Experiment No. – 3

Title: Interfacing sensors and actuators with Arduino/Raspberry-pi.

Hardware Requirements: Arduino Board, LM 35 Temp. Sensor, IR Sensor, Ultrasonic Sensor

Software Requirements: Arduino IDE

Theory:

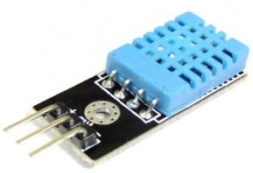
LM 35 Temperature Sensor:

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.

DHT11Sensor:

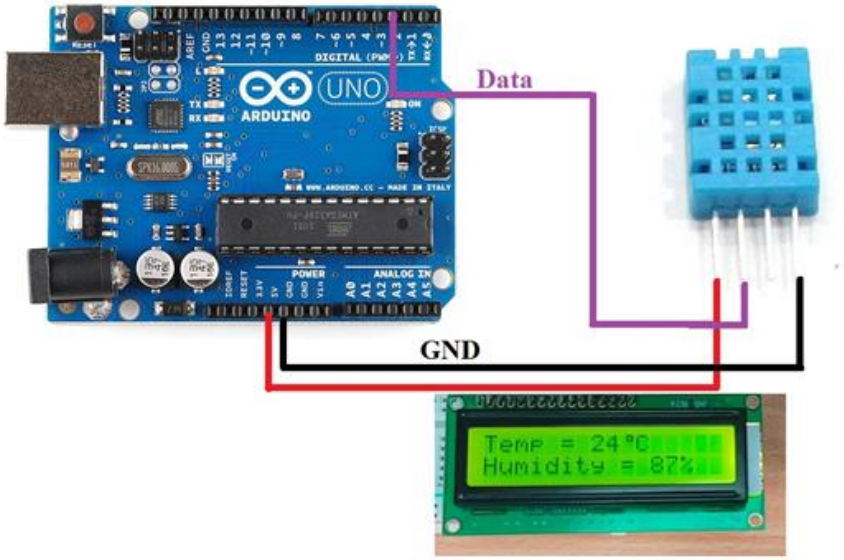


DHT11 is a Humidity and Temperature Sensor, which generates calibrated digital output. DHT11 can be interface with any microcontroller like Arduino, Raspberry Pi, etc. and get instantaneous results. DHT11 is a low cost humidity and temperature sensor which provides high reliability and long term stability.

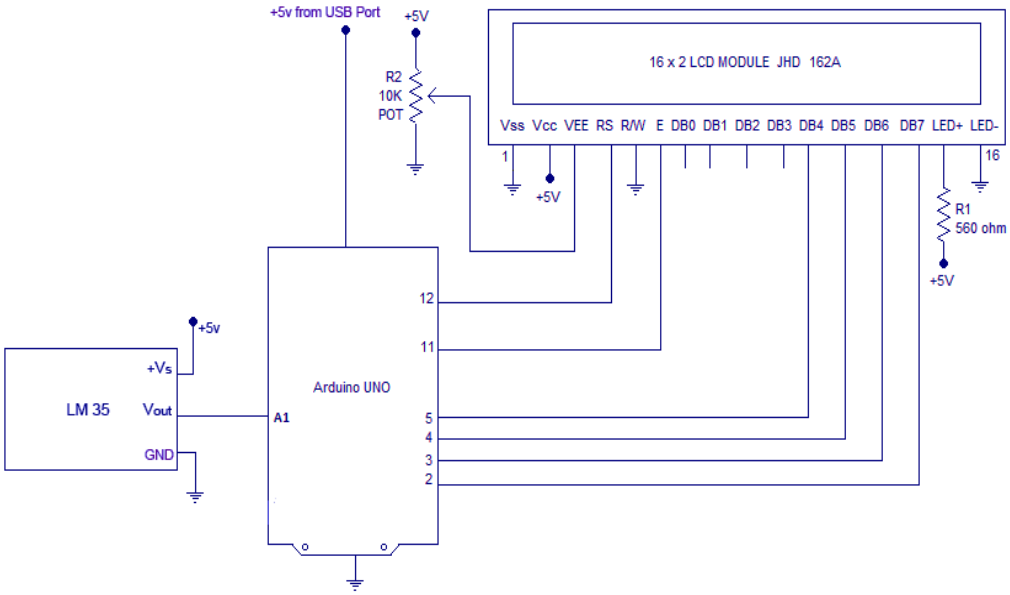
It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and outputs a digital signal on the data pin (no analog input pins needed). Its very simple to use, and libraries and sample codes are available for Arduino and Raspberry Pi.

This module makes is easy to connect the DHT11 sensor to an Arduino or microcontroller as includes the pull up resistor required to use the sensor. Only three connections are required to be made to use the sensor - Vcc, Gnd and Output.

It has high reliability and excellent long-term stability, thanks to the exclusive digital signal acquisition technique and temperature & humidity sensing technology.



