



มหาวิทยาลัย
ราชภัฏนครปฐม



บทเรียนออนไลน์

รายวิชา 6562213 ไมโครโปรเซสเซอร์ 1
6562213 Microprocessor 1

โดย

ผู้ช่วยศาสตราจารย์ วีระ กาญจนสินธุ์
วท.ม.(ฟิสิกส์) วท.บ.(ฟิสิกส์)
คณะวิทยาศาสตร์และเทคโนโลยี

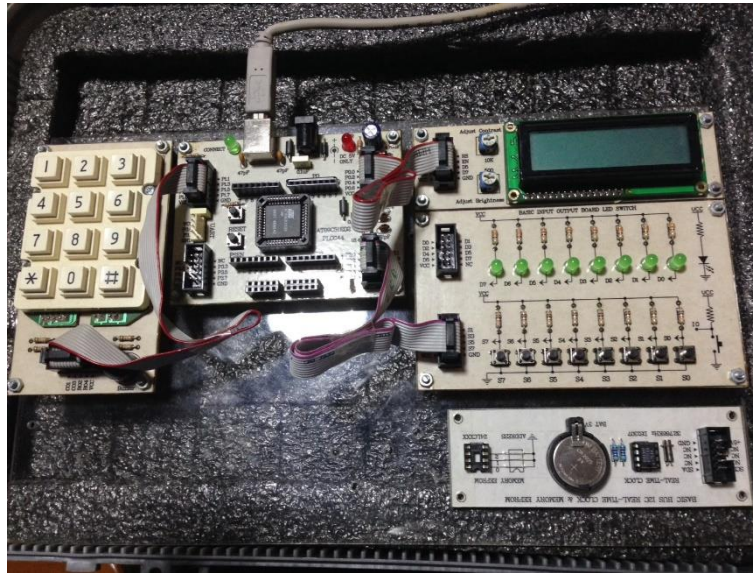


จุดประสงค์การเรียนรู้ของรายวิชา 6562213 ไมโครโพรเซสเซอร์ 1



2) เพื่อให้ผู้เรียนสามารถ
พัฒนาซอฟต์แวร์ภาษาซี
ในการควบคุมการทำงานของ
ฮาร์ดแวร์ที่ต้องวงจรร่วมกับ
ชิปตระกูล MCS-51

1) เพื่อให้ผู้เรียนสามารถ
ออกแบบและพัฒนาฮาร์ดแวร์
ที่มีไมโครคอนโทรลเลอร์ หรือ
ชิปตระกูล MCS-51
เป็นหน่วยประมวลผล



3) เพื่อให้ผู้เรียนสามารถ
พัฒนาเจตคติและมีจรรยาบรรณ
ในการคิดค้นและพัฒนา
สิ่งประดิษฐ์ โดยใช้ชิปตระกูล
MCS-51 เป็นหน่วยประมวลผล





- **บทที่ 1** สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน
- **บทที่ 2** โครงสร้างภายในของไมโครคอนโทรลเลอร์
- **บทที่ 3** ชุดคำสั่งแยกตามประเภทการใช้งาน
- **บทที่ 4** ตัวอย่างการเขียนโปรแกรมเบื้องต้นและโปรแกรมใช้งาน
- **บทที่ 5** การออกแบบวงจรหน่วยความจำและอุปกรณ์ต่อร่วม
- **บทที่ 6** การประยุกต์ใช้งานไมโครคอนโทรลเลอร์



มหาวิทยาลัย
ราชภัฏนครปฐม



รายวิชา 6562213 ไมโครโพรเซสเซอร์ 1

บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

โดยผู้ช่วยศาสตราจารย์ธีระ กาญจนสินธุ์

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

บทนำ



ในบทที่ 6 นี้ จะศึกษาเกี่ยวกับงานควบคุมอุปกรณ์ที่เชื่อมต่อกับ MCS-51 โดยเริ่มจากงานควบคุมอุปกรณ์ที่เชื่อมต่อกับ MCS-51 แบบ Byte-control และแบบ Bit-control ได้แก่ การหมุนดีซีมอเตอร์แบบตามเข็มนาฬิกา / งานควบคุมการติด-ดับของดวงไฟ LED ตั้งแต่ 1 ดวงถึง 8 ดวง(ซึ่งนำไปสู่การจำลองไฟจราจร) / งานเกี่ยวกับทางเลือก (condition) ที่เน้นการใช้ If-Condition และ Case-condition ในการพัฒนาโปรแกรมควบคุมที่มีทั้งอินพุตและเอาต์พุต โดยอินพุตมีทั้งสวิตช์แบบ Button switches และแบบ Matrix (3x4) switches รวมถึงเอาต์พุตมีตั้งแต่ หลอด LED -7 segment display – DC Motor และ Stepper motor/ นอกจากนี้ ในตอนท้ายเป็นตัวอย่างการขัดจังหวะการทำงานของ MCS-51 ด้วย Interrupt Routine Service เป็นตัวอย่างการขัดจังหวะงานหลักด้วยการกดสวิตช์ภายนอกที่ต่อกับขา Int.0(P3.2) และขา Int.1(P3.3)/



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

หัวข้อสำคัญในบทที่ 6



6.1 งานควบคุม
ดีซีมอเตอร์



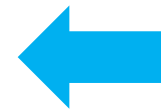
6.2 งานควบคุม
การติด/ดับของดวงไฟ



6.3 งานเกี่ยวกับ
ทางเลือก (condition)



6.4 Matrix switch



6.5 Matrix switch
Vs
8-LEDs & 7-segment
display



6.6 ตัวอย่างโปรแกรม
เกี่ยวกับการขัดจังหวะ
(Interrupt)



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ก) งานควบคุมดีซีมอเตอร์ทั่วไป

นิยาม

motor

Brushed DC motor

Brushless DC motor



motor

- อุปกรณ์เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล มีทั้งแบบ AC motor และแบบ DC motor
- ส่วนมาก DC motor ที่นำมาใช้งานควบคุมได้แก่ DC motor และ stepper motor
- รูปแบบการควบคุม เป็นการควบคุมทิศทาง (direction) และความเร็ว (speed)

DC Motor Types

Brushed DC



Advantages

- Cheapest and simplest motor
- Speed linear to applied voltage
- Simple motor control

Disadvantages

- High maintenance
- Low life-span (due to physical wear on brushes)

Brushless DC



Advantages

- High efficiency
- Little to no maintenance
- Long life span
- High output power per frame size

Disadvantages

- More complicated motor control
- Large initial costs

Stepper



Advantages

- Accurate position control
- Excellent low speed torque
- Long life

Disadvantages

- Low efficiency
- Prone to resonances, noise, and torque ripple
- Cannot accelerate loads rapidly



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ก) งานควบคุมดีซีมอเตอร์ทั่วไป

นิยาม

motor

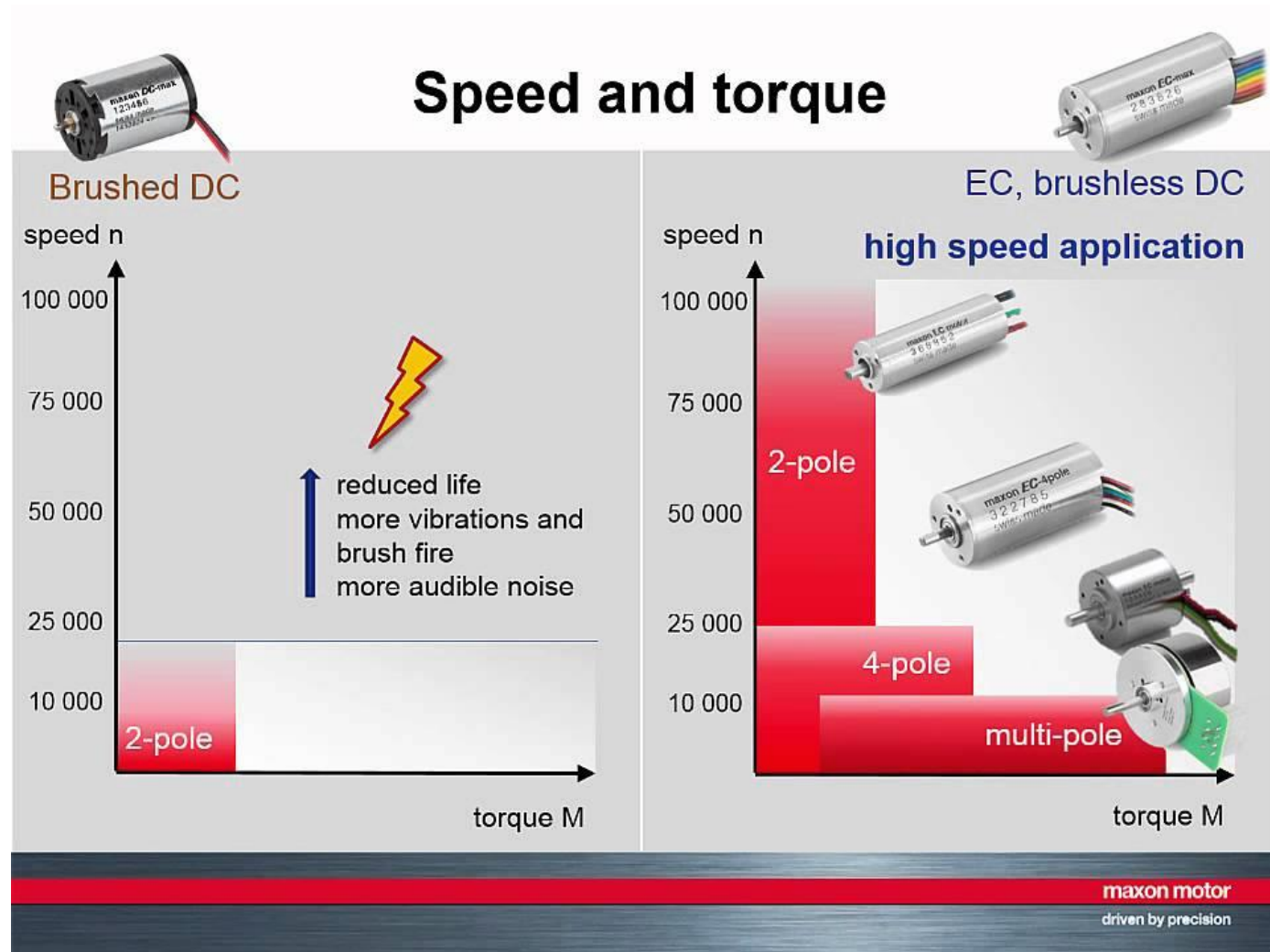
Brushed DC motor

Brushless DC motor



DC motor

- Brushed DC มีประสิทธิภาพต่ำกว่า brushless
- Brushed DC มีแรงบิด(torque)ต่ำกว่า brushless
- Brushed DC ใช้พลังงานไฟฟ้าต่ำกว่า brushless
- Brushed DC ราคาต่ำกว่า brushless





บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

4.6 (ก) ตัวอย่างที่ 5 : 1-button switch & 1-LED แบบ CA(control byte)

ตัวอย่างที่ใช้เงื่อนไข (if) สองทางเลือก

สวิตช์เปิด/ปิด ดวงไฟ

วงจรสวิตช์ใช้ port P1.7

วงจรดวงไฟใช้ port P2.4

ผังความคิด (flowchart)

เริ่ม

สวิตช์พร้อมกด

ดวงไฟพร้อมใช้

กดสวิตช์ ?

true

เปิดไฟ

end

false

รูปแบบโปรแกรม

If(statement is real) { do process }

```
#include<reg51.h>
```

```
void main ()
```

```
{
```

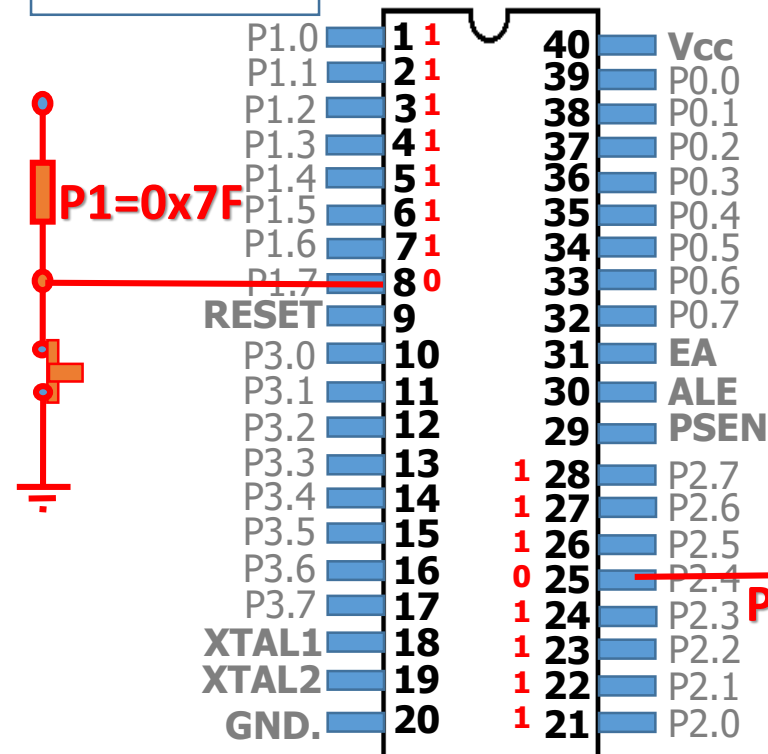
```
P1=0xFF;P2=0xFF;
while(1)
```

```
{ if(P1==0x7F)
```

```
{P2=0xEF;}
```

```
}
```

```
}
```





บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

4.6 (ก) ตัวอย่างที่ 5 : 1-button switch & 1-LED แบบ CA(control byte)

ตัวอย่างที่ใช้เงื่อนไข (if) สองทางเลือก

สวิตช์เปิด/ปิด ดวงไฟ

วงจรสวิตช์ใช้ port P1.7

วงจรดวงไฟใช้ port P2.4

ผังความคิด (flowchart)

เริ่ม

สวิตช์พร้อมกด

ดวงไฟพร้อมใช้

กดสวิตช์ ?

true

เปิดไฟ

false

ปิดไฟ

end

รูปแบบโปรแกรม

If(statement is real) { do process }

```
#include<reg51.h>
```

```
void main ()
```

```
{
```

```
P1=0xFF;P2=0xFF;
while(1)
```

```
{ if(P1==0x7F)
```

```
{P2=0xEF;}
```

```
else {P2=0xFF;}
```

```
}
```

```
}
```



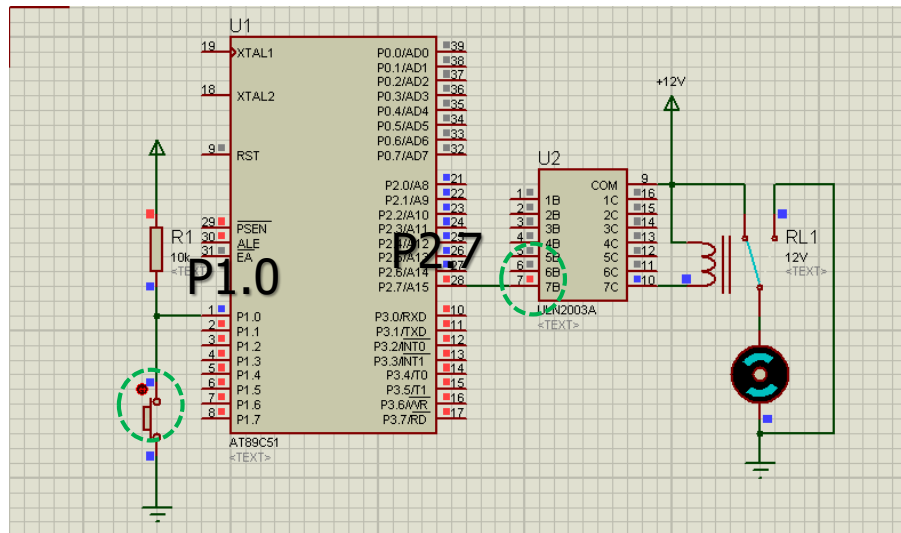

บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ก) งานควบคุมดีซีมอเตอร์ทั่วไป

ตัวอย่างที่ใช้เงื่อนไข (if) สองทางเลือก

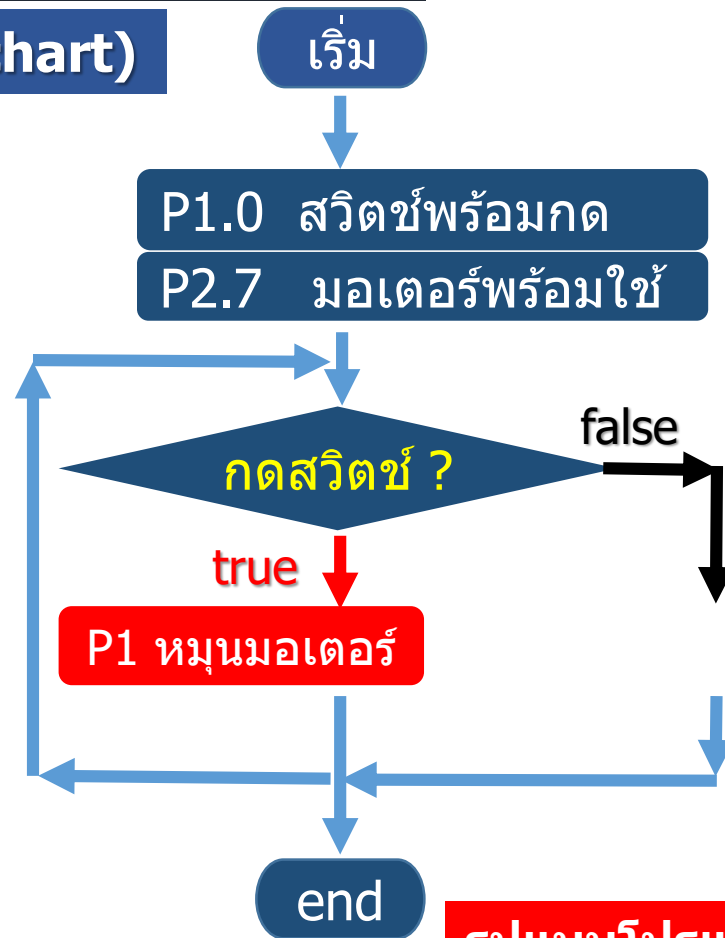
สวิตช์เปิด/ปิด มอเตอร์

ผังความคิด (flowchart)



วงจรสวิตช์ใช้ port P1.0

วงจรมอเตอร์ใช้ port P2.7



รูปแบบโปรแกรม

If(statement is real) { do P1 }else{do P2}

```
#include<reg51.h>
sbit s0=P1^0;
sbit m0=P2^7;
void main ()
{ P1=0xFF; P2=0x00;
  while(1)
  { if(s0==0) {m7=1;}
  }
}
```



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ก) งานควบคุมดีซีมอเตอร์ทั่วไป

ตัวอย่างที่ใช้เงื่อนไข (if) สองทางเลือก

สวิตช์เปิด/ปิด มอเตอร์

ผังความคิด (flowchart)

เริ่ม

P1.0 สวิตช์พร้อมกด

P2.7 มอเตอร์พร้อมใช้

กดสวิตช์ ?

false

true

P1 หมุนมอเตอร์

P2 หยุดมอเตอร์

end

รูปแบบโปรแกรม

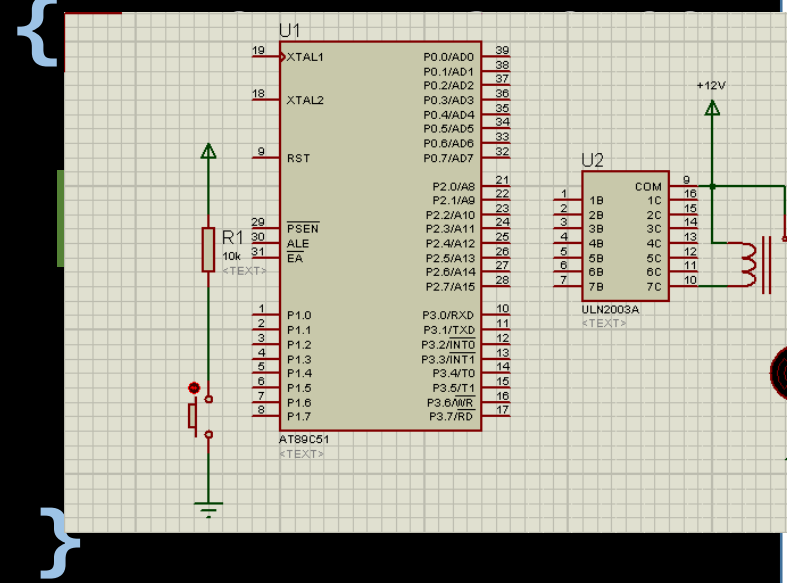
```
#include<reg51.h>
```

```
sbit s0=P1^0;
```

```
sbit m0=P2^7;
```

```
void main ()
```

```
{
```



```
}
```

