```c
/*****************************************************
This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 3/5/2016
Author  : Reza
Company :
Comments:


Chip type               : ATmega64
Program type            : Application
AVR Core Clock frequency: 8.000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 1024
*****************************************************/
#include <mega64.h>
#include <delay.h>
//
//#DEFINE SS_DP PORTC //SSDP:Seven segement data port
//#DEFINE SSDPD DDRC //SSDD:Seven segement data port direction
//
//#DEFINE SS_AP PORTA//SSAP:Seven segement activation port
//#DEFINE SSAPD DDRA//SSDD:Seven segement activation port direction

#define PORT_OUTPUT 0XFF
#define PORT_INPUT  0X00
#define SS_0 0X3F
#define SS_1 0X06
#define SS_2 0X5B
#define SS_3 0X4F
#define SS_4 0X66
#define SS_5 0X6D
#define SS_6 0X7D
#define SS_7 0X07
#define SS_8 0X8F
#define SS_9 0X6F

#define TIMER_CLK_PRE_1024 0x05
#define TIMER_CTC 0x08

// Declare your global variables here
int OutVal1 = 1;
int OutVal2 = 1;
int OutVal3 = 1;
int OutVal4 = 9;

void ActivateSevenSegement(int index);
void SetSevenSegement(int OutNum);

// Timer1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
```

```
// Place your code here
    if(OutVal4 == 0){
        OutVal4 = 9;
        return;
    }
    OutVal4 --;

}

void RefreshSevenSegement(int val1,int val2,int val3,int val4)
{
    ActivateSevenSegement(1);
    SetSevenSegement(val1);
    delay_ms(1);
    ActivateSevenSegement(2);
    SetSevenSegement(val2);
    delay_ms(1);
    ActivateSevenSegement(3);
    SetSevenSegement(val3);
    delay_ms(1);
    ActivateSevenSegement(4);
    SetSevenSegement(val4);
    delay_ms(1);
}


void InitSevenSegement()
{
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
    PORTA=0x00;
    DDRA=PORT_OUTPUT;

    // Port C initialization
    // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
    // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
    PORTC=0x00;
    DDRC=PORT_OUTPUT;

    //enable first seven segement
    //PORTA |= 0x01;
}

void ActivateSevenSegement(int index)
{   //PORTA = 0x00;
    if( index > 0 && index <5)
        PORTA = (1 << (index - 1));
}

void InitTimer()
{
    // Timer/Counter 1 initialization
    // Clock source: System Clock
    // Clock value: 7.813 kHz
    // Mode: CTC top=OCR1A
    // OC1A output: Discon.
    // OC1B output: Discon.
    // OC1C output: Discon.
    // Noise Canceler: Off
    // Input Capture on Falling Edge
    // Timer1 Overflow Interrupt: Off
    // Input Capture Interrupt: Off
    // Compare A Match Interrupt: On
    // Compare B Match Interrupt: Off
    // Compare C Match Interrupt: Off

    TCCR1B=TIMER_CLK_PRE_1024 | TIMER_CTC;

    OCR1AH=0x1E;
    OCR1AL=0x85;

    // Timer(s)/Counter(s) Interrupt(s) initialization
    TIMSK=0x10;

}

void SetSevenSegement(int OutNum)
{
    switch (OutNum)
    {
        case 0:
            PORTC = SS_0;
            break;

        case 1:
            PORTC = SS_1;
            break;

        case 2:
            PORTC = SS_2;
            break;

        case 3:
            PORTC = SS_3;
            break;

        case 4:
            PORTC = SS_4;
            break;

        case 5:
            PORTC = SS_5;
            break;
        case 6:
            PORTC = SS_6;
            break;
        case 7:
            PORTC = SS_7;
            break;
         case 8:
            PORTC = SS_8;
            break;
         case 9:
            PORTC = SS_9;
            break;
         default:
            break;


    }
}


void main(void)
{
// Declare your local variables here

//Initialize timer
InitTimer();
//initialize seven segement
InitSevenSegement();

// Global enable interrupts
#asm("sei")

while (1)
      {
      // Place your code here
      //SetSevenSegement(OutVal);
      RefreshSevenSegement(OutVal1,OutVal2,OutVal3,OutVal4);
      }
}
```