สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้งานต่อพ่วงคำสั่ง PC กับตัวบัส I^2C ไมโครคอนโทรเลอร์

PC – INTERFACING WITH I^2C BUS MICROCONTROLLER APPLICATION

T 050252

นายจิระ เดชไพฑูรย์กุล
นายบรรพต ด่านกลาง
นายภีรศักดิ์ เผ่าผาง

เลขหมู่.....
เลขทะเบียน.....50252
วัน,เดือน,ปี.....28 เม.ย. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC – INTERFACING WITH I^2C BUS MICROCONTROLLER APPLICATION

JIRA DETPAITOONKUL

BUNPOT DANKLANG

PERASAK PAOPANG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMANT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF THCHNOLOGY LADKRABANG

2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	การประยุกต์ใช้งานต่อพ่วงคำสั่ง PC กับตัวบัส I^2C ไมโครคอนโทรเลอร์ PC – INTERFACING WITH I^2C BUS MICROCONTROLLER APPLICATION		
นักศึกษาผู้จัดทำ	นายจิระ	เดชไพฑูรย์กุล	รหัสประจำตัว 43015504
	นายบรรพต	ด่านกลาง	รหัสประจำตัว 43015514
	นายภีรศักดิ์	เผ่าผาง	รหัสประจำตัว 43015527
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมการวัดคุม		
ปีการศึกษา	2545		

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
รศ. พิพัฒน์ เลาหสงคราม	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 22 เมษายน 2546

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การประยุกต์ใช้งานต่อพ่วงคำสั่ง PC กับตัวบัส I^2C ไมโครคอนโทรเลอร์
	PC – INTERFACING WITH I^2C BUS MICROCONTROLLER
	APPLICATION
นักศึกษาผู้จัดทำ	นายจิระ เดชไพฑูรย์กุล
	นายบรรพต ด่านกลาง
	นายภีรศักดิ์ เผ่าผาง
อาจารย์ที่ปรึกษา	รศ. พิพัฒน์ เลาหสงคราม
ปีการศึกษา	2545

บทคัดย่อ

ปริญญานิพนธ์นี้เสนอแนวทางศึกษาการใช้ I^2C Bus สำหรับ MCS-51 กับอุปกรณ์อื่นๆโดยตัวบอร์ดที่ออกแบบมา สามารถที่จะประยุกต์ใช้กับกระบวนการได้ โดยในปริญญานิพนธ์นี้จะทดลองต่อกับกระบวนการจำลองการควบคุมระดับของของเหลว และสามารถที่จะต่อกับ PC เพื่อที่จะแสดงการทำงานของกระบวนการได้ และสามารถที่จะทำงานในโหมด Stand Alone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	PC – Interfacing with I^2C Bus Microcontroller Application
Authors	Mr. Jira Detpaitoonkul
	Mr.Bunpot Dankalang
	Mr. Perasak Paopang
Thesis Advisor	Assoc.Prof.Phiphat Laohasongkram
Year	2002

ABSTRACT

This thesis represents the administration of PC-Interfacing With I^2C Bus Microcontroller Application as the circuit designed in order to apply with this process. And this, was experimented to combine with the liquid level controling model process and also with PC in order to display the process of this administration including can be worked together in the Stand Alone mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำปรึกษาแนะนำและเสนอแนะแนวทางการดำเนินงานเป็นอย่างดี ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ รองศาสตราจารย์ พิพัฒน์ เลาหสงคราม ที่ให้คำแนะนำแนวทางการดำเนินงานเป็นอย่างดี ขอขอบพระคุณ อาจารย์ภาควิชาวัดคุมทุกท่าน ที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ที่ให้ความช่วยเหลือในด้านการเงินและเป็นกำลังใจด้วยดีตลอดมา ตลอดจนพี่ๆ เพื่อนๆทุกคนที่ให้คำแนะนำและความช่วยเหลือในด้านต่างๆ

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

บทที่ 2 การศึกษาทฤษฎีต่างๆที่เกี่ยวข้องกับปริญญานิพนธ์.....2

2.1 การเชื่อมต่อไมโครคอนโทรลเลอร์บนระบบบัส I^2C.....2

2.2 คุณสมบัติโดยทั่วไปของบัส I^2C.....2

2.3 หลักการของบัส I^2C.....5

2.4 สภาวะที่เกิดขึ้นบนบัส I^2C.....5

2.4.1 บัสว่าง.....5

2.4.2 เริ่มต้นการถ่ายทอดข้อมูล.....6

2.4.3 ข้อมูลดำรงอยู่บนบัส.....6

2.4.4 รับรู้ข้อมูล.....7

2.4.5 หยุดการถ่ายทอดข้อมูล.....8

2.5 การทำงานบนบัส I^2C.....8

2.5.1 การอ้างถึงแบบ 7 บิต.....8

2.5.2 การอ้างถึงแบบ 10 บิต.....9

2.6 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID.....10

2.6.1 การปรับแต่งค่าของตัวควบคุม PID โดยวิธีของ Ziegler-Nichols.....11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 อุปกรณ์ร่วมของระบบฐานบัส I^2Cและโปรแกรม.....14

3.1 วงจรแสดงการทำงานของระบบ.....14

3.2 วงจรเปลี่ยนสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า.....15

3.3 ไมโครคอนโทรลอร์ และอุปกรณ์ร่วมของระบบฐานบัส I^2C.....16

3.4 การเชื่อมต่อกับสัญญาณอะนาลอกผ่านพอร์ตอนุกรม.....24

3.5 EEPROM 2402.....26

3.6โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด(LCD 16x2).....31

3.7 การทำงานของ PID.....32

3.8 Main โปรแกรมหลัก.....33

3.9 โปรแกรมย่อย PID.....34

บทที่ 4 การทดลองและผลการทดลอง.....36

การทดลองที่ 1 แสดงการควบคุมกระบวนการจำลองควบคุมระดับโดยผ่าน Computer และแสดงผลผ่านทาง Computer.....36

การทดลองที่2 แสดงการควบคุมกระบวนการจำลองควบคุมระดับโดย Mode Stand Alone.....40

สรุปผลการทดลอง.....41

บทที่ 5 สรุปหลักการทำงานและการวิจารณ์.....42

5.1 สรุปหลักการทำงานของระบบ.....42

5.2 วิจารณ์.....42

บรรณานุกรม.....43

ภาคผนวก.....44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 Serial Port Control Register.....19

3.2 แสดงโหมดต่างๆ ของการรับส่งข้อมูลแบบอนุกรม.....20

3.3 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Timer 0.....21

3.4 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Counter 0.....21

3.5 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Timer 1.....21

3.6 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Counter 1.....22

3.7 รีจิสเตอร์ใช้เฉพาะ PSW (Program Status Word) เข้าถึงข้อมูลได้ในระดับบิต.....22

3.8 แสดงตำแหน่งและหน้าที่ของขาใน 24LC02.....26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างวงจรเอาต์พุตของอุปกรณ์ที่ใช้การเชื่อมต่อบนบัส I^2C.....3

2.2 แสดงการเชื่อมต่ออุปกรณ์บนระบบบัส I^2C ที่ใช้ไฟเลี้ยงไม่เท่ากัน.....4

2.3 การต่อตัวต้านทานเพื่อป้องกันแรงดันกระชากที่อาจปะปนเข้ามาในไฟเลี้ยงของอุปกรณ์ในระบบบัส I^2C.....4

2.4 แสดงสภาวะช่วงที่บัส I^2C ว่างอยู่.....6

2.5 แสดงสภาวะจุดเริ่มต้น START.....6

2.6 แสดงสภาวะที่ข้อมูลที่ถูกส่งไม่มีการเปลี่ยนแปลง.....7

2.7 ไดอะแกรมเวลาแสดงสภาวะต่างๆ ที่เกิดขึ้นบนระบบบัส I^2C.....7

2.8 รูปแบบของข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ระบบบัส I^2C.....8

2.9 รูปแบบข้อมูลอนุกรมที่ใช้ในการติดต่อกับอุปกรณ์ระบบบัส I^2C แบบ 7 บิต.....9

2.10รูปแบบข้อมูล แบบ 10 บิต.....10

2.11 แสดงโครงสร้างของระบบควบคุมด้วยตัวควบคุม PID.....11

2.12 แสดงค่าที่พุ่งเกินที่ต้องการเมื่อใช้การปรับด้วยวิธี Ziegler-Nichols.....11

3.1 วงจรแสดงการทำงานของระบบ.....14

3.2 แสดงบล็อกไดอะแกรมการทำงานของระบบ.....15

3.3วงจรเปลี่ยนสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า.....15

3.4 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์.....17

3.5 แสดงรายละเอียดข้อมูลควบคุมที่เขียนลงในรีจิสเตอร์ควบคุมภายในไอซี PCF8591.....24

3.6 แสดงโครงสร้างขาใช้งานของ EEPROM 24LC02.....26

3.7 แสดงโครงสร้างภายใน EEOROM 24LC02.....27

3.8 แสดงลักษณะการ Device Address 27

3.9 แสดง Timing Diagram ของการเขียนข้อมูลแบบ Byte.....28

3.10 แสดง Timing Diagram ของการอ่านข้อมูลแบบ Page.....29

3.11 การอ่านข้อมูลในตำแหน่งปัจจุบันของ EEPROM 24LC02.....30

3.12 แสดง Timing Diagram ของการอ่านข้อมูลแบบสุ่มของ EEPROM 24LC02.....30

3.13 แสดง Timing Diagram ของการอ่านข้อมูลแบบลำดับของ EEPROM 24LC02.....31

3.14 แสดงรูปร่างและการจัดขาโมดูล LCD แบบอักขระ.....31

3.15 FLOW CHARTแสดงโปรแกรมหลัก.....33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.16 FLOW CHARTแสดงโปรแกรมย่อย PID.....34

3.17 แสดงแผงวงจรของ PC – Interfacing With I^2C Bus Microcontroller Application.....35

3.18 แสดงกระบวนการจำลองควบคุมระดับของเหลว.....35

4.1 กราฟที่ได้จากที่ SP = 50%.....37

4.2 กราฟที่ได้จากที่ SP = 75%.....38

4.3 กราฟที่ได้จากการสั่งค่า SP เป็น Step.....39

4.4 กราฟจากเครื่อง Recorderแสดงสภาวะการทำงานแต่ละ Step.....40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1

บทนำ

การเชื่อมต่อข้อมูลในปัจจุบันได้มีการใช้มาตรฐานการเชื่อมต่อแบบบัส I^2C กันอย่างแพร่หลายในงานอุตสาหกรรม เนื่องจากความสามารถในการส่งข้อมูล ได้ระยะไกลอีกทั้งการต่ออุปกรณ์บนบัสนั้นก็สามารถใช้ไฟเลี้ยงที่ไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้

ในโครงงานนี้จะใช้บัส I^2C เป็นตัวเชื่อมโยงข้อมูลระหว่าง ไมโครคอนโทรลเลอร์และอุปกรณ์อื่นๆ เช่น วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ไอซีรีลไทม์คล็อก หน่วยความจำ EEPROM โดยมีการกำหนดรูปแบบการติดต่อสื่อสารบนบัสหรือที่เรียกว่า โปรโตรคอล มาพัฒนาใช้งานกับวงจรอื่นๆได้

วัตถุประสงค์

1. เพื่อศึกษามาตรฐานการเชื่อมต่อแบบบัส I^2C
2. เพื่อนำมาตรฐานการเชื่อมต่อแบบบัส I^2C มาประยุกต์และพัฒนาใช้ในการรับส่งข้อมูลบนบัส I^2C
3. สามารถนำโครงงานนี้ไปประยุกต์ใช้กับกระบวนการจริงได้

ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้จะกล่าวถึง การพัฒนาและการออกแบบรูปแบบการเชื่อมต่อระหว่างไมโครคอนโทรเลอร์กับอุปกรณ์อื่นๆบนบัส I^2C โดยใช้การสร้างโปรโตคอลในการติดต่อสื่อสารและสามารถที่จะนำไปประยุกต์ใช้กับกระบวนการจริงได้ ซึ่งสามารถรับส่งค่าระหว่างคอมพิวเตอร์กับบอร์ดไมโครคอนโทรเลอร์ โดยผ่านระบบการเชื่อมต่อแบบ RS-232 และแสดงผลทาง Monitor ของคอมพิวเตอร์ สามารถที่จะนำบอร์ดไมโครคอนโทรเลอร์ไปต่อใช้งานโดยไม่มีคอมพิวเตอร์ต่อพ่วงคอยควบคุม และสามารถเก็บค่าผลการวัดไว้ในตัวบอร์ดไมโครคอนโทรเลอร์ได้ เพื่อเป็นการบันทึกการเปลี่ยนแปลงของระบบ และนำข้อมูลเหล่านี้ไป Load เก็บไว้ในคอมพิวเตอร์ ได้

ขั้นตอนการทำปริญญานิพนธ์

1. เริ่มจากการศึกษาหลักการทำงานของระบบบัส I^2C
2. ศึกษาและทำการเขียนโปรแกรมภาษา แอสเซมบลี้(Assembly)
3. ศึกษาและทำการเขียนโปรแกรม Visual Basic เพื่อใช้แสดงผลบน คอมพิวเตอร์
4. ศึกษาหลักการทำงานของระบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การศึกษาทฤษฎีต่างๆที่เกี่ยวข้องกับปริญญานิพนธ์

2.1 การเชื่อมต่อไมโครคอนโทรลเลอร์บนระบบบัส I^2C

I^2C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูล อีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัวจะใช้รหัสข้อมูลและการกำหนดสภาวะ ลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

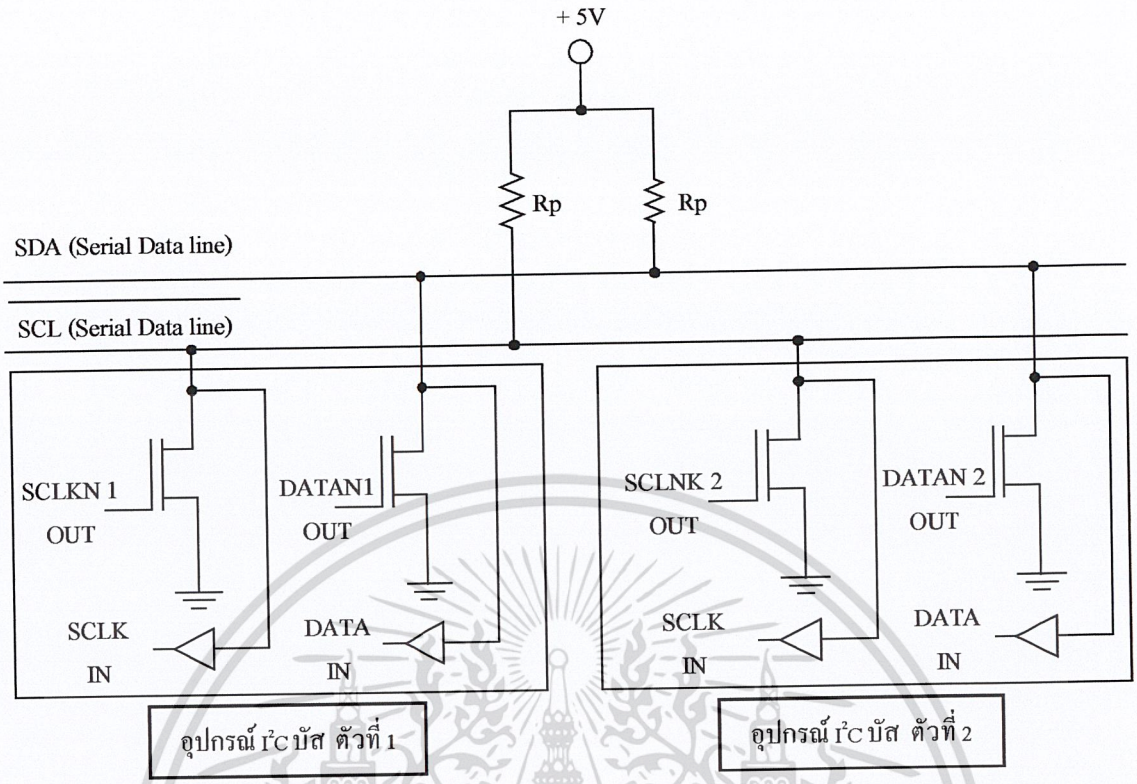
สายข้อมูลบนบัส I^2C มีชื่อเรียกอีกอย่างเป็นทางการว่า **สายข้อมูลอนุกรม** หรือ **SDA (Serial Data Line)** ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า **สายสัญญาณนาฬิกาอนุกรม** หรือ **SCL (Serial Clock Line)** ในการอธิบายต่อไปนี้ จะเรียกสายสัญญาณทั้งสองว่า สาย **SDA** และ **SCL**

อุปกรณ์ที่ใช้ในการเชื่อมต่อด้วยบัส I^2C มีหลากหลาย ไม่ว่าจะเป็น ไอซีแปลงสัญญาณอะนาลอกเป็นดิจิตอล (ADC) และแปลงสัญญาณดิจิตอลเป็นอะนาลอก (DAC), ไอซีรีลไทม์คล็อก (RTC), หน่วยความจำอีอีพรอมและไมโครคอนโทรลเลอร์

2.2 คุณสมบัติโดยทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (Bit-Directional Line) ต้องมีการต่อตัวต้านทานพูลอัปกับแรงดัน +5V ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสองวงจรเอาต์พุตของอุปกรณ์ที่ติดอยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด(Open-Drain)หรือคอลเล็กเตอร์เปิด (Open-Collector) ดังแสดงรายละเอียดในรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

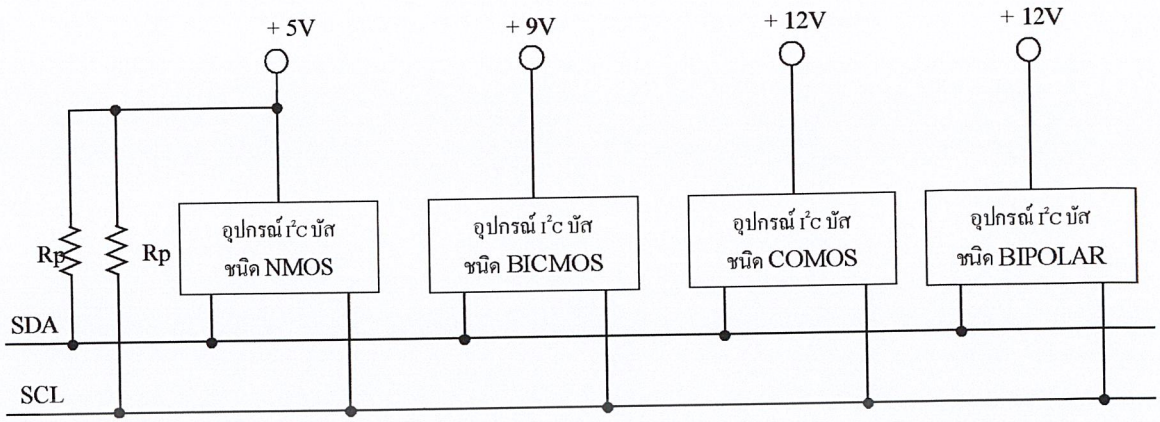


รูปที่ 2.1 โครงสร้างวงจรเอาต์พุตของอุปกรณ์ที่ใช้การเชื่อมต่อบนบัส I^2C

อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (Fast Mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2Cจะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400pF การเข้าถึงอุปกรณ์บนบัส I^2Cใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7 บิต (7-Bit Addressing) หรือ 10 บิต (10-Bit Addressing)

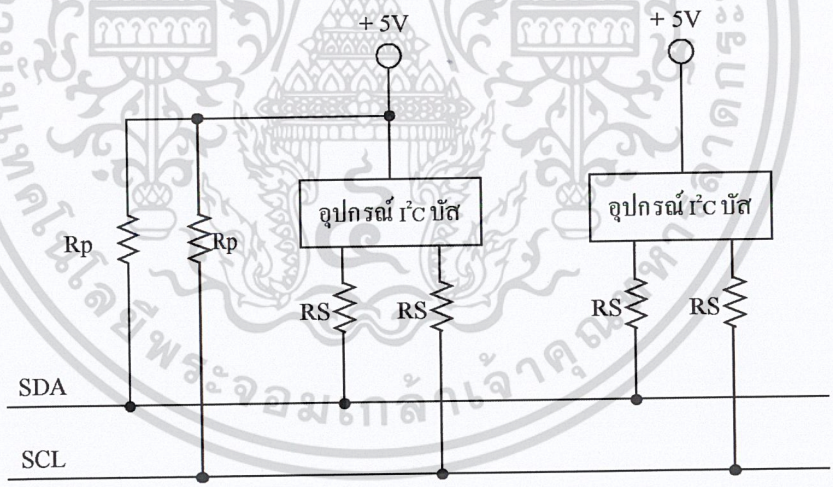
ข้อเด่นอีกประการหนึ่งของบัส คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้โดยอุปกรณ์บนบัสตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12V การต่อร่วมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทาน พูลอัป (R_p) เข้ากับแรงดัน + 5V ไว้ด้วยเสมอ ดังแสดงในรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการเชื่อมต่ออุปกรณ์บนระบบบัส I^2C ที่ใช้ไฟเลี้ยงไม่เท่ากัน

ในกรณีที่อาจมีแรงดันไฟกระชากขนาดใหญ่ปะปนเข้ามาในบัส I^2C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัวต้องต่อตัวต้านทานอนุกรมกับขา SDA และ SCL เรียกว่า R_s ก่อนต่อเข้าสู่บัส I^2C ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 การต่อตัวต้านทานเพื่อป้องกันแรงดันกระชากที่อาจปะปนเข้ามาในไฟเลี้ยงของอุปกรณ์ในระบบบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หลักการของบัส I^2C

บัส I^2C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า **โปรโตคอล (protocol)** เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I^2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I^2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ***ตัวส่ง (Transmitter)***

อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ***ตัวรับ (Reciever)***

ในอุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว

อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการทำงานหรือการติดต่อบนบัส I^2C เรียกว่า ***มาสเตอร์ (Master)***

อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า ***สเลฟ (Slave)***

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

2) ในระหว่างการถ่ายทอดข้อมูลเมื่อใดก็ตามที่สายSCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