วงจรสวิตช์ใช้ port P1.0

วงจรมอเตอร์ใช้ port P2.7

If(statement is real) { do P1 }else{do P2}

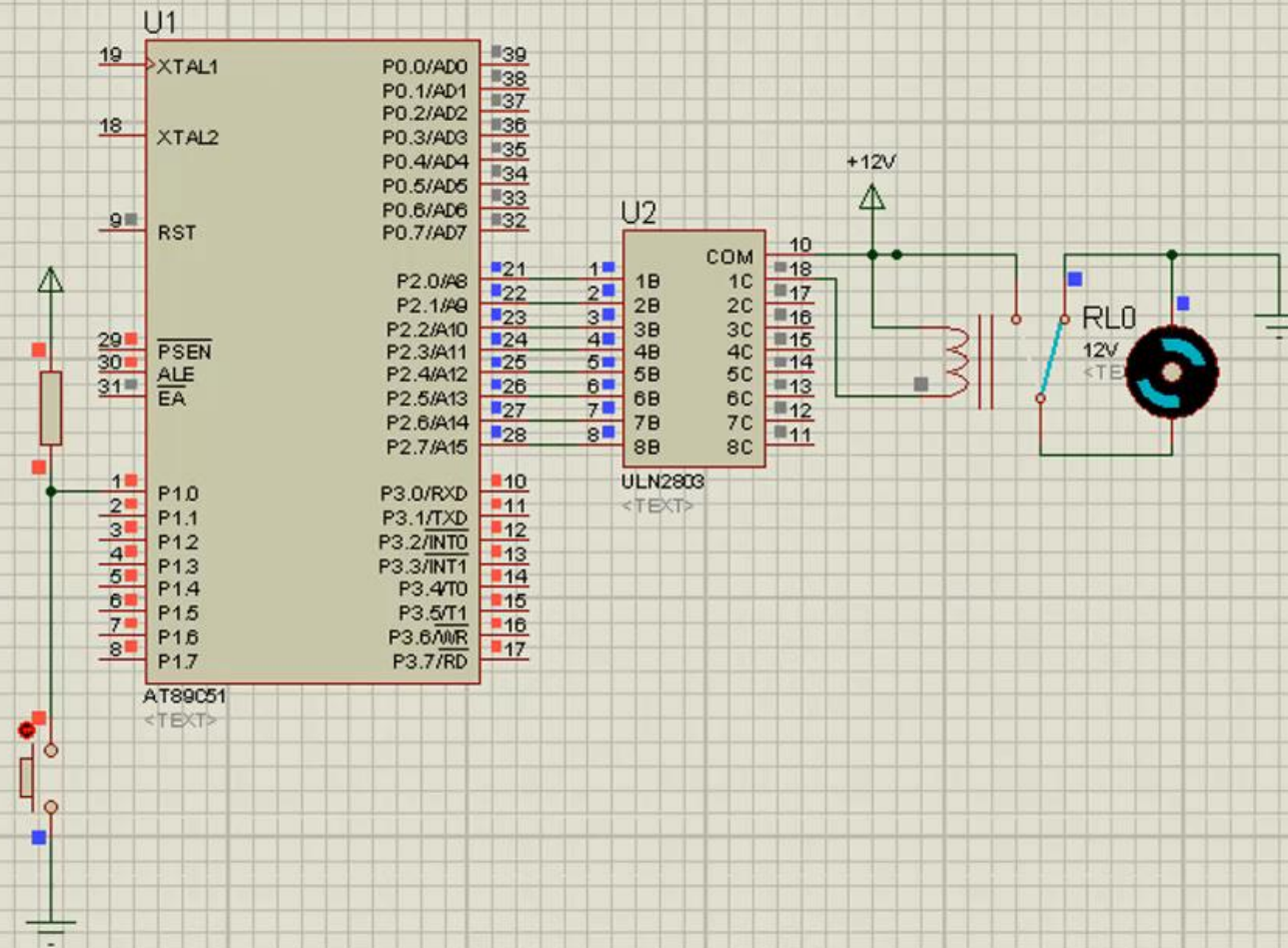


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ก) งานควบคุมดีซีมอเตอร์ทั่วไป



```
1 #include<reg51.h>
2 void main ()
3 {int x;
4   while(1)
5   { P1=0xFF;x=P1;
6     if(x==0xFE){P2=(0x01);}
7     else {P2=(0x00);} }
8 }
```



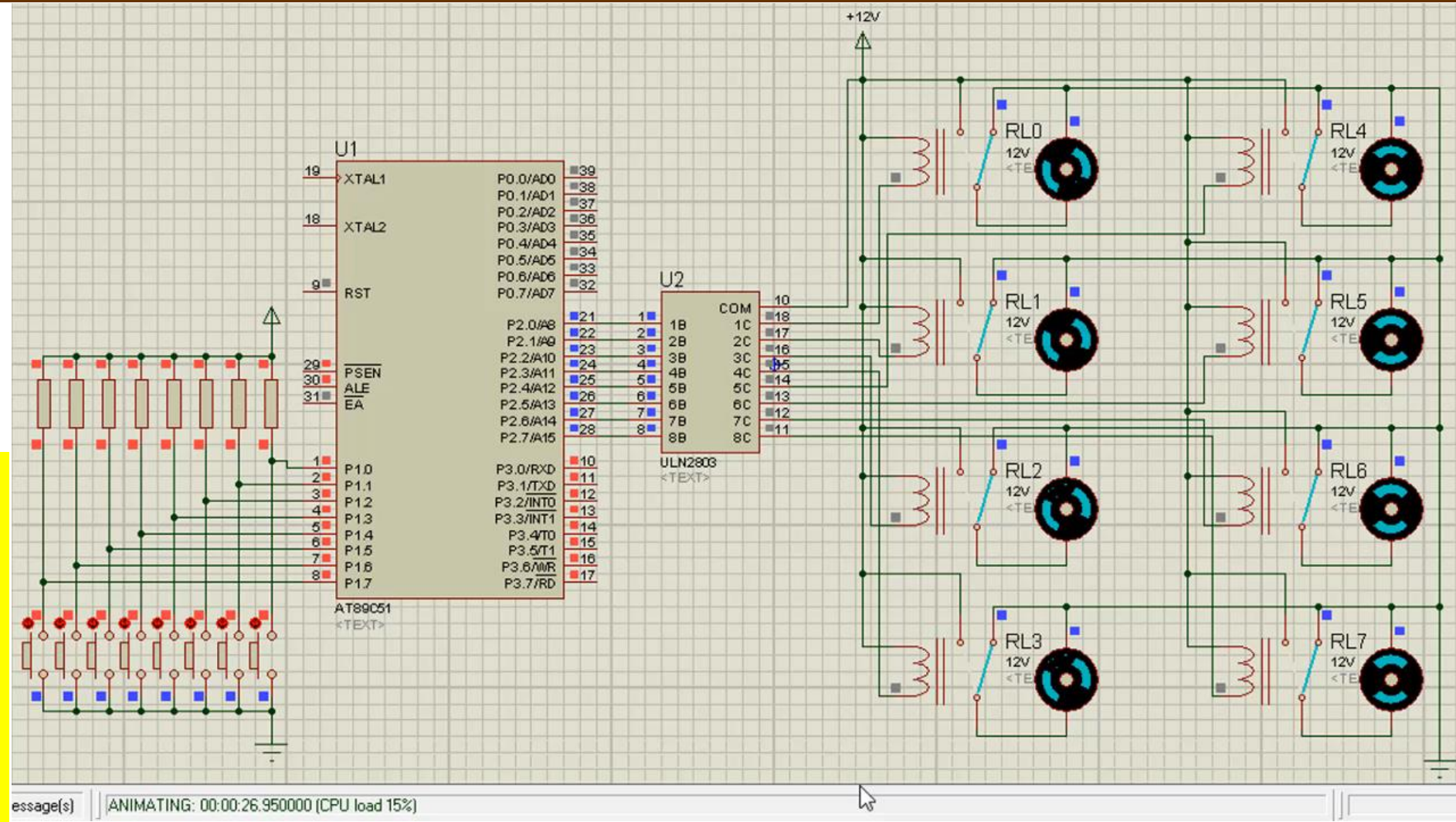


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ก) งานควบคุมดีซีมอเตอร์ทั่วไป



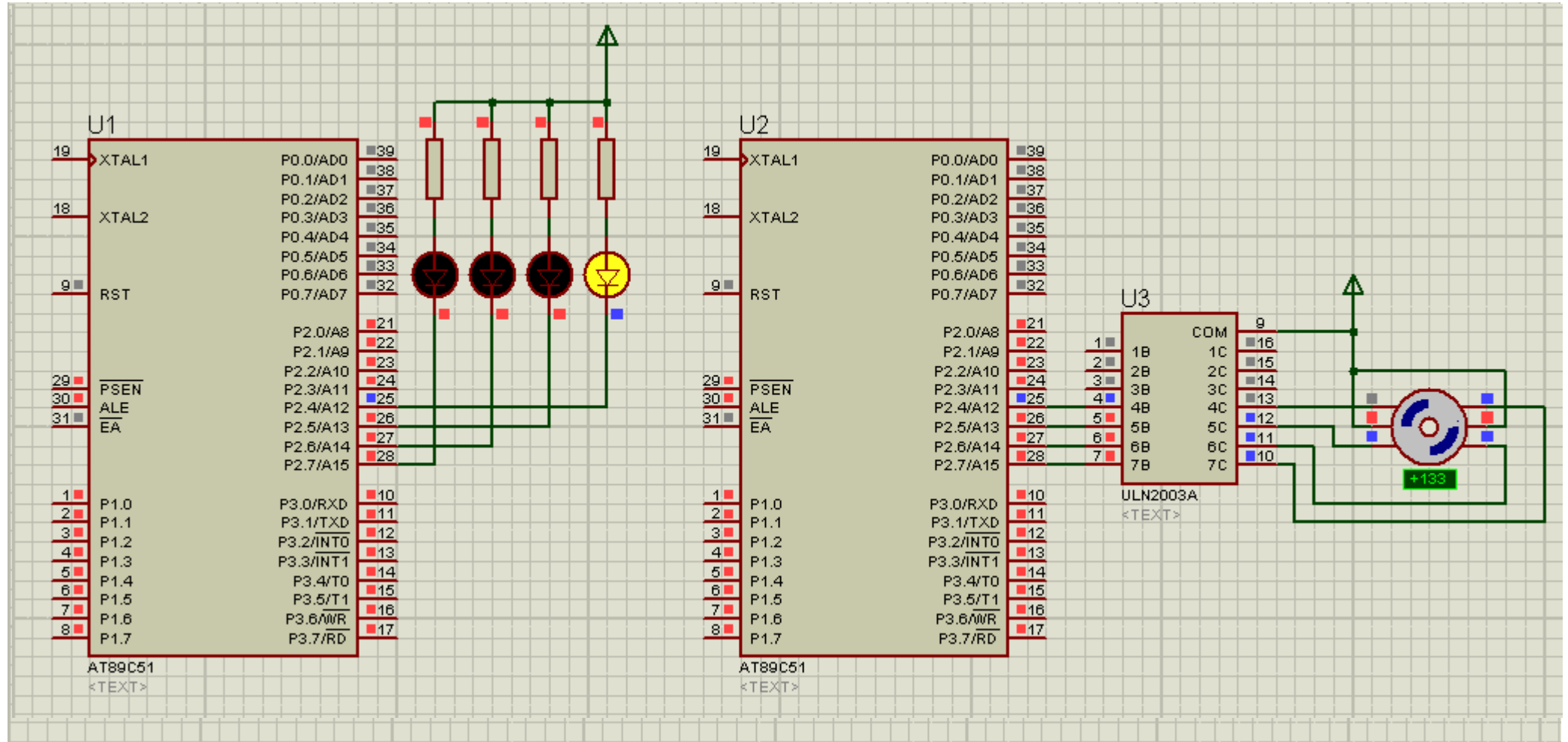
```
01 #include<reg51.h>
02 void main ()
03 {int x;
04 while(1)
05 { P1=0xFF;x=P1;
06 switch(x)
07 {case 0xFE: {P2=(0x0F);break;}
08 case 0xFD: {P2=(0xF0);break;}
09 case 0xFB: {P2=(0xAA);break;}
10 case 0xF7: {P2=(0x55);break;}
11 case 0xEF: {P2=(0xCC);break;}
12 case 0xDF: {P2=(0x33);break;}
13 case 0xBF: {P2=(0x03);break;}
14 case 0x7F: {P2=(0xFC);break;}
15 default: {P2=(0xFF);break;} }
16 }
17 }
```





บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ข) การควบคุมการหมุนของมอเตอร์แบบขั้น (stepper Motor)

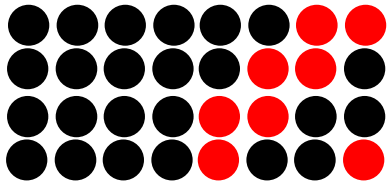




6.1 (ข) การควบคุมการหมุนของมอเตอร์แบบขั้น (stepper Motor)

นิยาม

stepper motor



stepper motor

- มอเตอร์แบบขั้น ที่มีขดลวดภายใน 2 ชุดๆ ละ 2 เฟส และมีจุดร่วมกัน 2 เฟส/จุด
- การควบคุมทิศทางหมุน(direction) แบบง่ายทำได้โดยการป้อนไฟแต่ละเฟสตามลำดับดังตารางขวามือ
- การควบคุมความเร็วในการหมุนแบบง่ายทำได้โดยการกำหนดค่าหน่วยเวลาในการป้อนไฟแต่ละเฟส

```
1a > _____
2b > _____

01 #include<reg51.h>
02 void delay(unsigned int t);
03 void main ()
04 {while(1)
05 { P2=(0x03);delay(600);
06  P2=(0x06);delay(600);
07  P2=(0x0C);delay(600);
08  P2=(0x09);delay(600);}
09
10 }
11 void delay(unsigned int t)
12 {for (t=t;t>0;t--);}
```

หมุนตามเข็มนาฬิกา

หมุนทวนเข็มนาฬิกา

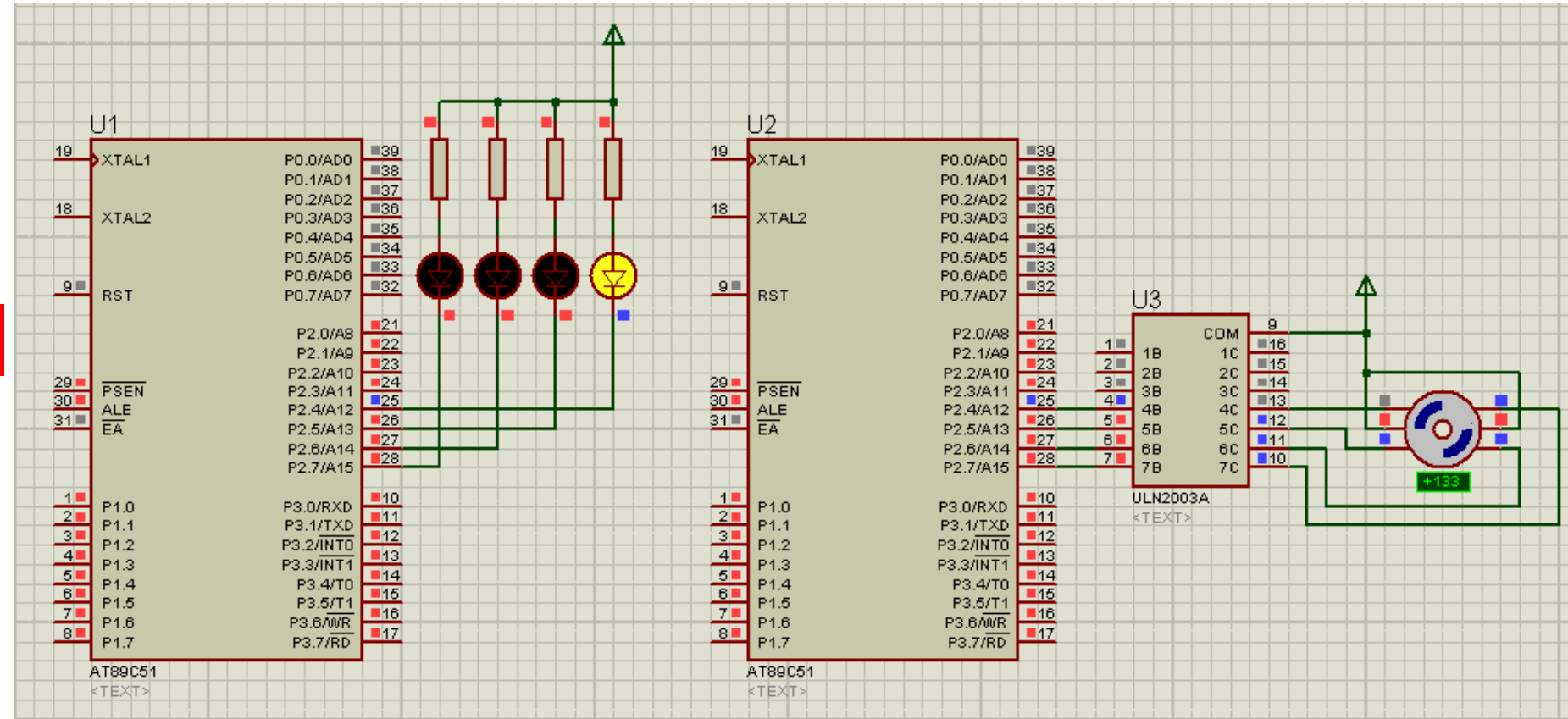


6.1 (ข) การควบคุมการหมุนของมอเตอร์แบบขั้น (stepper Motor)



มอเตอร์แบบขั้นใช้ port P2

```
#include<reg51.h>
void delay(unsigned int t);
void main ()
{while(1)
{ P2=(0x03);delay(600);
  P2=(0x06);delay(600);
  P2=(0x0C);delay(600);
  P2=(0x09);delay(600);}}
void delay(unsigned int t)
{for (t=t;t>0;t--);}
```



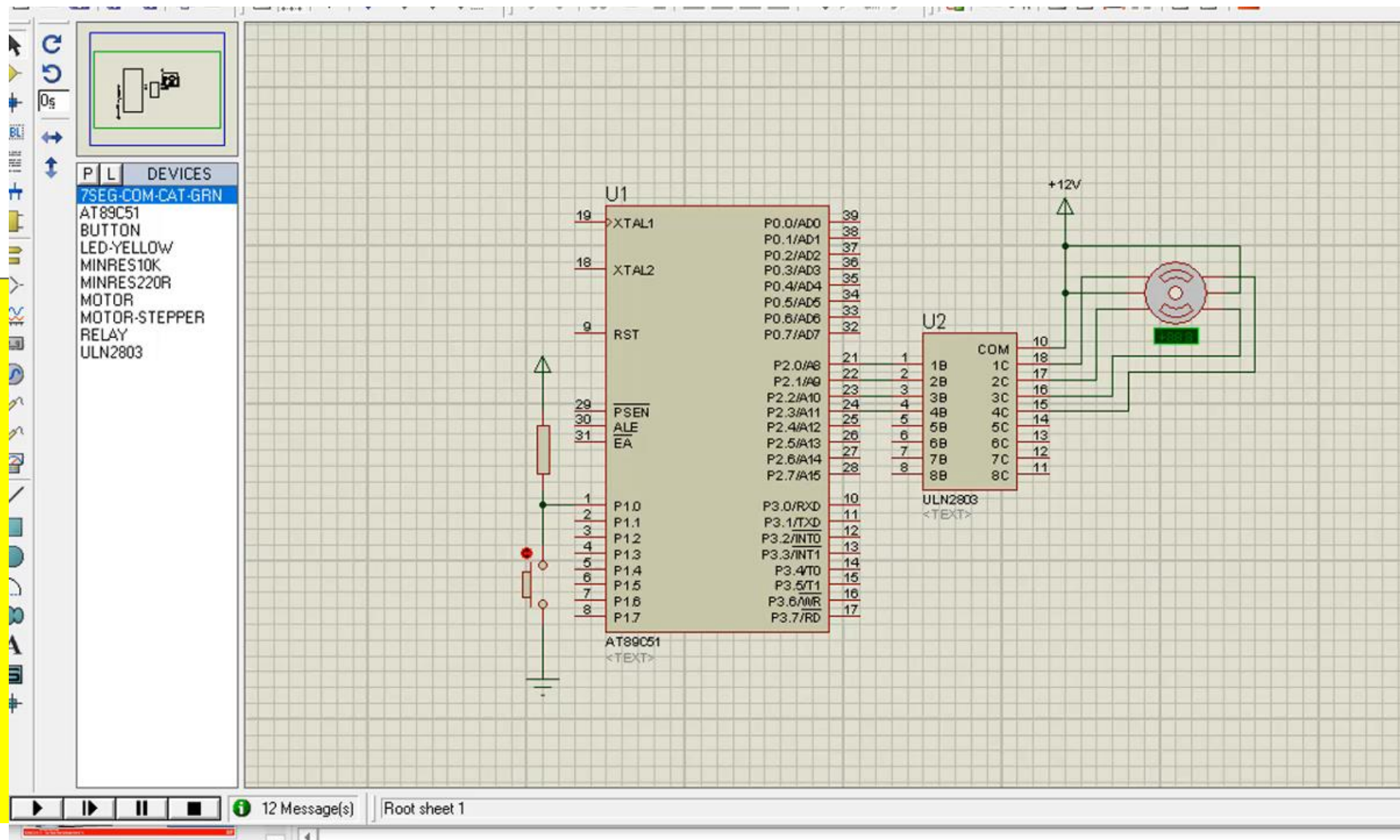


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ข) การควบคุมการหมุนของมอเตอร์แบบขั้น (stepper Motor)



```
01 #include<reg51.h>
02 void delay(unsigned int t);
03 void main ()
04 {int x;
05  while(1)
06  { P1=0xFF;x=P1;
07    if(x==0xFE)
08    {P2=(0x03);delay(600);
09     P2=(0x06);delay(600);
10     P2=(0x0C);delay(600);
11     P2=(0x09);delay(600);}
12    else
13    {P2=(0x09);delay(600);
14     P2=(0x0C);delay(600);
15     P2=(0x06);delay(600);
16     P2=(0x03);delay(600);} }
17 }
18 void delay(unsigned int t)
19 {for (t=t;t>0;t--);}
```



