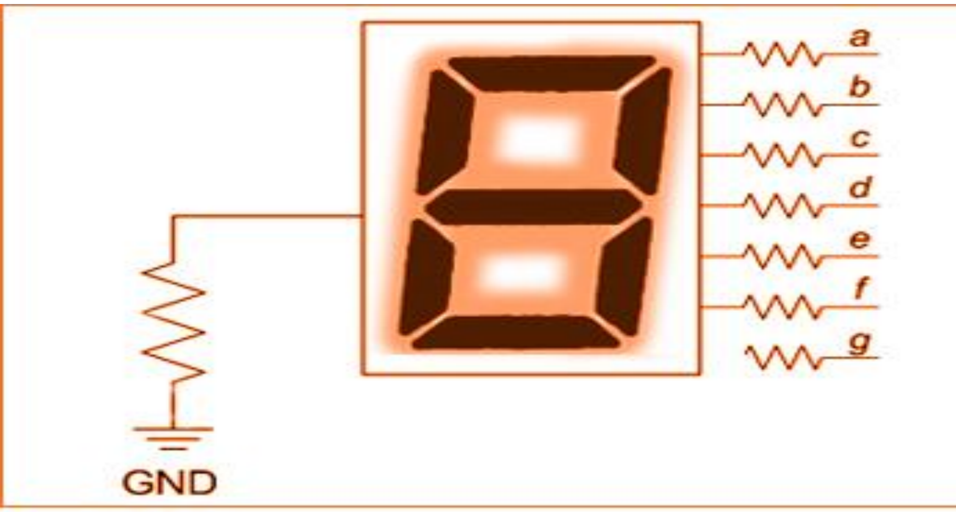
PANTECH SOLUTIONS®
Technology Beyond the Dreams

PIC HOW-TO GUIDE

Interfacing 7SEG with PIC16F877A



Contents at a Glance

PIC16F/18F Primer Board

The PIC16F/18F Primer board is specifically designed to help students to master the required skills in the area of embedded systems. The kit is designed in such way that all the possible features of the microcontroller will be easily used by the students. The kit supports in system programming (ISP) which is done through USB port.

Microchip's PIC (PIC16F877A), PIC16F/18F Primer Kit is proposed to smooth the progress of developing and debugging of various designs encompassing of High speed 8-bit Microcontrollers.

Seven Segment Display

A **seven segment display** is the most basic electronic display device that can display digits from 0-9. The most common configuration has an array of eight LEDs arranged in a special pattern to display these digits. They are laid out as a squared-off figure '8'.

Interfacing Seven Segment Display

Fig. 1 shows how to interface the seven segments with microcontroller. A seven segment is generally available in ten pin package. While eight pins correspond to the eight LEDs, the remaining two pins (at middle) are common and internally shorted. These segments come in two configurations, namely, Common cathode (CC) and Common anode (CA).

