สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

**ภาควิชาครุศาสตร์วิศวกรรม**

**คณะครุศาสตร์อุตสาหกรรม**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ใบรับรองปริญญานิพนธ์**

---

**ปริญญานิพนธ์** ระบบตรวจสอบบุคคลเข้าหน่วยงาน

PERSONAL CHECK-IN SYSTEM

**ชื่อนักศึกษา**

| | | | | |
|---|---|---|---|---|
| 1. นายกิติพงศ์ | พรหมเลิศ | รหัสประจำตัว | 40031403 |
| 2. นายโกมิน | สัตย์มั่น | รหัสประจำตัว | 40031404 |
| 3. นายชิษณุพงศ์ | ธัญญลักษณ์ | รหัสประจำตัว | 40031410 |

**หลักสูตร** ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** อิเล็กทรอนิกส์และคอมพิวเตอร์

**อาจารย์ผู้ควบคุมปริญญานิพนธ์**

1. ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา
2. อาจารย์โกศล ตราชู

T032815

| คณะกรรมการสอบปริญญานิพนธ์ | | ลายมือชื่อ |
|---|---|---|
| 1. อาจารย์อำพล | ทองระอา | ..... |
| 2. อาจารย์โกศล | ตราชู | ..... |
| 3.อาจารย์สุระชัย | พิมพ์สาลี | ..... |
| 4. อาจารย์อมรชัย | ชัยชนะ | ..... |
| 5. อาจารย์ปิยะ | จิตธรรมมาภิรมย์ | ..... |

**วันเดือนปีที่สอบ** วันที่ 23 พฤศจิกายน 2541 **เวลา** 05.00 น. ถึง 06.00 น.

**สถานที่สอบ** ห้อง ค.310 คณะครุศาตร์อุตรสาหกรรม สจล.

**ภาควิชารับรองแล้ว**

ลงชื่อ .....

( ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา )

**หัวหน้าภาควิชาครุศาสตร์วิศวกรรม**

**วันที่** 12 **เดือน** ก.ย. **พ.ศ.** 41

**เลขหมู่**.....
**เลขทะเบียน** 32815
**วัน, เดือน, ปี** 10 ธ.ค. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ปริญญานิพนธ์

## ระบบตรวจสอบบุคคลเข้าหน่วยงาน

## PERSONAL CHECK- IN SYSTEM


นายกิติพงศ์ พรหมเลิศ

นายโกมิน สัตย์มั่น

นายชิษณุพงศ์ ธัญญลักษณ์


ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ปริญญานิพนธ์

**เรื่อง** ระบบตรวจสอบบุคคลเข้าหน่วยงาน

PERSONAL CHECK- IN SYSTEM

**ผู้จัดทำ**

1. นายกิติพงศ์ พรหมเลิศ
2. นายโกมิน สัตย์มั่น
3. นายชิษณุพงศ์ ธัญญลักษณ์

**อาจารย์ที่ปรึกษา**

**ลงนาม** .....

( ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา )

**ลงนาม** .....

( อาจารย์โกศล ตราชู )

**หัวหน้าภาควิชาครุศาสตร์วิศวกรรม**

**ลงนาม** .....

( ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ปริญญานิพนธ์

**เรื่อง** ระบบตรวจสอบบุคคลเข้าหน่วยงาน

PERSONAL CHECK- IN SYSTEM

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ และบัตรรหัสบาร์โค้ด
2. เพื่อออกแบบระบบตรวจสอบบุคคลเข้าหน่วยงาน
3. เพื่อสร้างระบบตรวจสอบบุคคลเข้าหน่วยงาน
4. เพื่อนำระบบตรวจสอบบุคคลเข้าหน่วยงานไปใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้มีความรู้ความเข้าใจเกี่ยวกับไมโครคอนโทรลเลอร์ และบัตรรหัสบาร์โค้ด
2. ได้ต้นแบบระบบตรวจสอบบุคคลเข้าหน่วยงาน
3. ได้เครื่องต้นแบบระบบตรวจสอบบุคคลเข้าหน่วยงาน
4. นำระบบตรวจสอบบุคคลเข้าหน่วยงานไปใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ระบบตรวจสอบบุคคลเข้าหน่วยงาน

นายกิติพงศ์ พรหมเลิศ
นายโกมิน สัตย์มั่น
นายชิษณุพงศ์ ธัญญลักษณ์

อาจารย์ที่ปรึกษา
ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา
อาจารย์โกศล ตราชู

ปีการศึกษา 2541

**บทคัดย่อ**

ปริญญานิพนธ์ฉบับนี้ เสนอระบบตรวจสอบบุคคลเข้าหน่วยงาน โดยใช้บัตรรหัสบาร์โค้ด โครงสร้างของระบบประกอบด้วยหน่วยประมวลผล 2 ส่วน ได้แก่ ส่วนไมโครคอมพิวเตอร์และส่วนไมโครโปรเซสเซอร์เชื่อมต่อกันเป็นระบบโครงข่ายโดยมีไมโครคอมพิวเตอร์เป็นศูนย์กลางทำหน้าที่ควบคุมและรับส่งข้อมูลแบบอนุกรมกับไมโครโปรเซสเซอร์แต่ละตัวโดยผ่าน RS-485 ซึ่งสามารถเชื่อมต่อกับไมโครโปรเซสเซอร์ได้ถึง 32 จุดต่อ แต่ละจุดต่อทำหน้าที่อ่านข้อมูลบัตรรหัสบาร์โค้ดและจากการกดรหัสด้วยปุ่มกดมีการบันทึกฐานข้อมูลและเวลาของการเข้าบนหน่วยความจำของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PERSONAL CHECK- IN SYSTEM

MR.KITIPONG PROMLERT
MR.KOMIN SATMUN
MR.CHITSANUPONG THANYALUK

ADVISORS
Assited Prof.Dr.TEERAPON TEPHATSADIN NA AYUTAYA
MR.KOSON TRACHU
1998

**ABSTRACT**

This thesis presents the project of check- in personal system using barcode card. The system consists of microcomputer and microprocessor units.The microcomputer unit is a centralized host that controls and communicates data with microprocessor unit. The microcomputer unit can be connected to maximum 32 microprocessor units via RS-485. Each microprocessor unit reads code from barcode cards or key matrix switches. Check- in time will be consequently recorded in microcomputer unit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลือในการให้คำแนะนำจากคณาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน โดยเฉพาะอย่างยิ่งจากอาจารย์ผู้ควบคุมปริญญานิพนธ์ ขอขอบพระคุณไว้ ณ ที่นี้ และขอขอบคุณกำลังใจจากทุกท่านที่เกี่ยวข้องที่ช่วยผลักดันให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้ และที่สำคัญที่ขอกราบขอบพระคุณ คุณพ่อ และคุณแม่ ผู้บังเกิดเกล้าผู้เป็นแรงกำลังอันยิ่งใหญ่ทั้งกำลังใจ กำลังทรัพย์ และเป็นผู้ให้ตลอดมา อนึ่ง ประโยชน์และคุณความดีใด ๆ ก็ตามที่เกิดจากปริญญานิพนธ์ฉบับนี้ ขอมอบให้แก่ คุณพ่อ คุณแม่ ผู้ให้กำเนิด และคุณครูบาอาจารย์ ที่ได้ประสิทธิ์ประสาทวิชาตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบาญ


| เรื่อง | หน้า |
|---|---|
| **บทคัดย่อภาษาไทย** | **I** |
| **บทคัดย่อภาษาอังกฤษ** | **II** |
| **กิตติกรรมประกาศ** | **III** |
| **สารบาญ** | **IV** |
| **สารบาญตาราง** | **VI** |
| **สารบาญภาพ** | **VII** |
| **บทที่ 1 บทนำ** | 1 |
| 1.1 หลักการ และเหตุผล | 1 |
| 1.2 วัตถุประสงค์ของปริญญานิพนธ์ | 1 |
| 1.3 เนื้อหาโดยสังเขปของปริญญานิพนธ์ | 1 |
| **บทที่ 2 ทฤษฎี และหลักการ** | 3 |
| 2.1กล่าวนำ | 3 |
| 2.2 บาร์โค้ด | 3 |
| 2.2.1 ความเป็นมาของบาร์โค้ด | 3 |
| 2.2.2 หลักการบาร์โค้ด | 3 |
| 2.3 การสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-485 | 9 |
| 2.3.1 รูปแบบการสื่อสารข้อมูลแบบ RS-485 | 10 |
| 2.3.2 RS-485 Protocol | 12 |
| 2.4 ตัวแสดงผล Lcd แบบจุด | 16 |
| **บทที่ 3 การออกแบบ การสร้าง และการทำงาน** | 22 |
| 3.1 กล่าวนำ | 22 |
| 3.2 การออกแบบวงจร | 23 |
| 3.2.1 ส่วนของวงจรรับข้อมูล | 23 |
| 3.2.2 ส่วนควบคุมอุปกรณ์โซลินอยด์,ลำโพง | 26 |
| 3.2.3 ส่วนสื่อสารข้อมูล | 30 |
| 3.3 การออกแบบโปรแกรม | 31 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบาญ (ต่อ)


| เรื่อง | หน้า |
|---|---|
| **บทที่ 4 การทดลอง และผลการทดลอง** | 36 |
| 4.1 การทดลองส่วนรับข้อมูล | 36 |
| 4.1.1 การทดลองส่วนรับข้อมูลจากบัตรรหัสบาร์โค้ด | 36 |
| 4.1.2 การทดลองส่วนรับข้อมูลจากเมทริกซ์สวิตช์ | 37 |
| 4.2 การทดลองการแสดงผลของระบบ | 38 |
| 4.3 การทดลองการสื่อสารข้อมูล | 39 |
| **บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา** | 41 |
| 5.1 สรุป | 41 |
| 5.2 ปัญหา | 40 |
| 5.3 แนวทางแก้ไข และการพัฒนาโครงงาน | 42 |
| **ภาคผนวก ก โปรแกรมการทำงาน** | 43 |
| **ภาคผนวก ข วงจร และแผ่นวงจรพิมพ์** | 67 |
| **ภาคผนวก ค รายละเอียดข้อมูล และคุณสมบัติของอุปกรณ์** | 73 |
| **ประวัติผู้แต่ง** | 92 |
| **บรรณานุกรม** | 95 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบาญตาราง


| ตาราง | หน้า |
|---|---|
| ตารางที่ 2.1 อักษรของรหัส 39 เทียบกับค่าของแต่ละอักษร | 8 |
| ตารางที่ 2.2 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลทาง EIA | 10 |
| ตารางที่ 4.1 ผลจากการวัดแรงดันของวงจรรับข้อมูลจากบัตรรหัสบาร์โค้ด | 37 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบาญภาพ


| รูปภาพ | หน้า |
|---|---|
| รูปที่ 2.1 การแทนค่าเลขฐานสองของแถบต่างๆ | 4 |
| รูปที่ 2.2 สัญลักษณ์ที่อ่านได้จากหัวอ่าน | 5 |
| รูปที่ 2.3 โครงสร้างของรหัสบาร์โค้ด 39 | 7 |
| รูปที่ 2.4 โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-485 | 9 |
| รูปที่ 2.5 เครือข่ายของ RS-485 แบบที่ใช้สายนำสัญญาณสองเส้น (Two Wire) | 11 |
| รูปที่ 3.1 โครงสร้างการทำงานของระบบตรวจสอบบุคคลเข้าหน่วยงาน | 22 |
| รูปที่ 3.2 ชุดหัวอ่านบัตรรหัสบาร์โค้ด | 24 |
| รูปที่ 3.3 วงจรส่วนรับข้อมูลจากหัวอ่านบัตรรหัสบาร์โค้ด | 24 |
| รูปที่ 3.4 คีย์เมทริกซ์สวิตช์ | 25 |
| รูปที่ 3.5 ส่วนรับข้อมูลคีย์เมทริกซ์สวิตช์ | 26 |
| รูปที่ 3.6 วงจรควบคุมการทำงานของลำโพง | 27 |
| รูปที่ 3.7 วงจรควบคุมโซลินอยด์ | 28 |
| รูปที่ 3.8 อุปกรณ์แสดงผลชุดหัวอ่าน | 29 |
| รูปที่ 3.9 วงจรสร้างสัญญาณเตือน เมื่อมีการรูดบัตรรหัสบาร์โค้ดไม่ผ่าน | 29 |
| รูปที่ 3.10 วงจรสื่อสารข้อมูลผ่าน RS-485 | 30 |
| รูปที่ 3.11 ผังโปรแกรมของการรูดบัตรและการกดคีย์สวิตช์ | 32 |
| รูปที่ 3.12 ผังโปรแกรมของการสื่อสารข้อมูล | 33 |
| รูปที่ 4.1 วงจรส่วนรับข้อมูลจากบัตรรหัสบาร์โค้ด | 36 |
| รูปที่ 4.2 วงจรส่วนรับข้อมูลจากคีย์เมทริกซ์สวิตช์ | 38 |
| รูปที่ 4.3 การต่อวงจรส่วนแสดงผล LCD | 39 |
| รูปที่ 4.4 การต่อวงจรสื่อสารข้อมูล | 40 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 หลักการ และเหตุผล

ปัจจุบันหน่วยงานต่างๆทั้งของภาครัฐบาล และเอกชนเริ่มตระหนักถึงความสำคัญของการตรวจสอบข้อมูลบุคคลที่ผ่านเข้าออกหน่วยงานโดยเฉพาะพิจารณาถึงความปลอดภัยภายในหน่วยงาน ความสะดวกรวดเร็วในการตรวจสอบ และสามารถนำข้อมูลมาประมวลผลตามวัตถุประสงค์ของหน่วยงานได้

เหตุผลที่ใช้คอมพิวเตอร์มาทำงาน เพราะเครื่องคอมพิวเตอร์มีทั่วไปในหน่วยงานต่างๆ ทำงานได้รวดเร็ว ถูกต้อง แม่นยำ และแก้ไขเพิ่มเติมการทำงานต่างๆได้ง่าย

เครื่องตรวจสอบบุคคลเข้าหน่วยงาน สามารถนำไปใช้งานโดยการตรวจสอบรหัสบาร์โค้ด หรือการกดรหัสด้วยปุ่มกดที่ใช้ไมโครคอมพิวเตอร์เป็นศูนย์กลาง ทำหน้าที่ควบคุม และรับส่งข้อมูลอนุกรมกับไมโครโปรเซสเซอร์แต่ละตัว

## 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ และบัตรรหัสบาร์โค้ด
2 เพื่อศึกษา และออกแบบการทำงานของระบบตรวจสอบบุคคลเข้าหน่วยงาน
3. เพื่อสร้างระบบตรวจสอบบุคคลเข้าหน่วยงาน
4. เพื่อสามารถนำระบบตรวจสอบบุคคลเข้าหน่วยงานไปใช้งานจริงได้

## 1.3 เนื้อหาโดยสังเขปของปริญญานิพนธ์

ในปริญญานิพนธ์นี้ได้แบ่งเนื้อหาออกเป็น 5 บท โดยมีเนื้อหาโดยสังเขป ในบทที่ 2 ถึง 5 ดังนี้

บทที่ 2 ทฤษฎี และหลักการ กล่าวถึงทฤษฎีของรหัสบาร์โค้ด หลักการพื้นฐาน รหัสโค้ด 39 มาตรฐานการรับส่งข้อมูลอนุกรม RS-485 และ LCD

บทที่ 3 การออกแบบ การสร้าง และการทำงานของระบบตรวจสอบบุคคลเข้าหน่วยงาน กล่าวถึงรายละเอียดโครงสร้างของระบบวงจรที่ใช้งานจริงการทำงานของวงจรต่างๆในแต่ละส่วนที่เป็นส่วนประกอบของเครื่อง และการออกแบบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลอง ผลการทดลอง และการทดสอบ กล่าวถึงผลการทดลองของระบบ โดยทดลองส่วนรับส่งข้อมูล ส่วนควบคุมอุปกรณ์ และส่วนการแสดงผล

บทที่ 5 บทสรุป ปัญหา แนวทางการแก้ไข และการพัฒนา เป็นบทสรุป และวิจารณ์ระบบที่ออกแบบ รวมถึงข้อเสนอแนะ และแนวทางในการพัฒนา

ในส่วนสุดท้ายของปริญญานิพนธ์เป็นภาคผนวกซึ่งมีรายละเอียดดังต่อไปนี้

ภาคผนวก ก โปรแกรมการทำงาน

ภาคผนวก ข วงจรแผ่นพิมพ์ และรายอุปกรณ์

ภาคผนวก ค รายละเอียดข้อมูล และคุณสมบัติของอุปกรณ์

ประวัติผู้แต่ง

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎี และหลักการ

## 2.1 กล่าวนำ

ในระบบการควบคุมการเข้าออกโดยใช้บัตรนักศึกษามาใช้งานเป็นบัตรสำหรับผ่านเข้าออกระบบ โดยใช้ข้อมูลบนบัตรเป็นข้อมูลประจำตัวของผู้ถือบัตร รวมถึงการกดรหัสผ่านซึ่งบัตรนักศึกษามีรูปแบบของข้อมูลบันทึกอยู่ในรหัสบาร์โค้ด

## 2.2 บาร์โค้ด ( Bar Code )

### 2.2.1 ความเป็นมาของบาร์โค้ด

สหรัฐอเมริกาได้ออกสิทธิบัตรรับรองบาร์โค้ดขึ้นเมื่อปี ค.ศ.1949 ในแบบที่เรียกว่า Circular bar code ต่อมาในปี ค.ศ. 1960 ก็มีการรับรองบาร์โค้ดแบบนี้เรียกว่า Rall identification symbol หลังจากนั้นเป็นต้นมาเทคนิคของบาร์โค้ดรูปแบบต่างๆ ก็มีมากขึ้น และเริ่มใช้งานจริงจังเมื่อปี ค.ศ. 1970 เมื่อคณะกรรมการบริหารด้านห้างสรรพสินค้าของสหรัฐอเมริกา ได้นำรหัสที่เรียกว่า UPC (Universal Product Code) ซึ่งเป็นรหัสที่ใช้กันมากในสินค้า ออกเผยแพร่และใช้กันแพร่หลายในสหรัฐอเมริกา และยุโรป ตั้งแต่ ค.ศ. 1973 และ ค.ศ. 1977 ตามลำดับ

การใช้งานในด้านอื่นๆ เริ่มตั้งแต่ปี ค.ศ. 1980 เป็นต้นมา แต่คนส่วนใหญ่เริ่มคุ้นเคยกับบาร์โค้ดเป็นอย่างดีจากรหัสสินค้าและการชำระเงินที่คอมพิวเตอร์รวมออกมาจากการอ่านบาร์โค้ดสินค้าเหล่านั้น จากความสะดวกเหล่านี้สามารถลดพนักงาน ณ จุดนี้ลงได้

ในปี ค.ศ. 1981 มีห้างสรรพสินค้ามากกว่า 4,000 แห่ง ในสหรัฐอเมริกาและแคนนาดาใช้บาร์โค้ดในธุรกิจนี้ นอกจากนี้ยังใช้กับกิจการอื่น เช่น ห้องสมุด บริการสุขภาพ งานบริการ การผลิตสินค้า เป็นต้น

### 2.2.2 หลักการของบาร์โค้ด

บาร์โค้ดเป็นการแทนข้อมูลที่เป็นรหัสฐานสอง (Binary Code) ในรูปแบบของแถบขาว-ดำที่มีความกว้างแคบต่างกัน วางเรียงขนานสลับกัน ด้วยจำนวนของแถบรหัสสลับกันไป ขึ้นอยู่กับข้อมูลที่ต่างกันและชนิดของบาร์โค้ดที่เลือกใช้ต่างกันด้วย

การเข้ารหัสของบาร์โค้ด (Bar Code) แบ่งเป็น 2 วิธีการคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. จะใช้สีของบาร์โค้ดนำมาเข้ารหัส โดยใช้แถบสีดำแทน "1" และแถบสีขาวหรือแถบว่างเป็น "0" ลักษณะเช่นนี้เรียกว่า "เดลต้าโค้ด" (Delta Code)

2. จะใช้ความกว้างของแถบรหัสนำมาเข้ารหัส โดยถ้าเป็นแถบกว้างจะเป็น "1" และแถบแคบจะเป็น "0" ลักษณะเช่นนี้เรียกว่า "วิดท์โค้ด" (Width Code) การเข้ารหัสชนิดนี้จะไม่สนใจสีของแถบเลย

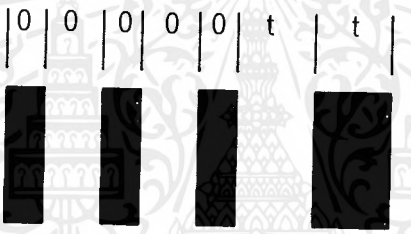
แถบขาว-ดำ มีลักษณะ และชื่อที่ใช้คือ

-แถบสีดำที่มีความกว้างมาก เรียกว่า Wide Bar

-เส้นหรือแถบสีดำที่มีความกว้างน้อย เรียกว่า Narrow Bar

-ช่องว่าง (แถบสีขาว) ที่มีความกว้างมาก เรียกว่า Wide Space

-ช่องว่าง (เส้นสีดำ) ที่มีความกว้างน้อย เรียกว่า Narrow Space




**รูปที่ 2.1** การแทนค่าเลขฐานสองของแถบต่างๆ

**การอ่านบาร์โค้ด**

ในการอ่านบาร์โค้ดใช้หลักการเปลี่ยนรหัสบาร์โค้ดให้เป็นรหัสแอสกี้ โดยอาศัยความแตกต่างกันระหว่างแถบเข้มกับพื้นที่ว่าง โดยพื้นที่ว่าง (ปกติจะเป็นสีขาวหรือสีอ่อน) จะมีการสะท้อนกลับของแสงได้มากกว่าบริเวณที่เป็นแถบเข้ม (ซึ่งใช้สีดำหรือสีที่มีความเข้มมาก) หัวอ่าน (Bar Code Reader) จะประกอบด้วยตัวกำเนิดแสงที่ผ่านเลนส์ออกมา โดยถูกบังคับทิศทางให้มีจุดรวมแสงเล็กที่สุด กับตัวรับแสงที่มีความไวสูง ทั้งสองอย่างนี้จะบรรจุไว้ในหัวอ่านเดียวกัน ที่มีหลายรูปแบบ แต่แบบที่เป็นพื้นฐานที่สุดอยู่ในรูปคล้ายปากกาขนาดใหญ่ (Wand type)

หัวอ่านจะสแกนผ่านรหัสบาร์โค้ดในขณะที่ตัวกำเนิดแสง จะทำให้เกิดแสงส่งผ่านเลนส์ไปกระทบบนรหัสบาร์โค้ด และสะท้อนกลับจากแถบกลับไปยังตัวรับแสง (Photosensor) ที่เกิดค่าความแตกต่างขึ้น ตามหลักการสะท้อนกลับในแต่ละแถบ ทำให้เกิดสภาวะลอจิก "1" และสภาวะลอจิก "0" ขึ้นตามที่กล่าวมาข้างต้น ซึ่งเมื่อรวมสภาวะลอจิก "1" และ "0" ทั้งหมด ตลอดความกว้างของทุกแถบแล้วจะตรงกับแพตเทอร์นที่ได้กำหนดไว้แล้ว ในหัวอ่านรหัสบาร์โค้ดจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

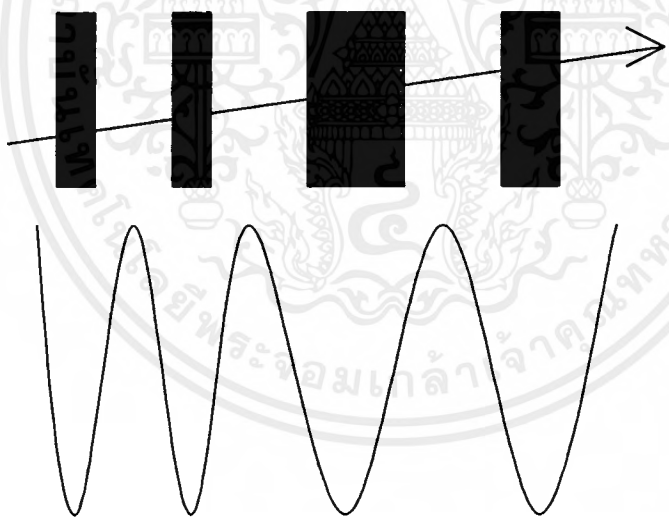
ตัวกำเนิดแสงสีแดงหรือสีขาว แต่ส่วนใหญ่จะใช้สีแดงเนื่องจากแสงสีขาวต้องการพลังงาน และความเข้มของแสงสูงมากกว่าแสงสีแดง แสงสีแดงสามารถอ่านรหัสที่พิมพ์ด้วยสีต่างๆได้ทุกสี ยกเว้นรหัสที่พิมพ์ด้วยสีแดง

องค์ประกอบสำคัญสองประการที่จำเป็นอย่างมากในการอ่านบาร์โค้ดได้ถูกต้อง

ประการแรกคือ พื้นที่ภายในแถบและช่องว่าง จะต้องทำให้เกิดความแตกต่างของการสะท้อนกลับอย่างมาก (Contrast) เช่น แถบสีดำและช่องว่างสีขาว เป็นต้น ซึ่งปกติความแตกต่างนี้จะต้องอยู่ในช่วงระหว่างอัตรา 80-90% ขึ้นไป

ประการที่สองคือ ความกว้างระหว่างแถบกว้าง หรือช่องว่างต่อแถบแคบ หรือช่องว่างแคบจะเป็นอัตราส่วน 2 : 0.5 , 2 : 1 และ 3 : 1

ตามหลักของบาร์โค้ดแล้ว สัญญาณที่อ่านได้จากหัวอ่านจะไม่ขึ้นกับชนิดของบาร์โค้ด แต่จะขึ้นกับแถบขาว-ดำ ที่รูดผ่าน คือ ถ้าเป็นแถบดำสัญญาณที่ได้จากหัวอ่านจะเป็น “1” ถ้าเป็นแถบขาว สัญญาณที่อ่านได้จะเป็น “0” ความกว้างของสัญญาณที่อ่านได้จะเท่ากับความกว้างของแถบขาว-ดำ



**รูปที่ 2.2** แสดงสัญลักษณ์ที่อ่านได้จากหัวอ่าน

**ชนิดของบาร์โค้ด**

ปัจจุบันบาร์โค้ดที่นิยมใช้กันแพร่หลาย แบ่งได้เป็น

1. ชนิดรหัส 2 ใน 5 (2 of 5 code)
2. ชนิดรหัส 2 ใน 5 แบบสอดแทรก (interleaved 2 of 5)
3. ชนิดรหัส 3 ใน 9 (3 of 9 or 39 code)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ชนิดรหัส Codabar

5. ชนิดรหัส UPC (Universal Product Code)

6. ชนิดรหัส EAN (European Article Numbering)

7. ชนิดรหัส Code 93

ในเครื่องตรวจสอบบุคคลเข้าหน่วยงานชุดนี้ใช้ชนิดรหัส 3 ใน 9 ในการตรวจสอบข้อมูล มีรายละเอียดดังนี้

