

```
/******
```

```
This program was produced by the  
CodeWizardAVR V2.03.8a Evaluation  
Automatic Program Generator  
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project : Braille Watch  
Version :  
Date : 12/3/2008  
Author : Freeware, for evaluation and non-commercial use only  
Company :  
Comments:
```

```
Chip type : ATmega8515  
Program type : Application  
Clock frequency : 8.000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 128  
*****/
```

```
#include <mega8515.h>  
#include <delay.h>
```

```
char sec = 0;  
char H1 = 1; //hard code values  
char H2 = 2;  
char M1 = 4 ;  
char M2 = 2;
```

```
char turnedOff=1;
```

```
// Timer 1 output compare A interrupt service routine  
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
```

```
{  
// Place your code here  
PORTB.0= ~PORTB.0;
```

```
    sec++;  
    turnedOff = PIND.2;  
    if(sec==60)  
    {  
        sec = 0;  
        M2++;  
    }
```

```
    if(M2==10)  
    {  
        M2=0;  
        M1++;  
    }
```

```
    if (M1 == 6)  
    {  
        M1 = 0;  
        H2 ++;  
    }
```

```
    if(H2 == 10)  
    {  
        H2 = 0;  
        H1++;  
    }
```

```
    if (H1 == 3)
```

```

{ H1=0; }

if(!turnedOff)
{
    turnedOff = 1; //set back to true
    PORTA.4 = 1;
    PORTA.5 = 1;
    PORTA.6 = 1;
    PORTA.7 = 1;

    delay_ms(1000);

    if(H1==0)
    {
        PORTA.4=0;
        PORTA.5=0;
        PORTA.7=0;
    }

    else if(H1 == 1)
    { PORTA.6=0;}

    else if (H1== 2)
    {
        PORTA.6=0;
        PORTA.7= 0;
    }

    //HOURS 1'S DIGIT DISPLAY

    delay_ms(1200);
    PORTA.4 = 1;
    PORTA.5 = 1;
    PORTA.6 = 1;
    PORTA.7 = 1;
    delay_ms(500);

    if(H2==0)
    {
        PORTA.4=0;
        PORTA.5=0;
        PORTA.7=0;
    }

    else if(H2 == 1)
    { PORTA.6=0;}

    else if (H2== 2)
    {
        PORTA.6=0;
        PORTA.7= 0;
    }

    else if(H2 == 3)
    {
        PORTA.4 = 0;
        PORTA.6 = 0;
    }

    else if(H2 == 4)
    {
        PORTA.4 = 0;
        PORTA.5 = 0;
        PORTA.6 = 0;
    }
}

```

```
else if(H2 == 5)
{
    PORTA.5 = 0;
    PORTA.6 = 0;
}

else if(H2 == 6)
{
    PORTA.4 = 0;
    PORTA.6 = 0;
    PORTA.7 = 0;
}

else if(H2 == 7)
{
    PORTA.4 = 0;
    PORTA.5 = 0;
    PORTA.6 = 0;
    PORTA.7 = 0;
}
else if(H2 == 8)
{
    PORTA.5 = 0;
    PORTA.6 = 0;
    PORTA.7 = 0;
}

else if(H2 == 9)
{
    PORTA.4 = 0;
    PORTA.7 = 0;
}

//Display Minutes tens digit

delay_ms(1200);
PORTA.4 = 1;
    PORTA.5 = 1;
    PORTA.6 = 1;
    PORTA.7 = 1;

    delay_ms(800);
    sec++;

if(M1==0)
{
    PORTA.4=0;
    PORTA.5=0;
    PORTA.7=0;
}

else if(M1 == 1)
{ PORTA.6=0;}
else if (M1 == 2)
{
    PORTA.6=0;
    PORTA.7= 0;
}

else if(M1 == 3)
{
    PORTA.4 = 0;
    PORTA.6 = 0;
}

else if(M1 == 4)
```

```

{
    PORTA.4 = 0;
    PORTA.5 = 0;
    PORTA.6 = 0;
}

else if(M1 == 5)
{
    PORTA.5 = 0;
    PORTA.6 = 0;
}

//Display minutes one's digit

delay_ms(1200);
PORTA.4 = 1; //reset motors to off
PORTA.5 = 1;
PORTA.6 = 1;
PORTA.7 = 1;

delay_ms(1200);

if(M2==0)
{
    PORTA.4=0;
    PORTA.5=0;
    PORTA.7=0;
}

else if(M2 == 1)
{ PORTA.6=0;}

else if (M2== 2)
{
    PORTA.6=0;
    PORTA.7= 0;
}

else if(M2 == 3)
{
    PORTA.4 = 0;
    PORTA.6 = 0;
}

else if(M2 == 4)
{
    PORTA.4 = 0;
    PORTA.5 = 0;
    PORTA.6 = 0;
}

else if(M2 == 5)
{
    PORTA.5 = 0;
    PORTA.6 = 0;
}

else if(M2 == 6)
{
    PORTA.4 = 0;
    PORTA.6 = 0;
    PORTA.7 = 0;
}

else if(M2 == 7)
{

```

```

        PORTA.4 = 0;
        PORTA.5 = 0;
        PORTA.6 = 0;
        PORTA.7 = 0;
    }

    else if(M2 == 8)
    {
        PORTA.5 = 0;
        PORTA.6 = 0;
        PORTA.7 = 0;
    }

    else if(M2 == 9)
    {
        PORTA.4 = 0;
        PORTA.7 = 0;
    }

    delay_ms(1200);
    PORTA.4 = 1; //reset motors to off
    PORTA.5 = 1;
    PORTA.6 = 1;
    PORTA.7 = 1;
}
}

```

// Declare your global variables here

```
void main(void)
```

```
{
```

// Declare your local variables here

// Input/Output Ports initialization

// Port A initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In Func2=In Func1=In Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=T State2=T State1=T State0=0

PORTA=0x00;

DDRA=0xF1;

// Port B 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

PORTB=0x00;

DDRB=0x00;

// Port C 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

PORTC=0x00;

DDRC=0x00;

// Port D 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

PORTD=0x00;

DDRD=0x00;

// Port E initialization

// Func2=In Func1=In Func0=In

// State2=T State1=T State0=T

PORTE=0x00;

DDRE=0x00;

// Timer/Counter 0 initialization

// Clock source: System Clock

```

// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OCO output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 62.500 kHz
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x03;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x3d;
OCR1AL=0x01;
OCR1BH=0x00;
OCR1BL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
EMUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x40;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;

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

while (1)
{
    // Place your code here
};
}

```