอติดต่อของหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณมาที่ขานี้ 2 ครั้งในแต่ละแมตชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใด ๆ ออกมา

ขา EA / Vpp (External Access enable / Program voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าเป็นขานี้เป็น “0” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายนอก แต่ถ้าหากเป็น “1” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำในไมโครคอนโทรลเลอร์ สำหรับในไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม + 12 V

ขา XTAL 1 และ XTAL 2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

3.2.2 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตแบบ 2ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชมีวงจรแลตช์วงจรขับตลอดจนบัฟเฟอร์อินพุต ดังแสดงให้เห็นในสถาปัตยกรรม ภาพที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 หน้าที่พิเศษของพอร์ต 1 ในตัวไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลช

ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.0	AT89C52 / AT89Sxx	ขา T2 เป็นขาอินพุตนับค่าของไทเมอร์ / เคาน์เตอร์ 2 และเป็นขาเอาต์ของวงจรกำเนิดสัญญาณนาฬิกาโดยไทเมอร์ 2
P1.1	AT89C52 / AT89Sxx	ขา T2EX เป็นขาอินพุตทริกเกอร์สำหรับการแคปเจอร์ / รีโหลด และการควบคุมทิศทางของสัญญาณ
P1.4	AT89Sxx	ขา SS (Slave Select) เป็นขาเลือกการติดต่อในกรณีไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบการติดต่อแบบ SPI
P1.5	AT89Sxx	ขา MOSI (Master dada output Slave dada input) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT89Sxx	ขา MISO (Master dada input Slave dada input) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

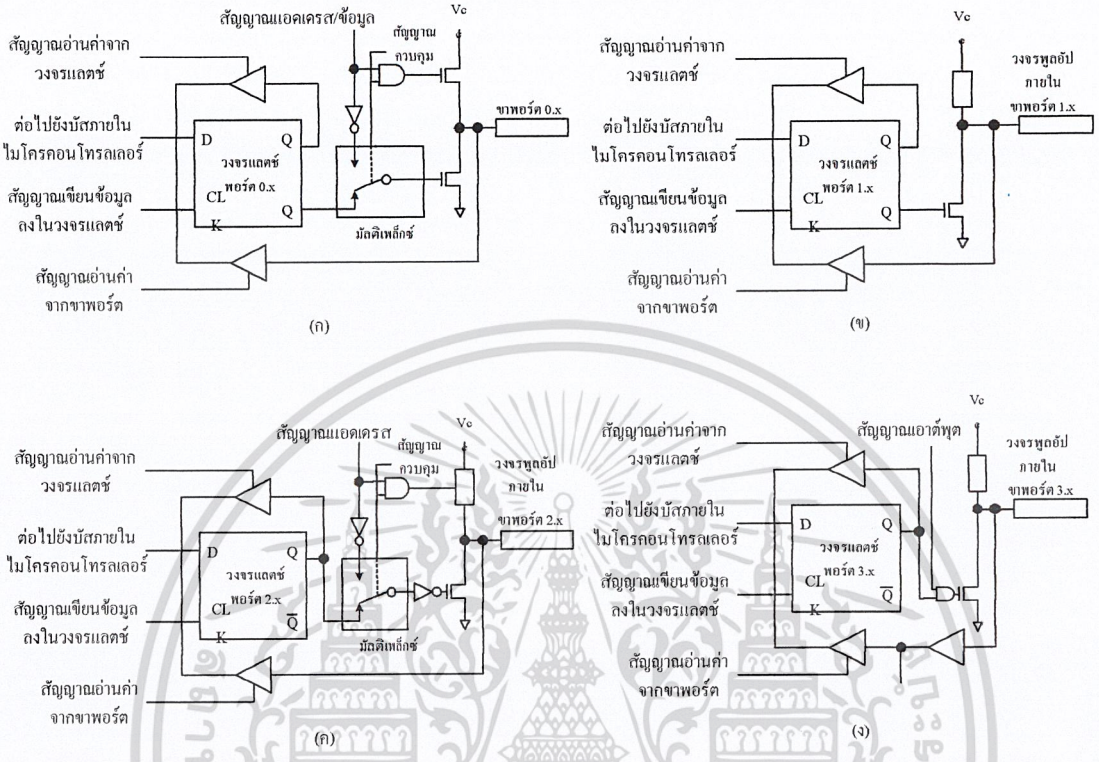
ที่พอร์ต 0 และพอร์ต 2 จะใช้งานและพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3ทั้งพอร์ตและพอร์ต 1 บางขานอกจากการใช้งานเป็นขาอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชเบอร์ใด ดังสรุปได้ในตารางที่ 2-2

ในภาพที่ 3.9 แสดงวงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชโดยในภาพที่ 2-5 (ก) เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของวงจรสามารถกระทำได้อย่างอิสระต่อกันด้วยสัญญาณที่แยกจากกัน นั่นเอง สัญญาณอ่านข้อมูลจากพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิฟลอปในขณะที่ข้อมูลจะส่งผ่านมาทางขาบัสข้อมูลภายใน เข้าสู่ขา D ของดีฟลิฟลอป MCS – 51

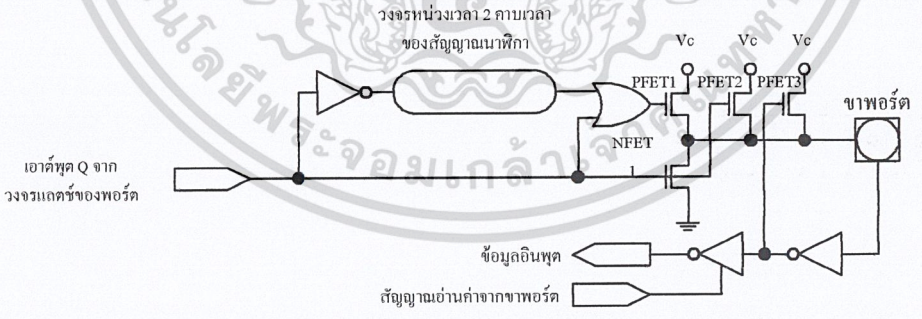
ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่า ต้องการใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตปกติหรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกที่ขาพอร์ต 0 ทุกขาด้วย



ภาพที่ 3.9 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลช



วงจรพูลอัปประกอบด้วยเฟตชนิดพีแชนแนล 3 ตัวคือ PFET1-PFET3 โดย NFET1 จะทำงานเมื่อได้รับลอจิก "1" จากขา Q และหยุดทำงานเมื่อ ได้รับลอจิก "0" วงจรพูลอัปจะเริ่มต้นทำงานเมื่อ NFET1 ได้รับลอจิก "1" PFET1 จะทำงานนานประมาณ 2 คาบเวลาของสัญญาณนาฬิกาภายใน หลังจากที่เกิดการเปลี่ยนแปลงจากลอจิก "0 " เป็นลอจิก "1" ในขณะที่ PFET1 ทำงานจะทำให้ PFET3 ทำงานตามไปด้วย ทำให้เกิดการพูลอัป ขาพอร์ต

ภาพที่ 3.10 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในภาพที่ 3.9 (ข) เป็นวงจรพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์เนื่องจากพอร์ตนี้จะไม่มีใช้ในการติดต่อกับหน่วยความจำภายนอกแต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน สำหรับรายละเอียดของวงจรพูลอัปแสดงในภาพที่ 2-6

ในภาพที่ 3.9 (ค) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรพูลอัปเพิ่มเติมเข้ามา ส่วนภาพที่ 3.9 (ง) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์ และวงจรอินพุตเอาต์พุตเมื่อทำงานในฟังก์ชั่นพิเศษเข้ามาเนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

3.2.3 การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุตต้องเริ่มต้นด้วยการเขียนข้อมูล “1” มาที่แต่ละบิต ของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้น ๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรงส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น “1” สามารถสัญญาณลอจิก “0” จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจนบัฟเฟอร์ภายในพอร์ตแล้วรอให้ซีพียูมาอ่านค่าเข้าไปเมื่อเป็นเช่นนี้อุปกรณ์ภายนอกเชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์MCS – 51 แบบแฟลชควรกำหนดให้การทำงานในสภาวะลอจิก “ 0” จะดีและสะดวกที่สุด

3.2.4 การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้วขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายละตรงไปตรงมากล่าวคือเมื่อต้องการส่งข้อมูล “0” ออกไป ทางเอาต์พุตก็ให้เขียนข้อมูล “0” ไปยังวงจรแลตช์ซึ่งก็จะส่งต่อไปขับเฟตทำให้เฟตทำงาน ขาพอร์ตที่กำหนดให้ทำงานก็จะเกดลอจิก “0” ขึ้น ซึ่งคล้ายกับการกำหนดเพียงแตกต่างกัน ที่กระบวนการในการเคลื่อนย้ายข้อมูลโดยถ่ายข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การอ่านค่าลอจิกจากพอร์ต

ในไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชสามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะคือ อ่านจากขาพอร์ตโดยตรงและอ่านจากวงจรแลตช์ของแต่ละพอร์ต