**บาร์โค้ดชนิด 3 ใน 9 (3 of 9 or Code 39)**

ชื่อของบาร์โค้ดชนิดนี้บอกถึงโครงสร้างของรหัสว่า 3 ใน 9 ส่วนต่อตัวอักษรจะเป็นแถบกว้าง และที่เหลือเป็นแถบแถบอีก 5 แถบ แต่ละตัวอักษรใน code 39 จะแสดงเป็นกลุ่มของแถบดำ (bar) 5 แถบ และแถบว่าง (space) 4 แถบ แต่ละตัวอักษรที่สมบูรณ์จะต้องรวมตัวอักษร start และ stop เข้าไปด้วย โดยใช้เป็นตัว asterisk (*) ข้อมูลทั้ง 43 ตัวอักษรประกอบด้วยตัวเลข 10 ตัว คือ 0-9 ตัวอักษร 26 ตัว A-Z space และเครื่องหมายอีก 6 ตัว (-, ., &, /, +, %)

คุณสมบัติที่สำคัญของ code 39 คือมี self-checking ซึ่งทำให้ข้อมูลปลอดภัยสูง และถ้ามีเครื่องอ่านที่ดี และถ้าบาร์โค้ดที่มีคุณภาพสูง จะพบข้อผิดพลาด (error) ได้เพียง 1 ครั้งใน 70 ล้านครั้งของการอ่าน และถ้าบาร์โค้ดมีการพิมพ์มาอย่างดี (ใช้เครื่องพิมพ์ที่มีคุณภาพดีกว่า dot matrix) จะพบความผิดพลาดเพียง 1 ครั้งใน 3 ล้านครั้ง

code 39 เป็นรหัสที่มีความยาวเปลี่ยนแปลงได้ ขึ้นกับเครื่องอ่านที่ใช้และ code 39 มี self-checking จึงไม่ต้องมี check character code 39 เป็นตัวอักษรชนิดไม่ต่อเนื่อง โดยจะมีช่องว่างระหว่างตัวอักขระ และเป็นรหัสที่อ่านได้ 2 ทาง คือสามารถอ่านได้ทั้งจากซ้ายไปขวา หรือขวาไปซ้ายก็ได้ ขนาดของรหัส 39 จะเปลี่ยนแปลงไปตามความกว้าง สำหรับรหัส 39 ที่มีความหนาแน่นสูงจะมี 9.4 ตัวอักษรต่อนิ้ว และรหัส 39 ที่มีความหนาแน่นต่ำจะมี 1.4 ตัวอักษรต่อนิ้ว

**สรุปคุณลักษณะของรหัส 39**

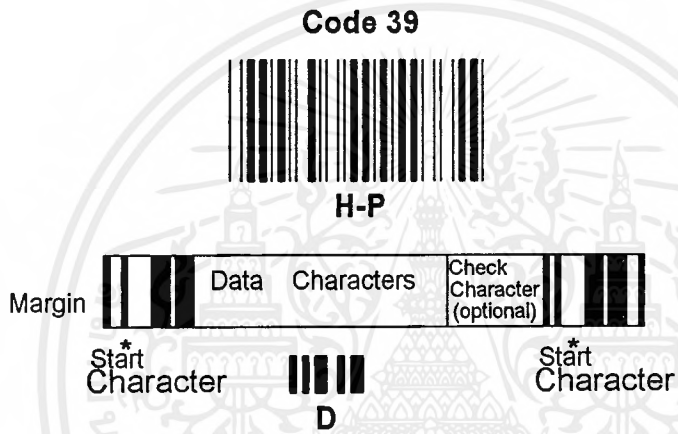
| | |
|---|---|
| รูปแบบตัวอักษรทั้งหมด | อักษรอังกฤษตัวใหญ่ 26 ตัว (A-Z)<br>ตัวเลข 10 ตัว อักษรพิเศษ 7 ตัว<br>สามารถขยายได้เป็น 128 อักษร |
| ความยาวของแถบรหัส | เปลี่ยนแปลงได้ |
| อักษรที่ใช้ตรวจสอบความผิดพลาด | เป็นส่วนที่มีหรือไม่มีก็ได้ |
| อักษร overload | 2 ตัวต่อ 1 แถบรหัส |
| ความหนาแน่น | มากสุดได้ 9.8 หลักต่อ 1 นิ้ว |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**คำอธิบายของบาร์โค้ดชนิด 39**

ทุกๆแถบรหัสของบาร์โค้ดชนิด 39 ประกอบด้วย

1. นำด้วย quite zone
2. รูปแบบตัวอักษรเริ่มต้น
3. อักษรข้อมูล
4. รูปแบบอักษรสิ้นสุด
5. ปิดท้ายด้วย quite zone




รูปที่ 2.3 โครงสร้างของรหัสบาร์โค้ด 39

**การตรวจสอบความถูกต้องของรหัส 39**

อักษรตรวจสอบใส่ช่วยตรวจในความถูกต้องของข้อมูล อักขระที่ใช้ตรวจสอบนี้จะใส่เป็นหลักของข้อมูล หลังจากที่ถูกเข้ารหัสแล้ว โดยค่าของหลักที่ใช้ตรวจนี้ จะคำนวณได้ตามนี้

1. ใช้ตารางที่ 2.1 ให้ค่าแก่อักขระข้อมูลต่างๆ
2. รวมผลของทุกๆข้อมูล
3. หาเศษที่ได้จากผลรวมที่ได้จากข้อ 2 หารด้วย 43
4. นำเศษที่ได้จากข้อ 3 ไปเทียบในตารางที่ 2 ว่าเป็นอักขระอะไร อักษรนั้นจะเป็นอักษรตรวจสอบ

ตัวอย่างเช่นแถบรหัสมีข้อมูล "TEST" ผลรวมจะเป็น 29+14+28+29 = 100 ผลหารของ 100 ด้วย 43 คือ 2 เศษ 14 ฉะนั้นอักษรตรวจสอบคือ "E"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.1** อักษรของรหัส 39 เทียบกับค่าของแต่ละอักษร

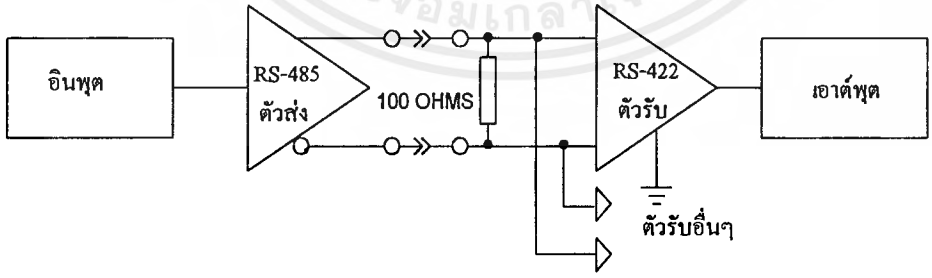
| ASCII | CODE 39 | ACSII | CODE 39 | ASCII | CODE 39 | ASCII | CODE 39 |
|---|---|---|---|---|---|---|---|
| NUL | %U | SP | Space | @ | %V | ` | %w |
| SOH | $A | ! | /A | A | A | a | +A |
| STX | $B | " | /B | B | B | b | +B |
| ETX | $C | # | /C | C | C | c | +C |
| EOT | $D | $ | /D | D | D | d | +D |
| ENQ | $E | % | /E | E | E | e | +E |
| ACK | $F | & | /F | F | F | f | +F |
| BEL | $G | ' | /G | G | G | g | +G |
| BS | $H | ( | /H | H | H | h | +H |
| HT | $I | ) | /I | I | I | I | +I |
| LF | $J | | /J | J | J | J | +J |
| VT | $K | + | /K | K | K | K | +K |
| FF | $L | , | /L | L | L | L | +L |
| CR | $M | - | - | M | M | M | +M |
| SO | $N | . | . | N | N | N | +N |
| SI | $O | / | /O | O | O | O | +O |
| DEL | $P | 0 | O | P | P | P | +P |
| DC1 | $Q | 1 | 1 | Q | Q | Q | +Q |
| DC2 | $R | 2 | 2 | R | R | R | +R |
| DC3 | $S | 3 | 3 | S | S | S | +S |
| DC4 | $T | 4 | 4 | T | T | T | +T |
| NAK | $U | 5 | 5 | U | U | U | +U |
| SYN | $V | 6 | 6 | V | V | V | +V |
| ETB | $W | 7 | 7 | W | W | W | +W |
| CAN | $X | 8 | 8 | X | X | X | +X |
| EM | $Y | 9 | 9 | Y | Y | Y | +Y |
| SUB | $Z | : | /Z | Z | Z | Z | +Z |
| ESC | %A | ; | %F | \| | %K | { | %P |
| FS | %B | < | %G | \ | %L | \| | %R |
| GS | %C | = | %H | ] | %M | } | %S |
| RS | %D | > | %I | ↑ | %N | ~ | %T,%X |
| US | %E | ? | %J | _ | %O | DEL | %Y,%Z |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 การสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-485

การสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-232C นั้นเป็นมาตรฐานการสื่อสารข้อมูลในแบบที่ใช้การสื่อสารข้อมูลระหว่างอุปกรณ์หรือแบบจุดต่อจุด (Point to Point) ส่วน RS-422A นั้นเป็นมาตรฐานที่พัฒนามาจาก RS-232A เพื่อทำให้สามารถทำการสื่อสารข้อมูลได้ในระยะทางที่ไกลขึ้นและอัตราการสื่อสารข้อมูลเพิ่มขึ้นแต่ก็ยังเป็นการสื่อสารข้อมูลจากอุปกรณ์หนึ่งไปยังอุปกรณ์อื่นๆ ได้สูงสุดเพียงไม่เกิน 10 ตัวเท่านั้นและไม่สามารถส่งข้อมูลย้อนกลับจากอุปกรณ์ 10 ตัวนั้นกลับมายังอุปกรณ์หลักได้ หรืออาจกล่าวได้ว่าการสื่อสารข้อมูลตามมาตรฐาน RS-422A นั้นเป็นการสื่อสารข้อมูลแบบซิมเพล็กซ์ คือมีทิศทางของข้อมูลเป็นแบบทางเดียวตลอด ดังนั้นจึงไม่สามารถทำการออกแบบระบบให้เป็นโครงข่ายข้อมูลได้

ด้วยเหตุนี้จึงได้มีการพัฒนามาตรฐานการสื่อสารข้อมูลแบบใหม่ขึ้น เพื่อรองรับความต้องการดังกล่าวนี้ ซึ่งก็คือมาตรฐาน RS-485 ซึ่งเป็นมาตรฐานที่อาศัยหลักการทำงานของสัญญาณแบบดิฟเฟอเรนเชียลเช่นเดียวกับมาตรฐาน RS-422A แต่จะสามารถสื่อข้อมูลได้ทั้ง 2 ทิศทางในสายสัญญาณเพียงคู่เดียว ซึ่งเป็นการสื่อสารข้อมูลแบบฮาล์ฟดูเพล็กซ์ จากผลของการใช้สัญญาณในลักษณะดิฟเฟอเรนเชียลนี้จะทำให้ได้ระยะทาง และความเร็วในการสื่อสารข้อมูลสูงเช่นเดียวกับมาตรฐานการสื่อสารข้อมูล RS-422A แต่มาตรฐาน RS-485 นั้นสามารถที่จะสื่อสารระหว่างอุปกรณ์ทั้งการรับและการส่งได้สูงสุดถึง 32 ตัว หรืออาจกล่าวได้ว่าการสื่อสารข้อมูลมาตรฐาน RS-485 นั้น เป็นการสื่อสารแบบหลายจุด โดยโครงสร้างในการสื่อสารข้อมูลแบบ RS-485 แสดงดังรูปที่ 2.4




**รูปที่ 2.4** โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-485

RS-485 ไม่จำกัดรูปแบบของโปรโตคอล ที่จะนำมาใช้งานกับระบบที่พัฒนาขึ้น โดยขึ้นอยู่กับผู้พัฒนาระบบเองว่าจะเลือกใช้โปรโตคอลแบบไหนมาใช้งาน นอกจากนี้ ตัวรับและตัวส่ง มีราคาไม่สูงมากนักจึงทำให้ RS-485 สามารถนำมาใช้งานอย่างแพร่หลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1 รูปแบบการสื่อสารข้อมูลแบบ RS-485

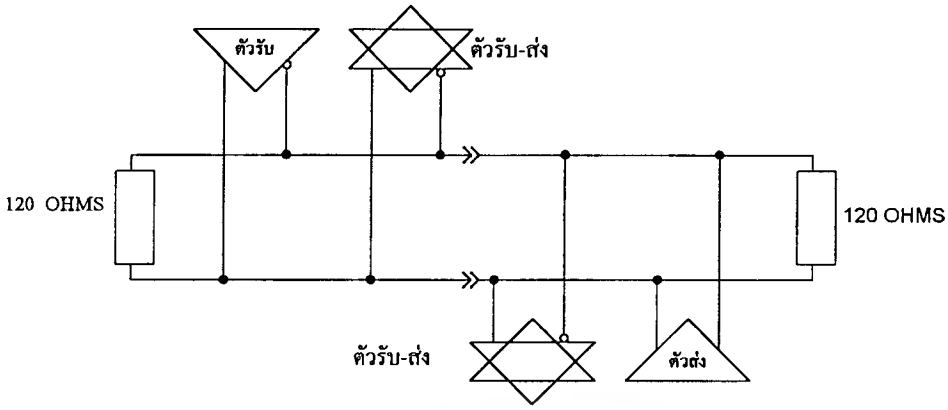
**เครือข่ายแบบที่ใช้สายนำสัญญาณสองเส้น (Two Wire)**

RS-485 แบบสองสายนี้ ถือเป็นการสื่อสารแบบซิมเพล็กซ์คือ ลักษณะของการรับ-ส่งข้อมูลจะเป็นไปในลักษณะผลัดกันรับ-ส่ง โดยการกำหนดว่าจะเป็นการรับหรือส่ง จะถูกกำหนดด้วยตัวแม่ข่าย (MASTER) ข้อดีของเครือข่ายแบบนี้ คือ จะช่วยประหยัดสายส่งสัญญาณที่ใช้ในการวางเครือข่าย

ตารางที่ 2.2 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลทาง EIA

| พารามิเตอร์ | RS-232C | RS-423A | RS-422A | RS-485 |
|---|---|---|---|---|
| โหมดการทำงาน | Single-ended | Single-ended | Differential | Differential |
| จำนวนของตัวรับและตัวส่งที่ยอมรับได้ | 1 ตัวส่ง<br>1 ตัวรับ | 1 ตัวส่ง<br>10 ตัวรับ | 1 ตัวส่ง<br>10 ตัวรับ | 32 ตัวส่ง<br>32 ตัวรับ |
| ความยาวของคู่สายสัญญาณรับส่งข้อมูล | 50 ฟุต | 4,000 ฟุต | 4,000 ฟุต | 4,000 ฟุต |
| อัตราการส่งข้อมูลสูงสุด(บิตต่อวินาที) | 20k | 100k | 10M | 10M |
| แรงดันไฟฟ้าโหมดร่วมสูงสุด | ±2.5V | ±6V | +6V<br>-2.5V | +12V<br>-7V |
| Driver output | ±5V ต่ำสุด<br>± 15V สูงสุด | ±3.6V ต่ำสุด<br>±6.0V สูงสุด | ±2V ต่ำสุด | ±1.5V ต่ำสุด |
| Driver load(W) | 3k ถึง 7k | 450 ต่ำสุด | 100 ต่ำสุด | 60 ต่ำสุด |
| Driver slow rate | 30 V /Aสูงสุด | | NA | NA |
| กระแสลิมิตเมื่อเอาท์พุตลัดวงจร | 500mA ลัดวงจรกับ $V_{CC}$หรือ Gnd | 150 mA ลัดวงจรกับ Gnd | 150 mA ลัดวงจรกับ Gnd | 150 mA ลัดวงจรกับ Gnd 250 mA ลัดวงจรกับ 8 V หรือ 12 V |
| ค่าความต้านทานเอาท์พุตของตัวส่ง(W) | NA-Power On<br>300-Power Off | NA-Power On<br>60k-Power Off | NA-Power On<br>60k-PowerOff | 120 k<br>Power On, Off |
| ค่าความต้านทานอินพุตของตัวส่ง(W) | 3 k ถึง 7k | 4k | 4k | 12k |
| ความไวตัวรับ | ±3V | ±200 Ma | ±200 mA | ±200 mA |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.5 เครือข่ายของ RS-485 แบบที่ใช้สายนำสัญญาณสองเส้น (Two Wire)

**เครือข่ายแบบที่ใช้สายนำสัญญาณสี่เส้น (Four Wire)**

การสื่อสารข้อมูล RS-485 แบบนี้ ถือเป็นการสื่อสารแบบฟลูดูเพล็กซ์ คือลักษณะของการรับ-ส่งข้อมูลสามารถทำการรับและส่งข้อมูลได้ในเวลาเดียวกัน เนื่องจากว่ามีบัสข้อมูลอยู่เป็นจำนวน 4 เส้น ข้อดีของเครือข่ายแบบนี้คือ ความเร็วในการรับ-ส่งข้อมูลจะเร็วกว่าแบบที่ใช้สายนำสัญญาณสองเส้น แต่ก็มีข้อเสียคือ จะสิ้นเปลืองค่าใช้จ่ายในส่วนของสายนำสัญญาณมากกว่าแบบที่ใช้สายนำสัญญาณสองเส้น 2 ชุด โดยที่โหลด 1 ชุด จะประกอบด้วยตัวส่ง 1 ตัว และตัวรับ 1 ตัว และค่าของความต้านทานที่ต่อคร่อมระหว่างสายสัญญาณมีค่า 60 โอห์ม

- เอาต์พุตของตัวส่งในสภาวะออฟ มีกระแสรั่วไม่เกิน 100 mA ในช่วงแรงดันไฟฟ้าระหว่าง -7 ถึง 12V
- เอาต์พุตของตัวส่งให้แรงดันไฟฟ้าเอาต์พุต 1.5 ถึง 5V ในช่วงแรงดันไฟฟ้าระหว่าง -7 ถึง 12V
- ตัวส่งมีวงจรป้องกันตัวเองที่ส่วนเอาต์พุต ในกรณีที่ตัวส่งหลายๆ ตัว ส่งข้อมูลออกมาพร้อมๆกัน

**คุณสมบัติเฉพาะของตัวรับ RS-485**

- ค่าความต้านทานอินพุตมีค่าสูง โดยมีค่าไม่น้อยกว่า 12 กิโลโอห์ม
- ตัวรับ มีค่าแรงดันไฟฟ้าอินพุตระหว่าง -7 ถึง 12V
- ตัวรับ สามารถตอบสนองต่อสัญญาณที่แตกต่างจากสัญญาณโหมดร่วมได้

**คุณสมบัติเฉพาะของคู่สายสัญญาณ RS-485**

- คู่สายสัญญาณที่ใช้สำหรับการรับ-ส่งสัญญาณในระบบ RS-485 นั้น ควรพันสลับเป็นเกลียวเพื่อการลดทอนสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ความหมายของยูนิตโหลด (Unit Load)**

ยูนิตโหลด เป็นจำนวนที่มากที่สุดของตัวรับ และตัวส่งที่สามารถใช้งานบนคู่สายสัญญาณรับ-ส่งคู่หนึ่ง โดยจะขึ้นกับค่า Unit Load (U.L.) ซึ่ง RS-485 ยอมรับได้ที่ 32 Unit Load ต่อคู่สายสัญญาณ 1 คู่

นิยามของยูนิตโหลด เป็นโหลดที่ใช้กระแส 1 mA ที่แรงดันไฟฟ้าโหมดร่วม 12 V ซึ่งโหลดนี้ประกอบด้วยตัวส่ง และ/หรือ ตัวรับ แต่ไม่รวมค่าความต้านทานที่เกิดจากความต้านทานที่ต่อคร่อมคู่สายสัญญาณรับ-ส่ง

- เป็นไปตามมาตรฐาน RS-485, RS-485, CCITT V.11 และ X.27
- เอาท์พุตของตัวส่งเป็นแบบ 3 สเตท(State) ยกเว้น SN 75176B
- เอาท์พุตตัวส่งสามารถขับกระแสได้สูงสุด 60 mA
- ค่าความต้านทานอินพุตของตัวรับมีค่าประมาณ 20k (น้อยที่สุด)
- ตัวรับมีค่าความไวอินพุต (input sensitivity) ประมาณ $\pm$200 mV
- ใช้ไฟเลี้ยง 5V

**การคำนวณหาจำนวนคู่ตัวรับ-ส่ง**

ตัวส่ง

- ค่าของกระแสรั่วเมื่ออยู่ในสภาวะออฟที่ 12V มีค่าไม่เกิน 0.1 mA
- ตัวส่งมีค่า (U.L.) = 0.1 mA/1.0 mA = U.L.

ตัวรับ

- ค่าของกระแสอินพุตที่แรงดันไฟฟ้าอินพุต 12V มีค่าไม่เกิน 0.6 mA
- ตัวรับมีค่า (U.L.) = 0.6 mA/1.0 mA = 0.6 mA
- เมื่อพิจารณาโหลด 1 (ตัวรับและตัวส่ง) มีค่า = (0.1+0.6)mA/1.0 mA
  = 0.7 U.L.
- สายสัญญาณรับ-ส่งข้อมูลคู่หนึ่งสามารถรองรับตัวรับ-ส่งได้ = 32/0.7 = 45 คู่

### 2.3.2 RS-485 Protocol

RS-485 เป็นเงื่อนไขของการรับและส่งข้อมูลที่ได้พัฒนาขึ้นมาเพื่อใช้กับระบบการสื่อสารข้อมูลแบบ RS-485 โดยได้ประยุกต์มาจาก Protocol. BSC Protocol, HDLC Protocol และ Xebec Protocol เพื่อสร้างชุดของข้อความในการรับ-ส่งข้อมูล และการประยุกต์ มีรายละเอียดดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เฟรมของข้อมูลประกอบด้วย Header Portion และ Data Portion

| SYN | SYN | SYN | HEADER | DATA PORTION | SYN | SYN | SYN |
|---|---|---|---|---|---|---|---|

**Header Portion**

ในส่วนของ Header Portion ของ RS-485 Protocol ได้ประยุกต์มาจากบางส่วนของ

DDCMP (Digital Data Communication Message Protocol)

HDLC (High Level Data Link Control)

Xebec Protocol โดยมีรายละเอียดดังต่อไปนี้คือ

| SOH | TYD | EXC | CDAT | FSEQTF | Fserr | ADDRESS | LRC # 1 |
|---|---|---|---|---|---|---|---|
| 8 bit | 2 bit | 4 bit | 10 bit | 8 bit | 8 bit | 24 bit | 8 bit |

**SOH : Start of Header (8 bit)**

เป็นส่วนแสดงการเริ่มต้นของข้อความของ Header Portion มีขนาด 8 บิต

**TYD : Type of Data (2 bit)**

เป็นส่วนที่ใช้แสดงถึงชนิดของข้อมูลที่บรรจุอยู่ในส่วนของ Data Portion

| TPD | ลักษณะของข้อความ |
|---|---|
| 00 | ข้อความเป็นข้อมูล |
| 01 | ข้อความเป็นคำสั่ง |
| 10 | ข้อความเป็นคำสั่งและพารามิเตอร์ |
| 11 | ข้อความเป็นข้อความตอบรับจากตัวตอบรับ |

**CDAT : Count of Data (10 bit)**

เป็นส่วนแสดงจำนวนของข้อมูลที่บรรจุอยู่ใน Data Field ของ Data Portion โดยนับข้อมูลแบบไบนารีได้สูงสุด 1 Kbyte/1 เฟรม (สามารถขยายได้ 8 Kbyte/1เฟรม)

**EXC : Expansion of Count (4 bit)**

เป็นส่วนแสดงบิตขยายของ CDAT มีขนาด 4 บิต

บิตแรก = 0 แสดงว่า CDAT มีขนาด 10 บิต

= 1 แสดงว่า CDAT มีขนาด 13 บิต (เพิ่มจาก 3 บิตของEXC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**FSEQTE : File Sequence Transfer (8 bit)**

0 : ส่งข้อความทีละเฟรม ไม่ต้องพิจารณาอีก บิตของ FSEQTE และ Fserr

1 : ส่งข้อความแบบทีละหลายๆเฟรม พิจารณาอีก 7 บิตของ FSEQTE และ Fserr

**NFs : Number of Frame Sequence Begin to Send (3 bit)**

แสดงลำดับเฟรมข้อความที่จะส่ง (0-7) มีขนาด 3 บิต

**NFr : Number of The Next Frame be Recive (3 bit)**

แสดงลำดับเฟรมข้อความที่จะรับ (0-7) มีขนาด 3 บิต

**P/F : Poll/Final bit (1 bit)**

กรณีเซนเตอร์ส่งข้อความให้เทอร์มินอล บิท P/F เป็น P : Poll

P = 1 ยินยอมให้เทอร์มินอลตอบรับได้

P = 0 ยังไม่ยินยอมให้เทอร์มินอลสามารถตอบรับได้

กรณีเซนเตอร์ส่งข้อความให้เทอร์มินอล บิท P/F เป็น F : finall

F = 1 ยินยอมให้เซนเตอร์สามารถตอบรับได้

F = 0 ยังไม่ยินยอมให้เซนเตอร์ตอบรับได้

**Fserr : Frame Sequence error**

แสดงข้อความผิดพลาด มีขนาด 8 บิต Fserr ประกอบด้วย

| F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 |
|---|---|---|---|---|---|---|---|
| 1 bit | 1 bit | 1 bit | 1 bit | 1 bit | 1 bit | 1 bit | 1 bit |

ตัวอย่าง เช่น ถ้าเราส่งข้อมูล 5 เฟรม (0.4) ผล เฟรม 1 และเฟรม 3 เกิดผิดพลาด

Fserr = 00001010b

**ADDRESS : Adress of bode (24bit)**

เป็นส่วนแสดงค่าตำแหน่งของตัวรับที่ตัวส่งต้องการติดต่อด้วย มีขนาด 24 บิต

เมื่อตัวรับเป็นตัวเซนเตอร์

ADDRESS = CXX : C = 43h (ASCII)

: XX = 01-32 (ASCII 2 BYTE)

เมื่อตัวรับเป็นเทอร์มินอล

TERMINAL = TXX : T = 54h (ASCII)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**LRC # 1 : Longitudinal redundacy Check**

เป็นส่วนแสดงความผิดพลาดและเป็นรหัสเช็คความผิดพลาดของ Header มีขนาด 8 บิต

**DATA PORTION**

ในส่วนของ Data Portion ของ RS-485 Protocol ได้ประยุกต์มาจากบางส่วนของ

HDLC Protocol : High-Level Data Link Control Protocol

BSC Protocol : Binary Synchronous Communication Protocol

Xcbee Protocol

โดยมีรายละเอียดของ Data Portion ดังนี้

| STX | DATA FIELD | ETX | LRC # 2 |
|---|---|---|---|
| 8 bit | 1-8 Kbyte | 8 bit | 8 bit |

