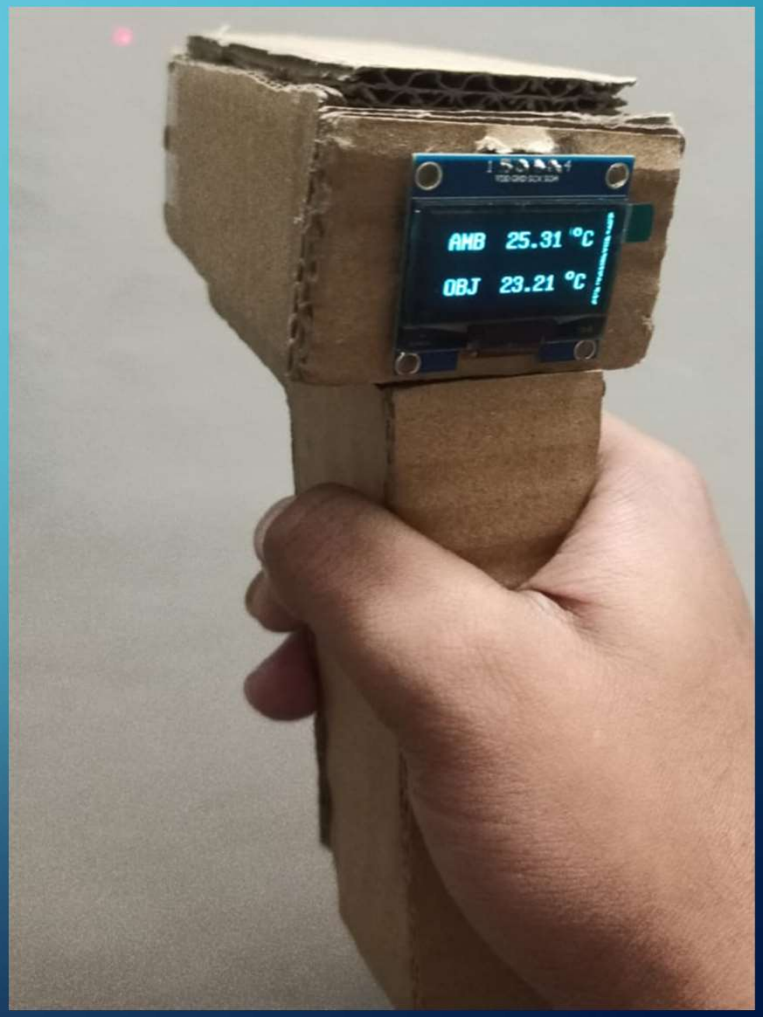
INFRARED TEMPERATURE GUN

GROUP MEMBERS –

- BT21ECE060 – ASHAY ATKAR
- BT21ECE061 – SHREYA GUPTA
- BT21ECE063 – KISHORE P CHANDRA



CONTENTS

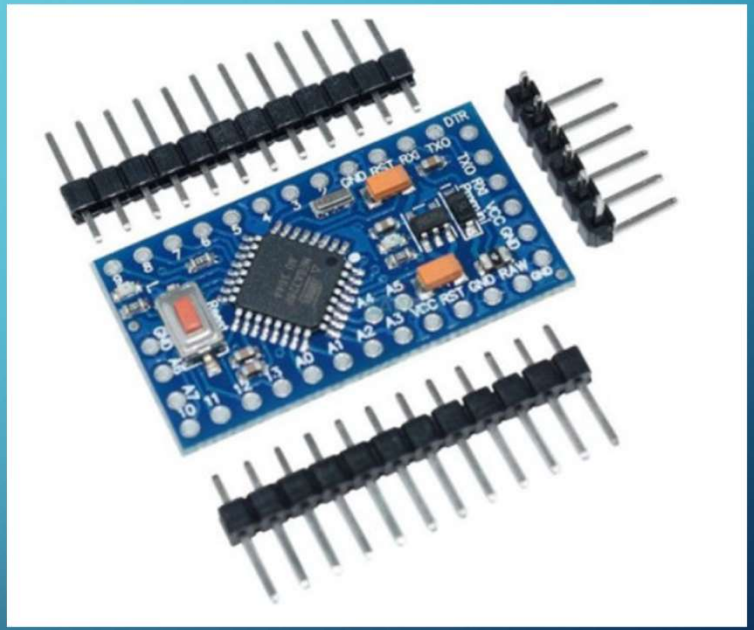
1. Hardware Requirements
2. Software Requirements
3. Source code for Arduino Pro Mini
4. Circuit Diagram
5. Images of Temperature Gun

