EXPERIMENT NO. 11 (Group B)

- **Aim:** Write a program read the temperature sensor and send the values to the serial monitor on the computer

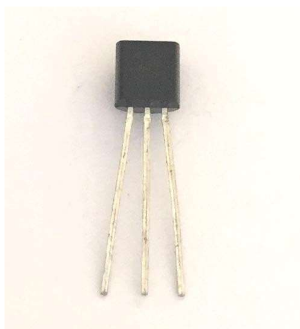
- **Outcome:** Understanding working principle of DHT11, LM35 temperature sensor, Relationship between different temperature scales

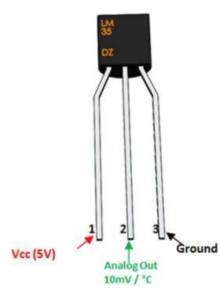
- **Hardware Requirement:** Arduino, LED, LM35, DHT11, etc

- **Software Requirement**: Arduino IDE

- **Theory:**

LM35 Temperature Sensor





LM35 Temperature Sensor Pinout
LM35 Sensor Pinout Configuration

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Connected to ground of circuit

- **LM35 Sensor Features**
  - Minimum and Maximum Input Voltage is 35V and -2V respectively. Typically 5V.
  - Can measure temperature ranging from -55°C to 150°C
  - Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of 10mV (0.01V) for every 1°C rise in temperature.
  - ±0.5°C Accuracy
  - Drain current is less than 60uA
  - Low cost temperature sensor
  - Small and hence suitable for remote applications
  - Available in TO-92, TO-220, TO-CAN and SOIC package

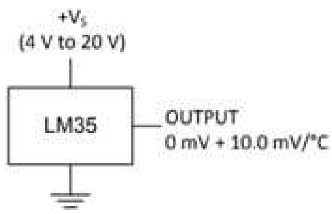
- **LM35 Temperature Sensor Equivalent**

LM34, DS18B20, DS1620, LM94022

How to use LM35 Temperature Sensor:

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.

Power the IC by applying a regulated voltage like +5V (VS) to the input pin and connected the ground pin to the ground of the circuit. Now, you can measure the temperature in form of voltage as shown below.



If the temperature is 0°C, then the output voltage will also be 0V. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature. The voltage can converted into temperature using the below formulae.

$$V_{OUT} = 10\ \text{mv/°C} \times T$$

where

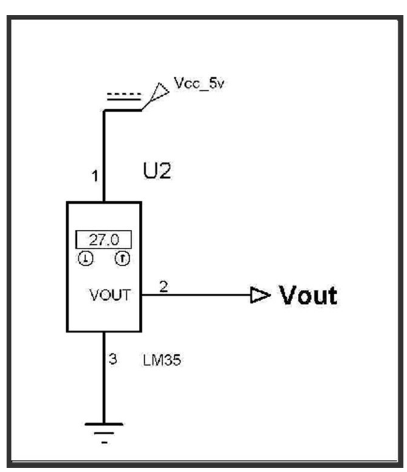
- V_{OUT} is the LM35 output voltage
- T is the temperature in °C

- **LM35 Temperature Sensor Applications**
  - Measuring temperature of a particular environment
  - Providing thermal shutdown for a circuit/component
  - Monitoring Battery Temperature
  - Measuring Temperatures for HVAC applications.

- **How Does LM35 Sensor Work?**

Main advantage of LM35 is that it is linear i.e. 10mv/°C which means for every degree rise in temperature the output of LM35 will rise by 10mv. So if the output of LM35 is 220mv/0.22V the temperature will be 22°C. So if room temperature is 32°C then the output of LM35 will be 320mv i.e. 0.32V.

- **LM35 Interfacing Circuit**



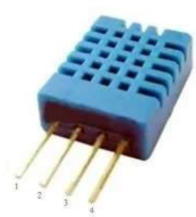
As such no extra components required to interface LM35 to ADC as the output of LM35 is linear with 10mv/degree scale. It can be directly interfaced to any 10 or 12 bit ADC. But if you are using an 8-bit ADC like ADC0808 or ADC0804 an amplifier section will be needed if you require to measure 1°C change.
LM35 can also be directly connected to Arduino. The output of LM35 temperature can also be given to comparator circuit and can be used for over temperature indication or by using a simple relay can be used as a temperature controller.

