DATE OF CONDUCTION -: 15/2/24

DATE OF SUBMISSION -: 22/2/24

NAME -: Arjunsingh Gautam

PRN -: 22070123043

ENTC A2 2022-26

EXPERIMENT 5

Study and working of LM35 Temperature Sensor

AIM

To understand the functionality of a Temperature Sensor, design a signal conditioning circuit on a breadboard and implement on Arduino IDE Software.

HARDWARE USED

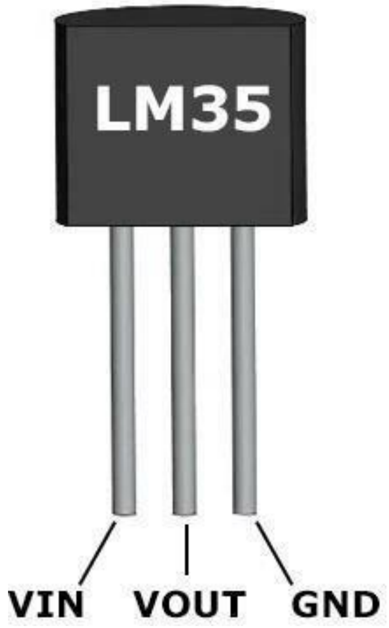
1. Arduino Board
2. Multiple Led
3. Breadboard
4. Jumper wires (male-to-female, male-to-male)
5. USB Cable
6. Computer with Arduino IDE
7. Arduino Code
8. Power Source (optional, e.g., battery pack)
9. LM35 Temperature Sensors

SOFTWARE USED

1. Arduino IDE for programming and microcontroller
2. Serial monitor for data visualization software for data analysis.

SYESETM INFORMATION

The TMP36 is a low-cost analog temperature sensor that provides a voltage output proportional to temperature. It has a linear output scale, making it easy to interface with Arduino's analog input pins. It measures temperatures from -40°C to +125°C with an accuracy of ±2°C.



APPLICATIONS

It is commonly used in various projects requiring temperature monitoring or control. Its simplicity, accuracy, and low cost make it popular for hobbyists and professionals alike. With its analog output, interfacing with microcontrollers or analog-to-digital converters is straightforward, allowing temperature data to be easily integrated into electronic systems for temperature sensing and control applications.

FUNCTIONS USED IN CODE

- int lm35_val: This declares an integer variable named "lm35_val" which will be used to store the raw analog value read from the LM35 temperature sensor.

- float real_val: This declares a floating-point variable named "real_val" which will be used to store the converted temperature value in Celsius.

- lm35_val = analogRead(sensor): This line reads the analog voltage value from the LM35 temperature sensor connected to the analog pin defined earlier (A0) and stores it in the variable "lm35_val".

- Serial.println(lm35_val): This line prints the raw analog value read from the LM35 temperature sensor to the serial monitor. This value represents the voltage level proportional to the temperature.

- real_val = (lm35_val * 4.88): This line converts the raw analog value read from the LM35 sensor into millivolts. The conversion factor 4.88 is used assuming a 5V reference voltage for the Arduino's analog-to-digital converter.

- real_val = (real_val / 10): This line further adjusts the converted value to obtain the temperature in Celsius, assuming that every 10 millivolts represent 1 degree Celsius.

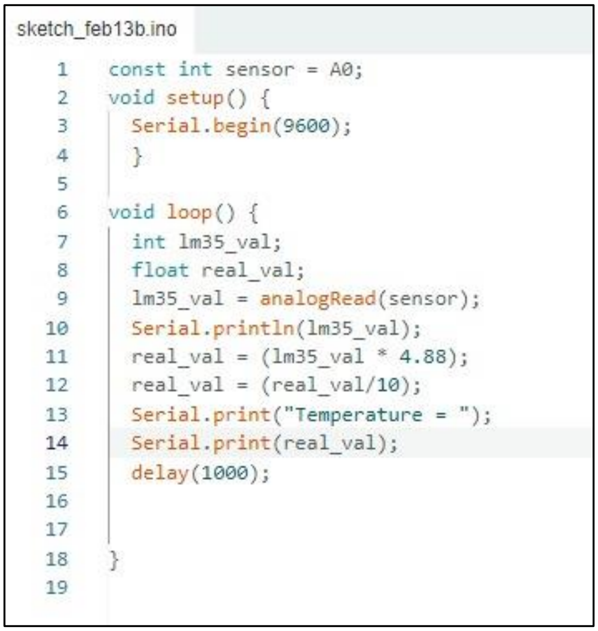
- Serial.print("Temperature = "): This line prints the text "Temperature = " to the serial monitor.