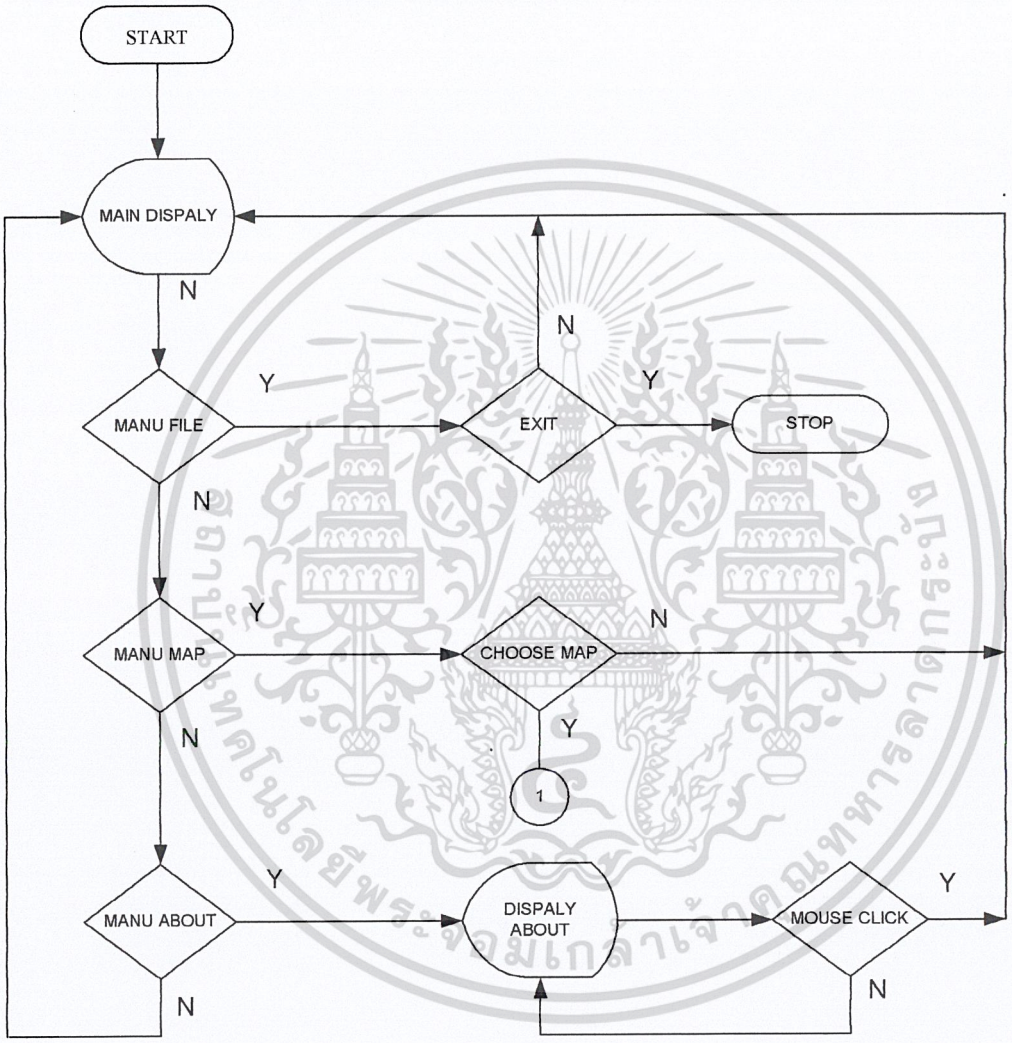
ในกรณีที่พอร์ตต่อกับขาเบสทรานซิสเตอร์ชนิด NPN และขาอิมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล “1” ไปยังทรานซิสเตอร์ จะทำให้ทรานซิสเตอร์ทำงานสถานะลอจิกที่ขาพอร์ตจะเป็น “0” เนื่องจากเมื่อทรานซิสเตอร์ทำงานเสมือนว่าขาพอร์ตนั้น ต่อลงกราวด์ทำให้ค่าลอจิกที่ขาพอร์ตจะได้ผลตรงกันข้ามกับที่ส่งออกมาแต่ถ้าหากทำงาน อ่านค่าลอจิกที่วงจรแลตช์จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริงดังนั้น ในการค่าลอจิกพอร์ต จึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

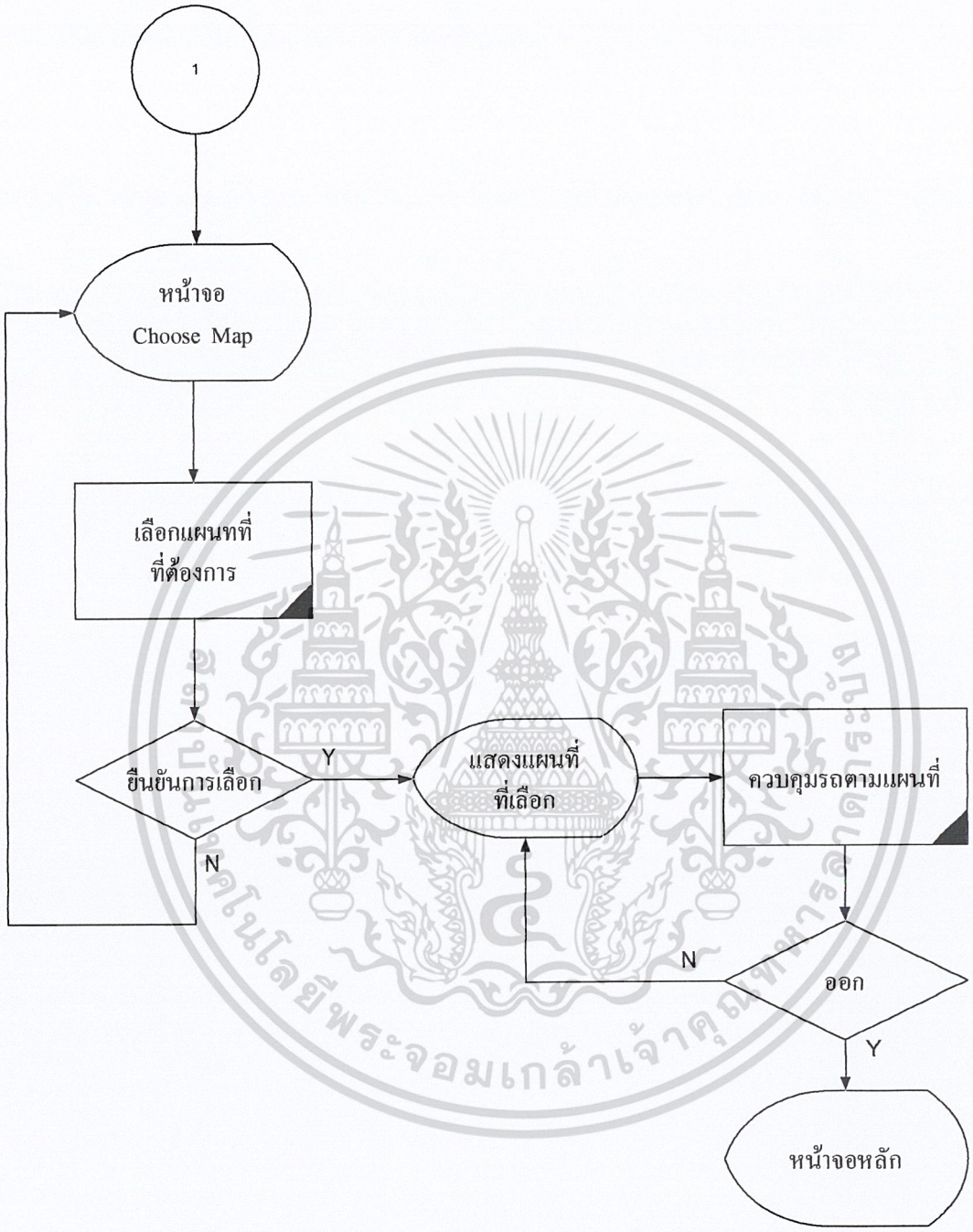
4.1 Flow Chart ส่วนของ Main Program



ภาพที่ 4.1 Flow chart แสดงการทำงานของ Main Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

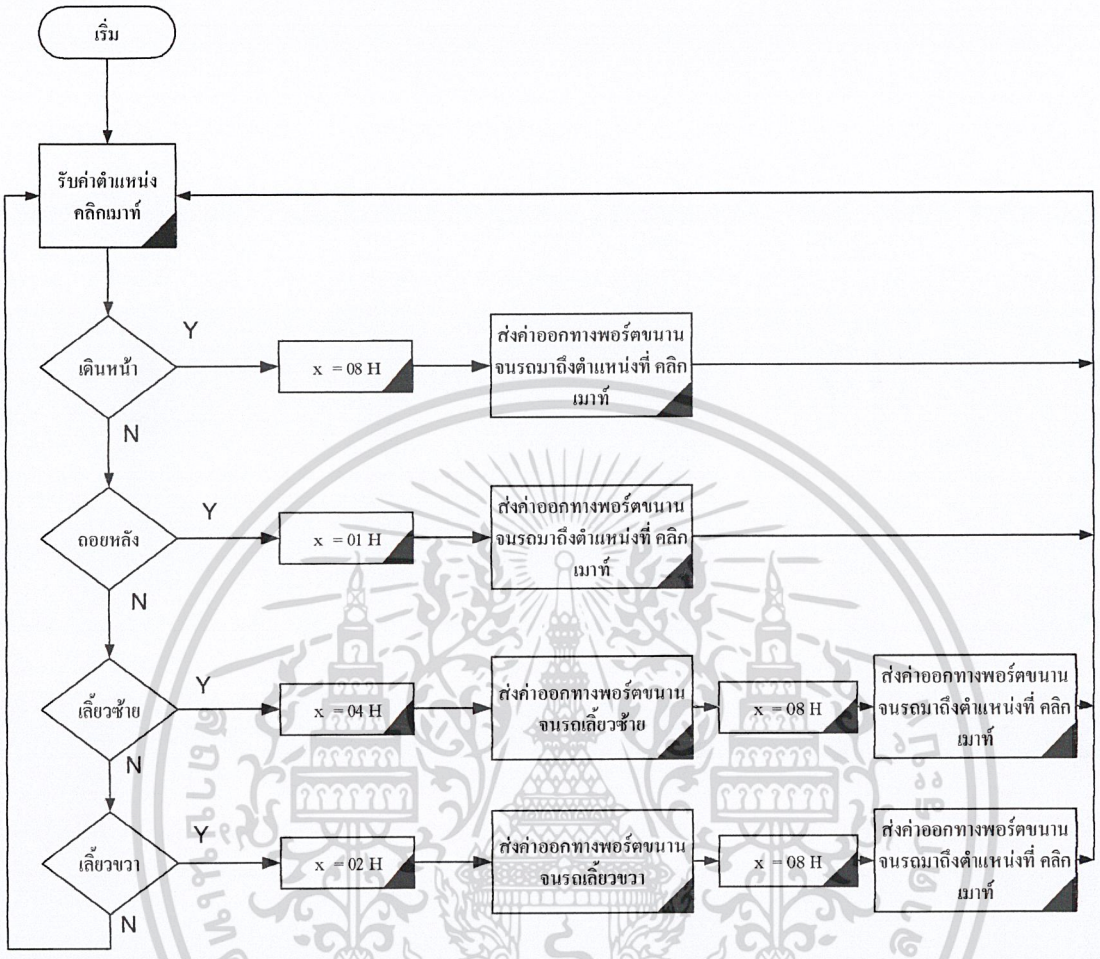
4.2 Flow Chart ส่วนของ Sub Program



ภาพที่ 4.2 Flow Chart แสดงส่วนของ Choose Map ใน Sub Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 Flow Chart ส่วนของการควบคุมรถ