HARDWARE REQUIREMENTS

1. Arduino Pro Mini board
2. IR Temperature sensor (MLX90614)
3. OLED Display (128 x 64)
4. Rechargeable Lithium Ion Battery (3.7V, 2500mAH)
5. Jumper Wires (For connections)
6. Push Switch (For switching on the Temperature Gun)
7. Battery Protector Module
8. Laser Diode (For pointing towards the object)

Arduino Pro Mini Board

The Arduino Pro Mini is a microcontroller board. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. (Brain of our Temperature Gun).



IR Temperature sensor (MLX90614)

This is the sensing unit of our gun.

It uses Infrared radiations for sensing the temperature. Every objects and living beings emit **IR Energy** and therefore the intensity of this emitted IR energy is going to be directly proportional to the temperature of that object or living being.

1. Object temperature range = -70 to 382 degrees celsius
2. Ambient temperature range = -40 to 125 degree celsius
3. Distance between object and sensor - 3 - 5 cm

$$\frac{P}{A} = \sigma T^4 \; j/m^2 s \quad \text{Stefan-Boltzmann Law}$$

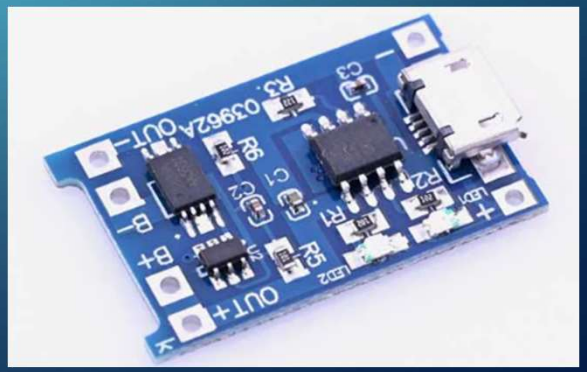
$$\sigma = 5.6703 x 10^{-8} \; watt/m^2 K^4$$

OLED Display

This acts as an interface between User and Temperature gun . It converts digital signals to a readable form and gives information about Ambient temperature and Object temperature.

