

Appendix A

System Code

/*****

This program was created by the

CodeWizardAVR V3.04 Evaluation

Automatic Program Generator

© Copyright 1998-2013 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :Hemo control System

Version : 1.0

Date : 12/16/2014

Author :Eng.Mazin Mohammed

Company :

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 16.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

```

#include <mega16.h>

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <delay.h>

#include <string.h>

// Alphanumeric LCD functions

#include <alcd.h>

// Global variables are Declared here

floata,b,c;

voidsdelay(){

unsigned char i,j,k;

for(i=0;i<6;i++)

for(j=0;j<255;j++)

for(k=0;k<255;k++);}

// Voltage Reference: AVCC pin

#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))

// Read the AD conversion result

unsignedintread_adc(unsigned char adc_input)

{

ADMUX=adc_input | ADC_VREF_TYPE;

// Delay needed for the stabilization of the ADC input voltage

```

```

delay_us(10);

// Start the AD conversion

ADCSRA|=(1<<ADSC);

// Wait for the AD conversion to complete

while ((ADCSRA & (1<<ADIF))==0);

ADCSRA|=(1<<ADIF);

return ADCW;

}

void main(void)

{

// Declare your local variables here

// Input/Output Ports initialization

// Port A initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (0<<DDA1) | (0<<DDA0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(0<<DDB2) | (0<<DDB1) | (0<<DDB0);

```

```

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) |
(0<<DDC2) | (0<<DDC1) | (0<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) |
(0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) |
(0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (0<<CS00);

```

```

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected

// OC1B output: Disconnected

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

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

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

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

```

```

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;

TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22)
| (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

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

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

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

```

```

MCUCSR=(0<<ISC2);

// USART initialization

// USART disabled

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is
// connected to the AIN0 pin

// The Analog Comparator's negative input is
// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC)
| (0<<ACIS1) | (0<<ACIS0);

// ADC initialization

// ADC Clock frequency: 125.000 kHz

// ADC Voltage Reference: AVCC pin

// ADC Auto Trigger Source: Free Running

ADMUX=ADC_VREF_TYPE;

ADCSRA=(1<<ADEN) | (0<<ADSC) | (1<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);

SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);

// SPI initialization

```

```

// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

// Alphanumeric LCD initialization

// Connections are specified in the

// Project|Configure|CCompiler|Libraries|Alphanumeric LCD menu:

// RS - PORTD Bit 0

// RD - PORTD Bit 1

// EN - PORTD Bit 2

// D4 - PORTD Bit 4

// D5 - PORTD Bit 5

// D6 - PORTD Bit 6

// D7 - PORTD Bit 7

// Characters/line: 8

lcd_init(8);

while (1)
{
    a=read_adc(1);
    b=read_adc(2);

```

```

    c=read_adc(0);
    a=(a*5.0/1023.0) ;
    b=(b*5.0/1023.0);
    c=(c*5.0/1023.0);
if ((a>=1.98)&(b>=1.98)&(c>=1.98))
    {lcd_clear();
    PORTC = 0x1C;
    lcd_gotoxy(0,0);
    lcd_putsf("valve is off");
    sdelay(); }
else
    {lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("valve is on");
    PORTC = 0x29 ;
    sdelay(); }
    }
}

```