

**ภาควิชาครุศาสตร์วิศวกรรม**  
**คณะครุศาสตร์อุตสาหกรรม**  
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
**ใบรับรองปริญญานิพนธ์**

หัวข้อข้อปริญญานิพนธ์ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

8051 SINGLE BOARD MICROCONTROLLER

|              |                 |           |              |          |
|--------------|-----------------|-----------|--------------|----------|
| ชื่อนักศึกษา | 1. นายจิตติพงษ์ | พิสิฐนฤดม | รหัสประจำตัว | 40031411 |
|              | 2. นายวาทิน     | ทิพนงค์   | รหัสประจำตัว | 40031426 |
|              | 3. นายศราวุธ    | สุชีฐาน   | รหัสประจำตัว | 40031428 |
|              | 4. นายสืบศักดิ์ | สุนิพันธ์ | รหัสประจำตัว | 40031434 |

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ผู้ควบคุมปริญญานิพนธ์

|                   |                 |
|-------------------|-----------------|
| 1. อาจารย์ไพบูลย์ | พวงวงศ์ตระกูล   |
| 2. อาจารย์ปิยะ    | จิตรธรรมาภิรมย์ |
| 3. อาจารย์รววิทย์ | สมหา            |



| คณะกรรมการสอบปริญญานิพนธ์ |               | ลายมือชื่อ |
|---------------------------|---------------|------------|
| 1. อาจารย์กิตติพงศ์       | มะโน          |            |
| 2. อาจารย์อำพล            | ทองระอา       |            |
| 3. อาจารย์ไพบูลย์         | พวงวงศ์ตระกูล |            |
| 4. อาจารย์สุรพงษ์         | สิริพงศ์ดี    |            |
| 5. อาจารย์สุระชัย         | พิมสาลี       |            |

วัน/เดือน/ปี ที่สอบ วันอาทิตย์ที่ 22 พฤศจิกายน พ.ศ. 2541 เวลา 13.00-14.00 น.

สถานที่สอบ ห้อง ค.310 คณะครุศาสตร์อุตสาหกรรม สจล.

นพ.  
จ 339 ๗  
2541



ภาควิชารับรองแล้ว

ผู้อำนวยการฯ (ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

ภาควิชาครุศาสตร์วิศวกรรม

คณาจารย์ผู้ควบคุม

เดือน... พ.ศ....

เลขหมู่.....

เลขทะเบียน..... 32830

วัน, เดือน, ปี 110 ส.ย. 2542

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ปริญญานิพนธ์

เรื่อง 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว  
8051 SINGLE BOARD MICROCONTROLLER

### ผู้จัดทำ

1. นายฐิติพงษ์ พิสิฐนฤดม
2. นายวาทิน ทิพนงค์
3. นายศรารุท สุชีวาน
4. นายสืบศักดิ์ สุนิพันธ์

### อาจารย์ที่ปรึกษา

ลงนาม.....  
(อาจารย์ไพฑูริย์ พวงวงศ์ตระกูล)

ลงนาม.....  
(อาจารย์ปิยะ จิตธรรมมาภิรมย์)

ลงนาม.....  
(อาจารย์วรวิทย์ สมหา)

### หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....

เอกสารนี้เป็น (ผศ.ดร.ธีระพล) แห่งมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ปริญญานิพนธ์

เรื่อง 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว  
8051 SINGLE BOARD MICROCONTROLLER

### วัตถุประสงค์

1. เพื่อศึกษาระบบการทำงานของไมโครคอนโทรลเลอร์ ตระกูล MCS-51
2. ออกแบบแผ่นพิมพ์เดี่ยว (ลายวงจรพิมพ์) ของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ได้
3. สร้างแผ่นพิมพ์เดี่ยวของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ได้
4. ศึกษาการเขียนโปรแกรมควบคุมระบบการทำงานของชิพไมโครคอนโทรลเลอร์ ตระกูล MCS-51
5. สามารถนำแผ่นพิมพ์เดี่ยวไปประยุกต์ใช้งานและควบคุมอุปกรณ์ได้
6. เขียนโปรแกรมการทดลองในการใช้งานได้

### ประโยชน์ที่คาดว่าจะได้รับ

1. ความรู้ ความเข้าใจในระบบการทำงานของไมโครคอนโทรลเลอร์ ตระกูล MCS-51
2. แผ่นพิมพ์เดี่ยว (ต้นแบบ) ของไมโครคอนโทรลเลอร์ ตระกูล MCS-51
3. ลายวงจรแผ่นพิมพ์เดี่ยวของไมโครคอนโทรลเลอร์ ตระกูล MCS-51
4. เขียนโปรแกรมระบบควบคุมการทำงานของชิพไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ได้
5. นำแผ่นพิมพ์เดี่ยวไปประยุกต์ใช้งานและควบคุมอุปกรณ์ไฟฟ้าได้
6. โปรแกรมที่ใช้ในการทดลอง 10 โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

|              |           |
|--------------|-----------|
| นายจิตติพงษ์ | พิสิฐนฤดม |
| นายวาทิน     | ทิพนงค์   |
| นายศราวุธ    | สุชีฐาน   |
| นายสืบศักดิ์ | สุนิพันธ์ |

### อาจารย์ที่ปรึกษา

|                |                  |
|----------------|------------------|
| อาจารย์ไพบูลย์ | พวงวงศ์ตระกูล    |
| อาจารย์ปิยะ    | จิตรธรรมมาภิรมย์ |
| อาจารย์วรวิทย์ | สมหา             |

ปีการศึกษา 2541

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการนำเสนอเกี่ยวกับ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว โดยสร้างเป็นชุดศึกษาการทำงานของ 8051 ไมโครคอนโทรลเลอร์ เพื่อให้ง่ายต่อการศึกษาเรียนรู้และเข้าใจการทำงาน ตลอดจนนำไปใช้ประโยชน์ได้อย่างมีประสิทธิภาพ

การทำงานของ 8051 ไมโครคอนโทรลเลอร์ มีลักษณะการทำงานโดยการป้อนโปรแกรมที่อยู่ในรูปภาษาเครื่องบนแผ่นพิมพ์เดี่ยว เพื่อศึกษาการทำงานในแต่ละส่วนบนแผ่นพิมพ์เดี่ยว จะมีการแสดงผลที่จอผลึกเหลว และไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## II

### 8051 SINGLE BOARD MICROCONTROLLER

|               |                |
|---------------|----------------|
| MR.THITIPHONG | PHISITNARUEDOM |
| MR.VATIN      | TIPANONK       |
| MR.SARAVUTH   | SUKHEETAN      |
| MR.SEAUBSAK   | SUNIPAN        |

#### ADVISORS

|             |                  |
|-------------|------------------|
| MR.PIBOON   | POUNGVONGTRAKOON |
| MR.PIYA     | JITTAMMAPIROM    |
| MR.WORRAVIT | SOMHA            |

1998

#### ABTRACT

This thesis presents the project of 8051 Single Board Microcontroller. It's created in the set for support to study the operating of Microcontroller 8051, for easy to study and good sufficiency used.

The operating of the Single Board Microcontroller 8051 by keying machine code program for study the operating in part on Single Board. This opareting mode will be showing by LCD Display and LED connect with Port 1 of Microcontroller 8051.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี เนื่องมาจากการอนุเคราะห์ของอาจารย์ที่ปรึกษาปริญญานิพนธ์ และอาจารย์ประจำภาควิชาคณะครุศาสตร์วิสุทธรณทุกท่านที่ได้กรุณาให้คำปรึกษา, ข้อเสนอแนะ รวมทั้งรุ่นพี่และเพื่อน ๆ ทุกคนภายในภาควิชาครุศาสตร์วิสุทธรณ ตลอดจนบุคลากรผู้สนับสนุนการศึกษาดลอดมา

นอกจากนี้คณะผู้จัดทำขอขอบพระคุณภาควิชาครุศาสตร์วิสุทธรณ คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้กรุณาให้งบประมาณสนับสนุนในการจัดทำปริญญานิพนธ์ครั้งนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

| เรื่อง                                                        | หน้า |
|---------------------------------------------------------------|------|
| บทคัดย่อภาษาไทย                                               | I    |
| บทคัดย่อภาษาอังกฤษ                                            | II   |
| กิตติกรรมประกาศ                                               | III  |
| สารบัญ                                                        | IV   |
| สารบัญตาราง                                                   | VIII |
| สารบัญรูปภาพ                                                  | IX   |
| บทที่ 1 บทนำ                                                  | 1    |
| 1.1 ความเป็นมาและความสำคัญของโครงการ                          | 1    |
| 1.2 ชี้ความสามารถของโครงการ                                   | 1    |
| 1.3 เนื้อหาโดยสังเขป                                          | 2    |
| บทที่ 2 ทฤษฎีและหลักการ                                       | 3    |
| 2.1 กล่าวนำ                                                   | 3    |
| 2.2 คุณสมบัติของไมโครคอนโทรลเลอร์ MCS-51                      | 4    |
| 2.3 โครงสร้างของ MCS-51                                       | 4    |
| 2.3.1 หน่วยประมวลผลกลาง                                       | 4    |
| 2.3.2 หน่วยความจำข้อมูล                                       | 5    |
| 2.3.3 อุปกรณ์อินพุต/เอาต์พุต                                  | 6    |
| 2.4 สถาปัตยกรรมภายใน MCS-51                                   | 6    |
| 2.5 การจัดขั้วลักษณะภายนอกของ MCS-51                          | 6    |
| 2.6 การจัดหน่วยความจำของ MCS-51                               | 9    |
| 2.6.1 หน่วยความจำโปรแกรม                                      | 9    |
| 2.6.2 หน่วยความจำข้อมูล                                       | 10   |
| 2.7 ออสซิลเลเตอร์และสัญญาณนาฬิกาของ MCS-51                    | 10   |
| 2.8 การเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับหน่วยความจำภายนอก | 11   |
| 2.8.1 สัญญาณในการเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอก         | 12   |
| 2.8.2 สัญญาณในการเชื่อมต่อกับหน่วยความจำข้อมูลภายนอก          | 13   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

| เรื่อง                                                           | หน้า |
|------------------------------------------------------------------|------|
| 2.8.3 การเชื่อมต่อหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลภายนอก   | 15   |
| 2.9 ไทม์เมอร์/เคาน์เตอร์                                         | 16   |
| 2.9.1 ไทม์ค 0                                                    | 17   |
| 2.9.2 ไทม์ค 1                                                    | 17   |
| 2.9.3 ไทม์ค 2                                                    | 18   |
| 2.9.4 ไทม์ค 3                                                    | 18   |
| 2.10 การสื่อสารข้อมูลแบบอนุกรม                                   | 19   |
| 2.10.1 ไทม์ค 0                                                   | 20   |
| 2.10.2 ไทม์ค 1                                                   | 20   |
| 2.10.3 ไทม์ค 2                                                   | 21   |
| 2.10.4 ไทม์ค 3                                                   | 22   |
| 2.10.5 อัตราความเร็วในการรับส่งข้อมูล                            | 22   |
| 2.11 การอินเตอร์รัพต์                                            | 23   |
| 2.11.1 การอินเตอร์รัพต์แฟล็กไทม์เมอร์                            | 24   |
| 2.11.2 การอินเตอร์รัพต์พอร์ตอนุกรม                               | 24   |
| 2.11.3 การอินเตอร์รัพต์จากภายนอก                                 | 24   |
| 2.11.4 การควบคุมการอินเตอร์รัพต์                                 | 24   |
| 2.12 การเข้าถึงข้อมูลของ MCS-51                                  | 24   |
| 2.12.1 การเข้าถึงข้อมูลในรีจิสเตอร์                              | 25   |
| 2.12.2 การเข้าถึงข้อมูลโดยตรง                                    | 25   |
| 2.12.3 การเข้าถึงข้อมูลที่อยู่ในรีจิสเตอร์โดยอ้อม                | 25   |
| 2.12.4 การเข้าถึงข้อมูลโดยทันที                                  | 25   |
| 2.12.5 การเข้าถึงข้อมูลโดยใช้ตัวอ้างอิง                          | 25   |
| บทที่ 3 การออกแบบและการสร้าง                                     | 27   |
| 3.1 กล่าวนำ                                                      | 27   |
| 3.2 การพัฒนาการโปรแกรมของ 8051 ไมโครคอนโทรลเลอร์แบบแฟมพิมพ์เดียว | 27   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ในนโยบายด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

| เรื่อง                                     | หน้า |
|--------------------------------------------|------|
| 3.2.1 การพัฒนาโปรแกรมบนแผ่นพิมพ์เดี่ยว     | 28   |
| 3.2.2 การพัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์  | 28   |
| 3.3 ส่วนประกอบโดยทั่วไป                    | 28   |
| 3.4 การออกแบบและการทำงานในส่วนต่าง ๆ       | 30   |
| 3.4.1 ส่วนชุดควบคุม                        | 30   |
| 3.4.2 ส่วนเลือกสัญญาณแอนะล็อก              | 30   |
| 3.4.3 ส่วนคีย์บอร์ด                        | 30   |
| 3.4.4 ส่วนจอแสดงผลแบบผลึกเหลว              | 31   |
| 3.4.5 ส่วนการแสดงผลทางลำโพง                | 32   |
| 3.4.6 ส่วนการรับส่งข้อมูลทางพอร์ตนุกรม     | 33   |
| 3.4.7 ส่วนไดโอดเปล่งแสงแสดงผลที่พอร์ตนุกรม | 33   |
| 3.4.8 ส่วนแหล่งจ่ายไฟ                      | 34   |
| 3.5 การทำงานในส่วนของโปรแกรมมอเนเตอร์      | 35   |
| 3.5.1 ส่วนการทำงานของโปรแกรมหลัก           | 35   |
| 3.5.2 ส่วนโปรแกรมการสแกนคีย์บอร์ด          | 35   |
| 3.5.3 ส่วนโปรแกรมการประมวลผล               | 35   |
| 3.6 การประกอบวงจรในส่วนต่าง ๆ              | 38   |
| บทที่ 4 ผลการทดลองและทดสอบ                 | 41   |
| 4.1 การทดลองและผลการทดลอง                  | 41   |
| 4.1.1 การทดลองโปรแกรมการทดลองที่ 1         | 42   |
| 4.1.2 การทดลองโปรแกรมการทดลองที่ 2         | 43   |
| 4.1.3 การทดลองโปรแกรมการทดลองที่ 3         | 44   |
| 4.1.4 การทดลองโปรแกรมการทดลองที่ 4         | 46   |
| 4.2 สรุปผลการทดลอง                         | 49   |
| บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา   | 50   |
| 5.1 บทสรุป                                 | 50   |
| 5.2 ปัญหาและแนวทางแก้ไข                    | 50   |
| 5.3 แนวคิดในการพัฒนา                       | 52   |

## VIII

### สารบัญตาราง

| ตาราง                                                                      | หน้า |
|----------------------------------------------------------------------------|------|
| ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51                                | 3    |
| ตารางที่ 3.1 การทำงานของส่วนประกอบภายในไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 29   |
| ตารางที่ 5.1 ปัญหาและแนวทางการแก้ไขปัญหาในการทำปริญญานิพนธ์                | 51   |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

| รูปภาพ                                                                   | หน้า |
|--------------------------------------------------------------------------|------|
| รูปที่ 3.10 ผังการทำงานการสแกนคีย์บอร์ด                                  | 37   |
| รูปที่ 3.11 ผังการทำงานของการทำงานประมวลผลทางโปรแกรม                     | 38   |
| รูปที่ 3.12 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว                     | 39   |
| รูปที่ 3.13 การต่อ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียวกับพอร์ตอนุกรม | 39   |
| รูปที่ 3.14 พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว       | 40   |
| รูปที่ 3.15 โปรแกรมมอนิเตอร์                                             | 40   |



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของโครงการ

ปัจจุบันนี้ไมโครคอนโทรลเลอร์ ได้ถูกนำมาใช้ในการควบคุมการทำงานของเครื่องใช้ไฟฟ้าต่าง ๆ เช่น ควบคุมความเร็วหรือทิศทางการหมุนของมอเตอร์ไฟฟ้า, ควบคุมการทำงานของเครื่องปรับอากาศ ฯลฯ ตลอดจนระบบการควบคุมภายในโรงงานอุตสาหกรรม ด้วยเหตุที่ว่าการใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานของวงจรอิเล็กทรอนิกส์นั้น จะมีความถูกต้องเที่ยงตรงเชื่อถือได้ มีการทำงานที่รวดเร็วแม่นยำ และยังมีความยืดหยุ่นการทำงานได้โดยการเปลี่ยนแปลงทางฮาร์ดแวร์ ทำให้สะดวกและคล่องตัวในการใช้งาน นอกจากนี้ไมโครคอนโทรลเลอร์ที่ผลิตหรือออกแบบมาในปัจจุบัน มีการเพิ่มขีดความสามารถต่าง ๆ ให้สะดวกต่อการใช้งานมากขึ้น อีกทั้งยังช่วยลดความยุ่งยากในการออกแบบวงจรประกอบวงจรประยุกต์ใช้งานทำให้เปลี่ยนแปลงการทำงานได้ง่ายขึ้น

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ผลิตโดยบริษัทอินเทล เช่น เบอร์ 8031, 8032 และ 8051 เป็นต้น ข้อดีของไมโครคอนโทรลเลอร์ตระกูลนี้ คือสามารถเขียนโปรแกรมการควบคุมระบบได้สะดวก และมีเครื่องมือที่ช่วยในการพัฒนาระบบค่อนข้างมาก พร้อมทั้งมีพอร์ตอินพุต และเอาต์พุตภายในตัวทำให้สะดวกต่อการนำไปใช้งาน เหมาะสำหรับการนำไปในใช้งานควบคุมขนาดกลาง ดังนั้นจึงทำให้บริษัทต่าง ๆ ได้พยายามที่จะผลิตไมโครคอนโทรลเลอร์ที่มีการทำงานเหมือนกับ MCS-51 และมีความสามารถทางฮาร์ดแวร์ที่มีประสิทธิภาพสูง

จากที่กล่าวมา ในการนำไมโครคอนโทรลเลอร์ไปประยุกต์ใช้งานในการควบคุมให้มีประสิทธิภาพและมีความถูกต้อง จะต้องมีความเข้าใจการทำงานของคำสั่งหรือซอฟต์แวร์ที่ใช้สำหรับการควบคุมการทำงานอย่างละเอียด เพื่อให้ระบบควบคุมมีประสิทธิภาพในการทำงานต่อไป

### 1.2 ขีดความสามารถของโครงการ

โครงการนี้มีขีดความสามารถดังต่อไปนี้

1. สามารถติดต่อกับคอมพิวเตอร์ ผ่านทางพอร์ตอนุกรม RS-232 ได้

2. แสดงผลเป็นจอแสดงผลแบบผลึกเหลว 16 ตัวอักษร 2 บรรทัด

เอกสารนี้เป็นเอกสารที่ควรรวไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่ควรมานำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. มีพอร์ต 8255 จำนวน 2 ชุด ที่ชุดแรกจะใช้ในการรับและการส่งข้อมูลแบบขนานระหว่าง ไมโครคอนโทรลเลอร์กับตัวอุปกรณ์ภายนอก และชุดที่สองใช้สำหรับควบคุมการแสดง ผลของจอแสดงผลแบบผลึกเหลว
4. มีโปรแกรมมอนิเตอร์ สามารถใช้ติดต่อกับฮาร์ดแวร์ได้
5. ใช้หน่วยความจำโปรแกรมและหน่วยความจำข้อมูลขนาด 32 กิโลไบต์
6. ใช้เมทริกสวิตช์ขนาด 6 x 4 จุด

### 1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่าง ๆ เพื่อความสะดวกต่อการศึกษา และทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 2 ทฤษฎีและหลักการ ประกอบด้วยเนื้อหาในทางทฤษฎีที่เกี่ยวข้อง ซึ่งทำให้ผู้อ่านได้ มีความรู้ความเข้าใจเป็นพื้นฐานก่อน ซึ่งจะเป็นประโยชน์ต่อการทำความเข้าใจการทำงานของ ไมโครคอนโทรลเลอร์ MCS-51

บทที่ 3 การออกแบบการสร้างและการทำงาน โดยจะอธิบายถึงการออกแบบวงจรและการ สร้างในแต่ละส่วนของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

บทที่ 4 ผลการทดลองและการทดสอบ จะกล่าวถึงขั้นตอนในการทดลองและการทดสอบ ประสิทธิภาพในการทำงานของฮาร์ดแวร์ และซอฟต์แวร์ของปฏิญานิพนธ์นี้ เพื่อตรวจสอบว่า ปฏิญานิพนธ์นี้ สามารถทำงานได้ตรงตามวัตถุประสงค์ หรือไม่

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา เป็นการสรุปผลการทำงาน และได้เสนอแนะแนวทางในการแก้ไข และแนวทางการพัฒนา ให้มีประสิทธิภาพในการใช้งานได้อย่างมีประสิทธิภาพมากขึ้น

ภาคผนวก ก วงจรและการวางอุปกรณ์

ภาคผนวก ข การใช้งาน 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

ภาคผนวก ค ไปงานการทดลอง

ภาคผนวก ง ชุดคำสั่งในการทำงาน

ภาคผนวก จ โปรแกรมมอนิเตอร์ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

## ทฤษฎีและหลักการ

### 2.1 กล่าวนำ

ปัจจุบันนี้ในการออกแบบไมโครคอมพิวเตอร์จะใช้ไมโครคอนโทรลเลอร์ควบคุมระบบการทำงาน เนื่องจากใช้เวลาในการทำงานได้รวดเร็วและแม่นยำ ไมโครคอนโทรลเลอร์ที่ผลิตออกมาจะเป็นลักษณะชิปเดี่ยว (Single Chip) ซึ่งมีความสะดวกในการใช้งาน และการเขียนโปรแกรมใช้ในการควบคุม พร้อมทั้งยังมีเครื่องมือที่ช่วยในการพัฒนาระบบให้มีประสิทธิภาพค่อนข้างมาก

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ผลิตโดยบริษัทอินเทล ได้มีการนำมาใช้งานกันอย่าง กว้างขวางในปี ค.ศ. 1980 ต่อมาบริษัท Phillips และ บริษัท Siemens ได้รับลิขสิทธิ์ในการผลิตจำหน่ายและได้มีการเพิ่มประสิทธิภาพมากในการใช้งานมากขึ้น ทำให้ในปัจจุบันมีการผลิตของไมโครคอนโทรลเลอร์ที่มีมาตรฐานมาจากตระกูล MCS-51 ของบริษัทอินเทลเป็นจำนวนมาก

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วย ไมโครคอนโทรลเลอร์หลายรุ่น ซึ่งจะมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่จะมีขนาดของหน่วยความจำภายในและภายนอกที่แตกต่างกัน เพื่อให้เหมาะสำหรับการนำไปใช้งานตามความต้องการของงานประเภทต่าง ๆ ซึ่งลักษณะคุณสมบัติต่าง ๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้แสดงดังตารางที่ 2.1

ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

| EMBEDDED CONTROLLERS             |        |        |       |         |         |       |        |        |      |
|----------------------------------|--------|--------|-------|---------|---------|-------|--------|--------|------|
| Feature                          | 8051AH | 8031AH | 8051H | 80C51BH | 80C31BH | 87CH1 | 8052AH | 8052AH | 8752 |
| Program Memory (Bytes)           | 4 K    | -      | 4 K   | 4 K     | -       | 4 K   | -      | 4 K    | 4 K  |
| Data Memory (Bytes)              | 128    | 128    | 128   | 128     | 128     | 128   | 128    | 128    | 128  |
| Program Memory Expantion (Bytes) | 64 K   | 64 K   | 64 K  | 64 K    | 64 K    | 64 K  | 64 K   | 64 K   | 64 K |
| Data Memory Expantion (Bytes)    | 64 K   | 64 K   | 64 K  | 64 K    | 64 K    | 64 K  | 64 K   | 64 K   | 64 K |
| Max Clock Frequency (MHz)        | 12     | 12     | 12    | 16      | 16      | 16    | 16     | 12     | 16   |
| Typical Intruction Time (us)     | 1      | 1      | 1     | 0.75    | 0.75    | 0.75  | 1      | 1      | 1    |
| 16-Bit Timer/Counter             | 2      | 2      | 2     | 2       | 2       | 2     | 3      | 3      | 3    |
| No. of I/O Lines                 | 32     | 16     | 32    | 32      | 16      | 32    | 32     | 16     | 32   |
| Interrupt Sources                | 5      | 5      | 5     | 5       | 5       | 5     | 6      | 6      | 6    |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 คุณสมบัติของไมโครคอนโทรลเลอร์ MCS-51

ลักษณะคุณสมบัติโดยทั่วไปของไมโครคอนโทรลเลอร์ MCS-51 มีดังต่อไปนี้

1. หน่วยประมวลผลกลางขนาด 8 บิต
2. หน่วยความจำโปรแกรมภายใน (Program Memory) ขนาด 4 กิโลไบต์
3. หน่วยความจำข้อมูลภายใน (Data Memory) ขนาด 128 ไบต์
4. อ้างตำแหน่งหน่วยความจำโปรแกรมได้ถึง 64 กิโลไบต์
5. อ้างตำแหน่งหน่วยความจำข้อมูลได้ถึง 64 กิโลไบต์
6. หน่วยความจำโปรแกรมและหน่วยความจำข้อมูลที่อยู่ภายนอกชิพ แยกจากกันอย่างละ 64 กิโลไบต์
7. มีพอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 4 พอร์ต (32บิต) แยกกันอย่างอิสระ
8. มีวงจรนับ/จับเวลาขนาด 16 บิต 2 ชุด ทำงานได้ 4 โหมด
9. มีพอร์ตการสื่อสารอนุกรม (Universal Asynchronous Receiver Transmitter : UART)
 

รับส่งข้อมูลได้ในเวลาเดียวกัน (Full Duplex) สามารถเลือกรูปแบบการส่งได้ 4 รูปแบบ
10. รับสัญญาณอินเทอร์รัพท์ได้ 6 แหล่ง กระโดดไปทำงานตอบสนองได้ 5 ตำแหน่ง
11. มีวงจรออสซิลเลเตอร์ภายใน
12. นำข้อมูลมา AND, OR หรือทำ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

## 2.3 โครงสร้างของ MCS-51

ภายในไมโครคอนโทรลเลอร์ MCS-51 จะประกอบด้วยเทคนิคต่าง ๆ เช่น AND, OR และ NOT ซึ่งเทคนิคเหล่านี้จะนำเอามาออกแบบให้มีหน้าที่การทำงานต่าง ๆ เช่น วงจรถอดรหัสคำสั่ง วงจรสร้างสัญญาณนาฬิกา เป็นต้น ซึ่งมีโครงสร้างดังรูปที่ 2.1 โดยลักษณะโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 จะมีส่วนประกอบ 3 ส่วนหลัก ๆ ดังนี้

### 2.3.1 หน่วยประมวลผลกลาง (Central Processing Unit)

ส่วนนี้จะทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่น ๆ เรียกว่า วงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุม ได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ, สัญญาณการรับข้อมูลเข้าและส่งออก, ส่วนควบคุมการขัดจังหวะ และ ส่วนการควบคุมบัส เป็นต้น การสร้างสัญญาณจากวงจรควบคุมจากหน่วยประมวลผลกลางนี้จะสร้างสัญญาณโดยการถอดรหัสคำสั่งที่กำหนดไว้ และสัญญาณควบคุมที่สร้างขึ้นนี้จะทำการอ้างอิงกับสัญญาณนาฬิกาของวงจรออสซิลเลเตอร์เพื่อให้ทุก ๆ ส่วนทำงานได้อย่างถูกต้อง

### 2.3.2 หน่วยความจำข้อมูล (Memory)

1. สัญญาของตำแหน่งที่ต้องการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูล ซึ่งภายในตัวไมโครคอนโทรลเลอร์ MCS-51 จะมีหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลขนาดสูงสุดชนิดละ 65,536 ตำแหน่ง (64 กิโลไบต์) ดังนั้นในการอ้างตำแหน่งของหน่วยความจำจะต้องใช้เส้นตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น

- ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สัญญาณควบคุมที่ส่งยังหน่วยความจำ เพื่อบอกว่าต้องการอ่านหรือเขียนข้อมูลโดยวงจรถอดรหัสคำสั่งจะทำการสร้างสัญญาณควบคุมจากคำสั่งที่อ่านเข้ามาจากหน่วยความจำโปรแกรม

### 2.3.3 อุปกรณ์ อินพุต/เอาต์พุต (Input/Output Device)

เป็นส่วนที่ใช้ในการส่งข้อมูลเข้าหรือนำข้อมูลออกจากไมโครคอนโทรลเลอร์ MCS-51 ซึ่งจะทำให้ติดต่อกับอุปกรณ์ อินพุต/เอาต์พุต ได้ สามารถแบ่งลักษณะได้ 3 ประเภทดังนี้

#### 1. อินพุต/เอาต์พุต พอร์ต (Input/Output Port)

ใช้สำหรับรับข้อมูลที่เป็นสัญญาณเข้าหรือออกจากตัวไมโครคอนโทรลเลอร์ MCS-51 มีทั้งหมด 4 พอร์ต โดยแต่ละพอร์ตจะรับข้อมูลได้ 8 บิต มีพอร์ต P0, P1, P2 และ P3

#### 2. ไทม์เมอร์/เคาน์เตอร์ 0, ไทม์เมอร์/เคาน์เตอร์ 1 (Timer/Counter 0, Timer/Counter 1)

เป็นวงจรที่สามารถนับจำนวนไซเคิลของสัญญาณที่ต่อภายนอกของไมโครคอนโทรลเลอร์ MCS-51 สามารถที่จะตั้งค่าเริ่มต้นการนับและอ่านค่าการนับได้โดยหน่วยประมวลผลกลาง

#### 3. พอร์ตอนุกรม (Serial Port)

หน่วยประมวลผลกลางจะอ่านและเขียนข้อมูลกับพอร์ตอนุกรมเป็นแบบ 8 บิต แต่ละข้อมูลจะถูกส่งออกจาก ไมโครคอนโทรลเลอร์ MCS-51 เรียงไปทีละบิตออกจากขา TXD และในการรับข้อมูลก็จะรับเข้ามาทีละบิตทางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิต เพื่อให้หน่วยประมวลผลกลางอ่านไปใช้งานต่อไป

## 2.4 สถาปัตยกรรมภายในของ MCS-51

สถาปัตยกรรมภายในของ MCS-51 ซึ่งจะประกอบไปด้วย หน่วยประมวลผลกลาง, หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล, พอร์ตอินพุตและพอร์ตเอาต์พุต, ไทม์เมอร์/เคาน์เตอร์สถานะข้อมูล เป็นต้น ซึ่งจะติดต่อกับบัสข้อมูลขนาด 8 บิต และจะมีบัฟเฟอร์สำหรับการติดต่อข้อมูลกับภายนอกผ่านพอร์ตอินพุต/เอาต์พุต โดยมีลักษณะโครงสร้างสถาปัตยกรรมดังรูปที่ 2.2

## 2.5 การจัดขาลักษณะภายนอกของ MCS - 51

ลักษณะการจัดเรียงขาของชิพ MCS-51 จะบรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งมีการจัดขาลักษณะภายนอกดังรูปที่ 2.3 และจำนวนขาของชิพจะมีทั้งหมด 40 ขา ดังนี้

VCC : (ขา 40) ต่อไฟเลี้ยง +5 โวลต์

VSS : (ขา 20) ต่อลงกราวด์

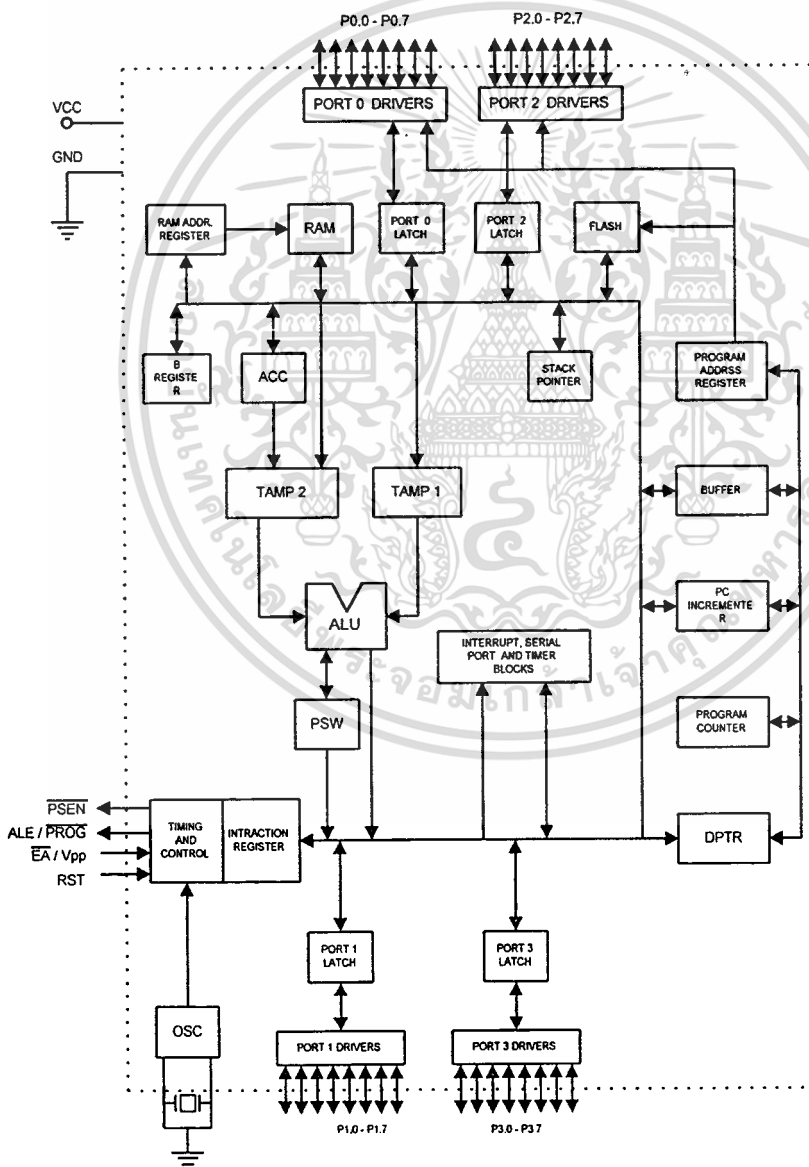
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 0 : (ขา 32-39) มีอยู่ทั้งหมด 8 บิต คือ P0.0-P0.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ต และ ใช้เป็นตัวส่งแอดเดรสไบต์ต่ำ (A0-A7) และรับข้อมูล (D0-D7) จากหน่วยความจำภายนอก

Port 1 : (ขา 1-8) มีอยู่ทั้งหมด 8 บิต คือ P1.0-P1.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ต

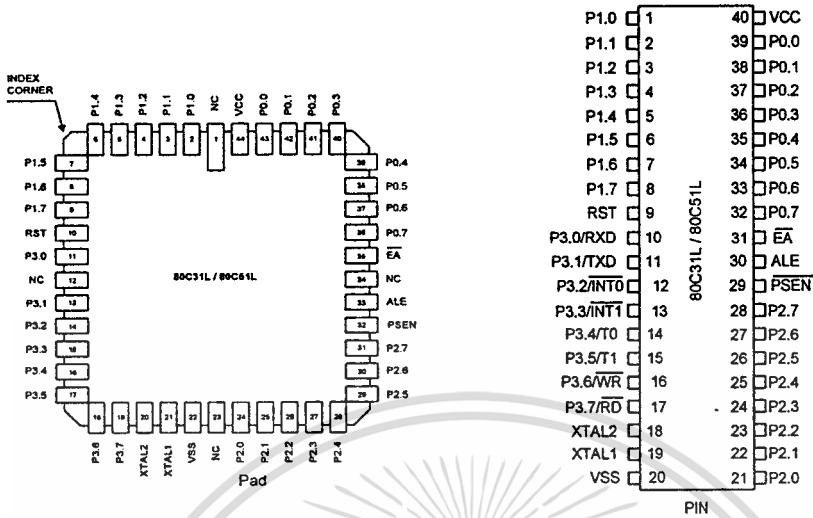
Port 2 : (ขา 21-28) มีอยู่ทั้งหมด 8 บิต คือ P2.0-P2.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ต และ ใช้เป็นตัวส่งแอดเดรสไบต์สูง (A8-A15) เพื่อใช้ติดต่อกับหน่วยความจำภายนอก

Port 3 : (ขา 10-17) มีอยู่ทั้งหมด 8 บิต คือ P3.0-P3.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ต และ ใช้งานในหน้าที่พิเศษดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไขตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รูปที่ 2.2 สถาปัตยกรรมภายในของ MCS-51



รูปที่ 2.3 ลักษณะการจัดเรียงขาภายนอกของ MCS-51

P3.0/RXD (Serial Input Port) : ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) : ใช้ส่งข้อมูลแบบอนุกรม

P3.2/INT0 (External Interrupt) : ใช้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

P3.3/INT1 (External Interrupt) : ใช้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

P3.4/T0 (Timer / Counter 0 External Input) : ใช้เป็นอินพุตให้วงจรรนับ/จับเวลา ชุดที่ 0

P3.5/T1 (Timer / Counter 1 External Input) : ใช้เป็นอินพุตให้วงจรรนับ/จับเวลา ชุดที่ 1

P3.6/WR (External Data Memory Write Strobe) : ควบคุมการเขียนข้อมูลไปยังหน่วย

ความจำภายนอก

P3.6/RD (External Data Memory Write Strobe) : ควบคุมการอ่านข้อมูลจากหน่วยความ

จำภายนอก

RST : (ขา 9) Reset ใช้สำหรับรีเซ็ตทุกอย่างภายในชิพ เพื่อเริ่มต้นการทำงานใหม่ ซึ่งในการรีเซ็ตจะต้องป้อนลอจิก “1” นานอย่างน้อย 2 แมกซีนไซเคิล

ALE : (ขา 30) Address Latch Enable เป็นขาส่งสัญญาณออกไปภายนอกเพื่อควบคุมการแลตช์ค่าแอดเดรสไบต์ต่ำจากพอร์ต 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**PSEN** : (ขา 29) Program Strob Enable เป็นขาส่งสัญญาณเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อขานี้มีลอจิกเป็น “0” จะอ่านโปรแกรมจากหน่วยความจำโปรแกรมภายนอก และถ้าเป็นการอ่านหน่วยความจำโปรแกรมภายในขาจะไม่แอคทีฟ

**EA** : (ขา 31) External Access เป็นขาที่ใช้สำหรับการเลือกการทำงานของหน่วยความจำโปรแกรมภายใน หรือภายนอกหน่วยความจำโปรแกรมภายนอกชิพ เมื่อขานี้มีลอจิกเป็น “0”

**XTAL1** : (ขา 19) ใช้ในการต่อคริสตัลภายนอกชิพ โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์

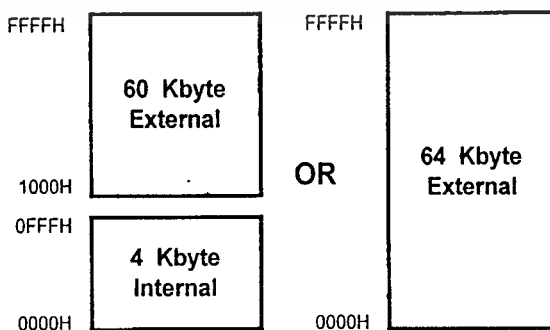
**XTAL2** : (ขา 18) ใช้ต่อคริสตัลภายนอก โดยเป็นเอาต์พุตออกจากวงจรออสซิลเลเตอร์

## 2.6 การจัดหน่วยความจำของ MCS-51

หน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 จะแยกแอดเดรสสำหรับหน่วยความจำของโปรแกรม และหน่วยความจำของข้อมูลออกจากกัน โดยสามารถแบ่งหน่วยความจำออกได้เป็น 2 แบบตามลักษณะการใช้งาน ดังนี้

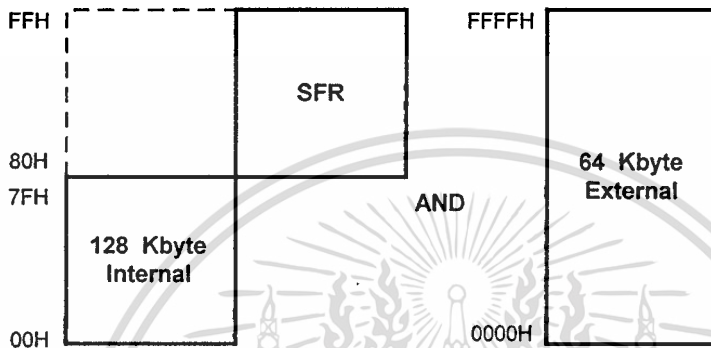
### 2.6.1 หน่วยความจำโปรแกรม (Program Memry)

เป็นหน่วยความจำที่ใช้ในการเก็บคำสั่งในรูปของภาษาเครื่อง ซึ่งต้องการให้ MCS-51 ทำงาน เมื่อ MCS-51 ทำงานจะทำการอ่านข้อมูลที่เก็บอยู่ในหน่วยความจำโปรแกรมไปถอดรหัสแล้วทำการสร้างสัญญาณควบคุมส่วนต่าง ๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำโปรแกรมสามารถขยายได้ถึง 64 กิโลไบต์และจำนวนไบต์ต่ำ 4 กิโลไบต์จะอยู่ภายในชิพ MCS-51 และ การใช้งานจะต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นภาษาเครื่องของ MCS-51 ตามลำดับการทำงานที่ต้องการ ซึ่งลักษณะการจัดพื้นที่ของหน่วยความจำโปรแกรม ดังรูปที่ 2.4



## 2.6.2 หน่วยความจำข้อมูล (Data Memory)

เป็นหน่วยความจำที่ใช้สำหรับการเก็บข้อมูล สามารถทำการอ่านและเขียนข้อมูลได้ ซึ่งจะมีหน่วยความจำภายในชิพ MCS-51 ขนาดเพียง 128 ไบต์ ส่วนหน่วยความจำข้อมูลภายนอกชิพจะมีขนาด 64 กิโลไบต์ ดังรูปที่ 2.5



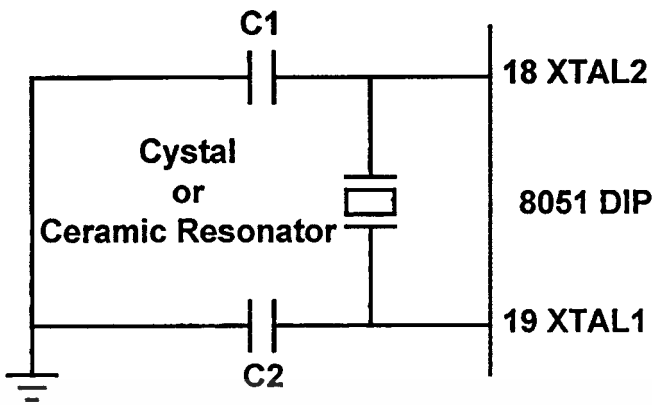
รูปที่ 2.5 การจัดพื้นที่หน่วยความจำข้อมูล

## 2.7 ออสซิลเลเตอร์และสัญญาณนาฬิกาของ MCS-51

วงจรกำเนิดสัญญาณพัลส์ซึ่งเกิดขึ้นพร้อมกับการทำงานภายในชิพ MCS-51 ขา XTAL1 และขา XTAL2 จะต่อกับวงจรรีโซแนนซ์ เป็นออสซิลเลเตอร์ สามารถใช้คริสตัล (Crystal) หรือตัวเก็บประจุ ดังรูปที่ 2.6 ความถี่คริสตัลเป็นตัวกำเนิดความถี่นาฬิกาภายในของไมโครคอนโทรลเลอร์ ผู้ออกแบบจะออกแบบให้สามารถทำงานที่ความถี่สูงสุดและต่ำสุด เช่น 1MHz - 6MHz

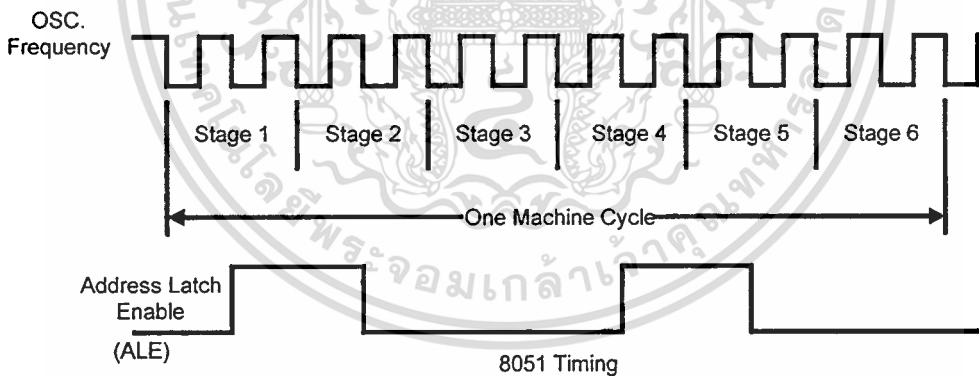
การสื่อสารจำเป็นต้องบอกความถี่ออสซิลเลเตอร์ เนื่องจากความต้องการให้เคาน์เตอร์ภายในคำนวณอัตราสัญญาณนาฬิกาให้เป็นบอดเรตมาตรฐาน ถ้าความถี่นาฬิกาหารแล้วเหลือเศษความถี่การสื่อสารจะไม่มาตรฐาน

เชรามิกรีโซแนนซ์ถูกนำมาใช้อย่างกว้างขวางเพราะมีราคาถูก แต่จะมีเสถียรภาพทางความถี่ลดลงและไม่มีความแน่นอน ถ้าเป็นการสื่อสารแบบอนุกรมด้วยความเร็วสูงอาจเกิดภาวะวิกฤติขึ้นได้ ออสซิลเลเตอร์ที่เป็นแบบคริสตัล, ตัวเก็บประจุ และอินเวอร์เตอร์บนชิพ โดยจะมีหน้าที่สร้างขบวนพัลส์ที่ความถี่คริสตัล ดังรูปที่ 2.7



รูปที่ 2.6 วงจรออสซิลเลเตอร์ภายใน MCS-51

รอบการทำงานของคำสั่ง เป็นค่าที่น้อยที่สุดในการทำคำสั่งหนึ่ง ถ้าเป็นคำสั่งที่ซับซ้อนมาก จะใช้เวลา 2-3 เมกไซเคิล โดยใน 1 เมกไซเคิล จะประกอบด้วยสัญญาณนาฬิกาจำนวน 12 ลูก โดยสัญญาณนาฬิกาแต่ละลูก เรียกว่า เฟส (Phase) สัญญาณนาฬิกา 2 เฟส รวมกันเป็น 1 สเตต (State) เพราะฉะนั้นใน 1 เมกไซเคิลจึงมี 6 สเตต ดังรูปที่ 2.7