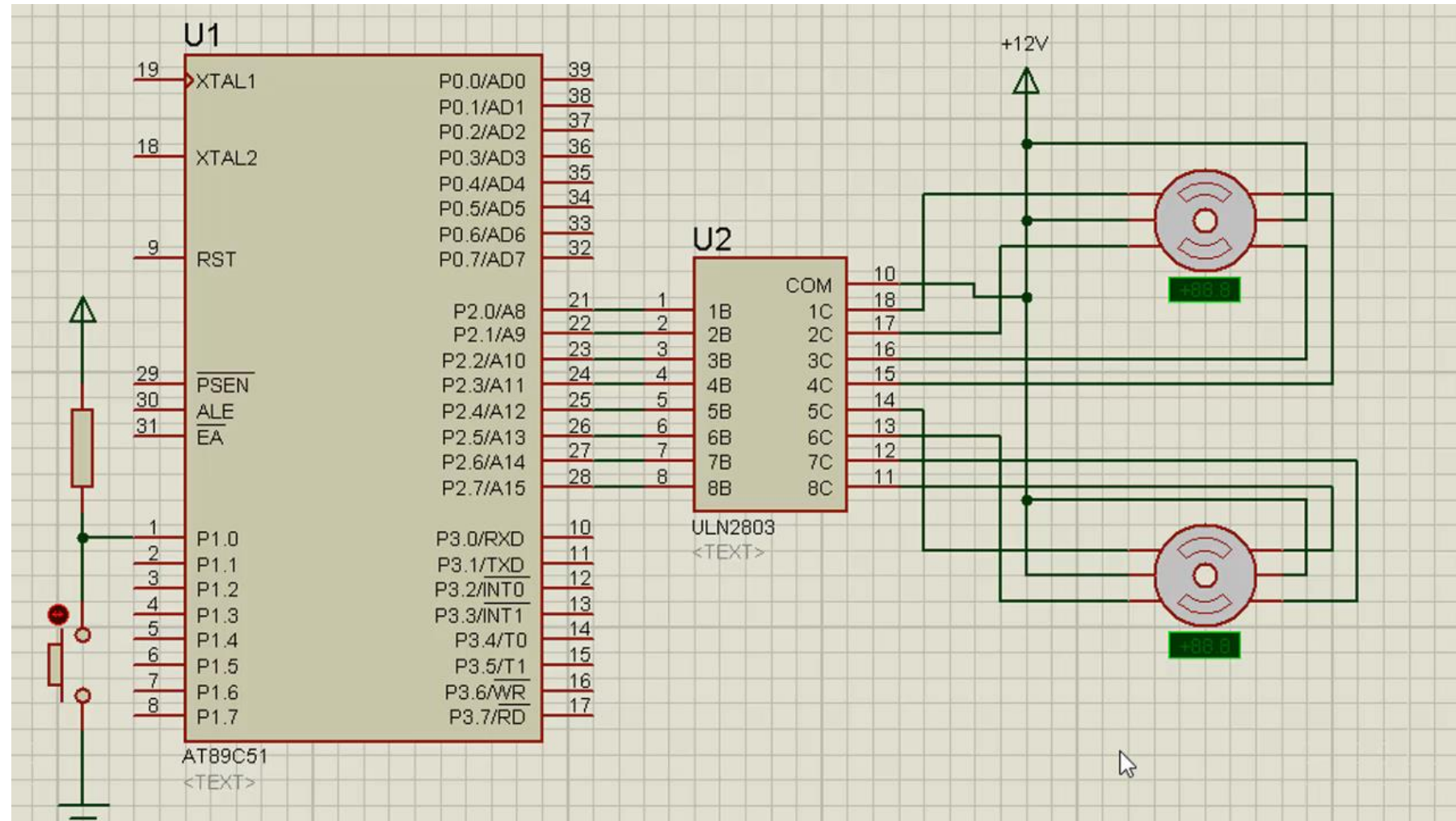


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.1 (ข) การควบคุมการหมุนของมอเตอร์แบบขั้น (stepper Motor)



```
01 #include<reg51.h>
02 void delay(unsigned int t);
03 void main ()
04 {int x;
05  while(1)
06  { P1=0xFF;x=P1;
07    if(x==0xFE)
08    {P2=(0x93);delay(600);
09     P2=(0xC6);delay(600);
10     P2=(0x6C);delay(600);
11     P2=(0x39);delay(600);}
12    else
13    {P2=(0x39);delay(600);
14     P2=(0x6C);delay(600);
15     P2=(0x96);delay(600);
16     P2=(0xC3);delay(600);} }
17 }
18 void delay(unsigned int t)
19 {for (t=t;t>0;t--);}
```



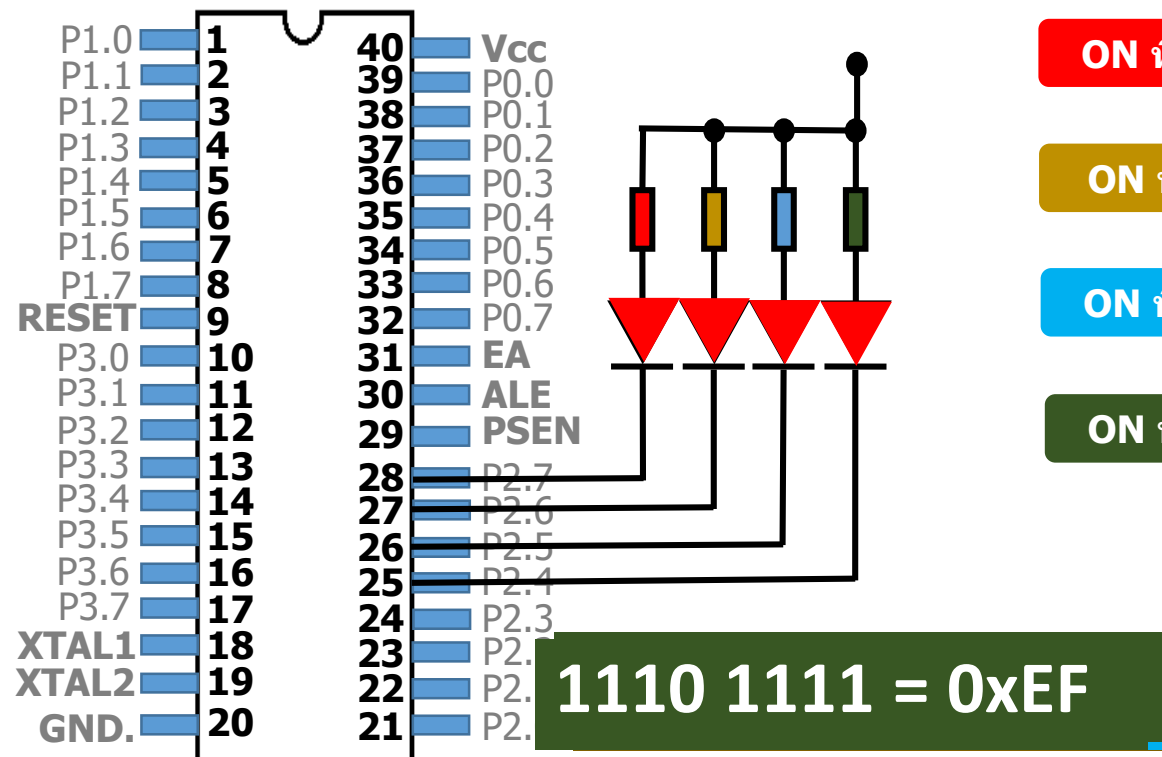


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.2 (ก) งานควบคุมการติด/ดับของดวงไฟ 4 บิต / 8 บิต

ตัวอย่างที่ใช้เอาต์พุต

ผังความคิด (flowchart)



```
#include<reg51.h>
Int x;

void delay (unsigned int t);
void main ()
{
    P2=0x80; x=P2;
    while(1)
    {
        P2= ~x; delay(100);
        x=(x>>1);
        if(x==0x08) {x=0x80;}
    }
}

void delay (unsigned int t)
{
    for (t=t; t>0; t--);
}
```

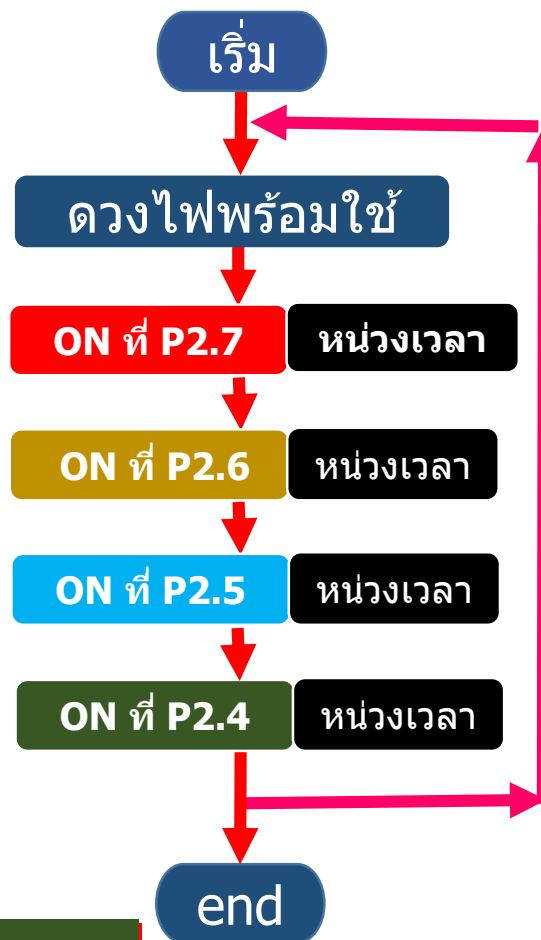
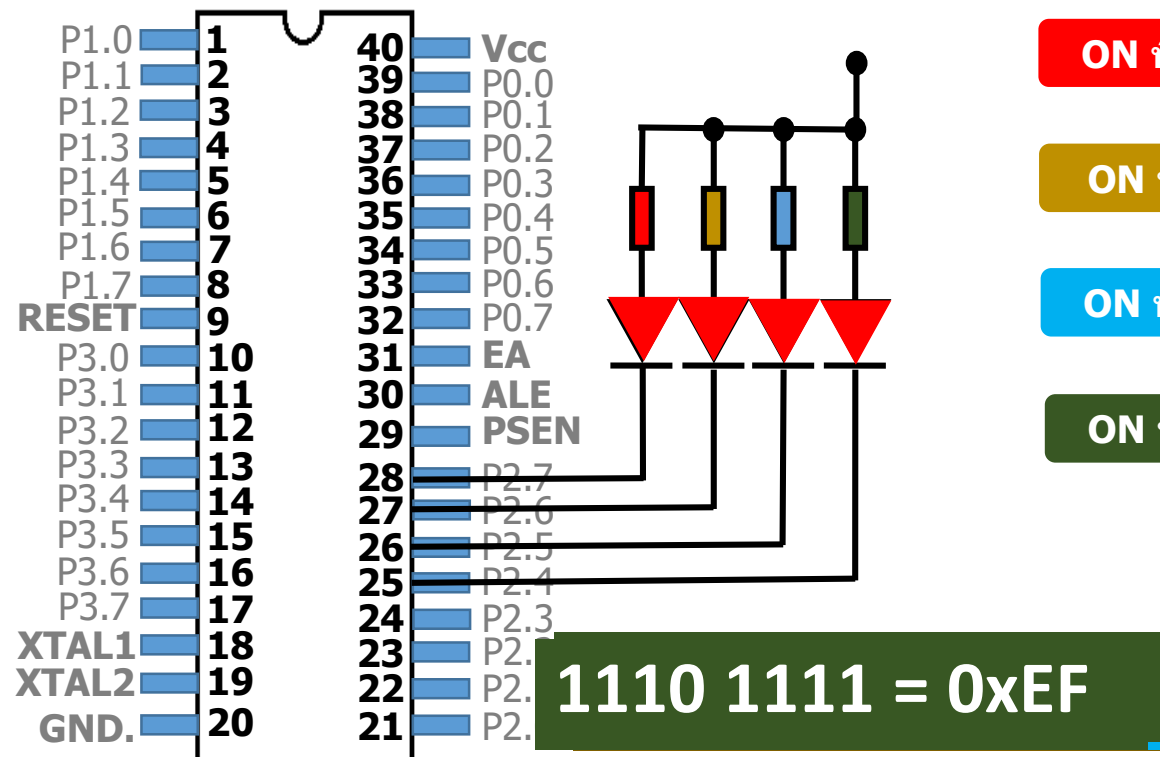


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.2 (ก) งานควบคุมการติด/ดับของดวงไฟ 4 บิต / 8 บิต

ตัวอย่างที่ใช้เอาต์พุต

ผังความคิด (flowchart)



```
#include<reg51.h>
Int x;
void delay (unsigned int t);
void main ()
{ P2=0x80;
  while(1)
  { P2= ~x; delay(100);
    x=(x>>1);
    if(x==0x08) {x=0x80;}
  }
}

void delay (unsigned int t)
{ for (t=t; t>0; t--); }
```



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

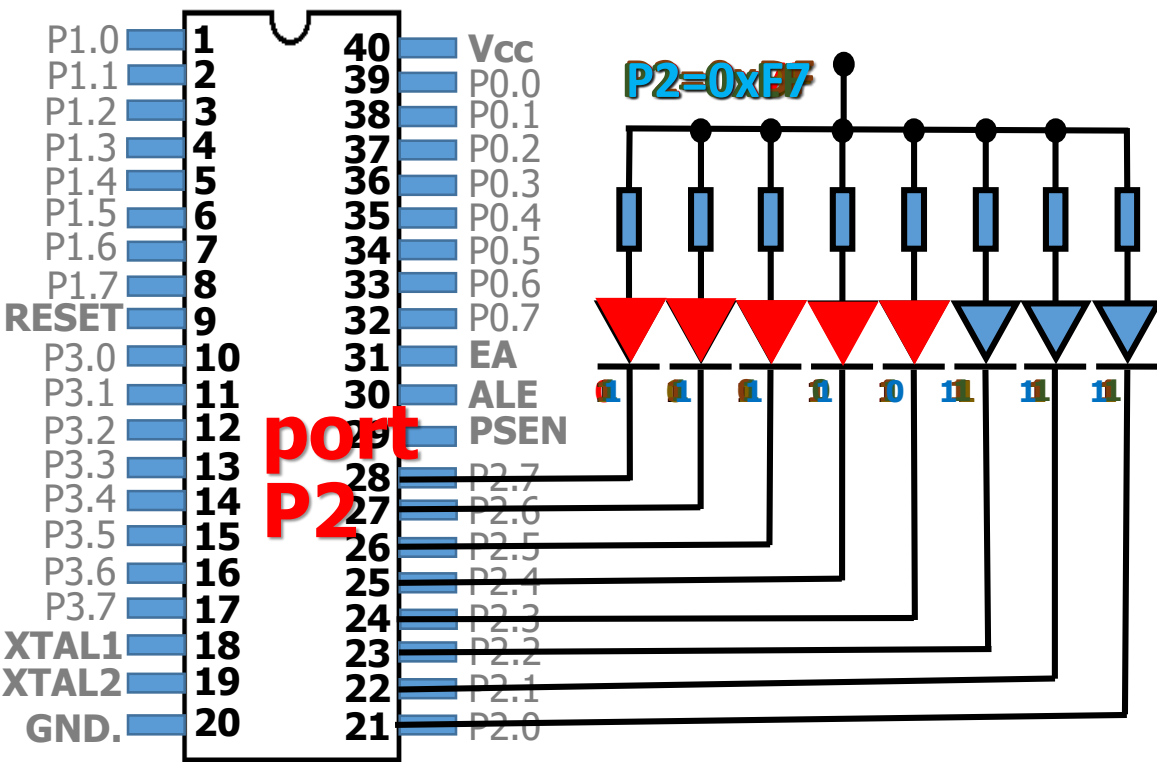
6.2 (ก) งานควบคุมการติด/ดับของดวงไฟ 4 บิต / 8 บิต

ตัวอย่างที่ใช้เอาต์พุต

เปิด/ปิด ดวงไฟ4 ดวง

ผังความคิด (flowchart)

วงจรดวงไฟใช้ port P2



```
#include<reg51.h>
```

```
void delay (long t) ;
```

```
void main ();
```

```
{while(1)  
{ P2=0xFF;
```

```
  P2=0x7F; delay(100);
```

```
  P2=0xBF; delay(100);
```

```
  P2=0xDF; delay(100);
```

```
  P2=0xEF; delay(100);
```

```
  P2=0xF7; delay(100);
```

```
}
```

```
}
```

```
void delay (long t)
```

```
{for (t=t;t>0;t--);}
```




บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.2 (ข) งานควบคุมส่วนแสดงผล 7 ส่วน (7-segment display)

ตัวอย่างที่ใช้เอาต์พุต

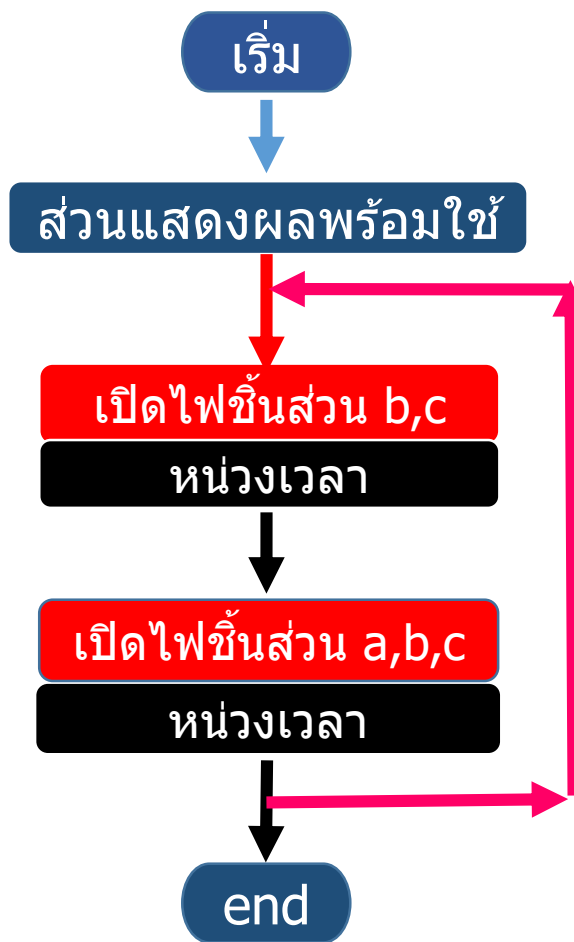
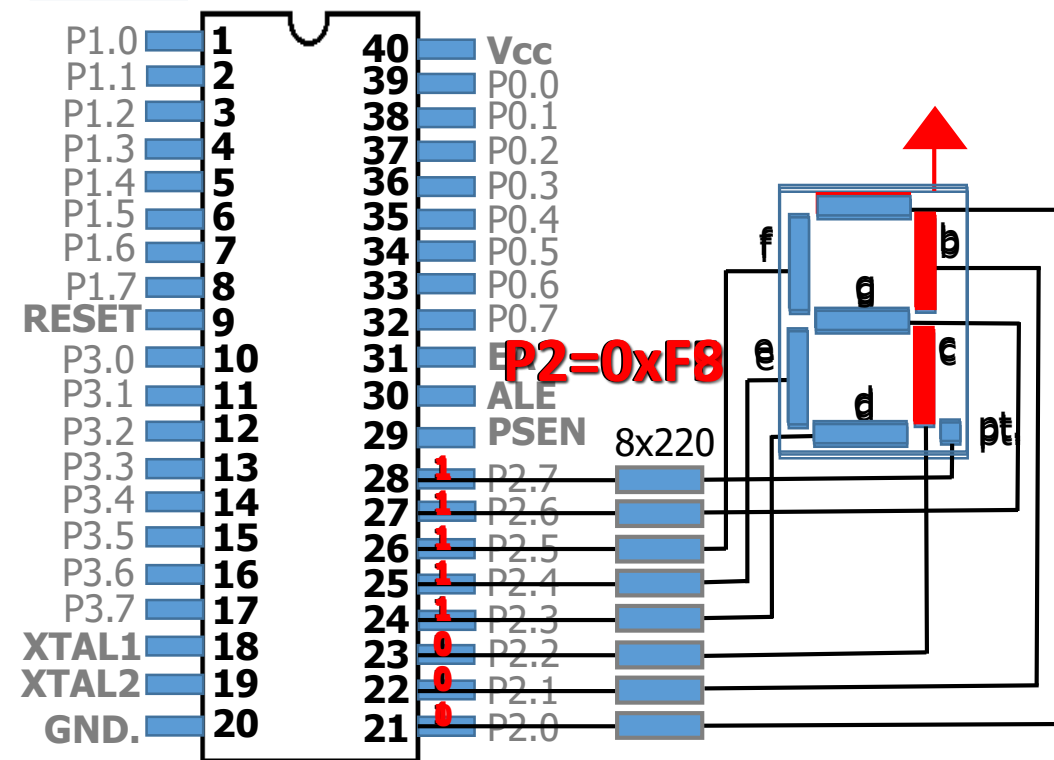
แสดงผลเจ็ดส่วนแบบบวกร่วม

ส่วนแสดงผลใช้ port P2

ผังความคิด (flowchart)

ส่วนแสดงผลแบบบวกร่วม

สั่งให้ติดสว่างด้วยลอจิก "0"



