PSG COLLEGE OF TECHNOLOGY

(AUTONOMOUS INSTITUTION)

COIMBATORE – 641004

ELECTRONICS AND COMMUNICATION ENGINEERING

19L511- MICROPROCESSORS AND MICROCONTROLLERS LABORATORY

TOPIC: Get the input from ADC and map it to a sensor to show the temperature from 10^{o}C to 30^{o}C in led display

SUBMITTED BY,

20L147 - R SURESHKUMAR

20L148 - M TAMIZHARASAN

MAPPING ADC TO TEMPERATURE

AIM:

To obtain values from ADC, map them to the temperature values from 10^{o}C to 30^{o}C and display them on a led display.

SOFTWARE REQUIRED:

KEIL IDE, Program ISP

HARDWARE REQUIRED:

8051 Microcontroller, ADC 0804, 7 Segment display

THEORY:

ADC0804 is a very commonly used 8-bit analog to digital convertor. It is a single channel IC, i.e., it can take only one analog signal as input. The digital outputs vary from 0 to a maximum of 255. The step size can be adjusted by setting the reference voltage at pin9. When this pin is not connected, the default reference voltage is the operating voltage, i.e., V_{cc}. The step size at 5V is 19.53mV (5V/255), i.e., for every 19.53mV rise in the analog input, the output varies by 1 unit. To set a particular voltage level as the reference value, this pin is connected to half the voltage. For example, to set a reference of 4V (V_{ref}), pin9 is connected to 2V ($V_{ref}/2$), thereby reducing the step size to 15.62mV (4V/255).

ADC0804 needs a clock to operate. The time taken to convert the analog value to digital value is dependent on this clock source. An external clock can be given at the Clock IN pin. ADC 0804 also has an inbuilt clock which can be used in absence of external clock. A suitable RC circuit is connected between the Clock IN and Clock R pins to use the internal clock.

CS stands for Chip select. CS Pin is used to turning on the device. When we are using multiple ADC then this pin is used to select the required device at that time. It's active at a low state. RD stands for reading. It is an input pin. RD pin is used when we want to receive the output value from the internal register. High to Low pulse will be used to perform the operation. WR stands for Write. WR is an input pin used to start the conversion by applying high to low pulse. INTR is an interrupt pin used to indicate that conversion is complete. It goes low when the conversion is completed. To use the internal Clock just use the RC circuit with a 10K resistor and 100pF capacitance at the clock pin. By using these standards ADC0804 will be able to operate.

After setting up everything on ADC, we have to do calculation for mapping digital values ranging from 0 to 255 to temperature values ranging from 10°C to 30°C. For this conversion, we have used proportionality constant.

$$k=0.0784313725$$

By multiplying this constant with obtained digital value, we can map values as follows,

$$0 \rightarrow 0$$
$$255 \rightarrow 20$$

To make it ranging from 10 to 30, we have to add 10 with the above procedure and then forming final formula as,