รูปที่ 2.7 ช่วงเวลาของสัญญาณนาฬิกาใน 1 เมกไซเคิล

## 2.8 การเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับหน่วยความจำภายนอก

ไมโครคอนโทรลเลอร์ MCS-51 สามารถติดต่อกับหน่วยความจำภายนอกได้ 2 ส่วน คือ หน่วยความจำโปรแกรมภายนอกและหน่วยความจำข้อมูลภายนอก ในการใช้งานอาจจะต้องการใช้งานร่วมกันหรือต่อแยกการใช้งานก็ได้ ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8.1 สัญญาณในการเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอก

ในการเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำโปรแกรมภายนอก ข้อที่สำคัญ คือ จะต้องต่อขาของสัญญาณควบคุมหน่วยความจำโปรแกรมให้ถูกต้อง สามารถตรวจสอบสัญญาณการเชื่อมต่อได้ดังรูปที่ 2.8 ซึ่งเป็นลักษณะการต่อของขาสัญญาณควบคุม และการทำงานสามารถแบ่งลักษณะตามขาของสัญญาณได้ ดังนี้

### 1. สัญญาณ $\overline{\text{CE}}$ (Chip Enable)

สัญญาณนี้ทำหน้าที่เลือกให้อุปกรณ์ทำงาน โดยถ้าขานี้แอกทีฟหรือเป็นระดับลอจิกต่ำจะยอมให้หน่วยความจำโปรแกรมทำงานตามปกติ แต่ถ้าเป็นระดับลอจิกสูงจะไม่มีการทำงานใด ๆ ภายใน หน่วยความจำโปรแกรม

### 2. สัญญาณ $\overline{\text{OE}}$ (Output Enable)

เมื่อขาสัญญาณนี้แอกทีฟหรือระดับลอจิกต่ำจะทำให้มีการนำข้อมูลภายในตำแหน่งที่ถูกระบุด้วย บัสแอดเดรส ส่งมายังบัสข้อมูลของ หน่วยความจำโปรแกรม

### 3. สัญญาณ A0-A15

ขาสัญญาณนี้เป็นขาแอดเดรสที่ใช้เป็นตัวกำหนดตำแหน่งทั้งหมดของข้อมูลที่อยู่ภายใน หน่วยความจำโปรแกรม

### 4. สัญญาณ D0-D7

ขาสัญญาณนี้เป็นขาที่ใช้สำหรับการรับและส่งค่าของข้อมูลออกจากหน่วยความจำให้กับ ไมโครคอนโทรลเลอร์

### 5. สัญญาณ $\overline{\text{EA}}$ (External Access)

ใช้ในการกำหนดว่าจะให้ไมโครคอนโทรลเลอร์อ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกหรือหน่วยความจำโปรแกรมภายในของตัวไมโครคอนโทรลเลอร์เอง ซึ่งหากจะเป็นระดับลอจิกต่ำจะอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก และถ้าเป็นลอจิกสูงจะอ่านข้อมูลจากหน่วยความจำโปรแกรมภายใน และมีการใช้งานแอดเดรสที่อยู่ในช่วงที่สูงเกินค่าสูงสุดของหน่วยความจำโปรแกรมภายใน ไมโครคอนโทรลเลอร์จะทำการอ่านคำสั่งหรือข้อมูลตำแหน่งที่สูงเกินกว่าค่าสูงสุดมาจากหน่วยความจำโปรแกรมภายนอกโดยอัตโนมัติ

### 6. สัญญาณ $\overline{\text{PSEN}}$ (Program Strobe Enable)

ขาสัญญาณนี้จะใช้ในการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกจะแอกทีฟที่ระดับลอจิกต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 7. สัญญาณ P0.0-P0.7

ขาสัญญาณนี้จะใช้ในการกำหนดค่าของแอดเดรสไบต์ต่ำ (A0-A7) ของหน่วยความจำและใช้เป็นบัสข้อมูล ในการรับและส่งข้อมูล

### 8. สัญญาณ P2.0-P2.7

ขาสัญญาณนี้จะใช้ในการกำหนดค่าของแอดเดรสไบต์สูง (A8-A15) ของหน่วยความจำโปรแกรม

### 9. สัญญาณ ALE (Address Latch Enable)

ขาสัญญาณนี้จะใช้ในการควบคุมการแลตช์ค่าของแอดเดรสไบต์ต่ำ



รูปที่ 2.8 สัญญาณการติดต่อกับหน่วยความจำโปรแกรม

## 2.8.2 สัญญาณในการเชื่อมต่อกับหน่วยความจำข้อมูลภายนอก

ในการเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำข้อมูลภายนอก ข้อที่สำคัญ คือ จะต้องขอขาสัญญาณควบคุมหน่วยความจำข้อมูลให้ถูกต้องสามารถตรวจสอบสัญญาณการเชื่อมต่อขาสัญญาณได้ ดังรูปที่ 2.9 ซึ่งเป็นลักษณะของขาสัญญาณควบคุม และการทำงาน โดยสามารถแบ่งลักษณะตามขาของสัญญาณได้ ดังนี้

### 1. สัญญาณ $\overline{\text{CE}}$ (Chip Enable)

ขาสัญญาณนี้ทำหน้าที่ในการเลือกให้แรมทำงานโดยถ้าขานี้แอคทีฟหรือเป็นระดับลอจิกต่ำ จะยอมให้แรมทำงานโดยปกติ แต่ถ้ามีระดับลอจิกสูงจะไม่มีการทำงานใด ๆ ภายในหน่วยความจำข้อมูล

### 2. สัญญาณ $\overline{\text{EA}}$ (External Access)

เมื่อขาสัญญาณนี้แอคทีฟหรือเป็นระดับลอจิกต่ำ จะทำให้มีการนำข้อมูลภายในตำแหน่งที่ถูกระบุด้วยบัสแอดเดรส ส่งมายังบัสข้อมูลของหน่วยความจำข้อมูล

### 3. สัญญาณ $\overline{\text{WR}}$ (Write Enable)

ไม่ว่ากรณีใด ขาสัญญาณนี้เป็นสัญญาณลบและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อขาสัญญาณนี้แอคทีฟหรือมีสัญญาณลอจิกต่ำ จะทำการเขียนข้อมูลลงในหน่วยความจำในตำแหน่งที่ถูกระบุด้วยบัสแอดเดรส

#### 4. สัญญาณ A0-A15

ขาสัญญาณนี้จะเป็นขาแอดเดรสที่ใช้เป็นในการกำหนดตำแหน่งทั้งหมดของข้อมูลภายในหน่วยความจำข้อมูล

#### 5. สัญญาณ D0-D7

ขาสัญญาณนี้เป็นขานำข้อมูลเข้าและส่งออกจากหน่วยความจำข้อมูล

#### 6. สัญญาณ P0.0-P0.7

ขาสัญญาณนี้ใช้เป็นตัวกำหนดแอดเดรสไบต์ต่ำ (A0-A7) ของหน่วยความจำและใช้เป็นบัสข้อมูล

#### 7. สัญญาณ P2.0-P2.7

ขาสัญญาณนี้ใช้เพื่อกำหนดแอดเดรสไบต์สูง (A8-A15) ของหน่วยความจำข้อมูล

#### 8. สัญญาณ ALE (Address Latch Enable)

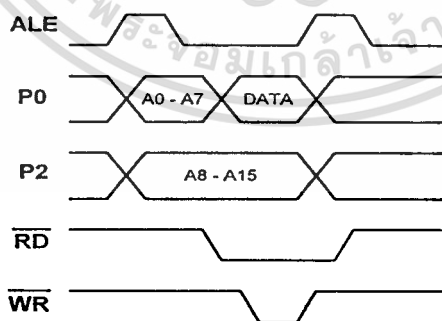
ขาสัญญาณนี้ใช้สำหรับควบคุมการแลตช์ค่าแอดเดรสไบต์ต่ำ

#### 9. สัญญาณ P3.6 (RD)

ขาสัญญาณนี้ใช้สำหรับควบคุมการเขียนข้อมูลลงในหน่วยความจำข้อมูล

#### 10. สัญญาณ P3.7 (WR)

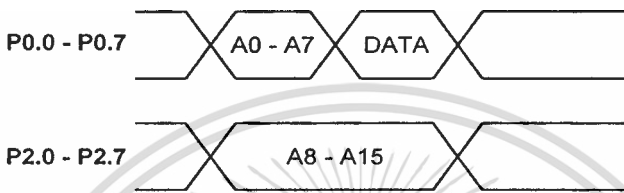
ขาสัญญาณนี้ใช้สำหรับควบคุมการอ่านข้อมูลออกมาจากหน่วยความจำข้อมูล



รูปที่ 2.9 สัญญาณการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

### 2.8.3 การเชื่อมต่อกับหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลภายนอก

ในไมโครคอนโทรลเลอร์ระบบบัสแอดเดรสและบัสข้อมูลของ MCS-51 เป็นลักษณะของการมัลติเพล็กซ์จากพอร์ตเดียวกัน คือ ในระยะเวลาเริ่มต้น เส้นสัญญาณของพอร์ตจะใช้ส่งค่าของแอดเดรสตำแหน่งที่ต้องการติดต่อด้วย ในช่วงเวลาต่อมาจึงจะเปลี่ยนเป็นสภาวะอิมพีแดนซ์สูงเพื่อใช้งานในฐานะของบัสข้อมูล ซึ่งจะมีลักษณะของสัญญาณ ดังรูปที่ 2.10



รูปที่ 2.10 การมัลติเพล็กซ์บัสแอดเดรสและบัสข้อมูลของพอร์ต 0

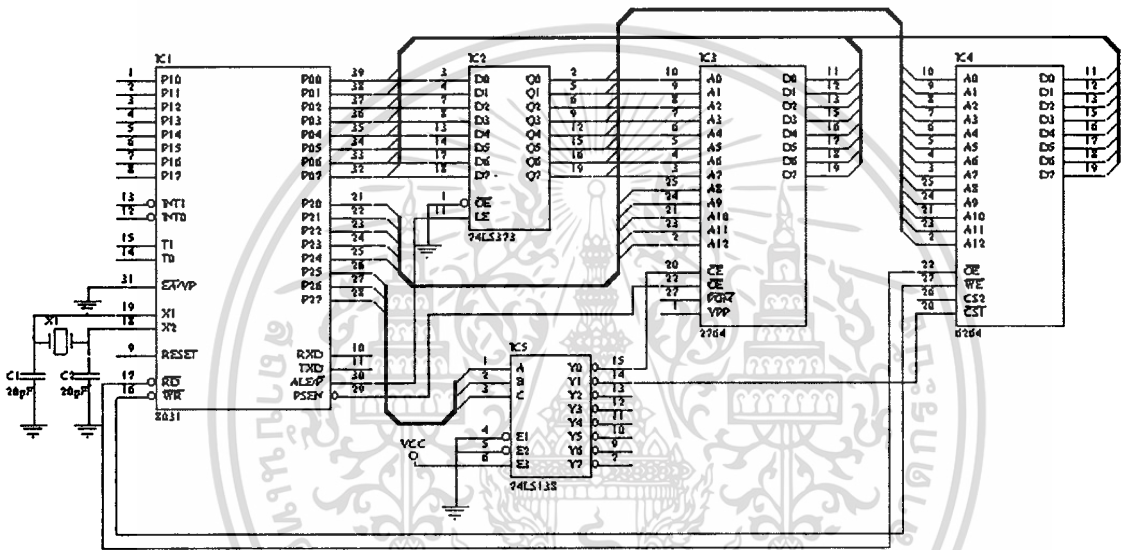
แต่เนื่องจากว่า หน่วยความจำโปรแกรม ที่ใช้งานกันทั่วไปนั้น ไม่ใช้การมัลติเพล็กซ์และมีขาสัญญาณบัสแอดเดรสและบัสข้อมูลแยกออกจากกันโดยชัดเจน ดังนั้นการเชื่อมต่อหน่วยความจำเพื่อทำหน้าที่เป็นหน่วยความจำโปรแกรม จึงจำเป็นต้องมีวงจรแลตช์สัญญาณ (Latch) ประกอบเพิ่มเติมขึ้น เพื่อทำการค้างค่าของแอดเดรสที่ส่งออกมาจากชิพของ MCS-51 ในช่วงแรกให้กับขาสัญญาณแอดเดรสของหน่วยความจำโปรแกรมต่อไป

การติดต่อกับหน่วยความจำโปรแกรมภายนอกชิพ MCS-51 จะสังเกตว่าในช่วงเวลาของเมกซีนไคเกิดหนึ่ง ๆ นั้นไมโครคอนโทรลเลอร์ MCS-51 ซึ่งจะมีการนำข้อมูลมาถึงสองครั้งด้วยกัน ดังนั้นการทำงานของไมโครคอนโทรลเลอร์ MCS-51 จึงจะไปอ่านข้อมูลโปรแกรมมาเป็นทวีคูณเสมอ ในช่วงของเวลาเริ่มต้นของการติดต่อกับหน่วยความจำภายนอก ที่พอร์ต 0 จะเป็นค่าแอดเดรสไบต์ต่ำ (A0-A7) และช่วงเวลาต่อมาจึงจะเป็นบัสข้อมูล การส่งค่าของแอดเดรสไบต์นี้จะอยู่ในช่วงเวลาขอบลอจิกต่ำ ดังนั้นในการออกแบบวงจร ALE ในการทำให้ไอซีแลตช์ภายนอกค้างระดับสัญญาณของแอดเดรสเหล่านี้ไว้ ส่วนสัญญาณ  $\overline{\text{PSEN}}$  จะใช้ในการเลือกให้หน่วยความจำโปรแกรมทำงานและอ่านค่าข้อมูลกลับมา

ขณะเมื่อสัญญาณ  $\overline{\text{PSEN}}$  เป็นระดับลอจิกต่ำหน่วยความจำโปรแกรมก็จะทำการถอดรหัสค่าแอดเดรสและส่งข้อมูลที่อยู่ในตำแหน่งนั้นออกมาโดยสัญญาณ  $\overline{\text{PSEN}}$  นี้จะค้างสภาวะการเป็นลอจิกต่ำไว้ช่วงเวลาหนึ่งเพื่อให้ข้อมูลถูกส่งออกมาจากหน่วยความจำโปรแกรมเรียบร้อยแล้วจึงกลับไปเป็นลอจิกสูงตามเดิม และในช่วงจังหวะขาขึ้นของสัญญาณ  $\overline{\text{PSEN}}$  ชิพ MCS-51 จะทำการสุ่มอ่านข้อมูลเข้ามาสำหรับข้อมูลทางพอร์ต 2 ซึ่งจะเป็นค่าแอดเดรสไบต์สูง (A8-A15) นั้น จะถูกส่งออกมา

ในช่วงกึ่งกลางระหว่างที่สัญญาณ ALE เป็นลอจิกสูง ซึ่งจะเป็นเวลาใกล้เคียงที่ใช้กับการส่งค่าของแอดเดรสไบต์ต่ำออกมาทางพอร์ต 0 สำหรับค่าแอดเดรสพอร์ต 2 นั้นจะค้างค่าอยู่ตลอดช่วงรอบเวลาของการติดต่อกับหน่วยความจำโปรแกรม

ในกรณีการเชื่อมต่อกับหน่วยความจำข้อมูลภายนอกนั้น จะเหมือนกับการเชื่อมต่อกับหน่วยความจำโปรแกรม แต่จะแตกต่างกันที่สัญญาณควบคุมการอ่านและการเขียนข้อมูลเท่านั้น ลักษณะการเชื่อมต่อหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลจะเป็นลักษณะดังรูปที่ 2.11



รูปที่ 2.11 การเชื่อมต่อหน่วยความจำข้อมูลและหน่วยความจำโปรแกรม

## 2.9 ไทม์เมอร์/เคาน์เตอร์ (Timer/Counter)

ภายในชิพไมโครคอนโทรลเลอร์ MCS-51 จะมี ไทม์เมอร์/เคาน์เตอร์ 2 ตัว ขนาด 16 บิต คือ ไทม์เมอร์/เคาน์เตอร์ 0 และ ไทม์เมอร์/เคาน์เตอร์ 1 ส่วนเบอร์ 8032/8052 จะมีเพิ่มอีก 1 ชุด คือ ไทม์เมอร์/เคาน์เตอร์ 2 ขณะที่แต่ละ ไทม์เมอร์/เคาน์เตอร์ สามารถที่จะติดตั้งให้ทำงานได้เป็นตัวจับเวลาหรือตัวนับก็ได้ โดยวิธีการเซตหรือการเคลียร์บิตที่ควบคุมในรีจิสเตอร์ TMOD ในกลุ่ม SFR

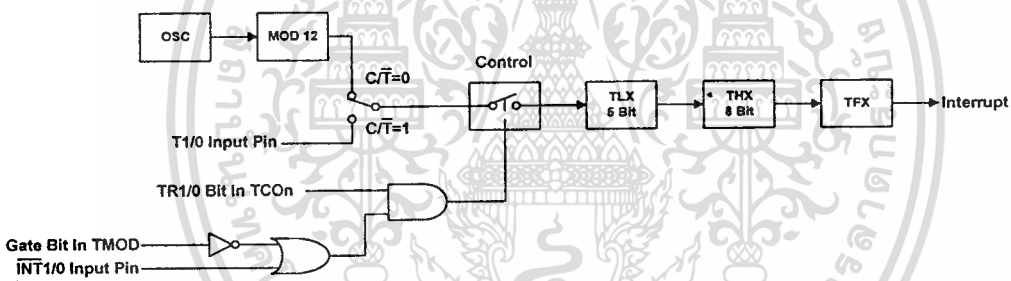
ในฟังก์ชันไทม์เมอร์ ตัวรีจิสเตอร์จะเพิ่มค่าทุก ๆ รอบเมกซ์ซินไซเคิล ดังนั้นค่าของตัวเลขในรีจิสเตอร์จะเป็นจำนวนของรอบเมกซ์ซินไซเคิล และเนื่องจากแต่ละรอบเมกซ์ซินไซเคิลประกอบด้วย 12 คาบของออสซิลเลเตอร์ อัตราการนับแต่ละครั้งจะใช้เวลา 1/12 ของความถี่ออสซิลเลเตอร์

เอกสารนี้ในฟังก์ชันตัวนับรีจิสเตอร์จะเพิ่มค่าทุกครั้งที่มีการเปลี่ยนแปลงสถานะจาก “1” เป็น “0” ที่เข้ามาที่ขา T0 หรือ T1 ในฟังก์ชันนี้สัญญาณภายนอกที่เข้ามาจะถูกสุ่ม (Sampling) ระหว่างช่วงไปใช้

S5P2 ของทรูบอบแมชชีนไซเคิล โดยถ้าสัญญาณเข้ามาเป็นระดับสูงในหนึ่งรอบ ดังนั้น ถ้าในหนึ่งรอบต่อมาของสัญญาณเข้าเป็นระดับต่ำ รีจิสเตอร์จะนับเพิ่มหนึ่งค่า โดยที่ค่าใหม่ของตัวนับ จะปรากฏที่รีจิสเตอร์ช่วง S3P1 ของรอบแมชชีนไซเคิล ซึ่งค่าหนึ่ง ๆ ที่รับเข้าไป จะต้องใช้เวลา 2 แมชชีนไซเคิล (เท่ากับ 24 คาบ) ในการรับค่าช่วงการเปลี่ยน 1 เป็น 0 ดังนั้น ค่าสูงสุดในการนับจะมีอัตรา  $1/24$  ของความถี่ออสซิลเลเตอร์ และสัญญาณอินพุตที่นับนั้นจะไม่มีช่วงระยะห่างที่แน่นอนของคิวต์ไซเคิล แต่จะถูกนับเมื่อระดับแรงดันที่ถูกสุ่มในแต่ละครั้งจะต้องมีช่วงคงที่อย่างน้อย 1 รอบแมชชีนไซเคิลก่อนที่จะเปลี่ยนค่าระดับแรงดันใหม่ในการเลือกทำงานระหว่างไทม์เมอร์กับเคาน์เตอร์จะเลือกได้ 4 โหมด คือ โหมด 0, โหมด 1, โหมด 2 และ โหมด 3

### 2.9.1 โหมด 0

ในโหมด 0 จะเป็นการใช้งานไทม์เมอร์ขนาด 13 บิต โดยจะใช้ไทม์เมอร์ไบต์สูง (THX) ขนาด 8 บิต ต่ออนุกรมกับไทม์เมอร์ไบต์ต่ำ (TLX) อีก 5 บิต (บิต 0-บิต 4) รวมเป็น 13 บิต ส่วนบิตที่เหลือของไทม์เมอร์ต่ำจะไม่ถูกนำมาใช้งาน ลักษณะการทำงานของไทม์เมอร์ โหมด 0 ดังรูปที่ 2.12

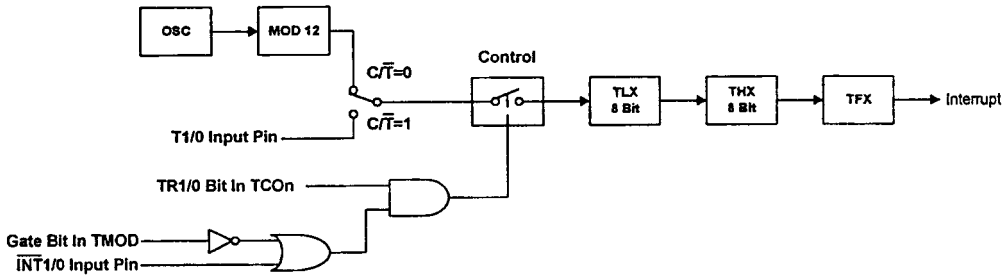


รูปที่ 2.12 การทำงานของไทม์เมอร์โหมด 0

### 2.9.2 โหมด 1

ในโหมด 1 เป็นการใช้งานไทม์เมอร์ขนาด 16 บิต ซึ่งจะทำงานเหมือนกับโหมด 0 (ต่างกันที่โหมด 1 จะใช้ทั้ง 16 บิต) สัญญาณนาฬิกาจะป้อนให้รีจิสเตอร์ไทม์เมอร์ที่เกิดจาก TLX และ THX ต่อกัน เมื่อได้รับสัญญาณนาฬิกา ไทม์เมอร์จะเริ่มนับจาก 0000H ไปเรื่อย ๆ จนถึง FFFFH ในรีจิสเตอร์ไทม์เมอร์ (THX และ TLX) สามารถที่จะทำการอ่านหรือเขียนได้โดยซอฟต์แวร์เช่นเดียวกัน ซึ่งลักษณะการทำงานของไทม์เมอร์ โหมด 0 ดังรูปที่ 2.13

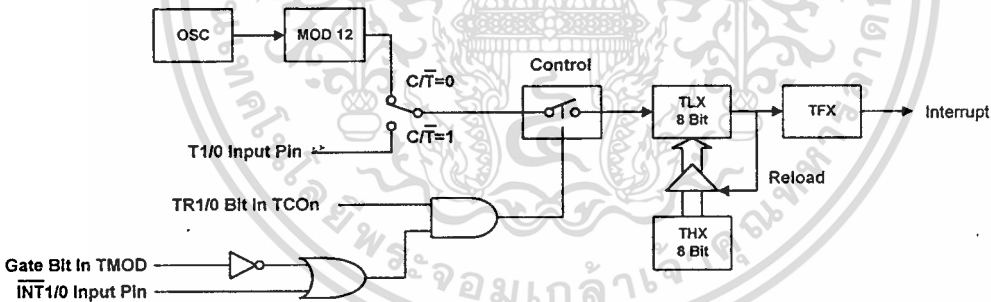
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การทำงานของไทม์เมอร์โหมด 1

### 2.9.3 โหมด 2

ในโหมด 2 จะให้ไทม์เมอร์ทำงานเป็นแบบ 8 บิต โดยจะใช้ไบต์สูงจะใช้เก็บค่าที่ต้องการให้โหลดใหม่เอาไว้เมื่อเกิดโอเวอร์โฟลว์ซึ่งจะเกิดขึ้นเมื่อการนับเปลี่ยนจาก FFH เป็น 00H แฟล็กของไทม์เมอร์จะเซต และค่าใน THX จะถูกโหลดเข้าไปเก็บไว้ใน TLX อัตโนมัติ ซึ่งลักษณะการทำงานของไทม์เมอร์ โหมด 2 ดังรูปที่ 2.14



รูปที่ 2.14 การทำงานของไทม์เมอร์โหมด 2

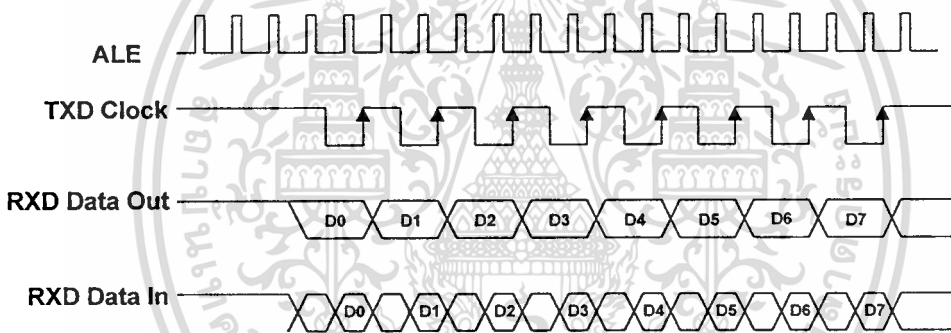
### 2.9.4 โหมด 3

ในโหมด 3 จะเป็นการแยกไทม์เมอร์ให้เป็นอิสระจากกัน โดยในโหมด 3 นี้ไทม์เมอร์ 0 จะถูกแยกเป็นไทม์เมอร์เวลาขนาด 8 บิต 2 ชุด คือ TLO และ TH0 ไทม์เมอร์ทั้ง 2 ชุดจะทำงานแยกจากกันอย่างอิสระ โดยแฟล็ก TF0 จะเซตเมื่อ TLO เกิดการโอเวอร์โฟลว์ และแฟล็ก TF1 จะเซตเมื่อ TH0 เกิดการ โอเวอร์โฟลว์ เช่นกัน ลักษณะการทำงานของไทม์เมอร์ โหมด 3 ดังรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.10.1 โหมด 0

ในการใช้งานโหมด 0 จะต้องกำหนดให้บิต SM0 และ SM1 ที่อยู่ในรีจิสเตอร์ SCON มีค่าเป็น 0 ทั้งคู่ ซึ่งจะทำให้รีจิสเตอร์ SBUF รับหรือส่งข้อมูลได้ขนาด 8 บิต โดยใช้ขา RXD ต่อกับสายสัญญาณที่จะใช้ส่งข้อมูลภายนอก ส่วนขา TXD จะใช้ต่อกับวงจรกำเนิดสัญญาณนาฬิกา ที่อยู่ภายในชิพ MCS-51 โดยใช้ส่งสัญญาณนาฬิกาที่กำหนดความถี่อ้างอิงในการส่งข้อมูลออกมาภายนอก ซึ่งความถี่ที่ส่งออกมาที่ขา TXD จะมีค่าเท่ากับ  $1/12$  ของความถี่ออสซิลเลเตอร์ เมื่อทำการส่งข้อมูลออก ข้อมูลจะถูกเลื่อนออกมาทีละบิตผ่านขา RXD ในเวลาเดียวกัน สัญญาณพัลส์จะถูกส่งออกมาทางขา TXD ด้วย และในเวลาเดียวกัน เมื่อรับข้อมูลเข้าก็จะรับมาทางขา RXD การรับข้อมูลจะเกิดขึ้นเมื่อเซตให้ REN=1 และ RI=0 เมื่อบิต RI ถูกเคลียร์ สัญญาณพัลส์จะถูกส่งออกมาที่ขา TXD เพื่อทำการซิงโครไนซ์กับข้อมูลที่รับเข้ามาที่ขอบด้านบวกของพัลส์ ลักษณะดังรูปที่ 2.16



รูปที่ 2.16 การทำงานของพอร์ตอนุกรมโหมด 0

### 2.10.2 โหมด 1

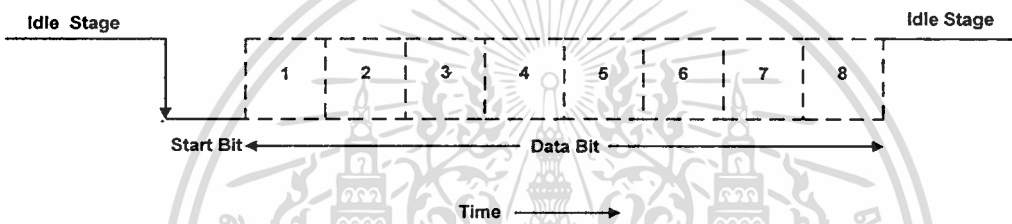
ในการใช้งานโหมด 1 จะต้องกำหนดให้บิต SM0 และบิต SM1 ในรีจิสเตอร์ SCON มีค่าเป็น 0 และ 1 ซึ่งเป็นการกำหนดให้รีจิสเตอร์ SBUF สามารถมีการรับส่งข้อมูลได้ขนาด 10 บิต เป็นรูปแบบการรับส่งข้อมูล 2 ทิศทาง ซึ่งสามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน (ต่างกับโหมด 0 คือจะสามารถรับหรือส่งในเวลาหนึ่ง ๆ ได้เพียงอย่างเดียวเท่านั้น) โดยจะใช้ขา RXD รับสัญญาณอนุกรมที่เข้ามา และ TXD ส่งข้อมูลแบบอนุกรมออกไปภายนอก

การส่งข้อมูลจะเริ่มขึ้นด้วยการส่งบิตเริ่มต้น ออกไปก่อน แล้วตามด้วยบิตข้อมูลอีก 8 บิต จากนั้นจึงส่งบิตหยุด ส่วนแฟล็ก TI จะเซตเมื่อส่งข้อมูลครบทั้ง 10 บิต

เอกสารนี้เป็นการรับข้อมูลจะเริ่มจากขอบขาขึ้นของบิตเริ่มต้น เมื่อครบทั้ง 8 บิต แล้วจะเกิดสภาวะดังนี้ การคำนวณการนับบิตที่ 9 (Stop bit) จะถูกเก็บไว้ใน RB8 ของ SCON ถ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SBUF จะทำการโหลดข้อมูลทั้ง 8 บิต ไปเก็บไว้
- แฟล็ก RI จะเซตเป็น “1” ซึ่ง สิ่งทีกล่าวมาทั้งหมดนี้จะเป็นจริงเมื่อ
- แฟล็ก RI = 0
- SM2 = 1 และ Stop bit ที่รับเข้ามาเป็น 1 หรืออีกกรณีหนึ่งคือ SM2 = 0

ถ้าไม่เกิดทั้ง 2 กรณี การรับข้อมูลก็จะสมบูรณ์ ถ้าเกิดทั้งสองกรณี ตัวบิตหยุดจะไปเก็บที่ RB8 และข้อมูลจะเข้า SBUF และ RI จะแอกทีฟสูง ช่วงเวลานี้ไม่ว่าจะเกิดขึ้นทั้ง 2 กรณีหรือไม่ หน่วยควบคุมการรับข้อมูลจะกลับไปตรวจการเปลี่ยนแปลงสถานะการส่งข้อมูลของ RXD ใหม่ ซึ่งลักษณะการทำงานของพอร์ตอนุกรม ในโหมด 1 ดังรูปที่ 2.17

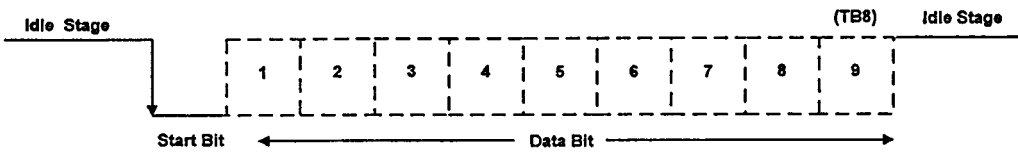


รูปที่ 2.17 การทำงานของพอร์ตอนุกรมโหมด 1

### 2.10.3 โหมด 2

ในการทำงานของโหมด 2 มีการทำงานคล้ายกับโหมด 1 มีความต่าง กันที่โหมด 2 จะเป็นการรับส่งข้อมูลแบบ 11 บิต คือ บิตเริ่มต้น 1 บิต, บิตข้อมูล และ บิตหยุด อีก 1 บิต เมื่อการส่งข้อมูลที่บิตที่ 9 จะได้จากบิต TB8 ในรีจิสเตอร์ SCON ส่วนเวลารับข้อมูลบิตที่ 9 จะถูกเก็บไว้ในบิต RB8

ข้อกำหนดในการเซตบิต RI ในโหมด 2 คือ บิต RI จะต้องเป็น 0 ก่อนบิตสุดท้ายที่จะเข้ามาภายในชิพ MCS-51 และ บิต SM2 จะต้องเป็น 0 (หรือบิตที่ 9 ของข้อมูลจะต้องเป็น 1) การรับส่งข้อมูลในโหมดนี้ส่วนมากจะใช้ในการรับส่งข้อมูลระหว่างไมโครโปรเซสเซอร์ด้วยกัน (ซึ่งจะทำให้การต่อสายสัญญาณเข้ากันโดยตรง) เนื่องจากอัตราการรับส่งข้อมูลจะสูงมากกว่าโหมด 1 โดยลักษณะการทำงานของพอร์ตอนุกรม ในโหมด 2 ดังรูปที่ 2.18



รูปที่ 2.18 การทำงานของพอร์ตอนุกรมโหมด 2

### 2.10.4 โหมด 3

ในการทำงานในโหมด 3 จะมีการทำงานเหมือนกับโหมด 2 ยกเว้นอัตราบอดเท่านั้น โดยจะมีความแน่นอนมากกว่าโหมด 2 โดยซึ่งใช้การคำนวณเหมือนโหมด 1 และจะใช้ตัวจับเวลา 1 ทำการสร้างเวลาที่สำหรับการส่งสัญญาณแทน

### 2.10.5 อัตราความเร็วในการรับส่งข้อมูล (Buad Rate)

อัตราความเร็วของการรับส่งข้อมูลจะมีค่าขึ้นอยู่กับการทำงานในแต่ละโหมดของพอร์ตการสื่อสารอนุกรม ดังนี้

$$\text{อัตราเร็วการรับส่งข้อมูล โหมด 0} = \frac{\text{frequency oscillator}}{12}$$

ในโหมด 2 ค่าความเร็วในการรับส่ง ข้อมูลขึ้นอยู่กับค่าของบิต SMOD ที่อยู่ในรีจิสเตอร์ใช้งานเฉพาะ PCON โดย

บิต SMOD = 0 ค่าความเร็วในการรับส่งข้อมูลจะเป็น 1/64 ของความถี่ออสซิลเลเตอร์ที่ใช้

บิต SMOD = 1 ค่าความเร็วในการรับส่งข้อมูลจะเป็น 1/32 ของความถี่ออสซิลเลเตอร์ที่ใช้

$$\text{อัตราเร็วในการรับส่งข้อมูล โหมด 2} = \frac{[2^{\text{SMOD}} (\text{frequency oscillator})]}{64}$$

อัตราเร็วการรับส่งข้อมูลในโหมด 1 และ 3 จะถูกกำหนดโดยอัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 1 และมีไทม์เมอร์ 2 เป็นตัวกำหนดอัตราเร็วในการรับส่งข้อมูลได้อีกทำให้มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์ที่สามารถกำหนดอัตราการรับส่งข้อมูลรวมจำนวน 2 ตัวโดยอาจใช้ตัวหนึ่งเป็นกำหนดอัตราเร็วสำหรับการรับข้อมูล ส่วนอีกตัวหนึ่งกำหนดอัตราเร็วสำหรับการส่งข้อมูล ทำให้การรับข้อมูลมีค่าอัตราเร็วในการรับส่งข้อมูลที่ต่างกันได้ การใช้ไทม์เมอร์ 1 หรือไทม์เมอร์ 2 มีการกำหนดค่าอัตราเร็วในการรับส่งข้อมูล มีดังนี้

เมื่อไทม์เมอร์ 1 ถูกใช้เป็นตัวกำหนดอัตราเร็วในการรับส่งข้อมูล สำหรับการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 1 และ 3 ค่าของอัตราเร็วในการรับส่งข้อมูลที่ได้จะถูกกำหนดด้วยอัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 1 และขึ้นอยู่กับบิต SMOD ในรีจิสเตอร์ PCON ซึ่งสามารถหาอัตราเร็วในการรับส่งข้อมูล ได้ดังนี้

$$\text{อัตราเร็วในการรับส่งข้อมูล โหมด 1 และโหมด 3} = \frac{[2^{SMOD} (\text{over flow timer})]}{32}$$

เนื่องจากเมื่อเกิดโอเวอร์โฟลวในไทม์เมอร์ตัวใดตัวหนึ่งจะทำให้เกิดสัญญาณอินเตอร์รัพต์เพื่อบอกให้หน่วยประมวลผลกลางทราบ ดังนั้นจะต้องนำไทม์เมอร์ 1 มาเป็นตัวกำหนดอัตราเร็วในการรับส่งข้อมูล จึงไม่ควรให้เกิดการอินเตอร์รัพต์ขึ้นในระหว่างการรับหรือส่งข้อมูล

การใช้งานพอร์ตสื่อสารแบบอนุกรมที่พบบากนั้น ไทม์เมอร์จะถูกกำหนดการทำงานเป็นไทม์เมอร์ในโหมด 2 (Auto Reload) ในกรณีนี้อัตราเร็วในการรับส่งข้อมูลจะถูกกำหนดโดยสมการดังนี้

$$\text{อัตราเร็วในการรับส่งข้อมูล โหมด 1 และโหมด 3} = \frac{[2^{SMOD} (\text{frequency ocillator})]}{32 * 12 * [256 - (TH1)]}$$

## 2.11 การอินเตอร์รัพต์

โปรแกรมคอมพิวเตอร์มีเพียง 2 วิธีเท่านั้นที่จะหาเงื่อนไขที่เป็นจริงของวงจรภายในและภายนอกวิธีแรก คือ ใช้คำสั่งซอฟต์แวร์ กระโดดบนสภาวะแฟลก และพอร์ตพิน วิธีที่สอง คือ สัญญาณฮาร์ดแวร์ที่เรียกว่า อินเตอร์รัพต์ ซึ่งทำให้โปรแกรมไปเรียกโปรแกรมย่อย เทคนิคของซอฟต์แวร์ใช้เวลาของโปรเซสเซอร์ ซึ่งสามารถไปใช้งานอื่น ๆ อินเตอร์รัพต์จะใช้เวลาของโปรเซสเซอร์เมื่อถูกทำเมื่อต้องการใช้โปรแกรมประยุกต์การใช้งานเกือบทั้งหมดของไมโครคอนโทรลเลอร์เกี่ยวข้องกับเหตุการณ์ ที่เร็วพอจะควบคุมภาวะแวดล้อมที่สร้างเหตุการณ์

ในชิพ MCS-51 มีอินเตอร์รัพต์ 5 ชนิด มี 3 ชนิดจะถูกสร้างโดยฮาร์ดแวร์ โดยไทม์เมอร์แฟลก 0, ไทม์เมอร์แฟลก 1 และ อินเตอร์รัพต์พอร์ตอนุกรม (RI หรือ TI) อินเตอร์รัพต์อีก 2 ตัวถูกทริกโดยสัญญาณภายนอกโดยวงจรที่ต่อกับขา  $\overline{INT0}$  และ  $\overline{INT1}$  หน้าที่ของการอินเตอร์รัพต์จะอยู่ภายใต้การควบคุมของโปรแกรมมอนิเตอร์ ซึ่งสามารถที่จะเปลี่ยนบิตการควบคุมในรีจิสเตอร์ IE, รีจิสเตอร์ IP และรีจิสเตอร์ TCON โปรแกรมมอนิเตอร์สามารถหยุดการอินเตอร์รัพต์ทั้งหมดหรือบางตัวจากโปรแกรมโดยการรีเซต หรือ การเคลียร์บิตในรีจิสเตอร์ เหล่านี้หลังจากอินเตอร์รัพต์ถูกจัดการโดยโปรแกรมย่อย ซึ่งทำโดยผู้เขียนโปรแกรมลงในหน่วยความจำโปรแกรม โปรแกรมการอินเตอร์รัพต์ต้องทำงานที่คำสั่งที่เกิดการอินเตอร์รัพต์โปรแกรมถูกทำโดยเก็บค่า PC ไปบนสแตกในหน่วยความจำข้อมูลก่อนเปลี่ยนรีจิสเตอร์ PC ซึ่งเป็นค่าตำแหน่งของการอินเตอร์รัพต์ในหน่วยความจำโปรแกรม คำรีจิสเตอร์ PC จะได้จากสแตกหลังจากคำสั่ง RETI ได้ทำแล้วที่ส่วนท้ายของโปรแกรมย่อย การอินเตอร์รัพต์สามารถแบ่งตามลักษณะการเกิดได้ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.11.1 อินเทอร์รัพต์แฟลกไทม์เมอร์

เมื่อไทม์เมอร์/เคาน์เตอร์ เกิดโอเวอร์โฟลว์ ผลคือ ไทม์เมอร์แฟลก (TF0 และ TF1) จะเซตเป็น 1 แฟลกถูกเคลียร์เป็น 0 เมื่ออินเทอร์รัพต์ทำให้โปรแกรมทำให้โปรแกรมเรียกโปรแกรมย่อยของไทม์เมอร์ในหน่วยความจำโปรแกรม

### 2.11.2 การอินเทอร์รัพต์ฟอร์ตอนุกรม

เมื่อทำการรับข้อมูลแล้ว บิตอินเทอร์รัพต์ (RI) ในรีจิสเตอร์ SCON ต้องเซตเป็น 1 เมื่อทำการส่งข้อมูลเรียบร้อยแล้ว บิตอินเทอร์รัพต์ (TI) ต้องเซต และจะนำมา OR กันเพื่อหาตัวอินเทอร์รัพต์แก่หน่วยประมวลผลกลาง ซึ่งเป็นการอินเทอร์รัพต์ฟอร์ตอนุกรมบิตเหล่านี้จะไม่ถูกเคลียร์เมื่อมีการเรียกโปรแกรมอินเทอร์รัพต์ถูกทำโดยหน่วยประมวลผลกลาง โปรแกรมที่จัดการสื่อสารข้อมูลออนุกรมต้องรีเซตบิต RI หรือ TI เพื่อจะทำข้อมูลถัดไป

### 2.11.3 การอินเทอร์รัพต์ภายนอก

ขา  $\overline{INT0}$  และ  $\overline{INT1}$  จะใช้โดยวงจรภายนอก การอินพุตบนขาเหล่านี้จะต้องเซตแฟลก IE0 และแฟลก IE1 ในรีจิสเตอร์ SCON เป็น 1 โดยมี 2 วิธี แฟลก  $\overline{INTX}$  บิต IT0 และ บิต IT1 จะต้องโปรแกรมขา  $\overline{INTX}$  สำหรับการอินเทอร์รัพต์ที่ 0 และเมื่อเซตเป็น 0 จะต้องทำการโปรแกรมขา  $\overline{INTX}$  สำหรับอินเทอร์รัพต์ที่การเปลี่ยนลอจิกเมื่อเซตเป็น 1

แฟลก IEX จะทำการรีเซตเมื่อการอินเทอร์รัพต์ที่เกิดจากการเปลี่ยนลอจิก พบโดยการโปรแกรมและเข้าถึงโปรแกรมย่อย วงจรภายนอกต้องเป็น 1 ก่อนที่บิต RETI จะถูกจัดความผิดพลาดในเรื่องนี้จะทำให้เกิดการอินเทอร์รัพต์ทันทีหลังจากบิต RETI ด้วยแหล่งอินเทอร์รัพต์เดียวกัน

### 2.11.4 การควบคุมการอินเทอร์รัพต์

โปรแกรมต้องสามารถหยุดการอินเทอร์รัพต์ทั้งหมดหรือบางตัว เพื่อให้งานที่สำคัญได้ทำจนเสร็จ รีจิสเตอร์ IE จะเก็บค่าบิตที่โปรแกรมได้ เพื่อให้ทำการอินเทอร์รัพต์ได้ตามต้องการ และเมื่อเลือกอินเทอร์รัพต์แล้ว แหล่งอินเทอร์รัพต์แต่ละแหล่งอาจถูกเลือกหรือไม่ก็ได้

ในการที่ต้องการอินเทอร์รัพต์ที่สำคัญกว่าโดยทันที บิตของรีจิสเตอร์ IP อาจจะถูกเซตโดยโปรแกรมเพื่อกำหนดความสำคัญของแต่ละแหล่งอินเทอร์รัพต์ เพื่อให้อินเทอร์รัพต์ที่สำคัญกว่าทำการอินเทอร์รัพต์ก่อน เมื่อมีการอินเทอร์รัพต์พร้อมกัน 2 แหล่งขึ้นไป

## 2.12 การเข้าถึงข้อมูลของ MCS- 51

เอกสารนี้วิธีการเข้าถึงข้อมูลในรีจิสเตอร์ในไมโครคอนโทรลเลอร์ MCS-51 มี 5 วิธี ดังต่อไปนี้ ขั้นตอนการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.12.1 การเข้าถึงข้อมูลในรีจิสเตอร์

เป็นการกำหนดเลขที่อยู่แบบรีจิสเตอร์ โดยจะให้ข้อมูลเข้าถึงโดยการใช้รีจิสเตอร์กลุ่มที่ถูกเลือกจากการติดตั้งใช้งานในกลุ่มรีจิสเตอร์เบงก์นั้น ๆ โดยแต่ละเบงก์จะมี 8 รีจิสเตอร์ และใช้บิตต่ำสุด 3 บิตแรกของรหัสคำสั่งเป็นตัวกำหนดเลือกรีจิสเตอร์ใช้งานตัวใดตัวหนึ่งใน 8 ตัว ทั้ง 3 บิตนี้จะรวมอยู่ในออปโค้ดกับโอเปอร์เรนด์ ตำแหน่งรวมอยู่ในรูปแบบตำแหน่งสั้น ๆ เพียงหนึ่งไบต์

MOV A, R0

### 2.12.2 การเข้าถึงข้อมูลโดยตรง

เป็นการกำหนดเลขที่อยู่โดยตรงเป็นวิธีเดียวที่เข้าถึงข้อมูลทางฮาร์ดแวร์รีจิสเตอร์ เช่น กลุ่มของรีจิสเตอร์ SFR และสามารถกำหนดเลขที่อยู่โดยตรงบริเวณตำแหน่งต่าง ๆ ของแรมภายในจำนวน 128 ไบต์ ด้วยการใช้ไบต์โอโปรเรนด์ตัวต่อจากออปโค้ดของคำสั่ง เป็นตัวกำหนดตำแหน่งที่ถูกใช้

MOV A, 40H

### 2.12.3 การเข้าถึงข้อมูลที่อยู่ในรีจิสเตอร์โดยอ้อม

เป็นการใช้ค่าข้อมูลที่อยู่ R0 หรือ R1 ในเบงก์ที่ถูกติดตั้งให้ทำงานเท่านั้นจะเป็นค่าดัชนีหรือ ตัวชี้ตำแหน่งข้อมูลภายใน 256 ไบต์ แบ่งเป็นบล็อกค่า จำนวน 128 ไบต์ของแรมภายใน และบล็อกสูงอีก 128 ไบต์ของหน่วยความจำข้อมูลภายใน การกำหนดให้ข้อมูลแรมภายในที่สามารถกำหนดตัวชี้ตำแหน่งด้วยค่า R0 และ R1 ด้วยการเลือกการทำงานนี้รหัสคำสั่งออปโค้ดที่บิตหลักต่ำสุด (LSB)

