Problem :

Design a system to measure human body temperature without any touch to surface of the body. Provide wired and wireless solution for the same. Output data will be given to a PC which further will process it accordingly.

Also look for proper design solution to implement that in real life use case. System may contain any microcontroller, 5V power supply, sensors if needed.

For measuring the temperature without touching surface of body we can use:- Non Contact Temperature Sensors: These temperature sensors use convection & radiation to monitor temperature.

The sensor used here is TMP006/B Infrared Thermopile Sensor (manufacturer Texas instruments).

Brief description of TMP006/B Infrared Thermopile Sensor:-

Thermopile Principles and Operation

The TMP006 and TMP006B sense radiation by absorbing the radiation on a hot junction. The thermopile then generates a voltage proportional to the temperature difference between the hot junction, Thot, and the cold junction, Tcold.

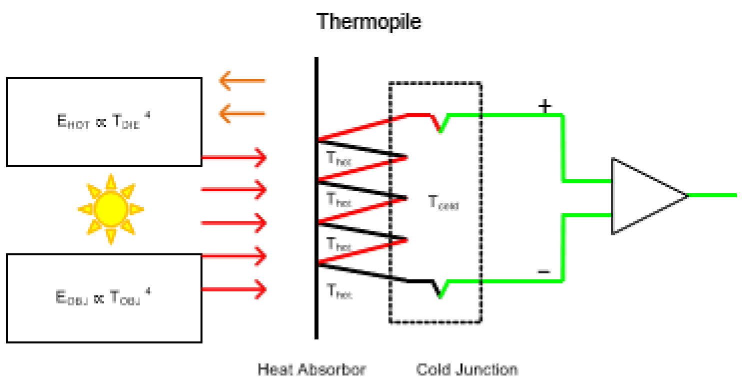


Figure 9. Principle of Thermopile Operation

The cold junction is thermally grounded to the die, and is effectively TDIE, the die temperature. In thermal equilibrium, the hot junction is determined by the object temperature, TOBJ. The energy emitted by the object, EOBJ, minus the energy radiated by the die, EDIE, determines the temperature of the hot junction. The output voltage, VOUT, is therefore determined by the relationship shown in Equation 2: $V_{OUT} = V_{SENSOR} = C \times (T_{HOT} - T_{COLD}) \propto (T_{OBJ}4 - T_{DIE}4)$

where • C is a constant depending on the design of the sensing element.

Object Temperature Calculation

The TMP006 and TMP006B generate a sensor voltage, VSensor, in register 00h that is representative of the energy radiated by the object. In an ideal situation, the Stefan-Boltzman law relates the energy radiated by an object to its temperature by the relationship shown in Equation 5:

$$\text{Energy}_{Rad} = \varepsilon\sigma T_{OBJ}^4$$

where

- σ = Stefan-Boltzman constant = 5.7×10^{-12} W/cm^2/K^4
- ε = Emissivity, $0 < \varepsilon < 1$, an object dependent factor, $\varepsilon = 1$ for a perfect black body

where • σ = Stefan-Boltzman constant = 5.7 × 10-12 W/cm2/K4 • ε = Emissivity, 0 < ε < 1, an object dependent factor, ε = 1 for a perfect black body (5) A similar relationship holds for the sensing element itself that radiates heat at a rate determined by TDIE. The net energy absorbed by the sensor is then given by the energy absorbed from the object minus the energy radiated by the sensor, as shown in Equation 6:

$$V_{SENSOR} \propto E_{ABSORBED} - E_{RADIATED} = \varepsilon\sigma\left(T_{OBJ}^4 - T_{DIE}^4\right) \tag{6}$$

In an ideal situation, the sensor voltage relates to object temperature as shown in Equation 7:

$$T_{OBJ} = \sqrt[4]{T_{DIE}^4 + \frac{V_{SENSOR}}{\varepsilon\sigma}} \tag{7}$$

$$T_{OBJ} = \sqrt[4]{T_{DIE}^4 + \left(\frac{f\{V_{OBJ}\}}{S}\right)}$$

where

- S is a system-dependent parameter incorporating the object emissivity (ε), FOV, and sensor characteristics. The parameters S0, A1, and A2 are used in determining S.
- $f(V_{OBJ})$ is a function that compensates for heat flow other than radiation, such as convection and conduction, from nearby objects. The parameters B0, B1, and B2 are used to tune this function to a particular system and environment. (8)

Field of View and Angular Response

The TMP006 and TMP006B sense all radiation within a defined field of view (FOV). The FOV (or full-angle of θ) is defined as 2Φ. These devices contain no optical elements, and thus sense all radiation within the hemisphere to the front of the device. Figure shows the angular dependence of the sensor response and the relative power for a circular object that subtends a half angle of phi (Φ). Figure 7 defines the angle Φ in terms of object diameter and distance. Figure 7 assumes that the object is well approximated as a plane that is perpendicular to the sensor axis.

