สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจำลองกระบวนการด้วยภาพสำหรับระบบ PLC

VISUALIZE PROCESS SIMULATION FOR PLC SYSTEM

T055058

นายขจรจักร สมสมัย

นายดนัย ไชยสาลี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่..........

เลขทะเบียน.....55058.....

วัน,เดือน,ปี.....7 เม.ย. 2548.....

b.....

i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VISUALIZE PROCESS SIMULATION FOR PLC SYSYTEM

KACHONCHAK SOMSAMAI

DANAI CHAISALEE

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

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

....

หัวข้อปริญญานิพนธ์	การจำลองกระบวนการด้วยภาพสำหรับระบบ PLC VISUALIZE PROCESS SIMULATION FOR PLC SYSTEM	
นักศึกษาผู้จัดทำ	นายขจรจักร สมสมัย	**รหัสประจำตัว** 44015508
	นายดนัย ไชยสาลี	**รหัสประจำตัว** 44015516
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมการวัดคุม	
ปีการศึกษา	2546	

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผศ.ทวีพล ซื่อสัตย์	

วัน/เดือน/ปีที่สอบ วันพุธที่ 24 มีนาคม พ.ศ.2547
สถานที่สอบ ณ.ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(รศ. ประสิทธิ์ จุลเสรีวงศ์)
หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การจำลองกระบวนการด้วยภาพสำหรับระบบ PLC

VISUALIZE PROCESS SIMULATION FOR PLC SYSTEM

นักศึกษาผู้จัดทำ นายขจรจักร สมสมัย

นายดนัย ไชยสาลี

อาจารย์ที่ปรึกษา ผศ.ทวีพล ซื่อสัตย์

ปีการศึกษา 2546

บทคัดย่อ

ในปัจจุบันเครื่องควบคุม PLC ถูกใช้งานในระบบควบคุมอัตโนมัติอย่างแพร่หลาย เนื่องจากมีความน่าเชื่อถือสูง และใช้งานได้อย่างสะดวก โดยในปริญญานิพนธ์นี้เป็นการนำเสนอการจำลองกระบวนการด้วยภาพสำหรับระบบ PLC ซึ่งสามารถใช้งานได้ในคอมพิวเตอร์ที่ทำงานภายใต้ระบบปฏิบัติการวินโดว์ 95 หรือ 98 โดยใช้ Microsoft visual basic 6.0 ในการเขียนโปรแกรม

ปริญญานิพนธ์นี้มีความสามารถในการจำลองกระบวนการในการทดลองของ PLC และเพื่อใช้เป็นเครื่องมือสนับสนุนในการศึกษาการเขียนโปรแกรมของ PLC โดยจะแสดงผลทางหน้าจอภาพของคอมพิวเตอร์เสมือนทำการควบคุมกระบวนการจริง โดยการจำลองภาพออกมาเพื่อให้เข้าใจยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Visualize Process Simulation For PLC System
Authors	Mr.Kachonchak Somsamai
	Mr.Danai Chaisalee
Thesis Advisor	Asst.Prof. Taweepol Suesut
Year	2003

ABSTRACT

Present, the Programmable Logic Controller is reliability and convenience to use, thus it is widely used in industrial such automatic control system. This project proposes the tool for developing the PLC system. By design the visualize process simulation for PLC system. This consists of hardware for interfacing between the input/output of PLC and computer via RS232 serial port, and software that developed by microsoft visual basic 6.0 for windows operating system.

This project can be supported education for PLC programming by using the computer as the visual process such cylinder control, traffic light, lift control etc. The visualize process on the computer can be controlled as well as the visualize process.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณภาควิชาวิศวกรรมการวัดคุมที่ให้การสนับสนุนเครื่องมือ และอุปกรณ์ต่างๆ ตลอดจนสถานที่ที่ใช้ทำปริญญานิพนธ์

ขอขอบพระคุณอาจารย์ที่ปรึกษา อ.ทวีพล ซื่อสัตย์ ที่ให้การสนับสนุนทางด้านวิชาการ และให้คำแนะนำปรึกษาทางด้านเทคนิคต่าง ๆ ตลอดช่วงเวลาในการทำปริญญานิพนธ์จน ปริญญานิพนธ์ได้สำเร็จลุล่วงไปได้ด้วยดี

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย....I

บทคัดย่อภาษาอังกฤษ....II

กิตติกรรมประกาศ....III

สารบัญ....IV

สารบัญตาราง....V

สารบัญรูป....VI

บทที่ 1 บทนำ....1

1.1 ความสำคัญของปริญญานิพนธ์....1

1.2 วัตถุประสงค์ของการทำปริญญานิพนธ์....1

1.3 ขอบเขตของปริญญานิพนธ์....1

1.4 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์....2

บทที่ 2 ทฤษฎีเบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์ 8051....3

2.1 กล่าวนำ....3

2.2 พอร์ตอินพุทและเอาท์พุทของไมโครคอนโทรลเลอร์ 8051....3

2.2.1 พอร์ตแบบขนานของไมโครคอนโทรลเลอร์ 8051....4

2.2.2 โครงสร้างของพอร์ตการทำงาน....4

2.2.3 การใช้งานพอร์ตเป็นอินพุต....4

2.2.4 การใช้งานพอร์ตเป็นเอาต์พุต....5

2.2.5 ลักษณะสมบัติของพอร์ตอินพุต/เอาต์พุต....6

2.3 การจัดการหน่วยความจำและการเชื่อมต่อ....7

2.3.1 หน่วยความจำโปรแกรม....7

2.3.2 หน่วยความจำข้อมูล....8

2.3.3 หน่วยความจำข้อมูลภายใน....8

2.3.4 หน่วยความจำขนาด 128 ไบต์แรก....8

2.3.5 หน่วยความจำขนาด128ไบต์ถัดไป....9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4 รีจิสเตอร์หน้าที่พิเศษ....10
2.5 การอินเตอร์รัพต์ในไมโครคอนโทรลเลอร์ 8051....13
2.5.1 ประเภทของการ interrupt....13
2.5.2 โครงสร้างการ interrupt....13
2.5.3 การควบคุม interrupt....14
2.6 วงจรนับ/จับเวลา8051....14
2.6.1 การ interupt วงจรนับตรวจ/จับเวลา....14
2.6.2 การทำงานเป็นตัวจับเวลา....14
2.6.3 การทำงานเป็นตัวนับสัญญาณ....18
2.6.4 วงจรนับ/จับเวลา2(Timer2)....18

บทที่ 3 ความรู้เบื้องต้นเกี่ยวกับพอร์ตอนุกรม....21
3.1 กล่าวนำ....21
3.2 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม....22
3.3 รูปแบบของการสื่อสารข้อมูลอนุกรม....22
3.4 การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232....23
3.5 ไอซี MAX232 ,L232....24

บทที่ 4 ทฤษฎีและหลักการ PLC....25
4.1 หน่วยประมวลผลกลาง (CPU)....25
4.2 หน่วยความจำ (Memory Unit)....26
4.2.1. RAM (Random Access Memory)....26
4.2.2. EPROM (Erasable Programmable Read Only Memory....26
4.2.3. EEPROM (Electrical Erasable Programmable Read Only Memory)....26
4.3. ส่วนที่เป็นอินพุท/เอาท์พุท (I/O Unit)....26
4.4. อุปกรณ์ติดต่อภายนอก (Peripheral Device)....27
4.5 ความแตกต่างระหว่างคอมพิวเตอร์กับ PLC....28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 5 ทฤษฎีเบื้องต้น Visual Basic Version 6.0......29

5.1 ส่วนประกอบของหน้าต่าง VB......29

5.2 รู้จักกับฟอร์ม (Form)......31

5.3 ปุ่มคอนโทรล (Control) ต่างๆ......31

5.4 การเขียนโปรแกรมเบื้องต้น......32

5.5 ตัวแปร (Variable)......33

5.6 การเขียนโปรแกรมติดต่อ และควบคุม Serial Port กับ Visual Basic......35

บทที่ 6 ผลการทดลอง......39

6.1 หลักการทำงานของวงจรสื่อสารข้อมูล......39

6.2 อุปกรณ์ที่ใช้ในการทดลอง......40

6.3 กระบวนการจำลองระบบจ่ายน้ำ......41

6.4 กระบวนการจำลองการทำงานลูกสูบ......44

6.5 การจำลองการทำงานของสัญญาณไฟจราจร......48

6.6 กระบวนการคัดแยกด้วยกล้องวิดีโอ......51

6.7 กระบวนการจำลองการผสมสารจาก ไซโล......53

6.8 รายละเอียดของวงจรที่ใช้ในการทดลอง......56

6.9 สรุปผลการทดลอง......57

6.10 ประโยชน์ที่คาดว่าจะได้รับ......57

6.11 ปัญหาและอุปสรรค......58

บรรณานุกรม......59

ภาคผนวก......60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท INTEL..........................6

2.2 แสดงรีจีสเตอร์แบงค์ของแอดเดรสใช้งาน..........................8

2.3 แสดงรีจีสเตอร์หน้าที่พิเศษ..........................10

4.1 แสดงประเภทหน้าที่ของอุปกรณ์ติดต่อภายนอก..........................27

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 แสดงโครงสร้างแต่ละบิตภายในพอร์ตอินพุต/เอาท์พุตของ8051......6
2.2 แสดงการติดต่อกับหน่วยความจำภายนอก......7
2.3 แสดงโครงสร้างการอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ 8051......13
2.4 ส่วนควบคุมของวงจรนับ/ตรวจจับเวลา......15
2.5 การทำงานของ TIMER/COUNTER 0 หรือ 1 ในโหมด 0......16
2.6 การทำงานของ TIMER/COUNTER 0 หรือ 1 ในโหมด 1......16
2.7 การทำงานของ TIMER/COUNTER 0 หรือ 1 ในโหมด 2......17
2.8 วงจรนับ/จับเวลา 0 ในโหมด 3......18
2.9 แสดงการทำงานของโหมดแคปเจอร์ของ TIMER2......19
2.10 แสดงการทำงานในโหมดกำเนิดอัตราบอดของ TIMER2......19
3.1 แสดงการส่งข้อมูลแบบขนานและแบบอนุกรม......21
3.2 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที......22
3.3 แสดงการส่งข้อมูลขนาด 8 บิตอนุกรม......23
3.4 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL......24
3.5 แสดงตำแหน่งขาของไอซี MAX232, L232 และการต่อใช้งาน......24
5.1 แสดงการส่วนประกอบของหน้าต่าง VB......29
5.2 หน้าต่างของProperties......30
5.3 หน้าต่างของForm Layout......30
5.4 หน้าต่างของลักษณะ Form......31
5.5 รายการเครื่องมือคอนโทรลต่าง ๆ......32
5.6 ปุ่มการรันโปรแกรม......32
5.7 เพิ่มคอมโพเนนต์ MSComm......36
5.8 การเลือก MSComm ออกมาใช้......36
5.9 คอนโทรล MSComm พร้อมทำงาน......37
6.1 แสดงภาพรวมของวงจรสื่อสารข้อมูล......39
6.2 กระบวนการจำลองระบบจ่ายน้ำ......41
6.3 กระบวนการจำลองการทำงานของลูกสูบ......44
6.4 การจำลองกระบวนการทำงานของสัญญาณไฟจราจร......48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ หน้า

6.5 กระบวนการคัดแยกด้วยกล้องวิดีโอ..51

6.6 กระบวนการจำลองการผสมสารจากไซโล...53

6.7 แสดงบล็อกไดอะแกรมของวงจรในส่วนที่ใช้ในการควบคุมการติดต่อสื่อสารข้อมูล..............56

6.8 แสดงส่วนของวงจรที่ใช้ในการส่งสัญญาณออกไปควบคุม PLC....................................56

6.9 แสดงส่วนของวงจรที่ใช้ในการรับสัญญาณจาก PLC..57

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันอุตสาหกรรมในประเทศไทยมีความก้าวหน้าไปมากเครื่องมือต่างๆ อาทิเช่น เซ็นเซอร์ อุปกรณ์ประเภทนิวแมติกส์และอื่นๆ ได้มีส่วนสำคัญในกระบวนการอุตสาหกรรมเพื่อที่จะให้ระบบนั้นทำงานอย่างมีประสิทธิภาพและเป็นขั้นตอน ดังนั้นในการที่จะทำให้ระบบทำงานอย่างมีประสิทธิภาพจึงจำเป็นจะต้องมีเครื่องมือที่ใช้ในการควบคุม ซึ่งเครื่องมือที่ว่านี้ก็คือ PROGRAMABLE LOGIC CONTROLLER หรือที่รู้จักกันในชื่อ PLC

PLC เป็นเครื่องมือที่มีประโยชน์ในกระบวนการอุตสาหกรรม แต่ในความเป็นจริงการที่จะศึกษาการทำงานและกระบวนการในการควบคุมให้เข้าใจถ่องแท้นั้นเป็นไปได้ยากโดยเฉพาะในระดับอุดมศึกษาและอาชีวศึกษาเนื่องจากไม่ได้เห็นอุปกรณ์และเครื่องมือต่างๆ ในกระบวนการว่าสามารถทำงานได้ตามที่ต้องการหรือไม่

การจำลองกระบวนการด้วยภาพสำหรับระบบPLC หรือ Visualize Process Simulation for PLC system ก็เพื่อที่จะใช้เป็นชุดทดลองความสัมพันธ์ระหว่าง PLC กับกระบวนการต่างๆ ในอุตสาหกรรมว่ามีความเกี่ยวข้องกันเช่นไร รวมทั้งเพื่อให้ผู้ใช้งานมีความเข้าใจในการใช้คำสั่งควบคุมการทำงานในระบบควบคุมแบบ PLC ต่อไป

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อให้ผู้ใช้งานมีความเข้าใจในการทำงานของเครื่องควบ PLC มากยิ่งขึ้น

2. สร้างชุดส่งผ่านข้อมูลระหว่าง PLC กับ แบบจำลองกระบวนการต่างๆ บนระบบคอมพิวเตอร์เพื่อใช้ในการศึกษาความสัมพันธ์ระหว่างอุปกรณ์

3.เพื่อศึกษากระบวนการในการควบคุมของ PLC รวมทั้งใช้คำสั่งต่างๆ ในการควบคุมกระบวนการ

1.3 ขอบเขตของปริญญานิพนธ์

1.สร้างชุดส่งผ่านข้อมูลระหว่างกระบวนการบนระบบคอมพิวเตอร์กับ PLC และเขียนโปรแกรมที่ใช้ในการควบคุมการส่งผ่านข้อมูลโดยใช้ไมโครคอนโทรลเลอร์ 8051 เป็นตัวควบคุมการทำงาน

2.ออกแบบโปรแกรมจำลองการกระบวนการบนเครื่องคอมพิวเตอร์โดยใช้โปรแกรมวิชวลเบสิกและแลดเดอร์ไดอะแกรมเพื่อใช้ควบคุมกระบวนการทำงานบนหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์

1. ความรู้และเพิ่มทักษะในการออกแบบวงจรรวมทั้งการเลือกใช้อุปกรณ์อิเล็กทรอนิกส์ต่างๆให้มีความเหมาะสมกับวงจร

2. เพิ่มทักษะในการเขียนโปรแกรมบนไมโครคอนโทรลเลอร์และวิชวลเบสิกมากยิ่งขึ้น

3. มีความรู้ในการใช้อุปกรณ์ติดต่อสื่อสารข้อมูลจากโครงงานคือ RS232

4. มีความรู้ความเข้าใจในระบบ PLC มากยิ่งขึ้น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์

2.1 กล่าวนำ

ไมโครคอนโทรลเลอร์8051ได้มีการผลิตครั้งแรกโดยบริษัท Intel และใช้ชื่อว่า MCS-51 ซึ่งปรากฏว่านิยมใช้กันอย่างแพร่หลายในไมโครคอนโทรลเลอร์ตระกูล 16 บิต ต่อมาได้มีหลายบริษัทที่รับลิขสิทธ์จากบริษัท Intel ให้มีการนำ MCS-51ไปผลิตเพื่อจำหน่ายทำให้เกิด 8051 ไมโครคอนโทรลเลอร์ หลายรูปแบบจำนวนมากขึ้นมาซึ่งผู้บริโภคสามารถเลือกใช้ตามความเหมาะสมของลักษณะงาน

ตารางที่ 2.1 แสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท INTEL

EMBEDDED CONTROLLERS										
Feature	8051AH	8031AH	8751H	80C51BH	80C31BH	87C51	8052AH	8032AH	8752	8044H
Program Memory (Bytes)	4K	-	4K	4K	-	4K	8K	-	8K	4K
RAM Memory (Bytes)	128	128	128	128	128	128	256	256	256	192
Program Memory Expansion (Off Chip) (Bytes)	64K									
Data Memory Expansion(Off Chip) (Bytes)	64K									
Max Clock Frequency (MHz)	12	12	12	16	16	16	16	12	12	12
Typical instruction Time (uS)	1	1	1	0.75	0.75	0.75	1	1	1	1
16-Bit Timer/Counter	2	2	2	2	2	2	3	3	3	2
Serial Communications	Synchronous Mode, Asynchronous Mode, 9 or 10 -Bit Programmable									HDLC/SDLC
No. of I/O Lines	32	16	32	32	16	32	32	16	32	32
Interrupt Sources	5	5	5	5	5	5	6	6	6	5
(Two Priority Levels) Power Requirements 125	125	250	24	24	29	175	175	175	200	
(ICC Max. mA)										
Programmable Power Modes	-	-	-	4.8 mA	4.0 mA	4.8 mA	-	-	-	-
Idle PowerDown				50 uA	51 uA	52 uA				30 mA

2.2 พอร์ตอินพุทและเอาท์พุทของไมโครคอรโทรลเลอร์ 8051

พอร์ตมีความหมายถึงแอดเดรสหนึ่งที่ได้รับการกำหนดไว้ เพื่อการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อขึ้นอยู่กับทิศทางการไหลของข้อมูลเมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก ดังนั้นการนำเข้าข้อมูลจากวงจรภายนอกจึงเรียกว่าการอินพุท และในกรณีตรงกันข้ามเพื่อส่งออกข้อมูลก็จะเรียกว่าการเอาท์พุทเมื่อพิจารณาถึงวิธีการส่งข้อมูลภายในพอร์ตจะสามารถแยกประเภทของพอร์ตออกได้เป็นสองลักษณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือพอร์ตแบบขนาน ซึ่งทำการส่งจำนวนบิตข้อมูลทั้งหมดออกมา หรือนำเข้าไปพร้อมกันในคราวเดียว และพอร์ตแบบอนุกรม ซึ่งทำการโอนย้ายข้อมูลคราวละบิต ๆ จนครบจำนวนแต่สำหรับในบทนี้จะกล่าว ถึงเฉพาะในส่วนของพอร์ตแบบขนานเท่านั้น สำหรับการทำงานของพอร์ตแบบอนุกรมจะได้กล่าวภายหลัง

2.2.1 พอร์ตแบบขนานของไมโครคอนโทรลเลอร์ 8051

8051 มีโครงสร้างของพอร์ตที่สามารถใช้งานแบบขนานได้จำนวนทั้งหมดสี่พอร์ตเรียกชื่อเรียงตามลำดับว่าพอร์ต 0, 1, 2 และ 3 และเป็นพอร์ตขนาด 8 บิตทั้งหมด การใช้งานพอร์ตสามารถทำได้ทั้งในลักษณะของเส้น สัญญาณเดี่ยวๆหรือกลุ่มของสัญญาณได้ นอกจากนี้พอร์ต 0, 2 และ 3 ยังสามารถนำไปใช้งานอื่นๆที่ไม่ใช่เป็นพอร์ตอินพุต/เอาต์พุตได้โดยพอร์ต 0 จะทำหน้าที่มัลติเพล็กซ์ ระหว่างบัสแอดเดรสไบต์ต่ำและกับบัสข้อมูลสำหรับการติดต่อ กับวงจรประกอบร่วมข้อมูลบัสแอดเดรสไบต์สูงซึ่งจะส่งออกมาทางพอร์ต 2 สำหรับพอร์ต 3 นั้น นอกเหนือไปจากความ-สามารถ เช่นพอร์ตปกติแล้วสามารถนำไปเป็นขาสัญญาณของการอินเตอร์รัปต์ต่างๆ ซึ่งรวมทั้งการสร้างสัญญาณ ควบคุม RDและ WR เพื่อทำหน้าที่อ่านหรือเขียนหน่วยความจำข้อมูลภายนอกด้วย การใช้งานพอร์ตลักษณะงานแบบ อื่นๆที่ไม่ใช่เป็นพอร์ต /เอาต์พุทนี้จะดำเนินการ โดย 8051 เองโดยอัตโนมัติ

2.2.2 โครงสร้างของพอร์ตการทำงาน

จากลักษณะโครงสร้างของแต่ละบิตภายในพอร์ตทั้งหมดของ 8051 นั้นจะเห็นว่ามีความคล้ายคลึงกันตามลักษณะโครงสร้างที่เรียกว่า Quasi-bidirectional port ยกเว้นพอร์ต 0 ซึ่งเพียงแต่ไม่มีตัวต้านทาน Pull-upสัญญาณ ไว้ภายใน วงจรประกอบอื่นภายในยังมีฟลิปฟลอปแบบ D ซึ่งมีผลทำให้พอร์ตสามารถแลตช์หรือค้างสภาวะของสัญญาณได้นอกจากนี้ ในส่วนเอาต์พุตของฟลิปฟลอปเฉพาะของพอร์ต 0 และพอร์ต 2 จะมีโครงสร้างที่ทำหน้าที่คล้ายกับสวิตช์เพิ่มเติมขึ้นเพื่อควบคุมให้เอาต์พุตนี้ต่อเข้ากับส่วนของ ทรานซิสเตอร์ในระหว่างที่ไม่ได้มีการทำงานในลักษณะของบัสแอดเดรสหรือบัสข้อมูลด้วย สำหรับบัฟเฟอร์จำนวนสองตัวของทุกบิตในพอร์ตนั้นมีการทำงานแยกกัน โดยอิสระ โดยตัวที่อยู่ทางด้านบนจะยอมให้สัญญาณผ่านได้ก็ต่อ เมื่อมีการอ่านค่าข้อมูลที่ค้างไว้ส่วนอีกตัวหนึ่งซึ่งอยู่ทางด้านล่าง จะถูกใช้งานเฉพาะเมื่อได้มีการอ่านสถานะของขาสัญญาณเท่านั้น

2.2.3 การใช้งานพอร์ตเป็นอินพุต

การใช้งานพอร์ตเป็นการอินพุตข้อมูลจะต้องเริ่มด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตนั้นก่อนเป็นลำดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในซึ่งมีผลให้บิตนั้นๆของพอร์ต 1, 2 และ 3 เป็น สภาวะของลอจิกสูง ตัวต้านทานนี้มีค่าประมาณ 50 กิโลโอห์ม ซึ่งเป็นค่าที่สูงมาก และทำให้อุปกรณ์ภายนอกสามารถขับสัญญาณของพอร์ตเหล่านี้

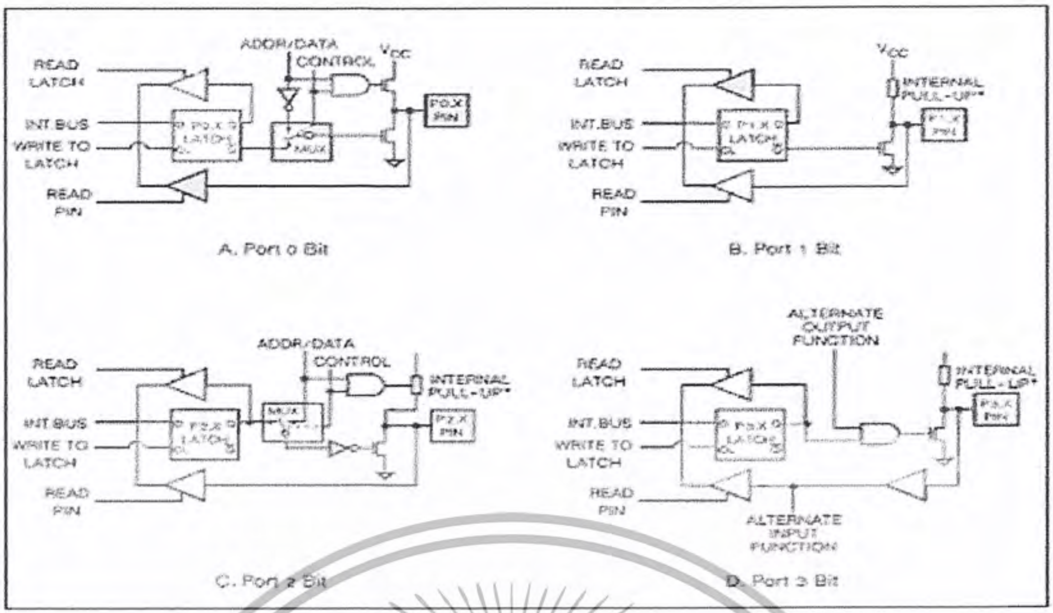
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นลอจิกต่ำได้ง่าย สำหรับบิตของพอร์ต 0 นั้น แม้ว่าจะมีหลักการทำงานที่คล้ายคลึงกัน กับบิตของพอร์ตอื่นๆ แต่เนื่องจากการที่ไม่มีตัวต้านทานทำหน้าที่ Pull-up ภายในไว้ ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่ ขับสัญญาณเอาต์พุตนั้นหยุดการทำงาน ก็จะเป็นผลให้ขาสัญญาณนี้อยู่ในสภาวะอิมพีแดนส์สูงแทน

2.2.4. การใช้งานพอร์ตเป็นเอาต์พุต

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับฟลิปฟลอปซึ่งจะค้างค่านี้ไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสภาวะลอจิกเป็นลอจิกต่ำส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้นในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ต 1, 2 หรือ 3 จะทำ ให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตนั้นหยุดการทำงานมีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัวต้านทานที่ Pull-up อยู่ภายในนั้น แต่สำหรับการทำงานในแต่ละบิตทางพอร์ต 0 นั้นจะมีผลที่แตกต่างออกไปโดยขาสัญญาณจะเป็นสภาวะอิมพีแดนซ์สูงแทนเนื่องจากไม่มีตัวต้านทานภายในเชื่อมต่ออยู่นั่นเอง ดังนั้นในการใช้งานพอร์ต 0 เป็นการเอาต์พุตข้อมูลจึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับลอจิกสูงแทนความสามารถอีกประการหนึ่งเกี่ยวกับพอร์ตอินพุต /เอาต์พุตของ 8051 เป็นวิธีการอ่านลิจิกจากพอร์ตซึ่งมีได้สองวิธีคือ การอ่านค่าลอจิกที่ขาสัญญาณ (Port pin) และวิธีการอ่านลอจิกของการแลตซ์ที่พอร์ต (Port latch) ดังจะสังเกตได้จาก วิธีการอ่านค่าจากพอร์ต ทั้งสองแบบนี้จะช่วยให้ระบบทำงานได้ด้วยความถูกต้องมาก ยิ่งขึ้น ยกตัวอย่างเช่น หากว่าพอร์ตถูกนำไปต่อกับขาเบสของทรานซิสเตอร์แบบ NPN และขาอิมิตเตอร์ต่อกับกราวด์ ของระบบ เมื่อมีการส่งค่า 1 ออกไปจะมีผลทำให้ทรานซิสเตอร์ทำงาน ในขณะนั้นถ้าซีพียูมีการอ่านค่าลิจิกจากขาสัญญาณของพอร์ตนี้ก็จะได้ค่าลอจิกต่ำเนื่องจากมองเห็นค่าศักย์ไฟฟ้าระหว่างขาเบสและขาอิมิตเตอร์ซึ่งมีค่าประมาณ 0.6 โวลต์แทนดังนั้นในกรณีเช่นนี้ หากว่าเป็นการอ่านค่าจากลอจิกของการแลตซ์ ก็จะได้รับค่าระดับลอจิกสูง ซึ่งเป็นค่า ที่ถูกต้องสภาพที่เป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 แสดงโครงสร้างแต่ละบิตภายในพอร์ตอินพุต/เอาท์พุตของ8051