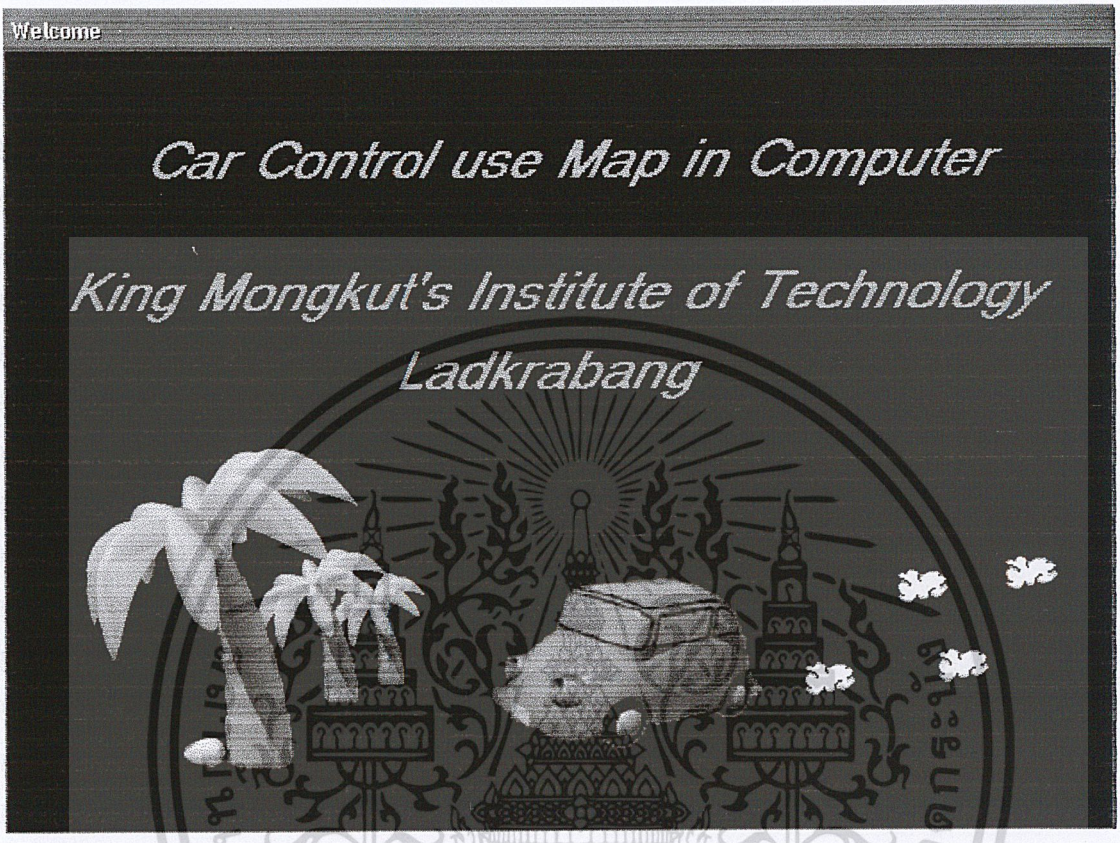


ภาพที่ 4.3 Flow Chart แสดงส่วนของการควบคุมรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การแสดงหน้าจอของการเรียก Program

Welcome screen



ภาพที่ 4.4 แสดงหน้าจอต้อนรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การแสดงหน้าจอของการเรียก Main Program

ภาพที่ 4.5 แสดงหน้าจอของ Main Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การแสดงหน้าจอของการเรียก Main File

ภาพที่ 4.6 แสดงหน้าจอของ Main File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การแสดงหน้าจอของการเรียก Main Map

ภาพที่ 4.7 แสดงหน้าจอของ Main Map

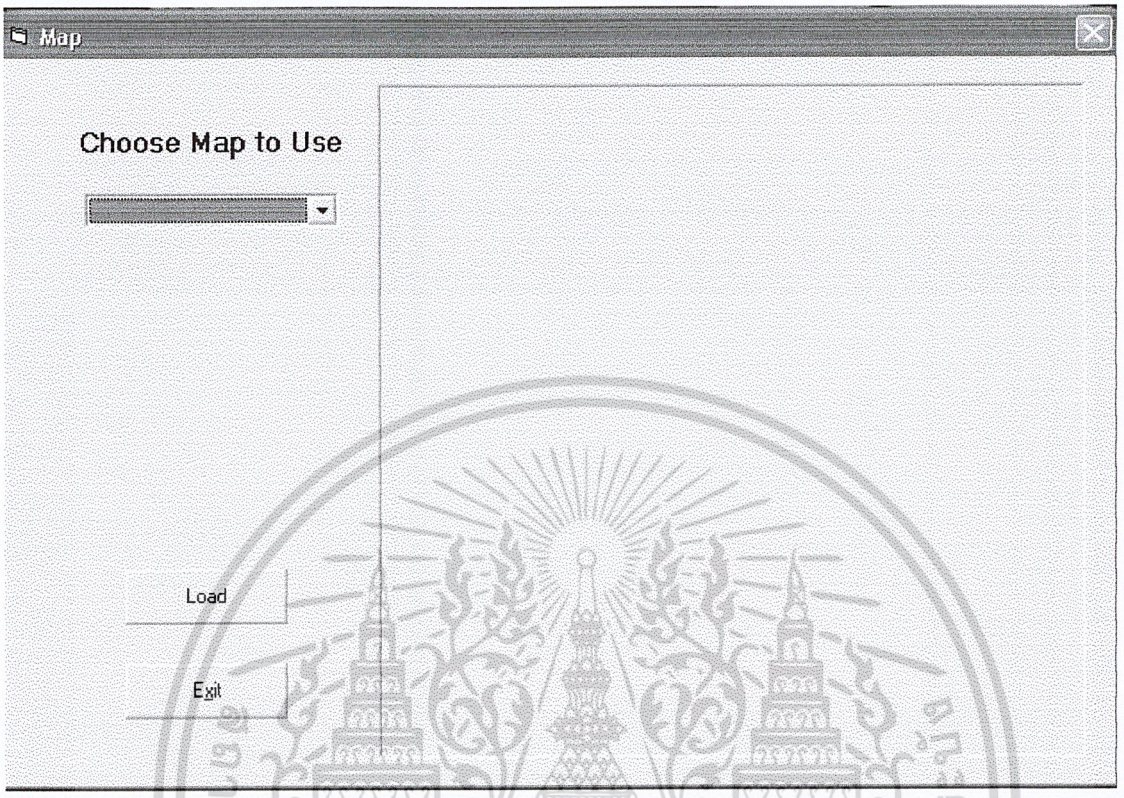
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 การแสดงหน้าจอของการเรียก Main About

ภาพที่ 4.8 แสดงหน้าจอของ Main About

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การแสดงหน้าจอของการเรียก Map