- **DHT11 interfacing with arduino and weather station**

DHT11 sensor is used to measure the **temperature** and **humidity**. It has a resistive humidity sensing component and a negative temperature coefficient (NTC). An 8 bit MCU is also connected in it which is responsible for its fast response. It is very inexpensive but it gives values of both temperature and humidity at a time.

- **Specification of DHT11**
  - It has humidity range from 20 to 90% RH
  - It has temperature range from 0 – 50 C
  - It has signal transmission range of 20 m
  - It is inexpensive
  - It has fast response and it is also durable

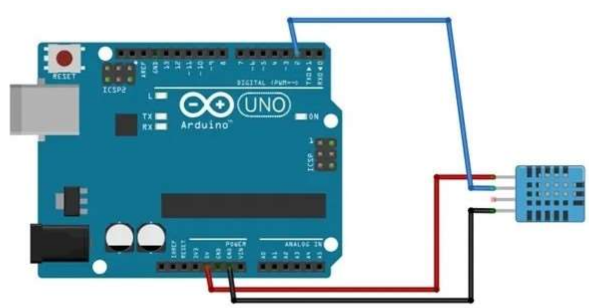
- **DHT11 Pin out**



  - The first pin of the DHT11 is vcc pin.
  - The second pin of the DHT is Data pin.
  - The third pin is not used.
  - The fourth pin of the DHT sensor is ground.

- **DHT11 interfacing with arduino**

First of all connect the ground and the VCC of the DHT11 temperature and humidity sensor to the ground and 5v of the **Arduino**. Then connect the data pin of the DHT11 sensor to the pin 2 of the Arduino.



- **Installing the DHT11 Library**

To run the following code in Arduino IDE you will first have to install the DHT library in you Arduino directory.

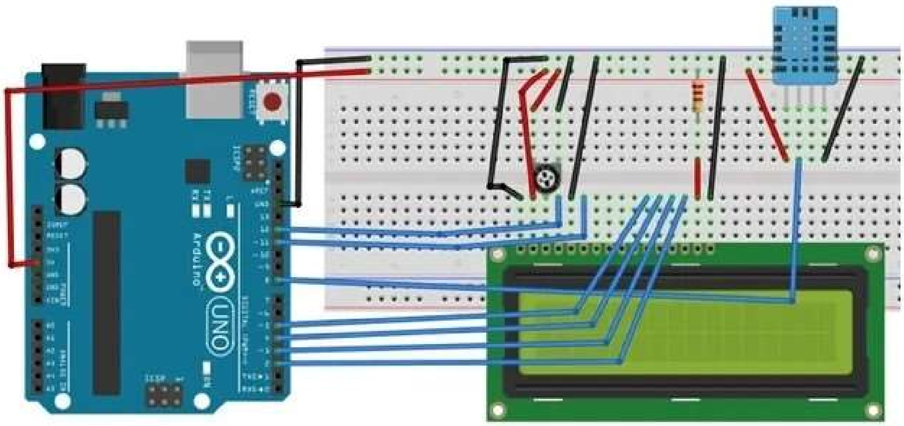
Download the zip file from here and place it in your Arduino library folder. The path to Arduino library folder for my computer is
Documents/ Arduino/ Libraries
Unzip the downloaded file and place it in this folder.
After copying the files, the Arduino library folder should have a new folder named DHT containing the dht.h and dht.cpp. After that copy the following code in the Arduino IDE and upload the code.

- **Code of DHT11 interfacing with arduino**

```
// Code for DHT11 Temperature and humidity sensor.
 #include " DHT.h "    // including the library of DHT11 temperature and
humidity sensor
 #define DHTPIN 2      // Selecting the pin at which we have connected
DHT11
 #define DHTTYPE DHT11 // Selecting the type of DHT sensors
 DHT dht ( DHTPIN, DHTTYPE ) ;
 void setup ( ) {
   Serial.begin ( 9600 ) ;
   dht.begin (  ) ;    // The sensor will start working
 }
 void loop ( ) {
   // Reading temperature or humidity may take about 2 seconds because it
is a very slow sensor.

float humidity = dht.readHumidity ( ) ; // Declaring h a variable and
storing the humidity in it.
  float temp = dht.readTemperature ( ) ; // Declaring t a variable and
storing the temperature in it.
  // Checking if the output is correct. If these are NaN, then there is
something in it.
   if ( isnan ( t ) || isnan ( h ) ) {
      Serial.println ( " Sensor not working " ) ;
   }
 else
{

Serial.print ( " Temp is " ) ;
      Serial.print ( temp ) ;          // Printing the temperature on
display.
      Serial.println ( " *C " ) ;      // Printing “ *C ” on display.
      Serial.print ( " Humidity in % is : " ) ;
      Serial.print ( humidity ) ;      // Printing the humidity on display
      Serial.print ( " % \t " ) ;      // Printing “%” on display

   }
 }
```

