FYMCA-B **SEM-II** **DATE:19/05/2022**

IOT LAB **PRACTICAL NO:04** **ROLL NO: 24**

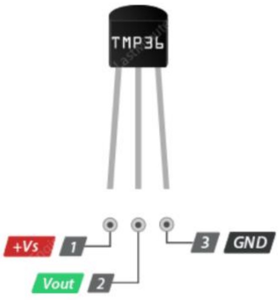
AIM: Programs based on interfacing DHT11/TMP36 temperature sensor Programs based on interfacing Passive infrared sensors (PIR), Ultrasonic of Arduino.

THEORY:

TMP36 Temperature Sensor:

The TMP36 is a low voltage, precision centigrade temperature sensor manufactured by Analog Devices. It is a chip that provides a voltage output that is linearly proportional to the temperature in °C and is, therefore, very easy to use with an Arduino.

The TMP36 temperature sensor is fairly precise, never wears out, works under many environmental conditions and requires no external components to work. In addition, the TMP36 sensor does not require calibration and provides a typical accuracy of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range.



+Vs is the power supply for the sensor which can be anywhere between 2.7V to 5.5V.

Vout pin produces an analog voltage that is directly proportional (linear) to the temperature. It should be connected to an Analog (ADC) input.

GND is a ground pin.

16x2 LCD:

It is called 16x2 Because you can write 16 characters or numbers in column wise and 2 in row wise. This display has total of 16 pins. Here I only use 12 pins. Here we use the pins except D0, D1, D2, D3. Because here I interface the LCD in 4-bit mode.

GND(VSS): Connect the ground pin of the power supply to this pin.

VCC: Connect this pin to 5v

Contrast (VEE): This pin is used to adjust the contrast of Display. Connect a potentiometer (POT) to this pin. Rotate the knob of the POT to adjust the contrast.

RS: RS pin means Register select pin. Selects command register when the pin is LOW. And selects data register when this pin is HIGH.

RW: It represent the Read Write pin. When this pin is LOW, the MCU write to register. And when the pin is HIGH, MCU read from the register. Here we want to write. It connects it permanently to GND.

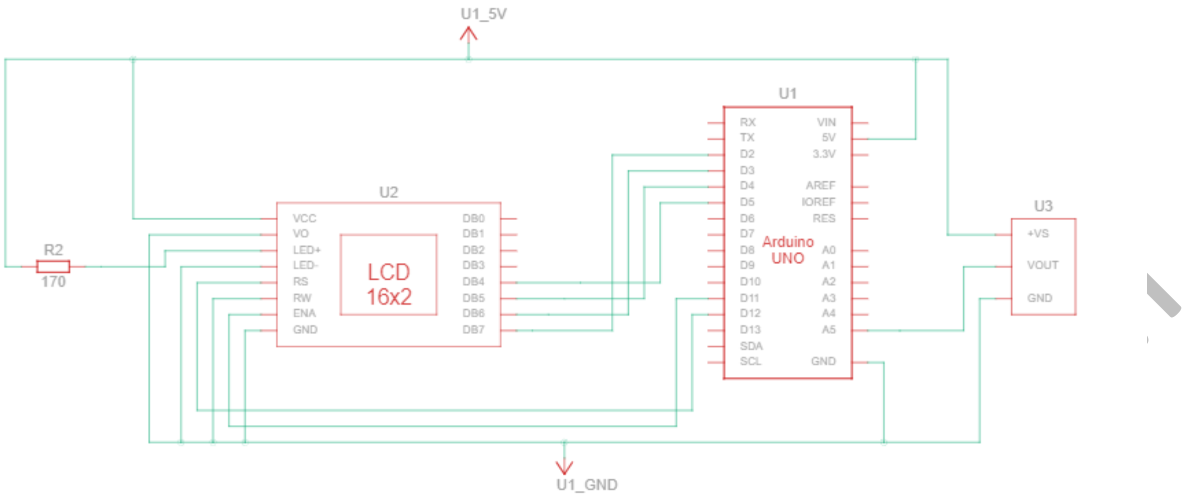
EN (E): EN pin means the Enable pin. Send data to data pins when a HIGH to LOW pulse is given.

