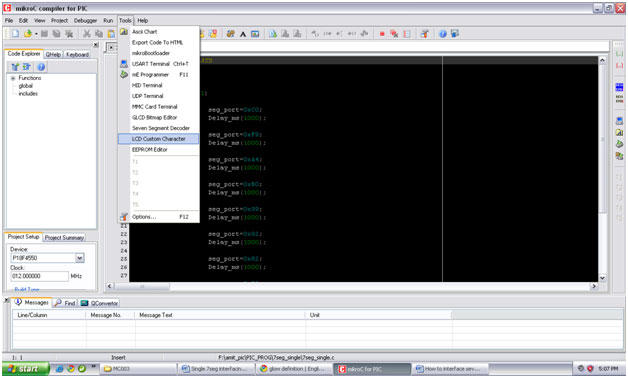
**[How to interface Seven Segment Display with PIC18F4550 Microcontroller](http://www.engineersgarage.com/embedded/pic-microcontroller-projects/7-segment-display-interfacing-circuit" \o "How to interface Seven Segment Display with PIC18F4550 Microcontroller)**

The [seven segments](http://www.engineersgarage.com/electronic-components/7-segment-display) are used to display decimal and hexadecimal (0-9, A-F) values. A seven segment is cheapest option for applications requiring numeric value display as output. Calculators, watches, lift’s floor indication panel etc. are examples of such applications. The interfacing and operation of a seven-segment display with [PIC18F4550](http://www.engineersgarage.com/components/pic18f4550-microcontroller) has been explained here.

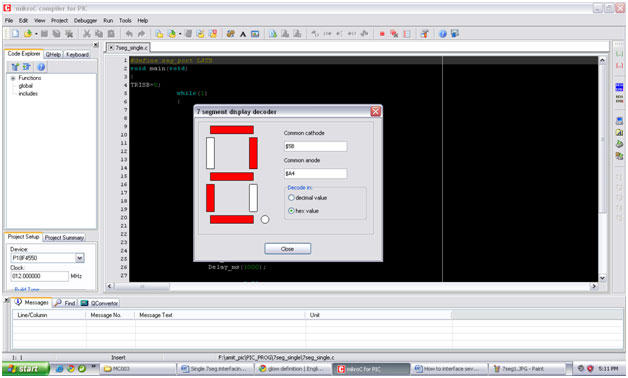
A typical [seven-segment](http://www.engineersgarage.com/electronic-components/7-segment-display) consists of 8 LEDs arranged in a pattern to display values. A seven-segment can be either of the two types, namely, Common Anode (CA) and Common Cathode (CC). For more details, refer Seven segments.

A single seven-segment requires a minimum of 7 data pins of controller to display different values. The connections of seven-segment with [PIC18F4550](http://www.engineersgarage.com/components/pic18f4550-microcontroller) are shown in the adjoining circuit.

To display a numeric value, a particular data-byte is sent by the [microcontroller](http://www.engineersgarage.com/microcontroller). Refer [seven segment interfacing with 8051](http://www.engineersgarage.com/microcontroller/8051projects/interface-seven-segment-AT89C51-circuit). This data byte can either be calculated manually or by using Seven Segment Editor tool of mikroC IDE. (Also see Working with mikroC) Following are the steps to use this tool:

1. Go to Tools -> Seven Segment Editor       

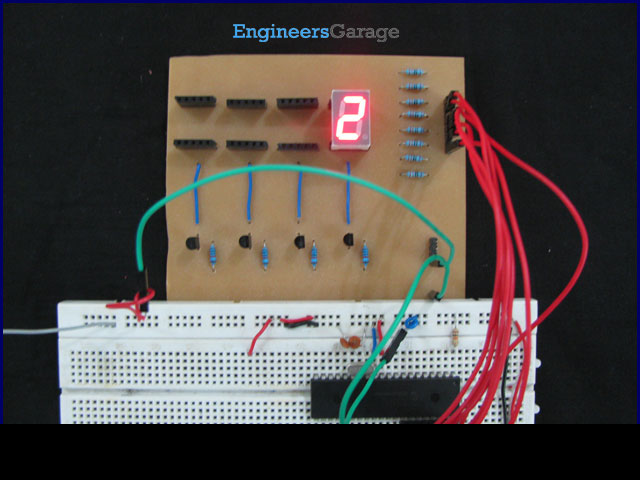
2. Generate the numeric value by clicking on the [LED](http://www.engineersgarage.com/electronic-components/leds-light-emitting-diode) segments and the desired decimal/hexa-decimal values for Common cathode & Common anode configurations can be noted from the adjoining text boxes.

