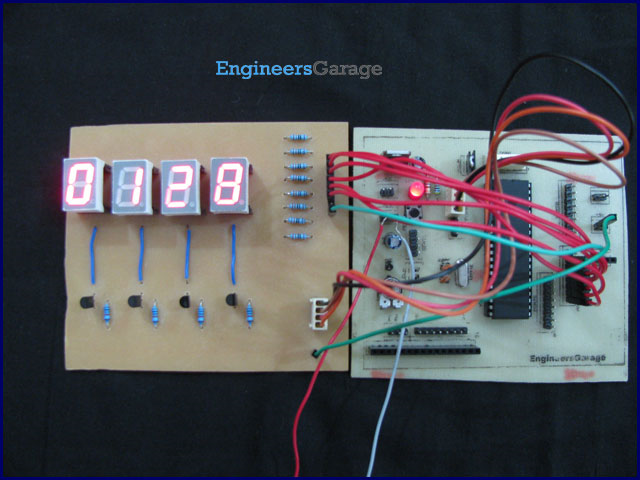
**[Seven Segment Multiplexing using PIC18F4550 Microcontroller](http://www.engineersgarage.com/embedded/pic-microcontroller-projects/7-segment-multiplexing-circuit" \o "Seven Segment Multiplexing using PIC18F4550 Microcontroller)**



As explained earlier, a [seven segment interfaced with PIC](http://www.engineersgarage.com/embedded/pic-microcontroller-projects/7-segment-display-interfacing-circuit) uses almost an entire port (minimum 7 pins) to display a value. But a real time application, like watch, calculator etc., usually requires at least 3-4 seven segments. In such a case it is not advisable to use a port of the controller for each seven segment. In these cases, multiplexing technique is used to work with more than one [seven segment](http://www.engineersgarage.com/electronic-components/7-segment-display). Here multiplexing of four seven-segments has been explained with [PIC18F4550](http://www.engineersgarage.com/components/pic18f4550-microcontroller) to display four-digit count from 0000 to 9999.

The data pins (a-g) of all the [seven-segments](http://www.engineersgarage.com/electronic-components/7-segment-display) are connected to a single port (Port D\*) as shown in the circuit diagram. Transistors [BC547](http://www.engineersgarage.com/electronic-components/transistor-bc547-datasheet) are connected to COM pins of seven-segment for switching. The switching of COM pins is controlled by four pins of PortA.

\*Please note that the pins of PortD are not continuous and care has to be taken while making the connection.

The multiplexing concept is based on the principle of persistence of human vision. A human eye cannot detect a visual change if the frames change at a rate of 25 (or more) frames per sec. This means that if events occur continuously with a time difference of less than or equal to 0.04 sec (1/25 sec), then we cannot notice the transition between those events.

Considering this, the seven-segments are switched on one by one with a very small time delay. Thus, even though only one segment glows at a time, it appears that all the segments are glowing together. (See video) Thus the key factor in multiplexing is switching time of the segments.