Figure 7. FOV Geometry Definition

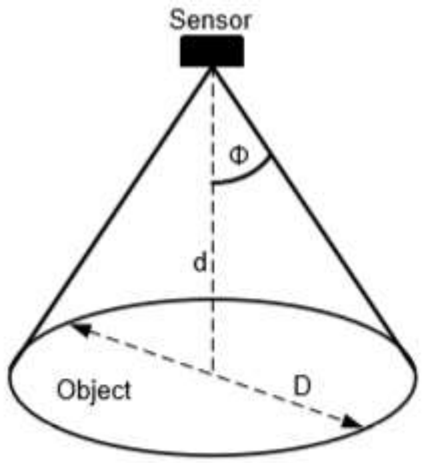


Figure 7. FOV Geometry Definition

In this case, the maximum contribution is from the portion of the object directly in front of the TMP006 or TMP006B (Φ = 0); with the sensitivity per solid angle, dR/dΦ decreases as Φ increases. Approximately 50% of the energy sensed by the TMP006 and TMP006B is within a FOV (θ) = 90°.

Now to communicate with the TMP006 sensor we are using an ARDUINO as we need to use I^2C communication protocol.

Connections:

- VCC → 5V (or 3.3V)
- GND → GND
- SCL → A5 (or SCL for R3)
- SDA → A4 (or SDA for R3)

