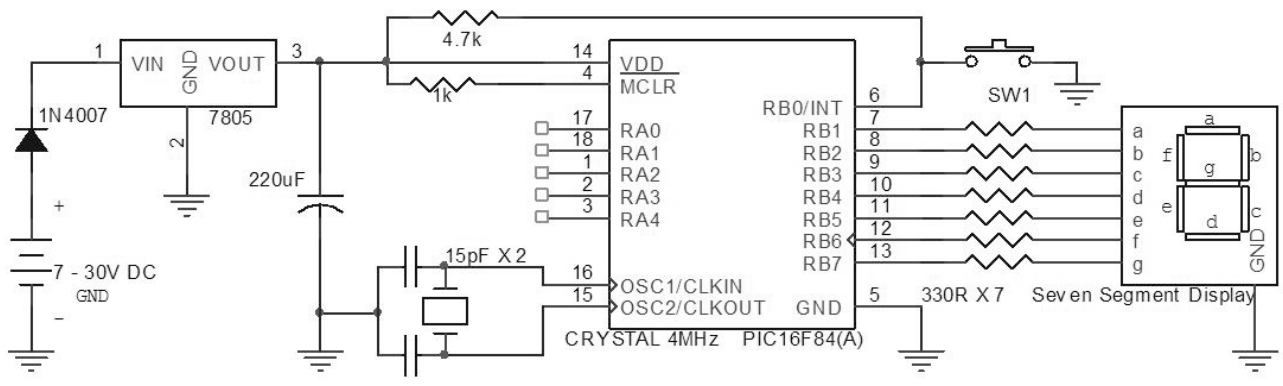
MICROCONTROLLERS

14 ලිපිය - 9 කොටස

බාහිර අතුරුබිඳුම් භාවිත කරන අයුරු

PIC 16184(A) මයික්‍රොකොන්ට්‍රෝලරයට බාහිරින් අතුරුබිඳුමක් ලබා දීම සහ ඊට අදාළ ක්‍රමලේඛය C පරිගණක භාෂාව ඇසුරින් ගොඩනංවා ගන්නා ආකාරය මෙම ලිපිය තුළින් විස්තර කෙරේ.



රූපසටහන අංක 1

රූප සටහන අංක 1න් අදාළ පරිපථය දැක්වේ. එය 6.6 ලිපියේ සඳහන් වූ පරිපථය ම වේ. SW 1 බොත්තම වරක් එබූ විට RB0 අග්‍රය හරහා බාහිර අතුරුබිඳුමක් ජනනය වේ. එවිට සප්තඛණ්ඩ ප්‍රදර්ශකයේ දක්වන අගය එකකින් වැඩි වන ආකාරයට ක්‍රමලේඛය සැකසිය යුතු ය. එලෙස සැකසූ ක්‍රමලේඛය රූප සටහන අංක 3න් දැක්වේ.

සුපුරුදු පරිදි එම ක්‍රමලේඛයේ මුලින් ම භාවිත වන උපදෙස් ගොනු සහ විචල්‍යයන් හඳුන්වා දී ඇත. ඒ සඳහා void ssdecodec (int i); සහ int Digit 1; උපදෙස් යොදාගෙන ඇත. අතුරුබිඳීම් සඳහා වූ උපදෙස් ගොනුව void interrupt () යටතේ ඉදිරිපත් කර ඇත. අතුරුබිඳුමක් ජනනය වූ වහාම ප්‍රධාන උපදෙස් ගොනුවෙන් මෙම උපදෙස් ගොනුව පැමිණේ. එසේ පැමිණි විට එම අතුරුබිඳුම ජනනය වූයේ බාහිර අතුරුබිඳුමක් හෙවත් RB0 අග්‍රයට සම්බන්ධ SW ස්විචය එබීම නිසා දැයි පරීක්ෂා කළ යුතු වේ. එසේ පරීක්ෂා කිරීම සඳහා if(INTCON.INTF) උපදෙස යොදාගෙන ඇත. එසේ වී නම් සප්තඛණ්ඩ ප්‍රදර්ශකයේ ද ප්‍රදර්ශනය කළ යුතු අගය දැක්වෙන Digit 1 විචල්‍යයේ අගය එකකින් වැඩි කෙරේ. එමෙන්ම එම අගය "10"ට පැමිණි විට නැවත "D" බවට පත් කිරීම ද සිදු කොට ඇත.

අවසානයේ දී INTCON.INTF = 0; මගින් බාහිර අතුරුබිඳුමක් ජනනය වූ බව දැක්වෙන බිටුව (Interrupt flag) "0" බවට පත් කරයි. එය අනිවාර්යයෙන් ම කළ යුත්තකි. එසේ

```
    // Introduce functions and variables
void ssdecode(int i);  //Function for 7 Segment decoding
int Digit1;            // Variable for digit1

    // Interrupt Service Routine (ISR)
void interrupt()
{
        if (INTCON.INTF)          // Check for Timer 0 interrupt
        {
        Digit1 = Digit1 +1;       // Increase Digit 1
        if(Digit1>9) Digit1=0;
        INTCON.INTF = 0;          // Clear RB0 interrupt flag
        }
}

    //Main Function
void main()
{
        TRISB = 0b00000001;       //RB0 input
        INTCON = 0b10010000;      //Enable RB0 interrupts
        Digit1 = 0;               //Start from 0

        while (1) // loop forever
        {
        ssdecode(Digit1);         //Display digit 1
        Delay_ms(200);            //Small delay
        }
}

    // Sewven Segment Decoder function
void ssdecode(int i)
{
    switch (i)
    {
    case 0: PORTB = 0b01111111; break;
    case 1: PORTB = 0b00001101; break;
    case 2: PORTB = 0b10110111; break;
    case 3: PORTB = 0b10011111; break;
    case 4: PORTB = 0b11001101; break;
    case 5: PORTB = 0b11011011; break;
    case 6: PORTB = 0b11111011; break;
    case 7: PORTB = 0b00001111; break;
    case 8: PORTB = 0b11111111; break;
    case 9: PORTB = 0b11011111; break;
    }
}
```