ภาพที่ 4.9 แสดงหน้าจอของการเรียก Map

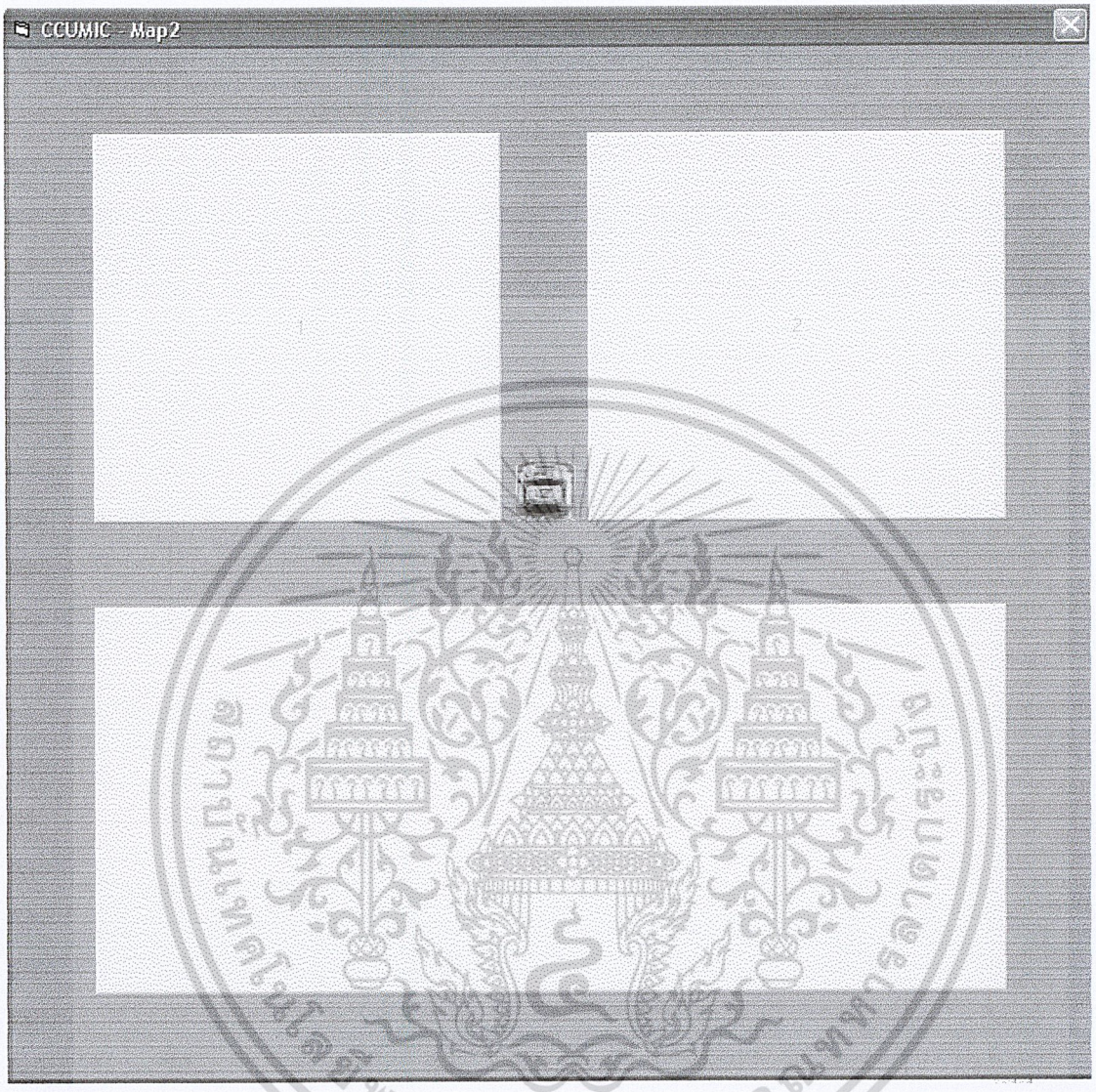
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10 การแสดงหน้าจอของการเรียก Map 1

ภาพที่ 4.10 แสดงหน้าจอของ Map 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

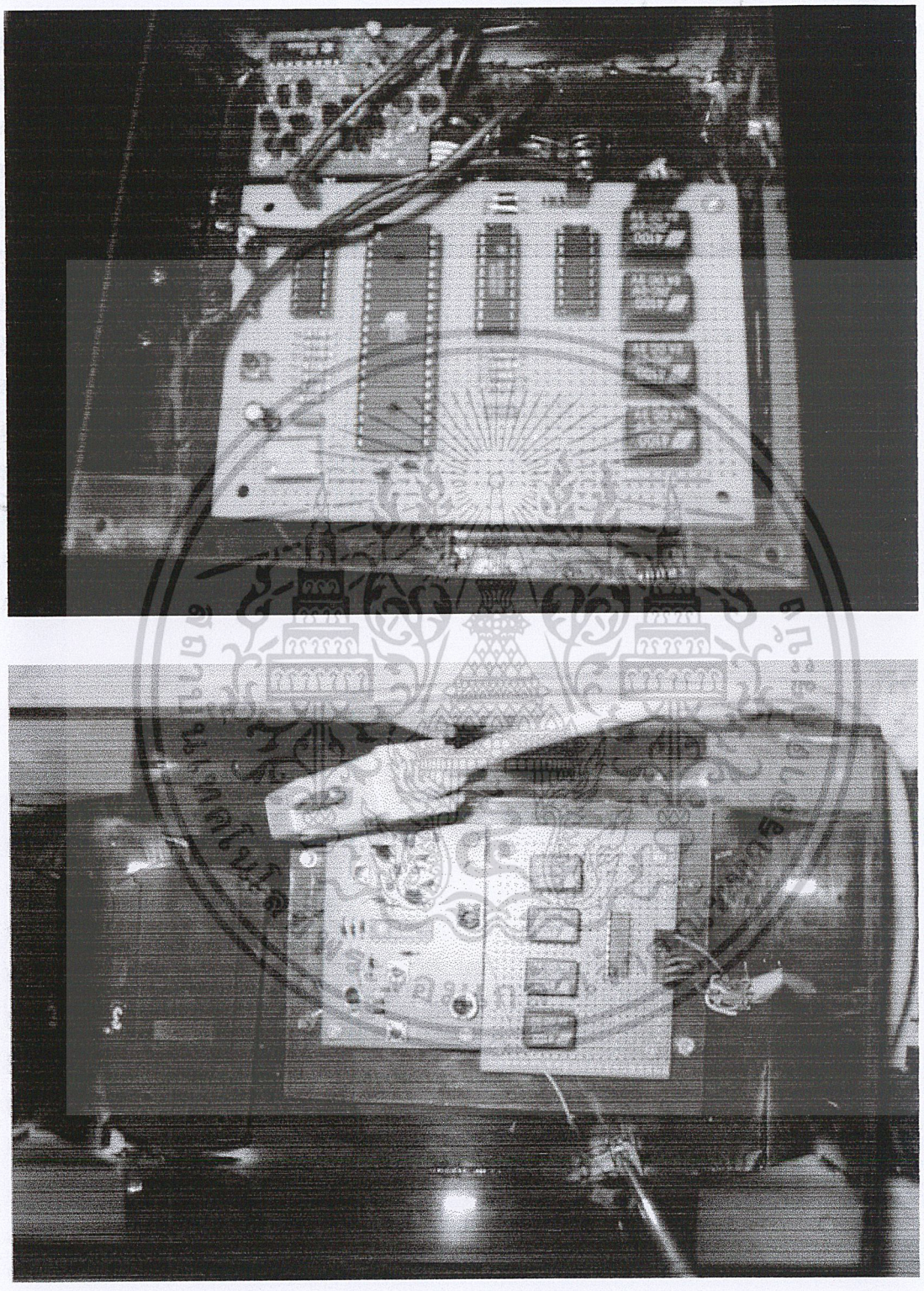
4.11 การแสดงหน้าจอของการเรียก Map 2



ภาพที่ 4.11 แสดงหน้าจอของ Map 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.12 รูปชิ้นงาน



ภาพที่ 4.12 แสดงภาพชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและปัญหา

จากผลการทดลอง เราสามารถใช้แผนที่ที่สร้างขึ้นในคอมพิวเตอร์มาควบคุมรถบังคับที่ออกแบบสร้างขึ้นมาได้ โดยใช้พอร์ตขนานจากคอมพิวเตอร์ต่อกับชุดส่งในการควบคุมรถบังคับ

ปัญหาที่เกิดขึ้นในการใช้งาน คือ แผนที่ที่ใช้ยังไม่มีมาตรฐานเข้ามาควบคุมว่าควรใช้ขนาดและอัตราส่วนเป็นเท่าใด ดังนั้นในการสร้างแผนที่จึงแล้วแต่ผู้สร้างแผนที่เอง และอีกปัญหาที่พบคือ ระยะทางในการส่งและรับสัญญาณระหว่างชุดรับและชุดส่งค่อนข้างใกล้

ข้อเสนอแนะแนวคิดในการพัฒนา จากปัญหาที่เกิดขึ้นมีดังนี้

1. แผนที่ในคอมพิวเตอร์ ควรสามารถกำหนดอัตราส่วนได้ ซึ่งจะทำให้สามารถนำเอาสถานที่จริงที่เราต้องการนำรถไปใช้งานมาออกแบบแผนที่ได้อย่างเที่ยงตรงและแม่นยำ

2.การออกแบบ ชุดรับ ชุดส่ง ให้สามารถทำงานได้ในระยะทางไกล ๆ เนื่องจากอุปกรณ์ที่ใช้อยู่เป็นชุดรับส่งชุดเล็ก ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ข้อมูลของไมโครคอนโทรเลอร์

Features

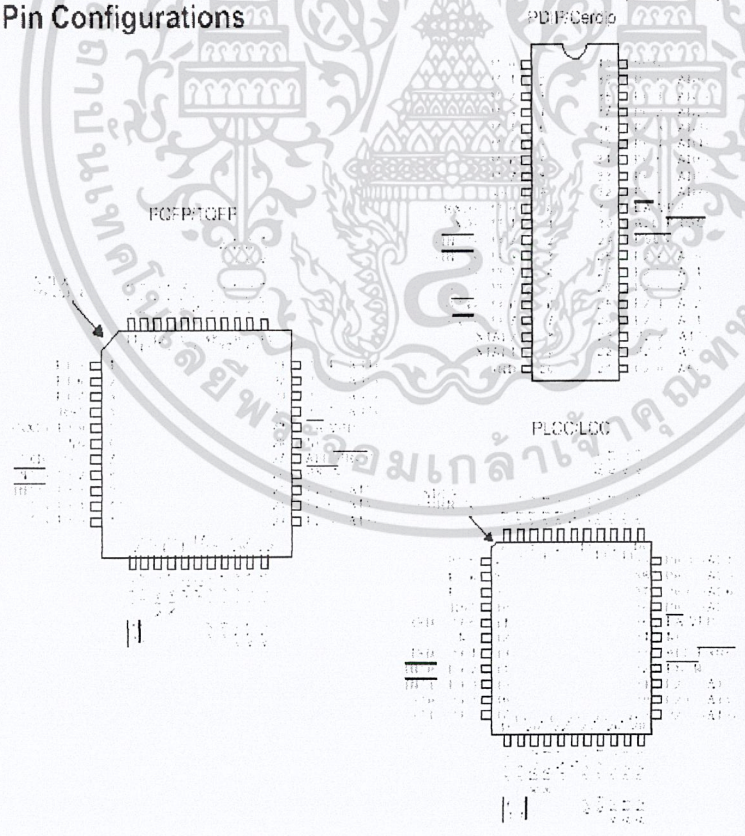
- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt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

