**Infrared Thermometer**

When debugging an electronics circuit or testing a new hardware design, oftentimes I tend to check if the components on the board are getting hot abnormally by touching them. And if something is messed up (which usually is in the first try) these components could get as hot as 80°C or more burning not only the component but also my finger along with it. After burning my fingers for more times than I could remember I decided to build my own Temperature Gun using an Arduino and an Infrared Temperature Sensor. This Thermal gun will be built using a non-contact temperature sensor called MLX90614; hence it can not only be used to measure component temperatures but can also be used for measuring body temperature, surface temperature, Heat ventilation and much more. Of course, these thermal guns are readily available in the market from renowned manufacturers like Fluke, Flir etc. But they are not light on your pockets and on top of that what is more fun than building your own gadgets.

**Material Required:**

• Arduino Pro Mini (189/-)

• MLX90614 Infrared Temperature Sensor

[Amazon:1269/- // Alibaba:(7-784/-)]

• OLED Display – SSD1306 (250/-)

• Laser Diode (70/-)

• 9V Battery (29/-)

• Push button (3/-)

• Battery Clip (20/-)

• Connecting wires (7/-)

**MLX90614 Infrared Thermometer:**

Before we proceed with the tutorial it is important to know how the MLX90614 sensor works. There are many temperature sensors available in the market and we have been using the DHT11 Sensor and LM35 extensively for many applications where atmospheric humidity or temperature has to be measured.

But here, for a thermal gun we need a sensor that could sense the temperature of a particular object (not ambient) without directly getting in contact with the object. For this purpose we have contact less temperature sensors which utilizes Laser or IR to calculate the temperature of an object. The MLX90614 is one such sensor that uses IR energy to detect the temperature of an object.

MLX90614 sensor is manufactured by Melexis Microelectronics Integrated system, it has two devices embedded in it, one is the infrared thermopile detector (sensing unit) and the other is a signal conditioning DSP device (computational unit). It works based on Stefan-Boltzmann law which states that all objects emit IR energy and the intensity of this energy will be directly proportional to the temperature of that object. The sensing unit in the sensor measures how much IR energy is emitted by a targeted object and the computational unit converts it into temperature value using a 17-bit in-built ADC and outputs the data through I2C communication protocol. The sensor measures both the object temperature and ambient temperature to calibrate the object temperature value.