MOV A, @R0

### 2.12.4 การเข้าถึงข้อมูลโดยทันที

เป็นการใช้ค่าคงที่ที่โปรแกรมอยู่ในหน่วยความจำข้อมูลหรือหน่วยความจำโปรแกรม และไม่สามารถเปลี่ยนแปลงค่าเหล่านี้ได้ในระหว่างการทำงานตามโปรแกรมในภาษาแอสเซมบลี การเรียกใช้งานที่กระทำในโหมด การกำหนดเลขที่อยู่จะมีรูปแบบของการเขียนด้วยสัญลักษณ์ “#” ภายในโอโปรเรนด์ฟิวด์ด้วยค่าที่ทำสัญลักษณ์นี้สามารถเขียนเป็นตัวเลขหรือใช้สัญลักษณ์ชื่อตัวแปร หรือใช้สูตรคำนวณทางคณิตศาสตร์ที่ให้ค่าออกมาคงที่จำนวนหนึ่ง

MOV A, #12

### 2.12.5 การเข้าถึงข้อมูลโดยใช้ ตัวชี้อ้างอิง

ในการทำงานของคำสั่งนี้จะใช้ค่าของรีจิสเตอร์ใช้งานเฉพาะด้าน DPTR หรือ PC มารวมกับค่าในรีจิสเตอร์ A เพื่อชี้ไปยังตำแหน่งของหน่วยความจำสำหรับเก็บข้อมูลโปรแกรมซึ่งเก็บข้อมูลไว้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นค่าในรีจิสเตอร์ใช้งานเฉพาะ DPTR หรือ PC จะต้องมีย้ายเท่ากับตำแหน่งต้นของหน่วยความจำ ส่วนที่เก็บข้อมูลที่ต้องการ ส่วนค่าในรีจิสเตอร์ A จะเป็นตัวเลือกข้อมูลที่อยู่ในหน่วยความจำ

```
MOV A,@A+DPTR
```

```
MOV A,@A+PC
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบและการสร้าง

#### 3.1 กล่าวนำ

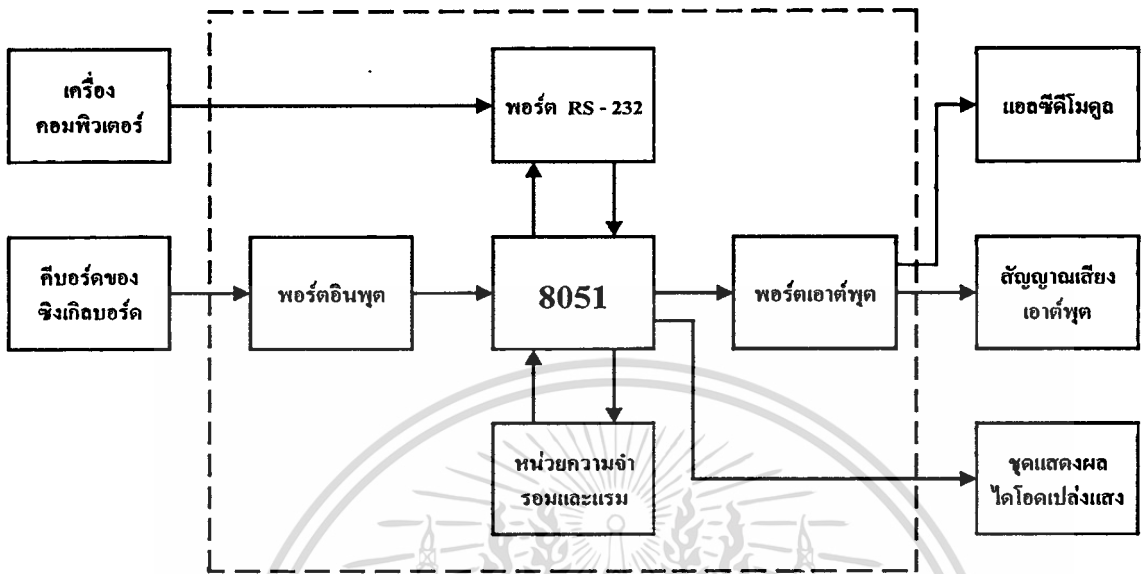
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ผลิตขึ้นโดยบริษัทอินเทล เบอร์ที่นิยมใช้ในงานควบคุมกัน ได้แก่ เบอร์ 8031, 8032, 8051, 8052, 8751 และ 8752 ทุกเบอร์จะมีพอร์ตอินและพอร์ตเอาต์ชุด 4 พอร์ต โดยข้อดีของไมโครคอนโทรลเลอร์ตระกูลนี้ สามารถที่จะเขียนโปรแกรมควบคุมการทำงานได้สะดวก และมีเครื่องมือช่วยในการพัฒนาระบบมาก ทำให้บริษัทอื่น ๆ พยายามผลิตไมโครคอนโทรลเลอร์ที่มีลักษณะการทำงานเหมือนกับ MCS-51 โดยสามารถใช้ได้ในซอฟต์แวร์คอมพิวเตอร์ตัวเดียวกัน

ผู้จัดทำเห็นว่า 8051 ไมโครคอนโทรลเลอร์ เป็นไมโครคอนโทรลเลอร์ในตระกูล MCS-51 มีประสิทธิภาพการทำงานสูง จึงได้นำ 8051 ไมโครคอนโทรลเลอร์นี้ มาทำการออกแบบและสร้างเป็นลักษณะของชุดซิงเกิลบอร์ดไมโครคอนโทรลเลอร์ เพื่อใช้ในการศึกษาและเรียนรู้คำสั่งการทำงานของตัวไมโครคอนโทรลเลอร์เบอร์ 8051 พร้อมทั้งเขียนโปรแกรมมอนิเตอร์ควบคุมการทำงานของระบบ ชุดไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวที่สร้างขึ้นจะมีส่วนประกอบต่าง ๆ ทางด้าน ฮาร์ดแวร์และซอฟต์แวร์ คือ ส่วนของวงจรประกอบรวมต่าง ๆ และ ส่วนโปรแกรมมอนิเตอร์ที่ใช้ควบคุมการทำงาน โดยส่วนประกอบในส่วนต่าง ๆ ด้านฮาร์ดแวร์มี ดังนี้ คือ ส่วนจอแสดงผลแบบผลึกเหลวขนาด 16 ตัวอักษร 2 แถว เป็นจอการแสดงผลบนแผ่นพิมพ์เดี่ยว, ส่วนการแสดงผลทางไดโอดเปล่งแสงจำนวน 8 ดวง ที่ต่ออยู่กับพอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์, ส่วนการรับข้อมูลคีย์บอร์ดของชุดไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวจะเป็นแบบเมทริกซ์ (Matrix) ขนาด 6 x 4, ส่วนการถอดรหัสภายนอกของอุปกรณ์อินพุตและอุปกรณ์เอาต์พุต โดยจะเชื่อมต่อกับส่วนรับข้อมูลคีย์บอร์ดของแผ่นพิมพ์เดี่ยว, ส่วนแสดงผลทางลำโพง และ ส่วนสุดท้ายคือ ส่วนควบคุมการรับส่งข้อมูลทางพอร์ตอนุกรม RS-232 และ พอร์ตผู้ใช้งาน

#### 3.2 การพัฒนาโปรแกรมของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

จากรูปที่ 3.1 เป็นผังการทำงานของ 8051 ไมโครคอนโทรลเลอร์ โดยสามารถที่จะทำการพัฒนาโปรแกรม หรือ การทำงานได้ 2 ลักษณะ คือ

1. การพัฒนาโปรแกรมบนแผ่นพิมพ์เดี่ยว
  2. การพัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ผังการทำงานของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

### 3.2.1 การพัฒนาโปรแกรมบนแผ่นพิมพ์เดี่ยว

การใช้งาน 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวจะต้องทำการเขียนโปรแกรมอยู่ในรูปของภาษาเครื่องบนคีย์บอร์ดของแผ่นพิมพ์เดี่ยว โดยจะประกอบด้วยคีย์คำสั่งควบคุมการทำงานในส่วนต่าง ๆ เช่น คีย์คำสั่ง, คีย์ข้อมูล และ คีย์ฟังก์ชัน เป็นต้น ซึ่งจะอำนวยความสะดวกในการเขียนโปรแกรม

### 3.2.2 การพัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์

การใช้งานจะต้องเขียนโปรแกรมที่อยู่ในรูปของภาษาแอสเซมบลีบนคีย์บอร์ดของเครื่องคอมพิวเตอร์ และทำการประมวลผลโปรแกรมผ่านทางพอร์ตการรับส่งข้อมูลอนุกรมโดยใช้พอร์ต RS-232

## 3.3 ส่วนประกอบโดยทั่วไป

การทำงานของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว มีส่วนประกอบที่เป็นส่วนฮาร์ดแวร์และซอฟต์แวร์ โดยการทำงานด้านฮาร์ดแวร์มีการทำงานด้วยกัน 8 ส่วน ดังตารางที่ 3.1 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.1 การทำงานของส่วนประกอบภายในไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

| ส่วนประกอบ                              | คำอธิบาย                                                                                                                                      |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 1. ส่วนชุดควบคุม                        | เป็นส่วนควบคุมการทำงานในส่วนต่าง ๆ ของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว ตามโปรแกรมมอนิเตอร์ที่เขียนขึ้น โดยใช้ 8051 ไมโครคอนโทรลเลอร์ |
| 2. ส่วนเลือกสัญญาณแอนาเดรส              | เป็นส่วนเลือกสัญญาณสำหรับการติดต่อกับ อุปกรณ์ต่าง ๆ ให้ทำงาน                                                                                  |
| 3. ส่วนคีย์บอร์ด                        | เป็นส่วนรับค่าข้อมูลที่ป้อนให้กับซิงเกิลบอร์ด โดยค่าที่รับมานี้จะมี 2 รหัส คือ รหัสคำสั่งและ รหัสข้อมูล                                       |
| 4. ส่วนจอแสดงผลแบบผลึกเหลว              | เป็นส่วนการใช้แสดงฟังก์ชันการทำงาน, แสดงการรับข้อมูลจากผู้ใช้งาน และเป็นหน่วยแสดงผลทางจอภาพของแผ่นพิมพ์เดี่ยว                                 |
| 5. ส่วนการแสดงผลทางลำโพง                | ทำหน้าที่แสดงผลอยู่ซึ่งอยู่ในรูปของเสียง ออกทางลำโพงขณะไมโครคอนโทรลเลอร์ทำงาน                                                                 |
| 6. ส่วนการรับส่งข้อมูลทางพอร์ตอนุกรม    | เป็นส่วนที่ใช้เชื่อมต่อเพื่อรับสัญญาณควบคุมการทำงานทางพอร์ตอนุกรม โดยใช้ RS 232 ในการรับส่งข้อมูลของเครื่องคอมพิวเตอร์ PC                     |
| 7. ส่วนไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่ง | เป็นส่วนการแสดงผลของข้อมูลที่พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์                                                                             |
| 8. ส่วนของแหล่งจ่ายไฟ                   | ทำหน้าที่แปลงแรงดันไฟฟ้าให้มีขนาดแรงดันไฟตรงขนาด 5 โวลต์ เพื่อป้อนให้กับส่วนต่าง ๆ ภายในวงจรประกอบรวมของแผ่นพิมพ์เดี่ยว                       |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

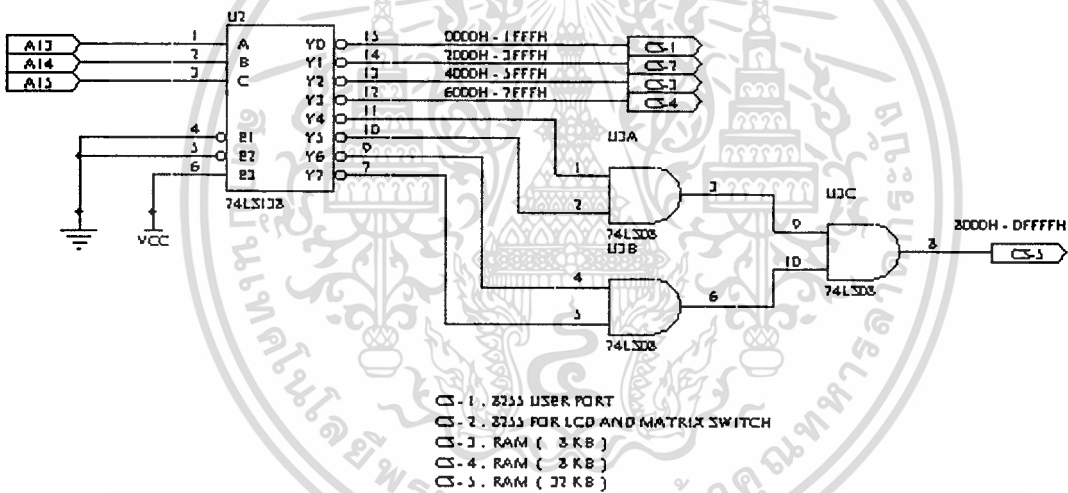
### 3.4 การออกแบบและการทำงานในส่วนต่าง ๆ

#### 3.4.1 ส่วนชุดควบคุม

วงจรส่วนควบคุมทั้งหมดคณะผู้จัดทำโครงการ ได้เสนออยู่ในส่วนที่ภาคผนวก ก

#### 3.4.2 ส่วนเลือกสัญญาณแอดเดรส

เป็นส่วนในการเลือกสัญญาณสำหรับการติดต่อกับอุปกรณ์ภายในวงจรทั้งหมด ซึ่งสัญญาณที่ได้นี้ มาจากสัญญาณขาแอดเดรสของ 8051 ไมโครคอนโทรลเลอร์ ซึ่งจะต่อกับใช้ไอซี 74LS138 โดยเป็นตัวเลือกสัญญาณว่าจะให้อุปกรณ์ต่อร่วมตัวใดทำงานโดยจะนำสัญญาณเอาต์พุตที่ได้ไปต่อเข้ากับ  $\overline{CS}$  ของอุปกรณ์ต่อร่วมเพื่อใช้ในการควบคุมการทำงานของอุปกรณ์นั้น ๆ ให้ทำงานได้ตามต้องการโดยลักษณะของวงจรเลือกสัญญาณแอดเดรส ดังรูปที่ 3.2



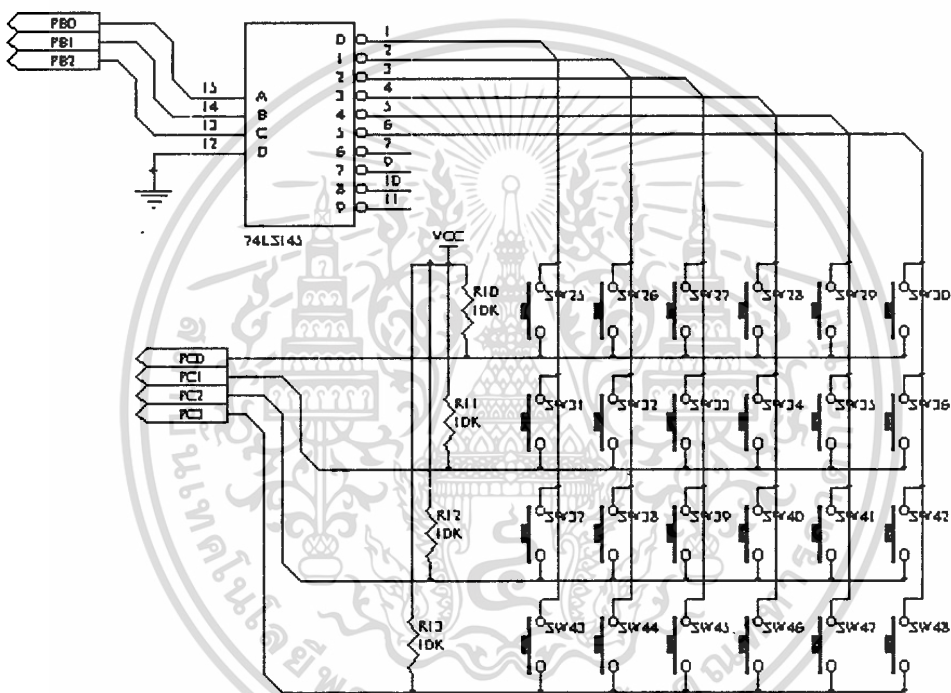
รูปที่ 3.2 วงจรส่วนเลือกสัญญาณแอดเดรส

#### 3.4.3 ส่วนคีย์บอร์ด

การออกแบบวงจรส่วนคีย์บอร์ด โดยลักษณะการทำงานจะใช้พอร์ต B และพอร์ต C เป็นพอร์ตอินพุตและพอร์ตเอาต์พุต คือ พอร์ต PB0-PB2 จะเป็นอินพุตให้กับไอซี 74LS145 และพอร์ต PC0-PC3 จะเป็นเอาต์พุต การทำงานจะใช้งานร่วมกับการควบคุมทางด้านซอฟต์แวร์ ซึ่งวงจรส่วนคีย์บอร์ดดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรในรูปที่ 3.3 จะใช้วิธีการสแกน เพื่อตรวจสอบการทำงานของสวิตช์ ถ้าทำการกดโดยเริ่มจากคอลัมน์ “0” จะมีผลทำให้เกิดการส่งค่าลอจิก “0” มาที่พอร์ต์เอาต์พุต ทำให้ไอซี 74LS145 ซึ่งเป็นตัวสแกนคอลัมน์ เมื่อได้รับลอจิก “0” ทางอินพุตจะทำให้ไอซี 74LS145 ให้สถานะเอาต์พุต Y0 เป็นลอจิก “0” ส่วนในคอลัมน์อื่นจะเป็น “1” แล้วจะทำการตรวจสอบทีละแถว โดยเริ่มจากแถวที่ 0 ก่อนซึ่งจะตรวจสอบว่ามีระดับลอจิกเป็น “0” หรือไม่ ถ้าเกิดมีการกดสวิตช์ เป็นผลแสดงว่าสวิตช์ตำแหน่งนี้ถูกกด แต่ถ้าไม่มีการกดสวิตช์ก็จะทำการเลื่อนไปแถวถัดไป



รูปที่ 3.3 วงจรแสดงวงจรส่วนคีย์บอร์ด

จากนั้นจะทำการตรวจสอบเช่นเดิม และจะทำการตรวจสอบทั้งด้านคอลัมน์และแถวถัดไปเรื่อย ๆ จนกว่าจะหมด หรือจนกระทั่งพบว่ามิสวิตช์ถูกกด และเมื่อพบว่าสวิตช์ใดถูกกดแล้วจะต้องทราบอีกว่าสวิตช์นั้น คือ สวิตช์ตำแหน่งใดเพื่อจะได้แปลค่าความหมายหรือสร้างรหัสคำสั่งขึ้นตามค่าของสวิตช์ตำแหน่งนั้น ๆ เพื่อนำไปใช้งานต่อไป

#### 3.4.4 ส่วนจอแสดงผลแบบผลึกเหลว

ในส่วนของการแสดงผลของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว ได้ทำการออกแบบการแสดงผลเป็นแบบจอผลึกเหลว โดยจอผลึกเหลวนี้นี้จะเป็นขนาด 16 ตัวอักษร 2 บรรทัด ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ซึ่งในโมดูลของจอผลึกเหลวประกอบด้วย 3 ส่วน ดังนี้

### 1. คอเทมทริกซ์ ในจอผลึกเหลว (Dot Matrix LCD)

เป็นส่วนการแสดงผลให้ผู้ใช้งานได้มองเห็นการทำงานของไมโครคอนโทรลเลอร์ โดยเป็นการทำงานในลักษณะการปิดและเปิดตัวเองกับแสง คือ ส่วนที่เป็นกระจกบรรจุผลึกเหลว

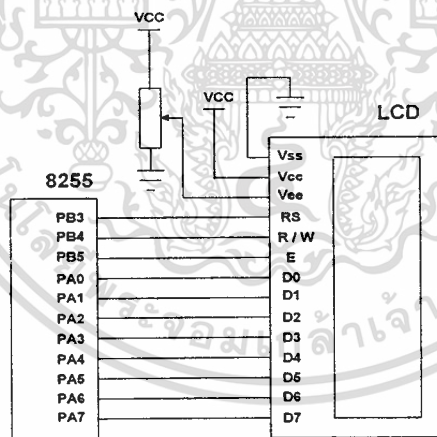
### 2. ไดรเวอร์ (Driver)

เป็นส่วนที่รับสัญญาณจากภาคการควบคุมมาทำการขับส่วนการแสดงผลทางจอผลึกเหลว

### 3. ตัวควบคุม (Controller)

เป็นส่วนที่รับข้อมูลจากอุปกรณ์ต่อรวมภายนอกและการควบคุมจอแสดงผลแบบผลึกเหลวให้ทำงานแสดงผลต่าง ๆ เช่น การเกิดตัวอักษร, การลบหน้าจอ เป็นต้น

ในการออกแบบจะนำเอาสายข้อมูลผ่านทางพอร์ต PA และพอร์ต PB ของ 8255 และทำการเชื่อมต่อกับ 8051 ไมโครคอนโทรลเลอร์เพื่อส่งสัญญาณควบคุมและสัญญาณข้อมูลที่ใช้ในการแสดงผลและการอ่านข้อมูลต่าง ๆ กับจอแสดงผลแบบผลึกเหลว โดยขา D0-D7 ซึ่งเป็นขาข้อมูลของจอแสดงผล ในการปรับเปลี่ยนระดับความสว่างและความเข้มของจอแสดงผล สามารถทำได้ โดยการปรับค่าตัวต้านทานปรับค่าได้ 10 กิโลโอห์ม ซึ่งลักษณะของวงจรส่วนจอแสดงผลแบบผลึกเหลว ดังรูปที่ 3.4

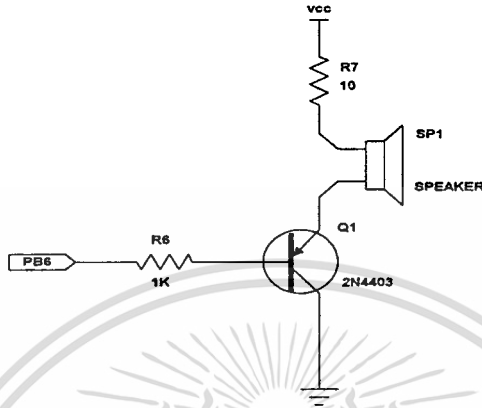


รูป ที่ 3.4 วงจรจอแสดงผลแบบผลึกเหลวขนาด 16 ตัวอักษร 2 บรรทัด

### 3.4.5 ส่วนการแสดงผลทางลำโพง

ในส่วนการแสดงผลทางลำโพง ลักษณะของเสียงที่เกิดขึ้นจะใช้พอร์ต PB6 ของ 8255 เป็นตัวสร้างสัญญาณการขยาย โดยจะประกอบด้วยทรานซิสเตอร์ 2N4403 เป็นตัวทำหน้าที่ขยายสัญญาณการกำเนิดเสียง ทำได้โดยการส่งข้อมูลให้พอร์ต PB6 ในรูปของพัลส์ ซึ่งลักษณะของวงจรการแสดงผลทางลำโพง ดังรูปที่ 3.5 ห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำเนิดเสียง ทำได้โดยการส่งข้อมูลให้พอร์ต PB6 ในรูปของพัลส์ ซึ่งลักษณะของวงจรการแสดงผลทางลำโพง ดังรูปที่ 3.5



รูปที่ 3.5 วงจรส่วนการแสดงผลทางลำโพง

### 3.4.6 ส่วนการรับส่งข้อมูลทางพอร์ตอนุกรม

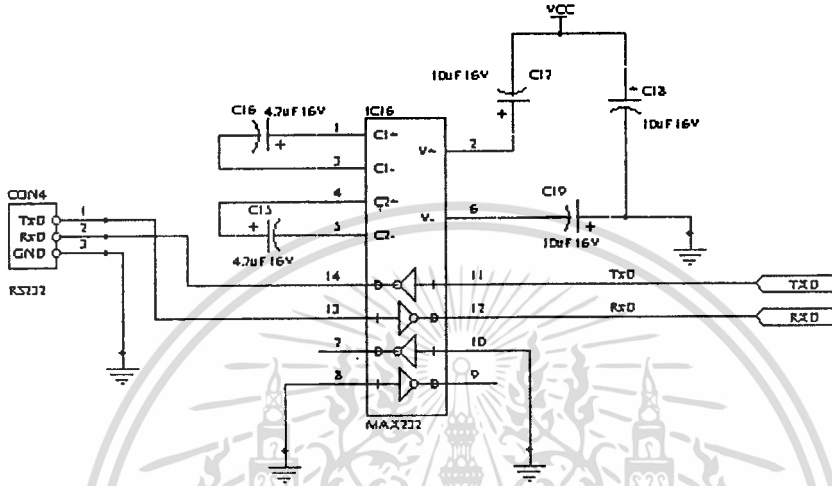
ในส่วนประกอบของการรับส่งข้อมูลทางพอร์ตอนุกรม จะเป็นการรับข้อมูลและส่งข้อมูล ในลักษณะกลุ่มของบิต คราวละหนึ่งบิต เรียงลำดับกันไปเรื่อย ๆ โดยที่พอร์ตอนุกรมของ 8051 ไมโครคอนโทรลเลอร์ มีโครงสร้างการทำงานในแบบ 2 ทิศทาง (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกันโดยทางค่านางจรของตัวส่ง (Transmitter) ประกอบด้วยข้อมูลออกไปยัง พอร์ตอนุกรม ทางขาสัญญาณ TXD (พอร์ต 3.1) ส่วนวงจรด้านตัวรับ (Receiver) ประกอบด้วย SBUF เช่นเดียวกันสัญญาณข้อมูลที่ได้รับเข้ามาทางขาสัญญาณ RXD (พอร์ต 3.0) ซึ่งลักษณะของวงจรการรับส่งข้อมูลทางพอร์ตอนุกรมมีลักษณะ ดังรูปที่ 3.6

### 3.4.7 ส่วนไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่ง

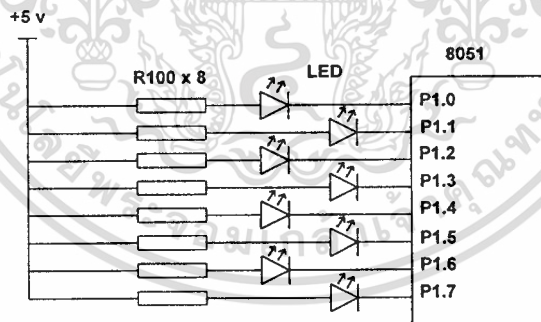
ในส่วนประกอบของการแสดงผลทางไดโอดเปล่งแสง จะทำการต่อเข้าที่พอร์ตหนึ่งของตัว 8051 ไมโครคอนโทรลเลอร์ โดยทำการต่อขั้วแอนโอดของตัวไดโอดเปล่งแสงเข้ากับแหล่งจ่ายไฟบวก 5 โวลต์โดยมีค่าตัวต้านทาน 100 โอห์มจำกัดกระแส และขั้วคาโทดของไดโอดเปล่งแสงต่อเข้ากับพอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์ แต่ในการเขียนโปรแกรมการส่งข้อมูลให้ทำการแสดงผลทางไดโอดเปล่งแสงจะต้องทำการกลับค่าของข้อมูลก่อนที่จะทำการส่ง ส่วนวงจรไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่ง ดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผลทางไดโอดเปล่งแสงจะต้องทำการกลับค่าของข้อมูลก่อนที่ทำการส่ง ส่วนวงจรไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่ง ดังรูปที่ 3.7



รูปที่ 3.6 วงจรส่วนการรับส่งข้อมูลทางพอร์ตอนุกรม

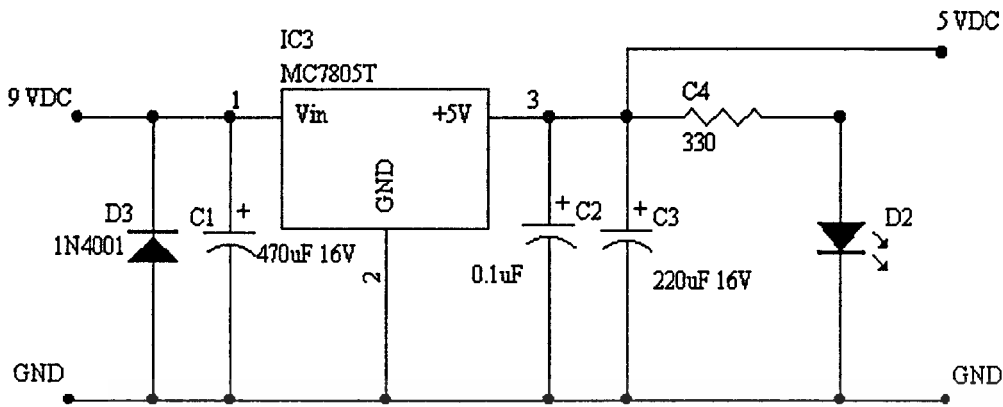


รูปที่ 3.7 วงจรไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่งของ 8051

### 3.4.8 ส่วนของแหล่งจ่ายไฟ

ในส่วนประกอบของวงจรแหล่งจ่ายไฟ จะประกอบด้วยวงจรรักษาระดับแรงดันโดยจะใช้ไอซีเรกูเลเตอร์ เบอร์ MC7405T ซึ่งจะให้ค่าของแรงดันเอาต์พุตเป็นแรงดันไฟตรงขนาด 5 โวลต์ เพื่อจ่ายให้กับตัวอุปกรณ์ในส่วนต่าง ๆ ภายในแผ่นพิมพ์เขียวทั้งหมด โดยลักษณะของวงจรแหล่งจ่ายไฟ ดังรูปที่ 3.8

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรแสดงส่วนของแหล่งจ่ายไฟ

### 3.5 การทำงานในส่วนของโปรแกรมมอนิเตอร์

ในการออกแบบการทำงานด้านซอฟต์แวร์นั้นสามารถแบ่งการทำงานออกเป็น ส่วน ๆ คือ ส่วนของโปรแกรมหลัก, ส่วนการสแกนค่าของคีย์บอร์ด และส่วนการนำโปรแกรมไปประมวลผล

#### 3.5.1 ส่วนการทำงานของโปรแกรมหลัก

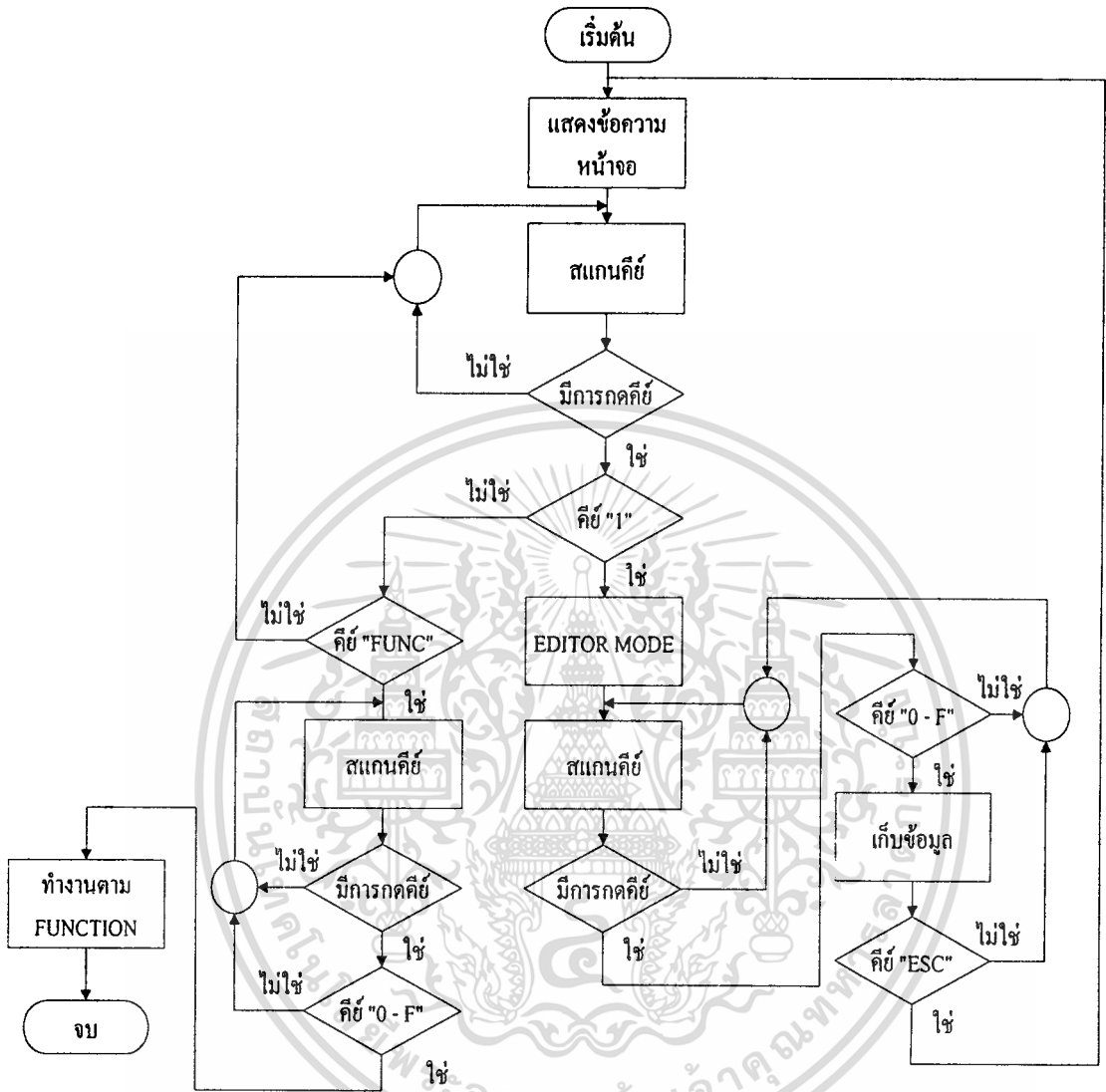
การทำงานของโปรแกรมหลักจะมีการทำงานส่วนย่อยอีกมาก ที่นำมาประกอบกัน ซึ่งทำให้ตัวโปรแกรมหลักนั้นสามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งการทำงานของโปรแกรมหลักจะแสดงในผังการทำงานของโปรแกรมหลัก ดังรูปที่ 3.9

#### 3.5.2 ส่วนโปรแกรมการสแกนคีย์บอร์ด

การทำงานในส่วนของโปรแกรมสแกนคีย์บอร์ด จะมีขั้นตอนการทำงานที่ซับซ้อนมาก เพราะต้องคอยรับค่าของคีย์ต่าง ๆ เพื่อทำการตรวจสอบค่าของคีย์บอร์ด ว่าเป็นคีย์คำสั่งหรือคีย์ข้อมูล ซึ่งการทำงานของโปรแกรมการสแกนคีย์บอร์ด นั้นมีการทำงาน ดังผังการทำงาน รูปที่ 3.10

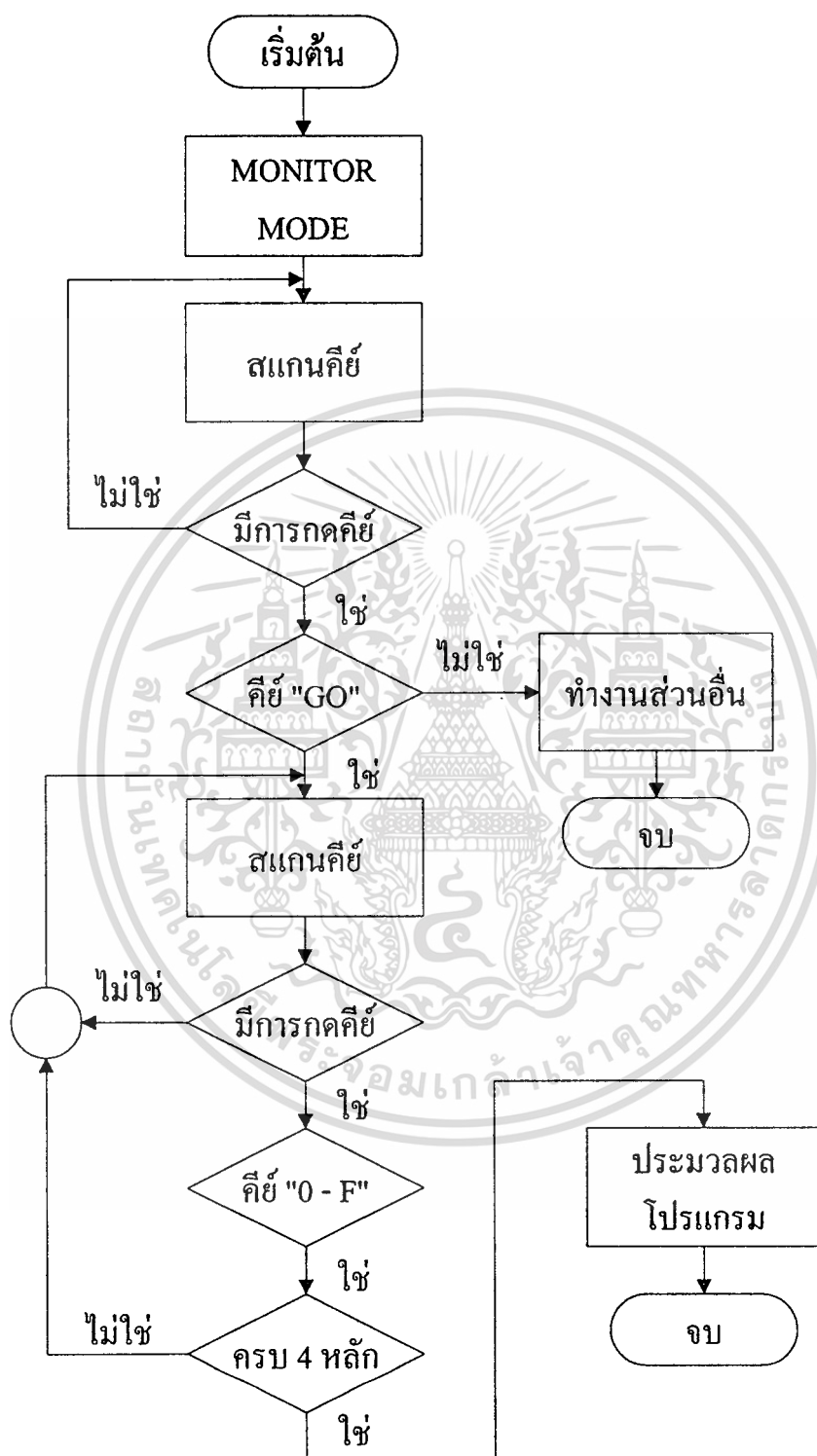
#### 3.5.3 ส่วนโปรแกรมการประมวลผล

โปรแกรมการประมวลผลเป็นโปรแกรมที่มีความสำคัญมาก ซึ่งจะมีหน้าที่ในการประมวลผลโปรแกรมของผู้ใช้งาน ที่ทำการป้อนคำสั่งให้กับ 8051 ไมโครคอนโทรลเลอร์ทำงาน ตามที่ผู้ใช้งานต้องการ โดยโปรแกรมการประมวลผลมีขั้นตอนการทำงาน ดังผังการทำงาน รูปที่ 3.11



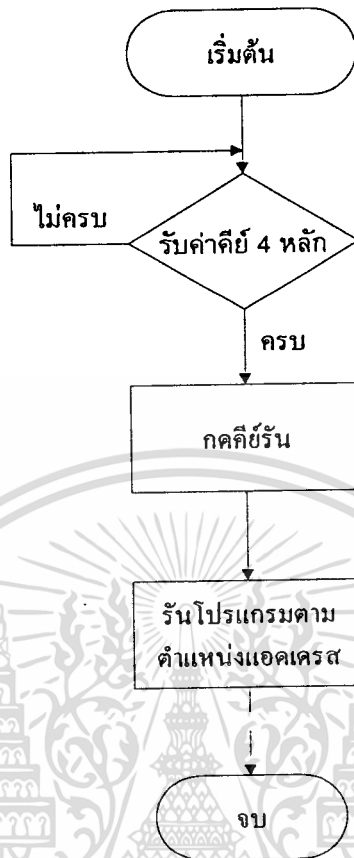
รูปที่ 3.9 ผังการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ฟังก์ชันการทำงานการสแกนคีย์บอร์ด

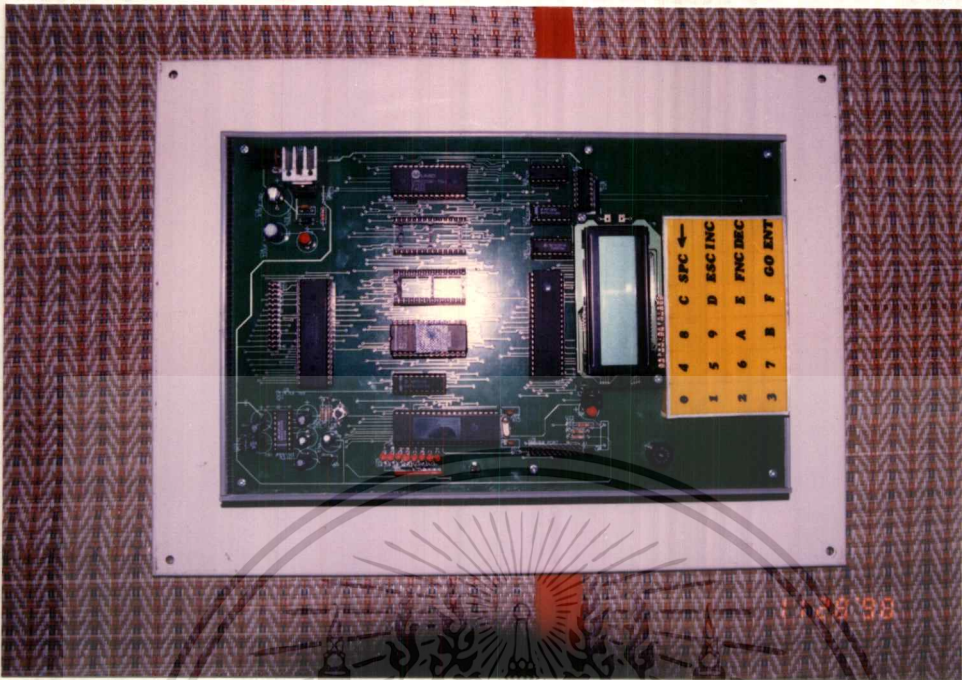
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



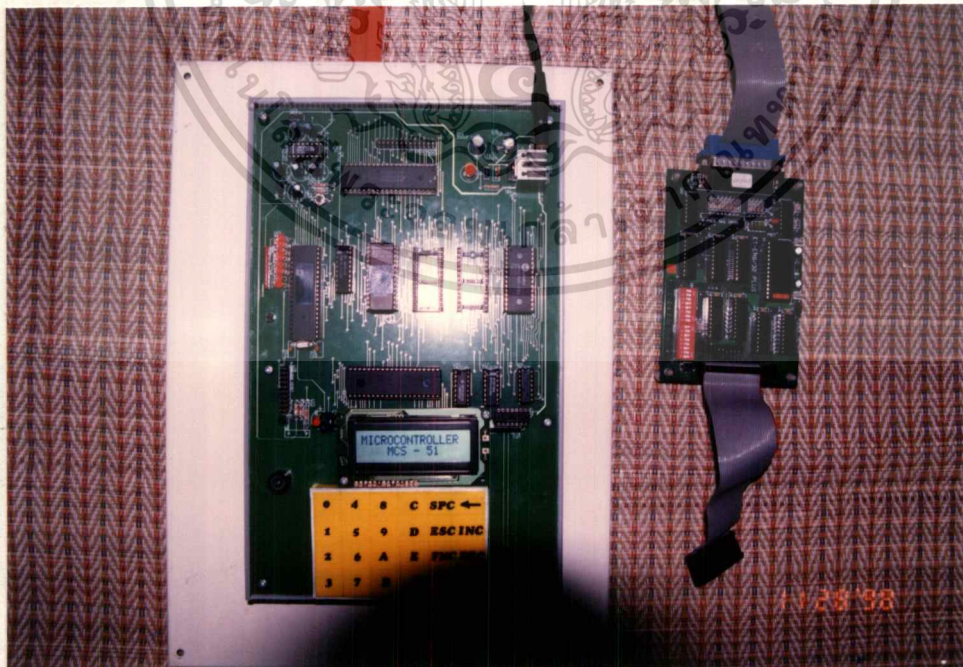
รูปที่ 3.11 ผังการทำงานของ การประมวลผลทางโปรแกรม

### 3.6 การประกอบวงจรส่วนต่าง ๆ

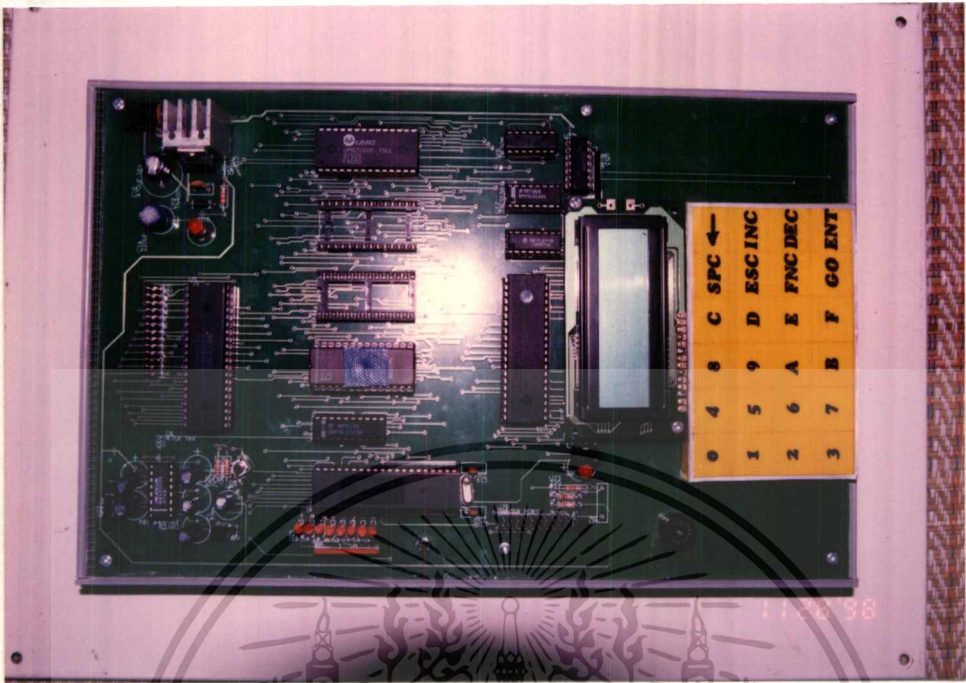
จากการทำงานในส่วนต่าง ๆ นำมาทำการออกแบบเป็นแผ่นพิมพ์เดี่ยว และส่งไปให้ทางร้านทำลายวงจรพิมพ์ แล้วทำการประกอบวงจรในส่วนต่าง ๆ ในแผ่นพิมพ์เดี่ยวให้ครบสมบูรณ์ ซึ่งเมื่อทำการประกอบวงจรครบจะมีลักษณะของ ชุด 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว ดังรูปที่ 3.12, ส่วนในการต่อการรับส่งข้อมูลทางพอร์ตอนุกรมของตัว 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวกับคอมพิวเตอร์ PC จะมีลักษณะ ดังรูปที่ 3.13 , ส่วนไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่งของไมโครคอนโทรลเลอร์ เมื่อทำการส่งข้อมูลให้ทำการแสดงผล จะมีลักษณะดังรูปที่ 3.14 และ ส่วนของโปรแกรมมอนิเตอร์ ที่ใช้ในการควบคุมส่วนต่าง ๆ ของวงจรจะมีลักษณะของโปรแกรมมอนิเตอร์ ดังรูปที่ 3.15



รูปที่ 3.12 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีรูปที่ 3.13 การ์ด 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวกับพอร์ตอนุกรมที่มีการนำไปใช้



รูปที่ 3.14 พอร์ต 1 ของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิใช้รูปที่ 13.5 โปรแกรมมอนิเตอร์ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลองและการทดสอบ

การทำงานของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว สามารถทำการสั่งงานได้จากคีย์บอร์ดบนแผ่นพิมพ์เดี่ยว โดยการเขียนโปรแกรมสั่งงาน ผลการทดลองในการเขียนโปรแกรมสั่งงานที่เขียนไว้นี้ ในการทดลองได้แบ่งการทดลองออกเป็นส่วน ๆ ซึ่งแบ่งได้ 4 ส่วน ดังนี้

1. การส่งข้อมูลออกทางพอร์ตหนึ่งของไมโครคอนโทรลเลอร์ 8051
2. การเขียนและอ่าน ข้อมูลลงในหน่วยความจำข้อมูล
3. การทำงานของพอร์ตผู้ใช้งาน
4. การรับส่งข้อมูลทางพอร์ตอนุกรม

#### 4.1 การทดลองและผลการทดลอง

การเขียนโปรแกรมทดลองการทำงานของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว มีลำดับขั้นตอนการทดลอง ดังนี้

1. ทำการจ่ายไฟให้ไมโครคอนโทรลเลอร์
2. ที่จอแสดงผลแบบผลึกเหลว จะปรากฏข้อความ “ SINGLE BOARD ” ที่แถวบนและมีข้อความ “ VERSION 1.00 ” ที่แถวล่าง จากนั้นจะมีข้อความ “ ED. ENGINEER # 19 ” ที่แถวบนและมีข้อความ “ ELEC&COM 2/2 ” ที่แถวล่าง
3. รอจนจอแสดงผลแบบผลึกเหลวปรากฏข้อความ “ MICROCONTROLLER ” ที่แถวบนและมีข้อความ “ MCS-51 ” ที่แถวล่าง ให้ทำการกดคีย์หมายเลข “ 1 ” เพื่อเข้าสู่ Editor Program
4. ตั้งค่า Addr ที่จะทำการป้อนโปรแกรมแล้วกดคีย์ “ Enter ” ทำการป้อนโปรแกรมซึ่งจะอยู่ในรูปแบบของภาษาเครื่อง เมื่อป้อนเสร็จเรียบร้อยแล้ว ต้องการออกจากโหมด Editor Program ให้ทำการกดคีย์ “ ESC ” จะทำให้กลับเข้าสู่โปรแกรมมอนิเตอร์
5. กดคีย์ “ GO ” แล้วตั้งค่า Addr ที่ป้อนข้อมูลไว้ แล้ว กดคีย์ “ Enter ” หน่วยประมวลผลของไมโครคอนโทรลเลอร์จะทำการประมวลผลโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.2 โปรแกรมการทดลองที่ 1

ป้อนโปรแกรมตาม โปรแกรมการทดลองที่ 1 แล้วทำการสังประมวลผล บันทึกผลการทดลอง

EXAM1.ASM

