

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องแสดงข้อความอักษรเบรลล์
BRAILLE CHARACTER DISPLAY



โดย
นางสาว พัชรัตน์ เจริญนรงค์
นาย เอกวุฒิ หวังคุณ

เลขหมู่.....
เลขทะเบียน..... 72958
วัน,เดือน,ปี..... 26 ส.ย. 2550

b. 1172519x
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาดานหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

ผ่านการตรวจรูปเล่มแล้ว
(ลงชื่อ).....ผู้ตรวจ
ผ่านการตรวจชิ้นงานแล้ว
(ลงชื่อ).....ผู้ตรวจ
เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปเผยแพร่จนดานการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องแสดงข้อความอักษรเบรลล์
BRAILLE CHARACTER DISPLAY

โดย

นางสาว พัชรวิทย์ เจริญธนรัตน์ 46010344

นาย เอกวุฒิ หวังกุล 46010998

อาจารย์ที่ปรึกษา

ผศ. เกรียงไกร วงศ์โรจนภรณ์

รศ.ดร. สุวิพล สัทธชีวกาศ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องแสดงข้อความอักษรเบรลล์

BRaille CHARACTER DISPLAY

ผู้จัดทำ

1. นางสาวพัชรวิชน์ เจริญนรัชต์ 46010344

2. นายเอกวุฒิ หวังคุณ 46010998

.....  อาจารย์ที่ปรึกษา
(ผศ.เกรียงไกร วงศ์โรจนภรณ์)

.....  อาจารย์ที่ปรึกษา
(รศ.ดร. สุวิทย์ สิริชีวะภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องแสดงข้อความอักษรเบรลล์

BRaille CHARACTER DISPLAY

โดย นางสาวพัชรวิรัตน์ เจริญนรินทร์ 46010344

นายเอกวุฒิ หวังคุณ 46010998

อาจารย์ที่ปรึกษา ผศ. เกரியงไกร วงศ์โรจนภรณ์

รศ.ดร. สุวิพล สิทธีชีวกาศ

บทคัดย่อ

โครงการนี้เป็นการนำเสนออุปกรณ์แปลงข้อความเป็นอักษรเบรลล์สำหรับคนตาบอด เพื่ออ่านข้อความบนคอมพิวเตอร์ โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมแผงเหล็กซึ่งจัดเรียงในลักษณะเป็นแมทริก เพื่อให้แผงเหล็กมีการเปลี่ยนแปลงตามรูปแบบของตัวอักษรเบรลล์

ABSTRACT

This project is to initiate a research and development of the Braille character display for a vision-impaired person. A message on a computer is being translated into Braille characters. A microcontroller device is used to control and interact with an array of metal buttons arranged in matrix format. This is to create physical display of Braille characters for a vision-impaired person.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จได้ ด้วยความกรุณาของผู้ช่วยศาสตราจารย์ เกรียงไกร วงศ์โรจนภรณ์ และ รองศาสตราจารย์ดร. สุวิพล สิริพิชิตภาค อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ซึ่งได้ให้คำปรึกษา ชี้แนะ และความช่วยเหลือในทุกๆ ด้าน ด้วยความเต็มใจ จึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอกราบขอบพระคุณ คณะอาจารย์ภาควิชาโทรคมนาคมทุกท่าน ที่ได้ให้ความรู้ ให้คำแนะนำให้ กำลังใจ

ขอบคุณและขอใจ นาย จิรณทัตย์ มาวิจักขณ์ ที่ได้ให้คำแนะนำด้านการศึกษาระเขียนโปรแกรม ตลอดจน พี่ เพื่อน และน้องภาควิชาโทรคมนาคมทุกคน รวมถึงผู้มีพระคุณทุกท่านที่มิได้เอ่ยนามไว้ ณ ที่นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

| | หน้า |
|---|------|
| บทที่ 1 บทนำ | |
| 1.1 ความเป็นมาของปริญญาโท | 1 |
| 1.2 วัตถุประสงค์ของปริญญาโท | 1 |
| 1.3 ขอบเขตของปริญญาโท | 1 |
| บทที่ 2 ทฤษฎีและหลักการ | |
| 2.1 ความเป็นมาและลักษณะของอักษรเบรลล์ | 2 |
| 2.1.1 ความเป็นมาของอักษรเบรลล์ | 2 |
| 2.1.2 ลักษณะของอักษรเบรลล์ | 2 |
| 2.1.2.1 กรณีที่ข้อมูลเป็นอักษรภาษาอังกฤษ | 2 |
| 2.1.2.2 กรณีข้อมูลเป็นตัวเลข | 3 |
| 2.2 ทฤษฎีไมโครคอนโทรลเลอร์ MCS-51 | 6 |
| 2.2.1 คุณลักษณะพื้นฐานของ MCS-51 | 6 |
| 2.2.2 ลักษณะการจัดขาของ MCS-51 | 6 |
| 2.2.3 พอร์ตอินพุตและพอร์ตเอาต์พุต | 8 |
| 2.2.3.1 การใช้งานพอร์ตเป็นอินพุต | 8 |
| 2.2.3.2 การใช้งานพอร์ตเป็นเอาต์พุต | 9 |
| 2.2.4 หน่วยความจำโปรแกรมของ MCS-51 | 9 |
| 2.2.5 หน่วยความจำข้อมูลของ MCS-51 | 10 |
| 2.2.6 รีจิสเตอร์หน้าที่พิเศษ | 13 |
| 2.3 ทฤษฎีเกี่ยวกับระบบบัสแบบไอสแควร์ซี (I ² C) | 14 |
| 2.3.1 คุณสมบัติทั่วไปของบัสไอสแควร์ซี | 14 |
| 2.4 ทฤษฎีเกี่ยวกับการสื่อสารข้อมูลแบบอนุกรม | 17 |
| 2.4.1 พื้นฐานการสื่อสารอนุกรม | 17 |
| 2.4.2 รูปแบบการสื่อสารข้อมูล | 18 |
| 2.4.2.1 การส่งข้อมูลแบบซิงโครนัส | 19 |
| 2.4.2.2 บิตเริ่มต้นและบิตสุดท้าย | 19 |
| 2.4.3 อัตราการรับส่งข้อมูล | 20 |
| 2.4.4 มาตรฐาน RS 232 | 20 |
| 2.4.5 การต่อ MCS-51 เข้ากับ RS-232 | 22 |
| 2.5 ทฤษฎีเกี่ยวกับวงจรเรียงกระแส (Rectifier) | 23 |
| 2.5.1 วงจรเรียงกระแสเต็มคลื่น (Full-Wave Rectifier) | 23 |
| 2.5.1.1 วงจรเรียงกระแสแบบบริดจ์ | 23 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | หน้า |
|--|------|
| 2.6 ทฤษฎีเกี่ยวกับไอซีเรกกูเลเตอร์ (IC Regulator) LM78xx, LM79xx | 24 |
| 2.7 ทฤษฎีเกี่ยวกับวงจรไบอัสกระแสตรง (DC Biasing Circuit) | 25 |
| 2.8 ทฤษฎีเกี่ยวกับการทำงานของโซลินอยด์ (Solenoid) | 25 |
| 2.9 รหัสแอสกี (ASCII Code) | 26 |
| บทที่ 3 การออกแบบและการสร้าง | |
| 3.1 ส่วนฮาร์ดแวร์ | 27 |
| 3.1.1 ภาคจ่ายไฟ | 27 |
| 3.1.2 ภาครับ-ส่งข้อมูล | 28 |
| 3.1.3 ภาคประมวลผล | 29 |
| 3.1.4 ภาคแสดงผล | 30 |
| 3.2 ส่วนซอฟต์แวร์ | 34 |
| 3.2.1 ฟังก์ชันการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์ | 41 |
| 3.2.2 ฟังก์ชันการทำงานของโปรแกรมที่ใช้ในการแปลงค่า | 42 |
| บทที่ 4 การทดลองและผลการทดลอง | |
| 4.1 ขั้นตอนการทดลอง | 43 |
| 4.2 ผลการทดลองที่ได้ | 45 |
| บทที่ 5 บทวิจารณ์และบทสรุป | |
| ภาคผนวก | 49 |
| กิตติกรรมประกาศ | 74 |
| บรรณานุกรม | 75 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

| | หน้า |
|--|------|
| บทที่ 1 บทนำ | |
| บทที่ 2 ทฤษฎีและหลักการ | |
| รูปที่ 2.1 ภาพลักษณะตำแหน่งจุด 6 จุดของอักษรเบรลล์ | 2 |
| รูปที่ 2.2 แสดงการเปรียบเทียบตัวอักษรภาษาอังกฤษกับตัวอักษรเบรลล์ | 4 |
| รูปที่ 2.3 แสดงการเปรียบเทียบตัวเลขกับตัวเลขเบรลล์ | 5 |
| รูปที่ 2.4 แสดงการเปรียบเทียบสัญลักษณ์กับสัญลักษณ์เบรลล์ | 5 |
| รูปที่ 2.5 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ของ ATME1 | 6 |
| รูปที่ 2.6 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C51 | 7 |
| รูปที่ 2.7 การใช้ไมโครคอนโทรลเลอร์เป็นเอาต์พุตและอินพุตพอร์ต | 8 |
| รูปที่ 2.8 การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 | 9 |
| รูปที่ 2.9 ตารางตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน | 10 |
| รูปที่ 2.10 แสดงตำแหน่งของหน่วยความจำแบบ ไบต์และแบบบิต | 12 |
| รูปที่ 2.11 การเชื่อมต่อด้วยระบบบัสไอส์แควร์ซี และวงจรทางเอาต์พุต | 15 |
| รูปที่ 2.12 จังหวะสัญญาณเวลาบนบัส | 16 |
| รูปที่ 2.13 ไบต์แอดเดรส | 17 |
| รูปที่ 2.14 ลักษณะการรับส่งข้อมูลแบบอนุกรมและแบบขนาน | 18 |
| รูปที่ 2.15 การรับส่งข้อมูลแบบซิมเพล็กซ์, ฮาล์ฟดูเพล็กซ์ และฟูลดูเพล็กซ์ | 18 |
| รูปที่ 2.16 เฟรมรหัสแอสกี ของตัว "A" | 19 |
| รูปที่ 2.17 ขั้วต่อแบบ DB-25 และหน้าที่ของขาต่างๆ | 21 |
| รูปที่ 2.18 ขั้วต่อแบบ DB-9 และหน้าที่ของขาต่างๆ | 21 |
| รูปที่ 2.19 โครงสร้างภายใน MAX232 และการเชื่อมต่อกับ MCS-51 | 22 |
| รูปที่ 2.20 วงจรเรียงกระแสแบบบริดจ์ | 23 |
| รูปที่ 2.21 ลักษณะรูปร่างภายนอก และการใช้งานของไอซี LM 7805 | 24 |
| รูปที่ 2.22 วงจรใช้งานไอซีเรกกูเลเตอร์ LM7805 | 24 |
| รูปที่ 2.23 วงจรไบอัสคงที่ | 25 |
| รูปที่ 2.24 แสดงการเคลื่อนที่ของแกนกระตุ่ง | 26 |
| บทที่ 3 การออกแบบและการสร้าง | |
| รูปที่ 3.1 บล็อกไดอะแกรมของโครงการงาน | 27 |
| รูปที่ 3.2 วงจรภาคจ่ายไฟ | 27 |
| รูปที่ 3.3 วงจรภาครับ-ส่ง ผ่านRS-232 | 28 |
| รูปที่ 3.4 วงจรภาคประมวลผล | 29 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | หน้า |
|--|------|
| รูปที่ 3.5 วงจรภาคแสดงผล | 30 |
| รูปที่ 3.6 อัตราย้ายไฟฟ้ากระแสตรง | 31 |
| รูปที่ 3.7 วงจรขับกระแส | 31 |
| รูปที่ 3.8 บิตที่ใช้ในการควบคุมโซลินอยด์ | 32 |
| รูปที่ 3.9 ภาพของวงจรส่วนขับกระแส | 33 |
| รูปที่ 3.10 ภาพของวงจรส่วนของ โซลินอยด์ | 33 |
| รูปที่ 3.11 ภาพของวงจรที่ต่อเรียบร้อยแล้ว | 34 |
| รูปที่ 3.12 แสดงผังการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์ | 41 |
| รูปที่ 3.13 แสดงผังการทำงานของโปรแกรมที่ใช้แปลงค่า | 42 |
| บทที่ 4 การทดลองและผลการทดลอง | |
| รูปที่ 4.1 เมื่อทำการเปิดโปรแกรม แปลงอักษรเบรลล์ | 43 |
| รูปที่ 4.2 ก่อนทำการเปิดเครื่องจ่ายไฟ | 43 |
| รูปที่ 4.3 หลังทำการเปิดเครื่องจ่ายไฟ | 44 |
| รูปที่ 4.4 ผลที่ได้บนโปรแกรม เมื่อป้อนอินพุต | 44 |
| รูปที่ 4.5 การแสดงผลของโซลินอยด์เมื่อใส่อินพุต “bb” | 45 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

| | หน้า |
|--|------|
| บทที่ 1 บทนำ | |
| บทที่ 2 ทฤษฎีและหลักการ | |
| ตารางที่ 2.1 แสดงเบอร์ไอซีและแรงดันที่ใช้งาน | 24 |
| ตารางที่ 2.2 รายละเอียดตาราง รหัสแอสกี | 26 |
| บทที่ 3 การออกแบบและการสร้าง | |
| ตารางที่ 3.1 แสดงการเปรียบเทียบระหว่าง รหัสแอสกี กับ รหัสที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ตสำหรับตัวอักษรภาษาอังกฤษ | 34 |
| ตารางที่ 3.2 แสดงการเปรียบเทียบระหว่าง รหัสแอสกี กับ รหัสที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ตสำหรับตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ | 36 |
| ตารางที่ 3.3 แสดงการเปรียบเทียบระหว่าง รหัสแอสกี กับ รหัสที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ตสำหรับตัวเลข | 38 |
| ตารางที่ 3.4 แสดงการเปรียบเทียบระหว่าง รหัสแอสกี กับ รหัสที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ตสำหรับตัวอักษร | 39 |
| บทที่ 4 การทดลองและผลการทดลอง | |
| ตารางที่ 4.1 ตารางเปรียบเทียบผลที่ได้จาก โค โอดเปล่งแสงกับอักษรเบรลล์ (ตัวอักษร) | 45 |
| ตารางที่ 4.2 ตารางเปรียบเทียบผลที่ได้จาก โค โอดเปล่งแสงกับอักษรเบรลล์ (ตัวเลข) | 52 |

บทที่ 1

บทนำ

1.1 ความเป็นมาของปริญญานิพนธ์

ในปัจจุบันนี้ มีวิธีการเข้าถึงข้อมูลข่าวสารหลายวิธี ทั้งอินเทอร์เน็ต โทรทัศน์ วิทยุ หนังสือพิมพ์ นิตยสาร รวมไปถึง สื่อสิ่งพิมพ์ทุกชนิด คนทั่วไปสามารถเลือกได้ว่าจะรับข่าวสารจากสื่อใด ขึ้นอยู่กับความพึงพอใจของแต่ละบุคคล แต่ทว่าในสังคมเรา ยังมีคนอีกกลุ่มที่ไม่สามารถเข้าถึงสื่อเหล่านี้ หรือเข้าถึงได้ไม่เต็มร้อยเหมือนคนทั่วไป บุคคลเหล่านี้ คือ ผู้พิการทางสายตา หรือ คนตาบอด ซึ่งแม้จะมีสิทธิและโอกาสทางการศึกษาเท่าเทียมกับบุคคลทั่วไปตามที่ระบุไว้ในรัฐธรรมนูญ แต่ในความเป็นจริงแล้ว สถานศึกษาและหน่วยงานที่เกี่ยวข้องกับการจัดการศึกษาสำหรับผู้พิการทางสายตา ยังขาดแคลนสื่อและเทคโนโลยีการศึกษาที่เหมาะสมและมีคุณภาพ ที่จะช่วยส่งเสริมให้คนพิการได้พัฒนาการเรียนรู้ของตนเองได้อย่างมีประสิทธิภาพ แม้จะมีการประดิษฐ์คิดค้นเครื่องมือและอุปกรณ์ต่างๆ ที่ช่วยให้ผู้พิการทางสายตาสามารถเข้าถึงข้อมูลได้มากขึ้น แต่เครื่องมือและอุปกรณ์เหล่านั้นก็มีราคาสูงมาก และไม่มีการนำมาใช้อย่างแพร่หลาย ข้อจำกัดสำคัญในการรับรู้ข้อมูลนี้ส่งผลให้ผู้พิการทางสายตา ไม่สามารถพัฒนาการเรียนรู้ได้ทัดเทียมกับบุคคลทั่วไป ตลอดจนไม่ทันต่อสภาพเศรษฐกิจและสังคมที่เปลี่ยนแปลงไปอย่างรวดเร็วตลอดเวลา

1.2 วัตถุประสงค์ของปริญญานิพนธ์

เพื่อสร้างอุปกรณ์ในการแปลงอักษรคนตาดีบนหน้าจอคอมพิวเตอร์ให้มีลักษณะคล้ายอักษรเบรลล์ ซึ่งผู้พิการทางสายตาสามารถอ่านและเข้าใจได้โดยใช้งบประมาณที่จำกัด เป็นการเพิ่มโอกาสในการรับรู้ข่าวสารแก่ผู้พิการทางสายตา

1.3 ขอบเขตของปริญญานิพนธ์

ในการสร้างอุปกรณ์แปลงอักษรนั้น ได้กำหนดให้อุปกรณ์มีการรับค่าจากคอมพิวเตอร์โดยผ่าน RS-232 นำค่ามาเปรียบเทียบกับรหัสที่กำหนดไว้ ก่อนจะแสดงผลเป็นอักษรลักษณะคล้ายอักษรเบรลล์ที่ผู้พิการทางสายตาสามารถอ่านได้โดยการสัมผัส โดยใช้แท่งเหล็กที่มีลักษณะนูนขึ้นมาจากพื้นผิวเรียบ โดยใช้โซลินอยด์แบบพุช-พูล (push-pull Solenoid) เป็นแท่งเหล็กที่สามารถควบคุมให้นูนขึ้นหรือราบลงได้ตามลักษณะของตัวอักษร

บทที่ 2 ทฤษฎีและหลักการ

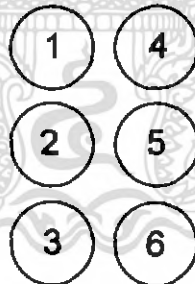
2.1 ความเป็นมาและลักษณะของอักษรเบรลล์

2.1.1 ความเป็นมาของอักษรเบรลล์

อักษรเบรลล์เริ่มต้นขึ้นโดยครูตาบอดชาวฝรั่งเศส หลุยส์ เบรลล์ เป็นผู้ประดิษฐ์อักษรสำหรับคนตาบอดได้ใช้อ่าน โดยใช้นิ้วสัมผัส เขาได้รับความคิดมาจากการส่งข่าวสารทางทหารในเวลากลางคืนของกัปตันชาร์ลส์ บาร์นีย์เออร์ ซึ่งใช้กระดาษแข็งบีบเป็นรหัสจุดและขีด โดยได้พัฒนามาเป็นระบบ 6 จุดซึ่งสามารถจัดกลุ่มของจุดได้ถึง 63 แบบใช้แทนอักษรตาดี อักษรเบรลล์ได้เข้ามาในประเทศไทยและพัฒนาให้ใช้กับภาษาไทยจากการนำมาเผยแพร่ของ เจมีวีฟ คอลฟิลด์ สุภาพสตรีอเมริกันตาบอด ในปี พ.ศ. 2482 จึงเป็นการเปิดศักราชแห่งการเรียนรู้หนังสือของคนตาบอดไทยคั้งแต่นั้นมา รวมทั้งได้ริเริ่มก่อตั้งโรงเรียนคนตาบอดกรุงเทพและมูลนิธิช่วยคนตาบอดแห่งประเทศไทย ทำให้คนตาบอดไทยได้รับความรู้จากการอ่านหนังสือผ่านอักษรเบรลล์คั้งแต่นั้นเป็นต้นมา

2.1.2 ลักษณะของอักษรเบรลล์

อักษรเบรลล์จะมีลักษณะเป็นจุดนูนเล็กๆ ใน 1 ช่องประกอบด้วยจุด 6 ตำแหน่งซึ่งนำมาจัดสลับไปมาเป็นรหัสแทนอักษรตาดี หรือสัญลักษณ์ทางคณิตศาสตร์ วิทยาศาสตร์ ไม้คนดนตรี ฯลฯ ได้ การเขียนใช้เครื่องมือเฉพาะเรียก สเตท และคินสอ การพิมพ์ใช้เครื่องพิมพ์เรียกว่า เบรลเลอร์ ใช้กระดาษหนาขนาดกระดาษวาดรูป



รูปที่ 2.1 ภาพลักษณะตำแหน่งจุด 6 จุดของอักษรเบรลล์

2.1.2.1 กรณีที่ข้อมูลเป็นอักษรภาษาอังกฤษ

การเขียนหนังสือเบรลล์ที่ใช้ในการแทนความหมายของตัวอักษรภาษาอังกฤษนั้น มีอยู่ด้วยกัน 2 ลักษณะ คือ

ลักษณะที่ 1: อักษรภาษาอังกฤษตัวพิมพ์ใหญ่

ลักษณะที่ 2: อักษรภาษาอังกฤษตัวพิมพ์เล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับหนังสือเบรลล์นั้นไม่ว่าจะเป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็กมีลักษณะที่เหมือนกัน ดังนั้นเงื่อนไขที่จะทำให้ผู้อ่านที่เป็นคนตาบอดรู้ว่า ตัวอักษรภาษาอังกฤษที่อ่านอยู่นั้น เป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก ก็สังเกตได้โดย ถ้าอักษรเบรลล์ที่ใช้ในการแทนความหมายของตัวอักษรภาษาอังกฤษ มีอักษรเบรลล์นำหน้าเพิ่มขึ้นมาเป็นตัว “;” ซึ่งจะมีรหัสเครื่องหมายเป็น (44) ก็จะเข้าใจว่าตัวอักษรภาษาอังกฤษนั้นเป็นตัวพิมพ์ใหญ่ แต่ถ้าไม่มีอักษรเบรลล์ที่ใช้นำหน้านั้น ผู้อ่านก็จะเข้าใจว่าตัวอักษรภาษาอังกฤษที่กำลังอ่านอยู่นี้เป็นตัวพิมพ์เล็ก

ส่วนการใช้ภาษาไทยและภาษาอังกฤษในข้อความของอักษรเบรลล์นั้น เมื่อเป็นคำภาษาอังกฤษแล้วต้องเว้นวรรคจากข้อความภาษาไทย ผู้อ่านก็จะสามารถที่จะอ่านเข้าใจได้

ตัวอย่างแสดงลำดับการเขียนหนังสือเบรลล์ จากข้อมูลภาษาอังกฤษที่เป็นตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก ประกอบคำอธิบาย ดังนี้ เช่น

พิมพ์คำว่า “What” จะมีการเพิ่มตัว “;” นำหน้าตัวพิมพ์ดังนี้ “,w-h-a-t”

พิมพ์คำว่า “book” จะไม่มีการเพิ่มตัว “;” นำหน้าตัวพิมพ์ดังนี้ “b-o-o-k”

พิมพ์คำว่า “BOOK” จะมีการเพิ่มตัว “;” นำหน้าตัวพิมพ์ดังนี้ “,B-O-O-K”

2.1.2.2 กรณีข้อมูลเป็นตัวเลข

สำหรับกรณีที่ข้อมูลเป็นตัวเลขนี้ การเขียนอักษรเบรลล์ในการใช้แทนความหมายของตัวเลขนั้น แบ่งออกเป็น 2 ลักษณะ คือ

ลักษณะที่ 1: ข้อมูลตัวเลขที่ใช้ในข้อความ

ลักษณะที่ 2: ข้อมูลตัวเลขที่เป็นทศนิยมหรือใช้ในแบบคณิตศาสตร์

โดยการเขียนของอักษรเบรลล์เมื่อเป็นข้อมูลตัวเลขที่ใช้ในข้อความนั้นจะเติมเครื่องหมาย # นำหน้าตัวเลขเหล่านั้น ส่วนตัวเลขก็ต้องเทียบเคียงด้วยตัวอักษรภาษาอังกฤษดังต่อไปนี้

1 = A, 2 = B, 3 = C, 4 = D, 5 = E, 6 = F, 7 = G, 8 = H, 9 = I, 0 = J

ส่วนข้อมูลตัวเลขที่ใช้กับแบบคณิตศาสตร์นั้นจะต้องเติมเครื่องหมาย # นำหน้าตัวเลขเหมือนกัน แต่จะจัดเรียงข้อมูลตัวเลขไว้แบบเดิม

ตัวอย่างแสดงลำดับการเขียนหนังสือเบรลล์จากข้อมูลที่เป็นตัวเลขดังนี้ เช่น

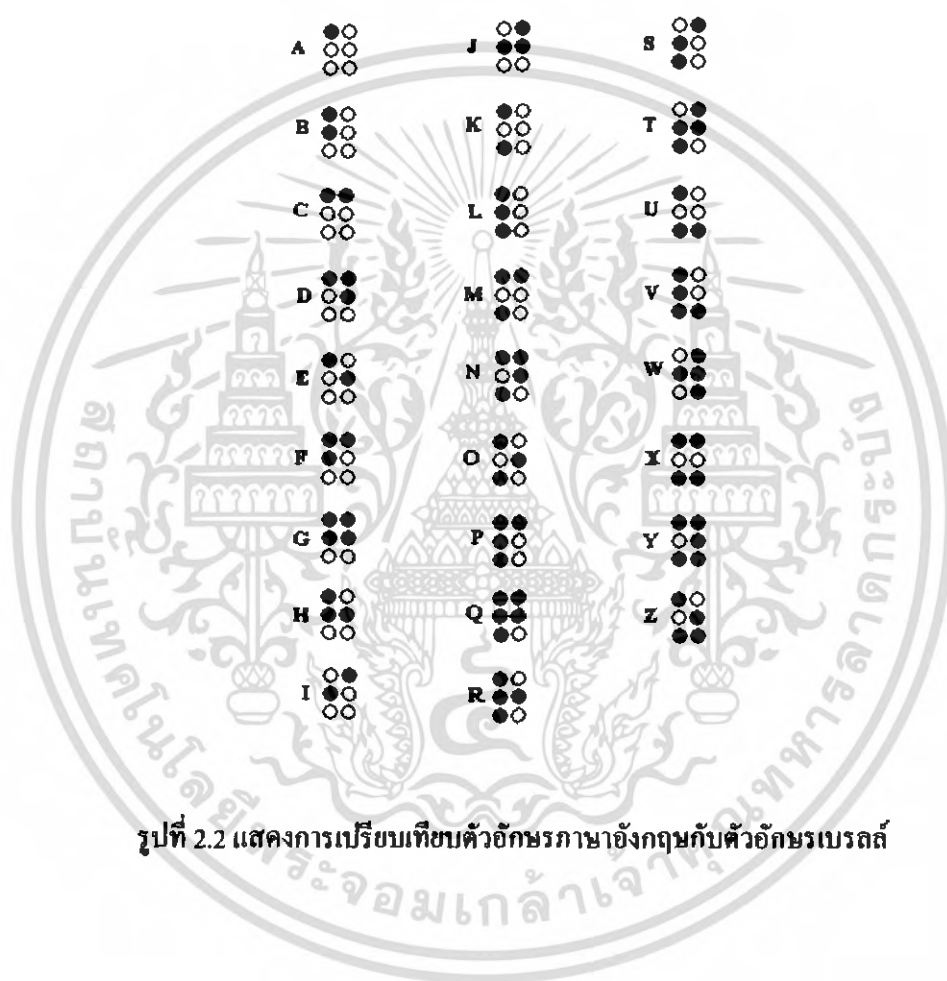
“หน้า 35” จะมีการเพิ่มรหัส “#” นำหน้าตัวพิมพ์ดังนี้ “h-n-ไม้โท-สระอา-#CE”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเลข “12” จะมีการเพิ่มรหัส “#” นำหน้าตัวพิมพ์ดังนี้ “#12”

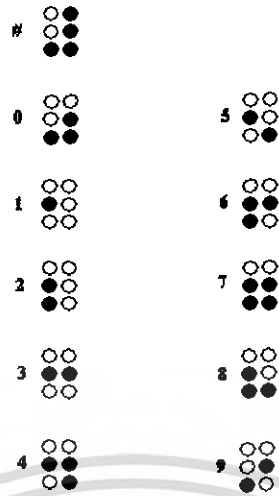
ตัวเลข “9578” จะมีการเพิ่มรหัส “#” นำหน้าตัวพิมพ์ดังนี้ “#9578”

