

Led_print

```
/*-----  
Led_print: disply integer number on 7-Segment Display 1 cycle  
input: integer number 0 - 9999  
-----*/
```

```
void led_print(unsigned int num)  
{  
    unsigned char seg[4];  
    int2seg (num,seg);  
    displ(seg);  
    return;  
}
```

19

โปรแกรมทดสอบ 1

```
void main( void )  
{  
  
    int_cpu();  
    while (i)  
    {  
        led_print(2549);  
    }  
}
```

20

โปรแกรมทดสอบ 2

```
void main( void )  
{  
    unsigned char snum[4];  
    unsigned int num,i,j;  
  
    int_cpu();  
    while(t)  
    {  
        num = 0;  
        for (j = 0; j < 10000; j++)  
        {  
            for (i = 0; i < 200; i++)  
            {  
                Led_print(num);  
                num++;  
            }  
        }  
    }  
}
```

21

การเขียนโปรแกรมอ่านการกดสวิตช์ 1 ตัว แบบที่ 1

```

if (P3_7 == 0)
{
    //ถ้ากดสวิตช์ให้ทำอะไร
    .....
}
else
{
    //ถ้าไม่กดสวิตช์ให้ทำอะไร
    .....
}

```

ค่าความต้านทานใช้แบบที่คำนวณมาแล้ว

25

การเขียนโปรแกรมอ่านการกดสวิตช์ 1 ตัว แบบที่ 2

ค่าความต้านทานคำนวณได้ดังนี้
จาก $V_{ILmax} = 0.2 \times V_{CC} - 0.1$
ที่ $V_{CC} = 5V$ $V_{ILmax} = 0.9V$
และ $I_{ILmax} = 50 \mu A$

$$R_{max} = \frac{V_{ILmax}}{I_{ILmax}} = \frac{0.9V}{50 \mu A} = 18K\Omega$$

ค่ากำลังงานสูญเสียสูงสุดที่ความต้านทานคำนวณได้ดังนี้
จาก $P_{Rmax} = V_{CC}^2 / R$
 $= 25 / 18K$
 $= 1.3 mW$

```

if (P2_7 == 1)
{
    //ถ้ากดสวิตช์ให้ทำอะไร
    .....
}
else
{
    //ถ้าไม่กดสวิตช์ให้ทำอะไร
    .....
}

```

26

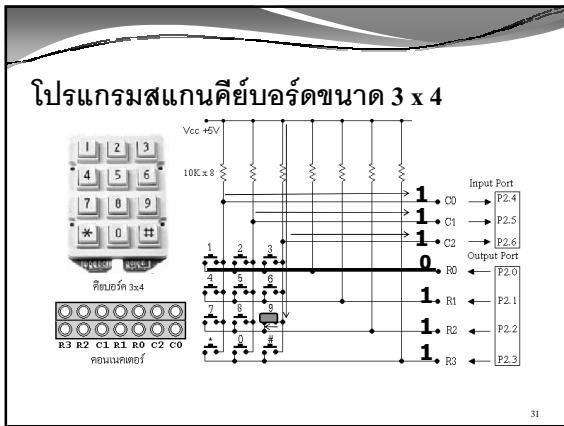
การเขียนโปรแกรมอ่านการกดสวิตช์หลายตัว แบบที่ 1