**STX : Start Text (8bit)**

เป็นส่วนแสดงการเริ่มต้นของข้อความส่วน Data Portion มีขนาด 8 บิต

**ETX : End of Text (8 bit)**

เป็นส่วนแสดงการสิ้นสุดของข้อความส่วน Data Portion มีขนาด 8 บิต

**LRC # 2 : Longitudinal Redundacy Check (8bit)**

เป็นส่วนแสดงรหัสตรวจสอบของ Data Portion มีขนาด 8 บิต

**DATA FIELD (1-8 Kbyte)**

เป็นส่วนแสดงข้อมูลที่ใช้ในการติดต่อ มีขนาดไม่เกิน 1024 ตัวอักษรต่อเฟรมข้อความ (ขยายได้ 8192 ตัวอักษร) โดยมีรูปแบบดังนี้

| COMMAND | SEP | SDB | DATA | EDB |
|---|---|---|---|---|
| 8 bit | 8 bit | 8 bit | ASCII CODE | 8 bit |

**COMMAND**

คือชุดคำสั่งของ RS-485 Protocol

**SEP : Seperator (8 bit)**

เป็นตัวกั้นระหว่างคำสั่งกับข้อมูลหรือพารามิเตอร์

**SDB : Start of Data Block (8 bit)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นส่วนแสดงการเริ่มต้นของข้อมูลหรือพารามิเตอร์

**EDB : End of Data Block (8 bit)**

เป็นส่วนแสดงการสิ้นสุดของข้อมูลหรือพารามิเตอร์

**DATA : Data or Parameter (ASCII)**

เป็นข้อมูล หรือ พารามิเตอร์ รูปแบบเป็นรหัส ASCII

## 2.4 ตัวแสดงผล Lcd แบบจุด

สามารถแบ่งตัวแสดงผล Lcd แบบจุด ออกได้เป็นดังนี้

1. Character Lcd Module
2. Graphic Lcd Module
3. Segment Display Type Lcd Module

โดยในแต่ละแบบนี้ก็จะมีส่วนประกอบใหญ่ ๆ แบ่งได้เป็น

**1. Dot Matrix Lcd** เป็นตัวแสดงผลในลักษณะการปิดและเปิดตัวเองกับแสงคือส่วนของที่เป็นตัวกระจกบรรจุผลึก

**2. Driver** เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก Lcd อีกทีหนึ่งโดยมีเบอร์ที่นิยมใช้เช่น HD44100H, MSM5259

**3. Controller** เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมา และจัดการควบคุม Lcd Module ให้ทำงานแสดงผลต่าง ๆ เช่น การลบจอภาพ การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ไอซีที่นิยมใช้กันคือ HD4478 ซึ่งจะใช้แบบแสดงตัวอักษรเป็นส่วนใหญ่ และเบอร์ HD61830 จะใช้ในแบบแสดงภาพกราฟฟิกส์

ในการทำงานและใช้งาน Lcd นั้น ในแต่ละบริษัทจะใช้ตัวควบคุม ที่มีหลักการทำงานเหมือน ๆ กันเป็นส่วนใหญ่ IC ที่นิยมมากที่สุดตัวหนึ่งที่เป็นตัวควบคุม Lcd ก็คือเบอร์ HD44780 โดยรูปแบบการทำงานของมันได้เป็นมาตรฐานให้กับตัวควบคุม Lcd ตัวอื่น ๆ ด้วย

**HD44780** เป็นไอซี LSI ตัวหนึ่งใช้ควบคุม Lcd โดยแสดงผลในรูปตัวอักษรหรือสัญลักษณ์ต่าง ๆ ตัวมันสามารถต่อใช้งานแบบ 4 บิต หรือ 8 บิต ก็ได้ โดยถ้าเราต่อแบบ 4 บิต จะต่อใช้งานที่ DB7 – DB4 เท่านั้น โดยข้อมูลครั้งแรกที่ส่งนั้น HD44780 จะถือเป็นข้อมูล 4 บิตบน และข้อมูลที่ส่งต่อมานั้นเป็นข้อมูล 4 บิ๊ตล่าง

เมื่อป้อนให้ HD44780 ทำงานก็จะทำการรีเซ็ตตัวมันเองโดยจะใช้เวลาประมาณ 10 ms หลังจากไฟ $V_{DD}$ 4.5 Volt แล้ว โดยจะเซ็ต ตัวเองดังนี้

1. ทำการลบข้อมูลจอภาพ Lcd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



2. เลือกเซ็ต ค่าภายใน

DL = 1 : เป็นการเซ็ตให้การติดต่อแบบ 8 บิต

N = 0 : เซ็ต เป็น 1 บรรทัดการแสดงผล

F = 0 : 5 x 7 Dot ต่อหนึ่งต่ออักษร

3. เปิด/ปิด การแสดงผล

D = 0 : Display off

C = 0 : Curror off

B = 0 : Blink off

4. เข้าสู่ Mode Set I/D = 1; +1 ( เพิ่มค่า Counter ขึ้น 1 )

S = 0 : No Shift

**รายละเอียดของคำสั่ง HD44780**

**1. ลบหน้าจอภาพ**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

คำสั่งนี้จะเป็นการเขียนช่องว่าง หรือ Space ( ASCII 20H ) เข้าไปใน DD Ram ทั้งหมด และทำการ Set DD Ram Addresser เป็นศูนย์ ตัว Curror จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ ภาพ Set I/D = 1 , S ไม่มีการเปลี่ยนแปลง

**2. กลับไปจุดเริ่มต้น**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | . |

คำสั่งนี้จะทำการ Set DD Ram Address เป็นศูนย์ ตัว Cursor จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ ข้อมูลในภาพไม่เปลี่ยนแปลง

**3. เซ็ตการเข้าสู่โหมด**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I/D | S |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32815

Bit I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้ว จะทำให้ DD Ram Address เพิ่มขึ้นหนึ่งหรือลดลงหนึ่งโดย

1 = เพิ่ม

0 = ลดลงหนึ่ง

BIT S : เป็นกำหนดแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูล แล้วตัว Cursor อยู่กับที่ ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่ที่ตัว Cursor จะถูกดันไปทางขวามือ

**4. ควบคุมการเปิด/ปิดหน้าจอภาพ**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B |

Bit D : เป็นบิตให้เปิดปิดหน้าจอภาพ โดย

D = 1 จะ on และ

D = 0 จะ off

Bit C : จะให้แสดง Cursor ให้บิต C = 1 และถ้าไม่ต้องการแสดง Cursor bit C = 0 โดยตัว Cursor จะอยู่ที่ line ที่ 8 ในแบบ 5x7 dot และจะอยู่ line ที่ 11 ในแบบ 5x10 dot

Bit B : เป็นบิต Set การกระพริบของ Cursor โดย B = 1 การกระพริบ B = 0 ไม่มีการกระพริบ โดยมีระยะเวลาการกระพริบประมาณ 379.2 ms

**5. กำหนดตำแหน่งของ Cursor**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | . | . |

เป็นคำสั่งกำหนดให้ตำแหน่ง Cursor หรือข้อมูลไปเกิดทางซ้ายหรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่านโดย

**6. Function Set**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | DL | N | F | . | . |

Bit DL : เป็นการเซ็ตการติดต่อว่าจะให้เป็นแบบ 8 บิตหรือ 4 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยถ้าต้องการติดต่อ 4 บิต DL = 0 และ 8 บิต DL = 1

N : เป็นการเซ็ตบรรทัดการแสดงผล N = 0 แสดง 1 บรรทัด

N = 1 แสดง 2 บรรทัด ในกรณีที่มากกว่า 2 บรรทัด ก็ให้ Set N = 1

F : เป็นการเซ็ตขนาด Dot การแสดงผล 5x7 หรือ 5x10 โดย

F = 0 เป็นแบบ 5x7 และ F = 1 เป็นแบบ 5x10

| N F | No. of display Line | Charcter font | Duty factor | Remarks |
|---|---|---|---|---|
| 0 0 | 1 | 5 x 7 dots | 1/8 | |
| 0 1 | 1 | 5 x 10 dots | 1/11 | |
| 1 * | 2 | 5 x 7 dots | 1/16 | Cannot display 2 line with 5 x 10 dot caracter font |

**7. เซ็ตตำแหน่งของ CG Ram**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | A | A | A | A | A | A |

← Higher Order bit Lower Order bit →

HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือหน่วยความจำข้อมูลแสดงผล ( DD Ram ) จำนวน 80x8 บิต และ หน่วยความจำรหัสอักษร ( CG Ram ) จำนวน 512 บิต และ 7200 บิต คำสั่งนี้จะเป็นตัวเซ็ตตำแหน่งใน CG Ram โดยต้องทำการเซ็ตตำแหน่งก่อนเขียนหรืออ่านข้อมูลจาก CG Ram ด้วย

**8. เซ็ตตำแหน่งของ DD Ram**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | A | A | A | A | A | A | A |

← Higher Order bit Lower Order bit →

เป็นคำสั่งเซ็ตค่าตำแหน่งใน DD Ram ในการเขียนหรืออ่านค่าจาก DD Ram ( DD Ram คือ ส่วนที่จะแสดงผลหน้าจอ Lcd ) โดยจำนวนตำแหน่ง ที่จะเกิดขึ้นบนจอ Lcd จะขึ้นอยู่กับ การเซ็ตค่า N ด้วย

N = 0 ( 1 บรรทัด ) ตำแหน่งจะอยู่ที่ 00H – 4FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N = 1 ( 2 บรรทัด ) ตำแหน่ง จะอยู่ 00H – 27H สำหรับบรรทัดที่ 1 และ 40H – 67H สำหรับบรรทัดที่ 2

แบบการจัดตำแหน่งของ DD Ram หน้าจอ Lcd แบบ 16 ตัวอักษร 2 บรรทัด

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F |

**9. อ่าน Busy Flag และตำแหน่ง**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | BF | A | A | A | A | A | A | A |

← Higher Order bit  Lower Order bit →

เป็นคำสั่งอ่านค่า Busy Flag ซึ่งจะเป็นตัวบอกว่าตัว HD44780 นี้อยู่ในขบวนการทำงานภายในอยู่หรืออยู่ในสภาพพร้อมรับข้อมูล โดย

BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูลตำแหน่งของ CG Ram หรือ DD Ram ด้วย

**10. เขียนข้อมูลลงใน CG หรือ DD Ram**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | D | D | D | D | D | D | D | D |

← Higher Order bit  Lower Order bit →

เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD Ram โดยเมื่อเขียนข้อมูลและ Address จะเพิ่มหรือลดโดยอัตโนมัติตามคำสั่งที่เซ็ตในการเข้าสู่โหมดข้อกำหนดที่จะรู้ว่าเป็นการเขียนข้อมูลของ CG Ram หรือ DD Ram ทำโดยการเซ็ตตำแหน่งของ CG Ram ขึ้นมาก่อนจะเขียนข้อมูล

**11. อ่านข้อมูลจาก CG หรือ DD Ram**

| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | D | D | D | D | D | D | D | D |

← Higher Order bit  Lower Order bit →

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่งอ่านค่าข้อมูลจาก CG Ram หรือ DD Ram โดยก่อนอ่านค่าจาก DD Ram นี้ควรจะใช้คำสั่งเซ็ตตำแหน่งก่อนเพื่อให้รู้ว่าข้อมูลที่อ่านได้นั้นเป็น DD หรือ CG Ram จากตารางการงานจะเห็นว่าการใช้งาน Lcd นั้นง่ายเพียงแต่เราส่งคำสั่งเริ่มแรกและเซ็ตความต้องการในขนาดตัวอักษร, Cursor หลังจากนั้นเราก็สามารถเขียนตัวอักษรเข้าไปใน DD Ram ก็จะเกิดอักษรในจอภาพ Lcd เรายังสามารถกำหนดตำแหน่งตัวอักษรที่จะให้เกิดบนจอได้ โดยการเซ็ตตำแหน่ง DD Ram ตามตารางที่ให้มาในหัวข้อเซ็ตตำแหน่ง DD Ram

**การเขียนข้อมูล CG Ram**

เราสามารถเขียนข้อมูลได้โดยกำหนดตำแหน่งของ CG Ram ก่อนโดยเขียนได้ 64 ตำแหน่ง Bit 5-Bit 0 และเมื่อกำหนดตำแหน่งแล้วก็จะทำการเขียนข้อมูลลงใน CG Ram โดยเป็นลักษณะ Bit ต่อ Bit บนจอ 1 ตัวอักษร คือ 5X7 Dot นั้นจะใช้ข้อมูล 8 Byte และเมื่อเขียนข้อมูลลงใน CG Ram แล้วเวลาเราจะใช้งานก็ให้เขียนข้อมูลลงใน DD RAM คือ ข้อมูลในตำแหน่งตารางรหัสตัวอักษร ที่ตำแหน่ง 00H-07H ขาต่างๆในการต่อใช้งาน ๆ HD44780

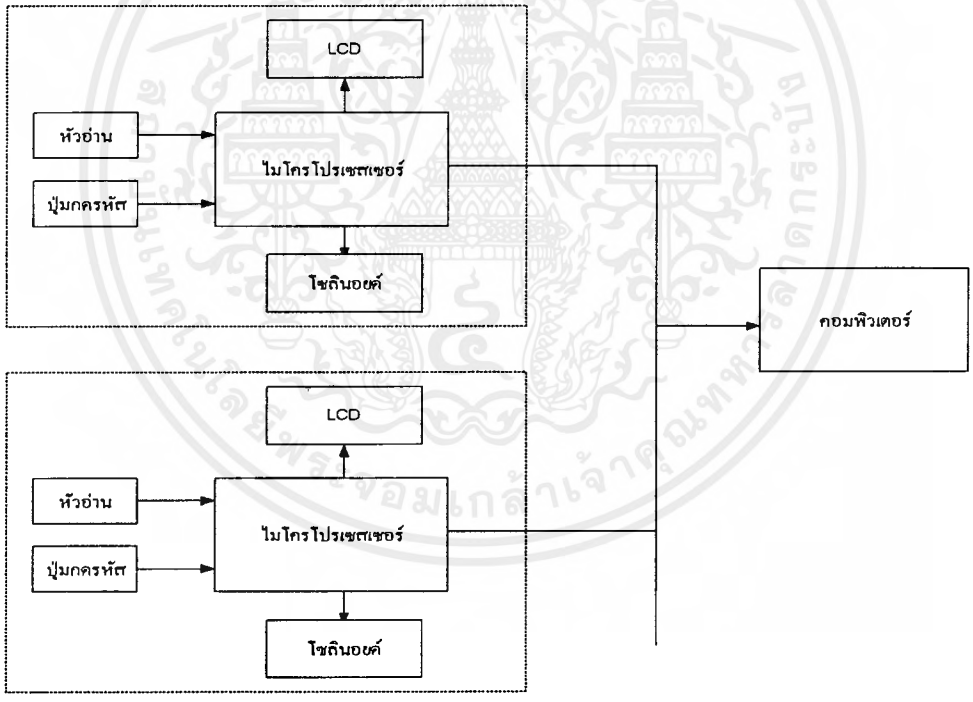
1. RS จะเป็นขาเลือก Registor ภายในซึ่งมีอยู่ 2 ตัว คือ
Instruction Registor ( IR ) และ Data Registor ( DR )
โดยถ้าเป็น 1 จะเป็นการเลือก Data และถ้าเป็น 0 จะเป็นการเลือกคำสั่ง
2. R/W ( Read/Write ) เป็นตัวเลือกว่าจะเขียนหรืออ่านข้อมูลจากตัว IC โดย
อ่านข้อมูล =1 ,เขียนข้อมูล = 0
3. E ( Enable Signal ) เป็นขากำหนดสภาพการรับเขียนอ่านข้อมูล
4. DB0-DB7 เป็นขารับส่งข้อมูลจากตัว IC
5. VDD ไฟเลี้ยงตัววงจร 5 V
6. VSS เป็นขากราวด์
7. VO เป็นขารับแรงดันในการขับ Lcd ให้สว่างหรือมืด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การออกแบบ การสร้างและการทำงาน

## 3.1 กล่าวนำ

การออกแบบระบบตรวจสอบบุคคลเข้าหน่วยงานประกอบด้วยสองส่วนคือ ส่วนประมวลผล และส่วนของวงจร ซึ่งส่วนประมวลผลได้แก่ไมโครคอมพิวเตอร์ และไมโครโปรเซสเซอร์ ส่วนของวงจร ได้แก่ ส่วนของวงจรรับข้อมูลจากบัตรรหัสบาร์โค้ด และวงจรรับข้อมูลจากคีย์เมทริกซ์สวิตช์ ,วงจรควบคุมอุปกรณ์โซลินอยด์,ลำโพง,บัซเซอร์,อุปกรณ์แสดงผล และส่วนสุดท้ายคือส่วนของวงจรสื่อสารข้อมูล โดยการทำงานทั้งหมด ได้แสดงโครงสร้างการทำงานของระบบตรวจสอบบุคคลเข้าหน่วยงาน ในรูปที่ 3.1 ซึ่งมีการทำงานดังนี้




**รูปที่ 3.1** โครงสร้างการทำงานของระบบตรวจสอบบุคคลเข้าหน่วยงาน

ส่วนของวงจรรับข้อมูลหัวอ่านรหัสบาร์โค้ด และส่วนของวงจรรับข้อมูลจากคีย์เมทริกซ์สวิตช์ จะต่ออยู่กับไมโครโปรเซสเซอร์ โดยมีไมโครคอมพิวเตอร์เป็นศูนย์กลาง โดยส่วนรับข้อมูลหัวอ่านบัตรรหัสบาร์โค้ด จะรับรหัสข้อมูลจากไอซี 89C51 เมื่อผู้ที่ต้องการผ่านเข้า ทำการรูดบัตรผ่านชุดหัวอ่านบัตร และส่วนรับข้อมูลจากคีย์เมทริกซ์สวิตช์จะรับรหัสข้อมูล เมื่อมีผู้ต้องการที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้ากดรหัสประจำตัวเป็นเลขจำนวน 8 ตัวบนปุ่มกดรหัส จากนั้นไอซี 89C51 จะรับรหัสข้อมูลมาจากชุดหัวอ่าน หรือจากปุ่มกดรหัส เพื่อทำการประมวลผลรหัสข้อมูลที่รับเข้ามา แล้วส่งข้อมูลไปยังไมโครคอมพิวเตอร์เพื่อนำไปประมวลผล โดยนำรหัสที่ได้มาตรวจสอบเปรียบเทียบกับแฟ้มข้อมูล โปรแกรมจะแสดงรายละเอียดต่างๆให้ผู้ควบคุมทราบ และจะส่งสัญญาณไปควบคุมโซลินอยด์ เพื่อเปิดสลักประตูให้เข้าและส่งรหัสสัญญาณข้อมูลไปยังไมโครคอมพิวเตอร์ โดยผ่านพอร์ตสื่อสารแบบอนุกรม RS–485 ได้แบ่งการสื่อสารข้อมูลออกเป็น 3 ขั้นตอน ขั้นตอนแรก คือ การสื่อสารระหว่างบัตรรหัสบาร์โค้ด หรือการกดรหัสด้วยปุ่มกดรหัสกับไมโครโปรเซสเซอร์ ขั้นตอนที่สอง คือ การสื่อสารระหว่างไมโครโปรเซสเซอร์กับไมโครคอมพิวเตอร์ และขั้นตอนที่สามเป็นการสื่อสารระหว่างไมโครโปรเซสเซอร์กับชุดควบคุมอุปกรณ์โซลินอยด์ ซึ่งในบทนี้จะกล่าวถึงการออกแบบ, การสร้าง และการทำงานของระบบ โดยแยกเป็นสองส่วนใหญ่ๆ คือ

1. ส่วนวงจร
2. ส่วนโปรแกรม

## 3.2 การออกแบบวงจร

การออกแบบวงจรระบบตรวจสอบบุคคลเข้าหน่วยงาน จะแยกเป็นสามส่วนคือ ส่วนของวงจรรับข้อมูล,ส่วนของวงจรควบคุมอุปกรณ์ โซลินอยด์,ลำโพง และส่วนของวงจรสื่อสารข้อมูลซึ่งมีการทำงานดังนี้

### 3.2.1 ส่วนของวงจรรับข้อมูล

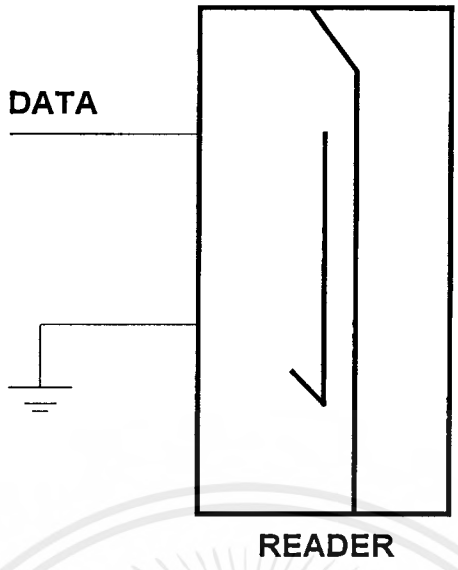
ในส่วนของวงจรรับข้อมูล ทำหน้าที่ในการรับข้อมูลเพื่อทำการส่งข้อมูลให้แก่ ไมโครโปรเซสเซอร์ เพื่อนำไปประมวลผล ซึ่งสามารถแบ่งได้เป็น สองส่วน คือ

1. ส่วนของวงจรรับข้อมูลจากหัวอ่านบัตรรหัสบาร์โค้ด
2. ส่วนของวงจรรับข้อมูลจากคีย์เมทริกซ์สวิตช์

**ส่วนรับข้อมูลจรกหัวอ่านบัตรรหัสบาร์โค้ด**

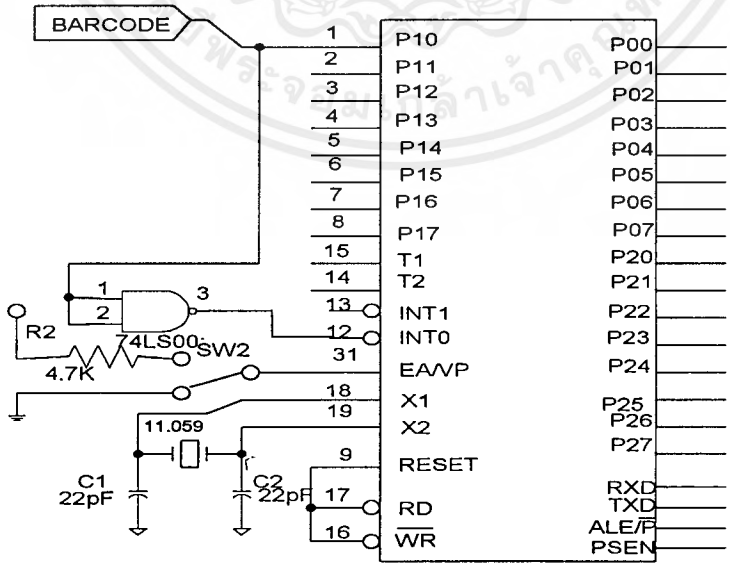
ในส่วนของวงจรรับข้อมูลจากหัวอ่านบัตรรหัสบาร์โค้ด ทำหน้าที่ในการรับข้อมูลจากการรูดบัตรรหัสบาร์โค้ด โดยผ่านชุดหัวอ่านบัตรรหัสบาร์โค้ด จากนั้นทำการส่งต่อข้อมูลให้แก่ พอร์ตอินพุตของไอซี 89C51 เพื่อนำไปประมวลผล แสดงในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.2 ชุดหัวอ่านบัตรรหัสบาร์โค้ด

เมื่อมีผู้ต้องการที่จะเข้าประตู ทำได้โดยการนำบัตรรหัสบาร์โค้ด ซึ่งถูกบันทึกด้วยตัวเลขรหัสประจำตัวของผู้ถือบัตรรหัสบาร์โค้ดจำนวน 8 ตัว โดยมีสัญลักษณ์ " * " บิดท้ายของข้อความชุดตัวเลขใน 8 ตัวและในแถบรหัสจะมีแถบนำซึ่งจะอยู่หน้า " * " ซึ่งทำหน้าที่บอกรูปแบบของการวางข้อมูลว่าส่วนไหนเป็นข้อมูลในการเริ่มต้น เมื่อโดยรูดบัตรรหัสบาร์โค้ดผ่านชุดหัวอ่าน รหัสข้อมูลจากบัตรจะถูกส่งรหัสข้อมูลออกเส้นทางข้อมูล (Data Bus)



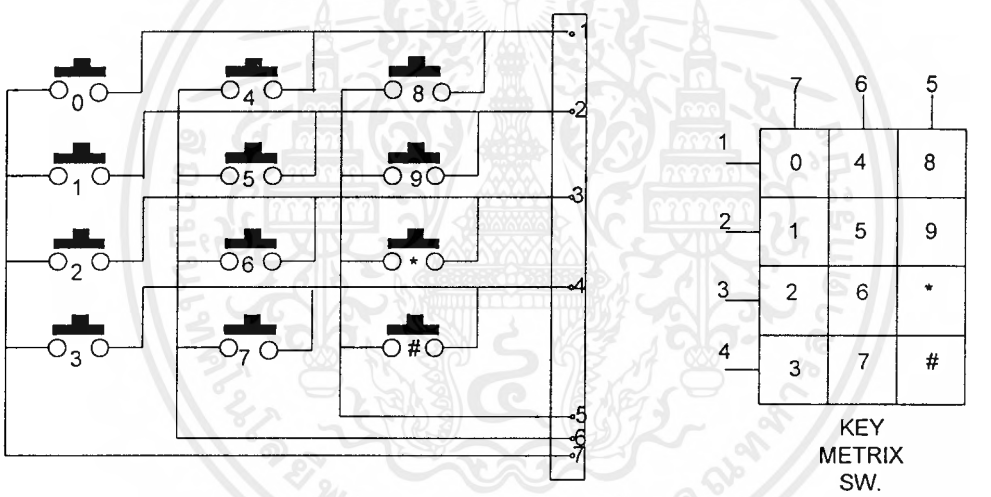

รูปที่ 3.3 วงจรส่วนรับข้อมูลจากหัวอ่านบัตรรหัสบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณรหัสข้อมูล (Data) ที่ออกมาจากเอาต์พุตของชุดหัวอ่านบัตรรหัสบาร์โค้ด จะถูกส่งไปยังพอร์ตอินพุตของไอซี 89C51 เข้าที่อินพุต P 1.0 และจะทำการตรวจสอบการทำงานจากโปรแกรมต่อ เพื่อทำการประมวลผล โดยทำการต่อ P1.0 เข้ากับ INT 0 จากนั้นก็จะเข้าสู่การตรวจเช็คข้อมูลและการอ่านระหัสบาโค้ดต่อไป

**ส่วนของวงจรรับข้อมูลคีย์เมทริกซ์สวิตช์**

ในส่วนของวงจรรับข้อมูลคีย์เมทริกซ์สวิตช์ จะทำหน้าที่ในการรับสัญญาณข้อมูลจากการกดรหัสประจำตัวของผู้ที่จะผ่านเข้าบนคีย์เมทริกซ์สวิตช์ โดยสัญญาณข้อมูลที่รับเข้ามานั้น จะเปลี่ยนเป็นสัญญาณรหัสฐานสอง 4 บิต เพื่อส่งไปยังพอร์ต 0 ของไอซี 89C51 เพื่อนำไปประมวลผล