$$Temperature = (digital\ value * k) + 10$$

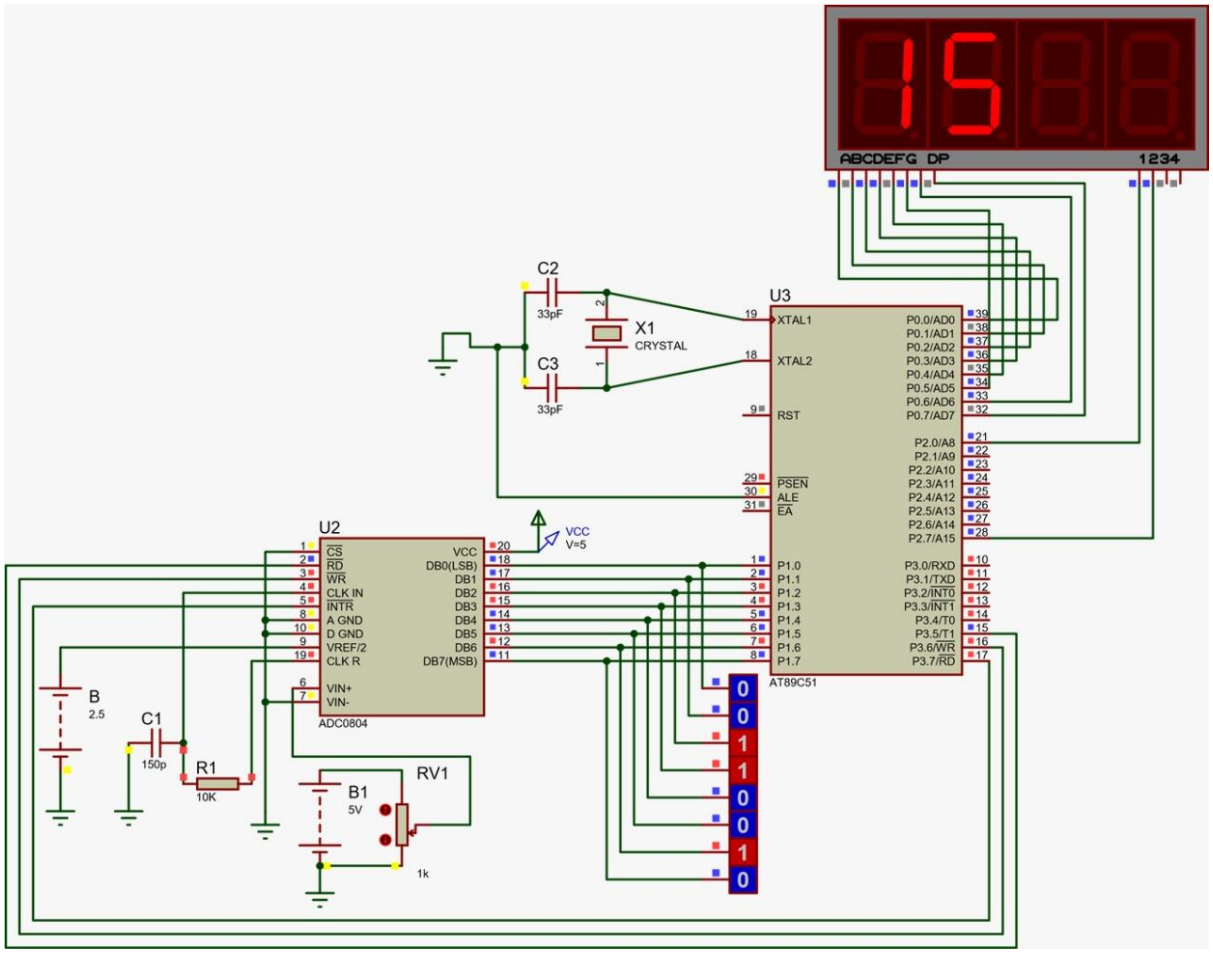
After adding 10, we get the output as,

$$0 \rightarrow 10$$
$$255 \rightarrow 30$$

After converting the values, we have used array to find corresponding hexadecimal code to be loaded into port where 7 segment led is connected. And finally the values are changed from 10 to 30 in display according to the change in voltage input to ADC which is varied using potentiometer.

FLOWCHART:

CIRCUIT DIAGRAM:



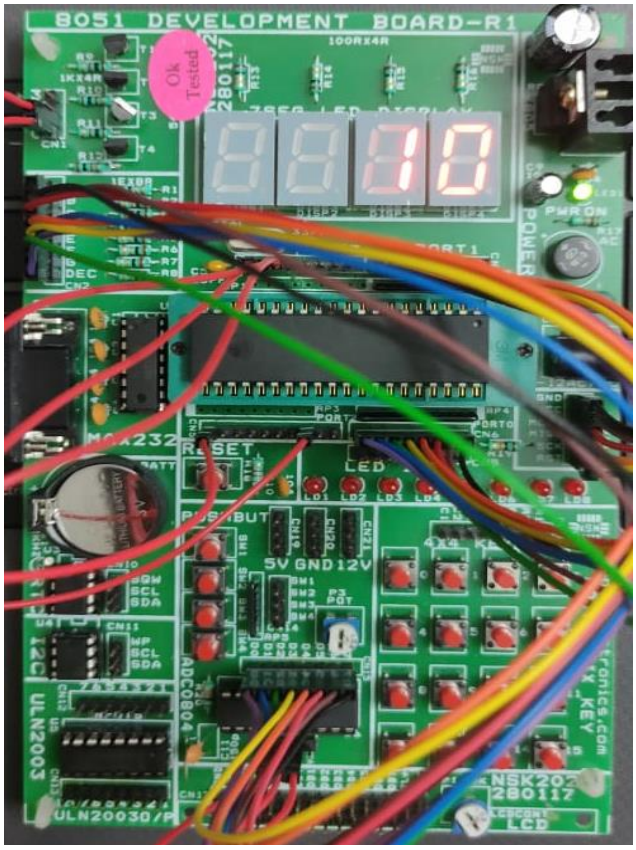
CODE:

```
#include<reg51.h>
#define k 0.0784313725
unsigned int d,d1,d2,j;
int ar[50]={0x3f,0x06,0x5b,0x4f,0x66,
0x6d,0x7d,0x07,0x7f,0x6f};   //for cathode 7 seg
void delay(int n);
float a,t;
int adc();
sbit rd=P3^5;
sbit wr=P3^6;
sbit intr=P3^7;
```

