



มหาวิทยาลัย
ราชภัฏนครปฐม



บทเรียนออนไลน์

รายวิชา 6562213 ไมโครโปรเซสเซอร์ 1
6562213 Microprocessor 1

โดย

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

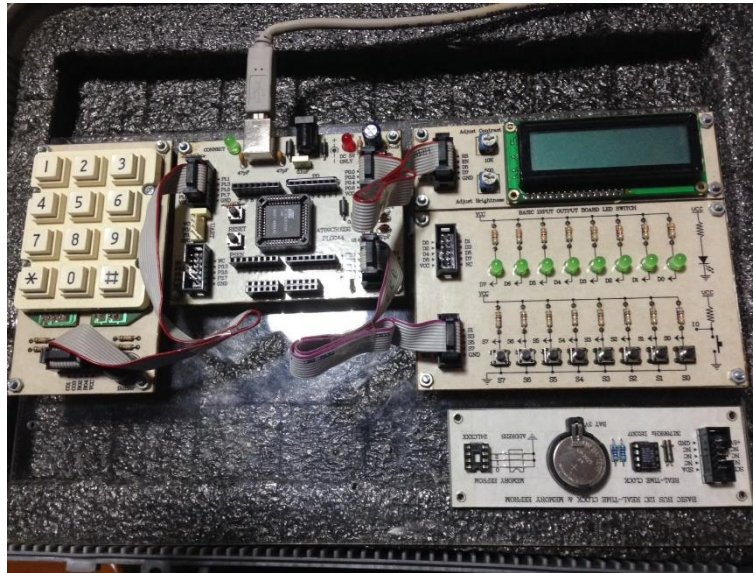


จุดประสงค์การเรียนรู้ของรายวิชา 6562213 ไมโครโพรเซสเซอร์ 1



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



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น



รายวิชา 6562213 ไมโครโปรเซสเซอร์ 1

บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

โดยผู้ช่วยศาสตราจารย์ธีระ กาญจนสินธุ์

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

บทนำ



ในบทนี้ มีตัวอย่างงานควบคุมแบบพื้นฐาน จากผังความคิดในการพัฒนางานควบคุม-การจำลอง วงจรอิเล็กทรอนิกส์-การเขียนโปรแกรมควบคุมแบบง่ายๆ เริ่มตั้งแต่การติดสว่างของหลอด LED 1 ดวง ถึง หลายดวง/ การแสดงผลของส่วนแสดงผล 7 ส่วน/ การใช้สวิตช์ 1 ตัว ถึง 8 ตัวควบคุมการติด-ดับของ หลอด LED/ ส่วนแสดงผล 7 ส่วน / การทำงานของ มอเตอร์



การนำเสนอผังความคิดนำไปสู่ฮาร์ดแวร์ที่จำลองได้บน ISIS Professional Proteus และซอฟต์แวร์ซึ่งพัฒนาได้บน Keil μ Vision เมื่อนำมาประกอบกันโดย ลำดับ นับเป็นกรณีศึกษาที่ช่วยพัฒนาความเข้าใจในไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้หลากหลาย ผู้เรียนจึงจำเป็นต้องสังเกตความแตกต่าง และปรับความ เข้าใจไปในแต่ละกรณี



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

หัวข้อสำคัญในบทที่ 4



4.1 วิธีการ
พัฒนางาน
ควบคุมบน
คอมพิวเตอร์

4.2 ตัวอย่าง 1:
on-off 1-LED
(byte control)

4.3 ตัวอย่าง 2 :
on-off 4-LEDs
(shift >> control)

4.4 ตัวอย่าง 3 :
on-off 5-LEDs
(byte control)

4.5 ตัวอย่าง 4 :
ส่วนแสดงผล 7 ส่วน
แบบ CA (byte control)

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

4.6 (ข) ตัวอย่าง 6:
1-button switch
& 1-DC motor
(bit control)

4.7 ตัวอย่าง 7:
3-button
switches
& 3-LEDs
(byte control)

4.8 ตัวอย่าง 8:
4-button
switches &
7-segment
(byte control)

4.9 ตัวอย่าง 9:
matrix (3x4)
switches



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.1 วิธีการพัฒนางานควบคุมบนคอมพิวเตอร์

ผังความคิด

สร้างวงจรบน Proteus

พัฒนาโปรแกรมบน Keil



เริ่ม

ดวงไฟพร้อมใช้

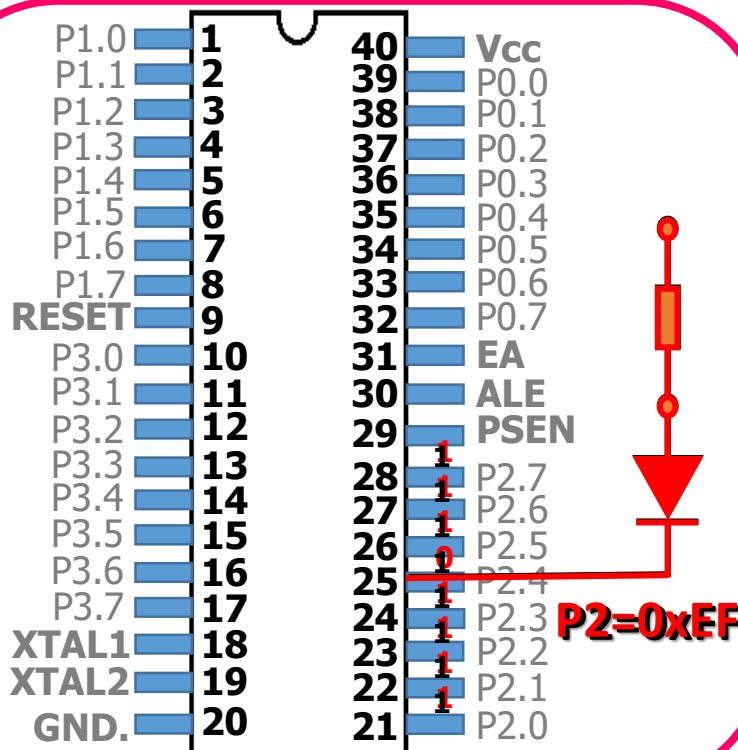
เปิดไฟ

หน่วงเวลา

ปิดไฟ

หน่วงเวลา

end



เปิด/ปิด ดวงไฟ 1 ดวง

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	Hex.
1	1	1	0	1	1	1	1	0xEF

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

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

```
void main ()
```

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

```
while(1)
```

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

```
P2=0xFF; delay(100);}
```

```
}
```