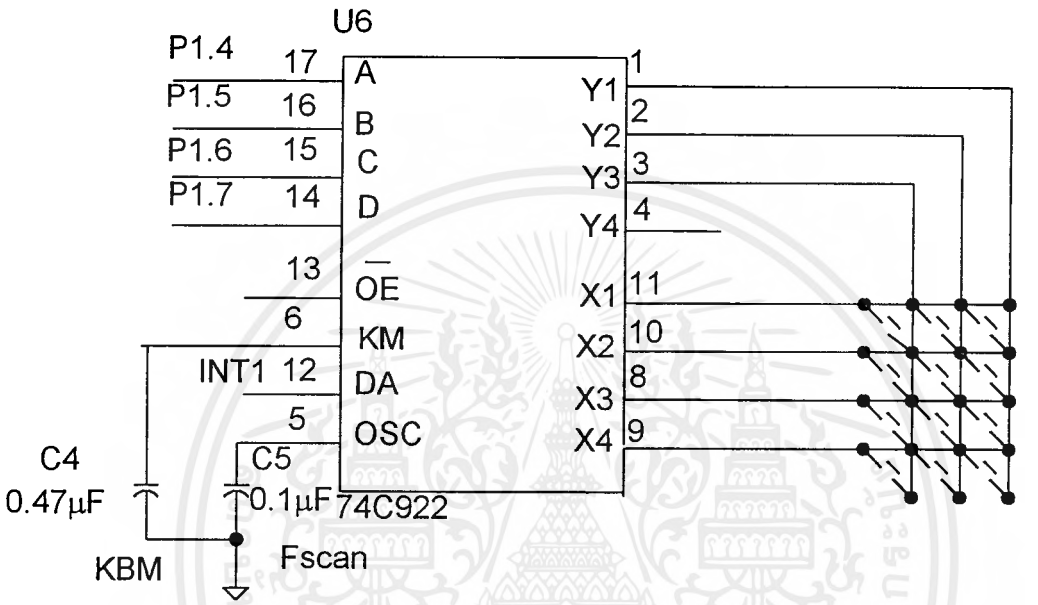



**รูปที่ 3.4** คีย์เมทริกซ์สวิตช์

คีย์เมทริกซ์สวิตช์มี 12 ปุ่มกด ประกอบด้วยตัวเลข 0-9 และ เครื่องหมาย 2 เครื่องหมาย คือ * และ # โดยแบ่งการเดินสายออกเป็นทางแนวนอน และแนวตั้งของแต่ละปุ่มกด ดังรูปที่ 3.4 คีย์เมทริกซ์สวิตช์จะให้ผู้ใช้กดรหัสประจำตัว 8 ตัว โยการกดแต่ละครั้งจะได้เอาต์พุตทางแนวตั้งและแนวนอนอย่างละหนึ่งค่า โดยที่เอาต์พุตจะให้รหัสที่ไม่ซ้ำกัน 12 ค่า เมื่อผู้ใช้กดสวิตช์ตัวใดตัวหนึ่งบนปุ่มกดรหัส รหัสข้อมูลจะถูกส่งไปให้อินพุตของไอซี MM74C922 ซึ่งเป็นอุปกรณ์ที่ทำหน้าที่เปลี่ยนรหัสสัญญาณจากการกดปุ่มรหัส ซึ่งเป็นรหัสฐานสอง 4 บิตซึ่งขา 5 และ ขา 6 จะต่ออยู่กับตัวเก็บประจุ และผ่านลงกราวด์ เพื่อกำหนดช่วงเวลาในการกดปุ่มรหัสซ้ำกัน ซึ่งสามารถปรับเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าได้ โดยเปลี่ยนค่าตัวเก็บประจุที่ขา 6 ของไดซี MM74C922 ส่วนขาที่ 8-11 จะรับสัญญาณจากคีย์เมทริกซ์สวิตช์ในแนวตั้ง และขาที่ 1-4 จะรับสัญญาณในแนวนอนจากคีย์เมทริกซ์สวิตช์ สัญญาณจะออกทางเอาต์พุตขาที่ 14-17 ซึ่งเป็นสัญญาณรหัสฐานสอง 4 บิต ส่วนขา DA จะเป็นสัญญาณอ้างอิง เมื่อมีการส่งสัญญาณออกทางเอาต์พุต ไปยังพอร์ต 0 ของไอซี 89C51




**รูปที่ 3.5** ส่วนรับข้อมูลคีย์เมทริกซ์สวิตช์

### 3.2.2 ส่วนควบคุมอุปกรณ์โซลินอยด์ และลำโพง

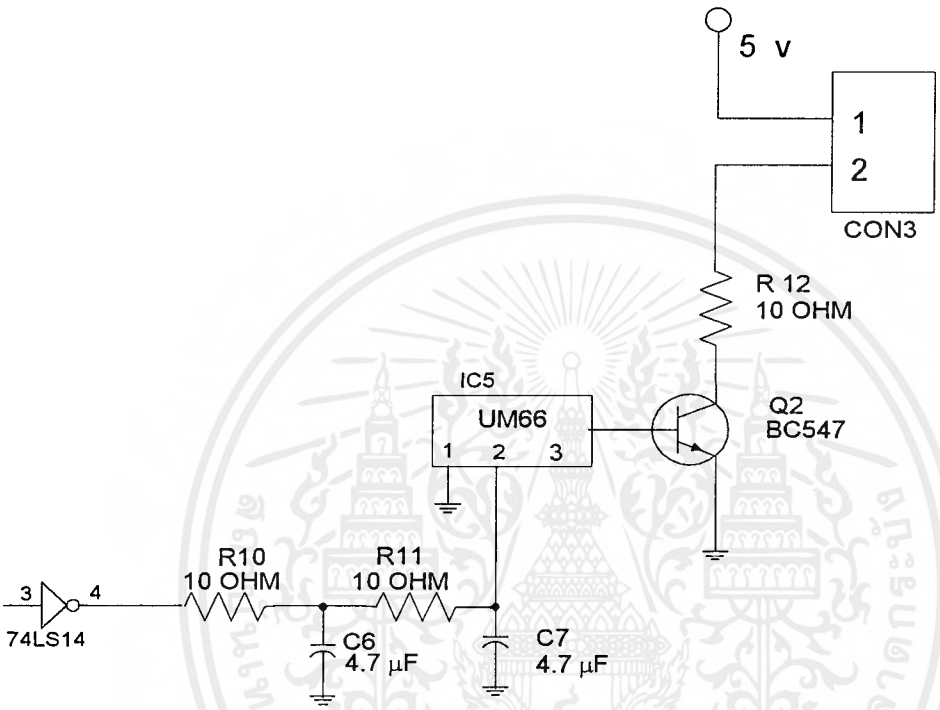
ในส่วนควบคุมอุปกรณ์โซลินอยด์ และลำโพง จะได้รับสัญญาณควบคุมที่ส่งมาจากพอร์ต 1.5 ของไอซี 89C51 มีผลทำให้โซลินอยด์ และลำโพง ทำงานเมื่อมีการตรวจสอบรหัสข้อมูลจากไมโครโปรเซสเซอร์เรียบร้อยแล้วว่าถูกต้อง จึงทำให้โซลินอยด์เปิดอุปกรณ์ล็อกประตู และลำโพงส่งสัญญาณเสียงดนตรี เพื่อแสดงให้ทราบว่า ให้เปิดประตูผ่านเข้าไปได้

**ส่วนควบคุมลำโพง**

เมื่อมีการรับรหัสข้อมูลจากชุดหัวอ่าน หรือ จากการกดคีย์เมทริกซ์สวิตช์ รหัสข้อมูลจะถูกตรวจสอบบนไอซี 89C51 และส่งสัญญาณควบคุมส่วนหนึ่งออกมาที่พอร์ต 1.5 ขาที่ 7 ของไอซี 89C51 เพื่อส่งสัญญาณไปยังอินพุต ขาที่ 5 ของไอซี 7414 ซึ่งเป็นนอทเกต ออกทางเอาต์พุต ขาที่ 6 และจะเก็บประจุไว้ที่ตัวเก็บประจุสองตัว ซึ่งทำหน้าที่ หน่วงเวลาในการไบอัสแรงดันให้กับไอซี UM66 จึงทำให้สามารถที่จะทำงานได้ระยะเวลานานขึ้น ซึ่งไอซี UM66 ทำหน้าที่กำเนิดเสียงดนตรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีจำนวน 3 ขา โดยขาที่ 2 เป็นอินพุต,ขาที่ 3 เป็นเอาต์พุตแรงดันให้กับไอซี UM66 จึงทำให้สามารถที่จะทำงานได้ระยะเวลานานขึ้น ซึ่งไอซี UM66 ทำหน้าที่กำเนิดเสียงดนตรี มีจำนวน 3 ขา โดยขาที่ 2 เป็นอินพุต, ขาที่3 เป็นเอาต์พุต



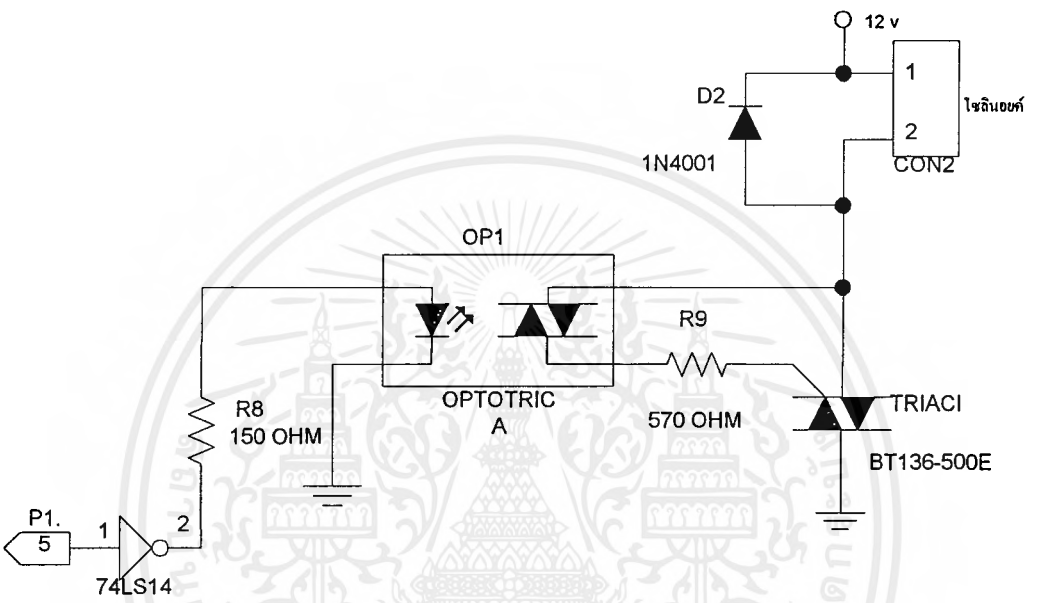

**รูปที่ 3.6** วงจรควบคุมการทำงานของลำโพง

ขาที่ 1 เป็นจุดกราวด์ร่วม เมื่อมีแรงดันมาไบอัสที่ขาอินพุตของไอซี UM66 จะทำให้เกิดสัญญาณเอาต์พุตเป็นจังหละเสียงดนตรีที่ถูกบันทึกไว้ในหน่วยความจำทรานซิสเตอร์ BC547 ซึ่งทำหน้าที่ขยายสัญญาณที่ได้จากเอาต์พุตของไอซี UM66 ให้มีขนาดกำลังสัญญาณที่เหมาะสม เพื่อไปขับลำโพงให้เกิดคลื่นเสียงที่สามารถรับฟังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ส่วนควบคุมโซลินอยด์**

ส่วนควบคุมลำโพง จะรับสัญญาณควบคุมสัญญาณเดียวกันกับส่วนควบคุมอุปกรณ์โซลินอย์ คือ พอร์ต 1.5 ของไอซี 89C51 เมื่อมีสัญญาณเป็นพัลซ์ลบ ป้อนเข้ามาที่ ขา 3 ของไอซี 74LS14 ซึ่งเป็นนอทเกต จะเปลี่ยนเป็นสัญญาณพัลซ์บวกที่เอาต์พุต



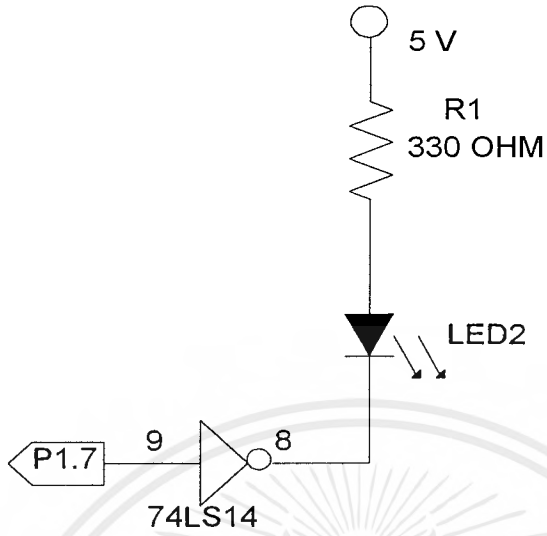

**รูปที่ 3.7** วงจรควบคุมโซลินอยด์

ขาที่ 2 ไบอัสให้แอลอีดีทำงาน ซึ่งอยู่ในชุดของออปโต้ไทรแอค ทำให้ไทรแอคที่อยู่ฆในชุดของออปโต้ไทรแอคนำกระแสไปทริกขาเกตของไทรแอค BT136 ให้ทำงานเปิดวงจรจึงมีกระแสไหลผ่านอุปกรณ์โซลินอยด์ดูดสลักกลับเข้ามา ทำให้ประตูสามารถเปิดได้ โดยจะติดตั้งอุปกรณ์โซนอยด์ไว้กับวงกบประตู

**อุปกรณ์แสดงผล**

อุปกรณ์แสดงผลเป็นอุปกรณ์แสดงผลว่าชุดหัวอ่านบัตรรหัสบาร์โค้ดพร้อมที่จะรับรหัสข้อมูล ถ้าพร้อมที่จะทำงานไดโอดเปล่งแสงจะแสดงผล โดยการกระพริบแสงออกมา เมื่อวงจรแสดงผลชุดหัวอ่านบัตรรหัสบาร์โค้ดพร้อมที่จะทำงาน จะได้รับสัญญาณพัลซ์ควบคุมจาก พอร์ต 1.7 ของขาที่ 8 ของไอซี 89C51 เมื่อแอลอีดีกระพริบตามสัญญาณพัลซ์ควบคุมอยู่ตลอดเวลา แสดงว่าชุดหัวอ่านพร้อมที่จะรับรหัสข้อมูล จากการรูดบัตรรหัสบาร์โค้ด

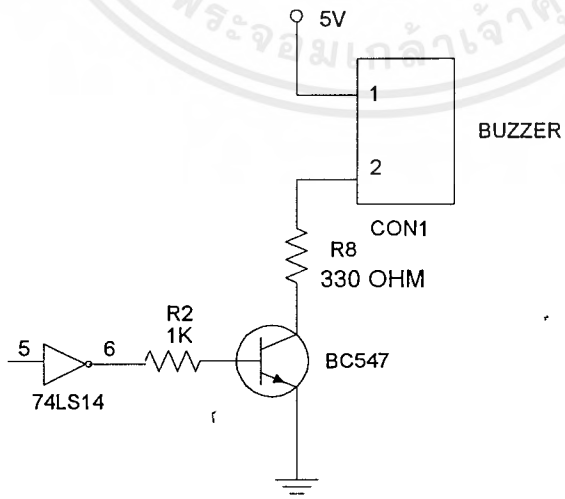
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.8** อุปกรณ์แสดงผลของชุดหัวอ่าน

**สัญญาณเตือน**

เมื่อมีการรูดบัตรรหัสบาร์โค้ดผ่านชุดหัวอ่านบัตรรหัสบาร์โค้ด แต่รหัสข้อมูลไม่ผ่านจะมีสัญญาณเสียงเตือนออกทางบัซเซอร์ โดยวงจรจะรับสัญญาณควบคุมมาจากพอร์ต 1.6 ของไอซี 89C51 มาไบอัสทรานซิสเตอร์ให้ทำงานขับบัซเซอร์ให้กำเนิดสัญญาณเสียงเตือนเมื่อมีผู้รูดบัตรรหัสบาร์โค้ดไม่ผ่าน



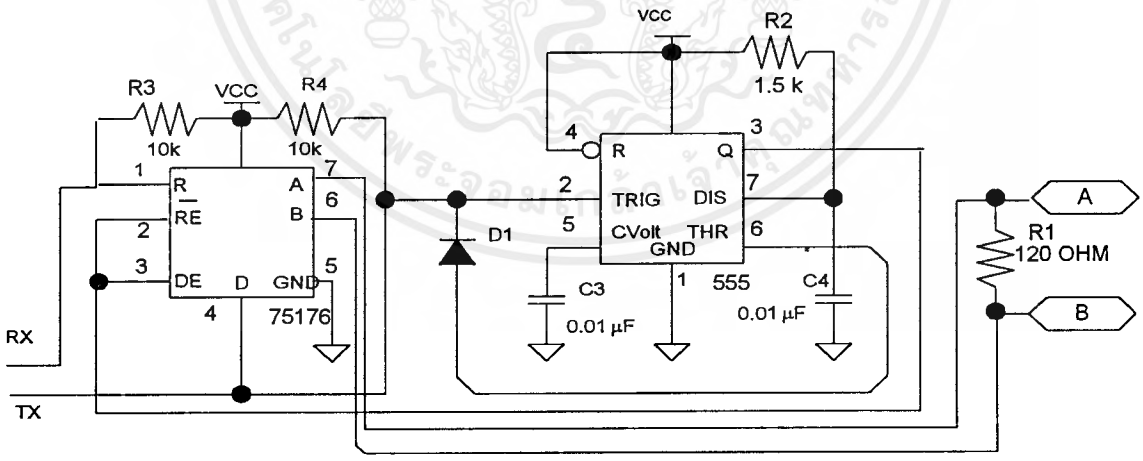

**รูปที่ 3.9** วงจรสร้างสัญญาณเตือน เมื่อมีการรูดบัตรรหัสบาร์โค้ดไม่ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3 ส่วนสื่อสารข้อมูล

ในการสื่อสารข้อมูลจากไมโครโปรเซสเซอร์ไปยังไมโครคอมพิวเตอร์ ได้กำหนดรูปแบบของเครือข่ายแบบบัส การทำงานจะใช้สายตัวกลางแยกออกไปเป็นกิ่งก้านซึ่งจะส่งข้อมูลไปยังตัวกลาง โดยมีไมโครคอมพิวเตอร์ ซึ่งทำหน้าที่เป็นศูนย์กลางประมวลผลสัญญาณซึ่งจะรับรหัสสัญญาณข้อมูลจากไมโครโปรเซสเซอร์ทุกตัว โดยผ่านทางพอร์ต RS-485 เมื่อไมโครโปรเซสเซอร์รับข้อมูลจากชุดหัวอ่านหรือคีย์เมทริกซ์สวิตช์ แล้วนำไปประมวลผลข้อมูล และทำการส่งข้อมูลไปยังไมโครคอมพิวเตอร์ ในการรับส่งจะใช้ไอซี SN75176 ซึ่งเป็นอุปกรณ์ที่มีตัวรับส่งอยู่ในเดียวกัน ในการส่งข้อมูลจากขา 11 ของไอซี 89C51 ซึ่งเป็นขา TxD จะต้องให้ขาที่ 2 และขาที่ 3 ของไอซี SN75176 ซึ่งจะได้รับพัลซ์ลบ ซึ่งเอาต์พุตออกทางขา 6 และ ขา 7 เพื่อไปต่อยังไมโครอมพิวเตอร์ และในการรับข้อมูลเข้ามาโดยผ่านขาที่ 6 และขาที่ 7 ของไอซี SN75176 ซึ่งจะให้ขาที่ 2 และขาที่ 3 เป็นพัลซ์บวก และต่อขาที่ 1 เข้ากับไมโครโปรเซสเซอร์ 89C51 ขาที่ 10 ซึ่งเป็นขา RxD

ในการสื่อสารข้อมูลจากไมโครโปรเซสเซอร์ไปยังไมโครคอมพิวเตอร์ ได้กำหนดรูปแบบของเครือข่ายแบบบัส การทำงานจะใช้สายตัวกลางแยกออกออกไปเป็นกิ่งก้าน ซึ่งจะส่งข้อมูลไปยังตัวกลาง โดยมีไมโครคอมพิวเตอร์ซึ่งทำหน้าที่เป็นศูนย์กลางจะรับรหัสสัญญาณข้อมูลจากไมโครโปรเซสเซอร์ทุกตัว ผ่านทางพอร์ต RS-485




**รูปที่ 3.10** วงจรสื่อสารข้อมูลผ่าน RS-485

เมื่อไมโครโปรเซสเซอร์รับข้อมูลจากชุดหัวอ่านและคีย์เมทริกซ์สวิตช์ แล้วนำไปประมวลผลและส่งข้อมูลไปยังคอมพิวเตอร์ การรับส่งจะใช้ไอซี SN75176 ซึ่งเป็นตัวรับส่งการส่งข้อมูลจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 11 ของไอซี 89C51 ซึ่งเป็นขา TxD จะต้องให้ขาที่ 2 และขาที่ 3 ของไอซีSN75176 ได้รับพัลซ์ลบ ซึ่งเอาต์พุตจะออกทางขา 6 และขา 7 ไปยังคอมพิวเตอร์

## 3.3 การออกแบบโปรแกรม

การออกแบบโปรแกรมของระบบตรวจสอบบุคคลเข้าหน่วยงานจะต้องออกแบบโปรแกรมควบคุมบนไมโครโปรเซสเซอร์ และควบคุมบนคอมพิวเตอร์ ซึ่งซอฟต์แวร์มีผังโปรแกรมควบคุมการทำงานดังนี้

**ผังโปรแกรมควบคุมบนไมโครโปรเซสเซอร์**

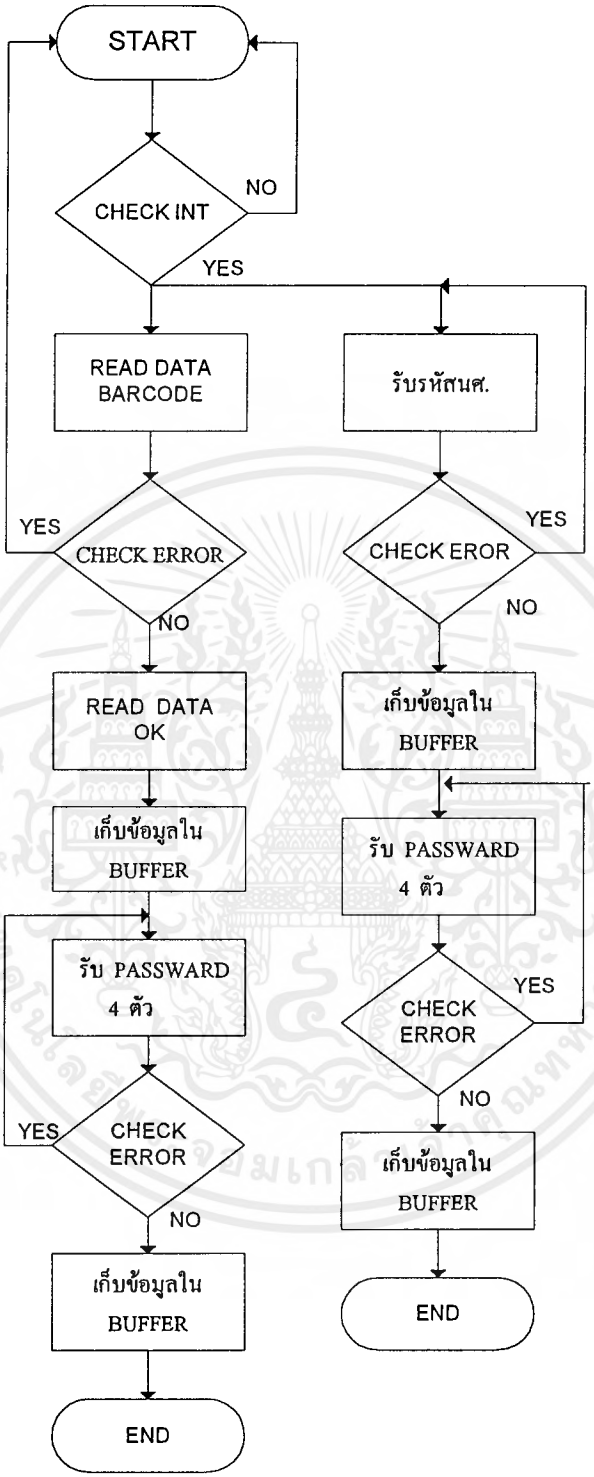
การออกแบบผังโปรแกรมเพื่อควบคุมให้ไมโครโปรเซสเซอร์ทำงานมีอยู่ในส่วนของการรูดบัตรรหัสบาร์โค้ด การกดคีย์สวิตช์ และการสื่อสารรับส่งข้อมูล

**การทำงานของระบบตรวจสอบบุคคลเข้าหน่วยงาน**

จากผังโปรแกรมหลักควบคุมไมโครโปรเซสเซอร์ โปรแกรมเริ่มการทำงานโดยการเช็คการอินเตอร์รัพต์ ว่ามีการอินเตอร์รัพต์จากบาร์โค้ดหรือเมทริกซ์สวิตช์ ถ้ามีการอินเตอร์รัพต์จากบาร์โค้ด โปรแกรมจะรับค่าเข้าไปตรวจสอบ ถ้าผิดพลาด โปรแกรมจะย้อนกลับไปรอรับการอินเตอร์รัพต์ใหม่ ในกรณีที่รหัสถูกต้องก็จะนำรหัสไปเก็บในหน่วยความจำ และทำการรับข้อมูลรหัสผ่านอีก 4 ตัว จากเมทริกซ์สวิตช์ ถ้ารหัสผ่านที่ป้อนผิดพลาดก็กลับไปรับรหัสผ่านใหม่ ถ้าถูกต้องจะนำรหัสผ่านเก็บในหน่วยความจำ จบการทำงานจากโปรแกรมรับข้อมูลจากบาร์โค้ด

ในกรณีที่มีการอินเตอร์รัพต์จากเมทริกซ์สวิตช์ โปรแกรมจะรับค่าเข้าไปตรวจสอบ โดยต้องป้อนรหัสให้ครบ 8 ตัวแล้วกดคีย์ Enter โปรแกรมจะนำรหัสที่ได้มาเก็บในหน่วยความจำ จากนั้นก็จะทำการรับรหัสผ่านจากคีย์เมทริกซ์สวิตช์ในกรณีที่รหัสถูกต้องก็จะนำรหัสไปเก็บในหน่วยความจำ และทำการรับข้อมูลรหัสผ่านอีก 4 ตัว จากเมทริกซ์สวิตช์ ถ้ารหัสผ่านที่ป้อนผิดพลาดก็กลับไปรับรหัสผ่านใหม่ ถ้าถูกต้องจะนำรหัสผ่านเก็บในหน่วยความจำ จบการทำงานจากโปรแกรมรับข้อมูลจากคีย์สวิตช์