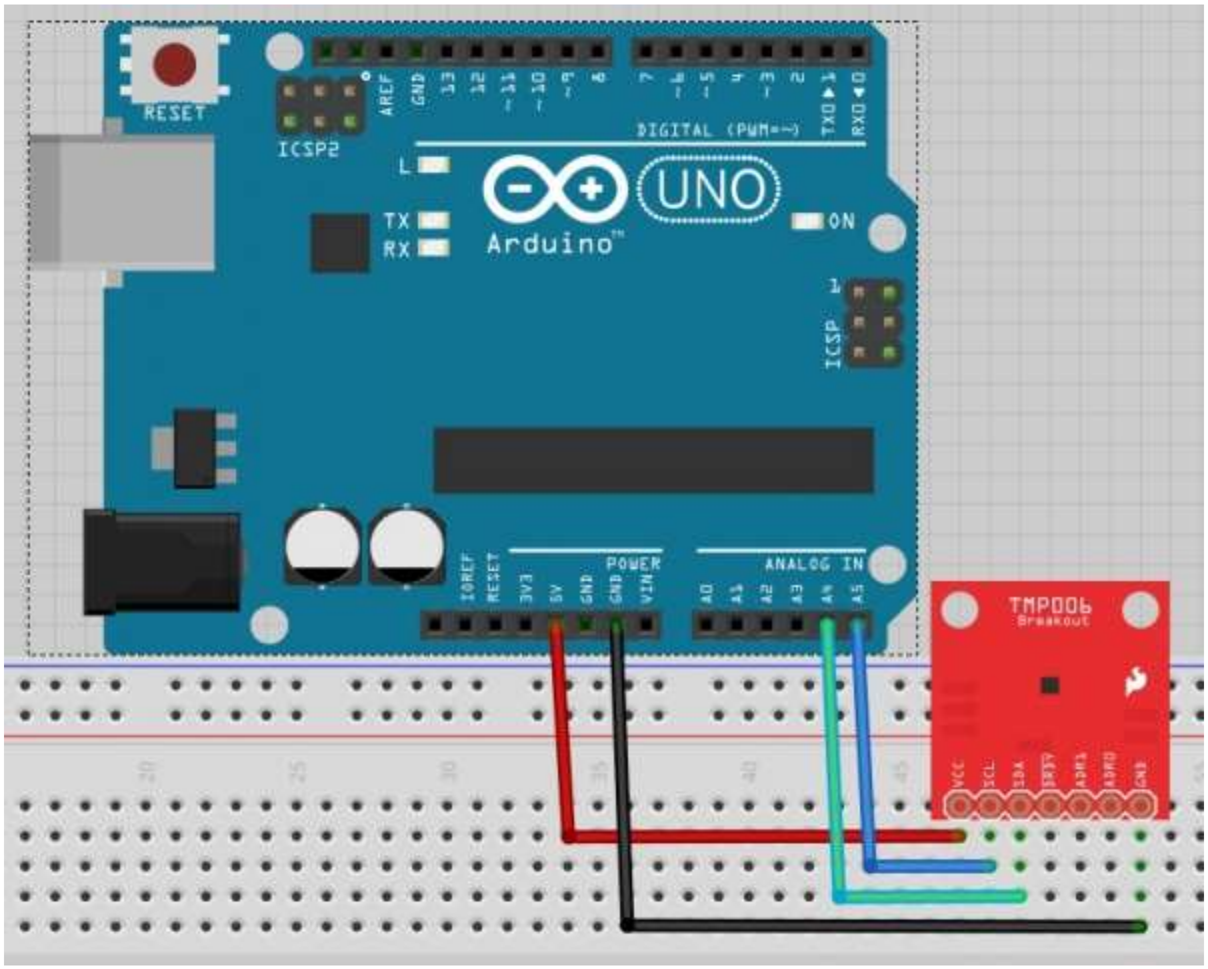


Diagram drawn in fritzing

The arduino is then connected to our PC or Laptop via usb and we can observe the temperature readings displayed on serial monitor of arduino IDE.

Therefore this is the wired solution of measuring temperature.

Arduino Code :

In the beginning, we have two global variables. One stores the I^{2}C address of the sensor, and the other stores how many times we'd like the sensor to sample per temperature reading/calculation. Feel free to try the defaults right away with the hardware setup described in the last section, no changes necessary. Here they are:

```
uint8_t sensor1 = 0x40; // I2C address of TMP006, can be 0x40-0x47

uint16_t samples = TMP006_CFG_8SAMPLE; // # of samples per reading, can be 1/2/4/8/16
```

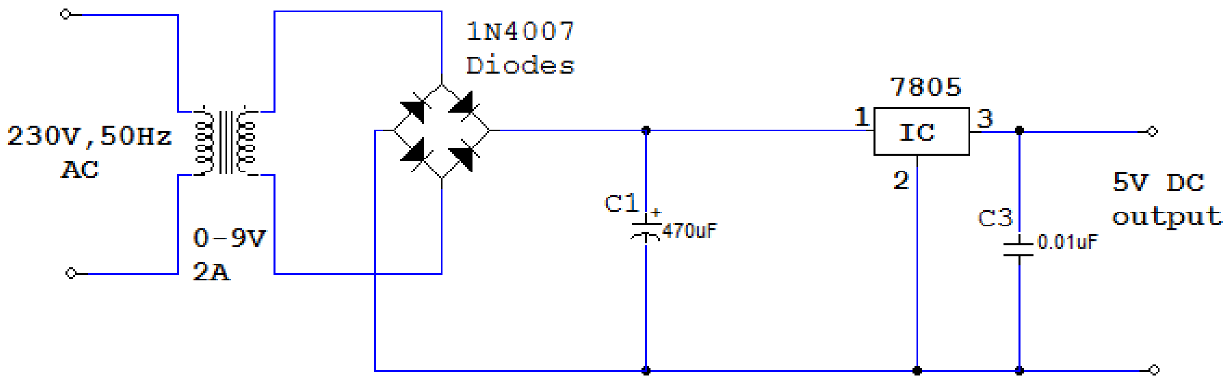
In the setup loop. Here we initialize serial output so we can display our readings. We also call a configuration function for our TMP006 sensor. It sets up some defaults for us to get going and also tells the sensor how many samples per reading we want. If you're using more than one sensor, you'll have to call this function for each one with the appropriate I2C address.

```
void setup()

{

  Serial.begin(9600);

  Serial.println("TMP006 temp measurement");



  config_TMP006(sensor1, samples);

}
```

