Experiment Name: Analog signal input in the microcontroller or Display ADC value in the virtual terminal.

A/D Theoretical Background:

The role of the ANALOG-TO-DIGITAL CONVERTER (A/D) is to convert analog voltage values to digital values. Let's explore the principle of operation of the A/D converter:

The ATD CONVERTER converts analog voltage to binary numbers. These binary numbers can be in different length - 2, 4, 8, 10-bit. The more bits the binary number has, the higher the resolution of the - A/D.

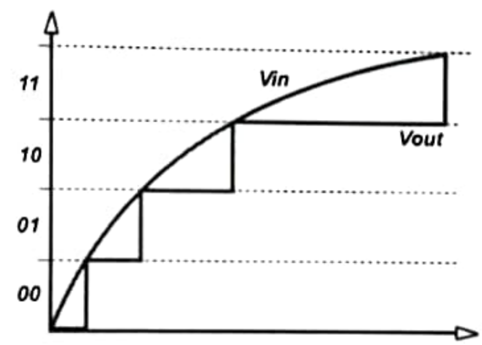
For example: Suppose that the voltage that supplied to the A/D converter varies from 0 to 5 volt, and the A/D converter converts the input voltage to a binary number of two-bits.

With two bits, we can ONLY display 4 different options:

00	01	10	11

That is, we can show the changes from 0 to 5 volt with 4 numbers, or more precisely four levels.

You can see the 4 levels in the following illustration:



BLUE line describes the changes in the input voltage of the ANALOG-TO-DIGITAL CONVERTER (A/D) of the microcontroller. RED line represents the digital levels at the output of the ANALOG-TO-DIGITAL CONVERTER (A/D) of the microcontroller.

We can see that the red signal far from being ideal, i.e. not close enough to the original analog input voltage values. Thus, we can say that A/D with the binary number of two-bits has a low resolution and there is a large gap between the real value of the analog input voltage and the values represented by the A/D.

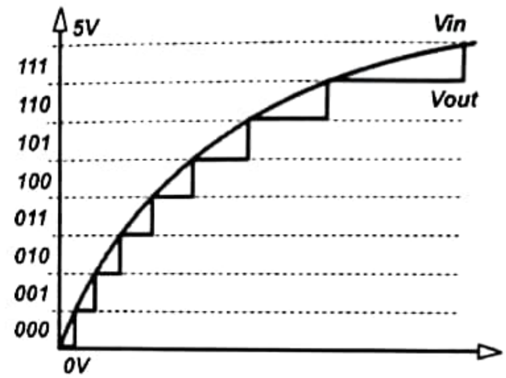
Now, suppose that the voltage that supplied to the A/D converter is still varies from 0 to 5 volt, however, the A/D converter converts the input voltage to a binary number of three-bits.

With three bits we can get 8 different options:

000	001	010	011	100	101	110	111

That is, we can show the changes from 0 to 5 volt with 8 numbers, or more precisely 8 levels.

You can see the eight levels in the following illustration:



Now we can see that the RED line represents the original signal "better" than the previous RED line. The gap between the analog signal and the digital signal smaller compared to the previous graph. Based on the "good" results that we received, we can say that current A/D converter has a high-resolution compare to previous case.

Therefore we can say that the ANALOG TO DIGITAL CONVERTER (A/D) of the microcontroller with a larger amount of bits has a higher resolution and better accuracy when converting from analog signal to digital signal.

There is another fact to mention about the resolution of the A/D converter. Since the converter converts the signals, it takes a certain time. The conversion time of the low resolution A/D takes less time than the conversion time of the high resolution A/D.

When you are planning special systems, you have to take into account this fact. If you are planning to build accurate and fast systems, you have to consider carefully which convert to choose: if you select A-to-D with high resolution – system will not be as "fast; but if you select an A-to-D with high-speed response – you will loose the resolution of the system.