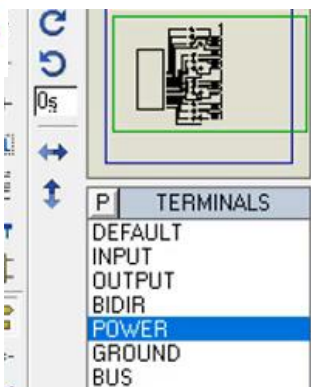
```

#include<reg51.h>
void delay (long t)
void main ()
{ P2=0xFF;
  while(1)
  { P2=0xF9; delay(100);
    P2=0xF8; delay(100);}
}
void delay (long t);
{for (t=t;t>0;t--);}
  
```

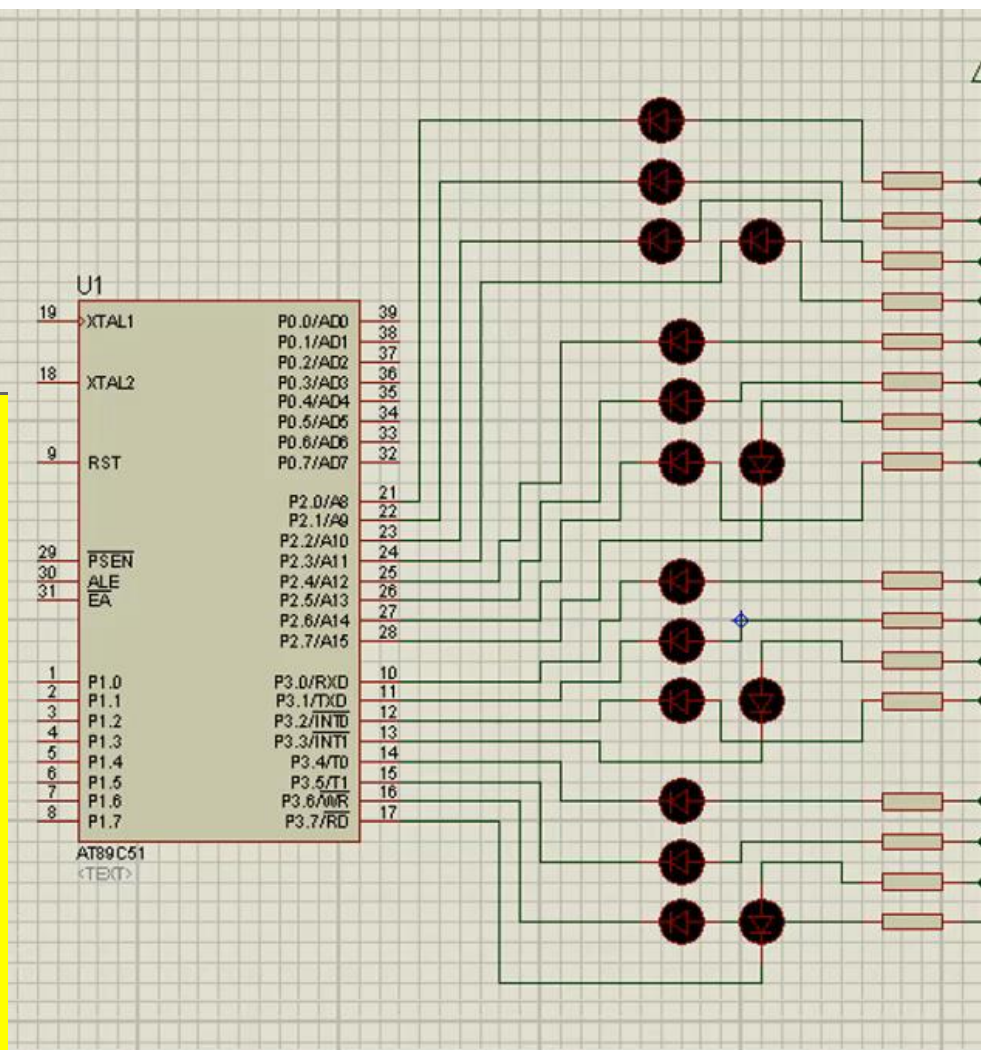



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.2 งานควบคุมไฟจราจร



```
01 #include<reg51.h>
02 int i,x,y;
03 void delay(long t);
04 void main ()
05 {while(1)
06     {for(i=0;i<5;i++)
07         {P2=0xEE;P3=0xBB;delay(250);
08           P2=0xDD;P3=0x33;delay(250);
09           P2=0xBB;P3=0xEE;delay(250);
10           P2=0x33;P3=0xDD;delay(250);} }
11 }
12 void delay(long t)
13 {for(t=t;t>0;t--);}
```



Shows the currently loaded terminal symbols.

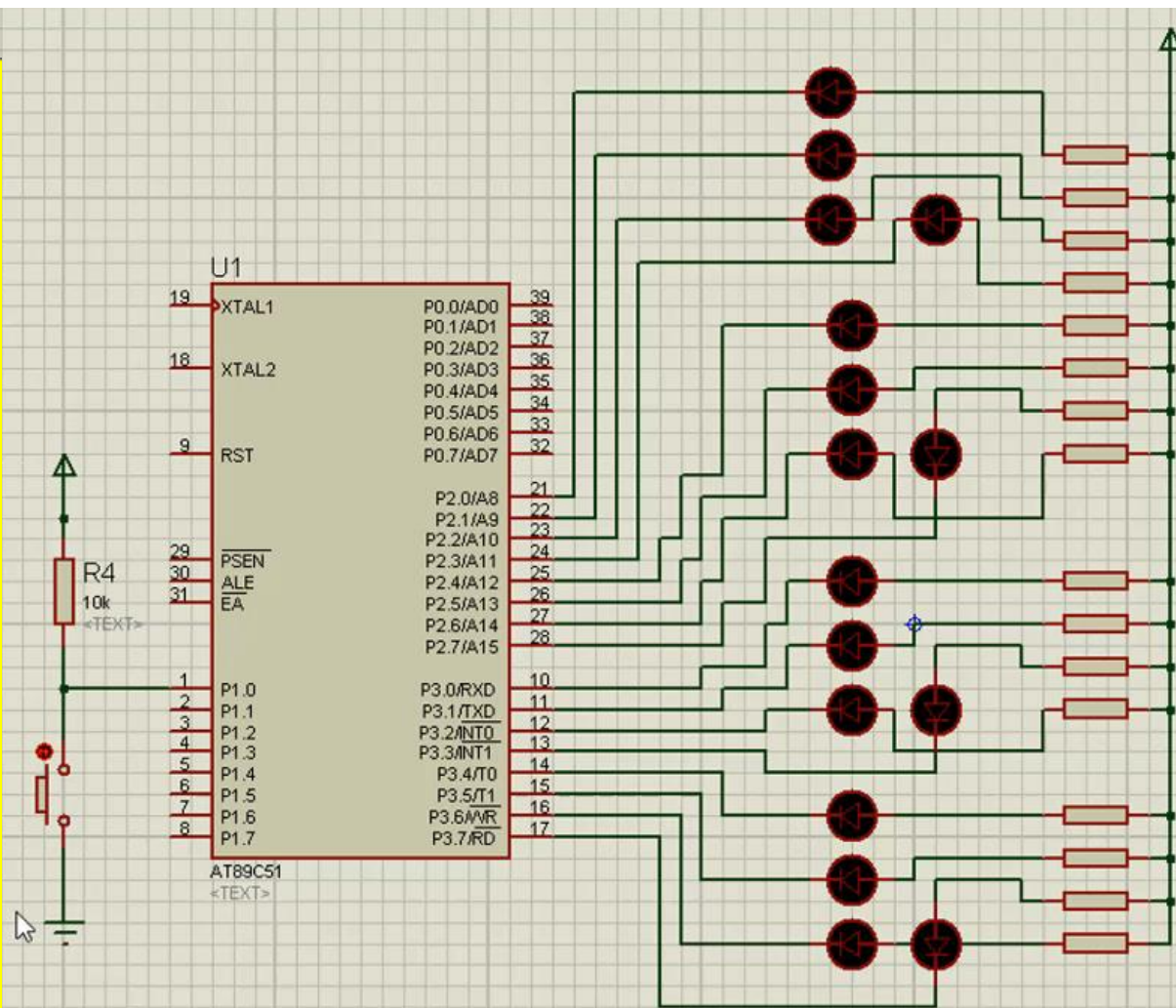


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.2 งานควบคุมไฟจราจร



```
01 #include<reg51.h>
02 int i,x,y,k;
03 void delay(long t);
04 void main ()
05 {P1=0xFF;P2=0xFF;P3=0xFF;
06  while(1)
07  {k=P1;
08
09   {if(k==0xFE)
10   {for(i=0;i<5;i++)
11   {P2=0xEE;P3=0xBB;delay(250);
12   P2=0xDD;P3=0x33;delay(250);
13   P2=0xBB;P3=0xEE;delay(250);
14   P2=0x33;P3=0xDD;delay(250);}} }
15   else if(k==0xFF)
16   {for(i=0;i<3;i++)
17   {P2=0xFF;P3=0x00;delay(250);
18   P2=0x00;P3=0xFF;delay(250);}} }
19   }
20 }
21 }
22 void delay(long t)
23 {for(t=t;t>0;t--);}
```





บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.3 งานเกี่ยวกับสวิตช์ทางเลือกหลายทาง (conditions)

ตัวอย่างที่ใช้ switch หลายทางเลือก

สวิตช์ 4 ตัวเปิด/ปิดควบคุมการแสดงผล 7 ส่วน ...0-9

ผังความคิด
(flowchart)

วงจรสวิตช์ใช้ port
P1.3 P1.2 P1.1 P1.0

วงจรส่วนแสดงผลใช้
port P2

เริ่ม

สวิตช์พร้อมกด
ดวงไฟพร้อมใช้

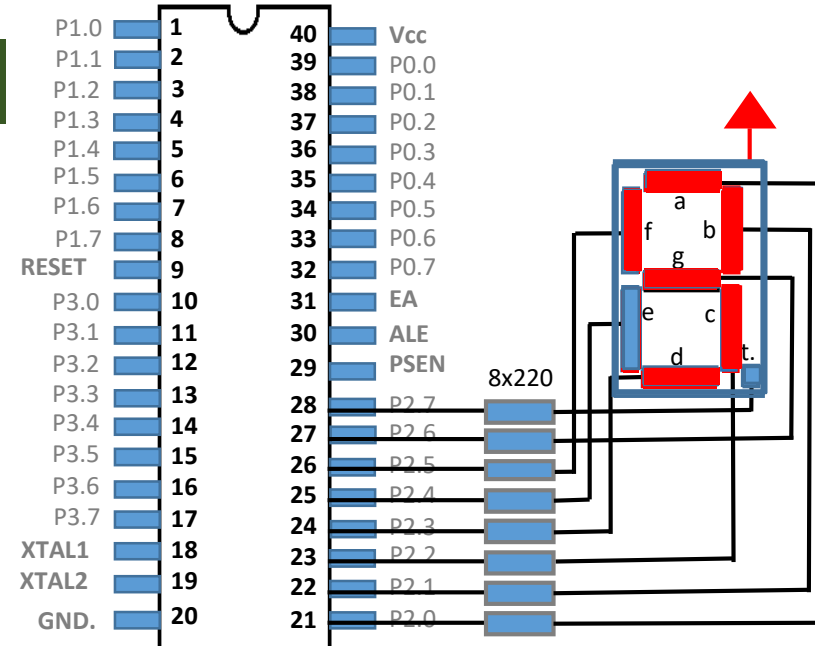
X=?

0xFF 0xFE 0xFD 0xFC 0xFB 0xFA 0xF9 0xF8 0xF7 0xF6 Default

แสดง "0" แสดง "1" แสดง "2" แสดง "3" แสดง "4" แสดง "5" แสดง "6" แสดง "7" แสดง "8" แสดง "9" แสดง " "

0xC0 0xF9 0xA4 0xB0 0x99 0x92 0x82 0xF8 0x80 0x90 0xFF

สั่งให้ติดสว่างด้วยลอจิก "0"



P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	Hex.
pt.	g	f	e	d	c	b	a	
1	0	0	0	0	0	0	0	0x00



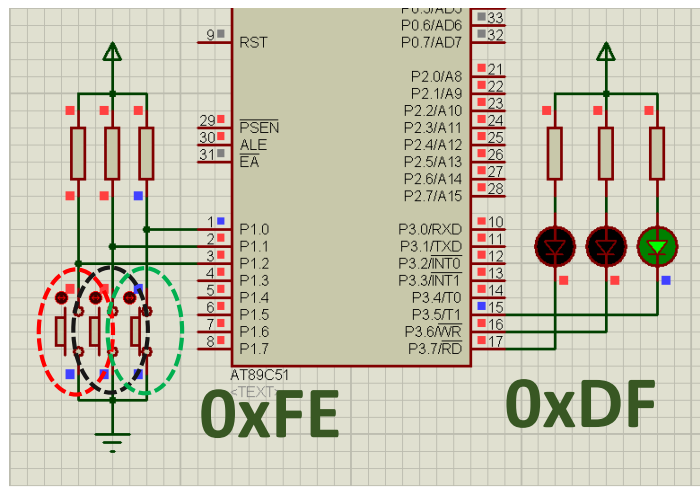
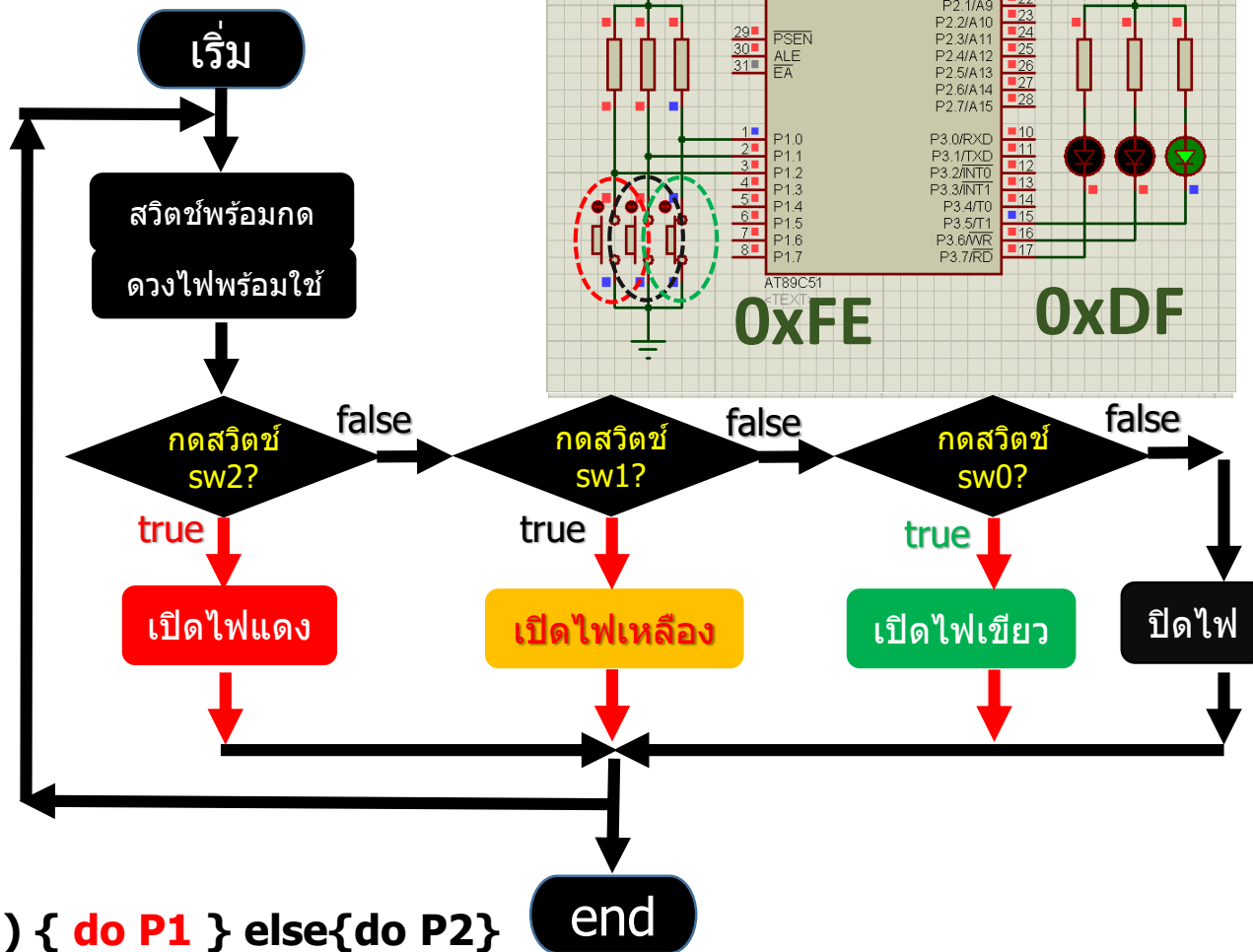
บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.3 (ก) งานเกี่ยวกับทางเลือก (if - condition)

ตัวอย่างที่ใช้เงื่อนไข (if) หลายทางเลือก

สวิตช์ 3 ตัวเปิด/ปิด ควบคุม ดวงไฟ 3 ดวง

ผังความคิด (flowchart)



วงจรดวงไฟใช้ port P3.7 P3.6 P3.5

```

#include<reg51.h>
void main ()
{
    Unsigned int xsw;
    while(1)
    {
        P1=0xFF;P3=0xFF;xsw=P1;
        if(xsw==0xFB){P3=0x7F;}
        else if(xsw==0xFD){P3=0xBF;}
        else if(xsw==0xFE){P3=0xDF;}
        else {P3=0xFF;}
    }
}
  
```

วงจรสวิตช์
ใช้ port
P1.2 P1.1 P1.0

รูปแบบโปรแกรม

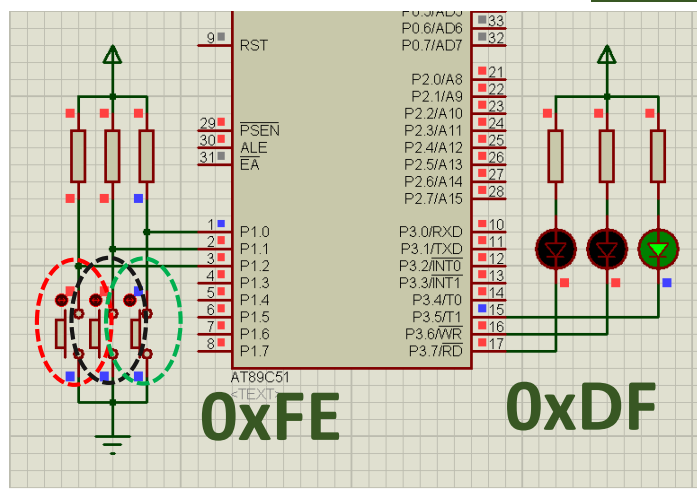
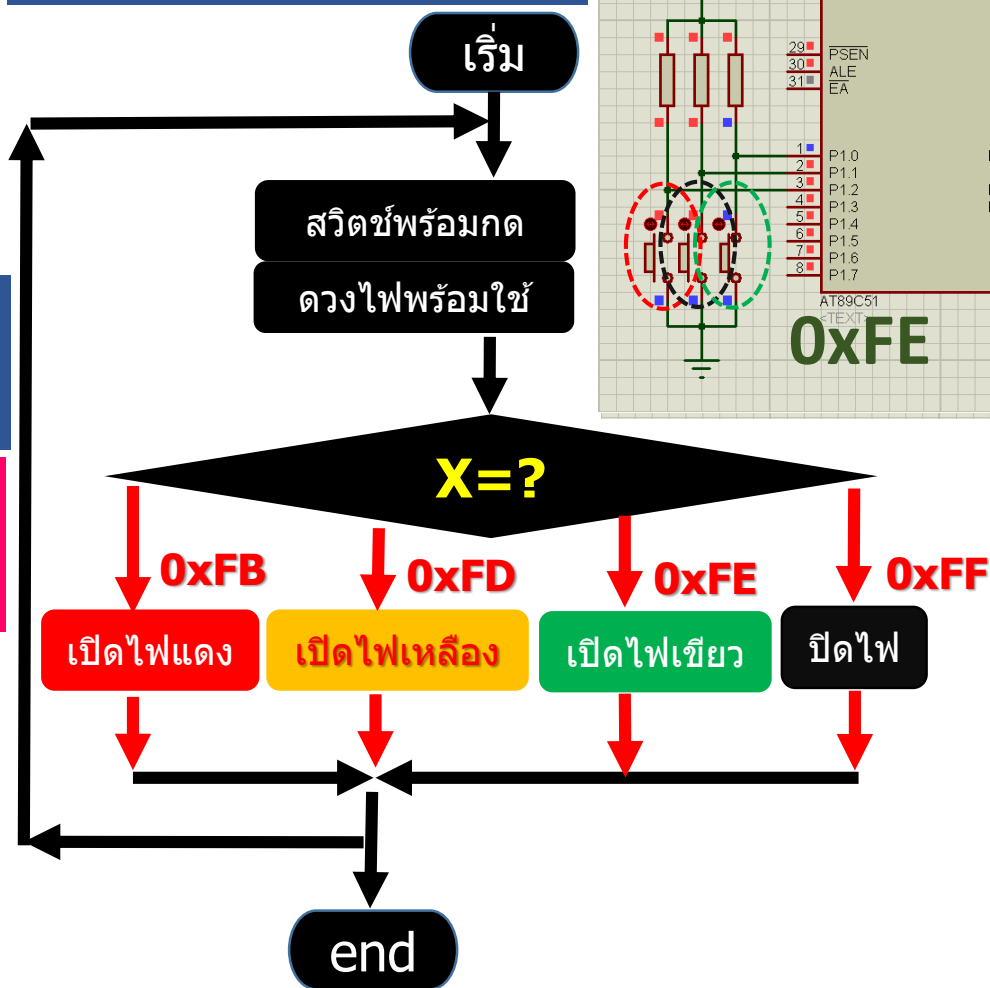
If(statement is real) { do P1 } else{do P2}



6.3 (ข) งานเกี่ยวกับทางเลือก (case - condition)

ตัวอย่างที่ใช้ switch หลายทางเลือก

สวิตช์ 3 ตัวเปิด/ปิด ความคุม ดวงไฟ 3 ดวง



```
#include<reg51.h>
int x;
void main ()
{ P1=0xFF; P3=0xFF;
  while(1)
  { X=P1;
```

switch(x)

Case 0xFB: {P3=0x7F;break;}

Case 0xFD: {P3=0xBF; break;}

Case 0xFE: {P3=0xDF; break;}

default: {P3=0xFF;}

}

}



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.3 (ข) งานเกี่ยวกับทางเลือก (case - condition)

สวิตช์ 4 ตัวเปิด/ปิดควบคุมการแสดงผล 7 ส่วน ...0-9

ผังความคิด
(flowchart)

วงจรสวิตช์ใช้ port
P1.3 P1.2 P1.1 P1.0

วงจรส่วนแสดงผลใช้
port P3

เริ่ม

สวิตช์พร้อมกด
ดวงไฟพร้อมใช้

X=?