**MLX90614 Features:**

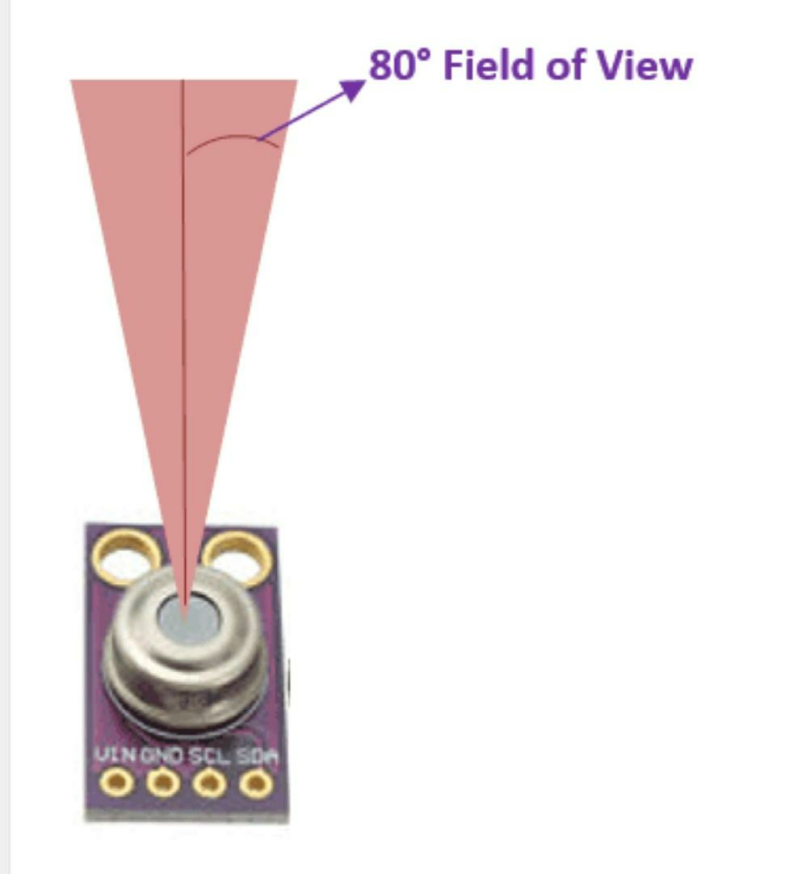
• Operating Voltage: 3.6V to 5V

• Object Temperature Range: -70°C to 382.2°C

• Ambient Temperature Range: -40°C to 125°C

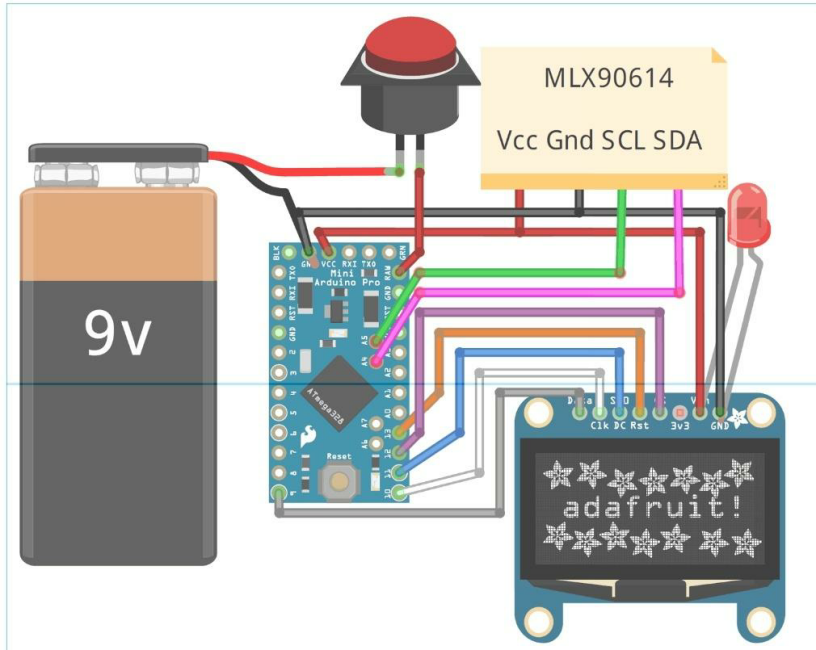
• Resolution/Accuracy: 0.02°C

**Distance between the Sensor and the Object:**

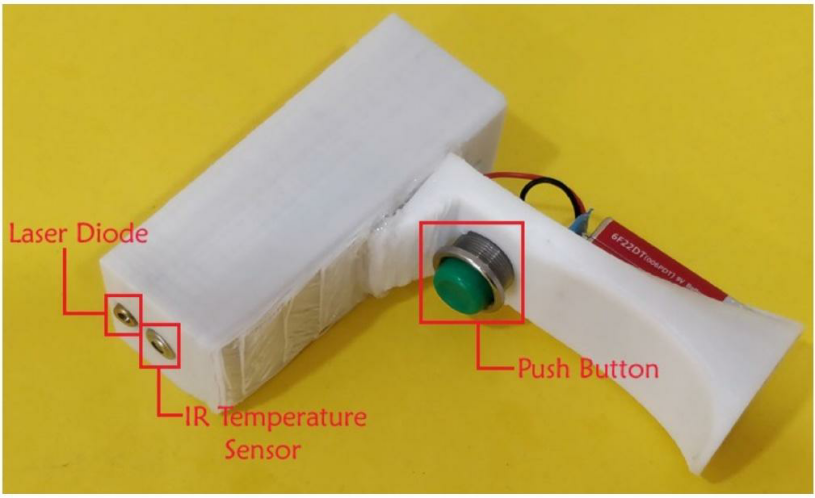
The value of this distance is given by the term Field of View (FOV), for our sensor the field of view is about 80°.

You can think of the sensing range to be in a conical shape from the point of sensor as shown above. So, as we go far from the measuring object the sensing area increases by two folds. Meaning for every 1cm we move away from the object the sensing area grows by 2cm. In our thermal gun we have placed a laser diode on top of the sensor to know where the sensing area of the sensor is currently pointing at. I found that the values were reliable if the gun is pointed at 2cm away from the object and the accuracy goes down as we move away.

