**CONTACTLESS**

Contactless Temperature Monitoring System

**SYSTEM**

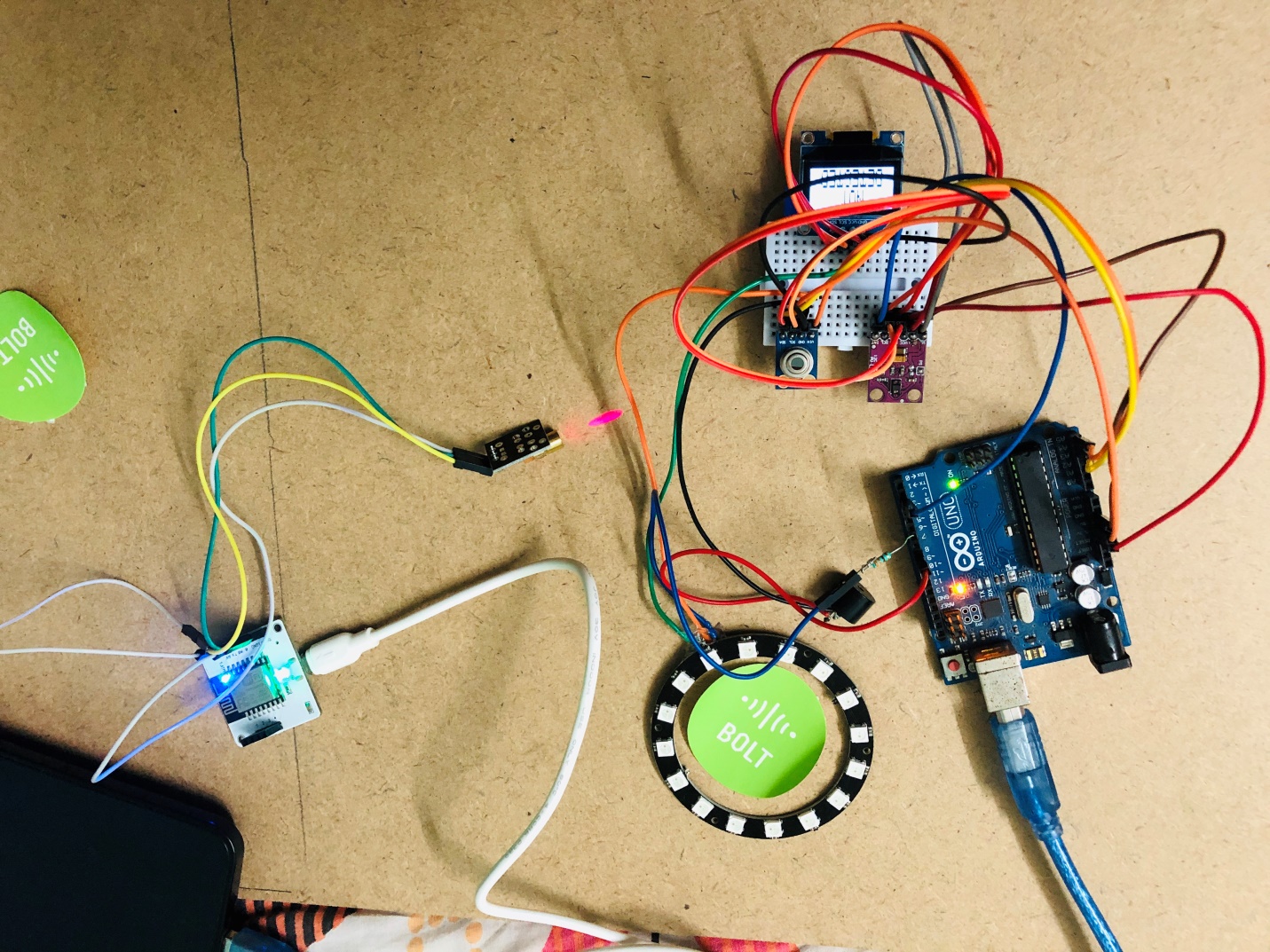
**MONITORING**

**TEMPERATURE**

1. Overview​
2. Things used in this project​
3. Hardware components​
4. Software apps and online services
5. Hand tools and fabrication machines
6. Story
7. INTRODUCTION
8. DEMONSTRATION
9. DETECTION OF TEMPERATURE
10. ALERTING IN ABNORMAL TEMPERATURE
11. SENDING SMS TO THE USER
12. Schematics​

OVERVIEW

Detect the temperature of a person contactlessly within a certain range and allowing only the one who have a normal body temperature by showing a wheel of green led, and showing the wheel of red led if anyone above normal body temperature comes and will let the securities know about that by turning on the buzzer and sensing message of such case to the admin or to the authorities.