Description

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

(continued)

Pin Configurations

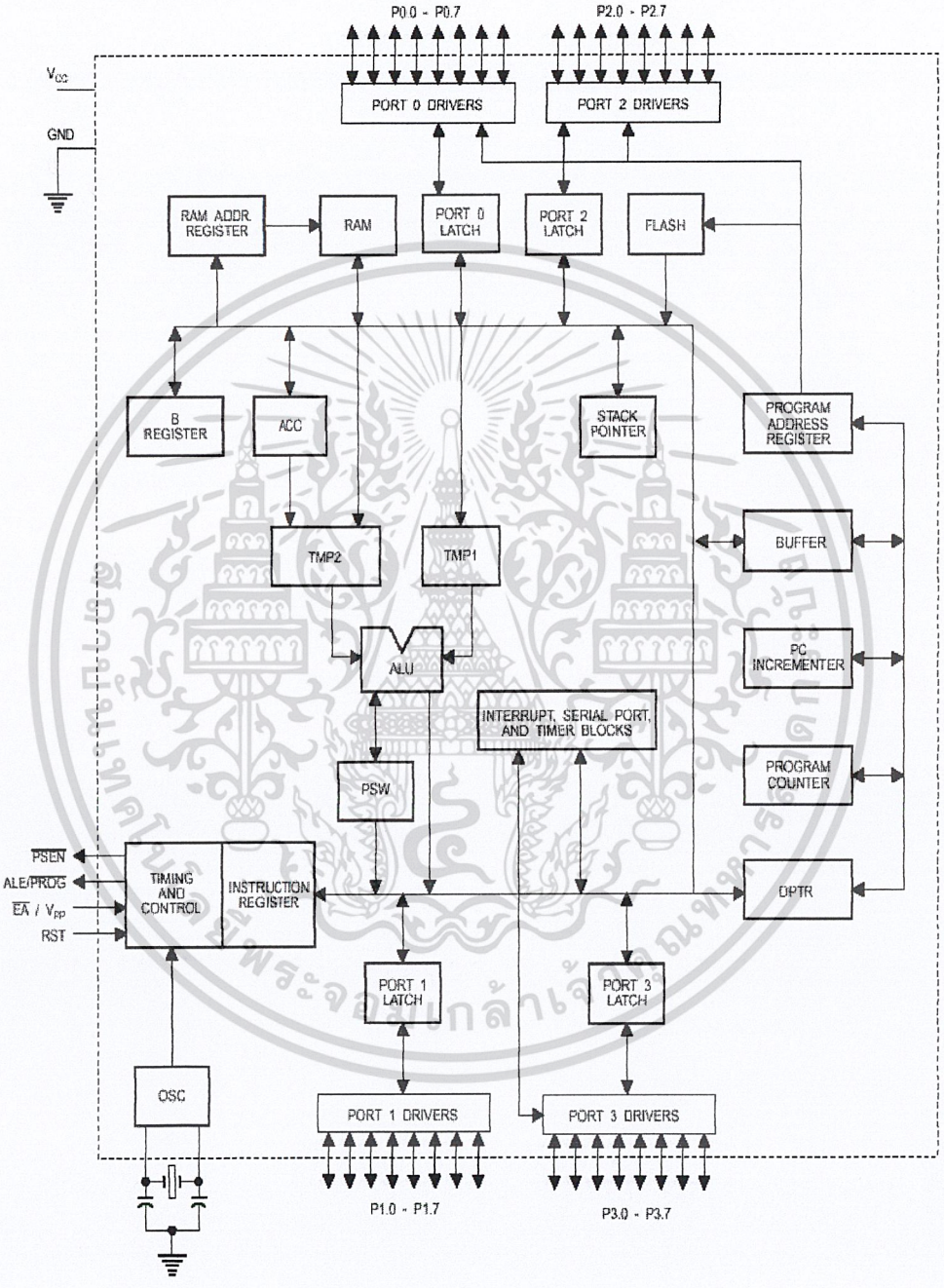


8-Bit Microcontrol with 4 Kbyte: Flash

AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Description (Continued)

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

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

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

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

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

Port 1

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

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

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{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 it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

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

Port 3

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

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

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

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

RST

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

ALE/$\overline{PROG}$

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

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

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

$\overline{PSEN}$

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

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

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

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

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

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

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

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

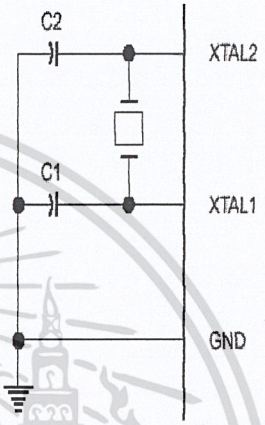
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

Idle Mode

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

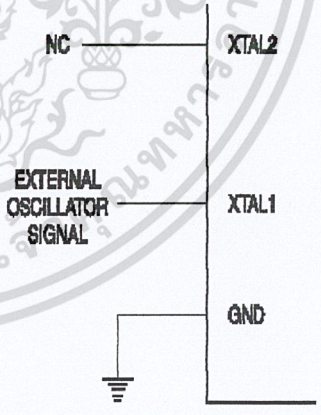
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hard-

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	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

เอกสารนี้เป็นเอกสารที่สงวนสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

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

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

Lock Bit Protection Modes

	Program Lock Bits			
	LB1	LB2	LB3	Protection Type
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 is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

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

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

	V_{PP} = 12 V	V_{PP} = 5 V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash (Continued)

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

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

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

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

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

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

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12 V programming
(032H) = 05H indicates 5 V programming

Programming Interface

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

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

Flash Programming Modes

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

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.
2. Chip Erase requires a 10 ms PROG pulse.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Programming the Flash

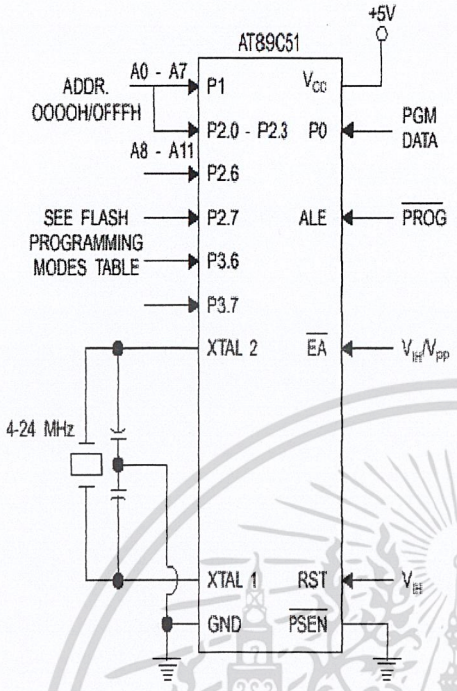
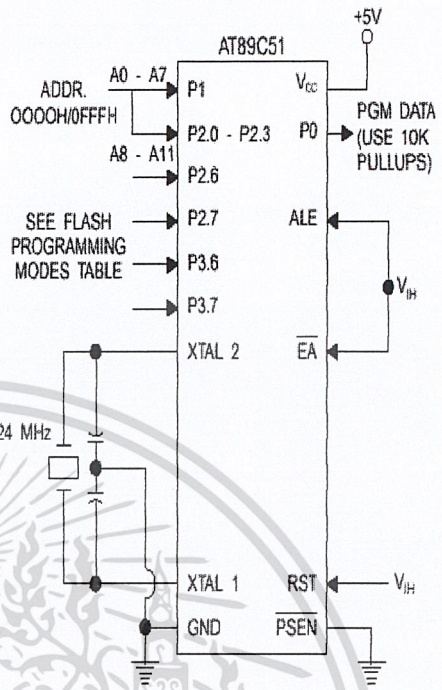


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

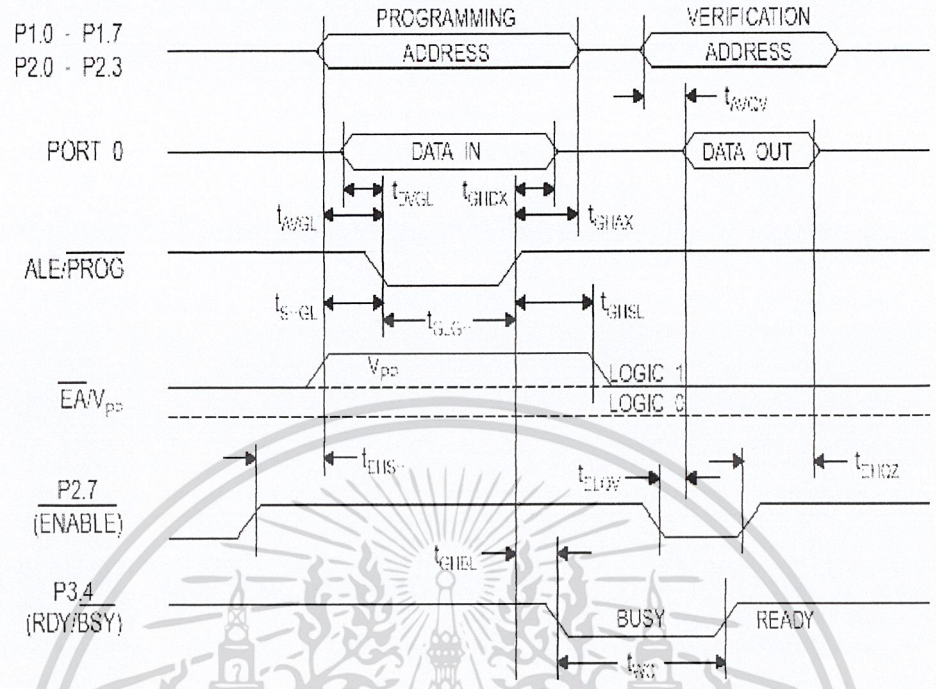
T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}[1]	Programming Enable Voltage	11.5	12.5	V
I_{PP}[1]	Programming Enable Current		1.0	mA
1/t_{CLCL}	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ 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_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48tCLCL		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}[1]	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		48tCLCL	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		48t_{CLCL}	
t_{EHQV}	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