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

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




บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.2 ตัวอย่าง 1 : on-off 1-LED (byte control)

ผังความคิด

สร้างวงจรบน Proteus

พัฒนาโปรแกรมบน Keil



เริ่ม

ดวงไฟพร้อมใช้

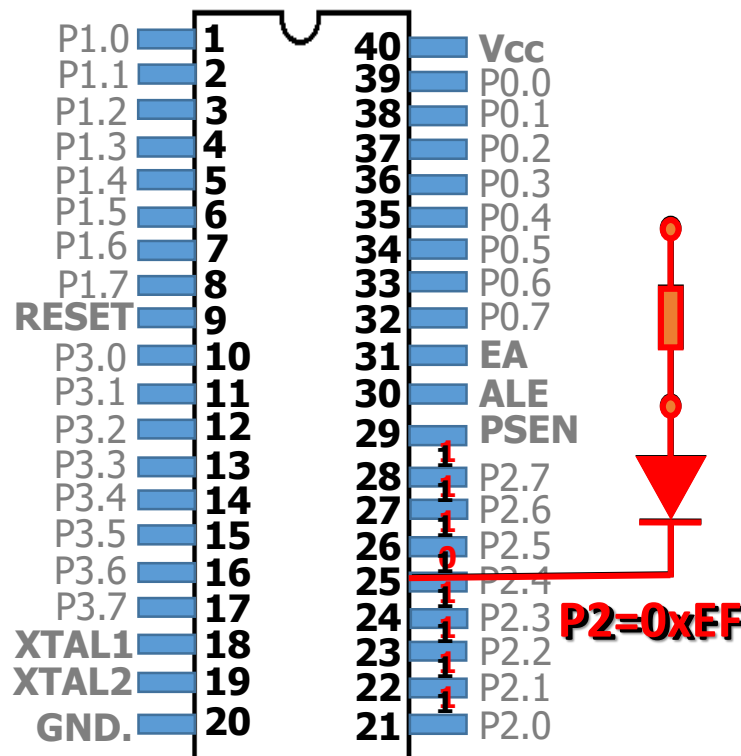
เปิดไฟ

หน่วงเวลา

ปิดไฟ

หน่วงเวลา

end



เปิด/ปิด ดวงไฟ 1 ดวง

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	Hex.
1	1	1	0	1	1	1	1	0xEF

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

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

```
void main ()
```

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

```
while(1)
```

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

```
P2=0xFF; delay(100);}
```

```
}
```

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

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




บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.3 ตัวอย่าง 2 : on-off 4-LEDs (shift bit >> control)

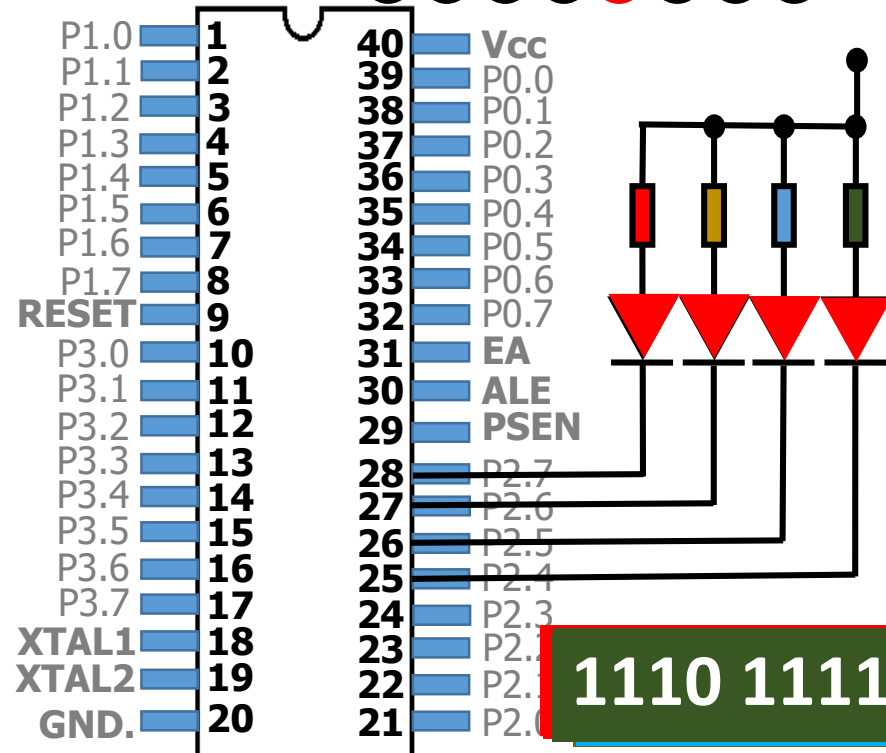
สร้างวงจรบน Proteus

ผังความคิด

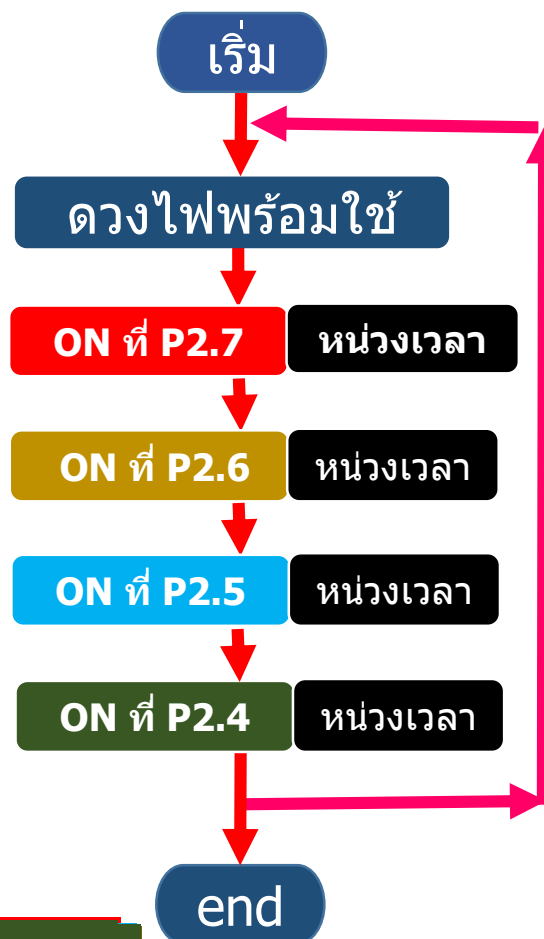
พัฒนาโปรแกรมบน Keil



●	●	●	●	●	●	●	●	0x80
●	●	●	●	●	●	●	●	0x40
●	●	●	●	●	●	●	●	0x20
●	●	●	●	●	●	●	●	0x10
●	●	●	●	●	●	●	●	0x08