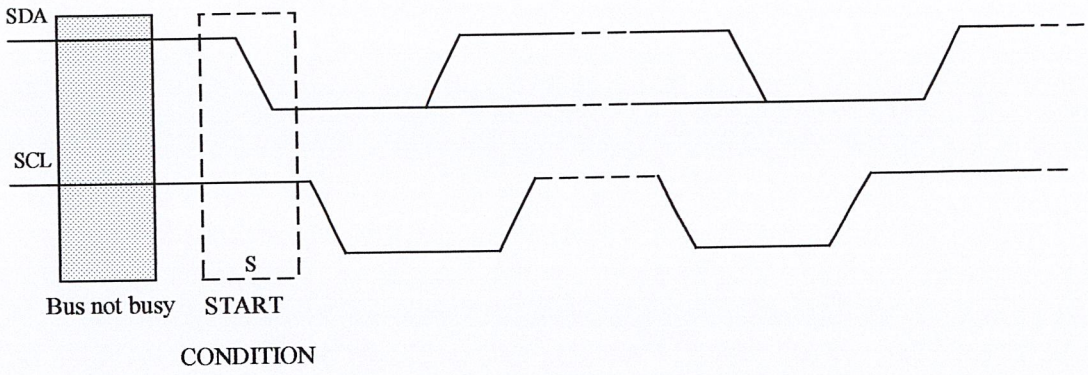
2.4 สภาวะที่เกิดขึ้นบนบัส I^2C

มีด้วยกัน 5 สภาวะ ดังนี้

2.4.1 บัสว่าง (Bus Not Busy)

สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสายSDAและSCLเป็นลอจิกสูงทั้งคู่นั่นหมายความว่าการถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

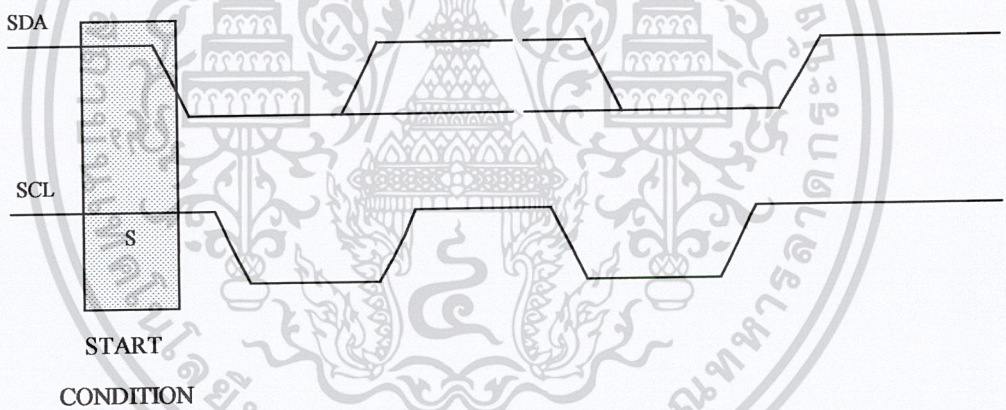
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงสภาวะช่วงที่บัส I^2C ว่างอยู่

2.4.2 เริ่มต้นการถ่ายทอดข้อมูล (Start Data Transfer)

เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น (START)



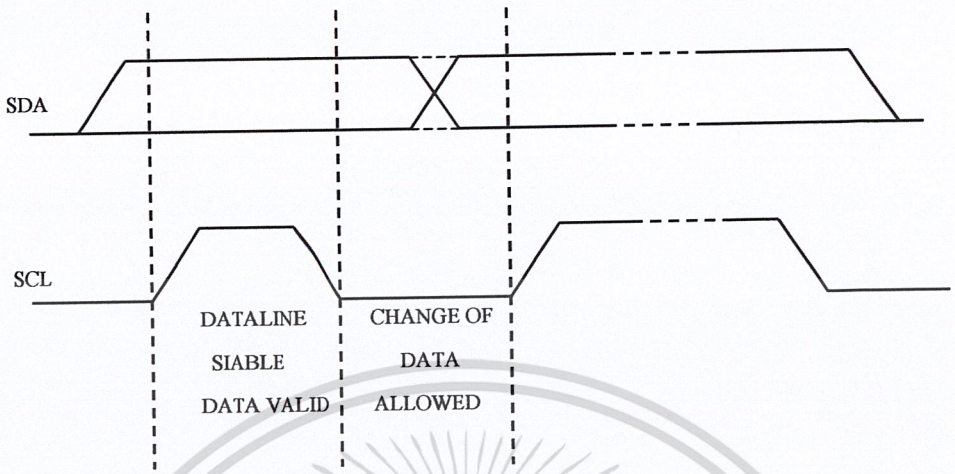
รูปที่ 2.5 แสดงสภาวะจุดเริ่มต้น START

2.4.3 ข้อมูลดำรงอยู่บนบัส (Data Valid)

สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

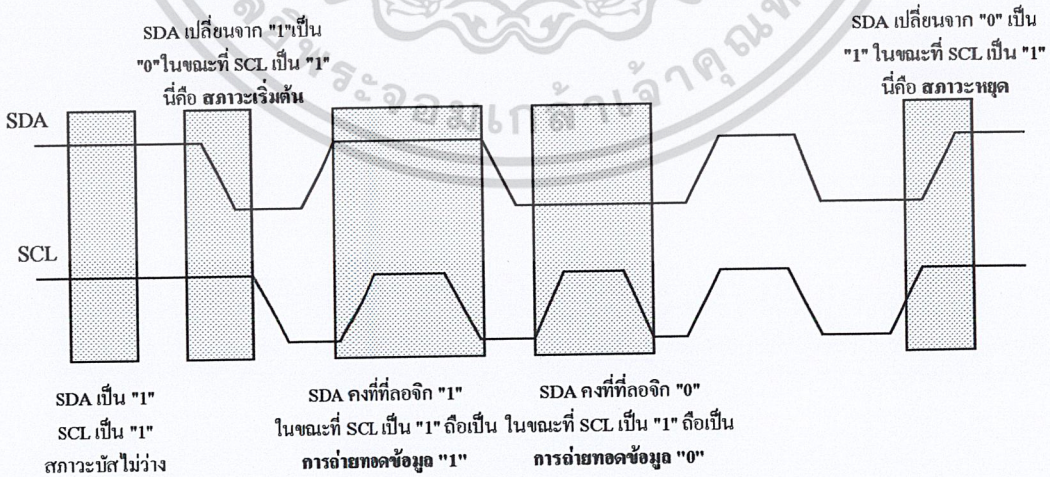
แปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น



รูปที่2.6 แสดงสภาวะที่ข้อมูลที่ถูกส่งไม่มีการเปลี่ยนแปลง

2.4.4 รับรู้ข้อมูล (Acknowledge)

เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิต เรียกว่า บิตรับรู้ (Acknowledge Bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา อุปกรณ์ สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้ที่มีสถานะลอจิกต่ำเพื่อตอบสนองให้ทราบว่าได้รับรู้ข้อมูลเรียบร้อยแล้ว



รูปที่ 2.7 ไดอะแกรมเวลาแสดงสภาวะต่างๆ ที่เกิดขึ้นบนระบบบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5 หยุดการถ่ายทอดข้อมูล (Stop Data Transfer)

เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูงในขณะที่สาย SCL มีสถานะ ลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)ในรูปที่ 2.4 เป็นไดอะแกรมเวลาที่แสดงถึงการเกิดสภาวะต่าง ๆ บนบัส I^2C ไม่ว่าจะเป็นสภาวะบัสว่าง, เริ่มต้น, ถ่ายทอดข้อมูล, รับรู้ และหยุดการถ่ายทอดข้อมูล

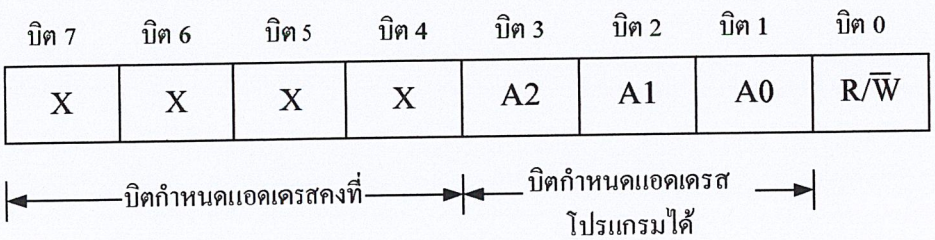
2.5 การทำงานบนบัส I^2C

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่าง ๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงอุปกรณ์เสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I^2C นั้นจะใช้การอ้างถึงแบบ 7 บิต หรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

ดังนั้น หัวใจสำคัญในอันดับแรกของการทำงานบนบัส I^2C คือการอ้างถึงอุปกรณ์แต่ละตัว ซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ

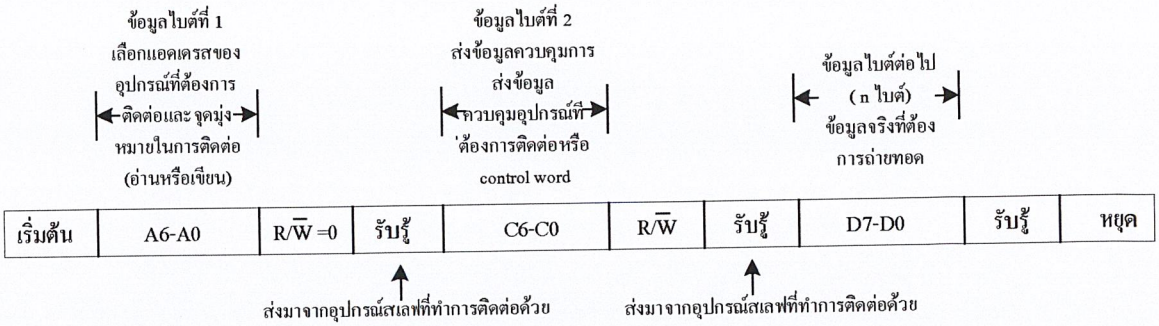
2.5.1 การอ้างถึงแบบ 7 บิต (7-Bit Addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ โดยมีรูปแบบแสดงในรูปที่2.8ใน 7 บิตบนรวมทั้งบิตMSBด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (Fixed Address Bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิต เป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (Programmable Address Bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I^2C ส่วนในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้น ๆ หากบิต LSB เป็น “0” หมายถึง ต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น “1” จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.8 รูปแบบของข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ระบบบัส I^2C



รูปที่ 2.9 รูปแบบข้อมูลอนุกรมที่ใช้ในการติดต่อกับอุปกรณ์ระบบบัส I^2C แบบ 7 บิต

ข้อมูลในไบต์ต่อมาคือข้อมูลควบคุม(Control Byte)ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไปยกตัวอย่าง ไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่าบิตใดเป็น อินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น

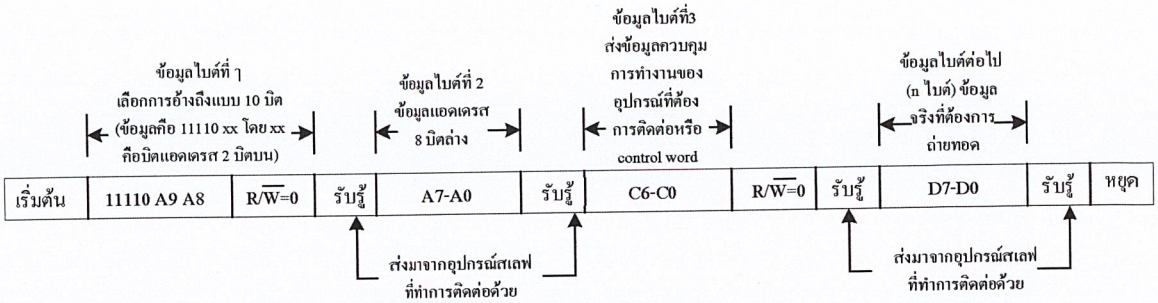
ข้อมูลในไบต์ต่อมา คือ ข้อมูลที่ทำการถ่ายทอดจริง(Data)หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.9 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I^2C ของการอ้างถึงแบบ 7 บิต

2.5.2 การอ้างถึงแบบ 10 บิต (10-Bit Addressing)

ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกับแบบ 7 บิตหากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะเป็นข้อมูลจริงที่ใช้ในการติดต่อ

เช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ในรูปที่2.10 แสดงรูปแบบข้อมูลอนุกรมของการอ้างถึงแบบ 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10รูปแบบข้อมูล แบบ 10 บิต

2.6 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID

การปรับแต่งหาค่าซึ่งสามารถหาได้ทางคณิตศาสตร์และทางปฏิบัติ แต่ส่วนใหญ่ในโรงงานอุตสาหกรรมก็จะนิยมใช้แบบทางด้านปฏิบัติมากกว่าเนื่องจากการหาค่าทางคณิตศาสตร์ จำเป็นจะต้องรู้ตัวแปรในกระบวนการทั้งหมด ซึ่งเป็นเรื่องที่ยุ่งยากมาก

การปรับแต่งหาค่าพารามิเตอร์ PID เป็นสิ่งจำเป็นเพื่อให้ได้การควบคุมที่ดีที่สุดซึ่งทำได้หลายวิธีบางวิธีจะพิจารณาจากผลตอบสนองของระบบและอาศัยประสบการณ์ บางวิธีจะอาศัยคณิตศาสตร์ และส่วนใหญ่จะพิจารณาจากเงื่อนไขในโดเมนของเวลามากกว่าในโดเมนของความถี่

พิจารณาฟังก์ชั่นถ่ายโอนของตัวควบคุมแบบ PID ดังนี้

$$G_c(s) = K_p\left(1+\frac{1}{T_i s}+T_d s\right) \tag{1}$$

โดยที่ K_p Proportional Gain

T_i Intergral Time

T_d Derivative Time

เมื่อ e(t) และ u(t) คืออินพุตและเอาท์พุตของตัวควบคุม จะได้ว่า

$$u(t) = K_p\left(e(t)+\frac{1}{T_i}\int_{-\infty}^{t} e(t)dt + T_d\frac{de(t)}{dt}\right) \tag{2}$$

นอกจากนี้ ฟังก์ชั่นถ่ายโอนของตัวควบคุมแบบ PID อาจจะเขียนอยู่ในรูปของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \qquad (3)$$

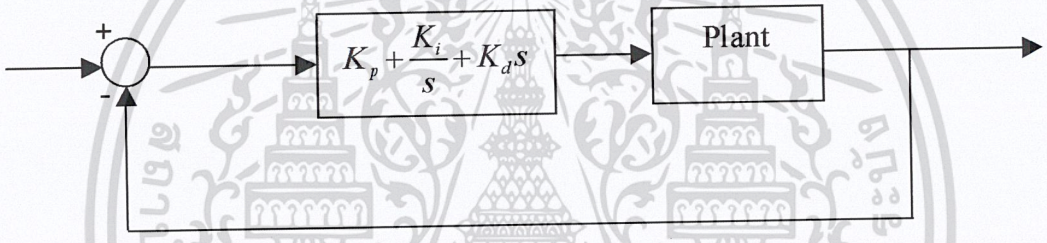
โดยที่ K_p Proportional Gain

K_i Intergral Gain

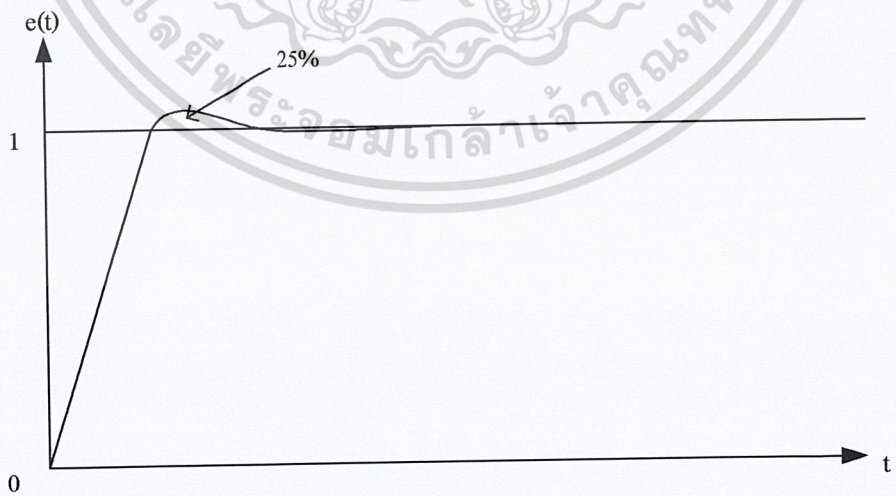
K_d Derivative Gain

2.6.1 การปรับแต่งค่าของตัวควบคุม PID โดยวิธีของ Ziegler-Nichols

วิธีของ Ziegler-Nichols ในการหาค่า K_p , T_i และ T_d จะขึ้นอยู่กับคุณลักษณะของผลตอบสนองชั่วขณะชั่วครู่ของระบบที่ถูกควบคุม ซึ่งมีอยู่ 2 วิธี แต่ละวิธีมีจุดมุ่งหมายที่จะทำให้ผลตอบสนองเวลาของระบบต่ออินพุตแบบ Unit Step มีค่าของ Maximum Overshoot ไม่เกิน 25% ดังรูป



รูปที่ 2.11 แสดงโครงสร้างของระบบควบคุมด้วยตัวควบคุม PID



รูปที่ 2.12 แสดงค่าที่พุ่งเกินที่ต้องการเมื่อใช้การปรับด้วยวิธี Ziegler-Nichols

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบ PID นั้นได้รวบรวมเอาการควบคุมกริยาแบบ Proportional ซึ่งเป็นอัตราขยายกริยาควบคุมแบบ Integral สามารถแก้ค่า offset และ overshoot ที่ทำให้เกิดการ Oscillate และกริยาควบคุมแบบ Derivative จะทำให้เกิดผลตอบสนองได้อย่างรวดเร็วซึ่งสมการของ PID มีดังนี้

$$u(t) = K_p e(t) + K_i \int e dt + k_d \frac{de}{dt} \tag{4}$$

K_p, K_i, K_d = PID gain constants

$u(t)$ = control signal

$e(t)$ = error signal

แปลงไปอยู่ในรูปแบบของ Discrete โดยพจน์ของเทอม Interal เป็นผลรวมของ rectangles $\sum e_i \times T$, เมื่อ T เป็นเวลาที่สุ่มและ e_i เป็นปริมาณของสัญญาณ error ที่เวลาสุ่ม i เขียนได้

$$\int e dt = \sum e_i T \tag{5}$$

ถ้าเวลาที่ T เป็นเวลาที่น้อยเทอม $\frac{de}{dt}$ สามารถประมาณได้ว่า

$$\frac{de}{dt} = \frac{e(n) - e(n-1)}{T} \tag{6}$$

เมื่อ $e(n)$ และ $e(n-1)$ เป็นปริมาณของสัญญาณ error e ที่เวลาระหว่าง n และ $n-1$

PID algoritum สามารถเขียนใหม่ได้เป็น

$$u(n) = K_p e(n) + K_i \sum e_i T + K_d [e(n) - e(n-1)] / T \tag{7}$$

รวมสมการที่2 เข้าไปในสมการ Difference สมการที่2 สำหรับเวลาระหว่าง $(n-2)$ เขียนเป็น

$$u(n-2) = K_p e(n-2) + K_i \sum_{i=1}^{n-2} e_i T + K_d [e(n-1) - e(n-2)] / T \tag{8}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลบสมการที่5 จากสมการที่4

$$u(n)-u(n-2)=K_p e(n)-e(n-2)+K_i\{e(n)+e(n-1)\}T$$
$$+(K_d/T)\{e(n)-2e(n-1)+e(n-2)\} \quad (9)$$

$$u(n)=u(n-2)+(K_p+K_d/T+K_iT)e(n)-(K_iT-2K_d/T)e(n-1)$$
$$+(K_d/T-K_p)e(n-2) \quad (10)$$

หรือ

$$u(n)=u(n-2)+K_1e(n)+K_2e(n-1)+K_3e(n-2)$$

ซึ่ง

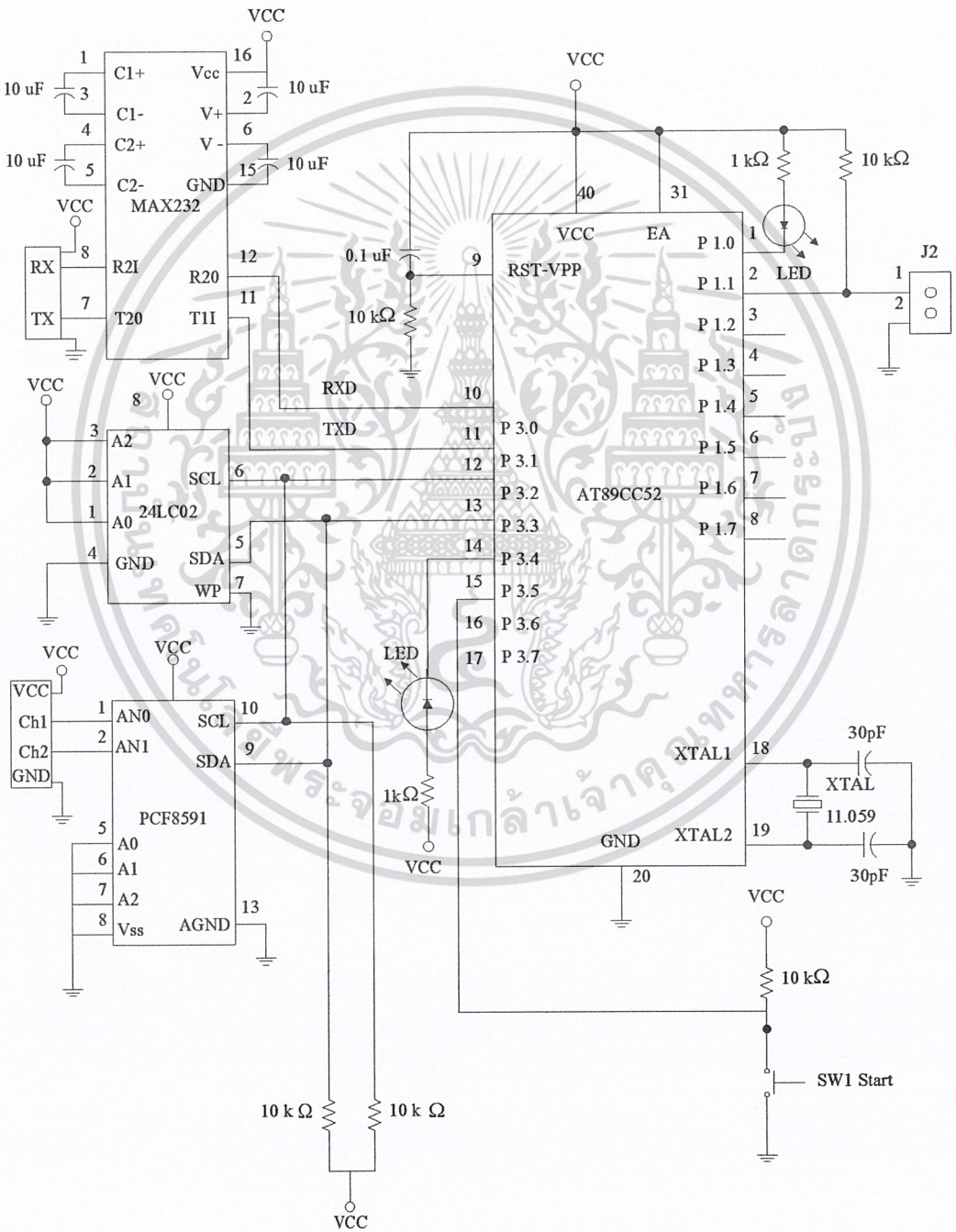
$$K_1=K_p+K_d/T+K_iT$$
$$K_2=K_iT-2K_d/T$$
$$K_3=K_d/T-K_p$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