คั้งที่ได้อธิบายถึงหลักในการแปลงจากคำภาษาอังกฤษ และตัวเลข เป็นอักษรเบรลล์ไปแล้วนั้นทำให้ทราบถึงหลักการแปลงอักษรเบรลล์โดยรวมได้อย่างถูกต้อง สำหรับตัวอักษรภาษาอังกฤษ และตัวเลขกับสัญลักษณ์ แต่ละตัวเปรียบเทียบกับอักษรเบรลล์นั้นได้แสดงไว้ดังรูปที่ 2.2, รูปที่ 2.3 และรูปที่ 2.4 ตามลำดับ

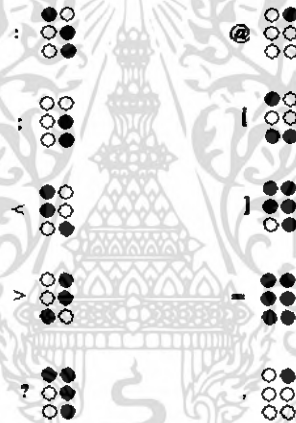


รูปที่ 2.2 แสดงการเปรียบเทียบตัวอักษรภาษาอังกฤษกับตัวอักษรเบรลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงการเปรียบเทียบตัวเลขกับตัวเลขเบรลล์



รูปที่ 2.4 แสดงการเปรียบเทียบสัญลักษณ์กับสัญลักษณ์เบรลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

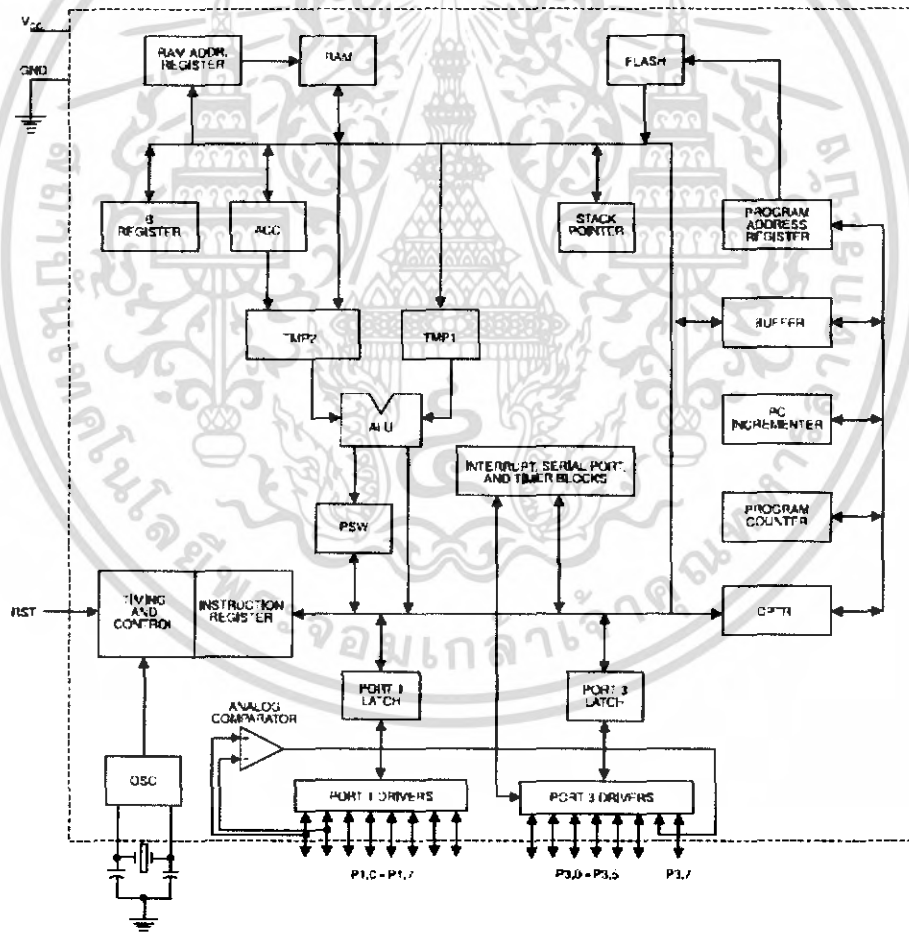
2.2 ทฤษฎีไมโครคอนโทรลเลอร์ MCS-51

2.2.1 คุณสมบัติพื้นฐานของ MCS-51

คุณสมบัติของไมโครคอนโทรลเลอร์ AT89C51

- มีหน่วยความจำโปรแกรมชนิด Flash Memory ขนาด 4 Kbytes
- มีหน่วยความจำแรม 8 Bit ขนาด 128 Byte
- ทำงานที่ความเร็วนาฬิกาสูงสุด 24 MHz
- มีอินพุทเอาต์พุทพอร์ตขนาด 15 บิต
- มีสัญญาณอินเทอร์รัพท์ได้ 3 แหล่ง
- มีพอร์ตสื่อสารแบบอนุกรม 1 ช่อง(UART)
- มีวงจรเปรียบเทียบสัญญาณอนาล็อก (Analog Comparator Input) 1 ช่อง
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

2.2.2 ลักษณะการจัดขาของ MCS-51



รูปที่ 2.5 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51

แบบแฟลช ของ ATMEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | |
|-----------|----|----|------|
| P1.0 | 1 | 40 | VCC |
| P1.1 | 2 | 39 | P0.0 |
| P1.2 | 3 | 38 | P0.1 |
| P1.3 | 4 | 37 | P0.2 |
| P1.4 | 5 | 36 | P0.3 |
| P1.5 | 6 | 35 | P0.4 |
| P1.6 | 7 | 34 | P0.5 |
| P1.7 | 8 | 33 | P0.6 |
| RST | 9 | 32 | P0.7 |
| P3.0/RXD | 10 | 31 | EA |
| P3.1/TXD | 11 | 30 | ALE |
| P3.2/INT0 | 12 | 29 | PSEN |
| P3.3/INT1 | 13 | 28 | P2.7 |
| P3.4/T0 | 14 | 27 | P2.6 |
| P3.5/T1 | 15 | 26 | P2.5 |
| P3.6/WR | 16 | 25 | P2.4 |
| P3.7/RD | 17 | 24 | P2.3 |
| XTAL2 | 18 | 23 | P2.2 |
| XTAL1 | 19 | 22 | P2.1 |
| VSS | 20 | 21 | P2.0 |

รูปที่ 2.6 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C51

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5 โวลต์

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 1 (P1.0-P1.7) มีขา 8 ขา แต่ละขาเรียกได้เป็น 1 บิตสามารถกำหนดให้เป็นทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย

ขาพอร์ต 3 (P3.0-P3.7) มีขา 7 ขา แต่ละขาเรียกได้เป็น 1 บิต แต่ในส่วนของวงจรรภายในไอซีจะมีขาของพอร์ต 3 อยู่ทั้งหมด 8 ขา เพียงแต่ขา P 3.6 จะไม่ได้ต่อออกมาใช้งานภายนอก แต่ใช้เป็นขารับสถานะของผลการเปรียบเทียบสัญญาณขาเข้าอนาล็อก (Analog Comparator Input) ระหว่างพอร์ต P1.0 และ P1.1 จากภายนอกของไอซี ดังนั้นขาทั้ง 7 ขาที่ต่อใช้งานสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นตอนต่อไปนี้

- P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD
- P3.1 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD
- P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{\text{INT0}}$
- P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา $\overline{\text{INT1}}$
- P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทมเมอร์จากภายนอกช่อง 0 หรือขา T0
- P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทมเมอร์จากภายนอกช่อง 1 หรือขา T1
- P3.7 ใช้เป็นขาอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

***P3.6 อยู่ในไอซีไม่ได้ค้อออกมาภายนอก แต่ใช้เป็นขาจับสถานะของการเปรียบเทียบ สัญญาณขาเข้าจากภายนอก ระหว่างพอร์ต P1.0 และ P1.1 จากภายนอก

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยใช้การป้อนสัญญาณเพื่อรีเซต สถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมกซ์ซินไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกา ยังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา XTAL1 และ XTAL2 เป็นขาสำหรับค้อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.2.3 พอร์ตอินพุทและพอร์ตเอาต์พุท

พอร์ต คือ แอดเดรสหนึ่งที่ได้รับคำสั่งไว้เพื่อการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อขึ้นอยู่กับทิศทางการไหลของข้อมูล เมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก จากรูปที่ 2.7(a) เป็นการใช้ไมโครคอนโทรลเลอร์เป็นเอาต์พุทพอร์ต และจากรูป 2.7(b) เป็นการใช้ไมโครคอนโทรลเลอร์เป็นอินพุทพอร์ต



รูปที่ 2.7 การใช้ไมโครคอนโทรลเลอร์เป็นเอาต์พุทและอินพุทพอร์ต

2.2.3.1 การใช้งานพอร์ตเป็นอินพุท

การใช้งานพอร์ตเป็นการอินพุทข้อมูลจะต้องเริ่มต้นด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตนั้นก่อนเป็นอันดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่จับสัญญาณเอาต์พุทของบิตนั้น ทำให้ขาสัญญาณของบิตถูกค้อเข้ากับตัวต้านทานซึ่งทำหน้าที่พูลอัพ (Pull-up) ภายใน ซึ่งมีผลทำให้บิตนั้นของพอร์ต 1, 2 และ 3 เป็นสถานะลอจิกสูง ตัวต้านทานนี้มีค่าประมาณ 50 กิโลโห์ม ซึ่งเป็นค่าที่สูงมาก และทำให้อุปกรณ์ภายนอกสามารถจับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับบิตของพอร์ต 0 นั้นแม้ว่าจะมีหลักการการทำงานที่คล้ายคลึงกันกับบิตของพอร์ตอื่นๆ แต่เนื่องจากไม่มีตัวต้านทานซึ่งทำหน้าที่พูลอัพภายในไว้ ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่จับสัญญาณเอาต์พุทนั้นหยุดการทำงาน ก็จะเป็นผลให้สัญญาณนี้อยู่ในสถานะอิมพีแดนซ์สูงแทน

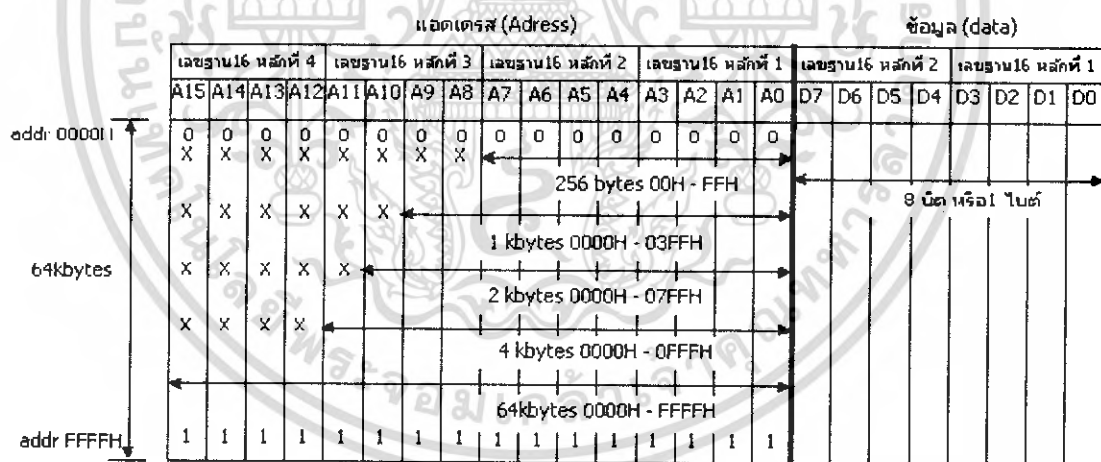
2.2.3.2 การใช้งานพอร์ตเป็นเอาต์พุต

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับ ฟลิปฟลอป ซึ่งจะค้างค่านีไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสถานะลอจิกเป็นลอจิกต่ำด้วย

ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ต 1, 2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตนั้นหยุดทำงาน มีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัวต้านทานที่พูล์อ์พอยู่ภายในนั้น แต่สำหรับการใช้งานในแต่ละบิตทางพอร์ต 0 นั้นจะมีผลแตกต่างออกไป โดยขาสัญญาณจะมีสถานะอิมพีแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านทานภายใน เชื่อมต่ออยู่นั่นเอง ดังนั้นการใช้งานพอร์ต 0 เป็นการนำข้อมูลออกจากเอาต์พุต จึงจำเป็นต้องใช้ตัวต้านทานภายนอกพูล์อ์พสัญญาณไว้กับลอจิกสูงแทน

2.2.4 หน่วยความจำโปรแกรมของ MCS-51

หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory) หรือหน่วยความจำรหัสคำสั่ง (Code Memory) จะทำหน้าที่เก็บชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ปฏิบัติตามคำสั่งนั้นๆ AT89C51 จะมีหน่วยความจำที่เก็บโปรแกรมได้ 4 กิโลไบต์ หน่วยความจำจะเป็นลักษณะแบบแฟลช ที่มีคุณสมบัติในการใช้งานโดยสามารถจะทำการลบข้อมูลด้วยไฟฟ้า และเก็บข้อมูลเข้าไว้ในตัวไมโครคอนโทรลเลอร์ได้กว่า 1000 ครั้ง

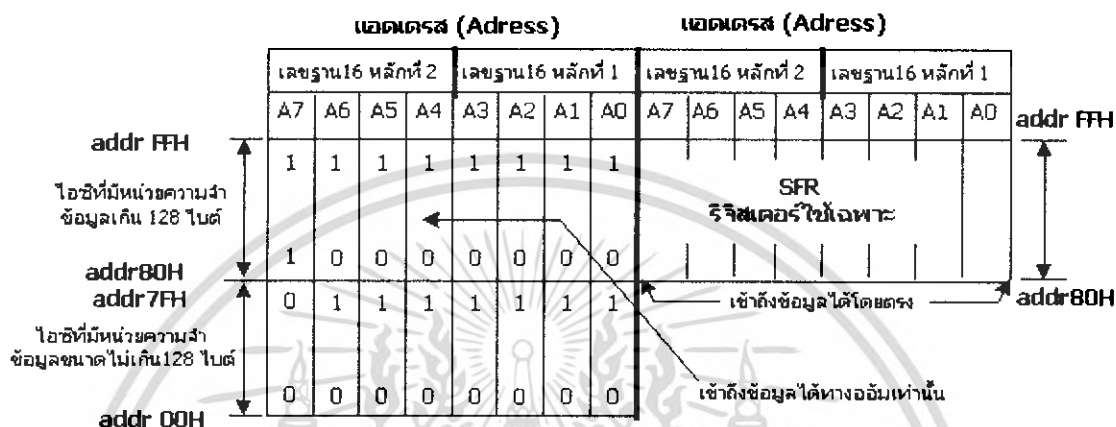


รูปที่ 2.8 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

ส่วนของแอดเดรส (ADDRESS) ไม่สามารถที่จะใช้ตำแหน่งเดียวกันได้ แต่ข้อมูล (DATA) สามารถที่จะมีข้อมูลเหมือนกันได้

2.2.5 หน่วยความจำข้อมูลของ MCS-51

หน่วยความจำข้อมูล (RAM) จะทำหน้าที่เก็บรักษาข้อมูล โดยข้อมูลอาจจะเก็บค่าหลังจากไมโครคอนโทรลเลอร์ทำการประมวลผล หรือเก็บค่าข้อมูลที่จะให้กับไมโครคอนโทรลเลอร์ประมวลผล ในขณะนั้นและจะทำหน้าที่เป็น สแตก (Stack) บางส่วน ไมโครคอนโทรลเลอร์ AT89C51 จะมีหน่วยความจำที่เก็บข้อมูลได้ 128 ไบต์



รูปที่ 2.9 ตารางตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายในยังแบ่งส่วนของการใช้งานได้อีกเป็นสองส่วนคือ หน่วยความจำข้อมูลภายใน 128 ไบต์จะเป็นหน่วยความจำที่ใช้งานทั่วไปอยู่ที่ตำแหน่งแอดเดรส 00H-7FH และหน่วยความจำในตำแหน่งแอดเดรสที่ 80H-FFH ซึ่งจะเป็นส่วนของรีจิสเตอร์เฉพาะ (Special Function Register)

โดยหน่วยความจำข้อมูลภายในที่ตำแหน่งแอดเดรส 00H-7FH ก็ยังสามารถแบ่งออกเป็น ส่วนย่อยได้ดังนี้

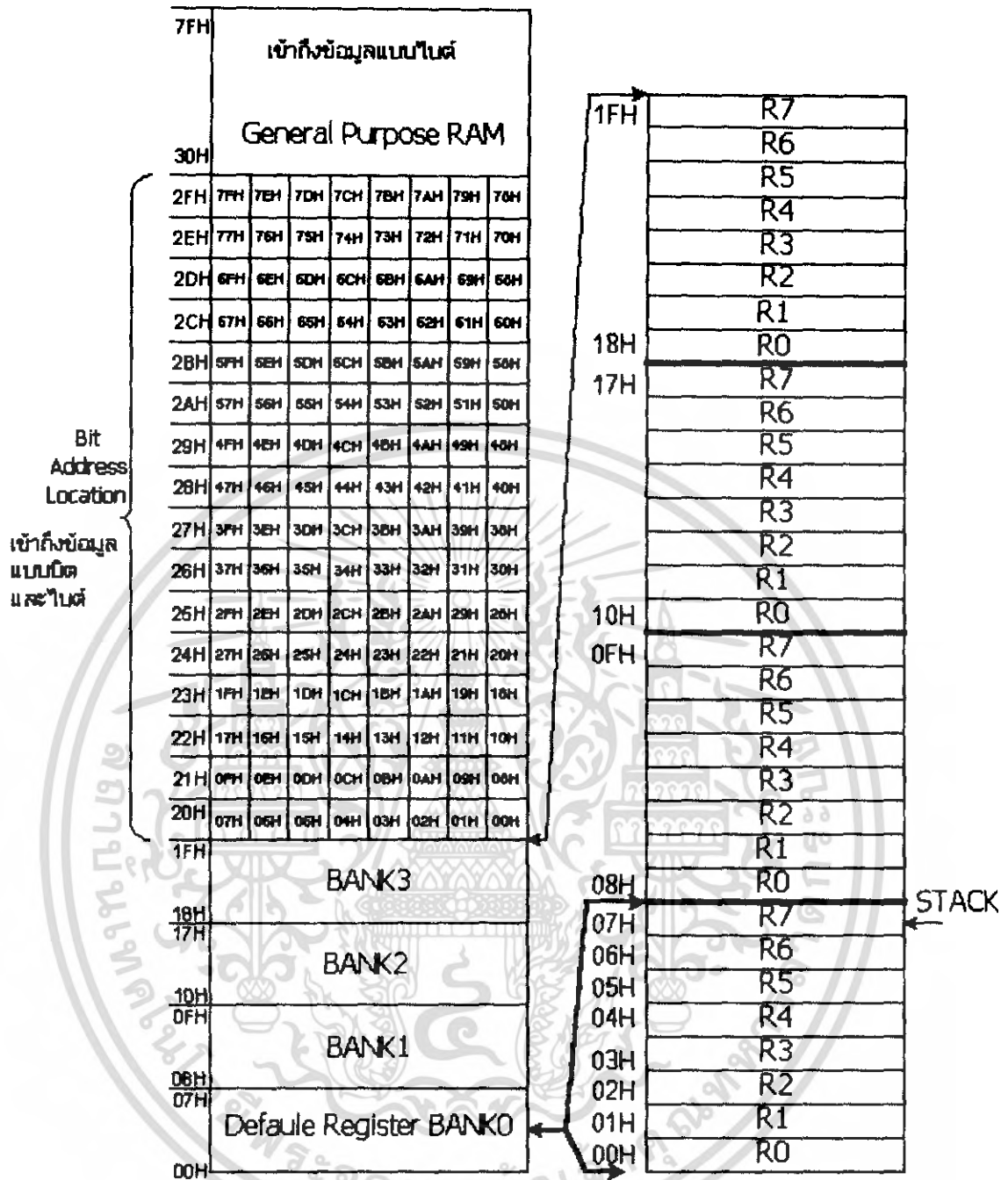
-พื้นที่ในหน่วยความจำข้อมูลตำแหน่งที่ 00H-1FH จำนวน 32 ไบต์ จะถูกแบ่งออกเป็น 4 กลุ่ม เรียกว่า แบงก์ (Bank) และในแต่ละแบงก์จะมี 8 ไบต์ พื้นที่ในแต่ละแบงก์จะถูกใช้งานเป็นรีจิสเตอร์ที่ใช้งานทั่วไป (R0-R7) โดยที่รีจิสเตอร์ R0 จะอยู่ในตำแหน่งแรกของแต่ละแบงก์ และรีจิสเตอร์ R7 จะอยู่ในตำแหน่งสุดท้ายของแต่ละแบงก์ ในการนำไปใช้งานจะเลือกใช้รีจิสเตอร์ได้เพียงแบงก์เดียว และเลือกใช้พื้นที่ของรีจิสเตอร์ R0-R7 ในแบงก์ใดๆก็ได้ โดยการกำหนดค่าของข้อมูลที่รีจิสเตอร์ PSW ในส่วนของรีจิสเตอร์เฉพาะ หากไม่มีการกำหนดใดๆเลย เมื่อทำการรีเซตให้กับไมโครคอนโทรลเลอร์จะถูกกำหนด ให้เริ่มต้นใช้งานที่รีจิสเตอร์ R0-R7 ในหน่วยความจำตำแหน่งแบงก์ 0 ให้เอง

-พื้นที่ในหน่วยความจำข้อมูลภายในตำแหน่งแอดเดรสที่ 20H-2FH จำนวน 16 ไบต์ เป็นส่วนที่ใช้งานในลักษณะการเข้าถึงข้อมูลแบบไบต์หรือแบบบิตได้ และสามารถอ้างตำแหน่งแบบบิตได้โดยตรง เพียงแต่ระบุตำแหน่งหรือชื่อของบิตนั้นๆได้ ซึ่งจะมีอยู่จำนวน 128 บิต แต่ละบิตจะมีหมายเลขตำแหน่งของบิตคือ 00H-7FH โดยตำแหน่งบิตที่ 00H ก็คือข้อมูลบิตที่ต่ำสุดในตำแหน่งแอดเดรสที่ 20H หรืออาจเรียกว่า (20H.1) และตำแหน่งของบิตที่ 7FH คือข้อมูลบิตที่สูงสุดในตำแหน่งแอดเดรสที่ 2FH หรืออาจเรียกว่า (20H.7) การอ้างตำแหน่งบิตจะทำให้โปรแกรมทำงานได้เร็วขึ้น

-พื้นที่ในหน่วยความจำข้อมูลภายในตำแหน่งแอดเดรสที่ 30H-7FH จะเป็นพื้นที่ของหน่วยความจำใช้งานทั่วไป และการติดต่อกับข้อมูลในตำแหน่งต่างๆของหน่วยความจำส่วนนี้จะอ้างตำแหน่งข้อมูลในลักษณะแบบไบต์เท่านั้นและพื้นที่ส่วนนี้อาจใช้เป็นสแตคได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 รีจิสเตอร์หน้าที่พิเศษ

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์หรือพอร์ตของ 89C51 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้อีก

-แอกคิวมูเลเตอร์ (Accumulator)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บข้อมูลที่ส่งให้หน่วยทำงานในซีพียู และเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ A

-รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณหารตัวเลข ในกรณีที่ไมใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

-โปรแกรมเคาน์เตอร์ (Program Counter)

เป็นรีจิสเตอร์ที่ใช้สำหรับการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

-สแตคพอยน์เตอร์ (Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแตค สำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม ค่าเริ่มต้นของสแตคจะอยู่ที่ตำแหน่ง 07H การใช้งานในโปรแกรมจะเรียกว่ารีจิสเตอร์ SP

-ตัวชี้ข้อมูลหรือเคาต์พอยน์เตอร์ (Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิต สองตัว คือ รีจิสเตอร์ DPH และ รีจิสเตอร์ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจเป็นแอดเดรสของอุปกรณ์ภายนอก

-โปรแกรมสเตตัสเวิร์ด (PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสถานะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์ ของรีจิสเตอร์ที่ใช้งานด้วย

-รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุทเอาต์พุทโดยตรงซึ่งเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะการอินพุทหรือการเอาต์พุทข้อมูลได้

-รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล

-รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ควบคุมการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรแกรมเซ็นเซอร์ การกำหนดอัตราทวิคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรมและแฟล็ก สถานะการทำงานทั่วไป

-รีจิสเตอร์ IP, IE, TMOD, SCON

เป็นกลุ่มรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอินเทอร์รีปต์ต่างๆ

2.3 ทฤษฎีเกี่ยวกับระบบบัสแบบไอสแควร์ซี (I²C)

เนื่องจากการต่อบัสข้อมูลของ MCS-51 กับชิปที่ใช้เป็นพอร์ตจะต้องใช้บัสข้อมูลแบบขนาน ซึ่งจะทำให้ต้องใช้สัญญาณในการเชื่อมต่อเป็นจำนวนมาก จึงได้มีการพัฒนาวิธีการเชื่อมต่อชิปไอซีต่างๆ เข้าด้วยกันโดยใช้การสื่อสารแบบพอร์ตอนุกรม ซึ่งจะทำให้ช่วยลดสายสัญญาณที่ใช้ในการสื่อสารข้อมูลลงได้มาก ระบบบัสแบบหนึ่งที่มีนิยมใช้กันคือ ระบบบัสแบบไอสแควร์ซี ซึ่งย่อมาจาก Inter-IC Communication ที่พัฒนาโดยบริษัท ฟิลลิปส์ (Phillips) สำหรับใช้ในการเชื่อมต่อชิปต่างๆ กับระบบไมโครคอนโทรลเลอร์โดยใช้สัญญาณแบบอนุกรมเพียงสองเส้น โดยเส้นหนึ่งเป็นสายข้อมูลมีชื่อว่า สายสัญญาณอนุกรม (SDA : Serial Data Line) อีกเส้นหนึ่งเป็นสายสัญญาณนาฬิกาสำหรับกำหนดจังหวะการสื่อสารข้อมูล มีชื่อว่า สายสัญญาณนาฬิกาอนุกรม (SCL : Serial Clock line)

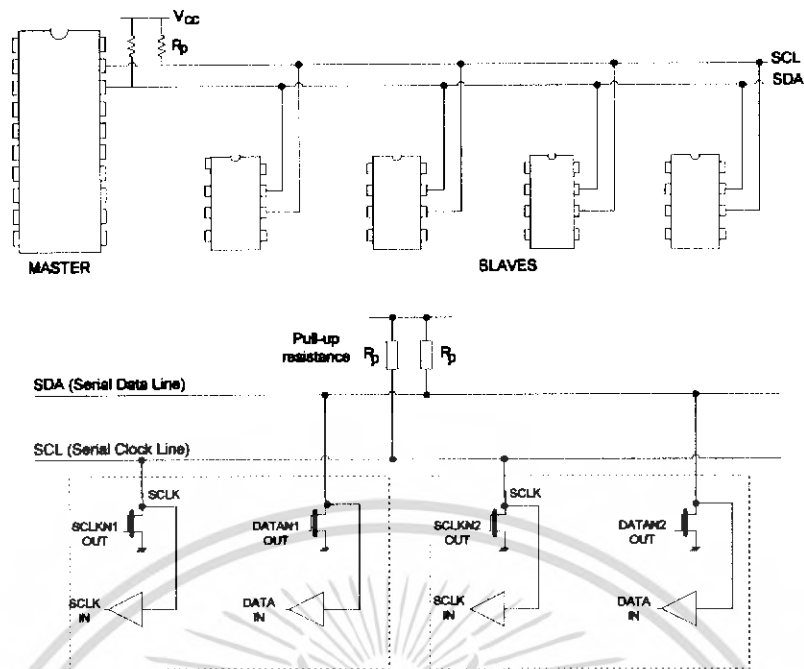
2.3.1 คุณสมบัติทั่วไปของบัสไอสแควร์ซี

สายสัญญาณอนุกรม และ สายสัญญาณนาฬิกาอนุกรม เป็นสายสัญญาณสองทิศทาง ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5 โวลต์ ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่อยุ่บนบัสไอสแควร์ซี ต้องมีลักษณะเป็นวงจรแคโทดเปิด (Open-drain) หรือ คอลเล็กเตอร์เปิด (Open-collector)