0xFF

0xFE

0xFD

0xFC

0xFB

0xFA

0xF9

0xF8

0xF7

0xF6

default

แสดง
"0"

แสดง
"1"

แสดง
"2"

แสดง
"4"

แสดง
"5"

แสดง
"6"

แสดง
"7"

แสดง
"8"

แสดง
"9"

แสดง
"0"

แสดง
" "

```
#include<reg51.h>
int x;
void main ()
{ P1=0xFF; P3=0xFF;
  while(1)
  { x=P1
    switch(x)
    { Case 0xFF: {P3=0xC0;break;}
      Case 0xFE: {P3=0xF9;break;}
      Case 0xFD: {P3=0xA4;break;}
      Case 0xFC: {P3=0xB0;break;}
      Case 0xFB: {P3=0x99;break;}
      Case 0xFA: {P3=0x92;break;}
      Case 0xF9: {P3=0x82;break;}
      Case 0xF8: {P3=0xF8;break;}
      Case 0xF7: {P3=0x80;break;}
      Case 0xF6: {P3=0x90;break;}
      default: {P3=0xFF;}
    }
  }
}
```



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.3 (ข) งานเกี่ยวกับทางเลือก (case - condition)

สวิตช์ 4 ตัวเปิด/ปิดควบคุมการแสดงผล 7 ส่วน0-9

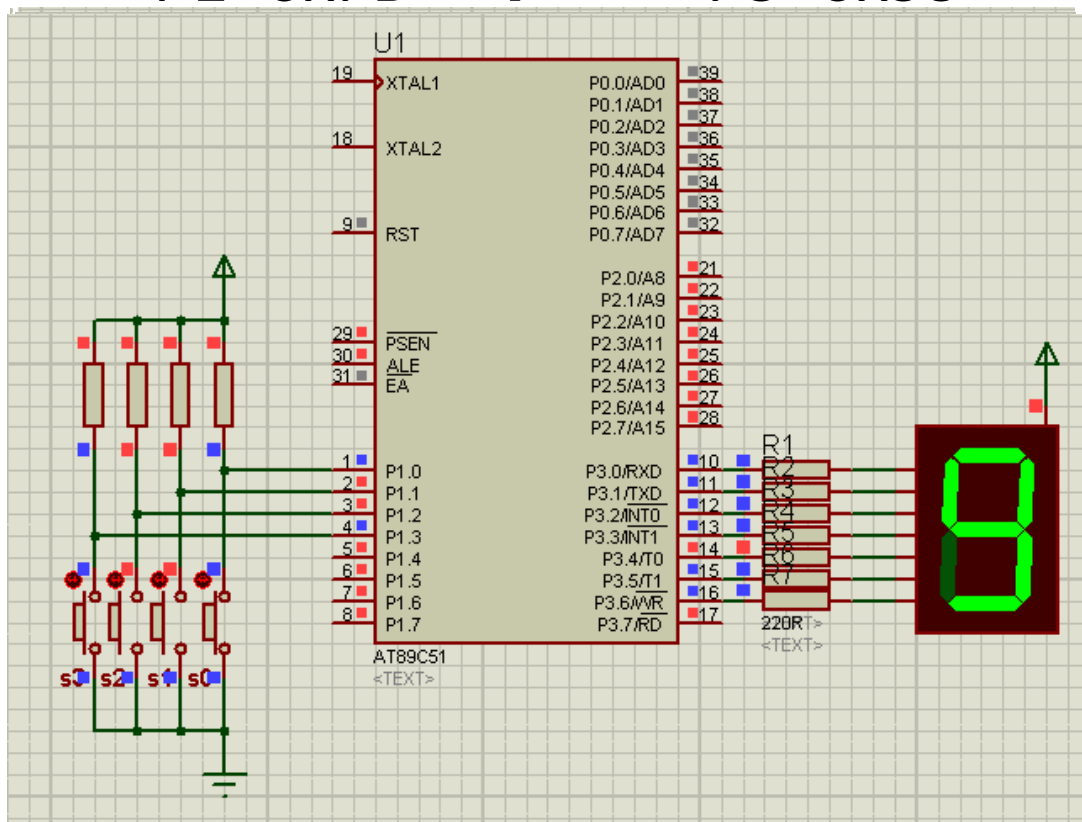
วงจรสวิตช์ใช้ port P1.3 P1.2 P1.1 P1.0

ส่วนแสดงผลใช้ port P3

P1=0xFF



P3=0x99



```
#include<reg51.h>
int x;
void main ()
{ P1=0xFF; P3=0xFF;
  while(1)
  { x=P1
    switch(x)
    { Case 0xFF: {P3=0xC0;break;}
      Case 0xFE: {P3=0xF9;break;}
      Case 0xFD: {P3=0xA4;break;}
      Case 0xFC: {P3=0xB0;break;}
      Case 0xFB: {P3=0x99;break;}
      Case 0xFA: {P3=0x92;break;}
      Case 0xF9: {P3=0x82;break;}
      Case 0xF8: {P3=0xF8;break;}
      Case 0xF7: {P3=0x80;break;}
      Case 0xF6: {P3=0x90;break;}
      default: {P3=0xFF;}
    }
  }
}
```




บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.3 (ข) งานเกี่ยวกับทางเลือก (case - condition)

สวิตช์ 4 ตัวเปิด/ปิดควบคุมการแสดงผล 7 ส่วน0-9

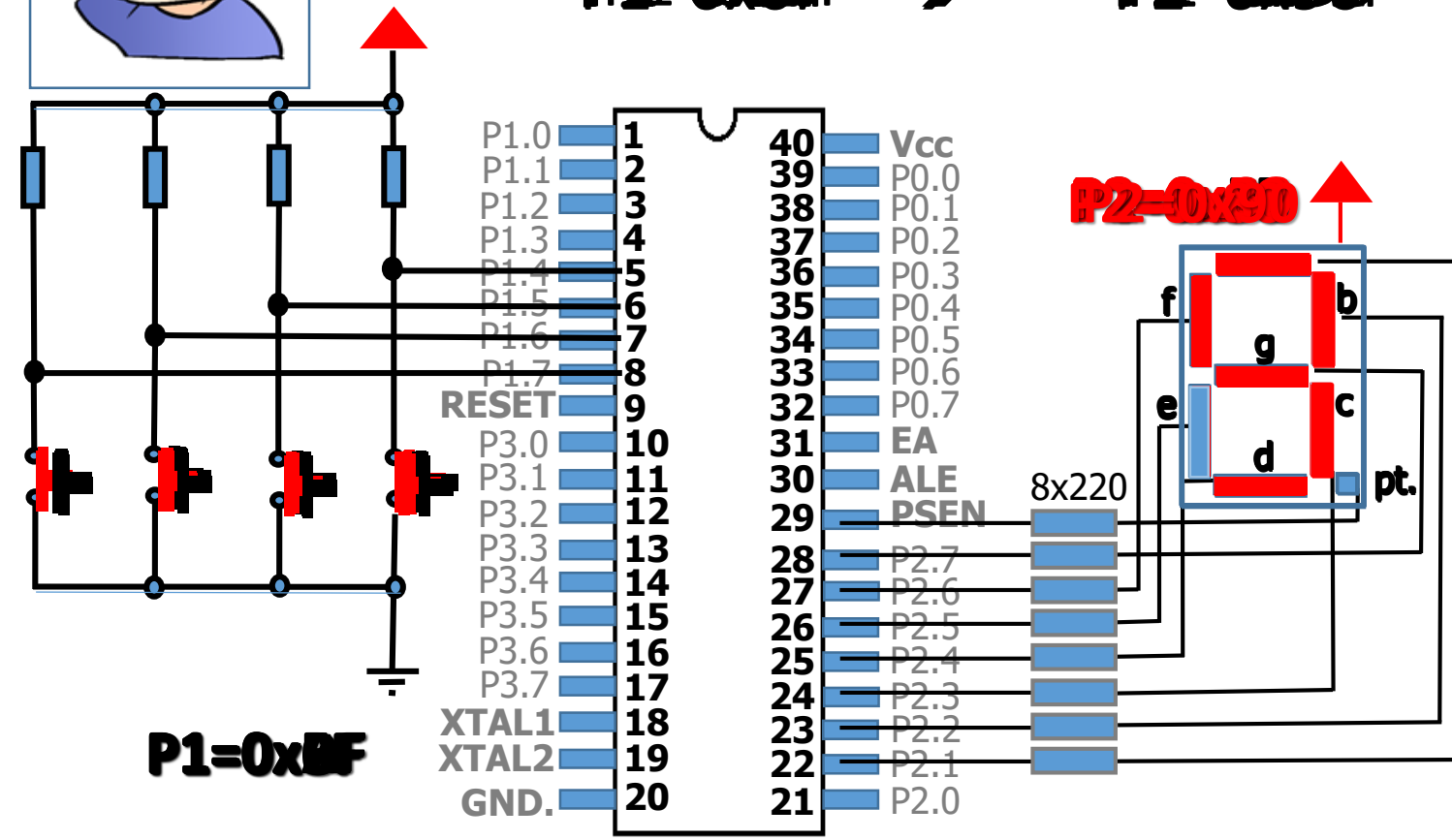
วงจรสวิตช์ใช้ port P1.7 P1.6 P1.5 P1.4

ส่วนแสดงผลใช้ port P3

~~P1=0xFF~~



~~P2=0x00~~



P1=0xBF

~~P2=0x00~~

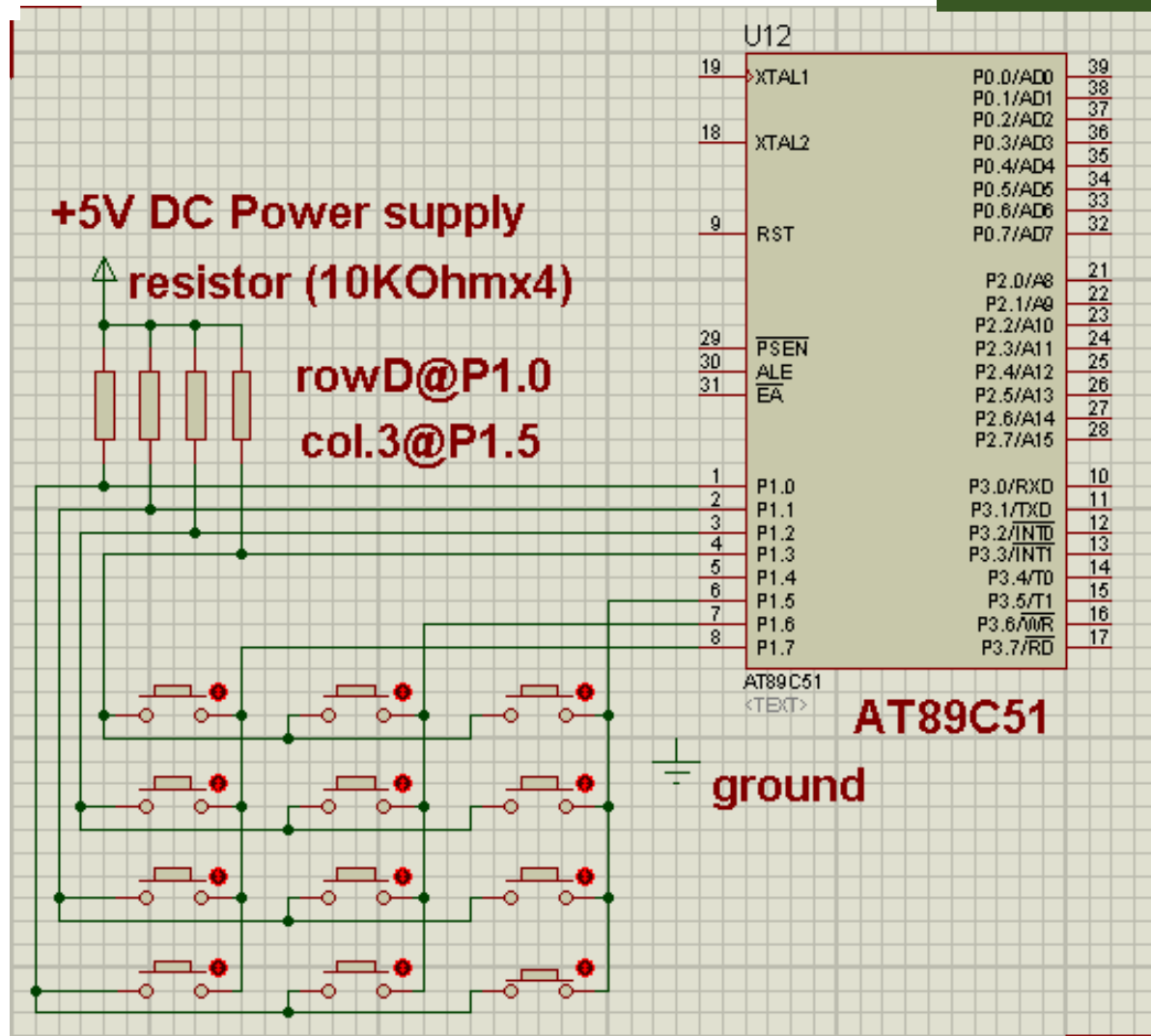
```
#include<reg51.h>
int x;
void main ()
{ P1=0xFF; P2=0xFF;
  while(1)
  { x=P1
    switch(x)
    { Case 0xFF: {P2=0xC0;break;}
      Case 0xEF: {P2=0xF9;break;}
      Case 0xDF: {P2=0xA4;break;}
      Case 0xCF: {P2=0xB0;break;}
      Case 0xBF: {P2=0x99;break;}
      Case 0xAF: {P2=0x92;break;}
      Case 0x9F: {P2=0x82;break;}
      Case 0x8F: {P2=0xF8;break;}
      Case 0x7F: {P2=0x80;break;}
      Case 0x6F: {P2=0x90;break;}
      default: {P2=0xFF;}
    }
  }
}
```




6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



row A, column 3

row B, column 3

row C, column 3

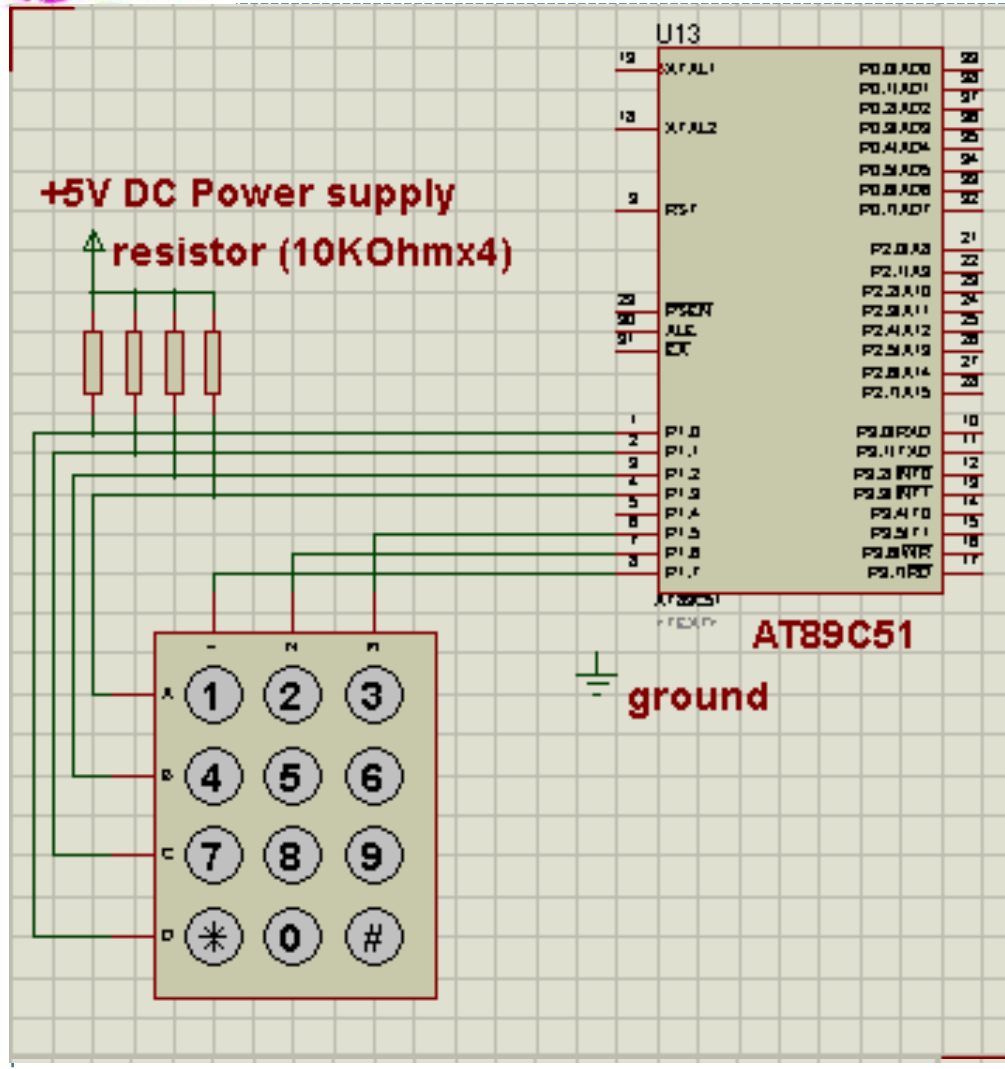
row D, column 3



6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว

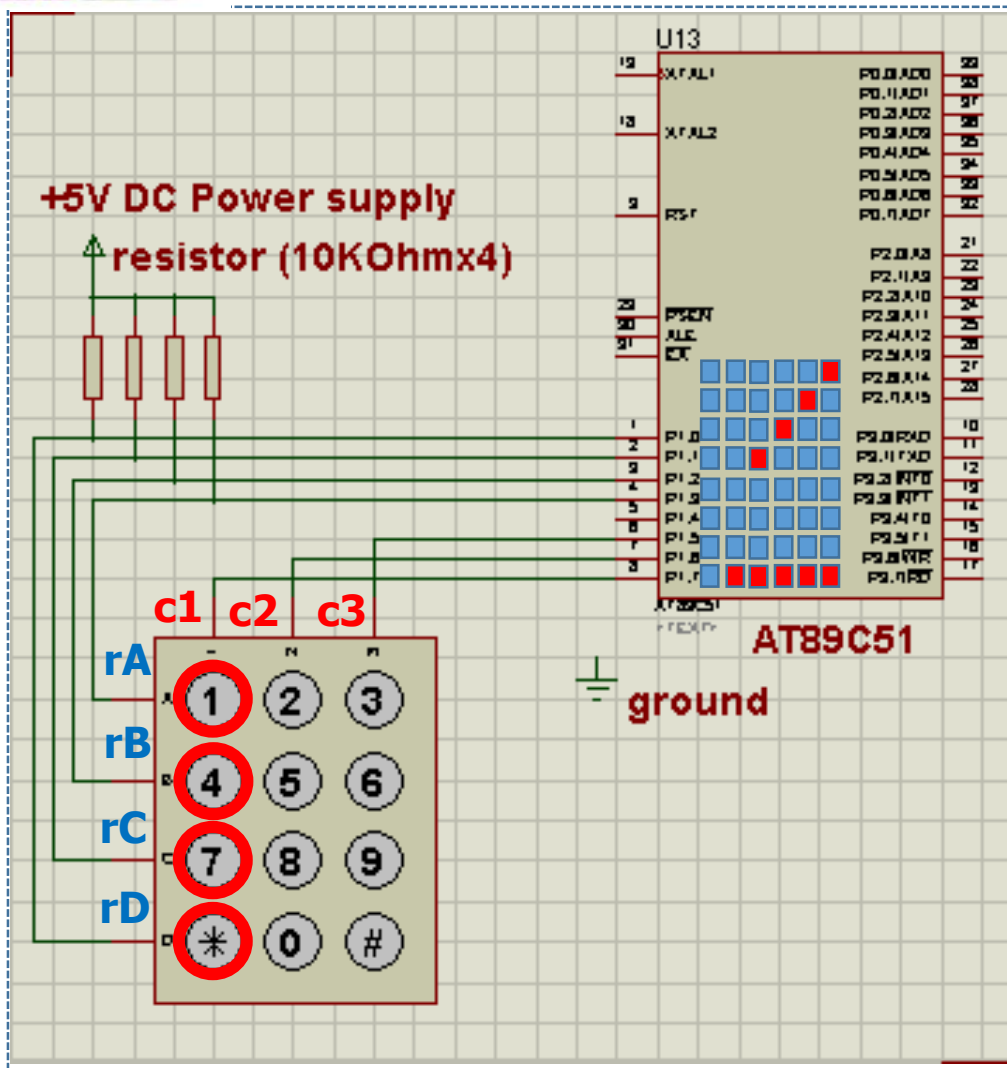




6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



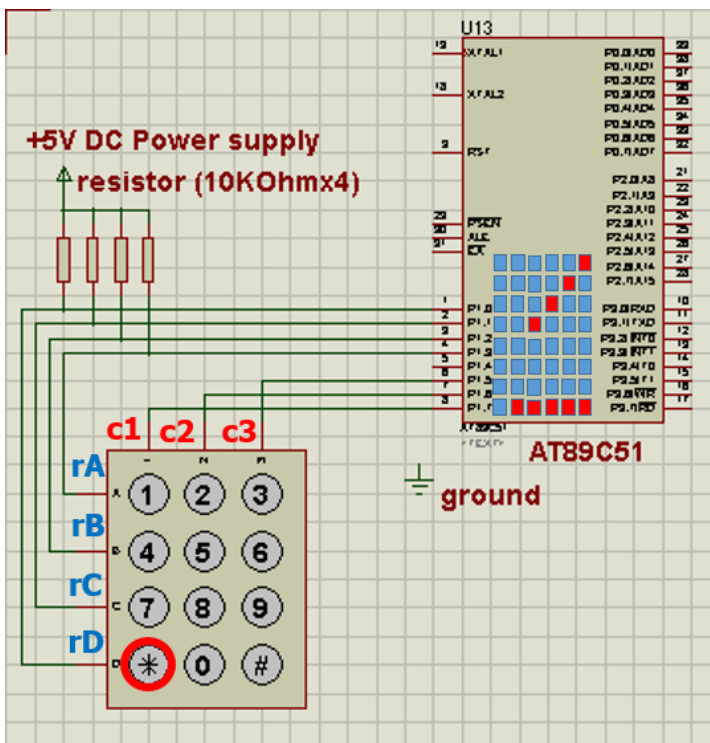
c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
0	1	1	1	1	1	1	1	0x7F	คอลัมน์ที่ 1
0	1	1	1	0	1	1	1	0x77	กดปุ่ม '1'
0	1	1	1	1	0	1	1	0x7B	กดปุ่ม '4'
0	1	1	1	1	1	0	1	0x7D	กดปุ่ม '7'
0	1	1	1	1	1	1	0	0x7E	กดปุ่ม '*'