2.2.5 ลักษณะสมบัติของพอร์ตอินพุต/เอาต์พุต

ดังได้กล่าวแล้วว่าพอร์ต 1,2 และ 3 ของ 8051 มีตัวต้านทาน (ซึ่งสร้างขึ้นจาก FET) ทำหน้าที่ Pull-up ขาสัญญาณไว้และมีค่าประมาณ 50 K โอห์ม ซึ่งถือว่ามีค่าที่สูงมาก เป็นผลให้การเปลี่ยนแปลงระดับสัญญาณลอจิก จากสูงไปต่ำทำได้อย่างรวดเร็ว แต่ในกรณีตรงข้ามจะใช้เวลาการเปลี่ยนแปลงระดับสัญญาณนานกว่ามาก ทั้งนี้เนื่องจากว่ากระแสจะไหลผ่านตัวต้านทานนี้ได้น้อยมาก ดังนั้นในการแก้ปัญหาจึงได้มีการออกแบบตัวต้านทานเพิ่มขึ้นอีกหนึ่ง ตัวขนานไว้โดยมีค่าประมาณ 1K โอห์ม เรียกว่า Speed-up resistor ซึ่งยอมให้กระแสไหลผ่านได้มากขึ้นประมาณ 50-100 เท่าและจะมีการเชื่อมต่อตัวต้านทานที่เพิ่มขึ้นนี้เฉพาะเมื่อมีการเปลี่ยนระดับสัญญาณจากลอจิกต่ำไปเป็นลอจิก สูงเท่านั้น โดยใช้เวลาประมาณ 2 คล็อกไซเคิล 4.6 คำสั่งการใช้งานพอร์ตอินพุต / เอาต์พุต เนื่องจาก 8051 ใช้หลักการที่เรียกว่า Memory mapped system กล่าวคือการอ้างถึงพอร์ตรีจิสเตอร์ หรืออุปกรณ์ต่างๆ ภายในระบบ จะเป็นการติดต่อกับหน่วยความจำตำแหน่งหนึ่งเท่านั้น ดังนั้นในการดำเนินการเพื่อนำ เข้าหรือส่งออกข้อมูลกับพอร์ต จึงใช้คำสั่งการอ่านค่าจากหน่วยความจำซึ่งถูกออกแบบให้เป็นตำแหน่งของพอร์ตหรือ คำสั่งการเขียนค่าข้อมูลไปยังตำแหน่งหน่วยความจำนั้นแทน ดังนั้นจะสังเกตเห็นได้ว่าในตารางชุดคำสั่งของ 8051 จะไม่มีคำสั่งที่เกี่ยวกับการทำงานพอร์ตแต่ประการใด เช่น คำสั่ง IN (นำเข้าข้อมูลจากพอร์ต)หรือคำสั่ง OUT(ส่งข้อมูลออกจากพอร์ต) เป็นต้น นอกจากนี้ 8051 ยังมีชุดคำสั่งที่จัดการข้อมูลแบบบิตได้โดยตรง ดังนั้นเรา สามารถที่จะใช้คำสั่งนี้จัดการพอร์ตอินพุต/เอาต์พุตทั้งหมดแบบเส้นสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดี่ยวได้โดยการใช้คำสั่ง SETB เพื่อ กำหนดค่าเป็น 1 หรือคำสั่ง CLR เพื่อทำให้บิตมีค่าเป็น 0 คำสั่งเหล่านี้มีประโยชน์มากและทำให้ลดความซับซ้อนในการใช้คำสั่งภายในโปรแกรมลงได้มาก

2.3 การจัดการหน่วยความจำและการเชื่อมต่อ

หน่วยความจำของไมโครคอนโทรเลอร์ 8051 แบ่งเป็น 2 ประเภท คือ

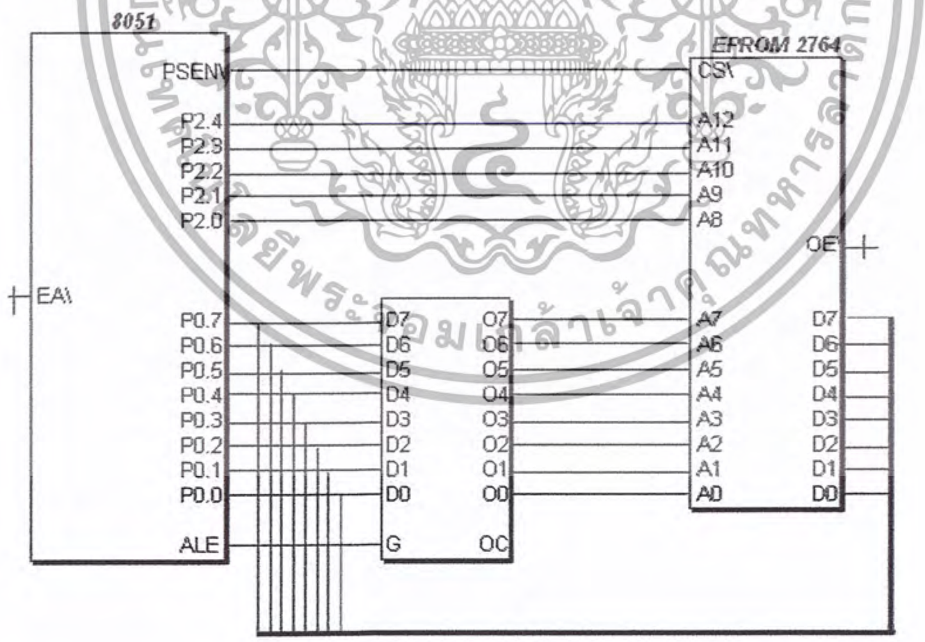
1. หน่วยความจำโปรแกรม(Program Memory) หน่วยความจำประเภทนี้ คือ ROM ใช้เก็บโปรแกรมที่ใช้ในการควบคุมระบบ ซึ่งเป็นหน่วยความจำประเภท non-volatile

2. หน่วยความจำข้อมูล (Data Memory) หน่วยความจำประเภทนี้ได้แก่ RAM

2.3.1 หน่วยความจำโปรแกรม

ใน8051 จะแบ่งหน่วยความจำประเภทนี้เป็นอีก 2 ประเภทหน่วยความจำภายนอกคือ ROM ที่มาต่อภายนอกตัว 8051 ส่วนอีกประเภทหนึ่ง คือ หน่วยความจำภายใน ได้แก่ ROM ภายในตัวไมโครคอนโทรเลอร์เองการเลือกใช้หน่วยความจำโปรแกรมภายนอกหรือหน่วยความจำโปรแกรมโปรแกรมภายในทำได้โดยการให้สัญญาณทางไฟฟ้าที่ขาEA/ โดย

- สัญญาณทางไฟฟ้าที่ขา EA/ เป็นลอจิก 0 หมายถึง หน่วยความจำโปรแกรมภายนอก
- สัญญาณทางไฟฟ้าที่ขา EA/ เป็นลอจิก 1 หมายถึง หน่วยความจำโปรแกรมภายใน



ภาพที่ 2.2 แสดงการติดต่อกับหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 หน่วยความจำข้อมูล

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูล หรือตัวแปรที่เกิดขึ้น ในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราวโดยพื้นฐานแล้วหน่วยความจำข้อมูลจัดเป็นหน่วยความจำ RAM แบบสแตติกดังนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ภายในหน่วยความจำนี้สูญหายไปพื้นที่ของหน่วยความจำข้อมูลของ 8051 สามารถมีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น

- หน่วยความจำโปรแกรมภายใน ซึ่งเป็น RAM ที่อยู่ภายในตัวของ ไอซีไมโคร-คอนโทรลเลอร์เอง

- หน่วยความจำข้อมูลภายนอกซึ่งเป็น การใช้ไอซีหน่วย ความจำ RAM มาเพิ่มเติมเข้าไปในวงจร

ลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

2.3.3 หน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายในของ 8051 มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกออกได้เป็นสองลักษณะ คือพื้นที่เฉพาะสำหรับตัวประมวลผลกลาง หรือเรียกว่ารีจิสเตอร์ R0-R7 และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา

2.3.4 หน่วยความจำขนาด 128 ไบต์แรก

บริเวณแอดเดรส 00H - 1FH จำนวน 32 ไบต์จำแนกออกเป็นกลุ่ม (Blank) 8 ไบต์ จำนวน 4 กลุ่มซึ่งมีชื่อเรียกว่ารีจิสเตอร์ R0-R7 ดังนี้

ตารางที่2.2 แสดงรีจิสเตอร์แบงค์ของแอดเดรสใช้งาน

แอดเดรส	รีจิสเตอร์แบงค์	ชื่อรีจิสเตอร์ใช้งาน
00H-07H	0	R0-R7
08H-0FH	1	R0-R7
10H-17H	2	R0-R7
18H-1FH	3	R0-R7

จะเป็นได้ว่าชื่อของรีจิสเตอร์ไม่ว่าจะอยู่ในรีจิสเตอร์แบงค์ใดก็จะมีชื่อ R0 ถึง R7เหมือนกันทั้งสิ้น ดังนั้นในการใช้งานผู้ใช้จะต้องให้ความระมัดระวังว่าต้องการรีจิสเตอร์นั้นๆ จากแบงค์ใด การสวิตซ์ เลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในรีจิสเตอร์ PSW เท่านั้นอย่างไรก็ตาม โดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0-R7 เฉพาะในแบงค์ 0 เท่านั้น ดังนั้นพื้นที่ของ แบงค์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วยความจำข้อมูลภายในปกติด้วยการอ้างถึงหมายเลขของ แอดเดรสนั้นๆ โดยตรง บริเวณแอดเดรส 20H -

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2FH จำนวน 16 ไบต์ บริเวณพื้นที่เป็นส่วนสำหรับผู้ใช้ซึ่งจะมีความพิเศษต่างไปหน่วยความจำส่วนอื่นๆ เนื่องจากผู้ใช้อาจสามารถอ้างถึงหน่วยความจำบริเวณนี้ได้ทั้งในลักษณะของ ไบต์ข้อมูล เช่น ปกติ หรืออาจจะเป็น บิตข้อมูล ได้โดยตรง ดังนั้นหากว่ามองในลักษณะบิตข้อมูลแล้ว ก็จะมีพื้นที่ตัวแปรแบบบิตให้ใช้ งานได้มากถึง128 บิต โดยตำแหน่งแรกของบิตจะเป็นบิตซึ่งเริ่มต้นนับจากบิตนัยสำคัญต่ำสุด (LSB) ของแอดเดรส 20H เรื่อยไปจนกระทั่งถึงบิตที่ 127 ซึ่งเป็นบิตนัยสำคัญสูงสุด (MSB) ของแอดเดรส 2FH เนื่องจากว่า 8051 ได้รับการออกแบบมาก็มักจะเป็นเพียงการอ่านค่าสถานะลอจิก ของเส้นสัญญาณ หรือกรณีการส่งออกข้อมูลก็จะเป็นการกำหนดสภาวะลอจิกให้กับวงจรภายนอกผ่านทางบิตใดบิต หนึ่งอยู่แล้ว ดังนั้นหากว่ามีการกำหนดบิตหรืออ่านค่าของบิตมาโดยตรง แทนที่จะต้องทำลอจิกขั้นต้นกับข้อมูลทั้งไบต์ เพื่อต้องการทราบผลเพียงหนึ่งบิตเช่นที่กระทำกันใน โปรเซสเซอร์โดยทั่วไป ก็จะเพิ่มความสะดวกและรวดเร็วในการเขียนโปรแกรมบริเวณแอดเดรส 30H - 7FH เป็นบริเวณที่สามารถนำไปใช้งานได้อย่างอิสระโดยสามารถอ้างถึงได้เฉพาะในลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

2.3.5 หน่วยความจำขนาด 128 ไบต์ถัดไป

พื้นที่ตั้งแต่บริเวณตั้งแต่แอดเดรส 80H-FFH เป็นบริเวณของหน่วยความจำที่มีการ ใช้งานเฉพาะจาก 8051 เท่านั้น โดยจะนำมาใช้เป็นตำแหน่งของ รีจิสเตอร์หน้าที่พิเศษจำนวน 20 ตำแหน่ง สำหรับไมโครคอนโทรลเลอร์เบอร์ 8051 จะมีหน่วยความจำข้อมูลภายในสำหรับการใช้งานเพิ่มมากขึ้น กว่าเบอร์อื่นๆ เช่น 8031 หรือ 8751 อีก 128 ไบต์ โดยจะอยู่บริเวณช่วงแอดเดรส 80H ถึง FFH เช่นกัน ซึ่งแม้ว่า จะเป็นพื้นที่ที่มีหมายเลขแอดเดรสเดียวกับส่วนของรีจิสเตอร์หน้าที่พิเศษ แต่ในความเป็นจริงแล้วจะเป็นพื้นที่หน่วย ความจำอีกบริเวณหนึ่ง ซึ่งมีการซ้อนเกย กันให้อยู่ในบริเวณแอดเดรสส่วนนี้ ซึ่งหากว่าผู้ใช้งานต้องการ จะเก็บข้อมูลในพื้นที่บริเวณนี้ก็จะต้องใช้การอ้างถึงหน่วยความจำแบบโดยอ้อม เท่านั้น

พระจอมเกล้าเจ้าคุณทหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่2.3 แสดงรีจีสเตอร์หน้าที่พิเศษ

ชื่อรีจิสเตอร์	ความสามารถในการอ้างถึงแบบบิตได้
ACC	*
B	*
PSW	*
SP	*
DPTR	*
P0	*
P1	*
P2	*
P3	*
IP	*
IE	
TMOD	
TCON	
TH0	
TL0	
TH1	
TL1	
SCON	
SUBF	* แทนได้
PCON	

2.4 รีจิสเตอร์หน้าที่พิเศษ

รีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่และการทำงานของอุปกรณ์หรือพอร์ตของ 8051 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อของรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของ รีจิสเตอร์นั้นก็ได้ตารางต่อไปนี้ แสดงเห็นลักษณะการจัดพื้นที่หน่วยความจำสำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อ สังเกตว่า รีจิสเตอร์ที่อยู่ตำแหน่งแอดเดรสที่เป็นจำนวนทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 แอคคิวมูเลเตอร์ (Accumulator) หรือ ACC

เป็นรีจิสเตอร์ขนาด 8 บิตทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในซีพียู และเก็บผลลัพธ์ ที่ได้จากการทำงานนั้น การทำงานของรีจิสเตอร์นี้มีลักษณะเช่นเดียวกับตัวแอกคิว-มูเลเตอร์ของโปรเซสเซอร์ทั่วไปการใช้งานภายในโปรแกรมจะเรียกว่ารีจิสเตอร์ A

2.4.2 รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณแลหารตัวเลขในกรณีที่ไม่ใช้ในการคำนวณทางด้านคณิตศาสตร์ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

2.4.3 โปรแกรมเคาน์เตอร์ (Program Counter)

เป็นรีจิสเตอร์ที่ใช้ในการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรมซึ่งจะต้องไปทำงานในลำดับถัดไปการใช้งานภายในโปรแกรมจะเรียกว่ารีจิสเตอร์PC

2.4.4 สแต็กพอยน์เตอร์ (StackPointer)

เป็นรีจิสเตอร์ขนาด 8 บิตทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ (Pointer) ของบริเวณ สแต็ก(Stack) สำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจาก โปรแกรม โดยปกติแล้วเมื่อทำการเริ่มต้น ระบบใหม่ภายหลังจากการเริ่มจ่ายไฟฟ้า หรือมีการรีเซต (Reset) เกิดขึ้นค่าภายในสแต็กพอยน์เตอร์จะมีค่า 07H ซึ่ง เป็นตำแหน่งแอดเดรสภายในบริเวณเนื้อที่ 128 ไบต์ แรกของหน่วยความจำข้อมูลภายใน การใช้งานภายในโปรแกรมจะเรียกว่า รีจิสเตอร์ SP

2.4.5 ตัวชี้ข้อมูล หรือ ดาต้าพอยน์เตอร์ (Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิตสองตัวคือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าของแอดเดรสของหน่วยความจำที่ต้องใช้งานภายในโปรแกรมหรืออาจจะเป็นแอดเดรสของอุปกรณ์ภายนอก ซึ่งกำหนดให้ติดต่อกัน โดยใช้ตำแหน่งของหน่วยความจำภายในโปรแกรม

2.4.6 โปรแกรมสเตตัสเวิร์ด (PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์(Bank)ของรีจิสเตอร์ที่ใช้งานด้วย

2.4.7 รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต / เอาท์พุต โดยตรง ซึ่งแต่ละตัวจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะของการอินพุต หรือการเอาต์พุตข้อมูลได้ การดำเนินการใดๆ ที่ เกี่ยวกับพอร์ตทั้งสี่นี้จะมีผลทำให้ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนแปลงไปเช่นกัน นอกจากนี้พอร์ต P0 และ P2 ยังสามารถนำมาใช้ในการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเดรส 8 บิตบนของหน่วยความจำ ส่วนพอร์ต P0 นั้นในช่วงเริ่มแรกจะเป็นค่าของแอดเดรส 8 บิตล่างของหน่วยความจำช่วงเวลาต่อมาจึงจะนำพอร์ต P0 ไปใช้เป็นบัสสำหรับการรับหรือ ส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ หน่วยความจำอุปกรณ์ภายนอก สำหรับพอร์ต P3 นั้นนอกเหนือจากจะใช้ในฐานะของพอร์ตอินพุต /เอาต์พุตเช่นปกติแล้วยังนำมาใช้ในฐานะบัสควบคุมเกี่ยวกับสัญญาณอินเตอร์รัปต์ได้อีกด้วย

2.4.8 รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็น จริงแล้วบัฟเฟอร์นี้มีอยู่ด้วยกันสองชุด และแยกจากกันอย่างชัดเจน สำหรับการส่งและการรับ โดยซีพียูจะทำการจัดการเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

2.4.9 รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมหน้าที่การทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรโซสเซอร์ (บิต IDL และ PD) การกำหนดอัตราการทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรม(บิต SMOD) และแฟล็กสภาวะสำหรับการใช้งานทั่วไป (บิต GR0 และ GR1)

2.4.10 บิต PD (Power down)

เป็นการกำหนดให้ลดกำลังไฟฟ้าที่จ่ายให้กับส่วนของโปรเซสเซอร์ภายในลงโดยยังคงมีกำลังไฟฟ้าจ่ายให้กับส่วนหน่วยความจำข้อมูลภายในผ่านทางขาสัญญาณ RST วิธีการนี้มักนำมาใช้ในกรณีที่มีการตรวจสอบการไม่มีกำลังไฟฟ้า (Power failure) โดยวงจรตรวจสอบภายนอกจะต้องมีการอินเตอร์รัปต์เข้ามา เพื่อทำการเก็บข้อมูลที่กำลังประมวลผลอยู่ก่อนและเมื่อมีกระแสไฟฟ้าจ่ายให้เป็นปกติแล้วจึงค่อยนำข้อมูลนั้นมาประมวลผลต่อไป

2.4.11 บิต IDL (Idle Mode)

เป็นการกำหนดให้โปรเซสเซอร์หยุดการทำงานชั่วขณะ (Sleep)และจะกลับมาอยู่ในสภาพปกติอีกครั้งเมื่อ ทำการรีเซตทางฮาร์ดแวร์ หรือมีการอินเตอร์รัปต์อย่างใดอย่างหนึ่งเกิดขึ้นการทำงานในลักษณะนี้สามารถเกิดขึ้นได้ ก็เนื่องจากว่าสภาวะการหยุดการทำงานชั่วขณะนั้น เป็นเพียงการห้ามไม่ให้มีสัญญาณนาฬิกาจ่ายให้ส่วนของ โปรเซสเซอร์ ส่วนของวงจรการอินเตอร์รัปต์พอร์ตอนุกรมและวงจรนับ/จับเวลา ยังคงมีสัญญาณนาฬิกาอยู่เป็นปกติ

2.4.12 รีจิสเตอร์ IP,IE,TMOD,TMOD,SCON

เป็นกลุ่มของรีจิสเตอร์ที่ทำหน้าที่กำหนดการควบคุม และการทำงานของการอินเตอร์รัปต์ต่าง ๆ ของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การอินเตอร์รัพต์ ในไมโครคอนโทรลเลอร์ 8051

2.5.1 ประเภทของการ interrupt

- External interrupt การตรวจสอบสัญญาณที่มา interrupt นี้จะสามารถกำหนดให้มีการตรวจสอบในลักษณะเมื่อได้มีการเปลี่ยนแปลงระดับสัญญาณ (Level-sensitive) ไปแล้ว หรือในช่วงเวลาขณะเริ่มมี การเปลี่ยนแปลงสัญญาณจาก logic สูงไปต่ำ (Edge-sensitive)

- Internal inetrrupt แหล่งกำเนิดสัญญาณนี้จะเป็นวงจรภายใน Microcontroller เอง เช่น วงจรนับ/จับเวลา วงจรเชื่อต่อสัญญาณอนุกรม เป็นต้น

2.5.2 โครงสร้างการ interrupt

เกิดได้ 5 ลักษณะ คือ

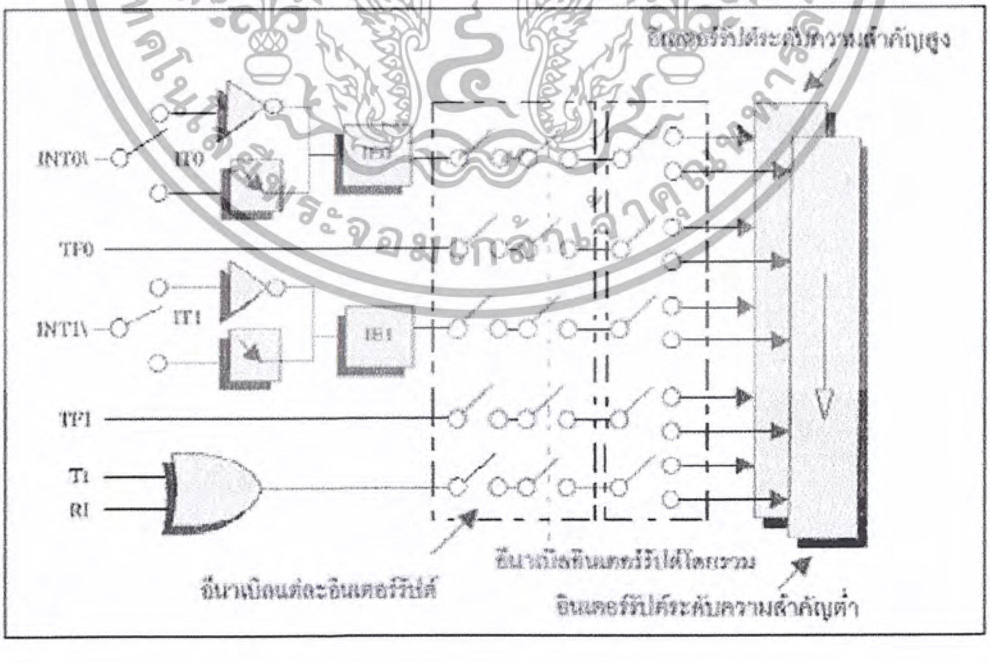
- INT0สัญญาณ interruptจากภายนอกทางขาสัญญาณP3.2 โดย8051จะทำการสุ่ม ตัวอย่างสัญญาณเมื่อสิ้นสุดทุก Machine Cycle

- INT1 สัญญาณ interrupt จากภายนอกทางขาสัญญาณ P3.3 โดย 8051 จะทำการสุ่มตัวอย่าง สัญญาณเมื่อสิ้นสุดทุก Machine Cycle

- Timer0 สัญญาณการเกิด Overflow ของ Timer 0

- Timer1 สัญญาณการเกิด Overflow ของ Timer 1

- Serial Port การเกิด interrupt ที่เกิดขึ้นจากการรับ / ส่งข้อมูลอนุกรม ทำให้มีผลต่อ flag interrupt RI และ TI ตามลำดับ



ภาพที่ 2.3 แสดงโครงสร้างการอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภาพโครงสร้างระบบ interrupt ของ 8051 จะเห็นว่าเมื่อเกิดการ interrupt สัญญาณต่างๆขึ้น จะส่งผลให้มีการควบคุมเพื่อสั่งให้ processor กระโดดไปทำงานที่ตำแหน่ง Address ต่างๆ ตามประเภทของแหล่งกำเนิดสัญญาณ interrupt ที่เกิดขึ้น ซึ่งปรกติควรมีการสร้าง program เหล่านี้ไว้ เพื่อทำหน้าที่ย่อยบริการ interrupt การกำหนดให้ 8051 สามารถตอบรับการ interrupt แต่ละประเภท ทำได้โดยกำหนด bit ของ ข้อมูลที่เกี่ยวข้อง ซึ่งมักอยู่ภายใน register TCON และ SCON หากได้ว่ามีการกำหนด ค่าของ bit ซึ่งอยู่ภายใน register IE (Intereupt Enable Register) ด้วยแล้ว ก็สามารถตอบรับการ interrupt ของสัญญาณนั้นๆ ได้ นอกจากนี้ตามแผนภาพที่2.3 ยังแสดงให้เห็นว่าสัญญาณ interrupt แต่ละ ประเภท ยังสามารถกำหนด piority ของการ interrupt ได้ 2 ลักษณะ คือ High Low piority กล่าว คือขณะที่ประมวลผลอยู่ภายในส่วนของ program ย่อย บริเวณinterrupt ของสัญญาณที่มีระดับความ สำคัญต่ำอยู่ ก็สามารถถูก interrupt ที่มี piorrity สูงกว่าได้ แต่หากว่าเป็นสัญญาณ interrupt ที่มี piority เดียวกันหรือต่ำกว่าแล้วก็จะต้องรอให้เสร็จสิ้นการประมวลผลที่ดำเนินอยู่ก่อน

2.5.3 การควบคุม interrupt

ตามโครงสร้างที่ด้านการจัดการ interrupt ของ 8051 สามารถกำหนดเรียกเพื่อยินยอม หรือไม่ยินยอม (Enable/Disabble) ให้มีการ interrupt แต่ละสัญญาณได้ โดยใช้วิธีการกำหนด ค่าของ bit ภาย ใน register IE

2.6 วงจรนับ/จับเวลา 8051

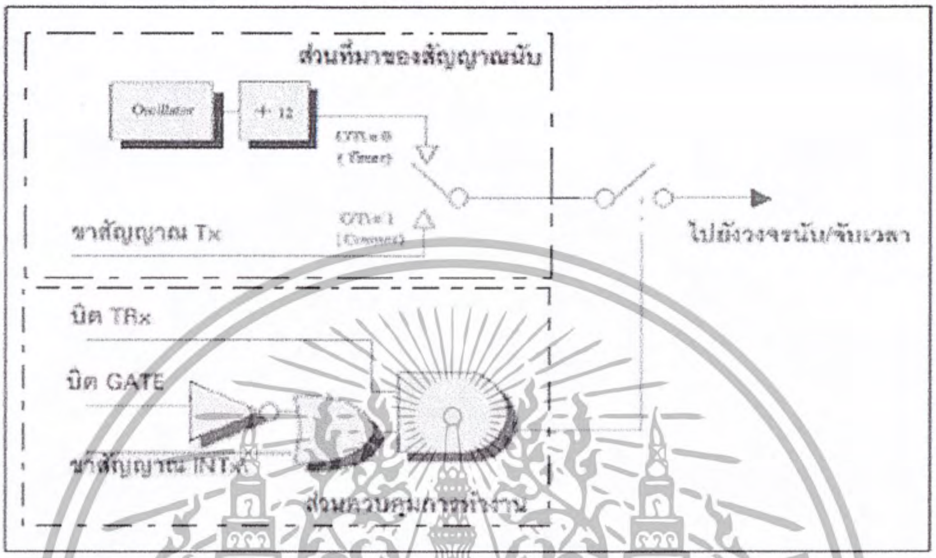
ประกอบด้วย register ขนาด 16 bit จำนวน 2 ตัว คือ T0 (Timer0) และ T1 (Timer1) ซึ่งสามารถนำไปใช้งานได้อย่างอิสระ โดยสามารถควบคุมให้ทำหน้าที่เป็นตัวจับเวลา (Timer) เพื่อนับจำนวน plus สัญญาณนาฬิกาภายใน หรือควบคุมให้ทำหน้าที่เป็นตัวนับ(Counter) เพื่อนับจำนวน plus ของระบบ ได้ ภายใน register แต่ละตัวยังสามารถแยกออก ได้เป็น register ขนาด 8 bit คือ TH0, TL0, TH1 และ TL1 โดยการทำงานของ register ทั้ง 2 ตัวนี้มีผลมาจากการกำหนดค่าของ bit ที่อยู่ภายใน TMOD (Timer mode control register) และ TCON (Timer/Counter control register) bit ต่างๆภายใน register TMOD bit ต่างๆภายใน register TCON