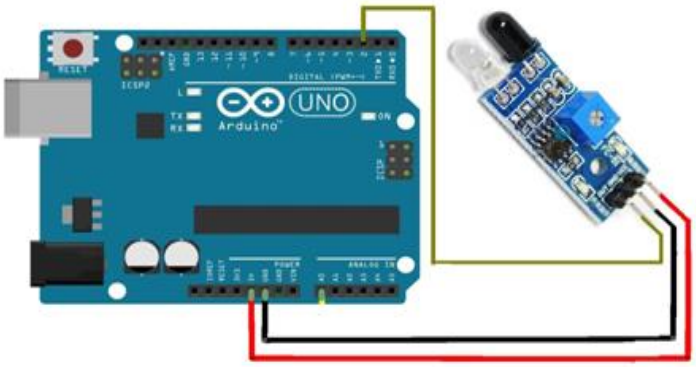
Interfacing DHT 11 to Arduino



Interfacing LM35 to Arduino

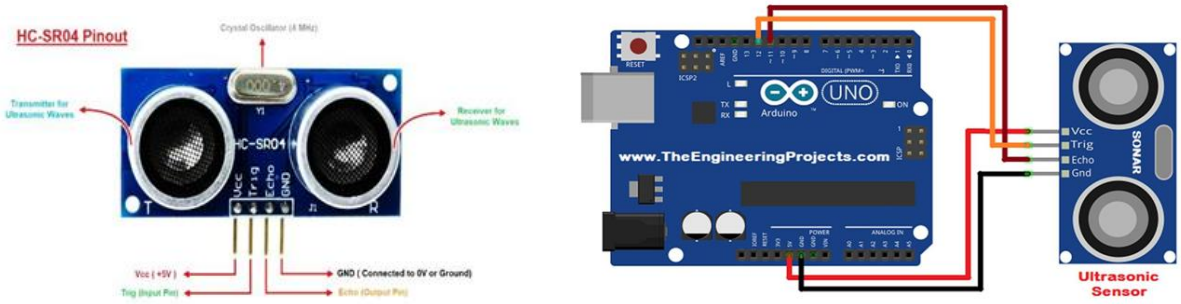
IR Sensor:

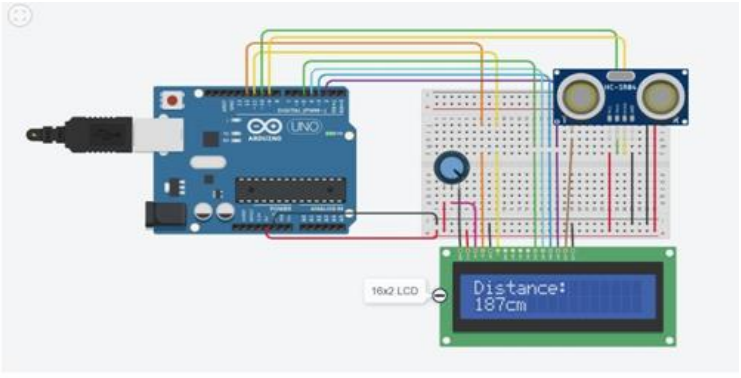
IR sensor is an electronic device that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations.



Interfacing of IR Sensor

Ultrasonic Sensor:





Interfacing of Ultrasonic Sensor

Procedure:

- Make the connection as per circuit diagram
- Open Arduino IDE
- Select the Arduino Board
- Select the Arduino Port
- Write code in Editor Window and save file
- Compile the code
- Upload the code
- Connect the DHT11 sensor to CN8 (pin A1) with the Blue sensor side facing the buttons on the board. Open the Serial Monitor from **Tools→Serial Monitor** and observe the analog values for the temperature and Humidity. You can blow on the sensor and bring some heated element near the sensor .

Conclusion:

Code:

Interfacing of DHT11

```
#include <dht.h>
dht DHT;
#define DHT11_PIN A1
void setup()
{
Serial.begin(9600);
Serial.println();
Serial.println("Type,\tHumidity (%),\tTemperature (C)");
}
void loop()
{
// READ DATA
Serial.print("DHT11, \t");
int chk = DHT.read11(DHT11_PIN);
// DISPLAY DATA
Serial.print(DHT.humidity, 1);
Serial.print(",\t");
Serial.println(DHT.temperature, 1);
delay(2000);
}
```

Interfacing of Temperature Sensor (LM35):

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(7,6,5,4,3,2);
void setup()
{
        Serial.begin(19200);
         lcd.begin(16,2);
}
void loop()
{
        int analogvalue = analogRead(A3);
         int pecent = map(analogvalue,0,1023,0,600);
         Serial.print("analogvalue= ");
         Serial.println(analogvalue);
         lcd.setCursor(0,0);
         lcd.print(analogvalue);
         Serial.print("Pecent= ");
         Serial.println(pecent);
```

```
            lcd.setCursor(0,1);
            lcd.print(pecent);
            delay(1000);
            lcd.clear ();
}
```

Interfacing of IR Sensor:

```
int ledPin=13;
int inputPin=2;
int val=0;

void setup()
{
            pinMode(13,OUTPUT);
            pinMode( inputPin, INPUT);
            Serial.begin(9600);
}
void loop()
{
          val=digitalRead(inputPin); // check the pin status (High=1/Low=0) //Active Low
          output
          if(val==HIGH)
          {
            Serial.print(“Object Absent\n");
            digitalWrite(13,LOW);
            }
          else
          {
            Serial.print("Object Present\n");
            digitalWrite(13,HIGH);
            }
  }
```

Interfacing of Ultrasonic Sensor:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (12,11,5,4,3,2);
```

```
// defining the pins
const int trigPin = 10;
const int echoPin = 9;
// defining variables
long duration;
int distance;
void setup()
 {
        pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
        pinMode(echoPin, INPUT); // Sets the echoPin as an Input
        Serial.begin(9600); // Starts the serial communication
        lcd.begin(16,2);
}
void loop()
{
        // Clears the trigPin
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        // Sets the trigPin on HIGH state for 10 micro seconds
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        // Reads the echoPin, returns the sound wave travel time in microseconds
        duration = pulseIn(echoPin, HIGH);
        // Calculating the distance
        distance= duration*0.034/2;
        lcd.setCursor (0,0);
        lcd.print("Distance: ");
        delay(1000);
          lcd.setCursor (0,1);
          lcd.print(distance);
          lcd.print("cm");
}
```