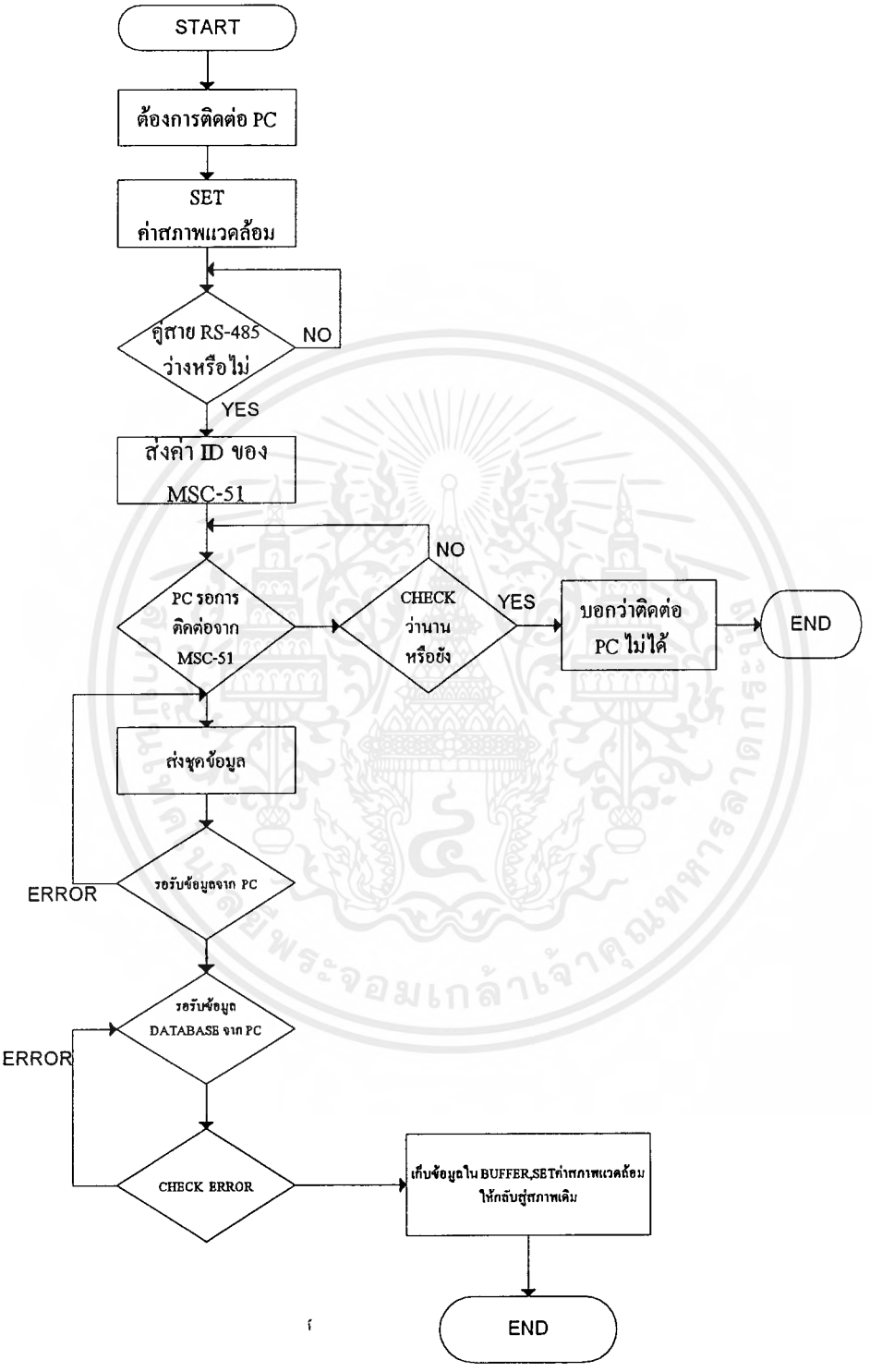
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.11** ผังโปรแกรมของการรูดบัตรและการกดคีย์สวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การสื่อสารรับส่งข้อมูล




**รูปที่ 3.12 ผังโปรแกรมของการสื่อสารข้อมูล**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในส่วนของรับส่งข้อมูลนี้ จะทำงานหลังจากที่ได้ทำการรับค่าของรหัสประจำตัวที่ได้จากบาร์โค้ดและ คีย์สวิตช์ รวมทั้งรหัสผ่าน ซึ่งจะเก็บเอาไว้ในส่วนของหน่วยความจำแรมภายนอกตามที่ได้ตกลงกันไว้ในเบื้องต้น และทำการเซ็ตค่า Registor Flag ที่บ่งบอกว่า 89C51 เองการที่จะติดต่อกับ PC เมื่อ 89C51 ต้องการที่จะติดต่อกับ PC จึงจำเป็นที่จะต้องเซ็ตค่าที่เกี่ยวกับการส่งข้อมูลผ่านทาง Serial Port รวมทั้งการ Disable INT0 และ INT1 ด้วย เพื่อป้องกันการรูดบัตรซ้ำหรือการกดคีย์ซ้ำนั่นเอง

จากนั้น ก็จะนำเอา ข้อมูลรหัสประจำตัว และข้อมูลรหัสผ่าน ไปจัดเรียงตามแบบ ที่ได้กำหนดไว้ ดังนี้

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

จาก รูปแบบการส่งข้อมูล นี้จะไม่อธิบายในการส่งในระดับบิตเพราะถือว่าได้มีความเข้าใจกันได้แล้ว ในที่นี้จะอธิบายเฉพาะเฟรม ข้อมูลทั้งหมดที่ต้องการส่งในแต่ละครั้ง

เฟรมข้อมูลหนึ่งชุดจะมีทั้งหมด 17 Character ซึ่งมีดังนี้คือ

Character Start คือ '<' และบิต Stop คือ '>' ปิดหัวและท้ายของข้อมูล

Character ที่ 2 คือ Character ชื่อเครื่องที่ส่ง

Character ที่ 3 คือ Character ชื่อเครื่องผู้รับ

Character ที่ 4 – 11 เป็น Character ข้อมูลรหัสของ ผู้ที่ต้องการจะใช้บริการ

Character ที่ 12 – 15 เป็น Character ข้อมูลรหัสผ่าน ของ ผู้ที่ต้องการใช้บริการ

Character ที่ 16 เป็น Character Check Sum ที่ได้จากผลบวกของของ Character ข้อมูลทุกตัวและ Character Password ทุกตัวรวมเข้าไว้ด้วยกัน ซึ่งค่าที่ได้นั้น ถ้าเกิดพ้นจำนวนของตัวแปรของ Character ค่า Check Sum ที่ได้จะมีค่าเท่ากับค่า Check Sum ที่ได้จริงหาร ด้วย 255 จนเหลือเศษค่าที่เหลือเศษคือค่า Character ที่ทำการส่งไปในเฟรมข้อมูลจริง ๆ

ในส่วนของ Farm Data Control จะมีลักษณะที่คล้ายกันดังนี้คือ

Character Start คือ " < " และบิต Stop คือ " > " ปิดหัวและท้ายของข้อมูล

Character ที่ 2 คือ Character ชื่อเครื่องที่ส่ง

Character ที่ 3 คือ Character ชื่อเครื่องผู้รับ

Character ที่ 4 เป็น Character ที่บ่งบอกถึงการผิดพลาดที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Character ที่ 5 เป็น Character Check Sum ที่ได้จากผลบวกของของ Character ที่บ่งบอกค่าความผิดพลาดที่เกิดขึ้นในส่วนของตัวแปรของ Character ค่า Check Sum ที่ได้จะมีค่าเท่ากับค่า Check Sum ที่ได้จริงหารด้วย 255 จนเหลือเศษค่าที่เหลือคือค่า Character ที่ทำการส่งไปในเฟรมข้อมูลจริง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# การทดลอง และผลการทดลอง

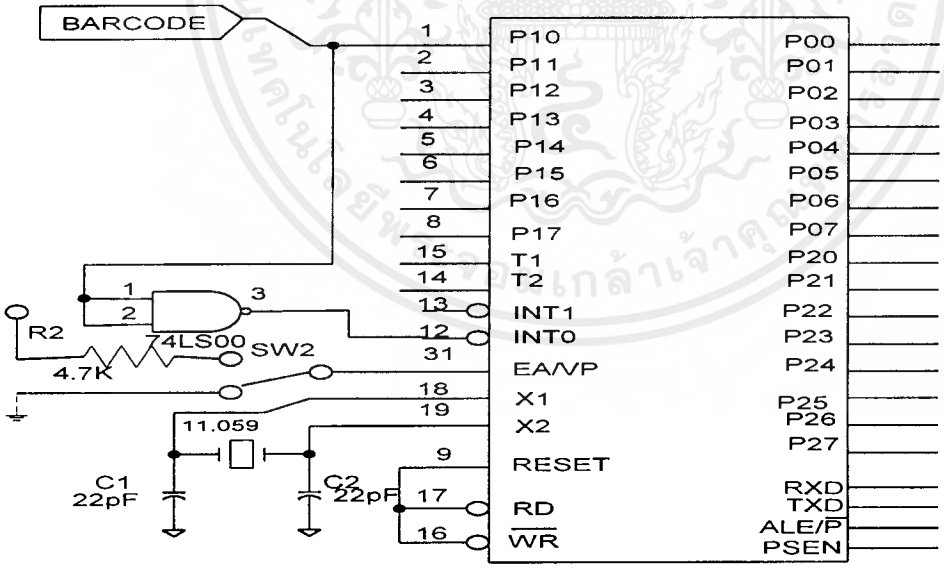
## 4.1 การทดลอง ส่วนรับข้อมูล

เพื่อให้ง่ายแก่การทดลอง และตรวจสอบการทำงานของระบบจึงได้แบ่งการทดลองออกเป็น 3 ส่วน คือ ส่วนแรกเป็นการทดลองส่วนรับข้อมูลจากบัตรรหัสบาร์โค้ด และส่วนรับข้อมูลจากคีย์เมทริกซ์สวิตช์ ส่วนที่สองเป็นการแสดงผลของระบบตรวจสอบบุคคลเข้าหน่วยงานจากบัตรรหัสบาร์โค้ด และการรับข้อมูลจากเมทริกซ์สวิตช์ และส่วนที่สามเป็นการทดลองการรับส่งสื่อสารข้อมูล

### 4.1.1 การทดลองส่วนรับข้อมูลจากบัตรรหัสบาร์โค้ด

**ขั้นตอนการทดลอง**

1. ประกอบวงจรตามรูปที่ 4.1




**รูปที่ 4.1** วงจรส่วนรับข้อมูลจากบัตรรหัสบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. นำบัตรรหัสบาร์โค้ดมาทำการรูดบัตรผ่านชุดหัวอ่าน ในทิศทางตามลูกศร

5. วัดแรงดันที่พอร์ต 1.0 ของไอซี 89C51

4. นำบัตรรหัสบาร์โค้ดมาทำการรูดบัตรผ่านชุดหัวอ่านบัตรรหัสบาร์โค้ด ในทิศทางสวนทางกับหัวลูกศร

5. วัดแรงดันที่พอร์ต 1.0 ของไอซี 89C51

**ผลการทดลอง**

จากการทดลองนี้ จะต้องนำแรงดันออกที่พอร์ต นำไปควบคุมอุปกรณ์ให้ทำงานจากการรับรหัสข้อมูลจากบัตรรหัสบาร์โค้ด โดยควบคุมอุปกรณ์โซลินอยด์,ลำโพง

**ตารางที่ 4.1** ผลจากการวัดแรงดันของวงจรรับข้อมูลจากบัตรรหัสบาร์โค้ด

| ลักษณะการรูดบัตร | พอร์ต 1.0 |
|---|---|
| แรงดันก่อนรูดบัตร | 0 v |
| แรงดันหลังรูดบัตรทิศทางตามลูกศร | 3.75 v |
| แรงดันหลังรูดบัตรทิศทางสวนทางลูกศร | 3.75 v |

**ปัญหาและการแก้ปัญหา**

จากการลอง แรงดันที่จ่ายให้กับวงจร มีค่าเท่า กับ 4.85 v แต่แรงดันที่ได้จากการรูดบัตรแต่ละครั้งมีค่าไม่เท่ากัน ขึ้นอยู่กับความเร็วในการรูดแต่ละครั้ง

### 4.1.2 การทดลองส่วนรับข้อมูลจากเมทริกซ์สวิตช์

**ขั้นตอนการทดลอง**

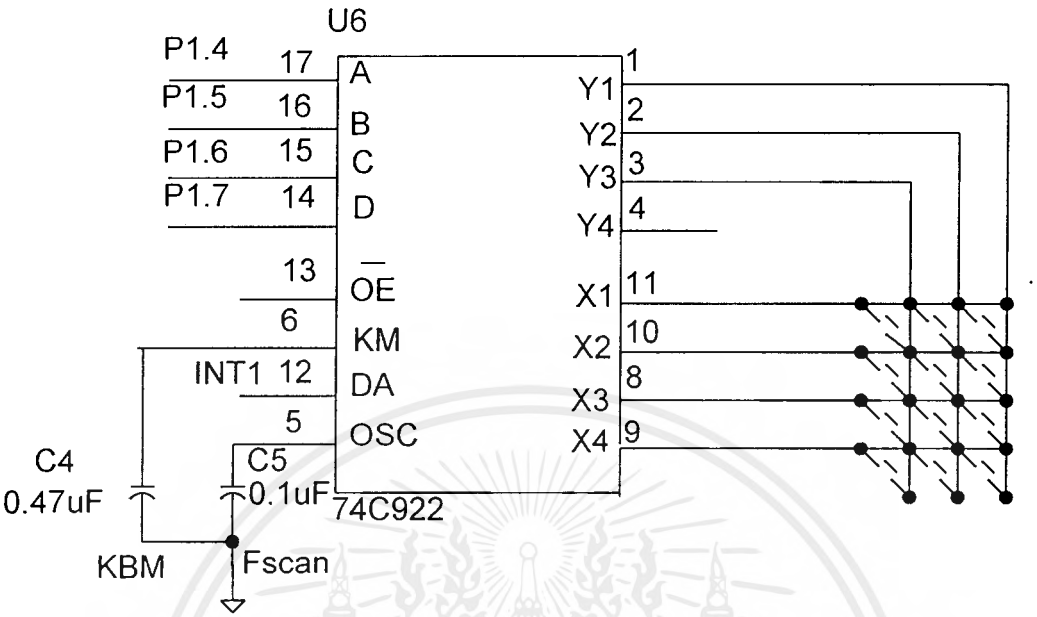
1.ประกอบวงจรตามรูปที่ 4.2

2.ตรวจสอบความเรียบร้อย และต่อสายของเมทริกซ์สวิตช์ให้ถูกต้อง

3.ป้อนแรงดันไฟฟ้ากระแสตรง 4.85 โวลต์

4.วัดแรงดันที่ขาเอาต์พุตของ IC74922 ขา D,C,B,A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.2** วงจรส่วนรับข้อมูลจากคีย์เมทริกซ์สวิตช์

**ผลการทดลอง**

แรงดันที่เอาต์พุตจะเท่ากับค่ารหัสคีย์ที่กด โดยบิท D เป็นบิทที่มีนัยสำคัญสูงสุด ส่วนเอ้าต์พุตที่ขา DA จะมีพัลซ์ออกมา 1 ลูกเมื่อกดคีย์หนึ่งครั้ง ซึ่งขา DA จะต่อกับขา INT0

## 4.2 การทดลองการแสดงผลของระบบ

ในส่วนที่สองของการทดลองระบบตรวจสอบบุคคลเข้าหน่วยงาน จะเป็นการทดลองการแสดงผลของส่วนของวงจรรับข้อมูลจากคีย์เมทริกซ์สวิตช์

**การทดลองการแสดงผลส่วนรับข้อมูลจากคีย์เมทริกซ์สวิตช์**

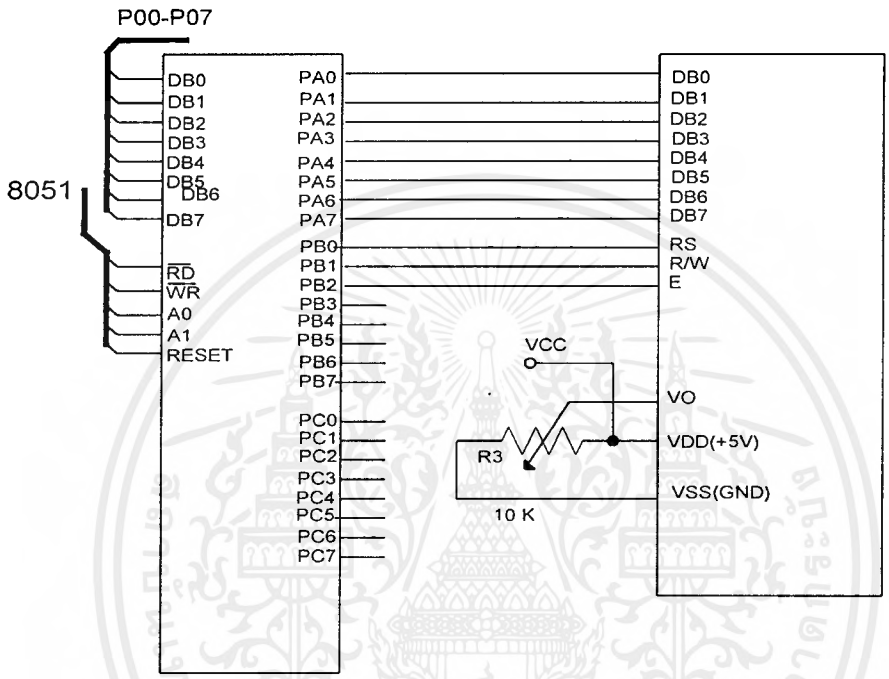
**ขั้นตอนการทดลอง**

1. เข้าโปรแกรมควบคุมอุปกรณ์การทำงาน
2. ต่อชุดวงจร ดังรูปที่ 4.6
3. สังเกตผลที่หน้าจอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ผลการทดลอง**

เมื่อเข้าโปรแกรมควบคุมอุปการณ์การทำงานบนไมโครคอมพิวเตอร์จากนั้นทำการรูดบัตรผ่านชุดหัวอ่านบัตรบาร์โค้ด 1 ครั้ง ผลที่ได้จากการทดลองนั้นจะแสดงบนหน้าจอของ LCD MODULE 16 ตัวอักษร 2 บรรทัด ซึ่งจะแสดงรหัสประจำตัวพร้อมทั้ง PASSWARD KEY 4 ตัว




**รูปที่ 4.3** การต่อวงจรทั้งหมดเข้าด้วยกัน

## 4.3 การทดลองสื่อสารข้อมูล

ในส่วนที่สามเป็นส่วนของการรับส่งสื่อสารข้อมูล จะผ่านขา RX และ TX จะต่อเข้ากับวงจร TTL/485

**การทดลองรับส่งสื่อสารข้อมูล**

**ขั้นตอนการทดลอง**

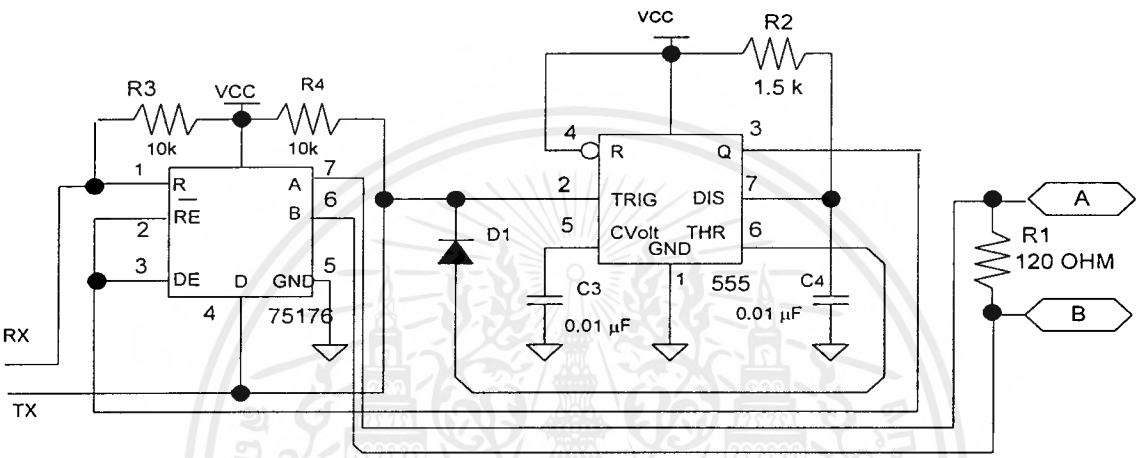
1. ต่อวงจรตามรูป 4.4
2. จ่ายพัลซ์ HIGH ให้กับขา D ของ SN75176
3. วัดไฟที่ ขา A, B
4. จ่ายพัลซ์ LOW ให้กับขา D ของ SN75176

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. วัดไฟที่ขา A,B

**ผลการทดลอง**

ใช้วงจรจ่ายพัลซ์ให้กับขา ENABLE DE ปรากฏว่าให้สัญญาณที่ เอ้าต์พุต ขา A,B ตรงตามคุณสมบัติ




**รูปที่ 4.4** การต่องวจรส่วนสื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

## 5.1 บทสรุป

ปริญญานิพนธ์ฉบับนี้เสนอผลงานเกี่ยวกับระบบตรวจสอบบุคคลเข้าหน่วยงาน ใช้ตรวจสอบบุคคลที่เข้ามาในหน่วยงาน โดยมีไมโครคอมพิวเตอร์เป็นศูนย์กลาง เชื่อมต่อกันเป็นโครงข่ายระหว่างชุดไมโครโปรเซสเซอร์หลายๆชุดโดยใช้ระบบการรูดบัตรที่มีรหัสบาร์โค้ดพร้อมทั้งกดคีย์รหัสผ่าน ซึ่งติดตั้งได้ง่ายที่ทางเข้าออกประตู เหมาะสำหรับทุกสถานที่ที่ต้องการตรวจสอบ

## 5.2 ปัญหา และแนวทางแก้ปัญหา

จากการทดลอง การทำงานของระบบตรวจสอบบุคคลเข้าหน่วยงานได้พบปัญหา และแก้ไขดังนี้

1. การต่อสัญญาณของ 8255 ผิดพลาดเพราะขา Vcc กับ Gnd ไม่ใช่ขา 40 และ 20 ตามลำดับแต่เป็นขา 26 และ 7 แทน

การทดลองแผงวงจรควบคุม 89C51 ผิดพลาดเนื่องจากวงจรถอดรหัสนำ OR มาใส่แทนทั้งที่ต้องการ สภาวะการทำงานที่เป็น AND จึงทำให้ไม่สามารถเลือก Rom ให้ทำงานได้

วิธีการแก้ไข ก่อนการทดลองควรระมัดระวังทำการตรวจสอบอุปกรณ์ก่อน

2. ทำการทดลองวงจรภาคคีย์สวิตช์ โดยการเขียนโปรแกรมควบคุมดู ปรากฏกว่าเมื่อทำการติดต่อกับ ภาคอินเตอร์รัพต์ มีผลที่วัดได้ที่ Port 1 คือ แรงดันที่ Port 1 ของ 89C51 นั้นไม่คงที่ทำให้ไม่สามารถที่จะนำเอาข้อมูลตรงที่ได้มา ทำการ Compile ต่อไปได้ สาเหตุเป็นเพราะการต่อ IC 74922 กับพร์อตโดยตรงทำให้ในบางสภาวะของ 74922 และ 89C51 นั้นมีการถ่ายเทกระแส หรือเกิดการ Short กัน

วิธีแก้ไข คือ ทำการย้ายภาค O/P ของ 74922 ไปต่อที่ 8255 แทนอาจจะดีขึ้นมาได้

3. ทำการเขียนโปรแกรมรับส่งข้อมูล ซึ่งเป็นส่วนที่รับข้อมูลมาจากโปรแกรมของการรูดบัตรบาร์โค้ดหรือกดคีย์เพื่อป้อนข้อมูลให้ 89C51 แล้ว

ในโปรแกรมส่วนนี้การทำงานจะเป็นการจัดรูปแบบของข้อมูลที่จะส่งผ่านสาย RS-485 ไปยังPC หรือ Server ซึ่งการทำงานส่วนใหญ่จะเป็นการจัดการเกี่ยวกับหน่วยความจำ ภายนอกทั้งสิ้น

ปัญหาที่เกิดขึ้นคือ ไม่สามารถที่จะติดต่อกับ Ram ภายนอกได้ จึงได้ทำการหาสาเหตุดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ทำการทดลองโดยการกำหนดเกี่ยวกับการจัดการข้อมูล ที่เกี่ยวกับหน่วยความจำภายนอกไปทำการประมวลผลที่หน่วยความจำภายในแทน ผลปรากฏว่าใช้งานได้ตามปกติ

3.2 ทำการทดลองโดยในขณะที่เข้าถึงหน่วยความจำภายนอก ให้ทำการแสดงค่าที่อยู่ในหน่วยความจำออกมาดู ปรากฏว่าไม่มีข้อมูลอยู่ในหน่วยความจำ

3.3 ทดลองโดยการเขียนโปรแกรมติดต่อกับหน่วยความจำภายนอก แล้ววัดสัญญาณที่ขา PSEN และ ALE ปรากฏว่า มีการตอบสนองสัญญาณ ในช่วงเวลาสั้น ๆ เท่านั้น ซึ่งจะต่างจากการติดต่อกับหน่วยความจำภายในที่จะให้ สัญญาณตอบสนองที่ช่วงเวลาที่กว้างกว่า

3.4 จากการทดลองในครั้งที่แล้ว กลัวว่าจะผิดพลาดในการทดลองจึงทำการทดลองใหม่ ซึ่งเป็นโปรแกรมที่เขียนในรูปแบบของแอสแซมบลี้ ปรากฏว่าสามารถที่จะติดต่อกับหน่วยความจำภายนอกได้ตลอดทั้งย่านที่ ถอดรหัสเอาไว้

จากการทดลองข้างต้นกลัวว่าจะมีปัญหากับส่วนของการถอดรหัสและอาจจะเป็นที่ตัว IC ที่อยู่ในแผงวงจร จึงได้ทำการทดลองโดยการเปลี่ยน IC และอุปกรณ์ต่าง ๆ ที่สามารถจะเปลี่ยนได้แล้วทำการทดลอง จากข้อ 1 - 4 อีกครั้ง ปรากฏว่าผลที่ได้เหมือนเดิม

ทำการทดลองโดยการใช้บอร์ดของ บริษัท ศิลา จำกัด แล้วทำการทดลองเหมือนเดิม ผลที่ได้คือมีบอร์ดอยู่รุ่นเดียวที่สามารถทำงานติดต่อกับหน่วยความจำภายนอกได้

## 5.3 แนวทางแก้ไข และการพัฒนาโครงงาน

เราสามารถที่จะปรับปรุงให้มีขีดความสามารถมากขึ้น

1. สามารถตรวจสอบว่ามีคนเข้าไปในห้องเป็นจำนวนคน ต่อการรูดบัตร หนึ่งครั้ง โดยมีการเตือนด้วย

2 ใช้ไมโครคอมพิวเตอร์เป็นงานอื่นๆได้ด้วย เมื่อมีการรูดบัตร จึงเข้าสู่ระบบตรวจสอบเข้าหน่วยงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก
# โปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมรับข้อมูลจากบาร์โค้ด และเมทริกซ์สวิตช์