Battery Protector module

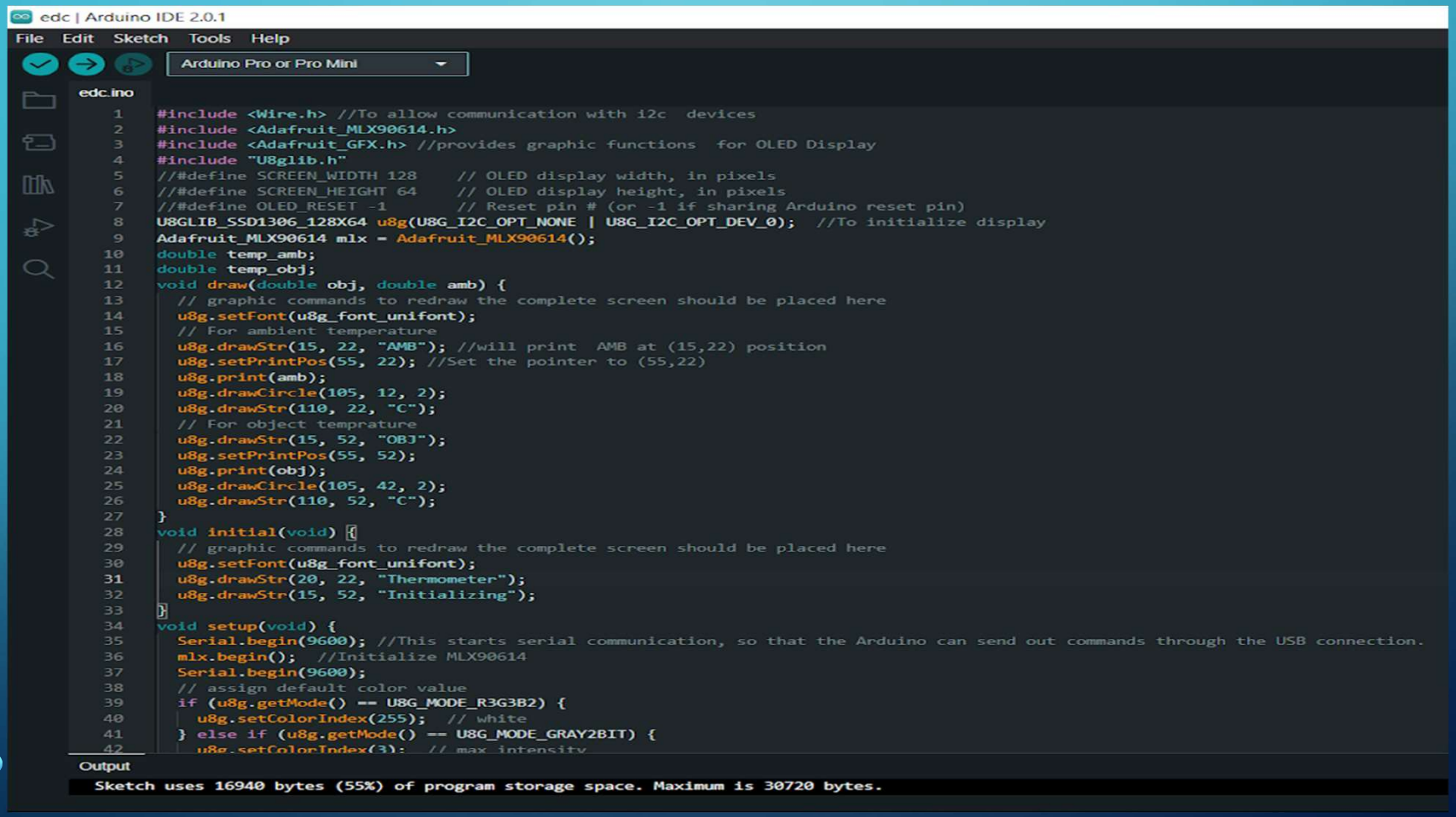
Battery Protector module allows us to charge our battery by a 5V DC source. Also, it protects the battery from too high and low voltages while charging. It also protects from overcharging.