1110 1111 = 0xEF



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

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




บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.4 ตัวอย่าง 3:on-off 5-LEDs(byte control)

พัฒนาโปรแกรมบน Keil

สร้างวงจรบน Proteus

ผังความคิด

#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);

```
}
```

```
}
```

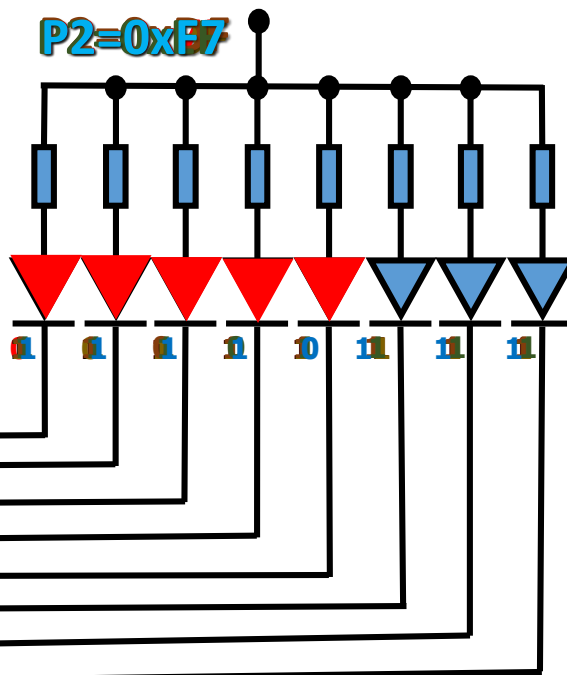
void delay (long t)

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



เปิด/ปิด ดวงไฟ 5 ดวง
วงจรดวงไฟใช้ port P2

P2=0xF7



เริ่ม

ดวงไฟพร้อมใช้

เปิดไฟที่ P2.7

หน่วงเวลา

เปิดไฟที่ P2.6

หน่วงเวลา

เปิดไฟที่ P2.5

หน่วงเวลา

เปิดไฟที่ P2.4

หน่วงเวลา

เปิดไฟที่ P2.3

หน่วงเวลา

end



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.5 ตัวอย่าง 4 : ส่วนแสดงผล 7 ส่วนแบบ CA (byte control)

สร้างวงจรบน Proteus

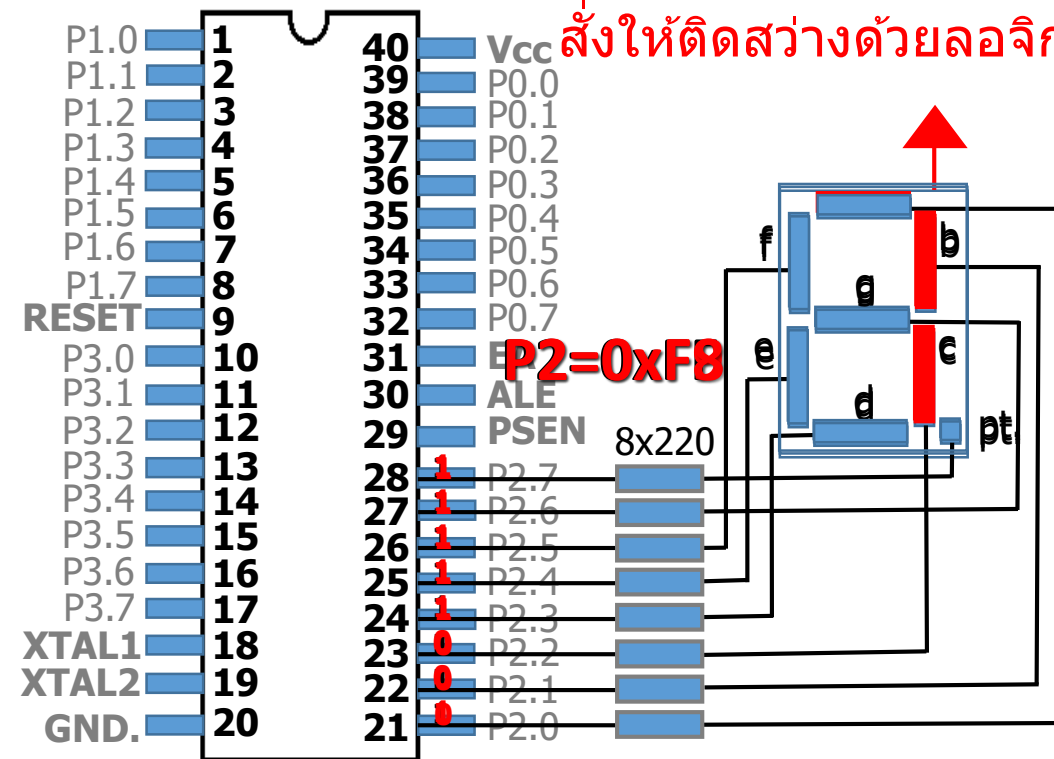
ฝังความคิด

พัฒนาโปรแกรมบน Keil



ส่วนแสดงผลแบบบวกร่วม

สั่งให้ติดสว่างด้วยลอจิก "0"