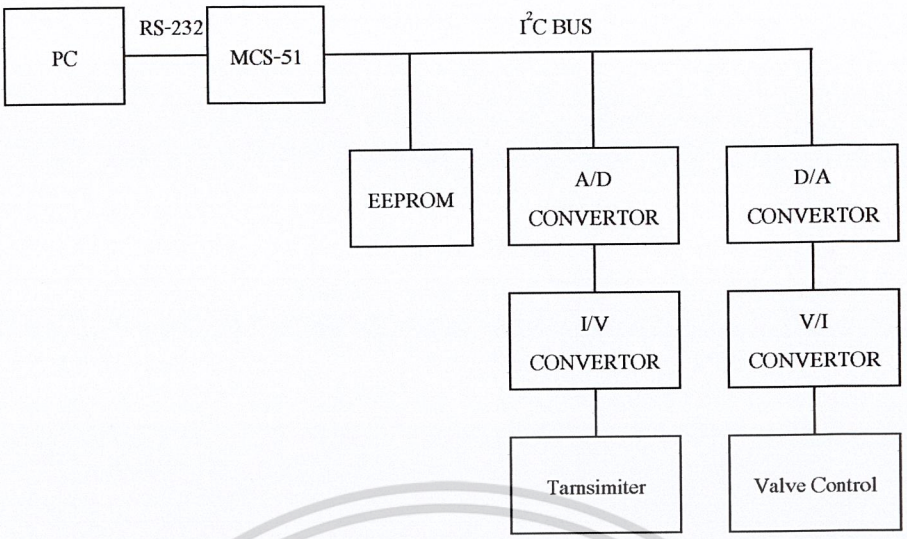
อุปกรณ์ร่วมของระบบฐานบัส I^2Cและโปรแกรม

3.1 วงจรแสดงการทำงานของระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

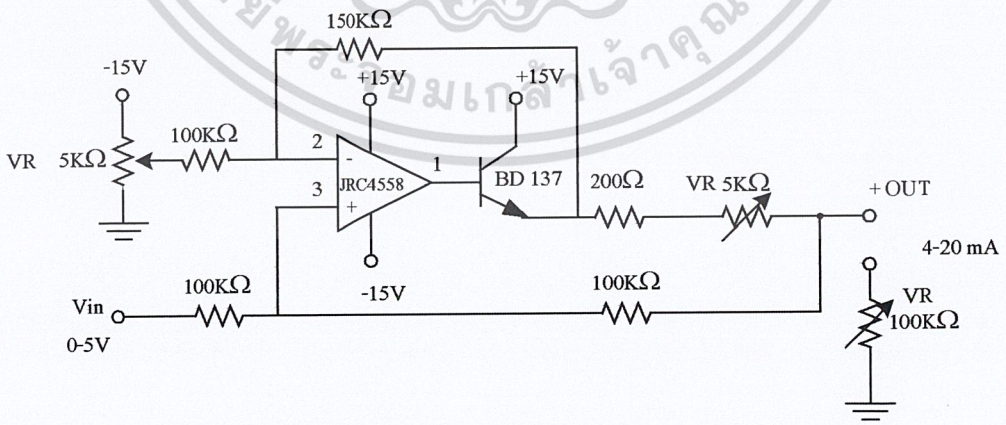
รูปที่ 3.1 วงจรแสดงการทำงานของระบบ



รูปที่ 3.2 แสดงบล็อกไดอะแกรมการทำงานของระบบ

3.2 วงจรเปลี่ยนสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า

นำแรงดันเอาท์พุทของวงจรแหล่งจ่ายไฟกระแสตรงที่มีการรักษาระดับแรงดันให้คงที่ มาใช้งานในวงจรส่วนแรกซึ่งประกอบด้วย ไอซี JRC4558 (เป็นออปแอมป์) ,ความต้านทานค่าต่างๆ ทั้งแบบธรรมดาและปรับค่าได้, ตัวเก็บประจุไฟฟ้า (Capacitor) ทั้งแบบมีขั้วและแบบไม่มีขั้ว โดยในวงจรจะรับอินพุตมาจากไอซี DAC เบอร์ PCF 8591เป็นสัญญาณแรงดันไฟตรง 0-5 โวลต์ สามารถทำการปรับแต่งได้โดยการปรับ ซีโร่ ที่ 5K (Zero) และสแปนที่ 10K (Zero-Span) โดยใช้แหล่งจ่ายสัญญาณแรงดันที่สามารถปรับค่าได้ อย่างต่อเนื่อง



รูปที่ 3.3วงจรเปลี่ยนสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ไมโครคอนโทรลอร์ และอุปกรณ์ร่วมของระบบฐานบัส I^2C

ไมโครคอนโทรลเลอร์ MCS-51

MCS-51เป็นไมโครคอนโทรลเลอร์แบบชิปเดี่ยวที่มีข้อดีเมื่อเทียบกับไมโครคอนโปรเซสเซอร์ขนาด 8 บิตตระกูลอื่น ดังนี้

- มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป (RAM) บรรจุไว้ภายใน 128-256 ไบต์
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในจำนวน4 กิโลไบต์
- มีวงจรตั้งเวลาวงจรนับขนาด 16 บิต 2 ตัว อยู่ภายใน
- มีวงจรรับส่งข้อมูลอนุกรมได้ 2 ทิศทาง
- มีสัญญาณนาฬิกาในตัว
- มีพอร์ทที่สามารถรับหรือส่งข้อมูลได้ 2 ทิศทางจำนวน 4 พอร์ทๆ ละ 8 บิต

นอกจากนี้ MCS-51 ยังมีคุณสมบัติอื่นๆที่น่าสนใจ คือ

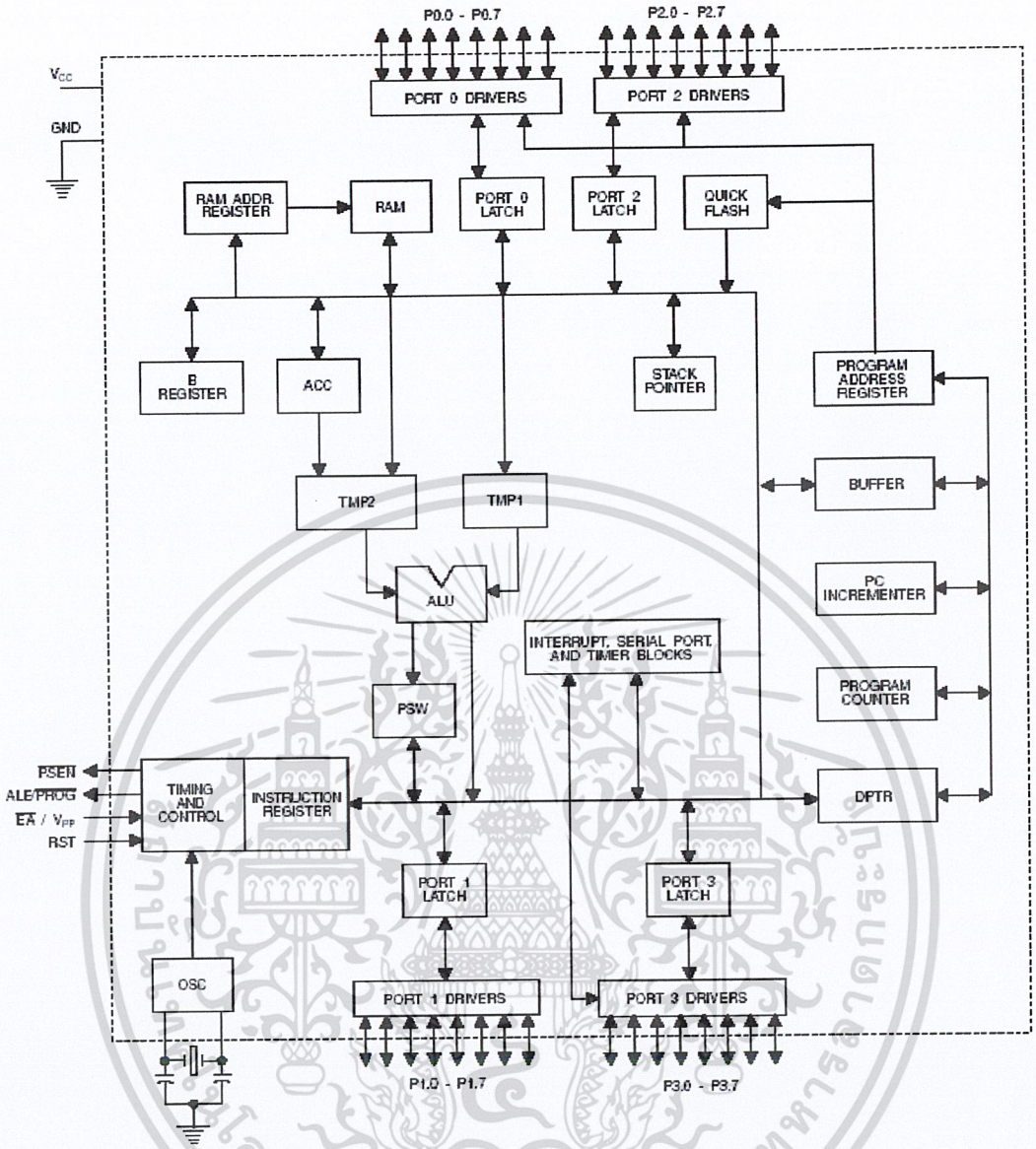
- ต้องการแหล่งจ่ายไฟ 5 V เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในชิป
- สามารถใช้หน่วยความจำ สำหรับโปรแกรม และข้อมูลที่อยู่ภายในชิปได้อย่างละ 64 กิโลไบต์
- มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
- จัดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ได้ 2 ระดับ
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราเร็วในการรับและส่งข้อมูลได้ตั้งแต่ 300-375 กิโลบิตต่อวินาที
- สามารถประมวลผลแบบบูลีนเพื่อใช้ในการควบคุมโดยเฉพาะ
- มีรีจิสเตอร์สำหรับใช้งานเป็นไทม์เมอร์ หรือ เคาน์เตอร์เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิป หรือนับการเปลี่ยนสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิตเพื่อให้การออกแบบโปรแกรมและการควบคุมระบบงานทำได้ง่าย ขึ้น

สถาปัตยกรรมของไมโครคอนโทรลเลอร์

โครงสร้างภานในชิปไมโครคอนโทรลเลอร์ 8051 ชิปเดี่ยวแสดงดังรูปที่ 3.4 ซึ่งอธิบายถึงส่วนย่อยๆ ภายใน 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 3.4 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์

ขาใช้งานต่างๆ ของไมโครคอนโทรเลอร์มีดังนี้

- V_{CC} ขา 40 เป็นขาที่ต้องมีการป้อนไฟเลี้ยง + 5 V เข้าไปเพื่อให้วงจรสามารถทำงานได้

- V_{SS} ขา 20 เป็นขาที่ต้องต่อกับกราวด์ของแหล่งจ่ายไฟ

- RST ขา 9 ขารีเซตนี้จะรีเซตการทำงานของ 8051 ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 ที่ขานี้จะเป็นการรีเซตการทำงาน กลับไปเริ่มการทำงานจากคำสั่งที่มีอยู่ในหน่วยความจำตำแหน่ง 0000

- ALE ขา 30 ใช้เป็นขาส่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตช์ค่าตำแหน่งไบต์ต่ำจากพอร์ท 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอก

- PSEN ขา 29 ใช้ส่งสัญญาณเพื่ออ่านคำสั่ง จากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50252

- XTLAL 1 ขา 19 ใช้ติดต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิเลเตอร์

- XTLAL 2 ขา 18 ใช้ติดต่อคริสตอลภายนอก โดยเป็นเอาท์พุตออกจากวงจรออสซิเลเตอร์

- PORT 0 เป็นพอร์ทขนาด 8 บิตอยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึง 7 ตามลำดับ พอร์ท 0 นี้ใช้ได้ทั้งการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้รับส่งข้อมูลก็ได้ นอกจากนี้ยังใช้งานได้หลายอย่างดังนี้

1. ใช้สำหรับส่งตำแหน่งหน่วยความจำที่ต้องการติดต่อด้วย โดย 8 บิตล่างถูกส่งออกไปทางพอร์ท 0 และ 8 บิตบนถูกส่งออกทางพอร์ท 2
2. ใช้รับส่งข้อมูลกับ Data Memory หรือใช้รับส่งข้อมูลจาก Program Memory
3. ใช้รับส่งข้อมูลออกจากพอร์ท โดยตรง

- PORT 1 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 1 ถึง 8 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับใช้ทำหน้าที่เป็นตัวรับส่งข้อมูลเท่านั้น ไม่สามารถส่งตำแหน่งได้

- PORT 2 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 8 ถึง 21 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับใช้งานเพียง 2 ลักษณะ คือ

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอก ที่ต้องการติดต่อทำงานร่วมกับพอร์ท 0
2. ใช้เป็นพอร์ทรับส่งข้อมูลกับภายนอก

- PORT 3 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 10 ถึง 17 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับ นอกจากจะใช้งานเหมือนพอร์ทอื่นๆ แล้วยังใช้งานอื่นโดยใช้คำสั่งควบคุม ดังนี้

P3.0 (RxD)	เป็นขาที่รับข้อมูลแบบอนุกรม
P3.1(TxD)	เป็นขาที่ส่งข้อมูลแบบอนุกรม
P3.2 (INT0)	ใช้รับสัญญาณขัดจังหวะจากภายนอก
P3.3 (INT1)	ใช้รับสัญญาณขัดจังหวะจากภายใน
P3.4 (T0)	ใช้เป็นขารับสัญญาณให้เคาน์เตอร์ของไทร์เมอร์ 0
P3.5 (T1)	ใช้เป็นขารับสัญญาณให้เคาน์เตอร์ของไทร์เมอร์ 1
P3.6 (WR)	ใช้เป็นขารับสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก
P3.7 (RD)	ใช้เป็นขารับสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก

- PORT 3 ของ MCS-51 ถูกใช้เป็นพอร์ทอนุกรมจะใช้ขาTxDและRxDในการรับส่งข้อมูล โดยขาทั้ง 2 จะอยู่ในพอร์ท 3 คือ P3.1 หรือขา 11 เป็น TxD และ P3.0 เป็นขา 10 เป็น RxD พอร์ทอนุกรม MCS-51 สามารถทำงานเป็นแบบ Full Duplex ได้ คือสามารถรับและส่งข้อมูลในเวลาเดียวกันได้ โดยมีการรับและส่งข้อมูลแบบบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

เอกสารนี้ได้รับการสนับสนุนให้เผยแพร่เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ที่สำคัญในการรับส่งข้อมูลคือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน Special Function Register โดยรีจิสเตอร์ Serial Port Buffer (SBUF) จะอยู่ในตำแหน่ง 99H ถ้าเขียนข้อมูลลงไปในตำแหน่งนี้จะเป็นการส่งข้อมูลออกทางพอร์ทอนุกรม และถ้าอ่านข้อมูลจากตำแหน่งนี้จะเป็นการรับข้อมูลจากทางพอร์ทอนุกรม โดยใน SBUF จะประกอบไปด้วยบัฟเฟอร์ 2 ตัว สำหรับและส่งข้อมูล

สำหรับ Serial Port Control Register (SCON) ซึ่งอยู่ที่ตำแหน่ง 98H จะเป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ รีจิสเตอร์จะทำหน้าที่ควบคุม และบอกสถานะต่างๆของการรับส่งข้อมูลแบบอนุกรม

สำหรับความเร็วของการส่งข้อมูล (Baud Rate) สามารถหาได้จากการหาสัญญาณนาฬิกาที่ใช้กับ MCS-51

ตารางที่ 3.1 Serial Port Control Register

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.7	SM0	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON.6	SM1	9EH	บิตเลือกโหมดการทำงานบิต
SCON.5	SM2	9DH	บิตเลือกโหมดการทำงานบิต
SCON.4	REN	9CH	บิตแฟลกกำหนดยอมให้มีการรับส่งข้อมูล
SCON.3	TB8	9BH	ค่าของบิต 9 สำหรับส่งข้อมูลในโหมด 2 และ 3 สามารถเซต และเคลียร์ได้โดยซอฟแวร์
SCON.2	RB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา
SCON.1	TI	99H	บิตแฟลกแสดงการอินเตอร์รัพท์ภายหลังการส่งข้อมูลออกไป โดยจะเซตเมื่อส่งข้อมูลออกไปหมดแล้ว และสามารถเคลียร์ได้โดยซอฟแวร์
SCON.0	RI	98H	แฟลกแสดงการอินเตอร์รัพท์ภายหลังรับข้อมูลเข้ามา สามารถเคลียร์ได้โดยซอฟแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 แสดงโหมดต่างๆ ของการรับส่งข้อมูลแบบอนุกรม

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

ไทม์เมอร์/เคาน์เตอร์

ไมโครคอนโทรลเลอร์ในตะกูลนี้ มีรีจิสเตอร์พิเศษที่สามารถเลือกใช้งานเป็นไทม์เมอร์หรือเคาน์เตอร์อย่างใดอย่างหนึ่ง รีจิสเตอร์ประเภทนี้มีอยู่ด้วยกัน 2 ตัวแต่ละตัวขนาด 16 บิต เรียกไทม์เมอร์0 และไทม์เมอร์ 1 ตามลำดับ

- ไทม์เมอร์นั้นค่าในรีจิสเตอร์ที่ใช้เป็นไทม์เมอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มขึ้นทุกแมชชีนไซเคิล
- เคาน์เตอร์นั้นค่าในรีจิสเตอร์ที่ใช้เป็นเคาน์เตอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มค่าทีละ1 เมื่อมีการเปลี่ยนสถานะ

ไทม์เมอร์ 0 และไทม์เมอร์ 1

สามารถเลือกการทำงานให้เป็น ไทม์เมอร์ หรือ เคาน์เตอร์ ได้โดยการกำหนดค่าบิตในรีจิสเตอร์ใช้งานเฉพาะ โดยหากบิตนี้มีค่าเป็น 0 หมายถึงเลือกใช้งานเป็นไทม์เมอร์ ถ้าหากบิตนี้เป็น 1 หมายถึงเลือกใช้งานเป็นเคาน์เตอร์

นอกจากจะเลือกการทำงานของรีจิสเตอร์ให้เป็นไทม์เมอร์ หรือเคาน์เตอร์ได้แล้วในแต่ละการทำงานยังมีการทำงานอยู่อีก 4 แบบ ตามความเหมาะสมของการใช้งาน

โหมด 0 จะใช้รีจิสเตอร์ขนาด 8 บิตเป็นตัวนับโดยมีการเพิ่มค่าครั้งละ 1 ทุกครั้งนับสัญญาณได้ครบ 32 ครั้ง โดยในโหมดนี้รีจิสเตอร์ที่ใช้นับเพียง 13 บิต (8 บิตในรีจิสเตอร์ TLx รวมกับ 5 [ตใน THx)

โหมด 1 การทำงานเหมือนกับโหมด 0 เว้นแต่ค่าในรีจิสเตอร์ถูกใช้งานครบทั้ง16บิตนั่นเอง คือไทม์เมอร์หรือเคาน์เตอร์ในโหมดนี้มีขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 2 ในโหมดนี้จะกำหนดรีจิสเตอร์ใช้งานในการนับเพียง 8 บิต (จากรีจิสเตอร์ TLx)ที่มีการโหลดค่าด้วยค่าในรีจิสเตอร์ THx การใช้งานโหมดนี้มีไว้เพื่อสร้างสัญญาณอินเตอร์รัพต์ที่มีคาบเวลาคงที่

โหมด 3 ในโหมดนี้ไทม์เมอร์ 1 จะไม่มีการนับแต่ไทม์เมอร์จะบังคับให้รีจิสเตอร์ TL0 ของไทม์เมอร์ 0 ถูกใช้เป็นไทม์เมอร์เพียงอย่างเดียว การทำงานโหมด 3 มีไว้เพื่อใช้งานที่ต้องการไทม์เมอร์หรือเคาน์เตอร์ขนาด 8 บิตเพิ่มขึ้น

ตารางที่ 3.3 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Timer 0

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

ตารางที่ 3.4 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Counter 0

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

ตารางที่ 3.5 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Timer 1

โหมด	ฟังก์ชั่นไทม์เมอร์ 1	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	00H	80H
1	16 bit Timer	10H	90H
2	8 bit Auto Reload	20H	A0H
3	Two 8 bit Counter	30H	B0H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่างๆ สำหรับ Counter 1

โหมด	ฟังก์ชั่นไทม์เมอร์ 1	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	40H	C0H
1	16 bit Timer	50H	D0H
2	8 bit Auto Reload	60H	E0H
3	Two 8 bit Counter	--	--

รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51

รีจิสเตอร์ A,B และรีจิสเตอร์ใช้งานทั่วไป R0-R7 ซึ่งอยู่ในหน่วยความจำสำหรบเก็บข้อมูลทั่วไปภายในชิปบริเวณ 128 ไบต์แรก รีจิสเตอร์ใช้งานทั่วไป R0-R7 ใน MCS-51 มีอยู่ด้วยกันทั้งหมด 4 กลุ่ม แต่ละกลุ่มประกอบด้วยรีจิสเตอร์จำนวน 8 ตัว (R0-R7) ซึ่งมีชื่อเรียกเหมือนกัน ดั้งนั้นรีจิสเตอร์ใช้งานทั่วไปR0-R7ในMCS-51จึงมีทั้งหมด 32 ตัว ในการทำงานขณะใดๆ รีจิสเตอร์ทั้ง 4 กลุ่ม (R0-R7) จะถูกใช้งานเพียงกลุ่มเดียวเท่านั้น การเลือกใช้งานรีจิสเตอร์ R0-R7 กลุ่มใดกลุ่มหนึ่งใน 4 กลุ่มกระทำโดยการเซตหรือเคลียร์บิต RS0,RS1 ในรีจิสเตอร์ใช้งานเฉพาะ PSW

รีจิสเตอร์ฟังก์ชั่นพิเศษใน MCS-51

ใน MCS-51 รีจิสเตอร์จะใช้หน่วยความจำ RAM ภายในชิป โดยส่วนหนึ่งเป็นรีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register: SFR) ซึ่งมีทั้งหมด 12 ตัว โดยรีจิสเตอร์ฟังก์ชั่นพิเศษต่างๆ จะเริ่มที่หน่วยความจำตั้งแต่ 80H-FFH ซึ่งมีทั้งหมด 128 ตำแหน่ง แต่จะเป็นรีจิสเตอร์ฟังก์ชั่นพิเศษเพียง 21 ตำแหน่ง แต่ถ้าเป็น AT89C51/AT89C52 จะใช้ 26 ตำแหน่งหรือมี SFR 26 ตัว