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <AT89x51.h>
#include "lcdcmn.h"
#include <intrins.h>

#define PORTA   0x6000
#define PORTB   0x6001
#define PORTC   0x6002
#define PORTP   0X6003

unsigned char idata XInt_0;
int a;

unsigned char xdata *p;
unsigned char idata Key_Char;
unsigned char idata ID_Code[8];
unsigned char idata Pwd_Code[3];
unsigned char akey,keybuf;
unsigned char row;
char *str;
char buf[LEN];
char i,l,skey,spwd,ID_OK;

bit bar;
bit key;
bit Serial;
bit pwd_key;

void Barcode(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Key_Sw(void);
void title(void);
void start(void);
void delay(unsigned int val);
void init8255 (void);
void initlcd(void);
void lcdctrl(int command);
void lcdwrite(char c);
void setpos(unsigned char row ,unsigned char col);
void clrscr(void);
void printflcd();
void lcdbarcode(void);
void lcdidkeys(void);
void lcdpwdkeys(void);
void inkey (void);
void Id_Char (void);
void getidcode(void);
void getpwdcode(void);
void Pwd_Char(void);
void Chk_Key(void);
void Del_Key(void);

/******************/
/** main program **/
/******************/

void main(void)
{
        char c;
        unsigned char i;

        IE      = 0x85;   /* Enable Interrupt 0,1                */
        TCON = 0x05;  /* Set Type Control Interrupt          */
                                /* '1' =  edge/low or '0' = level triggered */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
start();
init8255();
initlcd();
clrscr();
delay(200);
title();
for (i=0;i<5;i++)delay(30000);
lcdidkeys();
i=0;
bar = 0;
key = 0;
ID_OK=0;
Serial = 0;
while(!Serial){

        if(key){
          //lcdscankeys();
          if(!ID_OK)  Id_Char();
          if(ID_OK)  Pwd_Char();
          key = 0;
        }

        if(bar){
                lcdbarcode();
                bar = 0;
        }
}

/*************************************************************/
/***************** Function For Used  ************************/
/*****************  All of function  ************************/
/*************************************************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void title (void)               /* Display Title */
 {
          setpos(0,0);
          printf("Check in System ");
          setpos(1,0);
          printf("< By Computer! >");
          for (I=0;I<2;I++)delay(30000);
          /*setpos(0,0);
          printf("Create By EC.2/2");
          setpos(1,0);
          printf("Jigko MIN Frank!");
          for (I=0;I<2;I++)delay(30000);
          setpos(0,0);
          printf("** Advisor By **");
          setpos(1,0);
          printf("Mr. Koson Trachu");
          for (I=0;I<3;I++)delay(30000);*/
 }
void lcdbarcode (void)                          /* Display Barcode */
{
          clrscr();
          setpos(0,0);
          printf("INT. BY BARCODE!");
          setpos(1,0);
          printf("ENTER ID. CODE #");

void lcdidkeys (void)           /* Display Input ID. Code */
{
          setpos(0,0);
          printf("<Enter ID. Code>");
          ID_OK = 0;
          setpos(1,0);
          printf(" [          ] ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (skey==0){
             setpos(1,0);
             printf("* [         ] *");
             skey=1;
            }
}
void inkey (void)                              /* Display Charecter */
{
            akey&=0xf0;
            akey=keybuf>>4;
             switch (akey){
              case 0x00:{ putchar('0'); Key_Char = '0'; pwd_key=0;} break;
              case 0x01:{ putchar('1'); Key_Char = '1'; pwd_key=0;} break;
              case 0x02:{ putchar('2'); Key_Char = '2'; pwd_key=0;} break;
              case 0x03:{ putchar('3'); Key_Char = '3'; pwd_key=0;} break;
              case 0x04:{ putchar('4'); Key_Char = '4'; pwd_key=0;} break;
              case 0x05:{ putchar('5'); Key_Char = '5'; pwd_key=0;} break;
              case 0x06:{ putchar('6'); Key_Char = '6'; pwd_key=0;} break;
              case 0x07:{ putchar('7'); Key_Char = '7'; pwd_key=0;} break;
              case 0x08:{ putchar('8'); Key_Char = '8'; pwd_key=0;} break;
              case 0x09:{ putchar('9'); Key_Char = '9'; pwd_key=0;} break;
              case 0x0a:  Del_Key();    break;    /* KEY * = BACK SPACE */
              case 0x0b:{
                                        if (!ID_OK) getidcode ();
                                        else if ((ID_OK) && (i!=0))
                                                        getpwdcode();// i!=0;
                                       }  break;  /* KEY # = ENTER */

             default:  {
                                        i--;
                                       }

            }
void Id_Char (void)      /* Print charecter to LCD MODULE */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        setpos(1,i+4);
        inkey();
        delay(200);
        ID_Code[i] = Key_Char;
        if (!pwd_key){
        setpos(0,0);
        printf("%c",ID_Code[i]);
        }
        i++;
        if(i > 8) Chk_Key();

void Del_Key(void)      /* Delete Charecter*/
{
        if(i > 0){
                i--;
                setpos(1,i+4);
                putchar(' ');
                if (i < 0)  i=0;
                setpos(1,i+4);
                i--;

        }
        else {
                        i = 0;
                        setpos(1,i+4);
                        i--;
                }
}


void getidcode(void)
{
          if ( i < 8){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                setpos(0,0);
                printf("< Data Error!! >");
                delay(60000);
                setpos(0,0);
                printf("<Enter ID. Code>");
                i--;
        }
        if ( i > 8 ) i--;
}


void Chk_Key(void){
                if(!ID_OK){
                        if((i > 7)&&(akey==0x0b)){
                          setpos(0,0);
                          printf("* Get Data OK. *");
                          delay(30000);
                          delay(30000);
                          spwd = 0;
                          lcdpwdkeys();
                          ID_OK = 1;
                          i = 0;
                        }
                        else {
                                        ID_OK = 0;
                                        setpos(0,0);
                                        printf("< Data Error!! >");
                                        delay(30000);
                                        delay(30000);
                                        setpos(0,0);
                                        printf("<Enter ID. Code>");
                                        setpos(1,0);
                                        printf("* [        ] *");
                                        i = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
            }
            else if(ID_OK){
                    if((i > 3)&&(akey==0x0b)){
                      setpos(0,0);
                      printf("* Password OK! *");
                      delay(60000);
                      setpos(1,0);
                      printf("* Please Wait! *");
                      delay(60000);
                      skey = 0;
                      lcdidkeys();
                      ID_OK = 0;
                      i = 0;
                      Serial = 0;
                     }else {
                                ID_OK = 1;
                                setpos(0,0);
                                printf("Password Error!");
                                delay(30000);
                                delay(30000);
                                setpos(0,0);
                                printf("<Enter Password>");
                                setpos(1,0);
                                printf("* [     ] 4# *");
                                i = 0;
                                //i--;

                    }
            }
}

void lcdPwdkeys(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        setpos(0,0);
        printf("<Enter Password>");
        ID_OK = 1;
        if (spwd==0){
                setpos(1,0);
                printf("* [     ] 4# *");
                spwd=1;
        }   i=0;
}


void Pwd_Char (void)
{
        pwd_key=1;
        setpos(1,i+4);
        inkey();
        delay(200);
        Pwd_Code[i] = Key_Char;

                if (!pwd_key){
                        setpos(1,i+4);
                        putchar('*');
                        delay(200);

                }
                        i++;

        if(i > 4) Chk_Key();

void getpwdcode(void)
{
        if (i<4){
                                //if (!pwd_key){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        //pwd_key=1;
                                        setpos(0,0);
                                        printf("Password Error!");
                                        delay(30000);
                                        delay(30000);
                                        setpos(0,0);
                                        printf("<Enter Password>");
                                        //setpos(1,0);
                                        //printf("* [     ] 4# *");
                                        //i=0;
                                        i--;
                                        //pwd_key=1;
            }
            if (i>4) i--;
  //        }
}


void start (void)
{
        SCON = 0x50;    /* scon mode 1, 8 bit UART ,enable rcvr */
        TMOD = 0x20;    /* tmod timer1, mode22, 8 bit reload    */
        TH1 = 0xfd;             /* th1 reload value for 9600 baud      */
        TR1 = 1;                /* TIMER 1 RUN                         */
        T1 = 1;             /* t1 set t1 ti send first char of UART */

void delay (unsigned int val)
{

        for (;val;val--);
}

void initlcd (void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lcdctrl(SETFUNC);
    delay(20);
    lcdctrl(ON_OFF);
(   delay(20);
    lcdctrl(CLRSCR);
    delay(20);
    lcdctrl(0x14);
}


void init8255 (void)
{
        p=PORTP;
  *p=0x80;
        delay(50);
}
void lcdctrl(int command)
{
        delay(20);
        p = PORTB;
  *p = 0x00;
        p = PORTA;
  *p = command;
        p = PORTB;
  *p = 0x04;
        delay(20);
        p = PORTB;
  *p = 0x00;
}

char putchar (char c)
{
        p = PORTB;
  *p = 0x01;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            delay(20);
            p = PORTA;
    *p = c;
            p = PORTB;
    *p = 0x05;
            delay(20);
            p = PORTB;
    *p = 0x00;
            return c;
}


void setpos (unsigned char row , unsigned char col)
{
  delay(20);
  switch(row){
                case 0: lcdctrl((0x80+col|0x80)); break;
                case 1: lcdctrl((0xC0+col|0x80)); break;
  }
}


void clrscr()
{
  delay (20);
  lcdctrl(CLRSCR);
}


void Barcode(void) interrupt 0 using 1
{
            EA = 0;
            bar= 1;
//          delay(20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        EA = 1;
}

void Key_Sw(void) interrupt 2 using 2
{
        EA = 0;
        key= 1;
        delay(15000);
        keybuf = 0;
        keybuf = P1;
        EA = 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรมการสื่อสารข้อมูล

```
/*****************************************************************/
/*          Program Serial for Board 89C51                       */
/*             By Kitipong   Promlert                            */
/*             Admin Mr.Koson  Trahu                             */
/*****************************************************************/


#include <at89x51.h>
#include <stdio.h>
//#include <conio.h>



//********** ID for PC and 80C51 ****************************//

char xdata ID_C51 = 'A';
char xdata ID_PC = 'Z';
char xdata Std_Frame = '<';
char xdata Stp_Frame = '>';
//********** Send Data to serial ****************************//

char xdata Out_Data = 0;
char xdata Send_SumOut  = 0;
char xdata Send_Buf[20];
//char xdata S[10];
char xdata Send_PC_Error[10];



//********** receive Data to serial ***************************//

char xdata In_Data = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int  xdata In_Char = 0;
char xdata Check_SumIn = 0;
char xdata Recv_Buf[20];

//********** Flage Register Status ****************************//

unsigned int val;
char xdata ID_Code[]="40031404";
char xdata Pwd_Code[]="7700";
char xdata Error = 0xFF;
char xdata Right = 0xFE;


bit Send_PC;
bit Send_Error;
bit Recv_Error;
bit Yes_ID;
bit Pass;

void buf(void);
void Out_Buf(void);
void Check_ID(void);
void PC_Error(void);
void delay(int val);
void Check_Sum(void);
void Send_Data(void);
void Check_Send(void);
void Buf_to_Ram(void);
void Check_Error(void);
void Char_Amount(void);


/*************************************************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*              MAIN PROGRAM SERIAL                    */
/*************************************************************/

void main(void){ /* execution starts here after stack init   */

        Yes_ID = 0;
        Pass = 0;
        Send_Error = 1;
        Recv_Error = 0;

        SCON  = 0x50; /* SCON: mode 1, 8-bit UART, enable rcvr    */
        TMOD |= 0x20; /* TMOD: timer 1, mode 2, 8-bit reload      */
        TH1   = 0xfd; /* TH1:  reload value for 9600 baud         */
        TR1   = 1;    /* TR1:  timer 1 run                        */
        TI    = 1;    /* TI:   set TI to send first char of UART  */


 do{

         while (Send_Error){

                 Send_Error = 0;

                 Out_Buf();

                 Send_Data();


//               scanf("%s",&Recv_Buf);
                 buf();

                printf("\n\n\n");delay(1);
                printf("Print Form for \n");delay(1);
                for(In_Data  = 0;In_Data < 17;In_Data++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                printf ("%c",Send_Buf[In_Data]); //delay(1);

        }
        printf("\n");delay(1);
        for(In_Data  = 0;In_Data < 17;In_Data++){
                printf ("%c",Recv_Buf[In_Data]); //delay(1);
        }
        printf("\n");delay(1);


        printf("Pirnt from string\n");delay(1);
        printf("Send = %s\n",Send_Buf);delay(1);
        printf("Revc = %s\n",Recv_Buf);delay(1);
        printf("Check_Sum = %c\n",Send_SumOut);delay(1);

//      sprintf (S,"%s",Send_Buf);
//      delay(1);
//      sprintf (S,"%s",Recv_Buf);
//      delay(1);
//      printf("from s = [%s]",Recv_Buf);


        printf("Check_ ID = ");delay(1);
        Check_ID();
        if(Yes_ID){

//        Recv_Buf[15]=Error;
                printf("Yes_ID\n");delay(1);
                printf("Char_Amount = ");delay(1);
                Char_Amount();
//              In_Char++;
                printf("Char = %d\n",In_Char);delay(1);
                Check_Sum();
                printf("Check_SumIN = %c\n",Check_SumIn);delay(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if(Recv_Error){
                                printf("Recv_Error\n");delay(1);
                                printf("Send PC = ");
                                PC_Error();

                    }else{ printf("Recv_OK\n");delay(1);

                                }

                    if(Send_Error){
                                Yes_ID = 0;
                                Pass = 0;printf("Send_Error\n");delay(1);

                    }else{ printf("Send_OK\n");delay(1);

                                }
                }
        }

 // Recv_Error = 0; Send_Error = 0;
 if(!Recv_Error && !Send_Error) Buf_to_Ram();
 }while(Pass);
 printf("Serial Pass\n");delay(1);
}


//********** Put Data to Output Buffers **********//

void Out_Buf(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Send_SumOut = 0;
        for(Out_Data = 0;Out_Data < 17;Out_Data++){

                switch(Out_Data){
                        case 0 : Send_Buf[Out_Data] = Std_Frame;
                                        break;
                        case 1 : Send_Buf[Out_Data] = ID_C51;
                                        break;
                        case 2 : Send_Buf[Out_Data] = ID_PC;
                                        break;
                        case 15: Send_Buf[Out_Data] = Send_SumOut;
                                        break;
                        case 16: Send_Buf[Out_Data] = Stp_Frame;
                                        break;
                        default:{
                                if((Out_Data > 2) && (Out_Data < 11)){
                                        Send_Buf[Out_Data] = ID_Code[Out_Data - 3];
                                }
                                else if((Out_Data > 10) && (Out_Data < 15)){
                                        Send_Buf[Out_Data] = Pwd_Code[Out_Data - 11];
                                }
                                Send_SumOut = Send_SumOut + Send_Buf[Out_Data];
                        }
                }
        }
        Send_Buf[17] = '\0';
}

//*****************Receive Data To SERIAL PORT***************//

void buf(void){

        for(In_Data  = 0;In_Data < 17;In_Data++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Recv_Buf[In_Data] = Send_Buf[In_Data];delay(1);
//              printf ("[%c] = ",Send_Buf[In_Data]); //delay(1);
//              printf ("[%c]\n",Recv_Buf[In_Data]); //delay(1);

        }
        Send_Buf[17] = '\0';
}



//*******************Send Data To SERIAL PORT*****************//

void Send_Data(void){

                printf ("%s\n",Send_Buf);
                delay(1);

}


//********************* Function Delay **********************//

void delay(int val){
        while (val){
                                val--;
                        }
}


//*************** Check ID_C51 of Receiv from Serial *********//

void Check_ID(void){
        if(Recv_Buf[1] == ID_C51)Yes_ID = 1;
        else{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                In_Data    = 0;
                Yes_ID     = 0;
                delay(ID_C51);
        }
        delay(ID_C51+200);

}

//********* Check Sum Data to Receiver from Serial **********//

void Check_Sum(void){

        for(In_Data = 0;In_Data < (In_Char - 1);In_Data++){
                if((In_Data > 2) && (In_Data < 15)){
                Check_SumIn = Check_SumIn + Recv_Buf[In_Data];
                }
        }

//      Recv_Buf[In_Char - 1] = (Check_SumIn - 1);
        if(Check_SumIn == Recv_Buf[In_Char - 1]){
                Recv_Error = 0;
                Check_Send();
        }

        if(Check_SumIn != Recv_Buf[In_Char - 1])
                Recv_Error = 1;

}

//********** Check Send Error to Transceiver **************//

void Check_Send(void){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(Recv_Buf[3]== Error) Send_Error = 1;
        else Send_Error = 0;

}

//********* From Buffers of Receiv to RAM ID & Pwd *********//

void Buf_to_Ram(void){

        if(Recv_Buf[3]== Right) Pass = 1;

}

//****** Chack Char Amount of Frame Data From SERIAL ******//

void Char_Amount(void){

        In_Char = 0;

        while(Recv_Buf[In_Char] != Stp_Frame){
                In_Char++;
        }

}

//******** Chack Send Data Error for Data From PC ********//

void PC_Error(void){

        Send_PC_Error[0] = Std_Frame;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Send_PC_Error[1] = ID_C51;
        Send_PC_Error[2] = ID_PC;
        Send_PC_Error[3] = Error;
        Send_PC_Error[4] = Error;
        Send_PC_Error[5] = Stp_Frame;
        printf ("%s\n",Send_PC_Error);
//      delay(1);
}
```

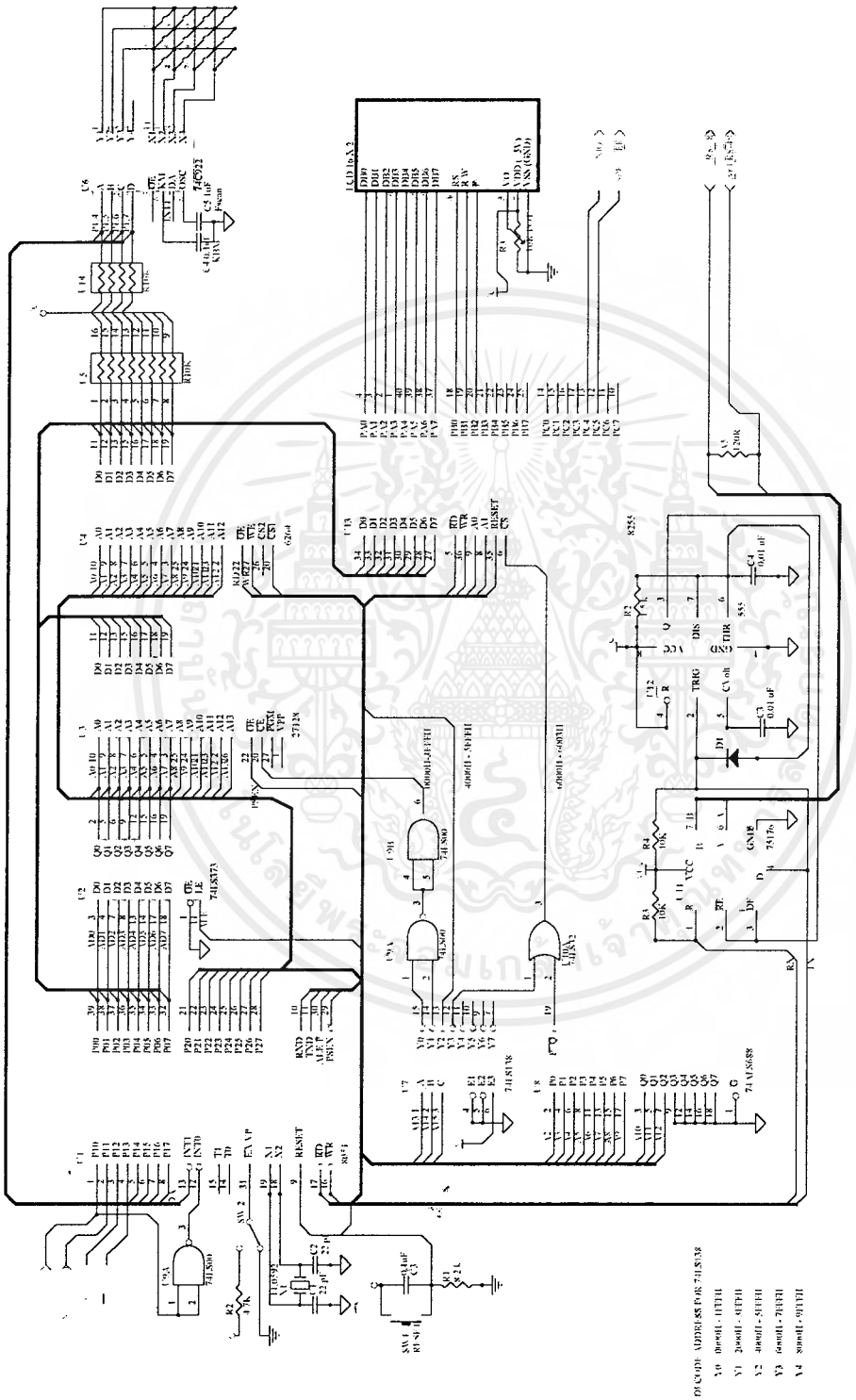


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ข

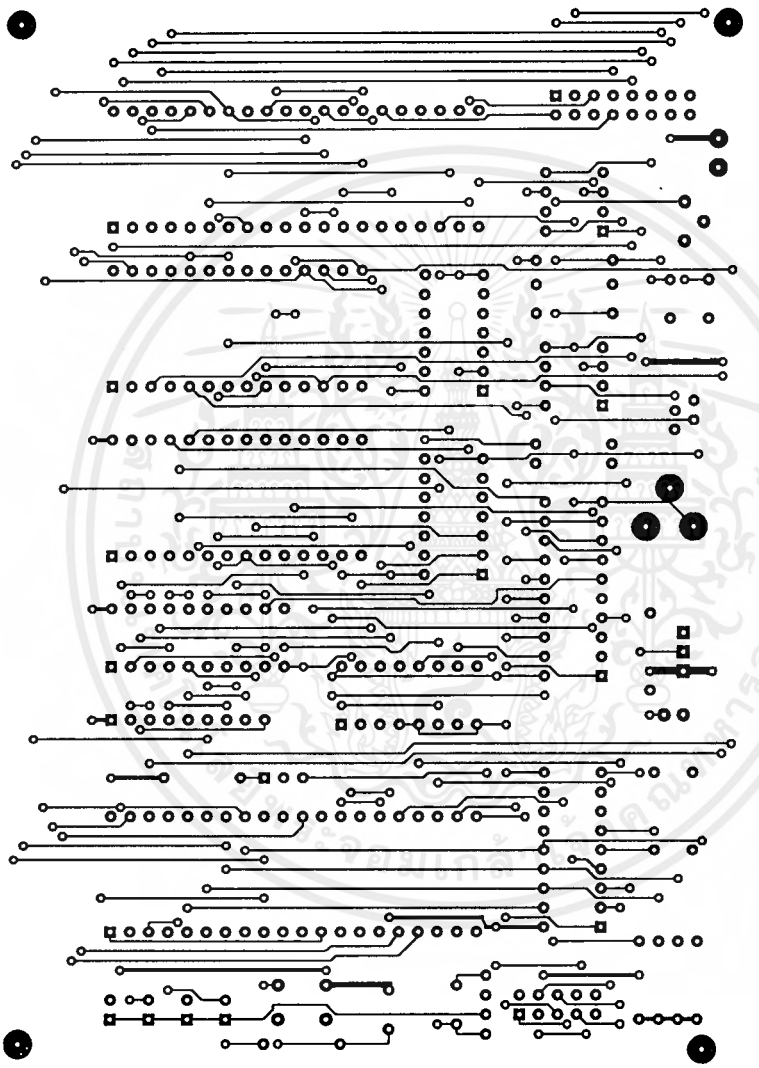
## วงจร แผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



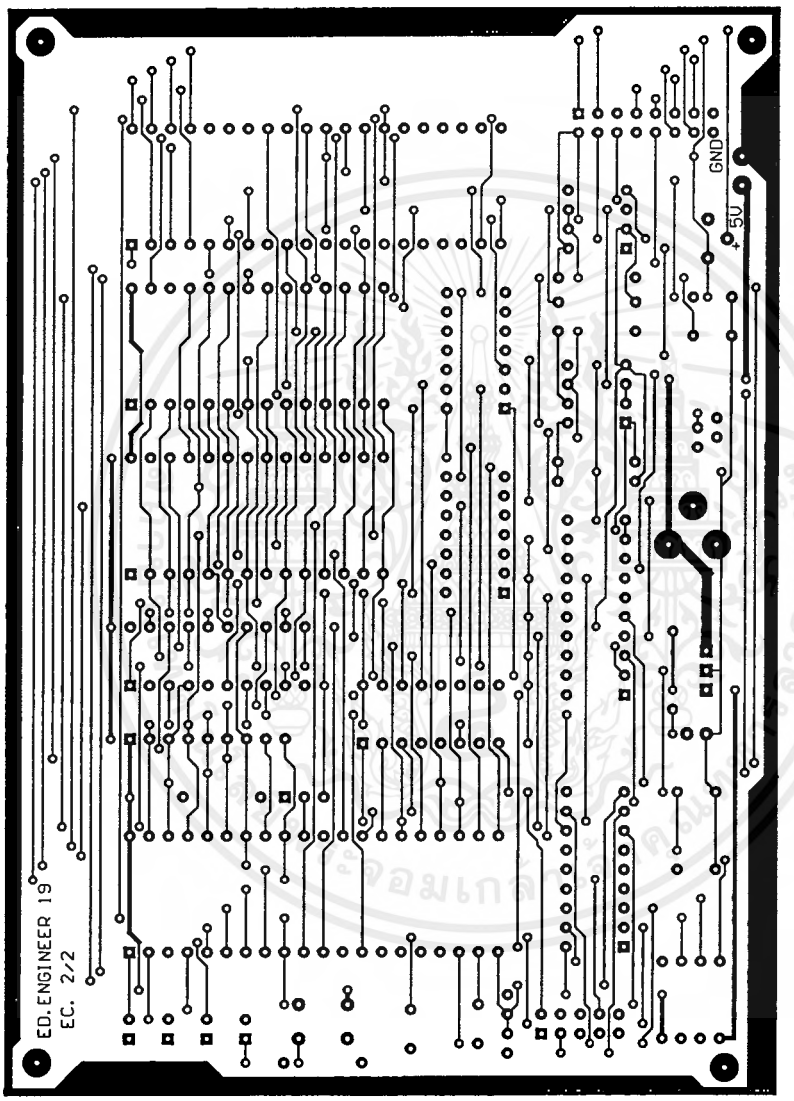
**รูปที่ 1 วงจรระบบตรวจสอบบุคคลเข้าหน่วยงาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