อัตราการถ่ายโอนข้อมูลบนบัสไอสแควร์ซี สูงถึง 100 กิโลบิตต่อวินาที ในโหมดปกติ (Standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (Fast mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัสไอสแควร์ซี จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสายสัญญาณอนุกรม และ สายสัญญาณนาฬิกาอนุกรม ไม่เกิน 400 พิโคฟารัด การเข้าถึงอุปกรณ์บนบัสไอสแควร์ซี มีสองแบบ คือ แบบ 7 บิต (7-bit addressing) หรือ 8 บิต (8-bit addressing)

นอกจากนี้ไอสแควร์ซี สามารถที่จะเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้โดยใช้อุปกรณ์บนบัสไอสแควร์ซี เช่น ตัวหนึ่งอาจใช้ไฟเลี้ยง +5 โวลต์ ในขณะที่อีกตัวหนึ่งอาจใช้ไฟเลี้ยง +12 โวลต์ โดยการต่อร่วมกันบนบัสไอสแควร์ซี สามารถที่จะใช้ได้ลักษณะเดียวกันกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงที่เท่ากัน กล่าวคือ ให้ต่อสายสัญญาณอนุกรม และ สายสัญญาณนาฬิกาอนุกรม ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัพ (R_p) เข้ากับแรงดัน +5 โวลต์ ไว้ด้วยเสมอ ในกรณีที่มิแรงดันไฟกระชากขนาดใหญ่ปะปนเข้ามาในบัสไอสแควร์ซี ที่ขาของสายสัญญาณอนุกรม และที่ขาของสายสัญญาณนาฬิกาอนุกรมของอุปกรณ์แต่ละตัว ต้องต่อตัวต้านทานอนุกรม (R_p) เข้ากับขาของสายสัญญาณอนุกรม และขาของสายสัญญาณนาฬิกาอนุกรม ก่อนจะต่อเข้าสู่บัสไอสแควร์ซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

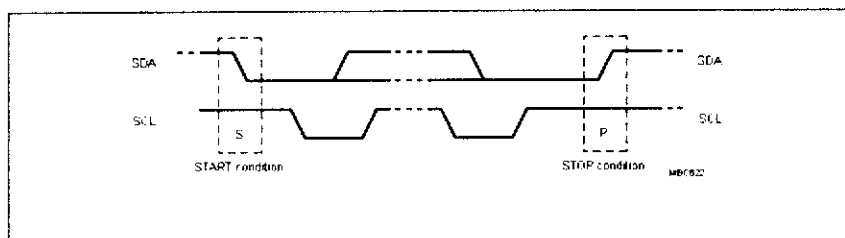


รูปที่ 2.11 การเชื่อมต่อด้วยระบบบัสไอสแควร์ซี และวงจรทางเอาท์พุท

สายสัญญาณสองเส้นที่เชื่อมต่อกันสามารถต่ออุปกรณ์ได้หลายตัว โดยแต่ละตัวจะมีแอดเดรสของตัวเอง การติดต่อกับอุปกรณ์แต่ละตัวจะมีรูปแบบการรับส่งข้อมูลหรือโพรโทคอล (protocol) ที่อุปกรณ์ตัวรู้จัก หากต้องการให้อุปกรณ์ตัวใดรับข้อมูล ตัวส่ง (Transmitter) ซึ่งก็คือ อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล จะส่งแอดเดรสของอุปกรณ์ตัวนั้นไปก่อน ถ้าหากอุปกรณ์ตัวใดมีแอดเดรสตรงกันก็จะรับข้อมูลไป ซึ่งอุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (Receiver) สำหรับตัวอุปกรณ์ที่ต้องการส่งข้อมูล จะเรียกว่ามาสเตอร์ (master) ซึ่งจะเป็นตัวที่สร้างจังหวะสัญญาณต่างๆ บนระบบบัส ส่วนอุปกรณ์ที่ถูกควบคุม จะเรียกว่าสเลฟ (slave)

หากต้องการให้ไมโครคอนโทรลเลอร์เป็นตัวส่งข้อมูล ตัวมันจะทำหน้าที่เป็นมาสเตอร์ ส่งสัญญาณไปบนบัส โดยจะส่งค่าแอดเดรสออกไปก่อน ถ้าแอดเดรสนี้ตรงกับแอดเดรสของชิปตัวใด ชิปตัวนั้นจะส่งสัญญาณตอบรับ (ACK) ออกมา และชิปตัวนั้นก็จะถูกไมโครคอนโทรลเลอร์ควบคุม แต่ถ้าค่าแอดเดรสไม่ตรงกับชิปตัวใดก็จะมีสิ่งใดเกิดขึ้น

การส่งข้อมูลต่างๆ จะต้องเกิดขึ้นเมื่อระบบบัสว่างเท่านั้น โดยทั่วไปแล้วสถานะที่มีบนระบบบัสจะมีอยู่ห้าสถานะ ดังรูปที่ 2.12



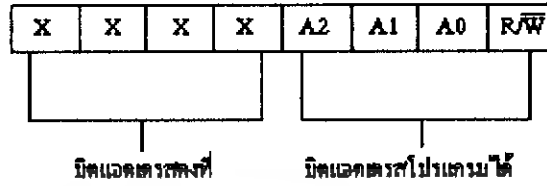
รูปที่ 2.12 จังหวะสัญญาณเวลาบนบัส

1. บัสว่าง (bus not busy) สถานะนี้ค่าลอจิกบนสายสัญญาณอนุกรม และสายสัญญาณนาฬิกาอนุกรม จะเป็นลอจิกสูงทั้งคู่
2. เริ่มส่งข้อมูล (start data transfer) สถานะนี้ สัญญาณนาฬิกาอนุกรมจะเป็นลอจิกสูง แต่สัญญาณอนุกรมจะเปลี่ยนจากลอจิกสูงไปเป็นลอจิกต่ำ เรียกว่าสถานะเริ่มต้น (start)
3. สถานะหยุด (stop) สถานะนี้สัญญาณนาฬิกาอนุกรมจะเป็นลอจิกสูง แต่สัญญาณอนุกรมจะเปลี่ยนจากลอจิกต่ำไปเป็นลอจิกสูง
4. สถานะมีข้อมูล (data valid) สถานะนี้จะอยู่ระหว่างสถานะเริ่มต้นและสถานะหยุด โดยการรับส่งข้อมูลต่างๆ จะเกิดในสถานะนี้ การรับส่งข้อมูลแต่ละบิตจะใช้สัญญาณนาฬิกาหนึ่งลูก โดยข้อมูลบนสายสัญญาณอนุกรมจะต้องคงที่ ขณะที่สายสัญญาณนาฬิกาอนุกรมเป็นลอจิกสูง และบิตข้อมูลในสายสัญญาณอนุกรมจะเปลี่ยนแปลงได้ขณะที่สายสัญญาณนาฬิกาอนุกรมเป็นลอจิกต่ำ ถ้าหากบิตบนสายสัญญาณอนุกรมมีการเปลี่ยนแปลงขณะที่สายสัญญาณนาฬิกาอนุกรมเป็นลอจิกสูง ระบบจะตีความว่าเป็นสถานะเริ่มต้นส่งข้อมูลหรือสถานะหยุดแทน
5. สถานะตอบรับ (Acknowledge) เมื่ออุปกรณ์มาสเตอร์ส่งข้อมูลออกมาครบหนึ่งไบต์แล้ว ในช่วงสัญญาณนาฬิกาอนุกรมลูกที่ 9 มาสเตอร์จะส่งข้อมูลลอจิกสูงออกมา และถ้าตัวรับได้รับข้อมูลครบแล้วมันจะส่งสัญญาณตอบรับ โดยทำให้ระดับลอจิกสูงบนสายสัญญาณอนุกรมให้กลับเป็นระดับลอจิกต่ำ แต่ถ้าตัวรับได้รับข้อมูลไม่ถูกต้อง ตัวรับจะบังคับให้ตัวส่งหยุดในสถานะรอ

ในการโอนถ่ายข้อมูลระหว่างสถานะเริ่มและสถานะหยุดสามารถโอนถ่ายข้อมูลได้ไม่จำกัด แต่เมื่อสเลฟได้รับข้อมูลแต่ละไบต์แล้ว มันจะส่งสัญญาณกลับมาทุกครั้ง

การเชื่อมต่ออุปกรณ์ต่างๆ ที่ใช้ระบบบัสแบบไอสแควร์ซีนั้น เราสามารถใช้อุปกรณ์หลายๆ ตัวมาต่อกับไมโครคอนโทรลเลอร์ได้ แต่จะใช้สายสัญญาณทั้งหมดเพียงสองเส้นเท่านั้น ดังนั้นการติดต่อกับอุปกรณ์แต่ละตัวจะต้องระบุว่าจะต้องการติดต่อกับอุปกรณ์ตัวใด ขั้นตอนการเขียนข้อมูลที่ใช้กับระบบบัสแบบไอสแควร์ซี มีสามขั้นตอน ดังนี้

1. เขียนข้อมูลอ้างอิงแอดเดรส (addressing) ข้อมูลไบต์แรกที่มาสเตอร์จะส่งออกไปคือข้อมูลที่ใช้อ้างอิงแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ โดยทั่วไปแล้วข้อมูลที่ใช้อ้างอิงแอดเดรสจะมีจำนวน 7 บิต ดังรูปที่ 2.13



รูปที่ 2.13 ไบต์แอดเดรส

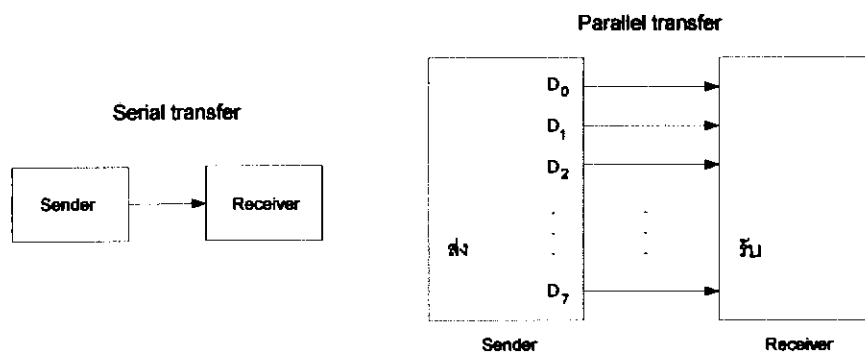
จากรูป จะเห็นว่า บิตค่าสุดท้ายจะเป็นตัวระบุว่าอุปกรณ์ที่ต้องการติดต่อนั้น เราจะใช้อ่านหรือเขียนข้อมูลลงไป ถ้าบิตนี้เป็นลอจิก “0” หมายความว่าเราจะใช้เขียนข้อมูลลงไป ถ้าบิตเป็นลอจิก “1” หมายความว่าใช้อ่านข้อมูล ส่วน 7 บิตบนจะเป็นแอดเดรส ซึ่งแบ่งออกเป็นบิตแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต โดยจะถูกโปรแกรมมาจากโรงงานที่ผลิตชิปแต่ละประเภท ส่วนอีก 3 บิตต่อมาจะเป็นบิตแอดเดรสที่ผู้ใช้สามารถโปรแกรมได้ (programmable address bit) ในกรณีที่มีการต่อชิปเบอร์เดียวกันหลายๆ ตัวในระบบบัสแบบไอสแควร์ซีนี้ จะต้องโปรแกรมให้บิตแอดเดรสที่ผู้ใช้โปรแกรมได้มีค่าแอดเดรสต่างกัน

2. การเขียนไบต์ควบคุม (control byte) ข้อมูลไบต์นี้จะขึ้นอยู่กับอุปกรณ์แต่ละประเภท เพื่อกำหนดการทำงานต่างๆ ของตัวมัน อุปกรณ์บางประเภทอาจจะไม่ต้องมีการเขียนไบต์นี้ก็ได้
3. การเขียนไบต์ข้อมูล (data byte) เป็นข้อมูลที่โอนถ่ายในระบบ ระหว่างสภาวะเริ่มต้นและสภาวะหยุด จะมีการเขียนข้อมูลจำนวนกี่ไบต์ก็ได้

2.4 ทฤษฎีเกี่ยวกับการสื่อสารข้อมูลแบบอนุกรม

2.4.1 พื้นฐานการสื่อสารอนุกรม

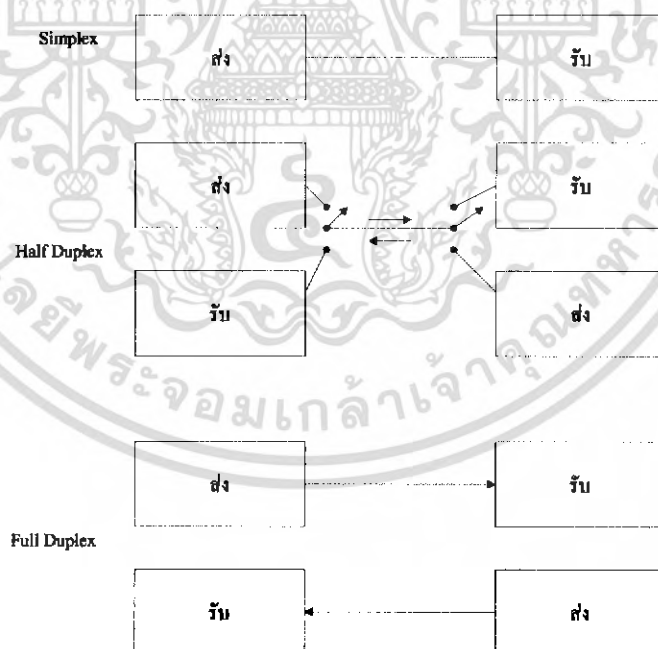
เมื่อไมโครโพรเซสเซอร์ต้องการติดต่อกับอุปกรณ์ภายนอก ตัวมันจะส่งข้อมูลออกมาเป็นขนาดเป็นไบต์ หรือ 8 บิต การโอนถ่ายข้อมูลต่างๆ จะทำแบบขนาน ถ้าต้องการส่งข้อมูลออกไปแบบอนุกรมจะต้องเปลี่ยนข้อมูลแบบขนานนี้ให้เป็นข้อมูลแบบอนุกรมเสียก่อนแล้วจึงส่งออกไป ส่วนการรับข้อมูลแบบอนุกรมนี้อาจจะรับข้อมูลเข้ามาครั้งละบิต และเปลี่ยนข้อมูลให้เป็นข้อมูลแบบขนาน เพื่อส่งให้ไมโครโพรเซสเซอร์ประมวลผลต่อไป ในระบบคอมพิวเตอร์ตัวที่เปลี่ยนข้อมูลอนุกรมเป็นขนานและเปลี่ยนข้อมูลขนานเป็นอนุกรมจะใช้อุปกรณ์ที่มีชื่อว่า UART (Universal Asynchronous Receiver-Transmitter)



รูปที่ 2.14 ลักษณะการรับส่งข้อมูลแบบอนุกรมและแบบขนาน

2.4.2 รูปแบบการสื่อสารข้อมูล

การสื่อสารข้อมูลระหว่างตัวรับและตัวส่งนั้นมีหลายวิธี ถ้าหากตัวส่งทำหน้าที่ส่งอย่างเดียวและตัวรับทำหน้าที่รับอย่างเดียวจะเรียกว่าการสื่อสารแบบซิมเพล็กซ์ (simplex) เช่น การส่งข้อมูลระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ แต่ถ้าหากตัวรับและตัวส่งสามารถรับและส่งข้อมูลได้แต่ทำในเวลาต่างกัน เรียกว่า ฮาล์ฟดูเพล็กซ์ (half duplex) เช่น การสื่อสารแบบเครื่องอินเทอร์เน็ตคอมพิวเตอร์ที่มีสายส่งสัญญาณได้ทางเดียว แต่ถ้าหากตัวรับและตัวส่งสามารถรับส่งข้อมูลได้สองทิศทางในเวลาเดียวกัน จะเรียกว่าเป็นการสื่อสารแบบฟูลดูเพล็กซ์ (full duplex) การสื่อสารแต่ละแบบแสดงได้ ดังรูปที่ 2.15



รูปที่ 2.15 การรับส่งข้อมูลแบบซิมเพล็กซ์, ฮาล์ฟดูเพล็กซ์ และฟูลดูเพล็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับวิธีในการส่งข้อมูลแบบอนุกรมมีสองวิธีคือ การส่งแบบเข้าจังหวะเวลา หรือแบบซิงโครนัส (synchronous) และการส่งแบบไม่เข้าจังหวะเวลา หรือแบบอะซิงโครนัส (asynchronous) โดยการส่งแบบเข้าจังหวะเวลาจะต้องมีการส่งสัญญาณนาฬิกาพร้อมไปด้วยเพื่อควบคุมการรับส่งข้อมูล แต่การส่งแบบไม่เข้าจังหวะเวลาจะไม่ต้องมี แต่จะใช้การกำหนดอัตราเร็วในการรับส่งข้อมูลให้มีค่าเท่ากันที่เรียกว่า อัตราบอด (baud rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที (bit per second : bps)

2.4.2.1 การส่งข้อมูลแบบซิงโครนัส

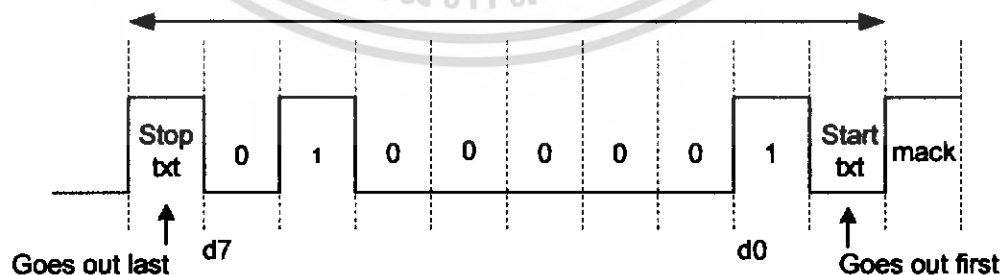
เมื่ออุปกรณ์สองตัวจะสื่อสารข้อมูลซึ่งกันและกันจะต้องกำหนดรูปแบบกฎเกณฑ์การสื่อสารซึ่งกันและกันเพื่อให้เข้าใจกันได้ รูปแบบการรับส่งข้อมูล หรือโพรโทคอลนี้ จะเป็นตัวบอกลักษณะของข้อมูล การเริ่มข้อมูล การจบข้อมูล

การรับส่งข้อมูลแบบซิงโครนัสจะประกอบด้วย ส่วน ดังนี้

1. บิตเริ่มต้น (start bit)
2. บิตข้อมูลอนุกรม
3. บิตตรวจสอบความถูกต้อง
4. บิตสุดท้าย (stop bit)

2.4.2.2 บิตเริ่มต้นและบิตสุดท้าย

การส่งข้อมูลแบบอะซิงโครนัสนั้น จะส่งข้อมูลออกไปเป็นชุด เรียกว่า เฟรม (framing) ภายในเฟรมจะประกอบด้วยข้อมูลหรือรหัสแอสกี (ASCII code) ก็ได้ ในแต่ละเฟรมจะเริ่มต้นด้วยบิตเริ่มต้น (start bit) ที่จะบอกว่าถึงที่ตามมาคือข้อมูล และจะจบข้อมูลด้วยบิตสุดท้าย (stop bit) ที่จะบอกว่าข้อมูลในเฟรมนั้นๆ สิ้นสุดลงแล้ว ในการส่งข้อมูลแบบนี้ถ้าหากยังไม่มีข้อมูลระดับลอจิกที่ส่งจะกลายเป็นลอจิก "1" เรียกว่าสภาวะรอ (waiting stage) ถ้าหากมีข้อมูลส่ง สัญญาณจะเป็นลอจิก "0" ในช่วงเวลาหนึ่งบิตข้อมูลนี้เรียกว่าบิตเริ่มต้น จากนั้นจะตามด้วยข้อมูล เช่น ถ้าหากจะส่งรหัสแอสกี ของตัว "A" จะเป็นดังรูปที่ 2.16 โดยส่งค่าข้อมูล 8 บิต (01000001) ออกไป จากนั้นจะจบด้วยบิตปิดท้ายหรือบิตหยุดที่มีสภาวะเป็นลอจิก "1" โดยบิตหยุดนี้อาจเป็นลอจิก "1" ในช่วงเวลา 1 บิต, 1.5 บิต หรือ 2 บิตก็ได้ ขึ้นกับการกำหนดลักษณะการส่งข้อมูล



รูปที่ 2.16 เฟรมรหัสแอสกี ของตัว "A"

สำหรับข้อมูลที่รับส่งกันนั้นอาจมีขนาด 7 บิต หรือ 8 บิตก็ได้ ขึ้นอยู่กับการออกแบบระบบ และข้อมูลสามารถทำการตรวจสอบความถูกต้องได้ระดับหนึ่ง เรียกว่า การตรวจสอบบิตพาริตี (parity bit) ซึ่งจะเป็นบิตข้อมูลที่จะเพิ่มเข้าไปก่อนบิตปิดท้าย โดยการตรวจสอบบิตพาริตีนี้สามารถกำหนดเป็นพาริตีคี่ (odd) หรือ พาริตีคู่ (even) ก็ได้ การกำหนดบิตพาริตีนี้หมายความว่า บิตข้อมูลทั้ง 7 หรือ 8 บิต ถ้าหากรวมกับบิตพาริตีแล้วจะต้องเป็นลอจิก “1” จำนวนคี่บิต ถ้าหากเป็นพาริตีคู่จะต้องรวมแล้วได้ลอจิก “1” จำนวนคู่บิต ตัวอย่างเช่น ถ้าหากส่งรหัสแอสกีของตัว “A” ออกไป ซึ่งข้อมูลจะเป็น 0100 0001 ถ้ากำหนดเป็นพาริตีคู่ บิตพาริตีจะต้องมีลอจิกเป็น “0” แต่ถ้ากำหนดเป็นพาริตีคี่บิตพาริตีจะต้องเป็นลอจิก “1” การสร้างบิตพาริตีนี้จะถูกสร้างจากภาคส่งของยูเออาร์ที ซึ่งเราสามารถโปรแกรมได้ว่าการส่งนั้นจะเป็นแบบพาริตีคี่หรือคู่ หรือไม่มีพาริตี

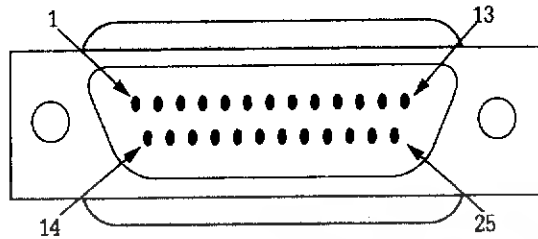
2.4.3 อัตราการรับส่งข้อมูล

ความเร็วในการรับส่งข้อมูลแบบอนุกรมจะบอกเป็นจำนวนบิตต่อวินาที หรืออัตราบอด พอร์ตอนุกรมของ MCS-51 สามารถติดต่อกับเครื่องคอมพิวเตอร์ส่วนบุคคลได้โดยการกำหนดอัตราเร็วให้เท่ากัน ในการสื่อสารทางพอร์ตอนุกรม RS-232 ของคอมพิวเตอร์ส่วนบุคคล ได้กำหนดอัตราเร็วไว้หลายค่าตั้งแต่ 100 ถึง 9600 บิตต่อวินาที สำหรับคอมพิวเตอร์ที่ใช้ไมโครโพรเซสเซอร์เพนเทียมสามารถส่งข้อมูลได้ความเร็วสูงถึง 56 กิโลบิตต่อวินาที

2.4.4 มาตรฐาน RS 232

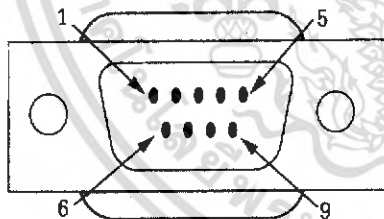
การสื่อสารแบบอนุกรมกับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) มักจะใช้รูปแบบมาตรฐาน RS-232 ซึ่งกำหนดโดย Electronics Industries Association หรือ EIA ในปี 1960 และได้มีการแก้ไขมาตรฐานในปี 1963, 1965 และ 1969 เรียกว่า RS-232A, RS-232B และ RS-232C ตามลำดับ ระดับแรงดันของลอจิกที่ใช้ในการสื่อสาร RS-232 นั้น ลอจิก “1” จะแทนด้วยแรงดัน -3 ถึง -25 โวลต์ ส่วนลอจิก “0” จะแทนด้วยแรงดัน +3 ถึง +25 โวลต์ แต่แรงดันในช่วง +3 โวลต์ จะไม่ถูกกำหนดให้ใช้งานซึ่งจะเห็นว่าแรงดันดังกล่าวไม่สามารถใช้กับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้ โดยทั่วไปแล้วหากต้องการให้ไมโครคอนโทรลเลอร์ติดต่อกับเครื่องคอมพิวเตอร์ส่วนบุคคลตามมาตรฐาน RS-232 จะต้องออกแบบวงจรอิเล็กทรอนิกส์เพิ่มเติม แต่ในปัจจุบันจะใช้ไอซี MAX232 ทำหน้าที่เปลี่ยนระดับแรงดันทางลอจิกให้อยู่ในมาตรฐาน RS-232 แต่ถ้าหากจะให้ไมโครคอนโทรลเลอร์สองตัวสื่อสารกันจะไม่ใช่มาตรฐานนี้ก็ได้

ในเครื่องคอมพิวเตอร์ส่วนบุคคลจะมีขั้วต่อ RS-232 หรือที่เรียกว่า คอนเนกเตอร์ (connector) อยู่สองแบบ คือ ขั้วต่อแบบ DB-25 และขั้วต่อแบบ DB-9



| Pin | Description |
|------|--------------------------------|
| 1 | Protective ground |
| 2 | Transmitted data (TxD) |
| 3 | Received data (RxID) |
| 4 | Request to send (RTS) |
| 5 | Clear to send (CTS) |
| 6 | Data set ready (DSR) |
| 7 | Signal ground (GND) |
| 8 | Data carrier detect (DCD) |
| 9/10 | Reserve for data testing |
| 11 | Unassigned |
| 12 | Secondary data carrier detect |
| 13 | Secondary clear to send |
| 14 | Secondary transmitted data |
| 15 | Transmit signal element timing |
| 16 | Secondary received data |
| 17 | Receive signal element timing |
| 18 | Unassigned |
| 19 | Secondary request to send |
| 20 | Data terminal ready (DTR) |
| 21 | Signal quality detector |
| 22 | Ring indicator |
| 23 | Data signal rate select |
| 24 | Transmit signal element timing |
| 25 | Unassigned |

รูปที่ 2.17 ขั้วต่อแบบ DB-25 และหน้าที่ของขาต่างๆ



| Pin | Description |
|-----|---------------------------|
| 1 | Data carrier detect (DCD) |
| 2 | Received data (RxID) |
| 3 | Transmitted data (TxID) |
| 4 | Data terminal ready (DTR) |
| 5 | Signal ground (GND) |
| 6 | Data set ready (DSR) |
| 7 | Request to send (RTS) |
| 8 | Clear to send (CTS) |
| 9 | Ring indicator (RI) |

รูปที่ 2.18 ขั้วต่อแบบ DB-9 และหน้าที่ของขาต่างๆ

ในระบบ RS-232 มีขาต่างๆ ที่สำคัญ ดังนี้

1. DTR (data terminal ready) เป็นสายสำหรับการทำแฮนด์เชก (handshake) จากอุปกรณ์ปลายทาง หรือ DTE (Data Terminal Equipment) ไปยังสถานีนำส่ง หรือ DCE (Data Communication Equipment)
2. DSR (data set ready) เป็นสายสำหรับการทำแฮนด์เชกจาก DCE ไปยัง DTE
3. RTS (request to send) เมื่ออุปกรณ์ DTE ต้องการส่งข้อมูล มันจะส่งสัญญาณที่ขานี้เป็นลอจิกต่ำออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. CTS (clear to send) เป็นสายสัญญาณสำหรับการทำแฮนด์เชกจาก DCE ไปยัง DTE
5. CDC (carrier detect หรือ data carrier detect)
6. RI (ring indicator) เป็นสายที่ใช้โดยโมเด็ม (modem) เพื่อบอกว่าตัวมันเองได้รับสัญญาณเรียกเข้ามา