ตารางที่ 3.7 รีจิสเตอร์ใช้เฉพาะ PSW (Program Status Word) เข้าถึงข้อมูลได้ในระดับบิต

บิต	ชื่อบิต	ตำแหน่ง	ความหมาย
PSW.7	CY	D7H	Carry Flag
PSW.6	AC	D6H	Auxiliary Flag
PSW.5	F0	D5H	Flag 0
PSW.4	RS1	D4H	บิตสำหรับเลือก รีจิสเตอร์ แบงค์ 1
PSW.3	RS0	D3H	บิตสำหรับเลือก รีจิสเตอร์ แบงค์ 0 00 = bank 0;Address 00H-07H 01 = bank 0;Address 08H-0FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต	ชื่อบิต	ตำแหน่ง	ความหมาย
			10 = bank 0;Address 10H-17H 11 = bank 0;Address 18H-1FH
PSW.2	OV	D2H	OverFlow Flag
PSW.1	-	D1H	Reserved
PSW.0	P	D0H	Even Parity Flag

3.4 การเชื่อมต่อกับสัญญาณอนาลอกผ่านพอร์ตอนุกรม

ปกติแล้วขอมูลในการติต่อกับพอร์ตอนุกรมของคอมพิวเตอร์นั้น จะเป็นสัญญาณดิจิตอลทั้งสิ้น แต่เมื่อนำมาเชื่อมต่อกับอุปกรณ์ภายนอกแล้ว ย่อมต้องเชื่อมต่อและประมวนผลสัญญาณอะนาลอกด้วย ในการเชื่อมต่อสัญญาณอะนาลอกต้องใช้ไอซีพิเศษที่ทำหน้าที่ในการแปลงสัญญาณอะนาลอกเป็นดิจิตอลหรือที่เรียกกันว่าไอซี ADC ในการทดลองต่อไปจะเป็นการใช้ไอซี ADC ร่วมกับวงจร เพื่อเชื่อมต่อสัญญาณอะนาลอกเข้ากับพอร์ตอนุกรม โดยใช้ไอซี ADC ที่ใช้เป็นเบอร์ PCF8591 ซึ่งมีรูปแบบการเชื่อมต่อแบบบัส I^2C

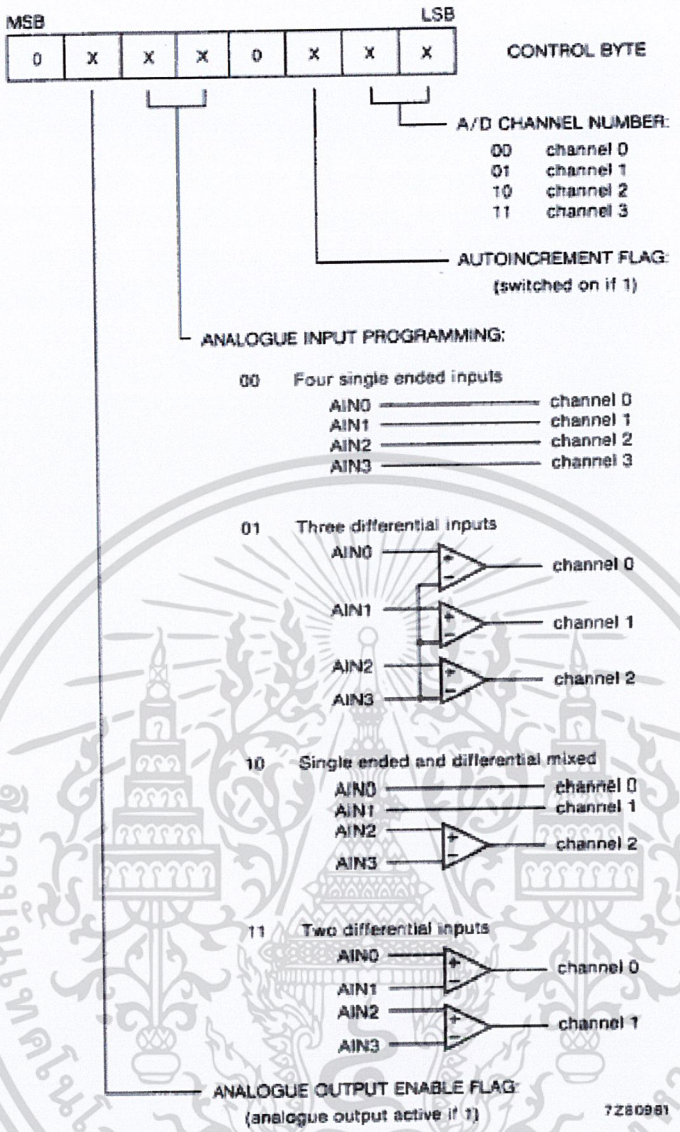
การปรับค่าและการส่งค่าของ IC PCF8591

รูปแบบข้อมูลกำหนดแอดเดรสที่ใช้การอ้างถึงแบบ 7 บิต ข้อมูลกำหนดแอดเดรส 4 บิตบนจะเป็นค่าแอดเดรสของอุปกรณ์ตัวนั้นๆที่กำหนดมาจากผู้ผลิต ผู้ใช้งานไม่สามารถเปลี่ยนแปลงได้ สำหรับ IC PCF8591 จะมีค่าเท่ากับ 1001_2 ข้อมูล 3 บิตถัดมา จะเป็นค่าของแอดเดรสที่ผู้ใช้งานสามารถกำหนดได้ ทางฮาร์ดแวร์ ส่วนบิต LSB ใช้ในการกำหนดว่าต้องการอ่านหรือเขียนข้อมูล ถ้าเป็น 0 หมายถึงต้องการเขียนข้อมูล ถ้าเป็น 1 หมายถึงต้องการอ่านข้อมูล

ข้อมูลควบคุม

หลังจากส่งข้อมูลกำหนดแอดเดรสแล้วต้องส่งข้อมูลควบคุมตามไปด้วย เพื่อกำหนดคุณสมบัติของ IC PCF8591 รายละเอียดแสดงดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงรายละเอียดข้อมูลควบคุมที่เขียนลงในรีจิสเตอร์ควบคุมภายในไอซี PCF8591

การอ่านข้อมูลอินพุตอะนาลอกจาก PCF8591มีลำดับขั้นตอนดังต่อไปนี้

1. **ส่งสัญญาณ Start**
2. **ส่งข้อมูลกำหนดแอดเดรส** บิตที่กำหนดแอดเดรสสามารถกำหนดได้ว่าจะติดต่อกับ IC PCF8591ตัวใด เช่น ให้ขา A0,A1,A2 มีลอจิกเป็น 000 และให้ทำงานในโหมดเขียนข้อมูล คือ LSB มีลอจิกเป็น "0" จะได้ข้อมูลกำหนดแอดเดรสเป็น 10010000_2 หรือเท่ากับ 90H
3. **รอสัญญาณ ACK หรือรอรับการตอบกลับจาก IC PCF8591**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ส่งข้อมูลควบคุม IC PCF8591ตัวอย่างเช่นต้องการอินาเบิลอะนาลอกเอาท์พุต, กำหนดให้อินพุตอะนาลอกทำงานในโหมดซิงเกิล, กำหนดให้ใช้การอ่านข้อมูลแบบต่อเนื่อง และเริ่มอ่านข้อมูลจากช่องที่ 1 จะได้ข้อมูลควบคุมดังนี้ 01000101_2
5. รอรับสัญญาณ ACK จาก PCF8591
6. ส่งสัญญาณ Stop
7. ส่งสัญญาณ Start
8. ส่งข้อมูลกำหนดแอดเดรสอีกครั้ง โดยครั้งนี้กำหนดให้เป็นโหมดอ่านข้อมูลคือให้ LSB มีลอจิกเป็น “1”
9. รอรับสัญญาณ ACK จาก PCF8591
10. อ่านค่าจากขาอินพุตของวงจงแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 1 โดยโปรแกรมย่อยที่ใช้ในการรับค่า
11. ส่งสัญญาณ MACK (Master ACK) ไปยัง PCF8591
12. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 2
13. ส่งสัญญาณ MACK (Master ACK) ไปยัง PCF8591
14. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 3
15. ส่งสัญญาณ MACK (Master ACK) ไปยัง PCF8591
16. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 4
17. รอรับสัญญาณ ACK จาก PCF8591
18. ส่งสัญญาณ Stop

การเขียนข้อมูลไปยังวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกของ PCF8591

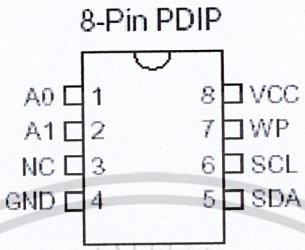
มีขั้นตอนดังนี้

1. ทำการส่งสัญญาณ Start
2. ส่งข้อมูลกำหนดแอดเดรสโดยใช้ทำงานในโหมดเขียนข้อมูลคือ บิต LSB มีลอจิก “0”
3. รอรับสัญญาณ ACK
4. ส่งข้อมูลควบคุม 01000100_2 หรือ 44H ไปยัง PCF8591
5. รอรับสัญญาณ ACK
6. ส่งข้อมูลไปยังเอาท์พุตอะนาลอก โดยค่าที่ส่งออกไปจะต้องมีค่าอยู่ระหว่าง 0-255
7. รอรับสัญญาณ ACK
8. ส่งสัญญาณ Stop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 EEPROM 2402

IC 24LC02 เป็น EEPROM ที่มีขนาด 2 Kbyte ที่มีการสื่อสารแบบ I^2C สามารถต่อร่วมกับอุปกรณ์แบบ I^2C ตัวอื่น ๆ ได้ในสายสัญญาณคู่เดียวกัน แต่ไม่สามารถต่อ 24LC02 ในสายสัญญาณคู่เดียวกัน เพราะ 24LC02 ตำแหน่งที่คงที่เพียงตำแหน่งเดียว คือ ตำแหน่ง 00H โครงสร้างภายนอกและขาสัญญาณแสดงดังรูปด้านล่าง

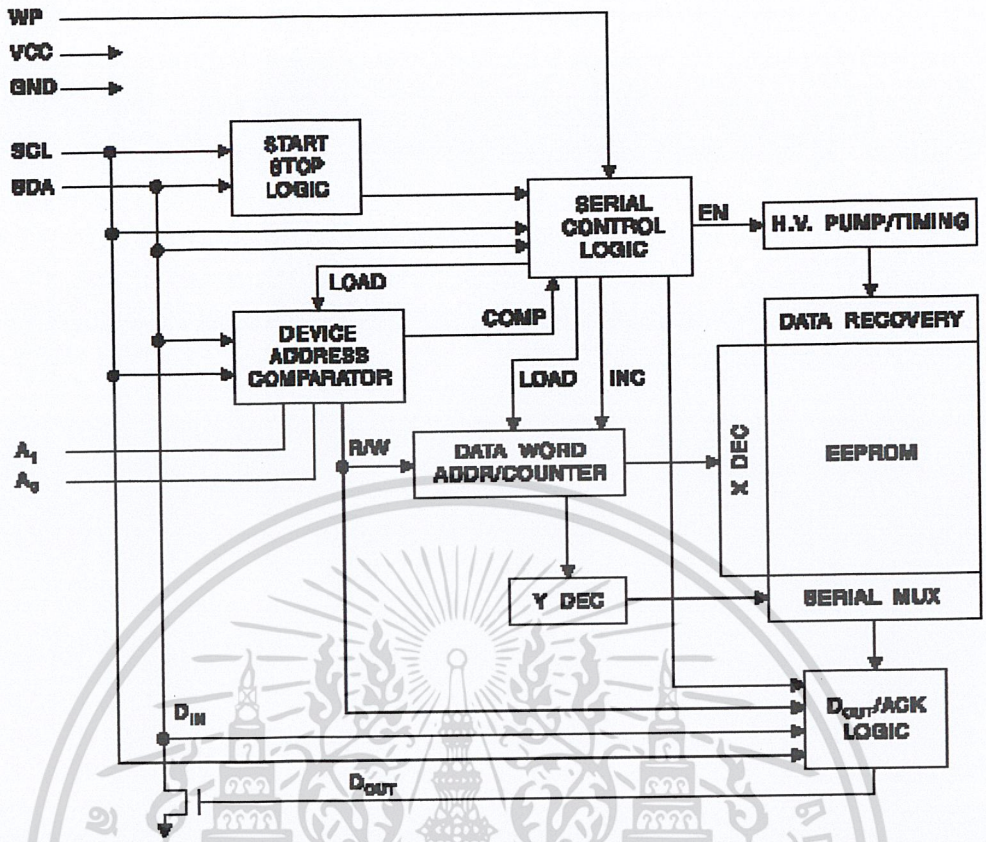


รูปที่3.6 แสดงโครงสร้างขาใช้งานของ EEPROM 24LC02

ตารางที่ 3.8 แสดงตำแหน่งและหน้าที่ของขาใน 24LC02

ตำแหน่ง	สัญลักษณ์	รายละเอียด
1	A0	ขากำหนดตำแหน่ง A0
2	A1	ขากำหนดตำแหน่งA1
3	A2	ขากำหนดตำแหน่ง A2
4	Vss	กราวด์
5	SDA	สัญญาณข้อมูล I^2C
6	SCl	สัญญาณนาฬิกา
7	WP	ขาป้องกันการเขียน
8	Vcc	แปลงจ่ายไฟ 5 โวลท์

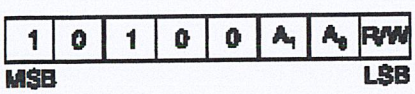
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.7 แสดงโครงสร้างภายใน EEOROM 24LC02

Device Addressing

Control Byte จะเป็น Byte แรกต่อจากบิต START Control Byte จะประกอบด้วย Control Code 4 bit จะอยู่ในบิตที่ 7-4 24LC02มีค่าเป็น1010 และต่อมาอีก 3 Bit จะเป็น (B2,B1,B0) เป็นบิตที่ใช้เลือก Bank ของหน่วยความจำ บิตที่ 0 เป็นบิต RW มีหน้าที่ กำหนดเงื่อนไขว่า จะเขียนหรืออ่านข้อมูลจาก 24LC02



รูปที่3.8 แสดงลักษณะการ Device Address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

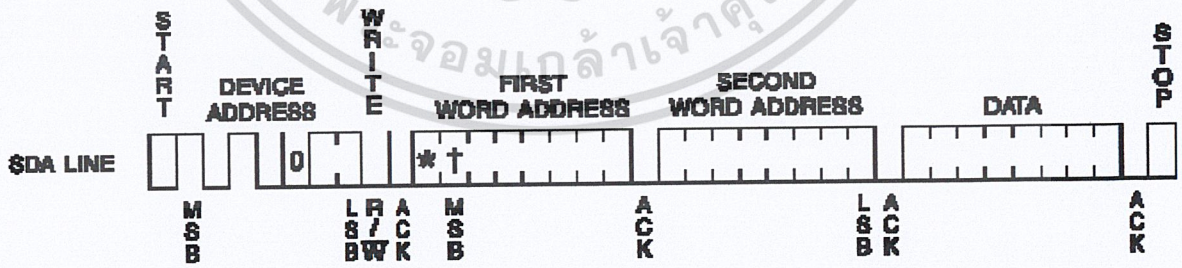
การเขียนข้อมูล

การเขียนข้อมูลลงในหน่วยความจำของ24LC02สามารถทำได้โดยเริ่มจากส่งเงื่อนไข Start และตามด้วย Control Byte บิต R/W ใน Control Byte จะต้องกำหนดค่าให้เป็น 0 ซึ่งการเขียนข้อมูลลงในหน่วยความจำของ 24LC02 สามารถแบ่งออกได้เป็นการเขียนข้อมูลครั้งละ 1 ไบต์ และการเขียนข้อมูลครั้งละ 1 Page

การเขียนข้อมูลแบบ Byte

การเขียนข้อมูลแบบไบต์ หรือการเขียนข้อมูลครั้งละ 1 ไบต์ ลงในหน่วยความจำของ 24LC02 สามารถทำได้โดยเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ส่งเงื่อนไข Start ให้กับ 24LC02 และตามด้วย Control Byte โดยกำหนดค่า B2,B1,B0 ใน Control Byte ตามค่าตำแหน่งหน่วยความจำที่ต้องการเขียนข้อมูล ส่วนบิต R/W จะถูกกำหนดให้เป็น 0 หลังจากส่ง Control Byte จะต้องรอรับค่าบิต ACKจาก 24LC02 ตามกลับออกมา ซึ่งบิต ACK ที่ตอบกลับออกมาจะมีค่าเป็น 0 หลังจากได้รับบิต ACK ที่ตอบกลับออกมาแล้ว ให้ส่งไบต์ตำแหน่งของหน่วยความจำที่ต้องการเขียนข้อมูลออกไปอีก 1 ไบต์ หลังจากนั้นให้รอรับบิต ACK ที่จะตอบกลับออกมาเป็นลอจิก 0 จาก 24LC02 เมื่อได้รับบิต ACK ตอบกลับออกมาแล้ว ให้ส่งค่าข้อมูลที่ต้องการเขียนลงในตำแหน่งหน่วยความจำออกไป และรอรับค่าบิต ACK จาก 24LC02 หลังจากได้รับบิต ACK ตอบกลับออกมาแล้วให้ส่งเงื่อนไขการ Stop เพื่อยกเลิกการติดต่อ

ถ้าขั้นตอนใดขั้นตอนหนึ่งไม่มีบิต ACK ตอบกลับออกมาให้เริ่มส่งเงื่อนไข Start ใหม่ อีกครั้งและเริ่มขบวนตั้งแต่ต้นใหม่



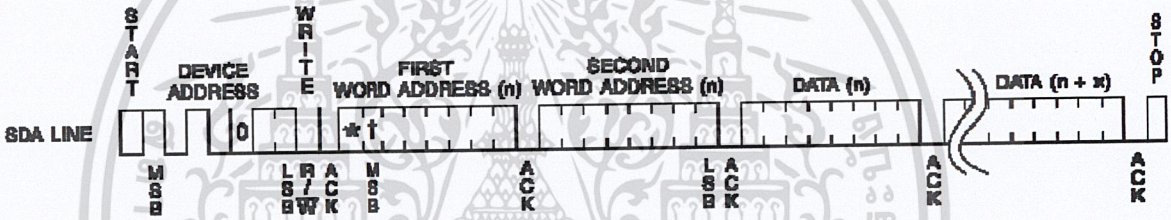
รูปที่ 3.9 แสดง Timing Diagram ของการเขียนข้อมูลแบบ Byte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนข้อมูลแบบ Page

การเขียนข้อมูลแบบ Page หรือการเขียนข้อมูลครั้งละ 1 Page ภายใน 24LC02 แบ่งการเก็บข้อมูลไว้เป็น Page โดยแต่ละ Page สามารถเก็บข้อมูลได้ 16 ไบต์ ดังนั้นการเขียนข้อมูลครั้งละ Page ก็คือการเขียนข้อมูลครั้งละ 16 ไบต์นั่นเอง

การเขียนข้อมูลครั้งละ Page จะเหมือนกับการเขียนข้อมูลครั้งละ 1 ไบต์ จะต้องส่งเงื่อนไข Start ออกไปก่อน แล้วตามด้วย Control Byte หลังจากนั้นรอรับ ACK จาก 24LC02 หลังจากได้ค่า ACK แล้วให้ส่งตำแหน่งหน่วยความจำออกไป และรอรับบิต ACK จาก 24LC02 จึงจะส่งข้อมูลไบต์ต่อไปจนครบ 16 ไบต์ โดยข้อมูลทั้ง 16 ไบต์ จะถูกเก็บเรียงกันในหน่วยความจำเริ่มต้นตั้งแต่ที่ตำแหน่งที่ถูกส่งออกไปใน Byte Word Address หลังจากที่ได้รับบิต ACK หลังจากส่งข้อมูลออกไปครบทั้ง 16 ไบต์ ให้ส่งบิต Stop เพื่อยกเลิกการเขียนข้อมูล



รูปที่3.10 แสดง Timing Diagram ของการอ่านข้อมูลแบบ Page

การอ่านข้อมูล

ขั้นตอนการอ่านข้อมูลจะเหมือนกับขั้นตอนการเขียนข้อมูลจะแตกต่างกันที่บิท R/W จะต้องถูกเซ็ต เป็น 1 ซึ่งสามารถแบ่งลักษณะการอ่านข้อมูลได้ 3 แบบ

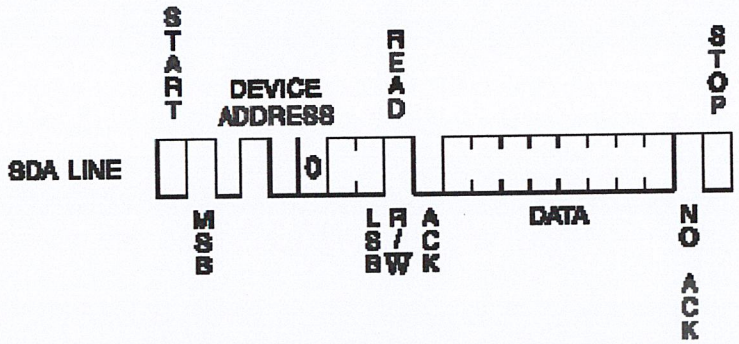
การอ่านข้อมูลในตำแหน่งปัจจุบัน (Current Address Read)

ภายใน24LC02จะมีตัวชี้ตำแหน่งของหน่วยความจำ ทุกครั้งที่เราส่งตำแหน่งให้กับ24LC02 นั้นค่าของตัวชี้ จะถูกเปลี่ยนตามค่าของตำแหน่งที่ถูกส่งเข้าไป และทุกครั้งที่มีการเขียนเข้าใน 24LC02 หรืออ่านข้อมูลออกจาก24LC02ในแต่ละไบต์ค่าของตัวชี้จะเพิ่มขึ้นเป็น 1 ค่าโดยอัตโนมัติ

ดังนั้นการอ่านข้อมูลในตำแหน่งปัจจุบันคือการอ่านข้อมูลจาก 24LC02 ณ ตำแหน่งที่ตัวชี้อยู่ ซึ่งสามารถทำได้โดยการส่งเงื่อนไข Start และตามด้วย Control Byte โดยกำหนดให้บิต R/W มีค่าเป็น 1 หลังจากนั้นให้รอรับบิต ACK หลังจากที่ได้รับบิต ACK แล้วให้อ่านข้อมูลกลับออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก 24LC02 จำนวน 1 ไบต์ หลังจากนั้นให้ส่งบิต NO ACK ซึ่งก็คือส่งลอจิก 1 และส่งเงื่อนไข Stop ให้กับ 24LC02

