

## Practical : 10

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

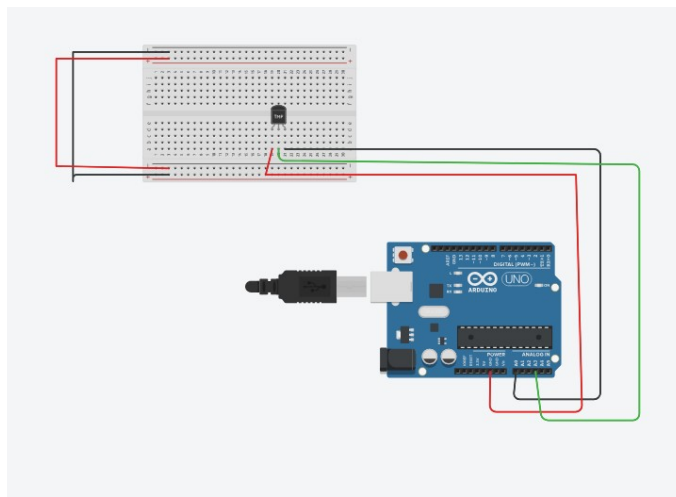
CODE:

```
10
11 // Include the DHT11 library for interfacing with the sensor.
12 #include <DHT11.h>
13
14 // Create an instance of the DHT11 class.
15 // - For Arduino: Connect the sensor to Digital I/O Pin 2.
16 // - For ESP32: Connect the sensor to pin GPIO2 or P2.
17 // - For ESP8266: Connect the sensor to GPIO2 or D4.
18 DHT11 dht11(2);
19
20 void setup() {
21     // Initialize serial communication to allow debugging and data readout.
22     // Using a baud rate of 9600 bps.
23     Serial.begin(9600);
24
25     // Uncomment the line below to set a custom delay between sensor readings (in milliseconds).
26     // dht11.setDelay(500); // Set this to the desired delay. Default is 500ms.
27 }
28
29 void loop() {
30     int temperature = 0;
31     int humidity = 0;
32
33     // Attempt to read the temperature and humidity values from the DHT11 sensor.
34     int result = dht11.readTemperatureHumidity(temperature, humidity);
35
36     // Check the results of the readings.
37     // If the reading is successful, print the temperature and humidity values.
38     // If there are errors, print the appropriate error messages.
39     if (result == 0) {
40         Serial.print("Temperature: ");
41         Serial.print(temperature);
42         Serial.print(" °C\tHumidity: ");
43         Serial.print(humidity);
44         Serial.println(" %");
45     } else {
46         // Print error message based on the error code.
47         Serial.println(DHT11::getErrorString(result));
48     }
49 }
50
```

O/P:

### 1. Set up your hardware

- Connect the temperature sensor (eg. LM35) to the Arduino board.
- Connect its VCC pin to 5V, GND pin to GND, and the output pin to an analog input pin (e.g. A0) on the Arduino
- Connect your Arduino board to your computer using a USB cable.



2. Open the Arduino IDE on your computer.
3. Create a new sketch by selecting "File" > "New".
4. Write the program code inside the Arduino IDE:

```
10
11 // Include the DHT11 library for interfacing with the sensor.
12 #include <DHT11.h>
13
14 // Create an instance of the DHT11 class.
15 // - For Arduino: Connect the sensor to Digital I/O Pin 2.
16 // - For ESP32: Connect the sensor to pin GPIO2 or P2.
17 // - For ESP8266: Connect the sensor to GPIO2 or D4.
18 DHT11 dht11(2);
19
20 void setup() {
21     // Initialize serial communication to allow debugging and data readout.
22     // Using a baud rate of 9600 bps.
23     Serial.begin(9600);
24
25     // Uncomment the line below to set a custom delay between sensor readings (in milliseconds).
26     // dht11.setDelay(500); // Set this to the desired delay. Default is 500ms.
27 }
28
29 void loop() {
30     int temperature = 0;
31     int humidity = 0;
32
33     // Attempt to read the temperature and humidity values from the DHT11 sensor.
34     int result = dht11.readTemperatureHumidity(temperature, humidity);
35
36     // Check the results of the readings.
37     // If the reading is successful, print the temperature and humidity values.
38     // If there are errors, print the appropriate error messages.
39     if (result == 0) {
40         Serial.print("Temperature: ");
41         Serial.print(temperature);
42         Serial.print(" °C\tHumidity: ");
43         Serial.print(humidity);
44         Serial.println(" %");
45     } else {
46         // Print error message based on the error code.
47         Serial.println(DHT11::getErrorString(result));
48     }
49 }
50
```

5. Verify and upload the code to your Arduino board by selecting "Sketch"> "Upload"
6. Open the serial monitor in the Arduino IDE by selecting "Tools" > "Serial Monitor" or by pressing Ctrl+Shift+M. Make sure the baud rate in the serial monitor matches the one set in the code (9600 baud).

7. The program will now read the temperature sensor values and send them to the serial monitor. You will see the temperature readings displayed in degrees Celsius

```
31 | int humidity = 0;
```

Output

Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM3')

Temperature: 32 °C	Humidity: 23 %
Temperature: 32 °C	Humidity: 24 %
Temperature: 32 °C	Humidity: 29 %
Temperature: 32 °C	Humidity: 27 %
Temperature: 32 °C	Humidity: 26 %
Temperature: 32 °C	Humidity: 28 %
Temperature: 32 °C	Humidity: 29 %
Temperature: 32 °C	Humidity: 29 %
Temperature: 32 °C	Humidity: 31 %
Temperature: 32 °C	Humidity: 31 %
Temperature: 32 °C	Humidity: 33 %
Temperature: 32 °C	Humidity: 34 %
Temperature: 32 °C	Humidity: 38 %
Temperature: 32 °C	Humidity: 41 %
Temperature: 32 °C	Humidity: 42 %