



NON-CONTACT THERMOMETER

USING MLX90614



Agneya Kumar Singh 19BEC0375
Vineeth Krishna M 19BEC0418
GROUP - 26

Abstract:

Nowadays due to the pandemic there is an increased demand of the non-contact thermometer, this is our attempt at making one. The thermo gun which is generally used is expensive, requires human intervention and cannot be monitored remotely. Moreover, it does not store the data for future reference. This project is to reduce the overall cost of the device and make this process as much as free of human interaction. In this project we will be using the IR sensor MLX90614 which has a high resolution of 0.02°C and can accurately measure temperature in the range -20°C to 120°C (approximately -29°F to 49°F). We will be making a webpage to display the output remotely.

Table of Contents:

1. Introduction
2. Experiment and Result
3. Conclusion
4. References
5. Appendices

Introduction:

Our aim is to provide an effective way to measure temperature of an individual without involving any human interference. Measuring of temperature is becoming necessary day by day due to the removal of lockdown by the government. So as make the human interaction minimum we have made this project as a DIY method to measure the temperature remotely from a safe place. For this we are using the IR sensor MLX90614. This sensor has a high resolution of 0.02°C , also it has a range of -20°C to 120°C .

Components used in the project:

1. NODE MCU
2. MLX90614 IR Sensor
3. Jumper Wires
4. Bread-Board
5. Website for monitoring the temperature.

The MLX90614(from here on mentioned as IR sensor) comes with an on-board signal conditioning circuit. The IR sensor has four leads, Ground, V_{in} (3.3V), SCL and SDA. SCL and SDA are part of serial communication protocol called I2C.

Why we chose NodeMCU over Arduino?

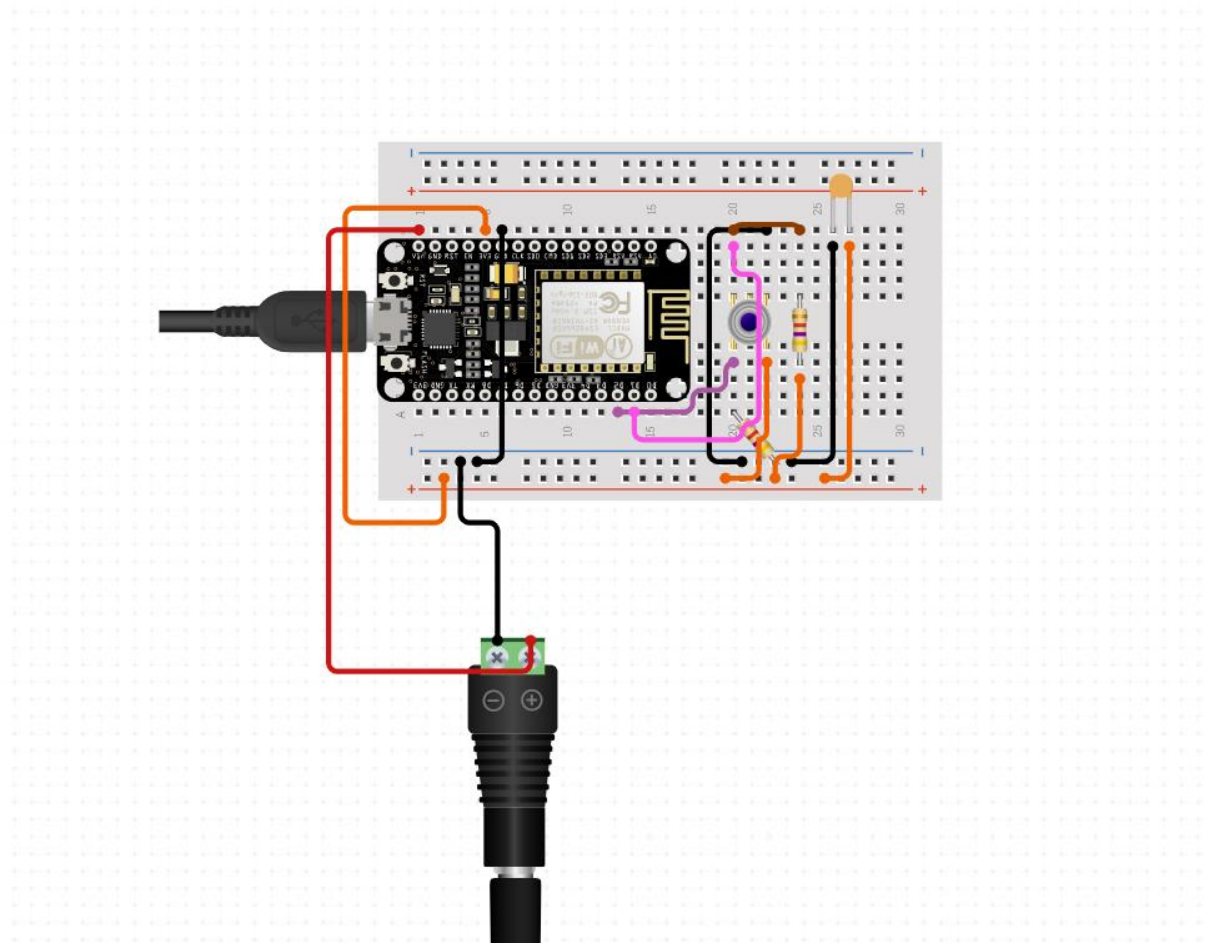
1. NodeMCU is cheaper than Arduino.
2. It comes with Wi-Fi module built onto the board. This helps in making the thermometer more compact and thus reduces the overall cost of the project.
3. It is capable of starting web servers.
4. It also supports I2C protocol so it can communicate with the sensor.
5. NodeMCU has a better CPU than Arduino.