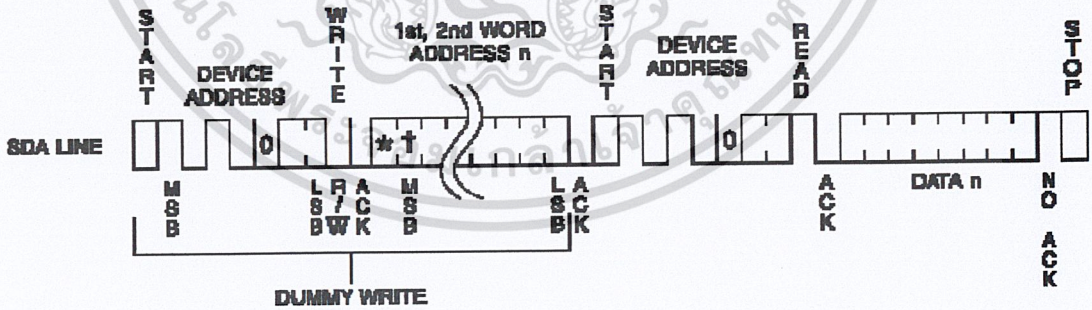


รูปที่3.11 การอ่านข้อมูลในตำแหน่งปัจจุบัน

การอ่านข้อมูลแบบสุ่ม (Random Read)

การอ่านข้อมูลแบบสุ่มจะเป็นการอ่านค่าตาม Address ที่ระบุที่ตำแหน่งที่ต้องการอ่าน โดยมีขั้นตอนการอ่านค่าข้อมูลในหน่วยความจำดังนี้

ส่งเงื่อนไข Start และตามด้วย Control Byte โดยให้บิต R/W เป็น 0 และ Work Address ซึ่งเป็นตำแหน่งของหน่วยความจำที่ต้องการอ่านข้อมูล หลังจากนั้นให้รอรับบิต ACK เมื่อได้รับบิต ACK แล้วจึงอ่านข้อมูลออกมา 1 ไบต์ ซึ่งเป็นข้อมูลตำแหน่งเดียวกับตัวชี้ ชี้อยู่ หลังจากได้ข้อมูลครบทั้ง 8 บิทแล้วให้ส่งบิท NO ACK และบิท Stop ให้กับ 24LC02



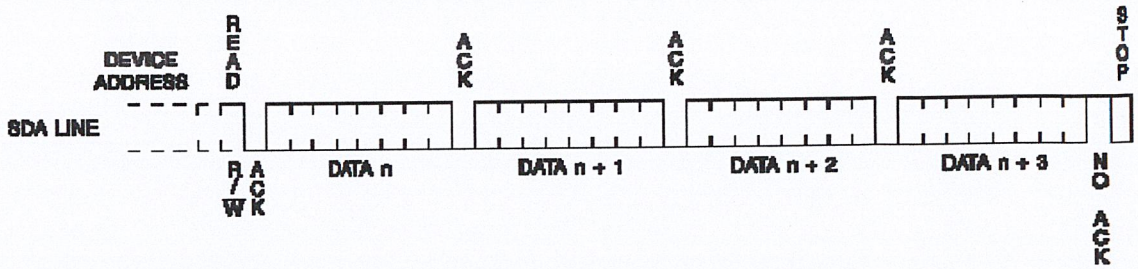
รูปที่ 3.12 แสดง Timing Diagram ของการอ่านข้อมูลแบบสุ่ม

การอ่านข้อมูลเป็นลำดับ (Sequential Read)

การอ่านข้อมูลในหน่วยความจำแบบลำดับ มีขั้นตอนการอ่านข้อมูลในลำดับเดียวกันกับการอ่านข้อมูลแบบสุ่ม จะแตกต่างที่การอ่านข้อมูลแบบสุ่มจะอ่านข้อมูลออกมาเพียงไบต์เดียว แต่การอ่านข้อมูลแบบลำดับจะอ่านข้อมูลออกมา X จนกระทั่งส่งบิต NO ACK และบิต Stop ออกไป

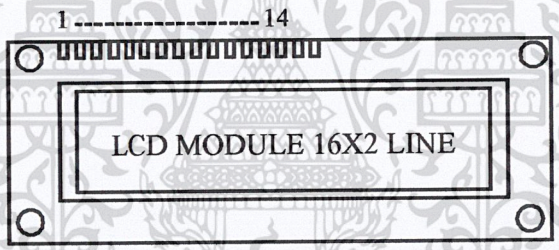
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ซึ่งทุกครั้งที่มีการอ่านข้อมูลออกจาก 24LC02 นั้น ค่าของตัวชี้ที่อยู่ใน 24LC02 จะเพิ่มค่าขึ้น 1 ค่า เพราะฉะนั้นข้อมูลที่อ่านออกมาได้จึงเป็นข้อมูลที่อยู่ต่อเป็นลำดับต่อ ๆ กันออกไป



รูปที่ 3.13 แสดง Timing Diagram ของการอ่านข้อมูลแบบลำดับ

3.6 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด(LCD 16x2)



รูปที่ 3.14 แสดงรูปร่างและการจัดขาโมดูล LCD แบบอักขระ

โมดูล LCD ขนาด 16x2 มีขาต่อใช้งานทั้งสิ้น 14 ขามีการจักขาดังรูปที่ 3.6 สำหรับรายระเอียดการทำงานของแต่ละขามีดังนี้

V_{SS} (ขา1) : ต่อลงกราวด์

V_{DD} (ขา2) : ต่อไฟเลี้ยง +5 V

V_O (ขา3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลทีททำการประมงลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยที่ขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดง

R/W (ขา5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับโมดูล LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

E (ขา6) : เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเอเบิลโมดูล LCD ให้ทำงาน

D0-D7 (ขา7) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอก ขนาด 8 บิต

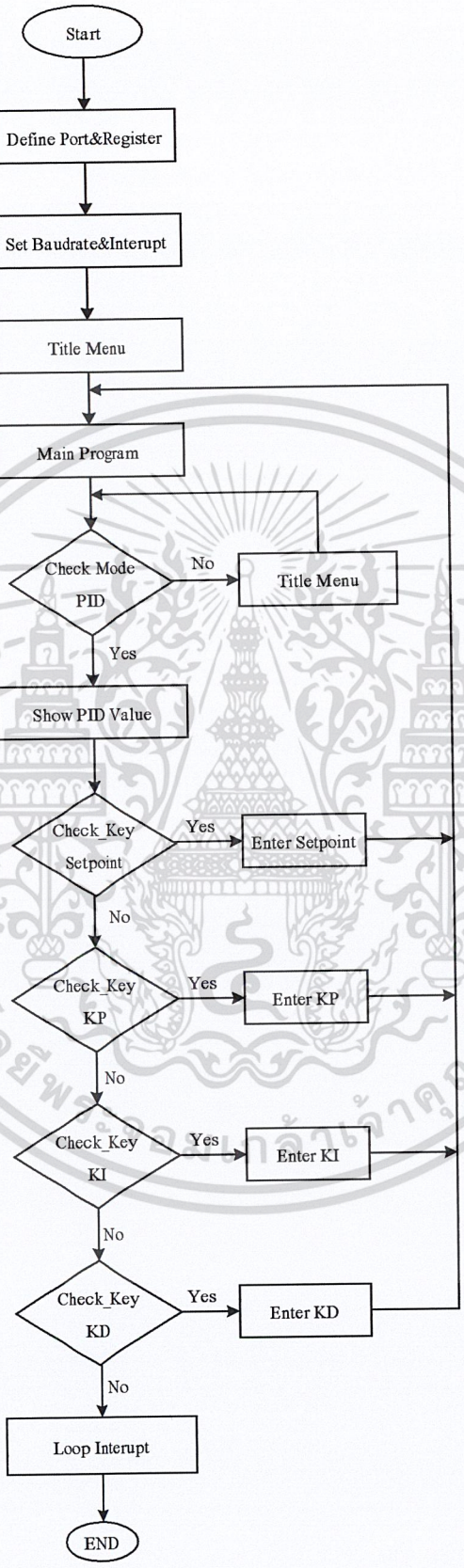
3.7 การทำงานของ PID

หน่วยประมวลผลหลักที่ใช้ไมโครคอนโทรเลอร์ ที่ได้ออกแบบเพื่อควบคุมกระบวนการ รับค่าจาก เอาท์พุต ของ Process ที่จะทำการควบคุมสั่งมายัง อินพุต ของเครื่องควบคุมและของเครื่องควบคุมและทำการประมวลผลข้อมูล Input ตามโปรแกรมที่ผู้ใช้เครื่องควบคุมได้โปรแกรมไว้ และส่งข้อมูลที่ได้จากการคำนวณของเครื่องควบคุมไปยัง เอาท์พุต เพื่อไปควบคุมกระบวนการดังกล่าว

หน่วย อินพุต ของเครื่องควบคุมนั้นจะรับสัญญาณ 4-20 mA ที่มาจาก Process แล้วเปลี่ยนเป็นสัญญาณ 0-5 Vdc โดยมี IC RCV420 ที่ใช้ในการเปลี่ยนสัญญาณดังกล่าว ส่วนเอาท์พุตของเครื่องควบคุมนั้นจะส่งสัญญาณไปควบคุมกระบวนการตามที่ผู้ใช้ได้โปรแกรมไว้ ซึ่งสัญญาณเอาท์พุตที่ได้จากเครื่องควบคุมนั้นจะเป็นสัญญาณแรงดัน 0-5 Vdc จึงต้องทำการเปลี่ยนสัญญาณดังกล่าวเป็นสัญญาณ 4-20 mA โดยใช้วงจร V to I เพื่อทำการเปลี่ยนสัญญาณที่จะไปควบคุมกระบวนการต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

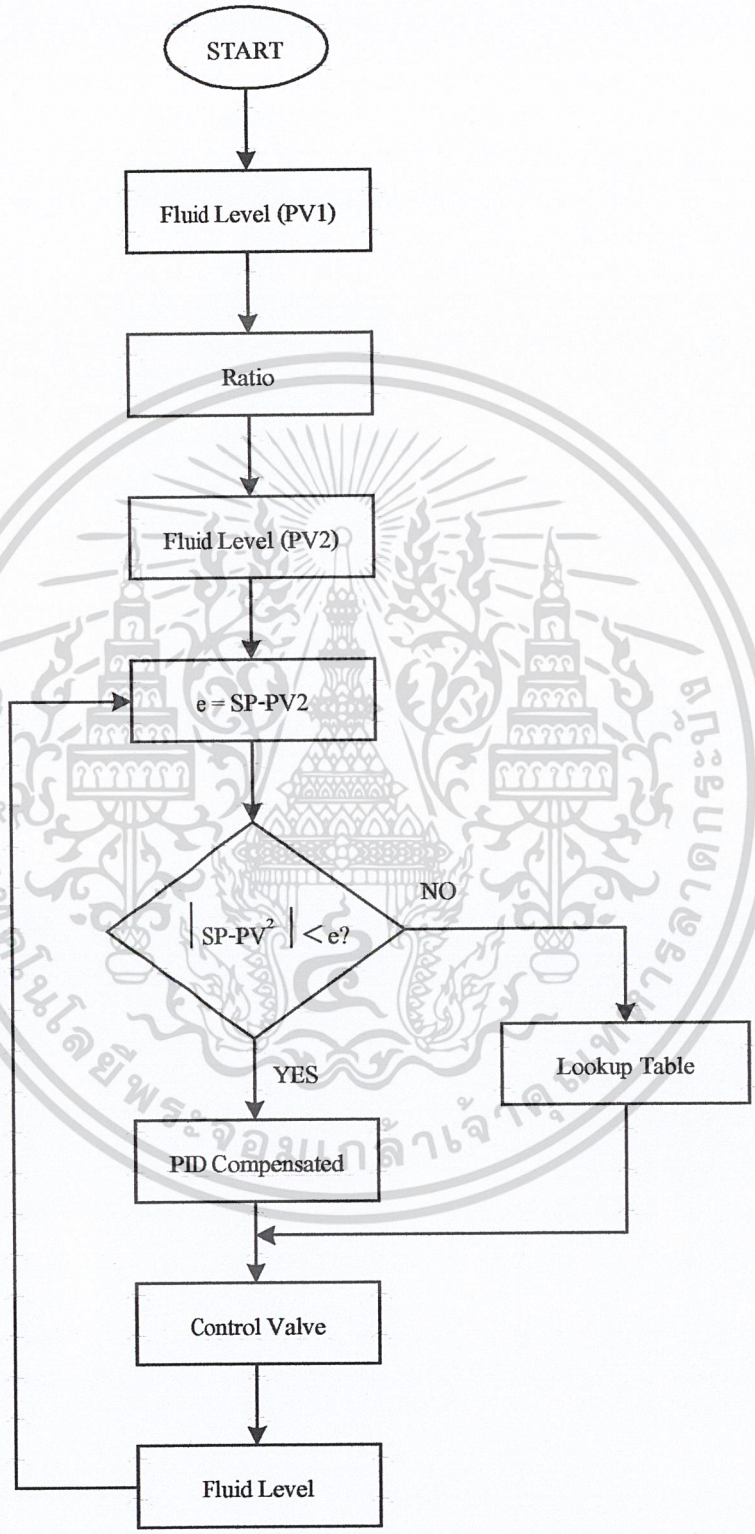
3.8 Main โปรแกรมหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

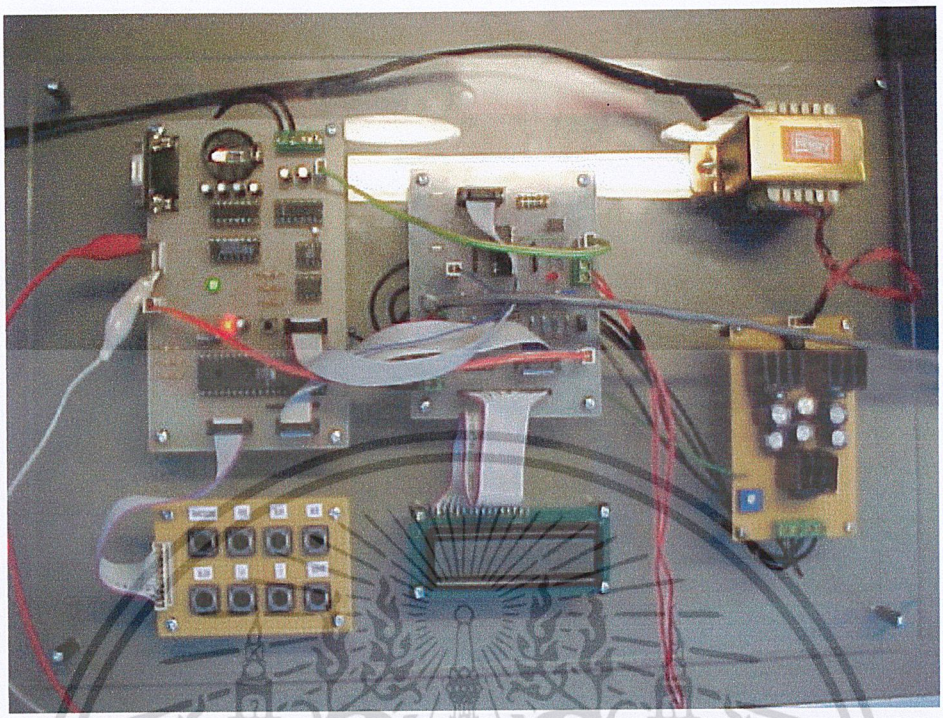
รูปที่ 3.15 FLOW CHARTแสดงโปรแกรมหลัก

3.9 โปรแกรมย่อย PID

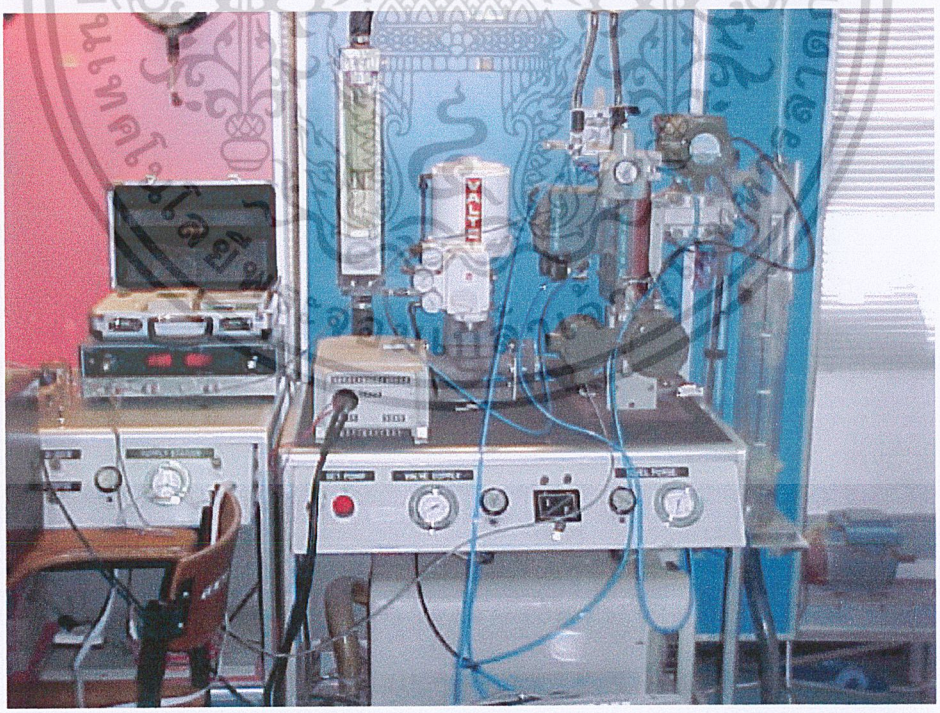


รูปที่ 3.16 FLOW CHARTแสดงโปรแกรมย่อย PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แสดงแผงวงจรของ PC – Interfacing With I^2C Bus Microcontroller Application



รูปที่ 3.18 แสดงกระบวนการจำลองควบคุมระดับของเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

โดยทำการเตรียมเครื่องมือที่ใช้ในการทดลองแต่ละกระบวนการคือ กระบวนการจำลองการควบคุมระดับของของเหลว

การทดลองที่ 1

แสดงการควบคุมกระบวนการจำลองควบคุมระดับโดยผ่าน Computer และแสดงผลผ่านทาง Computer

อุปกรณ์ที่ใช้ในการทดลอง

1. เครื่อง Computer พร้อมโปรแกรม Visual Basic พร้อมสายต่อพวง
2. บอร์ด PC – Interfacing With I^2C Bus Microcontroller Application
3. กระบวนการจำลองควบคุมระดับของของเหลว

ขั้นตอนการทดลอง

1. นำเครื่อง Computer ต่อสายสัญญาณเข้ากับเข้ากับพอร์ท RS323 บอร์ดไมโครคอนโทรลเลอร์ ที่สร้างขึ้นมา เชื่อมต่อกับกระบวนการจำลองควบคุมระดับ
2. กำหนดค่าพารามิเตอร์ที่เหมาะสมกับกระบวนการ

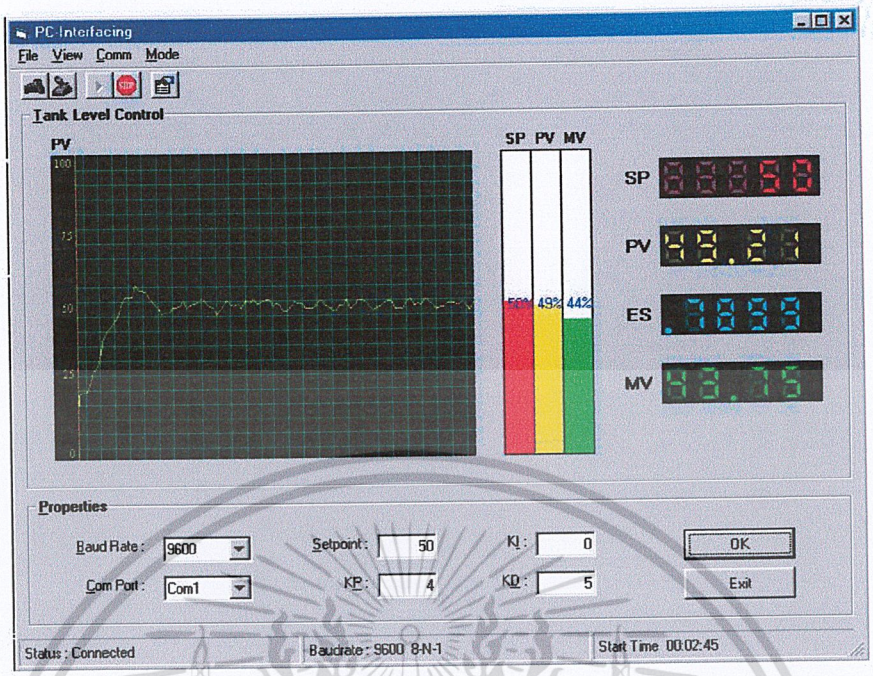
   SP = 50%

   KP = 4

   KI = 0

   KD = 5
3. ทำการทดลองตามโปรแกรมโดยใช้โปรแกรม Visual Basic เป็นตัวสังเกตการณ์เปลี่ยนแปลงดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 กราฟที่ได้จากที่ SP = 50%

ทำการเปลี่ยนค่าพารามิเตอร์

SP = 75%

KP = 2

KI = 0

KD = 5

ผลที่ได้จากการเปลี่ยนแปลงค่าพารามิเตอร์แสดงดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 กราฟที่ได้จากที่ SP = 75%

ทำการเปลี่ยนค่า SP ทั้งหมด 3 ค่า

SP = 20%

SP = 50%

SP = 75%

KP = 2

KI = 0

KD = 5

ผลที่ได้จากการเปลี่ยนแปลงค่าพารามิเตอร์แสดงดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 กราฟที่ได้จากการสั่งค่า SP เป็น Step

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่2

แสดงการควบคุมกระบวนการจำลองควบคุมระดับโดย Mode Stand Alone

อุปกรณ์ที่ใช้ในการทดลอง

1. บอร์ด PC – Interfacing With I^2C Bus Microcontroller Application
2. กระบวนการจำลองควบคุมระดับของของเหลว
3. เครื่องบันทึก Recorder

ขั้นตอนการทดลอง

1. นำบอร์ด PC – Interfacing With I^2C Bus Microcontroller Application ต่อเข้ากับกระบวนการจำลองควบคุมระดับ โดยผ่านทางพอร์ท RS232
2. กำหนดค่าพารามิเตอร์ผ่านทาง Key Switch