රූපසටහන අංක 3

INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **TMR0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE**: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF**: RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).
0 = None of the RB7:RB4 pins have changed state

රූපසටහන අංක 2

නො කළ හොත් වරක් අතුරුබිඳුම ජනනය වූ විට එය දිගින් දිගට ම ක්‍රියාත්මක වේ. අතුරුබිඳුම් හා සම්බන්ධ සකස්කිරීම් සඳහා INTCON හෙවත් Interrupt Configuration Register නැමැති රෙජිස්ටරය පිළිබඳව දැන සිටීම අත්‍යවශ්‍ය වේ. රූප සටහන අංක 2 මගින් එම රෙජිස්ටරයේ එක් එක් බිටුව මගින් සිදු කරන කාර්යයන් දැක්වේ.

ප්‍රධාන උපදෙස් ගොනුවේ දී මුලින් ම සිදු කොට ඇත්තේ B තොටුපළේ RB0 අග්‍රය ප්‍රදනයක් ලෙස සහ අනෙකුත් අග්‍ර ප්‍රතිදන ලෙස සකස් කිරීමයි. ඒ සඳහා TRIS B රෙජිස්ටරයට 00000001 යන බිටු සැකැස්ම ලියා ඇත. 0 මගින් ප්‍රතිදන ලෙස ද "1" මගින් ප්‍රදන ලෙස ද සකස් කරයි. ඉන් පසුව ඇති INTCON = 0b10010000; උපදෙස මගින් INTCON රෙජිස්ටරයේ සිදු කළ යුතු සකස් කිරීම කරනු ලබයි. එහි සත්වන බිටුව හෙවත් bit 7 මගින් Global Interrput Enable හෙවත් සියලු අතුරුබිඳුම් ක්‍රියාත්මක කිරීම හෝ නො කිරීම තීරණය කරයි. එම බිටුව තාර්කික 1 බවට පත් කළ යුතු වේ. එසේ කිරීමෙන් අනතුරුව අපට අවශ්‍ය වන අතුරුබිඳුම් වර්ග වෙන වෙන ම ක්‍රියාත්මක කළ යුතු ය. අපට අවශ්‍ය වන බාහිර RB0/INTE බාහිර අතුරුබිඳුම් ක්‍රියාත්මක කිරීම සඳහා bit 4 හෙවත් 4 වන බිටුව ද තාර්කික 1 බවට පත් කළ යුතු ය. එසේ කළ පසු බාහිර අතුරුබිඳුම් ජනනය කළ හැකි තත්ත්වයට පත් වේ.

INTCON රෙජිස්ටරයේ අනෙකුත් බිටු මගින් එක් එක් අතුරුබිඳුම් වර්ගවලට අදාළ සකස් කිරීම් සිදු කරගත හැකි ය. එහි bit 1 හෙවත් බිටු අංක 1 මගින් RB0 අග්‍රය හරහා අතුරුබිඳුමක් ජනනය වී තිබේ ද නැද්ද යන වග පෙන්නුම් කරයි. බාහිර අතුරුබිඳුමක් ජනනය වූ වහාම එම බිටුව තාර්කික 1 බවට පත් වේ. අතුරුබිඳුමට අදාළ කාර්යයන් හෙවත් Interrupt Service Routine (ISR) සිදු කිරීමෙන් අනතුරුව එම බිටුව "0" බවට පත් කළ යුතු ය.

ක්‍රමලේඛය while(1) යටතේ ssdecode(Digit 1); උපදෙස දැක්වේ. එම උපදෙසට අනුව ssdecode උපදෙස් ගොනුවට ගොස් Digit 1 විචල්‍යයේ අගය අනුව සප්තඛණ්ඩ ප්‍රදර්ශකයේ දැල්විය යුතු ඛණ්ඩක තෝරා ඒවා දැල්වීම සිදු කරයි. බොත්තම වරක් එබූ විට ප්‍රදර්ශනය කළ යුතු අගය එකකින් වැඩි කරයි. එවිට එම අගයට අදාළ ඛණ්ඩක දැල්වේ. මෙය දිගින් දිගට ම සිදු වේ. එනම් 0 සිට 9ට වැඩි වී නැවත 0ට පැමිණේ. අවශ්‍ය නම් තවත් සප්තඛණ්ඩකයක් යොදා 0 සිට 99ට ගණන් කිරීමට ද මෙම ක්‍රමලේඛය සහ පරිපථය වැඩිදියුණු කොට ඇත.

මොරටුව විශ්වවිද්‍යාලයයේ විද්‍යුත් හා විදුලි සංදේශ අංශයේ
ගාමිණී ජයසිංහ
කෝලිත ධර්මප්‍රිය