2.6.1 การ interupt วงจรนับตรวจ/จับเวลา

จากกระบวนการทำงานของวงจรนับ/จับเวลาของ 8051 จำเป็นต้องกำหนดค่าเริ่มต้นให้กับ register T0 หรือ T1 ค่านี้เป็นค่าจำนวน plus ภายในที่จะต้องนับหรือค่าของจำนวน plus ภายนอกที่เข้ามาทางขาสัญญาณสัญญาณ T0 หรือ T1 ค่าตัวเลขภายใน register นี้จะต้องลดให้มีค่าที่น้อยกว่าค่าที่ต้องการอยู่หนึ่งค่า ทั้งนี้เนื่องจากการทำงานของ register จะเพิ่มค่าจากที่กำหนดไปเรื่อยๆ จนถึง ค่าสูงสุดของ register และกลับไปเป็นค่า 0 เมื่อมีการเกิด Overflow เกิดขึ้น ทำให้เกิดการกำหนดค่า flag เพื่อแจ้งให้ CPU ได้รับทราบ ดังนั้นโปรแกรมทั่วไปจึงมักใช้สภาวะของ flag นี้(TF0 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TF1) ซึ่งเป็น bit อยู่ภายใน register TCON เพื่อตรวจสอบว่ากระบวนการนั้นได้เสร็จสิ้นลงแล้ว หรือใช้เพื่อ ทำการ interrupt program ต่อไป ส่วนควบคุมการทำงานของวงจรนับ/จับเวลา ซึ่งประกอบด้วยส่วนของการกำหนดที่มาของสัญญาณ (Timer) หรือ (Counter) และ bit หรือขาสัญญาณสำหรับการหยุดหรือทำงานของวงจรนับ

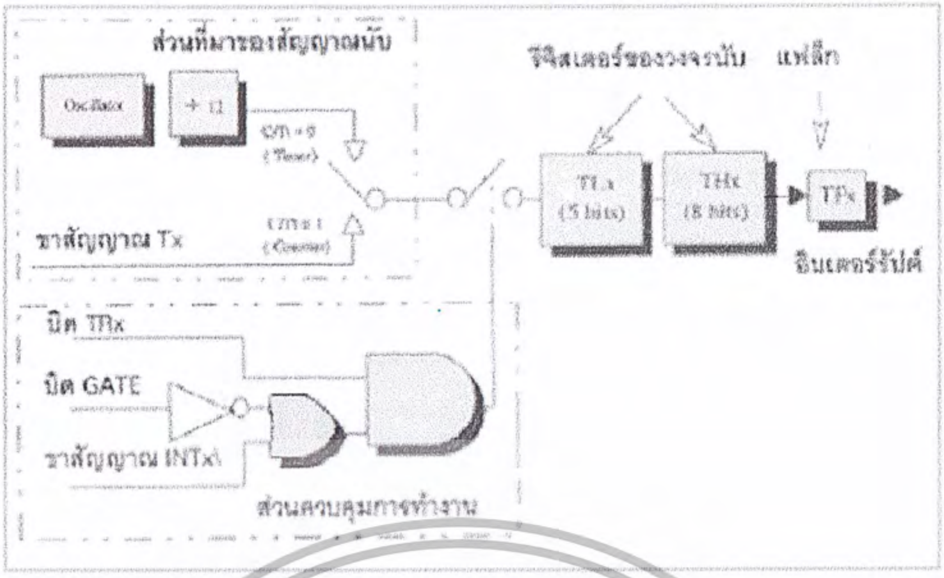


ภาพที่ 2.4 ส่วนควบคุมของวงจรนับ/ตรวจจับเวลา

2.6.2 การทำงานเป็นตัวจับเวลา

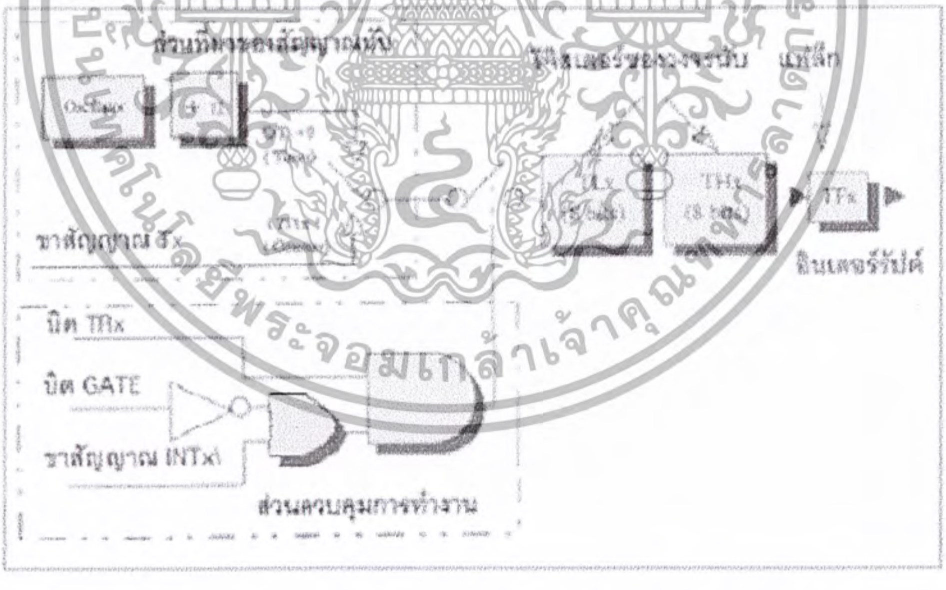
- การจับเวลาใน mode 0 : การทำงานใน mode 0 วงจรนับจับเวลาจะทำหน้าที่เป็นตัวนับขนาด 13 bit (โดยใช้ register TH0 หรือ TH1 เป็นตัวนับขนาด 8 bit และ register TL0 หรือ TL1 มีขนาด 5 bit) ตามลักษณะของ แผนภาพในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.5 การทำงานของ TIMER/COUNTER 0 หรือ 1 ในโหมด 0

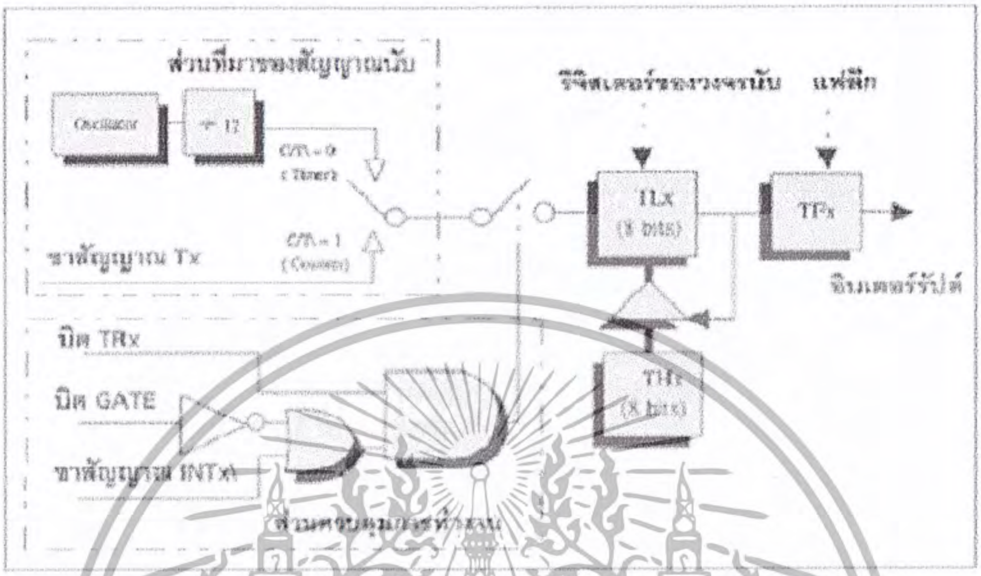
- การจับเวลาใน mode 1 : การทำงานใน mode 1 มีความคล้ายคลึงใน mode 0 มาก แตกต่างกันที่ mode 1 เป็นตัวนับขนาด 16 bit เต็ม ดังรูป



ภาพที่ 2.6 การทำงานของ TIMER/COUNTER 0 หรือ 1 ในโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

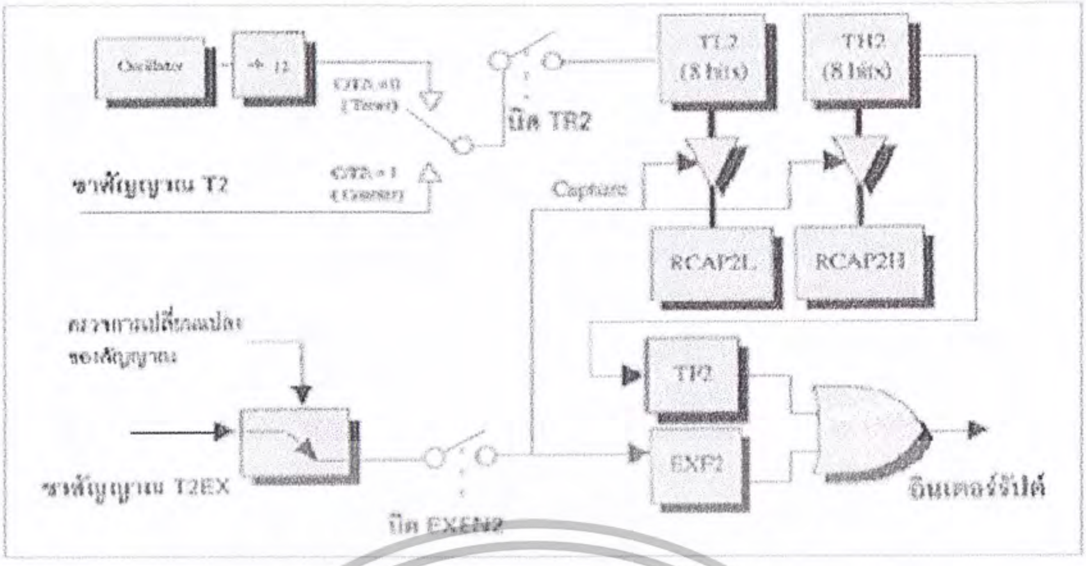
- การจับเวลาใน mode 2 : การทำงานใน mode 2 ของวงจรนับ/จับเวลาแตกต่างกันออกไป เพียงใช้ register TL0(TL1) เป็น ตัวนับขนาด 8 bit ส่วน register TH0 (TH1) เก็บค่าเริ่มต้นของการนับไว้ดังรูป



ภาพที่ 2.7 การทำงานของ TIMER/COUNTER 0 หรือ 1 ในโหมด 2

- การจับเวลาใน mode 3 : การทำงานใน mode 3 นี้จะสามารถใช้ได้เฉพาะกับ Timer 0 เท่านั้น หากว่านำไปกำหนดให้กับ Timer 1 จะทำให้หยุดการทำงานไป เมื่อ Timer 0 ได้รับการกำหนดทำงานใน mode 3 จะมีผลทำให้ register ของมันแยกการทำงานเป็นอิสระ โดย register TL0 จะถูกควบคุมจาก bit ภายใน register TCON และ ขาสัญญาณ INT0 ดังแสดงในรูป และเมื่อมีการ Overflow เกิดขึ้น จากค่า 0FFH เป็น 00H ก็ จะมีผลให้ flag TF0 มีการเปลี่ยนแปลงเกิดขึ้น สำหรับ register TH0 จะถูกกำหนดให้ทำงานในแบบ ของตัวจับเวลาภายใต้การควบคุมของ bit TR1ใน registerTCON เท่านั้น และหากเกิด Overflow จะมีผลเฉพาะต่อ flag TF1 ในส่วน Timer 1 ขณะเมื่อ Timer 0 ถูกกำหนดให้ทำงานmode3 ก็ยังจะสามารถทำงานใน mode อื่นๆ ที่ไม่ใช้ mode 3 ได้ เช่นเดิม ยกเว้นจะไม่มีการ interrupt เกิดขึ้นเท่านั้น (เนื่องจากflag TF1 ถูก ใช้โดย Timer 0 ไปแล้ว) รูปการทำงานใน mode 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.8 วงจรนับ/จับเวลา 0 ในโหมด 3

2.6.3 การทำงานเป็นตัวนับสัญญาณ

การใช้งานในลักษณะตัวนับ(counter) โดยหลักแล้วจะเหมือนกับลักษณะการทำงานเป็นตัวจับเวลา (Timer) ดังได้กล่าวในหัวข้อที่ผ่านมา ข้อแตกต่างประการเดียวคือ แทนที่จะนับ plus สัญญาณภายในและผ่านวงจรหาร 12 มาเป็นการนับ plus สัญญาณทางขาสัญญาณ T0(P3.4) ให้ กับ Timer0 หรือขาสัญญาณ T1(P3.5) ให้กับ Timer1 เท่านั้น นอกจากนี้ก่อนการเริ่มต้นใช้งานจะ ต้องกำหนดค่าของ bit C/T ภายใน register TCON ให้มีค่าเป็น 1 เสียก่อน

2.6.4 วงจรนับ/จับเวลา2(Timer2)

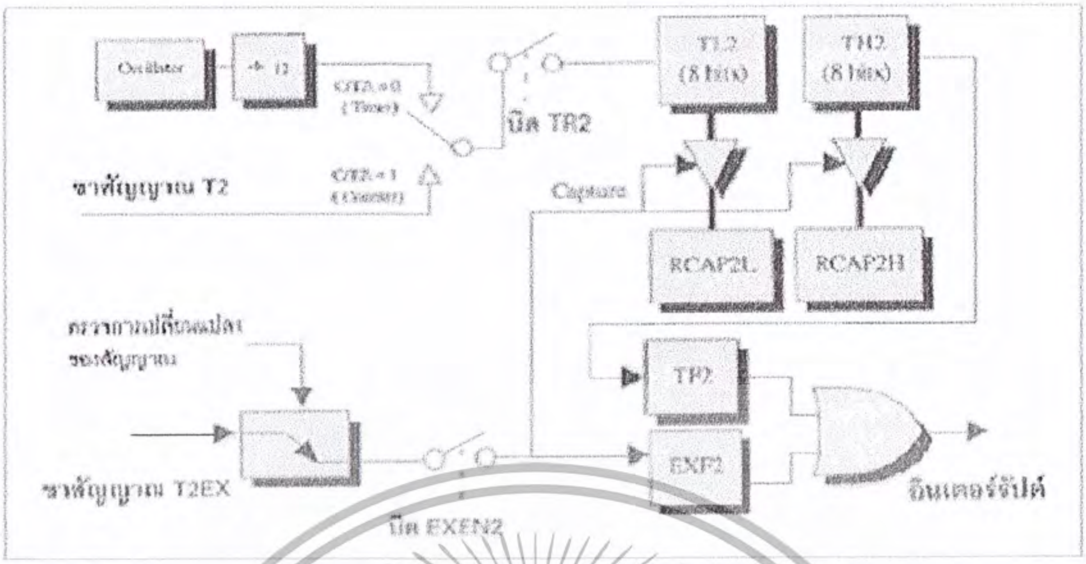
ซึ่ง mode การทำงานของ Timer2 ประกอบไปด้วย

- Capture mode: สามารถเลือกใช้งานได้ 2 ลักษณะ ด้วยการกำหนดให้กับ bit EXEN 2 ของ register T2CON ดังนี้

1. เมื่อกำหนดbit EXEN2 เป็น 0 Timer 2 ยังทำงานเป็นวงจรนับ/ตรวจจับเวลา เมื่อมีการ overflow ขึ้น bit ใน register TF2 จะถูกเซต และสามารถนำไปสร้างการ interrupt ขึ้นได้

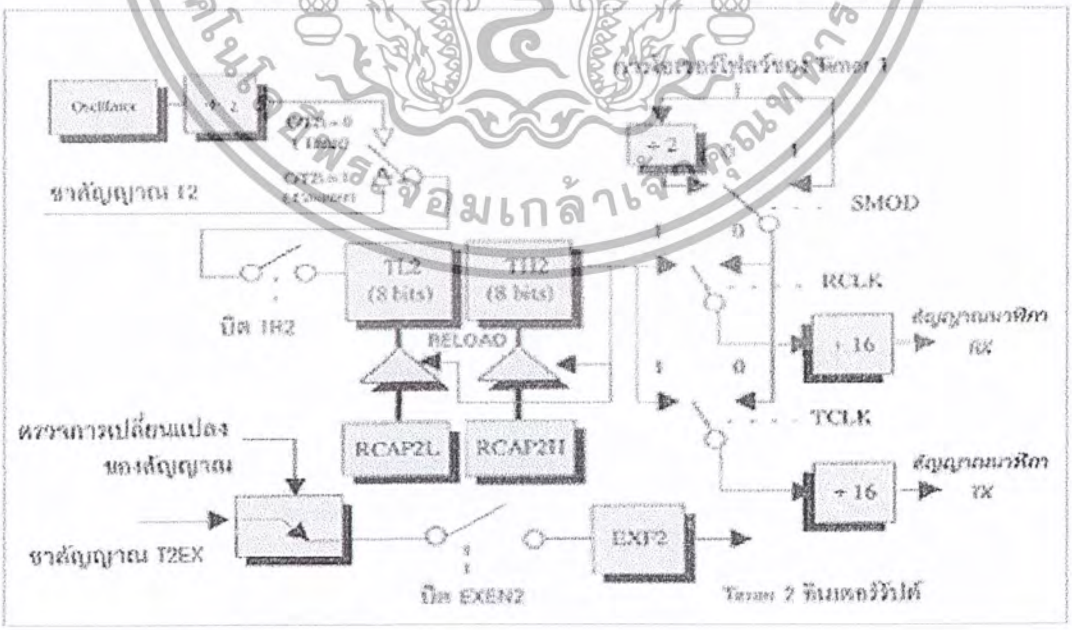
2. เมื่อกำหนดค่า bit EXEN2เป็น 1 การทำงานจะครอบคลุมการทำงานลักษณะข้างต้น แต่จะเพิ่มเมื่อมีการเปลี่ยนแปลงระดับ สัญญาณทางขาสัญญาณ T2EX จาก logic สูง ไปเป็น logic ต่ำ จะมีผลทำให้ค่าข้อมูลภายใน register ของ Timer2 คือ TL2 และ TH2 ถูกนำไปใส่ (Capture) ให้กับ register RCAP2L และ RCAP2H ซึ่งเป็น register หน้าที่พิเศษ หรือ SFR ที่มีใน Microcontroller เบอร์ 8052 เท่านั้น นอกจากนี้จะมีผลทำให้ bit EXF2 ภายใน register T2CON มีค่าเป็น 1 สามารถนำไปใช้งานในการ interrupt ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.9 แสดงการทำงานของโหมดแคปเจอร์ของ TIMER2

- Auto-reload mode: สามารถทำงานได้ 2 ลักษณะเช่นเดียวกัน
- Baud rate Generator : ของ Timer 2 จะมีความแตกต่างจาก Timer 0 และ Timer 1 โดยวงจรรับ และการส่ง สามารถเป็นค่าที่ต่างกันได้ ขึ้นอยู่กับการกำหนดค่าให้กับ bit TCLK และ RCLK ของ Timer 2



ภาพที่ 2.10 แสดงการทำงานในโหมดกำเนิดอัตราบอดของ TIMER2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

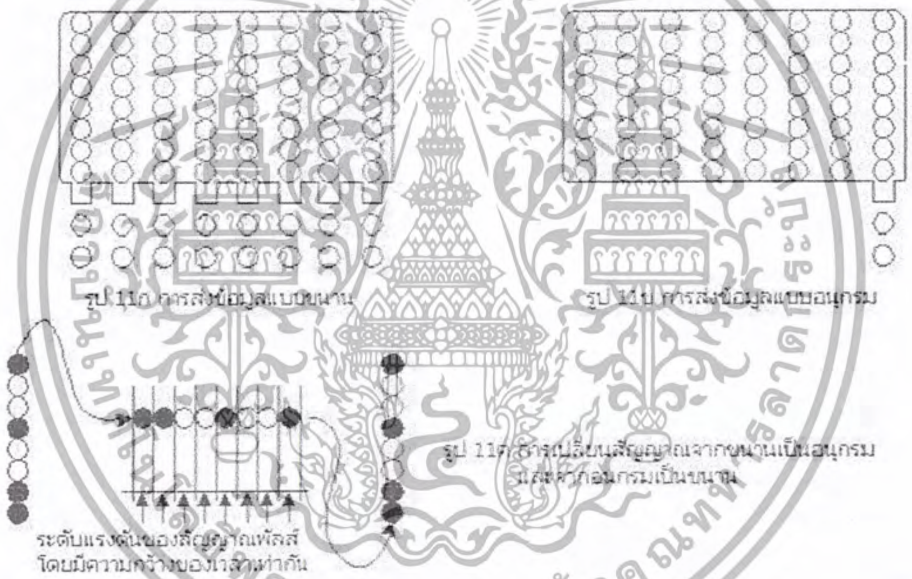
การทำงานของโหมดนี้คล้ายคลึงกับการทำงานใน Auto-reload mode กล่าวคือค่าใน register TH2 เปลี่ยนแปลงจากค่า0FFH ไปเป็นค่า 0 หรือที่เรียกว่า overflow จะมีผลให้มีการโหลดข้อมูลขนาด 16 bit จาก register RCAP2H และ RCAP2L ซึ่งมีการเตรียมค่าล่วงหน้าแล้วโดยอัตโนมัติในการจับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ความรู้เบื้องต้นเกี่ยวกับพอร์ตอนุกรม

3.1 กล่าวนำ

ข้อมูลในคอนโทรลเลอร์จะเป็นข้อมูลที่มีความยาวขนาด 1 ไบต์ หรือ 8 บิตซึ่งโดยถ้าเราให้ส่งข้อมูลพร้อมๆกันไป 8 บิตจะเป็นวิธีการส่งข้อมูลแบบขนาน แสดงได้ดังรูปที่ 2.1 จะเป็นการส่งข้อมูลขนาด 8 บิตพร้อมกันไปยังอุปกรณ์ภายนอก และจะต้องมีจำนวนของสายสัญญาณจำนวน 8 เส้น เพื่อให้พอดีกับจำนวนของบิตที่ต้องการจะส่ง การส่งข้อมูลแบบขนานจึงทำให้มีการส่งข้อมูลที่มีความรวดเร็ว แต่ถ้าหากมีการสื่อสารข้อมูลในระยะไกล ก็จะต้องใช้จำนวนของสายและระยะทางของสายมากขึ้นจึงทำให้มีการสิ้นเปลืองค่าใช้จ่ายสูง



ภาพที่ 3.1 แสดงการส่งข้อมูลแบบขนานและแบบอนุกรม

ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงถูกนำมาใช้ ในการสื่อสาร โดยจะใช้สายเพียงเส้นเดียวในการส่งข้อมูล หรือรับข้อมูล (คำว่าเส้นเดียวหมายความว่าสายส่ง (TxD) 1 เส้น สายรับ (RxD) 1 เส้น และสายกราวด์ร่วม (Ground) 1 เส้น นำมาใช้สื่อสารข้อมูลกับอุปกรณ์ภายนอกในระยะทางที่ไกล ดังในรูปที่ 2.1 ถ้าหากต้องการส่งข้อมูลขนาด 8 บิต ก็จะทำการส่งข้อมูลออกไปทีละบิตเป็นลำดับไปจนกว่าจะรบจำนวนทั้ง 8 บิต จะแสดงการเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม ข้อมูลจะถูกส่งไปตามสายสัญญาณที่ละบิตตามจังหวะเวลาที่กำหนด เป็นความกว้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของพัลส์ โดยจังหวะเวลาที่กล่าวนี้จะต้องมีมาตรฐาน ของฝ่ายส่ง และฝ่ายรับด้วย ในการรับสัญญาณที่ส่งมาทีละบิตจะทำการตรวจสอบระดับแรงดันของสัญญาณที่เข้ามาเพื่อแปลงเป็นลอจิก "1" หรือ "0" เมื่อรับข้อมูลเข้ามาครบใน 1 ไบต์ที่กำหนดไว้ก็จะถูกเปลี่ยนให้อยู่ในรูปแบบของข้อมูลแบบขนานเหมือนดังเช่นเดิม

3.2 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

ในการสื่อสารข้อมูลแบบอนุกรม เพื่อรับหรือส่งข้อมูล จะเป็นลักษณะของกลุ่มข้อมูล ดังนั้นอัตราความเร็วจะต้องมีค่าเท่ากันระหว่างการรับและการส่งโดยทั่วไปเราจะระบุความเร็วของจำนวนบิตในการรับและส่งข้อมูล เป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด(Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1200, 2400, 4800 และ 9600 บิตต่อวินาที ในรูป ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104.1 ไมโครวินาที และเวลาในการรับ และส่งข้อมูลทั้ง 8 บิต จะมีค่าเท่ากับ 8 x 104.1 หรือ 832.8 ไมโครวินาที

ภาพที่ 3.2 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

3.3 รูปแบบของการสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมแบบอะซิงโคนัส เป็นวิธีการรับและส่งข้อมูล โดยไม่ต้องอาศัยสัญญาณนาฬิกาส่งร่วมไปด้วย แต่จะใช้อัตราความเร็วของจำนวนข้อมูลต่อวินาที และจะทำการเพิ่มบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง เพื่อจะได้ทำการตรวจสอบข้อมูลได้อย่างถูกต้องมากยิ่งขึ้นดังแสดงดังภาพที่ 3.2 ซึ่งประกอบด้วยกัน 4 ส่วนคือ

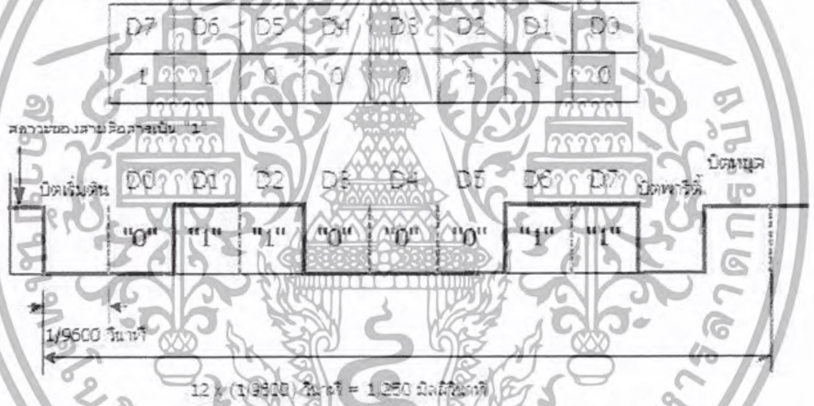
1. บิตเริ่มต้น(Start bit) จะมีขนาด 1 บิต จะเป็นละดับลอจิกตรงกันข้ามกับ ระดับลอจิกของสภาวะสายสื่อสารขณะที่ยังไม่มีการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. บิตข้อมูล(Data bit) จะเริ่มจากบิตที่มีนัยสำคัญต่ำสุดก่อนหรือ บิต LSB ก่อน โดยข้อมูลที่จะส่งอาจจะมีขนาด 5 , 6, 7 หรือ 8 บิตก็ได้

3. บิตแสดงสภาวะเลขคู่หรือเลขคี่ (Parity bit) มีขนาด 1 บิต โดยบิตนี้จะนำไปต่อท้ายกับบิตข้อมูลค่า ของบิตนี้ขึ้นอยู่กับจำนวนค่าของข้อมูลที่เป็น "1" โดยเลือกการส่งข้อมูลเป็นแบบพาริตี้คู่ หรือ พาริตี้คี่ ตัวอย่าง ถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตี้คู่ แต่ข้อมูลมีเลข 1 เป็นจำนวนคี่ ก็จะให้บิตพาริตี้นี้เป็น "1" เพื่อจะได้จำนวนเลข "1" เป็นคู่นั่นเอง ทำนองเดียวกันทางด้านรับเองก็ต้องมีการตรวจสอบจำนวนข้อมูลที่รับเข้ามาเป็น "1" รวมทั้งบิตพาริตี้ 1 บิต ถ้ามีค่า "1" เป็นจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง * สามารถกำหนดการรับและส่งข้อมูลเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตี้บิตก็ได้

4. บิตสุดท้ายหรือบิตหยุด (Stop bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล โดยจะทำให้ขาของข้อมูลมีสถานะ ลอจิกเป็น "1" ซึ่งอาจมีจะจำนวนมากกว่าหนึ่งบิตก็ได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต