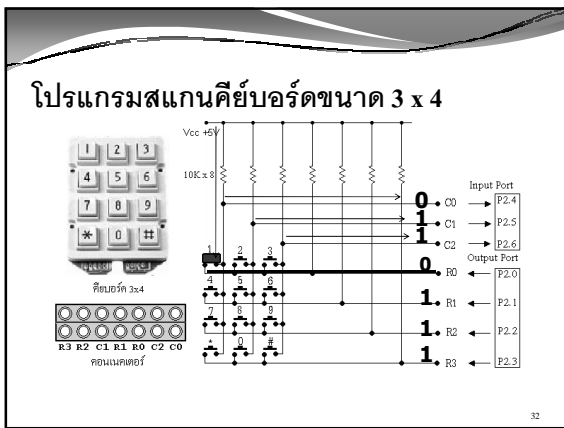
```

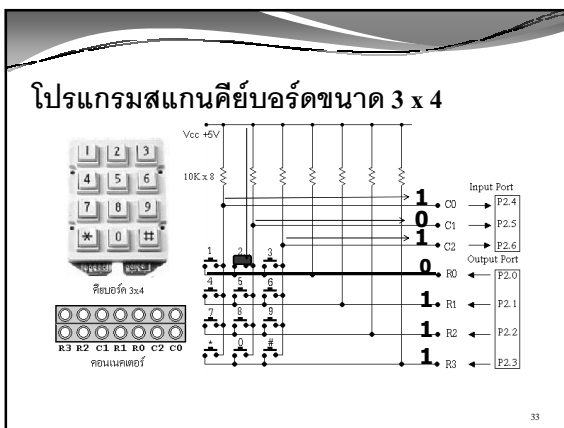
if (P2_7 == 0)
{
    .....
}
if (P2_6 == 0)
{
    .....
}
if (P2_5 == 0)
{
    .....
}
if (P2_0 == 0)
{
    .....
}
}

```

27







โปรแกรมสแกนคีย์บอร์ดขนาด 3 x 4

Input Port
C0 → P2.4
C1 → P2.5
C2 → P2.6

Output Port
R0 ← P2.0
R1 ← P2.1
R2 ← P2.2
R3 ← P2.3

34

โปรแกรมสแกนคีย์บอร์ดขนาด 3 x 4

Input Port
C0 → P2.4
C1 → P2.5
C2 → P2.6

Output Port
R0 ← P2.0
R1 ← P2.1
R2 ← P2.2
R3 ← P2.3

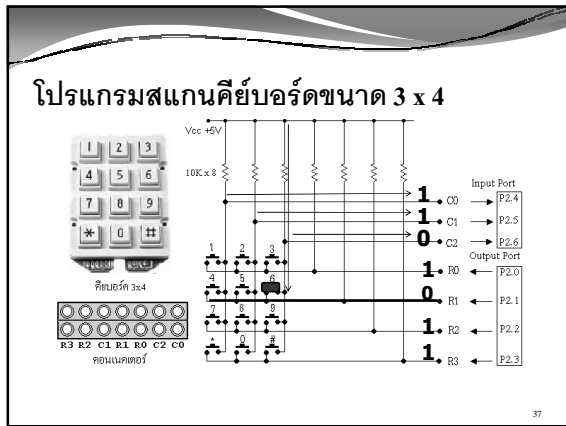
35

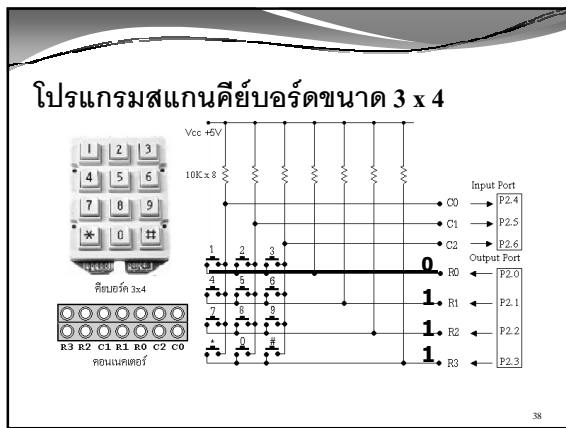
โปรแกรมสแกนคีย์บอร์ดขนาด 3 x 4

Input Port
C0 → P2.4
C1 → P2.5
C2 → P2.6

Output Port
R0 ← P2.0
R1 ← P2.1
R2 ← P2.2
R3 ← P2.3

36





```

/*
Scan1: Scan Keyboard 1 cycle
Output: Keycode
-----*/
unsigned char scan1(void)
{
    unsigned int i;
    char const dig[] = {0xFE, 0xFD, 0xFB, 0xF7};
    char const key[] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09,
0X0A, 0x00, 0x0B};
    unsigned char nt;
    for ( i = 0; i < 4; i++ )
    {
        P2 = dig[i];
        nt = P2 & 0x70;
        nt = _crot_(nt,4);
        if (nt < 7) break;
    }
    if (i > 3)
    {
        nt = 255;
    }
    else
    {
        nt = 6 - nt;
        if (nt == 3) (nt = 2);
        nt = nt + (3*i);
        nt = key[nt];
    }
    return(nt);
}

