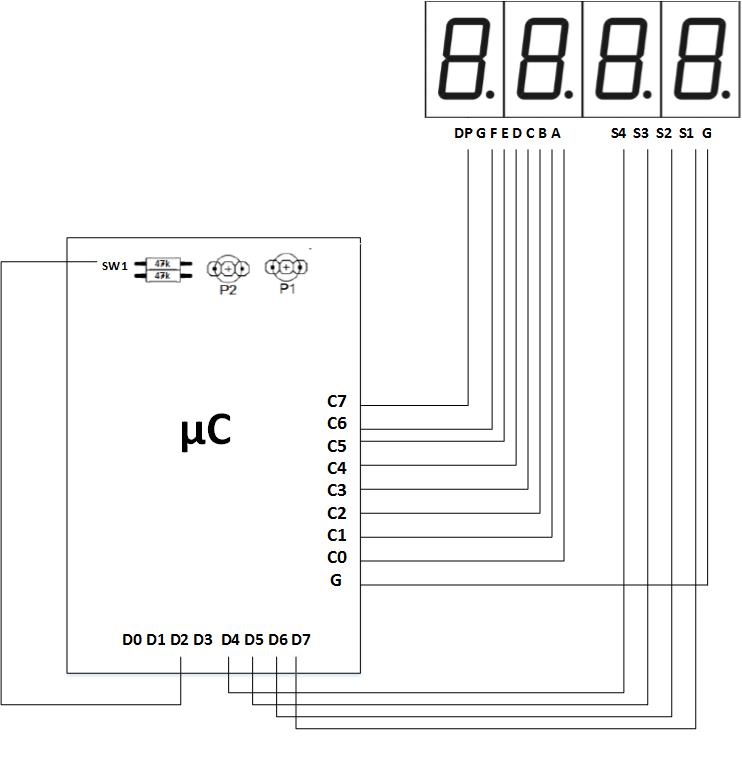
For developing a stopwatch, a push button switch is connected to INT0 (PD.2) pin of the ATmega32 microcontroller. Take one four digit seven segment display. Connect the data pins (A to DP) with PORTC and control pins (S1 to S4) with last four pins of PORTD (PD4 to PD7). If switch 1 is pressed for the first time, then the stopwatch will start counting. After the second press, time counting will pause and after the third press, it will reset.

For the code, the component connection should be as shown in the figure below.



Make the necessary connections according to the diagram. Connect the AVR kit with PC using the communication cable.

The code is developed in CVAVR. In this software, select ATmega32 as the chip with 16MHz frequency. Then select all pins of PORTC and last four pins of PORTD as the output pins. Enable the INT0 with falling edge. Select Timer 1. In Timer1 choose clock source, “15.625 kHz” in clock value and click on check box marked “Compare Match A Interrupt”. In Compare box enter the appropriate value that you obtain from your calculation for generating 1 second interval. Then generate the codes.

The code should be as follows:

#include <mega32.h>

#include <delay.h>

char sec=45,min=57,s1,s2,m1,m2,press=0;

char digit\_cathod[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

char digit\_cathod\_decimal[10]={0xBF,0x86,0xDB,0xCF,0xE6,0xED,0xFD,0x87,0xFF,0xEF};

void seperate\_digits(void)

{

s1=sec%10;

s2=sec/10;

m1=min%10;

m2=min/10;

}

void display\_digits(void)

{

PORTB=0b01111111;

PORTC=digit\_cathod[s1];

delay\_ms(10);

PORTB=0b10111111;

PORTC=digit\_cathod[s2];

delay\_ms(10);

PORTB=0b11011111;

PORTC=digit\_cathod\_decimal[m1];

delay\_ms(10);

PORTB=0b11101111;

PORTC=digit\_cathod[m2];

delay\_ms(10);

}

interrupt [EXT\_INT0] void ext\_int0\_isr(void)

{

press++;

if (press==3)

{

press=0;

min=0;

sec=0;

}

}

// Timer1 output compare A interrupt service routine

interrupt [TIM1\_COMPA] void timer1\_compa\_isr(void)

{

TCNT1H=0x00;

TCNT1L=0x00;

if (press==1)

{

sec++;

if (sec==60)

{

sec=0;

min++;

if (min==60) min=0;

}

}

}

void main(void)

{

DDRC=0xFF;

DDRB=0xFF;

TCCR1A=(1<<COM1A1) | (1<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (1<<CS12) | (0<<CS11) | (1<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x3D;

OCR1AL=0x09;

OCR1BH=0x00;

OCR1BL=0x00;

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (1<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

GICR|=(0<<INT1) | (1<<INT0) | (0<<INT2);

MCUCR=(0<<ISC11) | (0<<ISC10) | (1<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

GIFR=(0<<INTF1) | (1<<INTF0) | (0<<INTF2);

#asm("sei")

while (1)

{

seperate\_digits();

display\_digits();

}

}

After writing the code, build the project. Then run Extreme Burner software. Select Atmega32 as the chip, open the hex file, select write all and see the effect.