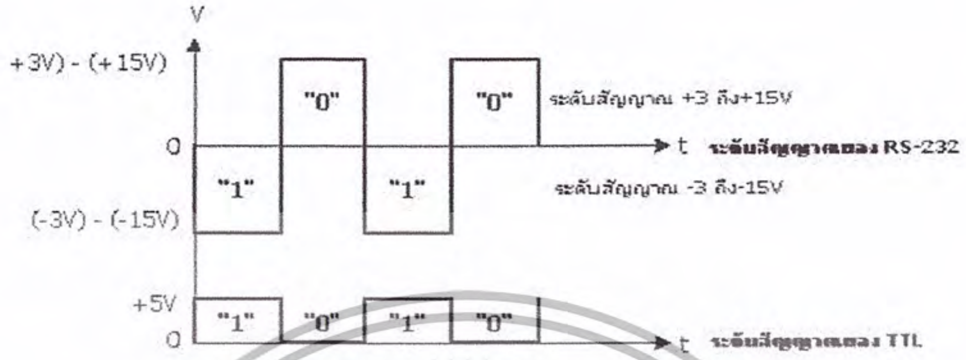
ภาพที่ 3.3 แสดงการส่งข้อมูลขนาด 8 บิตอนุกรม

3.4 การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232

การกำหนดมาตรฐานการเชื่อมต่อแบบอนุกรม EIA RS-232 (x) เป็นมาตรฐานทางอุตสาหกรรม โดยคณะกรรมการสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association) ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบ อะซิงโครนัส 2 ทิศทาง เพื่อให้มีการใช้งาน ในการเชื่อมต่อที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณจะส่งสัญญาณจะกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุต โดยมีระดับ สัญญาณตั้งแต่ 3 โวลท์ จนถึง 15 โวลท์ สำหรับลอจิก "0" และมีระดับแรงดันที่ - 3 โวลท์ จนถึง - 15 โวลท์ สำหรับลอจิก "1" ดังนั้นสังเกตได้ว่าจะมีระดับแรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่างออกไปจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

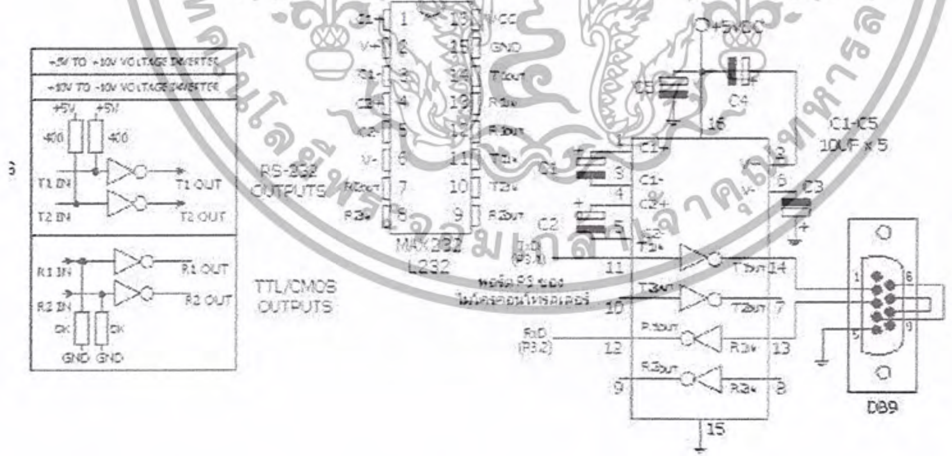
ระบบ ไอซี ดิจิตอลทั่วๆไป การต่อใช้งานจึงต้องมีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดันจาก 0 - 5 โวลท์ จากไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า + 3 หรือต่ำกว่า - 3 โดยจะมีไอซีสำเร็จรูปพร้อมใช้งาน หรืออาจจะต่อวงจรทรานซิสเตอร์ได้



ภาพที่ 3.4 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL

3.5 ไอซี MAX232 ,L232

ไอซี MAX232, L232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232 เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้



ภาพที่3.5 แสดงตำแหน่งขาของไอซี MAX232, L232 และการต่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ทฤษฎีและหลักการ PLC

โปรแกรมเมเบิ้ลลอจิก คอนโทรลเลอร์ (PLC) เป็นอุปกรณ์ที่ใช้สำหรับควบคุมเครื่องจักร หรือระบบกระบวนการให้ทำงานตามคำสั่งของผู้ใช้ และข้อมูลต่างๆ ที่ได้รับจากหน่วย อินพุท และเอาท์พุทของ PLC การทำงานของ PLC เป็นได้ทั้งการทำงานตามช่วงเวลา ตามลำดับขั้นตอน ฟังก์ชันทางคณิตศาสตร์และอื่นๆ

การใช้ PLC สำหรับควบคุมเครื่องจักรหรืออุปกรณ์ต่างๆ ในโรงงานอุตสาหกรรมจะมีข้อได้เปรียบมากกว่าการใช้ระบบของรีเลย์ (Relay) ซึ่งจำเป็นจะต้องเดินสายไฟ หรือที่เรียกว่า Hard-Wired ฉะนั้นเมื่อมีความจำที่ต้องเปลี่ยนกระบวนการผลิต หรือลำดับการทำงานใหม่ ก็จะต้องเดินสายไฟใหม่ ซึ่งเสียเวลาและค่าใช้จ่ายสูง แต่เมื่อเปลี่ยนมาใช้ PLC แล้ว การเปลี่ยนแปลงกระบวนการผลิต หรือลำดับการทำงานใหม่นั้นทำได้โดยการเปลี่ยนใหม่เท่านั้น นอกจากนี้แล้ว PLC ยังใช้ระบบโซลิดสเตท ซึ่งมีความน่าเชื่อถือกว่าระบบเดิม การกินกระแสไฟฟ้าก็น้อยกว่า และสะดวกกว่าเมื่อต้องการขยายขั้นตอนการทำงานของเครื่องจักร

ส่วนประกอบของ PLC

แบ่งได้เป็น 4 ส่วนด้วยกัน คือ

1. ส่วนที่เป็นหน่วยประมวลผลกลาง (Central Processing Unit : CPU)
2. หน่วยความจำ (Memory Unit)
3. ส่วนที่เป็นอินพุท / เอาท์พุท (Input/Output)
4. ส่วนที่เป็นอุปกรณ์การโปรแกรม (Programming Device)

4.1 หน่วยประมวลผลกลาง (CPU)

หน่วยประมวลผลกลางเป็นส่วนมันสมองของระบบ โดยภายในประกอบด้วยวงจร Logic Gate ชนิดต่างๆ หลายชนิด และมี Microprocessor-base ใช้สำหรับแทนอุปกรณ์จำพวกรีเลย์ (Relay) เคาท์เตอร์ (Counter)ไทเมอร์ (Timer) และซีเควนเซอร์ (Sequencer) เพื่อให้ผู้ออกแบบใช้วงจรรีเลย์แลดเดอร์ลอจิก (Relay Ladder Logic) เข้าไปได้

หน่วยประมวลผลกลาง จะรับข้อมูล (Read Input Data) จากอุปกรณ์ให้สัญญาณ (Sensing Device) ต่างๆ จากนั้นจะปฏิบัติการเก็บข้อมูลโดยใชโปรแกรมจากหน่วยความจำและส่งข้อมูลที่เหมาะสมถูกต้องไปยังอุปกรณ์ควบคุม (Control Device) แหล่งของกระแสไฟฟ้าตรง (DC Current) สำหรับสร้างแรงดันไฟฟ้าต่ำๆ (Low Level Voltage) ซึ่งใช้โดยโปรเซสเซอร์ (Processor) และ I/O Module ซึ่งแหล่งจ่ายไฟนี้จะเก็บไว้ที่หน่วยประมวลผลกลางหรือแยกออกไปติดตั้งที่จุดอื่นๆ ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลกของหน่วยประมวลผลกลางจากโปรแกรมทำได้โดยรับข้อมูลจากหน่วยอินพุทเข้ามาทำการประมวลผล แล้วส่งข้อมูลที่ได้ไปยังเอาท์พุทจากนั้นก็วกกลับไปรับข้อมูลเข้ามาอีก การทำในลักษณะนี้เรียกว่าการสแกน (Scan Tine)

4.2 หน่วยความจำ (Memory Unit)

เป็นองค์ประกอบที่สำคัญของระบบเพราะใช้เป็นที่เก็บโปรแกรมและข้อมูลขนาดของหน่วยความจำจะเป็นสิ่งกำหนดความสามารถของระบบ ระบบที่มีหน่วยความจำมากจะทำให้ผู้ใช้สามารถเขียนโปรแกรมที่มีความซับซ้อนได้มากขึ้น

หน่วยความจำของ PLC ประกอบด้วยหน่วยความจำชนิด RAM และ ROM หน่วยความจำชนิด RAM ทำหน้าที่เก็บโปรแกรมของผู้ใช้และข้อมูลสำหรับที่ใช้ในการปฏิบัติงานของ PLC ส่วน ROM ทำหน้าที่เก็บโปรแกรมสำหรับใช้ในการปฏิบัติงานของ PLC ตามโปรแกรมของผู้ใช้ คำว่า ROM ย่อมาจาก Read Only Memory สามารถโปรแกรมได้แต่ลบไม่ได้ถ้าชำรุดแล้วซ่อมไม่ได้

หน่วยความจำชนิดต่างๆ

4.2.1. RAM (Random Access Memory)

หน่วยความจำประเภทนี้จะมีแบตเตอรี่เล็กๆ ต่อไว้เพื่อใช้เลี้ยงข้อมูลเมื่อเกิดไฟดับ การอ่านและการเขียนโปรแกรมลงใน RAM ทำได้ง่ายมากจึงเหมาะกับการใช้งานในระยะทดลองเครื่องที่มีการเปลี่ยนแปลงแก้ไขโปรแกรมบ่อยๆ

4.2.2. EPROM (Erasable Programmable Read Only Memory)

หน่วยความจำชนิดนี้จะต้องใช้เครื่องมือพิเศษในการเขียนโปรแกรม การลบโปรแกรม ทำได้โดยใช้แสงอุลตร้าไวโอเลตหรือตากแดดร้อนๆ เป็นเวลานานมีข้อดีตรงที่ไม่ต้องการเปลี่ยนโปรแกรม

4.2.3. EEPROM (Electrical Erasable Programmable Read Only Memory)

หน่วยความจำชนิดนี้ไม่ต้องใช้เครื่องมือพิเศษในการเขียนและลบโปรแกรม โดยใช้วิธีทางไฟฟ้าเหมือนกับ RAM นอกจากนั้นไม่จำเป็นจะต้องมีแบตเตอรี่สำรองไฟไว้ เมื่อไฟดับซึ่งมีราคาแพงกว่าแต่จะรวมคุณสมบัติที่ดีของ RAM และ EPROM เอาไว้ด้วยกัน

4.3. ส่วนที่เป็นอินพุท/เอาท์พุท (I/O Unit)

หน่วยอินพุทจะทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอกที่เป็นสวิตช์ และ ตัวตรวจจับชนิดต่างๆ ของเครื่องจักรและกระบวนการ แล้วแปลงสัญญาณเป็น AC หรือ DC ที่เหมาะสมเพื่อส่งให้แก่หน่วยประมวลผลกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของเอาท์พุท จะทำหน้าที่รับค่าสภาวะที่ได้จากการประมวลผลของหน่วยประมวลผลกลาง แล้วนำค่าที่ได้ไปขยายสัญญาณออกให้มีขนาดใหญ่พอจะขับอุปกรณ์ภายนอก เช่น รีเลย์, โซลินอยด์ หรือ หลอดไฟ, มอเตอร์, วาล์ว, ปั้ม เป็นต้น นอกจากนั้นแล้วยังทำหน้าที่แยกสัญญาณของหน่วยประมวลผลกลางออกจากอุปกรณ์เอาท์พุท เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้นได้

โดยปกติเอาท์พุทนี้จะสามารถขับโหลดได้ด้วยกระแสไฟฟ้าประมาณ 1-2 แอมแปร์ ถ้าโหลดต้องการกระแสไฟฟ้ามากกว่านี้ จะต้องต่อเข้ากับอุปกรณ์ขับหรือขยายอีกที่หนึ่ง เช่น รีเลย์, โซลิดสเตทรีเลย์ และคอนแทรกเตอร์ เป็นต้น

4.4. อุปกรณ์ติดต่อภายนอก (Peripheral Device)

เป็นอุปกรณ์ต่างๆ ที่อำนวยความสะดวกในการพัฒนาโปรแกรม หน้าที่ของอุปกรณ์ภายนอกได้แก่

- ป้อนโปรแกรมเข้าไปใน Memory ของระบบ
- ใช้ในการแก้ไข (Debug) โปรแกรม
- ใช้ในการเก็บรักษาโปรแกรม
- ใช้ในการพิมพ์โปรแกรม
- ใช้ในการแสดงสภาวะการควบคุม

ตารางที่ 4.1 แสดงประเภทหน้าที่ของอุปกรณ์ติดต่อภายนอก

อุปกรณ์ต่อร่วม	หน้าที่การใช้งานเกี่ยวกับโปรแกรม				
	ป้อน	แก้ไข	โหลดใหม่	พิมพ์	สภาวะ
1. Programing Console	*	*			*
2.EPROM Writer			*		
3.Printer				*	
4.Graphic Programming	*	*			*
5.CRT Monitor	*	*			*
6.Audio Cassette			*		
7.Ladder Solfware	*	*	*	*	*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ความแตกต่างระหว่างคอมพิวเตอร์กับ PLC

4.5.1 PLC ถูกออกแบบมีความทนทานต่อสภาพแวดล้อมของโรงงานอุตสาหกรรม เช่น ความร้อน ความเย็น ระบบไฟฟ้ารบกวน การสั่นสะเทือน

4.5.2 การใช้โปรแกรมของ PLC จะไม่ยุ่งยากเหมือนของคอมพิวเตอร์ เนื่องจาก PLC มีระบบตรวจสอบตัวมันเอง ทำให้ใช้งานง่ายและบำรุงรักษาง่าย

4.5.3 PLC ทำงานตามที่โปรแกรมไว้เพียงโปรแกรมเดียว ทำให้ไม่ยุ่งยาก ส่วนที่คอมพิวเตอร์จะทำงานที่โปรแกรมหลายๆ โปรแกรมพร้อมกันจึงมีความยุ่งยากมากกว่า

4.5.4 PLC ใช้ควบคุมการผลิตทุกชนิด ทั้งแบบอนาล็อกและแบบลอจิก (ON-OFF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ทฤษฎีเบื้องต้น Visual Basic Version 6.0

Visual Basic เป็นภาษาคอมพิวเตอร์ที่มีศักยภาพสูง และออกแบบมาเพื่อทำงานบนระบบวินโดวส์ โดยเฉพาะ คำว่า Visual จะหมายถึงวิธีการที่ใช้สร้างส่วนติดต่อกับผู้ใช้บนกราฟฟิก (GUI = Graphic User Interface) ส่วน Basic เป็นคำที่ย่อมาจาก Beginner All-Purpose Symbolic Instruction Code นั่นเอง จากนี้ขอเรียก Visual Basic สั้นๆ ว่า VB

5.1 ส่วนประกอบของหน้าต่าง VB

ภาพที่ 5.1 แสดงการส่วนประกอบของหน้าต่าง VB

5.1.1 Manu Bar

เป็นส่วนที่เก็บเครื่องมือในการทำงานทั่วไปของ VB เช่น Ran, Save

5.1.2 Tool Box

เป็นที่เก็บคอนโทรลต่างๆ ที่ใช้งานทั้งหมด ซึ่งจะอธิบายรายละเอียดต่อไป

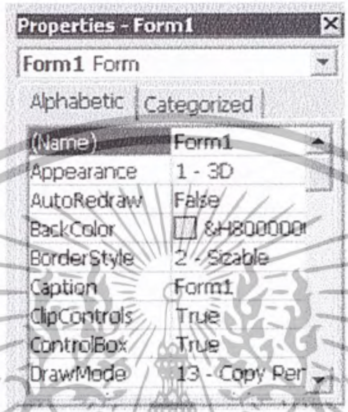
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 Project Explorer

เป็นหน้าต่างแสดงรายละเอียดของฟอร์ม (Form) ที่สร้างขึ้นรวมทั้ง Module และ Class ของงานนั้นๆ ด้วย ถ้าต้องการซ่อนหรือแสดงหน้าต่าง Project Explorer กด Ctrl+R บนแป้นพิมพ์

5.1.4 Properties Window

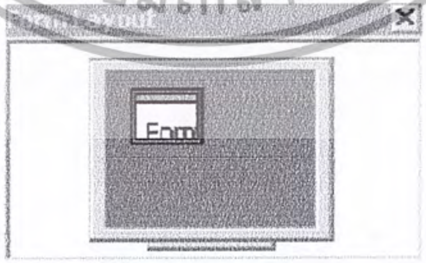
เป็นหน้าต่างที่ใช้ในการกำหนดคุณลักษณะของ (Control), ฟอร์ม (Form) ซึ่งจะอำนวยความสะดวกในการออกแบบฟอร์มโดยไม่ต้องเขียนโปรแกรมให้ยุ่งยาก



ภาพที่ 5.2 หน้าต่างของProperties

5.1.5. Form Layout

เป็นหน้าต่างที่ใช้ในการกำหนดตำแหน่งของฟอร์มบนหน้าจอในขณะรัน (Run) ตำแหน่งหน้าต่างของฟอร์มนั้นๆ สามารถ โดยการกำหนดคลิกเมาส์แล้วลากกรอบที่ปรากฏอยู่ในหน้าต่าง Form Layout โดยที่กรอบที่อยู่ภายใน อยู่ ณ ตำแหน่งใด ฟอร์มที่ขณะรัน โปรแกรมก็จะแสดงอยู่ตามตำแหน่งนั้น



ภาพที่ 5.3 หน้าต่างของForm Layout

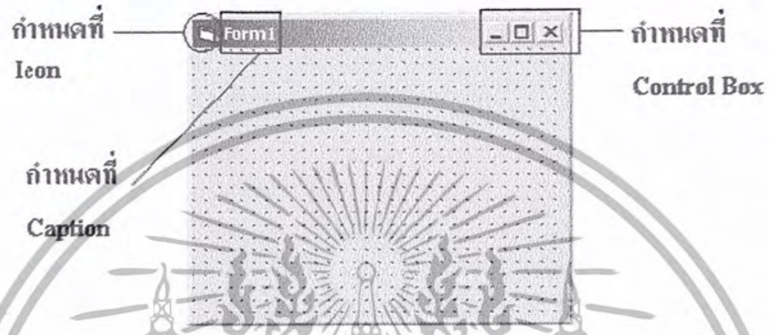
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 รู้จักกับฟอร์ม (Form)

ฟอร์ม (Form) เป็นพื้นที่สำหรับการออกแบบ โดยนำคอนโทรลมาวางตามความเหมาะสม ค่าต่างๆให้กำหนดในส่วนของ (Properties) ดังรูปด้านล่าง

5.2.1 รู้จักกับพรอพเพอร์ตี้ (Properties), ออปเจ็ก (Object) , และอีเวนท์ (Event)

VB เป็นภาษาที่อาศัยแนวคิดของ OOP (Object Oriented Programming) ซึ่งสิ่งที่หมายถึง ออปเจ็ก (Object) ในภาษา VB นั่นก็คือเครื่องมือคอนโทรล (Control) ต่างๆยิ่งจะกล่าวต่อได้อีก



ภาพที่ 5.4 หน้าต่างของลักษณะ Form

ว่าในแต่ละออปเจ็กจะมีลักษณะเฉพาะเช่น ความยาว ความกว้าง เรียกว่าพรอพเพอร์ตี้ (Properties) ในแต่ละออปเจ็กนั้นยังมีพฤติกรรมประจำตัว เรียกว่า เมธอด (Method) เช่น การถูกเมาส์ชี้เป็นต้น ในการเขียนโปรแกรมนั้นจะมี อีเวนต์ (Event) ซึ่งในความหมายจะคล้ายกับเมธอด ดังนั้นเราอาจสรุปได้ว่า

Properties คือ การกำหนดคุณลักษณะของคอนโทรล (Control) และ ฟอร์ม (Form)

Method คือ สิ่งที่คอนโทรลและฟอร์มสามารถกระทำได้ เช่น สามารถวาดรูปได้

Event คือ ความสามารถตอบสนองของฟอร์มหรือคอนโทรลตัวนั้น เช่น กด Enter หรือ Esc บนแป้นพิมพ์

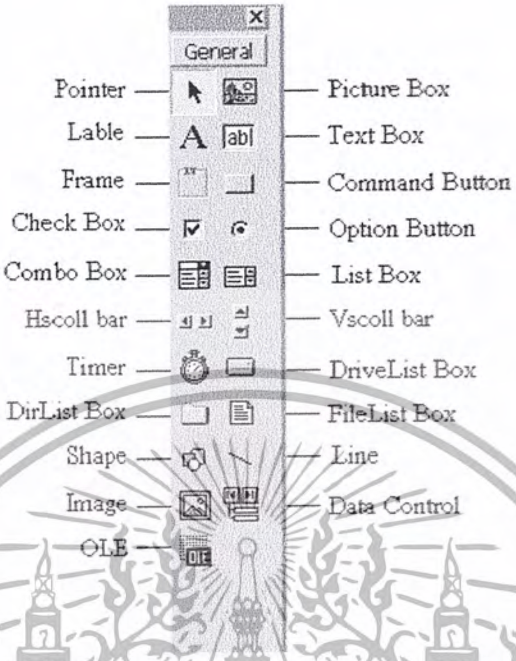
ฉะนั้นเราจะหลีกเลี่ยงไม่ได้เลยกับการกำหนดค่า Properties ซึ่งโดยรวมๆ แล้วค่า Properties จะมีการกำหนดที่ใกล้เคียงกันแต่จะมีบางตัวที่กำหนดต่างกันซึ่งจะอธิบายแยกเฉพาะ Properties ที่ต่างกันออกไป

5.3 ปุ่มคอนโทรล (Control) ต่างๆ

เมื่อเริ่มการทำงานจะปรากฏ ฟอร์ม 1 (Form 1) มาให้ แต่ถ้าไม่มีเราสามารถเพิ่มฟอร์มใช้งานได้เอง โดยใช้คำสั่ง ***Project->Add Form*** หรือ คลิกที่ปุ่ม Add form จากนี้เราสามารถสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าตาคอนโทรลฟอร์มได้ โดยเลือกคอนโทรลแต่ละตัวแล้วนำมาสร้างลงบนฟอร์ม หรือ จะทำการดับเบิ้ลคลิ๊กที่ตัวคอลโทรลนั้นๆ เลยก็ได้

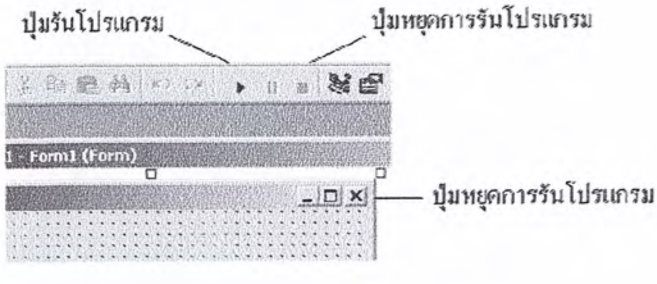


ภาพที่ 5.5 รายการเครื่องมือคอนโทรลต่าง ๆ

5.4 การเขียนโปรแกรมเบื้องต้น

การเขียนโปรแกรมต่างๆ ใน VB จะเป็นการตอบสนองตามเหตุการณ์ต่างๆ ในแต่ละฟอร์มและคอนโทรล การตอบสนองตามเหตุการณ์ต่างๆ นี้เรียกว่า Event Driven ในแต่ละเหตุการณ์หล่านั้นมีตัวจุดฉนวน (Trigger) และในการทำงานของฟอร์มและคอนโทรลจะถูกสร้างมาเพื่อสนองตอบต่อเหตุการณ์ ในแต่ละเหตุการณ์จะถูกบรรจุอยู่ใน โปรซีเยอร์ (Procedure) จึงเป็นหน้าที่ที่สำคัญของผู้ซึ่งที่ ออกแบบโปรแกรมจะต้องระบุตำแหน่งของโปรซีเยอร์ ให้ถูกต้องเสียก่อน

เมื่อได้เรียนรู้การออกแบบฟอร์มด้วยคอนโทรลต่างๆ แล้ว ถ้าต้องการรัน ฟอร์มนั้นให้กด F5 หรือ คลิกที่ปุ่มรันโปรแกรม และเมื่อต้องการให้หยุดรัน ให้คลิกที่ปุ่มหยุดการรันโปรแกรม



ภาพที่ 5.6 ปุ่มการรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 ตัวแปร (Variable)

เมื่อออกแบบคอนโทรลต่างๆ แล้ว เราจะต้องเขียนโปรแกรมแยกตามออปเจ็ค (Object) ตัวอย่างเช่น ถ้าออปเจ็คเป็นปุ่มคอมมานด์ (Command Button) เมื่อเขียนโปรแกรมเกี่ยวกับปุ่มคอมมานด์เราก็จะต้องเข้าไปเขียนในโปรซีเยอร์ของปุ่มคอมมานด์เท่านั้น ในการเขียนโปรแกรมนี้สิ่งที่ต้องคำนึงถึงเสมอก็คือ **ตัวแปร** เราจะต้องระลึกเสมอว่าโปรแกรมที่เขียนขึ้นนั้นต้องใช้ตัวแปรใดบ้าง เพื่อตอบสนองงานอะไรต้องใช้ตัวแปรมากน้อยแค่ไหน จึงหลีกเลี่ยงไม่ได้เลยที่ต้องทำความรู้จักกับ ตัวแปรทุกตัวเสียก่อน

5.5.1 ชนิดของตัวแปร

ตัวแปร เป็นชื่อที่เราตั้งขึ้นเพื่อใช้ในการเก็บค่าข้อมูล ซึ่งสามารถแปรเปลี่ยนค่าไปตามที่เรากำหนด ชนิดของตัวแปรก็จะขึ้นอยู่กับชนิดของข้อมูลที่ตัวแปรนั้นเก็บค่าอยู่

1. String	$	เป็นตัวแปรที่เก็บข้อความ
2. Integer	%	2 ไบต์ เก็บค่าได้ -32768 ถึง 32768
3. Long	&	4 ไบต์ เก็บค่าได้ -2147483648 ถึง 2147483647
4. Single	!	4 ไบต์ เก็บค่าได้ -3402823 E38 ถึง -1401298 E-45
5. Double	#	8 ไบต์ เก็บค่าได้ -179769313486232 E308 ถึง -494065645841247 E-324 หรือไม่ติดลบ
6. Currency	@	8 ไบต์ -9223372036854775808 ถึง 9223372036854775808
7. Variant		เป็นตัวแปรที่เก็บค่าได้ทุกชนิดยกเว้น Object และ Type
8. Constant		คือการกำหนดค่าให้กับตัวแปร
9. Object		ตัวแปรแทนชุดคำสั่ง
10.Boolean		เป็นตัวแปรเก็บค่าเพียง 2 ค่า คือ True/False

5.5.2 หลักการตั้งชื่อตัวแปร

1 ขึ้นต้นด้วยอักษร เช่น name, surname, no เป็นต้น

2 ความยาวต้องไม่เกิน 255 ตัวอักษรในขณะที่ชื่อฟอร์ม , คอนโทรลและโมดูล (Module) ต้องไม่เกิน 40 ตัวอักษร

3 ห้ามใช้คำหลัก (Keyword) มาตั้งเป็นชื่อตัวแปรเช่น If, For, Function, Len, Or, Mod

5.5.3 การประกาศตัวแปร

รูปแบบ **Dim** ชื่อตัวแปรการตามกฎการตั้งชื่อ **As** ชนิดของตัวแปร

หรือ **Dim** ชื่อตัวแปรการตามกฎการตั้งชื่อ สัญลักษณ์ของตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.3.1 ตัวแปรรับข้อความ : สตริง (String)

การประกาศตัวแปรสตริง

รูปแบบ Dim ตัวแปร As String หมายถึง ตัวแปรที่จะรับค่าเฉพาะข้อความ

หรือ Dim ตัวแปร $ หมายถึง ตัวแปรประกาศตัวเป็น String

หรือ Dim ตัวแปร As String*40 หมายถึง ตัวแปรเก็บข้อความได้ 40

5.5.3.2 ตัวแปรรับตัวเลข : Integer, Long, Single, Double, Currency

ในการใช้ตัวแปรตัวเลขนี้จะใช้ต่างกันตรงปริมาณตัวเลข เช่น Integer จะรับค่าตัวเลขจำนวนเต็มแต่ Long จะเก็บค่าตัวเลขจำนวนเต็มที่เก็บค่าได้มากกว่า Integer ส่วน Single จะเก็บค่าตัวเลขที่มีทศนิยมที่น้อยกว่า Double เป็นต้น