```

1          ;*****
2          ;*
3          ;* PROGRAM TEST PORT 1.x / MCS-51 *
4          ;*
5          ;*****
6
8000 7      ORG 8000H
8
8000 7401 9      MOV A,#01H
8002 F4   10     LOOP: CPL A
8003 F590 11     MOV P1,A
8005 12800C 12    CALL DELAY
8008 F4   13     CPL A
8009 23   14     RL A
800A 80F6 15     SJMP LOOP
16
800C 78FF 17     DELAY: MOV R0,#0FFH
800E 79FF 18     DELAY1: MOV R1,#0FFH
8010 D9FE 19             DJNZ R1,$
8012 D8FA 20             DJNZ R0,DELAY1
8014 22   21             RET
22
0000 =    23             END

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำเอกสารนี้ไปเผยแพร่หรือใช้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์

ผลการทดลองโปรแกรมการทดลองที่ 1 เมื่อทำการประมวลผลหลังจากการป้อนโปรแกรม จะมีการแสดงผลของไดโอดเปล่งแสงที่พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์ โดยโปรแกรม

การทดลองที่ 1 นี้จะเป็นโปรแกรมวนลูป ลักษณะการแสดงผลของไดโอดเปล่งแสงจะสว่างทีละดวง แล้ววิ่งจากซ้ายไปขวา

#### 4.1.2 โปรแกรมการทดลองที่ 2

ป้อนโปรแกรมตาม โปรแกรมการทดลองที่ 1 แล้วทำการสังประมวลผล บันทึกผลการทดลอง

EXAM2.ASM

```

1          ;*****
2          ;*                                     *
3          ;* PROGRAM TEST DATARAM *
4          ;* ( WRITE / READ ) *
5          ;*                                     *
6          ;*****
7
9000=      8          RAM EQU 9000H
9
8000      10         ORG 8000H
11
8000 909000 12        MOV DPTR,#RAM
8003 7400   13        MOV A,#0H
14
8005 F0     15        LOOP: MOVX @DPTR,A
8006 04     16         INC A
8007 A3     17         INC DPTR
8008 B410FA 18        CJNE A,#10H,LOOP
19
800B 909000 20        LOOP1: MOV DPTR,#RAM
800E E0     21        LOOP2: MOVX A,@DPTR
800F F4     22         CPL A
8010 F590   23        MOV PI,A
8012 F4     24         CPL A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

8013 A3      25      INC    DPTR
8014 12801F  26      CALL   DELAY
8017 B40FF4  27      CJNE   A,#0FH,LOOP2
801A 7590FF  28      MOV    P1,#0FFH
801D 80EC    29      SJMP   LOOP1
          30
801F 78FF    31      DELAY: MOV   R0,#0FFH
8021 79FF    32      DELAY1: MOV  R1,#0FFH
8023 D9FE    33      DJNZ   R1,$
8025 D8FA    34      DJNZ   R0,DELAY1
8027 22      35      RET
          36
0000=       37      END

```

ผลการทดลองโปรแกรมการทดลองที่ 2 เมื่อทำการประมวลผลหลังจากการป้อนโปรแกรมการทดลองที่ 2 โปรแกรมจะนำข้อมูลจากหน่วยความจำข้อมูลที่ตำแหน่ง 9000H ถึง 900FH มาทำการแสดงผลที่ไดโอดเปล่งแสงที่พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์ ซึ่งมีค่า 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E และ 0F

### 4.1.3 โปรแกรมการทดลองที่ 3

ทำการต่อพอร์ตผู้ใช้งานกับ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพีไฟฟ์เดี่ยว และทำการป้อนโปรแกรมการทดลองที่ 3 ทำการประมวลผลโปรแกรม และ บันทึกผลการทดลอง

EXAM3.ASM

```

1      ;*****
2      ;*
3      ;*   PROGRAM TEST USERPORT   *
4      ;*       ( 8255 USER PORT )   *
5      ;*
6      ;*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|             |    |                |
|-------------|----|----------------|
| 6000=       | 9  | CONA EQU 6000H |
| 6001=       | 10 | CONB EQU 6001H |
| 6002=       | 11 | CONC EQU 6002H |
| 6003=       | 12 | CTRL EQU 6003H |
|             | 13 |                |
| 8000        | 14 | ORG 8000H      |
|             | 15 |                |
| 8000 7480   | 16 | MOV A,#80H     |
| 8002 906003 | 17 | MOV DPTR,#CTRL |
| 8005 F0     | 18 | MOVX @DPTR,A   |
|             | 19 |                |
| 8006 74AA   | 20 | MOV A,#0AAH    |
| 8008 906000 | 21 | MOV DPTR,#CONA |
| 800B F0     | 22 | MOVX @DPTR,A   |
|             | 23 |                |
| 800C 74AA   | 24 | MOV A,#0AAH    |
| 800E 906001 | 25 | MOV DPTR,#CONB |
| 8011 F0     | 26 | MOVX @DPTR,A   |
|             | 27 |                |
| 8012 74AA   | 28 | MOV A,#0AAH    |
| 8014 906002 | 29 | MOV DPTR,#CONC |
| 8017 F0     | 30 | MOVX @DPTR,A   |
|             | 31 |                |
| 0000=       | 32 | END            |

ผลการทดลองโปรแกรมการทดลองที่ 3 เมื่อทำการประมวลผลหลังจากการป้อนโปรแกรมการทดลองที่ 3 ไมโครคอนโทรลเลอร์จะแสดงผลข้อมูลออกทางไดโอดเปล่งแสงแสดงผลของ 8255 ที่พอร์ต A ซึ่งมีค่า แสดงเป็น AAH, ที่พอร์ต B จะแสดงค่าเป็น AAH และ ที่พอร์ต C จะแสดงค่าเป็น AAH ตามลำดับไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.3 โปรแกรมการทดลองที่ 4

ป้อนโปรแกรมการทดลองที่ 4 และกดคีย์ “GO” ทำการประมวลผลโปรแกรมการทดลอง จากนั้นทำการคอมไพล์ โปรแกรม Serial1.C ซึ่งเป็นโปรแกรมภาษาซี ซึ่งโปรแกรม Serial1.C นี้ จะทำหน้าที่ส่งค่าเป็น รหัสแอสกี ออกทางพอร์ตอนุกรม จนกว่าจะครบตามจำนวนที่กำหนด และจะทำการจบโปรแกรมการทำงาน สังเกตผลการทดลอง และบันทึกผลการทดลอง

EXAM4.ASM

```

1          ;*****
2          ;*
3          ;*   PROGRAM TEST SERIAL PORT   *
4          ;* SEND DATA FROM KEYBOARD (PC) *
5          ;*   TO PORT 1.x/MCS-51        *
6          ;*
7          ;*****
8
9          ; INPUT = SERAIL PORT
10         ;      = DPTR
11         ; OUT  = A
12
13
14 9000=    RAM   EQU  9000H
15
16 8000     16          ORG   8000H
17 8000 909000 17          MOV  DPTR,#RAM
18 8003 759852 18          MOV  SCON,#52H
19 8006 758920 19          MOV  TMOD,#20H
20 8009 758DFD 20          MOV  TH1,#0FDH
21 800C D28E   21          SETB  TR1
22 800E 3098FD 22  LOOP:  JNB   RI,LOOP
23 8011 C298   23          CLR   RI
24 8013 E599   24          MOV  A,SBUF

```

```

8015 F0      25      MOVX @DPTR,A
8016 F4      26      CPL    A
8017 F590    27      MOV    P1,A
8019 F4      28      CPL    A
801A A3      29      INC     DPTR
801B B40DF0 30      CJNE   A,#0DH,LOOP
801E 909000 31      MOV     DPTR,#RAM
8021 E0      32  LOOP1: MOVX   A,@DPTR
8022 F4      33      CPL    A
8023 F590    34      MOV    P1,A
8025 F4      35      CPL    A
8026 A3      36      INC     DPTR
8027 112E    37      ACALL  DELAY
8029 B40DF5 38      CJNE   A,#0DH,LOOP1
802C 80FE    39      SJMP   $
                40
802E 78FF    41  DELAY: MOV    R0,#0FFH
8030 79FF    42  DELAY1: MOV    R1,#0FFH
8032 D9FE    43      DJNZ   R1,$
8034 D8FA    44      DJNZ   R0,DELAY1
8036 22      45      RET
                46
0000=        47      END

```

ส่วนของโปรแกรม Serial1.C มีลักษณะการเขียนโปรแกรม ดังนี้

```

#include <bios.h>
#include <conio.h>
#include <stdio.h>

```

#define COM 0

เอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

#define DATA\_READY 0x100 ให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define TRUE    1
#define FALSE   0

#define SETTINGS ( 0xE0 | 0x00 | 0x00 | 0x03)

/* the communications port is set to

9600 baud  (0xE0 = 9600)

Odd parity (0x08 = ODD,0x18 = EVEN,0x00 = NONE)

1 stop bit (0x00 = 1 STOP,0x04 = 2 STOP BIT)

8 data bits (0x03 = 8 DATA BIT,0x02 = 7 DATA BIT) */

int main(void)
{
    int in, out,DONE = FALSE,i,data[10]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0x00};
    clrscr();
    bioscom(0, SETTINGS, COM);
    printf("TRANSMISSION\n");
    i=0;
    while (!DONE)
    { if (kbhit())
        { if ((in = getch()) == 27)
            DONE = TRUE;
            bioscom(1,in,COM);
        }
    }
    return 0;
}

```

ผลการทดลองโปรแกรมการทดลองที่ 4 เมื่อทำการประมวลผลหลังจากการป้อนโปรแกรมการทดลองที่ 4 ในการกดคีย์บอร์ดของคอมพิวเตอร์ คำรหัสแอสกีของคีย์บอร์ดที่กด จะไปแสดงผลที่แอลอีดีแสดงผลที่พอร์ตหนึ่ง จากนั้นเมื่อทำการกดคีย์ “Enter” ที่คีย์บอร์ดของคอมพิวเตอร์ ส่วนการแสดงผลที่พอร์ตหนึ่งจะทำการแสดงผล เป็นคำรหัสแอสกี ที่ทำการกดคีย์ตั้งแต่ตัวแรกจนถึงตัวสุดท้ายที่กดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 สรุปผลการทดลอง

จากผลการทดลองทั้ง 4 โปรแกรมสรุปผลการทดลองได้ว่า ในการใช้งานพอร์ตหนึ่งของ 8051 ซึ่งเก็ลบอร์ดไมโครคอนโทรลเลอร์ สามารถใช้งานได้ตามปกติ ส่วนหน่วยความจำข้อมูล สามารถทำการอ่านและเขียนข้อมูลเข้าไปได้ โดยดูผลการทดลองจากโปรแกรมการทดลองที่ 2 ส่วนพอร์ตผู้ใช้งาน 8255 ซึ่งต่อใช้งานในลักษณะของเอาต์พุตพอร์ต สามารถใช้งานได้ และการรับส่งข้อมูลอนุกรมสามารถใช้งานได้ โดยดูผลการทดลองจากโปรแกรมทดลองที่ 4 ซึ่งเป็นการรับค่ารหัสแอสกี จากคีย์บอร์ดของเครื่องคอมพิวเตอร์ นำค่าที่ได้ไปแสดงผลที่พอร์ตหนึ่ง



## บทที่ 5

### สรุป อภิปรายและข้อเสนอแนะ

#### 5.1 บทสรุป

8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวในปริยฐานิพนธ์ฉบับนี้ สร้างขึ้นเพื่อนำไปใช้ในการศึกษาและเรียนรู้คำสั่งการทำงานต่าง ๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งเป็น ไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพสูงตัวหนึ่ง ในการสร้างคณะผู้จัดทำได้วางขอบเขตการสร้างและการพัฒนาเพื่อเป็นแนวทางในการออกแบบไว้ดังนี้

1. ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นหน่วยประมวลผลกลาง

2. การพัฒนาโปรแกรมสามารถป้อนโปรแกรมโดยแปลงโปรแกรมที่เขียนขึ้นเป็นภาษาเครื่องก่อน แล้วป้อนผ่านทางคีย์บอร์ดบนแผ่นพิมพ์เดี่ยว ให้หน่วยประมวลผลกลางประมวลผล หรือจะพัฒนาโปรแกรมผ่านทางเครื่องคอมพิวเตอร์ โดยเขียนเป็นภาษาแอสเซมบลีบนเครื่องคอมพิวเตอร์แล้วส่งผ่านทางพอร์ตอนุกรมโดยใช้ RS-232 ไปยังไมโครคอนโทรลเลอร์เพื่อทำการประมวลผล

3. ใช้หน่วยความจำโปรแกรมขนาด 32 กิโลไบต์ในเก็บข้อมูลโปรแกรมมอนิเตอร์ของไมโครคอนโทรลเลอร์

4. การแสดงผลของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว จะแสดงผลบนจอผลึกเหลวแบบ 16 ตัวอักษร 2 แถว โดยอาศัยวงจรดีโคเดอร์เป็นตัวเลือกการแสดงผลและควบคุมการแสดงผลของอุปกรณ์

#### 5.2 ปัญหาและแนวทางแก้ไข

ในระหว่างการทำปริยฐานิพนธ์ คณะผู้จัดทำได้พบปัญหาและอุปสรรคต่าง ๆ ที่ทำให้สูญเสียเวลาในการแก้ไขข้อบกพร่องผิดพลาดเหล่านั้น บางปัญหาอาจจะมองดูเล็กน้อย แต่ถ้าไม่ทำการแก้ไขจะเป็นผลทำให้เสียเวลามากในการแก้ไขภายหลัง โดยปัญหาต่าง ๆ และแนวทางแก้ไข ได้ทำการสรุปผล ดังตารางที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 ปัญหาและแนวทางการแก้ไขปัญหาในการทำปริญญานิพนธ์

| ปัญหา                                                                                                                                                                                                                                                                       | แนวทางแก้ไข                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. ไดโอดเปล่งแสงแสดงผลที่พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว ซึ่งในการออกแบบครั้งแรกได้ทำการต่อโดยให้ ขั้วอาโนดต่อเข้ากับพอร์ตและขั้วคาโทดต่อลงกราวด์ ผลการส่งข้อมูลมาทำการแสดงผลทำให้ ไดโอดเปล่งแสง นั้นมีความสว่างน้อยมาก                              | 1. ทำการกลับขั้วของไดโอดเปล่งที่แสดงผลให้ ขั้วของคาโทดต่อกับพอร์ตหนึ่งและขั้วอาโนด ต่อเข้ากับแหล่งจ่ายไฟบวก 5 โวลต์โดยมีค่า ความต้านทาน 100 โอห์มจำกัดกระแส ผลของ การส่งข้อมูลมาทำการแสดงผลทำให้ไดโอด เปล่งแสงมีความสว่างมากขึ้น แต่ในการเขียน โปรแกรมแสดงผลการทำงาน จะต้องทำการ คอมพิไรมินต์ค่าข้อมูลก่อนที่จะทำการส่งข้อมูล ออกที่พอร์ต |
| 2. ในการออกแบบทำแผ่นลายวงจรพิมพ์โดยใช้ โปรแกรม Protel for win 1.5 (Print Circuit Board) เป็นโปรแกรมที่ใช้ในการออกแบบ เมื่อ ทำการส่งไปให้ทางร้านทำแผ่นลายวงจรพิมพ์ ทางร้านไม่สามารถที่จะทำได้ เนื่องจากผิดข้อกำหนดค่าสุดในการทำแผ่นลายวงจรพิมพ์ของ ทางร้าน                   | 2. ก่อนที่จะทำการออกแบบลายวงจรพิมพ์โดยใช้โปรแกรม Protel for win 1.5 (Print Circuit Board) เป็นโปรแกรมที่ใช้ในการออกแบบ จะต้องไปทำการติดต่อทางร้านทำแผ่นลายวงจร พิมพ์ก่อนออกแบบ และทำการสอบถามข้อกำหนดค่าสุดในการทำแผ่นลายวงจรพิมพ์ของ ทางร้าน เพื่อให้ทราบข้อกำหนดต่าง ๆ ของ ทางร้าน                                                      |
| 3. ในการออกแบบลายวงจรพิมพ์ โดยใช้ โปรแกรม Protel for win 1.5 (Print Circuit Board) เป็นโปรแกรมที่ใช้ในการออกแบบ เมื่อ ใช้คำสั่งให้ทำการออกแบบโดยอัตโนมัติ จะเกิด ข้อผิดพลาดขึ้นมากและมีผลทำให้ลายวงจรที่ได้ มีขนาดใหญ่เกินไป ทำให้ไม่สามารถที่นำไป สร้างแผ่นลายวงจรพิมพ์ได้ | 3. ในการออกแบบลายวงจรพิมพ์ โดยใช้ โปรแกรม Protel for win 1.5 (Print Circuit Board) เป็นโปรแกรมใช้ในการออกแบบ เมื่อใช้ คำสั่งให้ทำการออกแบบโดยอัตโนมัติ จะ สามารถใช้คำสั่งนี้ได้เพียงบางส่วนเท่านั้นและ ต้องทำการลากเส้นเอง จึงจะเกิดความสมบูรณ์ พร้อมที่นำไปสร้างแผ่นลายวงจรพิมพ์ได้                                                      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 5.1 (ต่อ) ปัญหาและแนวทางการแก้ไขปัญหาในการทำปริญญานิพนธ์

| ปัญหา                                                                                                                                                                                                                                                        | แนวทางแก้ไข                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4. ในการเขียนโปรแกรมมอนิเตอร์ควบคุมการทำงาน บางฟังก์ชันนั้นมีข้อผิดพลาดอยู่ ซึ่งมีผลทำให้โปรแกรมเกิดการผิดพลาดและส่งผลให้โปรแกรมมอนิเตอร์ควบคุมการทำงาน เกิดการผิดพลาดจากความเป็นจริง ทำให้ในช่วงแรกของการเขียนโปรแกรมมีข้อผิดพลาดมาก จึงทำงานไม่ค่อยคืบหน้า | 4. แก้ไขการเขียนโปรแกรมให้สมบูรณ์มากยิ่งขึ้น โดยศึกษาการเขียนโปรแกรมจากโปรแกรมมอนิเตอร์ของไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว รุ่น JAZZ-31 แล้วนำมาดัดแปลงใช้ร่วมด้วยในการเขียนโปรแกรม และขอคำแนะนำการเขียนโปรแกรม จากอาจารย์ผู้มีประสบการณ์ในการเขียนโปรแกรมบางส่วน |

## 5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

คณะผู้จัดทำได้พยายามจัดทำและสร้าง 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวขึ้นให้มีประสิทธิภาพและสมบูรณ์ที่สุดเท่าที่จะทำได้โดยตลอดภาคการศึกษา ได้ทำการพัฒนามาการทำงานอย่างต่อเนื่อง แต่ระยะเวลาและงบประมาณที่มีอยู่จำกัดตลอดจนขาดประสบการณ์ในการเขียนโปรแกรมมอนิเตอร์เพื่อควบคุมการทำงานทำให้โปรแกรมมอนิเตอร์นั้นยังมีข้อบกพร่องอยู่ในบางส่วน ซึ่งจากการเขียนโปรแกรมมอนิเตอร์ที่ผ่านมาจึงมีข้อคิดและข้อเสนอแนะว่า ควรที่จะศึกษาข้อมูลการเขียนโปรแกรมมอนิเตอร์ของทางบริษัทได้จัดทำและผลิตออกมาจำหน่าย ซึ่งบริษัทเหล่านี้ได้ทำการพัฒนาและทดลองมาเป็นระยะเวลาหลายปี จนเกิดความสมบูรณ์และมีประสิทธิภาพทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์ ดังนั้นสำหรับผู้เขียนโปรแกรมมอนิเตอร์จึงควรที่จะอาศัยโปรแกรมของบริษัทเหล่านี้เป็นแนวทาง ในการดัดแปลงแก้ไขมาเป็นโปรแกรมมอนิเตอร์ที่ต้องการจะทำให้ได้โปรแกรมมอนิเตอร์ที่สมบูรณ์และใช้เวลาไม่นาน

ที่ผ่านมาในการจัดทำชุด 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว ได้พบข้อบกพร่องที่ควรได้รับการแก้ไขให้มีประสิทธิภาพต่อไปดังนี้

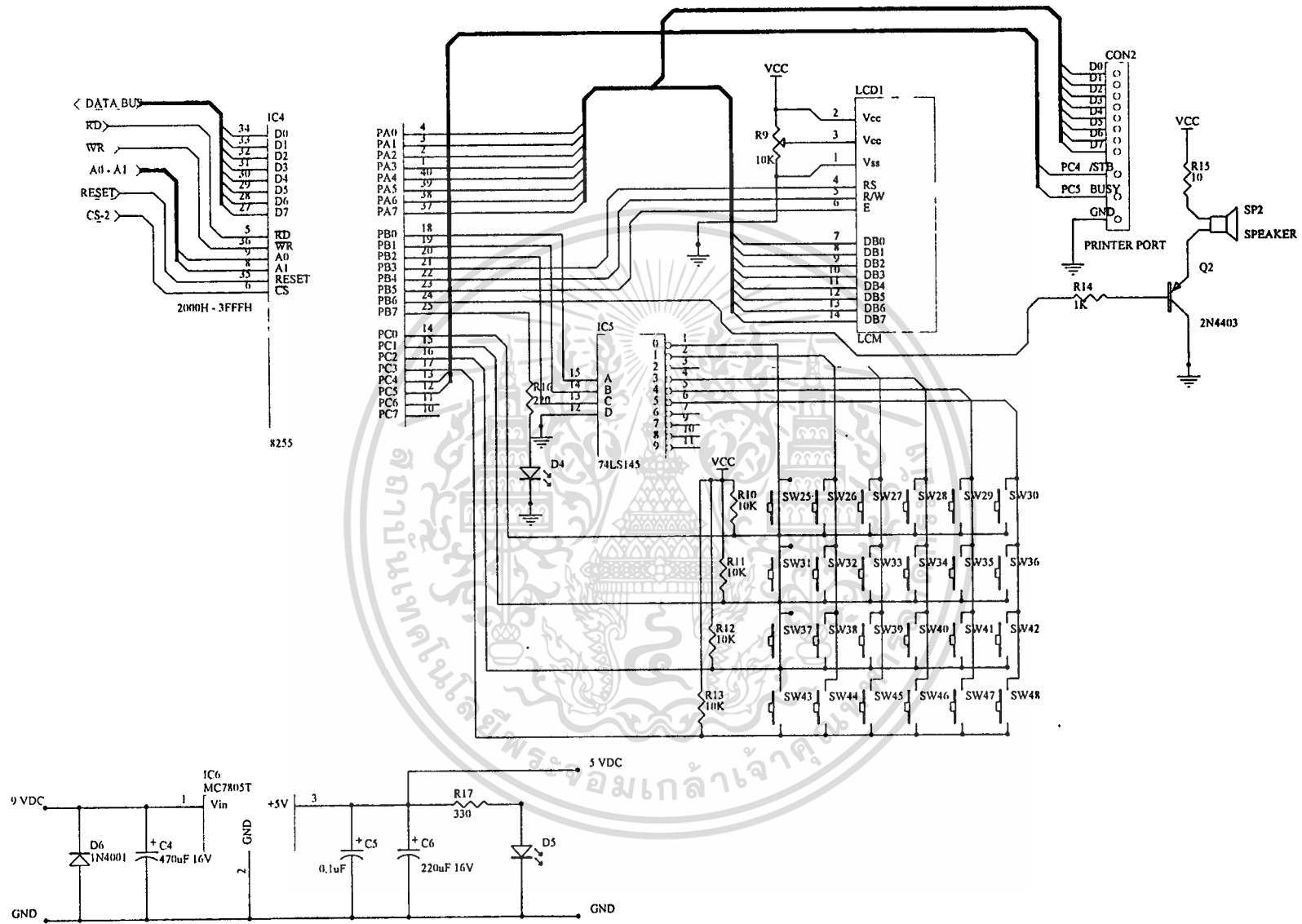
1. ขนาดของแผ่นวงจรพิมพ์ ควรที่จะทำพัฒนาให้มีขนาดเล็กลง และมีน้ำหนักเบาเพื่อความสะดวกในการใช้งานมากขึ้น

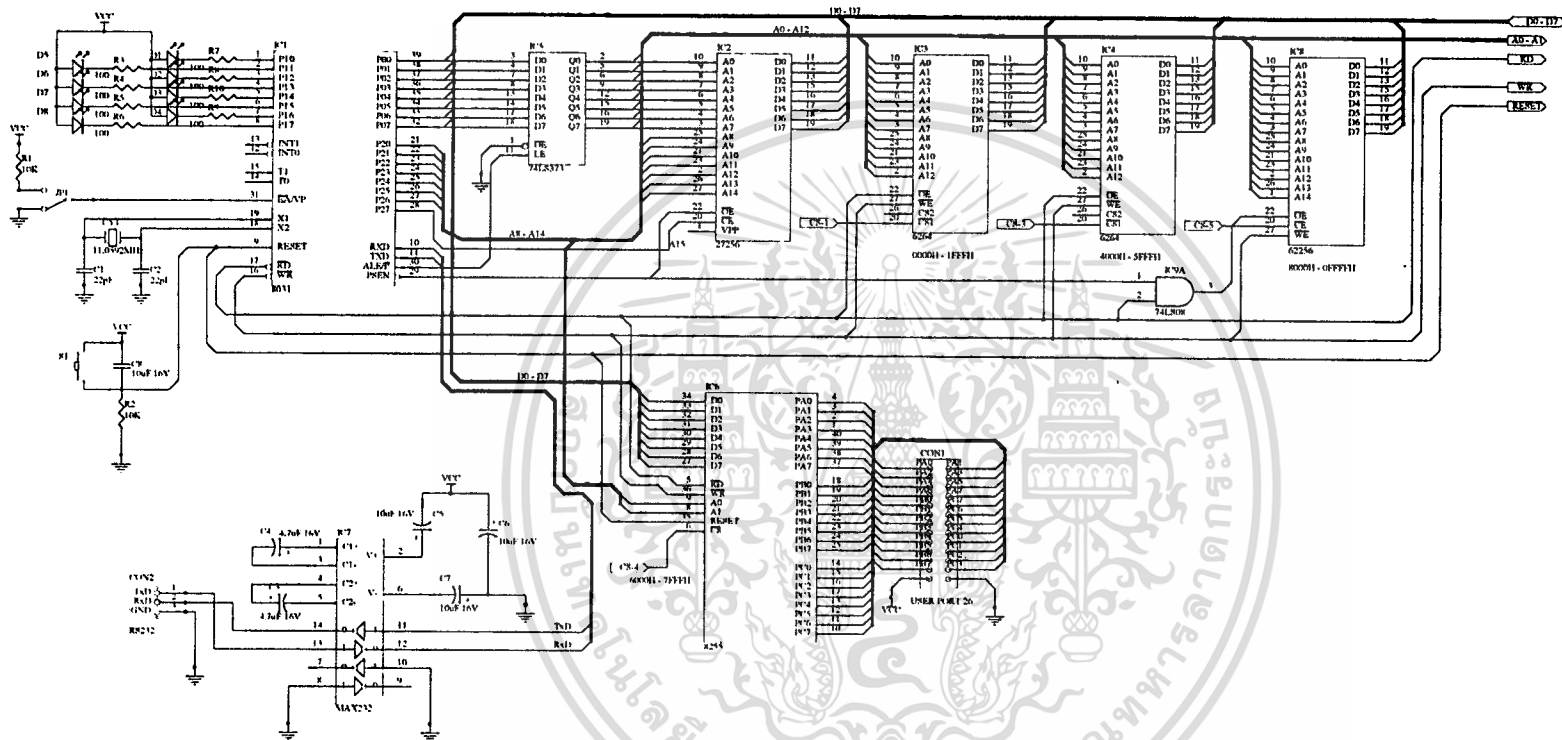
2. โปรแกรมมอนิเตอร์ที่เขียนขึ้น ยังคงไม่สมบูรณ์ ควรเขียนโปรแกรมให้มีความสมบูรณ์และมีประสิทธิภาพมากขึ้น เช่น ควรมีฟังก์ชันการใช้งานเพิ่มขึ้น และ เขียนโปรแกรมการประมวลผลแบบทีละคำสั่ง เป็นต้น

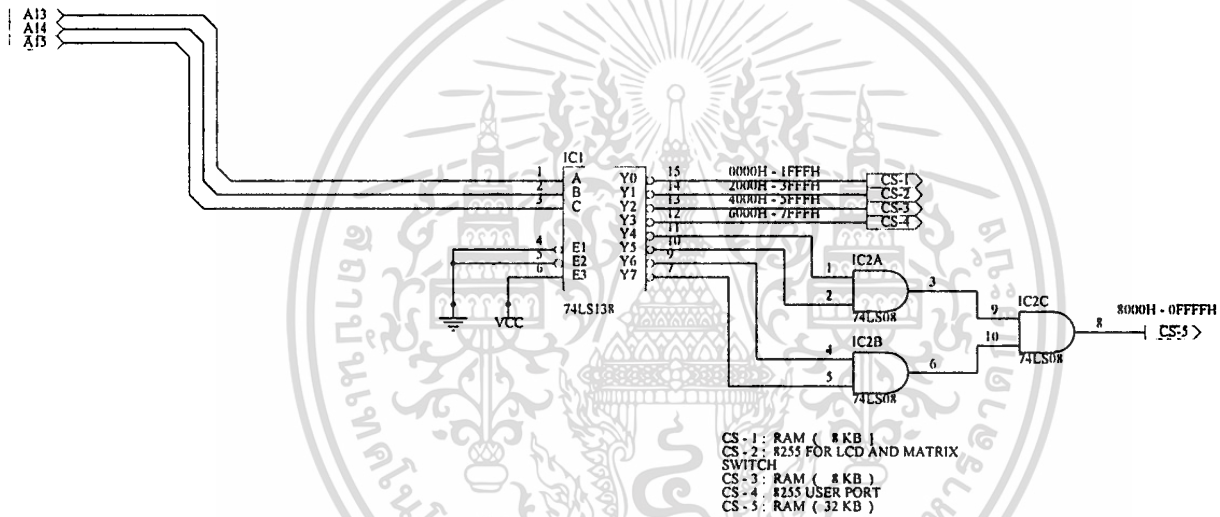


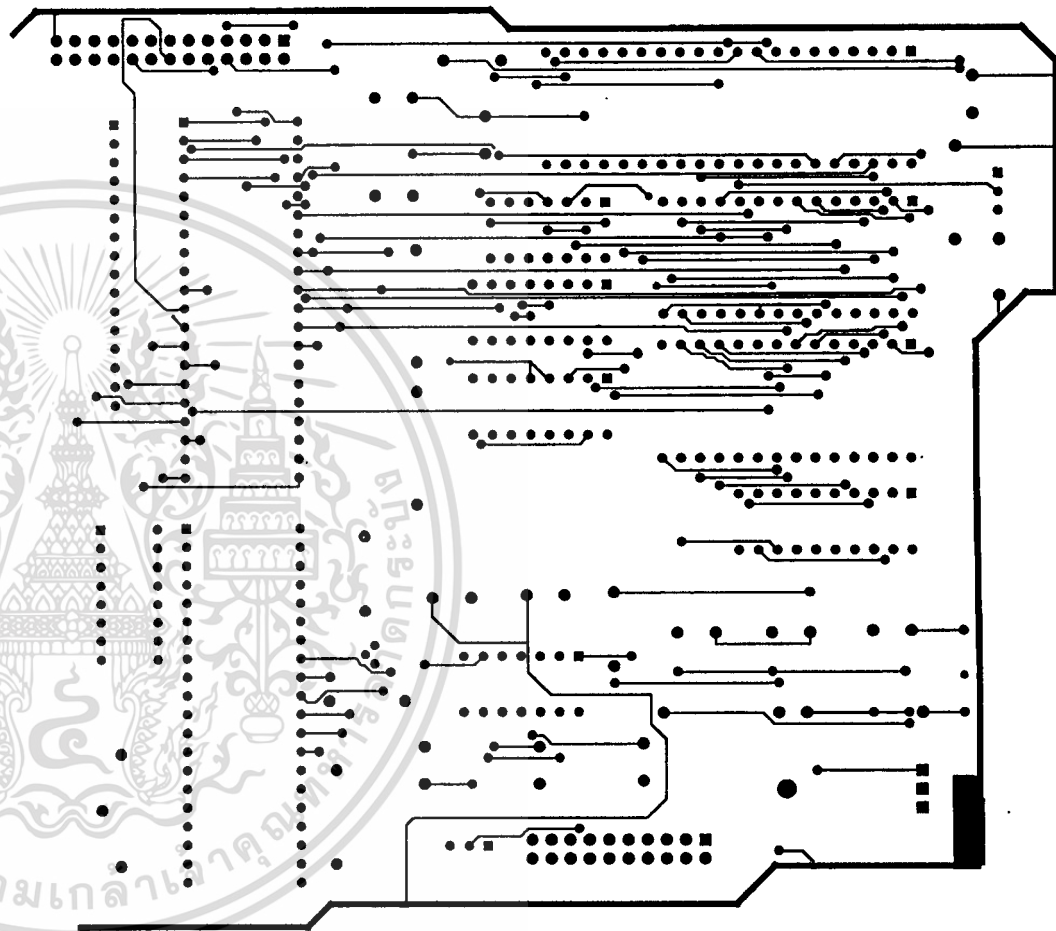
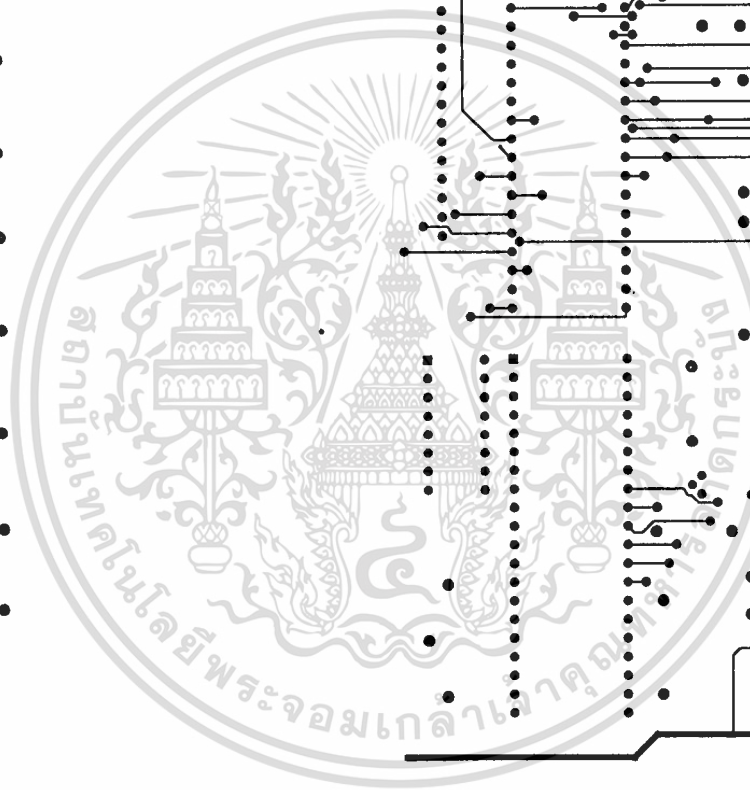
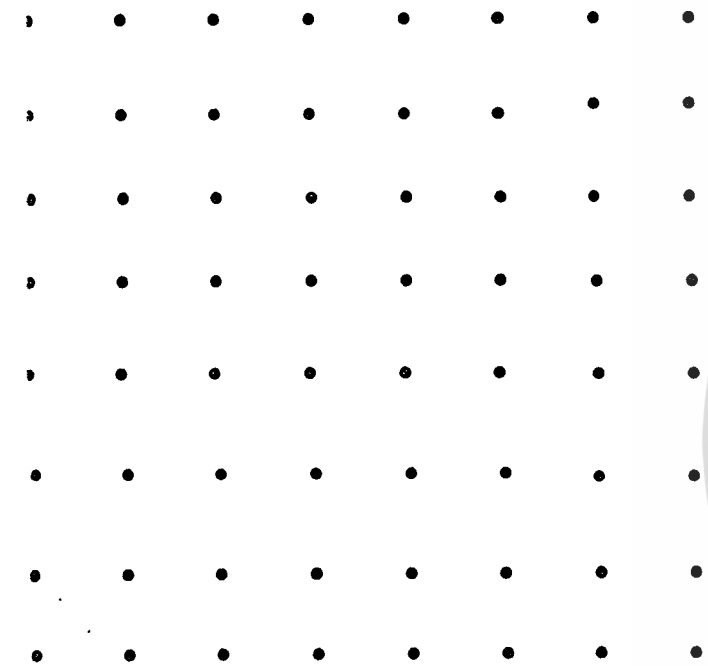
ภาคผนวก ก  
วงจรรและการวางอุปกรณ์

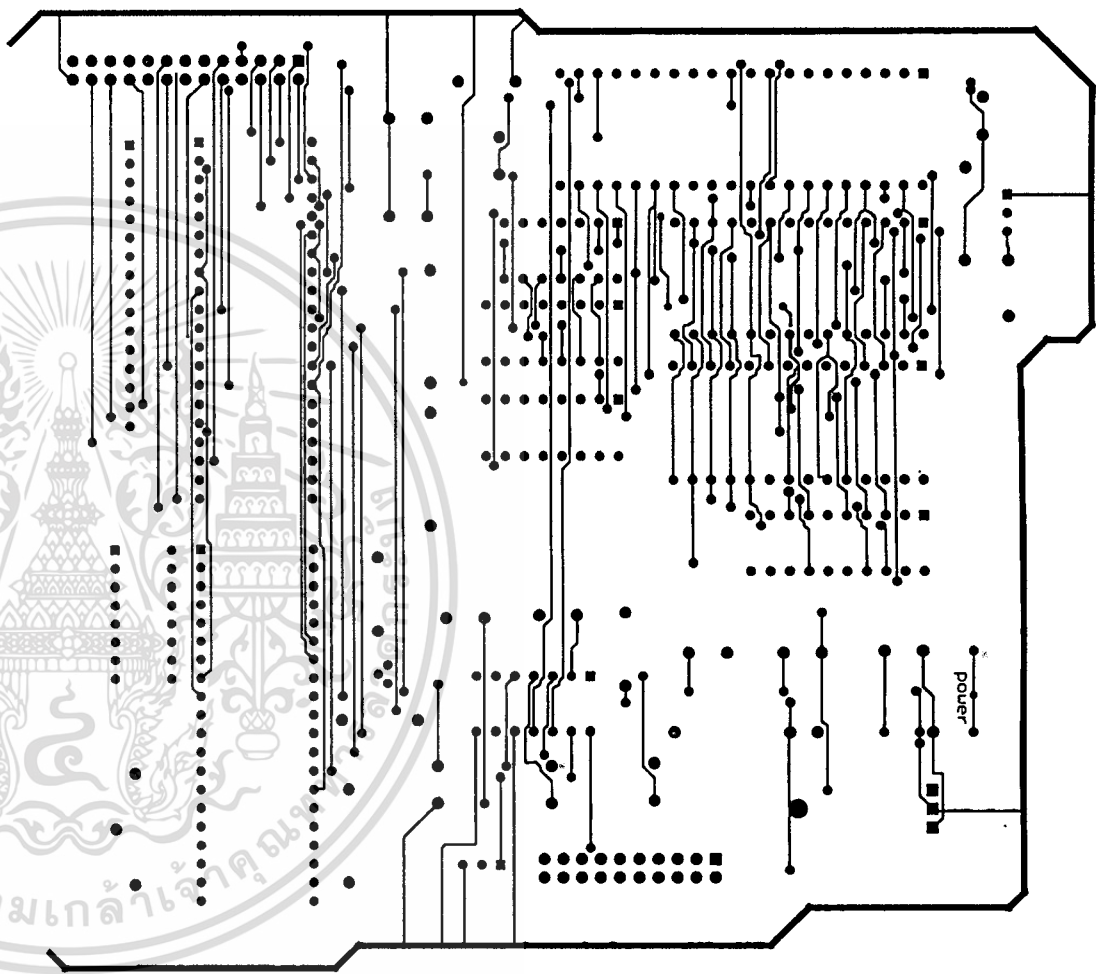
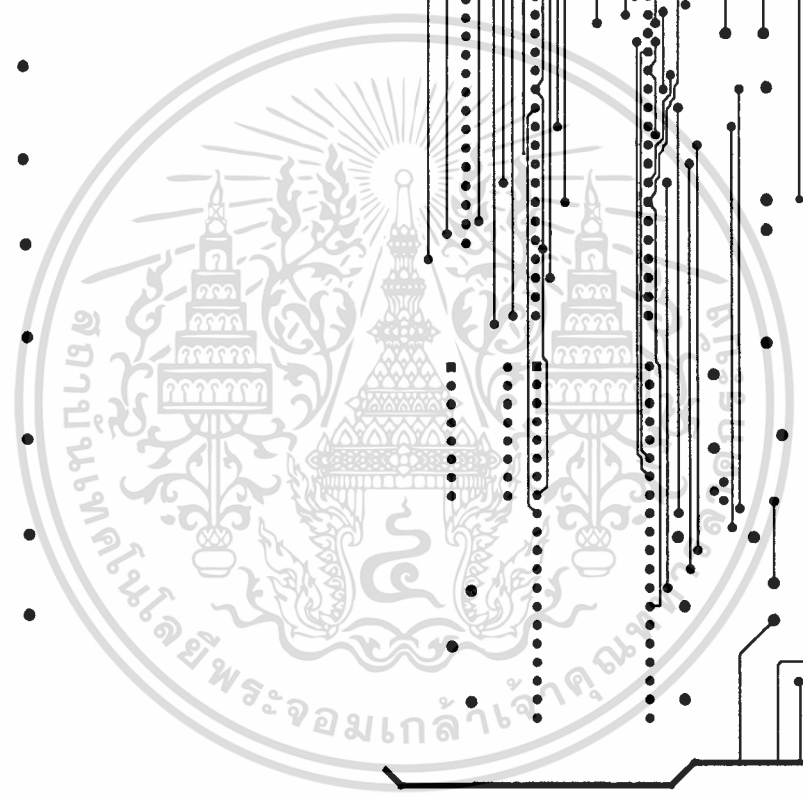
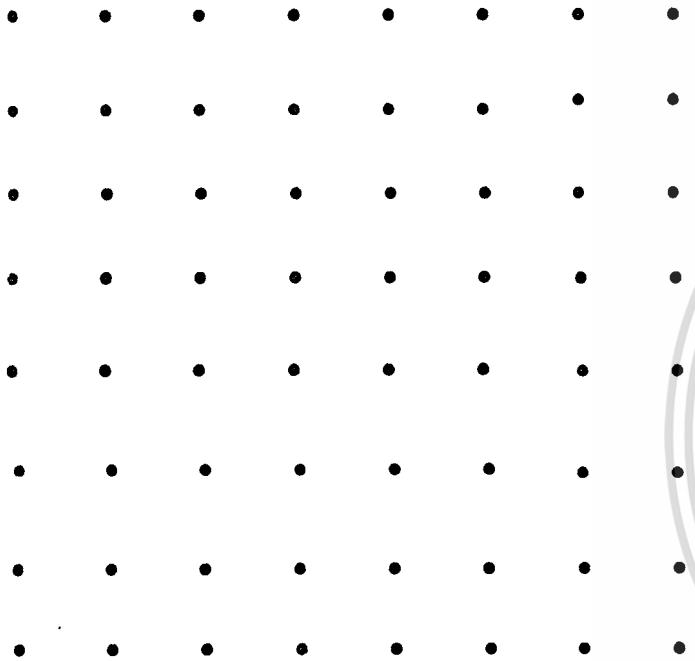
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