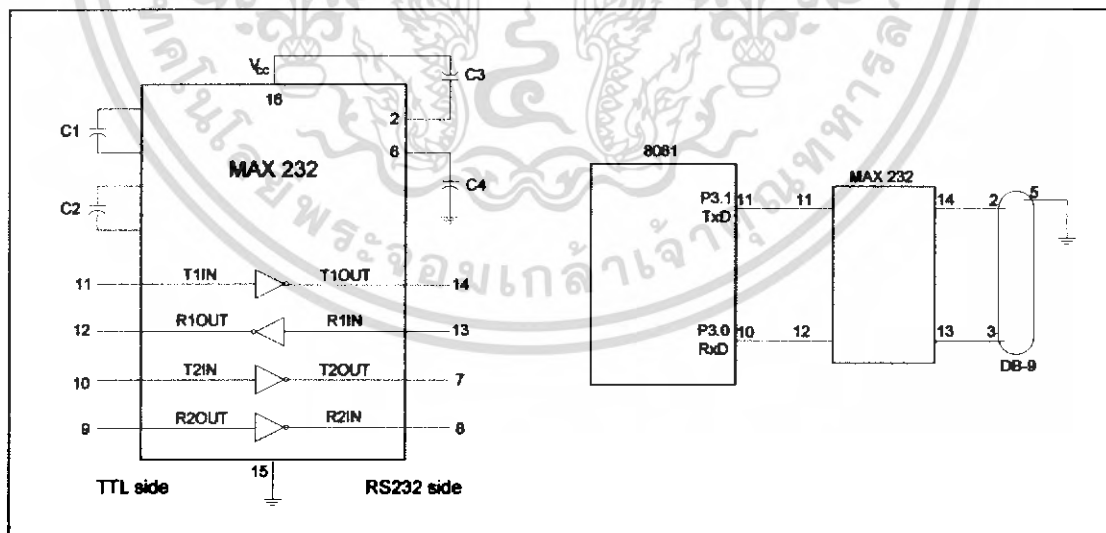
ในเครื่องคอมพิวเตอร์ส่วนบุคคล IBM ที่ใช้ไมโครโพรเซสเซอร์ตระกูล X86 จะมีพอร์ตสื่อสารอนุกรมอยู่สองพอร์ต เรียกว่าพอร์ต COM1 และ COM2 โดยใช้คอนเนกเตอร์แบบ DB-25 และ DB-9 เราสามารถนำพอร์ตทั้งสองนี้มาเชื่อมต่อกับพอร์ตอนุกรมของ 8051 ได้

2.4.5 การต่อ MCS-51 เข้ากับ RS-232

ในการต่อพอร์ตอนุกรมของ MCS-51 กับมาตรฐาน RS-232 จะต้องนำชิป MAX232 มาปรับระดับแรงดันใน 8051 เนื่องจากระดับสัญญาณของการสื่อสาร RS-232 ไม่เป็นไปตามมาตรฐานแรงดัน TTL

ขา RxD และ TxD ของ 8051

พอร์ตที่ใช้ในการรับส่งข้อมูลแบบอนุกรมของ 8051 จะรับส่งออกมาทางขา RxD และ TxD ซึ่งอยู่ในพอร์ต 3 (P3.0 และ P3.1) โดย TxD จะเป็นขาที่ 9 และ RxD จะเป็นขาที่ 10 การรับส่งข้อมูลกับมาตรฐาน RS-232 เราจะต้องติดต่อผ่านขาทั้งสองนี้โดยสามารถนำไอซี MAX232 มาเชื่อมต่อได้ ไอซีตัวนี้ใช้ไฟเลี้ยง +5 โวลต์ เท่ากับไฟเลี้ยงของไมโครคอนโทรลเลอร์ แต่สามารถยกระดับแรงดันตั้งแต่ -25 โวลต์ ถึง +25 โวลต์ได้โดยไม่ต้องใช้แหล่งจ่ายไฟแบบคู่ (dual power supplies)



รูปที่ 2.19 โครงสร้างภายใน MAX232 และการเชื่อมต่อกับ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ทฤษฎีเกี่ยวกับวงจรเรียงกระแส (Rectifier)

วงจรเรียงกระแส คือ วงจรซึ่งทำหน้าที่เปลี่ยนไฟฟ้ากระแสสลับ (Alternating Current : AC) จากด้านอินพุตให้เป็นไฟฟ้ากระแสตรง (Direct Current : DC) ที่ด้านเอาต์พุต แบ่งออกเป็นสองประเภท ดังนี้

1. วงจรเรียงกระแสครึ่งคลื่น
2. วงจรเรียงกระแสเต็มคลื่น

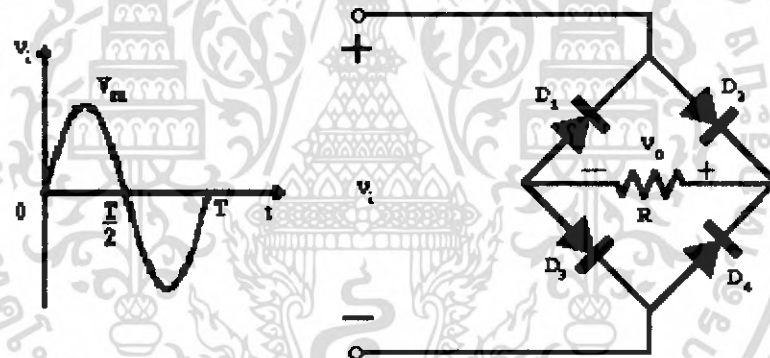
2.5.1 วงจรเรียงกระแสเต็มคลื่น (Full-Wave Rectifier)

วงจรเรียงกระแสเต็มคลื่น หมายถึง วงจรที่ได้รับไฟฟ้ากระแสสลับ จำนวนหนึ่งไซเคิล แล้วเปลี่ยนไปเป็นไฟฟ้ากระแสตรงที่มีค่าเฉลี่ยรูปคลื่นเอาต์พุตครบหนึ่งไซเคิล แบ่งออกเป็นสองประเภท คือ

- วงจรเรียงกระแสแบบบริดจ์ (Bridge Rectifier)
- วงจรเรียงกระแสแบบหม้อแปลงแทปกลาง (Center-tapped Transformer)

2.5.1.1 วงจรเรียงกระแสแบบบริดจ์

วงจรเรียงกระแสแบบบริดจ์ เป็นวงจรเรียงกระแสที่ประกอบด้วยไดโอดสี่ตัว ต่อวงจรเหมือนกับวงจรบริดจ์ ดังรูปที่ 2.20



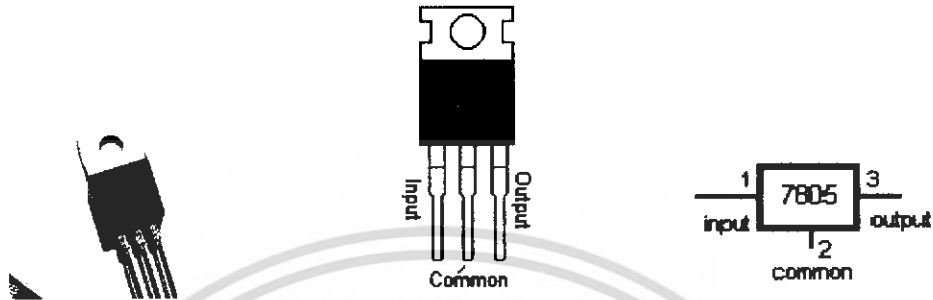
รูปที่ 2.20 วงจรเรียงกระแสแบบบริดจ์

เมื่อจ่ายสัญญาณแรงดันอินพุตให้กับวงจรเรียงกระแสแบบบริดจ์ ขณะสัญญาณบวกเข้าสู่วงจร ไดโอด D_2 และไดโอด D_3 จะได้รับไบอัสตรง ส่วนไดโอด D_1 และไดโอด D_4 จะได้รับไบอัสกลับ ทำให้ไดโอด D_2 และไดโอด D_3 เปรียบเสมือนสวิตช์วงจร ส่วนไดโอด D_1 และไดโอด D_4 เปิดวงจร

ในทำนองเดียวกัน ขณะสัญญาณลบเข้าสู่วงจร ไดโอด D_1 และไดโอด D_4 จะได้รับไบอัสตรง ส่วนไดโอด D_2 และไดโอด D_3 จะได้รับไบอัสกลับ ทำให้ไดโอด D_1 และไดโอด D_4 เปรียบเสมือนสวิตช์วงจร ส่วนไดโอด D_2 และไดโอด D_3 เปิดวงจร

2.6 ทฤษฎีเกี่ยวกับไอซีเรกกูเลเตอร์ (IC Regulator) LM78xx, LM79xx

ไอซีเรกกูเลเตอร์ 78xx เป็นไอซีที่ใช้ในวงจรควบคุมแรงดันให้คงที่ โดยมีค่าแรงดัน 5 โวลต์, 12 โวลต์, 15 โวลต์ สามารถจ่ายกระแสได้ 1 แอมแปร์ ส่วนไอซีเรกกูเลเตอร์ เบอร์ 79xx จะให้แรงดันไฟลบออกมา เช่น -15 โวลต์, -9 โวลต์ เป็นต้น

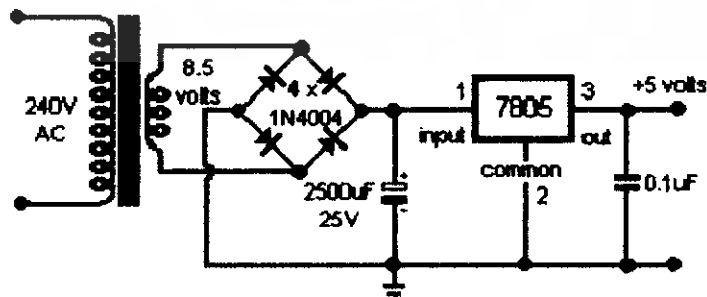


รูปที่ 2.21 ลักษณะรูปร่างภายนอก และขาใช้งานของไอซี LM 7805

ตารางที่ 2.1 แสดงเบอร์ไอซีและแรงดันที่ใช้งาน

| NAME | VOLTAGE |
|--------|-----------|
| LM7805 | +5 volts |
| LM7809 | +9 volts |
| LM7812 | +12 volts |
| LM7815 | +15 volts |
| LM7905 | -5 volts |
| LM7909 | -9 volts |
| LM7912 | -12 volts |
| LM7915 | -15 volts |

การต่อวงจร ใช้งาน ไอซีเรกกูเลเตอร์นี้ สามารถทำได้ดังรูปที่ 2.22



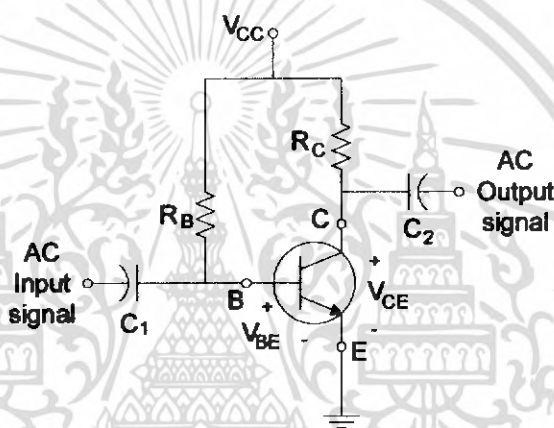
รูปที่ 2.22 วงจร ใช้งาน ไอซีเรกกูเลเตอร์ LM7805

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.22 จะเห็นว่าใช้แหล่งจ่ายไฟฟ้ากระแสสลับ 8.5 โวลต์ มีไดโอดที่ต่อวงจรบริดจ์ทำหน้าที่เป็นวงจรเรกติไฟเออร์ โดยเปลี่ยนไฟฟ้ากระแสสลับ ให้เป็นไฟฟ้ากระแสตรง และกลับขั้วไฟให้ถูกต้อง ตัวเก็บประจุ 2500 ไมโครฟารัด จะทำหน้าที่กรองแรงดันหรือฟิลเตอร์ (Filter) ในการลดค่าแรงดันริบเบิล (Ripple) ที่เป็นการกระเพื่อมของแรงดันไฟตรง ซึ่งเกิดจากการเปลี่ยนแรงดันไฟฟ้ากระแสสลับให้เป็นแรงดันไฟฟ้ากระแสตรง ตัวเก็บประจุ 0.1 ไมโครฟารัดนั้น ทำหน้าที่ป้องกันสัญญาณรบกวนความถี่สูงที่เข้ามาทำความเสียหายให้กับไอซี และแรงดันเอาต์พุตที่ได้ เมื่อใช้ไอซี LM7805 จะมีค่า +5 โวลต์

2.7 ทฤษฎีเกี่ยวกับวงจรไบอัสกระแสตรง (DC Biasing Circuit)

วงจรไบอัสคงที่ (Fixed Bias Circuit) เป็นวงจรไบอัสคงตัวของวงจรแบบอิมิตเตอร์ร่วม การไบอัสของวงจรเกิดจากแหล่งจ่ายแรงดัน V_{CC} เท่านั้น

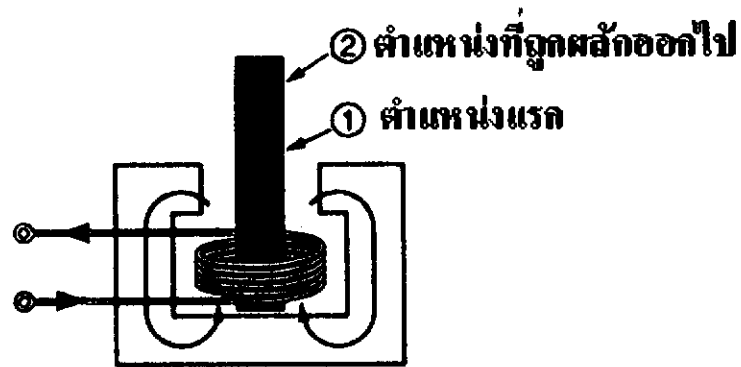


รูปที่ 2.23 วงจรไบอัสคงที่

2.8 ทฤษฎีเกี่ยวกับการทำงานของโซลินอยด์ (Solenoid)

การทำงานของโซลินอยด์ เป็นไปตามกฎของ อันส กริสเตียน เฮอร์ สเตด ซึ่งกล่าวว่า เมื่อมีกระแสไฟฟ้าไหลในขดลวดตัวนำใดๆ ก็ตาม จะเกิดสนามแม่เหล็กขึ้นรอบๆ ตัวนำนั้น และทิศทางเส้นแรงแม่เหล็กเป็นไปตามกฎมือขวา คือ ถ้าเอามือขวากำรอบเส้น โดยนิ้วหัวแม่มือแทนทิศทางกระแสไหล นิ้วที่เหลือจะแสดงทิศทางเส้นแรงแม่เหล็กจากขั้วได้ไปขั้วเหนือ เมื่อนำเส้นลวดมาขดเป็นวงหลายๆ วง เป็นขดลวด สนามแม่เหล็กที่เกิดจากขดลวดแต่ละขดจะอยู่ในทิศทางเสริมกัน

เพื่อที่จะไม่ให้สนามแม่เหล็กที่เกิดขึ้นกระจัดกระจาย จึงได้มีการใส่แกนเหล็กอ่อนรูปตัวซี (C) เข้ามารอบขดลวด เพื่อให้สนามแม่เหล็กมากขึ้น และถ้าเอาแกนกระทุ้ง (plunger) มาใส่เข้าไปตรงกลางขดลวดในตำแหน่งที่ 1 แกนกระทุ้งจะถูกผลัก ให้ออกไปที่ตำแหน่งที่ 2 ซึ่งระยะทางไกลมากเท่าไร แรงผลักก็จะมากขึ้นเท่านั้น



รูปที่ 2.24 แสดงการเคลื่อนที่ของแกนกระตุ้ง

ข้อแตกต่างระหว่างโซลินอยด์ไฟตรง และโซลินอยด์ไฟสลับ คือ ในโซลินอยด์ไฟตรง กระแสที่ไหลในขดลวด จะค่อนข้างคงที่ไม่เปลี่ยนแปลง ไม่ว่าแกนกระตุ้งจะอยู่ในตำแหน่งใดก็ตาม แต่โซลินอยด์ไฟสลับ กระแสในขณะที่แกนกระตุ้ง อยู่ในขดลวดจะมีค่าสูง และเมื่อแกนกระตุ้งถูกผลักออกไป กระแสจะลดต่ำลง ลักษณะแบบนี้ทำให้ต้องระวัง ไม่ให้เกิดการกระตุ้งในโซลินอยด์ไฟสลับ เนื่องจากจะทำให้เกิดกระแสมากขึ้น ไหลค้างอยู่ ทำให้ขดลวดร้อนขึ้น และ อาจจะมีไหม้เสียหายได้

2.9 รหัสแอสกี (ASCII Code)

รหัสแอสกี ย่อมาจาก American Standard Code for Information Interchange เป็นโค้ดแบบ 7 บิต ดังแสดงรายละเอียดตามตารางที่ 2.2

ตารางที่ 2.2 รายละเอียดตาราง รหัสแอสกี

| Hex. | Binary | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------|--------|-----|-----|----|---|---|---|---|---|
| 0 | 0000 | NUL | | SP | 0 | @ | P | . | p |
| 1 | 0001 | | DC1 | ! | 1 | A | Q | a | q |
| 2 | 0010 | | DC2 | " | 2 | B | R | b | r |
| 3 | 0011 | | DC3 | # | 3 | C | S | c | s |
| 4 | 0100 | | DC4 | \$ | 4 | D | T | d | t |
| 5 | 0101 | | | % | 5 | E | U | e | u |
| 6 | 0110 | | | & | 6 | F | V | f | v |
| 7 | 0111 | BEL | | ^ | 7 | G | W | g | w |
| 8 | 1000 | BS | | (| 8 | H | X | h | x |
| 9 | 1001 | HT | |) | 9 | I | Y | i | y |
| A | 1010 | LF | | * | : | J | Z | j | z |
| B | 1011 | VT | ESC | + | ; | K | [| k | (|
| C | 1100 | FF | | , | < | L | \ | l | |
| D | 1101 | CR | | - | = | M |] | m |) |
| E | 1110 | SO | | . | > | N | ^ | n | ~ |
| F | 1111 | SI | | / | ? | O | _ | o | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

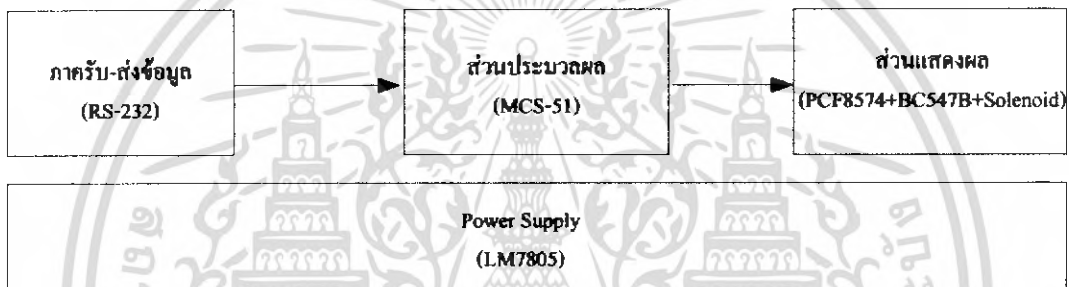
บทที่ 3 การออกแบบและการสร้าง

โครงการนี้ในการออกแบบแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ

1. ส่วนฮาร์ดแวร์
2. ส่วนซอฟต์แวร์

โดยในส่วนของฮาร์ดแวร์ ถูกแบ่งออกเป็น 4 ส่วน

1. ภาคจ่ายไฟ
2. ภาครับ-ส่งข้อมูล
3. ภาคประมวลผล
4. ภาคแสดงผล

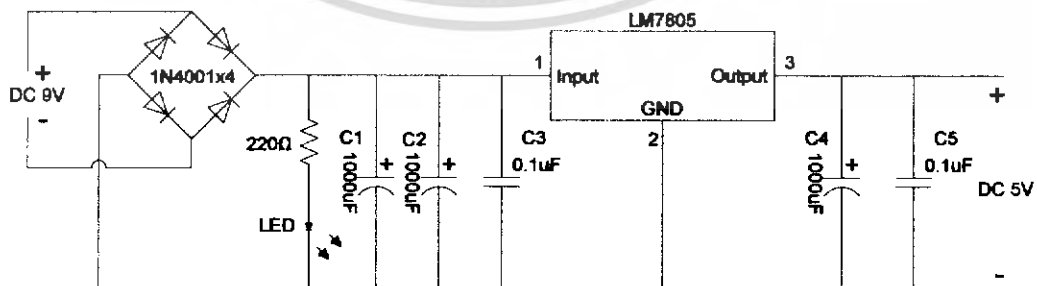


รูปที่ 3.1 บล็อกไดอะแกรมของโครงการ

3.1 ส่วนฮาร์ดแวร์

3.1.1 ภาคจ่ายไฟ

เนื่องจากอุปกรณ์ต่างๆในโครงการนี้ใช้ไฟกระแสตรงขนาด 5 โวลต์ในการทำงาน จึงต้องมีการออกแบบวงจรเพื่อให้สามารถใช้งานได้ถูกต้องดังรูปที่ 3.2



รูปที่ 3.2 วงจรภาคจ่ายไฟ

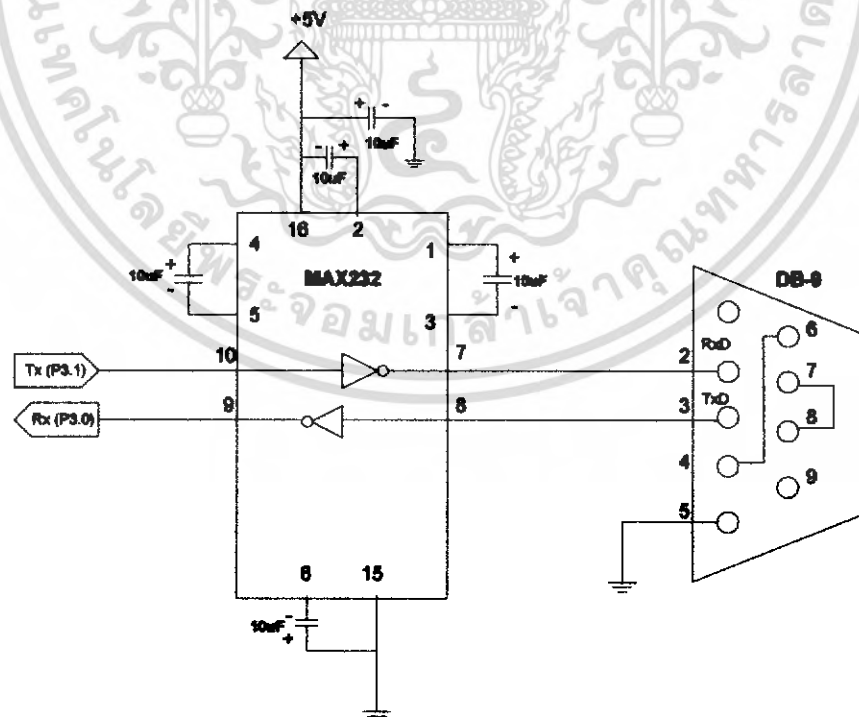
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากวงจรนี้สามารถรับอินพุตได้ทั้งสัญญาณไฟฟ้ากระแสตรงและกระแสสลับเพื่อให้เกิดความสะดวกในการใช้งาน เราจึงใช้วงจรบริดจ์ โดยถ้าหากเราป้อนไฟเข้ามาเป็นไฟฟ้ากระแสสลับวงจรบริดจ์นี้จะทำหน้าที่เป็นวงจรเรกติไฟเออร์ ทำให้ไฟฟ้ากระแสสลับกลายเป็นไฟฟ้ากระแสตรง หรือหากเราป้อนไฟเข้ามาเป็นไฟฟ้ากระแสตรง ก็จะทำให้สามารถต่อสลับขั้วกันได้โดยมีไดโอดทำหน้าที่แปลงขั้วบวก-ลบให้ถูกต้องดังรูปที่ 3.2 โดยในโครงงานนี้ใช้ไฟที่ได้มาจาก อแดปเตอร์กระแสตรง (DC adapter) ที่มีค่า 9 โวลต์

โดยในโครงงานนี้เราเลือกใช้ไอซี LM7805 ซึ่งเป็น ไอซีโวลต์เทจเรกกูเลเตอร์ ซึ่งทำหน้าที่รักษาระดับของไฟให้มีค่าคงที่ที่ 5 โวลต์ และมีตัวเก็บประจุ C1, C2 เพื่อกรองแรงดันหรือฟิลเตอร์ เนื่องจากการเปลี่ยนแรงดันไฟฟ้ากระแสสลับให้เป็นแรงดันไฟฟ้ากระแสตรง จะยังมีการกระเพื่อมของแรงดันไฟตรงที่เราเรียกว่าริปลิง ดังนั้นเราจึงใช้ตัวเก็บประจุเพื่อลดค่าแรงดันริปลิงลงไป โดยการเก็บค่าประจุไว้เมื่อช่วงแรงดันสูง และจะจ่ายประจุให้กับโหลดเมื่อมีการกระเพื่อมทางด้านต่ำ ดังนั้นโหลดจะได้แรงดันที่ราบเรียบขึ้น C4 ทำหน้าที่กรองแรงดันที่ออกมาจากเอาต์พุตของไอซี C3, C5 ทำหน้าที่กรองสัญญาณความถี่สูงทั้งไป ซึ่งอาจเกิดจากอุปกรณภายในตัวไอซีเอง

3.1.2 ภาครับ-ส่งข้อมูล

ในโครงงานนี้ใช้ไอซี MAX232 เพื่อรองรับการส่งผ่านข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 โดยใช้การเชื่อมต่อแบบ 9 ขา (DB-9) และแปลงไปกลับระหว่างระดับแรงดันของ TTL หรือ CMOS โดยใช้ไฟเลี้ยง 5 โวลต์ ดังรูปที่ 3.3

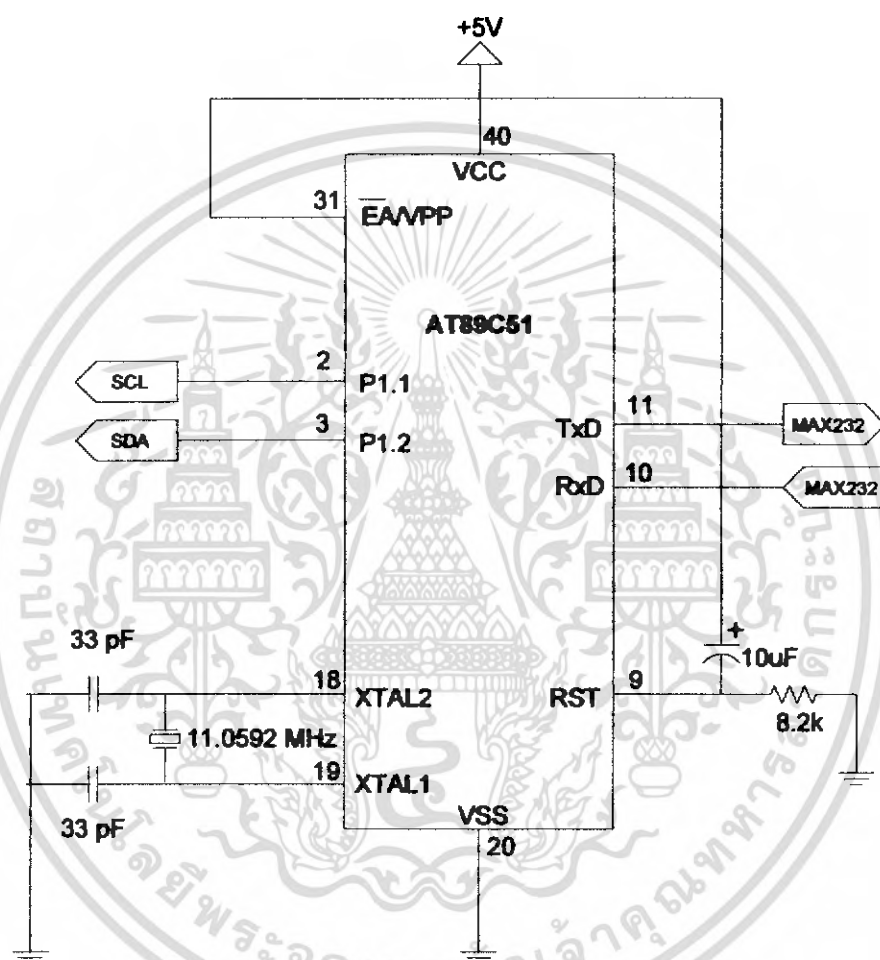


รูปที่ 3.3 วงจรภาครับ-ส่ง ผ่าน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ภาคประมวลผล