- Serial.print(real_val): This line prints the converted temperature value in Celsius to the serial monitor.

CODE: TYPED CODE

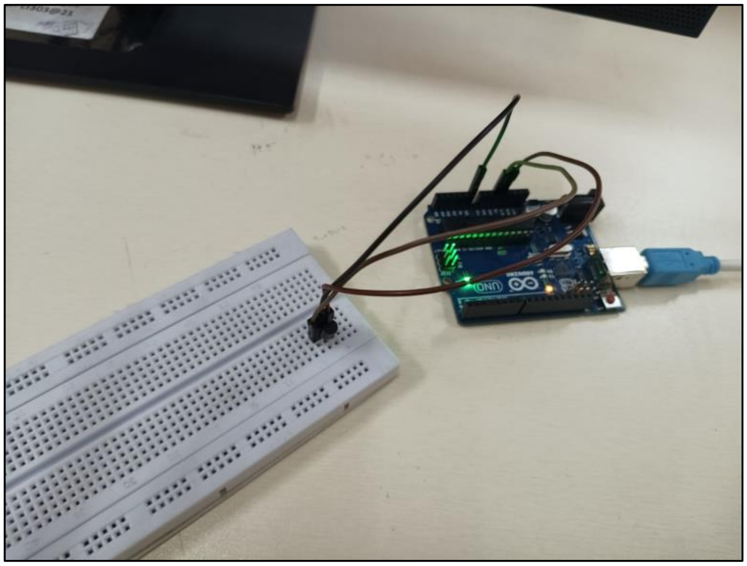
```
//Arjunsingh Gautam
//22070123043
Const int sensor =  A0;
Void setup(){
Serial.begin(9600);
}
Void loop (){
Int lm35_val;
Float real_val;
Lm35_val = analogRead(sensor);
Serial.println(lm35_val);
Real_val = (lm35_val * 4.88);
Real_val = (real_val/10);
Serial.print(“Temperature = “);
Serial.print(real_val);
Delay(1000);
}
```

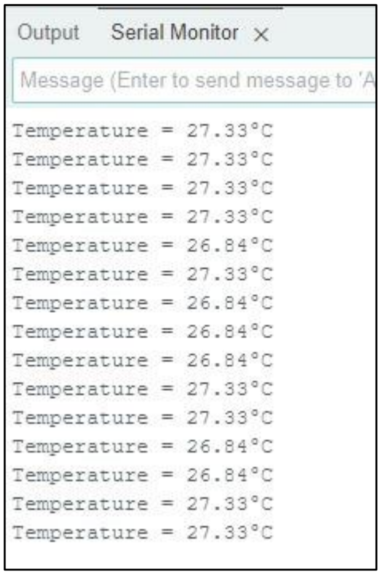
CODE: SCREENSHOOT



```
const int sensor = A0;
void setup() {
  Serial.begin(9600);
  }

void loop() {
  int lm35_val;
  float real_val;
  lm35_val = analogRead(sensor);
  Serial.println(lm35_val);
  real_val = (lm35_val * 4.88);
  real_val = (real_val/10);
  Serial.print("Temperature = ");
  Serial.print(real_val);
  delay(1000);


}

```

PICTURES OF THE EXPERIMENT





```
Temperature = 27.33°C
Temperature = 27.33°C
Temperature = 27.33°C
Temperature = 27.33°C
Temperature = 26.84°C
Temperature = 27.33°C
Temperature = 26.84°C
Temperature = 26.84°C
Temperature = 26.84°C
Temperature = 27.33°C
Temperature = 27.33°C
Temperature = 26.84°C
Temperature = 26.84°C
Temperature = 27.33°C
Temperature = 27.33°C
```

CONCLUSION

In this experiment, we used an LM35 Analog Temperature sensor along with an Arduino board. Initially, we connected the LM35 sensor to the Arduino board using jumper wires, ensuring correct polarity. Then, we transferred the provided code to the Arduino IDE and uploaded it to the board. This code sets up serial communication, reads the analog voltage from the LM35 sensor, converts it to Celsius, and then prints the temperature value onto the serial monitor. Once the code was uploaded, we monitored the temperature readings through the serial monitor. Since the program operates in a loop, it continually fetches the temperature from the sensor and showcases it in real-time. This experiment effectively showcases the LM35 sensor's capability in accurately measuring temperature and integrating it with an Arduino board for processing and display purposes.