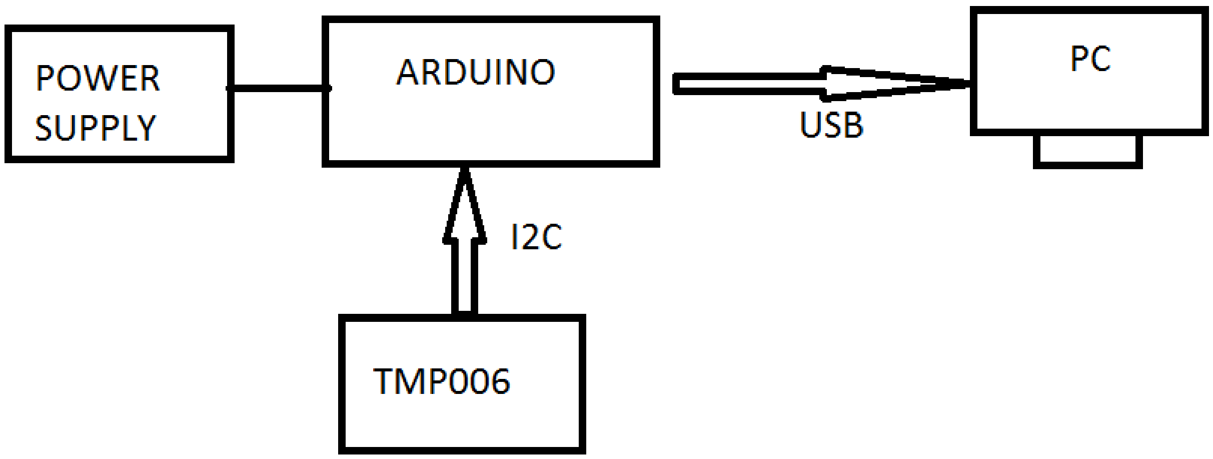
Within the loop function, we call two main functions. The first gives us the temperature of the object in front of the sensor, and the second gives us the temperature of the sensor itself. Both are then sent via serial to your computer and can be viewed using the Serial Monitor. Again, you’ll need to add duplicates of these functions if you’re talking to multiple temperature sensors. Power supply to be used :

```
void loop()

{

  float object_temp = readObjTempC(sensor1);

  Serial.print("body Temperature: ");

  Serial.print(object_temp); Serial.println("*C");



  float sensor_temp = readDieTempC(sensor1);

  Serial.print("Sensor Temperature: ");

  Serial.print(sensor_temp); Serial.println("*C");



  delay(2000); // delay 1 second for every 4 samples per reading

}
```

Arduino requires 5v power supply:



FINAL DESIGN:



For wireless transfer of the output data

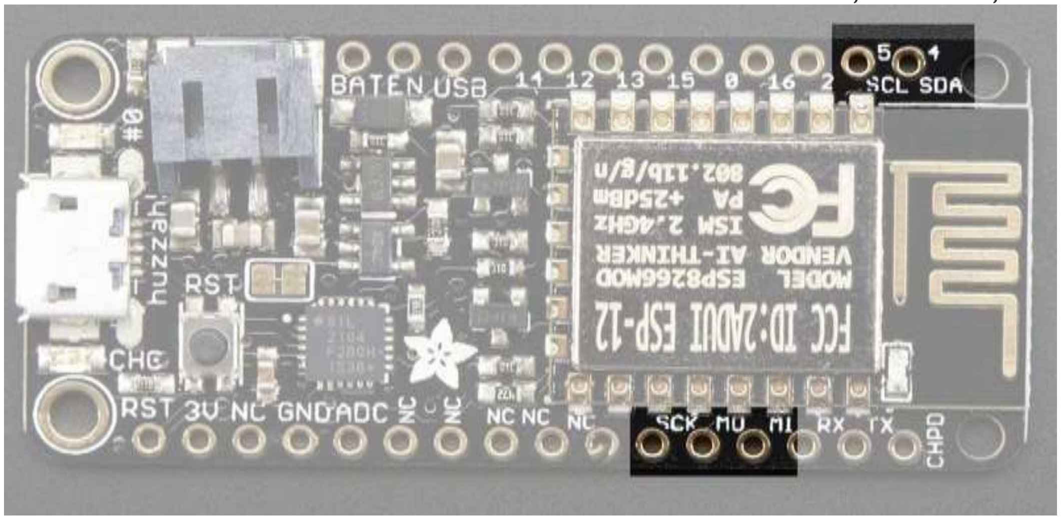
If we want to use the same sensor TMP006 then we need to look for a module that can perform I2C communication as well as has wifi connectivity, or zigbee, Bluetooth compatible.

There are multiple methods available example:-

- We can transmit the I2C data of the sensor using RF transmitter and then receive it on the other end and use an arduino on receivers side and this arduino is connected to PC to display it.
- We can use arduino on the transmitter side and use ESP8266 to transmit data online which can be accessed online through PC. The ESP8266 can be used to form a server that can be accessed online.

Using I2C & SPI pins of ESP8266

We can use the ESP8266 to control I2C and SPI devices, sensors, outputs, etc.



We can use any pins for I2C and SPI but to make it easier for people using existing Arduino code, libraries, sketches we set up the following:

- **I2C SDA** = **GPIO #4** (default)

- **I2C SCL** = **GPIO #5** (default)

If you want, you can connect to I2C devices using other 2 pins in the Arduino IDE, by calling `Wire.pins(sda, scl)` before any other Wire code is called (so, do this at the begining of `setup()`