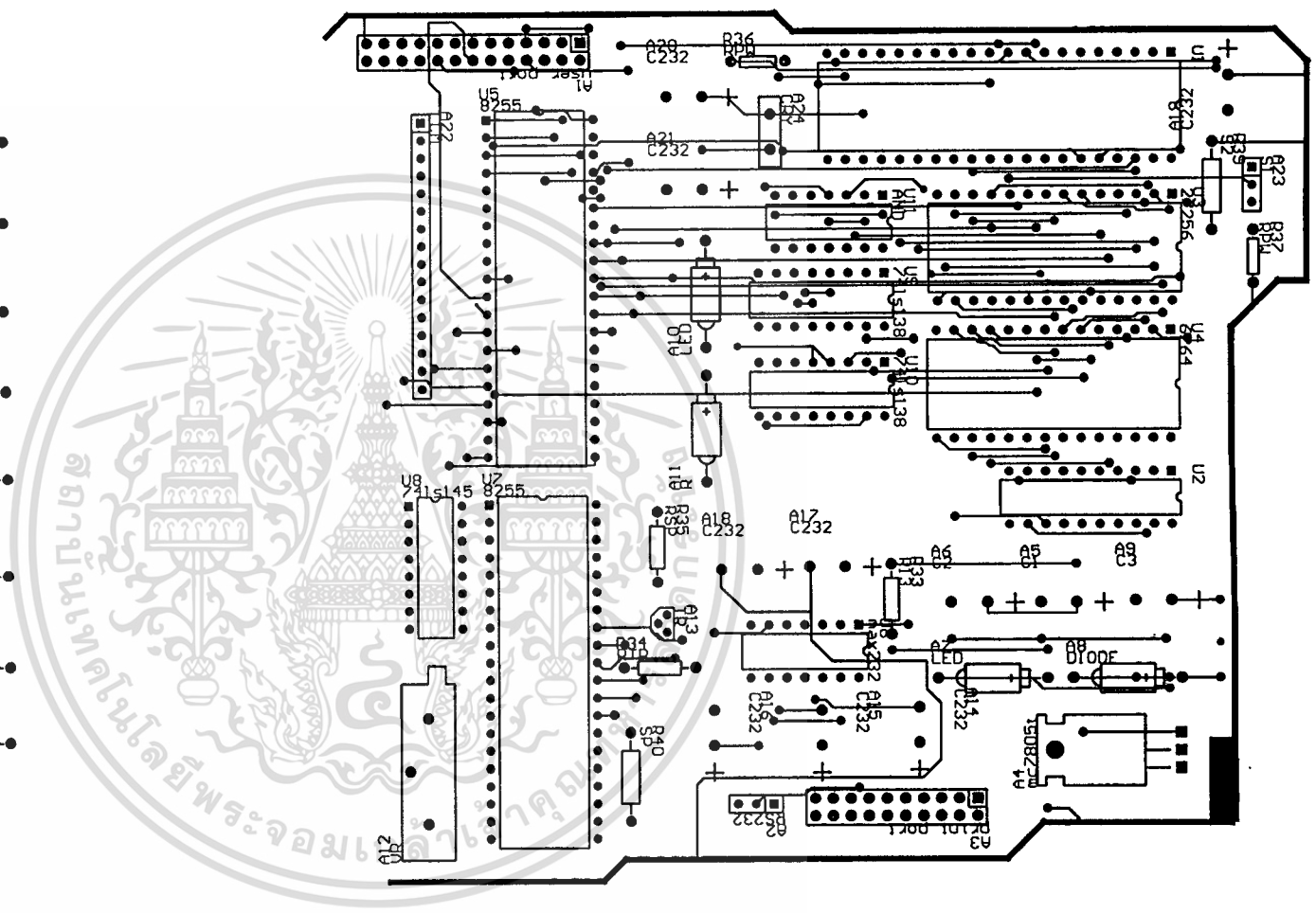
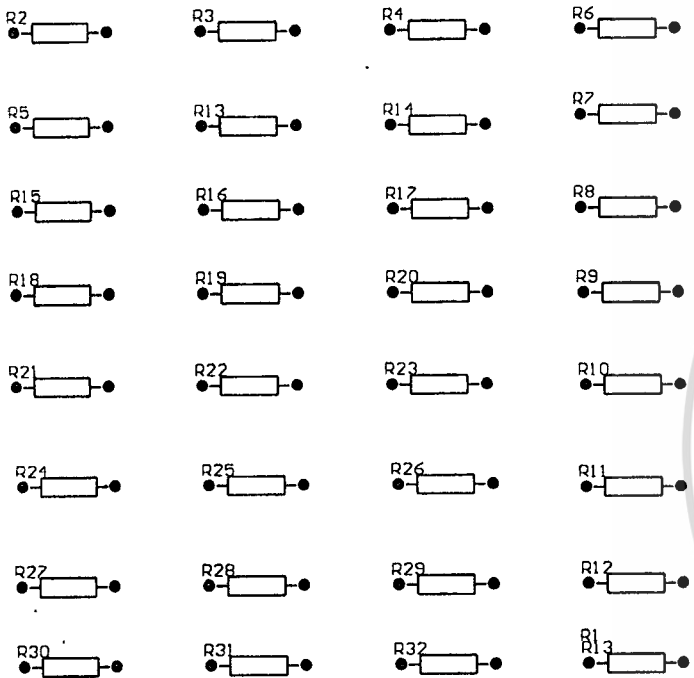












ภาคผนวก ข  
การใช้งาน 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การใช้งาน 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว

เมื่อป้อนไฟให้กับเครื่อง จะมีข้อความแสดงที่จอแสดงผลแบบผลึกเหลว ดังนี้ “SINGLE Board Version 1 ED. Engineer#19. Elec&com 2/2” และข้อความ “MICROCONTROLLER MCS-51” ที่จอแสดงผลจะแสดงถึงแค่ข้อความนี้ แล้วจะรอรับการกดคีย์คำสั่ง ซึ่งจะมีคีย์ Go, Spacebar, back, space, Function, Increment, Decrement, ESC และคีย์ Enter เมื่อต้องการที่จะป้อนคำสั่งการทำงานให้กับ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวทำงาน ซึ่งมีโหมดการทำงานอยู่ 2 โหมดด้วยกันคือ Editor Mode ซึ่งจะทำการป้อนคำสั่งลงในตัวไมโครคอนโทรลเลอร์ และโหมดการทำงานร่วมกับคอมพิวเตอร์

การใช้งานในโหมด Editor Mode สามารถปฏิบัติได้ดังนี้

1. เมื่อโปรแกรมอยู่ในส่วนของ monitor Mode โดยจะมีข้อความ “MICROCONTROLLER MCS-51” แล้วให้กดคีย์ “1” เพื่อเข้าสู่การป้อนคำสั่งซึ่งจะเป็นการป้อนคำสั่งที่เป็นภาษาเครื่องในเลขฐาน 16 ป้อนตำแหน่งแอดเดรสเริ่มต้นของโปรแกรมซึ่งจะเป็นตัวเลข 4 หลัก เมื่อป้อนคำสั่งครบเรียบร้อยแล้วทำการกดคีย์ “Enter”
2. ทำการป้อนข้อมูลลงในตำแหน่งข้อมูลของแอดเดรสที่เลือกซึ่งจะเป็นตัวเลข 2 หลัก แล้วทำการกดคีย์ “Enter” โปรแกรมจะทำการเพิ่มค่าแอดเดรสให้อัตโนมัติ
3. เมื่อทำการป้อนโปรแกรมเสร็จแล้วต้องการออกจากส่วน Editor Mode ให้ทำการกดคีย์ “ESC” เพื่อออกจาก Editor Mode ซึ่งจะมีข้อความแสดงขึ้นมาบนจอแสดงผลแบบผลึกเหลว “MICROCONTROLLER MCS-51” แล้วจะรอรับการกดคีย์คำสั่งต่อไป

เมื่อป้อนโปรแกรมเสร็จแล้ว ต้องการทำการที่จะประมวลผลโปรแกรมที่ทำการป้อน คำสั่งแล้วสามารถปฏิบัติได้ดังนี้

1. กดคีย์ “Go” เพื่อเข้าสู่การประมวลผลโปรแกรม
2. ป้อนตำแหน่งแอดเดรสที่จะทำการประมวลแล้วกดคีย์ “Enter”
3. โปรแกรมจะทำการประมวลผลของโปรแกรมผู้ใช้งานแล้วกลับเข้าสู่สภาวะปกติโดยจะมีข้อความ “MICROCONTROLLER MCS-51” แสดงอยู่บนจอแสดงผลแบบผลึกเหลว

การใช้งานโหมดร่วมกับคอมพิวเตอร์ สามารถปฏิบัติได้ดังนี้

1. เขียนโปรแกรมในโปรแกรม Editor ตัวใดก็ได้ แล้วทำการบันทึกเป็นแฟ้มข้อมูลที่มีนามสกุลของแอสเซมบลี ( File.asm )
2. ทำการ คอมไพล์โปรแกรมให้เป็นแฟ้มข้อมูลของ Intel Hex file ( File.hex )

3. ในขณะที่ไมโครคอนโทรลเลอร์ทำการโปรแกรมของโปรแกรมอยู่ในส่วนของ monitor

เอกสารนี้เป็นเอกสารที่วางไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mode โดยจะมีข้อความ “MICROCONTROLLER MCS-51” แล้วให้กดคีย์ “0” เพื่อเข้าสู่โหมดของการทำงานในโหมดร่วมกับคอมพิวเตอร์ที่แสดงผลแบบผลึกเหลวมีข้อความ “Download Hex File”

4. ให้ส่งแฟ้มข้อมูลแบบ Intel hex File มาทางพอร์ตอนุกรมโดยใช้คำสั่งตามรูปแบบดังนี้

C:Mode com1:96,n,8,1,p

C:copy File.hex com1

5. คอมพิวเตอร์จะส่งแฟ้มข้อมูลลงไปที่ไมโครคอนโทรลเลอร์เสร็จแล้ว จากนั้น จะกลับเข้าสู่โปรแกรม มอนิเตอร์ โหมด

6. ทำการประมวลผลโดยการกดคีย์ “Go” แล้วทำการป้อนค่าแอดเดรสที่มีข้อมูลจากการส่งจากคอมพิวเตอร์

หมายเหตุ ก่อนที่จะทำการส่งแฟ้มข้อมูลให้ต่อสายเชื่อมต่อจากคอมพิวเตอร์มาเข้าพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ จากตัวอย่างคำสั่งนั้นเป็นการเชื่อมต่อจากพอร์ต Com1 ของคอมพิวเตอร์

## ฟังก์ชันการทำงาน

ในการทำงานของโปรแกรมมอนิเตอร์จะมีโปรแกรมฟังก์ชัน สนับสนุนการทำงาน ให้สามารถใช้งานโปรแกรมส่วนอื่นได้สะดวกต่อการใช้ ซึ่งสามารถเลือกใช้ฟังก์ชันได้โดยเมื่อโปรแกรมอยู่ใน Monitor Mode โดยจะมีการข้อความแสดง “MICROCONTROLLER MCS-51” ที่จอแสดงผลแบบผลึกเหลว ผู้ใช้งานสามารถ กดคีย์ “FUNC” เพื่อเข้าสู่การเลือก โปรแกรมฟังก์ชัน แล้วทำการกดคีย์เลือกการใช้ฟังก์ชัน แต่ละฟังก์ชันมีการทำงาน ดังนี้

ฟังก์ชัน 0 แสดงผลที่จอแสดงผลแบบผลึกเหลว

ฟังก์ชัน 1 สร้างเสียงเพลงตามโน้ต

ฟังก์ชัน 3 Short & Open

ฟังก์ชัน 4 Clear Data memory

ฟังก์ชัน 5 Move Block

ฟังก์ชัน 6 Receive Serial Port

ฟังก์ชัน 7 Test Serial Port

ฟังก์ชัน 8 Generator pulse

ฟังก์ชัน F แสดงรายชื่อคณะผู้จัดทำโครงการทีละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ฟังก์ชัน 0 แสดงผลที่จอแสดงผลแบบพลิกเหลว

ฟังก์ชันนี้จะทำหน้าที่ นำข้อมูลจากหน่วยความจำที่เราได้ป้อนข้อมูลเข้าไปเก็บไว้แล้วโดยการป้อนเป็น รหัส แอสกี ซึ่งจะสามารถนำมาแสดงผลแบบพลิกเหลวได้ ครั้งละบรรทัด และครั้งละ 2 บรรทัด ซึ่งแต่ละบรรทัดสามารถป้อนได้ 16 ตัวอักษร ถ้าต้องการป้อนให้แสดงบรรทัดเดียว สามารถป้อนข้อมูลเข้าไปจำนวน 16 ตัวอักษรหรือจะไม่ถึงก็ได้ แต่ต้องจบด้วยการป้อนค่า 00H เพื่อเป็นการบอกให้โปรแกรมรู้ว่าจบข้อความนี้ ส่วนการแสดงผลข้อมูล 2 บรรทัดนั้นก็ทำการป้อนข้อมูลคล้ายกับการแสดงแบบ 1 บรรทัดแต่เมื่อจบบรรทัดแรกต้องมีรหัส 13H เพื่อบอกให้โปรแกรมรู้ว่าจบบรรทัดแรก ส่วนในบรรทัดที่สองทำการป้อนข้อมูลคล้ายกับการแสดงแบบหนึ่งบรรทัด

การใช้งานฟังก์ชัน 0

1. ป้อนไฟเข้าเครื่อง
2. กดคีย์ “1” เพื่อเข้าสู่ Editor Mode
3. ป้อนข้อมูลที่จะแสดงข้อความลงในแอดเดรสของหน่วยความจำข้อมูล
4. กลับเข้าสู่ Monitor Mode
5. กดคีย์ฟังก์ชันเพื่อเข้าสู่ฟังก์ชัน 0
6. ป้อนค่าแอดเดรสให้ข้อมูล แล้วกดคีย์ “Enter”
7. โปรแกรมจะนำเอาข้อมูลมาแสดงผลและจะแสดงค้างไว้ กดคีย์ใด ๆ เพื่อเข้าสู่ มอนิเตอร์ โหมด

## ฟังก์ชัน 1 สร้างเสียงเพลงตามโน้ต

เป็นฟังก์ชันที่เอาโน้ตของเสียงมาแสดงผลออกทางลำโพงซึ่งจะมีรหัสของโน้ต อยู่ 25 โน้ต ดังตารางที่ ค.1

ทำการป้อนข้อมูลของโน้ตเข้าสู่หน่วยความจำข้อมูล ซึ่งในการใส่ข้อมูลนั้นจะเป็นลักษณะข้อมูลขนาด 2 ไบต์ คือ ไบต์แรกเป็นรหัสของโน้ต ส่วนไบต์ที่สองจะเป็นการหน่วงเวลาในการค้างของเสียง และเมื่อป้อนข้อมูลหมดตามต้องการแล้ว ให้ทำการใส่ค่า 0FFH เพื่อบอกให้รู้ว่าหมดรหัสของโน้ตแล้ว

การใช้ฟังก์ชัน 1

1. เมื่อโปรแกรมอยู่ใน Monitor Mode ให้กดคีย์ฟังก์ชัน
2. กดคีย์ “1” เพื่อเข้าสู่ฟังก์ชันสร้างเสียงเพลงตามตัวโน้ต
3. ที่จอแสดงผลแบบพลิกเหลวจะแสดงข้อความ “Function Song” แล้วให้ทำการป้อนตำแหน่งแอดเดรสที่ต้องการป้อนข้อมูล แล้วทำการป้อนข้อมูลเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามนำเนื้อหาและข้อมูลข้างต้นไปเผยแพร่ต่อสาธารณะทุกครั้งที่มีการนำไปใช้

4. กดคีย์ “Enter” โปรแกรมจะกำเนิดเสียงออกจากลำโพงจนจบข้อมูล โปรแกรมจะกลับสู่การเลือกฟังก์ชันให้กดคีย์ “ESC” เพื่อเข้าสู่ Monitor Mode

ตารางที่ ข.1 รหัสของค่าตัวโน้ตในการใช้งานฟังก์ชัน 1

| NOTE | OCTAVE 1 | OCTAVE 2 | OCTAVE 3 | OCTAVE 4 |
|------|----------|----------|----------|----------|
| C    | ---      | 05H      | 11H      | 1DH      |
| C#   | ---      | 06H      | 12H      | 1EH      |
| D    | ---      | 07H      | 13H      | 1FH      |
| D#   | ---      | 08H      | 14H      | 20H      |
| E    | ---      | 09H      | 15H      | 21H      |
| F    | ---      | 0AH      | 16H      | 22H      |
| F#   | ---      | 0BH      | 17H      | 23H      |
| G    | 00H      | 0CH      | 18H      | 24H      |
| G#   | 01H      | 0DH      | 19H      | ---      |
| A    | 02H      | 0EH      | 1AH      | ---      |
| A#   | 03H      | 0FH      | 1BH      | ---      |
| B    | 04H      | 10H      | 1CH      | ---      |

## ฟังก์ชัน 2 Short & Open

เป็นฟังก์ชันสำหรับตรวจสอบอุปกรณ์(ต่อหรือขาด) โดยจะใช้พอร์ต TOOL I/O เป็นอินพุต ซึ่งถ้าขั้วบวกและลบของ TOOL I/O ต่อถึงกัน เครื่องจะส่งเสียงและแสดงเครื่องหมายจุดที่จอแสดงผลแบบผลึกเหลว จะแสดงผลให้ทราบและกระทำอย่างต่อเนื่อง จนกว่าขั้วบวกและลบนั้นจะแยกจากกัน ฟังก์ชันย่อยนี้ใช้เป็นเครื่องมือพื้นฐานในการทดลองต่าง ๆ ได้เป็นอย่างดี ไม่ว่าจะเป็นการวัดความต่อเนื่องบนแผ่นพิมพ์เขียวหรือการตรวจสอบสายไฟฟ้า ขาดภายในหรือไม่ มีวิธีการเรียกใช้ดังนี้

### การใช้ฟังก์ชัน 2

1. เมื่อ โปรแกรมอยู่ใน Monitor Mode กดคีย์ฟังก์ชัน เพื่อเข้าสู่เมนูการเลือกฟังก์ชัน
2. กดคีย์ 2 เพื่อเข้าสู่ฟังก์ชัน Short & Open
4. บนจอแสดงผลแบบผลึกเหลว จะแสดงข้อความ “Short & Open”
4. นำสายต่อจากจุดต่อ เพื่อเช็คอุปกรณ์หรือลายแผ่นวงจรพิมพ์ว่า ช็อตหรือไม่

5. เมื่อต้องการออกจากฟังก์ชันให้กดรีเซ็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ฟังก์ชัน 3 Clear Data Memory

เมื่อต้องการที่จะลบข้อมูลในหน่วยความจำให้มีค่าเป็น 00H โดยจะเคลียร์ค่าเฉพาะใน 30 กิโลไบต์ ของหน่วยความจำที่ 8000H–F7FFH (DATA + PROGRAM MEMORY) ซึ่งเป็นส่วนสำหรับให้ผู้พัฒนาโปรแกรม มีประโยชน์เพื่อการเริ่มต้นงานใหม่ ช่วยให้สามารถเขียนโปรแกรมได้อย่างเป็นระเบียบยิ่งขึ้น มีวิธีการเรียกใช้ดังนี้

#### การใช้ฟังก์ชัน 3

1. เมื่อโปรแกรมอยู่ใน Monitor Mode กดคีย์ฟังก์ชันเพื่อเข้าสู่เมนูการเลือกฟังก์ชัน
2. กดคีย์ “2” เพื่อการสู่ฟังก์ชัน Clear Data Memory
3. ที่จอแสดงผลแบบผลึกเหลวจะแสดงข้อความ “Clear Data Memory Press Enter”
4. กดคีย์ “Enter” เพื่อทำการเคลียร์ หน่วยความจำข้อมูล
5. โปรแกรมจะกลับเข้าสู่ Monitor Mode

### ฟังก์ชัน 4 Move Block

เมื่อมีความต้องการที่จะย้ายข้อมูลจากช่วงของแอดเดรสที่ต้องการไปยังแอดเดรสอื่น ๆ โดยจะสามารถกระทำกับส่วนหน่วยความจำข้อมูล เท่านั้น การย้ายข้อมูลนี้สามารถกระทำบนแอดเดรสที่ซ้อนทับกันได้ เช่น ต้องการย้ายข้อมูลจาก 8100H เนื่องจากฟังก์ชันย่อยนี้จะมีระบบตรวจสอบเพื่อเลือกลักษณะการย้ายให้เป็นแบบ จากแอดเดรสต่ำไปยังแอดเดรสสูง หรือจากแอดเดรสสูงไปยังแอดเดรสต่ำก็ได้อีกประการหนึ่ง การย้ายข้อมูลนี้ จะไม่ทำให้ข้อมูลในต้นฉบับมีการเปลี่ยนแปลงแต่อย่างใด (ถ้าไม่มีแอดเดรสที่ซ้อนกัน) มีวิธีการเรียกใช้งานดังนี้

#### การใช้ฟังก์ชัน 4

1. เมื่อโปรแกรมอยู่ในโหมดเตอร์ โหมด กดคีย์ฟังก์ชัน
2. กดคีย์ “4” เพื่อเข้าสู่ฟังก์ชัน Move Block
3. ป้อนแอดเดรสเริ่มต้นของโปรแกรมที่ต้องการย้ายแล้วกดคีย์ “Enter”
4. ป้อนแอดเดรสสิ้นสุดของโปรแกรมที่ต้องการย้ายแล้วกดคีย์ “Enter”
5. แอดเดรสปลายทางของโปรแกรมที่ต้องการย้ายแล้วกดคีย์ “Enter”
6. โปรแกรมจะทำการย้ายข้อมูลในบล็อกไปเก็บไว้ที่แอดเดรสปลายทาง
7. โปรแกรมจะกลับเข้าสู่ Monitor Mode

### ฟังก์ชัน 5 Receive Serial Port

เป็นฟังก์ชันที่ใช้สำหรับรับข้อมูลจากคีย์บอร์ดของคอมพิวเตอร์ ซึ่งค่าที่รับเข้ามานั้นจะเป็นรหัสแอสกี ซึ่งค่าที่รับเข้ามาจะเก็บไว้ใน แอดเดรส 8000H ซึ่งเป็นแอดเดรสเริ่มต้น และจะทำการไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มแอดเดรสเองอัตโนมัติ ซึ่งฟังก์ชันนี้จะมีโปรแกรมใช้งานร่วมจากโปรแกรมภาษาซี ซึ่งมาจากไฟล์ Send1.C

การใช้ฟังก์ชัน 5

1. เมื่อโปรแกรมอยู่ใน Monitor Mode ให้กดคีย์ฟังก์ชัน
2. กดคีย์ “5” เพื่อเข้าสู่ฟังก์ชัน Receive Serial Port
3. ทำการประมวลผลโปรแกรมใน Send1.EXE บนคอมพิวเตอร์
4. กดคีย์บอร์ดของคอมพิวเตอร์ รหัสแอสกีของคีย์ที่ทำการกดนั้นจะไปแสดงที่พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
5. กดคีย์ “Enter” เพื่อทำการหยุดรับข้อมูล

### ฟังก์ชัน 6 Test Serial Port

เป็นฟังก์ชันใช้สำหรับทดสอบพอร์ตการรับส่งข้อมูลอนุกรม ว่าสามารถที่จะใช้งานได้หรือไม่ โดยฟังก์ชันนี้จะใช้งานร่วมกับโปรแกรมภาษาซี ซึ่งมาจากไฟล์ Serial.C ในการทำงานร่วมกัน โปรแกรมภาษาซีจะทำการส่งข้อมูลมาทางพอร์ตการรับส่งข้อมูลอนุกรมมาที่ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว รับข้อมูลแล้วนำไปแสดงที่พอร์ตหนึ่งของไมโครคอนโทรลเลอร์

การใช้งานฟังก์ชัน 6

1. เมื่อโปรแกรมอยู่ใน Monitor Mode ให้กดคีย์ฟังก์ชัน
2. กดคีย์ “6” เพื่อเข้าสู่ฟังก์ชัน Test Serial Port โดยจะมีข้อความแสดงบนจอแสดงผลแบบผลึกเหลว ว่า “Test Serial Port”
3. ทำการประมวลผลโปรแกรมภาษาซี เพื่อใช้งานร่วมกับฟังก์ชันชื่อไฟล์ Send.C
4. โปรแกรมภาษาซีจะทำการส่งข้อมูลมาทางพอร์ตการรับส่งข้อมูลอนุกรม
5. ไมโครคอนโทรลเลอร์ จะเอาข้อมูลที่ได้นำมาแสดงที่พอร์ต หนึ่ง

### ฟังก์ชัน 7 Generator

เป็นฟังก์ชันที่ใช้สำหรับการผลิตพัลส์ ซึ่งเป็นแบบสแควร์เวฟออกมาที่พอร์ต P1.7 ของไมโครคอนโทรลเลอร์ จะมีการป้อนค่าให้กับรีจิสเตอร์ R6 และ R7 เป็นตัวเก็บค่ากำหนดการเกิดพัลส์ ซึ่งขึ้นอยู่กับว่าผู้ใช้งานต้องการพัลส์ขนาดเท่าใด โดยมีวิธีการใช้งาน ดังนี้

การใช้งานฟังก์ชัน 7

1. เมื่อโปรแกรมอยู่ในมอนิเตอร์ โหมด กดคีย์ฟังก์ชัน
2. กดคีย์ “7” เพื่อเข้าสู่โปรแกรมฟังก์ชัน Generator
3. บนจอแสดงผลแบบผลึกเหลว จะแสดงข้อความ “Generator Pulse Square Wave”
4. ป้อนค่าให้กับรีจิสเตอร์ R6 และ R7

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้และจะออกในรูปแบบของแผ่นไมโครคอมพิวเตอร์ใช้เพื่อประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ที่มีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. กดคีย์ฟังก์ชันเพื่อให้โปรแกรมกำเนิดพัลส์

6. เมื่อต้องการออกจากฟังก์ชันกด รีเซ็ต

## ฟังก์ชัน F แสดงรายชื่อคณะผู้จัดทำโครงการงานทีละคน

เป็นฟังก์ชันแสดงรายชื่อคณะผู้จัดทำทีละคน

การใช้ฟังก์ชัน F

1. เมื่อโปรแกรมอยู่ใน Monitor Mode ให้กดคีย์ฟังก์ชัน
2. กดคีย์ “F” เพื่อเข้าดูรายชื่อคณะผู้จัดทำ
3. กดคีย์ใด ๆ เมื่อต้องการออกจากฟังก์ชัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค  
ใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 1

### เรื่อง การติดต่อกับพอร์ตภายในของ 8051

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถอธิบายการทำงาน โปรแกรมการติดต่อกับพอร์ตภายในของ 8051 ได้

#### ทฤษฎี

พอร์ตภายในของ 8051 ไมโครคอนโทรลเลอร์จะประกอบไปด้วย พอร์ต 0, พอร์ต 1, พอร์ต 2 และ พอร์ต 3 เป็นกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ ซึ่งสามารถกระทำการแบบบิตได้ และเป็นตัวรีจิสเตอร์ที่แลตช์ของพอร์ต 0, พอร์ต 1, พอร์ต 2 และ พอร์ต 3 โดยที่ พอร์ต 0 เป็นบัสข้อมูลและบัสตำแหน่งไบต์ต่ำ, พอร์ต 1 จะเป็นพอร์ตที่ใช้งานทั่วไป, พอร์ตที่ 2 เป็นบัสตำแหน่งไบต์สูง และ พอร์ต 3 เป็นพอร์ตที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |

#### ลำดับขั้นตอนการทดลอง

1. นำคำสั่งจากโปรแกรมการทดลองที่ 1 ไปทำการแปลงเป็นคำสั่งของภาษาเครื่อง โดยการเปิดคู่มือการแปลรหัสคำสั่ง
2. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว โดยเริ่มจากตำแหน่ง 8000H
3. เมื่อป้อนโปรแกรมเสร็จแล้ว สังเกตการประมวลผล โดยกดคีย์ "GO" และตามด้วย ตำแหน่งข้อมูล
4. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมทดลองที่ 1

|        |      |         |
|--------|------|---------|
|        | ORG  | 8000H   |
| START: | MOV  | A,#0FFH |
|        | CPL  | A       |
|        | MOV  | P1,A    |
|        | SJMP | \$      |
|        | END  |         |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ผลการทดลอง

.....

.....

.....

.....

### สรุปผลการทดลอง

.....

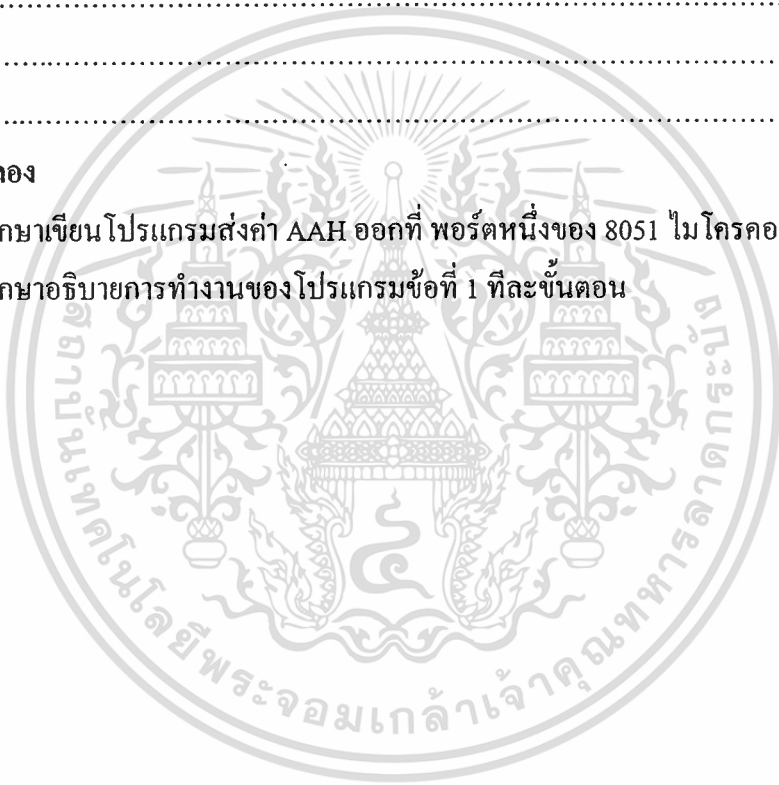
.....

.....

.....

### คำถามท้ายการทดลอง

1. ให้นักศึกษาเขียนโปรแกรมส่งค่า AAH ออกที่ พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์
2. ให้นักศึกษาอธิบายการทำงานของโปรแกรมข้อที่ 1 ทีละขั้นตอน



## ใบงานที่ 2

### เรื่อง โปรแกรมไฟวิ่งจากขวาไปซ้ายที่พอร์ต P1

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งาน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถอธิบายการทำงาน โปรแกรมควบคุมการทำงานของพอร์ตภายใน 8051 ได้

#### ทฤษฎี

โครงสร้างพอร์ตและการทำงานใน 8051 ไมโครคอนโทรลเลอร์มีพอร์ต 4 พอร์ต และทั้ง 4 พอร์ตเป็นแบบสองทิศทาง แต่ละพอร์ตจะประกอบด้วยเลตช์เป็น P0 ถึง P3 ของ SFR จะมีตัวขับเอาต์พุตและบัฟเฟอร์ (Buffer) อินพุตตัวขับเอาต์พุตของพอร์ต 0 และ พอร์ต 2 และบัฟเฟอร์อินพุตของพอร์ต 0 จะใช้งานสำหรับการเข้าถึงหน่วยความจำภายนอก ในการใช้งานลักษณะนี้ เอาต์พุตพอร์ต 0 จะทำหน้าที่เป็นตัวกำหนดไบต์ต่ำของตำแหน่งหน่วยความจำภายนอก

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |

#### ลำดับขั้นตอนการทดลอง

1. นำคำสั่งจากโปรแกรมการทดลองที่ 2 ไปแปลงเป็นรหัสคำสั่งของภาษาเครื่อง
2. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว โดยเริ่มจากตำแหน่ง 8500H
3. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว สั่งทำการประมวลผล โดยกดคีย์ “GO” และตามด้วยตำแหน่งข้อมูล
4. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 2

|        |       |        |
|--------|-------|--------|
|        | ORG   | 8500H  |
| START: | MOV   | A,#01H |
| LOOP:  | CPL   | A      |
|        | MOV   | P1,A   |
|        | CPL   | A      |
|        | ACALL | DELAY  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                RL          A
                                SJMP       LOOP
DELAY:  MOV          R0,0FFH
DELAY1: MOV          R1,0FFH
                                DJNZ       R1,$
                                DJNZ       R0,DELAY1
                                RET
                                END

```

ผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

คำถามท้ายการทดลอง

- 1. ให้นักศึกษาอธิบายการทำงานของ โปรแกรมการทดลองที่ 2 ทีละขั้นตอน

### ใบงานที่ 3

#### เรื่อง โปรแกรมไฟวิ่งจากซ้ายไปขวาที่พอร์ต P1

##### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถเขียนโปรแกรมใช้งานร่วมกับพอร์ตภายในของ 8051 ได้

##### ทฤษฎี

โครงสร้างพอร์ตและการทำงานใน 8051 ไมโครคอนโทรลเลอร์มีพอร์ต 4 พอร์ต และทั้ง 4 พอร์ตเป็นแบบสองทิศทาง แต่ละพอร์ตจะประกอบด้วยแลตช์เป็น P0 ถึง P3 ของ SFR จะมีตัวขับเอาต์พุตและบัฟเฟอร์ (Buffer) อินพุตตัวขับเอาต์พุตของพอร์ต 0 และ พอร์ต 2 และบัฟเฟอร์อินพุตของพอร์ต 0 จะใช้งานสำหรับการเข้าถึงหน่วยความจำภายนอก ในการใช้งานลักษณะนี้ เอาต์พุตพอร์ต 0 จะทำหน้าที่เป็นตัวกำหนดไบต์ต่ำของตำแหน่งหน่วยความจำภายนอก

##### เครื่องมือและอุปกรณ์

1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว 1 เครื่อง
2. คู่มือการแปลรหัสคำสั่ง 1 เล่ม

##### ลำดับขั้นตอนการทดลอง

1. นำคำสั่งจากโปรแกรมการทดลองที่ 3 ไปแปลงเป็น รหัสคำสั่งของภาษาเครื่อง
2. ป้อนรหัสคำสั่งลงในไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว โดยเริ่มจากตำแหน่ง 9000H
3. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว สั่งทำการประมวลผล โดยกดคีย์ “GO” และตามด้วยตำแหน่งข้อมูล
4. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

##### โปรแกรมการทดลองที่ 3

|        |       |        |
|--------|-------|--------|
|        | ORG   | 9000H  |
| START: | MOV   | A,#80H |
| LOOP:  | CPL   | A      |
|        | MOV   | P1,A   |
|        | CPL   | A      |
|        | ACALL | DELAY  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิที่จะนำเอกสารนี้ไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|      |      |
|------|------|
| RR   | A    |
| SJMP | LOOP |

```
DELAY:      MOV      R0,0FFH
DELAY1:     MOV      R1,0FFH
            DJNZ     R1,$
            DJNZ     R0,DELAY1
            RET
            END
```

ผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

คำถามท้ายการทดลอง

- 1. ให้นักศึกษาเขียนโปรแกรมไฟวิ่งจาก ทางด้านข้างวิ่งเข้าหาตรงกลาง ของพอร์ต หนึ่ง



## ใบงานที่ 4

### เรื่อง การติดต่อกับอุปกรณ์ภายนอก ทางพอร์ตผู้ใช้งาน

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งาน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถอธิบายการทำงานของพอร์ตผู้ใช้งานได้

#### ทฤษฎี

การรับส่งของพอร์ตนั้น บางครั้งการส่งอาจจะไม่ได้ครั้งละ 8 บิตเสมอ ดังนั้นถ้ามีการส่งค่าข้อมูลที่ไม่ใช่ขนาด 8 บิต สามารถที่จะใช้โปรแกรมช่วยในการตรวจสอบการรับส่งข้อมูลได้เป็นค่าขนาดต่าง ๆ แล้วแต่ความต้องการของผู้ใช้งาน ซึ่งผู้ใช้สามารถกำหนดรูปแบบของการรับส่งข้อมูลได้ เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 | ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 4 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 4

โปรแกรมให้ 8255 ซึ่งอยู่ที่พอร์ต 6000H-6003H ให้ทุกพอร์ตทำงานเป็นเอาต์พุตทั้งหมดและให้ส่งข้อมูล

- 0AAH ออกไปยังพอร์ต A
- 23H ออกไปยังพอร์ต B
- 2BH ออกไปยังพอร์ต C

|        |     |       |
|--------|-----|-------|
| PORT_A | EQU | 6000H |
| PORT_B | EQU | 6001H |
| PORT_C | EQU | 6002H |
| CTRL   | EQU | 6003H |

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับงานเพื่อการศึกษานี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                ORG            8000H
START:          ACALL         DELAY
                MOV           DPTR,#CTRL
                MOV           A,#80H
                MOVX          @DPTR,A
                MOV           DPTR,#PORT_A
                MOV           A,#0AAH
                MOVX          @DPTR,A
                MOV           DPTR,#PORT_B
                MOV           A,#23H
                MOVX          @DPTR,A
                MOV           DPTR,#PORT_C
                MOV           A,#2BH
                MOVX          @DPTR,A
                SJMP          $
DELAY:          MOV           R0,#0FFH
DELAY1:         MOV           R1,#0FFH
                DJNZ          R1,$
                DJNZ          R0,DELAY1
                RET
                END

```

#### ผลการทดลอง

.....

.....

.....

#### สรุปผลการทดลอง

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### คำถามท้ายการทดลอง

1. ถ้าต้องการเอาข้อมูล FFH ออกมาที่พอร์ต A, พอร์ต B และ พอร์ต C จะต้องทำการเปลี่ยนคำสั่งในตำแหน่งใดบ้าง
2. ให้นักศึกษาอธิบายการควบคุมการทำงานของพอร์ตผู้ใช้งาน มาพอสังเขป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 5

### เรื่อง โปรแกรมเซตบิต 7 ของ 8255

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถเซตบิตของพอร์ตผู้ใช้งานได้

#### ทฤษฎี

นอกจากผู้ใช้งานจะใช้ พอร์ต A, พอร์ต B และ พอร์ต C ในการโปรแกรมให้เป็นอินพุตและเอาต์พุต แล้วยังสามารถที่จะโปรแกรมพอร์ต C บิต PC0-PC7 ให้มีสถานะ ลอจิกเป็น “0” หรือ “1” เพื่อใช้เป็นสัญญาณสโตรบ (Strobe) โดย โหมด 1 จะใช้พอร์ต A และ พอร์ต B ในการส่งข้อมูล โดยส่วนใช้พอร์ต C เป็นพอร์ตตรวจสอบความพร้อม ในการทำงาน

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 | ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 5 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 5

กำหนดให้พอร์ตของ 8255 อยู่ที่ตำแหน่งแอดเดรส 6000H-6003H

|        |     |       |
|--------|-----|-------|
| PORT_A | EQU | 6000H |
| PORT_B | EQU | 6001H |
| PORT_C | EQU | 6002H |
| CTRL   | EQU | 6003H |
|        | ORG | 8000H |

START: ACALL DELAY

MOV DPTR,#CTRL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ใช้งานเพื่อการเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาหรือข้อมูลใดๆ ซึ่งรวมถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      A,#80H
MOVX     @DPTR,A
MOV      A,#01H
MOV      DPTR,#PORT_A
MOVX     @DPTR,A
SJMP     $
DELAY:   MOV      R0,#0FFH
DELAY1:  MOV      R1,#0FFH
          DJNZ     R1,$
          DJNZ     R0,DELAY1
          RET
          END

```

ผลการทดลอง

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. ถ้าต้องการเคลียร์บิตที่ 7 ของ 8255 จะต้องแก้ไขตำแหน่งใดบ้าง
2. ถ้าต้องการเซตบิตอื่นนอกจากบิตที่ 7 ของ 8255 สามารถทำได้หรือไม่ เพราะเหตุใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 6

### เรื่อง โปรแกรมไฟวิ่งรูปแบบที่ 1 ที่พอร์ต C ของ 8255

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานพอร์ตของ 8255 ได้
2. เพื่อให้นักศึกษาสามารถนำพอร์ต 8255 ไปประยุกต์ใช้งานได้

#### ทฤษฎี

นอกจากผู้ใช้งานจะใช้ พอร์ต A, พอร์ต B และ พอร์ต C ในการโปรแกรมให้เป็นอินพุตและเอาต์พุต แล้วยังสามารถที่จะโปรแกรมพอร์ต C บิต PC0-PC7 ให้มีสถานะ ลอจิกเป็น “0” หรือ “1” เพื่อใช้เป็นสัญญาณสโตป (Strobe) โดย โหมด 1 จะใช้พอร์ต A และ พอร์ต B ในการส่งข้อมูล โดยส่วนใช้พอร์ต C เป็นพอร์ตตรวจสอบความพร้อม ในการทำงาน

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 | ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 6 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 6

กำหนดให้พอร์ตของ 8255 อยู่ที่แอดเดรส 6000H-6003H

|        |     |       |
|--------|-----|-------|
| PORT_A | EQU | 6000H |
| PORT_B | EQU | 6001H |
| PORT_C | EQU | 6002H |
| CTRL   | EQU | 6003H |
|        | ORG | 8000H |

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับใช้ประกอบการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้นำเนื้อหาในเอกสารนี้ไปเผยแพร่หรือแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      A,#80H
MOVX     @DPTR,A
LOOP:    MOV      R2,#00H
LOOP1:   MOV      A,R2
         RL       A
         ANL      A,#0FH
         SETB     ACC.0
         MOVX     @DPTR,A
         ACALL    DELAY
         CLR      ACC.0
         MOVX     @DPTR,A
         INC      R2
         CJNE     R2,#08H,LOOP1
         SJMP     LOOP
DELAY:   MOV      R0,#0FFH
DELAY1:  MOV      R1,#0FFH
         DJNZ     R1,$
         DJNZ     R0,DELAY1
         RET
         END

```

#### ผลการทดลอง

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุปผลการทดลอง

.....

.....

.....

.....