ส่วนแสดงผลพร้อมใช้

เปิดไฟขึ้นส่วน b,c
หน่วงเวลา

เปิดไฟขึ้นส่วน a,b,c
หน่วงเวลา

end

```
#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 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

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

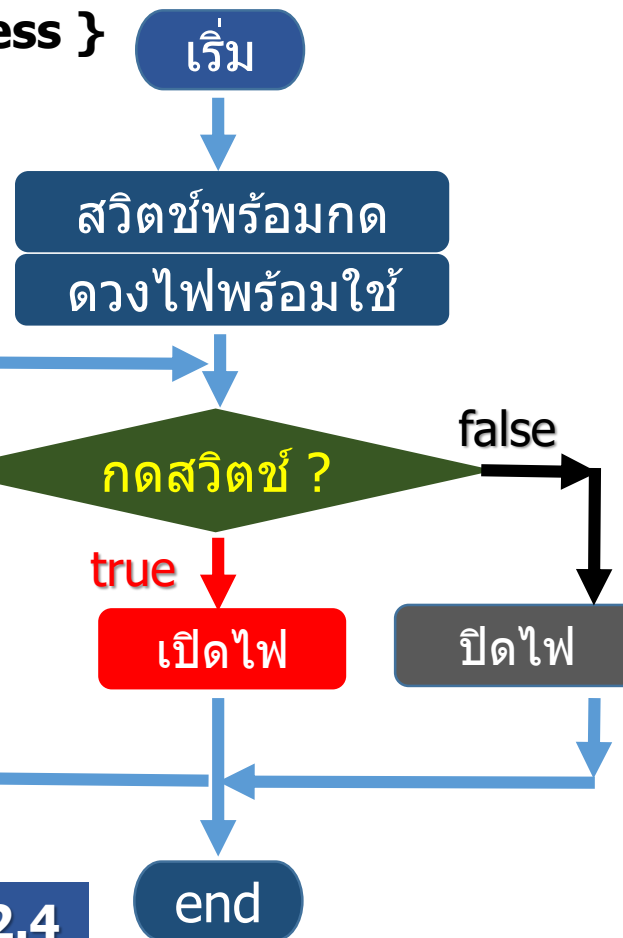
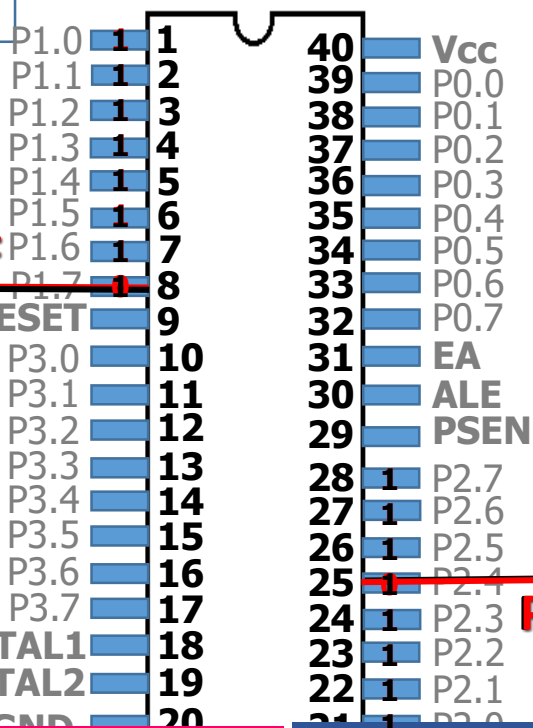
สร้างวงจรบน Proteus

ฝังความคิด

พัฒนาโปรแกรมบน Keil

รูปแบบคำสั่ง ทางเลือก (if condition)

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;}
```

```
}
```

```
}
```

วงจรสวิตช์ใช้ port P1.7

วงจรดวงไฟใช้ port P2.4



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.6(ข) ตัวอย่าง 6 : 1-button switch & 1-DC motor (bit control)