6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
0	1	1	1	1	1	1	1	0x7F	คอลัมน์ที่ 1
0	1	1	1	0	1	1	1	0x77	กดปุ่ม '1'
0	1	1	1	1	0	1	1	0x7B	กดปุ่ม '4'
0	1	1	1	1	1	0	1	0x7D	กดปุ่ม '7'
0	1	1	1	1	1	1	0	0x7E	กดปุ่ม '*'

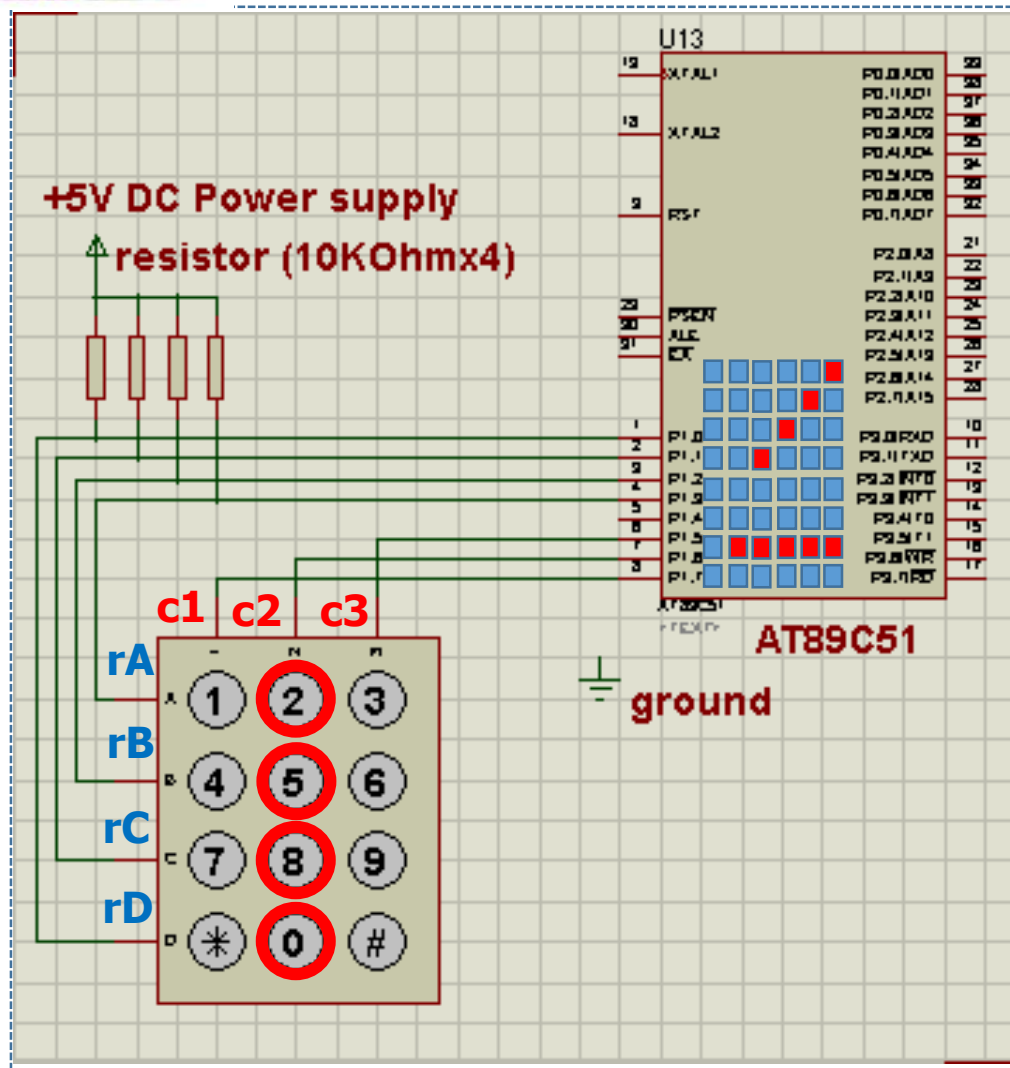
```
#include<reg51.h>
int x;
void main ()
{
    P1=0xFF; P2=0xFF;
    while(1)
    {
        P1=0x7F; x=P1;
        switch(x)
        {
            Case 0x77:
            Case 0x7B:
            Case 0x7D:
            Case 0x7E:
        }
    }
}
```



6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



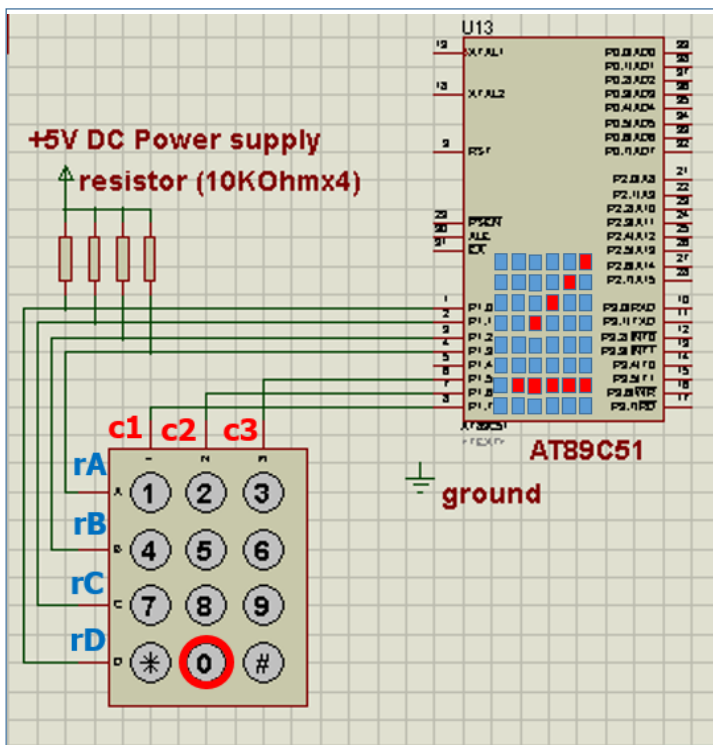
c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
1	0	1	1	1	1	1	1	0xBF	คอลัมน์ที่ 2
1	0	1	1	0	1	1	1	0xB7	กดปุ่ม '2'
1	0	1	1	1	0	1	1	0xBB	กดปุ่ม '5'
1	0	1	1	1	1	0	1	0xBD	กดปุ่ม '8'
1	0	1	1	1	1	1	0	0xBE	กดปุ่ม '0'



6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
1	0	1	1	1	1	1	1	0xBF	คอลัมน์ที่ 2
1	0	1	1	0	1	1	1	0xB7	กดปุ่ม '2'
1	0	1	1	1	0	1	1	0xBB	กดปุ่ม '5'
1	0	1	1	1	1	0	1	0xBD	กดปุ่ม '8'
1	0	1	1	1	1	1	0	0xBE	กดปุ่ม '0'

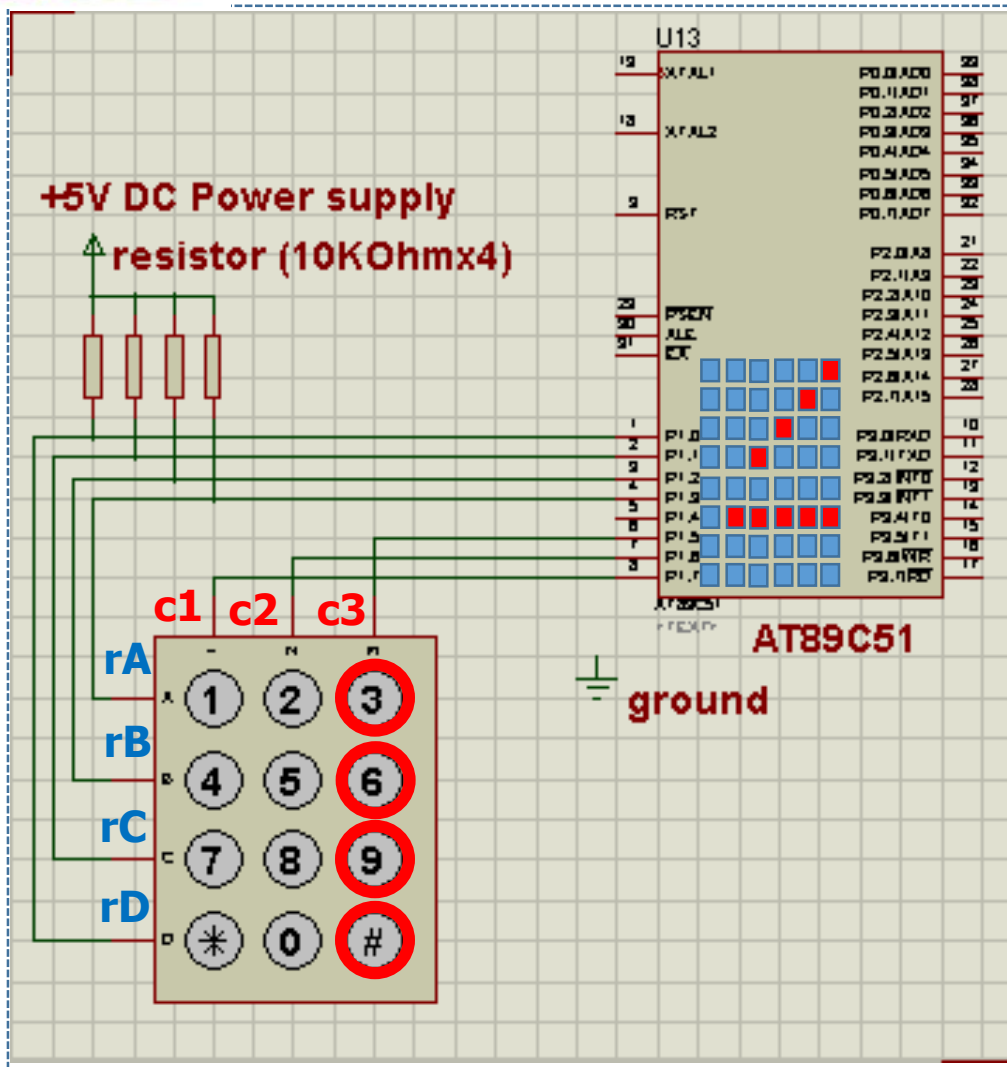
```
#include<reg51.h>
int x;
void main ()
{
    P1=0xFF; P2=0xFF;
    while(1)
    {
        P1=0xBF; x=P1;
        switch(x)
        {
            Case 0xB7:
            Case 0xBB:
            Case 0xBD:
            Case 0xBE:
        }
    }
}
```



6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



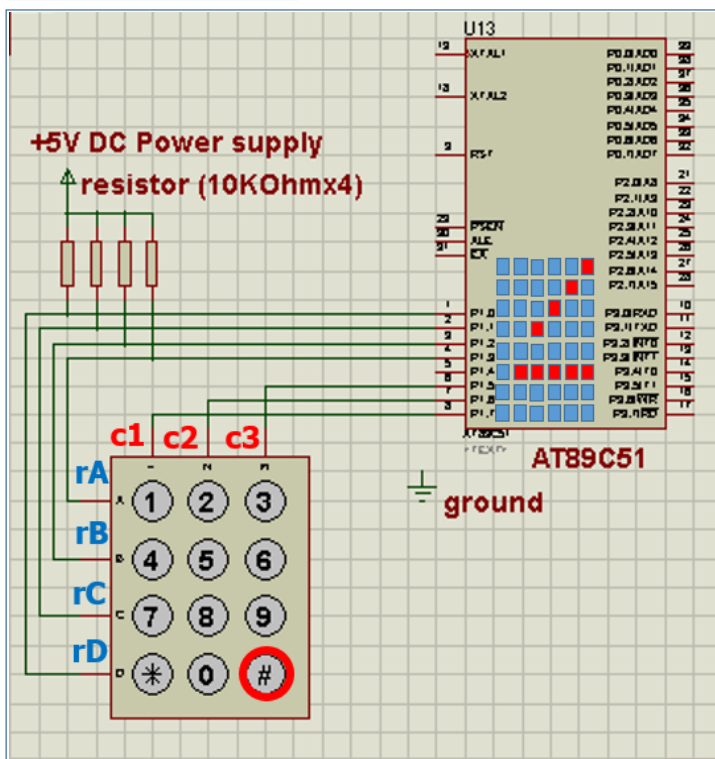
c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
1	1	0	1	1	1	1	1	0xDF	คอลัมน์ที่ 3
1	1	0	1	0	1	1	1	0xD7	กดปุ่ม '3'
1	1	0	1	1	0	1	1	0xDB	กดปุ่ม '6'
1	1	0	1	1	1	0	1	0xDD	กดปุ่ม '9'
1	1	0	1	1	1	1	0	0xDE	กดปุ่ม '#'



6.4 Matrix switch

ตัวอย่าง Keyphone Pad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
1	1	0	1	1	1	1	1	0xDF	คอลัมน์ที่ 3
1	1	0	1	0	1	1	1	0xD7	กดปุ่ม '3'
1	1	0	1	1	0	1	1	0xDB	กดปุ่ม '6'
1	1	0	1	1	1	0	1	0xDD	กดปุ่ม '9'
1	1	0	1	1	1	1	0	0xDE	กดปุ่ม '#'

```
#include<reg51.h>
```

```
int x;
```

```
void main ()
```

```
{ P1=0xFF; P2=0xFF;
```

```
while(1)
```

```
{ P1=0xDF; x=P1;
```

```
switch(x)
```

```
{ Case 0xD7:
```

```
Case 0xDB:
```

```
Case 0xDD:
```

```
Case 0xDE:
```

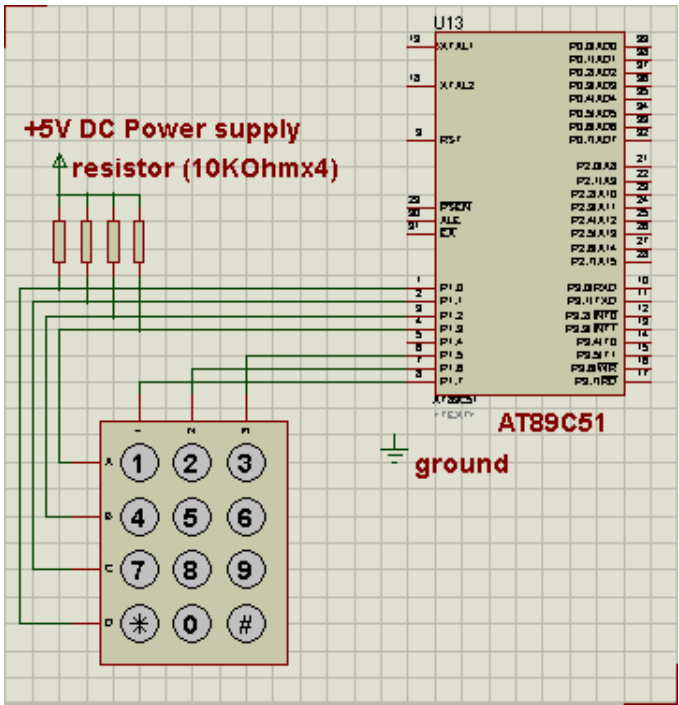
```
}
```

```
}
```

```
}
```


c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
0	1	1	1	1	1	1	1	0x7F	คอลัมน์ที่ 1
0	1	1	1	0	1	1	1	0x77	กดปุ่ม '1'
0	1	1	1	1	0	1	1	0x7B	กดปุ่ม '4'
0	1	1	1	1	1	0	1	0x7D	กดปุ่ม '7'
0	1	1	1	1	1	1	0	0x7E	กดปุ่ม '*'

c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
1	0	1	1	1	1	1	1	0xBF	คอลัมน์ที่ 2
1	0	1	1	0	1	1	1	0xB7	กดปุ่ม '2'
1	0	1	1	1	0	1	1	0xBB	กดปุ่ม '5'
1	0	1	1	1	1	0	1	0xBD	กดปุ่ม '8'
1	0	1	1	1	1	1	0	0xBE	กดปุ่ม '0'



c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
1	1	0	1	1	1	1	1	0xDF	คอลัมน์ที่ 3
1	1	0	1	0	1	1	1	0xD7	กดปุ่ม '3'
1	1	0	1	1	0	1	1	0xDB	กดปุ่ม '6'
1	1	0	1	1	1	0	1	0xDD	กดปุ่ม '9'
1	1	0	1	1	1	1	0	0xDE	กดปุ่ม '#'

```

#include<reg51.h>
int x;
void main ()
{P1=0xFF;P2=0xFF;P3=0x00;
while(1)
{P1=0x7F;x=P1;
switch(x)
{case 0x77: {P3=P1;break;}
case 0x7B: {P3=P1;break;}
case 0x7D: {P3=P1;break;}
case 0x7E: {P3=P1;break;}
}
P1=0xBF;x=P1;
switch(x)
{case 0xB7: {P3=P1;break;}
case 0xBB: {P3=P1;break;}
case 0xBD: {P3=P1;break;}
case 0xBE: {P3=P1;break;}
}
}

```

```

P1=0xDF;x=P1;
switch(x)
{case 0xD7: {P3=P1;break;}
case 0xDB: {P3=P1;break;}
case 0xDD: {P3=P1;break;}
case 0xDE: {P3=P1;break;}
}
P1=0xEF;x=P1;
switch(x)
{case 0xE7: {P3=P1;break;}
case 0xEB: {P3=P1;break;}
case 0xED: {P3=P1;break;}
case 0xEE: {P3=P1;break;}
}
}
}

```

```

#include<reg51.h>
int x;
void main ()
{P1=0xFF;P2=0xFF;P3=0x00;
while(1)
{P1=0x7F;x=P1;
switch(x)
{case 0x77: {P3=P1;P2=0xF8;break;}
case 0x7B: {P3=P1;P2=0x99;break;}
case 0x7D: {P3=P1;P2=0xF9;break;}
case 0x7E: {P3=P1;P2=0xDC;break;}
}
P1=0xBF;x=P1;
switch(x)
{case 0xB7: {P3=P1;P2=0x80;break;}
case 0xBB: {P3=P1;P2=0x92;break;}
case 0xBD: {P3=P1;P2=0xA4;break;}
case 0xBE: {P3=P1;P2=0xC0;break;}
}
}