```

39

```

/*-----
Scan: Scan Keyboard until key pressed
Output: Keycode
-----*/
unsigned char scan(void)
{
    unsigned char s = 255;
    while (s > 20)
    {
        s = scan1();
    }
    return(s);
}

```

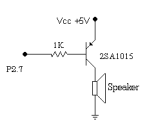
```

void main( void )
{
    unsigned char s;
    int_cpu();

    while(1)
    {
        s = scan();
        if (s < 12)
        {
            PO = s;
        }
    }
}

```

โปรแกรม สร้างเสียง



$F = \text{ความถี่เสียง}$
 $\text{Freq} = \text{ค่าตัวแปรความถี่}$
 $nf = \text{จำนวนรอบ}$
 $tp = \text{ตัวแปรระยะเวลา}$
 $np = \text{จำนวนรอบของเวลา}$
 1 รอบใช้เวลา = $10 \text{ us} \times 2 = 20 \text{ us}$

$nf = 50/\text{freq}$
 $F = 1/(nf \times 20\text{us}) = \text{freq}(50 \times 20 \text{ us}) = \text{freq}/1000 \text{ us} = \text{freq} / 1\text{ms} = \text{freq} \text{ kHz}$
 $Np = \text{freq} \times tp$
 เวลา = $np \times \text{เวลาใน 1รอบของเสียง} = np \times 1/F \text{ ms}$
 $= (\text{freq} \times tp) / F \text{ ms} = \text{freq} \times tp / \text{freq} = tp \text{ ms}$

```

d10us: 10 microsec delay Time
-----*/
void d10us (void)
{
    wait();
    wait();
    wait();
}

-----*/
/*-----*/
Main Program
-----*/
void main (void)
{
    unsigned char s;
    int_cpu();
    while(1)
    {
        s = scan();
        if (s < 12)
        {
            sound(s, 1000);
        }
    }
}

-----*/
/*-----*/
sound: Sound generator
Input : freq frequency in 1 KHz to 10KHz
      (1 KHz per step) freq = 1 to 10
      tp Period in tp ms
-----*/
void sound(unsigned char freq,unsigned int tp)
{
    unsigned int i,j, nf, np;
    nf = 50*freq;
    np = freq*tp;
    for ( j = 0; j < np; j++ )
    {
        P2_7 = 1;
        for( i = 0; i < nf; i++ )
        {
            d10us();
        }
        P2_7 = 0;
        for( i = 0; i < nf; i++ )
        {
            d10us();
        }
    }
}

```

43

```

Bios.h
-----*/
P2_0 = key_R0 and the LED 7 segment Display rightmost digit control
P2_1 = key_R1
P2_2 = key_R2
P2_3 = key_R3 and the LED 7 segment Display leftmost digit control
P2_4 = key_C0 as input port
P2_5 = key_C1 as input port
P2_6 = key_C2 as input port
-----*/
#define key_port P2
#define speaker P2_7
-----*/
/*-----*/
P0_0 = segment a
P0_1 = segment b
P0_2 = segment c
P0_3 = segment d
P0_4 = segment e
P0_5 = segment f
P0_6 = segment g
P0_7 = segment p
-----*/
#define seg_port P1

```

44

```

/*-----*/
Not CPL for standard mode
-----*/
void int_cpu(void)

-----*/
/*-----*/
Delay Time
-----*/
void wait (void)

-----*/
/*-----*/
d10us: 10 microsec delay Time
-----*/
void d10us (void)

-----*/
/*-----*/
d300us: 300 microsec delay time
-----*/
void d300us (void)

-----*/
/*-----*/
d100ms: 100 millisecc delay time
-----*/
void d100ms (void)