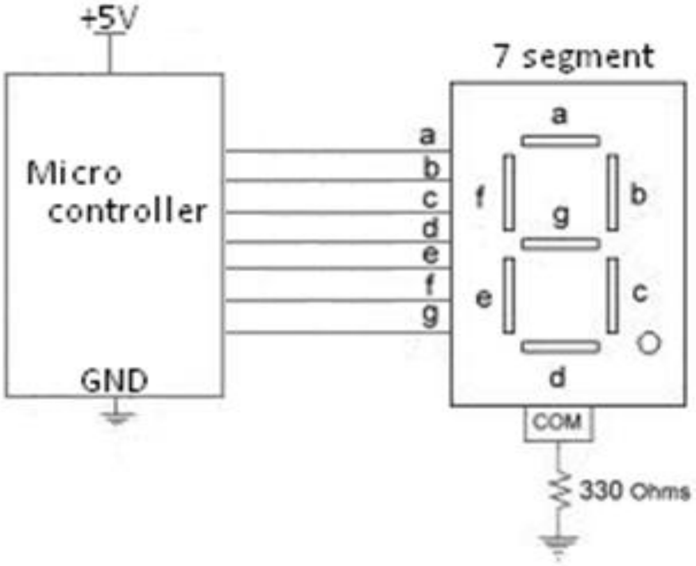


Fig. 1 Interfacing 7segment to Microcontroller

Interfacing Seven Segment with PIC16F877A

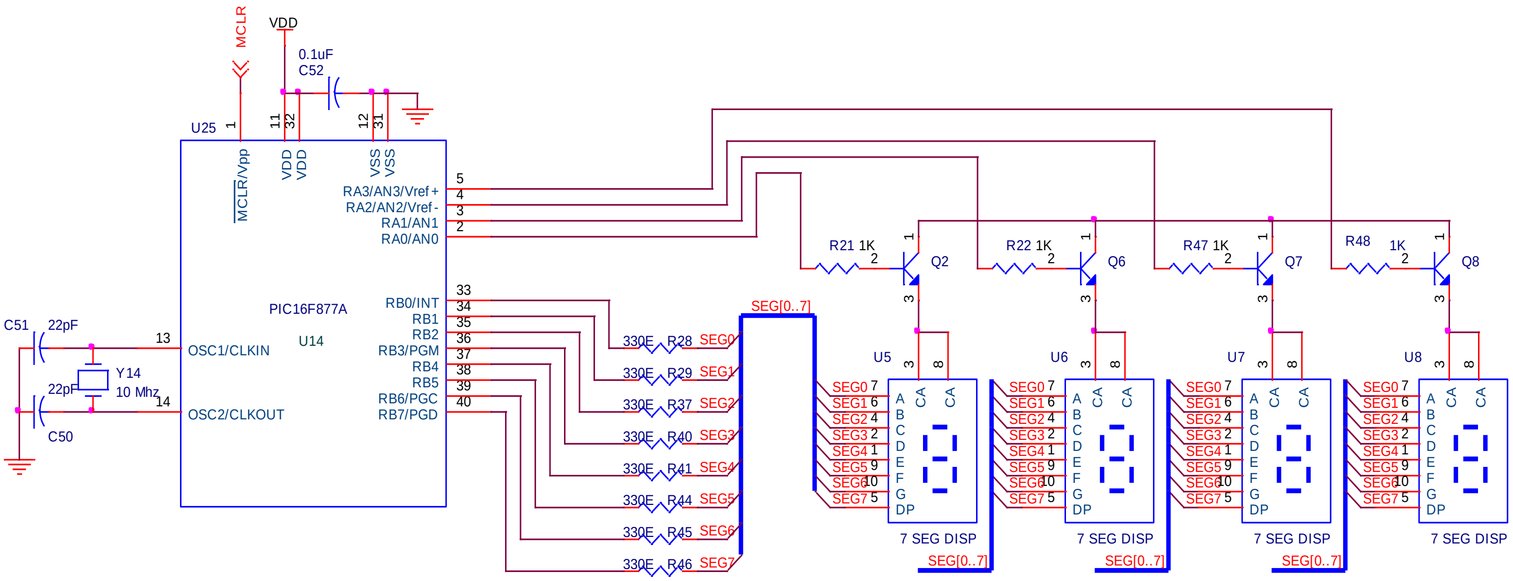
We now want to display a four digit number in PIC16F/18F Primer Board by using seven segment displays. The seven segment display is connected with PIC16F877A microcontroller.

In **PIC16f/18F Primer Kit**, 4 nos. of common anode seven segment displays are controlled by seven segment drivers.

Pin Assignment with PIC16F877A

	7-SEG Driver	PIC16F LINES	Connections
7-SEG Display	A	PORTB.0	* Connect PORTB with JP15 via FRC Cable. * JP15 is at Seven Segment Section
	B	PORTB.1	
	C	PORTB.2	
	D	PORTB.3	
	E	PORTB.4	
	F	PORTB.5	
	G	PORTB.6	**Output**
	DP	PORTB.7	* Digits will be incrementing from 0 to 9999
	CL1	PORTA.3	
	CL2	PORTA.2	
	CL3	PORTA.1	
	CL4	PORTA.0	

Circuit Diagram to Interface 7 segment with PIC16F877A



7 SEGMENT DISPLAY

Source Code

The Interfacing seven segment displays with PIC16F877A program is very simple and straight forward, which display a four digit number in seven segment display .The C programs are developed in Mplab software. Here we are increment a counter and display this value loaded into seven segment driver in PIC16F/18F PIC Primer Board.