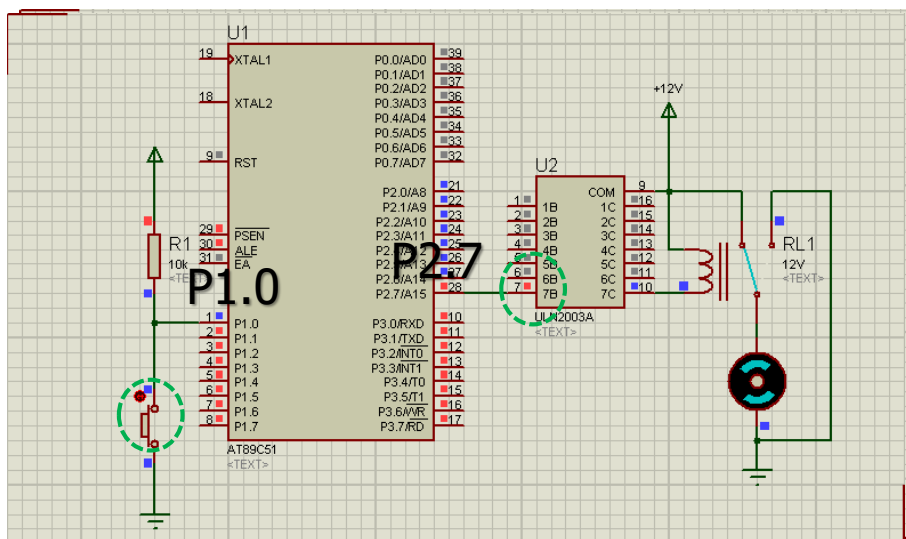
สร้างวงจรบน Proteus

ผังความคิด

พัฒนาโปรแกรมบน Keil

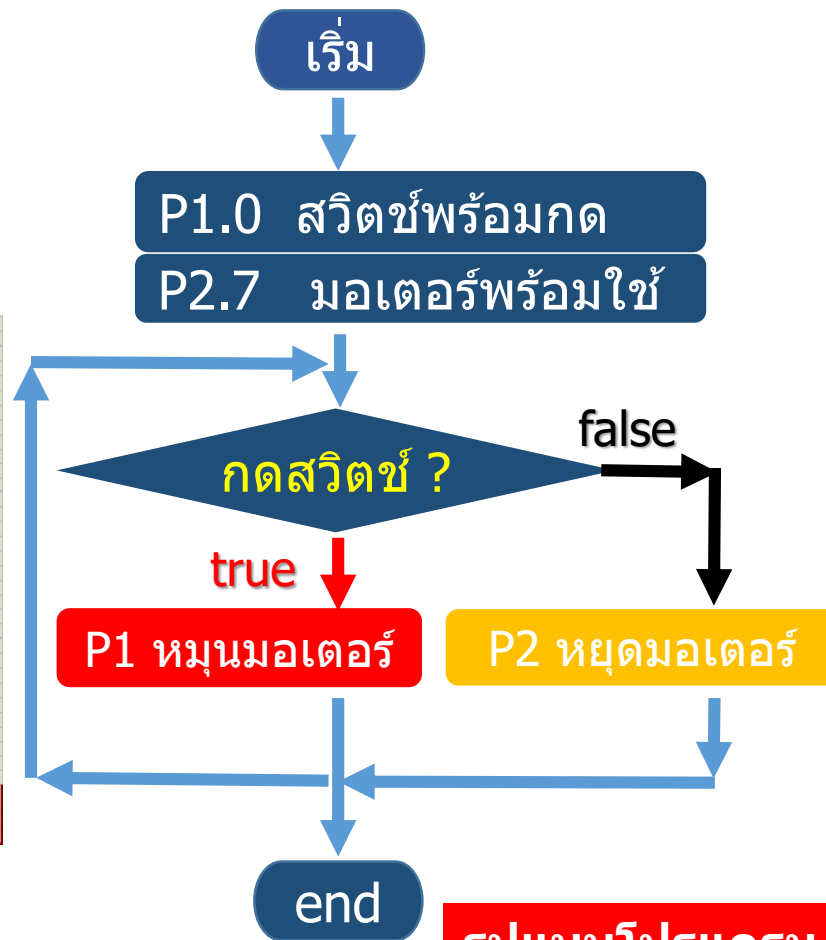
ตัวอย่างที่ใช้เงื่อนไข
(if) สองทางเลือก

สวิตช์เปิด/ปิด มอเตอร์



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 m7=P2^7;
void main ()
{ P1=0xFF; P2=0x00;
  while(1)
  { if(s0==0) {m7=1;}
    else {m7=0;}
  }
}
```




บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.7 ตัวอย่าง 7 : 3-button switches & 3-LEDs (byte control)

ผังความคิด

traffic light

พัฒนาโปรแกรมบน Keil



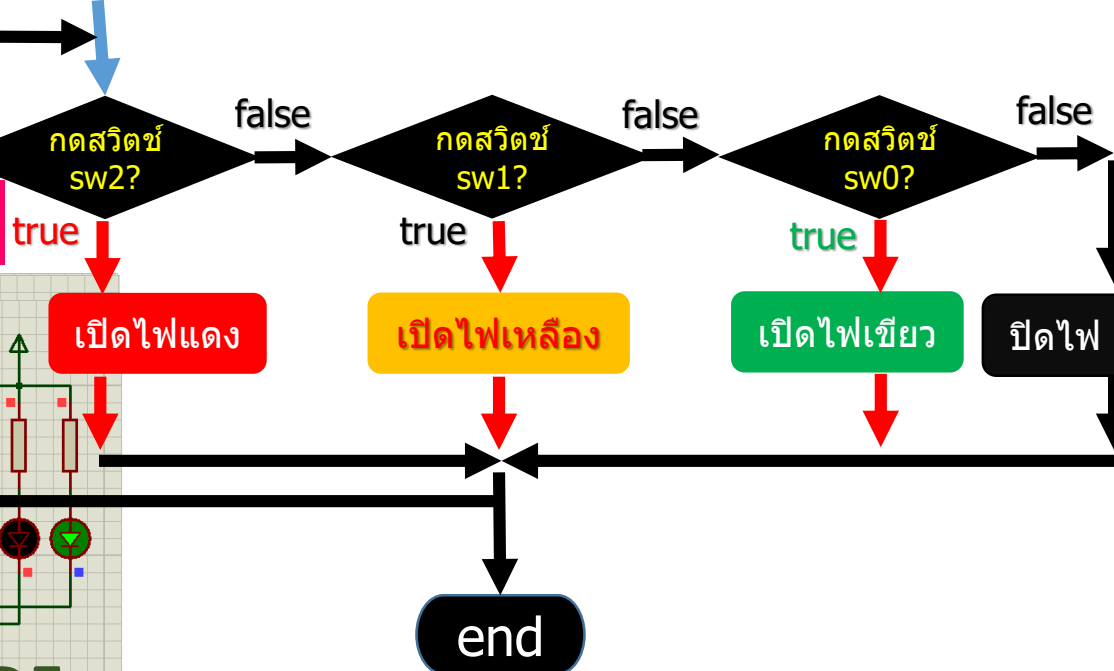
เริ่ม

Input : วงจรสวิตช์ใช้ port P1.2 P1.1 P1.0

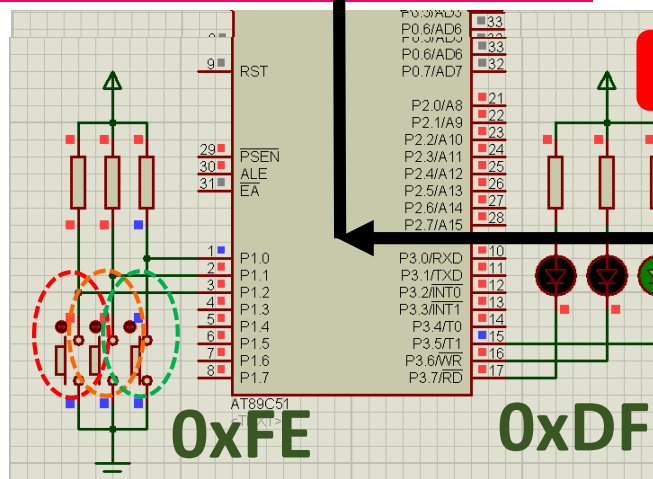
Output : วงจรดวงไฟใช้ port P3.7 P3.6 P3.5

สวิตช์พร้อมกด

ดวงไฟพร้อมใช้



สร้างวงจรบน Proteus



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