การประกาศตัวแปร

รูปแบบ Dim X As Integer

Dim Y As Long

Dim Z As Single

5.5.3.3 ตัวแปรแวเรียน : Varient

เป็นตัวแปรที่เก็บได้ทุกค่า โดยไม่ต้องประกาศตัวแปรใดก็ได้ VB จะประกาศตัวแปรเป็นแบบแวเรียนโดยอัตโนมัติ นั่นหมายถึง ตัวแปรแวเรียนจะมีการแปลงค่าจากตัวเลขไปเป็นข้อความ และจากข้อความไปเป็นตัวเลข แต่ตัวแปรแวเรียนจะไม่รับตัวแปรประเภท Object และ Type

การประกาศตัวแปร

รูปแบบ **Dim X** ตัวแปรถูกประกาศเป็นตัวแปรประเภทแวเรียน

เช่น Dim Valueประกาศตัวแปร Value เป็นประเภทแวเรียน

Value = "3"ตัวแปรรับข้อความ ซึ่งในที่นี้คือ 3

Value = "Class"& Value ผลลัพธ์ที่ได้คือ Class3

5.5.3.4 ค่าคงที่ : Constant

เป็นการกำหนดค่าตายตัวให้กับตัวแปร ซึ่งในการเขียนโปรแกรมจะไม่สามารถไปเปลี่ยนค่าภายในตัวแปรได้อีก

การประกาศตัวแปรสตริง

รูปแบบ Const ตัวแปร = ค่าคงที่

เช่น Const Date = #09/9/2001# เป็นการประกาศตัวแปร Dateเป็น วัน ที่ 09/9/2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.3.5 บูลีน : Boolean

เป็นตัวแปรที่เก็บค่าทางตรรกะ ซึ่งจะมีค่าเป็นได้เพียง 2 ค่า เท่านั้น คือ ค่าที่เป็น True (จริง) และค่าที่เป็น False (เท็จ)

การประกาศตัวแปรสตริง

รูปแบบ Dim ตัวแปร As Boolean

เช่น Bcc = True

5.5.3.6 ตัวแปรออปเจ็ก : Object

มีลักษณะเหมือนกับแวเรียนแต่มักใช้แทนการเขียนคำสั่งที่ยาวๆ และไม่นำไปคำนวณหรือแทนค่านอกจากนั้นยังใช้เขียนแทนชื่อของฟอร์ม , คอนโทรล เป็นต้น ส่วนตัวแปรออปเจ็กนั้น จะต้องใช้คู่กับคำสั่ง Set

การประกาศตัวแปรสตริง

รูปแบบ Dim ตัวแปร As Object

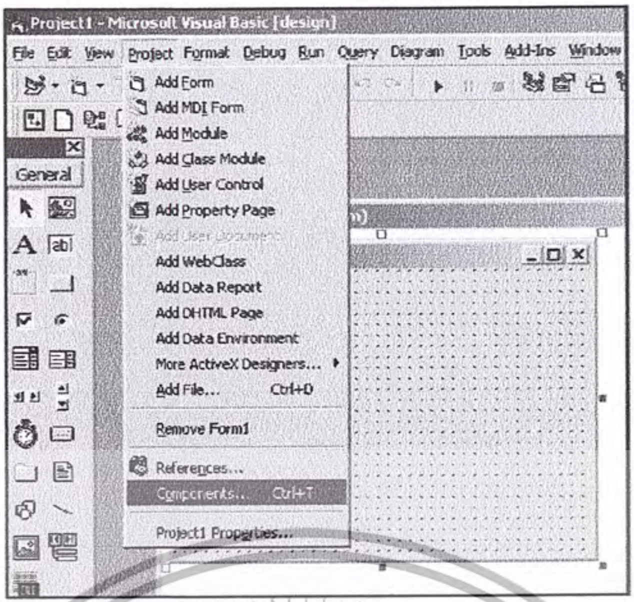
เช่น Set ObjPic = LoadPicture ("C:\My documents\pic1.bmp")

คือ กำหนดตัวแปร ObjPic เป็นคำสั่งดึงรูปภาพ pic1.bmp จากไดรฟ์ C ใน My documents

5.6 การเขียนโปรแกรมติดต่อ และควบคุม Serial Port กับ Visual Basic

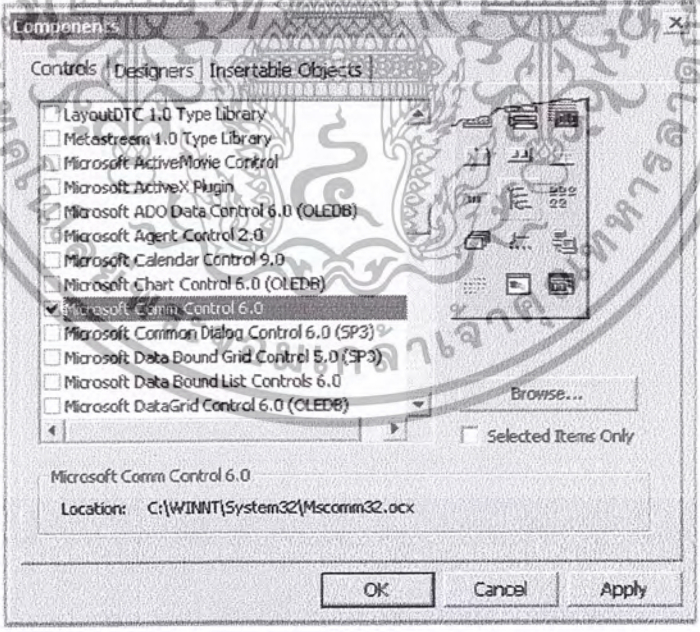
คอนโทรลที่สำคัญในการทำให้ Visual Basicสามารถสื่อสารผ่านพอร์ตอนุกรมได้นั่นก็คือ คอนโทรล MSComm ซึ่งไม่ใช้คอนโทรลมาตรฐาน ดังนั้นถ้าเราต้องการใช้งาน MSComm เราจะต้องทำการเพิ่มคอนโทรลนี้เข้าไปใน Toolbox ซึ่งสามารถกระทำได้โดยคลิกขวาที่ Toolbox แล้วเลือกเมนู Components ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.7 เพิ่มคอมโพเนนต์ MSComm

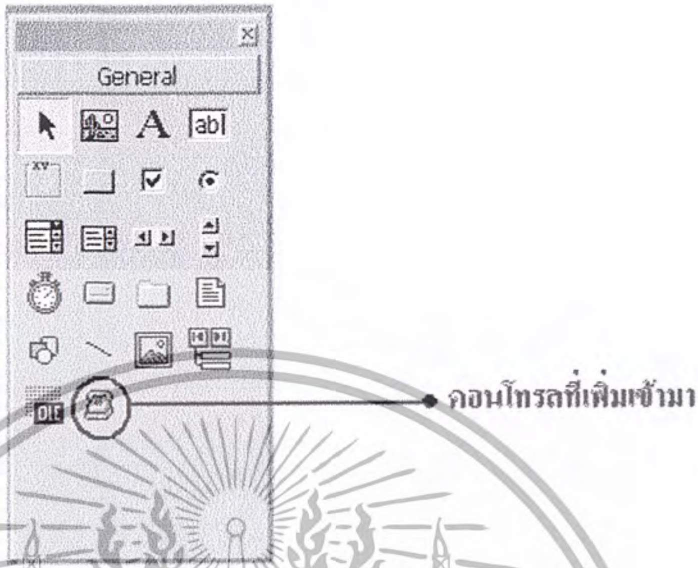
จากนั้นจะปรากฏไดอะล็อก Components ขึ้นมา จากนั้นให้คลิกเลือกที่ Microsoft Comm Control 6.0 แล้วปุ่ม O.K. ดังรูปข้างล่าง



ภาพที่ 5.8 การเลือก MSComm ออกมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นก็จะปรากฏภายใน Toolbox จะมีไอคอนรูปโทรศัพท์ซึ่งเป็นไอคอนของคอนโทรล MSComm ปรากฏขึ้นมาให้เราเลือกใช้งาน



ภาพที่ 5.9 คอนโทรล MSComm พร้อมทำงาน

5.6.1 พร็อพเพอร์ตี้ที่สำคัญในการใช้งาน

- CommPort ใช้ในการกำหนดหมายเลขของพอร์ตอนุกรมที่เราต้องการจะติดต่อ โดยมีรูปแบบของการใช้งานดังตัวอย่างดังนี้

MSComm1.CommPort = 1

- Settings ใช้ในการกำหนดอัตราบอด (Baud Rate) หรือ ความเร็วในการส่งข้อมูลมีหน่วยเป็นบิตต่อวินาที, พาริตี้, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย เช่น Baud Rate = 9600 บิตต่อวินาที ไม่มีพาริตี้ จำนวนข้อมูลเท่ากับ 8 บิตและมีบิตปิดท้าย 1บิต

MSComm1.Settings = " 9600,n,8,1 "

- PortOpen ใช้สำหรับเปิดและปิดการใช้งานพอร์ตอนุกรม โดยมีรูปแบบของการทำงานดังนี้

MSComm1.PortOpen = True (สั่งให้ Port1 ทำการเปิด)

MSComm1.PortOpen = False (สั่งให้ Port1 ทำการปิด)

- InBuffersize เป็นการกำหนดขนาดของ Buffer ในการรับข้อมูลเข้ามาโดยมีรูปแบบการทำงานดังนี้

Object.InBuffrsize [= value]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- OutBuffersize เป็นการกำหนดขนาดของ Buffer ในการส่งข้อมูลออกไป โดยมีรูปแบบการทำงานดังนี้

Object.OutBuffrsize [= value]

- Input ใช้ในการอ่านค่าของ ข้อมูลจากพอร์ตอนุกรม โดยมีรูปแบบในการอ่านค่า ดังนี้ เช่น เราทำการอ่านค่าจากบัฟเฟอร์ของพอร์ตอนุกรม แล้วนำมาเก็บในตัวแปรที่ชื่อว่า Data เราจะเขียนค่าได้ดังนี้

Data = MSComm1.Input

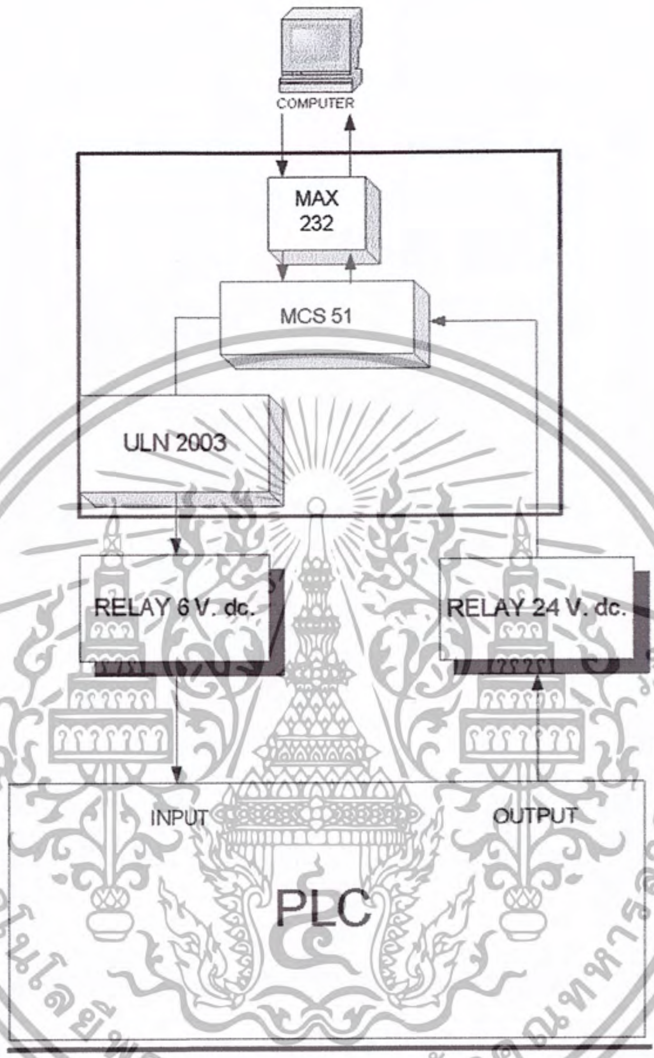
- Output ซึ่งใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบในการเขียนโปรแกรมดังนี้

Object.Output [= value]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

ผลการทดลอง



ภาพที่ 6.1 แสดงภาพรวมของวงจรสื่อสารข้อมูล

6.1 หลักการทำงานของวงจรสื่อสารข้อมูล

เมื่อมีสัญญาณจากคอมพิวเตอร์เข้ามา(เป็นอักขระต่างๆผ่านทาง RS 232 และ ICL232) 8051จะทำการตรวจเช็คว่าสัญญาณที่ได้มีค่าตรงกับรหัส ASCII ตัวใดเมื่อตรวจเช็คแล้วก็จะทำการส่งข้อมูลที่ได้ไปให้ PLC ทำการประมวลผลโดยผ่านRELAY 6V DCและเมื่อ PLC ทำการประมวล-ผลแล้วก็จะส่งสัญญาณออกมาควบคุมอุปกรณ์ต่างๆ(ในที่นี้แทนด้วย RELAY 24 V)เมื่อ PLC ส่งสัญญาณออกมาทำการควบคุมแล้วหลังจากนั้น สัญญาณที่ได้ก็จะถูกต่อให้ 8051 ทำการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

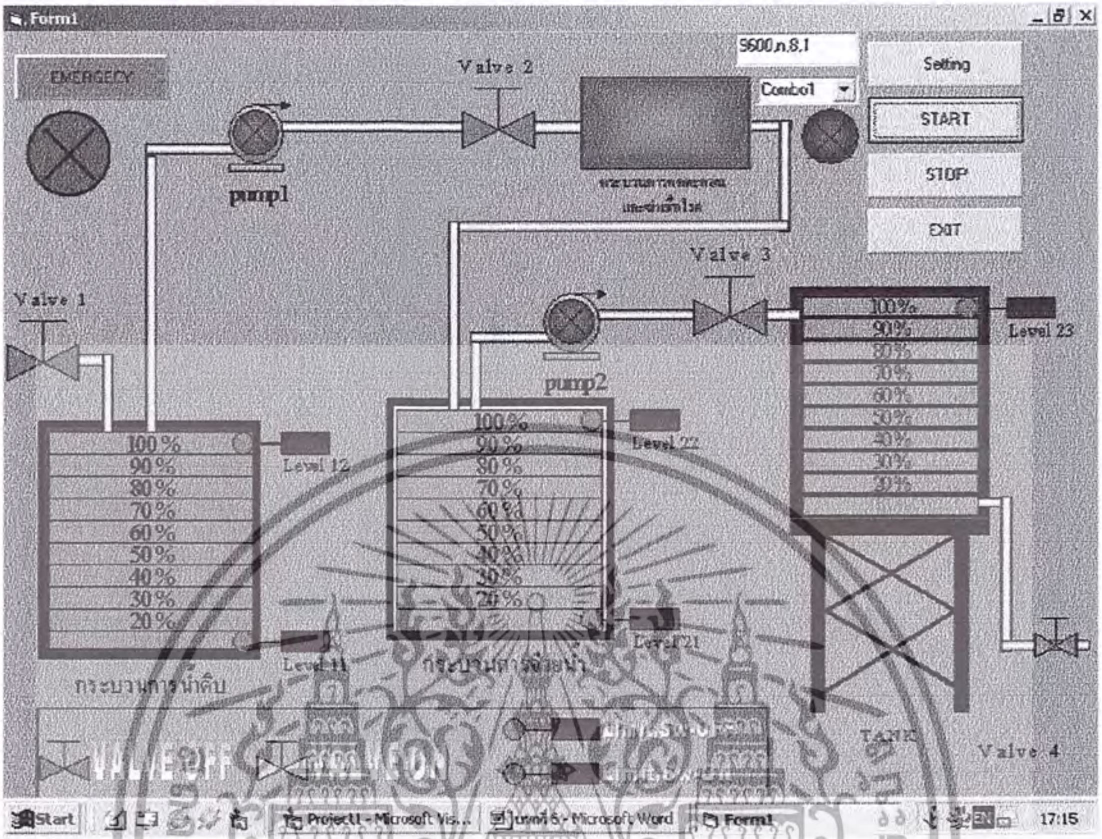
แล้วทำการส่งสัญญาณที่ได้เข้าสู่คอมพิวเตอร์เพื่อทำการควบคุมอุปกรณ์ที่จำลองไว้บนคอมพิวเตอร์ โดยผ่าน RS 232 และ ICL232 ต่อไป.

6.2 อุปกรณ์ที่ใช้ในการทดลอง

1. AT80C51
2. ULN 2003
3. MAX 232CPE
4. RELAY 24 V DC
5. RELAY 6V DC
6. LED
7. DIODE 1N4001
8. RESISTOR 4.7 K,0.33K
9. CAPACITOR 10\50V,33Pf
10. CRYSTRAL 11.0592 MHZ
11. สวิตช์กดติดปล่อยดับ (PB. Switch)
12. DB 9 (CONNECTER ตัวเมีย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 กระบวนการจำลองระบบจ่ายน้ำ



ภาพที่ 6.2 กระบวนการจำลองระบบจ่ายน้ำ

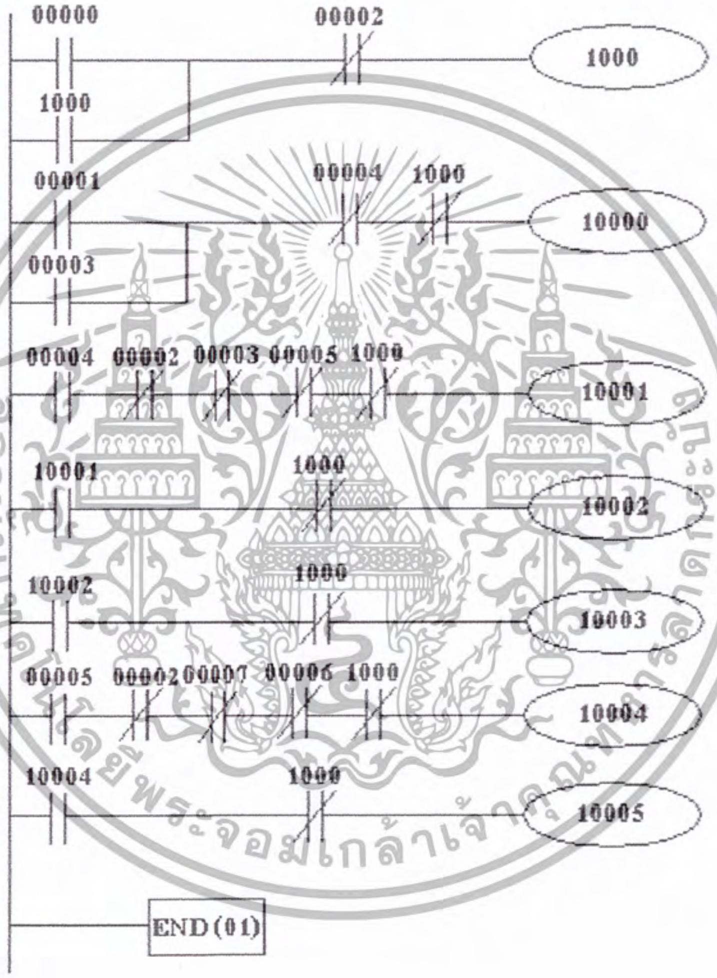
6.3.1 เงื่อนไขการทำงาน

ในการทำงานเมื่อเริ่มแรกจะให้กระบวนการน้ำดิบยังมีน้ำบรรจุอยู่รวมถึงกระบวนการการจ่ายน้ำ และ ปริมาณน้ำที่บรรจุอยู่ในแท็งก์ก็ยังไม่มีปริมาณน้ำอยู่เลย และเมื่อทำการเริ่มกดสวิตซ์ START จะให้ทำการเปิดวาล์ว 1 ก่อนเพื่อทำการเติมน้ำลงไปในถังของกระบวนการน้ำดิบเสียก่อนจนกระทั่งปริมารณน้ำภายในถังเต็มซึ่งจะทำให้ level 12 นั้นทำงานและจะไปการทำงานของวาล์ว 1 พร้อมกับไปสั่งให้ Pump 1 นั้นทำงานด้วย โดยทำการปั๊มน้ำเข้าไปในวาล์วที่ 2 ซึ่งถ้า Pump 1 นั้นยังไม่ทำงานวาลว์2 ก็จะไม่สามารถทำงานเปิดวาล์วได้ ซึ่งจะทำการจ่ายน้ำเข้าไปในกระบวนการตกตะกอนและฆ่าเชื้อโรค โดยที่ถ้าวาลว์2 ยังไม่ทำงานกระบวนการตกตะกอนและฆ่าเชื้อโรค ก็จะยังไม่สามารถทำงานได้เช่นกัน และเมื่อน้ำผ่านกระบวนการตกตะกอนและฆ่าเชื้อโรคแล้วก็จะเป็นน้ำที่สะอาดและจถูกส่งไปที่ ถังกระบวนการจ่ายน้ำทำการเก็บน้ำไว้จนเต็มถังจนถึงระดับ level 21 ก็จะไปสั่งให้ Pump 2 นั้นทำการปั๊มน้ำเข้าวาล์ว 3 โดยที่การทำงานของวาล์ว 3 ก็เช่นเดียวกับวาล์ว 2 ซึ่งก็จะทำการจ่ายน้ำเข้าไปเก็บไว้ใน TANK เพื่อทำการเก็บน้ำไว้พร้อมที่จะจ่ายออกไปทางวาล์ว 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ level 23 จะทำหน้าที่คอยตัดการทำงานกระบวนการทั้งหมดถ้า level 23 นั้นทำงาน และ level 22 ก็จะทำหน้าที่เป็นตัวตัดการทำงาน Pump 2 และสั่งให้ Pump1 ทำงานปั๊มน้ำเข้าไปในกระบวนการตกตะกอนและฆ่าเชื้อโรคต่อไป ส่วนใน level 11 จะเป็นตัวเซ็นเซอร์คอยตัดการทำงานของ Pump1 เพื่อไม่ให้ปั๊มโดยไม่มีน้ำภายในถังเพื่อเป็นการประหยัดพลังงานและยืดอายุการใช้งานของปั๊ม

6.3.2 LADDER DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.3 กำหนด Input และ Output

- **Input**

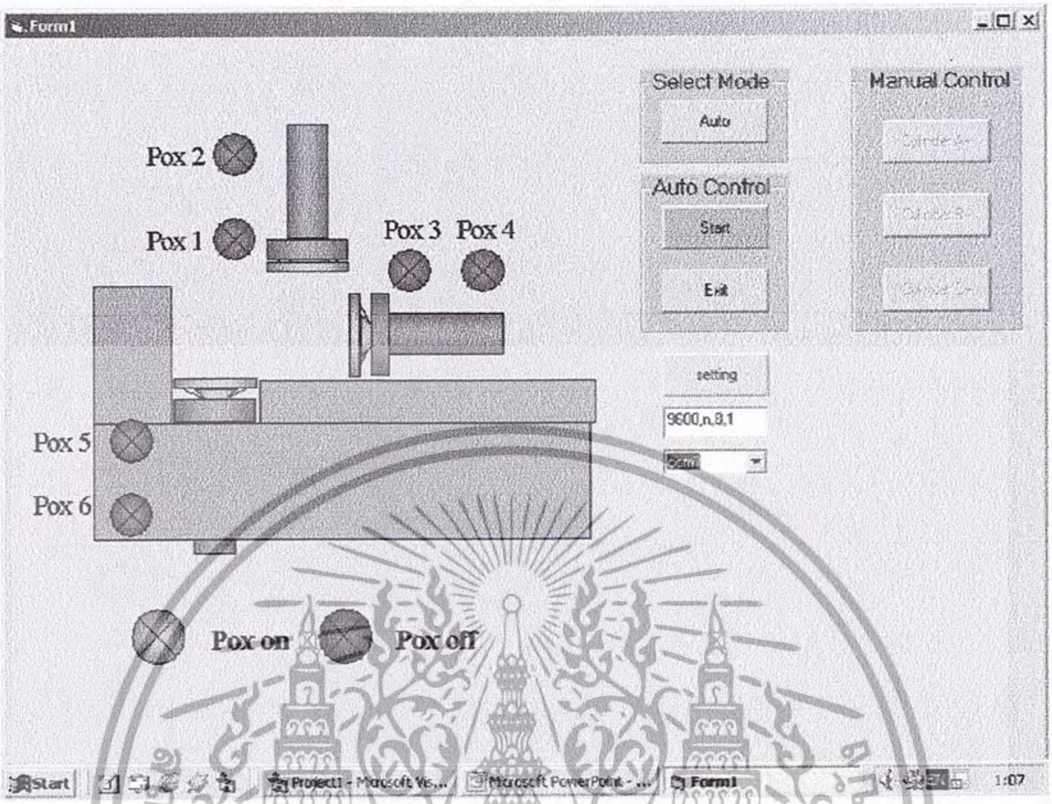
00000	=	Emergency Switch
00001	=	Start
00002	=	Stop
00003	=	Level sensor 11
00004	=	Level sensor 12
00005	=	Level sensor 21
00006	=	Level sensor 22
00007	=	Level sensor 23

- **Output**

10000	=	Valve 1
10001	=	Pump 1
10002	=	Valve 2
10003	=	ตกตะกอนและฆ่าเชื้อ
10004	=	Pump2
10005	=	Valve 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 กระบวนการจำลองการทำงานลูกสูบ

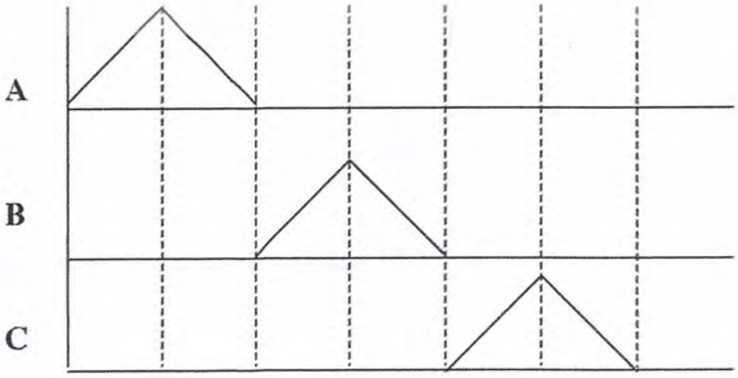


ภาพที่ 6.3 กระบวนการจำลองการทำงานของลูกสูบ

6.4.1 เงื่อนไขการทำงาน

กำหนดให้เมื่อทำการกดสวิตช์ START การทำงานของลูกสูบ A B และ C ตาม MOTIONSTEP DIAGRAM ดังนี้.