```

45

```

/*-----*/
Set serial port for 9600 baud at 11.0592 MHz. Note that we use Timer 1
for the baud rate generator.
-----*/
void int_serial (void)
/*-----*/
Scan1: Scan Keyboard 1 cycle
Output: Keycode
port: P2.0 P2.1 P2.2 P2.3 = output
      P2.4 P2.5 P2.6 = input
      P2.7 = speaker active low
-----*/
unsigned char scan1(void)
/*-----*/
Scan: Scan Keyboard until key pressed
Output: Keycode
-----*/
unsigned char scan(void)

```

46

```

/*-----*/
int2seg: Unsigned integer to 7-Segment pattern, 0 - 9999
input: 7-segment pattern 4 bytes
-----*/
void int2seg (unsigned int num,unsigned char segnum[4])
/*-----*/
hex2seg: Unsigned integer to hex 7-Segment pattern, 0 - F
Input: unsign one digit
Output: Hax-7-segment pattern
-----*/
unsigned char hex2seg (unsigned char hex)
/*-----*/
Disp1: 7-Segment Display 1 cycle
input: 7-segment pattern 4 bytes
-----*/
void disp1(unsigned char seg[4])

```

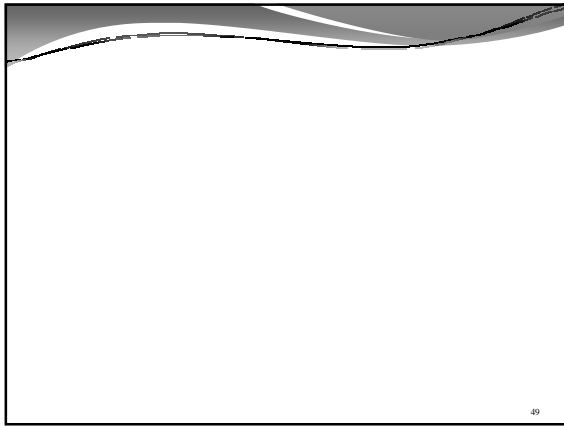
47

```

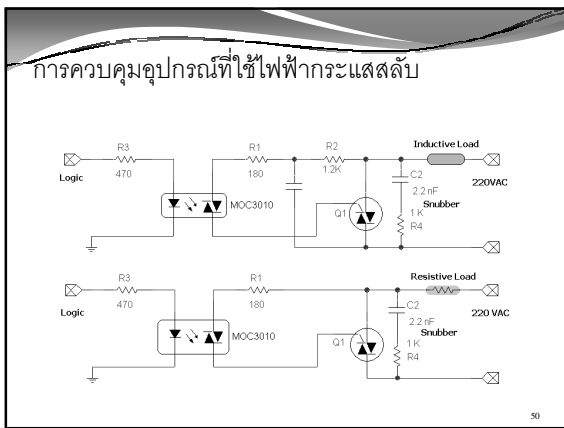
/*-----*/
sound: Sound generator
Input : freq frequency in 1 KHz to 10KHz (1 KHz per step)
      freq = 1 to 10
      tp   Period in tp ms
port: speaker = P2.7 = speaker active low
-----*/
void sound(unsigned char freq,unsigned int tp)

```

48



49



50

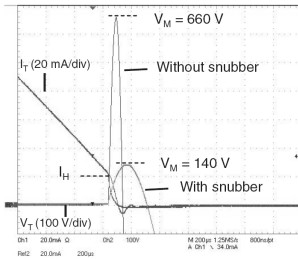
MOC3010

Opto Triac Random Phase:
6V 50mA. 30mA max
trigger current. Triac:
250V 100mA.

51

Snubber

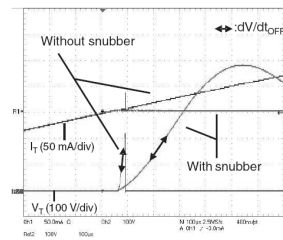
Overvoltage at TRIAC turn-off with and without snubber circuit
($C = 10 \text{ nF}$ and $R = 2.7 \text{ k}\Omega$)



52

Snubber กับ Inductive Load

TRIAC turn-off on inductive load without and with snubber circuit
($C = 10 \text{ nF}$ and $R = 2.7 \text{ k}\Omega$)



53