```
void main ()
```

```
{ Unsigned int xsw;
```

```
while(1)
```

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

```
if(xsw==0xFB){P3=0x7F;}
```

```
else if(xsw==0xFD){P3=0xBF;}
```

```
else if(xsw==0xFE){P3=0xDF;}
```

```
else {P3=0xFF;}
```

```
}
```

```
}
```

if condition

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



4.8 ตัวอย่าง 8: 4-button switches & 7-segment (byte control)

ผังความคิด

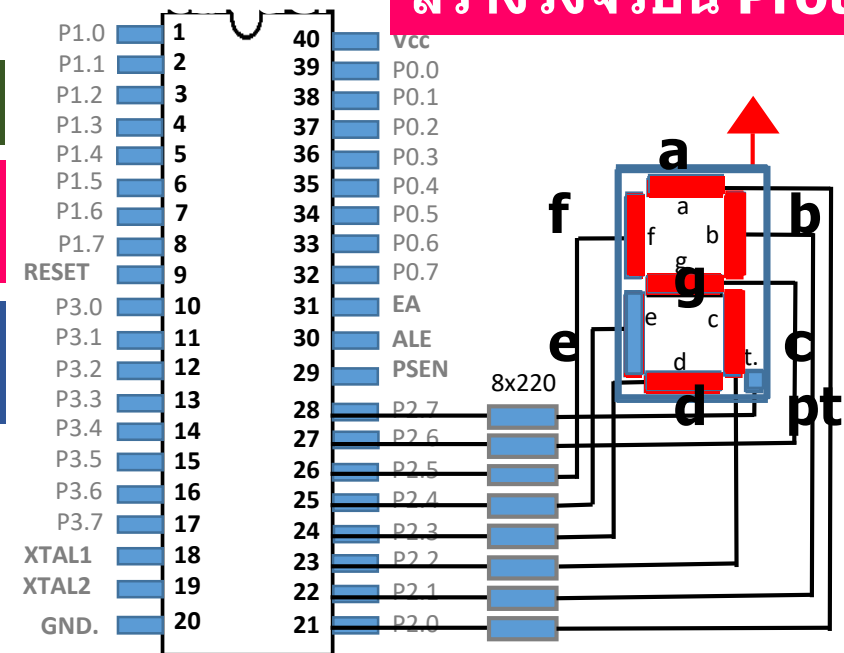
สวิตช์ 4 ตัวเปิด/ปิดควบคุมการแสดงผล 7 ส่วน ...0-9



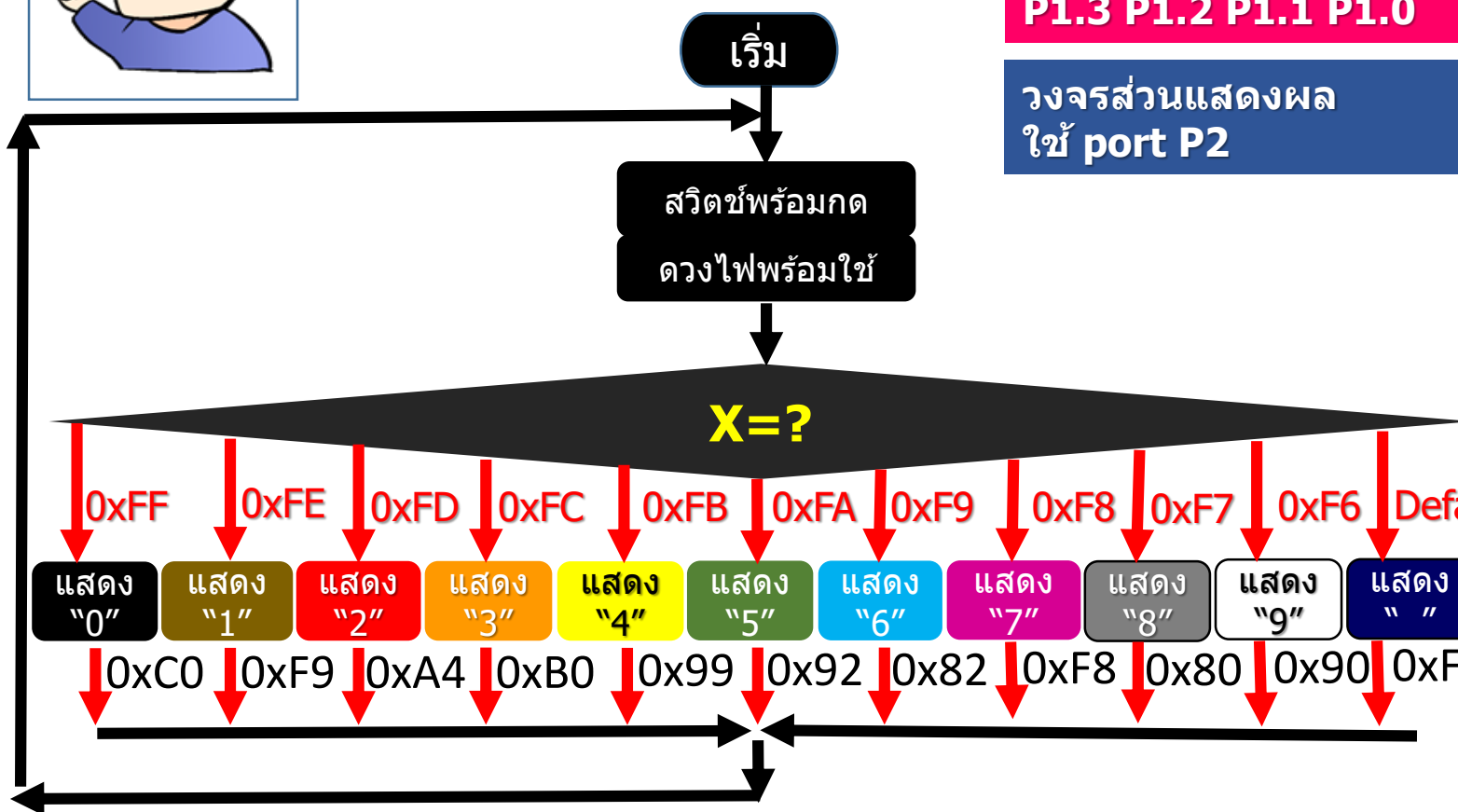
วงจรสวิตช์ใช้ port
P1.3 P1.2 P1.1 P1.0

วงจรส่วนแสดงผล
ใช้ port P2

สร้างวงจรบน Proteus



Active low : 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



4.8 ตัวอย่าง 8: 4-button switches & 7-segment

พัฒนาโปรแกรมบน Keil

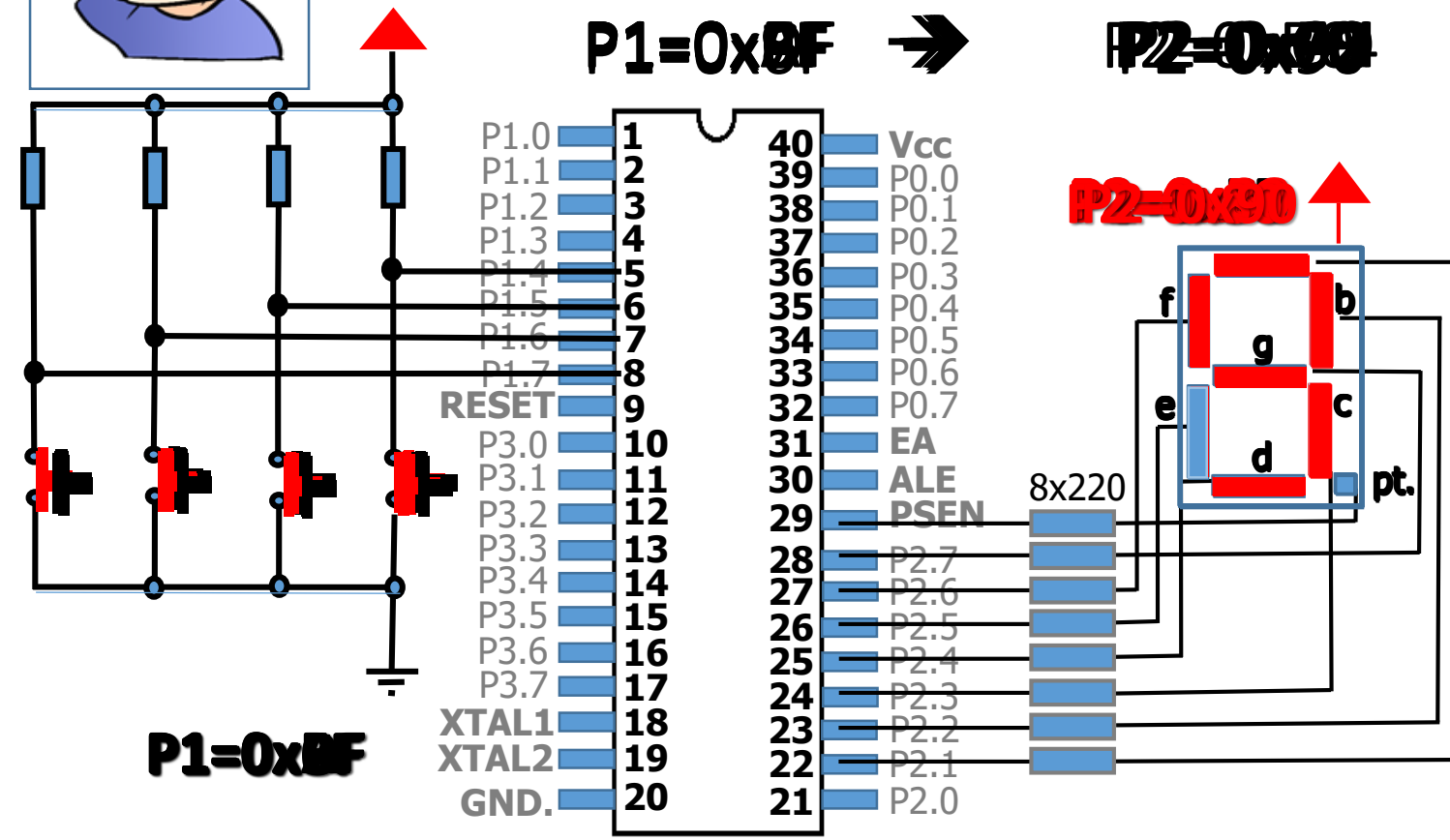
สร้างวงจรบน Proteus

ผังความคิด

สวิตช์ 4 ตัวเปิด/ปิดควบคุมการแสดงผล 7 ส่วน0-9

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

ส่วนแสดงผล port P3