## คำถามท้ายการทดลอง

1. ในบรรทัดที่ 8 ถ้าแก้จาก MOV A,#80H ไปเป็น MOV A,#00H จะเกิดอะไรขึ้น
2. จงเขียนโปรแกรมไฟวิ่งจากซ้ายไปขวาที่พอร์ต A ของ 8255



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 7

### เรื่อง โปรแกรมไฟวิ่งรูปแบบที่ 2 ที่พอร์ต C ของ 8255

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานพอร์ตของ 8255 ได้
2. เพื่อให้นักศึกษาสามารถนำพอร์ต 8255 ไปประยุกต์ใช้งานได้

#### ทฤษฎี

นอกจากผู้ใช้งานจะใช้ พอร์ต A, พอร์ต B และ พอร์ต C ในการโปรแกรมให้เป็นอินพุตและเอาต์พุต แล้วยังสามารถที่จะโปรแกรมพอร์ต C บิต PC0-PC7 ให้มีสถานะ ลอจิกเป็น “0” หรือ “1” เพื่อใช้เป็นสัญญาณสตrobe (Strobe) โดย โหมด 1 จะใช้พอร์ต A และ พอร์ต B ในการส่งข้อมูล โดยส่วนใช้พอร์ต C เป็นพอร์ตตรวจสอบความพร้อมในการทำงาน

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 | ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 7 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 7

กำหนดให้พอร์ตของ 8255 อยู่ที่แอดเดรส 6000H-6003H

|        |     |       |
|--------|-----|-------|
| PORTA  | EQU | 6000H |
| PORTB  | EQU | 6001H |
| PORTC  | EQU | 6002H |
| CTRL_P | EQU | 6003H |
| FLAG   | EQU | 7FH   |

ORG 8000H

START: ACALL DELAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น ถือว่าห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV        DPTR,#CTRL_P
MOV        A,#80H
MOVX       @DPTR,A
SETB       FLAG
LOOP:      MOV        R2,#00H
LOOP1:     MOV        A,R2
           RL         A
           ANL        A,#0FH
           JNB        FLAG,NEXT0
           SETB       ACC.0
           SJMP       NEXT1
NEXT0:     CLR        ACC.0
NEXT1:     MOVX       @DPTR,A
           ACALL      DELAY
           INC        R2
           CJNE       R2,#08H,LOOP1
           CPL        FLAG
           SJMP       LOOP
DELAY:     MOV        R0,#0FFH
DELAY1:    MOV        R1,#0FFH
           DJNZ       R1,S
           DJNZ       R0,DELAY1
           RET
           END

```

#### ผลการทดลอง

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุปผลการทดลอง

.....

.....

.....

.....

## คำถามท้ายการทดลอง

1. ในบรรทัดที่ 5 ถ้าแก้จาก “FLAG EQU 7FH” มาเป็น “FLAG EQU 70H”
2. จงเขียนโปรแกรมไฟว์ตามใบงานที่ 7 โดยให้แสดงผลที่พอร์ต A ของ 8255



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 8

### เรื่อง การใช้งานไทม์เมอร์ของ 8051

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถอธิบายการทำงานของไทม์เมอร์ของ 8051 ได้
2. เพื่อให้นักศึกษาสามารถนำหลักการของไทม์เมอร์ของ 8051 ไปประยุกต์ใช้งานได้

#### ทฤษฎี

ภายในชิพไมโครคอนโทรลเลอร์ MCS-51 จะมี ไทม์เมอร์/เคาน์เตอร์ 2 ตัว ขนาด 16 บิต คือ ไทม์เมอร์/เคาน์เตอร์ 0 และ ไทม์เมอร์/เคาน์เตอร์ 1 ส่วนเบอร์ 8032/8052 จะมีเพิ่มอีก 1 ชุด คือ ไทม์เมอร์/เคาน์เตอร์ 2 ขณะที่แต่ละ ไทม์เมอร์/เคาน์เตอร์ สามารถที่จะติดตั้งให้ทำงานได้เป็นตัวจับเวลาหรือตัวนับก็ได้ โดยวิธีการเซตหรือการเคลียร์บิตที่ตัวควบคุมในรีจิสเตอร์ TMOD ในกลุ่ม SFR

ในฟังก์ชันไทม์เมอร์ ตัวรีจิสเตอร์จะเพิ่มค่าทุก ๆ รอบแมชชีนไซเคิล ดังนั้นค่าของตัวเลขในรีจิสเตอร์จะเป็นจำนวนของรอบแมชชีนไซเคิล และเนื่องจากแต่ละรอบแมชชีนไซเคิลประกอบด้วย 12 คาบของออสซิลเลเตอร์ อัตราการนับแต่ละครั้งจะใช้เวลา 1/12 ของความถี่ออสซิลเลเตอร์

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 8 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 8

กำหนดให้ใช้ไทม์เมอร์ 1 โหมด 1

```

ORG          8000H

START:       MOV          TMOD,#01H

             MOV          A,#00H

LOOP:        MOV          PI,A

             ACALL        DELAY
  
```

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|        |      |          |
|--------|------|----------|
|        | CPL  | A        |
|        | SJMP | LOOP     |
| DELAY: | MOV  | TH0,#00H |
|        | MOV  | TL0,#00H |
|        | SETB | TR0      |
|        | JNB  | TF0,\$   |
|        | CLR  | TR0      |
|        | CLR  | TF0      |
|        | RET  |          |
|        | END  |          |

ผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. ถ้าต้องการใช้ไทม์เมอร์ 0 โหมด 1 ต้องแก้ไขที่บรรทัดใดบ้าง
2. ในบรรทัดที่ 3 ถ้าเปลี่ยนจาก “MOV A,#00H” เป็น “MOV A,#0AAH” จะเกิดอะไรขึ้น

## ใบงานที่ 9

### เรื่อง โปรแกรมการอินเทอร์รัพต์ของไทม์เมอร์ 0 โดยกำหนดให้ใช้โหมด 1

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถอธิบายการทำงานการอินเทอร์รัพต์จากไทม์เมอร์ได้
2. เพื่อให้นักศึกษาสามารถนำหลักการการอินเทอร์รัพต์จากไทม์เมอร์ไปประยุกต์ใช้งานได้

#### ทฤษฎี

เมื่อไทม์เมอร์/เคาน์เตอร์ เกิดโอเวอร์โฟลว์ ผลคือ ไทม์เมอร์แฟล็ก (TF0 และ TF1) จะเซตเป็น

1 แฟล็กถูกเคลียร์เป็น 0 เมื่ออินเทอร์รัพต์ทำให้โปรแกรมทำให้โปรแกรมเรียกโปรแกรมย่อยของไทม์เมอร์ในหน่วยความจำโปรแกรม

#### เครื่องมือและอุปกรณ์

- |                                             |       |         |
|---------------------------------------------|-------|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1     | เครื่อง |
| 2. คู่มือการแปลงรหัสคำสั่ง                  | 1     | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 ชุด | ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 9 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 9

|        |       |       |
|--------|-------|-------|
| PORT_A | EQU   | 6000H |
| PORT_B | EQU   | 6001H |
| PORT_C | EQU   | 6002H |
| CTRL_P | EQU   | 6003H |
|        | ORG   | 0000H |
|        | SJMP  | START |
|        | ORG   | 000BH |
|        | LJMP  | TOISR |
| START: | ACALL | DELAY |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      DPTR,#CTRL_P
MOV      A,#080H
MOVX     @DPTR,A
MOV      IE,#082H
MOV      TMOD,#01H
MOV      TH0,#00H
MOV      TL0,#00H
SETB     TR0
MOV      P1,#00H
MOV      DPTR,#PORT_A
MOV      A,#01H
LOOP:    MOVX     @DPTR,A
        ACALL    DELAY
        RL       A
        SJMP     LOOP
T0ISR:   CLR      TR0
        MOV      TH0,#00H
        MOV      TL0,#00H
        SETB     TR0
        CPL      P1.0
        RETI
DELAY:   MOV      R0,#0FFH
DELAY1:  MOV      R1,#0FFH
        DJNZ     R1,$
        DJNZ     R0,DELAY1
        RET
        END

```

ผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

คำถามท้ายการทดลอง

- 1. จงอธิบายหลักการการอินเทอร์รัพต์จากไทม์เมอร์มาพอสังเขป
- 2. จากคำสั่ง “MOV IE,#82H” หมายความว่าอย่างไร



## ใบงานที่ 10

### โปรแกรม Scan Matrix Switch โดยใช้หลักการของการอินเทอร์รัพต์

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถอธิบายหลักการทำงานของเมตริกสวิตช์ได้
2. เพื่อให้นักศึกษาสามารถนำเมตริกสวิตช์ไปประยุกต์ใช้งานได้

#### ทฤษฎี

คีย์บอร์ด เป็นอุปกรณ์ที่จะนำข้อมูลระบบไมโคร โปรเซสเซอร์ สามารถแบ่งออกได้เป็น 2 แบบ

คือ

1. แบบ ซิงเกิลคอลัมน์ (Single Column)
2. แบบ เอ็กซ์-วาย เมตริกซ์ (X-Y Matrix)

เนื่องจากโครงสร้างของคีย์บอร์ด ทำมาจากโลหะสัมผัสกัน ดังนั้นเวลา กด แล้วปล่อย จะทำให้เกิดอาการเด้ง ของสวิตช์ทำให้เกิด เบ๊าะ (Bounce) วิธีแก้ เบ๊าะ ทำได้ 2 วิธี คือ ใช้ฮาร์ดแวร์และซอฟต์แวร์

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. ไมโครคอมพิวเตอร์ส่วนบุคคล                | 1 | เครื่อง |
| 3. อิมูเลเตอร์                              | 1 | เครื่อง |

#### ลำดับขั้นตอนการทดลอง

1. นำเอาเมตริกซ์สวิตช์ มาต่อเข้ากับพอร์ต A และพอร์ต B ของ 8255 พอร์ตผู้ใช้งาน โดยที่ พอร์ต A ต่อเข้ากับคอลัมน์ และพอร์ต B ต่อเข้ากับแถว ของเมตริกซ์สวิตช์
2. ทำการแปลงคำสั่ง โปรแกรมการทดลองที่ 10 ให้เป็นคำสั่งภาษาเครื่อง
3. ป้อนคำสั่งภาษาเครื่องลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนคำสั่งเสร็จแล้ว กดคีย์ “GO” ตามด้วย ตำแหน่งเริ่มต้นของข้อมูล และกดคีย์ “Enter”
5. ทำการกดคีย์ที่เมตริกซ์สวิตช์ แล้วสังเกตผลการทดลอง และบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 10

กำหนดให้

-8255 อยู่ที่แอดเดรส 6000H-6003H

-ใช้อินเทอร์รัพต์ของไทม์เมอร์ 0 โหมด 1

-แสดงผลของการกดที่พอร์ต P1

เอกสารนี้เป็นเอกสารที่สร้างโดยระบบอัตโนมัติเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|        |       |              |
|--------|-------|--------------|
| PORT_A | EQU   | 6000H        |
| PORT_B | EQU   | 6001H        |
| PORT_C | EQU   | 6002H        |
| CTRL_P | EQU   | 6003H        |
| T0_H   | EQU   | 0F7H         |
| T0_L   | EQU   | 00H          |
| ROW    | EQU   | 78H          |
| COL    | EQU   | 79H          |
| KEY    | EQU   | 7AH          |
| F_KEY  | EQU   | 7FH          |
|        | ORG   | 0000H        |
|        | SJMP  | START        |
|        | ORG   | 000BH        |
|        | LJMP  | T01SR        |
| START: | ACALL | DELAY        |
|        | MOV   | DPTR,#CTRL_P |
|        | MOV   | A,#089H      |
|        | MOVB  | @DPTR,A      |
|        | MOV   | COL,#00H     |
|        | MOV   | KEY,#00H     |
|        | CLR   | F_KEY        |
|        | MOV   | IE,#082H     |
|        | MOV   | TMOD,#01H    |
|        | SETB  | TF0          |
| LOOP:  | JNB   | F_KEY,LOOP1  |
|        | MOV   | P1,KEY       |
| LOOP1: | MOV   | P1,#00H      |
|        | SJMP  | LOOP         |

|        |     |     |
|--------|-----|-----|
| T01SR: | CLR | TR0 |
|--------|-----|-----|

|  |      |     |
|--|------|-----|
|  | PUSH | DPH |
|  | PUSH | DPL |

เอกสารนี้เป็นเอกสารลิขสิทธิ์ส่วนตัวสำหรับใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่หรือดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH        ACC
PUSH        PSW
MOV         A,R0
PUSH        ACC
MOV         TH0,#T0_H
MOV         TL0,#T0_L
SETB        TR0
MOV         A,COL
MOV         DPTR,#PORT_B
MOVB        @DPTR,A
INC         DPL
JNB         F_KEY,T01SR2
MOV         A,KEY
SWAP        A
ANL         A,#0FH
CJNE        A,COL,T01SR0
MOV         A,KEY
ANL         A,#0FH
MOV         ROW,A
MOVB        A,@DPTR
ANL         A,#0FH
CJNE        A,#0FH,T01SR0
CLR         F_KEY
T01SR2:     MOVX        A,@DPTR
ANL         A,#0FH
CJNE        A,#0FH,T01SR3
SJMP        T01SR0
T01SR3:     MOV         KEY,A
MOV         A,COL
SWAP        A
ANL         A,#0FH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำหรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                ORL      A,KEY
                                MOV      KEY,A
                                SETB     F_KEY
T01SR0:  MOV      A,COL
                                INC      A
                                CJNE     A,#08H,T01SR1
                                MOV      A,#00H
T01SR1:  MOV      COL,A
                                POP      ACC
                                MOV      R0,A
                                POP      PSW
                                POP      ACC
                                POP      DPL
                                POP      DPH
                                RETI
DELAY:   MOV      R0,#0FFH
DELAY1:  MOV      R1,#0FFH
                                DJNZ     R1,$
                                DJNZ     R0,DELAY1
                                RET
                                END
```

ผลการทดลอง

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุปผลการทดลอง

.....

.....

.....

.....

## คำถามท้ายการทดลอง

1. “ORG 000BH” ในบรรทัดที่ 13 หมายความว่าอย่างไร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง  
เฉลยใบงานการทดลอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 1

### เรื่อง การติดต่อกับพอร์ตภายในของ 8051

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถอธิบายการทำงาน โปรแกรมการติดต่อกับพอร์ตภายในของ 8051 ได้

#### ทฤษฎี

พอร์ตภายในของ 8051 ไมโครคอนโทรลเลอร์จะประกอบไปด้วย พอร์ต 0, พอร์ต 1, พอร์ต 2 และ พอร์ต 3 เป็นกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ ซึ่งสามารถกระทำการแบบบิตได้ และเป็นตัวรีจิสเตอร์ที่แลตช์ของพอร์ต 0, พอร์ต 1, พอร์ต 2 และ พอร์ต 3 โดยที่ พอร์ต 0 เป็นบัสข้อมูลและบัสตำแหน่งไบต์ต่ำ, พอร์ต 1 จะเป็นพอร์ตที่ใช้งานทั่วไป, พอร์ตที่ 2 เป็นบัสตำแหน่งไบต์สูง และ พอร์ต 3 เป็นพอร์ตที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์

เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |

#### ลำดับขั้นตอนการทดลอง

1. นำคำสั่งจากโปรแกรมการทดลองที่ 1 ไปทำการแปลงเป็นคำสั่งของภาษาเครื่อง โดยการเปิดคู่มือการแปลรหัสคำสั่ง
2. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว โดยเริ่มจากตำแหน่ง 8000H
3. เมื่อป้อนโปรแกรมเสร็จแล้ว สั่งทำการประมวลผล โดยกดคีย์ “GO” และตามด้วย ตำแหน่งข้อมูล
4. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมทดลองที่ 1

|        |      |         |
|--------|------|---------|
|        | ORG  | 8000H   |
| START: | MOV  | A,#0FFH |
|        | CPL  | A       |
|        | MOV  | P1,A    |
|        | SJMP | \$      |
|        | END  |         |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ผลการทดลอง

จากการทดลอง ไคโอคเปล่งแสงที่พอร์ตหนึ่งของ 8051 จะติดทุกดวง

### สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า เราสามารถใช้งาน 8051 ไมโครคอนโทรลเลอร์ ได้ และ อธิบายการทำงานของโปรแกรมการติดต่อกับพอร์ตภายในของ 8051 ไมโครคอนโทรลเลอร์ได้

### คำถามท้ายการทดลอง

1. ให้นักศึกษาเขียนโปรแกรมส่งค่า AAH ออกที่ พอร์ตหนึ่งของ 8051 ไมโครคอนโทรลเลอร์

```

                ORG      8000H
START:         MOV      A,#0AAH
                CPL      A
                MOV      P1,A
                SJMP     $
                END

```

2. ให้นักศึกษาอธิบายการทำงานของโปรแกรมข้อที่ 1 ทีละขั้นตอน  
ขั้นตอนการทำงานของโปรแกรม

- บรรทัดที่ 1. กำหนดตำแหน่งเริ่มต้นของโปรแกรม
- บรรทัดที่ 2. นำค่า AAH ไปเก็บไว้ในรีจิสเตอร์ A
- บรรทัดที่ 3. กลับค่าที่อยู่ในรีจิสเตอร์เป็นค่าตรงกันข้าม (Complement)
- บรรทัดที่ 4. นำค่าที่อยู่ในรีจิสเตอร์ A ออกแสดงที่ไคโอคเปล่งแสงที่พอร์ตหนึ่ง
- บรรทัดที่ 5. กระโดดอยู่กับที่
- บรรทัดที่ 6. จบการทำงานของโปรแกรม

## ใบงานที่ 2

### เรื่อง โปรแกรมไฟวิ่งจากขวาไปซ้ายที่พอร์ต P1

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถอธิบายการทำงาน โปรแกรมควบคุมการทำงานของพอร์ตภายใน 8051 ได้

#### ทฤษฎี

โครงสร้างพอร์ตและการทำงานใน 8051 ไมโครคอนโทรลเลอร์มีพอร์ต 4 พอร์ต และทั้ง 4 พอร์ตเป็นแบบสองทิศทาง แต่ละพอร์ตจะประกอบด้วยแลตช์เป็น P0 ถึง P3 ของ SFR จะมีตัวขับเอาต์พุตและบัฟเฟอร์ (Buffer) อินพุตตัวขับเอาต์พุตของพอร์ต 0 และ พอร์ต 2 และบัฟเฟอร์อินพุตของพอร์ต 0 จะใช้งานสำหรับการเข้าถึงหน่วยความจำภายนอก ในการใช้งานลักษณะนี้ เอาต์พุตพอร์ต 0 จะทำหน้าที่เป็นตัวกำหนดไบต์ค่าของตำแหน่งหน่วยความจำภายนอก

#### เครื่องมือและอุปกรณ์

1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว 1 เครื่อง
2. คู่มือการแปลรหัสคำสั่ง 1 เล่ม

#### ลำดับขั้นตอนการทดลอง

1. นำคำสั่งจากโปรแกรมการทดลองที่ 2 ไปแปลงเป็นรหัสคำสั่งของภาษาเครื่อง
2. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว โดยเริ่มจากตำแหน่ง 8500H
3. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว สั่งทำการประมวลผล โดยกดคีย์ “GO” และตามด้วยตำแหน่งข้อมูล
4. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 2

|        |       |        |
|--------|-------|--------|
|        | ORG   | 8500H  |
| START: | MOV   | A,#01H |
| LOOP:  | CPL   | A      |
|        | MOV   | P1,A   |
|        | CPL   | A      |
|        | ACALL | DELAY  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|         |      |           |
|---------|------|-----------|
|         | RL   | A         |
|         | SJMP | LOOP      |
| DELAY:  | MOV  | R0,0FFH   |
| DELAY1: | MOV  | R1,0FFH   |
|         | DJNZ | R1,\$     |
|         | DJNZ | R0,DELAY1 |
|         | RET  |           |
|         | END  |           |

### ผลการทดลอง

จากการทดลอง เราจะสามารถสังเกตได้ ไคโอดเปล่งแสงที่พอร์ตหนึ่ง จะติดและดับและเลื่อนติดและดับไปที่หลอดเริ่มจากด้านขวาไปทางซ้าย

### สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า ทำให้เรามีความเข้าใจการทำงานของโปรแกรมการทำงานของการทำงานทดลองและสามารถที่จะอธิบายการทำงานเป็นลำดับขั้นตอนได้

### คำถามท้ายการทดลอง

1. ให้นักศึกษาอธิบายการทำงานของ โปรแกรมการทดลองที่ 2 ทีละขั้นตอนการทำงานของโปรแกรมการทดลองที่ 2

กำหนดตำแหน่งเริ่มต้นของโปรแกรม นำค่า “1” ไปเก็บไว้ในรีจิสเตอร์ A แล้วทำการกลับค่าที่อยู่ในรีจิสเตอร์ A แล้วส่งข้อมูลออกไปแสดงผลที่ไคโอดเปล่งแสงที่พอร์ตหนึ่ง จากนั้นก็ทำการหน่วงเวลา แล้วทำการกลับค่าที่อยู่ในรีจิสเตอร์ A ให้มีค่าเท่าเดิม เลื่อนข้อมูลในรีจิสเตอร์ A ไป 1 บิต กระโดดกลับไปวนการทำงานในตำแหน่งที่กำหนด

### ใบงานที่ 3

#### เรื่อง โปรแกรมไฟวิ่งจากซ้ายไปขวาที่พอร์ต P1

##### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียวได้
2. เพื่อให้นักศึกษาสามารถเขียนโปรแกรมใช้งานร่วมกับพอร์ตภายในของ 8051 ได้

##### ทฤษฎี

โครงสร้างพอร์ตและการทำงานใน 8051 ไมโครคอนโทรลเลอร์มีพอร์ต 4 พอร์ต และทั้ง 4 พอร์ตเป็นแบบสองทิศทาง แต่ละพอร์ตจะประกอบด้วยแลตช์เป็น P0 ถึง P3 ของ SFR จะมีตัวขับเอาต์พุตและบัฟเฟอร์ (Buffer) อินพุตตัวขับเอาต์พุตของพอร์ต 0 และ พอร์ต 2 และบัฟเฟอร์อินพุตของพอร์ต 0 จะใช้งานสำหรับการเข้าถึงหน่วยความจำภายนอก ในการใช้งานลักษณะนี้ เอาต์พุตพอร์ต 0 จะทำหน้าที่เป็นตัวกำหนดไบต์ค่าของตำแหน่งหน่วยความจำภายนอก

##### เครื่องมือและอุปกรณ์

1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียว 1 เครื่อง
2. คู่มือการแปลรหัสคำสั่ง 1 เล่ม

##### ลำดับขั้นตอนการทดลอง

1. นำคำสั่งจากโปรแกรมการทดลองที่ 3 ไปแปลงเป็น รหัสคำสั่งของภาษาเครื่อง
2. ป้อนรหัสคำสั่งลงในไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียว โดยเริ่มจากตำแหน่ง 9000H
3. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว สั่งทำการประมวลผล โดยกดคีย์ "GO" และตามด้วยตำแหน่งข้อมูล
4. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

##### โปรแกรมการทดลองที่ 3

|        |       |        |
|--------|-------|--------|
|        | ORG   | 9000H  |
| START: | MOV   | A,#80H |
| LOOP:  | CPL   | A      |
|        | MOV   | P1,A   |
|        | CPL   | A      |
|        | ACALL | DELAY  |
|        | RR    | A      |
|        | SJMP  | LOOP   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DELAY:    MOV        R0,0FFH
DELAY1:   MOV        R1,0FFH
          DJNZ       R1,$
          DJNZ       R0,DELAY1
          RET
          END

```

### ผลการทดลอง

จากการทดลอง จะสามารถสังเกตได้ว่า โดไดโอดเปล่งแสงที่พอร์ตหนึ่ง จะติดและดับและเลื่อนติดและดับไปที่ละดวงเริ่มจากด้านซ้ายไปทางขวา

### สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า ทำให้เรามีความเข้าใจการทำงานของโปรแกรมการทำงานของการทำงานของการทดลองและสามารถที่จะอธิบายการทำงานเป็นลำดับขั้นตอนได้

### คำถามท้ายการทดลอง

1. ให้นักศึกษาเขียนโปรแกรมไฟวิ่งจาก ทางด้านข้างวิ่งเข้าหาตรงกลาง ของพอร์ตหนึ่ง

```

          ORG        9000H
START:    MOV        R7,#04H
          MOV        DPTR,#TABLE
LOOP:     MOVX       @DPTR,A
          CPL        A
          MOV        P1,A
          ACALL      DELAY
          INC        DPTR
          DJNZ       R7,#00H,LOOP
          SJMP       START
          DELAY:     MOV        R0,#0FFH
          DELAY1:    MOV        R1,#0FFH
          DJNZ       R1,$
          DJNZ       R0,DELAY1
          RET
          END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 4

### เรื่อง การติดต่อกับอุปกรณ์ภายนอก ทางพอร์ตผู้ใช้งาน

#### วัตถุประสงค์

1. เพื่อให้ นักศึกษาสามารถใช้งาน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้ นักศึกษาสามารถอธิบายการทำงานของพอร์ตผู้ใช้งานได้

#### ทฤษฎี

การรับส่งของพอร์ตนั้น บางครั้งการส่งอาจจะไม่ได้ครั้งละ 8 บิตเสมอ ดังนั้นถ้ามีการส่งค่าข้อมูลที่ไม่ใช่ขนาด 8 บิต สามารถที่จะใช้โปรแกรมช่วยในการตรวจสอบการรับส่งข้อมูลได้เป็นค่าขนาดต่าง ๆ แล้วแต่ความต้องการของผู้ใช้งาน ซึ่งผู้ใช้สามารถกำหนดรูปแบบของการรับส่งข้อมูลได้ เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 | ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 4 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 4

โปรแกรมให้ 8255 ซึ่งอยู่ที่พอร์ต 6000H-6003H ให้ทุกพอร์ตทำงานเป็นเอาต์พุตทั้งหมดและ

#### ให้ส่งข้อมูล

- 0AAH ออกไปยังพอร์ต A
- 23H ออกไปยังพอร์ต B
- 2BH ออกไปยังพอร์ต C

|        |     |       |
|--------|-----|-------|
| PORT_A | EQU | 6000H |
| PORT_B | EQU | 6001H |
| PORT_C | EQU | 6002H |
| CTRL   | EQU | 6003H |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นหากมีให้ทดแทนเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                ORG            8000H
START:          ACALL         DELAY
                MOV           DPTR,#CTRL
                MOV           A,#80H
                MOVX          @DPTR,A
                MOV           DPTR,#PORT_A
                MOV           A,#0AAH
                MOVX          @DPTR,A
                MOV           DPTR,#PORT_B
                MOV           A,#23H
                MOVX          @DPTR,A
                MOV           DPTR,#PORT_C
                MOV           A,#2BH
                MOVX          @DPTR,A
                SJMP          $
DELAY:          MOV           R0,#0FFH
DELAY1:         MOV           R1,#0FFH
                DJNZ         R1,$
                DJNZ         R0,DELAY1
                RET
                END

```

#### ผลการทดลอง

จากผลการทดลอง ที่ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานพอร์ต 1 ไดโอดเปล่งแสงจะติด 1 ดวงแล้ว 1 ดวงสลับกันไป ที่พอร์ต B ไดโอดเปล่งแสงจะระดับ 2 ดวง, ติด 1 ดวง, ดับ 3 ดวง และ ติด อีก 2 ดวง จากบิตสูงมาบิตต่ำตามลำดับ ที่พอร์ต C ไดโอดเปล่งแสงจะระดับ 2 ดวง, ติด 1 ดวง, ดับ 1 ดวง, ติด 1 ดวง, ดับ 1 ดวง และ ติดอีก 2 ดวง จากบิตสูงมาบิตต่ำตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า ในการที่จะใช้งานพอร์ตผู้ใช้งานนั้นเราต้องเซตให้ 8255 เตรียมพร้อมการทำงานตามที่เรากำลังต้องการว่าจะให้เป็นพอร์ต เป็นอินพุตพอร์ตหรือเอาต์พุตพอร์ต

## คำถามท้ายการทดลอง

1. ถ้าต้องการเอาข้อมูล FFH ออกมาที่พอร์ต A, พอร์ต B และ พอร์ต C จะต้องทำการเปลี่ยนคำสั่งในตำแหน่งใดบ้าง

การแก้ไขโปรแกรม

... บรรทัดที่ 11 เปลี่ยนจาก MOV A,#0AAH เป็น MOV A,#0FFH  
 บรรทัดที่ 14 เปลี่ยนจาก MOV A,#023H เป็น MOV A,#0FFH  
 บรรทัดที่ 15 เปลี่ยนจาก MOV A,#02BH เป็น MOV A,#0FFH

2. ให้นักศึกษาอธิบายการควบคุมการทำงานของพอร์ตผู้ใช้งาน มาพอสังเขป

การควบคุมการทำงานของ 8255 พอร์ตผู้ใช้งาน

ต้องเซตให้ 8255 งานตามหน้าที่ ที่ผู้ใช้อยู่จะต้องให้เป็นพอร์ตอินพุต หรือว่าพอร์ตเอาต์พุตโดยมีคำสั่งในการเซต (ดูจากคู่มือการใช้งาน 8255) แล้วเราสามารถนำข้อมูลเข้าและข้อมูลออกผ่าน 8255 พอร์ตผู้ใช้งานได้

## ใบงานที่ 5

### เรื่อง โปรแกรมเซตบิต 7 ของ 8255

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยวได้
2. เพื่อให้นักศึกษาสามารถเซตบิตของพอร์ตผู้ใช้งานได้

#### ทฤษฎี

นอกจากผู้ใช้งานจะใช้ พอร์ต A, พอร์ต B และ พอร์ต C ในการโปรแกรมให้เป็นอินพุตและเอาต์พุตแล้วยังสามารถที่จะโปรแกรมพอร์ต C บิต PC0-PC7 ให้มีสถานะ ลอจิกเป็น “0” หรือ “1” เพื่อใช้เป็นสัญญาณสโตรบ (Strobe) โดย โหมด 1 จะใช้พอร์ต A และ พอร์ต B ในการส่งข้อมูล โดยส่วนใช้พอร์ต C เป็นพอร์ตตรวจสอบความพร้อม ในการทำงาน

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 | ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 5 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 5

กำหนดให้พอร์ตของ 8255 อยู่ที่ตำแหน่งแอดเดรส 6000H-6003H

|        |       |            |
|--------|-------|------------|
| PORT_A | EQU   | 6000H      |
| PORT_B | EQU   | 6001H      |
| PORT_C | EQU   | 6002H      |
| CTRL   | EQU   | 6003H      |
|        | ORG   | 8000H      |
| START: | ACALL | DELAY      |
|        | MOV   | DPTR,#CTRL |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      A,#80H
MOVX     @DPTR,A
MOV      A,#01H
MOV      DPTR,#PORT_A
MOVX     @DPTR,A
SJMP     $
DELAY:   MOV      R0,#0FFH
DELAY1:  MOV      R1,#0FFH
          DJNZ     R1,$
          DJNZ     R0,DELAY1
          RET
          END

```

### ผลการทดลอง

จากการทดลอง บิต 7 ของ 8255 พอร์ต 1 จะเป็นลอจิก “1”

### สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า เราสามารถที่จะทำการเซตบิต 7 ของ 8255 ได้หลายวิธี และในวิธีของโปรแกรมที่ 5 ก็เป็นวิธีหนึ่งในการเซตบิต ของ 8255

### คำถามท้ายการทดลอง

1. ถ้าต้องการเคลียร์บิตที่ 7 ของ 8255 จะต้องแก้ไขตำแหน่งใดบ้าง  
การแก้ไขโปรแกรม

บรรทัดที่ 10 เปลี่ยนจาก MOV A,#01H เป็น MOV A,#00H

2. ถ้าต้องการเซตบิตอื่นนอกจากบิตที่ 7 ของ 8255 สามารถทำได้หรือไม่ เพราะเหตุใด  
ได้ และ สามารถใช้วิธีในโปรแกรมที่ 5 เซตบิตอื่น ๆ ของ 8255 ได้เช่นกัน

```

MOV      A,#80H
MOVX     @DPTR,A
LOOP:    MOV      R2,#00H
LOOP1:   MOV      A,R2
         RL       A
         ANL      A,#0FH
         SETB     ACC.0
         MOVX     @DPTR,A
         ACALL    DELAY
         CLR      ACC.0
         MOVX     @DPTR,A
         INC      R2
         CJNE     R2,#08H,LOOP1
         SJMP     LOOP
DELAY:    MOV      R0,#0FFH
DELAY1:   MOV      R1,#0FFH
         DJNZ     R1,$
         DJNZ     R0,DELAY1
         RET
         END

```

### ผลการทดลอง

จากการทดลอง ไดโอดเปล่งแสงดวงที่ 1 ของพอร์ต C จะติด ส่วนดวงอื่นจะดับ และหลังจากนั้น ไดโอดเปล่งแสงดวงที่ 2 จะติด ในขณะที่ดวงอื่น ๆ จะดับ จะเป็นลักษณะเช่นนี้กับ ไดโอดเปล่งแสงทุก ๆ ดวง ซึ่งจะทำให้เหมือนกับ ไฟวิ่งจาก ขวามือ ไป ซ้ายมือ

### สรุปผลการทดลอง

จากผลการทดลองสรุปได้ว่า ถ้าเราใช้หลักการของการส่งข้อมูลทางพอร์ตควบคุมก่อน เราจะสามารถควบคุมการทำงานของพอร์ตต่าง ๆ ของ 8255 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำถามท้ายการทดลอง

1. ในบรรทัดที่ 8 ถ้าแก้จาก MOV A,#80H ไปเป็น MOV A,#00H จะเกิดอะไรขึ้น  
โปรแกรมจะไม่ทำงานเนื่องจากเป็นการย้ายข้อมูล 00H ไปเก็บไว้ที่ พอร์ตควบคุม
2. จงเขียนโปรแกรมไฟวิ่งจากซ้ายไปขวาที่พอร์ต A ของ 8255

```

PORT_A    EQU        6000H
PORT_B    EQU        6001H
PORT_C    EQU        6002H
CTRL      EQU        6003H

          ORG         8000H

START:    ACALL       DELAY
          MOV         DPTR,#CTRL
          MOV         A,#80H
          MOVX        @DPTR,A
LOOP:     MOV         R2,#00H
LOOP1:    MOV         A,R2
          RR          A
          ANL         A,#0FH
          SETB        ACC.0
          MOVX        @DPTR,A
          ACALL       DELAY
          CLR         ACC.0
          MOVX        @DPTR,A
          INC         R2
          CJNE        R2,#08H,LOOP1
          SJMP        LOOP

DELAY:    MOV         R0,#0FFH
DELAY1:   MOV         R1,#0FFH
          DJNZ        R1,$

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อ R0,DELAY1 นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ RET เปลี่ยนเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

## ใบงานที่ 7

### เรื่อง โปรแกรมไฟวิ่งรูปแบบที่ 2 ที่พอร์ต C ของ 8255

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถใช้งานพอร์ตของ 8255 ได้
2. เพื่อให้นักศึกษาสามารถนำพอร์ต 8255 ไปประยุกต์ใช้งานได้

#### ทฤษฎี

นอกจากผู้ใช้งานจะใช้ พอร์ต A, พอร์ต B และ พอร์ต C ในการโปรแกรมให้เป็นอินพุตและเอาต์พุต แล้วยังสามารถที่จะโปรแกรมพอร์ต C บิต PC0-PC7 ให้มีสถานะ ลอจิกเป็น “0” หรือ “1” เพื่อใช้เป็นสัญญาณสโตรบ (Strobe) โดย โหมด 1 จะใช้พอร์ต A และ พอร์ต B ในการส่งข้อมูล โดยส่วนใช้พอร์ต C เป็นพอร์ตตรวจสอบความพร้อมในการทำงาน

#### เครื่องมือและอุปกรณ์

- |                                            |           |
|--------------------------------------------|-----------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียว | 1 เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                  | 1 เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน    | 1 ชุด     |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 7 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 7

กำหนดให้พอร์ตของ 8255 อยู่ที่แอดเดรส 6000H-6003H

|        |       |       |
|--------|-------|-------|
| PORTA  | EQU   | 6000H |
| PORTB  | EQU   | 6001H |
| PORTC  | EQU   | 6002H |
| CTRL_P | EQU   | 6003H |
| FLAG   | EQU   | 7FH   |
|        | ORG   | 8000H |
| START: | ACALL | DELAY |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      DPTR,#CTRL_P
MOV      A,#80H
MOVX     @DPTR,A
SETB     FLAG
LOOP:    MOV      R2,#00H
LOOP1:   MOV      A,R2
         RL       A
         ANL      A,#0FH
         JNB      FLAG,NEXT0
         SETB     ACC.0
         SJMP     NEXT1
NEXT0:   CLR      ACC.0
NEXT1:   MOVX     @DPTR,A
         ACALL    DELAY
         INC      R2
         CJNE     R2,#08H,LOOP1
         CPL      FLAG
         SJMP     LOOP
DELAY:   MOV      R0,#0FFH
DELAY1:  MOV      R1,#0FFH
         DJNZ     R1,$
         DJNZ     R0,DELAY1
         RET
         END

```

### ผลการทดลอง

จากการทดลอง ไคโอดเปล่งแสงดวงที่ 1 ของพอร์ต C จะติด ส่วนดวงอื่นจะดับ และหลังจากนั้น ไคโอดเปล่งแสงดวงที่ 2 จะติด ในขณะที่ดวงอื่น ๆ จะดับ จะเป็นลักษณะเช่นนี้กับไคโอดเปล่งแสงทุก ๆ ดวง ซึ่งจะทำให้เหมือนกับ ไฟวิ่งจาก ขวามือไปซ้ายมือ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุปผลการทดลอง

จากผลการทดลองสรุปได้ว่า ถ้าเราใช้หลักการของการส่งข้อมูลทางพอร์ตควบคุม ก่อน เราจะสามารถ ควบคุมการทำงานของพอร์ตต่าง ๆ ของ 8255 ได้

## คำถามท้ายการทดลอง

1. ในบรรทัดที่ 5 ถ้าแก้จาก “FLAG EQU 7FH” มาเป็น “FLAG EQU 70H”

โปรแกรมยังคงทำงานได้เหมือนเดิม เนื่องจากที่ตำแหน่ง 70H ก็สามารถพักข้อมูลได้เช่นเดียวกับที่ ตำแหน่ง 7FH

2. จงเขียนโปรแกรมไฟวิ่งตามใบงานที่ 7 โดยให้แสดงผลที่พอร์ต A ของ 8255

```

PORTA EQU 6000H
PORTB EQU 6001H
PORTC EQU 6002H
CTRL_P EQU 6003H
FLAG EQU 7FH
ORG 8000H
START: ACALL DELAY
      MOV DPTR,#CTRL_P
      MOV A,#80H
      MOVX @DPTR,A
      MOV DPTR,#PORT C
      MOV A,#01H
      MOVX @DPTR,A
      SETB FLAG
LOOP:  MOV R2,#00H
LOOP1: MOV A,R2
      RL A
      ANL A,#0FH

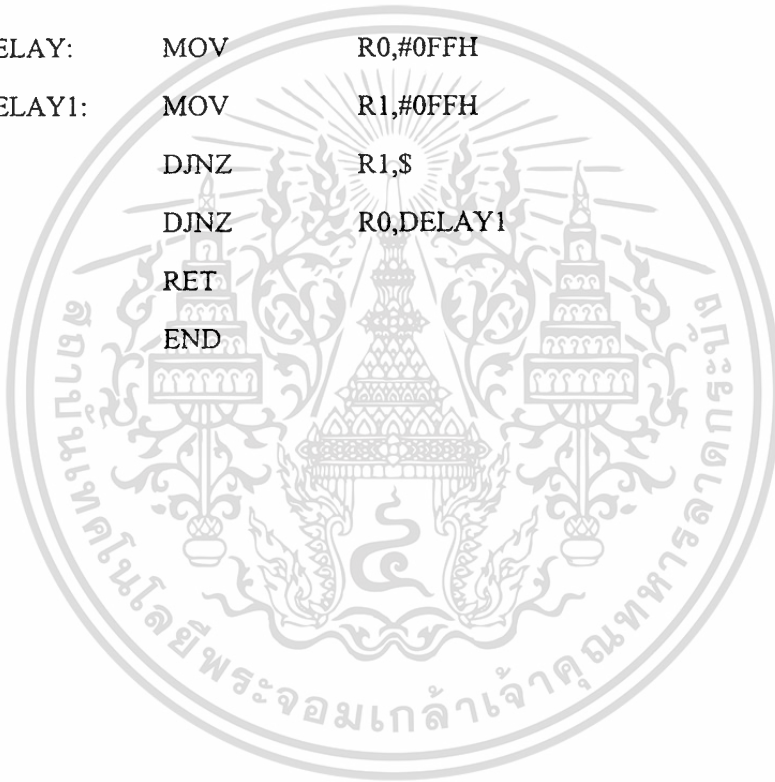
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีการเปลี่ยนแปลงเนื้อหาใดๆ อย่างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                SJMP      NEXT1
NEXT0:          CLR       ACC.0
NEXT1:          MOVX      @DPTR,A
                ACALL     DELAY
                INC       R2
                CJNE      R2,#08H,LOOP1
                CPL       FLAG
                SJMP      LOOP
DELAY:          MOV       R0,#0FFH
DELAY1:         MOV       R1,#0FFH
                DJNZ      R1,$
                DJNZ      R0,DELAY1
                RET
                END

```



## ใบงานที่ 8

### เรื่อง การใช้งานไทม์เมอร์ของ 8051

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถอธิบายการทำงานของไทม์เมอร์ของ 8051 ได้
2. เพื่อให้นักศึกษาสามารถนำหลักการของไทม์เมอร์ของ 8051 ไปประยุกต์ใช้งานได้

#### ทฤษฎี

ภายในชิพไมโครคอนโทรลเลอร์ MCS-51 จะมี ไทม์เมอร์/เคาน์เตอร์ 2 ตัว ขนาด 16 บิต คือ ไทม์เมอร์/เคาน์เตอร์ 0 และ ไทม์เมอร์/เคาน์เตอร์ 1 ส่วนเบอร์ 8032/8052 จะมีเพิ่มอีก 1 ชุด คือ ไทม์เมอร์/เคาน์เตอร์ 2 ขณะที่แต่ละ ไทม์เมอร์/เคาน์เตอร์ สามารถที่จะติดตั้งให้ทำงานได้เป็นตัวจับเวลาหรือตัวนับก็ได้ โดยวิธีการเซตหรือการเคลียร์บิตที่ตัวควบคุมในรีจิสเตอร์ TMOD ในกลุ่ม SFR

ในฟังก์ชันไทม์เมอร์ ตัวรีจิสเตอร์จะเพิ่มค่าทุก ๆ รอบแมชชีนไซเคิล ดังนั้นค่าของตัวเลขในรีจิสเตอร์จะเป็นจำนวนของรอบแมชชีนไซเคิล และเนื่องจากแต่ละรอบแมชชีนไซเคิลประกอบด้วย 12 คาบของออสซิลเลเตอร์ อัตราการนับแต่ละครั้งจะใช้เวลา 1/12 ของความถี่ออสซิลเลเตอร์

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |

#### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 8 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 8

กำหนดให้ใช้ไทม์เมอร์ 1 โหมด 1

```

ORG          8000H

START:       MOV          TMOD,#01H

              MOV          A,#00H

LOOP:        MOV          P1,A
              ACALL        DELAY
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|        |      |          |
|--------|------|----------|
|        | CPL  | A        |
|        | SJMP | LOOP     |
| DELAY: | MOV  | TH0,#00H |
|        | MOV  | TL0,#00H |
|        | SETB | TR0      |
|        | JNB  | TF0,\$   |
|        | CLR  | TR0      |
|        | CLR  | TF0      |
|        | RET  |          |
|        | END  |          |

### ผลการทดลอง

จากการทดลอง โค้ดเปิดแสงที่พอร์ต 1 ของ 8051 จะดับหมดทุกดวง หลังจากนั้นก็จะติดสว่างทุกดวง สลับกันไป จนดูเหมือนว่า ไฟกระพริบ

### สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า เราสามารถใช้ไทม์เมอร์มาเป็น โปรแกรมหน่วงเวลาได้ โดยก่อนที่จะทำการตั้งค่าเวลานั้น เราต้องทำการ เซตค่าพารามิเตอร์ต่าง ๆ ของไทม์เมอร์เสียก่อน เช่น TH0 หรือ TL0 เป็นต้น

### คำถามท้ายการทดลอง

1. ถ้าต้องการใช้ไทม์เมอร์ 0 โหมด 1 ต้องแก้ไขที่บรรทัดใดบ้าง  
แก้ไขบรรทัดที่ 8-13 เป็น

|        |      |          |
|--------|------|----------|
| DELAY: | MOV  | TH1,#00H |
|        | MOV  | TL1,#00H |
|        | SETB | TR1      |
|        | JNB  | TF1,\$   |
|        | CLR  | TR1      |
|        | CLR  | TF1      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ในบรรทัดที่ 3 ถ้าเปลี่ยนจาก “MOV A,#00H” เป็น “MOV A,#0AAH” จะเกิดอะไรขึ้น

ไดโอดเปล่งแสงจะกระพริบเป็นเลขฐานสิบหก คือ 0AAH และ 55H สลับกันที่พอร์ตหนึ่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ใบงานที่ 9

เรื่อง โปรแกรมการอินเทอร์รัพต์ของไทม์เมอร์ 0 โดยกำหนดให้ใช้โหมด 1

### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถอธิบายการทำงานการอินเทอร์รัพต์จากไทม์เมอร์ได้
2. เพื่อให้นักศึกษาสามารถนำหลักการการอินเทอร์รัพต์จากไทม์เมอร์ไปประยุกต์ใช้งานได้

### ทฤษฎี

เมื่อไทม์เมอร์/เคาน์เตอร์ เกิดโอเวอร์โฟลว์ ผลคือ ไทม์เมอร์เฟลก (TF0 และ TF1) จะเซตเป็น 1 เฟลกถูกเคลียร์เป็น 0 เมื่ออินเทอร์รัพต์ทำให้โปรแกรมทำให้โปรแกรมเรียกโปรแกรมย่อยของไทม์เมอร์ในหน่วยความจำโปรแกรม

### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. คู่มือการแปลรหัสคำสั่ง                   | 1 | เล่ม    |
| 3. ไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งาน     | 1 | ชุด     |

### ลำดับขั้นตอนการทดลอง

1. นำไดโอดเปล่งแสงแสดงผลที่พอร์ตผู้ใช้งานต่อกับพอร์ตผู้ใช้งาน
2. ทำการแปลงโปรแกรมการทดลองที่ 9 ให้เป็นรหัสคำสั่งภาษาเครื่อง
3. ป้อนรหัสคำสั่งลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนรหัสคำสั่งเสร็จแล้ว กดคีย์ “GO” แล้วตามตำแหน่งเริ่มต้นของข้อมูล
5. สังเกตผลการทดลอง แล้วบันทึกผลการทดลอง

### โปรแกรมการทดลองที่ 9

|        |       |       |
|--------|-------|-------|
| PORT_A | EQU   | 6000H |
| PORT_B | EQU   | 6001H |
| PORT_C | EQU   | 6002H |
| CTRL_P | EQU   | 6003H |
|        | ORG   | 0000H |
|        | SJMP  | START |
|        | ORG   | 000BH |
|        | LJMP  | T0ISR |
| START: | ACALL | DELAY |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      DPTR,#CTRL_P
MOV      A,#080H
MOVX     @DPTR,A
MOV      IE,#082H
MOV      TMOD,#01H
MOV      TH0,#00H
MOV      TL0,#00H
SETB     TR0
MOV      P1,#00H
MOV      DPTR,#PORT_A
MOV      A,#01H
LOOP:    MOVX     @DPTR,A
        ACALL    DELAY
        RL       A
        SJMP     LOOP
T0ISR:   CLR      TR0
        MOV      TH0,#00H
        MOV      TL0,#00H
        SETB     TR0
        CPL      P1.0
        RETI
DELAY:   MOV      R0,#0FFH
DELAY1:  MOV      R1,#0FFH
        DJNZ     R1,$
        DJNZ     R0,DELAY1
        RET
        END

```

### ผลการทดลอง

จากการทดลอง ไคโอดเปล่งแสงดวงที่ 1 ของพอร์ต C จะติด ส่วนดวงอื่นจะดับ และหลังจากนั้น ไคโอดเปล่งแสงดวงที่ 2 จะติด ในขณะที่ดวงอื่น ๆ จะดับ จะเป็นลักษณะเช่นนี้กับ ไคโอดเปล่งแสงทุก ๆ ดวง ซึ่งจะทำให้เหมือนกับ ไฟวิ่งจาก ขวามือ ไป ซ้ายมือ

### สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า เมื่อเกิดการอินเตอร์รัพต์จากไทม์เมอร์ โปรแกรมจะทำการกระโดดไปยังตำแหน่ง ของโปรแกรมบริการอินเตอร์รัพต์ทันที

### คำถามท้ายการทดลอง

1. จงอธิบายหลักการการอินเตอร์รัพต์จากไทม์เมอร์มาพอสังเขป

เมื่อไทม์เมอร์ได้มีการนับเลขขึ้น จนถึงค่า THx จะทำให้เกิดโอเวอร์โฟลว์ หลังจากนั้น โปรแกรมจะทำการกระโดดไปยังตำแหน่งของโปรแกรมบริการอินเตอร์รัพต์ทันทีและเมื่อทำงานที่ โปรแกรมบริการอินเตอร์รัพต์เสร็จเรียบร้อยแล้ว ก็จะมีการเคลียร์บิตโอเวอร์โฟลว์ และจะกระโดดกลับไปยังตำแหน่ง ของโปรแกรมหลักทันที

2. จากคำสั่ง “MOV IE,#82H” หมายความว่า อย่างไร

ยอมให้มีการอินเตอร์รัพต์จากไทม์เมอร์ 0 อย่างเดียว เท่านั้น

## ใบงานที่ 10

### โปรแกรม Scan Matrix Switch โดยใช้หลักการของการอินเตอร์รัพต์

#### วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถอธิบายหลักการทำงานของเมตริกสวิทช์ได้
2. เพื่อให้นักศึกษาสามารถนำเมตริกสวิทช์ไปประยุกต์ใช้งานได้

#### ทฤษฎี

คีย์บอร์ด เป็นอุปกรณ์ที่จะนำข้อมูลระบบไมโครโปรเซสเซอร์ สามารถแบ่งออกได้เป็น 2 แบบ คือ

1. แบบ ซิงเกิลคอลัมน์ (Single Column)
2. แบบ เอ็กซ์-วาย เมทริกซ์ (X-Y Matrix)

เนื่องจากโครงสร้างของคีย์บอร์ด ทำมาจากโลหะสัมผัสกัน ดังนั้นเวลา กด แล้วปล่อย จะทำให้เกิดอาการเค็ง ของสวิทช์ทำให้เกิด เบ๊าะ (Bounce)วิธีแก้ เบ๊าะ ทำได้ 2 วิธี คือ ใช้ฮาร์ดแวร์และซอฟต์แวร์

#### เครื่องมือและอุปกรณ์

- |                                             |   |         |
|---------------------------------------------|---|---------|
| 1. 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว | 1 | เครื่อง |
| 2. ไมโครคอมพิวเตอร์ส่วนบุคคล                | 1 | เครื่อง |
| 3. อิมูเลเตอร์                              | 1 | เครื่อง |

#### ลำดับขั้นตอนการทดลอง

1. นำเอาเมตริกสวิทช์ มาต่อเข้ากับพอร์ต A และพอร์ต B ของ 8255 พอร์ตผู้ใช้งาน โดยที่ พอร์ต A ต่อเข้ากับคอลัมน์ และพอร์ต B ต่อเข้ากับแถว ของเมตริกสวิทช์
2. ทำการแปลงคำสั่งโปรแกรมการทดลองที่ 10 ให้เป็นคำสั่งภาษาเครื่อง
3. ป้อนคำสั่งภาษาเครื่องลงใน ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดี่ยว
4. เมื่อป้อนคำสั่งเสร็จแล้ว กดคีย์ “GO” ตามด้วย ตำแหน่งเริ่มต้นของข้อมูล และกดคีย์ “Enter”
5. ทำการกดคีย์ที่เมตริกสวิทช์ แล้วสังเกตผลการทดลอง และบันทึกผลการทดลอง

#### โปรแกรมการทดลองที่ 10

กำหนดให้

-8255 อยู่ที่แอดเดรส 6000H-6003H

-ใช้อินเตอร์รัพต์ของไทม์เมอร์ 0 โหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
-แสดงผลของการกดที่พอร์ต P1

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PORT_A    EQU        6000H
PORT_B    EQU        6001H
PORT_C    EQU        6002H
CTRL_P    EQU        6003H
T0_H      EQU        0F7H
T0_L      EQU        00H
ROW       EQU        78H
COL       EQU        79H
KEY       EQU        7AH
F_KEY     EQU        7FH

ORG        0000H
SJMP      START
ORG        000BH
LJMP      T0ISR
START:    ACALL      DELAY
          MOV        DPTR,#CTRL_P
          MOV        A,#089H
          MOVX       @DPTR,A
          MOV        COL,#00H
          MOV        KEY,#00H
          CLR        F_KEY
          MOV        IE,#082H
          MOV        TMOD,#01H
          SETB       TF0
LOOP:     JNB        F_KEY,LOOP1
          MOV        P1,KEY
LOOP1:    MOV        P1,#00H
          SJMP       LOOP
T0ISR:    CLR        TR0
          PUSH       DPH
          PUSH       DPL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้นำไปเปลี่ยนแปลงเนื้อหาหรือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH      ACC
PUSH      PSW
MOV       A,R0
PUSH      ACC
MOV       TH0,#T0_H
MOV       TL0,#T0_L
SETB      TR0
MOV       A,COL
MOV       DPTR,#PORT_B
MOVB      @DPTR,A
INC       DPL
JNB       F_KEY,T01SR2
MOV       A,KEY
SWAP      A
ANL       A,#0FH
CJNE      A,COL,T01SR0
MOV       A,KEY
ANL       A,#0FH
MOV       ROW,A
MOVB      A,@DPTR
ANL       A,#0FH
CJNE      A,#0FH,T01SR0
CLR       F_KEY
T01SR2:   MOV      A,@DPTR
          ANL      A,#0FH
          CJNE     A,#0FH,T01SR3
          SJMP     T01SR0
T01SR3:   MOV      KEY,A
          MOV      A,COL
          SWAP     A
          ANL      A,#0FH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่หรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                ORL        A,KEY
                                MOV        KEY,A
                                SETB      F_KEY
T01SR0:  MOV        A,COL
                                INC        A
                                CJNE      A,#08H,T01SR1
                                MOV        A,#00H
T01SR1:  MOV        COL,A
                                POP        ACC
                                MOV        R0,A
                                POP        PSW
                                POP        ACC
                                POP        DPL
                                POP        DPH
                                RETI
DELAY:   MOV        R0,#0FFH
DELAY1:  MOV        R1,#0FFH
                                DJNZ      R1,S
                                DJNZ      R0,DELAY1
                                RET
                                END