Working Principle of MLX90614:

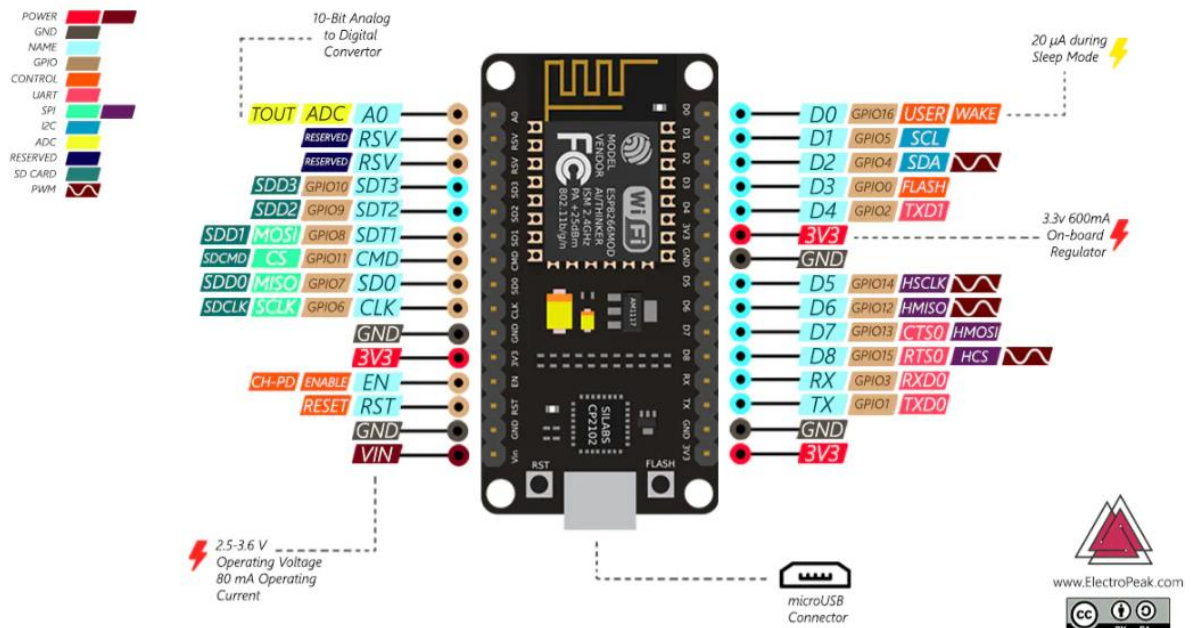
The IR sensor works on a law called as Stefan-Boltzmann Law. The law states that all objects and living beings emit Infrared Energy and the intensity of this emitted energy will be directly proportional to the temperature of that object. So, the sensor calculates the temperature of an object by measuring the amount of IR energy emitted. The sensor is factory calibrated and hence it acts like a plug and play sensor module to speed up development process. The sensor consists of two devices embedded as a single sensor, one device acts as a sensing unit and other device acts as a processing unit. The sensing unit is an Infrared Thermophile Detector called MLX81101 which senses the temperature and the processing unit is a Single Conditioning ASSP called MLX90302 which converts signal from the sensor to digital value and communicates using the I2C protocol. The MLX90302 has a low noise amplifier, 17-bit ADC and a powerful DSP which helps the sensor to have high accuracy and resolution.

