Internet of Things Laboratory

Assignment 8

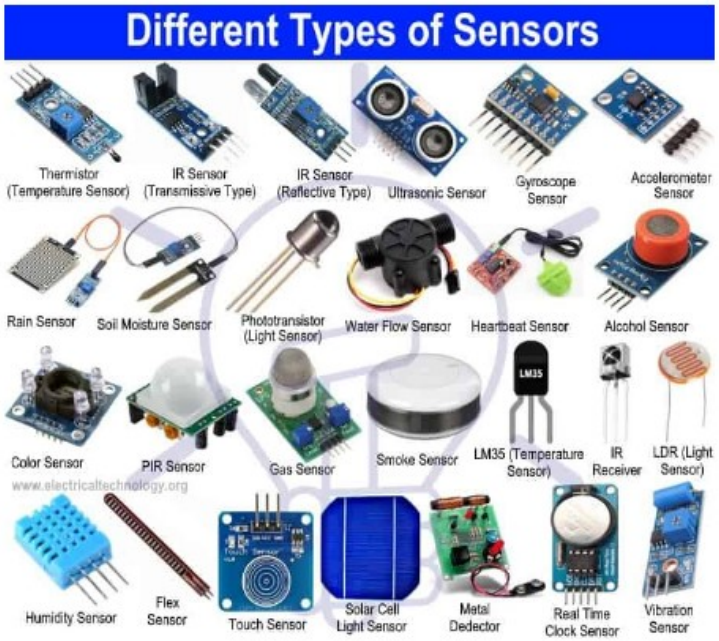
Prajakta Deokule

C22019221332

4329

Problem Statement: Write an application on Raspberry Pi to read the environment temperature. If temperature crosses a threshold value, the application indicates the user using LEDSs.

1. **What are different types of sensors?** :
   1. Position Sensors
   2. Pressure Sensors
   3. Temperature Sensors
   4. Force Sensors
   5. Vibration Sensors
   6. Piezo Sensors
   7. Fluid Property Sensors
   8. Humidity Sensors
   9. Strain gauges



Types of Sensors

2. Which sensor is used for temperature?

Thermocouples, RTDs, thermistors, and semiconductor based ICs are the main types of temperature sensors.

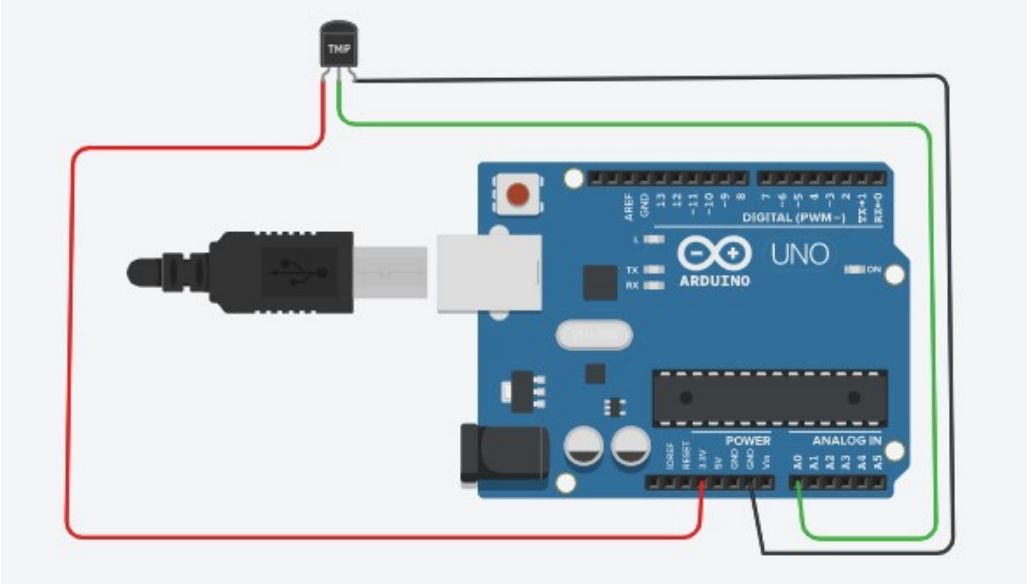
3. How many pins do temperature sensors have?

The temperature sensor LM35 has 3 legs, the first leg is Vcc, you can connect this to the 3.3V. The middle leg is Vout, where the temperature is read from. The third leg is ground.

4. Write a python program to read temperature using arduino.

```
int sensorPin=0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
//getting the voltage reading from the temperature sensor
  int reading=analogRead(sensorPin);
 // converting that reading to voltage, for 3.3v arduino use 3.3
  float voltage=reading*3.3;
```

```
 voltage/=1024.0;
//converting from 10 mv per degree wit 500 mV offset //to degrees ((voltage
- 500mV) times 100)
 float temperature=(voltage-0.5)*100;
 Serial.print(temperature);
 Serial.println("degree C");
 delay(1000);
//waiting a second
```




```
}
```

Output:

19.71 degree C

19.71 degree C

34.45 degree C

45.26 degree C

64.10 degree C

64.10 degree C

79.42 degree C

83.34 degree C

96.89 degree C

96.89 degree C

5. Enlist some applications implemented with Temperature sensor.

- **Motors**– there are many different aspects of motors and most of these require temperature measurement to ensure the motor itself does not overheat.
- **Surface plates** – ring terminal temperature sensors are often used on surface plates as they can be mounted onto a flat surface and measure temperature effectively.
- **Home appliances** – kettles, toasters, washing machines, dishwashers and coffee machines will all contain temperature sensors.
- **Computers**– within computers there are temperature sensors to ensure the system does not overheat
- **Industrial equipment** – temperature sensors used within these applications will need to be robust as the environment can be very demanding.
- **Warming Electrical Radiators** – NTC thermistors are used to control the heat on electric radiators.
- **Exhaust Gas Monitoring on Motorsport Vehicles** – Motorsport temperature sensors need to be highly reliable and durable to ensure performance is not compromised in this harsh environment.
- **Food Production**-3D printed chocolates – temperature sensors are used to monitor the temperature of the melted chocolate for 3D printing.
- **Alcohol breathalyser** – thermistors are used within alcohol breathalysers to measure the temperature of the subject's breath.

Code

```
import os
import glob
import time
import requests

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

led = 21
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(led, GPIO.OUT)

def read_temp_raw():
        f = open(device_file, 'r')
        lines = f.readlines()
        f.close()
        return lines

def read_temp():
        lines = read_temp_raw()
        while lines[0].strip()[-3:] != 'YES':
                time.sleep(0.2)
                lines = read_temp_raw()
        equals_pos = lines[1].find('t=')
        if equals_pos != -1:
                temp string = lines[1][equal pos+2]
                temp_c = float(temp_string) / 1000.0
        # temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c

try:
        while True:
                temp = read_temp()
        if(temp > 32.0):
                GPIO.output(led, GPIO.HIGH)
                time.sleep(5)
                GPIO.output(led, GPIO.LOW)
        print(temp)
        time.sleep(2)

except:
        print("Error occurred.")
```