```

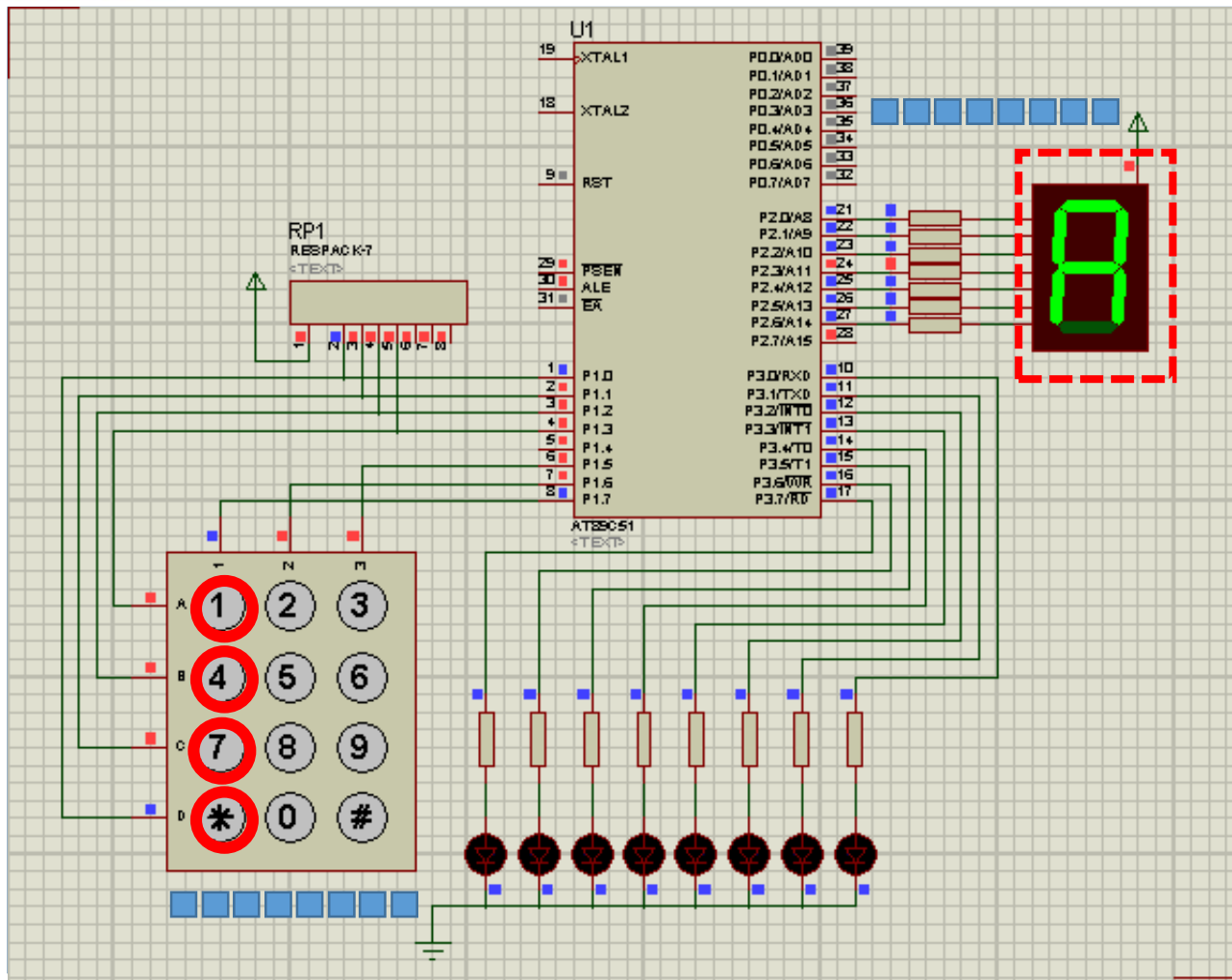
```

P1=0xDF;x=P1;
switch(x)
{case 0xD7: {P3=P1;P2=0x90;break;}
case 0xDB: {P3=P1;P2=0x82;break;}
case 0xDD: {P3=P1;P2=0xB0;break;}
case 0xDE: {P3=P1;P2=0xBE;break;}
}
P1=0xEF;x=P1;
switch(x)
{case 0xE7: {P3=P1;P2=0xB6;break;}
case 0xEB: {P3=P1;P2=0x89;break;}
case 0xED: {P3=P1;P2=0xBF;break;}
case 0xEE: {P3=P1;P2=0xE3;break;}
}
}
}

```




6.5 Matrix switch vs (8-LEDs & 7-segment display)



```
#include<reg51.h>
```

```
int x;
```

```
void main ()
```

```
{ P1=0xFF; P2=0xFF; P3=0x00;
```

```
while(1)
```

```
{ P1=0x7F; x=P1;
```

```
switch(x)
```

```
{ Case 0x77: {P2=0xF9;break;}
```

```
Case 0x7B: {P2=0x99;break;}
```

```
Case 0x7D: {P2=0xF8;break;}
```

```
Case 0x7E: {P2=0x88;break;}
```

```
}
```

```
}
```

```
}
```

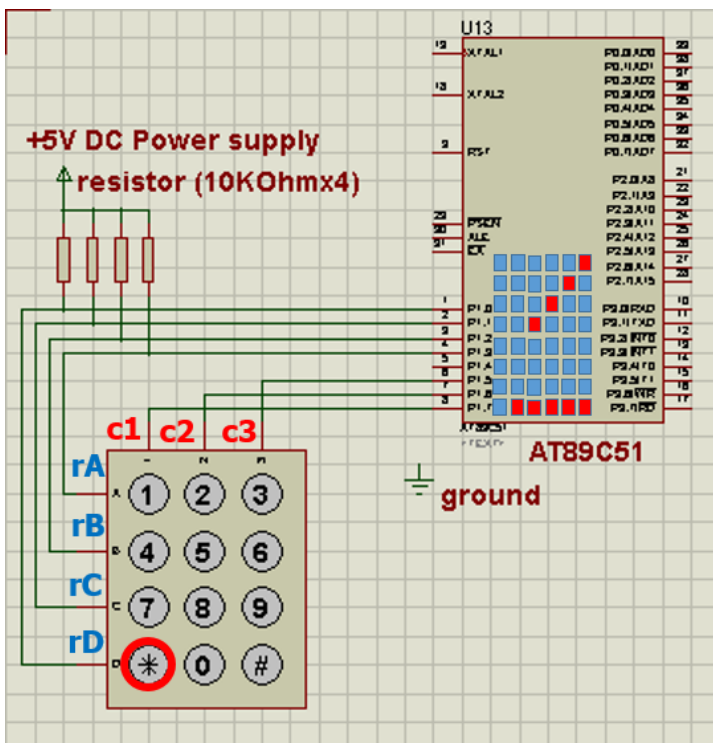



บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

3.8 ชุดคำสั่งเกี่ยวกับอินพุต(inputs)

ตัวอย่าง Keypad(4rows x 3columns)

ศึกษาสถานะการทำงานของสวิตช์ 12 ตัว



c1	c2	c3	X	rA	rB	rC	rD	hex.	
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	
1	1	1	1	1	1	1	1	0xFF	สถานะเริ่ม
0	1	1	1	1	1	1	1	0x7F	คอลัมน์ที่ 1
0	1	1	1	0	1	1	1	0x77	กดปุ่ม '1'
0	1	1	1	1	0	1	1	0x7B	กดปุ่ม '4'
0	1	1	1	1	1	0	1	0x7D	กดปุ่ม '7'
0	1	1	1	1	1	1	0	0x7E	กดปุ่ม '*'

```
#include<reg51.h>
```

```
int x;
```

```
void main ()
```

```
{ P1=0xFF; P2=0xFF;
```

```
while(1)
```

```
{ P1=0x7F; x=P1;
```

```
switch(x)
```

```
{ Case 0x77: {P2=0xC0;break;}
```

```
Case 0x7B: {P2=0xF9;break;}
```

```
Case 0x7D: {P2=0xA4;break;}
```

```
Case 0x7E: {P2=0x99;break;}
```

```
}
```

```
}
```

```
}
```

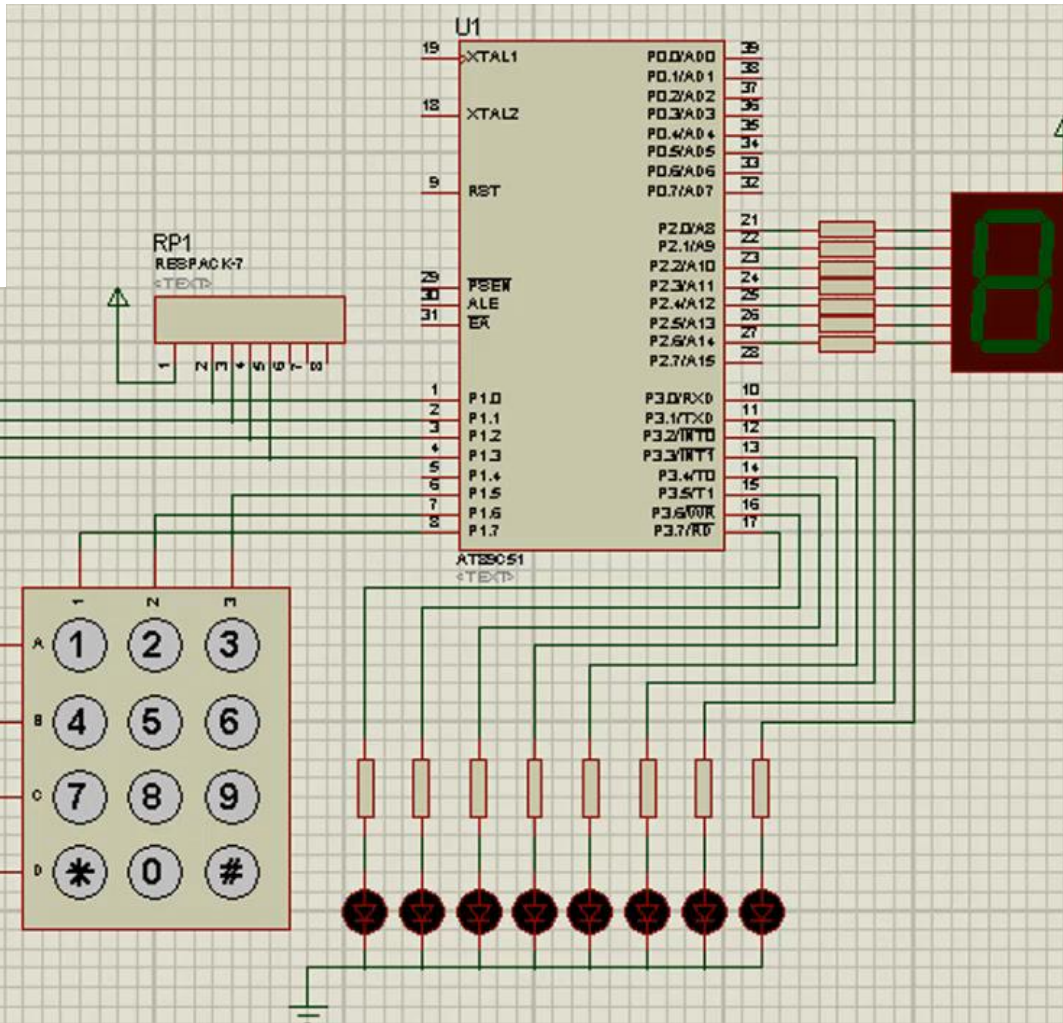


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

3.8 ชุดคำสั่งเกี่ยวกับอินพุต(inputs)

ศึกษาสถานการณ์การทำงานของสวิตช์ 12 ตัว

ตัวอย่าง Keypad(4rows x 3columns)



```

01 #include<reg51.h>
02 int c1,c2,c3;
03 void delay(unsigned int t);
04 void main ()
05 {P1=0xFF;P2=0xFF;P3=0x00;
06 while(1)
07 {P1=0x7F;c1=P1;
08 switch(c1)
09 {case 0x77: {P3=c1;P2=0xF9;delay(100);break;} //display_1
10 case 0x7B: {P3=c1;P2=0x99;delay(100);break;} //display_4
11 case 0x7D: {P3=c1;P2=0xF8;delay(100);break;} //display_7
12 case 0x7E: {P3=c1;P2=0x88;delay(100);break;} ///display_'A'
13 }
14 P1=0xBF;c2=P1;
15 switch(c2)
16 {case 0xB7: {P3=c2;P2=0xA4;delay(100);break;} //display_2
17 case 0xBB: {P3=c2;P2=0x92;delay(100);break;} //display_5
18 case 0xBD: {P3=c2;P2=0x80;delay(100);break;} //display_8
19 case 0xBE: {P3=c2;P2=0xC0;delay(100);break;} //display_9
20 }
21 P1=0xDF;c3=P1;
22 switch(c3)
23 {case 0xD7: {P3=c3;P2=0xB0;delay(100);break;} //display_3
24 case 0xDB: {P3=c3;P2=0x82;delay(100);break;} //display_6
25 case 0xDD: {P3=c3;P2=0x90;delay(100);break;} //display_9
26 case 0xDE: {P3=c3;P2=0x86;delay(100);break;} //display_'E'
27 default: {P3=0x00;P2=0xFF;}
28 } } }
29 void delay(unsigned int t)
30 {for(t=0;t<1500;t++);}

```




บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.6 (ก) ตัวอย่างโปรแกรมเกี่ยวกับการขัดจังหวะ(Interrupt: int.0)

ไมโครคอนโทรลเลอร์ : MCS-51

เมื่อต่อเชื่อมกับอุปกรณ์ภายนอก ระหว่างการทำงาน มีระบบตรวจสอบอุปกรณ์อินพุตหลายตัว ตลอดเวลา จึงต้องเสียเวลาไป ทุกรอบ(Polling method)

เพื่อลดการเสียเวลา ใน MCS-51 จัดให้มีขา int (P3.2, P3.3) ให้อุปกรณ์อินพุตใช้ติดต่อโดยตรง เมื่อมีการขัดจังหวะ (interrupt)

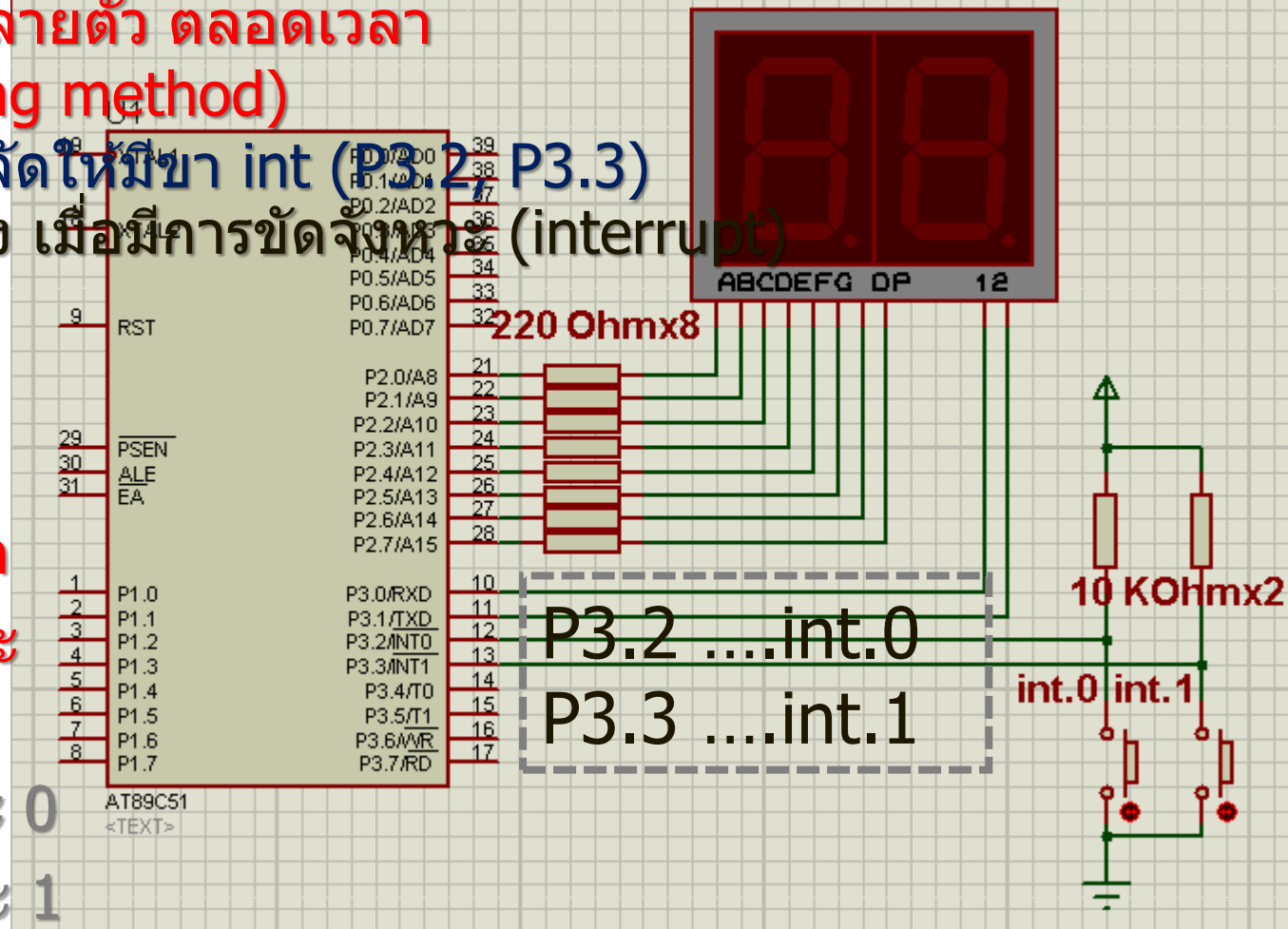
นิยาม การขัดจังหวะ (Interrupt)

กระบวนการ

ขณะที่ระบบทำงานในโปรแกรมหลัก หากมีการขัดจังหวะ เช่น กดสวิทช์ที่ขา int.0 ระบบจะพักงานในโปรแกรมหลัก และ กระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ (Interrupt Service Routine)

Void ext_int0() interrupt 0 //การขัดจังหวะ 0

Void ext_int1() interrupt 2 //การขัดจังหวะ 1





บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.6 (ก) ตัวอย่างโปรแกรมเกี่ยวกับการขัดจังหวะ(Interrupt: int.0)

```
#include<reg51.h>
```

X=0, 0xC0

```
Char hex2seg[]={0xC0,0xF9,0xA4,0xB0,0x99,0x82,0x58,0x80};
```

```
int x; //ตัวชี้ตำแหน่งในตัวแปรชุด hex2seg[]
```

```
Sbit d0=P3^0; //หลักแสดงผลขวาสุด
```

```
Void delay(int t);
```

```
Void ext_int0(); //การขัดจังหวะโดยสวิตซ์ที่ขา P3.1
```

```
void main ()
```

```
{ IT0=1; EX0=1; EA=1; //รูปแบบการขัดจังหวะ
```

```
while(1)
```

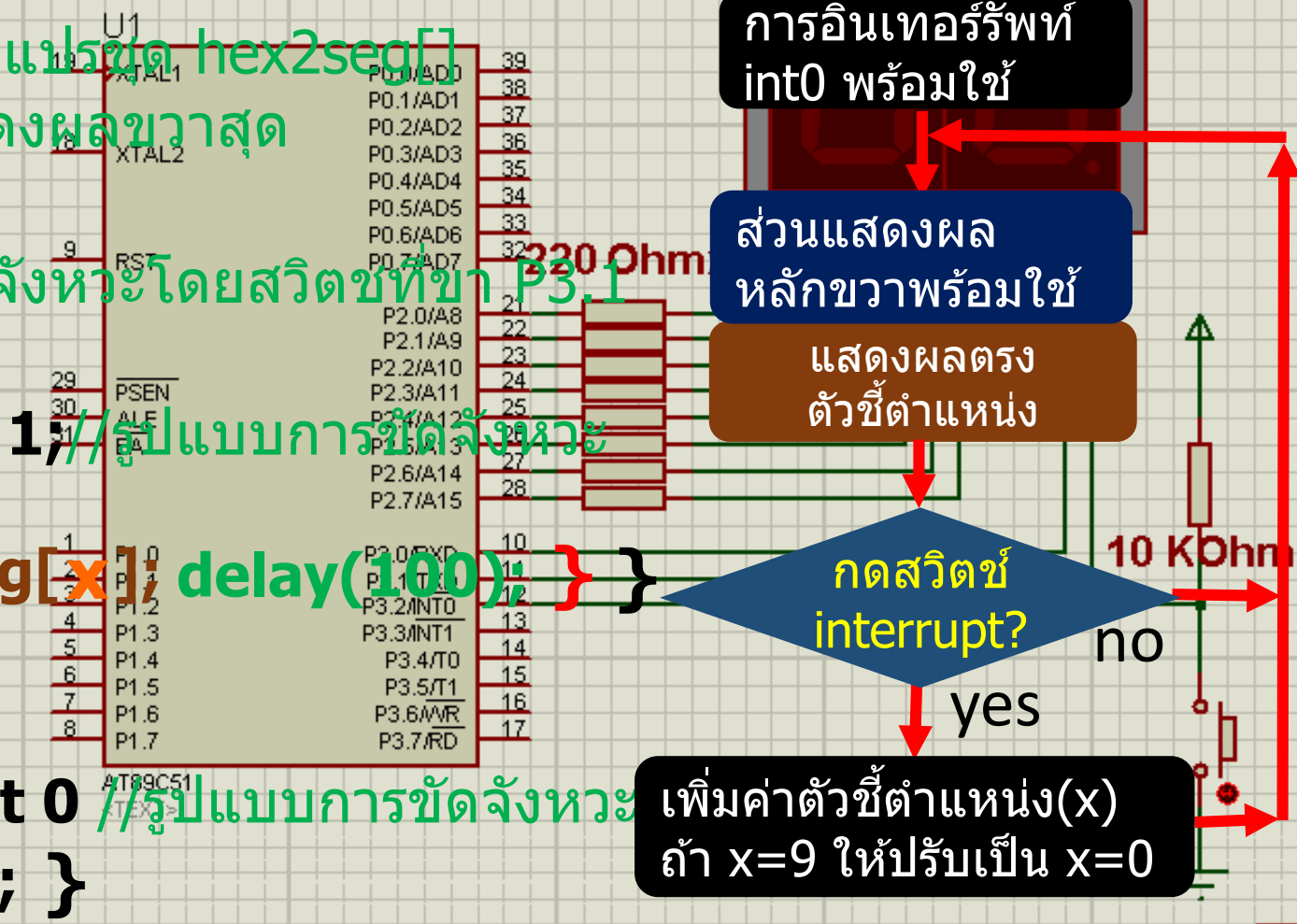
```
{ d0=0; P2=hex2seg[x]; delay(100); }
```

```
Void delay(int t)
```

```
{ for t=t; t>0; t--; }
```

```
Void ext_int0() interrupt 0 //รูปแบบการขัดจังหวะ
```

```
{ x++; if(x>9) {x=0}; }
```