Experiment and Results:

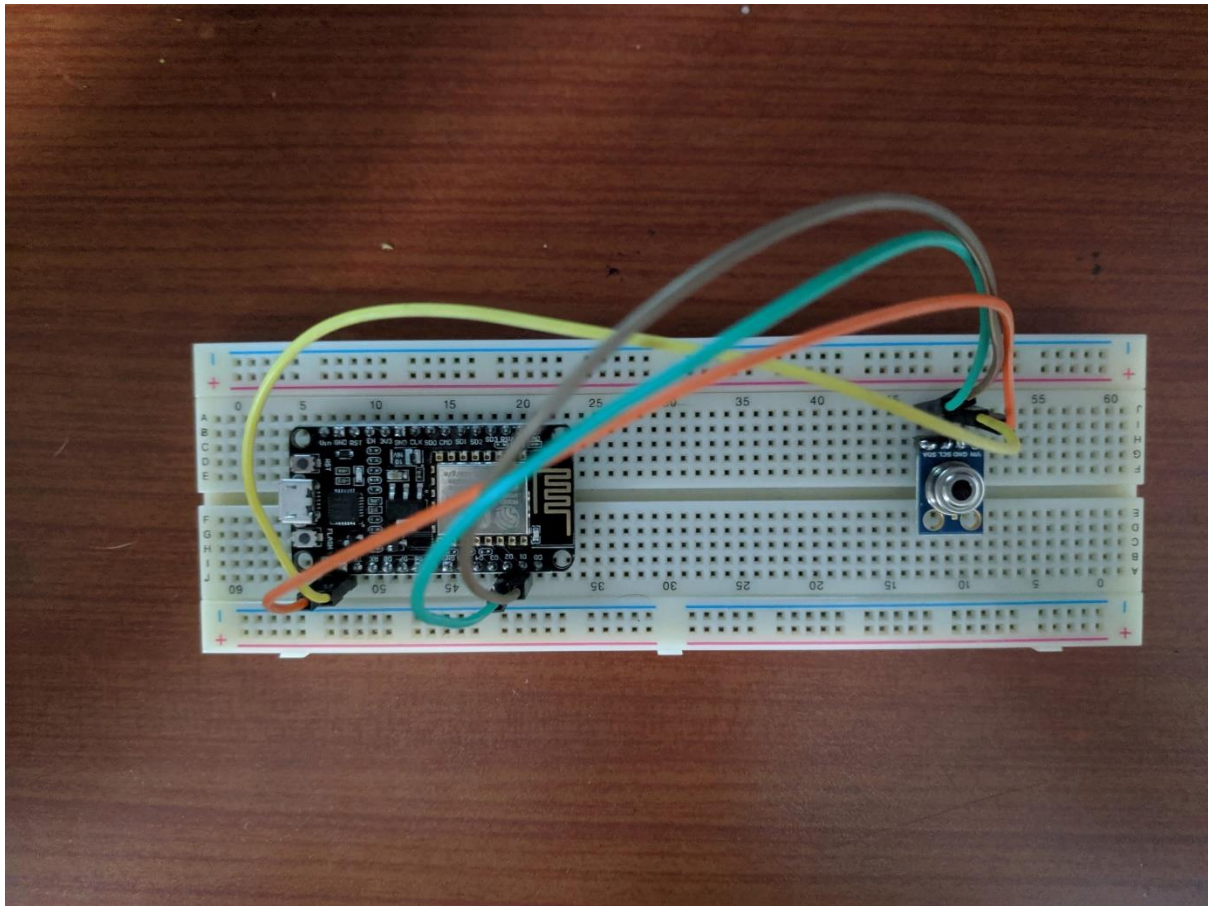
Circuit Diagram:



