```c
#include <stdlib.h>
#include <avr/io.h>
#include <stdint.h>
#include <avr/delay.h>
#include <string.h>
#define _BV(n) (1 << n)
#define F_CPU 4000000
#define LED5x PB1
#define LED10x PB0
#define LCD_Port PORTD      //Define LCD Port (PORTA, PORTB, PORTC, PORTD)
#define LCD_DPin  DDRD      //Define 4-Bit Pins (PD4-PD7 at PORT D)
#define RSPIN PD0           //RS Pin
#define ENPIN PD1           //E Pin

uint16_t adc_read(uint16_t adcx);         // Allows the ADC ports to be read
void adc_init();                          // Enables the ADC port
void LCD_Init (void);                     // Enables the LCD
void LCD_Clear();                         // CLears LCD screen and puts LCD cursor at line 1
void LCD_Print (char *str);               // Prints string on LCD
void LCD_LineJmp( unsigned char cmnd );   // Allows LCD Cursor to position at line 1/2
void LED (char *command );                // LED commands


int main(void)
{
    adc_init();         // Enables the ADC port
    LED("Enable");      // Enables the LED
    LCD_Init();         // Enables the LCD

    // Setting variables
    volatile uint16_t Vread10x = 0;
    volatile uint16_t Vread5x = 0;

    volatile uint16_t ADC0;
    volatile uint16_t ADC1;

    // Setting I.D variables
    volatile double Vres = 5.0/1024;
    volatile double Error = .88;    //0.9253382438;

    // Voltage difference
    volatile double Vmeas10x, Vmeas5x;

    while(1)
    {
            // Setting the ADC read out pin
            Vread10x = adc_read(0);
```

```
48              Vread5x = adc_read(1);
49
50              // Measuring the Voltage sum
51              Vmeas5x =  (float)Vread5x * Vres * 5 /Error;            //Vdiff for
                   Att 5x
52              Vmeas10x = (float)Vread10x * Vres * 10 / Error;        //Vdiff for
                  Att 10x
53
54              //value to be displayed
55              char DispVmeas5x[6];
56              char DispVmeas10x[6];
57              snprintf(DispVmeas5x,6,"%f", Vmeas5x);
58              snprintf(DispVmeas10x,6,"%f", Vmeas10x);
59
60              ADC0= Vread10x;
61              ADC1= Vread5x;
62
63              //Switch
64               if (ADC0 == 0 && ADC1 == 0)   // when connected theres no input
65                {
66                    LCD_Clear();                                //clears and starts
                       at position 1
67                    LCD_Print("         Connected to GND");
68                    LCD_LineJmp(0xC0);                          //Jumps to line two
69                    LCD_Print("Unit (V): ");
70                    LCD_Print(DispVmeas5x);                     //Displays Value
71                    LED("AOF");                                 // Turns off all
                        LED
72                    _delay_ms(1500);
73                    LED("AON");                                 // Turns on all LED
74                    _delay_ms(15000);
75                }
76               else if(ADC0 == ADC1 && ADC0 > 0 )  //WARNING CHECK Input
77                {
78                    for(int i=0; i < 3; i++)
79                    {  //Flashes
80                     LED("AON");                               //Turns on all LED
81                     LCD_Clear();                              // clears and starts
                         at position 1
82                     LCD_Print("           !!WARNING!!");
83                     LCD_LineJmp(0xC0);                        //Jumps to Line 2
84                     LCD_Print("   CHECK INPUT");
85                     _delay_ms(5000);
86                     LCD_Clear();                              // clears and starts
                         at position 1
87                     LED("AOF");                               //Turns off all LED
88                     _delay_ms(5000);
89                    }
90                 }
```

```
91                else if(ADC0 > 0 && ADC1 < ADC0)   //10x Position
92                {
93                    LED("Atx10");                              //Turns on 10x LED
94                    LCD_Clear();                               // clears and
                        starts at position 1
95                    LCD_Print("         HighVoltage <45V");
96                    LCD_LineJmp(0xC0);                         //jumps to line 2
97                    LCD_Print("Unit (V): ");
98                    LCD_Print(DispVmeas10x);                   //Displays Value
99                    _delay_ms(15000);