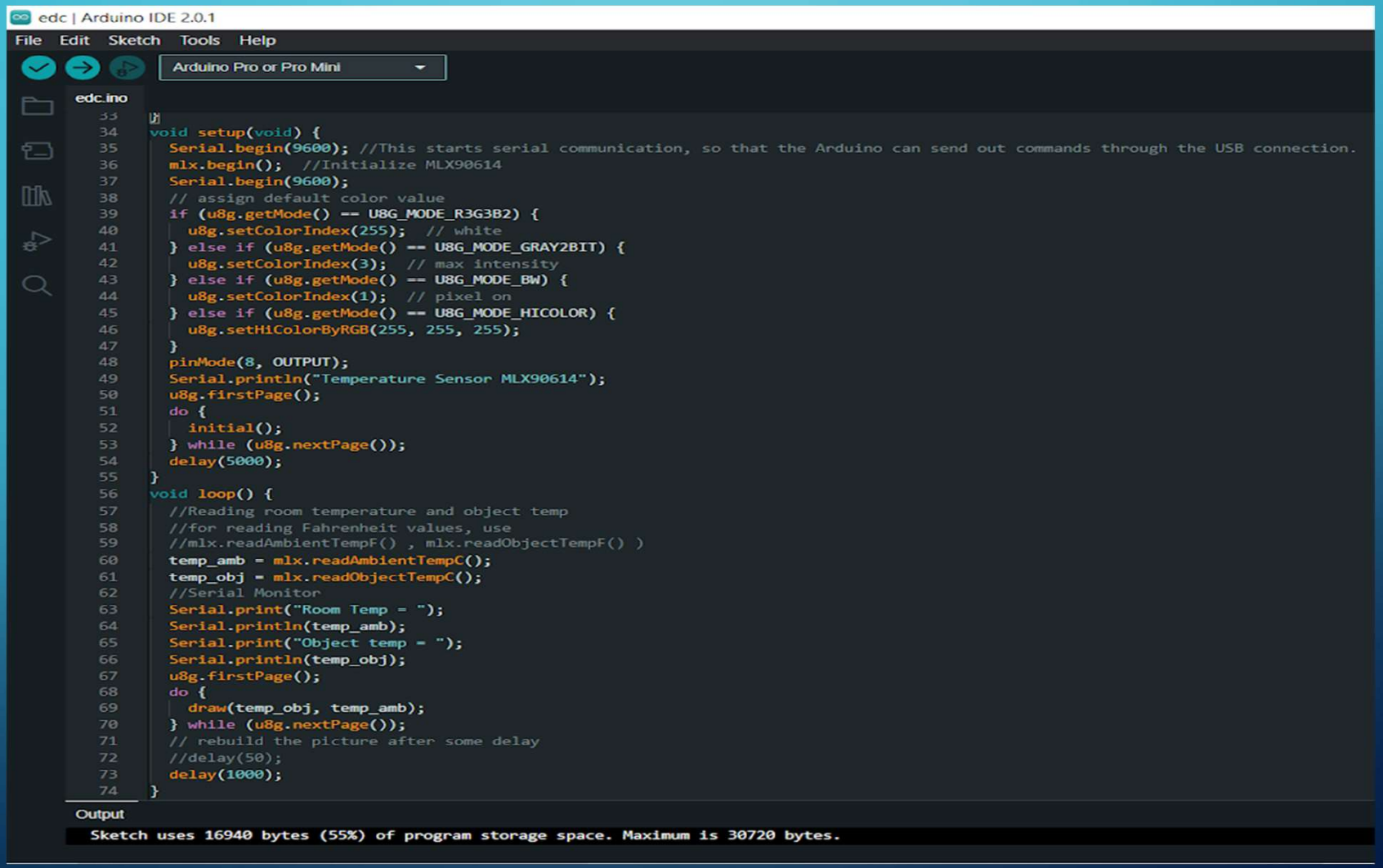


SOFTWARE REQUIREMENTS

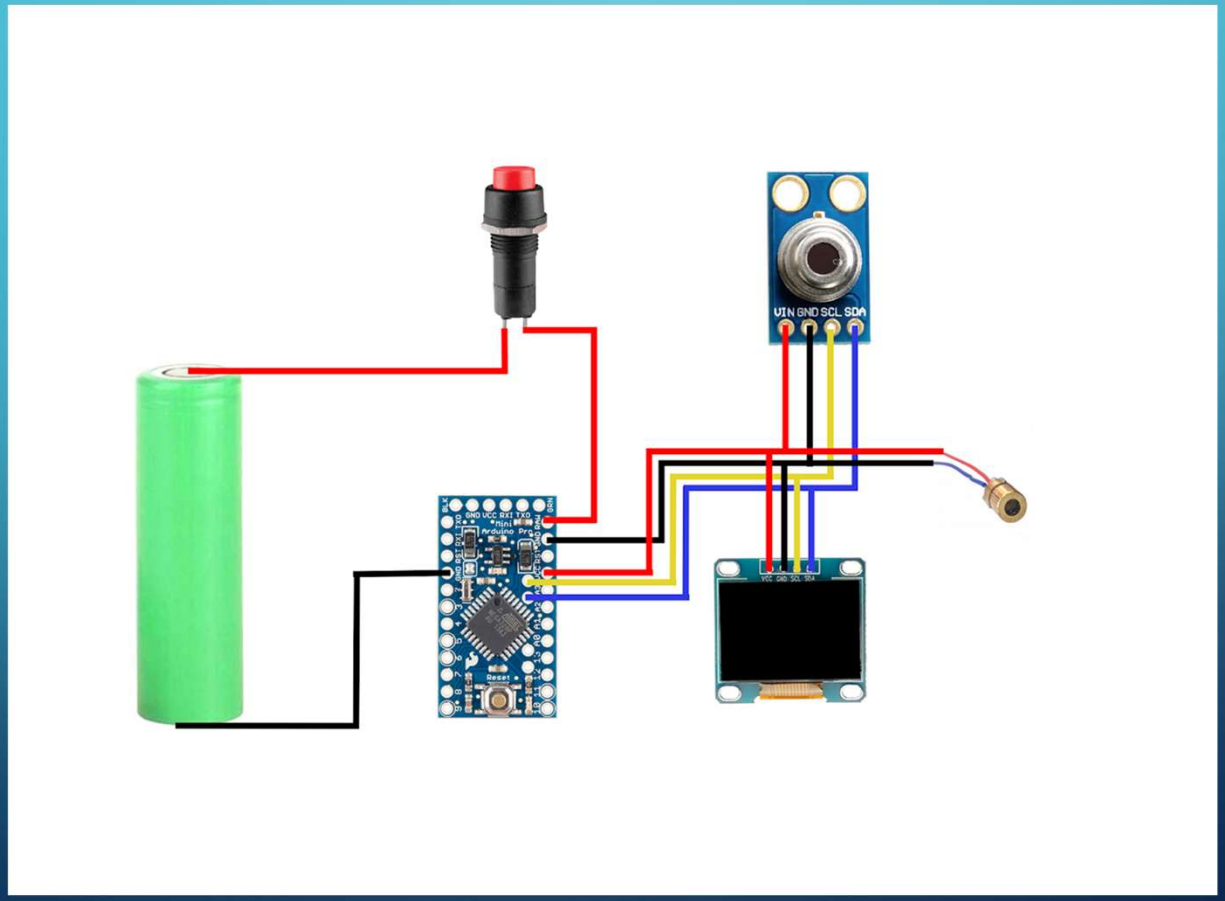
1. Arduino IDE (For writing and uploading code)
2. Wire Library
3. Adafruit_MLX90614 Library
4. Adafruit_GFX Library
5. U8glib Library