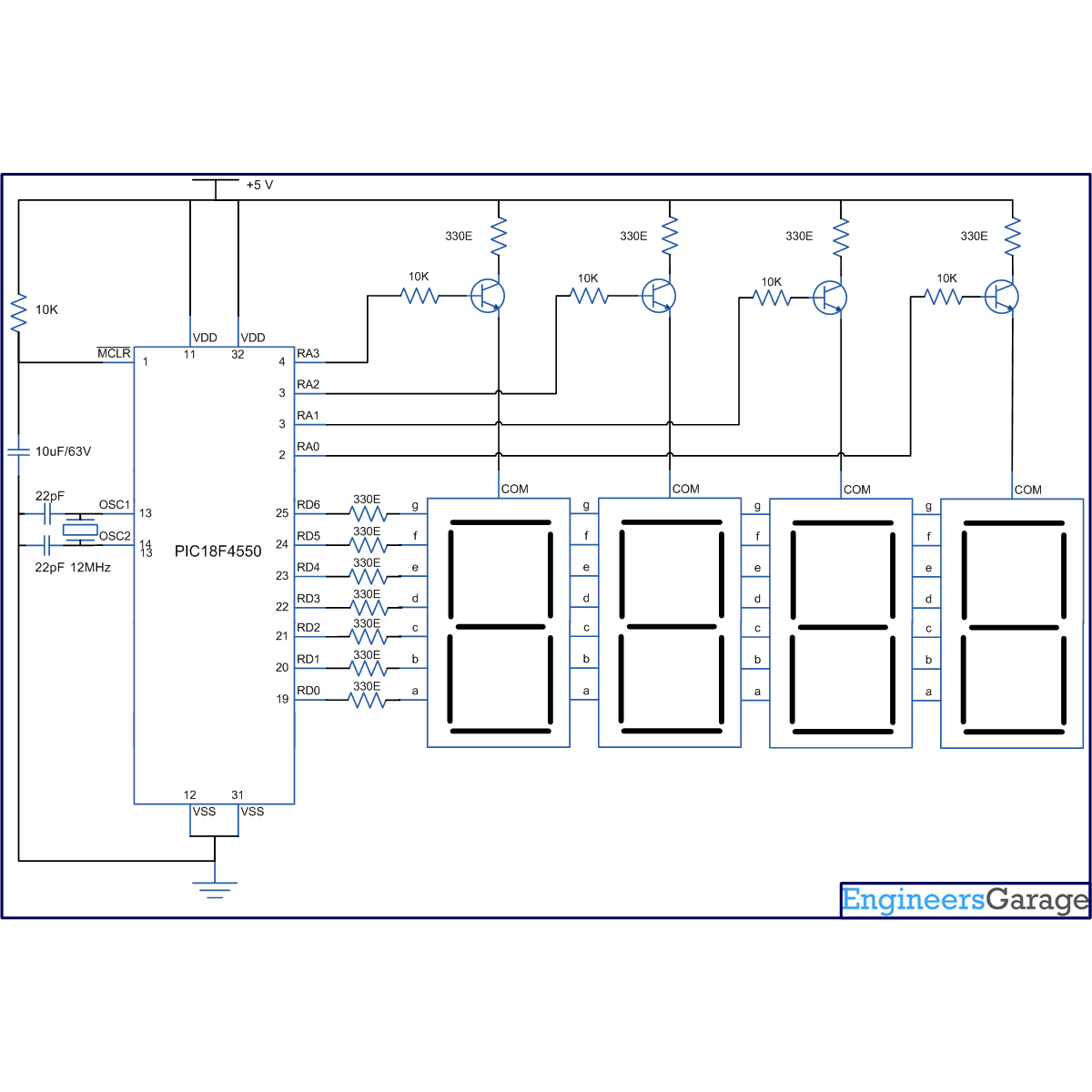
Programming steps:

·         The count value to be displayed is stored in a variable.

·         Extract the individual digits from the count value into different variables.

·         Turn on the seven-segments one by one with a small time delay.

·         Send the hexadecimal values corresponding to individual digits to PortD.



**// Program for Seven segment multiplexing using 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 LATD  
#define seg\_unit LATA.F0  
#define seg\_decade LATA.F1  
#define seg\_hundred LATA.F2  
#define seg\_thousand LATA.F3  
unsigned int i=0,j=0,k=0;  
  
void main(void)  
{  
 unsigned int value[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};  
 unsigned int count,num0,num1,num2,num3,num4;  
 TRISA=0; // Configure PortA as output port  
 LATA=0;  
 TRISD=0; // Configure PortD as output port  
 LATD=0;  
  
 for(count=0;count<9999;count++) // Counter from 0 to 9999  
 {  
 num0=count;  
  
 num1=num0%10; // Extract the value of unit digit  
 num0=num0-num1;  
 num0=num0/10;  
  
 num2=num0%10; // Extract the value of decade digit  
 num0=num0-num2;  
 num0=num0/10;  
   
 num3=num0%10; // Extract the value of hundred digit  
 num0=num0-num3;  
 num0=num0/10;  
   
 num4=num0%10; // Extract the value of thousand digit  
 num0=num0-num4;  
 num0=num0/10;  
   
 for(i=0;i<10;i++) // Delay= ((5msx4)x10) = 200ms between two consecutive counts  
 {  
   
 seg\_unit=1;seg\_decade=0;seg\_hundred=0;seg\_thousand=0; // Display unit digit  
 seg\_port=value[num1];  
 Delay\_ms(5);  
   
 seg\_unit=0;seg\_decade=1;seg\_hundred=0;seg\_thousand=0; // Display decade digit  
 seg\_port=value[num2];  
 Delay\_ms(5);  
   
 seg\_unit=0;seg\_decade=0;seg\_hundred=1;seg\_thousand=0; // Display hundred digit  
 seg\_port=value[num3];  
 Delay\_ms(5);  
   
 seg\_unit=0;seg\_decade=0;seg\_hundred=0;seg\_thousand=1 // Display thousand digit  
 seg\_port=value[num4];  
 Delay\_ms(5);  
 }  
 }  
}