100               }
101               else  //if(ADC1 > 0 && ADC1 > ADC0)
102               {    //5x Position
103                    LED("Atx5");                               // Turn on 5x LED
104                    LCD_Clear();                               // clears and
                        starts at position 1
105                    LCD_Print("         Low Voltage <25V");
106                    LCD_LineJmp(0xC0);                         //jumps to line
                        two
107                    LCD_Print("Unit (V): ");
108                    LCD_Print(DispVmeas5x);                    // Displays Value
109                    _delay_ms(15000);
110               }
111           }
112   }
113
114   void LED (char *command )
115   {
116       if (command == "Enable")
117       {
118           DDRB |= _BV(LED10x);
119           DDRB |= _BV(LED5x);
120       }
121       else if(command == "Atx10")
122       {
123            PORTB &= ~(1<<LED5x);       // Turns off Low Voltage Light
124            PORTB |= _BV(LED10x);      //turns on High Voltage LED
125       }
126       else if (command == "Atx5")
127       {
128            PORTB &= ~(1<<LED10x);    //turns off High Voltage LED
129            PORTB |= _BV(LED5x);     // Turns on Low Voltage Light
130       }
131       else if(command == "AOF")
132       {
133           PORTB &= ~(1<<LED10x);    //turns off High Voltage LED
134           PORTB &= ~(1<<LED5x);    // Turns off Low Voltage Light
135       }
136       else if(command == "AON")
```

```
137     {
138         PORTB |= _BV(LED10x);     //turns on High Voltage LED
139         PORTB |= _BV(LED5x);     //turns on Low Voltage LED
140     }
141     else if(command == "Blink")
142     {
143         PORTB &= ~(1<<LED10x);    //turns off High Voltage LED
144         PORTB &= ~(1<<LED5x);    // Turns off Low Voltage Light
145         _delay_ms(1500);
146         PORTB |= _BV(LED10x);     //turns on High Voltage LED
147         PORTB |= _BV(LED5x);     //turns on Low Voltage LED
148     }
149     else
150     {
151         return 0;
152     }
153 }
154
155 void LCD_Init (void)
156 {
157     LCD_DPin = 0xFF;        //Control LCD Pins (D4-D7)
158     _delay_ms(15);      //Wait before LCD activation
159     LCD_LineJmp(0x02);  //4-Bit Control
160     LCD_LineJmp(0x28);       //Control Matrix @ 4-Bit
161     LCD_LineJmp(0x0c);       //Disable Cursor
162     LCD_LineJmp(0x06);       //Move Cursor
163     LCD_LineJmp(0x01);       //Clean LCD
164     _delay_ms(2);
165 }
166
167 void LCD_LineJmp( unsigned char cmnd )
168 {
169     LCD_Port = (LCD_Port & 0x0F) | (cmnd & 0xF0);
170     LCD_Port &= ~ (1<<RSPIN);
171     LCD_Port |= (1<<ENPIN);
172     _delay_us(1);
173     LCD_Port &= ~ (1<<ENPIN);
174     _delay_us(200);
175     LCD_Port = (LCD_Port & 0x0F) | (cmnd << 4);
176     LCD_Port |= (1<<ENPIN);
177     _delay_us(1);
178     LCD_Port &= ~ (1<<ENPIN);
179     _delay_ms(2);
180 }
181
182 void LCD_Clear()
183 {
184     LCD_LineJmp (0x01);     //Clear LCD
185     _delay_ms(2);           //Wait to clean LCD
```

```
186     LCD_LineJmp (0x80);     //Move to Position Line 1, Position 1
187 }
188 
189 
190 void LCD_Print (char *str)
191 {
192     int i;
193     for(i=0; str[i]!=0; i++  )
194     {
195         LCD_Port = (LCD_Port & 0x0F) | (str[i] & 0xF0);
196         LCD_Port |= (1<<RSPIN);
197         LCD_Port|= (1<<ENPIN);
198         _delay_us(1);
199         LCD_Port &= ~ (1<<ENPIN);
200         _delay_us(200);
201         LCD_Port = (LCD_Port & 0x0F) | (str[i] << 4);
202         LCD_Port |= (1<<ENPIN);
203         _delay_us(1);
204         LCD_Port &= ~ (1<<ENPIN);
205         _delay_ms(2);
206     }
207 }
208 
209 // Enables
210 void adc_init()
211 {
212     ADMUX = (1<<REFS0) | (0<<REFS1);
213     // ADC Enable and prescaler of 128
214     // 8000000/128 = 62500
215     ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
216 }
217 
218 // reads the ADC port
219 uint16_t adc_read(uint16_t adcx)
220 {
221     ADMUX &= 0xf0;
222     ADMUX |= adcx;
223 
224     ADCSRA |= _BV(ADSC);
225     while( ADCSRA & _BV(ADSC)  );
226 
227     return (double)ADC;
228 }
229 
230 /*//Write on a specific location
231 void LCD_Printpos (char row, char pos, char *str)
232 {
233     if (row == 0 && pos<16)
234     LCD_LineJmp((pos & 0x0F)|0x80);
```

```
235     else if (row == 1 && pos<16)
236     LCD_LineJmp((pos & 0x0F)|0xC0);
237     LCD_Print(str);
238 }    */
239
240
241
```