How to convert analog values to digital values:

If we decide to work with an analog to digital converter (A/D) with three-bit length, we obtain eight different binary numbers which represent different voltage levels. For example:

Voltage levels [V]	Binary representation
0-0.62	000
0.621-1.25	001
1.251-1.87	010
1.871-2.5	011
2.51-3.12	100
3.121-3.75	101
3.751-4.37	110
4.371-5.00	111

In this example, the analog voltage values from 0v to 0.62v have a binary representation of 000; the analog voltage values from 0.621v to 1.25v have a binary representation of 001 so on.

The ADCON module located within the PIC microcontroller has a resolution of ten-bit length. Therefore, the converter can divide the analog input voltage between 0v and 5v to 2^ 10 levels, which are 1024 levels. We can say that the resolution of this component is very high.

How do we know what is the binary value/representation of the analog input voltage?

We can use the triangle method to calculate/find the binary representation of an analog input voltage. For example, lets calculate/find the binary value representation on the analog input voltage of 3.65 volt:

$$\begin{Bmatrix} 5V & \to 1024 \\ 3.65V & \to X \end{Bmatrix} \to X = \frac{1024 * 3.65}{5} = 747.52 \approx 748$$

The analog input voltage of 3.65v will be represented by decimal number 748 or by binary number 1011101100.Using similar way we can find a binary representation for any desired level of the analog input voltage.

Circuit Diagram:

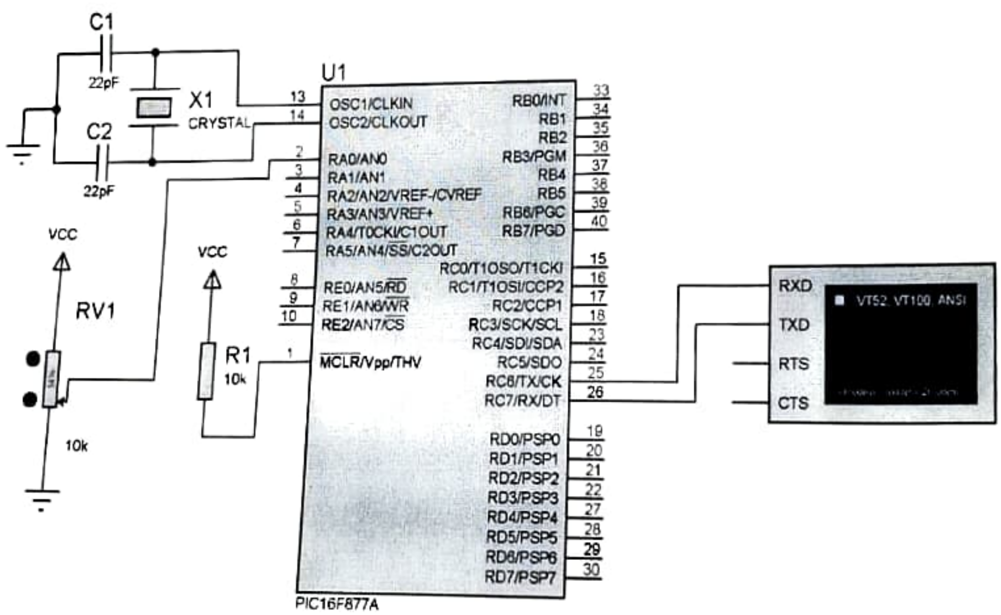


Figure: Displaying ADC value in Virtual Terminal

MikroC Code:

```
//create a variable that will hold the ADC value
int valADC;
//create a char array
char x[4];
void main()
{
  //initialize UART
  UART1_Init(9600);
```

```
//initialize ADC
ADC_Init();

//create a loop
while(1)
{
    //Read ADC value in RA0
    valADC = ADC_Read(0);
    //convert into string/char array
    IntToStr(valADC,x);
    //Print
    UART1_Write_Text("Analog value = ");
    UART1_Write_Text(x);
    //clear char array
    strcpy(x,"");
    UART1_Write(13);
    Delay_ms(1000);


    }
}
```