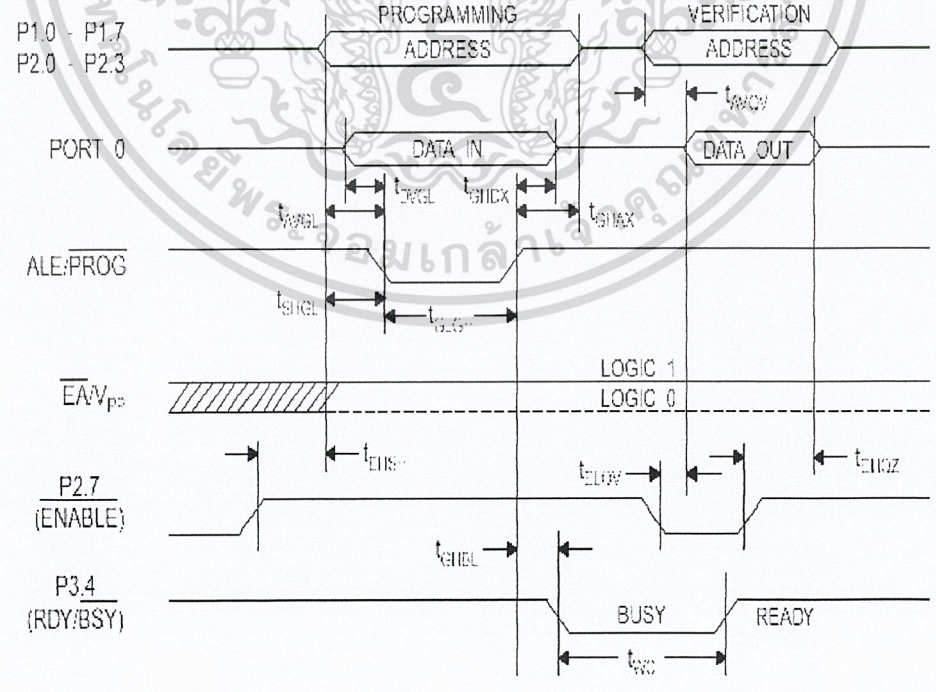
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

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

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

D.C. Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0 V ±20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-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 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 uA, V_{CC} = 5 V ±10%	2.4		V
		I_{OH} = -25 uA	0.75 V_{CC}		V
		I_{OH} = -10 uA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 uA, V_{CC} = 5 V ±10%	2.4		V
		I_{OH} = -300 uA	0.75 V_{CC}		V
		I_{OH} = -80 uA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2 V		-650	µA
I_{LI}	Input Leakage Current (Port 0, EA)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode[2]	V_{CC} = 6 V		100	µA
		V_{CC} = 3 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 IOL for all output pins: 71 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power Down is 2 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. Characteristics

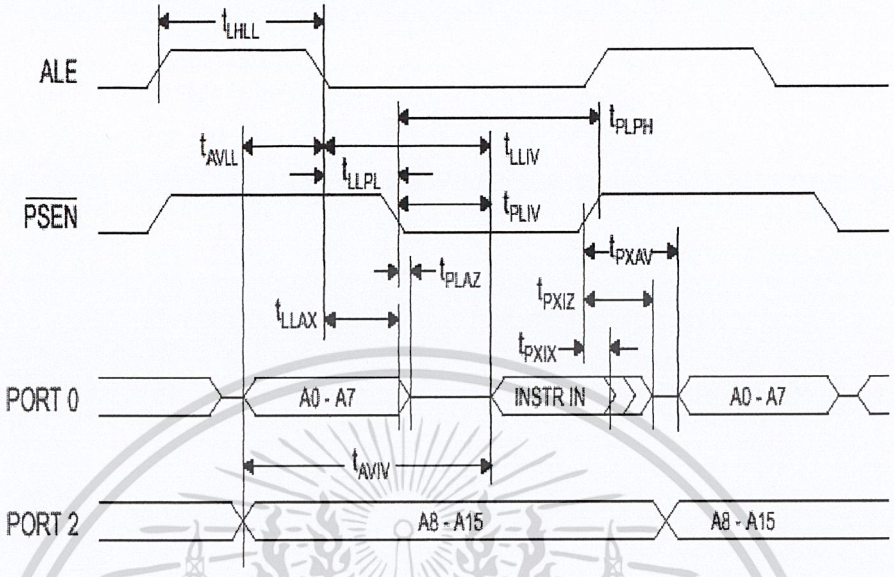
(Under Operating Conditions; Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

External Program and Data Memory Characteristics

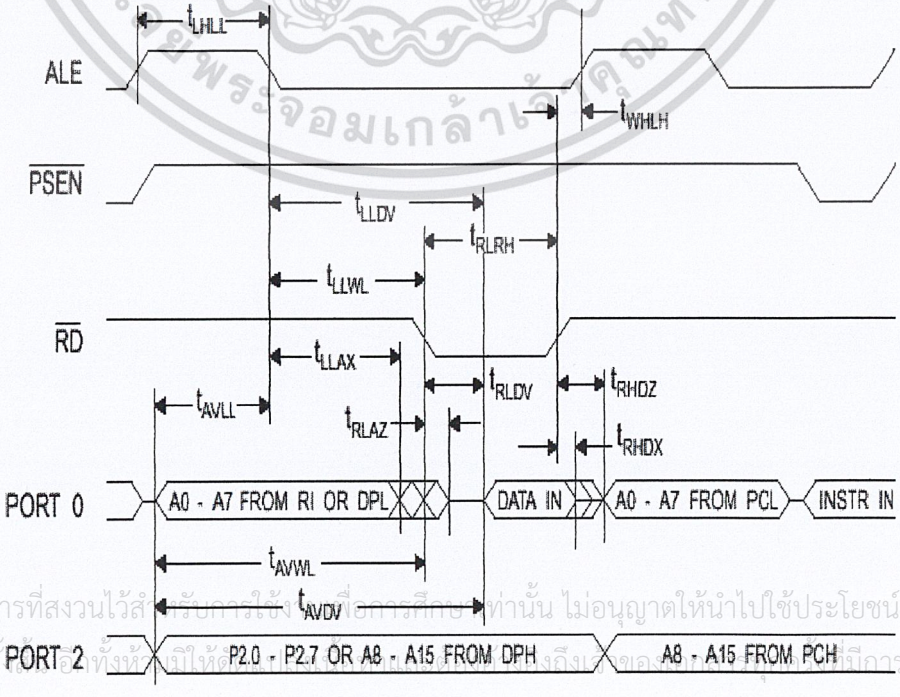
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	28		tCLCL-13		ns
t_{LLAX}	Address Hold After ALE Low	48		tCLCL-20		ns
t_{LLIV}	ALE Low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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

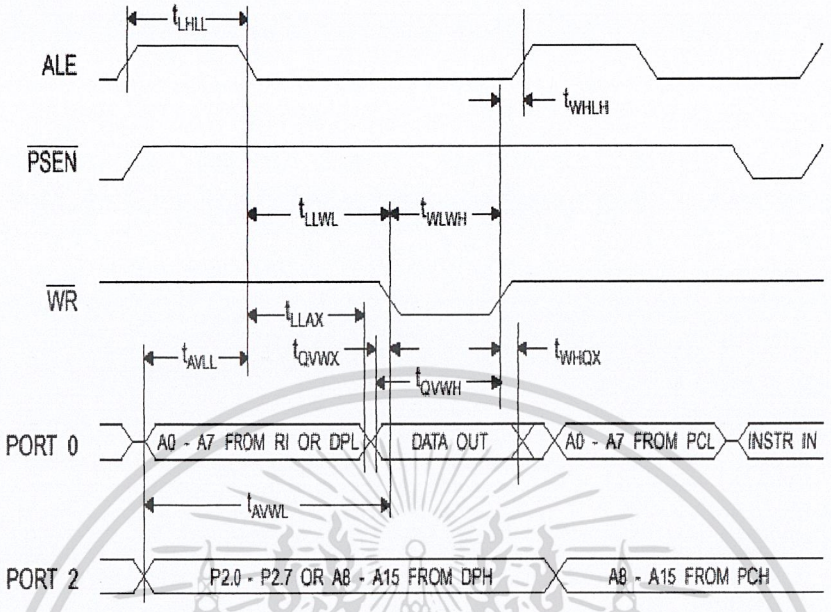


External Data Memory Read Cycle

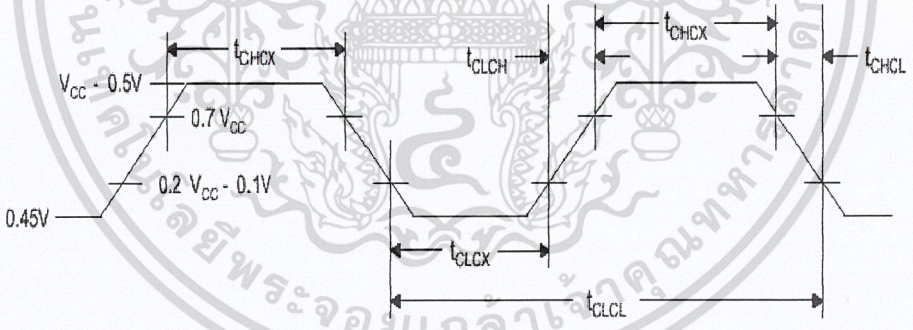


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

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

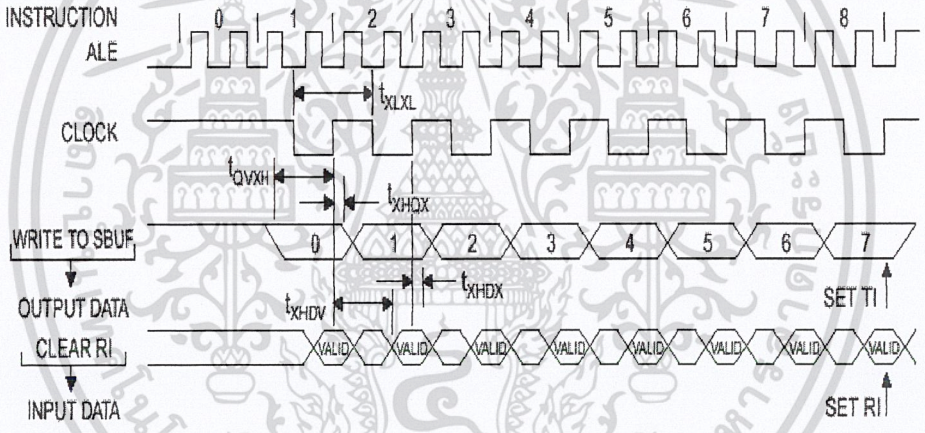
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