```
#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;} }
```



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.8 ตัวอย่าง 8: 4-button switches & 7-segment (byte control)

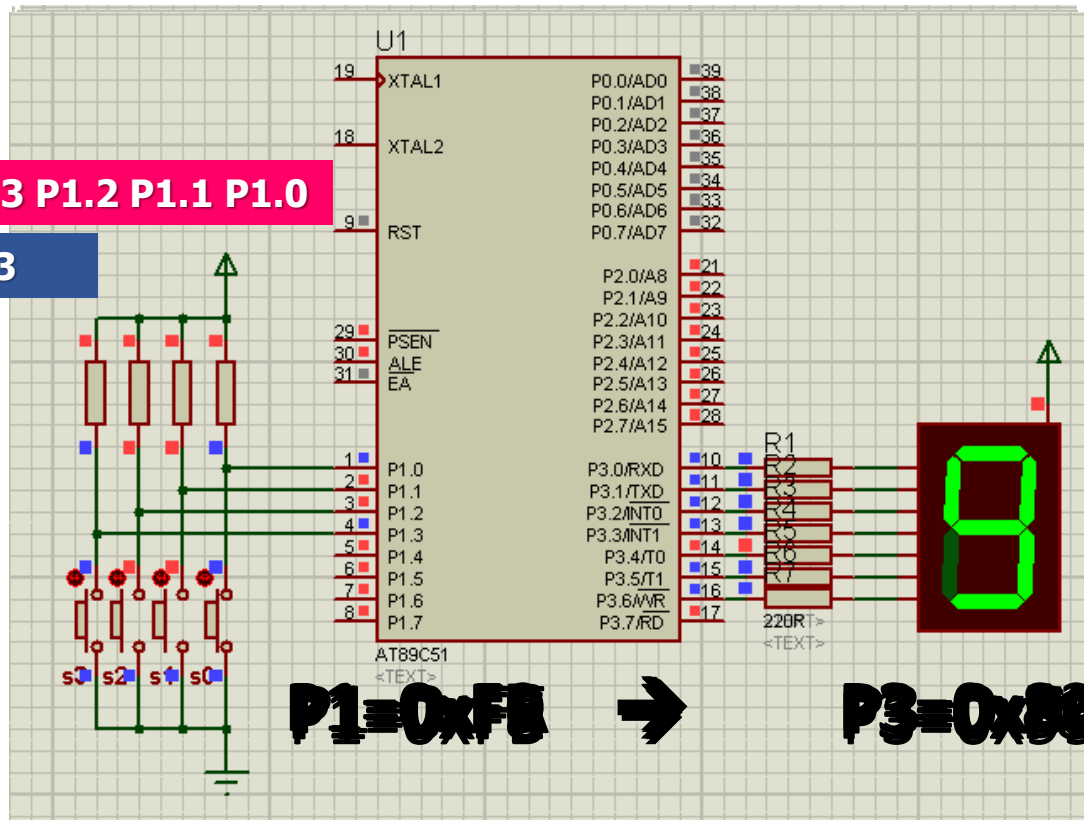
สร้างวงจรบน Proteus

สวิตช์ 4 ตัวเปิด/ปิดควบคุมการแสดงผล 7 ส่วน0-9



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

ส่วนแสดงผลใช้ port P3



P1=0xFB



P3=0x80

ตัวอย่างที่ใช้ switch case (หลายทางเลือก)

พัฒนาโปรแกรมบน Keil