C Program to 7 Segment Display using PIC16F877A

```
************************************************************************************
                 Title   : Program to Seven Segment display
************************************************************************************

#include<pic.h>  // Define PIC Registers

_CONFIG(0x3f72); // HS,Enable(PWRTE,BOREN),
                 // Disable (CPD,CP,WDTEN,In-circuit Debugger)

#define CNTRL_PORT PORTA
#define DATA_PORT  PORTB

void hex2dec(unsigned char);
void send_seg(unsigned char,unsigned char,
unsigned char,unsigned char);
void DelayMs(unsigned int);

unsigned char x;
unsigned char thou=0,hun=0,ten=0,single=0;
unsigned char CA[10]   =
{0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};

unsigned char CC[10]   =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

unsigned char CA_CNTRL[4]    =    {0x07,0x0b,0x0d,0x0e};
unsigned char CC_CNTRL[4]    =    {0x08,0x04,0x02,0x01};
unsigned char n=1;

void main()
{
   unsigned char number;
   nRBPU =0;
   TRISB=0x00;    //PORTB configured as O/P
   ADCON1=0x07;   //Configure PORTA & PORTE as Digital port
   TRISA=0x00;    //PORTA Configured as O/P
   while(1)
   {
      if(x == 200)
      {
         x=0;
         single++; //Increment up to 9 in unit place

         if(single>9)
         {
            single=0;
            ten++; //Increment up to 9 in Tenth place

            if(ten>9)
            {
               ten=0;
                hun++;//Increment up to 9 in Hundredth place

                if(hun>9)
                {
                  hun=0;
                  thou++;//Increment up to 9 in Thousandth place

                   if(thou>9)
                     thou=0;
                }
             }
          }
      }
      x++;

      send_seg(thou,hun,ten,single);
   }
}

void send_seg(unsigned char thou,unsigned char hun,
unsigned char ten,unsigned char single)
{
   if(n==1)
   {
      CNTRL_PORT=CA_CNTRL[0]; //Eanble Unit place 7-Segment
      DATA_PORT=CA[single];   //Display Unit Place Number
      n=2;
      DelayMs(5);
   }

   else if(n==2)
   {
      CNTRL_PORT=CA_CNTRL[1]; //Eanble Tenth place 7-Segment
     DATA_PORT=CA[ten];       //Display Tenth Place Number
     n=3;
      DelayMs(5);
   }

   else if(n==3)
   {
     CNTRL_PORT=CA_CNTRL[2]; //Enable Hundredth place 7-Segment
     DATA_PORT=CA[hun];      //Display Hundredth Place Number
     n=4;
     DelayMs(5);
   }

   else if(n==4)
   {
    CNTRL_PORT=CA_CNTRL[3];//Eanble Thousandth place 7-Segment
    DATA_PORT=CA[thou];    //Display Thousandth Place Number
    n=1;
    DelayMs(5);
   }
}

void DelayMs(unsigned int Ms)
{
   int delay_cnst;
   while(Ms>0)
   {
     Ms--;
      for(delay_cnst = 0;delay_cnst <220;delay_cnst++);
   }
}
```

To compile the above C code you must need the Mplab software and Hi-Tech C Compiler. They must be properly set up and a project with correct settings must be created in order to compile the code. To compile the above code, the C file must be added to the project.

In Mplab, you want to develop or debug the project without any hardware setup. You must compile the code for generating HEX file. In debugging Mode, you want to check the port output without PIC16F/18F Primer Board.

The PICKIT2 software is used to download the hex file into your microcontroller IC PIC16F877A through USB port.

Testing the I2C – Seven segment with PIC16F

Give +12V power supply to PIC16F/18F Primer Board; the four seven segment display is connected with the PIC16F/18F Primer Board. First check the entire seven segments LED's are properly working or not. Here we are display just 1234 in four seven segment. The entire seven segments receive it through I2C & display it in order.

If any data is not coming in seven segments, then you just check the entire seven segments LED's are working or not. Change the seven segment driver IC & Check the I2C connections. Check the four seven segments connections. Otherwise you just check the code with debugging mode in Mplab. If you want to see more details about debugging just see the videos in below link.

- How to create & Debug a Project in Mplab using PIC16F using Hi-Tech Compiler.

General Information

- For proper working use the components of exact values as shown in Circuit file. Wherever possible use new components.
- Solder everything in a clean way. A major problem arises due to improper soldering, solder jumps and loose joints.
- Use the exact value crystal shown in schematic.

- More instructions are available in following articles,
  - ➢ Interfacing UART with PIC16F877A.
  - ➢ Interfacing Keys with PIC16F877A.
  - ➢ User Manual of PIC16F/18F Primer Board.
  - ➢ Create & Debug a project in Mplab using PIC16F877A.

What do we sell?

Our products range from Various Microcontroller development boards, DSP Boards, FPGA/CPLD boards, Communication Kits, Power electronics, Basic electronics, Robotics, Sensors, Electronic components and much more . Our goal is to make finding the parts and information you need easier and affordable so you can create awesome projects and training from Basic to Cutting edge technology.