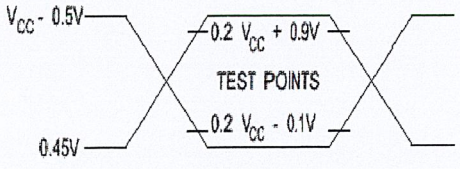
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

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_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		10tCLCL-133	ns

Shift Register Mode Timing Waveforms

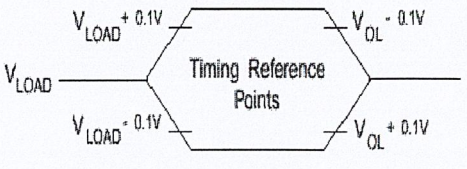


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V 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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

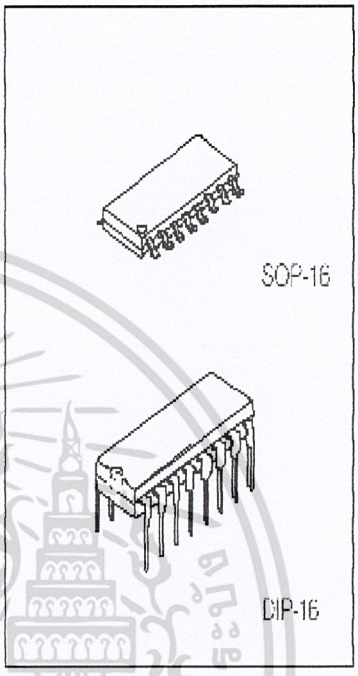
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5 V ± 20%	AT89C51-24AC AT89C51-24JC AT89C51-24PC AT89C51-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89C51-24AI AT89C51-24JI AT89C51-24PI AT89C51-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลของ ไอซี ULN 2003

UTC ULN2003 LINEAR INTEGRATED CIRCUIT

SEVEN DARLINGTON ARRAYS



DESCRIPTION

The UTC ULN2003 is high-voltage, high-current darlington drivers comprised of seven NPN darlington pairs.

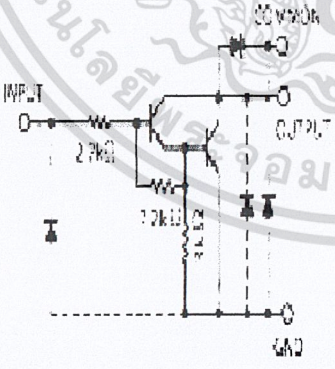
FEATURES

*Output current (single output) 500mA MAX.
*High sustaining voltage output 50V MIN.
*Output clamp diodes
*Inputs compatible with various types of logic.

APPLICATIONS

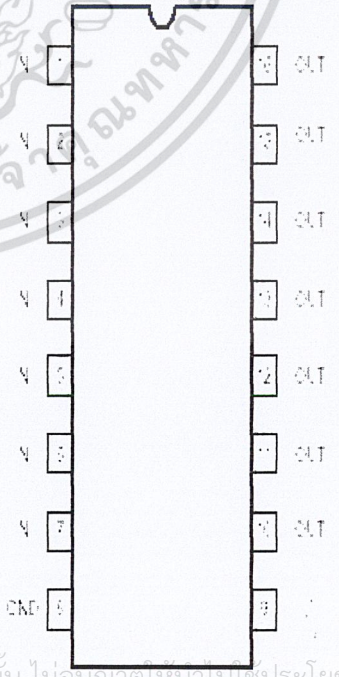
*Relay, hammer, lamp and display (LED) drivers

SCHEMATICS (EACH DRIVER)



Note: The input and output parasitic diodes cannot be used as clamp diodes.

PIN CONFIGURATIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTC ULN2003 LINEAR INTEGRATED CIRCUIT

ABSOLUTE MAXIMUM RATINGS

PARAMETER	SYMBOL	VALUE	UNIT
Input Voltage	V_{IN}	-0.5~30	V
Output Sustaining Voltage	$V_{CE(SUS)}$	-0.5~50	V
Output Current	$I_{O(1)}$	500	mA/ch
Clamp Diode Reverse Voltage	V_R	50	V
Clamp Diode Forward Current	I_F	500	mA
Power Dissipation	P_D	DIP 1.47 SOP 0.54(0.625)(Note)	W
Operating Ambient Temperature Range	T_{OPR}	-40 °C ~+85	C
Storage Temperature Range	T_{STG}	-55 to +150	C

Note: On glass epoxy PCB (30x30x1.6mm, Cu 50%)

ELECTRICAL CHARACTERISTICS (Ta = 25 °C, Unless otherwise specified)

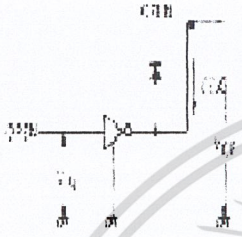
PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT	FIG
Output Leakage Current	$I_{O(OFF)}$	$V_{O(1)}$=50V, T_a=25 °C $V_{O(1)}$=50V, T_a=85 °C			50 100	μA	1
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	$I_{O(1)}$=350mA, I_{IN}=500μA $I_{O(1)}$=200mA, I_{IN}=350μA $I_{O(1)}$=100mA, I_{IN}=250μA		1.3 1.1 0.9	1.6 1.3 1.1	V	2
Input Current (output on)	$I_{I(ON)}$	V_{IN}=2.4V, I_O=350mA		0.4	0.7	mA	3
Input Current (output off)	$I_{I(OFF)}$	$I_{O(1)}$=500μA, T_a=85 °C	50	65		μA	4
Input Voltage (output on)	$V_{I(ON)}$	$V_{O(1)}$=2.0V, h_{FE}=500, $I_{O(1)}$=350mA $I_{O(1)}$=200mA			2.8 2.9	V	5
Clamp Diode Reverse Current	I_R	V_R=50V, T_a=25 °C V_R=50V, T_a=85 °C			50 100	μA	6
Clamp Diode Forward Voltage	V_F	I_F=350mA			2.0	V	7
Input Capacitance	C_I			15		pF	-
Turn-On Delay	t_{ON}	$V_{O(1)}$=50V, R_L=125Ω, C_L=15pF		0.1		μS	8
Turn-Off Delay	t_{OFF}	$V_{O(1)}$=50V, R_L=125Ω, C_L=15pF		0.2		μS	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