```
#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 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

4.9 ตัวอย่าง 9 : matrix (3x4) switches

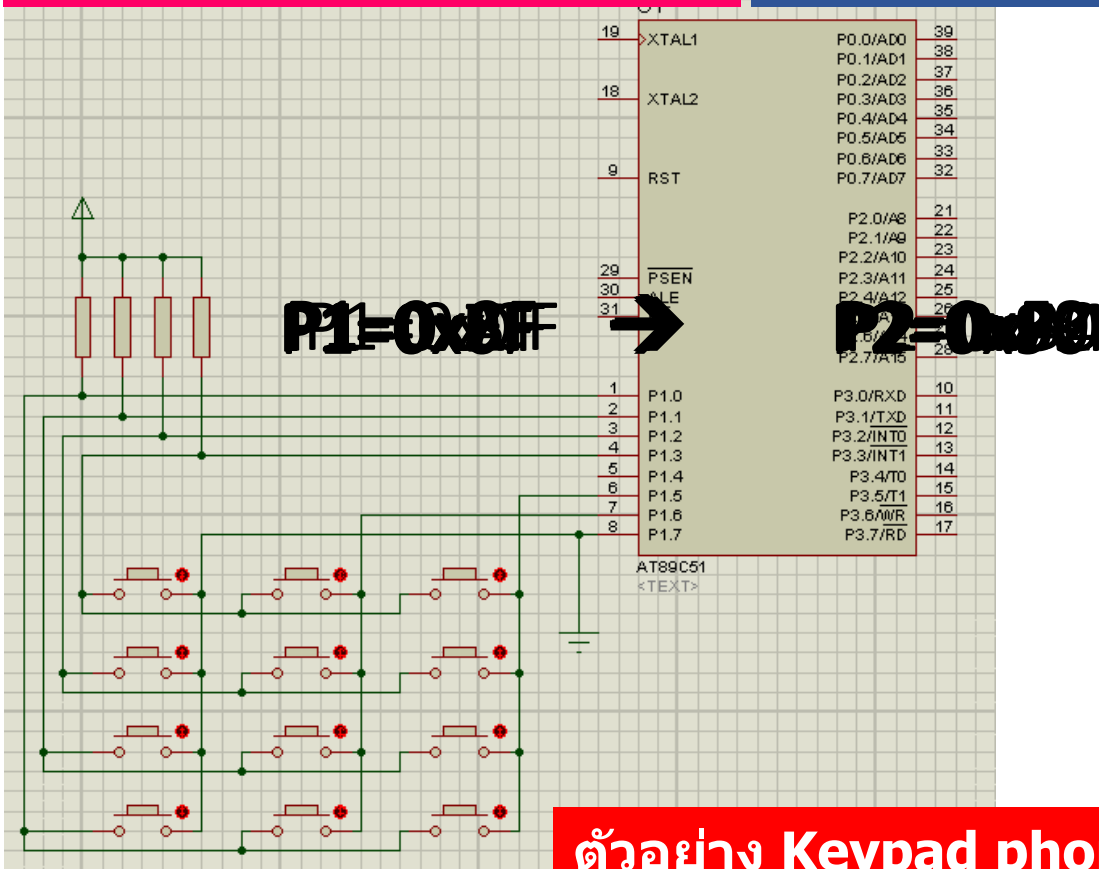
สร้างวงจรบน Proteus

พัฒนาโปรแกรมบน Keil

สวิตช์ 12 ตัว ควบคุมการทำงาน

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

ส่วนแสดงผลใช้ port P2



ตัวอย่าง Keypad phone

```
#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;} }
```



บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้น

เมื่อจบบทเรียนนี้แล้ว... นักศึกษาควรมีสมรรถนะดังนี้



4.1 เข้าใจ
วิธีการพัฒนา
งานควบคุมบน
คอมพิวเตอร์

4.2 เข้าใจการ
ควบคุม 1-LED
(byte control)

4.3 เข้าใจการ
ควบคุม 1-8 LEDs
(shift >> control)

4.6 เข้าใจการควบคุม
1-button switch & 1-LED
แบบ CA (byte control)

4.5 เข้าใจการควบคุม
การแสดงผล บน
7-segment(CA)

4.4 เข้าใจการ
ควบคุม 1-8 LEDs
(byte control)

4.7 เข้าใจการ
ควบคุม 1-button
switch & 1-DC
motor (bit control)

4.8 เข้าใจการ
ควบคุม 3 buttons
switches
& 3-LEDs
(byte control)

4.9 เข้าใจการ
ควบคุม
4-button
switches &
7-segment
(byte control)

4.10 เข้าใจการ
ควบคุม
matrix (3x4)
switches



มหาวิทยาลัย
ราชภัฏนครปฐม