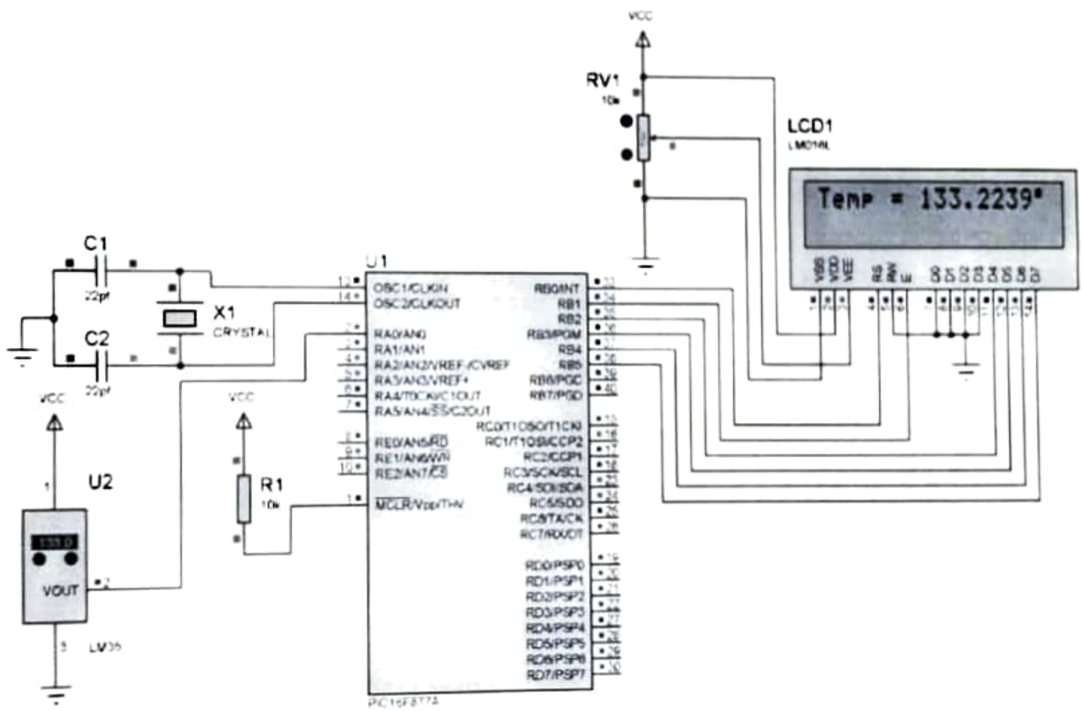
Experiment Name: LM35 Temperature Sensor Data Read and Display Using LCD Module

LM35 Temperature Sensor:

- LM35 is a temperature measuring device having an analog output voltage proportional to the temperature.
- It provides output voltage in Centigrade (Celsius). It does not require any external calibration circuitry.
- The sensitivity of LM35 is 10 mV/degree Celsius. As temperature increases, output voltage also increases. E.g. 250 mV means 25°C.

- It is a 3-terminal sensor used to measure surrounding temperature ranging from -55 °C to 150 °C.
- LM35 gives temperature output which is more precise than thermistor output.

Circuit Diagram:



MikroC Code:

```
// LCD module connections
sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;

sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
```

```
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;
// End LCD module connections

char display[16]="";

void main()
{
unsigned int result;
float volt,temp;

trisb=0x00;
trisa=0xff;
adcon1=0x80;
lcd_init();
lcd_cmd(_lcd_clear);
lcd_cmd(_LCD_CURSOR_OFF);

        while(1)
        {
                result=adc_read(0);
                volt=result*4.88;
                temp=volt/10;

                lcd_out(1,1,"Temp = ");

                floattostr(temp,display);
                lcd_out_cp(display);
                lcd_chr(1,16,223); //print at pos(row=1,col=13) "°" =223 =0xdf
                lcd_out_cp("  C"); //celcius

        }

}
```