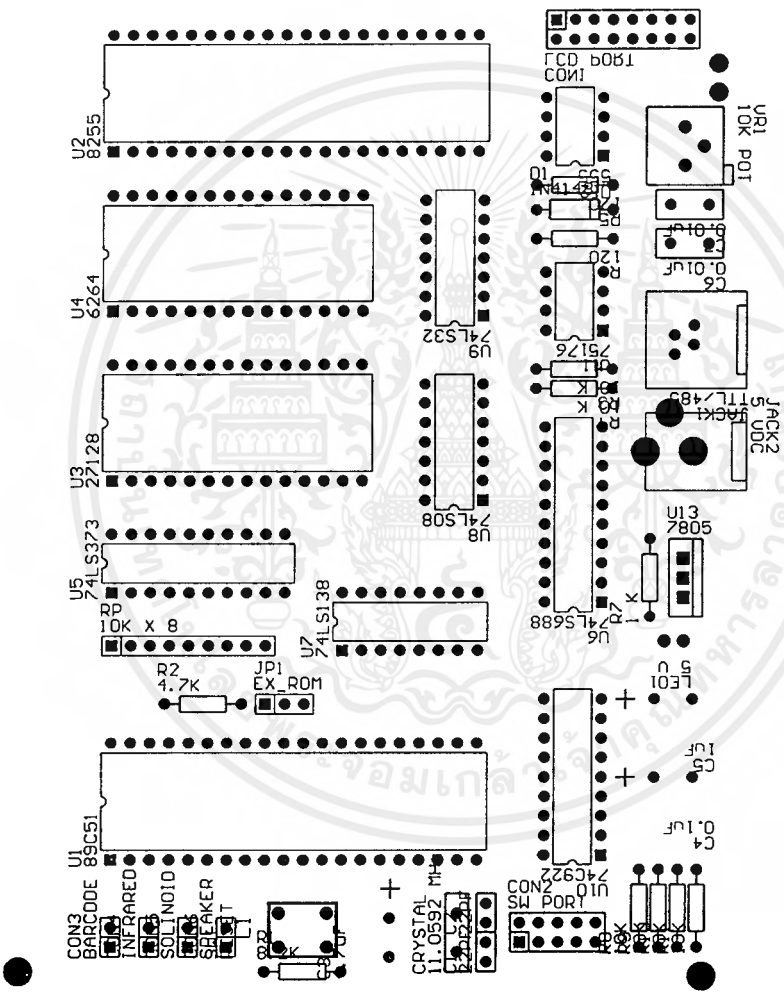
รูปที่ 1 ลายวงจรพิมพ์ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



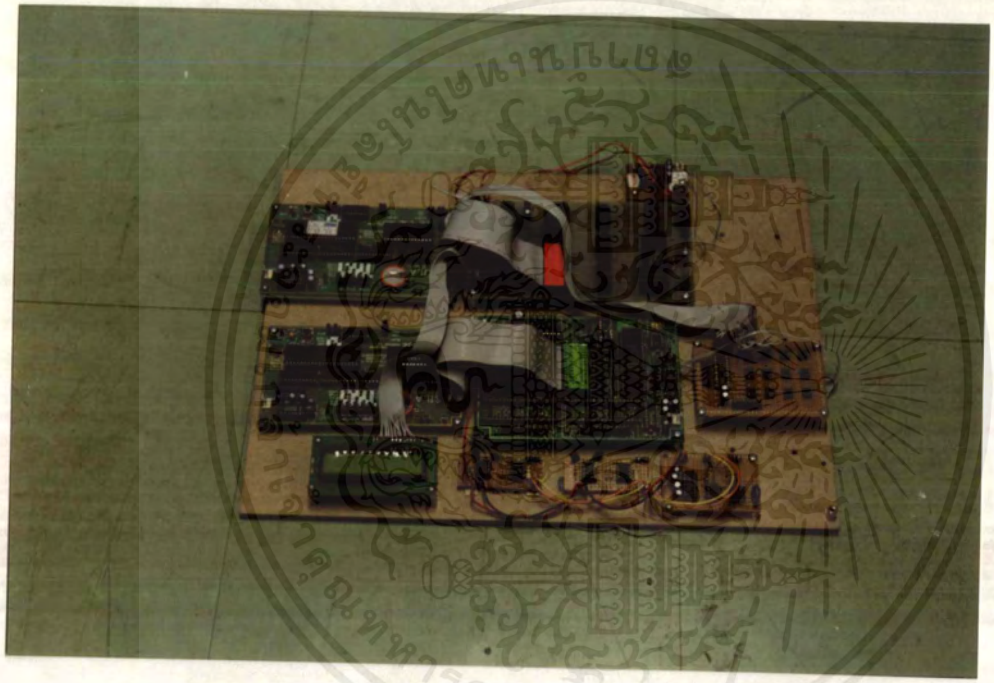

รูปที่ 2 ลายวงจรพิมพ์ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3 การวางอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 ระบบตรวจสอบบุคคลเข้าหน่วยงานของจริง

ลิขสิทธิ์มหาวิทยาลัยราชภัฏรำไพพรรณี
ระบบตรวจสอบบุคคลเข้าหน่วยงานโดยใช้ใบหน้าและลายนิ้วมือ กรณีศึกษา กองบัญชาการกองทัพไทย

# ภาคผนวก ค

# รายละเอียดข้อมูล และคุณสมบัติอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
  - Data Retention: 10 Years
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Five Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

# 8-Bit Microcontroller with 4 Kbytes Flash

# AT89C51

## Description

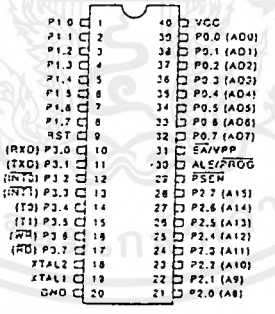
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and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

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 Bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five source two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is

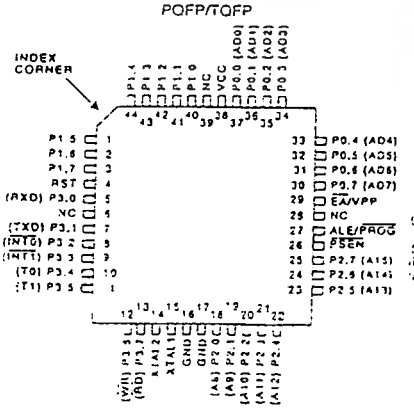
*continued*

## Pin Configurations

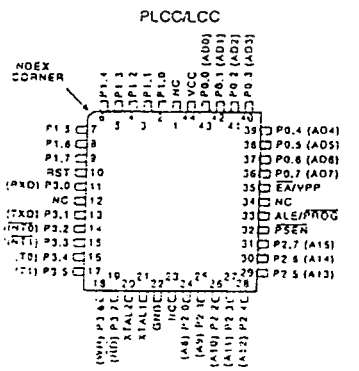
PDIP/Cerdip




PQFP/TQFP

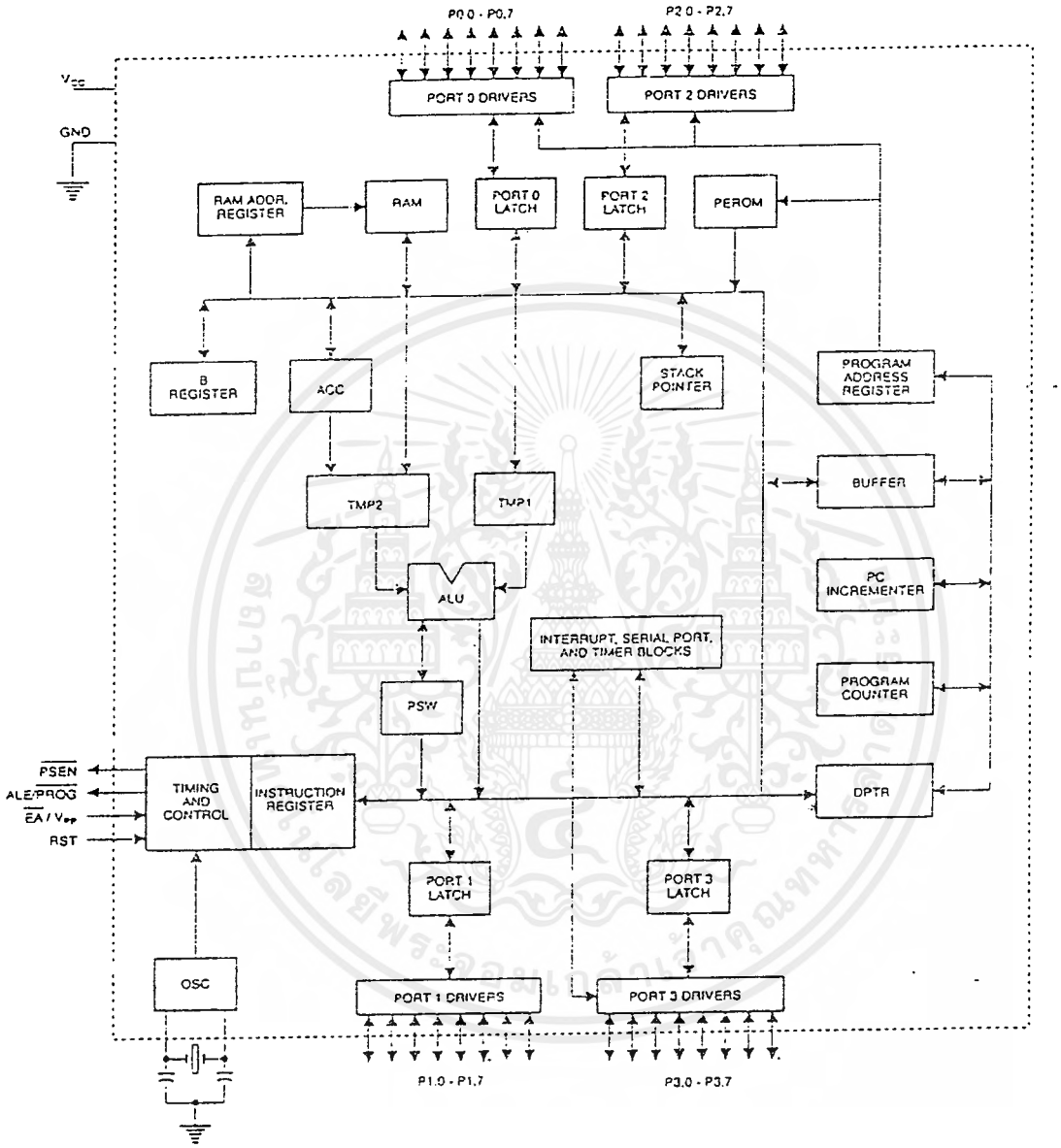



PLCC/LCC




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Block Diagram






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Description (Continued)

designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

## Pin Description

### $V_{CC}$

Supply voltage.

### GND

Ground.

### Port 0

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

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

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

### Port 1

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

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

### Port 2

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

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

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

### Port 3

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

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

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

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

### RST

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

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

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

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

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

### $\overline{PSEN}$

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

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

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

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

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

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

*continued*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Pin Description (Continued)

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

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

## Idle Mode

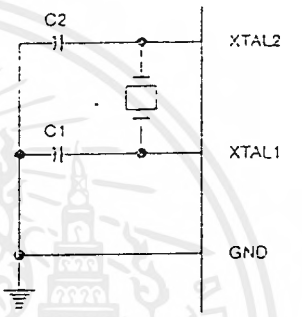
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

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

## Power Down Mode

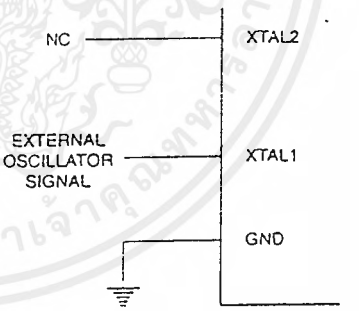
In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines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

Figure 1. Oscillator Connections




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

Figure 2. External Clock Drive Configuration




## Status of External Pins During Idle and Power Down

| Mode | Program Memory | ALE | $\overline{PSEN}$ | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power Down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power Down | External | 0 | 0 | Float | Data | Data | Data |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Program Memory Lock Bits

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

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

## Lock Bit Protection Modes

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

## Programming the Flash

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

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

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

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

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

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

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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

## Programming Interface

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

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

## Flash Programming Modes

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

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

Figure 3. Programming the Flash

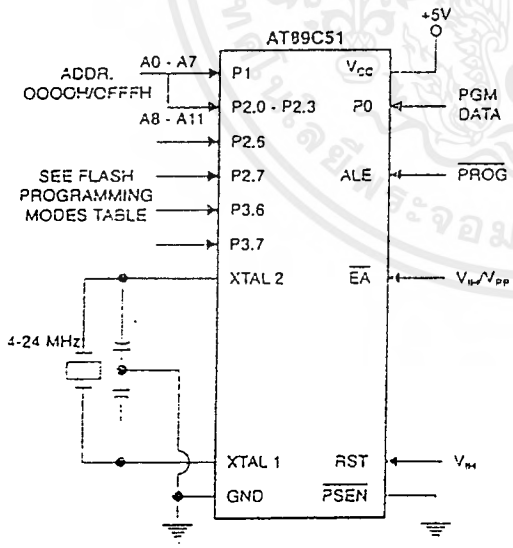
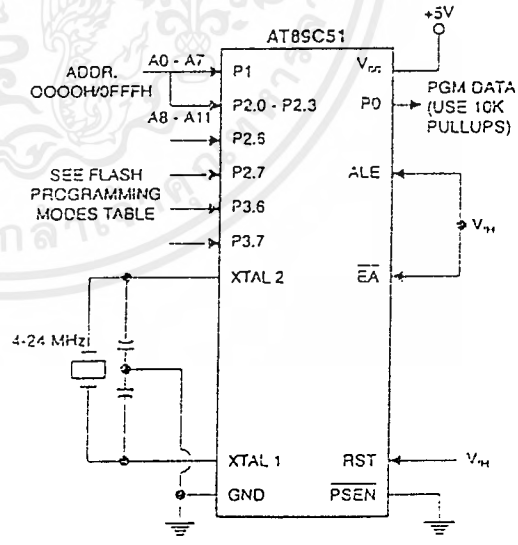



Figure 4. Verifying the Flash






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Flash Programming and Verification Characteristics

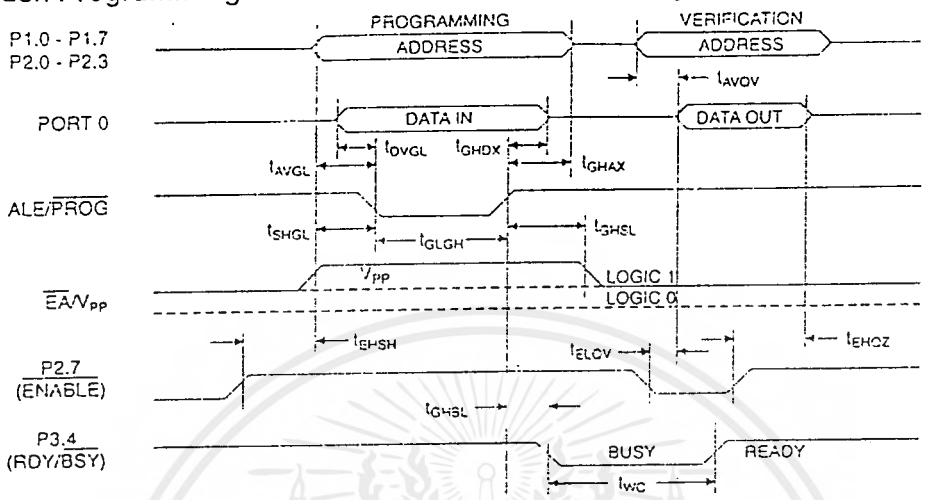
$T_A$ = 21°C to 27°C, $V_{CC}$ = 5.0 ± 10%

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $V_{PP}$[1] | Programming Enable Voltage | 11.5 | 12.5 | V |
| $I_{PP}$[1] | Programming Enable Current | | 1.0 | mA |
| $1/t_{CLCL}$ | Oscillator Frequency | 4 | 24 | MHz |
| $t_{AVGL}$ | Address Setup to $\overline{PROG}$ Low | 48$t_{CLCL}$ | | |
| $t_{GHAX}$ | Address Hold After $\overline{PROG}$ | 48$t_{CLCL}$ | | |
| $t_{DVGL}$ | Data Setup to $\overline{PROG}$ Low | 48$t_{CLCL}$ | | |
| $t_{GHDX}$ | Data Hold After $\overline{PROG}$ | 48$t_{CLCL}$ | | |
| $t_{EHSH}$ | P2.7 ($\overline{ENABLE}$) High to $V_{PP}$ | 48$t_{CLCL}$ | | |
| $t_{SHGL}$ | $V_{PP}$ Setup to $\overline{PROG}$ Low | 10 | | µs |
| $t_{GHSL}$[1] | $V_{PP}$ Hold After $\overline{PROG}$ | 10 | | µs |
| $t_{GLGH}$ | $\overline{PROG}$ Width | 1 | 110 | µs |
| $t_{AVQV}$ | Address to Data Valid | | 48$t_{CLCL}$ | |
| $t_{ELQV}$ | $\overline{ENABLE}$ Low to Data Valid | | 48$t_{CLCL}$ | |
| $t_{EHQV}$ | Data Float After $\overline{ENABLE}$ | 0 | 48$t_{CLCL}$ | |
| $t_{GHBL}$ | $\overline{PROG}$ High to $\overline{BUSY}$ Low | | 1.0 | µs |
| $t_{WC}$ | Byte Write Cycle Time | | 2.0 | ms |

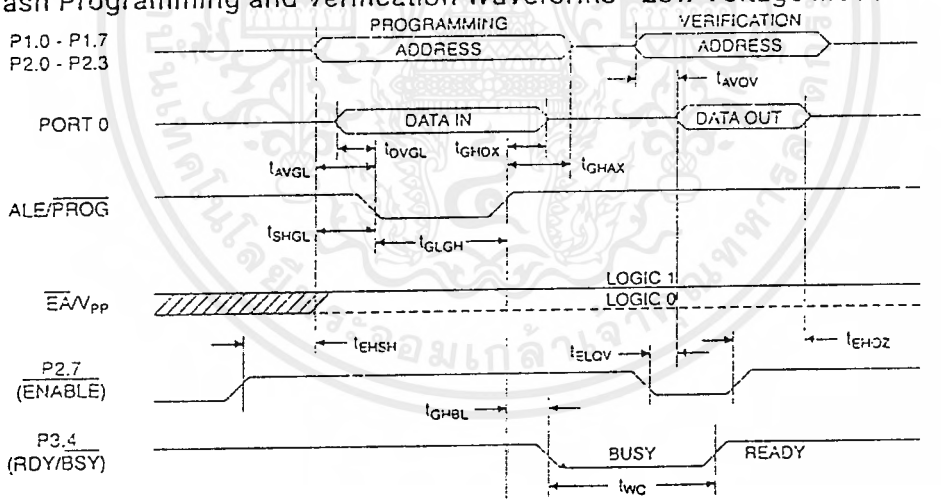
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Flash Programming and Verification Waveforms - High Voltage Mode




## Flash Programming and Verification Waveforms - Low Voltage Mode






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Absolute Maximum Ratings*

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

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

## D.C. Characteristics

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

| Symbol | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|
| $V_{IL}$ | Input Low Voltage | (Except $\overline{EA}$) | -0.5 | 0.2 $V_{CC}$-0.1 | V |
| $V_{IL1}$ | Input Low Voltage ($\overline{EA}$) | | -0.5 | 0.2 $V_{CC}$-0.3 | V |
| $V_{IH}$ | Input High Voltage | (Except XTAL1, RST) | 0.2 $V_{CC}$+0.9 | $V_{CC}$+0.5 | V |
| $V_{IH1}$ | Input High Voltage | (XTAL1, RST) | 0.7 $V_{CC}$ | $V_{CC}$+0.5 | V |
| $V_{OL}$ | Output Low Voltage[1] (Ports 1,2,3) | $I_{OL}$ = 1.6 mA | | 0.45 | V |
| $V_{OL1}$ | Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$) | $I_{OL}$ = 3.2 mA | | 0.45 | V |
| $V_{OH}$ | Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$) | $I_{OH}$ = -60 µA, $V_{CC}$ = 5 V ± 10% | 2.4 | | V |
| | | $I_{OH}$ = -25 µA | 0.75 $V_{CC}$ | | V |
| | | $I_{OH}$ = -10 µA | 0.9 $V_{CC}$ | | V |
| $V_{OH1}$ | Output High Voltage (Port 0 in External Bus Mode) | $I_{OH}$ = -800 µA, $V_{CC}$ = 5 V ± 10% | 2.4 | | V |
| | | $I_{OH}$ = -300 µA | 0.75 $V_{CC}$ | | V |
| | | $I_{OH}$ = -80 µA | 0.9 $V_{CC}$ | | V |
| $I_{IL}$ | Logical 0 Input Current (Ports 1,2,3) | $V_{IN}$ = 0.45 V | | -50 | µA |
| $I_{TL}$ | Logical 1 to 0 Transition Current (Ports 1,2,3) | $V_{IN}$ = 2 V | | -650 | µA |
| $I_{LI}$ | Input Leakage Current (Port 0, $\overline{EA}$) | 0.45 < $V_{IN}$ < $V_{CC}$ | | ±10 | µA |
| RRST | Reset Pulldown Resistor | | 50 | 300 | KΩ |
| $C_{IO}$ | Pin Capacitance | Test Freq. = 1 MHz, $T_A$ = 25°C | | 10 | pF |
| $I_{CC}$ | Power Supply Current | Active Mode, 12 MHz | | 20 | mA |
| | | Idle Mode, 12 MHz | | 5 | mA |
| | Power Down Mode[2] | $V_{CC}$ = 6 V | | 100 | µA |
| | | $V_{CC}$ = 3 V | | 40 | µA |

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## A.C. Characteristics

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

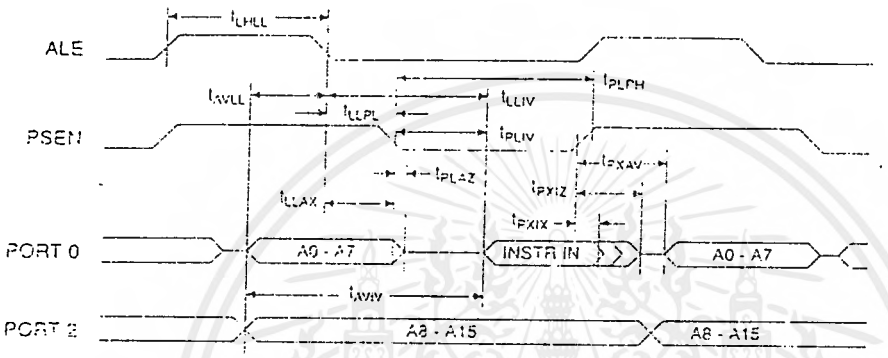
### External Program and Data Memory Characteristics

| Symbol | Parameter | 12 MHz Oscillator | | 16 to 24 MHz Oscillator | | Units |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| $1/t_{CLCL}$ | Oscillator Frequency | | | 0 | 24 | MHz |
| $t_{LHLL}$ | ALE Pulse Width | 127 | | $2t_{CLCL}-40$ | | ns |
| $t_{AVLL}$ | Address Valid to ALE Low | 28 | | $t_{CLCL}-13$ | | ns |
| $t_{LLAX}$ | Address Hold After ALE Low | 48 | | $t_{CLCL}-20$ | | ns |
| $t_{LLIV}$ | ALE Low to Valid Instruction In | | 233 | | $4t_{CLCL}-65$ | ns |
| $t_{LLPL}$ | ALE Low to $\overline{PSEN}$ Low | 43 | | $t_{CLCL}-13$ | | ns |
| $t_{PLPH}$ | $\overline{PSEN}$ Pulse Width | 205 | | $3t_{CLCL}-20$ | | ns |
| $t_{PLIV}$ | $\overline{PSEN}$ Low to Valid Instruction In | | 145 | | $3t_{CLCL}-45$ | ns |
| $t_{PXIX}$ | Input Instruction Hold After $\overline{PSEN}$ | 0 | | 0 | | ns |
| $t_{PXIZ}$ | Input Instruction Float After $\overline{PSEN}$ | | 59 | | $t_{CLCL}-10$ | ns |
| $t_{PXAV}$ | $\overline{PSEN}$ to Address Valid | 75 | | $t_{CLCL}-8$ | | ns |
| $t_{AVIV}$ | Address to Valid Instruction In | | 312 | | $5t_{CLCL}-55$ | ns |
| $t_{PLAZ}$ | $\overline{PSEN}$ Low to Address Float | | 10 | | 10 | ns |
| $t_{RLRH}$ | $\overline{RD}$ Pulse Width | 400 | | $6t_{CLCL}-100$ | | ns |
| $t_{WLWH}$ | $\overline{WR}$ Pulse Width | 400 | | $6t_{CLCL}-100$ | | ns |
| $t_{RLDV}$ | $\overline{RD}$ Low to Valid Data In | | 252 | | $5t_{CLCL}-90$ | ns |
| $t_{RHDX}$ | Data Hold After $\overline{RD}$ | 0 | | 0 | | ns |
| $t_{RHDZ}$ | Data Float After $\overline{RD}$ | | 97 | | $2t_{CLCL}-28$ | ns |
| $t_{LLDV}$ | ALE Low to Valid Data In | | 517 | | $8t_{CLCL}-150$ | ns |
| $t_{AVDV}$ | Address to Valid Data In | | 565 | | $9t_{CLCL}-165$ | ns |
| $t_{LLWL}$ | ALE Low to $\overline{RD}$ or $\overline{WR}$ Low | 200 | 300 | $3t_{CLCL}-50$ | $3t_{CLCL}+50$ | ns |
| $t_{AVWL}$ | Address to $\overline{RD}$ or $\overline{WR}$ Low | 203 | | $4t_{CLCL}-75$ | | ns |
| $t_{QVWX}$ | Data Valid to $\overline{WR}$ Transition | 23 | | $t_{CLCL}-20$ | | ns |
| $t_{QVWH}$ | Data Valid to $\overline{WR}$ High | 433 | | $7t_{CLCL}-120$ | | ns |
| $t_{WHQX}$ | Data Hold After $\overline{WR}$ | 33 | | $t_{CLCL}-20$ | | ns |
| $t_{RLAZ}$ | $\overline{RD}$ Low to Address Float | | 0 | | 0 | ns |
| $t_{WHLH}$ | $\overline{RD}$ or $\overline{WR}$ High to ALE High | 43 | 123 | $t_{CLCL}-20$ | $t_{CLCL}+25$ | ns |

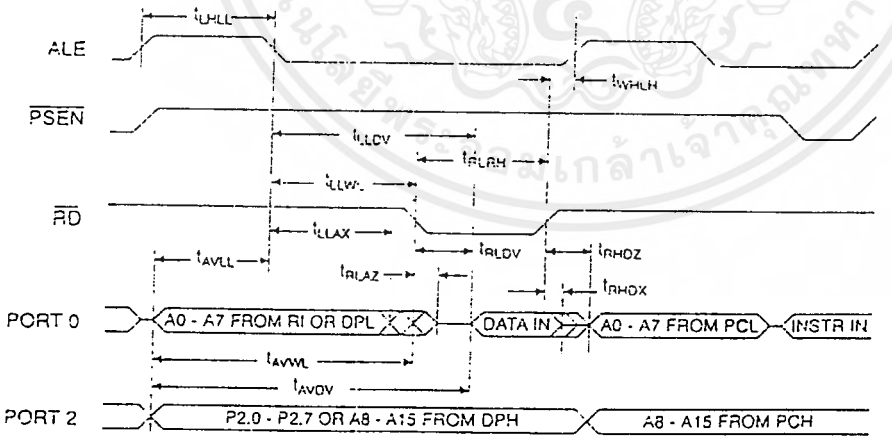


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## External Program Memory Read Cycle