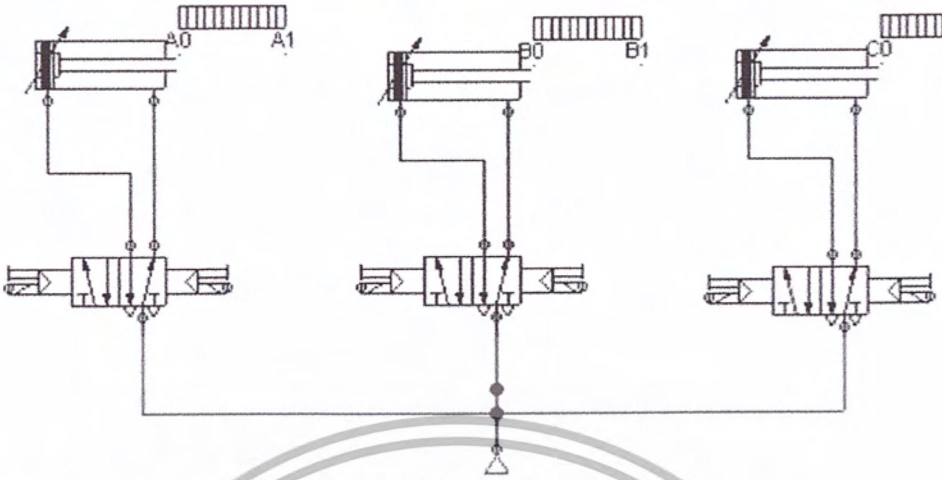
MOTIONSTEP DIAGRAM



สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

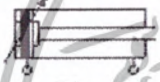
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

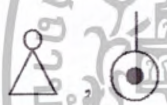
6.4.2 วงจรนิวแมติกส์ไฟฟ้า



สัญลักษณ์ในการเขียนแบบ

1. วาล์วควบคุมแบบ 5/2 ชนิดมีขดลวดโซลินอยด์

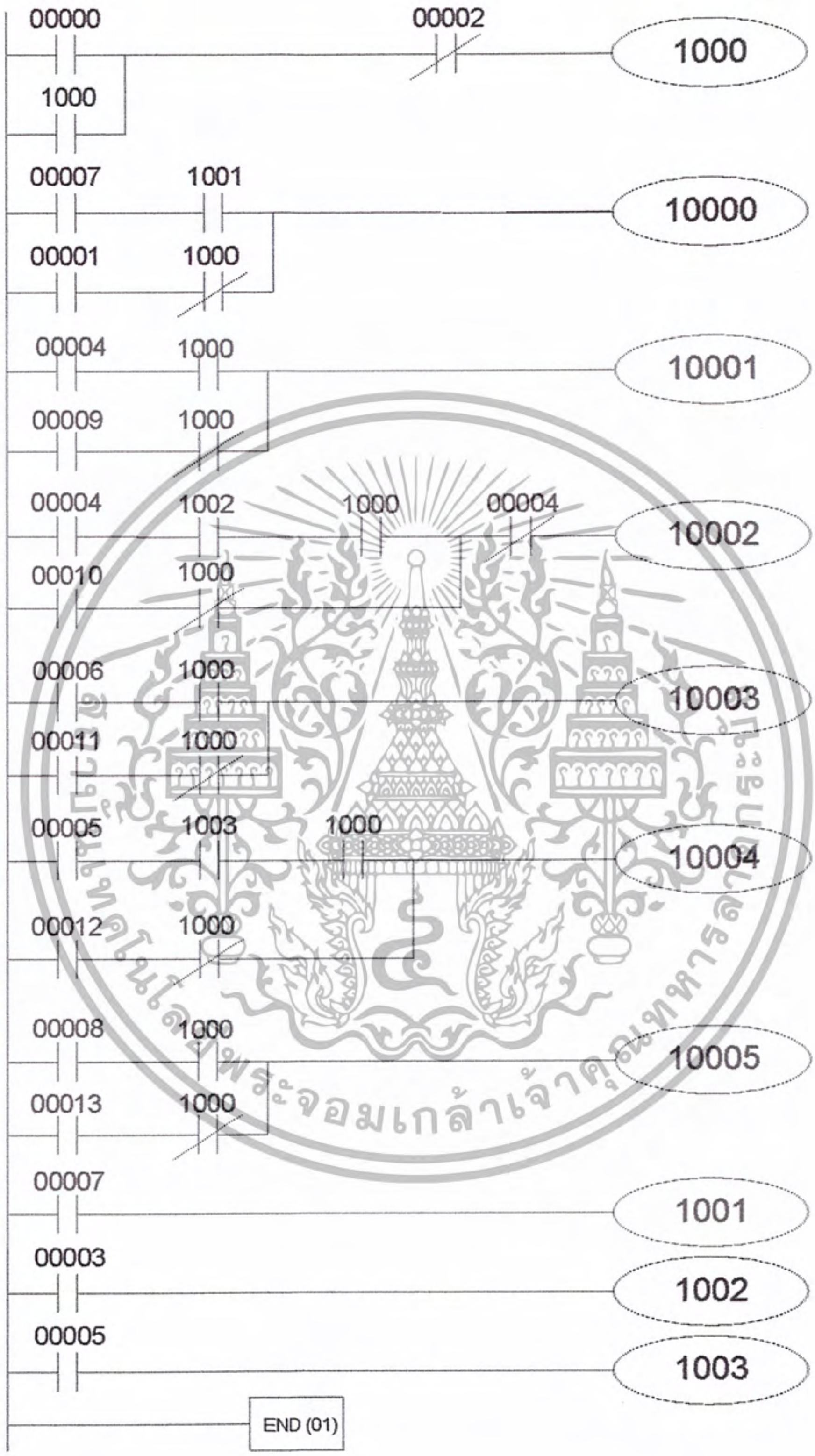
2.  กระบอกสูบชนิดสองทิศทาง

3.  แหล่งจ่ายลมให้กับระบบ

4. ——— เส้นทางเดินลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.4 LADDER DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.5 กำหนดค่า Input และ Output

- **Input**

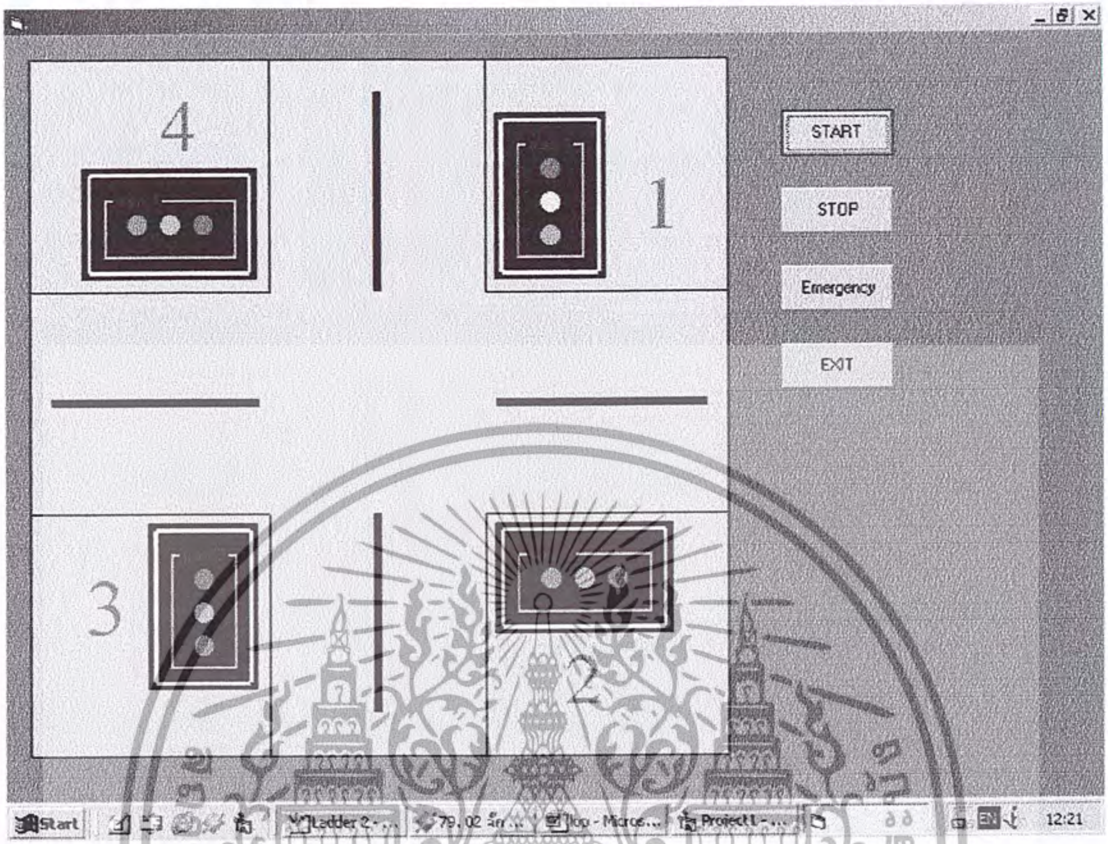
00000	=	Auto. Sw.
00001	=	Start (Man.Sw. A+)
00002	=	Stop
00003	=	pox1
00004	=	pox2
00005	=	pox3
00006	=	pox4
00007	=	pox5
00008	=	pox6
00009	=	Man.Sw. A-
00010	=	Man.Sw. B+
00011	=	Man.Sw. B-
00012	=	Man.Sw. C+
00013	=	Man.Sw. C-

- **Output**

10000	=	D1
10001	=	D2
10002	=	D3
10003	=	D4
10004	=	D5
10005	=	D6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 การจำลองการทำงานของสัญญาณไฟจราจร



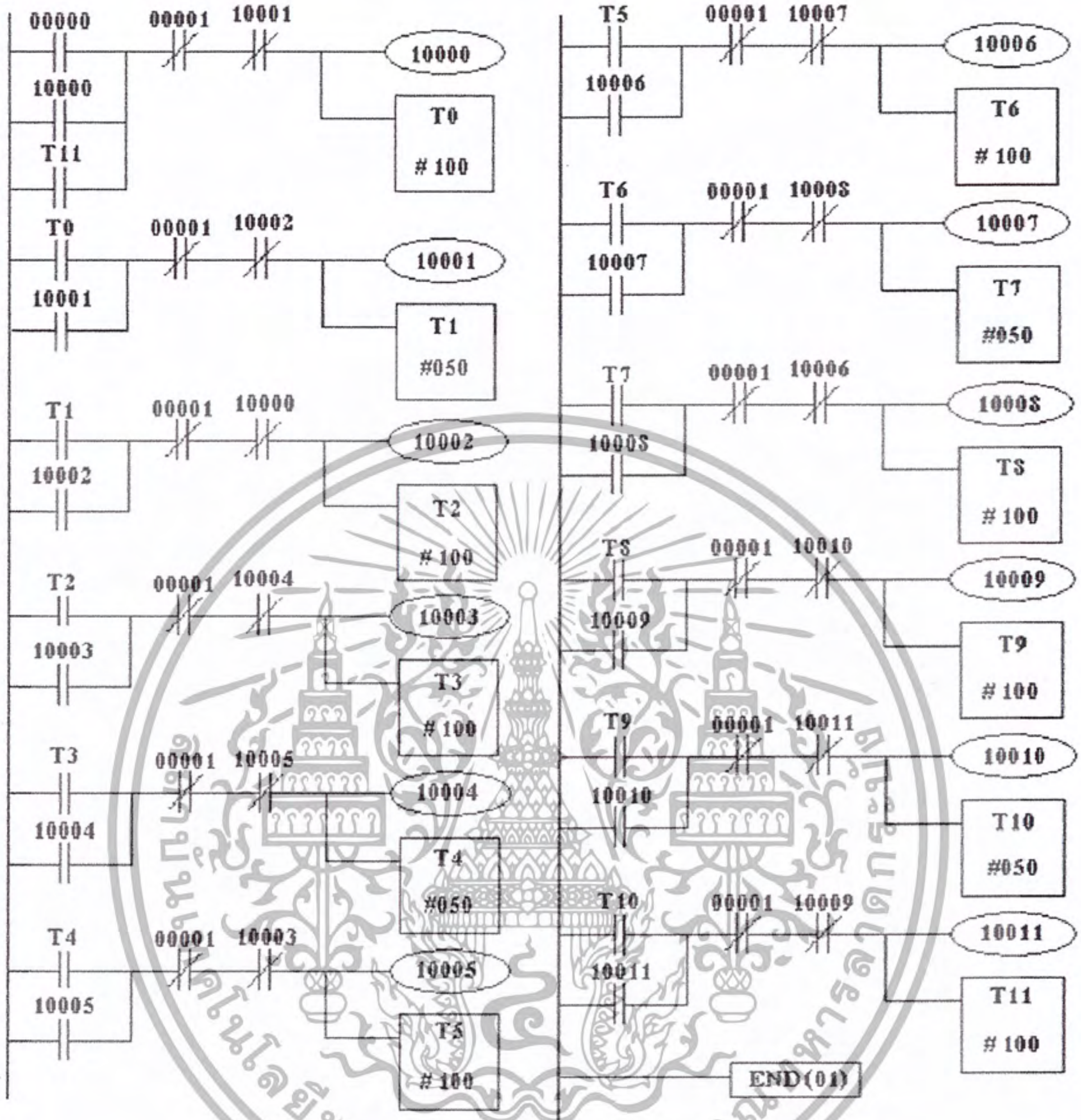
ภาพที่ 6.4 การจำลองกระบวนการทำงานของสัญญาณไฟจราจร

6.5.1 เงื่อนไขการทำงาน

เมื่อทำการเริ่มกดสวิตช์ START แล้วไฟเขียวแยกที่1 (green #1) จะเริ่มติดสว่างก่อนเป็นอันดับแรกก่อน เป็นเวลานาน 1 นาที และสัญญาณไฟแดงในแยกที่2 3 และ 4 นั้นก็จะติดสว่างด้วย จากนั้นไฟเหลืองในแยกที่1นี้ (yellow #1) ก็จะเริ่มทำงานสว่างต่อจากไฟเขียวแรก เป็นเวลานาน 30 วินาที หลังจากนั้นไฟแดง (red #1) ในแยกนี้ก็จะสว่างและเป็นผลให้ไฟเขียว (green #2) ในแยกที่ 2 นี้ติดสว่างขึ้นมาเป็นเวลานานเช่นเดียวกับไฟเขียวในแยกแรก และพอหลังจากนั้นไฟเหลือง (yellow #2) และ ไฟแดง (red #2) ก็จะการทำงานไล่กันตามลำดับเช่นเดียวกับการทำงานในแยกแรก และในสัญญาณไฟในแยกที่ 3 และ 4 การทำงานลำดับของสัญญาณไฟเขียว เหลือง และแดง ก็เช่นเดียวกัน และการทำงานก็จะวนอย่างนี้ไปเรื่อย ๆ จนกว่าจะทำการกดสวิตช์ STOP การทำงานของสัญญาณไฟทั้งหมดก็จะดับทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.2 LADDER DAIGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.3 กำหนดค่า Input และ Output

- **Input**

00000 = START

00001 = STOP

00003 = Emergency

- **Output**

10000 = green # 1

10001 = yellow # 1

10003 = red # 1

10004 = green # 2

10005 = yellow # 2

10006 = red # 2

10007 = green # 3

10008 = yellow # 3

10009 = red # 3

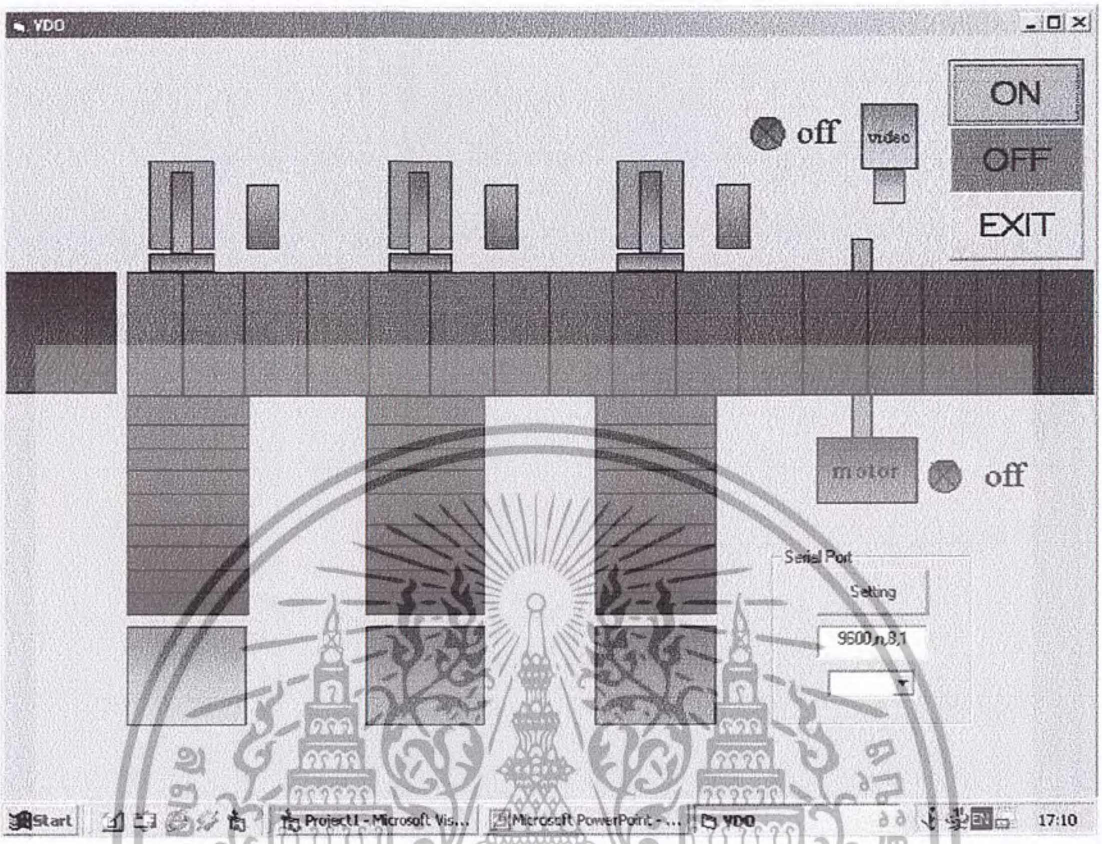
10010 = green # 4

10011 = yellow # 4

10012 = red # 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 กระบวนการคัดแยกด้วยกล้องวิดีโอ



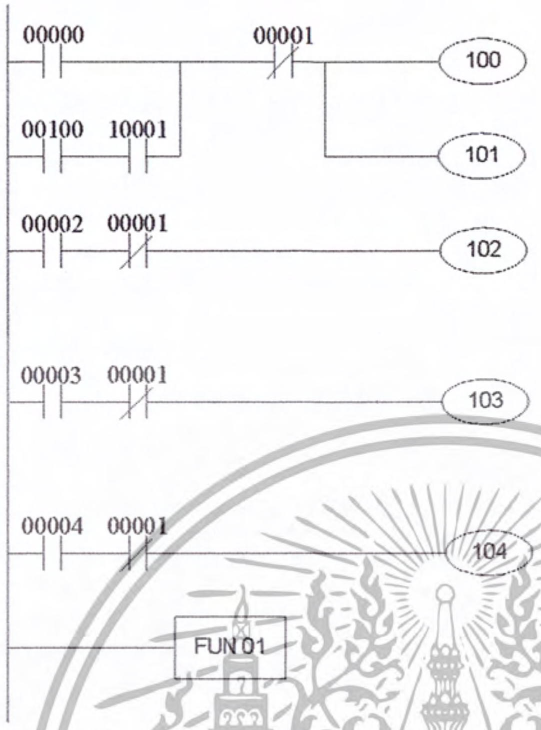
ภาพที่ 6.5 กระบวนการคัดแยกด้วยกล้องวิดีโอ

6.6.1 เงื่อนไขการทำงาน

การทำงานจะกำหนดให้การคัดแยกนั้นจะใช้กล้องวีดีโอเป็นตัวเซ็นเซอร์จับสีของวัตถุ เมื่อถ้าวัตถุที่มาตามสายพานนั้นมีสีตรงตามที่โปรแกรมที่ตั้งไว้ ลูกสูบก็จะทำการคัดแยกวัตถุนั้นตามสีที่โปรแกรมไว้ต่อไปตามลำดับ และเมื่อมีวัตถุชิ้นใหม่เข้ามาก็จะทำการจับสีของวัตถุ และลูกสูบก็จะทำการคัดแยกวัตถุเช่นนี้ไปเรื่อยๆ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6.2 LADDER DIAGRAM

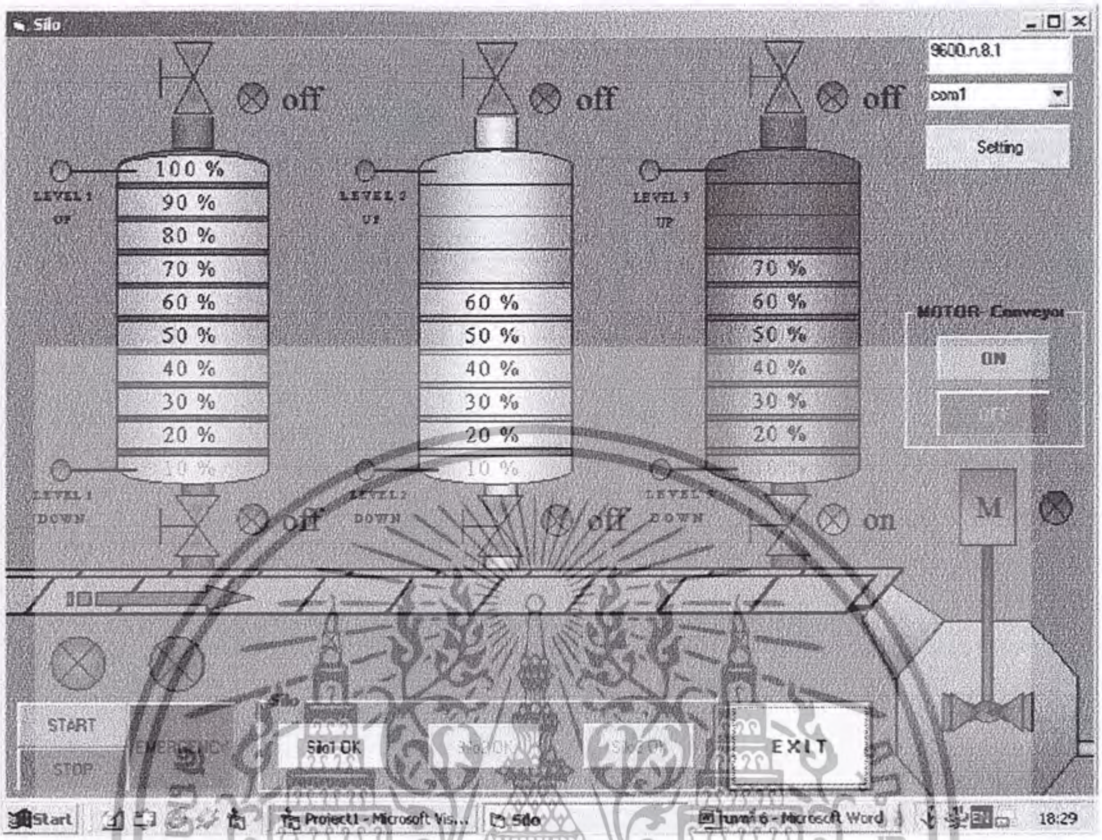


6.6.3 กำหนดค่า Input และ Output

- **Input**			- **Output**		
00000	=	Start	10000	=	motor
00001	=	Stop	10001	=	video
00002	=	red	10002	=	cylinder 1
00003	=	pink	10003	=	cylinder 2
00004	=	blue	10004	=	cylinder 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 กระบวนการจำลองการผสมสารจากไซโล



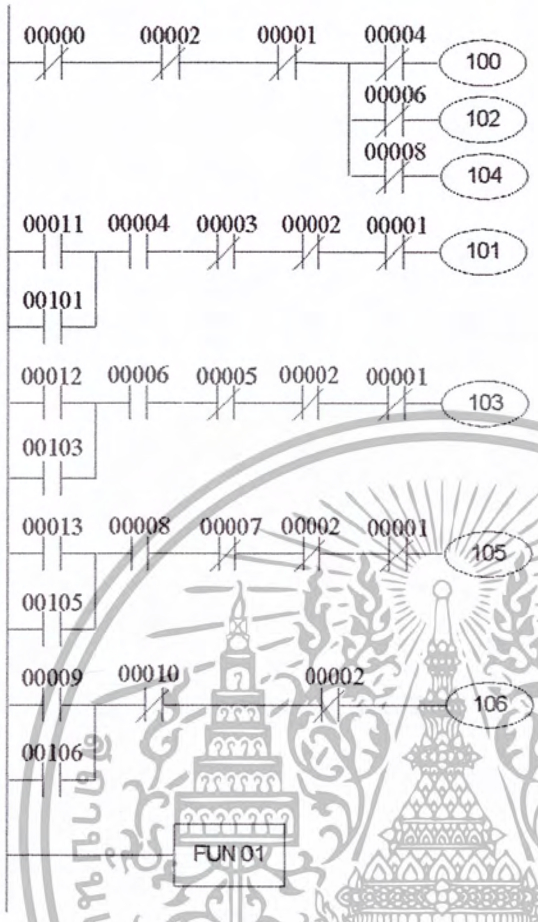
ภาพที่ 6.6 กระบวนการจำลองการผสมสารจากไซโล

6.7.1 เงื่อนไขการทำงาน

เมื่อเริ่มต้นการทำงานในตอนแรกของการทำงานนั้นไซโลทั้ง 3 ยังว่างเปล่าอยู่ ดังนั้นเมื่อทำการกดสวิตซ์ START จะต้องทำให้ให้วาล์วทั้ง 3 ตัว ทำการเปิดวาล์วทั้ง 3 พร้อมกันเพื่อให้สารทั้ง 3 ชนิดไหลลงมาในบรรจุเก็บอยู่ในไซโล จนเต็มจะทำให้ตัว Level Sw. ที่อยู่ด้านบนของถังไซโลทำงาน โดลการเปลี่ยนจากสีแดงเป็นสีเขียว จึงจะสามารถทำให้ กดสวิตซ์ สั่งให้วาล์วด้านล่างของไซโลแต่ละถังนั้นสามารถเปิดได้ (ปุ่ม Silo OK) และก็จะไปสั่งให้วาล์วด้านบนนั้นปิดลงเพื่อไม่ให้สารนั้นล้นออกมา ส่วนการทำงานของสายพานและมอเตอร์นั้นจะแยกจากการควบคุมของไซโล โดยมีหน้าที่ทำการเลื่อนลำเลียงเพื่อที่จะส่งสารที่ไหลลงมาเข้าไปในภาชนะ ที่จะทำการผสมสารทั้ง 3 ชนิดให้เข้ากัน และมอเตอร์ก็จะทำการขับใบพัดเพื่อทำการผสมสารทั้ง 3 ชนิดให้เข้ากัน โดยถ้าเมื่อสารที่อยู่ภายในไซโลได้ไหลลงมาหมดแล้วก็จะทำให้ Level Sw. ด้านล่างของถังจะเป็นตัวที่ไปตัดการทำงานของวาล์วให้ทำการปิดวาล์วลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7.2 LADDER DIAGRAM



6.7.3 กำหนดค่า Input และ Output

- **Input**

00000	=	Sw.Start	00011	=	open Silo 1
00001	=	Sw.Stop	00012	=	open Silo 2
00002	=	Emergency	00013	=	open Silo 3
00003	=	Level Down 1			
00004	=	Level Up 1			
00005	=	Level Down 2			
00006	=	Level Up 2			
00007	=	Level Down 3			
00008	=	Level Up 3			
00009	=	ON Motor			
00010	=	OFF Motor			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Output**

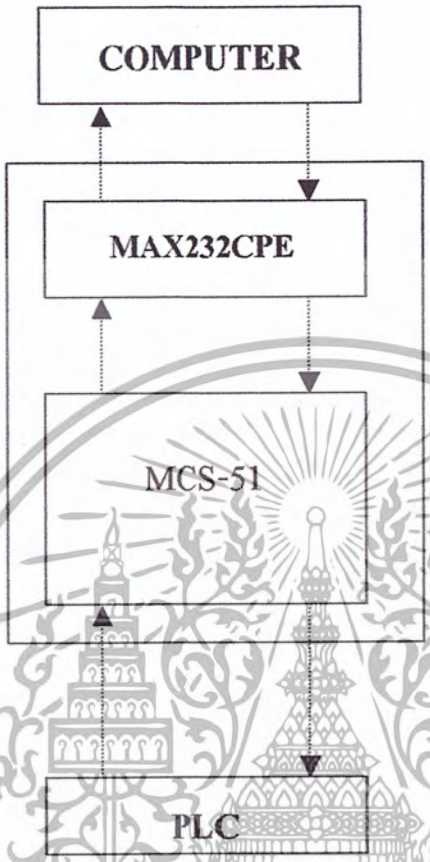
00100	=	Valve Down 1
00101	=	Valve Up 1
00102	=	Valve Down 2
00103	=	Valve Up 2
00104	=	Valve Down 3
00105	=	Valve Up 3
00106	=	Motor
00107	=	Emergency Lamp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8 รายละเอียดของวงจรที่ใช้ในการทดลอง

6.8.1 ส่วนที่ใช้ในการควบคุมการติดต่อสื่อสารข้อมูล

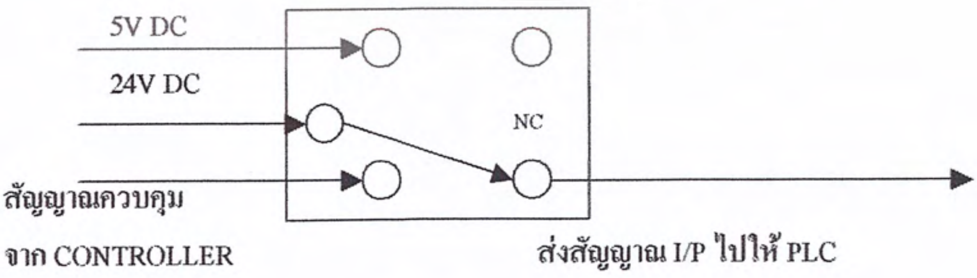
จะมีรายละเอียดต่างๆดังต่อไปนี้ (รายละเอียดของวงจรจริงอยู่ในภาคผนวก)