   SP = 20% KD = 5

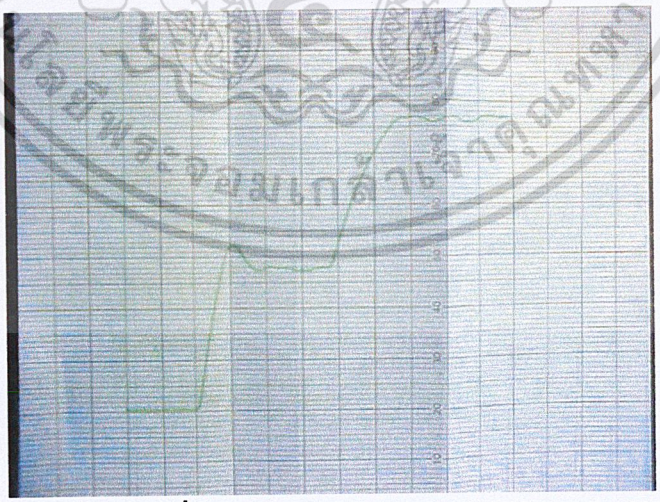
   SP = 50%

   SP = 75%

   KP = 4

   KI = 0

3.ทำการทดลองตามโปรแกรมโดยใช้ Recorder ทำการเปลี่ยนค่าพารามิเตอร์ผลที่ได้จากการเปลี่ยนแปลงค่าพารามิเตอร์แสดงดังรูปที่ 4.4



รูปที่ 4.4 กราฟจากเครื่อง Recorderแสดงสภาวะการทำงานแต่ละ Step

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากการทดลองทั้ง 2 จะพบว่าค่าที่ได้ออกมาสามารถควบคุมให้เข้าค่าเป้าหมายได้ ไม่ว่าจะเป็นMode Stand Alone หรือที่ต่อพ่วงกับคอมพิวเตอร์ และกราฟที่แสดงออกมาทางคอมพิวเตอร์ก็ให้ค่าออกมาตรงกับ กราฟที่ออกมาจากเครื่อง Recorder การทดลองทั้งสองนี้ทำแสดงให้เห็นว่าบอร์ดจำลองการทำงาน ของระบบบัส I^2C สามารถที่จะต่อพวงเข้ากับระบบจำลองควบคุมกระบวนการได้และสามารถควบคุมได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปหลักการทำงานและการวิจารณ์

5.1 สรุปหลักการทำงานของระบบ

จากการทดลองโดยนำเครื่อง PC – Interfacing With I^2C Bus Microcontroller Application มาต่อกับกระบวนการจำลองการควบคุมการระดับของของเหลวนั้นพบว่า การควบคุมแบบ PID นั้นสามารถควบคุมค่าตัวแปรของกระบวนการให้เข้าเป้าหมายได้ ซึ่งสามารถแสดงผลการทดลองเป็นกราฟออกมาทางคอมพิวเตอร์ผ่านทางโปรแกรม Visual Basic และยังสามารถที่จะนำไปใช้ใน Mode Stand Alone ซึ่งจะรับค่าพารามิเตอร์ต่างจากทาง Key Switch มาแสดงผลทางจอ LCD และที่จอ LCD ก็แสดงค่า PV ที่เปลี่ยนแปลงตามกระบวนการ

5.2 วิจารณ์

การเขียนโปรแกรมในปริญญานิพนธ์จะเป็นการเขียนโดยใช้ภาษาแอสเซมบี้ ซึ่งภาษาแอส-เซมบี้นั้นเป็นภาษาระดับต่ำทำให้การควบคุมค่าทาง PID ที่เป็นจุดทศนิยมไม่สามารถทำได้จึงทำให้การฟที่ออกมีการแกว่งอยู่ตลอดเวลาแต่ก็อยู่ในค่าเป้าหมาย ค่าที่ได้จึงหยาบกว่าการเขียนโปรแกรมด้วยภาษาอื่น ดังนั้นจึงเกิดความยุ่งยากในการเขียนโปรแกรม PID เป็นอย่างมากเนื่องจากส่วนของ PID นั้นจะต้องการค่าที่ระเอียดเป็นทศนิยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กฤษดา ใจเย็น, อรรถพล บุญยะโภค, ธัมวัฒน์ ลิ้มพรจิตรวิไล, "เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม", (C) INNOVATIVE EXPERIMENT CO.,LTD.

วรพจน์ กรแก้ววัฒนากุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 5 ฉบับ AT89C5X ของ ATMEL", (C) INNOVATIVE EXPERIMENT CO.,LTD.

สัจจะ จรัสรุ่งรวีวร, "คู่มือการเขียนโปรแกรมและการใช้งาน VISUAL BASIC 6.O", กรุงเทพฯ : อินโฟเพรส, 2544. 392 หน้า.

B.C. KUO, " AUTOMATIC CONTROL SYSTEM," 7TH EDITION, PRENTICE-HALL INTERNATIONAL EDITION, 1995.

R.C. DORF AND R.H. BISHOP, "MODERN CONTROL SYSTEM", ADDISON WESLEY, 7TH EDITION, 1995.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

PCF8591
8-bit A/D and D/A converter

Product specification
Supersedes data of 1997 Apr 02
File under Integrated Circuits, IC12

1998 Jul 02

Philips Semiconductors

PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTENTS

1	FEATURES
2	APPLICATIONS
3	GENERAL DESCRIPTION
4	ORDERING INFORMATION
5	BLOCK DIAGRAM
6	PINNING
7	FUNCTIONAL DESCRIPTION
7.1	Addressing
7.2	Control byte
7.3	D/A conversion
7.4	A/D conversion
7.5	Reference voltage
7.6	Oscillator
8	CHARACTERISTICS OF THE I²C-BUS
8.1	Bit transfer
8.2	Start and stop conditions
8.3	System configuration
8.4	Acknowledge
8.5	I²C-bus protocol
9	LIMITING VALUES
10	HANDLING
11	DC CHARACTERISTICS
12	D/A CHARACTERISTICS
13	A/D CHARACTERISTICS
14	AC CHARACTERISTICS
15	APPLICATION INFORMATION
16	PACKAGE OUTLINES
17	SOLDERING
17.1	Introduction
17.2	DIP
17.2.1	Soldering by dipping or by wave
17.2.2	Repairing soldered joints
17.3	SO
17.3.1	Reflow soldering
17.3.2	Wave soldering
17.3.3	Repairing soldered joints
18	DEFINITIONS
19	LIFE SUPPORT APPLICATIONS
20	PURCHASE OF PHILIPS I²C COMPONENTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 FEATURES

- Single power supply
- Operating supply voltage 2.5 V to 6 V
- Low standby current
- Serial input/output via I²C-bus
- Address by 3 hardware address pins
- Sampling rate given by I²C-bus speed
- 4 analog inputs programmable as single-ended or differential inputs
- Auto-incremented channel selection
- Analog voltage range from V_{SS} to V_{DD}
- On-chip track and hold circuit
- 8-bit successive approximation A/D conversion
- Multiplying DAC with one analog output.

2 APPLICATIONS

- Closed loop control systems
- Low power converter for remote data acquisition
- Battery operated equipment
- Acquisition of analog values in automotive, audio and TV applications.

3 GENERAL DESCRIPTION

The PCF8591 is a single-chip, single-supply low power 8-bit CMOS data acquisition device with four analog inputs, one analog output and a serial I²C-bus interface. Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I²C-bus without additional hardware. Address, control and data to and from the device are transferred serially via the two-line bidirectional I²C-bus.

The functions of the device include analog input multiplexing, on-chip track and hold function, 8-bit analog-to-digital conversion and an 8-bit digital-to-analog conversion. The maximum conversion rate is given by the maximum speed of the I²C-bus.

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCA8591P	DIP16	plastic dual in-line package; 16 leads (300 mil); long body	SOT38-1
PCA8591T	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 BLOCK DIAGRAM

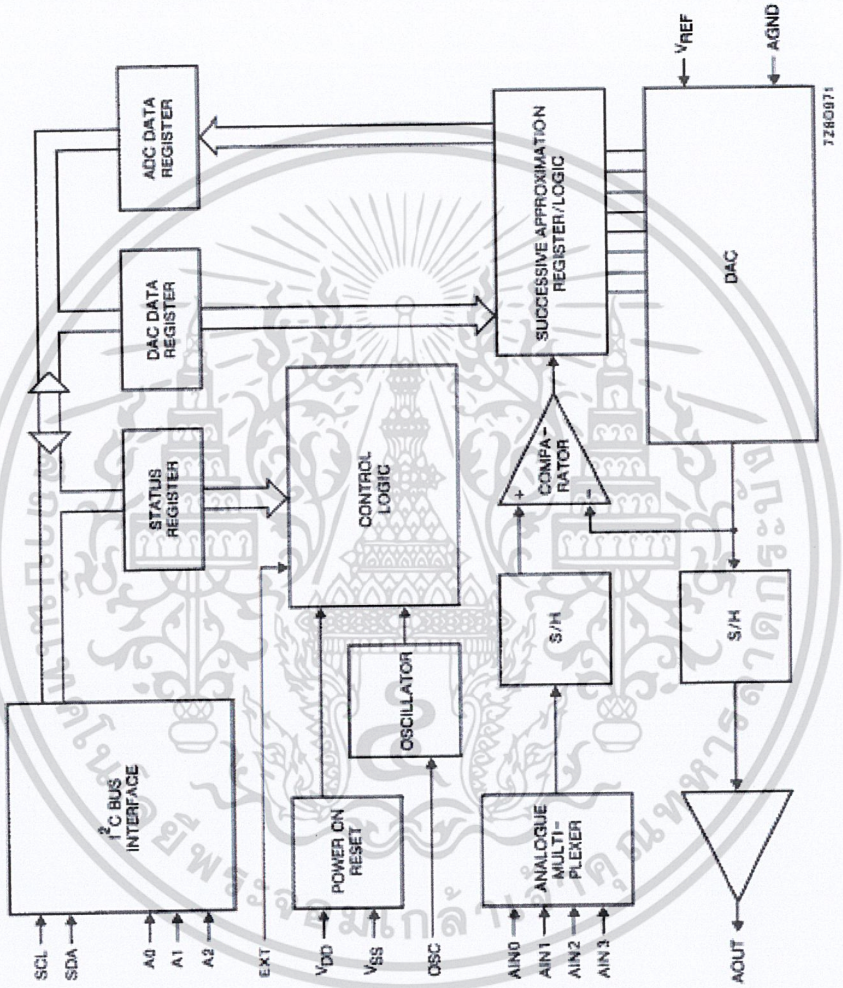


Fig.1 Block diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 PINNING

SYMBOL	PIN	DESCRIPTION
AIN0	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware address
A1	6	
A2	7	
V_{SS}	8	negative supply voltage
SDA	9	I²C-bus data input/output
SCL	10	I²C-bus clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground
V_{REF}	14	voltage reference input
AOUT	15	analog output (D/A converter)
V_{DD}	16	positive supply voltage

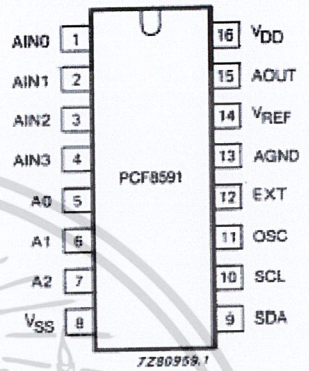


Fig.2 Pinning diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 FUNCTIONAL DESCRIPTION

7.1 Addressing

Each PCF8591 device in an I²C-bus system is activated by sending a valid address to the device. The address consists of a fixed part and a programmable part. The programmable part must be set according to the address pins A0, A1 and A2. The address always has to be sent as the first byte after the start condition in the I²C-bus protocol. The last bit of the address byte is the read/write-bit which sets the direction of the following data transfer (see Figs 3, 15 and 16).

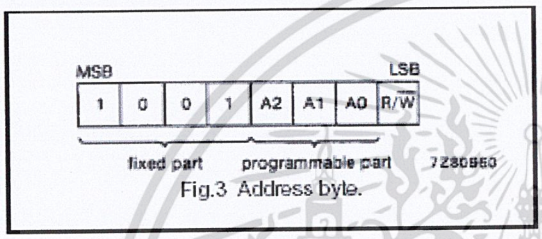


Fig.3 Address byte.

7.2 Control byte

The second byte sent to a PCF8591 device will be stored in its control register and is required to control the device function.

The upper nibble of the control register is used for enabling the analog output, and for programming the analog inputs as single-ended or differential inputs. The lower nibble selects one of the analog input channels defined by the upper nibble (see Fig.4). If the auto-increment flag is set the channel number is incremented automatically after each A/D conversion.

If the auto-increment mode is desired in applications where the internal oscillator is used, the analog output enable flag in the control byte (bit 6) should be set. This allows the internal oscillator to run continuously, thereby preventing conversion errors resulting from oscillator start-up delay. The analog output enable flag may be reset at other times to reduce quiescent power consumption.

The selection of a non-existing input channel results in the highest available channel number being allocated. Therefore, if the auto-increment flag is set, the next selected channel will be always channel 0. The most significant bits of both nibbles are reserved for future functions and have to be set to 0. After a Power-on reset condition all bits of the control register are reset to 0. The D/A converter and the oscillator are disabled for power saving. The analog output is switched to a high-impedance state.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

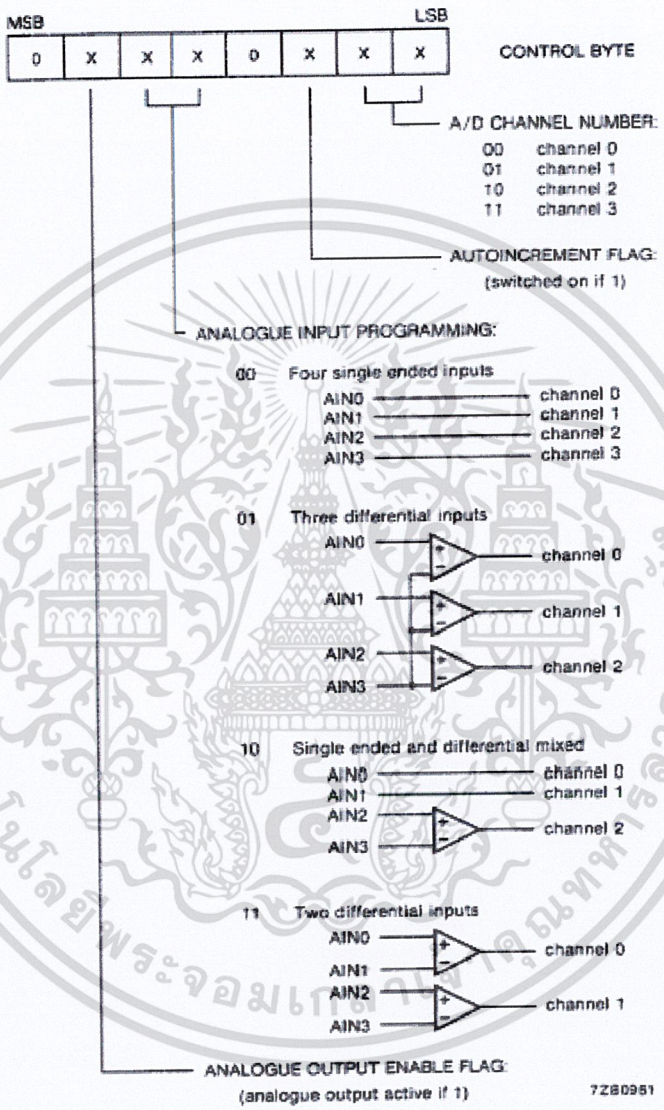


Fig.4 Control byte.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 D/A conversion

The third byte sent to a PCF8591 device is stored in the DAC data register and is converted to the corresponding analog voltage using the on-chip D/A converter. This D/A converter consists of a resistor divider chain connected to the external reference voltage with 256 taps and selection switches. The tap-decoder switches one of these taps to the DAC output line (see Fig.5).

The analog output voltage is buffered by an auto-zeroed unity gain amplifier. This buffer amplifier may be switched on or off by setting the analog output enable flag of the control register. In the active state the output voltage is held until a further data byte is sent.

The on-chip D/A converter is also used for successive approximation A/D conversion. In order to release the DAC for an A/D conversion cycle the unity gain amplifier is equipped with a track and hold circuit. This circuit holds the output voltage while executing the A/D conversion.

The output voltage supplied to the analog output AOUT is given by the formula shown in Fig.6. The waveforms of a D/A conversion sequence are shown in Fig.7.

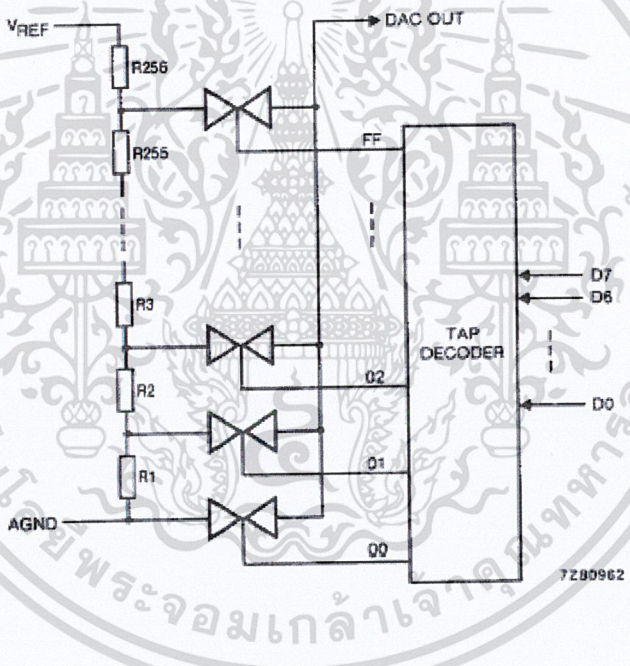


Fig.5 DAC resistor divider chain.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

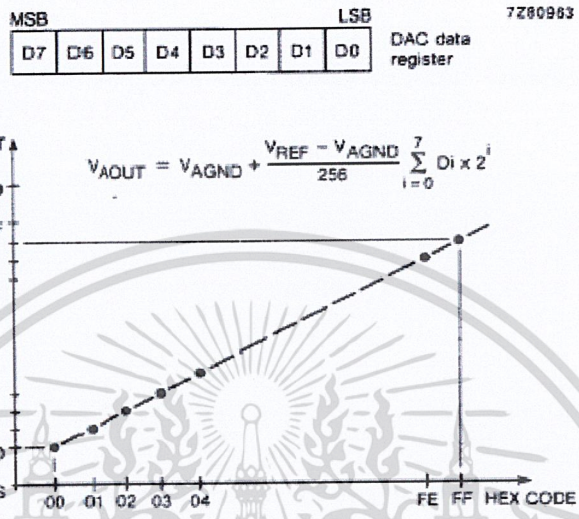


Fig.6 DAC data and DC conversion characteristics.

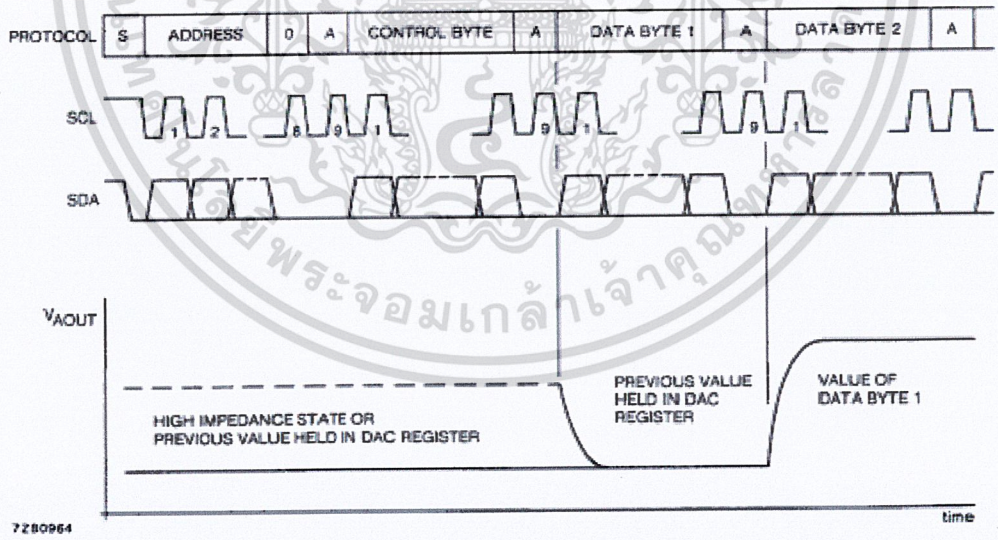


Fig.7 D/A conversion sequence.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 A/D conversion

The A/D converter makes use of the successive approximation conversion technique. The on-chip D/A converter and a high-gain comparator are used temporarily during an A/D conversion cycle.

An A/D conversion cycle is always started after sending a valid read mode address to a PCF8591 device. The A/D conversion cycle is triggered at the trailing edge of the acknowledge clock pulse and is executed while transmitting the result of the previous conversion (see Fig.8).

Once a conversion cycle is triggered an input voltage sample of the selected channel is stored on the chip and is converted to the corresponding 8-bit binary code. Samples picked up from differential inputs are converted to an 8-bit two's complement code (see Figs 9 and 10).

The conversion result is stored in the ADC data register and awaits transmission. If the auto-increment flag is set the next channel is selected.

The first byte transmitted in a read cycle contains the conversion result code of the previous read cycle. After a Power-on reset condition the first byte read is a hexadecimal 80. The protocol of an I²C-bus read cycle is shown in Chapter 8, Figs 15 and 16.

The maximum A/D conversion rate is given by the actual speed of the I²C-bus.