```

#### ผลการทดลอง

จากการทดลอง เมื่อมีการกดคีย์ ไค ๆ ที่เมทริกสวิตช์ ผลที่เกิดขึ้นไดโอดเปล่งแสงที่พอร์ตหนึ่งของ 8051 จะแสดงเป็นรหัสเลขฐานสิบหก ตามค่าตำแหน่งรหัสของคีย์ ที่เมทริกสวิตช์

#### สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า เราสามารถใช้หลักการสแกนคีย์บอร์ด มาต่อเข้ากับ 8255 เพื่อที่จะสามารถควบคุม การทำงานของคีย์บอร์ด ซึ่งจากไปงานนี้ เราใช้ไอซี 74LS145 เป็นตัวสแกนคีย์บอร์ด ทำให้ประหยัดขา (ของ 8255) ที่จะใช้งานได้นั้น เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### คำถามท้ายการทดลอง

1. “ORG 000BH” ในบรรทัดที่ 13 หมายความว่าอย่างไร  
เป็นค่าของตำแหน่งเริ่มต้นของโปรแกรมบริการ อินเทอร์เน็ต ไทม์เมอร์ 0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**ภาคผนวก จ**  
**โปรแกรมมอนิเตอร์ 8051 ไมโครคอนโทรลเลอร์แบบแผ่นพิมพ์เดียว**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## MONITOR.ASM

```

1
2 ;*****
3 ; FILENAME    MONITOR.ASM
4 ; DESCRIPTION MCS - 51 SINGLE BOARD MICROCONTROLLER
5 ; MONITOR PROGRAM ***** VERSION 1.00 *****
6 ; HARDWARE    MCS - 51 BOARD
7 ; ASSEMBLER    SXA51
8 ; FACULTY OF INDUSTRIAL EDUCATION
9 ; KING MONGKUT 'S INSTITUTE OF TECHNOLOGY LADKRABANG
10 ;*****
11
12 ;***** SYSTEM PORT*****
2000= 13     PORTA      EQU      2000H
2001= 14     PORTB      EQU      2001H
2002= 15     PORTC      EQU      2002H
2003= 16     PCTRL      EQU      2003H
17
18 ;***** USER PORT *****
6000= 19     USERA      EQU      6000H
6001= 20     USERB      EQU      6001H
6002= 21     USERC      EQU      6002H
6003= 22     UCTRL      EQU      6003H
23 ;*****
9000= 24     SERIADD     EQU      9000H
8000= 25     RAM        EQU      8000H
F800= 26     MEMADD     EQU      0F800H

27 ;***** SPECIAL KEY *****
28
0070= 29     KEYBUF      EQU      70H
001B= 30     ESC        EQU      1BH
0014= 31     BACKSP     EQU      14H
0022= 32     DECREASE   EQU      22H
0023= 33     INCREASE   EQU      23H
0013= 34     ENTER      EQU      13H
0020= 35     SPACE      EQU      20H
36
37 ;*****
0097= 38     TOOLBT      EQU      P1.7
0076= 39     STMFAG      EQU      76H ;RECV - STREAM FLAG
0078= 40     REMFAG      EQU      78H ;REM - REMOTE FUNCTION
FLAG

41
42 ;***** GENERRAL AREA *****
FF00= 43     STA        EQU      0FF00H
FF02= 44     ENDA       EQU      0FF02H
FF04= 45     DEAD       EQU      0FF04H
46 ;*****
0000 47     ORG          0000H
48
49
50

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่วาทกรรมใดๆ ที่อื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0000 7810    51      MOV      R0,#10H
0002 12036C  52      LCALL     DELAY3
0005 902003  53      MOV      DPTR,#PCTRL
0008 7489    54      MOV      A,#89H
000A F0      55      MOVX     @DPTR,A
000B 7590FF  56      MOV      P1,#0FFH
57
58 ;***** MAIN PROGRAM *****
59
000E 1202FD  60      LCALL     SETLCD
0011 900A14  61      MOV      DPTR,#STR4
0014 120383  62      LCALL     PRINT
0017 7820    63      MOV      R0,#20H
0019 12036C  64      LCALL     DELAY3
65
001C 1202FD  66      LCALL     SETLCD
001F 900A36  67      MOV      DPTR,#STR5
0022 120383  68      LCALL     PRINT
0025 7860    69      MOV      R0,#60H
0027 12036C  70      LCALL     DELAY3
71
002A 1204A9  72      MON:      LCALL     HBEEP
002D 1202FD  73      LCALL     SETLCD
0030 9009E7  74      MOV      DPTR,#STR1
0033 120383  75      LCALL     PRINT
0036 120290  76      MON1:      LCALL     KEY
0039 C0E0    77      PUSH     ACC
003B 1204A2  78      LCALL     OBEEP
003E D0E0    79      POP      ACC
0040 B43108  80      CJNE     A,#31H,MON3
0043 12013A  81      LCALL     ADDR
0046 120178  82      LCALL     ADDR4
0049 80DF    83      SJMP     MON
84
004B B41705  85      MON3:      CJNE     A,#17H,MON4
004E 12006C  86      LCALL     FUNC
0051 80D7    87      SJMP     MON
88
0053 B41905  89      MON4:      CJNE     A,#19H,MON5
0056 12069D  90      LCALL     RUN
0059 80CF    91      SJMP     MON
92
005B B430CC  93      MON5:      CJNE     A,#30H,MON
005E 1202FD  94      LCALL     SETLCD
0061 900ABE  95      MOV      DPTR,#STR3
0064 120383  96      LCALL     PRINT
0067 120713  97      LCALL     RECV
006A 80BE    98      SJMP     MON
99
100 ;***** FUNCTION MAIN *****
101
006C 1202FD  102     FUNC:      LCALL     SETLCD
006F 90091B  103     MOV      DPTR,#STR9
0072 120383  104     LCALL     PRINT
0075 120290  105     LCALL     KEY

```

|             |     |             |              |
|-------------|-----|-------------|--------------|
| 0078 C0E0   | 106 | PUSH        | ACC          |
| 007A 1204A2 | 107 | LCALL       | OBEEP        |
| 007D D0E0   | 108 | POP         | ACC          |
| 007F B41B01 | 109 | CJNE        | A,#ESC,FUNC0 |
| 0082 22     | 110 | RET         |              |
|             | 111 |             |              |
| 0083 B43011 | 112 | FUNC0: CJNE | A,#30H,FUNC1 |
| 0086 1202FD | 113 | LCALL       | SETLCD       |
| 0089 12013A | 114 | LCALL       | ADDR         |
| 008C 1202FD | 115 | LCALL       | SETLCD       |
| 008F 120383 | 116 | LCALL       | PRINT        |
| 0092 120290 | 117 | LCALL       | KEY          |
| 0095 80D5   | 118 | SJMP        | FUNC         |
|             | 119 |             |              |
| 0097 B43116 | 120 | FUNC1: CJNE | A,#31H,FUNC2 |
| 009A 1202FD | 121 | LCALL       | SETLCD       |
| 009D 9009D6 | 122 | MOV         | DPTR,#STR7   |
| 00A0 120383 | 123 | LCALL       | PRINT        |
| 00A3 7820   | 124 | MOV         | R0,#20H      |
| 00A5 12036C | 125 | LCALL       | DELAY3       |
| 00A8 12013A | 126 | LCALL       | ADDR         |
| 00AB 1203D9 | 127 | LCALL       | SONG         |
| 00AE 80BC   | 128 | SJMP        | FUNC         |
|             | 129 |             |              |
| 00B0 B4320E | 130 | FUNC2: CJNE | A,#32H,FUNC3 |
| 00B3 1202FD | 131 | LCALL       | SETLCD       |
| 00B6 9009C5 | 132 | MOV         | DPTR,#STR8   |
| 00B9 120383 | 133 | LCALL       | PRINT        |
| 00BC 1203AD | 134 | LCALL       | OSCK         |
| 00BF 80AB   | 135 | SJMP        | FUNC         |
|             | 136 |             |              |
| 00C1 B43305 | 137 | FUNC3: CJNE | A,#33H,FUNC4 |
| 00C4 120462 | 138 | LCALL       | CLRD         |
| 00C7 80A3   | 139 | SJMP        | FUNC         |
|             | 140 |             |              |
| 00C9 B43405 | 141 | FUNC4: CJNE | A,#34H,FUNC5 |
| 00CC 12058C | 142 | LCALL       | MOVE         |
| 00CF 809B   | 143 | SJMP        | FUNC         |
|             | 144 |             |              |
| 00D1 B43513 | 145 | FUNC5: CJNE | A,#35H,FUNC6 |
| 00D4 1202FD | 146 | LCALL       | SETLCD       |
| 00D7 9008F9 | 147 | MOV         | DPTR,#STR17  |
| 00DA 120383 | 148 | LCALL       | PRINT        |
| 00DD 7820   | 149 | MOV         | R0,#20H      |
| 00DF 12036C | 150 | LCALL       | DELAY3       |
| 00E2 120630 | 151 | LCALL       | SERIAL       |
| 00E5 8085   | 152 | SJMP        | FUNC         |
|             | 153 |             |              |
| 00E7 B43617 | 154 | FUNC6: CJNE | A,#36H,FUNC7 |
| 00EA 1202FD | 155 | LCALL       | SETLCD       |
| 00ED 9008D7 | 156 | MOV         | DPTR,#STR18  |
| 00F0 120383 | 157 | LCALL       | PRINT        |
| 00F3 7820   | 158 | MOV         | R0,#20H      |
| 00F5 12036C | 159 | LCALL       | DELAY3       |
| 00F8 120670 | 160 | LCALL       | SERIALB      |

```

00FB 12068F 161          LCALL      SERI2
00FE 02006C 162          LJMP       FUNC
163
0101 B43706 164      FUNC7:  CJNE       A,#37H,FUNCF
165
0104 1206AB 166          LCALL      PGEN
0107 02006C 167          LJMP       FUNC
168
169
010A B4462A 170      FUNC7:  CJNE       A,#46H,FUNCE
010D 1202FD 171          LCALL      SETLCD
0110 900A58 172          MOV        DPTR,#STR30
0113 120383 173          LCALL      PRINT
0116 7820   174          MOV        R0,#20H
0118 12036C 175          LCALL      DELAY3
011B 1202FD 176          LCALL      SETLCD
011E 900A7A 177          MOV        DPTR,#STR31
0121 120383 178          LCALL      PRINT
0124 7820   179          MOV        R0,#20H
0126 12036C 180          LCALL      DELAY3
0129 1202FD 181          LCALL      SETLCD
012C 900A9C 182          MOV        DPTR,#STR32
012F 120383 183          LCALL      PRINT
0132 7820   184          MOV        R0,#20H
0134 12036C 185          LCALL      DELAY3
0137 02006C 186      FUNCE:  LJMP       FUNC
187
188 *****
189
190
191
192 ***** FUNCTION ADDRESS *****
193 ; ENTER ADDRESS
194 ; OUTPUT = DPTR
195 *****
196
013A 1202FD 197      ADDR:  LCALL      SETLCD
013D 900A09 198          MOV        DPTR,#STR2
0140 120383 199          LCALL      PRINT
0143 7C04   200          MOV        R4,#4
0145 7D86   201          MOV        R5,#86H
0147 120228 202          LCALL      SCANF
203
014A B41302 204          CJNE       A,#ENTER,ADRET
014D 8001   205          SJMP       ADDR2
014F 22     206      ADRET:  RET
0150 900846 207      ADDR2:  MOV        DPTR,#DPP1
0153 E570   208          MOV        A,KEYBUF
0155 C3     209          CLR        C
0156 9430   210          SUBB       A,#30H
0158 93     211          MOVC      A,@A+DPTR
0159 C4     212          SWAP      A
015A FE     213          MOV        R6,A
015B E571   214          MOV        A,KEYBUF+1
015D C3     215          CLR        C

```

|             |                            |             |                  |
|-------------|----------------------------|-------------|------------------|
| 015E 9430   | 216                        | SUBB        | A,#30H           |
| 0160 93     | 217                        | MOVC        | A,@A+DPTR        |
| 0161 4E     | 218                        | ORL         | A,R6             |
| 0162 FE     | 219                        | MOV         | R6,A             |
| 0163 E572   | 220                        | MOV         | A,KEYBUF+2       |
| 0165 C3     | 221                        | CLR         | C                |
| 0166 9430   | 222                        | SUBB        | A,#30H           |
| 0168 93     | 223                        | MOVC        | A,@A+DPTR        |
| 0169 C4     | 224                        | SWAP        | A                |
| 016A FF     | 225                        | MOV         | R7,A             |
| 016B E573   | 226                        | MOV         | A,KEYBUF+3       |
| 016D C3     | 227                        | CLR         | C                |
| 016E 9430   | 228                        | SUBB        | A,#30H           |
| 0170 93     | 229                        | MOVC        | A,@A+DPTR        |
| 0171 4F     | 230                        | ORL         | A,R7             |
| 0172 FF     | 231                        | MOV         | R7,A             |
| 0173 8E83   | 232                        | MOV         | DPH,R6           |
| 0175 8F82   | 233                        | MOV         | DPL,R7           |
| 0177 22     | 234                        | RET         |                  |
|             | 235                        |             |                  |
| 236         | ;***** INPUT ADDRESS ***** |             |                  |
| 237         | ; INPUT = DPTR             |             |                  |
| 238         | ; REG = A,R4,R5            |             |                  |
| 239         | ;*****                     |             |                  |
| 240         |                            |             |                  |
| 0178 C083   | 241                        | ADDR4: PUSH | DPH              |
| 017A C082   | 242                        | PUSH        | DPL              |
| 017C 1202FD | 243                        | LCALL       | SETLCD           |
| 017F 1201E8 | 244                        | LCALL       | DISPADDR         |
| 0182 E0     | 245                        | MOVX        | A,@DPTR          |
| 0183 C0E0   | 246                        | PUSH        | ACC              |
| 0185 C4     | 247                        | SWAP        | A                |
| 0186 540F   | 248                        | ANL         | A,#0FH           |
| 0188 900836 | 249                        | MOV         | DPTR,#DPP        |
| 018B 93     | 250                        | MOVC        | A,@A+DPTR        |
| 018C 120335 | 251                        | LCALL       | LCDWD            |
| 018F D0E0   | 252                        | POP         | ACC              |
| 0191 540F   | 253                        | ANL         | A,#0FH           |
| 0193 900836 | 254                        | MOV         | DPTR,#DPP        |
| 0196 93     | 255                        | MOVC        | A,@A+DPTR        |
| 0197 120335 | 256                        | LCALL       | LCDWD            |
| 019A 900A10 | 257                        | MOV         | DPTR,#STR3       |
| 019D 120383 | 258                        | LCALL       | PRINT            |
| 01A0 D082   | 259                        | POP         | DPL              |
| 01A2 D083   | 260                        | POP         | DPH              |
|             | 261                        |             |                  |
| 01A4 7D8C   | 262                        | MOV         | R5,#8CH          |
| 01A6 7C02   | 263                        | MOV         | R4,#2            |
| 01A8 120228 | 264                        | LCALL       | SCANF            |
| 01AB B4220F | 265                        | CJNE        | A,#DECRESE,ADDR5 |
| 01AE C3     | 266                        | CLR         | C                |
| 01AF E582   | 267                        | MOV         | A,DPL            |
| 01B1 9401   | 268                        | SUBB        | A,#1             |
| 01B3 F582   | 269                        | MOV         | DPL,A            |

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้ทางการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 อื่นๆ ที่มิได้ห้ามมิให้ดัดแปลงเอกสารและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|             |     |                             |                   |
|-------------|-----|-----------------------------|-------------------|
| 01B5 E583   | 270 | MOV                         | A,DPH             |
| 01B7 9400   | 271 | SUBB                        | A,#0              |
| 01B9 F583   | 272 | MOV                         | DPH,A             |
| 01BB 80BB   | 273 | SJMP                        | ADDR4             |
| 01BD B42303 | 274 | ADDR5: CJNE                 | A,#INCREASE,ADDR6 |
| 01C0 A3     | 275 | INC                         | DPTR              |
| 01C1 80B5   | 276 | SJMP                        | ADDR4             |
|             | 277 |                             |                   |
| 01C3 B4131E | 278 | ADDR6: CJNE                 | A,#ENTER,ADDR7    |
| 01C6 C083   | 279 | PUSH                        | DPH               |
| 01C8 C082   | 280 | PUSH                        | DPL               |
| 01CA 900846 | 281 | MOV                         | DPTR,#DPP1        |
| 01CD E570   | 282 | MOV                         | A,KEYBUF          |
| 01CF C3     | 283 | CLR                         | C                 |
| 01D0 9430   | 284 | SUBB                        | A,#30H            |
| 01D2 93     | 285 | MOVC                        | A,@A+DPTR         |
| 01D3 C4     | 286 | SWAP                        | A                 |
| 01D4 FE     | 287 | MOV                         | R6,A              |
| 01D5 E571   | 288 | MOV                         | A,KEYBUF+1        |
| 01D7 C3     | 289 | CLR                         | C                 |
| 01D8 9430   | 290 | SUBB                        | A,#30H            |
| 01DA 93     | 291 | MOVC                        | A,@A+DPTR         |
| 01DB 4E     | 292 | ORL                         | A,R6              |
| 01DC D082   | 293 | POP                         | DPL               |
| 01DE D083   | 294 | POP                         | DPH               |
| 01E0 F0     | 295 | MOVX                        | @DPTR,A           |
| 01E1 A3     | 296 | INC                         | DPTR              |
| 01E2 8094   | 297 | SJMP                        | ADDR4             |
| 01E4 B41B91 | 298 | ADDR7: CJNE                 | A,#ESC,ADDR4      |
| 01E7 22     | 299 | RET                         |                   |
|             | 300 |                             |                   |
|             | 301 | ***** DISPLAY ADDRESS ***** |                   |
|             | 302 | ; INPUT = DPTR              |                   |
|             | 303 | ; REG = A,R6,R7             |                   |
|             | 304 | *****                       |                   |
|             | 305 |                             |                   |
| 01E8 C083   | 306 | DISPADDR: PUSH              | DPH               |
| 01EA C082   | 307 | PUSH                        | DPL               |
| 01EC AE83   | 308 | MOV                         | R6,DPH            |
| 01EE AF82   | 309 | MOV                         | R7,DPL            |
| 01F0 1202FD | 310 | LCALL                       | SETLCD            |
| 01F3 EE     | 311 | MOV                         | A,R6              |
| 01F4 C4     | 312 | SWAP                        | A                 |
| 01F5 540F   | 313 | ANL                         | A,#0FH            |
| 01F7 900836 | 314 | MOV                         | DPTR,#DPP         |
| 01FA 93     | 315 | MOVC                        | A,@A+DPTR         |
| 01FB 120335 | 316 | LCALL                       | LCDWD             |
|             | 317 |                             |                   |
| 01FE EE     | 318 | MOV                         | A,R6              |
| 01FF 540F   | 319 | ANL                         | A,#0FH            |
| 0201 900836 | 320 | MOV                         | DPTR,#DPP         |
| 0204 93     | 321 | MOVC                        | A,@A+DPTR         |
| 0205 120335 | 322 | LCALL                       | LCDWD             |
|             | 323 |                             |                   |
| 0208 EF     | 324 | MOV                         | A,R7              |

```

0209 C4      325      SWAP      A
020A 540F    326      ANL        A,#0FH
020C 900836  327      MOV        DPTR,#DPP
020F 93      328      MOVC       A,@A+DPTR
0210 120335  329      LCALL       LCDWD
330
0213 EF      331      MOV        A,R7
0214 540F    332      ANL        A,#0FH
0216 900836  333      MOV        DPTR,#DPP
0219 93      334      MOVC       A,@A+DPTR
021A 120335  335      LCALL       LCDWD
021D 900A10  336      MOV        DPTR,#STR3
0220 120383  337      LCALL       PRINT
0223 D082    338      POP        DPL
0225 D083    339      POP        DPH
0227 22      340      RET
341
342 ***** SCANF FUNCTION *****
343 ;INPUT R4 = COUNT CHAR
344 ;      R5 = POSITION
345 ;OUTPUT = A
346 ;REG = A,R0,R1,R2,R3,R4,R5,DPTR
347 *****
348
0228 C000    349      SCANF:  PUSH      0
022A C002    350      PUSH      2
022C C004    351      PUSH      4
022E C005    352      PUSH      5
0230 7B00    353      MOV        R3,#0
0232 7870    354      MOV        R0,#KEYBUF
355
0234 120290  356      SCANF1:  LCALL     KEY
0237 B42202  357      CJNE      A,#DECRESE,SCANF4
023A 8031    358      SJMP      SCANFO
023C B42302  359      SCANF4:  CJNE      A,#INCRESE,SCANF5
023F 802C    360      SJMP      SCANFO
0241 B41B02  361      SCANF5:  CJNE      A,#ESC,SCANF6
0244 8027    362      SJMP      SCANFO
0246 B41417  363      SCANF6:  CJNE      A,#BACKSP,SCANF2 ; BACK
SPACE
0249 EB      364      MOV        A,R3
024A 60E8    365      JZ         SCANF1
024C 1B      366      DEC        R3
024D 18      367      DEC        R0
024E ED      368      MOV        A,R5
024F 2B      369      ADD        A,R3
0250 120312  370      LCALL     LCDWI
0253 7420    371      MOV        A,#SPACE
0255 120335  372      LCALL     LCDWD
0258 F6      373      MOV        @R0,A
0259 7410    374      MOV        A,#10H
025B 120312  375      LCALL     LCDWI
025E 80D4    376      SJMP      SCANF1
377
0260 FA      378      SCANF2:  MOV        R2,A

```

```

0261 EB      379      MOV      A,R3
0262 C3      380      CLR      C
0263 9C      381      SUBB     A,R4
0264 7010     382      JNZ      SCANF3
0266 EA      383      MOV      A,R2
0267 C3      384      CLR      C
0268 942F     385      SUBB     A,#2FH
026A 50C8     386      JNC      SCANF1
026C EA      387      MOV      A,R2
026D D005     388      SCANFO: POP      5
026F D004     389      POP      4
0271 D002     390      POP      2
0273 D000     391      POP      0
0275 22      392      RET
0276 EA      394      SCANF3: MOV      A,R2
0277 C3      395      CLR      C ;CHECK HI
0278 9447     396      SUBB     A,#47H
027A 50B8     397      JNC      SCANF1
027C EA      398      MOV      A,R2
027D C3      399      CLR      C ;CHECK LOW
027E 9430     400      SUBB     A,#30H
0280 40B2     401      JC       SCANF1
0282 ED      402      MOV      A,R5 ;SET CURSOR
0283 2B      403      ADD      A,R3
0284 120312   404      LCALL     LCDWI
0287 EA      405      MOV      A,R2
0288 F6      406      MOV      @R0,A
0289 120335   407      LCALL     LCDWD
028C 08      408      INC      R0
028D 0B      409      INC      R3
028E 80A4     410      SJMP     SCANF1
0290 C083     417      KEY:    PUSH     DPH ;SAVE REGISTER
0292 C082     418      PUSH     DPL
0294 C000     419      PUSH     0
0296 C001     420      PUSH     1
0298 C002     421      PUSH     2
029A 7906     422      KEYS:   MOV      R1,#6
029C 7A00     423      MOV      R2,#0
029E EA      424      KEY1:   MOV      A,R2 ;6 LOOP
029F 902001   425      MOV      DPTR,#PORTB
02A2 F0      426      MOVX     @DPTR,A
02A3 A3      427      INC      DPTR
02A4 E0      428      MOVX     A,@DPTR
02A5 540F     429      ANL      A,#0FH
02A7 B40F05   430      CJNE     A,#0FH,KEY2
02AA 0A      431      INC      R2
02AB D9F1     432      DJNZ     R1,KEY1
02AD 80EB     433      SJMP     KEYS

```

\*\*\*\*\* READ KEYBOARD \*\*\*\*\*

; INPUT : NONE

; OUTPUT : ACC

\*\*\*\*\*

\*\*\*\*\*

```

02AF 7800    434    KEY2:    MOV        R0,#0        ; ENCODE KEY
02B1 30E00C  435                JNB        0E0H,KEY3
02B4 7801    436                MOV        R0,#1
02B6 30E107  437                JNB        0E1H,KEY3
02B9 7802    438                MOV        R0,#2
02BB 30E202  439                JNB        0E2H,KEY3
02BE 7803    440                MOV        R0,#3
02C0 7404    441    KEY3:    MOV        A,#4
02C2 8AF0    442                MOV        B,R2
02C4 A4      443                MUL        AB
02C5 28      444                ADD        A,R0
02C6 12035B  445                LCALL     DELAY        ; DELAY
02C9 9002E5  446                MOV        DPTR,#DATAKEY
02CC 93      447                MOVC     A,@A+DPTR
02CD C0E0    448                PUSH     ACC        ; CHECK KEY PRESS
02CF 902002  449                MOV        DPTR,#PORTC
02D2 E0      450    KEYPRESS: MOVX     A,@DPTR
02D3 540F    451                ANL        A,#0FH
02D5 B40FFA  452                CJNE     A,#0FH,KEYPRESS
02D8 D0E0    453    KEYPRESS1: POP        ACC
02DA D002    454    KEYO:    POP        2        ; RESTORE REGISTER
02DC D001    455                POP        1
02DE D000    456                POP        0
02E0 D082    457                POP        DPL
02E2 D083    458                POP        DPH
02E4 22      459                RET        ; RETURN VALUE IN
ACC
02E5 30313233 461
02E9 34353637 462    DATAKEY: DB        30H,31H,32H,33H
02ED 38394142 463                DB        34H,35H,36H,37H
02F1 43444546 464                DB        38H,39H,41H,42H
02F5 201B1719 465                DB        43H,44H,45H,46H
02F9 14232213 466                DB        20H,1BH,17H,19H
02F9 14232213 467                DB        14H,23H,22H,13H
02F9 14232213 468
02F9 14232213 469    ***** LCD INITIAL *****
02F9 14232213 470    ; INPUT , OUTPUT : NONE
02F9 14232213 471    *****
02F9 14232213 472
02FD 7438    473    SETLCD:    MOV        A,#38H        ;FUNCTION SET
02FF 120312  474                LCALL     LCDWI
0302 740E    475                MOV        A,#0EH        ;DISPLAY ON/OFF
0304 120312  476                LCALL     LCDWI
0307 7401    477                MOV        A,# 01H        ;CLEAR
0309 120312  478                LCALL     LCDWI
030C 7481    479                MOV        A,#81H        ;CLEAR
030E 120312  480                LCALL     LCDWI
0311 22      481                RET
0311 22      482
0311 22      483    ***** LCDWI *****
0311 22      484    ; LCD WRITE INSTRUCTION (RS=0)
0311 22      485    ; IN = A
0311 22      486    ; REG = A,DPTR
0311 22      487    *****

```

```

488
0312 C083 489 LCDWI: PUSH DPH
0314 C082 490 PUSH DPL
0316 902000 491 MOV DPTR,#PORTA ;DATA TO PORTA
0319 F0 492 MOVX @DPTR,A
031A 902001 493 MOV DPTR,#PORTB ;PORTB READ
;MODIFY WRITE
031D E0 494 MOVX A,@DPTR
031E C2E3 495 CLR ACC.3 ;RS=0
0320 C2E4 496 CLR ACC.4 ;R/W=0
0322 C2E5 497 CLR ACC.5 ;ENABLE=0
0324 F0 498 MOVX @DPTR,A
0325 D2E5 499 SETB ACC.5 ;ENABLE=1
0327 F0 500 MOVX @DPTR,A
0328 C2E5 501 CLR ACC.5 ;ENABLE=0
032A F0 502 MOVX @DPTR,A
032B 7400 503 MOV A,#0 ;DELAY
032D 14 504 LCDWI1: DEC A
032E 70FD 505 JNZ LCDWI1
0330 D082 506 POP DPL
0332 D083 507 POP DPH
033A 22 508 RET
509
510 ;***** LCDWD *****
511 ;LCD WRITE DATA (RS=1)
512 ;IN = A
513 ;REG = A,DPTR
514 ;*****
515
0335 C083 516 LCDWD: PUSH DPH
0337 C082 517 PUSH DPL
0339 902000 518 MOV DPTR,#PORTA ;DATA TO PORTA
033C F0 519 MOVX @DPTR,A
033D 902001 520 MOV DPTR,#PORTB ;PORTB READ
;MODIFY WRITE
0340 E0 521 MOVX A,@DPTR
0341 D2E3 522 SETB ACC.3 ;RS=1
0343 C2E4 523 CLR ACC.4 ;R/W=0
0345 C2E5 524 CLR ACC.5 ;ENABLE=0
0347 F0 525 MOVX @DPTR,A
0348 D2E5 526 SETB ACC.5 ;ENABLE=1
034A F0 527 MOVX @DPTR,A
034B C2E5 528 CLR ACC.5 ;ENABLE=0
034D F0 529 MOVX @DPTR,A
034E 12035B 530 LCALL DELAY
0351 7400 531 MOV A,#0
0353 14 532 LCDWD1: DEC A
0354 70FD 533 JNZ LCDWD1
0356 D082 534 POP DPL
0358 D083 535 POP DPH
035A 22 536 RET
537
538 ;***** DELAY FUNCTION *****
539 ;INPUT : R0 ,FOR DELAY3
540 ;OUTPUT : NONE

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ผู้ใช้ต้องรับผิดชอบต่อการใช้งานเอกสารทุกครั้งที่มีการนำไปใช้

```

541 ;*****
542
035B C000 543   DELAY:   PUSH      0
035D C001 544           PUSH      1
035F 7832 545           MOV      R0,#50
0361 7900 546   DELAY1:  MOV      R1,#0
0363 D9FE 547   DELAY2:  DJNZ     R1,DELAY2
0365 D8FA 548           DJNZ     R0,DELAY1
0367 D001 549           POP      1
0369 D000 550           POP      0
036B 22   551           RET
552
036C C000 553   DELAY3:  PUSH      0
036E C001 554           PUSH      1
0370 C002 555           PUSH      2
0372 7950 556   DELAY4:  MOV      R1,#50H
0374 7A00 557   DELAY5:  MOV      R2,#0H
0376 DAFE 558   DELAY6:  DJNZ     R2,$
0378 D9FA 559           DJNZ     R1,DELAY5
037A D8F6 560           DJNZ     R0,DELAY4
037C D002 561           POP      2
037E D001 562           POP      1
0380 D000 563           POP      0
0382 22   564           RET
565
566 ;***** FUNCTION SHOW CHARACTOR TO LCD *****
567 ; INPUT = DPTR
568 ;*****
569
0383 C083 570   PRINT:   PUSH      DPH
0385 C082 571           PUSH      DPL
0387 7400 572   PRINT1:  MOV      A,#0
0389 93   573           MOVC     A,@A+DPTR
038A B40005 574           CJNE     A,#0,PRINT2
038D D082 575           POP      DPL
038F D083 576           POP      DPH
0391 22   577           RET
0392 B40D08 578   PRINT2:  CJNE     A,#13,PRINT3
0395 74C0 579           MOV      A,#0C0H
0397 120312 580           LCALL     LCDWI
039A A3   581           INC      DPTR
039B 80EA 582           SJMP     PRINT1
039D 120335 583   PRINT3:  LCALL     LCDWD
03A0 A3   584           INC      DPTR
03A1 80E4 585           SJMP     PRINT1
586
587 ;***** BEEP FUNCTION *****
588 ; IN = R2 FREQUENCY
589 ;   = R3 LENGTH
590 ;*****
591
03A3 1202FD 592   BEEP:    LCALL     SETLCD
03A6 9008A4 593           MOV      DPTR,#STR34
03A9 120431 594           LCALL     SOUND
03AC 22   595           RET

```

```

596
597 ;***** OSCK FUNCTION *****
598 ; OPEN/SHORT CHECK
599 ;*****
600
03AD 902001 601 OSCK: MOV DPTR,#PORTB ;DIGIT 0
03B0 E0 602 MOVX A,@DPTR
03B1 54F8 603 ANL A,#0F8H
03B3 F0 604 MOVX @DPTR,A
605
03B4 D297 606 OSCK1: SETB TOOLBT ;PULL UP
03B6 2097FB 607 JB TOOLBT ;OSCK1
03B9 1203C6 608 CALL OSCKS
03BC 80F6 609 JMP OSCK1
610
03BE 7C02 611 OSCK9: MOV R4,#2
03C0 1204A9 612 LCALL HBEEP
03C3 02002A 613 LJMP MON
614
03C6 902000 615 OSCKS: MOV DPTR,#PORTA ;DOT & SOUND
03C9 7480 616 MOV A,#80H
03CB F0 617 MOVX @DPTR,A
03CC 7A30 618 MOV R2,#30H
03CE 7B04 619 MOV R3,#4
03D0 120431 620 LCALL SOUND
03D3 902000 621 MOV DPTR,#PORTA
03D6 E4 622 CLR A
03D7 F0 623 MOVX @DPTR,A
03D8 22 624 RET
625
626 ;***** SONG SUB *****
627 ; PLAY NOTE BY SONG TABLE
628 ; IN = DPTR TABLE (NN,LL,NN,LL,...,NN,LL,FF)
629 ; NN =0-25H LL=0-4
630 ; OUT = DPTR (NEXT)
631 ; REG = A,R2,R3,R4,R5,DPTR
632 ;*****
633
03D9 E4 634 SONG: CLR A
03DA 93 635 MOVC A,@A+DPTR
03DB A3 636 INC DPTR
03DC B4FF01 637 CJNE A,#0FFH,SONG1
03DF 22 638 RET ;END BY 0FFH
639
03E0 FA 640 SONG1: MOV R2,A ;MEMORY FRE.
03E1 E4 641 CLR A
03E2 93 642 MOVC A,@A+DPTR ;LENGTH
03E3 FB 643 MOV R3,A
03E4 A3 644 INC DPTR
645
03E5 C083 646 PUSH DPH
03E7 C082 647 PUSH DPL
03E9 EA 648 MOV A,R2
03EA 900406 649 MOV DPTR,#SONGT1
03ED 93 650 MOVC A,@A+DPTR

```

```

03EE FA      651      MOV      R2,A      ;FREQUENCY
                                652
03EF EB      653      MOV      A,R3
03F0 90042C  654      MOV      DPTR,#SONGT2
03F3 93      655      MOV      A,@A+DPTR
03F4 FB      656      MOV      R3,A      ;LENGTH
                                657
03F5 120431  658      CALL     SOUND
03F8 7A40    659      MOV      R2,#40H      ;DELAY
03FA 7B00    660      SONG4:  MOV      R3,#0
03FC DBFE    661      DJNZ     R3,$
03FE DAFA    662      DJNZ     R2,SONG4
0400 D082    663      POP      DPL
0402 D083    664      POP      DPH
0404 80D3    665      JMP      SONG
                                666
0406 00F4E4D9 667      SONGT1: DB      000H,0F4H,0F4H,0D9H ;G,G#,A,A#
040A CCC0B4AA668      DB      0CCH,0C0H,0B4H,0AAH ;B,C,C#,D
040E A1989088 669      DB      0A1H,098H,090H,088H ;D#,E,F,F#
0412 8078726B 670      DB      080H,078H,072H,06BH ;G,G#,A,A#
0416 655F5A55 671      DB      065H,05FH,05AH,055H ;B,C,C#,D
041A 504C4743 672      DB      050H,04CH,047H,043H ;D#,E,F,F#
041E 3F3B3835 673      DB      03FH,03BH,038H,035H ;G,G#,A,A#
0422 322F2C2A 674      DB      032H,02FH,02CH,02AH ;B,C,C#,D
0426 27252321 675      DB      027H,025H,023H,021H ;D#,E,F,F#
042A 1F01      676      DB      01FH,001H ;G,S
042C 084080C0 677      SONGT2: DB      008H,040H,080H,0C0H ;0,1,2,3
0430 00      678      DB      000H ;4
                                679
0431 7D00    680      ;***** SOUND SUB *****
0433 7C80    681      ; SOUND GENERATOR
                                682      ; IN = R2 FREQUENCY
                                683      ; = R3 LENGTH
                                684      ; REG = A,R2,R3,R4,R5,DPTR
                                685      ;*****
                                686
0431 7D00    687      SOUND:  MOV      R5,#0      ;END FLAG
0433 7C80    688      MOV      R4,#80H      ;DELAY
                                689
0435 12043C  689      SOUND1:  CALL     SOUND2
0438 BD01FA  690      CJNE     R5,#1,SOUND1
043B 22      691      RET
                                692
043C 902001  693      SOUND2:  MOV      DPTR,#PORTB ;OUT 1 (PORTA
                                694      BIT7)
043F E0      694      MOVX     A,@DPTR
0440 D2E6    695      SETB     ACC.6
0442 F0      696      MOVX     @DPTR,A
0443 120451  697      CALL     SOUND3
0446 902001  698      MOV      DPTR,#PORTB ;OUT 0
0449 E0      699      MOVX     A,@DPTR
044A C2E6    700      CLR      ACC.6
044C F0      701      MOVX     @DPTR,A
044D 120451  702      CALL     SOUND3
0450 22      703      RET

```

```

0451 EA 704
0452 120459 705 SOUND3: MOV A,R2 ;FREQUENCY DELAY
0455 14 706 SOUND4: CALL SOUND5
0456 70FA 707 DEC A
0458 22 708 JNZ SOUND4
0459 DC06 709 RET
710
045B 7C80 711 SOUND5: DJNZ R4,SOUND6 ;LENGTH COUNT
045D DB02 712 DOWN
045F 7D01 713 MOV R4,#80H
0461 22 714 DJNZ R3,SOUND6
715 MOV R5,#1
716 SOUND6: RET
717 ;***** CLRD FUNCTION *****
718 ; CLEAR USER RAM
719 ; CLEAR FORM ADDRESS 8100H - F7FFH
720 ;*****
721
0462 12047A 722 CLRD: CALL CONF
0465 908100 723 MOV DPTR,#8100H ;CLEAR PROCESS
0468 7AF7 724 MOV R2,#0F7H
046A 7BFF 725 MOV R3,#0FFH
046C 74FF 726 CLRD4: MOV A,#0FFH
046E F0 727 MOVX @DPTR,A
046F 12056C 728 CALL DPCOM
0472 A3 729 INC DPTR
0473 70F7 730 JNZ CLRD4
0475 1204A9 731 CALL HBEEP
0478 012A 732 JMP MON
733
734 ;***** CONF SUB *****
735 ; CONFIRM DISPLAY & WAIT ENTER KEY
736 ;*****
737
047A 1204BD 738 CONF: CALL RBEEP
047D E4 739 CLR A
047E 1202FD 740 LCALL SETLCD
0481 9009A3 741 MOV DPTR,#STR11
0484 120383 742 LCALL PRINT
0487 5190 743 CONF1: CALL KEY
0489 C0E0 744 PUSH ACC
048B 1204F7 745 LCALL UBEEP
048E D0E0 746 POP ACC
0490 B413F4 747 CJNE A,#13H,CONF1
0493 22 748 RET
749
750 ;***** NBEEP SUB *****
751 ; ANY BEEP SUBROUTINE
752 ; REG = A,R2,R3,R4,R5,DPTR
753 ;*****
0494 7A50 754 FBEEP: MOV R2,#50H ;DIRECT FUNCTION KEY
0496 7B10 755 MOV R3,#10H

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|             |     |         |      |           |                |
|-------------|-----|---------|------|-----------|----------------|
| 0498 9131   | 757 |         | CALL | SOUND     |                |
| 049A 22     | 758 |         | RET  |           |                |
|             | 759 |         |      |           |                |
| 049B 7A40   | 760 | XBEEP:  | MOV  | R2,#40H   | ;HEX KEY BEEP  |
| 049D 7B10   | 761 |         | MOV  | R3,#10H   |                |
| 049F 9131   | 762 |         | CALL | SOUND     |                |
| 04A1 22     | 763 |         | RET  |           |                |
|             | 764 |         |      |           |                |
| 04A2 7A30   | 765 | OBEEP:  | MOV  | R2,#30H   | ;OPERATE KEY   |
| BEEP        |     |         |      |           |                |
| 04A4 7B10   | 766 |         | MOV  | R3,#10H   |                |
| 04A6 9131   | 767 |         | CALL | SOUND     |                |
| 04A8 22     | 768 |         | RET  |           |                |
|             | 769 |         |      |           |                |
| 04A9 7A20   | 770 | HBEEP:  | MOV  | R2,#20H   | ;SUCCESS BEEP  |
| 04AB 7B40   | 771 |         | MOV  | R3,#40H   |                |
| 04AD 9131   | 772 |         | CALL | SOUND     |                |
| 04AF 7A10   | 773 |         | MOV  | R2,#10H   |                |
| 04B1 7B20   | 774 |         | MOV  | R3,#20H   |                |
| 04B3 9131   | 775 |         | CALL | SOUND     |                |
| 04B5 22     | 776 |         | RET  |           |                |
|             | 777 |         |      |           |                |
| 04B6 7A90   | 778 | LBEEP:  | MOV  | R2,#90H   | ;WARNING BEEP  |
| 04B8 7B90   | 779 |         | MOV  | R3,#90H   |                |
| 04BA 9131   | 780 |         | CALL | SOUND     |                |
| 04BC 22     | 781 |         | RET  |           |                |
|             | 782 |         |      |           |                |
| 04BD 7A04   | 783 | RBEEP:  | MOV  | R2,#4     | ;READY BEEP    |
| 04BF EA     | 784 | RBEEP1: | MOV  | A,R2      |                |
| 04C0 C0E0   | 785 |         | PUSH | ACC       |                |
| 04C2 7A70   | 786 |         | MOV  | R2,#70H   |                |
| 04C4 7B15   | 787 |         | MOV  | R3,#15H   |                |
| 04C6 9131   | 788 |         | CALL | SOUND     |                |
| 04C8 7A00   | 789 |         | MOV  | R2,#00H   |                |
| 04CA 7B18   | 790 |         | MOV  | R3,#18H   |                |
| 04CC 12057C | 791 |         | CALL | DMSEC     |                |
| 04CF D0E0   | 792 |         | POP  | ACC       |                |
| 04D1 FA     | 793 |         | MOV  | R2,A      |                |
| 04D2 DAEB   | 794 |         | DJNZ | R2,RBEEP1 |                |
| 04D4 22     | 795 |         | RET  |           |                |
|             | 796 |         |      |           |                |
| 04D5 7A90   | 797 | EBEEP:  | MOV  | R2,#90H   | ;ERROR BEEP    |
| 04D7 7B00   | 798 |         | MOV  | R3,#00H   |                |
| 04D9 9131   | 799 |         | CALL | SOUND     |                |
| 04DB 7AC0   | 800 |         | MOV  | R2,#0C0H  |                |
| 04DD 7B00   | 801 |         | MOV  | R3,#00H   |                |
| 04DF 9131   | 802 |         | CALL | SOUND     |                |
| 04E1 22     | 803 |         | RET  |           |                |
|             | 804 |         |      |           |                |
| 04E2 7E30   | 805 | PBEEP:  | MOV  | R6,#30H   | ;POWER-UP BEEP |
| 04E4 7F38   | 806 |         | MOV  | R7,#38H   |                |
| 04E6 EF     | 807 | PBEEP2: | MOV  | A,R7      |                |
| 04E7 FA     | 808 |         | MOV  | R2,A      |                |
| 04E8 7B02   | 809 |         | MOV  | R3,#02H   |                |
| 04EA 9131   | 810 |         | CALL | SOUND     |                |

```

04EC 7A00      811      MOV      R2,#00H
04EE 7B08      812      MOV      R3,#08H
04F0 12057C    813      CALL     DMSEC
04F3 1F        814      DEC      R7
04F4 DEF0      815      DJNZ     R6,PBEEP2
04F6 22        816      RET
04F7 7A15      817
04F7 7A15      818      UBEEP:   MOV      R2,#15H      ;USER BEEP
04F9 7B70      819      MOV      R3,#70H
04FB 9131      820      CALL     SOUND
04FD 22        821      RET
04FE 8E83      822
0500 8F82      823      ;***** MOVES SUB *****
0502 12056C    824      ;MOVE DATA MEMORY
0505 403D      825      ;IN = R2,R3 START
0507 603B      826      ;    = R4,R5 END
0509 8C83      827      ;    = R6,R7 DESTINATION
050B 8D82      828      ;REG = A,R2,R3,R4,R5,R6,R7,DPTR
050D 1205B9    829      ;*****
0510 EF        830
0511 2582      831      MOVES:   MOV      DPH,R6
0513 FF        832      MOV      DPL,R7
0514 EE        833      CALL     DPCOM
0515 3583      834      JC      MOVES4
0517 FE        835      JZ      MOVES4
0518 8C83      836
051A 8D82      837      MOV      DPH,R4      ;*** START < DEST ****
051C E0        838      MOV      DPL,R5      ;MOVE ADDRESS DOWN
051D 8E83      839      CALL     DPSUB      ;DPTR=END-START
051F 8F82      840      MOV      A,R7      ;R6,R7=R6,R7+DPTR
0521 F0        841      ADD     A,DPL
0522 8C83      842      MOV      R7,A
0524 8D82      843      MOV      A,R6
0526 12056C    844      ADDC     A,DPH
0529 7001      845      MOV      R6,A
052B 22        846
052C 8C83      847      MOVES2:  MOV      DPH,R4
052E 8D82      848      MOV      DPL,R5
0530 1205AF    849      MOVX     A,@DPTR
0533 AC83      850      MOV      DPH,R6
0535 AD82      851      MOV      DPL,R7
0537 8E83      852      MOVX     @DPTR,A
0538 8C83      853
0539 8D82      854      MOV      DPH,R4
053A 12056C    855      MOV      DPL,R5
053B 7001      856      CALL     DPCOM
053C 22        857      JNZ     MOVES21
053D 8C83      858      RET      ;FINISH
053E 8D82      859
053F 1205AF    860      MOVES21: MOV      DPH,R4
0540 8C83      861      MOV      DPL,R5
0541 8D82      862      CALL     DPDEC
0542 AC83      863      MOV      R4,DPH
0543 AD82      864      MOV      R5,DPL
0544 8E83      865      MOV      DPH,R6