โครงงานนี้ใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงานของระบบโดยใช้ระบบการทำงานแบบมาสเตอร์-สเลฟ โดยมี ไอซี AT89C51 เป็นมาสเตอร์ และใช้ไอซีขยายพอร์ตเบอร์ PCF8574AP เป็นสเลฟ โดยไมโครคอนโทรลเลอร์จะควบคุมการส่งแอดเดรส และส่งข้อมูล เพื่อนำไปแสดงผลในส่วนแสดงผล โดยในโครงงานนี้กำหนดให้ใช้พอร์ต 1.1 สำหรับส่งสัญญาณนาฬิกาเพื่อใช้ในการควบคุมการรับ-ส่งข้อมูลและ พอร์ต 1.2 สำหรับรับ-ส่งข้อมูลทั้งแอดเดรสและข้อมูลที่จะนำไปแสดงผล โดยมีการเชื่อมในส่วนของมาสเตอร์ดังรูปที่ 3.4



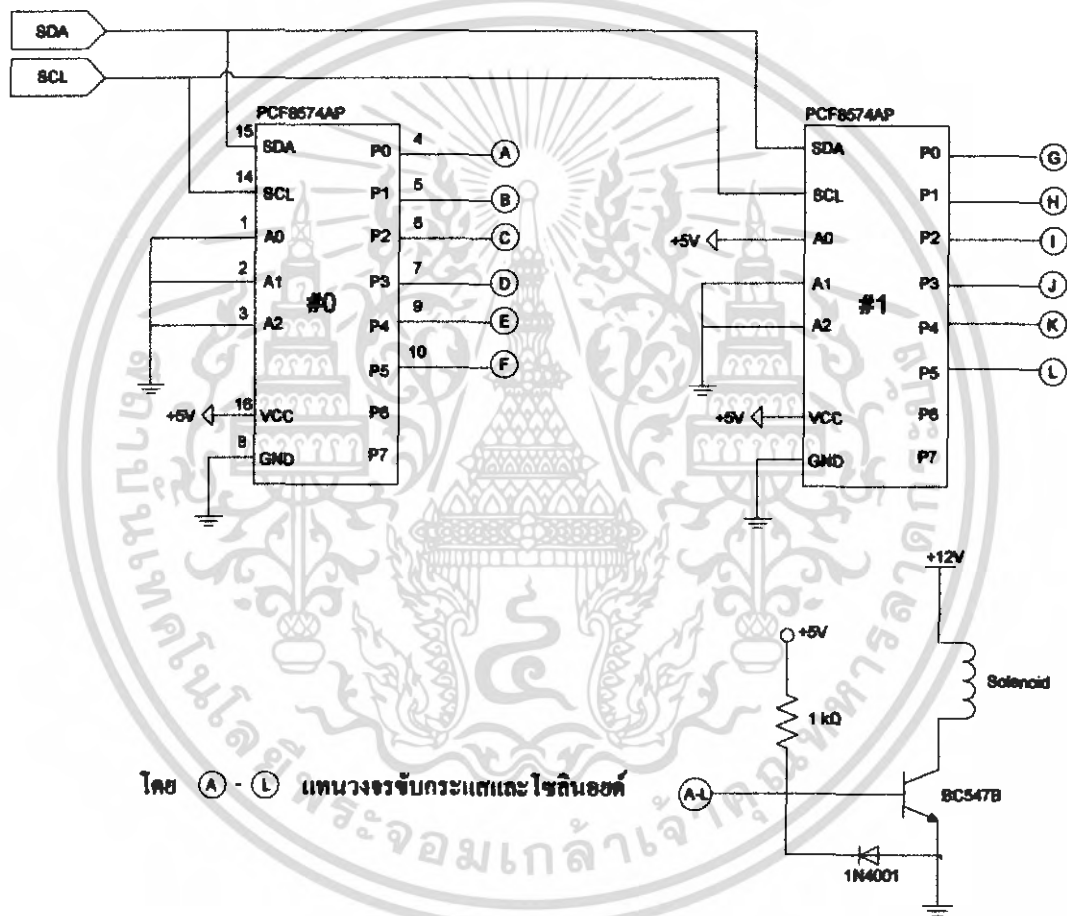
รูปที่ 3.4 วงจรภาคประมวลผล

ซึ่งในการต่อจะใช้การรีเซ็ตค่าของไมโครคอนโทรลเลอร์แบบ รีเซ็ตอัตโนมัติ เพื่อให้เกิดการรีเซ็ตค่าทุกครั้งเมื่อทำการเปิดใช้งาน โดยการป้อนสัญญาณขนาด 5 โวลท์ไปที่ขา RST โดยการต่อความต้านทาน 8.2 กิโลโห์ม และตัวเก็บประจุ 10 ไมโครฟารัด ดังรูปที่ 3.4 ซึ่งเมื่อเริ่มจ่ายกระแสไฟวงจรจะทำให้เกิดสัญญาณขนาด 5 โวลท์ แล้วลดลงเหลือ 0 โวลท์ในเวลาสั้นๆ และใช้คริสตัลขนาด 11.059 เมกกะเฮิรตซ์ ต่อคร่อมระหว่างขา XTAL1, XTAL2 เพื่อใช้ในการควบคุมสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 ภาคแสดงผล

ในการแสดงผล จะใช้เอาต์พุตที่ได้จาก PCF8574AP ไปขับโซลินอยด์ให้ทำงานตามที่ต้องการ โดยมีวงจรขับกระแสเป็นตัวขับกระแสให้โซลินอยด์ การทำงานของ PCF8574AP จะทำการรับข้อมูลทางพอร์ต SDA และถูกควบคุมการทำงานโดยสัญญาณ SCL โดยการเชื่อมต่อทำได้โดยต่อสายจากพอร์ต 1.1 (ขา2) และพอร์ต 1.2 (ขา3) ของไมโครคอนโทรลเลอร์เข้ากับ ขา 14 และ ขา 15 ของ PCF8574AP และทำการกำหนดแอดเดรสของไอซีแต่ละตัวโดยการป้อนไฟที่กระแสตรงขนาด 5 โวลต์ เมื่อต้องการให้เป็นลอจิก "1" หรือคอลลกกราวด์เมื่อต้องการให้เป็นลอจิก "0" เข้าที่ขา 1, 2, 3 ตามลำดับ โดยในการทดลองนี้ กำหนดแอดเดรส 0 และ แอดเดรส 1 ให้ไอซีแต่ละตัว ดังรูปที่ 3.5

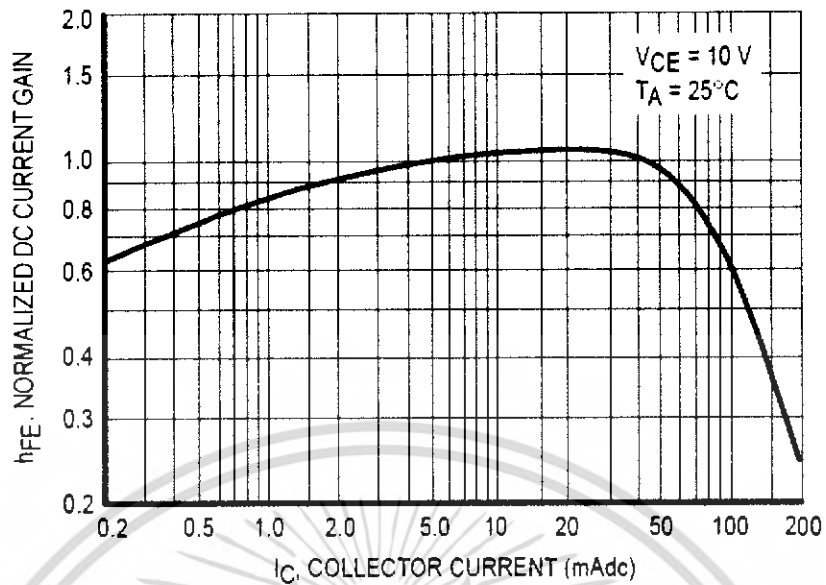


โดย (A) - (L) แทนวงจรขับกระแสและโซลินอยด์

รูปที่ 3.5 วงจรภาคแสดงผล

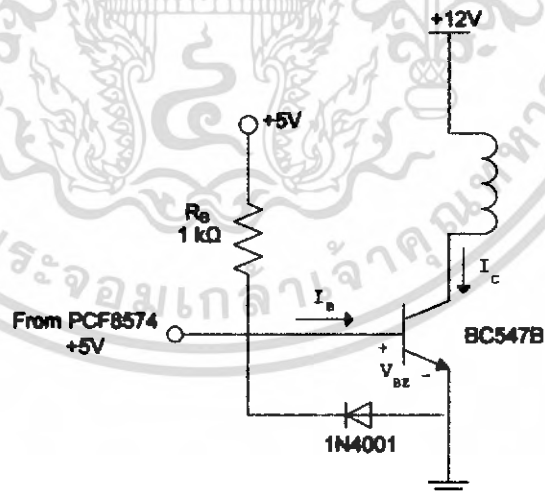
จากรูปที่ 3.5 จะเห็นว่าเราไม่สามารถต่อ โซลินอยด์เข้ากับเอาต์พุตพอร์ตโดยตรงได้ เนื่องจากตัว PCF8574AP นั้นไม่สามารถขับกระแสมากพอที่จะขับโซลินอยด์ได้ เราจึงต้องมีตัวต้านทาน 1 กิโลโห์ม และทรานซิสเตอร์ BC547B มาขับกระแสให้โซลินอยด์ทำงาน ซึ่งเราใช้โซลินอยด์ที่ใช้ไฟ 12 โวลต์ โดยในส่วนของวงจรขับกระแส ได้ทำการคำนวณ ซึ่งทรานซิสเตอร์ BC547B มีรูปกราฟของ อัตราขยายไฟที่กระแสตรง (Normalized DC current gain) ดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 อีกรายขยายไฟฟ้ากระแสตรง

จากการทดลอง ทำการวัดกระแส I_C ได้ 160 มิลลิแอมป์ ซึ่งจากกราฟ อ่านค่าอีกรายขยายไฟตรง (β) ได้ประมาณ 0.32 ซึ่งเป็นค่าที่ผ่านเกณฑ์มาตรฐาน ในการคำนวณ เราจะใช้ค่าอีกรายขยายก่อนทำการนอร์มอลไลซ์ ซึ่งมีค่าประมาณ 32



รูปที่ 3.7 วงจรขับกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7 สามารถคำนวณได้

$$I_B = \frac{V_{CC} - V_{BE}}{R_B}$$

เมื่อทำการแทนค่า โดย V_{CC} คือ 5V, V_{BE} คือ 0.7V และ R_B คือ 1 k Ω จะได้

$$I_B = \frac{5 - 0.7}{1000}$$

$$I_B = 4.3mA$$

และ

$$I_C = \beta I_B$$

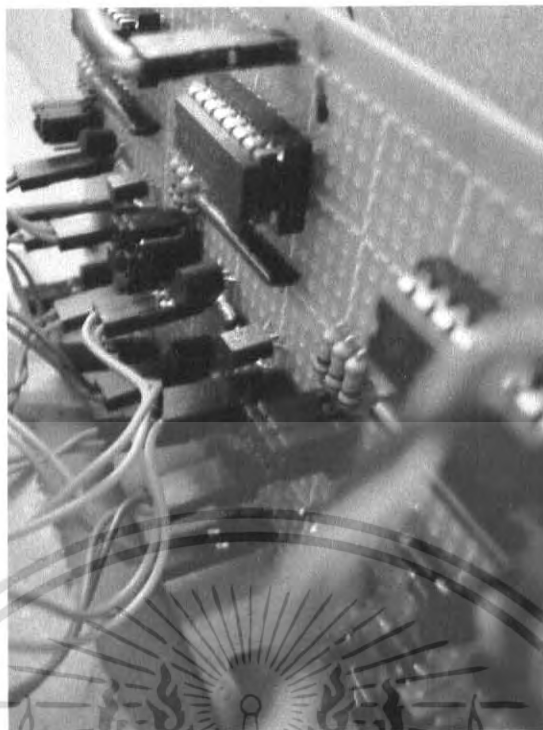
$$I_C = (32)(4.3mA)$$

$$I_C = 137.6mA$$

และที่เราเลือกใช้แค่ 6 พอร์ตนั้นเนื่องจาก ลักษณะของอักษรเบรลล์นั้น ใช้จำนวนจุด 6 จุดในการแสดงผลต่อ 1 ตัวอักษร โดยออกแบบให้บิต 1 บิต ควบคุมจุด 1 จุด เช่น บิตที่ 0 แสดงผลในตำแหน่งที่ 0 ของ อักษรเบรลล์ดังรูป

Bit5 – Solenoid5 ● ● Solenoid4 – Bit4
 Bit3 – Solenoid3 ● ● Solenoid2 – Bit2
 Bit1 – Solenoid1 ● ● Solenoid0 – Bit0

รูปที่ 3.8 บิตที่ใช้ในการควบคุม โซลินอยด์

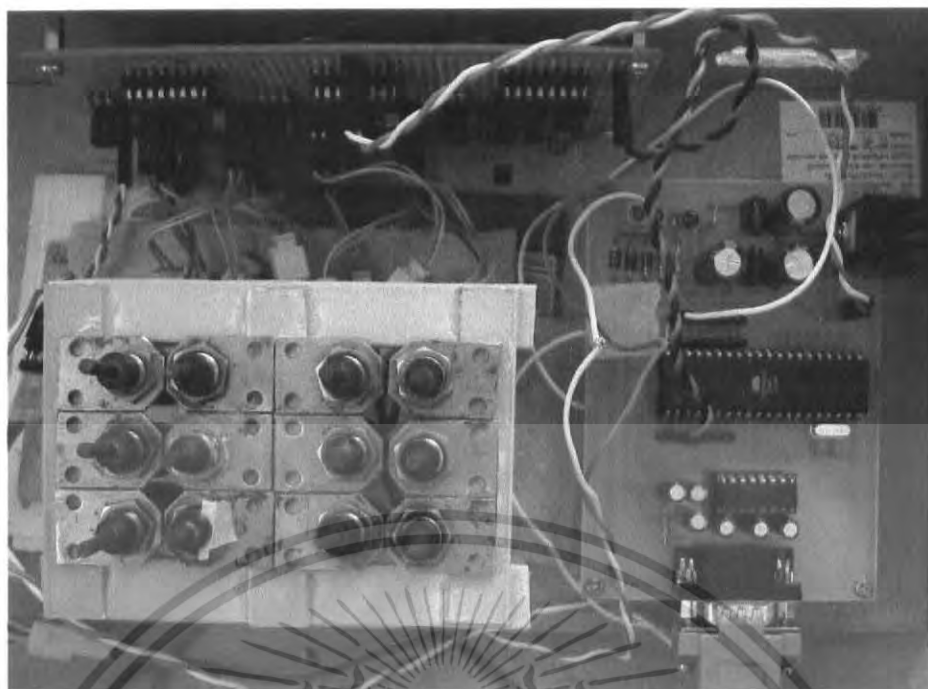


รูปที่ 3.9 ภาพของวงจรส่วนขับเคลื่อน



รูปที่ 3.10 ภาพของวงจรส่วนของโซลินอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ภาพของวงจรที่ต่อเรียบร้อยแล้ว

3.2 ส่วนซอฟต์แวร์

เป็นการเขียนโปรแกรมในไมโครคอนโทรลเลอร์เพื่อใช้ในการรับข้อมูลในรูปแบบของ รหัสแอสกี จากคอมพิวเตอร์ผ่าน RS-232 แล้วนำไปเปรียบเทียบค่า จากนั้นจะทำการแปลงค่าจากรหัสแอสกีนั้นให้กลายเป็นรูปแบบที่ต้องการตามลักษณะของอักษรเบรลล์และทำการส่งค่าออกทางพอร์ตโดยใช้ตัวอักษรที่กำหนดดังตารางที่ 3.1, ตารางที่ 3.2, ตารางที่ 3.3 และ ตารางที่ 3.4

ตารางที่ 3.1 แสดงการเปรียบเทียบระหว่าง รหัสแอสกี กับ รหัสที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์ และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ต สำหรับตัวอักษรภาษาอังกฤษ

| ตัวอักษร ภาษาอังกฤษ | ตัวอักษรเบรลล์ | รหัสแอสกี | รหัสในการ แสดงผลอักษร เบรลล์ | ตัวอักษรที่ส่ง แทน รหัสแสดงผล |
|------------------------|----------------|-----------|------------------------------------|-------------------------------------|
| a | ●○ ○○ ○○ | 61H | 20H | (Grave Accent) |
| b | ●○ ●○ ○○ | 62H | 28H | h |
| c | ●● ○○ ○○ | 63H | 30H | 0 (Zero) |
| d | ●● ●○ ○○ | 64H | 34H | 4 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตัวอักษร ภาษาอังกฤษ | ตัวอักษรเบรลล์ | รหัสแอสกี | รหัสในการ แสดงผลอักษร เบรลล์ | ตัวอักษรที่ส่ง แทน รหัสแสดงผล |
|------------------------|----------------|-----------|------------------------------------|-------------------------------------|
| e | ⠠ | 65H | 24H | d |
| f | ⠠ | 66H | 38H | 8 |
| g | ⠠ | 67H | 3CH | < |
| h | ⠠ | 68H | 2CH | l (small L) |
| i | ⠠ | 69H | 18H | X |
| j | ⠠ | 6AH | 1CH | \ |
| k | ⠠ | 6BH | 22H | b |
| l | ⠠ | 6CH | 2AH | j |
| m | ⠠ | 6DH | 32H | 2 |
| n | ⠠ | 6EH | 36H | 6 |
| o | ⠠ | 6FH | 26H | f |
| p | ⠠ | 70H | 3AH | z |
| q | ⠠ | 71H | 3EH | > |
| r | ⠠ | 72H | 2EH | n |
| s | ⠠ | 73H | 1AH | Z |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตัวอักษร ภาษาอังกฤษ | ตัวอักษรเบรลล์ | รหัสแอสกี | รหัสในการ แสดงผลอักษร เบรลล์ | ตัวอักษรที่ส่ง แทน รหัสแสดงผล |
|------------------------|----------------|-----------|------------------------------------|-------------------------------------|
| t | ⠠ | 74H | 1EH | ^ |
| u | ⠡ | 75H | 23H | c |
| v | ⠢ | 76H | 2BH | k |
| w | ⠣ | 77H | 1DH |] |
| x | ⠤ | 78H | 33H | 3 |
| y | ⠥ | 79H | 37H | 7 |
| z | ⠦ | 7AH | 27H | g |

ตารางที่ 3.2 แสดงการเปรียบเทียบระหว่าง ASCII Code กับ Code ที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ต สำหรับตัวอักษรภาษาอังกฤษพิมพ์ใหญ่

ซึ่งสำหรับตัวอักษรเบรลล์จะมีลักษณะเหมือนตัวอักษรพิมพ์เล็กที่มีตัว “.” หรือ



| ตัวอักษรภาษาอังกฤษ | รหัสแอสกี | รหัสในการแสดงผล อักษรเบรลล์ | ตัวอักษรที่ส่งแทน รหัสแสดงผล |
|--------------------|-----------|--------------------------------|---------------------------------|
| A | 41H | 08H 20H | H ` (Grave Accent) |
| B | 42H | 08H 28H | H h |
| C | 43H | 08H 30H | H 0 (Zero) |
| D | 44H | 08H 34H | H 4 |
| E | 45H | 08H 24H | H d |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้







| ตัวอักษรภาษาอังกฤษ | รหัสแอสกี | รหัสในการแสดงผล อักษรเบรลล์ | ตัวอักษรที่ส่งแทน รหัสแสดงผล |
|--------------------|-----------|--------------------------------|---------------------------------|
| F | 46H | 08H 38H | H 8 |
| G | 47H | 08H 3CH | H < |
| H | 48H | 08H 2CH | H 1 (small L) |
| I | 49H | 08H 18H | H X |
| J | 4AH | 08H 1CH | H \ |
| K | 4BH | 08H 22H | H b |
| L | 4CH | 08H 2AH | H j |
| M | 4DH | 08H 32H | H 2 |
| N | 4EH | 08H 36H | H 6 |
| O | 4FH | 08H 26H | H f |
| P | 50H | 08H 3AH | H z |
| Q | 51H | 08H 3EH | H > |
| R | 52H | 08H 2EH | H n |
| S | 53H | 08H 1AH | H Z |
| T | 54H | 08H 1EH | H ^ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตัวอักษรภาษาอังกฤษ | รหัสแอสกี | รหัสในการแสดงผล อักษรเบรลล์ | ตัวอักษรที่ส่งแทน รหัสแสดงผล |
|--------------------|-----------|--------------------------------|---------------------------------|
| U | 55H | 08H 23H | H c |
| V | 56H | 08H 2BH | H k |
| W | 57H | 08H 1DH | H J |
| X | 58H | 08H 33H | H 3 |
| Y | 59H | 08H 37H | H 7 |
| Z | 5AH | 08H 27H | H g |

ตารางที่ 3.3 แสดงการเปรียบเทียบระหว่าง รหัสแอสกี กับ รหัสที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์ และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ต สำหรับตัวเลข

โดยจะมีตัว “#” หรือ  ในอักษรเบรลล์ นำหน้า

| ตัวเลข | ตัวอักษรเบรลล์ | รหัสแอสกี | รหัสในการ แสดงผลอักษร เบรลล์ | ตัวอักษรที่ส่งแทน รหัสแสดงผล |
|--------|---|-----------|------------------------------------|---------------------------------|
| 0 |  | 30H | 07H | G |
| 1 |  | 31H | 08H | H |
| 2 |  | 32H | 0AH | J |
| 3 |  | 33H | 0CH | L |
| 4 |  | 34H | 0DH | M |
| 5 |  | 35H | 09H | I |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตัวเลข | ตัวอักษรเบรลล์ | รหัสแอสกี | รหัสในการ แสดงผลอักษร เบรลล์ | ตัวอักษรที่ส่งแทน รหัสแสดงผล |
|--------|----------------|-----------|------------------------------------|---------------------------------|
| 6 | ⠠ | 36H | 0EH | N |
| 7 | ⠡ | 37H | 0FH | O |
| 8 | ⠢ | 38H | 0BH | K |
| 9 | ⠣ | 39H | 06H | F |

ตารางที่ 3.4 แสดงการเปรียบเทียบระหว่าง รหัสแอสกี กับ รหัสที่กำหนดขึ้นตามลักษณะของอักษรเบรลล์ และตัวอักษรที่ใช้ส่งในการส่งออกทางพอร์ต สำหรับตัวอักษร

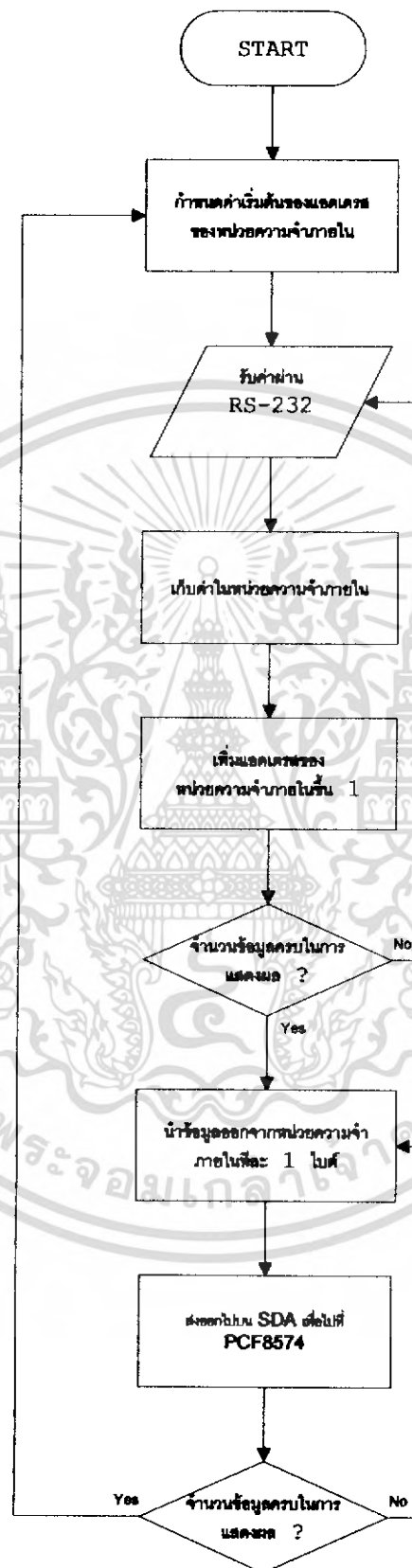
| อักขระ | ตัวอักษรเบรลล์ | รหัสแอสกี | รหัสในการ แสดงผลอักษร เบรลล์ | ตัวอักษรที่ส่งแทน รหัสแสดงผล |
|--------|----------------|-----------|------------------------------------|---------------------------------|
| # | ⠠ | 23H | 17H | W |
| : | ⠡ | 3AH | 25H | e |
| ; | ⠢ | 3BH | 05H | E |
| < | ⠣ | 3CH | 29H | I |
| > | ⠣ | 3EH | 16H | V |
| ? | ⠣ | 3FH | 35H | S |
| @ | ⠣ | 40H | 10H | P |
| [| ⠣ | 5BH | 23H | c |
|] | ⠣ | 5DH | 3DH | = |
| = | ⠣ | 3DH | 3FH | ? |
| , | ⠣ | 2CH | 08H | H |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| อักขระ | ตัวอักษรเบรลล์ | รหัสแอสกี | รหัสในการแสดงผลอักษรเบรลล์ | ตัวอักษรที่ส่งแทนรหัสแสดงผล |
|--------|----------------|-----------|----------------------------|-----------------------------|
| ! | ⠠ | 21H | 1BH | [|
| “ | ⠡ | 22H | 04H | D |
| \$ | ⠵ | 24H | 39H | 9 |
| % | ⠼ | 25H | 31H | 1 (one) |
| & | ⠠ | 26H | 3BH | ; |
| (| ⠠ | 28H | 2FH | o |
|) | ⠠ | 29H | 1FH | - |
| . | ⠠ | 27H | 02H | B |
| * | ⠠ | 2AH | 21H | a |
| + | ⠠ | 2BH | 13H | S |
| - | ⠠ | 2DH | 03H | C |
| / | ⠠ | 2FH | 12H | R |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

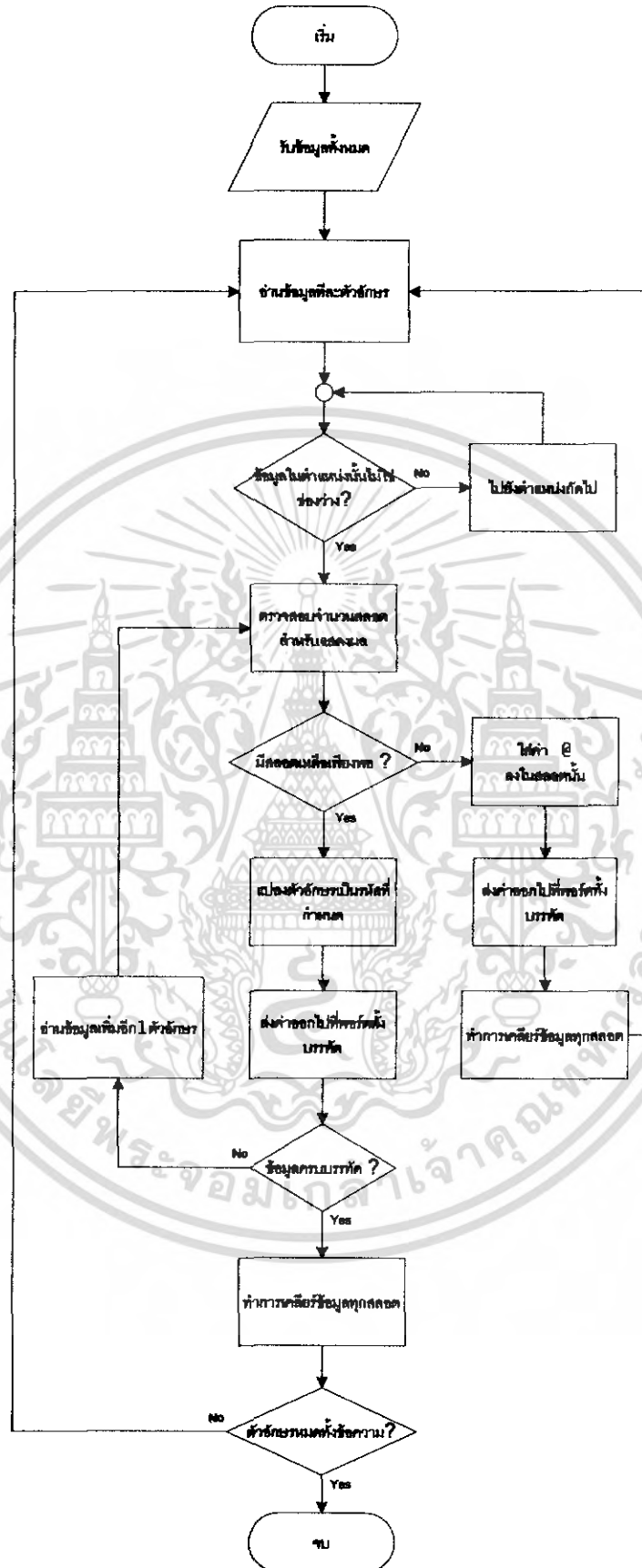
3.2.1 ฟังก์ชันการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์