6.6 (ก) ตัวอย่างโปรแกรมเกี่ยวกับการขัดจังหวะ(Interrupt: int.0)

```
#include<reg51.h>
```

```
Char hex2seg[]={0xC0,0xF9,0xA4,0xB0,0x99,0x82,0xF8,0x80};
```

```
int x;
```

```
Sbit d0=P3^0;
```

```
Void delay(int t);
```

```
Void ext_int0();
```

```
void main ()
```

```
{ IT0=1; EX0=1; EA=1;
```

```
while(1)
```

```
{ d0=0; P2=hex2seg[x]; delay(100); } }
```

```
Void delay(int t)
```

```
{ for t=t; t>0; t--; }
```

```
Void ext_int0() interrupt 0
```

```
{ x++; if(x>9) {x=0}; }
```





บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.6 (ก) ตัวอย่างโปรแกรมเกี่ยวกับการขัดจังหวะ(Interrupt: int.0)

```
#include<reg51.h>
Char hex2seg[]={0xC0,0xF9,0xA4,0xB0,0x99,0x82,0xF8,0x80};
int x;
Sbit d0=P3^0;
Void delay(int t);
Void ext_int0();
void main ()
{ IT0=1; EX0=1; EA=1;
  while(1)
  { d0=0;   P2=hex2seg[x];
    Void delay(int t)
    { for t=t; t>0; t--; }
    Void ext_int0() interrupt 0
    { x++; if(x>9) {x=0};}
```

X=0, 0xC0

X=2, 0xA4

X=4, 0x99

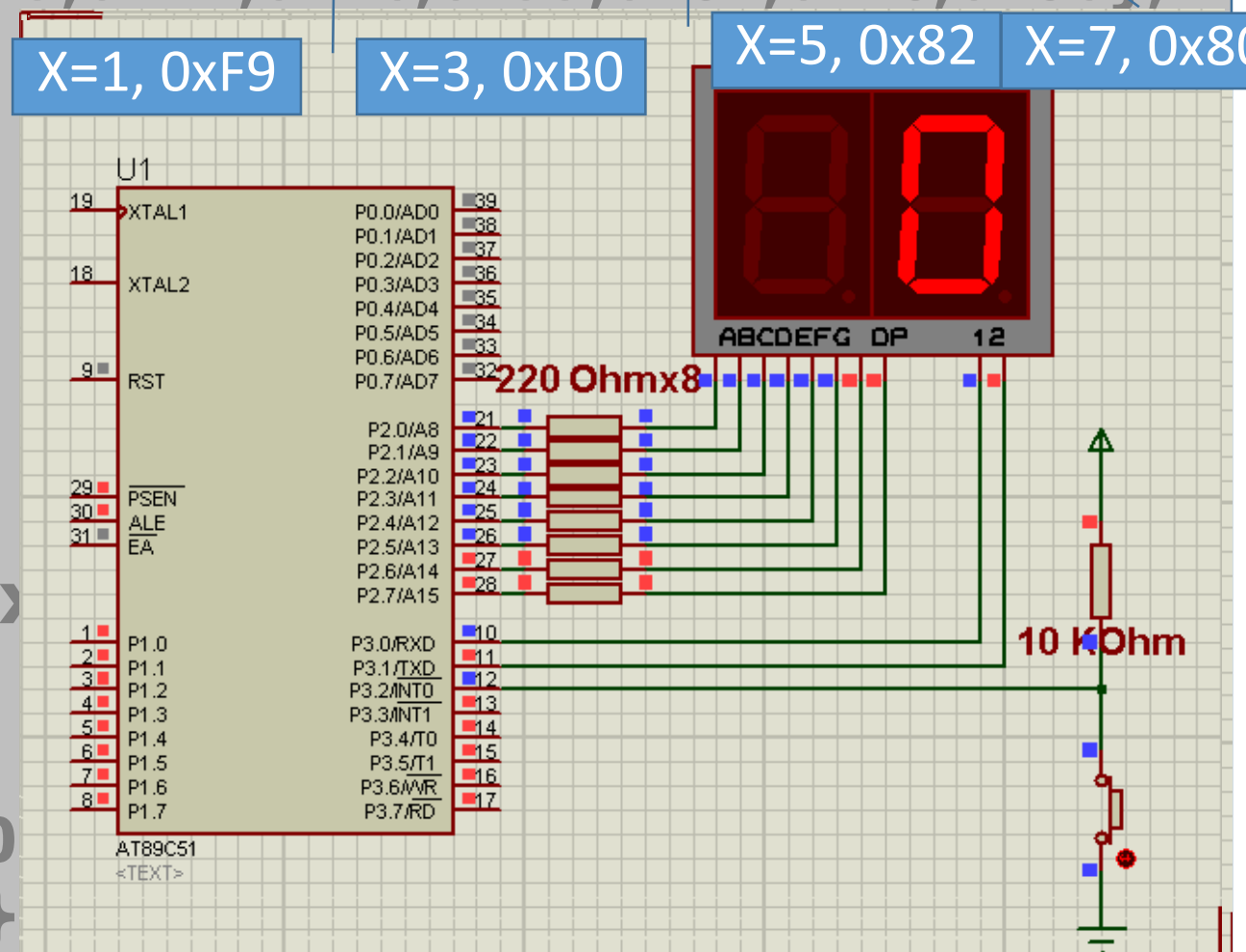
X=6, 0xF8

X=1, 0xF9

X=3, 0xB0

X=5, 0x82

X=7, 0x80





6.6 (ก) ตัวอย่างโปรแกรมเกี่ยวกับการขัดจังหวะ(Interrupt: int.0)

```
#include<reg51.h>
```

```
Char hex2seg[]={0xC0,0xF9,0xA4,0xB0,0x99,0x82,0xF8,0x80};
```

```
int x;
```

```
//ตัวชี้ตำแหน่งในตัวแปรชุด hex2seg[]
```

```
Sbit d0=P3^0; //หลักแสดงผลขวาสุด
```

```
Void delay(int t);
```

```
Void ext_int0(); //การขัดจังหวะโดยสวิตซ์ที่ขา P3.1
```

```
void main ()
```

```
{ IT0=1; EX0=1; EA=1; //รูปแบบการขัดจังหวะ
```

```
while(1)
```

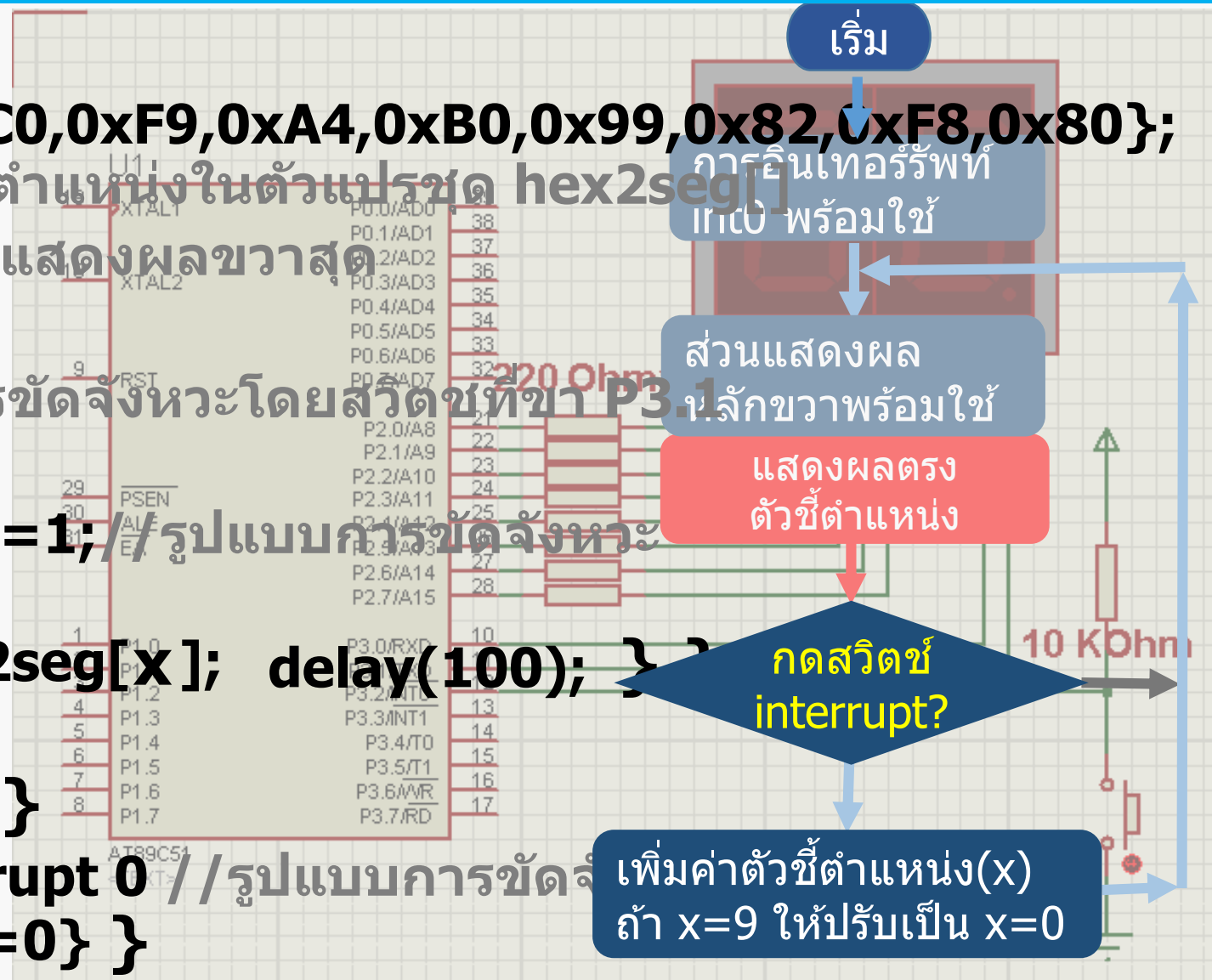
```
{ d0=0; P2=hex2seg[x]; delay(100); }
```

```
Void delay(int t)
```

```
{ for t=t; t>0; t--; }
```

```
Void ext_int0() interrupt 0 //รูปแบบการขัดจังหวะ
```

```
{ x++; if(x>9) {x=0} }
```





บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

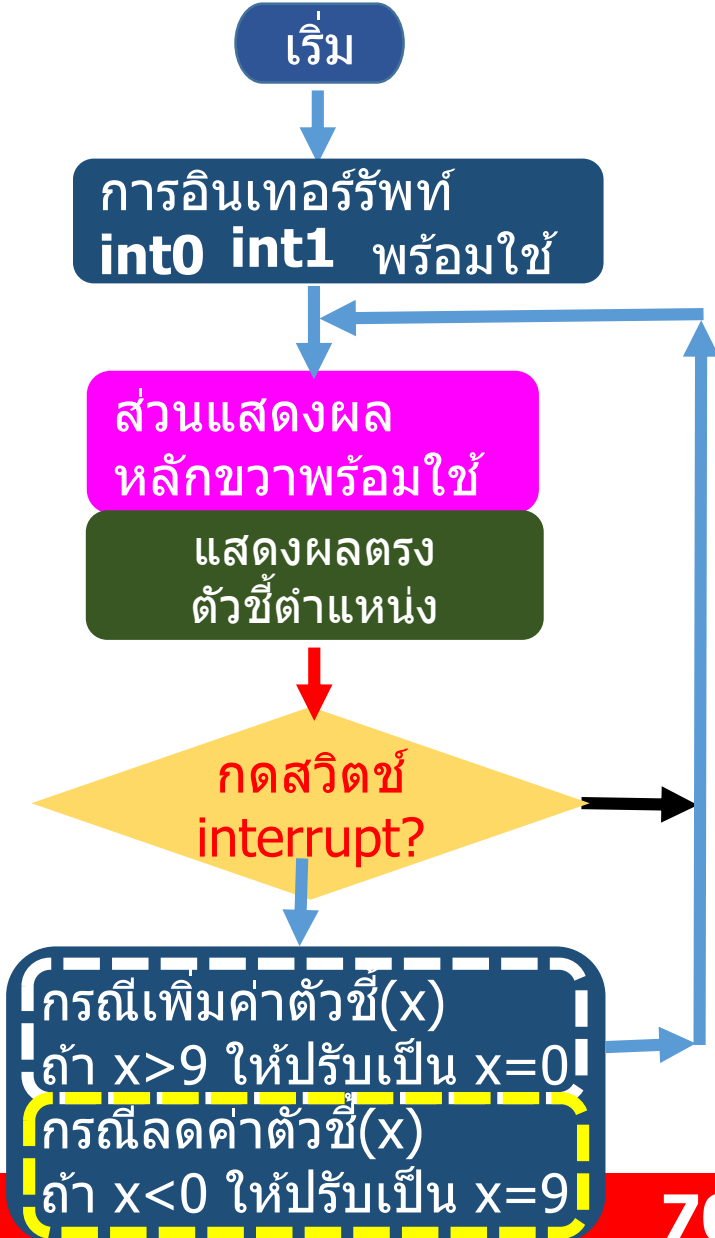
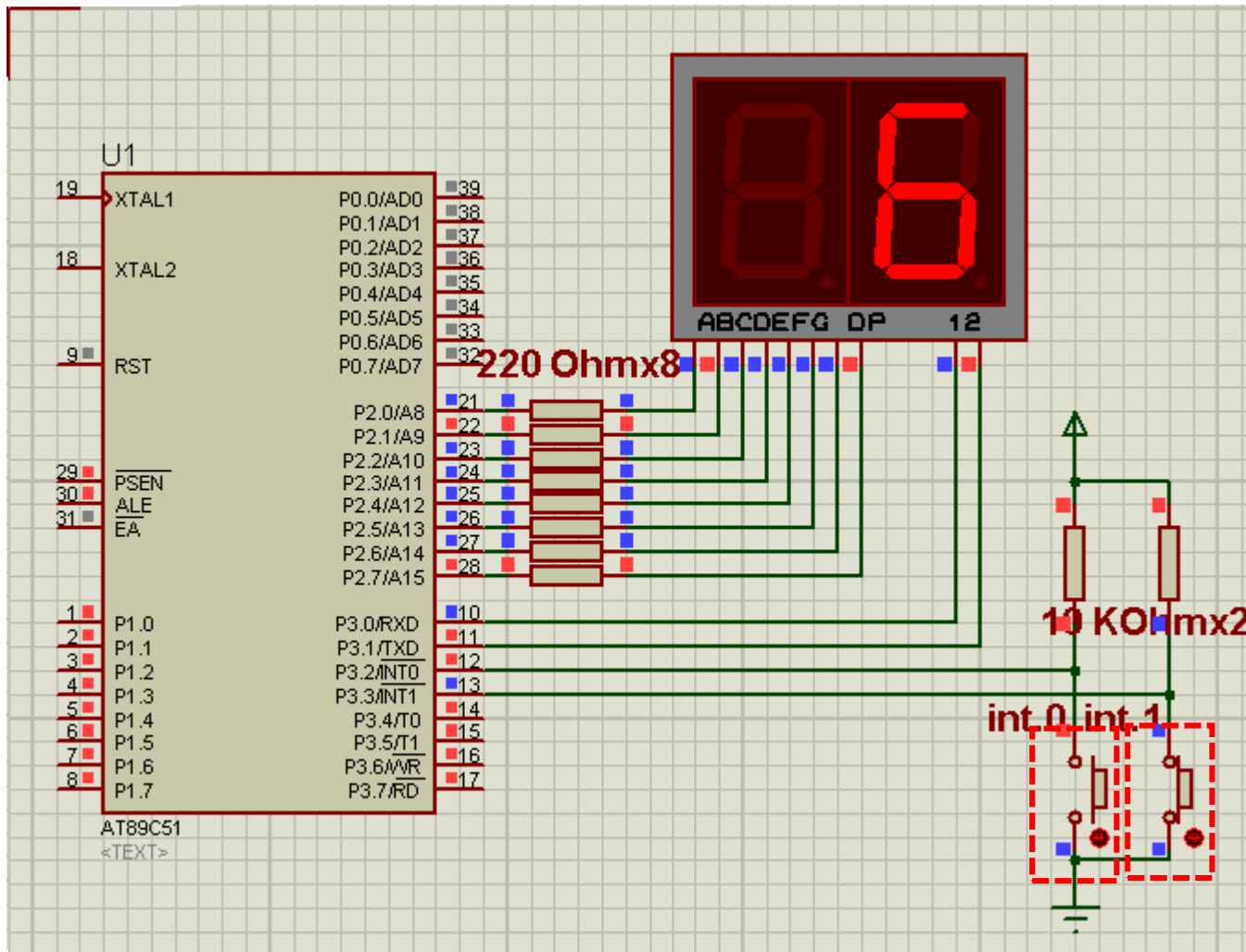
6.6 ตัวอย่างโปรแกรมเกี่ยวกับการขัดจังหวะ (Interrupt)





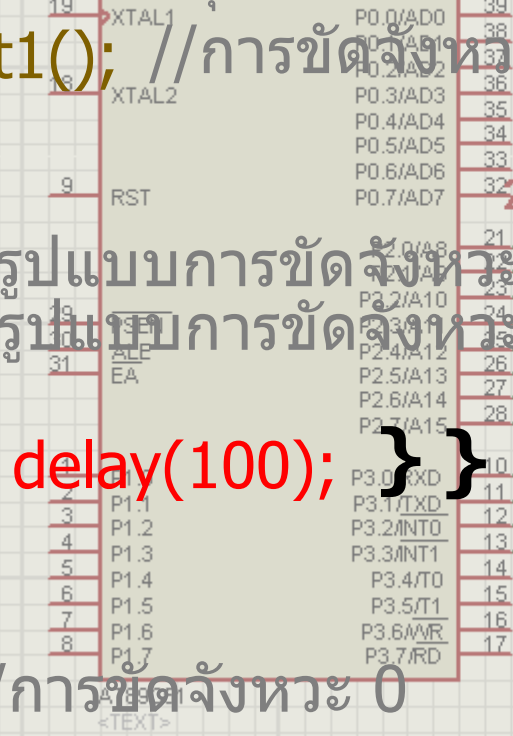
บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

6.6 (ข) ตัวอย่างโปรแกรมเกี่ยวกับการขัดจังหวะ(Interrupt: int.0 & int.1)





```
#include<reg51.h>
Char hex2seg[]={0xC0,0xF9,0xA4,0xB0,0x99,0x82,0xF8,0x80,0x90,0x8F};
int x; //ตัวชี้ตำแหน่งในตัวแปรชุด hex2seg[]
Sbit d0=P3^0; //หลักแสดงผลขวาสุด
Void ext_int0(); Void ext_int1(); //การขัดจังหวะโดย
Void delay(int t);
void main ()
{IT0=1; EX0=1; //รูปแบบการขัดจังหวะ
IT1=1; EX1=1; EA=1; //รูปแบบการขัดจังหวะ
while(1)
{d0=0; P2=hex2seg[x]; delay(100); }
Void delay(int t)
{ for t=t; t>0; t--; }
Void ext_int0() interrupt 0 //การขัดจังหวะ 0
{ x++; if(x>9) {x=0} }
Void ext_int1() interrupt 2 //การขัดจังหวะ 1
{ x--; if(x<0) {x=9} }
```



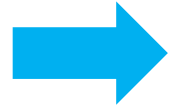


บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

เมื่อจบบทเรียนนี้... นักศึกษาควรมีสมรรถนะดังนี้



6.1 เข้าใจและสามารถพัฒนา
งานควบคุม
ดีซีมอเตอร์
หลายรูปแบบ



6.2 เข้าใจและ
สามารถพัฒนางาน
ควบคุมการ
ติด-ดับของดวงไฟ/
ไฟวิ่ง/ไฟจราจร/
ส่วนแสดงผล 7 ส่วน



6.3 เข้าใจและสามารถ
พัฒนางานเกี่ยวกับ
ทางเลือก (condition) ด้วย
คำสั่ง case/ if condition



6.6 เข้าใจหลักการ
และสามารถพัฒนา
งานที่เกี่ยวข้องกับ
การขัดจังหวะ
(Interrupt)
จากภายนอกชิป



6.5 เข้าใจหลักการและ
สามารถพัฒนางานที่มี
Matrix switch
เป็นอุปกรณ์อินพุต
และมี **8-LEDs & 7-
segment display**
เป็นอุปกรณ์เอาต์พุต



6.4 เข้าใจหลักการและ
สามารถพัฒนางานที่มี
Matrix switch
(3x4)/ (4x4)
เป็นอุปกรณ์อินพุต



มหาวิทยาลัย
ราชภัฏนครปฐม