```

```

0539 8F82      866      MOV      DPL,R7
053B 1205AF    867      CALL     DPDEC
053E AE83      868      MOV      R6,DPH
0540 AF82      869      MOV      R7,DPL
0542 80D4      870      JMP      MOVES2
               871
0544 8A83      872      MOVES4:  MOV      DPH,R2 ; START => DEST
0546 8B82      873      MOV      DPL,R3 ;MOVE ADDRESS UP
               874
0548 E0        875      MOVX     A,@DPTR
0549 8E83      876      MOV      DPH,R6
054B 8F82      877      MOV      DPL,R7
054D F0        878      MOVX     @DPTR,A
054E 8C83      879      MOV      DPH,R4
0550 8D82      880      MOV      DPL,R5
0552 12056C    881      CALL     DPCOM
0555 7001      882      JNZ      MOVES41
0557 22        883      RET              ;FINISH
               884
0558 8A83      885      MOVES41: MOV      DPH,R2 ;NEXT ADDRESS
055A 8B82      886      MOV      DPL,R3
055C A3        887      INC      DPTR
055D AA83      888      MOV      R2,DPH
055F AB82      889      MOV      R3,DPL
0561 8E83      890      MOV      DPH,R6
0563 8F82      891      MOV      DPL,R7
0565 A3        892      INC      DPTR
0566 AE83      893      MOV      R6,DPH
0568 AF82      894      MOV      R7,DPL
056A 80D8      895      JMP      MOVES4
               896
056C C082      904      DPCOM:  PUSH     DPL
056E C3        905      CLR      C
056F E582      906      MOV      A,DPL
0571 9B        907      SUBB     A,R3
0572 F582      908      MOV      DPL,A
0574 E583      909      MOV      A,DPH
0576 9A        910      SUBB     A,R2
0577 4582      911      ORL      A,DPL
0579 D082      912      POP      DPL
057B 22        913      RET
               914
057C 22        915      ;***** DMSEC SUB *****
057D 22        916      ; DELAY 1/1000 SECOND
057E 22        917      ; IN = R2,R3
057F 22        918      ; REG = A,R2,R3,R4
0580 22        919      ;*****
0581 22        920

```

```

057C 7CE6    921    DMSEC:    MOV        R4,#230        ;1 MSEC LOOP
057E 00      922    DMSEC1:   NOP
057F 00      923                NOP
0580 DCFC    924                DJNZ        R4,DMSEC1
0582 DBF8    925                DJNZ        R3,DMSEC
0584 EA      926                MOV        A,R2
0585 B40001  927                CJNE       A,#0,DMSEC2
0588 22      928                RET
0589 1A      929    DMSEC2:   DEC        R2
058A 80F0    930                SJMP       DMSEC
931
932 ;***** MOVE FUNCTION *****
933 ; MOVE BLOCK ON DATA MEMORY
934 ;*****
935
058C 1205C5  936    MOVE:    LCALL    INPSED
937
058F 90FF00  938                MOV        DPTR,#STA
0592 E0      939                MOVX       A,@DPTR
0593 FA      940                MOV        R2,A
0594 A3      941                INC        DPTR
0595 E0      942                MOVX       A,@DPTR
0596 FB      943                MOV        R3,A
944
0597 90FF02  945                MOV        DPTR,#ENDA
059A E0      946                MOVX       A,@DPTR
059B FC      947                MOV        R4,A
059C A3      948                INC        DPTR
059D E0      949                MOVX       A,@DPTR
059E FD      950                MOV        R5,A
951
059F 90FF04  952                MOV        DPTR,#DEAD
05A2 E0      953                MOVX       A,@DPTR
05A3 FE      954                MOV        R6,A
05A4 A3      955                INC        DPTR
05A5 E0      956                MOVX       A,@DPTR
05A6 FF      957                MOV        R7,A
958
05A7 1204FE  959                LCALL    MOVES
05AA 1204A9  960                LCALL    HBEEP
05AD 012A    961                JMP        MON
962
963 ;***** DPDEC SUB *****
964 ; DPTR = DPTR - 1
965 ; IN    = DPTR
966 ; OUT   = DPTR
967 ;      = A (ZERO FLAG)
968 ; REG   = NO
969 ;*****
970
05AF C582    971    DPDEC:    XCH        A,DPL
05B1 7002    972                JNZ        $+4
05B3 1583    973                DEC        DPH
05B5 14      974                DEC        A
05B6 C582    975                XCH        A,DPL

```

```

05B8 22      976      RET
977
978 ; ***** DPSUB SUB *****
979 ; DPTR = DPTR - R2,R3
980 ; IN = DPTR,R2,R3
981 ; OUT = DPTR
982 ; REG = A,DPTR
983
05B9 C3      984      DPSUB:      CLR      C
05BA E582    985      MOV      A,DPL
05BC 9B      986      SUBB     A,R3
05BD F582    987      MOV      DPL,A
05BF E583    988      MOV      A,DPH
05C1 9A      989      SUBB     A,R2
05C2 F583    990      MOV      DPH,A
05C4 22      991      RET
992
993 ; ***** INPSED SUB *****
994 ; INPUT START,END,DEST
995 ; IN = STTADD ENDADD DESADD
996 ; OUT = STTADD ENDADD DESADD
997 ; REG = ALL
998 ; *****
999
05C5 7400    1000     INPSED:      MOV      A,#00H
1001
05C7 1202FD  1002     LCALL     SETLCD
05CA 90094E  1003     MOV      DPTR,#STR20
05CD 120383  1004     LCALL     PRINT
05D0 7810    1005     MOV      R0,#10H
05D2 12036C  1006     LCALL     DELAY3
1007
05D5 1202FD  1008     LCALL     SETLCD
05D8 900970  1009     MOV      DPTR,#STR23
05DB 120383  1010     LCALL     PRINT
05DE 7810    1011     MOV      R0,#10H
05E0 12036C  1012     LCALL     DELAY3
05E3 12013A  1013     LCALL     ADDR
1014
05E6 8582F0  1015     MOV      B,DPL
05E9 E583    1016     MOV      A,DPH
05EB 90FF00  1017     MOV      DPTR,#STA
05EE F0      1018     MOVX     @DPTR,A
05EF A3      1019     INC      DPTR
05F0 E5F0    1020     MOV      A,B
05F2 F0      1021     MOVX     @DPTR,A
1022
05F3 1202FD  1023     LCALL     SETLCD
05F6 900981  1024     MOV      DPTR,#STR21
05F9 120383  1025     LCALL     PRINT
05FC 7810    1026     MOV      R0,#10H
05FE 12036C  1027     LCALL     DELAY3
0601 12013A  1028     LCALL     ADDR
1029
0604 8582F0  1030     MOV      B,DPL

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
 ไม่ควรแก้ไขหรือทำซ้ำโดยไม่ได้รับอนุญาต  
 หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง

```

0607 E583      1031      MOV      A,DPH
0609 90FF02    1032      MOV      DPTR,#ENDA
060C F0        1033      MOVX     @DPTR,A
060D A3        1034      INC      DPTR
060E E5F0      1035      MOV      A,B
0610 F0        1036      MOVX     @DPTR,A
                                1037
0611 1202FD    1038      LCALL     SETLCD
0614 900992    1039      MOV      DPTR,#STR22
0617 120383    1040      LCALL     PRINT
061A 7810      1041      MOV      R0,#10H
061C 12036C    1042      LCALL     DELAY3
061F 12013A    1043      LCALL     ADDR
                                1044
0622 8582F0    1045      MOV      B,DPL
0625 E583      1046      MOV      A,DPH
0627 90FF04    1047      MOV      DPTR,#DEAD
062A F0        1048      MOVX     @DPTR,A
062B A3        1049      INC      DPTR
062C E5F0      1050      MOV      A,B
062E F0        1051      MOVX     @DPTR,A
                                1052
062F 22        1053      RET
                                1054
                                1055 ;***** SERIAL PORT *****
                                1056 ; INPUT = SERIAL PORT
                                1057 ;      = DPTR
                                1058 ; OUT  = A
                                1059 ;*****
                                1060
0630 909000    1061      SERIAL:  MOV      DPTR,#SERIADD
0633 759852    1062      MOV      SCON,#52H
0636 758920    1063      MOV      TMOD,#20H
0639 758DFD    1064      MOV      TH1,#0FDH
063C D28E      1065      SETB     TR1
                                1066
063E 3098FD    1067      SERIAL1: JNB      RI,SERIAL1
0641 C298      1068      CLR      RI
0643 E599      1069      MOV      A,SBUF
0645 F0        1070      MOVX     @DPTR,A
0646 F4        1071      CPL      A
0647 F590      1072      MOV      P1,A
0649 F4        1073      CPL      A
064A A3        1074      INC      DPTR
064B B40DF0    1075      CJNE     A,#0DH,SERIAL1
064E 7820      1076      MOV      R0,#20H
0650 12036C    1077      LCALL     DELAY3
                                1078
0653 909000    1079      SERIAL2:  MOV      DPTR,#SERIADD
0656 E0        1080      SERIAL3:  MOVX     A,@DPTR
0657 F4        1081      CPL      A
0658 F590      1082      MOV      P1,A
065A F4        1083      CPL      A
065B A3        1084      INC      DPTR
065C 7820      1085      MOV      R0,#20H

```

```

065E 716C      1086      ACALL      DELAY3
0660 B40DF3    1087      CJNE       A,#0DH,SERIAL3
0663 7590FF    1088      MOV        P1,#0FFH
0666 1204BD    1089      LCALL      RBEEP
0669 1204BD    1090      LCALL      RBEEP
066C 02002A    1091      LJMP       MON
               1092
066F 22        1093      RET
               1094
0670 909000    1095      ;***** TEST SERIAL PORT *****
0673 759852    1096      ; INPUT = SERIAL PORT
0676 758920    1097      ;          = DPTR
0679 758DFD    1098      ; OUT  = A
067C D28E      1099      ;*****
               1100
067E 3098FD    1101      SERIALB:  MOV        DPTR,#SERIADD
0681 C298      1102      MOV        SCON,#52H
0683 E599      1103      MOV        TMOD,#20H
0685 F0        1104      MOV        TH1,#0FDH
0686 F590      1105      SETB       TR1
0688 A3        1106
0689 B400F2    1107      SERI1:   JNB        RI,SERI1
068C 909000    1108      CLR        RI
068F E0        1109      MOV        A,SBUF
0690 F4        1110      MOVX       @DPTR,A
0691 F590      1111      MOV        P1,A
0693 A3        1112      INC        DPTR
0694 715B      1113      CJNE       A,#00H,SERI1
0696 715B      1114
0698 B4FFF4    1115      SERI3:   MOV        DPTR,#SERIADD
069B 80EF      1116      SERI2:   MOVX       A,@DPTR
069D 9006AA    1117      CPL        A
06A0 C082      1118      MOV        P1,A
06A2 C083      1119      INC        DPTR
06A4 12013A    1120      ACALL      DELAY
06A7 7400      1121      ACALL      DELAY
06A9 73        1122      CJNE       A,#0FFH,SERI2
06AA 22        1123      SJMP       SERI3
06AD 73        1124
06AE 73        1125      ;***** RUN USER PROGRAM *****
06AF 73        1126      ; INPUT = NONE
06B0 73        1127      ; REG  = A,DPTR
06B1 73        1128      ;*****
06B2 73        1129
06B3 73        1130      RUN:     MOV        DPTR,#RUNP
06B4 73        1131      PUSH       DPL
06B5 73        1132      PUSH       DPH
06B6 73        1133      LCALL      ADDR
06B7 73        1134      MOV        A,#0H
06B8 73        1135      JMP        @A+DPTR
06B9 73        1136      RUNP:   RET
06BA 73        1137
06BB 73        1138      ;***** PULSE GEN FUNCTION *****
06BC 73        1139      ; INPUT = R6,R7
06BD 73        1140      ;

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ที่มีการนำเอกสารนี้ไปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|             |        |          |                 |
|-------------|--------|----------|-----------------|
|             | 1141 ; |          |                 |
| 06AB 1202FD | 1142   | PGEN:    | LCALL SETLCD    |
| 06AE 90086E | 1143   |          | DPTR,#STR16     |
| 06B1 120383 | 1144   |          | PRINT           |
| 06B4 7820   | 1145   |          | MOV R0,#20H     |
| 06B6 12036C | 1146   |          | LCALL DELAY3    |
| 06B9 1202FD | 1147   |          | LCALL SETLCD    |
| 06BC 900890 | 1148   |          | MOV DPTR,#STR15 |
| 06BF 120383 | 1149   |          | LCALL PRINT     |
| 06C2 7820   | 1150   |          | MOV R0,#20H     |
|             | 1151   |          |                 |
| 06C4 7C02   | 1152   |          | MOV R4,#2H      |
| 06C6 7D89   | 1153   |          | MOV R5,#89H     |
| 06C8 120228 | 1154   |          | LCALL SCANF     |
| 06CB 120700 | 1155   |          | CALL CONDATA    |
| 06CE FE     | 1156   |          | MOV R6,A        |
|             | 1157   |          |                 |
| 06CF 1202FD | 1158   |          | LCALL SETLCD    |
| 06D2 90089A | 1159   |          | MOV DPTR,#STR14 |
| 06D5 120383 | 1160   |          | LCALL PRINT     |
| 06D8 7C02   | 1161   |          | MOV R4,#2H      |
| 06DA 7D89   | 1162   |          | MOV R5,#89H     |
| 06DC 120228 | 1163   |          | LCALL SCANF     |
| 6DF 120700  | 1164   |          | CALL CONDATA    |
| 06E2 FF     | 1165   |          | MOV R7,A        |
| 06E3 1202FD | 1166   |          | LCALL SETLCD    |
| 06E6 90086E | 1167   |          | MOV DPTR,#STR16 |
| 06E9 120383 | 1168   |          | LCALL PRINT     |
|             | 1169   |          |                 |
|             | 1170   |          |                 |
| 06EC 758900 | 1171   | PGEN2:   | MOV TMOD,#00H   |
| 06EF 8E8C   | 1172   |          | MOV TH0,R6 ;FE  |
| 06F1 8F8A   | 1173   |          | MOV TL0,R7 ;EA  |
| 06F3 D28C   | 1174   |          | SETB TR0        |
| 06F5 308DFD | 1175   | PGEN1:   | JNB TF0,PGEN1   |
| 06F8 C28C   | 1176   |          | CLR TR0         |
| 06FA C28D   | 1177   |          | CLR TF0         |
| 06FC B297   | 1178   |          | CPL P1.7        |
| 06FE 80EC   | 1179   |          | SJMP PGEN2      |
|             | 1180   |          |                 |
|             | 1181   |          |                 |
| 0700 900846 | 1182   | CONDATA: | MOV DPTR,#DPP1  |
| 0703 E570   | 1183   |          | MOV A,KEYBUF    |
| 0705 C3     | 1184   |          | CLR C           |
| 0706 9430   | 1185   |          | SUBB A,#30H     |
| 0708 93     | 1186   |          | MOVC A,@A+DPTR  |
| 0709 C4     | 1187   |          | SWAP A          |
| 070A FE     | 1188   |          | MOV R6,A        |
| 070B E571   | 1189   |          | MOV A,KEYBUF+1  |
| 070D C3     | 1190   |          | CLR C           |
| 070E 9430   | 1191   |          | SUBB A,#30H     |
| 0710 93     | 1192   |          | MOVC A,@A+DPTR  |
| 0711 4E     | 1193   |          | ORL A,R6        |
| 0712 22     | 1194   |          | RET             |
|             | 1195   |          |                 |

1196 ; \*\*\*\*\* RECV FUNCTION \*\*\*\*\*

1197 ; RECEIVE INTEL-HEX FILE

1198 ; INPUT = SERIAL PORT

1199

|             |      |         |       |               |                 |
|-------------|------|---------|-------|---------------|-----------------|
| 0713 909000 | 1200 | RECV:   | MOV   | DPTR,#SERIADD |                 |
| 0716 759852 | 1201 |         | MOV   | SCON,#52H     |                 |
| 0719 758920 | 1202 |         | MOV   | TMOD,#20H     |                 |
| 071C 758DFD | 1203 |         | MOV   | TH1,#0FDH     |                 |
| 071F D28E   | 1204 |         | SETB  | TR1           |                 |
|             | 1205 |         |       |               |                 |
| 0721 C298   | 1206 |         | CLR   | RI            |                 |
| 0723 91BD   | 1207 |         | CALL  | RBEEP         |                 |
| 0725 7400   | 1208 |         | MOV   | A,#0H         |                 |
| 0727 90F800 | 1209 |         | MOV   | DPTR,#MEMADD  |                 |
| 072A F0     | 1210 |         | MOVX  | @DPTR,A       |                 |
| 072B A3     | 1211 |         | INC   | DPTR          |                 |
| 072C F0     | 1212 |         | MOVX  | @DPTR,A       |                 |
|             | 1213 |         |       |               |                 |
| 072D C276   | 1214 |         | CLR   | STMFAG        |                 |
| 072F 209705 | 1215 | RECV0:  | JB    | P1.7,RECV01   |                 |
| 0732 1204A9 | 1216 |         | LCALL | HBEEP         | ;BREAK TO       |
| 0735 012A   | 1217 |         | JMP   | MON           |                 |
| 0737 3098F5 | 1218 | RECV01: | JNB   | RI,RECV0      |                 |
| 073A 1207F3 | 1219 |         | LCALL | RBYTE         |                 |
| 073D B40D03 | 1220 |         | CJNE  | A,#0DH,\$+6   | ;DON'T CARE 0DH |
| 0740 02072F | 1221 |         | LJMP  | RECV0         |                 |
| 0743 B40A03 | 1222 |         | CJNE  | A,#0AH,\$+6   | ;DON'T          |
| CARE 0AH    |      |         |       |               |                 |
| 0746 02072F | 1223 |         | LJMP  | RECV0         |                 |
| 0749 B44F03 | 1224 |         | CJNE  | A,#'O',\$+6   |                 |
| 074C 0207DD | 1225 |         | LJMP  | RECVO         |                 |
| 074F B45303 | 1226 |         | CJNE  | A,#'S',\$+6   |                 |
| 0752 0207DB | 1227 |         | LJMP  | RECVS         |                 |
| 0755 B43A03 | 1228 |         | CJNE  | A,#'',\$+6    |                 |
| 0758 020764 | 1229 |         | LJMP  | RECV11        |                 |
| 075B 0207C4 | 1230 |         | JMP   | RECVE         |                 |
|             | 1231 |         |       |               |                 |
| 075E 1207F3 | 1232 | RECV1:  | LCALL | RBYTE         |                 |
| 0761 B43AFA | 1233 |         | CJNE  | A,#' ',RECV1  |                 |
| 0764 7F00   | 1234 | RECV11: | MOV   | R7,#0         | ;CLEAR CHECKSUM |
| 0766 120806 | 1235 |         | LCALL | RBYTEHC       | ;BYTE COUNT     |
| 0769 FD     | 1236 |         | MOV   | R5,A          | ;COUNTER        |
| 076A 120806 | 1237 |         | LCALL | RBYTEHC       |                 |
| 076D F583   | 1238 |         | MOV   | DPH,A         | ;ADDRESS HIGH   |
| 076F 120806 | 1239 |         | LCALL | RBYTEHC       |                 |
| 0772 F582   | 1240 |         | MOV   | DPL,A         | ;ADDRESS LOW    |
| 0774 120806 | 1241 |         | LCALL | RBYTEHC       |                 |
| 0777 B40026 | 1242 |         | CJNE  | A,#00H,RECV8  | ;END OF FILE    |
|             | 1243 |         |       |               |                 |
| 077A 120806 | 1244 | RECV2:  | CALL  | RBYTEHC       | ;DATA LOOP      |
| 077D F0     | 1245 |         | MOVX  | @DPTR,A       |                 |
| 077E C083   | 1246 |         | PUSH  | DPH           |                 |
| 0780 C082   | 1247 |         | PUSH  | DPL           |                 |
| 0782 902000 | 1248 |         | MOV   | DPTR,#PORTA   |                 |
| 0785 F0     | 1249 |         | MOVX  | @DPTR,A       |                 |

|             |      |               |                |                 |
|-------------|------|---------------|----------------|-----------------|
| 0786 D082   | 1250 | POP           | DPL            |                 |
| 0788 D083   | 1251 | POP           | DPH            |                 |
| 078A A3     | 1252 | INC           | DPTR           |                 |
| 078B DDED   | 1253 | DJNZ          | R5,RECV2       |                 |
| 078D EF     | 1254 | MOV           | A,R7           | ;CHECKSUM       |
| 078E F4     | 1255 | CPL           | A              |                 |
| 078F 04     | 1256 | INC           | A              |                 |
| 0790 F5F0   | 1257 | MOV           | B,A            |                 |
| 0792 120806 | 1258 | LCALL         | RBYTEHC        |                 |
| 0795 B5F02C | 1259 | CJNE          | A,B,RECVE      |                 |
| 0798 1207F3 | 1260 | LCALL         | RBYTE          | ;CR             |
| 079B 1207F3 | 1261 | LCALL         | RBYTE          | ;LF             |
| 079E 80BE   | 1262 | JMP           | RECV1          |                 |
|             | 1263 |               |                |                 |
| 07A0 EF     | 1264 | RECV8: MOV    | A,R7           | ;END OF FILE    |
| 07A1 F4     | 1265 | CPL           | A              |                 |
| 07A2 04     | 1266 | INC           | A              |                 |
| 07A3 F5F0   | 1267 | MOV           | B,A            |                 |
| 07A5 1207FB | 1268 | LCALL         | RBYTEH         |                 |
| 07A8 B5F019 | 1269 | CJNE          | A,B,RECVE      |                 |
| 07AB 1207F3 | 1270 | LCALL         | RBYTE          | ;CR             |
| 07AE 1207F3 | 1271 | LCALL         | RBYTE          | ;LF             |
| 07B1 1202FD | 1272 | LCALL         | SETLCD         |                 |
| 07B4 900AE0 | 1273 | MOV           | DPTR,#STR35    |                 |
| 07B7 120383 | 1274 | LCALL         | PRINT          |                 |
| 07BA 7820   | 1275 | MOV           | R0,#20H        |                 |
| 07BC 12036C | 1276 | LCALL         | DELAY3         |                 |
| 07BF 1204A9 | 1277 | LCALL         | HBEEP          |                 |
| 07C2 012A   | 1278 | JMP           | MON            |                 |
|             | 1279 |               |                |                 |
| 07C4 207804 | 1280 | RECVE: JB     | REMFAG,RECVEX  |                 |
| 07C7 91D5   | 1281 | CALL          | EBEEP          | ;ERROR          |
| 07C9 012A   | 1282 | JMP           | MON            |                 |
| 07CB 1202FD | 1283 | RECVEX: LCALL | SETLCD         |                 |
| 07CE 90085D | 1284 | MOV           | DPTR,#LOADE    |                 |
| 07D1 120383 | 1285 | LCALL         | PRINT          |                 |
| 07D4 7820   | 1286 | MOV           | R0,#20H        |                 |
| 07D6 12036C | 1287 | LCALL         | DELAY3         |                 |
| 07D9 012A   | 1288 | JMP           | MON            |                 |
|             | 1289 |               |                |                 |
|             | 1290 |               |                |                 |
| 07DB D276   | 1291 | RECVS: SETB   | STMFAG         | ;STREAM RECEIVE |
| 07DD 1207FB | 1292 | RECVO: CALL   | RBYTEH         | ;OFFSET RECEIVE |
| 07E0 90F800 | 1293 | MOV           | DPTR,#MEMADD   |                 |
| 07E3 E0     | 1294 | MOVX          | A,@DPTR        | ;ADDRESS HIGH   |
| 07E4 1207FB | 1295 | CALL          | RBYTEH         |                 |
| 07E7 90F801 | 1296 | MOV           | DPTR,#MEMADD+1 |                 |
| 07EA E0     | 1297 | MOVX          | A,@DPTR        | ;ADDRESS LOW    |
| 07EB 1207F3 | 1298 | CALL          | RBYTE          | ;CR             |
| 07EE 1207F3 | 1299 | CALL          | RBYTE          | ;LF             |
| 07F1 E12F   | 1300 | JMP           | RECV0          |                 |
|             | 1301 |               |                |                 |
|             | 1302 |               |                |                 |
|             | 1303 |               | RBYTE SUB      |                 |
|             | 1304 |               | OUT = A        |                 |

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ผู้จัดทำขอสงวนสิทธิ์ในข้อมูลและข้อมูลอ้างอิงที่มีอยู่ในเอกสารชุดนี้ซึ่งอาจจะมีการเปลี่ยนแปลงได้โดยไม่ต้องแจ้งให้ทราบล่วงหน้า

```

1305 ; REG = A
1306
07F3 3098FD 1307 RBYTE: JNB RI,$ ;WAIT FOR RECEIVE OK
07F6 C298 1308 CLR RI
07F8 E599 1309 MOV A,SBUF
07FA 22 1310 RET
1311
1312 ; ***** RBYTEH SUB *****
1313 ; READ HEX-ASCII FROM SERIAL (2 BYTE)
1314 ; OUT = A
1315 ; REG = A,R2,R3
1316
07FB F1F3 1317 RBYTEH: CALL RBYTE
07FD FA 1318 MOV R2,A
07FE 1207F3 1319 CALL RBYTE
0801 FB 1320 MOV R3,A
0802 12080D 1321 CALL ATOH
0805 22 1322 RET
1323
0806 1207FB 1324 RBYTEHC: CALL RBYTEH ;R7=CHECKSUM
0809 CF 1325 XCH A,R7
080A 2F 1326 ADD A,R7
080B CF 1327 XCH A,R7
080C 22 1328 RET
1329
1330
1331 ; ***** ATOH SUB *****
1332 ; ASCII TO HEX CONVERT
1333 ; IN = R2,R3 30H,41H
1334 ; OUT = A 0AH
1335 ; REG = A,R2
1336
080D EA 1337 ATOH: MOV A,R2
080E 120819 1338 CALL ATOHS
0811 C4 1339 SWAP A
0812 FA 1340 MOV R2,A
0813 EB 1341 MOV A,R3
0814 120819 1342 CALL ATOHS
0817 4A 1343 ORL A,R2
0818 22 1344 RET
1345
0819 B44100 1346 ATOHS: CJNE A,#'A',$+3
081C 4002 1347 JC ATOHS1
081E 2409 1348 ADD A,#9
0820 540F 1349 ATOHS1: ANL A,#0FH
0822 22 1350 RET
1351
1352 ; ***** SBYTE SUB *****
1353 ; SEND BYTE
1354 ; IN = A
1355 ; REG = NO
1356
0823 3099FD 1357 SBYTE: JNB TI,$ ;WAIT FOR SEND OK
0826 C299 1358 CLR TI
0828 F599 1359 MOV SBUF,A

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ในทางใดๆ ทั้งนี้หากมีข้อผิดพลาดใดๆ กรุณาแจ้งให้ทราบและต้องขอโทษล่วงหน้าและต้องขอโทษอีกครั้งที่นำเอกสารนี้มาใช้

```

082A 22      1360      RET
1361
1362 ; ***** DPADD SUB *****
1363 ; DPTR = DPTR + R2,R3
1364 ; IN = DPTR,R2,R3
1365 ; OUT = DPTR
1366 ; REG = A,DPTR
1367
082B E582    1368      DPADD:      MOV      A,DPL
082D 2B      1369      ADD      A,R3
082E F582    1370      MOV      DPL,A
0830 E583    1371      MOV      A,DPH
0832 3A      1372      ADDC     A,R2
0833 F583    1373      MOV      DPH,A
0835 22      1374      RET
1375
1376
1377
1378 ;*****
1379
0836 30313233 1380      DPP:      DB      '0123456789ABCDEF'
083A 34353637
083E 38394142
0842 43444546
0846 00010203 1381      DPP1:     DB      0,1,2,3,4,5,6,7,8,9,0,0,0,0,0,0,0AH,
                                0BH, 0CH,0DH, 0EH, 0FH
084A 04050607
084E 08090000
0852 00000000
0856 000A0B0C
085A 0D0E0F
1382
085D 20444F57 1383      LOADE:     DB      'DOWNLOAD ERROR ',0
0861 4C4F4144
0865 20455252
0869 4F522020
086D 00
086E 47454E45 1384      STR16:     DB      'GENERATOR PULSE ',13
0872 5241544F
0876 52205055
087A 4C534520
087E 0D
087F 20205351 1385      DB      ' SQUARE WAVE ',0
0883 55415245
0887 20574156
088B 45202020
088F 00
0890 52454720 1386      STR15:     DB      'REG R6 : ',0
0894 5236203A
0898 2000
089A 52454720 1387      STR14:     DB      'REG R7 : ',0
089E 5237203A
08A2 2000
08A4 2046554E 1388      STR34:     DB      'FUNCTION BEEP ',0
08A8 4354494F

```

|               |      |        |    |                       |
|---------------|------|--------|----|-----------------------|
| 08AC 4E204245 |      |        |    |                       |
| 08B0 45502020 |      |        |    |                       |
| 08B4 00       |      |        |    |                       |
| 08B5 20454449 | 1389 | STR19: | DB | ' EDIT DATA FROM ',13 |
| 08B9 54204441 |      |        |    |                       |
| 08BD 54412046 |      |        |    |                       |
| 08C1 524F4D20 |      |        |    |                       |
| 08C5 0D       |      |        |    |                       |
| 08C6 20202020 | 1390 |        | DB | ' KEYBOARD ',0        |
| 08CA 4B455942 |      |        |    |                       |
| 08CE 4F415244 |      |        |    |                       |
| 08D2 20202020 |      |        |    |                       |
| 08D6 00       |      |        |    |                       |
| 08D7 20202054 | 1391 | STR18: | DB | ' TEST SERIAL ',13    |
| 08DB 45535420 |      |        |    |                       |
| 08DF 53455249 |      |        |    |                       |
| 08E3 414C2020 |      |        |    |                       |
| 08E7 0D       |      |        |    |                       |
| 08E8 20202020 | 1392 |        | DB | ' PORT ',0            |
| 08EC 2020504F |      |        |    |                       |
| 08F0 52542020 |      |        |    |                       |
| 08F4 20202020 |      |        |    |                       |
| 08F8 00       |      |        |    |                       |
| 08F9 20205245 | 1393 | STR17: | DB | ' RECEIVE ASCII ',13  |
| 08FD 43454956 |      |        |    |                       |
| 0901 45204153 |      |        |    |                       |
| 0905 43494920 |      |        |    |                       |
| 0909 0D       |      |        |    |                       |
| 090A 20204652 | 1394 |        | DB | ' FROM KEYBOARD ',0   |
| 090E 4F4D204B |      |        |    |                       |
| 0912 4559424F |      |        |    |                       |
| 0916 41524420 |      |        |    |                       |
| 091A 00       |      |        |    |                       |
| 091B 53454C45 | 1395 | STR9:  | DB | 'SELECT FUNCTION ',0  |
| 091F 43542046 |      |        |    |                       |
| 0923 554E4354 |      |        |    |                       |
| 0927 494F4E20 |      |        |    |                       |
| 092B 00       |      |        |    |                       |
| 092C 20434C45 | 1396 | STR10: | DB | ' CLEAR INTERNAL ',13 |
| 0930 41522049 |      |        |    |                       |
| 0934 4E544552 |      |        |    |                       |
| 0938 4E414C20 |      |        |    |                       |
| 093C 0D       |      |        |    |                       |
| 093D 20202020 | 1397 |        | DB | ' REGISTER ',0        |
| 0941 52454749 |      |        |    |                       |
| 0945 53544552 |      |        |    |                       |
| 0949 20202020 |      |        |    |                       |
| 094D 00       |      |        |    |                       |
| 094E 2046554E | 1398 | STR20: | DB | ' FUNCTION MOVE ',13  |
| 0952 4354494F |      |        |    |                       |
| 0956 4E204D4F |      |        |    |                       |
| 095A 56452020 |      |        |    |                       |
| 095E 0D       |      |        |    |                       |
| 095F 20202020 | 1399 |        | DB | ' BLOCK ',0           |
| 0963 20424C4F |      |        |    |                       |

0967 434B2020  
096B 20202020  
096F 00  
0970 20535441 1400 STR23: DB 'START ADDRESS ',0  
0974 52542041  
0978 44445245  
097C 53532020  
0980 00  
0981 20202045 1401 STR21: DB 'END ADDRESS ',0  
0985 4E442041  
0989 44445245  
098D 53532020  
0991 00  
0992 44455354 1402 STR22: DB 'DESTINATION ADDR',0  
0996 494E4154  
099A 494F4E20  
099E 41444452  
09A2 00  
09A3 20434C45 1403 STR11: DB 'CLEAR DATA MEM ',13  
09A7 41522044  
09AB 41544120  
09AF 4D454D20  
09B3 0D  
09B4 20205052 1404 DB 'PRESS ENTER ',0  
09B8 45535320  
09BC 20454E54  
09C0 45522020  
09C4 00  
09C5 2053484F 1405 STR8: DB 'SHORT & OPEN ',0  
09C9 52542026  
09CD 204F5045  
09D1 4E202020  
09D5 00  
09D6 2046554E 1406 STR7: DB 'FUNCTION SONG ',0  
09DA 4354494F  
09DE 4E20534F  
09E2 4E472020  
09E6 00  
09E7 4D494352 1407 STR1: DB 'MICRO CONTROLLER',13  
09EB 4F20434F  
09EF 4E54524F  
09F3 4C4C4552  
09F7 0D  
09F8 20202020 1408 DB ' MCS - 51 ',0  
09FC 4D435320  
0A00 2D203531  
0A04 20202020  
0A08 00  
0A09 41444420 1409 STR2: DB 'ADD : ',0  
0A0D 3A2000  
0A10 203A2000 1410 STR3: DB ': ',0  
0A14 20205369 1411 STR4: DB 'Single Board ',13  
0A18 6E676C65  
0A1C 20426F61  
0A20 72642020

เอกสารนี้เป็นเอกสารที่ทางมหาวิทยาลัยสงขลานครินทร์ให้บริการใช้เพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
โดยไม่ได้รับอนุญาตทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|                    |        |    |                       |
|--------------------|--------|----|-----------------------|
| 0A24 0D            |        |    |                       |
| 0A25 20205665 1412 |        | DB | ' Version 1.00 ',0    |
| 0A29 7273696F      |        |    |                       |
| 0A2D 6E20312E      |        |    |                       |
| 0A31 30302020      |        |    |                       |
| 0A35 00            |        |    |                       |
| 0A36 45442E45 1413 | STR5:  | DB | 'ED.Engineer #19 ',13 |
| 0A3A 6E67696E      |        |    |                       |
| 0A3E 65657220      |        |    |                       |
| 0A42 23313920      |        |    |                       |
| 0A46 0D            |        |    |                       |
| 0A47 20456C65 1414 |        | DB | ' Elec & Com 2/2 ',0  |
| 0A4B 63202620      |        |    |                       |
| 0A4F 436F6D20      |        |    |                       |
| 0A53 322F3220      |        |    |                       |
| 0A57 00            |        |    |                       |
| 0A58 20205052 1415 | STR30: | DB | ' PRODUCER BY. ',13   |
| 0A5C 4F445543      |        |    |                       |
| 0A60 45522042      |        |    |                       |
| 0A64 592E2020      |        |    |                       |
| 0A68 0D            |        |    |                       |
| 0A69 20205448 1416 |        | DB | ' THITIPHONG P. ',0   |
| 0A6D 49544950      |        |    |                       |
| 0A71 484F4E47      |        |    |                       |
| 0A75 20502E20      |        |    |                       |
| 0A79 00            |        |    |                       |
| 0A7A 56415449 1417 | STR31: | DB | 'VATIN TIPPANONK',13  |
| 0A7E 4E202054      |        |    |                       |
| 0A82 49505041      |        |    |                       |
| 0A86 4E4F4E4B      |        |    |                       |
| 0A8A 0D            |        |    |                       |
| 0A8B 53415241 1418 |        | DB | 'SARAVUT SUKHETAN',0  |
| 0A8F 56555420      |        |    |                       |
| 0A93 53554B48      |        |    |                       |
| 0A97 4554414E      |        |    |                       |
| 0A9B 00            |        |    |                       |
| 0A9C 53554542 1419 | STR32: | DB | 'SUEBSAK SUNIPHAN',13 |
| 0AA0 53414B20      |        |    |                       |
| 0AA4 53554E49      |        |    |                       |
| 0AA8 5048414E      |        |    |                       |
| 0AAC 0D            |        |    |                       |
| 0AAD 5448414E 1420 |        | DB | 'THANKS FOR USER!',0  |
| 0AB1 4B532046      |        |    |                       |
| 0AB5 4F522055      |        |    |                       |
| 0AB9 53455221      |        |    |                       |
| 0ABD 00            |        |    |                       |
| 0ABE 20444F57 1421 | STR33: | DB | ' DOWNLOAD INTEL ',13 |
| 0AC2 4C4F4144      |        |    |                       |
| 0AC6 20494E54      |        |    |                       |
| 0ACA 454C2020      |        |    |                       |
| 0ACE 0D            |        |    |                       |
| 0ACF 20202020 1422 |        | DB | 'HEX FILE ',0         |
| 0AD3 48455820      |        |    |                       |
| 0AD7 46494C45      |        |    |                       |
| 0ADB 20202020      |        |    |                       |

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ไม่สามารถนำออกนอกมหาวิทยาลัยได้โดยไม่ได้รับอนุญาต  
 หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง  
 เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษานี้เท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
 หรือการอื่นใดได้  
 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|                    |        |     |                    |
|--------------------|--------|-----|--------------------|
| 0ADF 00            |        |     |                    |
| 0AE0 20444F57 1423 | STR35: | DB  | 'DOWNLOAD FILE',13 |
| 0AE4 4C4F4144      |        |     |                    |
| 0AE8 20204649      |        |     |                    |
| 0AEC 4C452020      |        |     |                    |
| 0AF0 0D            |        |     |                    |
| 0AF1 20202020 1424 |        | DB  | ' COMPLETE ',0     |
| 0AF5 434F4D50      |        |     |                    |
| 0AF9 4C455445      |        |     |                    |
| 0AFD 20202020      |        |     |                    |
| 0B01 00            |        |     |                    |
|                    | 1425   |     |                    |
| 0000=              | 1426   | END |                    |



## บรรณานุกรม

ปรเมษฐ์ ประณยานันท์ และ ปิยพงศ์ เผ่าวนิช, คู่มือประยุกต์การใช้งานไมโครคอนโทรลเลอร์

MCS-51 กรุงเทพมหานคร: พิมพ์ที่ บริษัท เอช.เอ็น กรุป, 2536

สุนทร วิฑูรพจน์, การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 กรุงเทพมหานคร: พิมพ์ที่

บริษัท เอช เอ็น กรุป, 2537

สุเจตน์ จันทรัมย์, “ไมโครคอนโทรลเลอร์ซีพเดียว 8051” กรุงเทพมหานคร:

วิทยาลัยมหานคร, 2535

รัชชัย อินทุโส และ ไตรภพ อินทุโส, ไมโครคอนโทรลเลอร์ 8051 กรุงเทพมหานคร: พิมพ์ที่

ฟิสิกส์เซ็นเตอร์, 2536

สมยศ จุณณะปิยะ, การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 กรุงเทพมหานคร:

พิมพ์ที่ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2536

พิพัฒน์ เลหาสงคราม, ไมโครคอนโทรลเลอร์ MCS-48 MCS-51 กรุงเทพมหานคร: พิมพ์ที่

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์

นายฐิติพงษ์ พิสิฐนฤดม

วันเดือนปีเกิด

1 เมษายน พ.ศ. 2517

สถานที่เกิด

จังหวัด ยะลา

ภูมิลำเนาเดิม

41 ถนน ประชาณุกุล อำเภอเมือง

ที่อยู่ปัจจุบัน

จังหวัด ยะลา

54 ถนน คูหามุข อำเภอเมือง

โทรศัพท์

จังหวัด ยะลา

073 - 215552

## ประวัติการศึกษา

ประถมศึกษา

โรงเรียนนิบงชนูปถัมภ์

มัธยมศึกษาตอนต้น โรงเรียน

โรงเรียนคณะราษฎรบำรุง

ประกาศนียบัตรวิชาชีพ (ปวช.)

วิทยาลัยเทคนิคยะลา

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

วิทยาลัยเทคนิคยะลา

ปริญญาตรี

สาขาวิชา อิเล็กทรอนิกส์ และ

คอมพิวเตอร์

คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับรางวัล

-

ทุนการศึกษา

-

คติพจน์

ความพยายามอยู่ที่ไหน ความสำเร็จอยู่-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ที่นั่น  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร

นายวาทีน ทิพนงค์

วันเดือนปีเกิด

25 สิงหาคม พ.ศ. 2519

สถานที่เกิด

จังหวัด สุรินทร์

ภูมิลำเนาเดิม

48 หมู่ 10 ตำบลเมืองบัว

ที่อยู่ปัจจุบัน

อำเภอชุมพลบุรี จังหวัด สุรินทร์

48 หมู่ 10 ตำบลเมืองบัว

อำเภอชุมพลบุรี จังหวัด สุรินทร์

โทรศัพท์

## ประวัติการศึกษา

ประถมศึกษา

โรงเรียนบ้านยางบ่ออี

มัธยมศึกษาตอนต้น โรงเรียน

โรงเรียนพยุคนกภูมิวิทยาการ

ประกาศนียบัตรวิชาชีพ (ปวช.)

วิทยาลัยเทคนิคบุรีรัมย์

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

สถาบันเทคโนโลยีราชมงคลวิทยาเขต-

ภาคตะวันออกเฉียงเหนือ นครราชสีมา

ปริญญาตรี

สาขาวิชา อิเล็กทรอนิกส์ และ

คอมพิวเตอร์

คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับรางวัล

ทุนการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 คติพจน์ ทำวันนี้ให้ดีที่สุด  
 ไม่มีใครได้ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์

นายศราวุธ สุชีฐาน

วันเดือนปีเกิด

9 กุมภาพันธ์ พ.ศ. 2520

สถานที่เกิด

จังหวัด ชลบุรี

ภูมิลำเนาเดิม

0574/2 หมู่ 2 ตำบลสัตหีบ

ที่อยู่ปัจจุบัน

อำเภอสัตหีบ จังหวัด ชลบุรี

58/62 หมู่ 9 ตำบลสัตหีบ

อำเภอสัตหีบ จังหวัด ชลบุรี

โทรศัพท์

038-436665 , 038-736517

## ประวัติการศึกษา

ประถมศึกษา

โรงเรียนสัตหีบ 2

มัธยมศึกษาตอนต้นโรงเรียน

โรงเรียนสิงห์สมุทร

ประกาศนียบัตรวิชาชีพ (ปวช.)

วิทยาลัยเทคนิคสัตหีบ

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

วิทยาลัยเทคนิคสัตหีบ

ปริญญาตรี

สาขาวิชา อิเล็กทรอนิกส์ และ

คอมพิวเตอร์

คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับรางวัล

-

ทุนการศึกษา

-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 คติพจน์ ทุกปัญหามีทางแก้ไข  
 ไม่วาการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์

นายสืบศักดิ์ สุนิพันธ์

วันเดือนปีเกิด

8 ตุลาคม พ.ศ. 2519

สถานที่เกิด

จังหวัด ชลบุรี

ภูมิลำเนาเดิม

42/102 หมู่ 9 ตำบลบางพระ

อำเภอศรีราชา จังหวัด ชลบุรี

ที่อยู่ปัจจุบัน

35/3 หมู่ 1 ตำบลสุรศักดิ์

อำเภอศรีราชา จังหวัด ชลบุรี

โทรศัพท์

038-324598 , 1500-747341

## ประวัติการศึกษา

ประถมศึกษา

โรงเรียนอัสสัมชัญศรีราชา

มัธยมศึกษาตอนต้นโรงเรียน

โรงเรียนอัสสัมชัญศรีราชา

ประกาศนียบัตรวิชาชีพ (ปวช.)

วิทยาลัยเทคนิคสัตหีบ

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

วิทยาลัยเทคนิคสัตหีบ

ปริญญาตรี

สาขาวิชา อิเล็กทรอนิกส์ และ

คอมพิวเตอร์

คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับรางวัล

-

ทุนการศึกษา

-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 กติพนัน  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
 ไม่เคยรอใคร