รูปที่ 3.12 แสดงฟังก์ชันการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ผังการทำงานของโปรแกรมที่ใช้ในการแปลงค่า



รูปที่ 3.13 แสดงผังการทำงานของโปรแกรมที่ใช้แปลงค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ขั้นตอนการทดลอง

1. เชื่อมต่อบอร์ดทดลองเข้ากับคอมพิวเตอร์ด้วยสาย DB-9 จากนั้นทำการเปิดโปรแกรม แปลงอักษรเบรลล์ขึ้นมา



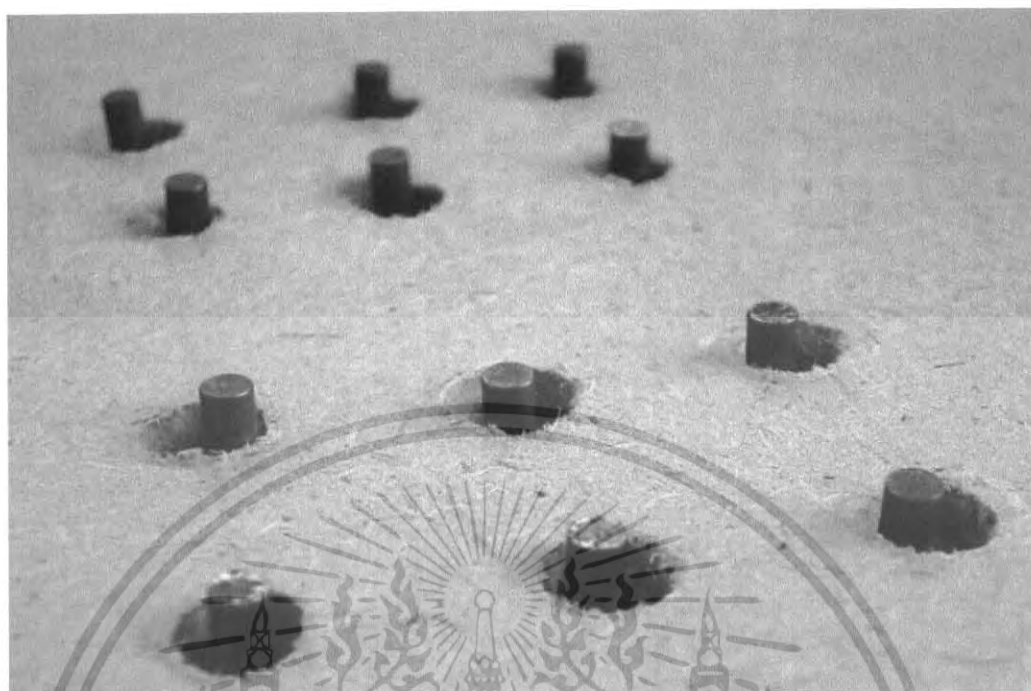
รูปที่ 4.1 เมื่อทำการเปิดโปรแกรม แปลงอักษรเบรลล์



รูปที่ 4.2 ก่อนทำการเปิดเครื่องจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการเปิดเครื่องจ่ายไฟที่อุปกรณ์ทดลอง จะเห็นว่าโซลินอยด์ทั้งหมดจะติดตัวขึ้น ดังรูปที่ 4.3



รูปที่ 4.3 หลังทำการเปิดเครื่องจ่ายไฟ

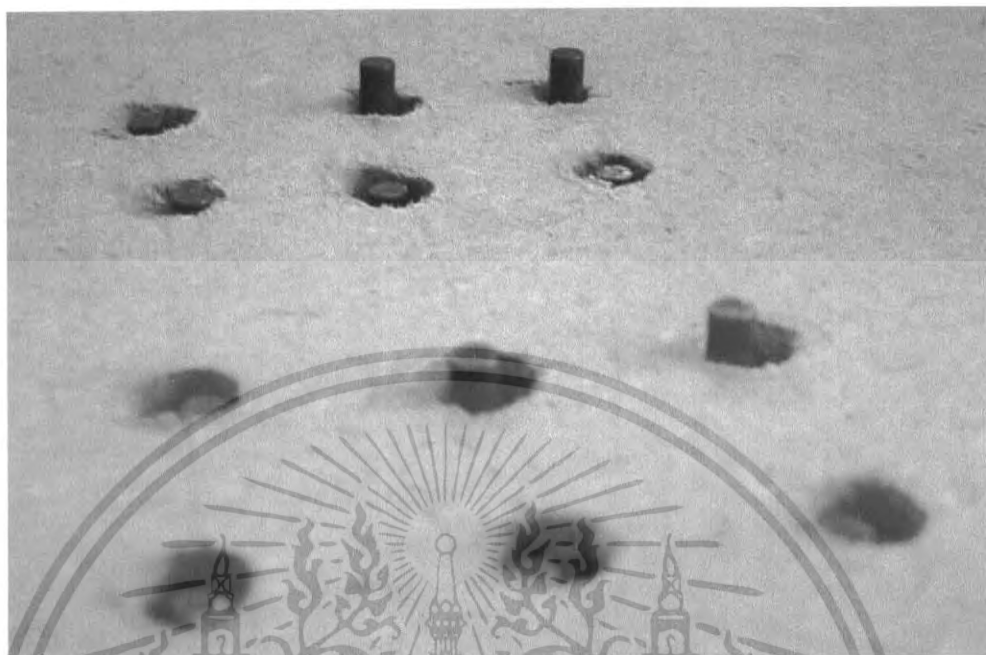
3. ทำการป้อนค่าโดยการกด “RESET” และกดคีย์บอร์ดทีละ 2 ตัวอักษร เช่น “bb” จากนั้นกด “Start” ผลบนหน้าจอคอมพิวเตอร์จะได้ดังรูปที่ 4.4



รูปที่ 4.4 ผลที่ได้บนโปรแกรม เมื่อป้อนอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สังเกตที่โซลินอยด์ที่ต่ออยู่กับ PCF8574AP จะเห็นว่าบางจุดจะตกกลับมาอยู่ในระดับราบ แสดงลักษณะอักษรเบรลล์ของตัวอักษรที่พิมพ์ไป ดังรูปที่ 4.5


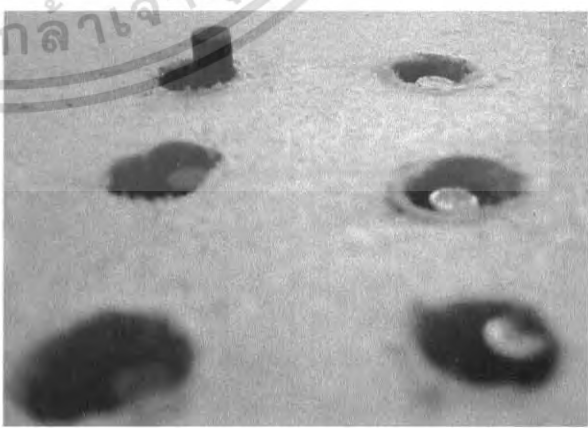


รูปที่ 4.5 การแสดงผลของโซลินอยด์เมื่อใส่อินพุต "bb"


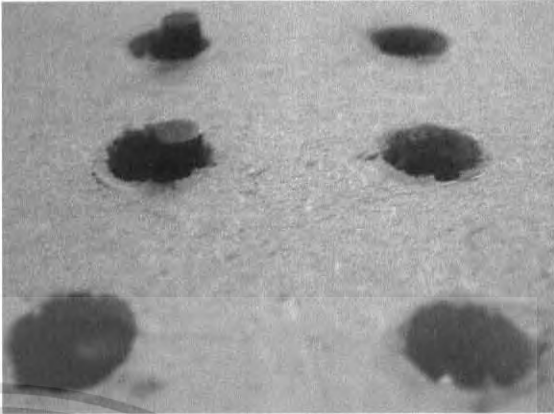

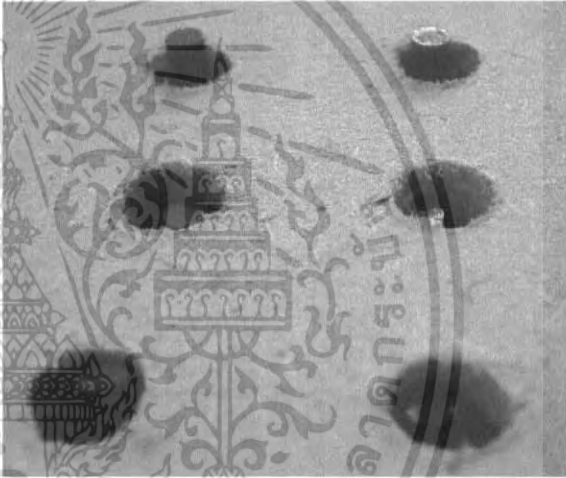


4.2 ผลการทดลองที่ได้

เมื่อพิจารณาที่ละตัวอักษร จะพบว่าการแสดงผลสามารถแสดงได้ถูกต้องตามลักษณะของอักษรเบรลล์ดังตัวอย่างในตารางที่ 4.1 และตารางที่ 4.2


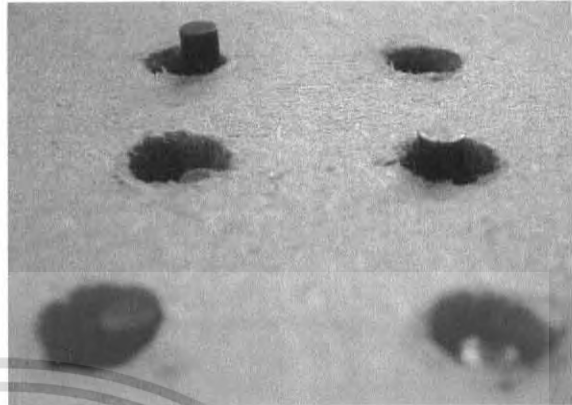
ตารางที่ 4.1 ตารางเปรียบเทียบผลที่ได้กับอักษรเบรลล์ (ตัวอักษร)

| ตัวอักษรภาษาอังกฤษ | อักษรเบรลล์ | ผลที่ได้ |
|--------------------|---|--|
| a |  |  |




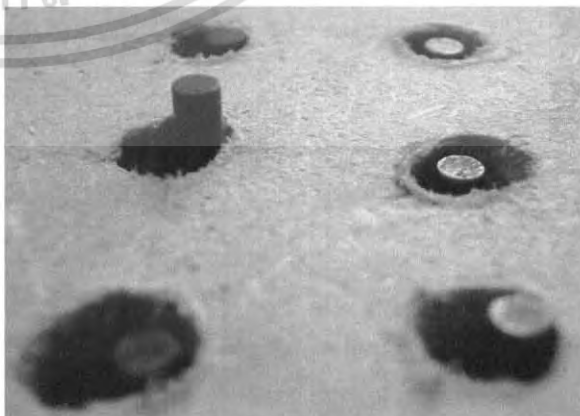
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตัวอักษรภาษาอังกฤษ | อักษรเบรลล์ | ผลที่ได้ |
|--------------------|---|--|
| b |  |  |
| c |  |  |
| d |  |  |


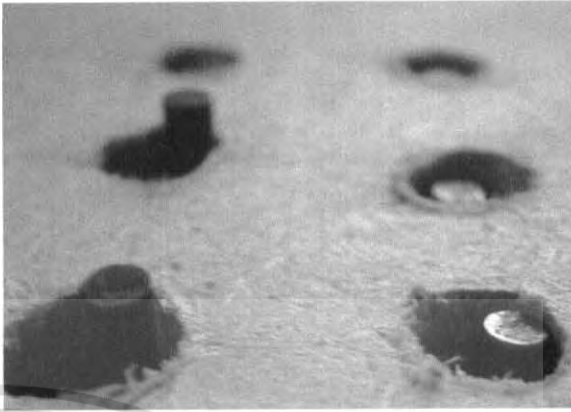


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตัวอักษรภาษาอังกฤษ | อักษรเบรลล์ | ผลที่ได้ |
|--------------------|---|--|
| e |  |  |

ตารางที่ 4.2 ตารางเปรียบเทียบผลที่ได้กับอักษรเบรลล์ (ตัวเลข)

| ตัวเลข | เลขเบรลล์ | ผลที่ได้ |
|--------|---|--|
| 0 |  |  |
| 1 |  |  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตัวเลข | เลขเบรลล์ | ผลที่ได้ |
|--------|--|---|
| 2 |  |  |
| 3 |  |  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และบทสรุป

จากการทดลองออกแบบวงจรเพื่อใช้ในการสร้างโครงงานเครื่องแสดงอักษรเบรลล์นั้น เมื่อพิจารณาที่หลักการสร้าง จะเห็นว่า เราสามารถสร้างอุปกรณ์ที่มีขายในท้องตลาดทั่วไปในราคาที่ค่อนข้างสูงได้ด้วยความรู้พื้นฐานที่มีเรียนในห้องเรียน ด้วยการนำองค์ความรู้ที่ได้เรียนมาเหล่านั้นมาประยุกต์ใช้ โดยเราสามารถแสดงให้เห็นว่า สามารถแสดงผลในลักษณะที่คล้ายกับอักษรเบรลล์ได้ โดยใช้โซลินอยด์แบบพวง-พุด ที่จะดันตัวสูงขึ้นเมื่อป้อนลอจิกสูง ซึ่งเสมือนว่าเป็นปุ่มนูนขึ้น ของจุดที่ใช้ในการแสดงผลอักษรเบรลล์ และจะอยู่ในระดับราบเมื่อป้อนลอจิกต่ำ ซึ่งเสมือนว่าเป็นจุดที่ราบเรียบในอักษรเบรลล์

โดยโครงงานนี้ยังมีข้อบกพร่องคือ จำนวนตัวอักษรที่แปลงได้ยังมีจำนวนน้อย เนื่องจากโซลินอยด์มีราคาค่อนข้างสูง และไม่สามารถนำโซลินอยด์ที่ทดลองทำขึ้นเองมาใช้ได้เนื่องจากใช้กระแสไฟมาก ความร้อนสูง ทั้งขนาดที่ได้ก็ยังมีขนาดใหญ่แต่กลับได้เส้นแรงแม่เหล็กน้อยจึงได้แรงแม่เหล็กที่ต่ำเมื่อเทียบกับโซลินอยด์ที่นำมาใช้ในโครงงานนี้ และขนาดของอักษรเบรลล์ที่ได้ไม่ได้มาตรฐานเนื่องจากโซลินอยด์ที่หาได้มีขนาดค่อนข้างใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. สมยศ จุณณะปิยะ. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์. กรุงเทพฯ: คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546
2. ชีรวัดน์ ประกอบผล. ภาษาแอสเซมบลีสำหรับ MCS-51. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2548
3. มงคล ทองสงคราม. อิเล็กทรอนิกส์เบื้องต้น. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วิเจ พรีนติ้ง
4. สมพร บุญอริน. “การใช้งาน ไอซีเรกูเลเตอร์ 78xx”
(<http://www.chontech.ac.th/~electric/html/regulator.htm>)
5. อติศักดิ์ ชิมะวงศ์. “สร้างบอร์ดแหล่งจ่ายไฟ 5 โวลท์”
(<http://www.adisak51.com/>)
6. ศุภชัย สมพานิช. คู่มือการเขียนโปรแกรมและใช้งาน Visual C#.NET ฉบับสมบูรณ์. กรุงเทพฯ: สำนักพิมพ์ Infopress Developer Book, 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ไลบรารีโปรแกรมแบบจัดการหน่วย Visual C#.Net 2005

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;

namespace คณิตยกตัว
{
    public partial class Form1 : Form
    {
        string tmp;
        string btmp;
        int i=0;
        SerialPort sp = new SerialPort();
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            tmp = maskedTextBox1.Text;
            button2_Click(sender,e);
            button1.Text = "Next";
        }

        private void button2_Click(object sender, EventArgs e)
        {
            char chartmp;
            int count;
            try
            {
                sp.Open();
                sp.ReadTimeout = 500;
            }
            catch (System.Exception ex)
            {
            }

            label1.Text = "";
            label2.Text = "";
mark1:
            if (i == tmp.Length)
            {
                button1.Enabled = false;
                goto end;
            }
            switch (tmp[i++])
            {
                case 'a': label1.Text = "`"; break;
                case 'b': label1.Text = "h"; break;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 'c': labell.Text = "0"; break;
case 'd': labell.Text = "4"; break;
case 'e': labell.Text = "d"; break;
case 'f': labell.Text = "8"; break;
case 'g': labell.Text = "<"; break;
case 'h': labell.Text = "1"; break;
case 'i': labell.Text = "X"; break;
case 'j': labell.Text = "\\"; break;
case 'k': labell.Text = "b"; break;
case 'l': labell.Text = "j"; break;
case 'm': labell.Text = "2"; break;
case 'n': labell.Text = "6"; break;
case 'o': labell.Text = "f"; break;
case 'p': labell.Text = "z"; break;
case 'q': labell.Text = ">"; break;
case 'r': labell.Text = "n"; break;
case 's': labell.Text = "Z"; break;
case 't': labell.Text = "^"; break;
case 'u': labell.Text = "c"; break;
case 'v': labell.Text = "k"; break;
case 'w': labell.Text = "}"; break;
case 'x': labell.Text = "3"; break;
case 'y': labell.Text = "7"; break;
case 'z': labell.Text = "g"; break;
case 'A': labell.Text = "H"; label2.Text = "`"; goto
end; break;
case 'B': labell.Text = "H"; label2.Text = "h"; goto
end; break;
case 'C': labell.Text = "H"; label2.Text = "0"; goto
end; break;
case 'D': labell.Text = "H"; label2.Text = "4"; goto
end; break;
case 'E': labell.Text = "H"; label2.Text = "d"; goto
end; break;
case 'F': labell.Text = "H"; label2.Text = "8"; goto
end; break;
case 'G': labell.Text = "H"; label2.Text = "<"; goto
end; break;
case 'H': labell.Text = "H"; label2.Text = "1"; goto
end; break;
case 'I': labell.Text = "H"; label2.Text = "X"; goto
end; break;
case 'J': labell.Text = "H"; label2.Text = "\\"; goto
end; break;
case 'K': labell.Text = "H"; label2.Text = "b"; goto
end; break;
case 'L': labell.Text = "H"; label2.Text = "j"; goto
end; break;
case 'M': labell.Text = "H"; label2.Text = "2"; goto
end; break;
case 'N': labell.Text = "H"; label2.Text = "6"; goto
end; break;
case 'O': labell.Text = "H"; label2.Text = "f"; goto
end; break;
case 'P': labell.Text = "H"; label2.Text = "z"; goto
end; break;
case 'Q': labell.Text = "H"; label2.Text = ">"; goto
end; break;
case 'R': labell.Text = "H"; label2.Text = "n"; goto
end; break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end; break;
case 'S': label1.Text = "H"; label2.Text = "Z"; goto
end; break;
case 'T': label1.Text = "H"; label2.Text = "^"; goto
end; break;
case 'U': label1.Text = "H"; label2.Text = "c"; goto
end; break;
case 'V': label1.Text = "H"; label2.Text = "k"; goto
end; break;
case 'W': label1.Text = "H"; label2.Text = "j"; goto
end; break;
case 'X': label1.Text = "H"; label2.Text = "3"; goto
end; break;
case 'Y': label1.Text = "H"; label2.Text = "7"; goto
end; break;
case 'Z': label1.Text = "H"; label2.Text = "g"; goto
end; break;
case '0': label1.Text = "G"; break;
case '1': label1.Text = "H"; break;
case '2': label1.Text = "J"; break;
case '3': label1.Text = "L"; break;
case '4': label1.Text = "M"; break;
case '5': label1.Text = "I"; break;
case '6': label1.Text = "N"; break;
case '7': label1.Text = "O"; break;
case '8': label1.Text = "K"; break;
case '9': label1.Text = "F"; break;
case '#': label1.Text = "W"; break;
case ':': label1.Text = "e"; break;
case ';': label1.Text = "E"; break;
case '<': label1.Text = "I"; break;
case '>': label1.Text = "V"; break;
case '?': label1.Text = "5"; break;
case '@': label1.Text = "P"; break;
case '[': label1.Text = "c"; break;
case ']': label1.Text = "="; break;
case '=': label1.Text = "?"; break;
case ',': label1.Text = "H"; break;
case '!': label1.Text = "{"; break;
case '"': label1.Text = "D"; break;
case '$': label1.Text = "9"; break;
case '%': label1.Text = "1"; break;
case '&': label1.Text = ";"; break;
case '(': label1.Text = "o"; break;
case ')': label1.Text = "_"; break;
case '\\': label1.Text = "B"; break;
case '*': label1.Text = "a"; break;
case '+': label1.Text = "S"; break;
case '-': label1.Text = "C"; break;
case '/': label1.Text = "R"; break;
case ' ': goto mark1; break;
}
if (i == tmp.Length)
{
    button1.Enabled = false;
    label2.Text = "@";
    goto end;
}
switch (tmp[i++])
{
    case 'a': label2.Text = "`"; break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 'b': label2.Text = "h"; break;
case 'c': label2.Text = "0"; break;
case 'd': label2.Text = "4"; break;
case 'e': label2.Text = "d"; break;
case 'f': label2.Text = "8"; break;
case 'g': label2.Text = "<"; break;
case 'h': label2.Text = "l"; break;
case 'i': label2.Text = "X"; break;
case 'j': label2.Text = "\\\"; break;
case 'k': label2.Text = "b"; break;
case 'l': label2.Text = "j"; break;
case 'm': label2.Text = "2"; break;
case 'n': label2.Text = "6"; break;
case 'o': label2.Text = "f"; break;
case 'p': label2.Text = "z"; break;
case 'q': label2.Text = ">"; break;
case 'r': label2.Text = "n"; break;
case 's': label2.Text = "Z"; break;
case 't': label2.Text = "^"; break;
case 'u': label2.Text = "c"; break;
case 'v': label2.Text = "k"; break;
case 'w': label2.Text = "j"; break;
case 'x': label2.Text = "3"; break;
case 'y': label2.Text = "7"; break;
case 'z': label2.Text = "g"; break;
case 'A': label2.Text = "@"; i--; goto end; break;
case 'B': label2.Text = "@"; i--; goto end; break;
case 'C': label2.Text = "@"; i--; goto end; break;
case 'D': label2.Text = "@"; i--; goto end; break;
case 'E': label2.Text = "@"; i--; goto end; break;
case 'F': label2.Text = "@"; i--; goto end; break;
case 'G': label2.Text = "@"; i--; goto end; break;
case 'H': label2.Text = "@"; i--; goto end; break;
case 'I': label2.Text = "@"; i--; goto end; break;
case 'J': label2.Text = "@"; i--; goto end; break;
case 'K': label2.Text = "@"; i--; goto end; break;
case 'L': label2.Text = "@"; i--; goto end; break;
case 'M': label2.Text = "@"; i--; goto end; break;
case 'N': label2.Text = "@"; i--; goto end; break;
case 'O': label2.Text = "@"; i--; goto end; break;
case 'P': label2.Text = "@"; i--; goto end; break;
case 'Q': label2.Text = "@"; i--; goto end; break;
case 'R': label2.Text = "@"; i--; goto end; break;
case 'S': label2.Text = "@"; i--; goto end; break;
case 'T': label2.Text = "@"; i--; goto end; break;
case 'U': label2.Text = "@"; i--; goto end; break;
case 'V': label2.Text = "@"; i--; goto end; break;
case 'W': label2.Text = "@"; i--; goto end; break;
case 'X': label2.Text = "@"; i--; goto end; break;
case 'Y': label2.Text = "@"; i--; goto end; break;
case 'Z': label2.Text = "@"; i--; goto end; break;
case '0': label2.Text = "G"; break;
case '1': label2.Text = "H"; break;
case '2': label2.Text = "J"; break;
case '3': label2.Text = "L"; break;
case '4': label2.Text = "M"; break;
case '5': label2.Text = "I"; break;
case '6': label2.Text = "N"; break;
case '7': label2.Text = "O"; break;
case '8': label2.Text = "K"; break;
case '9': label2.Text = "F"; break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case '#': label2.Text = "W"; break;
        case ':': label2.Text = "e"; break;
        case ';': label2.Text = "E"; break;
        case '<': label2.Text = "I"; break;
        case '>': label2.Text = "V"; break;
        case '?': label2.Text = "5"; break;
        case '@': label2.Text = "P"; break;
        case '[': label2.Text = "c"; break;
        case ']': label2.Text = "="; break;
        case '=': label2.Text = "?"; break;
        case ',': label2.Text = "H"; break;
        case '!': label2.Text = "["; break;
        case '"': label2.Text = "D"; break;
        case '$': label2.Text = "9"; break;
        case '%': label2.Text = "1"; break;
        case '&': label2.Text = ";"; break;
        case '(': label2.Text = "o"; break;
        case ')': label2.Text = " "; break;
        case '\\': label2.Text = "B"; break;
        case '*': label2.Text = "a"; break;
        case '+': label2.Text = "S"; break;
        case '-': label2.Text = "C"; break;
        case '/': label2.Text = "R"; break;

        default: label2.Text = "@"; break;
    }
end:

    btmp = label1.Text + label2.Text;
    try
    {
        sp.WriteLine(btmp);
    }
    catch (System.Exception ex)
    {
        label3.Text = ex.Message;
    }
}

private void button3_Click(object sender, EventArgs e)
{
    i = 0;
    maskedTextBox1.Text = "";
    label1.Text = "";
    label2.Text = "";
    label3.Text = "";
    label4.Text = "";
    label5.Text = "";
    label6.Text = "";
    button1.Text="Start";
    button1.Enabled=true;
}