NodeMCU:



Hardware Circuit:



Code:

```
#include<Wire.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <Adafruit_MLX90614.h>
const char 'ssid = "*****";
const char 'pass = "*****";
ESP8266WebServer server(80);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
float Ambient;
float Object;
void setup() {
  Serial.begin(9600);
  delay(10);
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("Got IP: "); Serial.println(WiFi.localIP());

  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);
```

```

server.begin();
Serial.println("HTTP server started");
mlx.begin();
}

void loop() {
  Serial.print("Ambient = ");
  Serial.print(mlx.readAmbientTempC());
  Serial.print("°C\tObject = ");
  Serial.print(mlx.readObjectTempC());
  Serial.println("°C");
  Serial.print("Ambient = ");
  Serial.print(mlx.readAmbientTempF());
  Serial.print("°F\tObject = ");
  Serial.print(mlx.readObjectTempF());
  Serial.println("°F");

  Serial.println();
  server.handleClient();
  delay(1000);
}

void handle_OnConnect() {
  Ambient = mlx.readAmbientTempF(); // Gets the values of the temperature
  Object = mlx.readObjectTempF(); // Gets the values of the humidity
  server.send(200, "text/html", SendHTML(Ambient,Object));
}

```

Output:

```

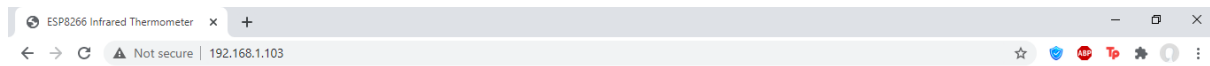
.....
WiFi connected
Got IP: 192.168.1.103
HTTP server started
Ambient = 23.87°C      Object = 22.13°C
Ambient = 74.97°F      Object = 71.83°F

Ambient = 23.87°C      Object = 22.17°C
Ambient = 74.97°F      Object = 71.91°F

Ambient = 23.85°C      Object = 22.19°C
Ambient = 74.93°F      Object = 71.94°F

Ambient = 23.85°C      Object = 22.31°C
Ambient = 74.93°F      Object = 72.16°F

```

ESP8266 NodeMCU Infrared Thermometer

Ambient Temperature (in F): 76 F

Object Temperature (in F): 72 F

The video link has been given below. It shows the working of the circuit:

<https://tinyurl.com/y2lelelb>

Conclusion:

We can observe that the sensor provides the data for both ambient temperature and the temperature of the body. The results can be seen on the webpage. This module can be put inside a protection case and thus protecting it from external damage and making the device more durable. An LCD display can also be attached to the circuit which can display the temperature to the person whose temperature is being measured but, in our project we have not included the LCD display because it was not available to us due to delivery restrictions.

References:

1. <https://components101.com/sensors/melexis-mlx90614-contact-less-ir-temperature-sensor> for details of the sensor and the working principle
2. The code was written and complied by the team members

Appendices:

Pin No.	Pin Name	Description
1	Vdd (Power supply)	Vdd can be used to power the sensor, typically using 5V
2	Ground	The metal can also act as ground
3	SDA – Serial Data	Serial data pin used for I2C Communication
4	SCL – Serial Clock	Serial Clock Pin used for I2C Communication