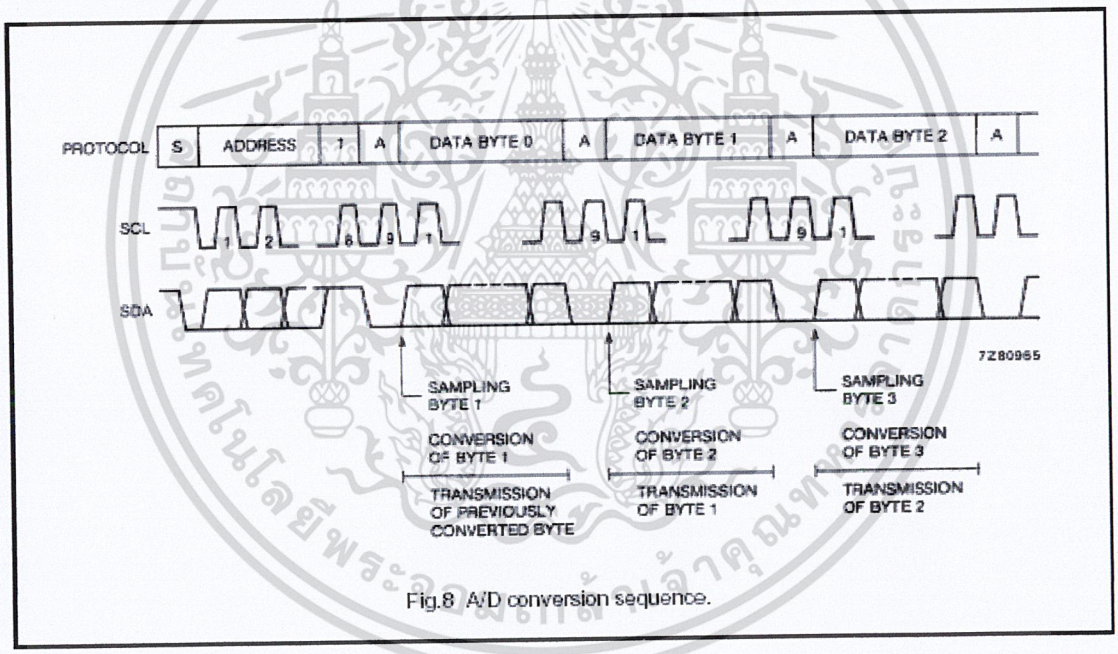


Fig.8 A/D conversion sequence.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

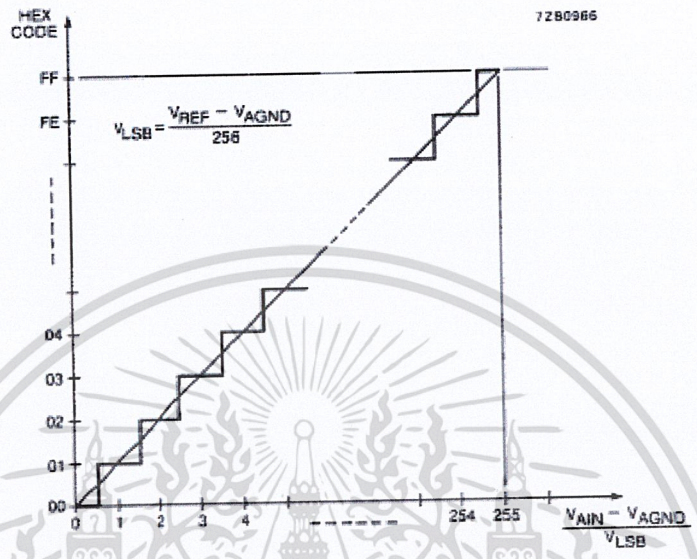


Fig.9 A/D conversion characteristics of single-ended inputs.

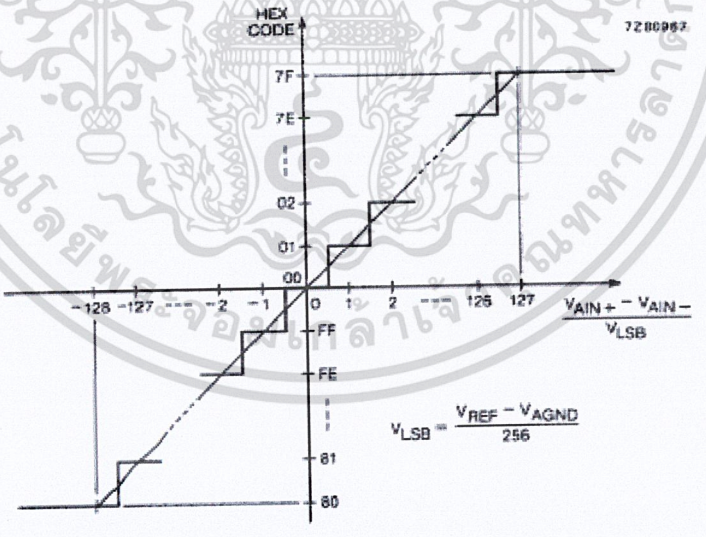


Fig.10 A/D conversion characteristics of differential inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 Reference voltage

For the D/A and A/D conversion either a stable external voltage reference or the supply voltage has to be applied to the resistor divider chain (pins V_{REF} and AGND). The AGND pin has to be connected to the system analog ground and may have a DC off-set with reference to V_{SS}.

A low frequency may be applied to the V_{REF} and AGND pins. This allows the use of the D/A converter as a one-quadrant multiplier; see Chapter 15 and Fig.6.

The A/D converter may also be used as a one or two quadrant analog divider. The analog input voltage is divided by the reference voltage. The result is converted to a binary code. In this application the user has to keep the reference voltage stable during the conversion cycle.

7.6 Oscillator

An on-chip oscillator generates the clock signal required for the A/D conversion cycle and for refreshing the auto-zeroed buffer amplifier. When using this oscillator the EXT pin has to be connected to V_{SS}. At the OSC pin the oscillator frequency is available.

If the EXT pin is connected to V_{DD} the oscillator output OSC is switched to a high-impedance state allowing the user to feed an external clock signal to OSC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8 CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is for bidirectional, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor. Data transfer may be initiated only when the bus is not busy.

8.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as a control signal.

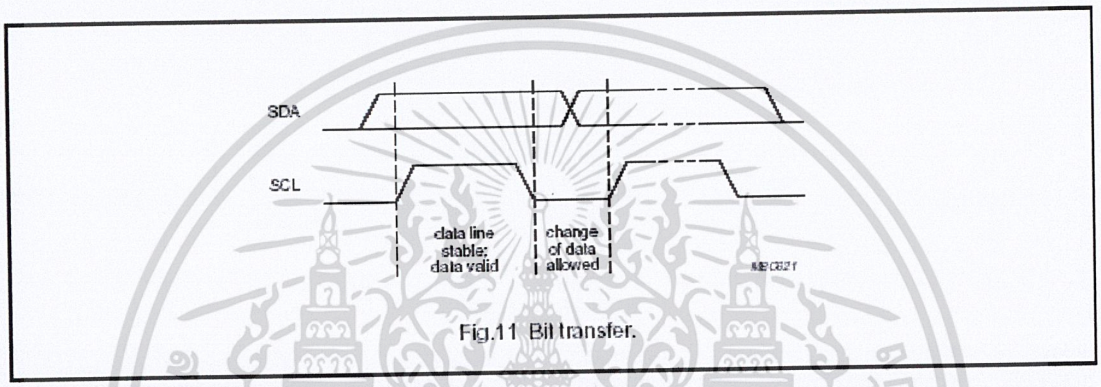


Fig.11 Bit transfer.

8.2 Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH, is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH, is defined as the stop condition (P).

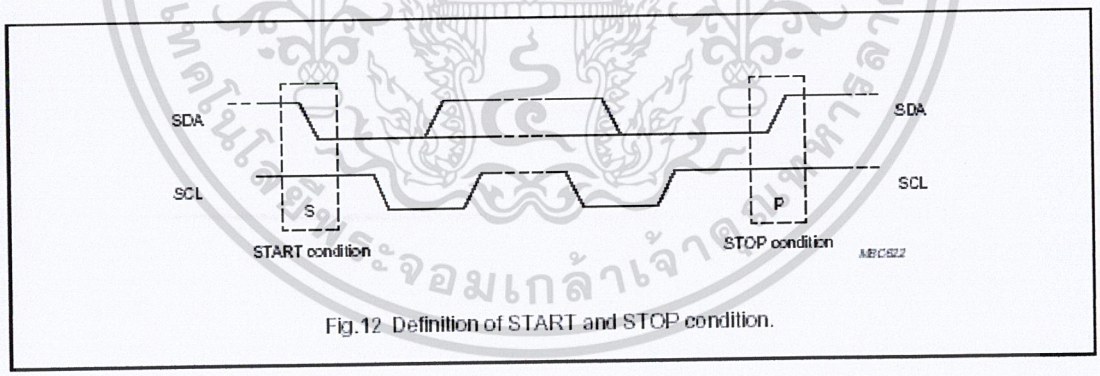


Fig.12 Definition of START and STOP condition.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 System configuration

A device generating a message is a 'transmitter', a device receiving a message is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves'.

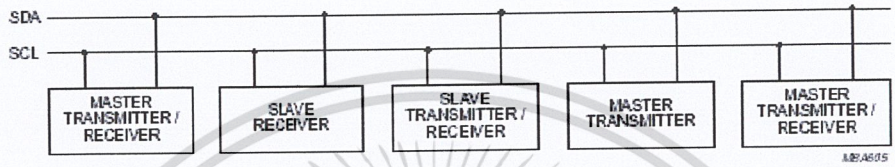


Fig.13 System configuration.

8.4 Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each data byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master also generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

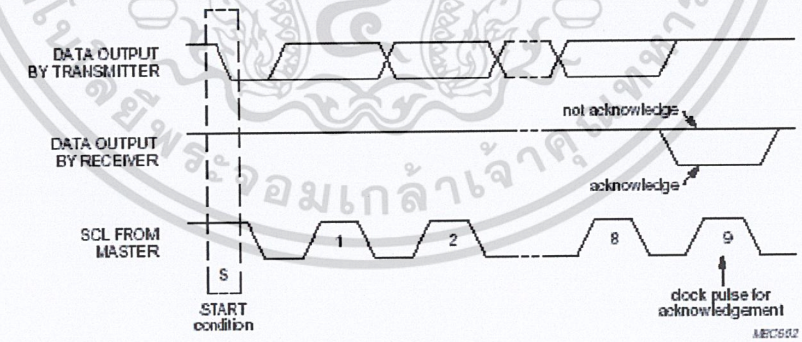


Fig.14 Acknowledgement on the I²C-bus.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.5 I²C-bus protocol

After a start condition a valid hardware address has to be sent to a PCF8591 device. The read/write bit defines the direction of the following single or multiple byte data transfer. For the format and the timing of the start condition (S), the stop condition (P) and the acknowledge bit (A) refer to the I²C-bus characteristics. In the write mode a data transfer is terminated by sending either a stop condition or the start condition of the next data transfer.

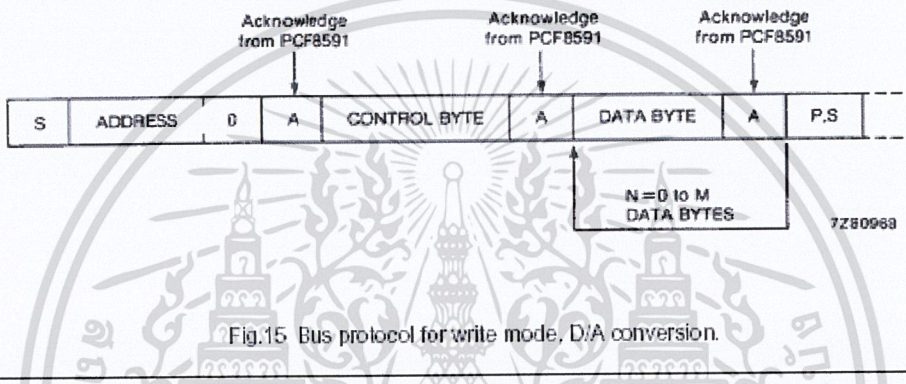


Fig.15 Bus protocol for write mode, D/A conversion.

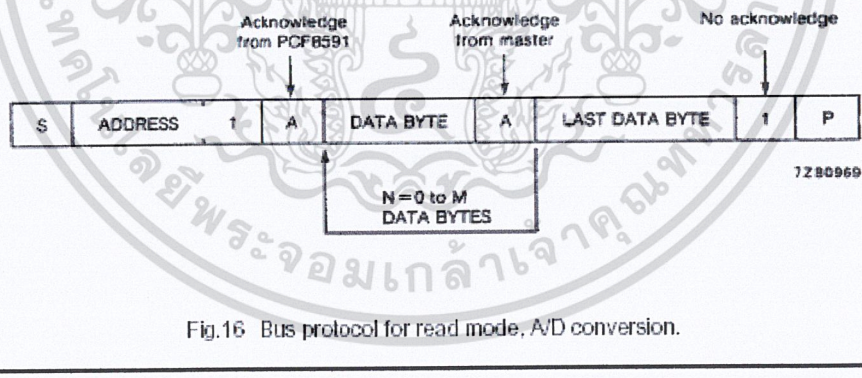


Fig.16 Bus protocol for read mode, A/D conversion.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{DD}	supply voltage (pin 16)	−0.5	+8.0	V
V_I	input voltage (any input)	−0.5	V_{DD} + 0.5	V
I_I	DC input current	–	±10	mA
I_O	DC output current	–	±20	mA
I_{DD}, I_{SS}	V_{DD} or V_{SS} current	–	±50	mA
P_{tot}	total power dissipation per package	–	300	mW
P_O	power dissipation per output	–	100	mW
T_{amb}	operating ambient temperature	−40	+85	°C
T_{stg}	storage temperature	−65	+150	°C

10 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take precautions appropriate to handling MOS devices. Advice can be found in Data Handbook IC12 under *"Handling MOS Devices"*.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11 DC CHARACTERISTICS

V_{DD} = 2.5 V to 6 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{DD}	supply voltage (operating)		2.5	–	6.0	V
I_{DD}	supply current					
	standby	$V_I = V_{SS}$ or V_{DD}; no load	–	1	15	μA
	operating, AOUT off	f_{SCL} = 100 kHz	–	125	250	μA
	operating, AOUT active	f_{SCL} = 100 kHz	–	0.45	1.0	mA
V_{POR}	Power-on reset level	note 1	0.8	–	2.0	V
Digital inputs/output: SCL, SDA, A0, A1, A2						
V_{IL}	LOW level input voltage		0	–	$0.3 \times V_{DD}$	V
V_{IH}	HIGH level input voltage		$0.7 \times V_{DD}$	–	V_{DD}	V
I_L	leakage current					
	A0, A1, A2	$V_I = V_{SS}$ to V_{DD}	−250	–	+250	nA
	SCL, SDA	$V_I = V_{SS}$ to V_{DD}	−1	–	+1	μA
C_i	input capacitance		–	–	5	pF
I_{OL}	LOW level SDA output current	V_{OL} = 0.4 V	3.0	–	–	mA
Reference voltage inputs						
V_{REF}	reference voltage	$V_{REF} > V_{AGND}$; note 2	V_{SS} + 1.6	–	V_{DD}	V
V_{AGND}	analog ground voltage	$V_{REF} > V_{AGND}$; note 2	V_{SS}	–	V_{DD} − 0.8	V
I_{LI}	input leakage current		−250	–	+250	nA
R_{REF}	input resistance	pins V_{REF} and AGND	–	100	–	kΩ
Oscillator: OSC, EXT						
I_{LI}	input leakage current		–	–	250	nA
f_{OSC}	oscillator frequency		0.75	–	1.25	MHz

Notes

1. The power on reset circuit resets the I²C-bus logic when V_{DD} is less than V_{POR}.
2. A further extension of the range is possible, if the following conditions are fulfilled:

$$\frac{V_{REF} + V_{AGND}}{2} \geq 0.8V, V_{DD} - \frac{V_{REF} + V_{AGND}}{2} \geq 0.4V$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12 D/A CHARACTERISTICS

V_{DD} = 5.0 V; V_{SS} = 0 V; V_{REF} = 5.0 V; V_{AGND} = 0 V; R_L = 10 kΩ; C_L = 100 pF; T_{amb} = −40 °C to +85 °C unless otherwise specified.

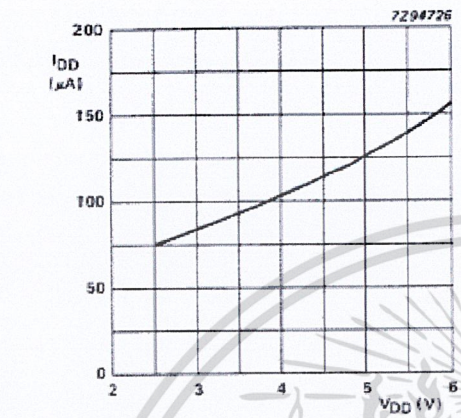
SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Analog output						
V_{OA}	output voltage	no resistive load	V_{SS}	–	V_{DD}	V
		R_L = 10 kΩ	V_{SS}	–	$0.9 \times V_{DD}$	V
I_{LO}	output leakage current	AOUT disabled	–	–	250	nA
Accuracy						
OS_e	offset error	T_{amb} = 25 °C	–	–	50	mV
L_e	linearity error		–	–	±1.5	LSB
G_e	gain error	no resistive load	–	–	1	%
t_{DAC}	settling time	to $\frac{1}{2}$ LSB full scale step	–	–	90	µs
f_{DAC}	conversion rate		–	–	11.1	kHz
SNRR	supply noise rejection ratio	f = 100 Hz; $V_{DDN} = 0.1 \times V_{PP}$	–	40	–	dB

13 A/D CHARACTERISTICS

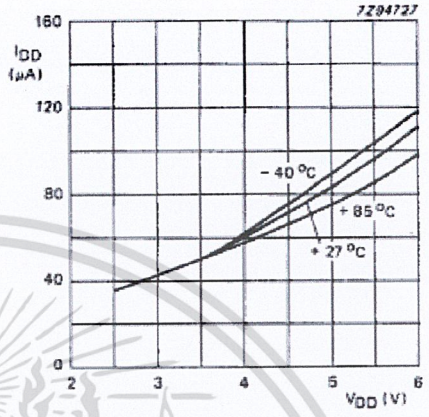
V_{DD} = 5.0 V; V_{SS} = 0 V; V_{REF} = 5.0 V; V_{AGND} = 0 V; R_S = 10 kΩ; T_{amb} = −40 °C to +85 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Analog inputs						
V_{IA}	analog input voltage		V_{SS}	–	V_{DD}	V
I_{LIA}	analog input leakage current		–	–	100	nA
C_{IA}	analog input capacitance		–	10	–	pF
C_{ID}	differential input capacitance		–	10	–	pF
V_{IS}	single-ended voltage	measuring range	V_{AGND}	–	V_{REF}	V
V_{ID}	differential voltage	measuring range; $V_{FS} = V_{REF} - V_{AGND}$	$\frac{-V_{FS}}{2}$	–	$\frac{+V_{FS}}{2}$	V
Accuracy						
OS_e	offset error	T_{amb} = 25 °C	–	–	20	mV
L_e	linearity error		–	–	±1.5	LSB
G_e	gain error		–	–	1	%
GS_e	small-signal gain error	ΔV_i = 16 LSB	–	–	5	%
CMRR	common-mode rejection ratio		–	60	–	dB
SNRR	supply noise rejection ratio	f = 100 Hz; $V_{DDN} = 0.1 \times V_{PP}$	–	40	–	dB
t_{ADC}	conversion time		–	–	90	µs
f_{ADC}	sampling/conversion rate		–	–	11.1	kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

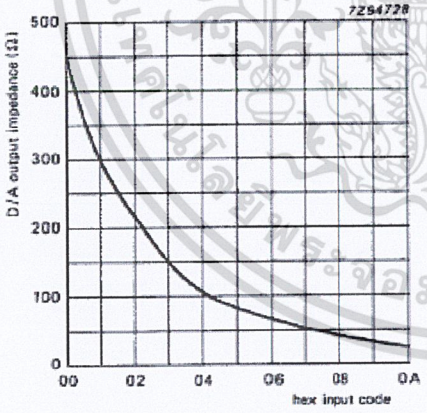


(a) Internal oscillator; T_{amb} = +27 °C.

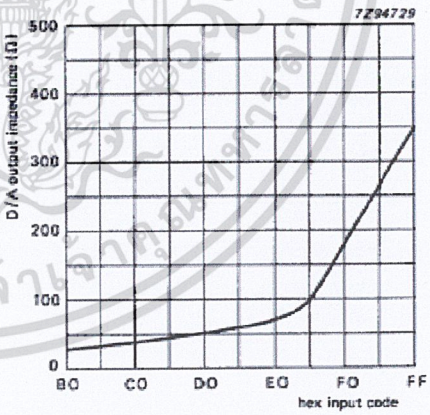


(b) External oscillator.

Fig.17 Operating supply current as a function of supply voltage (analog output disabled).



(a) Output impedance near negative power rail; T_{amb} = +27 °C.



(b) Output impedance near positive power rail; T_{amb} = +27 °C.

The x-axis represents the hex input-code equivalent of the output voltage.

Fig.18 Output impedance of analog output buffer (near power rails).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14 AC CHARACTERISTICS

All timing values are valid within the operating supply voltage and ambient temperature range and reference to V_{IL} and V_{IH} with an input voltage swing of V_{SS} to V_{DD}.

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
I²C-bus timing (see Fig.19; note 1)					
f_{SCL}	SCL clock frequency	–	–	100	kHz
t_{SP}	tolerable spike width on bus	–	–	100	ns
t_{BUF}	bus free time	4.7	–	–	µs
$t_{SU;STA}$	START condition set-up time	4.7	–	–	µs
$t_{HD;STA}$	START condition hold time	4.0	–	–	µs
t_{LOW}	SCL LOW time	4.7	–	–	µs
t_{HIGH}	SCL HIGH time	4.0	–	–	µs
t_r	SCL and SDA rise time	–	–	1.0	µs
t_f	SCL and SDA fall time	–	–	0.3	µs
$t_{SU;DAT}$	data set-up time	250	–	–	ns
$t_{HD;DAT}$	data hold time	0	–	–	ns
$t_{VD;DAT}$	SCL LOW-to-data out valid	–	–	3.4	µs
$t_{SU;STO}$	STOP condition set-up time	4.0	–	–	µs

Note

1. A detailed description of the I²C-bus specification, with applications, is given in brochure *"The I²C-bus and how to use it"*. This brochure may be ordered using the code 9398 393 40011.

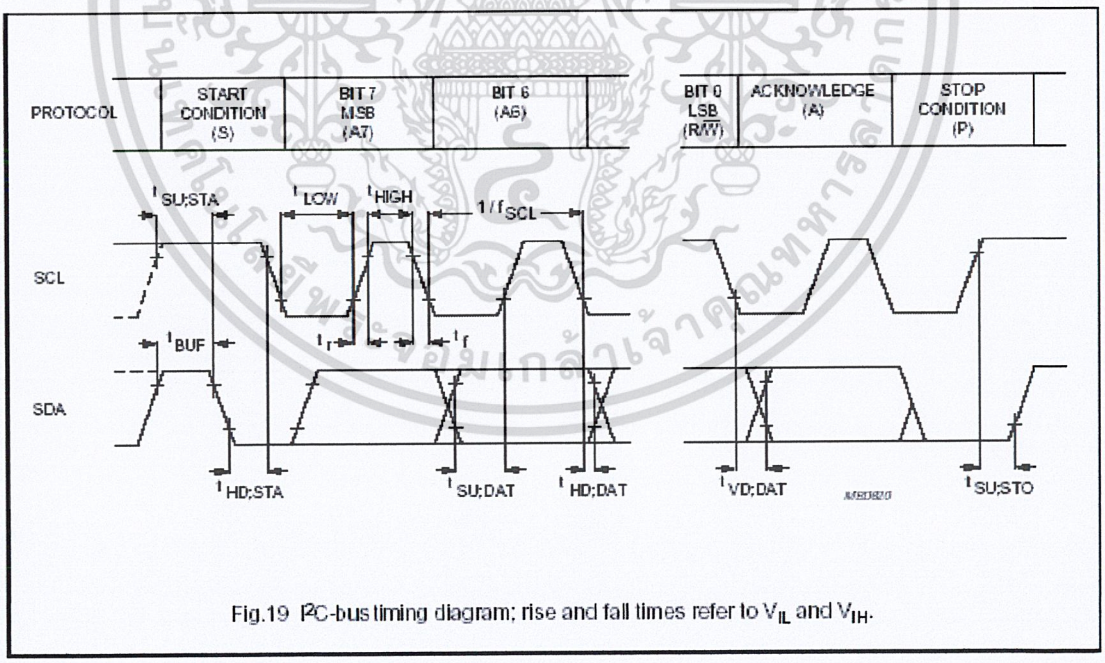


Fig.19 I²C-bus timing diagram; rise and fall times refer to V_{IL} and V_{IH}.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15 APPLICATION INFORMATION

Inputs must be connected to V_{SS} or V_{DD} when not in use. Analog inputs may also be connected to AGND or V_{REF}.