private void maskedTextBox1_MaskInputRejected(object sender,
MaskInputRejectedEventArgs e)
{
    tmp = maskedTextBox1.Text;
    maskedTextBox1.Text = "";
    i = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โค้ดของโปรแกรมในไมโครคอนโทรลเลอร์สำหรับ 2 ตัวอักษร

```
;SEND 2 CHARACTER ONLY!!!
; R2 = Slave Address
; R3 = Data
; RECEIVE FROM COMPUTER IN SERIAL (MODE1,2) 9600 BUAD
```

```
RTCSCL      EQU    P1.1
RTCSDA      EQU    P1.2
ERRFAG      EQU    7AH

      ORG 0000H
      MOV P1,#00H
      MOV P0,#0FFH
      MOV PCON,#00H
      MOV SCON,#50H
      MOV TMOD,#20H
      MOV TH1,#0FDH
      SETB TR1

MAIN: MOV R0,#30H
WAIT: JNB RI,WAIT
      CLR RI
      MOV A,SBUF
      MOV @R0,A
      INC R0
      CJNE R0,#32H,WAIT

PRE:  MOV R0,#30H
      MOV R1,#00H

DATA1: MOV A,@R0
      MOV SBUF,A
WAITSB: JNB TI,WAITSB
      CLR TI
ENDC:  MOV 70H,R1
      MOV R2,70H
      MOV R3,A
      CALL IPW
      JB ERRFAG,$
      CALL DELAY

CALB:  CJNE R0,#30H,BACK
      INC R1
      INC R0
      LCALL DATA1
BACK:  JB P0.0,BACK
BACKB: MOV A,#62H
      MOV SBUF,A
WAITB: JNB TI,WAITB
      CLR TI
      SJMP MAIN

DELAY: MOV R5,#0FH
DE1:  MOV R6,#0FFH
DE2:  MOV R7,#0FFH
      DJNZ R7,$
      DJNZ R6,DE2
      DJNZ R5,DE1
      RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****IPW SUB*****
;PCF8574 Write Port
;in  =    R2  Address
;    =    R3  Data
;Out  =    ErrFlag    0=OK  1=Error
;Reg  =    A,R2
;*****
IPW:  SETB RTCSDA
      SETB RTCSCL
      CLR RTCSDA
      CALL CDEL
      CLR RTCSCL
      SETB RTCSDA
      MOV A,R2
      RL A
      ANL A,#00001110B
      ORL A,#01110000B
      CALL IPWRB
      JB  ERRFAG,IPW9
      MOV A,R3
      CALL IPWRB
IPW9: CLR RTCSDA
      SETB RTCSCL
      CALL CDEL
      SETB RTCSDA
      RET
;*****IPWRB SUB*****
;PCF8574 WRITE BYTE
;IN   =    A
;OUT  =    ERRFAG    0=OK  1=ERROR
;REG  =    A,R2
;*****
IPWRB: CLR ERRFAG
      MOV R2,#8
IPWRB1: RLC A
      MOV RTCSDA,C
      CALL CHIGH
      CALL CLOW
      DJNZ R2,IPWRB1
      SETB RTCSDA
      CALL CHIGH
      JNB RTCSDA,IPWRB2
      SETB ERRFAG
IPWRB2: CALL CLOW
      RET
;*****CLOCK&DELAY*****
CHIGH: SETB RTCSCL
      NOP
      NOP
      NOP
      RET
CLOW:  CLR RTCSCL
      NOP
      NOP
      NOP
      RET
CDEL:  NOP
      NOP
      NOP
      RET
      END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus

PCF8574

1 FEATURES

- Operating supply voltage 2.5 to 6 V
- Low standby current consumption of 10 µA maximum
- I²C-bus to parallel port expander
- Open-drain interrupt output
- 8-bit remote I/O port for the I²C-bus
- Compatible with most microcontrollers
- Latched outputs with high current drive capability for directly driving LEDs
- Address by 3 hardware address pins for use of up to 8 devices (up to 16 with PCF8574A)
- DIP16, or space-saving SO16 or SSOP20 packages.



The device consists of an 8-bit quasi-bidirectional port and an I²C-bus interface. The PCF8574 has a low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. It also possesses an interrupt line (INT) which can be connected to the interrupt logic of the microcontroller. By sending an interrupt signal on this line, the remote I/O can inform the microcontroller if there is incoming data on its ports without having to communicate via the I²C-bus. This means that the PCF8574 can remain a simple slave device.

2 GENERAL DESCRIPTION

The PCF8574 is a silicon CMOS circuit. It provides general purpose remote I/O expansion for most microcontroller families via the two-line bidirectional bus (I²C-bus).

The PCF8574 and PCF8574A versions differ only in their slave address as shown in Fig.10.

3 ORDERING INFORMATION

| TYPE NUMBER | PACKAGE | | |
|--------------------------|---------|---|----------|
| | NAME | DESCRIPTION | VERSION |
| PCF8574P; PCF8574AP | DIP16 | plastic dual in-line package; 16 leads (300 mil) | SOT38-4 |
| PCF8574T; PCF8574AT | SO16 | plastic small outline package; 16 leads; body width 7.5 mm | SOT162-1 |
| PCF8574TS; PCF8574ATS | SSOP20 | plastic shrink small outline package; 20 leads; body width 4.4 mm | SOT266-1 |

Remote 8-bit I/O expander for I²C-bus

PCF8574

4 BLOCK DIAGRAM

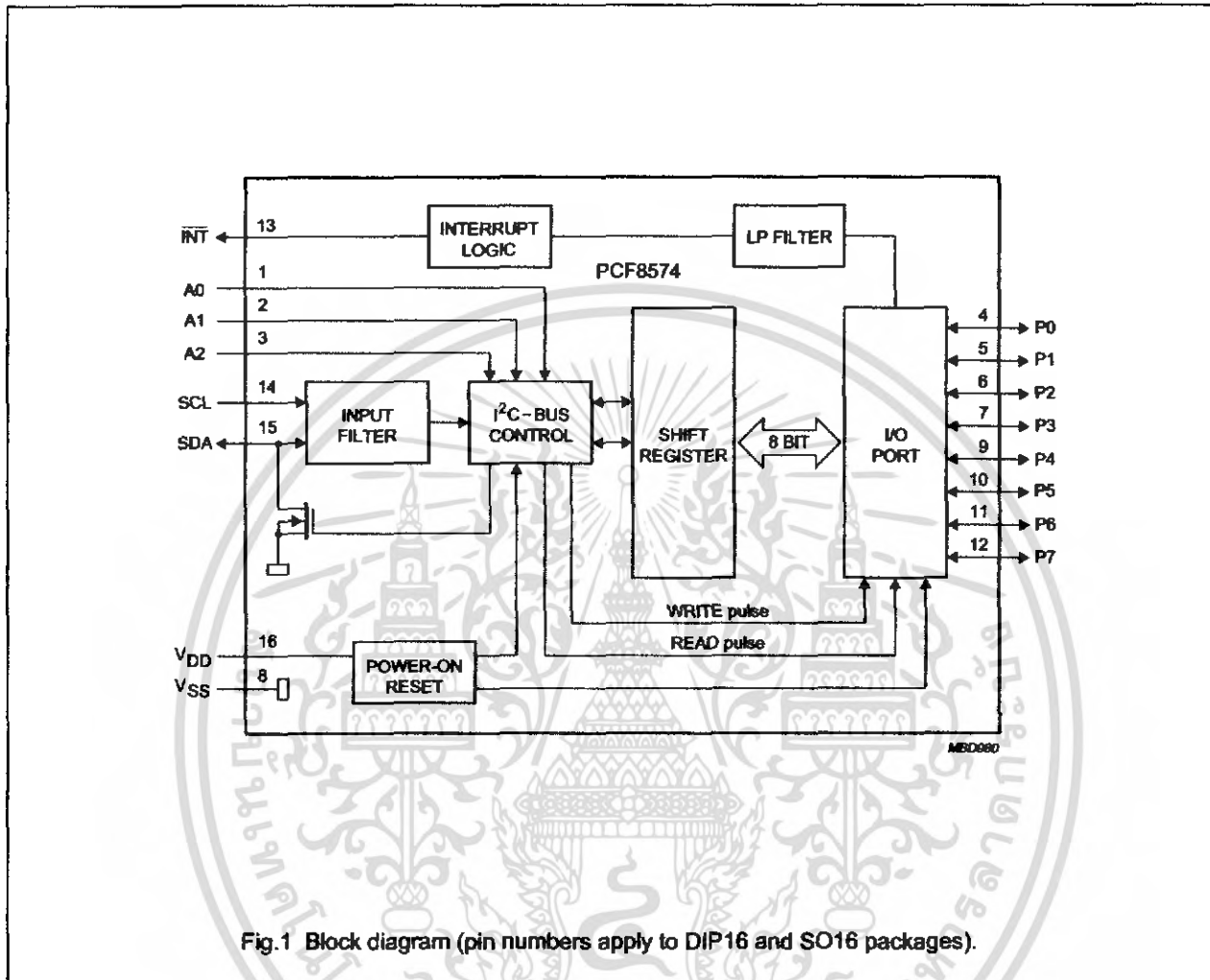


Fig.1 Block diagram (pin numbers apply to DIP16 and SO16 packages).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

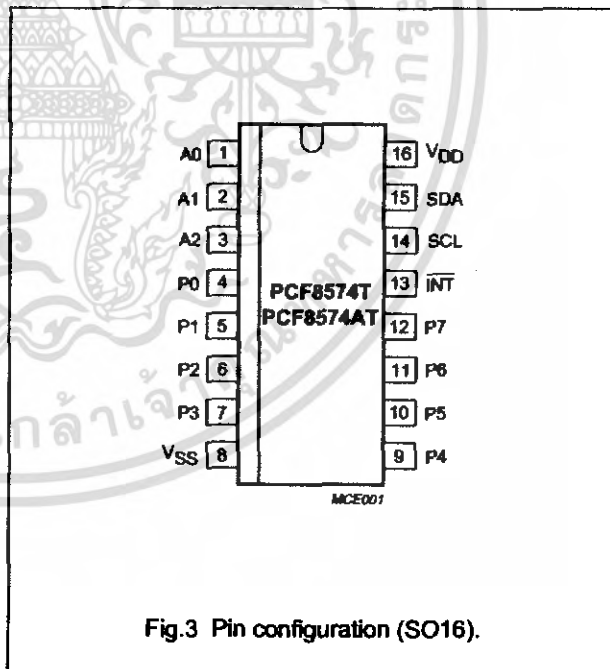
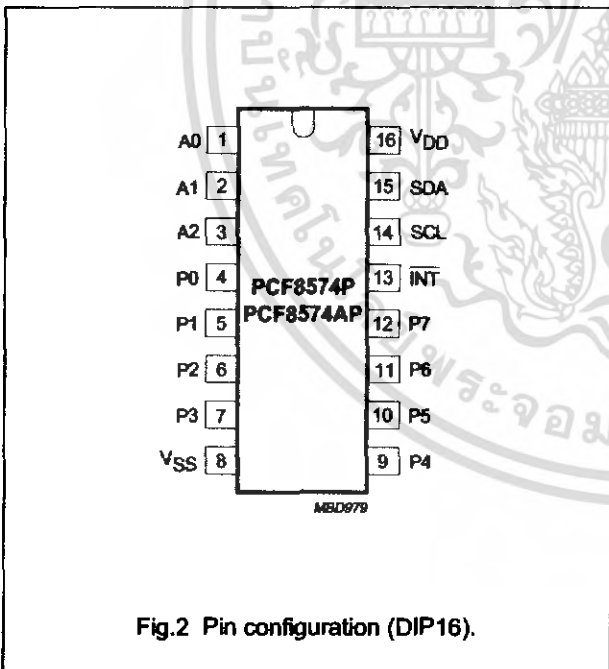
Remote 8-bit I/O expander for I²C-bus

PCF8574

5 PINNING

5.1 DIP16 and SO16 packages

| SYMBOL | PIN | DESCRIPTION |
|-----------------|-----|-------------------------------|
| A0 | 1 | address input 0 |
| A1 | 2 | address input 1 |
| A2 | 3 | address input 2 |
| P0 | 4 | quasi-bidirectional I/O 0 |
| P1 | 5 | quasi-bidirectional I/O 1 |
| P2 | 6 | quasi-bidirectional I/O 2 |
| P3 | 7 | quasi-bidirectional I/O 3 |
| V _{SS} | 8 | supply ground |
| P4 | 9 | quasi-bidirectional I/O 4 |
| P5 | 10 | quasi-bidirectional I/O 5 |
| P6 | 11 | quasi-bidirectional I/O 6 |
| P7 | 12 | quasi-bidirectional I/O 7 |
| INT | 13 | interrupt output (active LOW) |
| SCL | 14 | serial clock line |
| SDA | 15 | serial data line |
| V _{DD} | 16 | supply voltage |



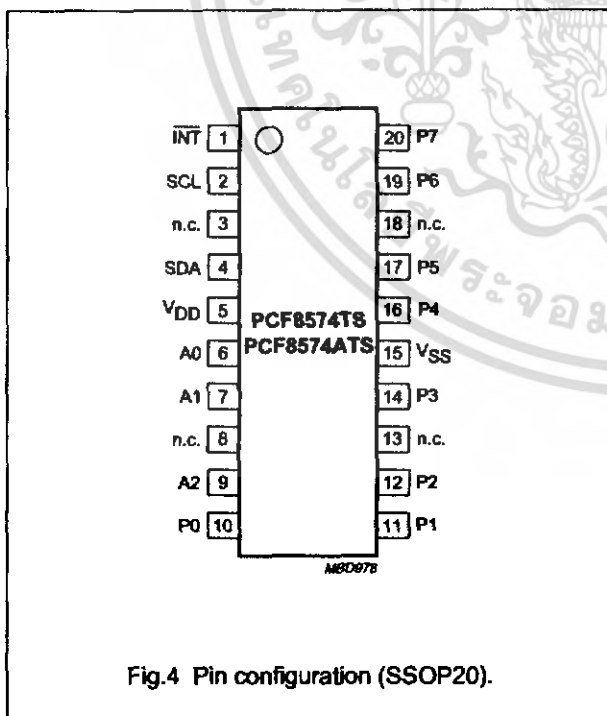
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus

PCF8574

5.2 SSOP20 package

| SYMBOL | PIN | DESCRIPTION |
|-----------------|-----|-------------------------------|
| INT | 1 | interrupt output (active LOW) |
| SCL | 2 | serial clock line |
| n.c. | 3 | not connected |
| SDA | 4 | serial data line |
| V _{DD} | 5 | supply voltage |
| A0 | 6 | address input 0 |
| A1 | 7 | address input 1 |
| n.c. | 8 | not connected |
| A2 | 9 | address input 2 |
| P0 | 10 | quasi-bidirectional I/O 0 |
| P1 | 11 | quasi-bidirectional I/O 1 |
| P2 | 12 | quasi-bidirectional I/O 2 |
| n.c. | 13 | not connected |
| P3 | 14 | quasi-bidirectional I/O 3 |
| V _{SS} | 15 | supply ground |
| P4 | 16 | quasi-bidirectional I/O 4 |
| P5 | 17 | quasi-bidirectional I/O 5 |
| n.c. | 18 | not connected |
| P6 | 19 | quasi-bidirectional I/O 6 |
| P7 | 20 | quasi-bidirectional I/O 7 |



Remote 8-bit I/O expander for I²C-bus

PCF8574

6 CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is for 2-way, 2-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 when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

6.1 Bit transfer

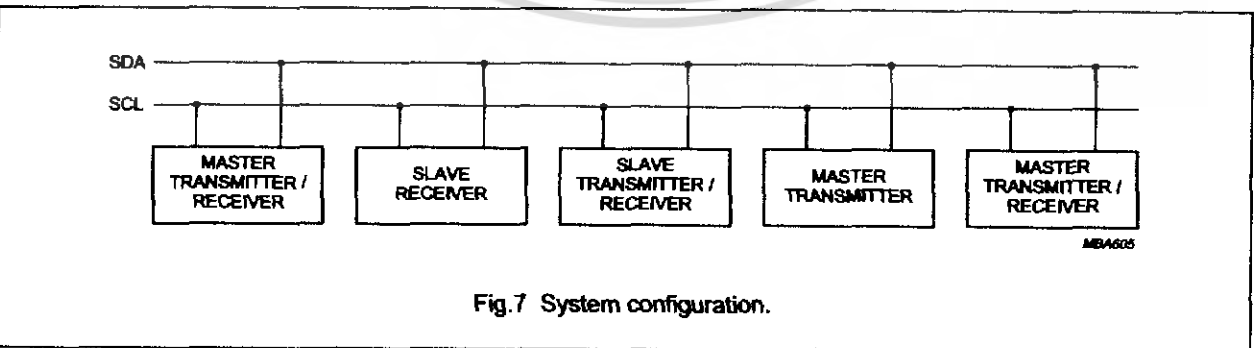
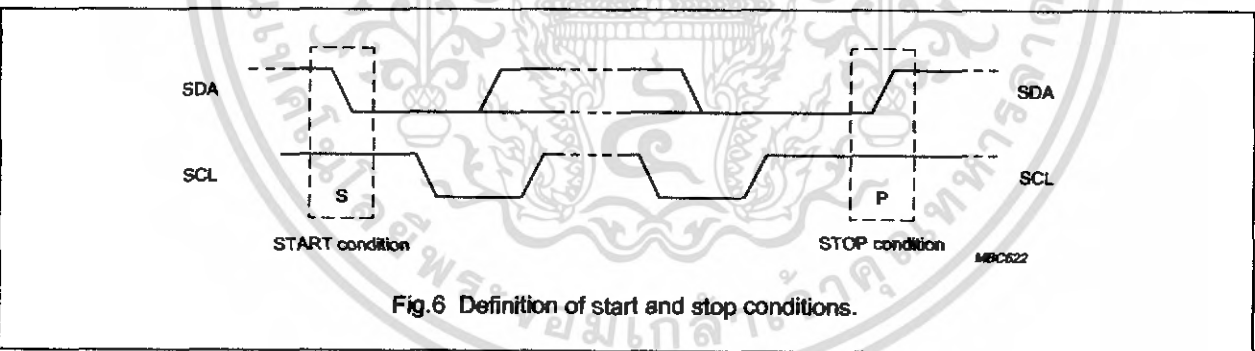
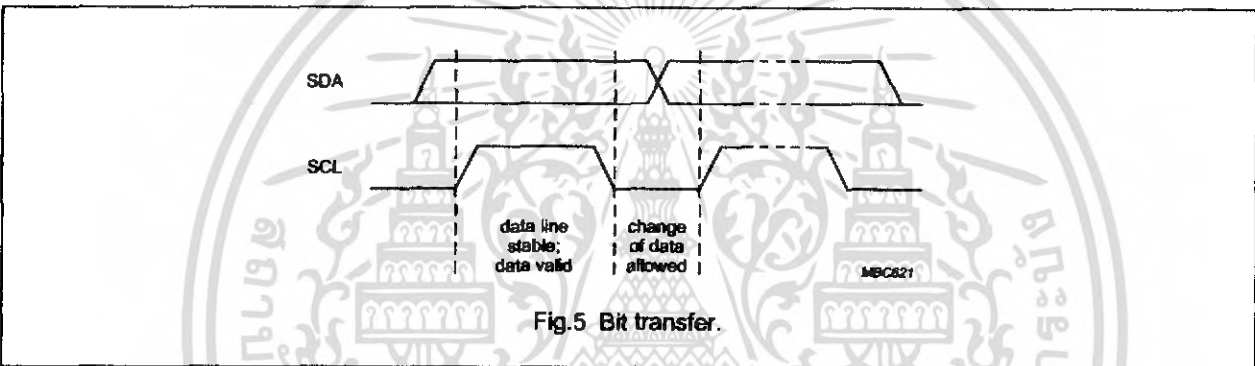
One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Fig.5).

6.2 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) (see Fig.6).

6.3 System configuration

A device generating a message is a 'transmitter', a device receiving is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves' (see Fig.7).



Remote 8-bit I/O expander for I²C-bus

PCF8574

6.4 Acknowledge

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit (see Fig.8). The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master 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, set-up and hold times must be taken into account.

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.

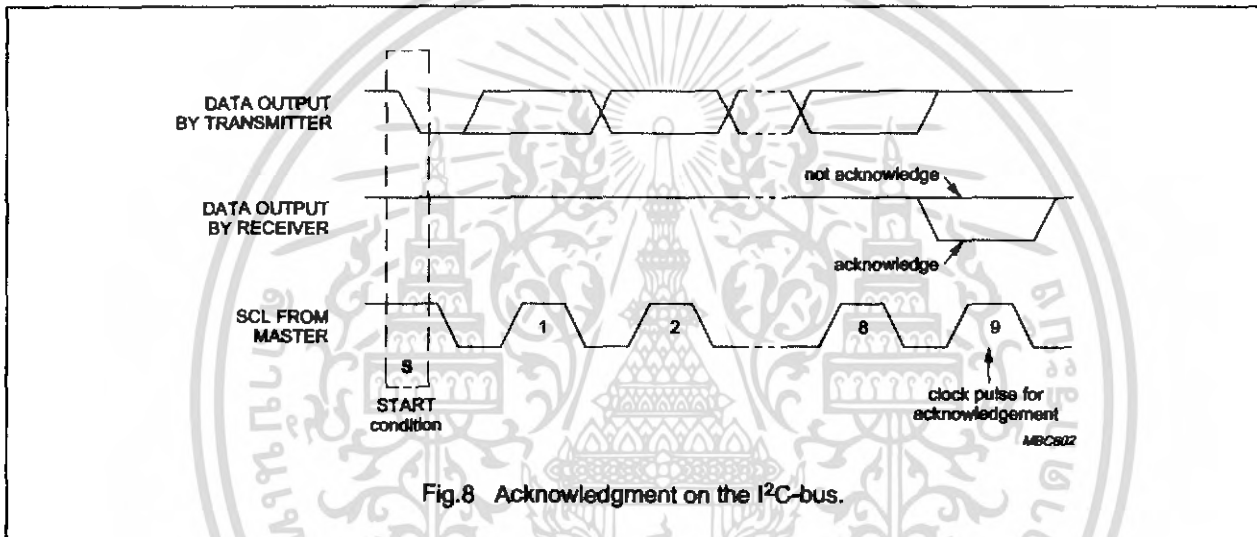
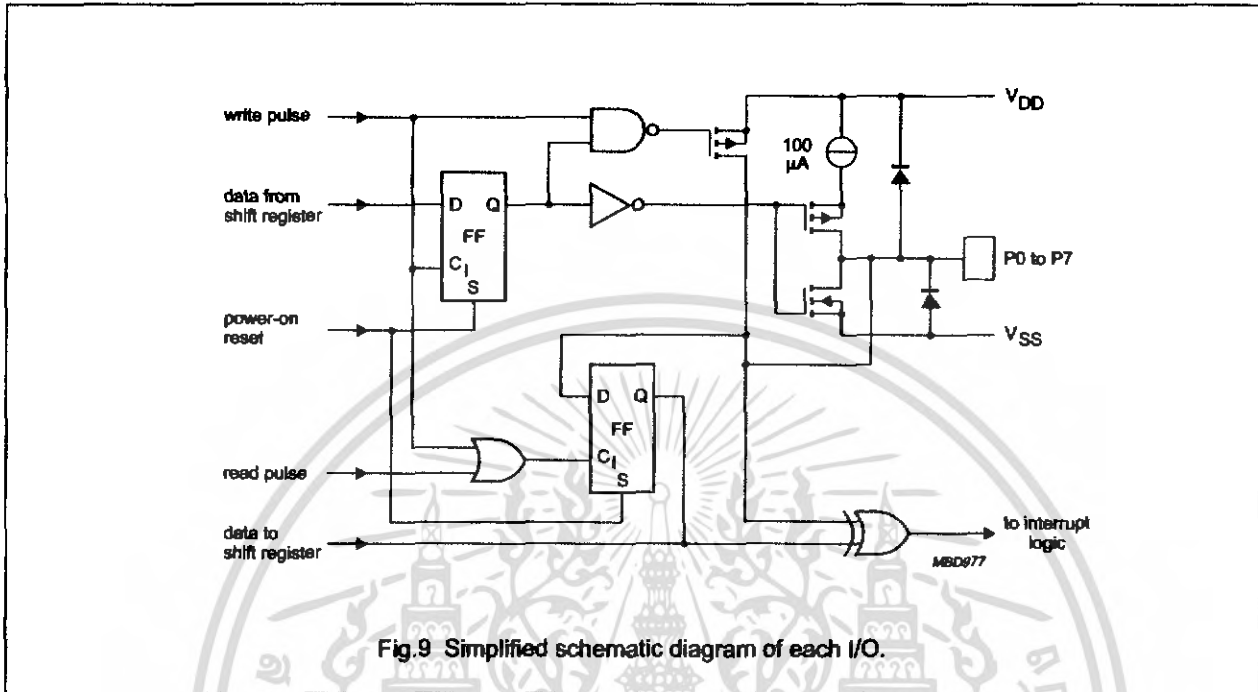


Fig.8 Acknowledgment on the I²C-bus.

Remote 8-bit I/O expander for I²C-bus

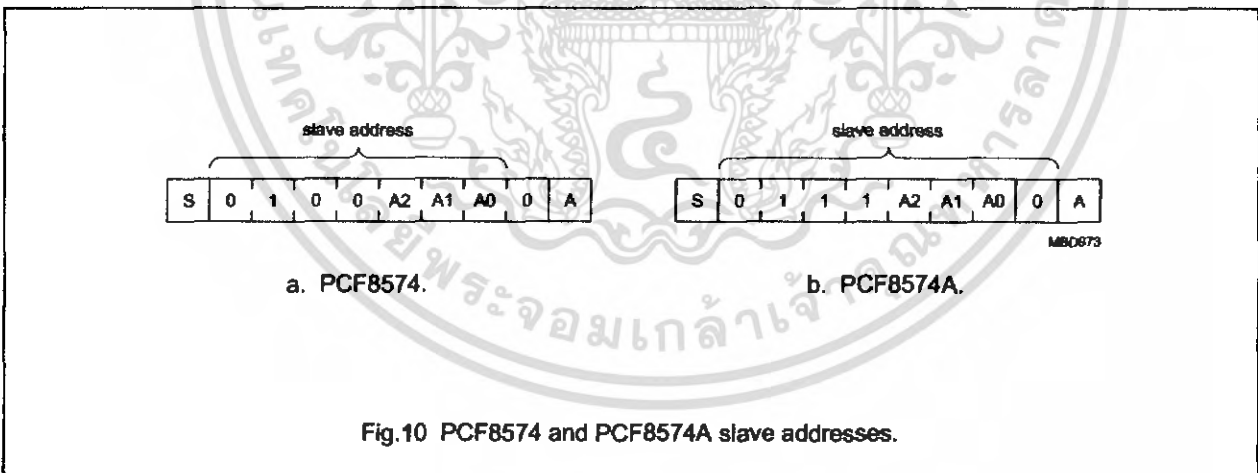
PCF8574

7 FUNCTIONAL DESCRIPTION



7.1 Addressing

For addressing see Figs 10, 11 and 12.



Each of the PCF8574's eight I/Os can be independently used as an input or output. Input data is transferred from the port to the microcontroller by the READ mode (see Fig.12). Output data is transmitted to the port by the WRITE mode (see Fig.11).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus

PCF8574

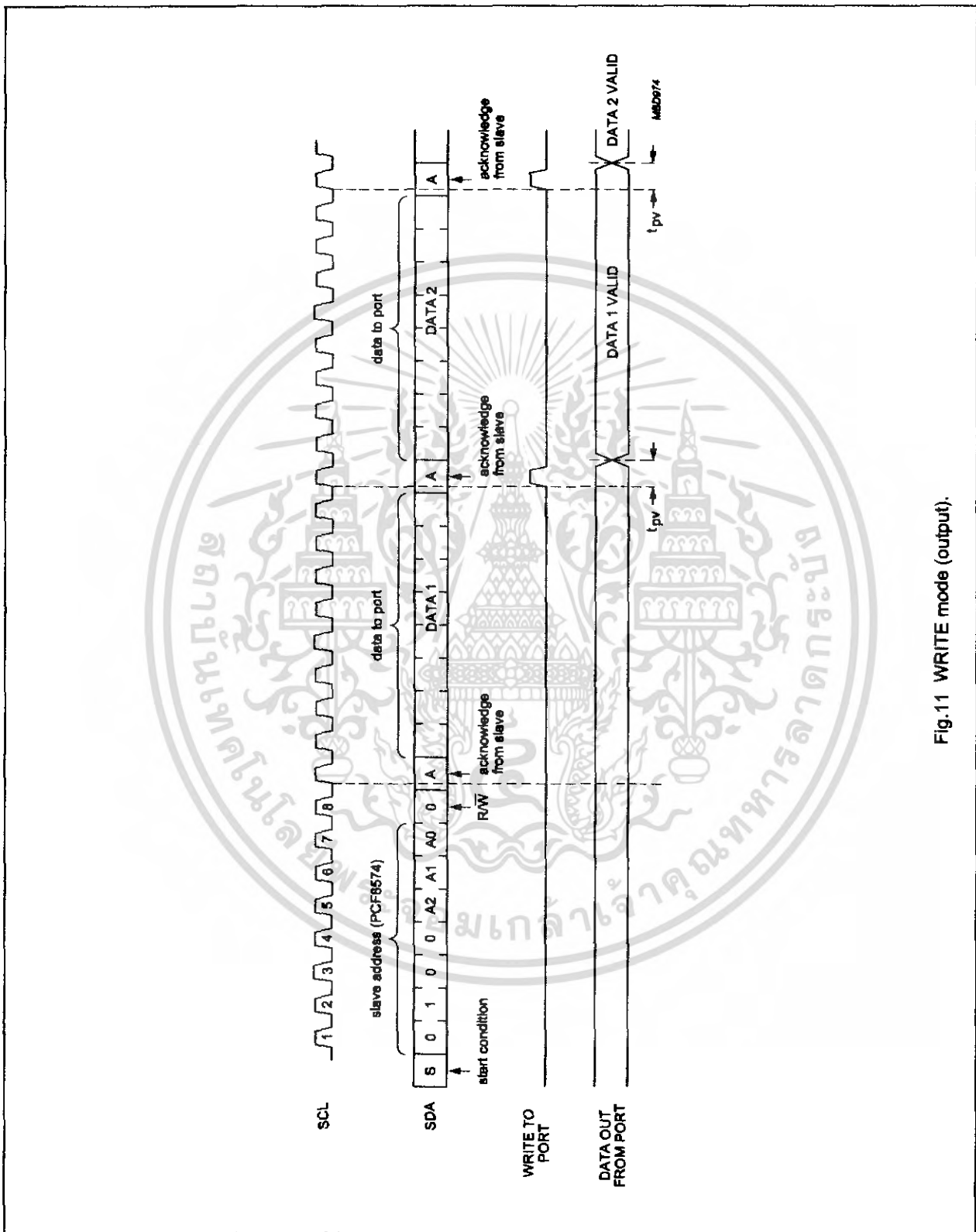


Fig.11 WRITE mode (output).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus

PCF8574

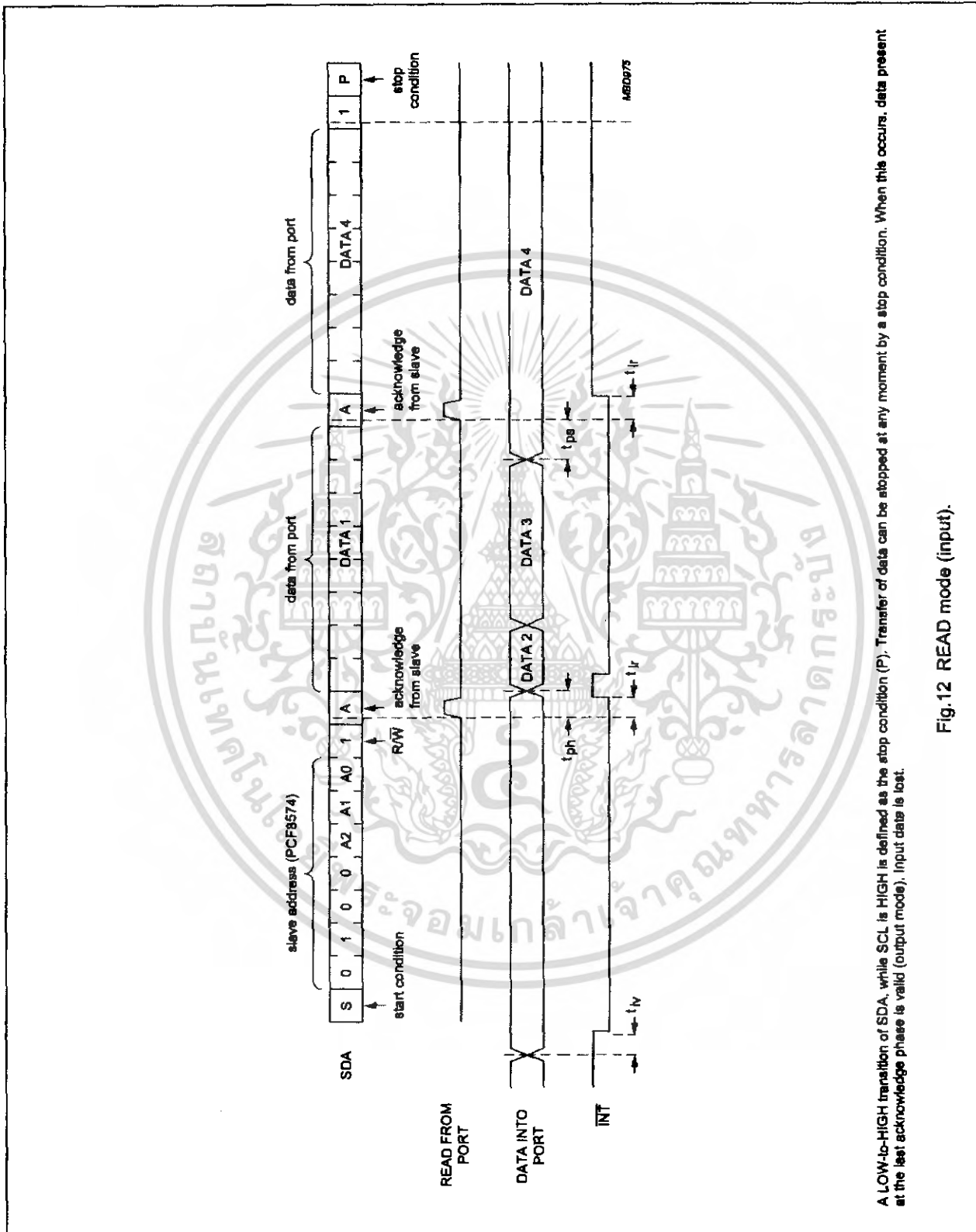


Fig.12 READ mode (input).

A LOW-to-HIGH transition of SDA, while SCL is HIGH is defined as the stop condition (P). Transfer of data can be stopped at any moment by a stop condition. When this occurs, data present at the last acknowledge phase is valid (output mode). Input data is lost.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus