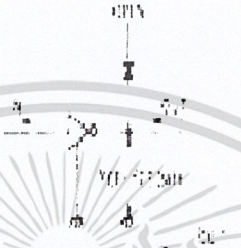
UTC ULN2003 LINEAR INTEGRATED CIRCUIT

TEST CIRCUIT

1. I_{CEX}

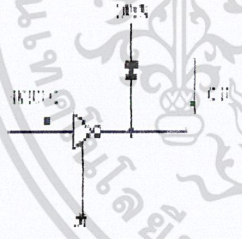


2. $V_{CE(sat)}$, h_{FE}



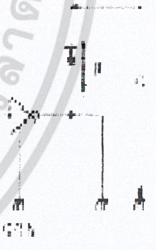
3. $I_{I(ON)}$

4. $I_{I(OFF)}$

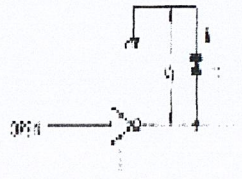


5. $V_{I(ON)}$

6. I_R

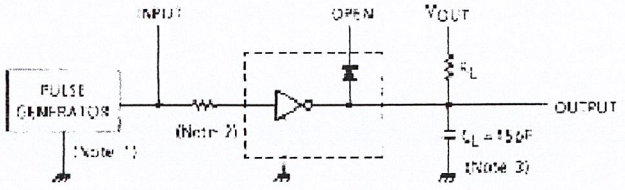


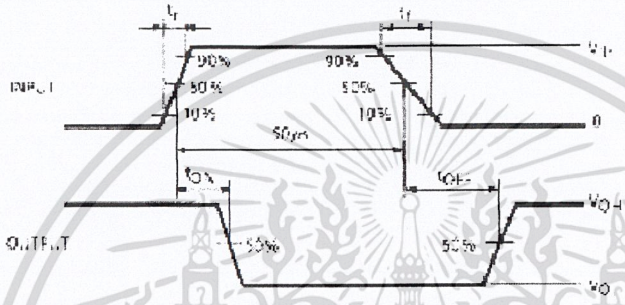
7. V_F



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ton, toff

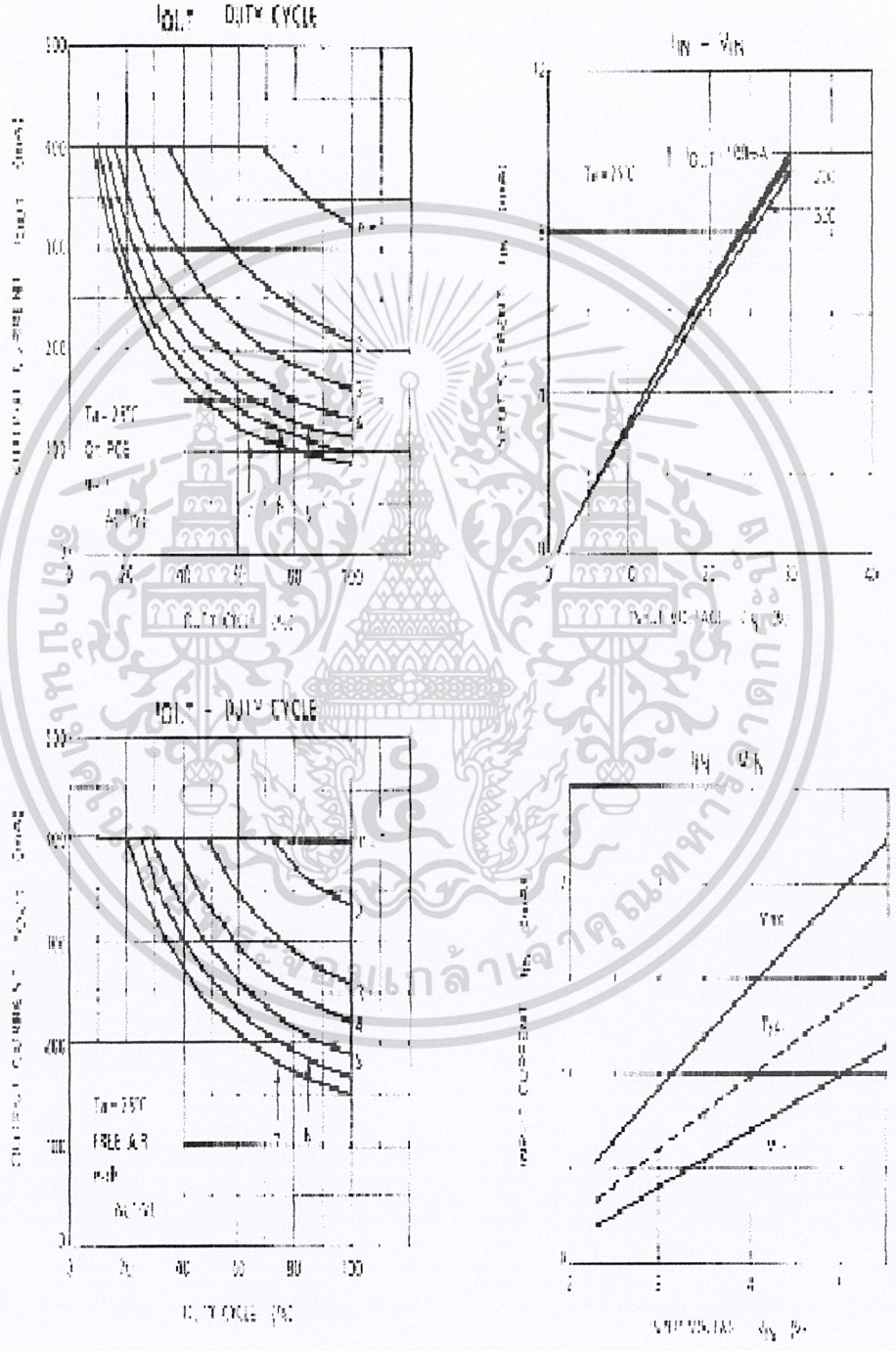




Note1. Pulse width 50μs, duty cycle 10%
Output impedance 50Ω, tr=5ns, tf=10ns
Note2. RL=0, V+=3V
Note3. CL includes probe and jig capacitance

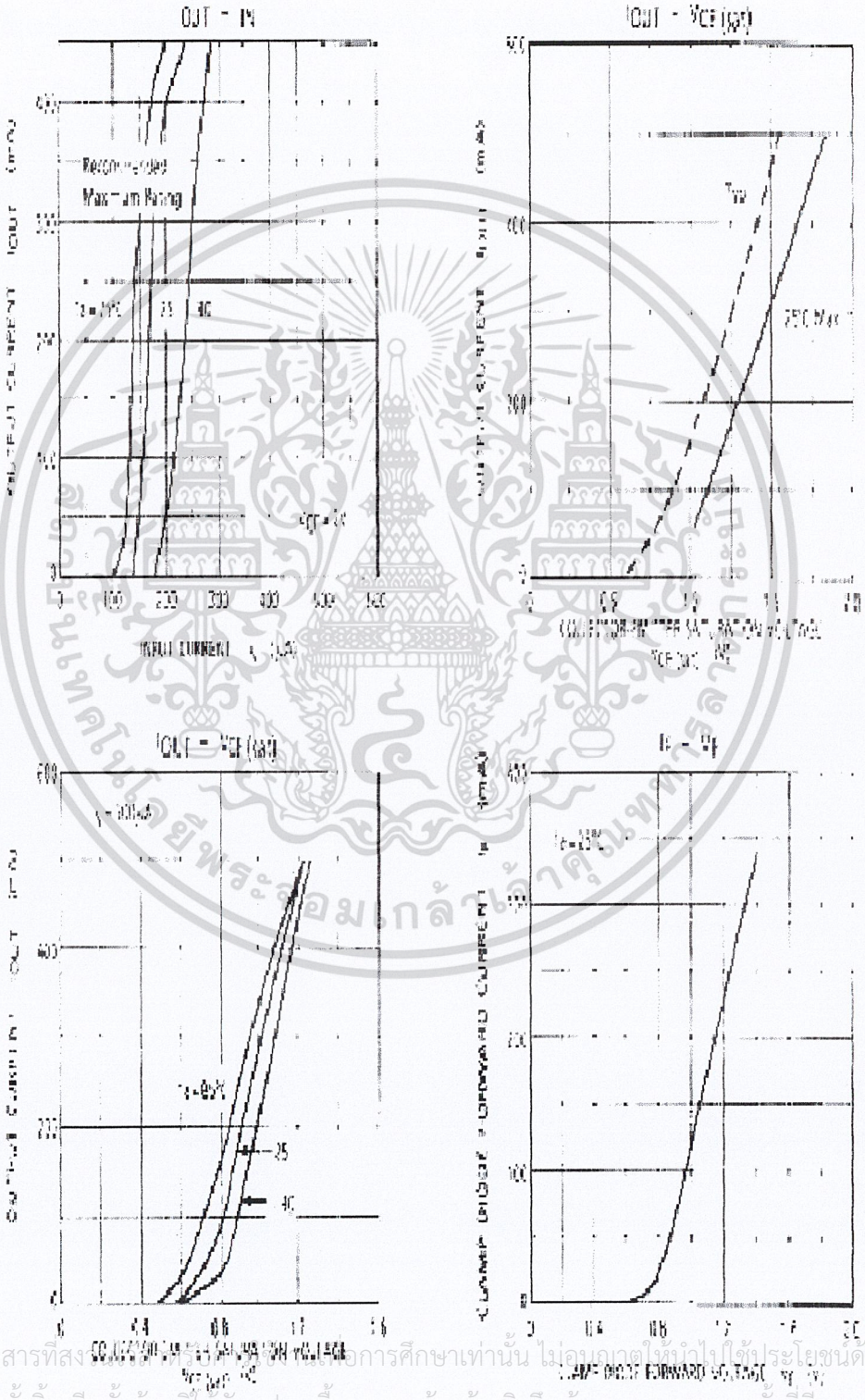
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTC ULN2003 LINEAR INTEGRATED CIRCUIT



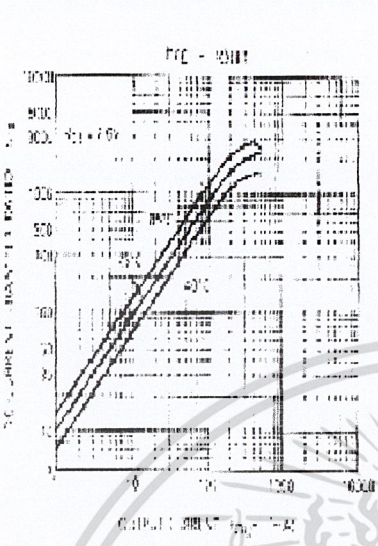
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

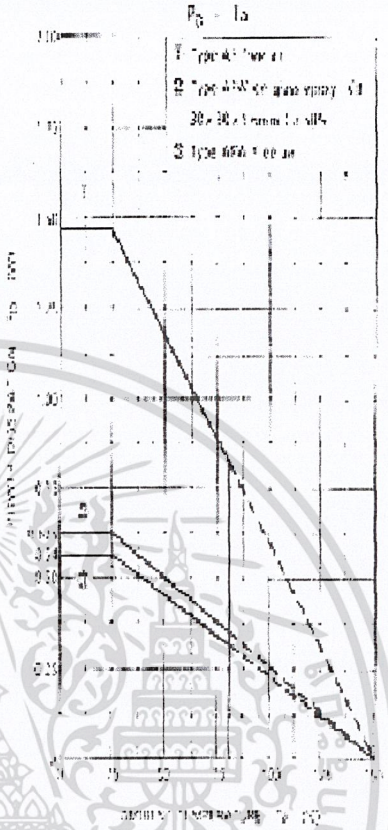
UTC ULN2003 LINEAR INTEGRATED CIRCUIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTC ULN2003 LINEAR INTEGRATED CIRCUIT





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้