D0-D7 (DB0-DB7): These are 8 data pins. Here I interface this LCD with Arduino is in 4 bit mode. So we need only D4 to D7

Backlight(+): This is the anode pin of the backlight of the display

Backlight(-): This is the cathode pin of the backlight of the display

CIRCUIT DIAGRAM:



SOURCE CODE:

```
// include the library code for LCD display:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Defining pin
#define temp A5

void setup()
{
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  pinMode(temp, INPUT);
  Serial.begin(9600);
  lcd.clear();
  lcd.print("Temperature: ");
}
float pre_temp = 0;
void loop() {
  float temperature = 0;
  temperature = (analogRead(temp) * 0.48828125)- 49.95 ;
  if(pre_temp != temperature)
  {
    lcd.setCursor(0,1);
    lcd.print("                ");
  }
  lcd.setCursor(0,1);
  lcd.print(temperature);
  lcd.print(" C");
  pre_temp = temperature;
}
```

Note:

0.48828125 where this number came from??

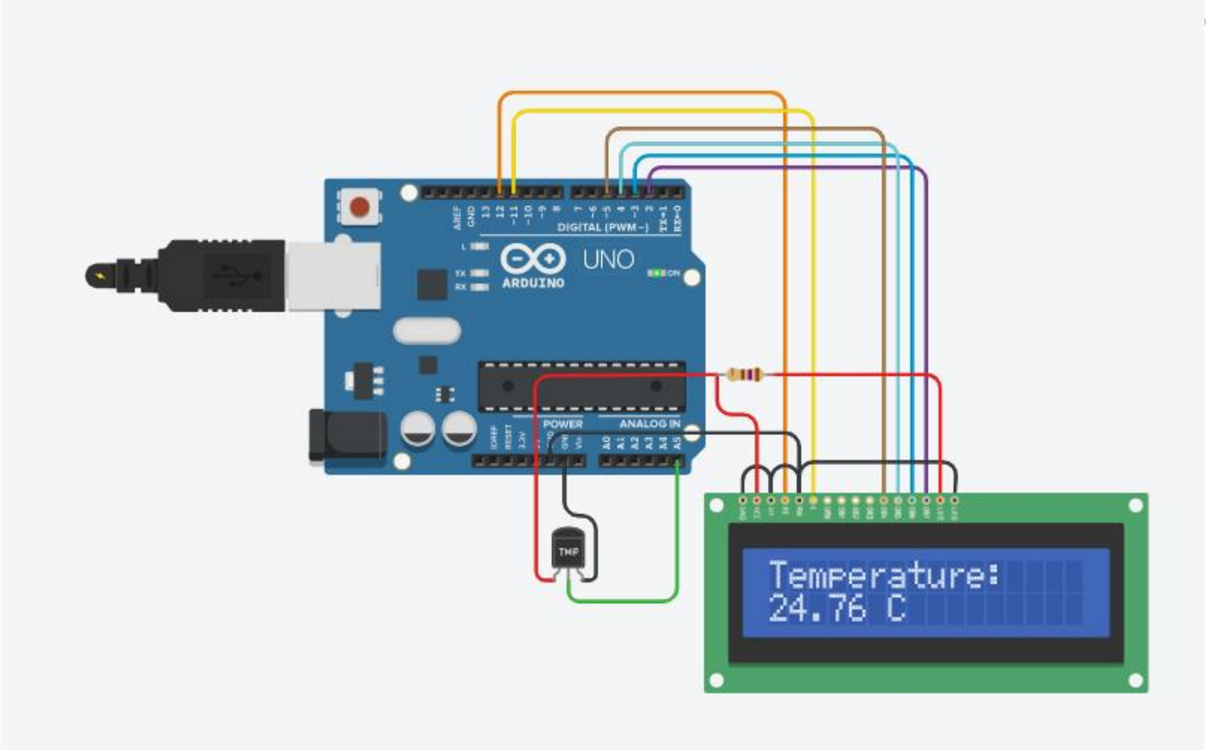
This is **(+Vcc * 1000 / 1024) / 10**

Where +Vcc is the supply voltage = +5V, 1024 is 2^10, value where the analog value can be represented by ATmega the actual voltage obtained by VOLTAGE_GET / 1024.

1000 is used to change the unit from V to mV & 10 is a constant as each 10 mV is directly proportional to 1 Celsius in LM35.

So (5.0 * 1000 / 1024) / 10 = 0.48828125.

OUTPUT:



```
Temperature:
24.76 C
```

CONCLUSION:

From this practical, I have learned and implemented the temperature sensor (TMP36) with LCD display in arduino.