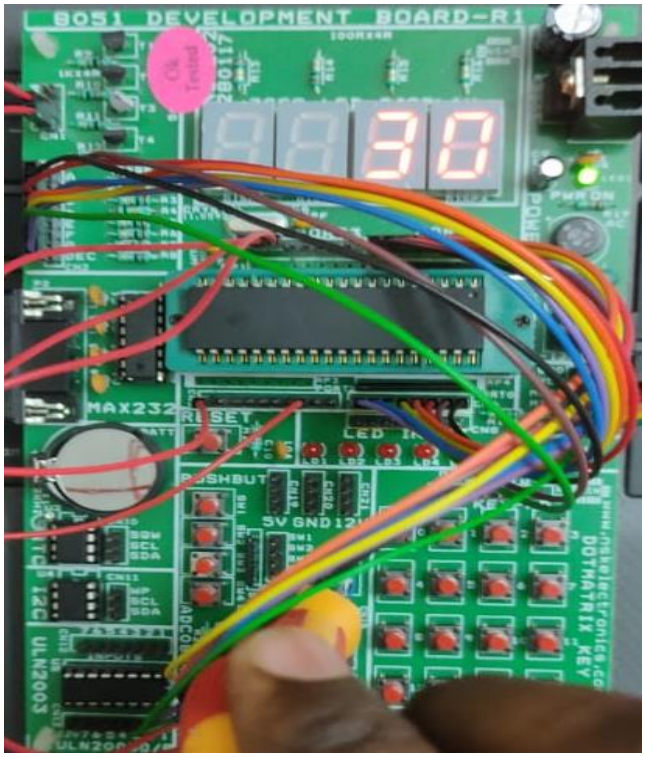
```
void main()
{
  while(1){
    a=adc();
    t=(a*k)+10;
    d=t;
    d1=d/10;
    d2=d%10;
    for(j=0;j<=10;j++)
    {
      P2=0x0f;
      P0=ar[d1];
      delay(100);
      P2=0xf0;
      P0=ar[d2];
      delay(100);
    }
    P2=0x00;
  }
}
int adc()
{
  P1=0xff; //P1 is connected to ADC
  rd=1;
  wr=0;
  wr=1;
  //P1=b;
  while(intr==1);
  rd=0;
  //b++;
  return P1;
}
void delay(int n)
{
  for(j=0;j<=n;j++);
}
```

OUTPUT:

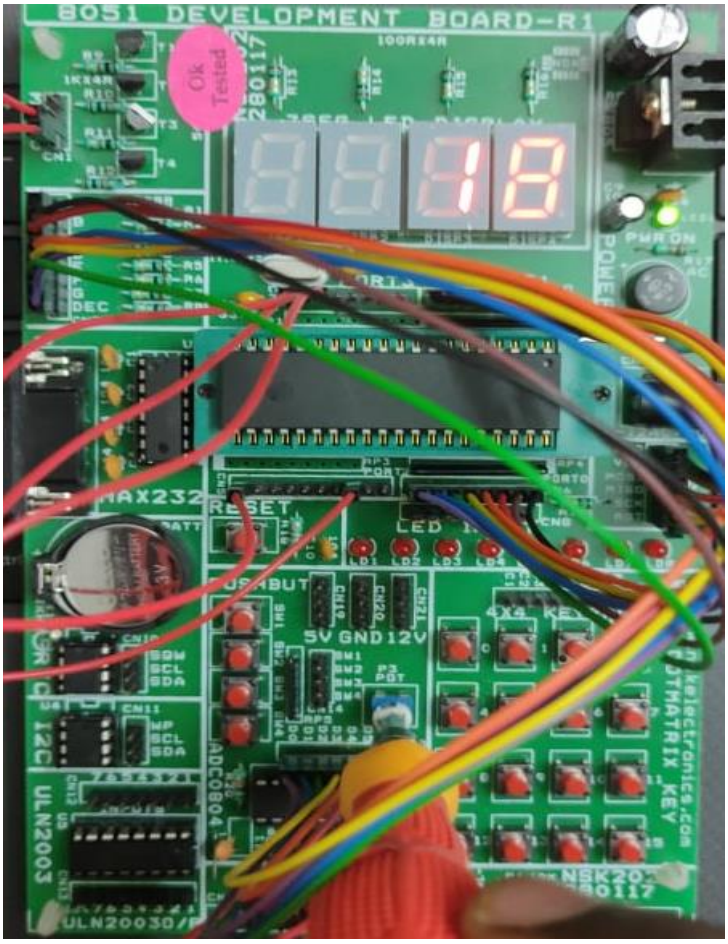
(0 → 10ºC)



(255 → 30ºC)



(102 → 18ºC)



RESULT:

Thus ADC output is obtained and mapped to the temperature values from 10ºC to 30ºC and then displayed it using 7 segment led.