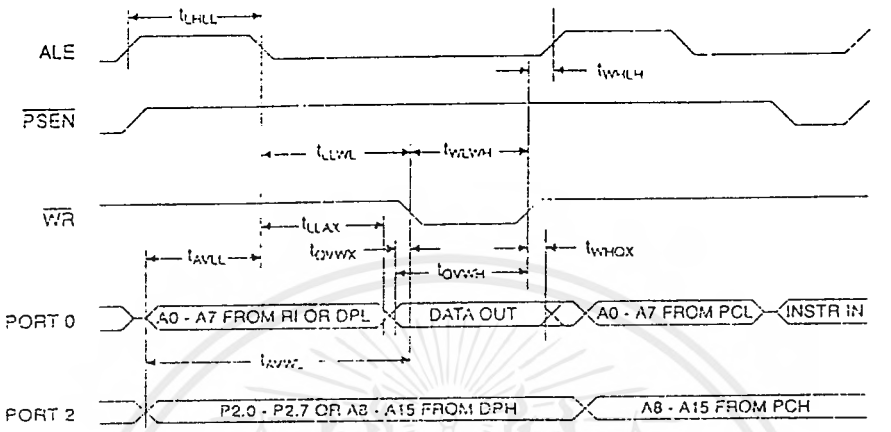



## External Data Memory Read Cycle

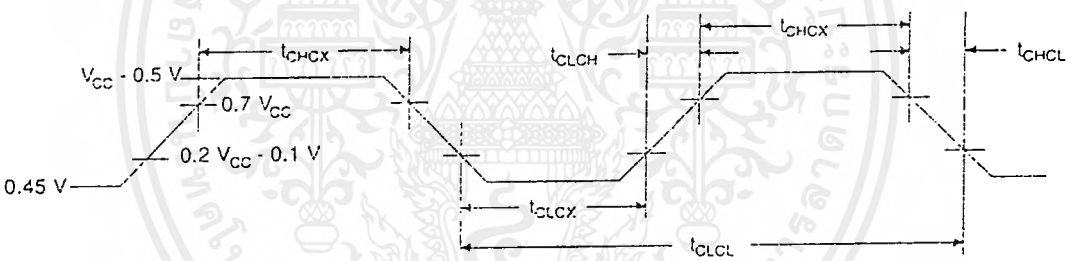



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## External Data Memory Cycle




## External Clock Drive Waveforms




## External Clock Drive

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| 1/$t_{CLCL}$ | Oscillator Frequency | 0 | 24 | MHz |
| $t_{CLCL}$ | Clock Period | 41.5 | | ns |
| $t_{CHCX}$ | High Time | 15 | | ns |
| $t_{CLCX}$ | Low Time | 15 | | ns |
| $t_{CLCH}$ | Rise Time | | 20 | ns |
| $t_{CHCL}$ | Fall Time | | 20 | ns |



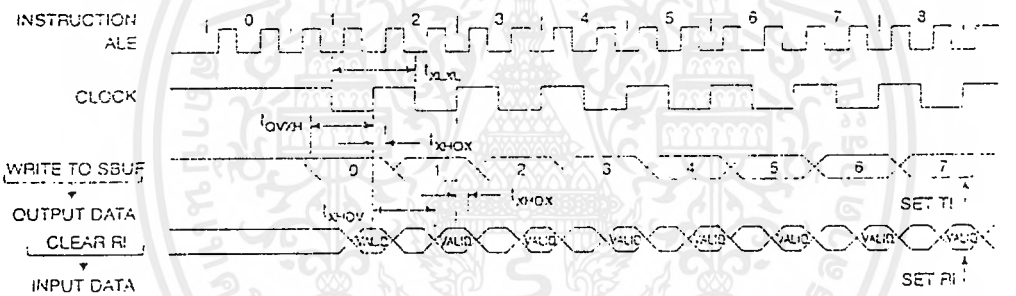
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Serial Port Timing: Shift Register Mode Test Conditions

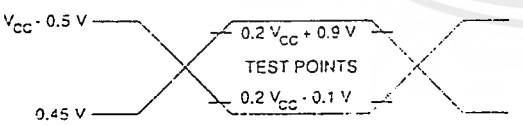
($V_{CC}$ = 5.0 V ± 20%; Load Capacitance = 80 pF)

| Symbol | Parameter | 12 MHz Osc | | Variable Oscillator | | Units |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| $t_{XLXL}$ | Serial Port Clock Cycle Time | 1.0 | | $12t_{CLCL}$ | | µs |
| $t_{QVXH}$ | Output Data Setup to Clock Rising Edge | 700 | | $10t_{CLCL}$-133 | | ns |
| $t_{XHQX}$ | Output Data Hold After Clock Rising Edge | 50 | | $2t_{CLCL}$-33 | | ns |
| $t_{XHDX}$ | Input Data Hold After Clock Rising Edge | 0 | | 0 | | ns |
| $t_{XHDV}$ | Clock Rising Edge to Input Data Valid | | 700 | | $10t_{CLCL}$-133 | ns |

## Shift Register Mode Timing Waveforms

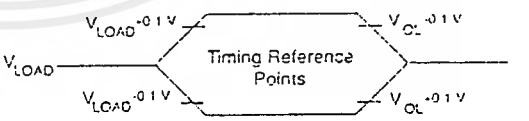



## AC Testing Input/Output Waveforms[1]




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

## Float Waveforms[1]

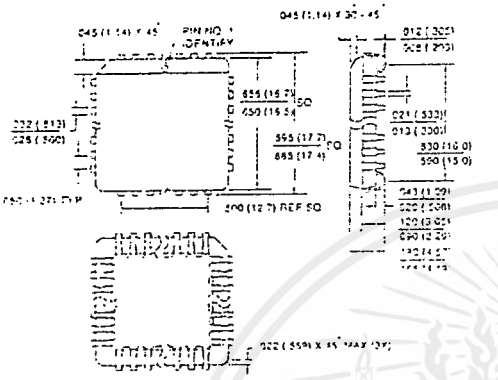



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded $V_{OH}/V_{OL}$ level occurs.

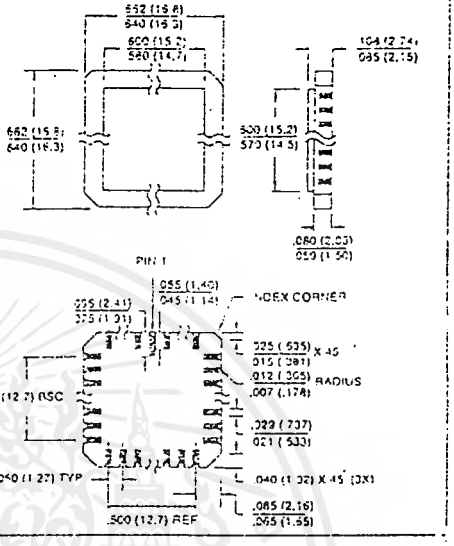
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Packaging Information

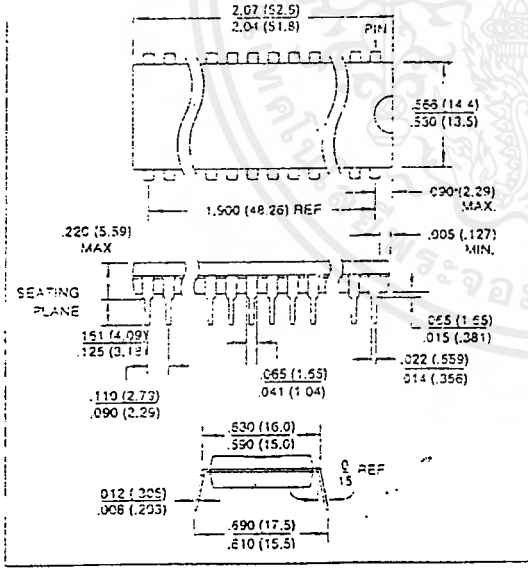
44J, 44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC OUTLINE MO-047 AC



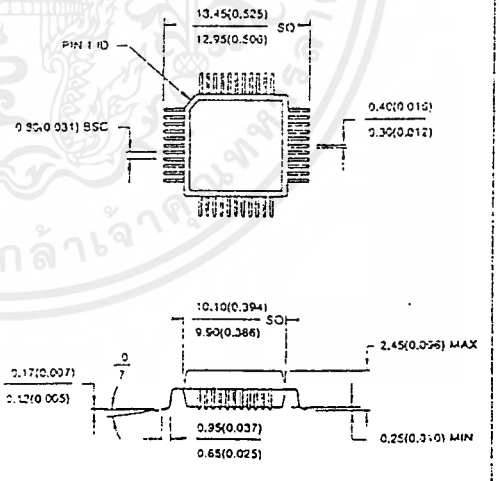
44L, 44 Pad, Non-Windowed,
Ceramic Leadless Chip Carrier (LCC)
Dimensions in Inches and (Millimeters)



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

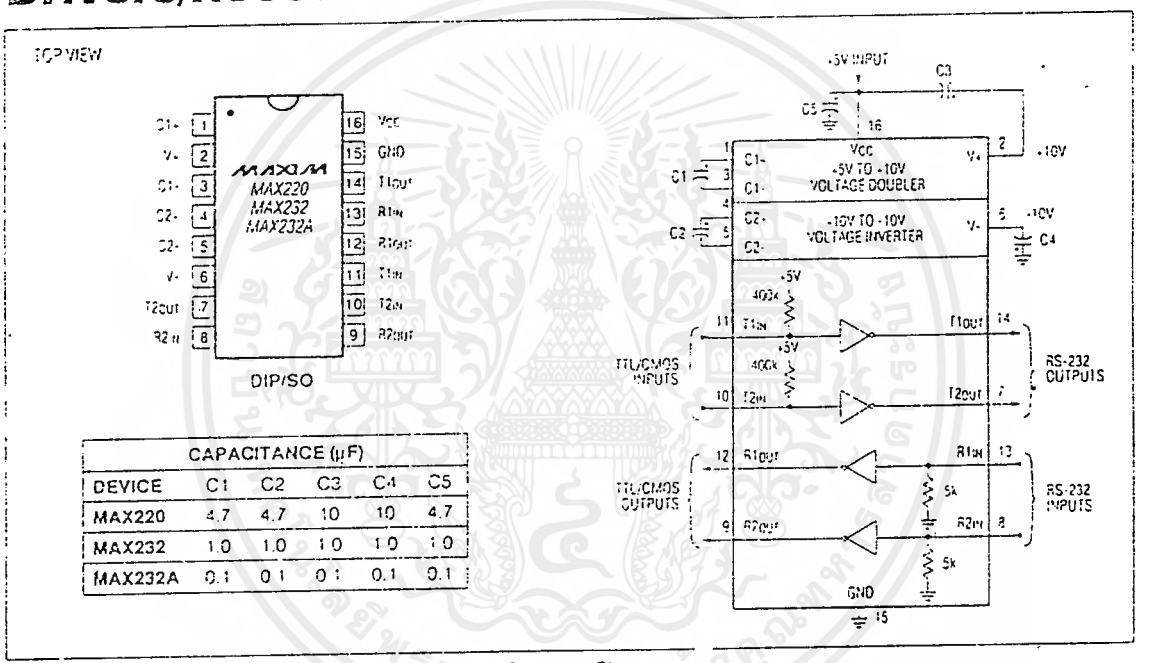


44Q, 44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)
Dimensions in Millimeters and (Inches)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# +5V-Powered, Multi-Channel RS-232 Drivers/Receivers




| CAPACITANCE (µF) | | | | | |
|---|---|---|---|---|---|
| DEVICE | C1 | C2 | C3 | C4 | C5 |
| MAX220 | 4.7 | 4.7 | 10 | 10 | 4.7 |
| MAX232 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| MAX232A | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |

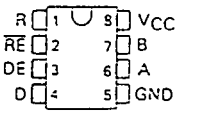
*MAX220/232/232A Pin Configuration and Typical Operating Circuit*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# SN75176A
# DIFFERENTIAL BUS TRANSCEIVER

- Bidirectional Transceiver
- Meets EIA Standards RS-422A and CCITT Recommendations V.11 and X.27
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Driver and Receiver Outputs
- Individual Driver and Receiver Enables
- Wide Positive and Negative Input/Output Bus Voltage Ranges
- Driver Output Capability. . . ±60 mA Max
- Thermal Shutdown Protection
- Driver Positive and Negative Current Limiting
- Receiver Input Impedance . . . 12 kΩ Min
- Receiver Input Sensitivity . . . ±200 mV
- Receiver Input Hysteresis . . . 50 mV Typ
- Operates from Single 5-V Supply
- Low Power Requirements

D OR P
DUAL-IN-LINE PACKAGE
(TOP VIEW)




FUNCTION TABLE (DRIVER)

| INPUT | ENABLE | OUTPUTS | |
|---|---|---|---|
| D | DE | A | B |
| H | H | H | L |
| L | H | L | H |
| X | L | Z | Z |

FUNCTION TABLE (RECEIVER)

| DIFFERENTIAL INPUTS | ENABLE | OUTPUT |
|---|---|---|
| A − B | $\overline{RE}$ | R |
| $V_{ID} > 0.2$ V | L | H |
| $-0.2$ V $< V_{ID} < 0.2$ V | L | ? |
| $V_{ID} < -0.2$ V | L | L |
| X | H | Z |

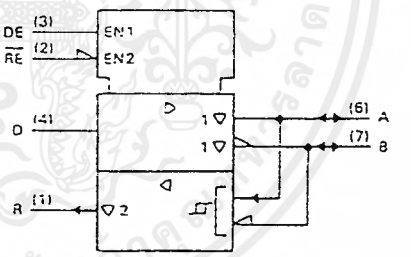
H = high level, L = low level, ? = indeterminate,
X = irrelevant, Z = high impedance (off)

## description

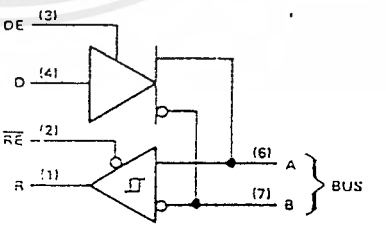
The SN75176A differential bus transceiver is a monolithic integrated circuit designed for bidirectional data communication on multipoint bus transmission lines. It is designed for balanced transmission lines and meets EIA Standard RS-422A and CCITT Recommendations V.11 and X.27.

The SN75176A combines a 3-state differential line driver and a differential-input line receiver both of which operate from a single 5-V power supply. The driver and receiver have active-high and active-low enables, respectively, that can be externally connected together to function as direction control. The driver differential outputs and the receiver differential inputs are connected internally to form differential input/output (I/O) bus ports that are designed to offer minimum loading to the bus whenever the driver is disabled or $V_{CC}$ = 0. These ports feature wide positive and negative common-mode voltage ranges making the device suitable for party-line applications.

## logic symbol




## logic diagram (positive logic)



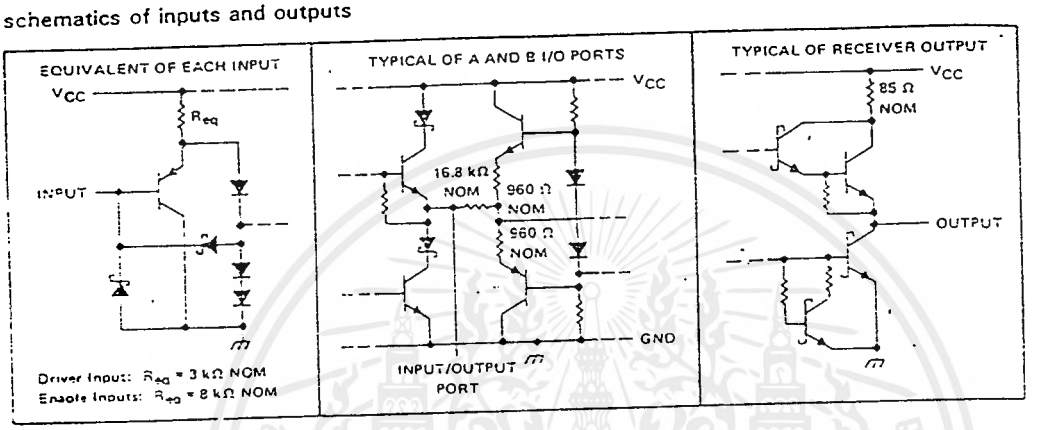

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

description (continued)

The driver is designed to handle loads up to 60 mA of sink or source current. The driver features positive- and negative-current limiting and thermal shutdown for protection from line fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 150°C. The receiver features a minimum input impedance of 12 kΩ, input sensitivity of ±200 mV, and a typical input hysteresis of 50 mV.

The SN75176A can be used in transmission line applications employing the SN75172 and SN75174 quadruple differential line drivers and the SN75173 and SN75175 quadruple differential line receivers.

schematics of inputs and outputs




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

Supply voltage, $V_{CC}$ (see Note 1) . . . . . 7 V
Voltage at any bus terminal . . . . . −10 V to 15 V
Enable input voltage . . . . . 5.5 V
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2):
D package . . . . . 725 mW
P package . . . . . 1000 mW
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . . . 260°C

NOTES: 1. All voltage values, except differential input/output bus voltage, are with respect to network ground terminal.
2. For operation above 25°C free-air temperature, derate the D package to 464 mW at 70°C at the rate of 5.8 mW/°C and derate the P package to 640 mW at 70°C at the rate of 8.0 mW/°C.

recommended operating conditions

| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| Supply voltage, $V_{CC}$ | | 4.75 | 5 | 5.25 | V |
| Voltage at any bus terminal (separately or common-mode), $V_I$ or $V_{IC}$ | | −7 | | 12 | V |
| High-level input voltage, $V_{IH}$ | D, DE, and $\overline{RE}$ | 2 | | | V |
| Low-level input voltage, $V_{IL}$ | D, DE, and $\overline{RE}$ | | | 0.8 | V |
| Differential input voltage, $V_{ID}$ (see Note 3) | | | | ±12 | V |
| High-level output current, $I_{OH}$ | Driver | | | −60 | mA |
| | Receiver | | | −400 | μA |
| Low-level output current, $I_{OL}$ | Driver | | | 60 | mA |
| | Receiver | | | 8 | |
| Operating free-air temperature, $T_A$ | | 0 | | 70 | °C |

NOTE 3: Differential-input/output bus voltage is measured at the noninverting terminal A with respect to the inverting terminal B.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DRIVER SECTION

driver electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted).

| PARAMETER | | TEST CONDITIONS | | MIN | TYP† | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $V_{IK}$ | Input clamp voltage | $I_I$ = −18 mA | | | | −1.5 | V |
| $V_{OH}$ | High-level output voltage | $V_{IH}$ = 2 V, $I_{OH}$ = −33 mA | $V_{IL}$ = 0.8 V, | | 3.7 | | V |
| $V_{OL}$ | Low-level output voltage | $V_{IH}$ = 2 V, $I_{OL}$ = 33 mA | $V_{IL}$ = 0.8 V, | | 1.1 | | V |
| $\|V_{OD1}\|$ | Differential output voltage | $I_O$ = 0 | | | | $\frac{1}{2}V_{OD2}$ | V |
| $\|V_{OD2}\|$ | Differential output voltage | $R_L$ = 100 Ω, | See Figure 1 | 2 | 2.7 | | V |
| | | $R_L$ = 54 Ω, | See Figure 1 | 1.5 | 2.4 | | |
| $\Delta\|V_{OD}\|$ | Change in magnitude of differential output voltage‡ | | | | | ±0.2 | V |
| $V_{OC}$ | Common-mode output voltage§ | $R_L$ = 54 Ω or 100 Ω, | See Figure 1 | | | 3 | V |
| $\Delta\|V_{OC}\|$ | Change in magnitude of common-mode output voltage‡ | | | | | ±0.2 | V |
| $I_O$ | Output current | Output disabled, See Note 4 | $V_O$ = 12 V | | | 1 | mA |
| | | | $V_O$ = −7 V | | | −0.8 | |
| $I_{IH}$ | High-level input current | $V_I$ = 2.4 V | | | | 20 | μA |
| $I_{IL}$ | Low-level input current | $V_I$ = 0.4 V | | | | −400 | μA |
| $I_{OS}$ | Short-circuit output current | $V_O$ = −7 V | | | | −250 | mA |
| | | $V_O$ = $V_{CC}$ | | | | 250 | |
| | | $V_O$ = 12 V | | | | 500 | |
| $I_{CC}$ | Supply current (total package) | No load | Outputs enabled | | 38 | 60 | mA |
| | | | Outputs disabled | | 26 | 40 | |

† All typical values are at $V_{CC}$ = 5 V and $T_A$ = 25°C.
‡ $\Delta|V_{OD}|$ and $\Delta|V_{OC}|$ are the changes in magnitude of $V_{OD}$ and $V_{OC}$ respectively, that occur when the input is changed from a high level to a low level.
§ In EIA Standard RS-422A, $V_{OC}$, which is the average of the two output voltages with respect to ground, is called output offset voltage, $V_{OS}$.
NOTE 4: This applies for both power on and power off. Refer to EIA Standard RS-422A for exact conditions.

driver switching characteristics, $V_{CC}$ = 5 V, $T_A$ = 25°C

| PARAMETER | | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $t_{DD}$ | Differential-output delay time | $R_L$ = 60 Ω, | See Figure 3 | | 40 | 60 | ns |
| $t_{TD}$ | Differential-output transition time | | | | 55 | 95 | ns |
| $t_{PZH}$ | Output enable time to high level | $R_L$ = 110 Ω, | See Figure 4 | | 55 | 80 | ns |
| $t_{PZL}$ | Output enable time to low level | $R_L$ = 110 Ω, | See Figure 5 | | 30 | 50 | ns |
| $t_{PHZ}$ | Output disable time from high level | $R_L$ = 110 Ω, | See Figure 4 | | 25 | 130 | ns |
| $t_{PLZ}$ | Output disable time from low level | $R_L$ = 110 Ω, | See Figure 5 | | 20 | 40 | ns |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้แต่ง

| | |
|---|---|
| ชื่อผู้ทำปริญญานิพนธ์ | นายกิติพงศ์ พรหมเลิศ |
| วันเดือนปีเกิด | 8 มกราคม 2520 |
| สถานที่เกิด | จังหวัดสุราษฎร์ธานี |
| ภูมิลำเนา | 107 หมู่ 8 ต.กะแดะ อ.กาญจนดิษฐ์ |
| | จ.สุราษฎร์ธานี 84160 |
| ที่อยู่ปัจจุบัน | 107 หมู่ 8 ต.กะแดะ อ.กาญจนดิษฐ์ |
| | จ.สุราษฎร์ธานี 84160 |
| โทรศัพท์ | 074-274494 |


**ประวัติการศึกษา**

| | |
|---|---|
| ประถมศึกษา | โรงเรียนบ้านปากกะแดะ |
| มัธยมศึกษาตอนต้น | โรงเรียนกาญจนดิษฐ์ |
| ประกาศนียบัตรวิชาชีพ(ปวช.) | วิทยาลัยเทคนิคสุราษฎร์ธานี |
| ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.) | วิทยาลัยเทคนิคหาดใหญ่ |

| | |
|---|---|
| ปริญญาตรี | สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ |
| | ภาควิชาครุศาสตร์วิศวกรรม |
| | คณะครุศาสตร์อุตสาหกรรม |
| ผลงานที่ได้รับรางวัล | |
| คติพจน์ | นำอดีตเป็นเครื่องมือเพื่อทำปัจจุบันให้มี |
| | อนาคตตามที่มุ่งหวัง |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้แต่ง

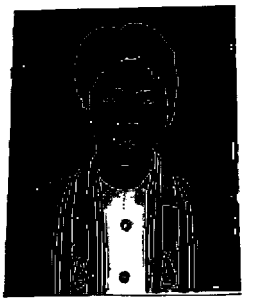
| | |
|---|---|
| ชื่อผู้ทำปริญญานิพนธ์ | นายโกมิน สัตย์มั่น |
| วันเดือนปีเกิด | 16 กรกฎาคม 2519 |
| สถานที่เกิด | จังหวัดภูเก็ต |
| ภูมิลำเนา | 78/31 ถ.ศักดิเดช ซ.นานาชาติ หมู่ 6 อ.เมือง จ.ภูเก็ต 83000 |
| ที่อยู่ปัจจุบัน | 78/31 ถ.ศักดิเดช ซ.นานาชาติ หมู่ 6 อ.เมือง จ.ภูเก็ต 83000 |
| โทรศัพท์ | 076-200505 |

**ประวัติการศึกษา**

| | |
|---|---|
| ประถมศึกษา | โรงเรียนอนุบาลภูเก็ต |
| มัธยมศึกษาตอนต้น | โรงเรียนภูเก็ตวิทยาลัย |
| ประกาศนียบัตรวิชาชีพ(ปวช) | วิทยาลัยเทคนิคภูเก็ต |
| ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส) | วิทยาลัยเทคนิคภูเก็ต |
| ปริญญาตรี | สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม |
| ผลงานที่ได้รับรางวัล | |
| คติพจน์ | ชีวิตไม่สำเร็จรูปเหมือนบะหมี่ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้แต่ง



| | |
|---|---|
| ชื่อผู้ทำปริญญานิพนธ์ | นายชิษณุพงศ์ ธัญญลักษณ์ |
| วันเดือนปีเกิด | 21 มกราคม 2520 |
| สถานที่เกิด | จังหวัดนครศรีธรรมราช |
| ภูมิลำเนา | 2/4 หมู่ 3 ต.มะขามเตี้ย อ.เมือง จ.สุราษฎร์ธานี 84000 |
| ที่อยู่ปัจจุบัน | 2/4 หมู่ 3 ต.มะขามเตี้ย อ.เมือง จ.สุราษฎร์ธานี 84000 |
| โทรศัพท์ | |

**ประวัติการศึกษา**

| | |
|---|---|
| ประถมศึกษา | โรงเรียนเทพมิตรศึกษา |
| มัธยมศึกษาตอนต้น | โรงเรียนประดู่วิทยาคม |
| ประกาศนียบัตรวิชาชีพ(ปวช) | วิทยาลัยเทคนิคสุราษฎร์ธานี |
| ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส) | วิทยาลัยเทคนิคหาดใหญ่ |
| ปริญญาตรี | สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม |

ผลงานที่ได้รับรางวัล

คติพจน์ สังคมเลวร้ายขึ้น เพราะคนดีท้อใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

1. สมพันธ์ ชาญศิลป์, " ภาษาซี " , คณะวิศกรรมศาสตร์ มหาวิทยาลัยขอนแก่น, 2537

2. รศ.สมยศ จุณณะปิยะ, " การประยุกต์ใช้งานไมโครคอนโทรเลอร์ตระกูล MSC-51 ",คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีเจ้าคุณทหารลาดกระบัง, 2541

3. Microcontroller AT89C51 Device Data, 1987

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้