PCF8574

7.2 Interrupt output

The PCF8574 provides an open-drain output ($\overline{\text{INT}}$) which can be fed to a corresponding input of the microcontroller (see Figs 13 and 14). This gives these chips a type of master function which can initiate an action elsewhere in the system.

An interrupt is generated by any rising or falling edge of the port inputs in the input mode. After time t_{IV} the signal $\overline{\text{INT}}$ is valid.

Resetting and reactivating the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from or written to the port which has generated the interrupt.

Resetting occurs as follows:

- In the READ mode at the acknowledge bit after the rising edge of the SCL signal
- In the WRITE mode at the acknowledge bit after the HIGH-to-LOW transition of the SCL signal

- Interrupts which occur during the acknowledge clock pulse may be lost (or very short) due to the resetting of the interrupt during this pulse.

Each change of the I/Os after resetting will be detected and, after the next rising clock edge, will be transmitted as $\overline{\text{INT}}$. Reading from or writing to another device does not affect the interrupt circuit.

7.3 Quasi-bidirectional I/Os

A quasi-bidirectional I/O can be used as an input or output without the use of a control signal for data direction (see Fig. 15). At power-on the I/Os are HIGH. In this mode only a current source to V_{DD} is active. An additional strong pull-up to V_{DD} allows fast rising edges into heavily loaded outputs. These devices turn on when an output is written HIGH, and are switched off by the negative edge of SCL. The I/Os should be HIGH before being used as inputs.

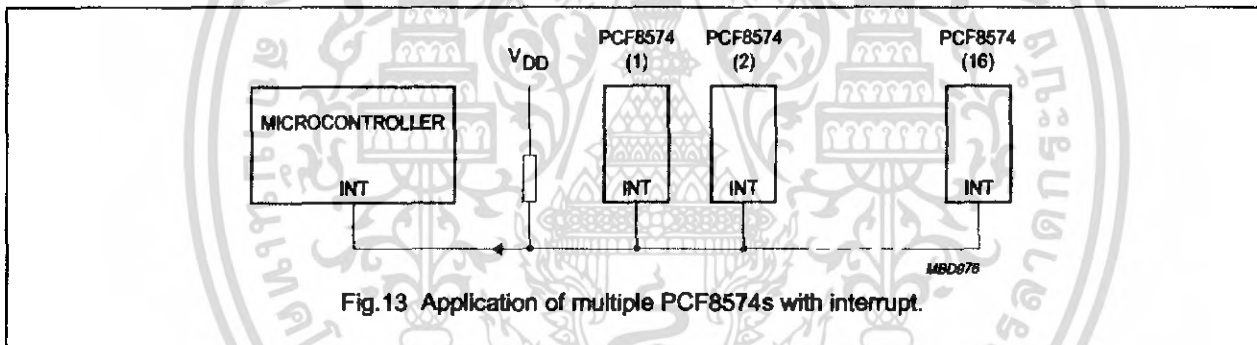


Fig.13 Application of multiple PCF8574s with interrupt.

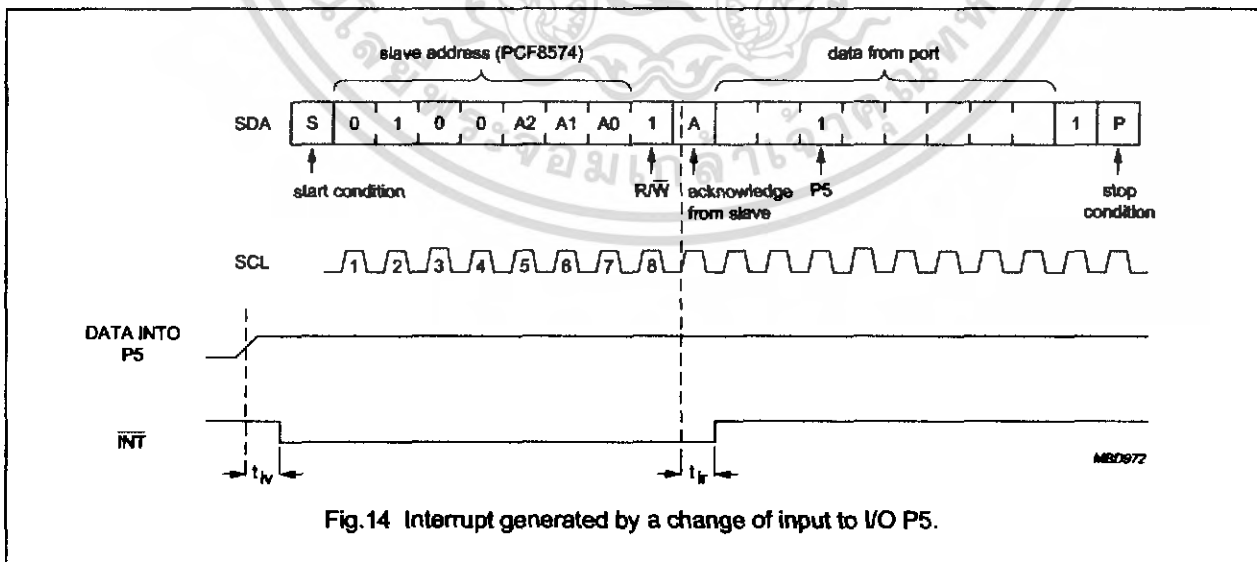


Fig.14 Interrupt generated by a change of input to I/O P5.

Remote 8-bit I/O expander for I²C-bus

PCF8574

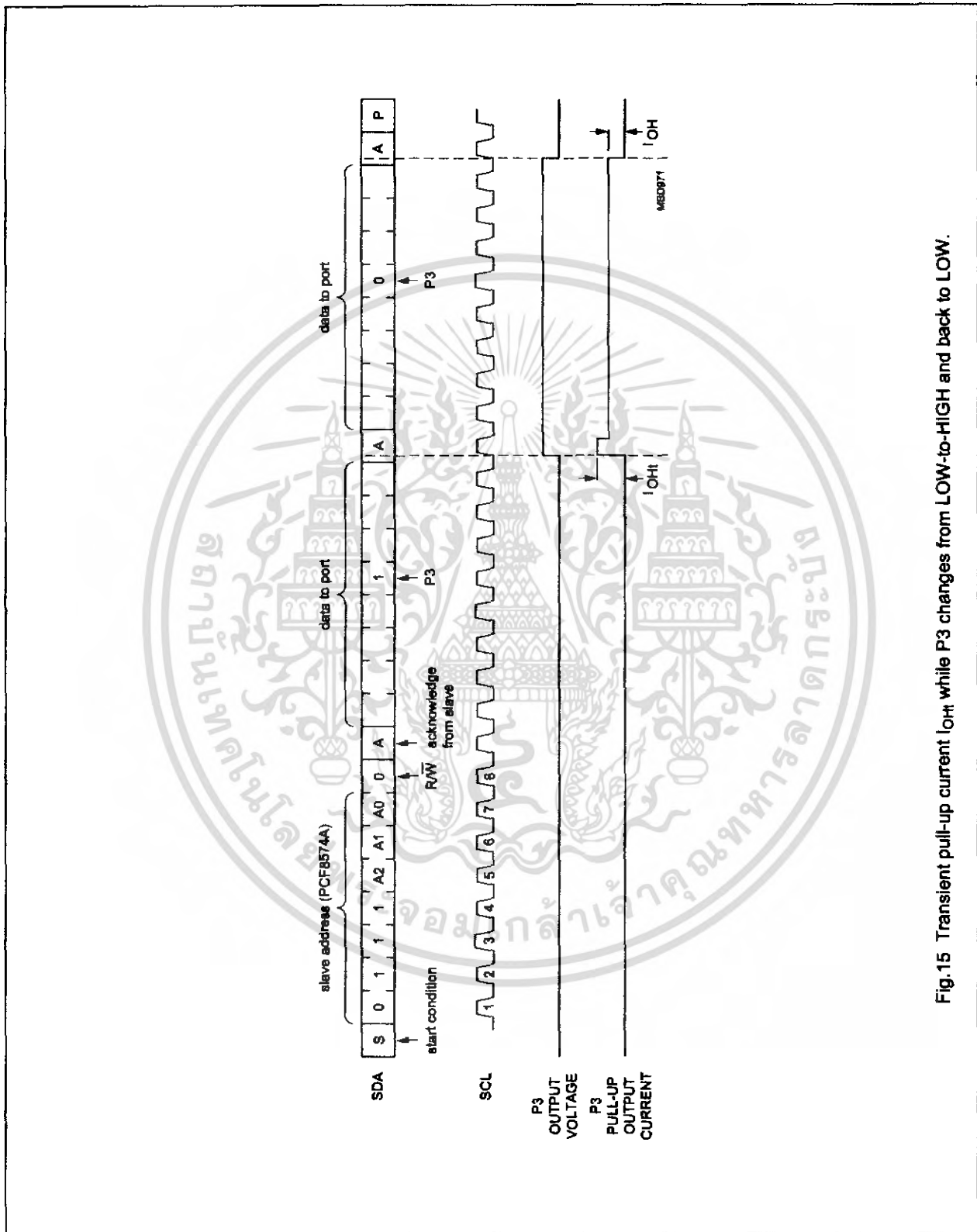


Fig.15 Transient pull-up current I_{OH} while P3 changes from LOW-to-HIGH and back to LOW.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus

PCF8574

8 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 60134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|------------------|------------------------------|-----------------------|-----------------------|------|
| V _{DD} | supply voltage | -0.5 | +7.0 | V |
| V _I | input voltage | V _{SS} - 0.5 | V _{DD} + 0.5 | V |
| I _I | DC input current | - | ±20 | mA |
| I _O | DC output current | - | ±25 | mA |
| I _{DD} | supply current | - | ±100 | mA |
| I _{SS} | supply current | - | ±100 | mA |
| P _{tot} | total power dissipation | - | 400 | mW |
| P _O | power dissipation per output | - | 100 | mW |
| T _{stg} | storage temperature | -65 | +150 | °C |
| T _{amb} | ambient temperature | -40 | +85 | °C |

9 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However it is good practice to take normal precautions appropriate to handling MOS devices (see "Handling MOS devices").

10 DC CHARACTERISTICS

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

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|------------------------------------|--------------------------|--|--------------------|------|-----------------------|------|
| Supply | | | | | | |
| V _{DD} | supply voltage | | 2.5 | - | 6.0 | V |
| I _{DD} | supply current | operating mode; V _{DD} = 6 V; no load; V _I = V _{DD} or V _{SS} ; f _{SCL} = 100 kHz | - | 40 | 100 | µA |
| I _{stb} | standby current | standby mode; V _{DD} = 6 V; no load; V _I = V _{DD} or V _{SS} | - | 2.5 | 10 | µA |
| V _{POR} | Power-on reset voltage | V _{DD} = 6 V; no load; V _I = V _{DD} or V _{SS} ; note 1 | - | 1.3 | 2.4 | V |
| Input SCL; input/output SDA | | | | | | |
| V _{IL} | LOW level input voltage | | -0.5 | - | +0.3V _{DD} | V |
| V _{IH} | HIGH level input voltage | | 0.7V _{DD} | - | V _{DD} + 0.5 | V |
| I _{OL} | LOW level output current | V _{OL} = 0.4 V | 3 | - | - | mA |
| I _L | leakage current | V _I = V _{DD} or V _{SS} | -1 | - | +1 | µA |
| C _i | input capacitance | V _I = V _{SS} | - | - | 7 | pF |

Remote 8-bit I/O expander for I²C-bus

PCF8574

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|--|---|--------------------|------|-----------------------|------|
| I/Os | | | | | | |
| V _{IL} | LOW level input voltage | | -0.5 | - | +0.3V _{DD} | V |
| V _{IH} | HIGH level input voltage | | 0.7V _{DD} | - | V _{DD} + 0.5 | V |
| I _{IHL(max)} | maximum allowed input current through protection diode | V _I ≥ V _{DD} or V _I ≤ V _{SS} | - | - | ±400 | µA |
| I _{OL} | LOW level output current | V _{OL} = 1 V; V _{DD} = 5 V | 10 | 25 | - | mA |
| I _{OH} | HIGH level output current | V _{OH} = V _{SS} | 30 | - | 300 | µA |
| I _{OHt} | transient pull-up current | HIGH during acknowledge (see Fig.15); V _{OH} = V _{SS} ; V _{DD} = 2.5 V | - | -1 | - | mA |
| C _i | input capacitance | | - | - | 10 | pF |
| C _o | output capacitance | | - | - | 10 | pF |
| Port timing; C_L ≤ 100 pF (see Figs 11 and 12) | | | | | | |
| t _{pv} | output data valid | | - | - | 4 | µs |
| t _{su} | input data set-up time | | 0 | - | - | µs |
| t _h | input data hold time | | 4 | - | - | µs |
| Interrupt INT (see Fig.14) | | | | | | |
| I _{OL} | LOW level output current | V _{OL} = 0.4 V | 1.6 | - | - | mA |
| I _L | leakage current | V _I = V _{DD} or V _{SS} | -1 | - | +1 | µA |
| TIMING; C_L ≤ 100 pF | | | | | | |
| t _v | input data valid time | | - | - | 4 | µs |
| t _r | reset delay time | | - | - | 4 | µs |
| Select inputs A0 to A2 | | | | | | |
| V _{IL} | LOW level input voltage | | -0.5 | - | +0.3V _{DD} | V |
| V _{IH} | HIGH level input voltage | | 0.7V _{DD} | - | V _{DD} + 0.5 | V |
| I _{LI} | input leakage current | pin at V _{DD} or V _{SS} | -250 | - | +250 | nA |

Note

1. The Power-on reset circuit resets the I²C-bus logic at V_{DD} < V_{POR} and sets all I/Os to logic 1 (with current source to V_{DD}).

Remote 8-bit I/O expander for I²C-bus

PCF8574

11 I²C-BUS TIMING CHARACTERISTICS

| SYMBOL | PARAMETER | MIN. | TYP. | MAX. | UNIT |
|---|------------------------------|------|------|------|------|
| I²C-bus timing (see Fig.16; note 1) | | | | | |
| f _{SCL} | SCL clock frequency | – | – | 100 | kHz |
| t _{SW} | 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

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

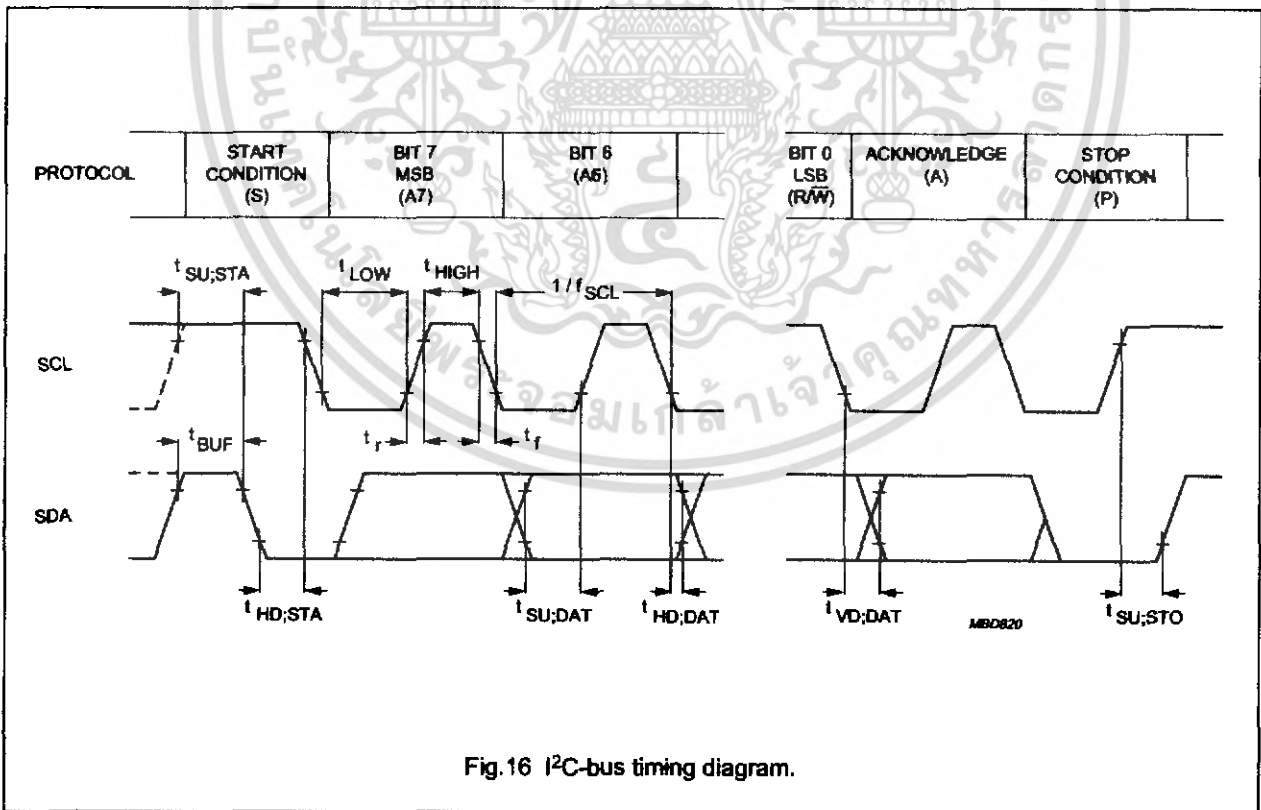


Fig.16 I²C-bus timing diagram.

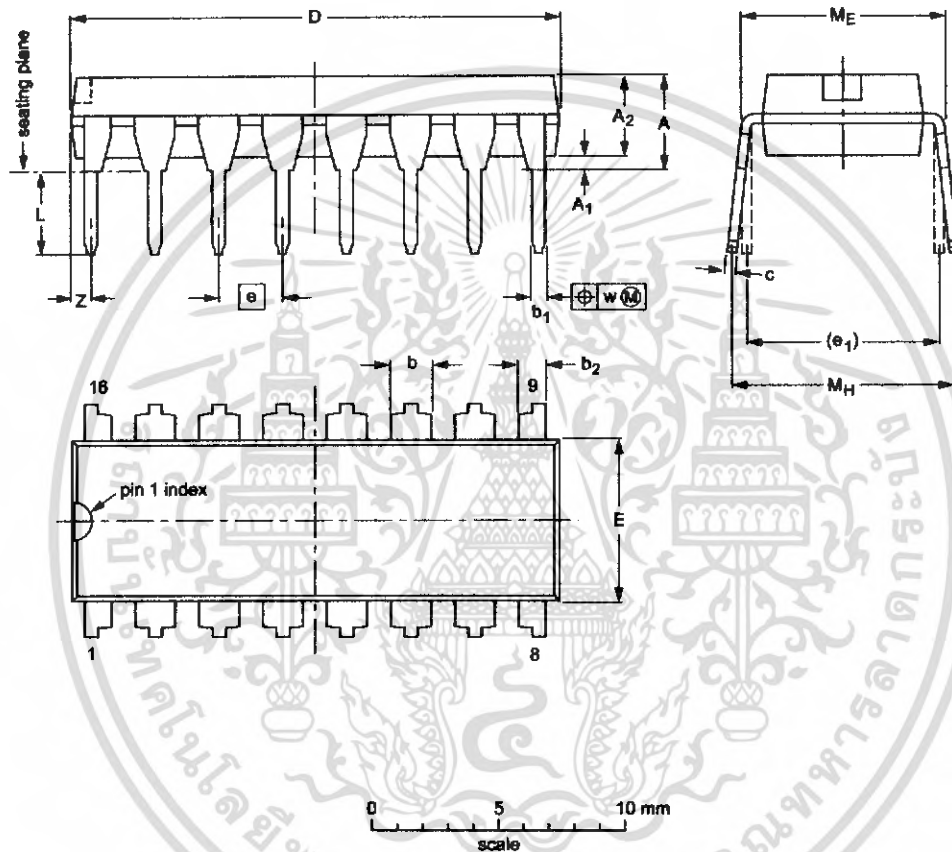
Remote 8-bit I/O expander for I²C-bus

PCF8574

12 PACKAGE OUTLINES

DIP16: plastic dual in-line package; 16 leads (300 mil)

SOT38-4



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

| UNIT | A max. | A ₁ min. | A ₂ max. | b | b ₁ | b ₂ | c | D ⁽¹⁾ | E ⁽¹⁾ | e | e ₁ | L | M _E | M _H | w | Z ⁽¹⁾ max. |
|--------|--------|---------------------|---------------------|----------------|----------------|----------------|----------------|------------------|------------------|------|----------------|--------------|----------------|----------------|-------|-----------------------|
| mm | 4.2 | 0.51 | 3.2 | 1.73 1.30 | 0.53 0.38 | 1.25 0.85 | 0.36 0.23 | 19.50 18.55 | 6.48 6.20 | 2.54 | 7.62 | 3.60 3.05 | 8.25 7.80 | 10.0 8.3 | 0.254 | 0.76 |
| inches | 0.17 | 0.020 | 0.13 | 0.068 0.051 | 0.021 0.015 | 0.049 0.033 | 0.014 0.009 | 0.77 0.73 | 0.26 0.24 | 0.10 | 0.30 | 0.14 0.12 | 0.32 0.31 | 0.39 0.33 | 0.01 | 0.030 |

Note

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

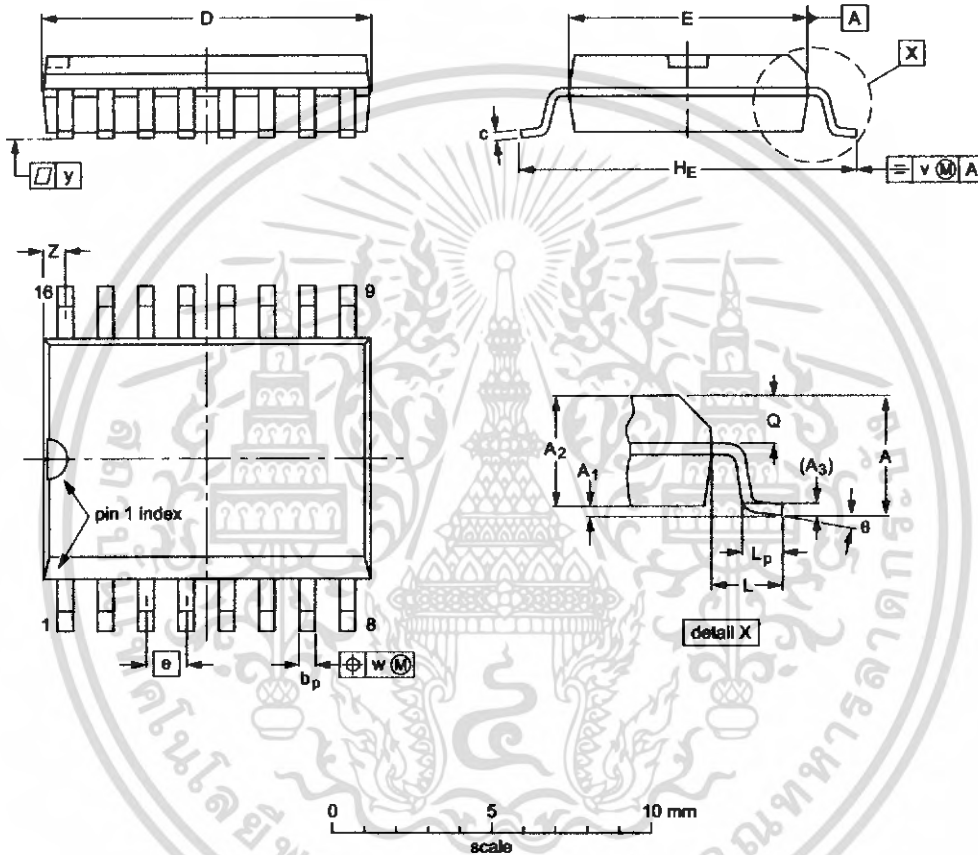
| OUTLINE VERSION | REFERENCES | | | | EUROPEAN PROJECTION | ISSUE DATE |
|-----------------|------------|-------|------|--|---------------------|----------------------|
| | IEC | JEDEC | EIAJ | | | |
| SOT38-4 | | | | | | 92-11-17 95-01-14 |

Remote 8-bit I/O expander for I²C-bus

PCF8574

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 | 7.8 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-013 | | | | 97-06-22 99-12-27 |

Remote 8-bit I/O expander for I²C-bus

PCF8574

SSOP20: plastic shrink small outline package; 20 leads; body width 4.4 mm

SOT266-1