ภาพที่ 6.7 แสดงบล็อกไดอะแกรมของวงจรในส่วนที่ใช้ในการควบคุมการติดต่อสื่อสารข้อมูล

6.8.2 ส่วนที่ใช้ในการส่งสัญญาณออกไปควบคุม PLC

จะมีรายละเอียดต่างๆดังต่อไปนี้ (รายละเอียดของวงจรจริงอยู่ในภาคผนวก)

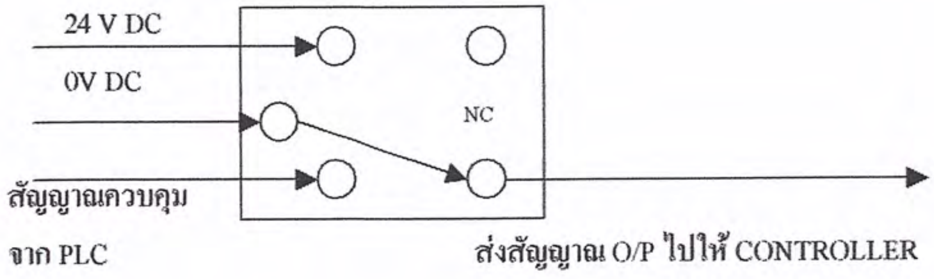


ภาพที่ 6.8 แสดงส่วนของวงจรที่ใช้ในการส่งสัญญาณออกไปควบคุม PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8.3 ส่วนที่ใช้ในการรับสัญญาณจาก PLC

จะมีรายละเอียดต่างๆดังต่อไปนี้ (รายละเอียดของวงจรจริงอยู่ในภาคผนวก)



ภาพที่ 6.9 แสดงส่วนของวงจรที่ใช้ในการรับสัญญาณจาก PLC

6.9 สรุปผลการทดลอง

จากการปฏิบัติและทำการทดลองที่ผ่านมานั้นในส่วนของอุปกรณ์ในการส่งผ่านข้อมูลที่ได้ดำเนินการออกแบบและสร้างขึ้นมาเพื่อใช้ในการเชื่อมต่อข้อมูลระหว่าง PLC และแบบจำลองบนคอมพิวเตอร์ซึ่งหลังจากการดำเนินการออกแบบและสร้างขึ้นมานั้นสามารถติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์และอุปกรณ์ส่งผ่านข้อมูล ได้จริงแต่ยังมีปัญหาในส่วนของ โปรแกรมที่ได้ทำการออกแบบ ไว้ ซึ่งในส่วนของเอาท์พุทนั้นจะสามารถติดต่อ ได้เพียง8บิทข้อมูลเท่านั้นเนื่องจากข้อจำกัดของตัว ไมโครคอน โทรลเลอร์เองและการเขียน โปรแกรมบนคอมพิวเตอร์ซึ่งเมื่อเกิน 8 บิทข้อมูลก็จะเกิดการซ้ำซ้อนของข้อมูลบนเครื่องคอมพิวเตอร์ ได้เป็นผลทำให้การจำลองกระบวนการไม่เป็นไปอย่างที่ได้ตั้งใจไว้

6.10 ประโยชน์ที่คาดว่าจะได้รับ

จากการที่ได้ทำการศึกษาและปฏิบัติการทดลองที่ผ่านมานั้นนั้นประโยชน์ที่ผู้จัดทำคาดว่าจะได้รับมีดังนี้

1. มีความรู้และเพิ่มทักษะในการออกแบบวงจรรวมทั้งการเลือกใช้อุปกรณ์อิเล็กทรอนิกส์ต่างๆให้มีความเหมาะสมกับวงจร
2. เพิ่มทักษะในการเขียน โปรแกรมบนไม โครคอน โทรลเลอร์ให้เพิ่มมากขึ้น
3. มีความรู้ในการ ใช้อุปกรณ์ติดต่อสื่อสารข้อมูลจาก โครงงานคือ RS232
4. มีความรู้ในการเขียนโปรแกรมวิชวลเบสิกในรูปแบบกราฟฟิก
5. มีความรู้และทักษะในการ ใช้เครื่องควบคุม PLC เพิ่มมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.11 ปัญหาและอุปสรรค

6.11.1 เนื่องจากอุปกรณ์ที่ใช้บางอย่างมีราคาค่อนข้างสูงและทางสโตร์ของทางภาควิชามีอุปกรณ์ที่จะให้นักศึกษาได้ขอยืมใช้ไม่เพียงพอทำให้เกิดความล่าช้าในการทดสอบและประกอบวงจร

6.11.2 เนื่องจากเป็นงานที่ไม่เคยทำมาก่อนการทำงานจึงเป็นไปได้ล่าช้าเนื่องจากจะต้องทำการศึกษาค้นคว้าข้อมูลในเรื่องของการออกแบบวงจรรวมทั้งในส่วนที่จะนำมาใช้ในการเชื่อมต่อซึ่งก็คือ PLC และโปรแกรมบนระบบคอมพิวเตอร์และวงจรที่ใช้เกิดความเสียหายไม่สามารถใช้งานได้ทำให้สิ้นเปลืองเงินทุนในการซื้ออุปกรณ์ที่มีจำกัด ทำให้การทำงานเป็นไปอย่างล่าช้าและไม่เป็นไปตามเป้าหมาย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

พัฒนพงศ์ พันธยะติวงศ์, การเขียนโปรแกรมคอมพิวเตอร์ด้วย Visual Basic, กรุงเทพมหานคร, ซอฟท์เพรส, 2545, 264 หน้า.

ชัยวัฒน์ ลิ้มพรจิตรวิไล, เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์MCS-51, กรุงเทพมหานคร, อินโนเวตีฟ เอ็กเพอริเมนต์.

สมศักดิ์ ศรีขจรเกียรติ, เรียน Visual Basic 6 ด้วยตนเอง, กรุงเทพมหานคร, บิบลิโอไฟล์, 2541.

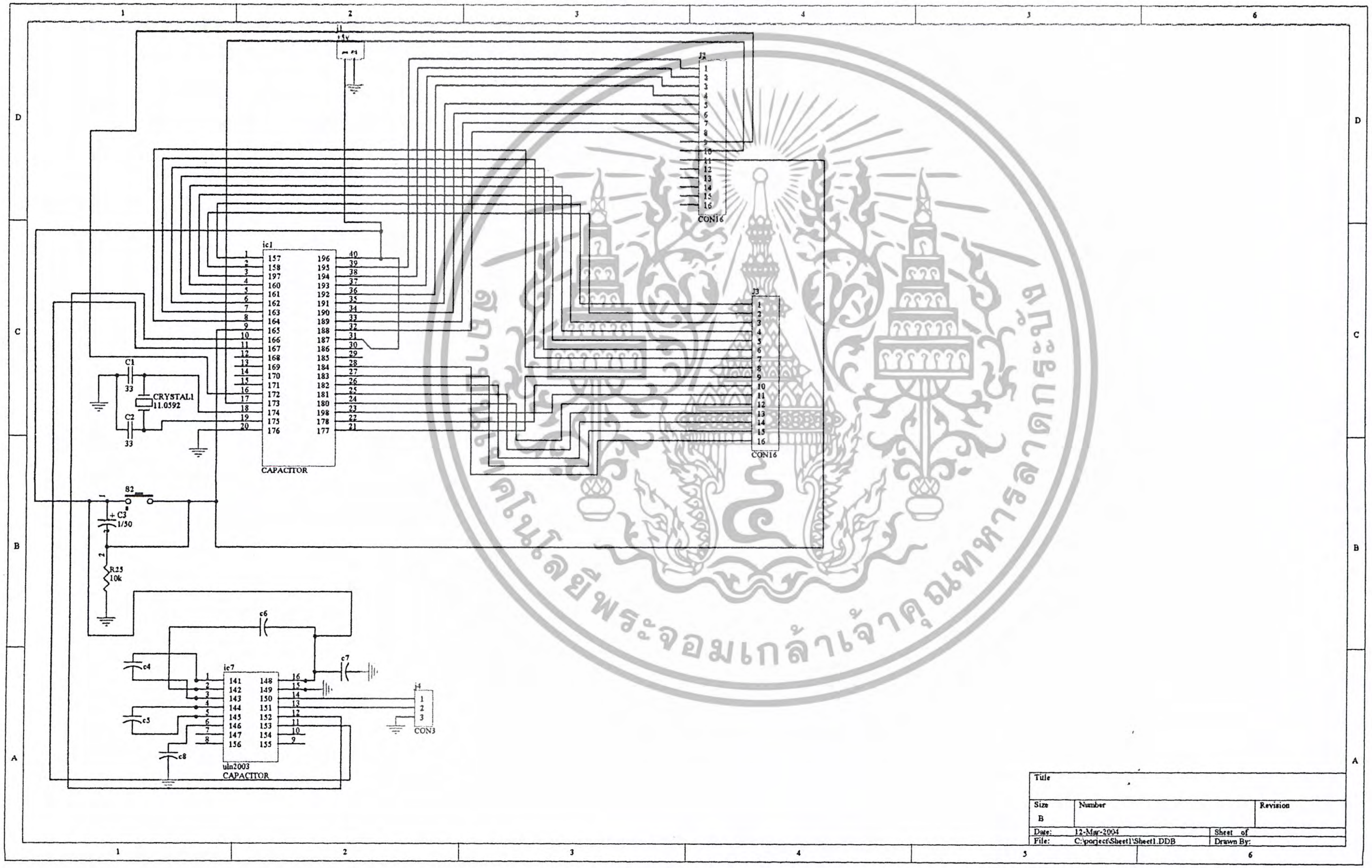
ธีรศิลป์ ทุมวิภาต, เรียนรู้ PLC ขั้นตนด้วยตนเอง, กรุงเทพมหานคร, ซีเอ็ดยูเคชั่น, 2545.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

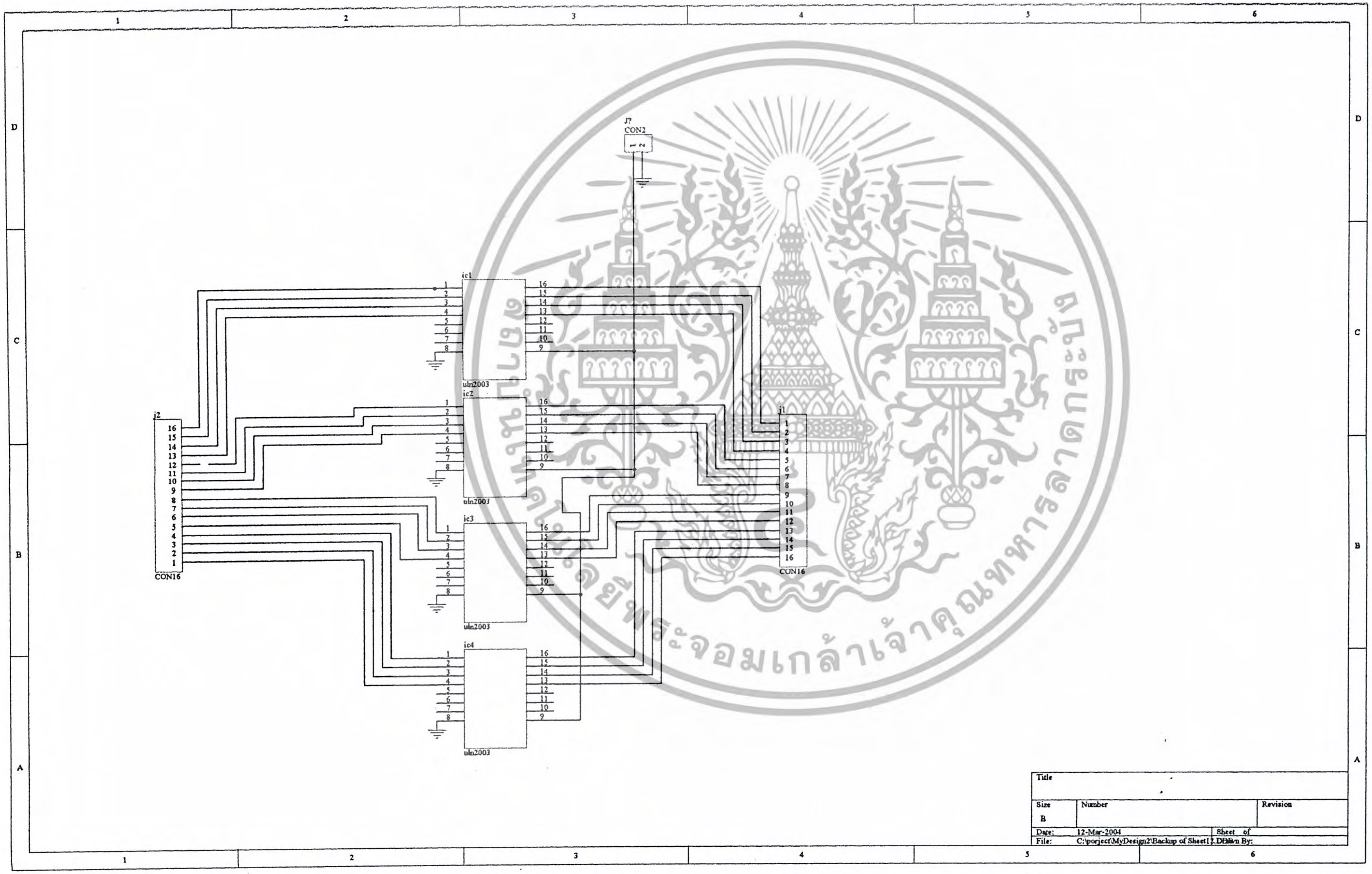
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

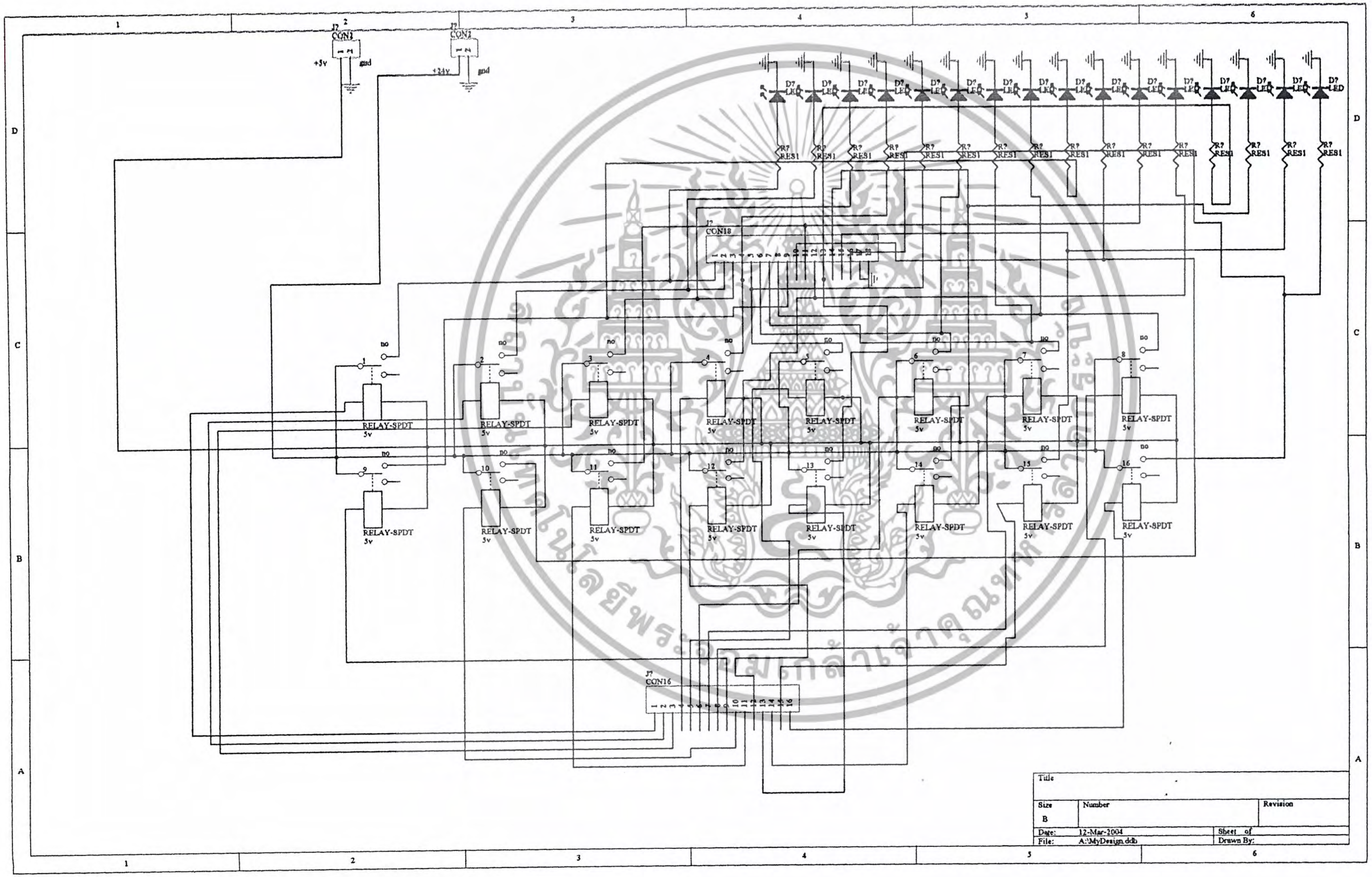
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



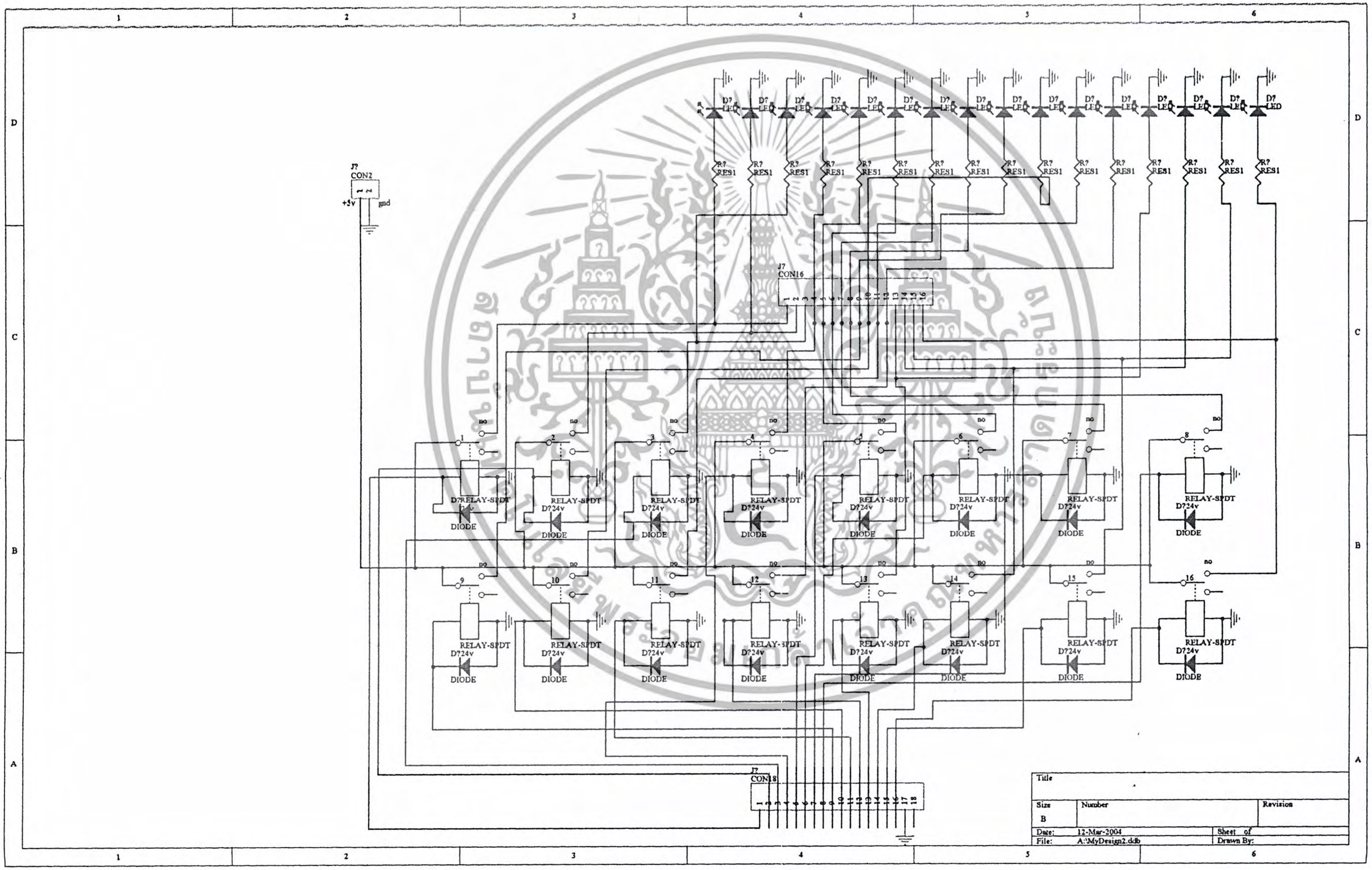
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



J?
CON1
+5v
gnd
CON2
+24v
D?
LED
R?
RES1
CON18
RELAY-SPDT
5v
no
CON16
Title
Size
B
Number
Revision
Date: 12-Mar-2004
File: A:\MyDesign.ddb
Sheet of
Drawn By:



J?
CON2
+5v
gnd
D?
LED
R?
RES1
J?
CON16
RELAY-SPDT
D?24v
DIODE
no
J?
CON18
Title
Size
B
Number
Revision
Date: 12-Mar-2004
File: A:\MyDesign2.ddb
Sheet of
Drawn By:

Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 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

8-bit Microcontroller with 4K Bytes Flash

AT89C51

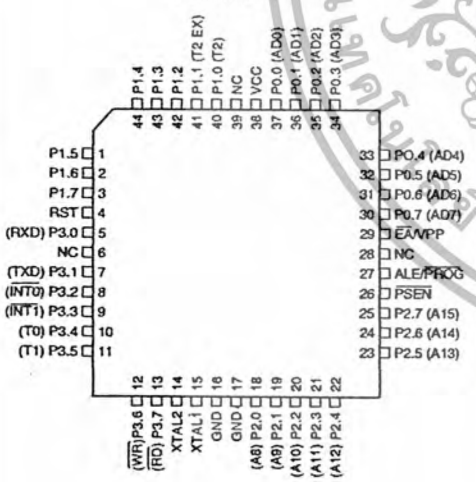
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 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.

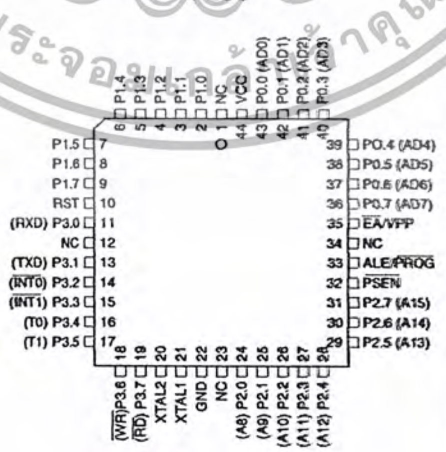
Pin Configurations

PDIP

PQFP/TQFP



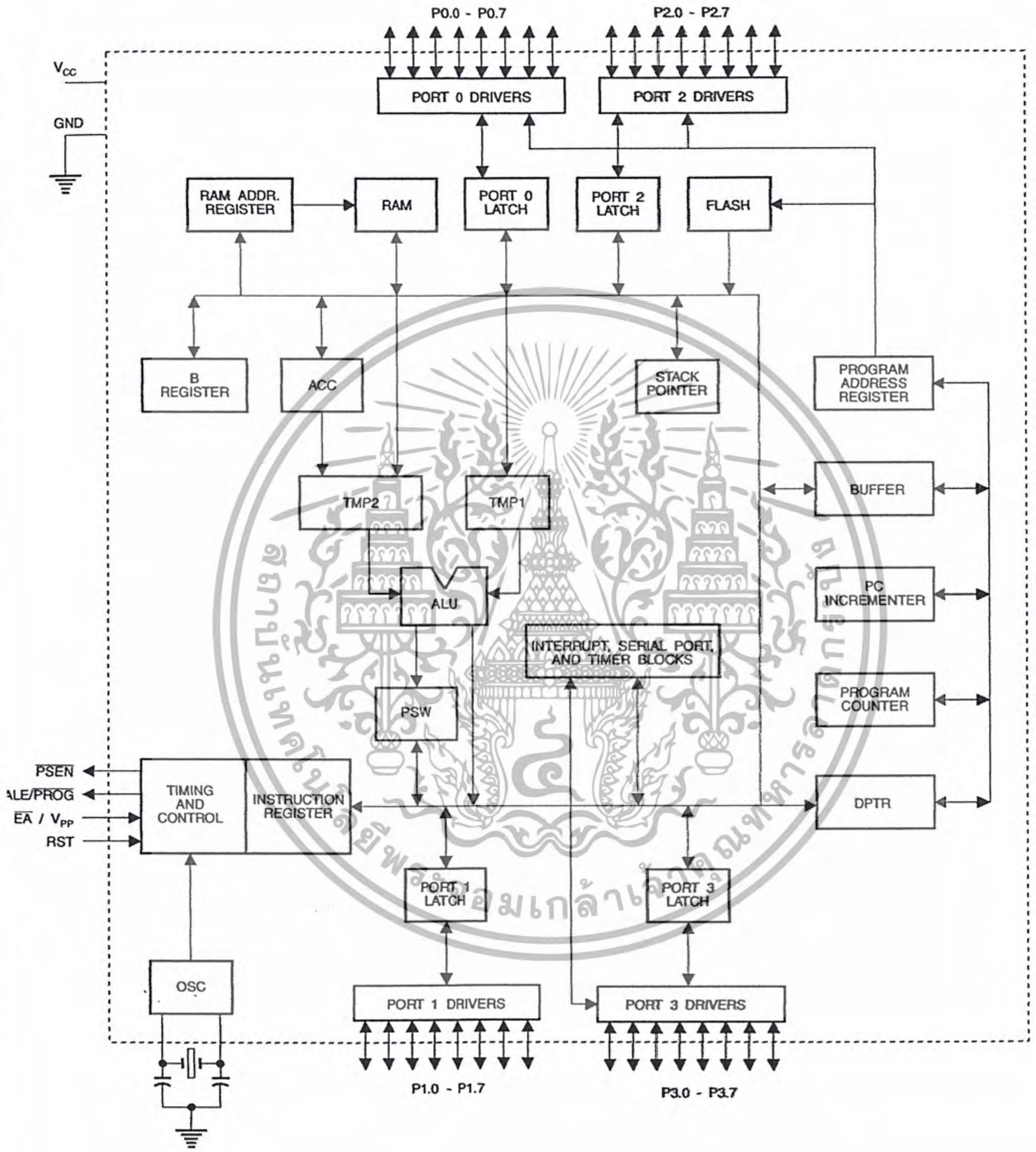
PLCC



Rev. 0265G–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes 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

VCC

Supply voltage.

GND

Ground.

Port 0

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

Port 0 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 bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

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

Port 2

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

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, 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 bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

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

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

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

RST

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pulse is skipped during each access to external Data Memory.

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

$\overline{\text{PSEN}}$

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

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

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

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

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

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

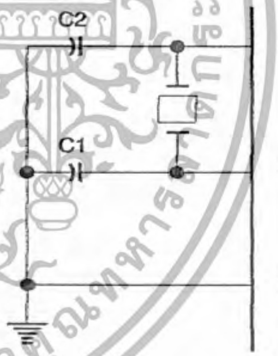
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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 hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle 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.