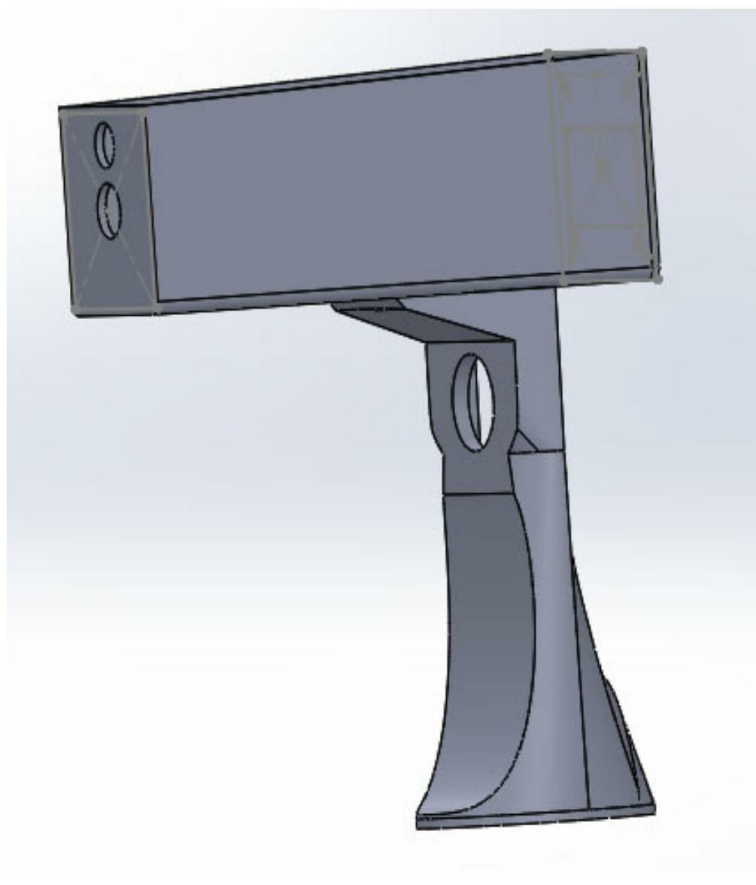
**Circuit Diagram:**

****

The Fritzing Software did not support a part for the MLX90614 sensor. We have used a note to mention its connections as shown above, also we have used a red colour LED in place of a laser diode. The entire circuit is powered by the 9V battery through a push button. When the push button is pressed the 9V battery is connected to the RAW pin of Arduino which is then regulated to 5V using the on-board voltage regulator. This 5V is then used to power the OLED module, Sensor and Laser diode.



**Block Diagram of Infrared Thermometer :**

To make the project more interesting and practically usable we have 3D modeled and printed our outer casing for our thermal gun. The design involves two parts, one is the top part which acts as the body of the gun housing the Arduino controller, OLED, Sensor and the Laser diode. The other is the bottom part which acts as a handle of the gun housing the battery and push button. The Push button here acts as the trigger. The model looks like this above diagram.

**Programming for Infrared Thermometer :**

The Program for Arduino should read the temperature value from the MLX90614 and display it on the OLED display.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Arduino Contactless thermometer

MLX90614 I2C connection

OLED 4-wire SPI connection

Dated: 14/04/2020

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <Wire.h>

#include <SparkFunMLX90614.h>

#include <SPI.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

// If using software SPI (the default case):

#define OLED\_MOSI 9

#define OLED\_CLK 10

#define OLED\_DC 11

#define OLED\_CS 12

#define OLED\_RESET 13

Adafruit\_SSD1306 display(OLED\_MOSI, OLED\_CLK, OLED\_DC, OLED\_RESET, OLED\_CS);

IRTherm therm;

void setup()

{

Serial.begin(9600);

therm.begin();

therm.setUnit(TEMP\_C);

display.begin(SSD1306\_SWITCHCAPVCC);

display.clearDisplay();

display.setRotation(2);

}

String temperature;

char runner;

void loop()

{

if (therm.read()) // On success, read() will return 1, on fail 0.

{

temperature = String(therm.object(), 2);

Serial.print("Object: ");

Serial.print(temperature); Serial.println("C");

display.clearDisplay();

runner++;

delay(5);

}

display.setTextSize(2);

display.setTextColor(WHITE);

display.setCursor(display.width()/4,display.height()/12);

display.println(temperature);

display.drawLine(display.width()/runner,display.height() - display.height()/2.5, display.width()/runner+1, display.height() - display.height()/2.5, WHITE);

display.setCursor(0,display.height()-display.height()/4);

display.setTextSize(1);

display.println(" Arduino Thermal Gun");

display.setCursor(display.width()- display.width()/4,display.height()/12);

display.println("deg C");

display.display();

if (runner>20)

runner=0;

}

By-Taufikur Rahaman

Asansol Engineering college (108)

Electronics & Communication Department