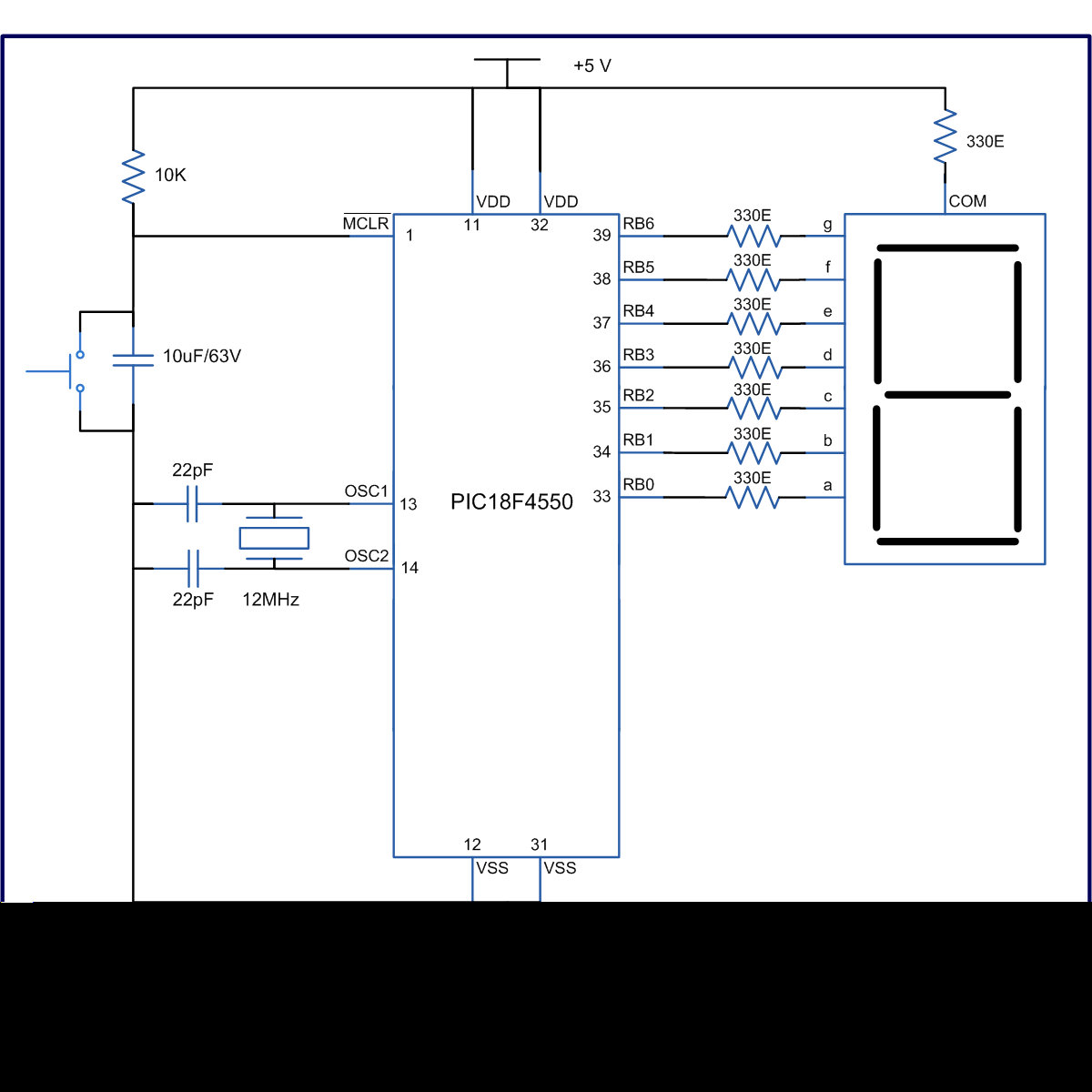


This tool works on the assumption that pins 0, 1, 2, 3, 4, 5, 6 and 7 of the controller’s port (PortB in this case) is connected to the pins a, b, c, d ,e ,f, g and h of the seven-segment respectively. The following table shows the hexadecimal values of the corresponding decimal value.

|  |  |
| --- | --- |
| **Value on Seven-segment** | **Hexadecimal value** |
| 0 | 0xC0 |
| 1 | 0xF9 |
| 2 | 0xA4 |
| 3 | 0xB0 |
| 4 | 0x99 |
| 5 | 0x92 |
| 6 | 0x82 |
| 7 | 0xF8 |
| 8 | 0x80 |
| 9 | 0x90 |

To print the numeric value in the left column above, the corresponding hexadecimal value from right column is sent to the controller’s port which is connected to the seven segment display.





**// Program to Interface seven segment display with PIC18F4550 Microcontroller**  
  
// Configuration bits  
/\* \_CPUDIV\_OSC1\_PLL2\_1L, // Divide clock by 2  
 \_FOSC\_HS\_1H, // Select High Speed (HS) oscillator  
 \_WDT\_OFF\_2H, // Watchdog Timer off  
 MCLRE\_ON\_3H // Master Clear on  
\*/  
  
#define seg\_port LATB  
void main(void)  
{  
 TRISB=0; // Configure PortB as output port  
 while(1)  
 {  
 seg\_port=0xC0; // Display '0'  
 Delay\_ms(1000);  
   
 seg\_port=0xF9; // Display '1'  
 Delay\_ms(1000);  
   
 seg\_port=0xA4; // Display '2'  
 Delay\_ms(1000);  
   
 seg\_port=0xB0; // Display '3'  
 Delay\_ms(1000);  
   
 seg\_port=0x99; // Display '4'  
 Delay\_ms(1000);  
   
 seg\_port=0x92; // Display '5'  
 Delay\_ms(1000);  
   
 seg\_port=0x82; // Display '6'  
 Delay\_ms(1000);  
   
 seg\_port=0xF8; // Display '7'  
 Delay\_ms(1000);  
   
 seg\_port=0x80; // Display '8'  
 Delay\_ms(1000);  
   
 seg\_port=0x90; // Display '9'  
 Delay\_ms(1000);  
 }  
}