Figure 1. Oscillator Connections



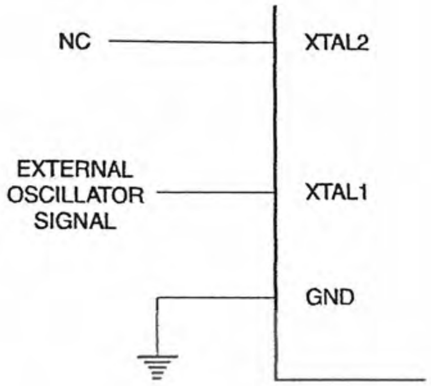
Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 2. External Clock Drive Configuration



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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} = 12V	V_{PP} = 5V
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 Figure 3 and Figure 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 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{\text{Data}}$ Polling: The AT89C51 features $\overline{\text{Data}}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

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

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

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "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 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming Modes

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

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

Figure 3. Programming the Flash

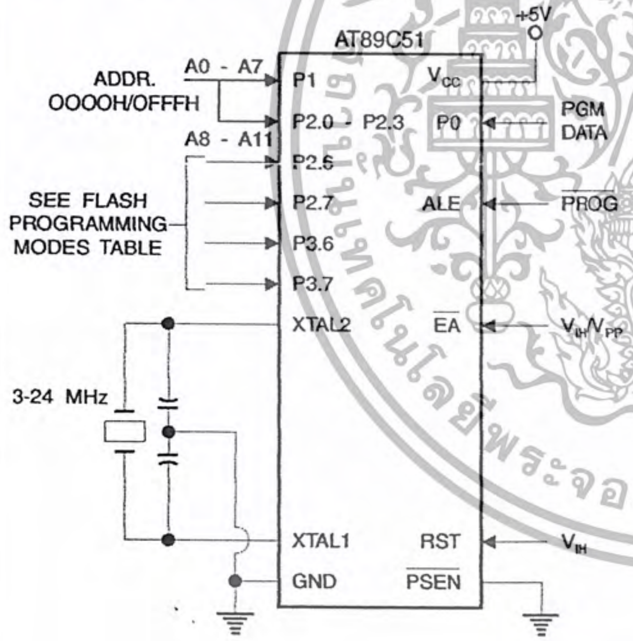
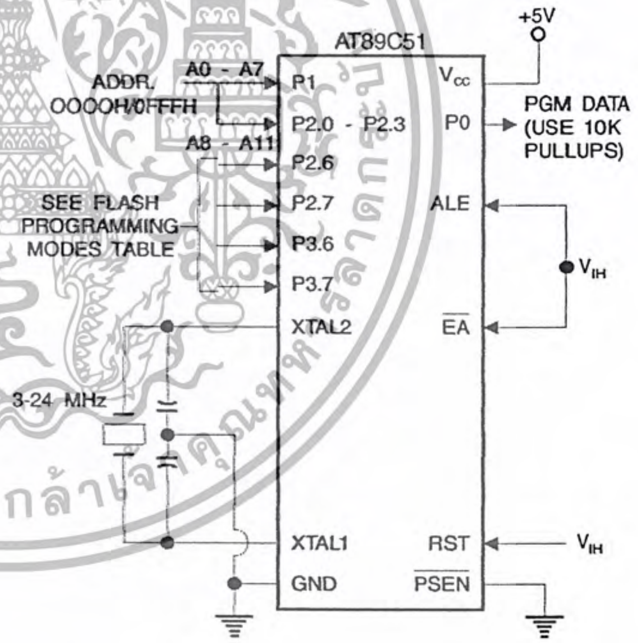
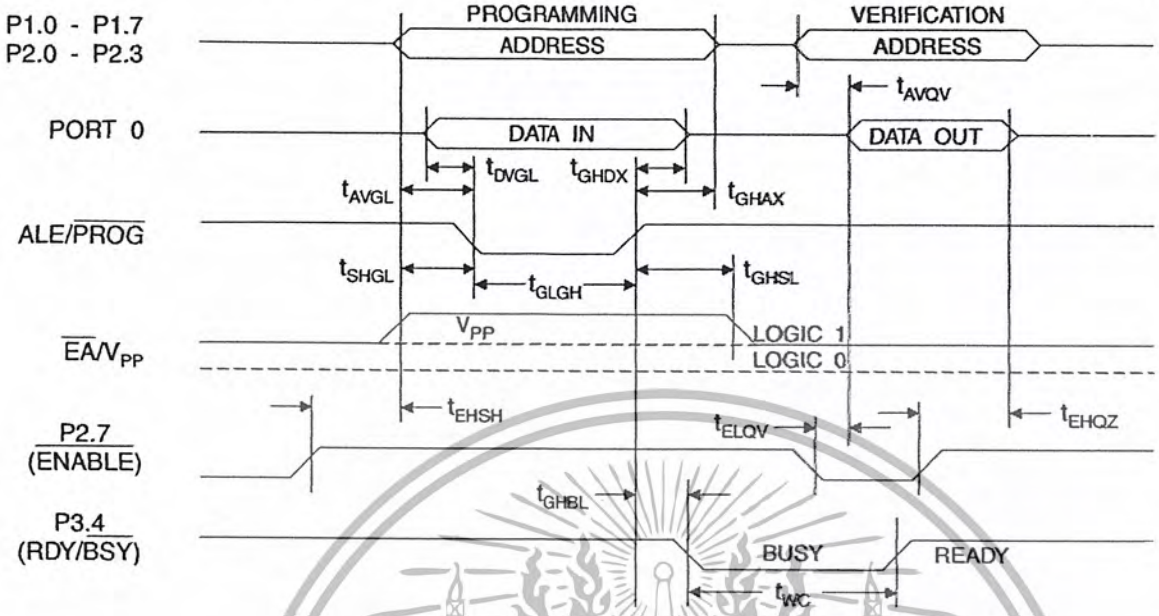


Figure 4. Verifying the Flash

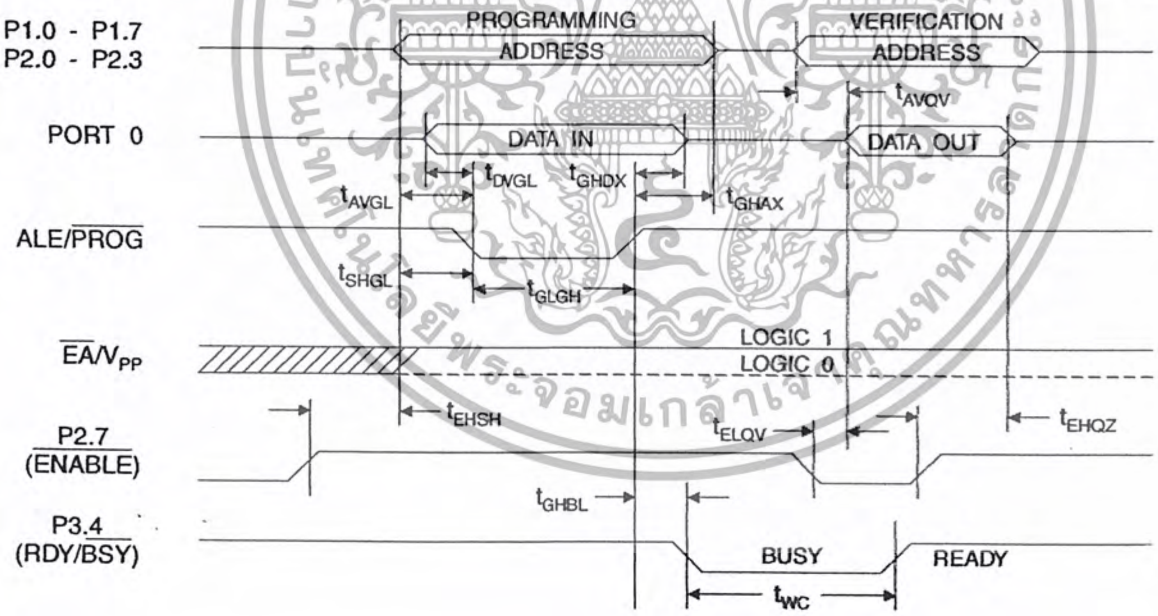


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Characteristics

T_A = 0°C to 70°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	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
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		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to BUSY Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms

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

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

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

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

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 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 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pull-down 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} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HARRIS SEMICONDUCTOR

ICL232

+5V Powered Dual RS-232 Transmitter/Receiver

December 1993

Features

- Meets All RS-232C Specifications
- Requires Only Single +5V Power Supply
- Onboard Voltage Doubler/Inverter
- Low Power Consumption
- 2 Drivers
  - ±9V Output Swing for +5V Input
  - 300Ω Power-off Source Impedance
  - Output Current Limiting
  - TTL/CMOS Compatible
  - 30V/μs Maximum Slew Rate
- 2 Receivers
  - ±30V Input Voltage Range
  - 3kΩ to 7kΩ Input Impedance
  - 0.5V Hysteresis to Improve Noise Rejection
- All Critical Parameters are Guaranteed Over the Entire Commercial, Industrial and Military Temperature Ranges

Applications

- Any System Requiring RS-232 Communications Port
  - Computer - Portable and Mainframe
  - Peripheral - Printers and Terminals
  - Portable Instrumentation
  - Modems
  - Dataloggers

Description

The ICL232 is a dual RS-232 transmitter/receiver interface circuit that meets all EIA RS-232C specifications. It requires a single +5V power supply, and features two onboard charge pump voltage converters which generate +10V and -10V supplies from the 5V supply.

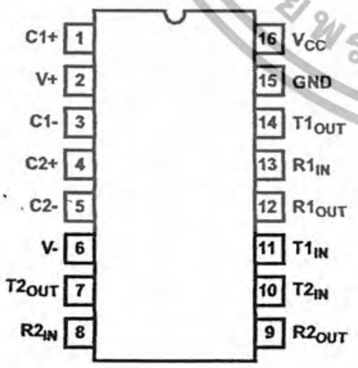
The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and 300Ω power-off source impedance. The receivers can handle up to +30V, and have a 3kΩ to 7kΩ input impedance. The receivers also have hysteresis to improve noise rejection.

Ordering Information

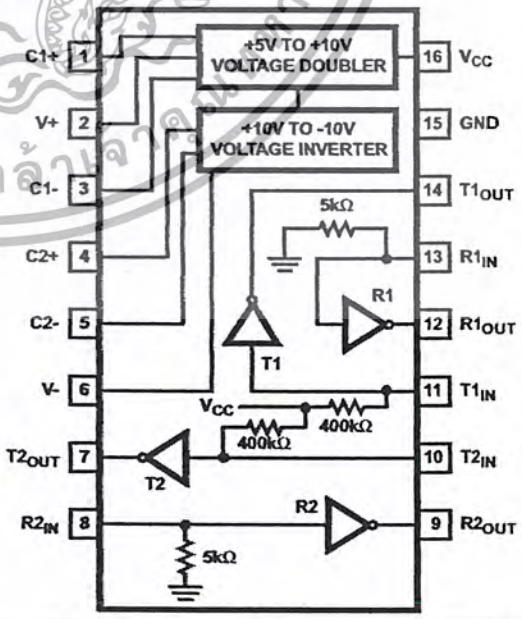
PART NUMBER	TEMPERATURE RANGE	PACKAGE
ICL232CPE	0°C to +70°C	16 Lead Plastic DIP
ICL232CJE	0°C to +70°C	16 Lead Ceramic DIP
ICL232CBE	0°C to +70°C	16 Lead SOIC (W)
ICL232IPE	-40°C to +85°C	16 Lead Plastic DIP
ICL232IJE	-40°C to +85°C	16 Lead Ceramic DIP
ICL232IBE	-40°C to +85°C	16 Lead SOIC (W)
ICL232MJE	-55°C to + 125°C	16 Lead Ceramic DIP

Pinouts

ICL232
(PDIP, CDIP, SOIC)
TOP VIEW



Functional Diagram



CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper I.C. Handling Procedures.

Copyright © Harris Corporation 1993

File Number 3020.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

V_{CC} to Ground . . . . (GND -0.3V) <V_{CC} < 6V
V+ to Ground . . . . (V_{CC} -0.3V) <V+ < 12V
V- to Ground . . . . -12V < V- < (GND +0.3V)
Input Voltages
$T1_{IN}$, $T2_{IN}$ . . . . (V- -0.3V) < V_{IN} < (V+ +0.3V)
$R1_{IN}$, $R2_{IN}$ . . . . ±30V
Output Voltages
$T1_{OUT}$, $T2_{OUT}$ . . . . (V- -0.3V) < V_{TXOUT} < (V+ +0.3V)
$R1_{OUT}$, $R2_{OUT}$ . . . . (GND -0.3V) < V_{RXOUT} < (V_{CC} +0.3V)
Short Circuit Duration
$T1_{OUT}$, $T2_{OUT}$ . . . . Continuous
$R1_{OUT}$, $R2_{OUT}$ . . . . Continuous
Storage Temperature Range . . . . -65°C to +150°C
Lead Temperature (Soldering 10s) . . . . +300°C

Thermal Information

Thermal Resistance	θ_{JA}	θ_{JC}
Ceramic DIP Package	80°C/W	24°C/W
Plastic DIP Package	100°C/W	-
SOIC Package	100°C/W	-

Maximum Power Dissipation 250mW
Operating Temperature Range
ICL232C . . . . 0°C to +70°C
ICL232I . . . . -40°C to +85°C
ICL232M . . . . -55°C to +125°C

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Electrical Specifications

Test Conditions: V_{CC} = +5V ±10%, T_A = Operating Temperature Range. Test Circuit as in Figure 8 Unless Otherwise Specified

PARAMETER	TEST CONDITIONS	LIMITS			UNITS
		MIN	TYP	MAX	
Transmitter Output Voltage Swing, T_{OUT}	$T1_{OUT}$ and $T2_{OUT}$ loaded with 3kΩ to Ground	±5	±9	±10	V
Power Supply Current, I_{CC}	Outputs Unloaded, T_A = +25°C	-	5	10	mA
T_{IN}, Input Logic Low, V_{IL}		-	-	0.8	V
T_{IN}, Input Logic High, V_{IH}		2.0	-	-	V
Logic Pullup Current, I_P	$T1_{IN}$, $T2_{IN}$ = 0V	-	15	200	μA
RS-232 Input Voltage Range, V_{IN}		-30	-	+30	V
Receiver Input Impedance, R_{IN}	V_{IN} = ±3V	3.0	5.0	7.0	kΩ
Receiver Input Low Threshold, V_{IN} (H-L)	V_{CC} = 5.0V, T_A = +25°C	0.8	1.2	-	V
Receiver Input High Threshold, V_{IN} (L-H)	V_{CC} = 5.0V, T_A = +25°C	-	1.7	2.4	V
Receiver Input Hysteresis, V_{HYST}		0.2	0.5	1.0	V
TTL/CMOS Receiver Output Voltage Low, V_{OL}	I_{OUT} = 3.2mA	-	0.1	0.4	V
TTL/CMOS Receiver Output Voltage High, V_{OH}	I_{OUT} = -1.0mA	3.5	4.6	-	V
Propagation Delay, t_{PD}	RS-232 to TTL	-	0.5	-	μs
Instantaneous Slew Rate, SR	C_L = 10pF, R_L = 3kΩ, T_A = +25°C (Notes 1, 2)	-	-	30	V/μs
Transition Region Slew Rate, SR_T	R_L = 3kΩ, C_L = 2500pF Measured from +3V to -3V or -3V to +3V	-	3	-	V/μs
Output Resistance, R_{OUT}	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V	300	-	-	Ω
RS-232 Output Short Circuit Current, I_{SC}	$T1_{OUT}$ or $T2_{OUT}$ shorted to GND	-	±10	-	mA

NOTES:

1. Guaranteed by design.
2. See Figure 4 for definition.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves

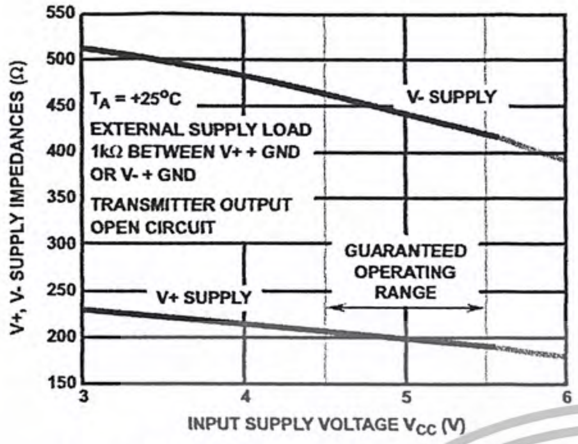


FIGURE 1. V+, V- OUTPUT IMPEDANCES vs V_{CC}

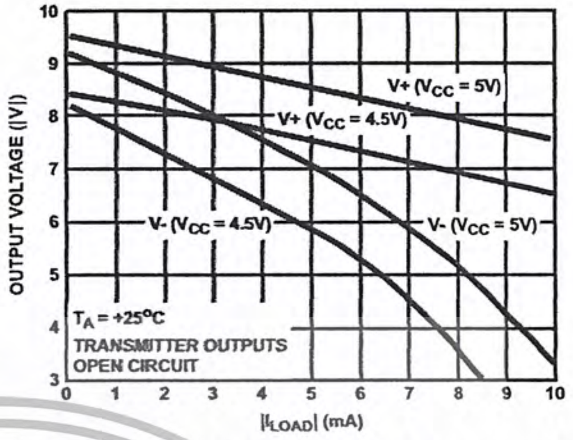


FIGURE 2. V+, V- OUTPUT VOLTAGES vs LOAD CURRENT

Pin Descriptions

PLASTIC DIP, CERAMIC DIP	SOIC	PIN NAME	DESCRIPTION
1	1	C1+	External capacitor "+" for internal voltage doubler.
2	2	V+	Internally generated +10V (typical) supply.
3	3	C1-	External capacitor "-" for internal voltage doubler.
4	4	C2+	External capacitor "+" internal voltage inverter.
5	5	C2-	External capacitor "-" internal voltage inverter.
6	6	V-	Internally generated -10V (typical) supply.
7	7	$T2_{OUT}$	RS-232 Transmitter 2 output ±10V (typical).
8	8	$R2_{IN}$	RS-232 Receiver 2 input, with internal 5K pulldown resistor to GND.
9	9	R2out	Receiver 2 TTL/CMOS output.
10	10	$T2_{IN}$	Transmitter 2 TTL/CMOS input, with internal 400K pullup resistor to V_{CC}.
11	11	$T1_{IN}$	Transmitter 1 TTL/CMOS input, with internal 400K pullup resistor to V_{CC}.
12	12	$R1_{OUT}$	Receiver 1 TTL/CMOS output.
13	13	$R1_{IN}$	RS-232 Receiver 1 input, with internal 5K pulldown resistor to GND.
14	14	$T1_{OUT}$	RS-232 Transmitter 1 output ±10V (typical).
15	15	GND	Supply Ground.
16	16	VCC	Positive Power Supply +5V ±10%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Detailed Description

The ICL232 is a dual RS-232 transmitter/receiver powered by a single +5V power supply which meets all EIA RS232C specifications and features low power consumption. The functional diagram illustrates the major elements of the ICL232. The circuit is divided into three sections: a voltage doubler/inverter, dual transmitters, and dual receivers.

Voltage Converter

An equivalent circuit of the dual charge pump is illustrated in Figure 3.

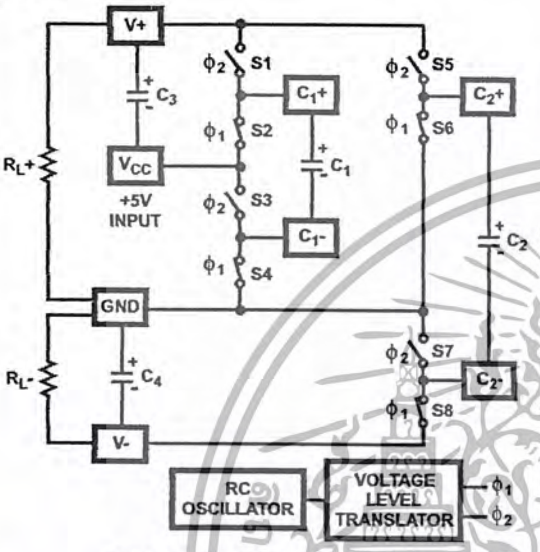
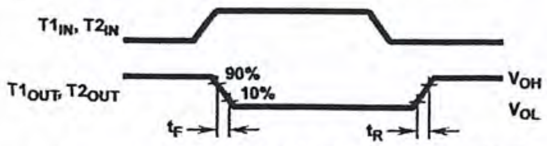


FIGURE 3. DUAL CHARGE PUMP

The voltage quadrupler contains two charge pumps which use two phases of an internally generated clock to generate +10V and -10V. The nominal clock frequency is 16kHz. During phase one of the clock, capacitor C1 is charged to V_{CC}. During phase two, the voltage on C1 is added to V_{CC}, producing a signal across C2 equal to twice V_{CC}. At the same time, C3 is also charged to $2V_{CC}$, and then during phase one, it is inverted with respect to ground to produce a signal across C4 equal to $-2V_{CC}$. The voltage converter accepts input voltages up to 5.5V. The output impedance of the doubler (V+) is approximately 200Ω, and the output impedance of the inverter (V-) is approximately 450Ω. Typical graphs are presented which show the voltage converters output vs input voltage and output voltages vs load characteristics. The test circuit (Figure 8) uses 1µF capacitors for C1-C4, however, the value is not critical. Increasing the values of C1 and C2 will lower the output impedance of the voltage doubler and inverter, and increasing the values of the reservoir capacitors, C3 and C4, lowers the ripple on the V+ and V- supplies.



$$\text{Instantaneous Slew Rate (SR)} = \frac{(0.8)(V_{OH} - V_{OL})}{t_R} \text{ or } \frac{(0.8)(V_{OL} - V_{OH})}{t_F}$$

FIGURE 4. SLEW RATE DEFINITION

Transmitters

The transmitters are TTL/CMOS compatible inverters which translate the inputs to RS-232 outputs. The input logic threshold is about 26% of V_{CC}, or 1.3V for V_{CC} = 5V. A logic 1 at the input results in a voltage of between -5V and V- at the output, and a logic 0 results in a voltage between +5V and (V+ -0.6V). Each transmitter input has an internal 400kΩ pullup resistor so any unused input can be left unconnected and its output remains in its low state. The output voltage swing meets the RS-232C specification of ±5V minimum with the worst case conditions of: both transmitters driving 3kΩ minimum load impedance, V_{CC} = 4.5V, and maximum allowable operating temperature. The transmitters have an internally limited output slew rate which is less than 30V/µs. The outputs are short circuit protected and can be shorted to ground indefinitely. The powered down output impedance is a minimum of 300Ω with ±2V applied to the outputs and V_{CC} = 0V.

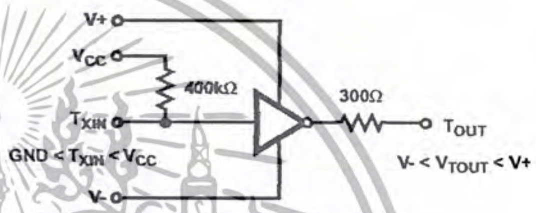


FIGURE 5. TRANSMITTER

Receivers

The receiver inputs accept up to ±30V while presenting the required 3kΩ to 7kΩ input impedance even it the power is off (V_{CC} = 0V). The receivers have a typical input threshold of 1.3V which is within the ±3V limits, known as the transition region, of the RS-232 specification. The receiver output is 0V to V_{CC}. The output will be low whenever the input is greater than 2.4V and high whenever the input is floating or driven between +0.8V and -30V. The receivers feature 0.5V hysteresis to improve noise rejection.

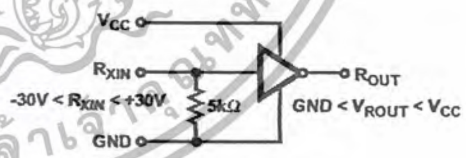
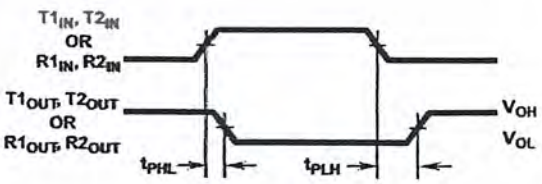


FIGURE 6. RECEIVER



$$\text{Average Propagation Delay} = \frac{t_{PHL} + t_{PLH}}{2}$$

FIGURE 7. PROPAGATION DELAY DEFINITION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuits

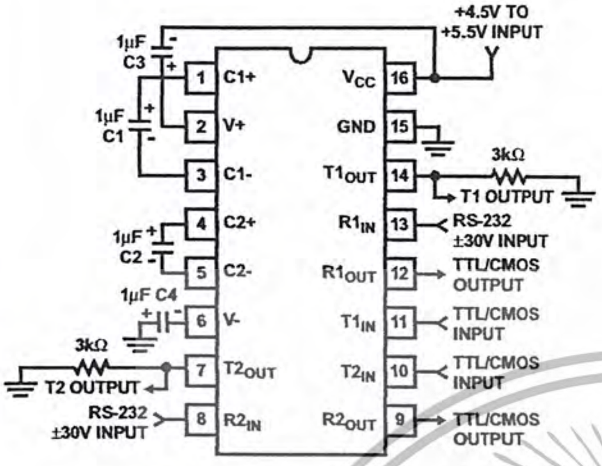


FIGURE 8. GENERAL TEST CIRCUIT

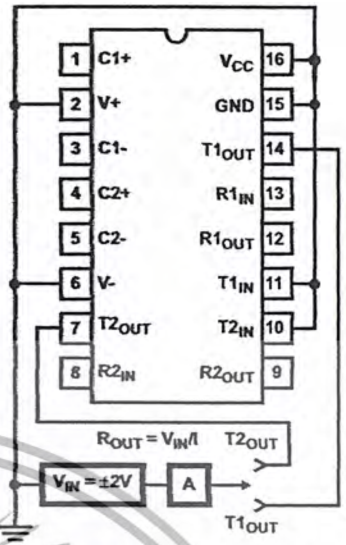


FIGURE 9. POWER-OFF SOURCE RESISTANCE CONFIGURATION

Applications

The ICL232 may be used for all RS-232 data terminal and communication links. It is particularly useful in applications where ±12V power supplies are not available for conventional RS-232 interface circuits. The applications presented represent typical interface configurations.

A simple duplex RS-232 port with CTS/RTS handshaking is illustrated in Figure 10. Fixed output signals such as DTR (data terminal ready) and DSRS (data signaling rate select) is generated by driving them through a 5kΩ resistor connected to V+.

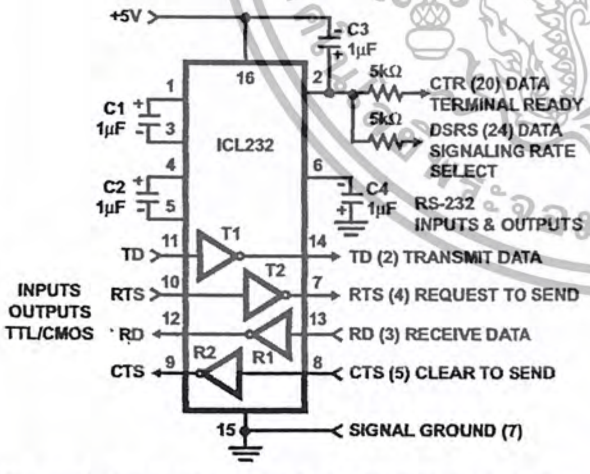


FIGURE 10. SIMPLE DUPLEX RS-232 PORT WITH CTS/RTS HANDSHAKING

In applications requiring four RS-232 inputs and outputs (Figure 11), note that each circuit requires two charge pump capacitors (C1 and C2) but can share common reservoir capacitors (C3 and C4). The benefit of sharing common reservoir capacitors is the elimination of two capacitors and the reduction of the charge pump source impedance which effectively increases the output swing of the transmitters.

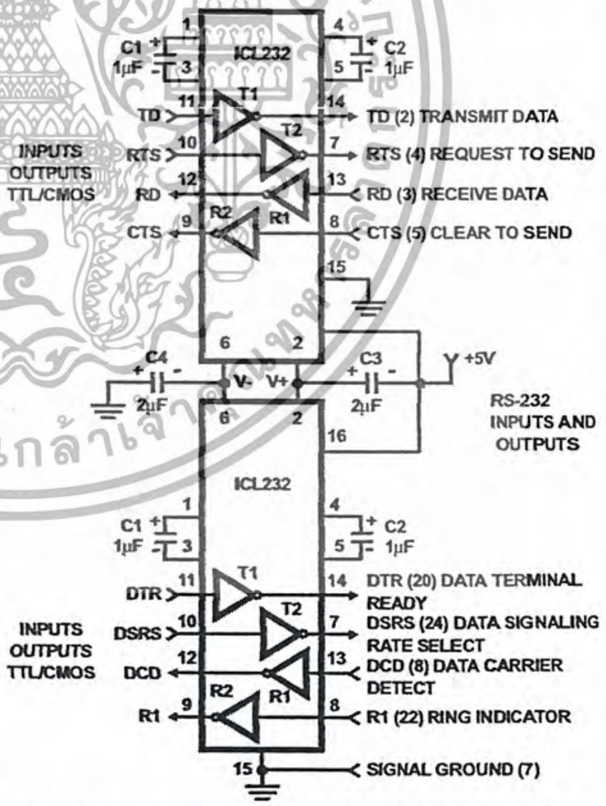


FIGURE 11. COMBINING TWO ICL232s FOR 4 PAIRS OF RS-232 INPUTS AND OUTPUTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้