In order to prevent excessive ground and supply noise and to minimize cross-talk of the digital to analog signal paths the user has to design the printed-circuit board layout very carefully. Supply lines common to a PCF8591 device and noisy digital circuits and ground loops should be avoided. Decoupling capacitors (>10 µF) are recommended for power supply and reference voltage inputs.

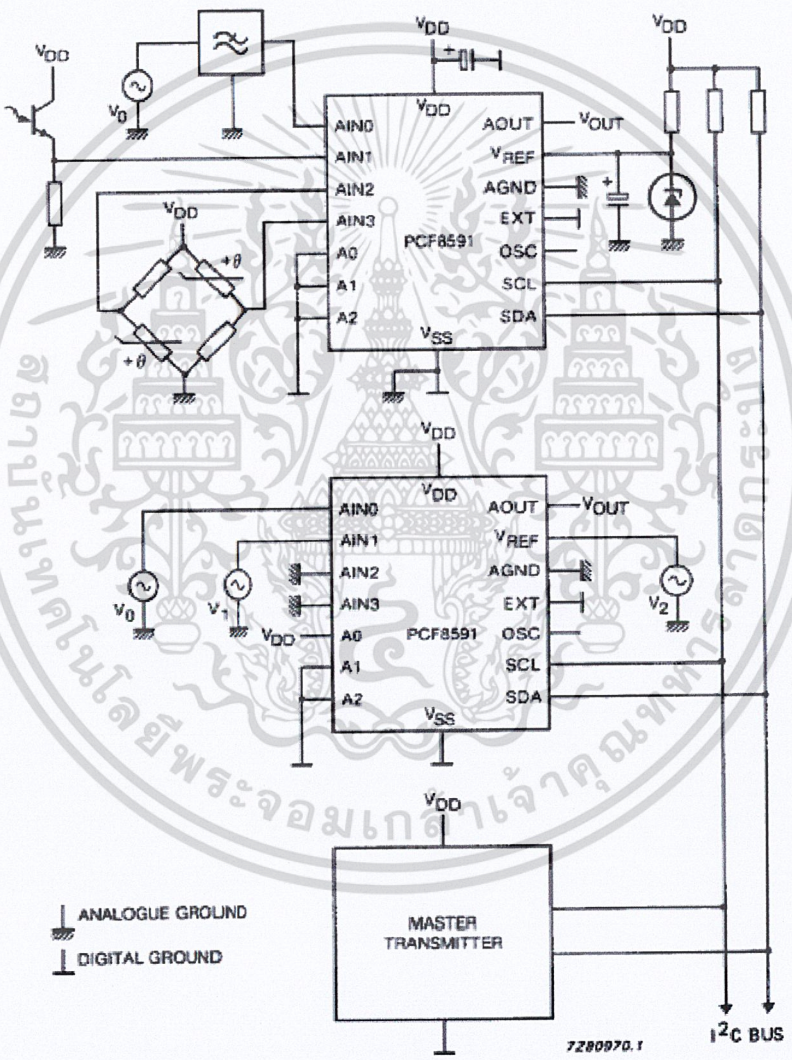
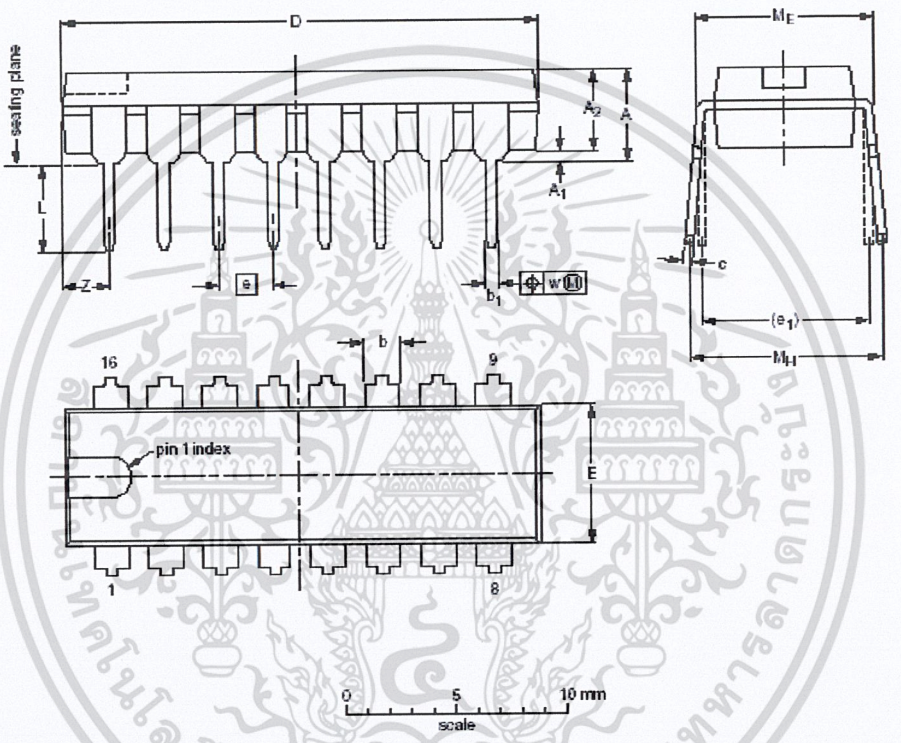


Fig.20 Application diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16 PACKAGE OUTLINES

DIP16: plastic dual in-line package; 16 leads (300 mil); long body **SOT38-1**



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.7	0.51	3.7	1.40 1.14	0.53 0.38	0.32 0.23	21.8 21.4	6.48 6.20	2.54	7.62	3.9 3.4	8.25 7.80	9.5 8.3	0.254	2.2
inches	0.19	0.020	0.15	0.055 0.045	0.021 0.015	0.013 0.009	0.86 0.84	0.26 0.24	0.10	0.30	0.15 0.13	0.32 0.31	0.37 0.33	0.01	0.087

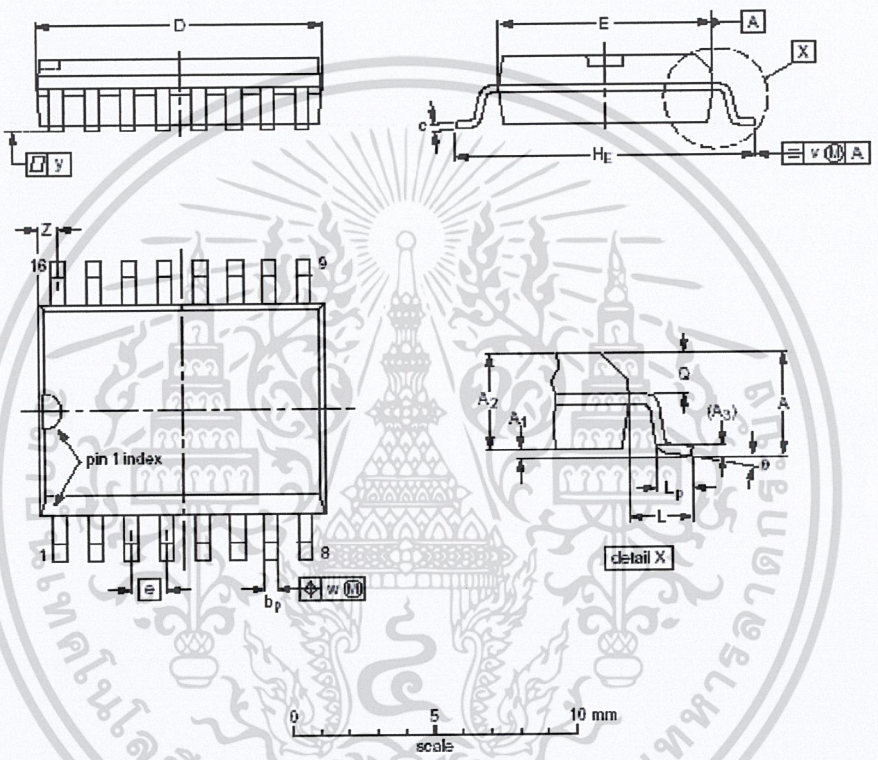
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT38-1	050G09	MO-001AE				~~92-10-02~~ 95-01-19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SO16: plastic small outline package; 16 leads; body width 7.5 mm SOT162-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	10.5 10.1	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.41 0.40	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT162-1	075E03	MS-013AA				~~95-01-24~~ 97-05-22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17 SOLDERING

17.1 Introduction

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"Data Handbook IC26; Integrated Circuit Packages"* (order code 9398 652 90011).

17.2 DIP

17.2.1 Soldering by dipping or by wave

The maximum permissible temperature of the solder is 260 °C; solder at this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg\ max}$). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

17.2.2 Repairing soldered joints

Apply a low voltage soldering iron (less than 24 V) to the lead(s) of the package, below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400 °C, contact may be up to 5 seconds.

17.3 SO

17.3.1 Reflow soldering

Reflow soldering techniques are suitable for all SO packages.

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt. Dwell times vary between 50 and 300 seconds depending on heating method. Typical reflow temperatures range from 215 to 250 °C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 minutes at 45 °C.

17.3.2 Wave soldering

Wave soldering techniques can be used for all SO packages if the following conditions are observed:

- A double-wave (a turbulent wave with high upward pressure followed by a smooth laminar wave) soldering technique should be used.
- The longitudinal axis of the package footprint must be parallel to the solder flow.
- The package footprint must incorporate solder thieves at the downstream end.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder is 10 seconds, if cooled to less than 150 °C within 6 seconds. Typical dwell time is 4 seconds at 250 °C.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

17.3.3 Repairing soldered joints

Fix the component by first soldering two diagonally-opposite end leads. Use only a low voltage soldering iron (less than 24 V) applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18 DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

19 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

20 PURCHASE OF PHILIPS I²C COMPONENTS

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming a 24LC00 EEPROM using the EZ-USB I²C Port

Introduction

The Cypress EZ-USB family contains a general-purpose I²C interface through which the 8051 accesses standard I²C devices connected to the SCL and SDA pins. In normal operation, a small EEPROM connected to the I²C port supplies vendor-specific ID information to the operating system to enable it to load the proper software driver associated with the device. These ID tags, called Vendor ID (VID), Product ID (PID), and Device ID (DID), occupy the first seven bytes of the EEPROM, as shown in *Table 1* below.

Table 1. Contents of 16-byte 24LC00 EEPROM

EEPROM Address	Value
0	0xB0
1	VID(L)
2	VID(H)
3	PID(L)
4	PID(H)
5	DID(L)
6	DID(H)
7	available

The EZ-USB chip uses the constant 0xB0 in the first byte to identify the subsequent bytes as VID/PID/DID information.

An excellent choice for the EEPROM is the Microchip 24LC00-OT, a 16-byte serial EEPROM in a 5-lead SOT-23 package. This device is similar to the widely available 24 series of EEPROMS such as the 24LC01 (128 bytes) and the 24LC02. The "LC" designation denotes low-voltage operation (2.5–6.0 volts) making the part ideal for a 3.3V USB peripheral. At this writing the Microchip 24LC00 is the only version available in the very small SOT-23 package.

The Microchip 24LC00 is used on the Cypress EZ-USB Development Board. It is provided in an 8-pin socketed DIP package so that other EEPROM types may be plugged in for study and debugging.

Two subtle differences between the 24LC00 and the other EEPROMS in the family (lack of address pins and lack of "page write") are mentioned in this note. This note concludes with a set of general-purpose 8051 subroutines that perform byte reads and writes to the 24LC00 on the EZ-USB Development Board. The routines assume the lowest common functionality in the 24LC family, making them usable for any of the family devices ('00, '01, and '02).

EEPROM Details

The 24LC00 responds to the I²C slave address 1010xxxd, where "d" is the direction bit (d=1 for read, d=0 for write). The 'xxx' indicates that the device responds to all eight slave addresses between 1010 000 and 1010 111. This is due to the fact that the 24LC00 lacks the A[2..0] pins found on the 24LC01 and 24LC02 devices.

The 24LC00 contains an internal address counter that must be written before reading or writing an EEPROM byte. This address counter automatically increments for sequential EEPROM byte reads, but it does not increment for sequential byte writes. This is the major difference between the 24LC00 and the 24LC01/02 devices. The 24LC01/02 devices also have a "page write" mechanism that allows loading eight bytes at a time, where each load automatically increments the internal address counter. By writing the example routines to explicitly write an address before each EEPROM byte write, the code is compatible with all three part types.

Although the 24LC series is called "electrically erasable", there is no explicit erase operation. Writing a byte first erases the byte, then reprograms it.

An EEPROM write operation consists of sending the bytes shown in *Table 2* to the 24LC00.

Table 2. 24LC00 Write Operation

Byte	Value	Meaning
1	10100000	Command byte—Write
2	0000aaaa	EEPROM address to write
3	Dddddddd	Data to write

An I²C START condition precedes the first byte, and a STOP condition follows the final byte. The STOP condition initiates the erase/write cycle, which takes a maximum of 4 milliseconds to complete in the 24LC00.

The 8051 checks for completion of an erase/write cycle by repeatedly sending "byte write" commands (byte 1 in *Table 2*) to the EEPROM, and checking the ACK bit. During the EEPROM programming cycle the EEPROM responds with ACK=0 (not acknowledge) on the I²C bus. When the programming cycle is complete it responds with ACK=1 (acknowledge) on the I²C bus. (These two conditions are shown in *Figure 3.*) Note that the I²C *bus* polarities are the opposite sense of the ACK bit in the I2CS register (I2CS.1), so the 8051 polling routine waits while I2CS.1=0 and exits when I2CS.1=1.

I²C Data Transfers

The 8051 communicates with the I²C bus using two registers, shown below:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I2CS			I^2C Control and Status				7FA5
b7	b6	b5	b4	b3	b2	b1	b0
START	STOP	LASTRD	ID1	ID0	BERR	ACK	DONE
R/W	R/W	R/W	R	R	R	R	R
0	0	0	x	x	0	0	0

I2DAT			I^2C Data				7FA6
b7	b6	b5	b4	b3	b2	b1	b0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
x	x	x	x	x	x	x	x

The 8051 initiates a transfer by setting the START bit (I2CS.7), and then writing a command byte to the I2DAT register. When the I^2C controller is ready, the 8051 then writes or reads data to/from I2DAT. Finally, the 8051 sets the STOP bit to terminate the transaction.

The I2CS register bits operate as follows:

DONE

When the 8051 initiates an I^2C transfer, the DONE bit (I2CS.0) goes LOW, and returns HIGH when the transfer completes and the EZ-USB I^2C controller is ready. The DONE bit is automatically cleared when the 8051 reads or writes I2DAT.

STOP

The 8051 terminates an I^2C transfer by setting STOP=1. This bit stays HIGH until the I^2C controller finishes sending the STOP condition on the I^2C bus, at which time it clears the STOP bit. Completion of the STOP condition on the I^2C bus has no effect on the DONE bit.

For an I^2C read operation, the 8051 must read the last data byte from I2DAT before the STOP condition completes (within about 11 microseconds). Therefore the 8051 code should read the I2DAT register immediately after setting the STOP bit. It is therefore good practice to begin every I^2C transfer routine with a check for STOP=0, indicating that any previous STOP condition has completed and the I^2C controller is "listening."

LastRD

The 8051 sets the LASTRD bit (I2CS.5) before clocking in the last byte in a read operation. This instructs the EZ-USB I^2C controller not to generate an ACK for the last transfer. The lack of an ACK from the I^2C master (the EZ-USB I^2C controller) signals the I^2C peripheral to stop sending.

ACK, BERR

After every transfer (when DONE goes HIGH), two status bits indicate if the transfer was acknowledged (ACK, I2CS.1) and if another I^2C device interfered by driving the bus at the same time as the EZ-USB controller (BERR, I2CS.2). The example code checks the ACK bit for completion of an erase/program cycle. For simplicity, and because no other I^2C device is present on the EZ-USB Development board, the example code does not check the BERR bit.

ID1–ID0

These bits, which indicate the EEPROM type detected by the EZ-USB I^2C boot loader, may be safely ignored for these examples. For reference, Chapter 4 of the *EZ-USB Technical Reference Manual* describes the meaning of these read-only bits.

Code Description

The code listing at the end of this note contains several subroutines:

1. Reset EEPROM address to 0000 ('reset_address', line 218).
2. Read first eight bytes from the EEPROM ('read8_eeprom', line 63). The 8051 code stores these bytes in internal registers at 0x80-0x87.
3. Write the sixteen EEPROM bytes with the data in 8051 internal registers at 0x80-0x8F ('write16_eeprom', line 125).
4. Test the above routines ('test', line 39), as follows. Call (1) to reset the EEPROM address pointer to zero, call routine (2) to read the first eight EEPROM bytes into 0x80-0x87, fill the eight bytes at 0x88-0x8F with the values 8,7,6,5,4,3,2,1, and finally call (3) to write the EEPROM using the sixteen bytes at 0x80-0x8F.

The result is an EEPROM that contains the first eight bytes previously stored in the EEPROM, and the count 8->1 in the last eight bytes. It should be easy to modify the example test routine to write any desired data.

Programming the LastRD Bit

The LastRD bit in I2CS.5 instructs the EZ-USB I^2C controller to float the SDA line at 'ACK' time (the ninth SCL of the byte transfer) to alert the slave (EEPROM) to stop sending. Looking at the code, it may appear that the LastRD bit is actually set two bytes before the last transfer, since the code sequence to read eight bytes is as follows:

1. Perform a dummy read of I2DAT to send out the first nine SCL pulses which clock the first byte into the I2DAT register.
2. Read the first six bytes from I2DAT.
3. Set the LastRD bit.
4. Read the seventh byte from I2DAT.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

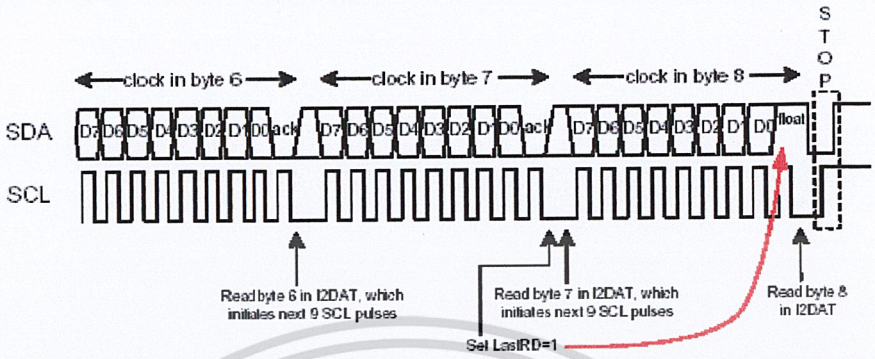


Figure 1. The 8051 Sets LastRD=1 Before Reading the Second-to-last byte from I2DAT

5. Read the eighth byte from I2DAT.

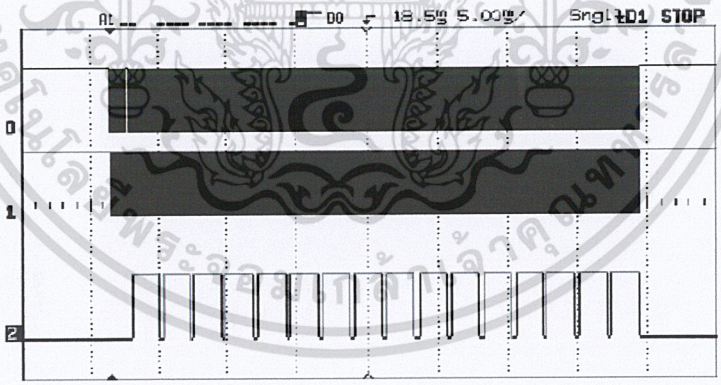
6. Set STOP=1.

The reason the LastRD is set to one before reading the seventh, and not the eighth byte, is that the 8051 must set LastRD=1 before the last nine clock pulses are sent out by the EZ-USB controller. Therefore LastRD must be set to 1 before the 8051 reads I2DAT to retrieve the second-to-last byte (byte 7). This is illustrated in *Figure 1*.

Waveforms

Figure 2 illustrates the sixteen EEPROM byte write operations. Trace 2 is the port pin PORTA.0, which is programmed to be HI while a write cycle is in progress (set HI in lines 174–176, and LO in lines 198–200). The sixteen byte write times can clearly be seen in trace 2. Note that the 16-byte write takes less than 40 milliseconds. *Figure 3* (next page) shows the same data magnified to show the last write command byte that returns a non-ACK (EEPROM still busy writing) followed by the command byte that returns ACK to indicate that the write cycle has finished.

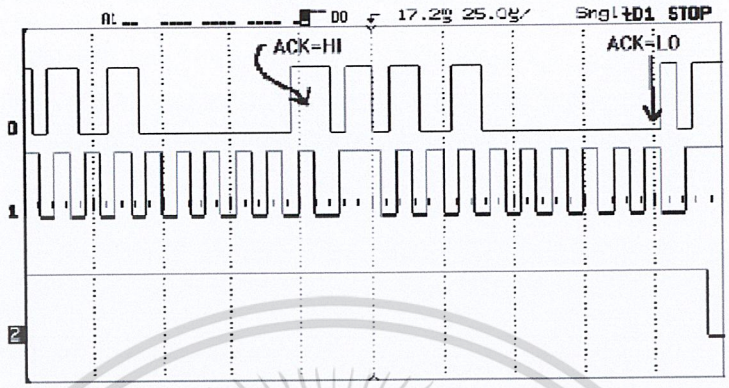
The 8051 sends the command byte '10100000' in both transfers in *Figure 3*. In the first transfer the 24LC00 sends ACK=1 (not-acknowledge) to indicate that a write cycle is still in progress. In the second transfer it sends ACK=0 (acknowledge) to indicate that the write cycle has completed and another byte can be erased/written.



Trace 0: SDA. Trace 1: SCL
Trace 2: PORTA.0–HI while waiting for EEPROM write cycle to complete.

Figure 2. Scope Traces for Listing A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Tracc 0: SDA. Tracc 1: SCL
Trace 2: PORTA.0–HI while waiting for EEPROM cycle to complete

Figure 3. Two Command Bytes Indicate 'busy' and 'done'.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้