- **Weather Station using DHT11 and arduino**

In this example we will make a weather station that will sense the humidity and temperature and will show it on the lcd attached to the Arduino. Make the circuit as shown in the diagram. The resistor in the circuit will make the black light darker. We have used the 220 ohm resistor but you can use any resistor having value near to that. The potentiometer we used in the circuit is used to set the screen contrast. We have used the 10 K ohm value but you can choose any value relative to that one.



- **Components Required**
    - Arduino Uno (you can use any)
    - 16 x 2 LCD
    - DHT11 Temperature and humidity sensor
    - 10 K ohm potentiometer
    - 220 ohm resistor

- **Code of weather station using arduino and DHT11**

```
// This code is for the weather station using the DHT11 humidity and temperature
sensor.
// Install the library of the DHT before uploading the code in the Arduino IDE
#include < dht.h >      // including the DHT library
#include < LiquidCrystal.h >   // including the LCD library
LiquidCrystal lcd ( 12, 11, 5, 4, 3, 2 ) ;  // initializing the lcd pins
dht DHT ;                        // declaring dht a variable
#define DHT11_PIN 8               // initializing pin 8 for dht
void setup ( ) {
  lcd.begin ( 16, 2 ) ;       // starting the 16 x 2 lcd
}
void loop ( )
{
  int chk = DHT.read11(DHT11_PIN ) ;         // Checking that either the dht is
working or not
  lcd.setCursor ( 0, 0 ) ;                  //  starting the cursor from  top left

  lcd.print ( " Temperature is : " ) ;       // printing the “ Temperature is : ” on
the lcd
  lcd.print ( DHT.temperature ) ;            // printing the temperature on the lcd
  lcd.print ( ( char ) 223 ) ;
  lcd.print ( " C " ) ;                     // Printing “ C “ on the display
  lcd.setCursor ( 0 , 1 );
  lcd.print ( " Humidity is : " ) ;         // printing “ humidity is : ” on the
display
  lcd.print ( DHT.humidity ) ;              // printing humidity on the display
  lcd.print ( " % " ) ;                     // printing “ % ” on display
  delay ( 1000 ) ;                          // Giving delay of 1 second.

}
```

Temperature Scales

Thermometers measure temperature according to well-defined scales of measurement. The three most common temperature scales are the Fahrenheit, Celsius, and Kelvin scales.

- **Celsius Scale & Fahrenheit Scale**

The Celsius scale has a freezing point of water as 0ºC and the boiling point of water as 100ºC. On the Fahrenheit scale, the freezing point of water is at 32ºF and the boiling point is at 212ºF. The temperature difference of one degree Celsius is greater than a temperature difference of one degree Fahrenheit. One degree on the Celsius scale is 1.8 times larger than one degree on the Fahrenheit scale 180/100=9/5.

- **Kelvin Scale**

Kelvin scale is the most commonly used temperature scale in science. It is an absolute temperature scale defined to have 0 K at the lowest possible temperature, called absolute zero. The freezing and boiling points of water on this scale are 273.15 K and 373.15 K, respectively. Unlike other temperature scales, the Kelvin scale is an absolute scale. It is extensively used in scientific work. The Kelvin temperature scale possesses a true zero with no negative temperatures. It is the lowest temperature theoretically achievable and is the temperature at which the particles in a perfect crystal would become motionless.

- **Relationship Between Different Temperature Scales**

The relationship between three temperature scales is given in the table below:

Relationship between different Temperature Scales

Conversion	Equation
Celsius to Fahrenheit	$T_{F^o} = \frac{9}{5} T_{c^o} + 32$
Fahrenheit to Celsius	$T_{C^o} = \frac{5}{9} T_{F^o} - 32$
Celsius to Kelvin	$T_K = T_{C^o} + 273.15$
Kelvin to Celsius	$T_{C^o} = T_K - 273.15$
Fahrenheit to Kelvin	$T_K = \frac{5}{9}(T(F^0) - 32) + 273.15$
Kelvin to Fahrenheit	$T_{F^0} = \frac{9}{5}(T(K) - 273.15) + 32$

Conclusion: -