SOURCE CODE

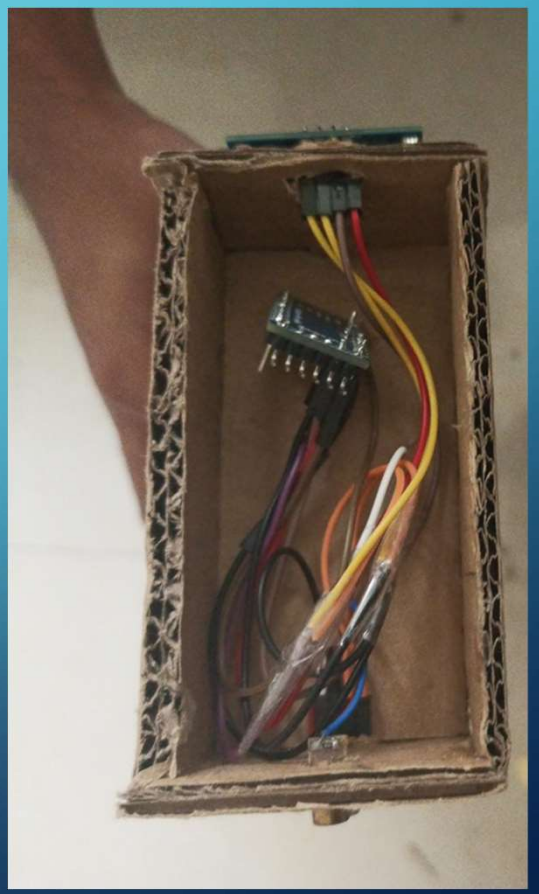

edc | Arduino IDE 2.0.1
File Edit Sketch Tools Help
Arduino Pro or Pro Mini
edc.ino
#include <Wire.h> //To allow communication with i2c devices
#include <Adafruit_MLX90614.h>
#include <Adafruit_GFX.h> //provides graphic functions for OLED Display
#include "U8glib.h"
//#define SCREEN_WIDTH 128 // OLED display width, in pixels
//#define SCREEN_HEIGHT 64 // OLED display height, in pixels
//#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE | U8G_I2C_OPT_DEV_0); //To initialize display
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
double temp_amb;
double temp_obj;
void draw(double obj, double amb) {
// graphic commands to redraw the complete screen should be placed here
u8g.setFont(u8g_font_unifont);
// For ambient temperature
u8g.drawStr(15, 22, "AMB"); //will print AMB at (15,22) position
u8g.setPrintPos(55, 22); //Set the pointer to (55,22)
u8g.print(amb);
u8g.drawCircle(105, 12, 2);
u8g.drawStr(110, 22, "C");
// For object temprature
u8g.drawStr(15, 52, "OBJ");
u8g.setPrintPos(55, 52);
u8g.print(obj);
u8g.drawCircle(105, 42, 2);
u8g.drawStr(110, 52, "C");
}
void initial(void) {
// graphic commands to redraw the complete screen should be placed here
u8g.setFont(u8g_font_unifont);
u8g.drawStr(20, 22, "Thermometer");
u8g.drawStr(15, 52, "Initializing");
}
void setup(void) {
Serial.begin(9600); //This starts serial communication, so that the Arduino can send out commands through the USB connection.
mlx.begin(); //Initialize MLX90614
Serial.begin(9600);
// assign default color value
if (u8g.getMode() == U8G_MODE_R3G3B2) {
u8g.setColorIndex(255); // white
} else if (u8g.getMode() == U8G_MODE_GRAY2BIT) {
Output
Sketch uses 16940 bytes (55%) of program storage space. Maximum is 30720 bytes.


edc | Arduino IDE 2.0.1
File Edit Sketch Tools Help
Arduino Pro or Pro Mini
edc.ino
}
void setup(void) {
Serial.begin(9600); //This starts serial communication, so that the Arduino can send out commands through the USB connection.
mlx.begin(); //Initialize MLX90614
Serial.begin(9600);
// assign default color value
if (u8g.getMode() == U8G_MODE_R3G3B2) {
u8g.setColorIndex(255); // white
} else if (u8g.getMode() == U8G_MODE_GRAY2BIT) {
u8g.setColorIndex(3); // max intensity
} else if (u8g.getMode() == U8G_MODE_BW) {
u8g.setColorIndex(1); // pixel on
} else if (u8g.getMode() == U8G_MODE_HICOLOR) {
u8g.setHiColorByRGB(255, 255, 255);
}
pinMode(8, OUTPUT);
Serial.println("Temperature Sensor MLX90614");
u8g.firstPage();
do {
initial();
} while (u8g.nextPage());
delay(5000);
}
void loop() {
//Reading room temperature and object temp
//for reading Fahrenheit values, use
//mlx.readAmbientTempF() , mlx.readObjectTempF())
temp_amb = mlx.readAmbientTempC();
temp_obj = mlx.readObjectTempC();
//Serial Monitor
Serial.print("Room Temp = ");
Serial.println(temp_amb);
Serial.print("Object temp = ");
Serial.println(temp_obj);
u8g.firstPage();
do {
draw(temp_obj, temp_amb);
} while (u8g.nextPage());
// rebuild the picture after some delay
//delay(50);
delay(1000);
}
Output
Sketch uses 16940 bytes (55%) of program storage space. Maximum is 30720 bytes.

CIRCUIT DIAGRAM



IMAGES OF TEMPERATURE GUN



Thankyou