THINGS USED IN THE PROJECT

Hardware components:-

1. Bolt IoT Bolt Wi-Fi Module
2. Arduino UNO
3. Buzzer
4. MLX90614 IR Temperature Sensor
5. APDS-9960 Proximity Sensor
6. 128x32 12C OLED Display
7. RGB Led Ring(WS2812 5050)
8. LASER module
9. Multi-Purpose PCB Board
10. Jumper wires
11. Usb cables for Bolt and Arduino

Software apps and online services:-

1. Arduino IDE
2. Google-Colab
3. Twilio

Hand tools and fabrication machines:-

1. Soildering Iron

Story

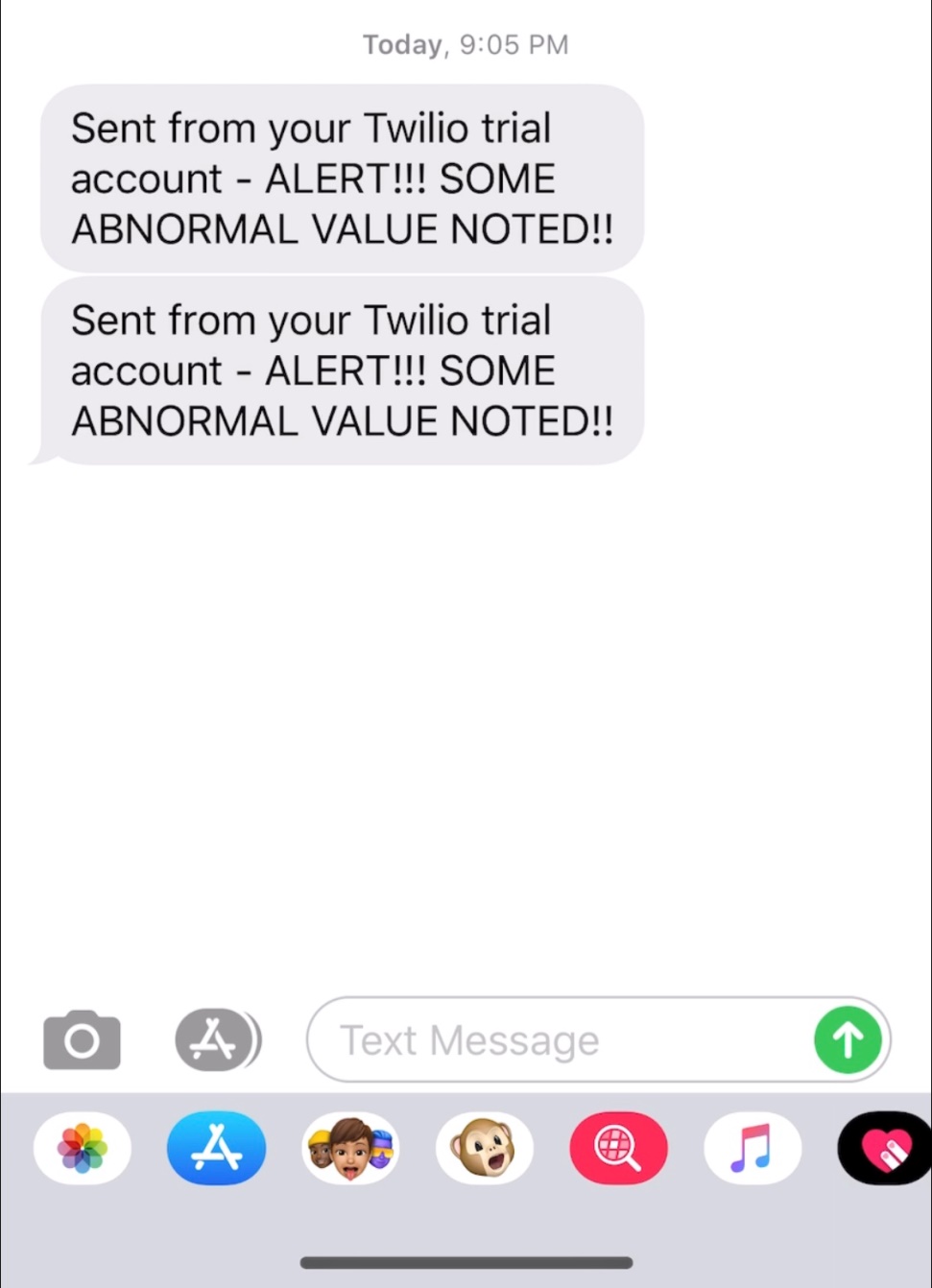
* + 1. Introduction:-

This project has primarily three functionalities:-

* 1. To check the person’s temp. and to let him enter if he/she is having normal temp. Range.
  2. To forbid the person, who is not having a normal temp, to allow, moreover it will indicate with a red indication to the person and also will alert the security by turning on the buzzer.
  3. Finally it will send message to the authorities that a person with abnormal temperature had entered and also to take necessary actions.

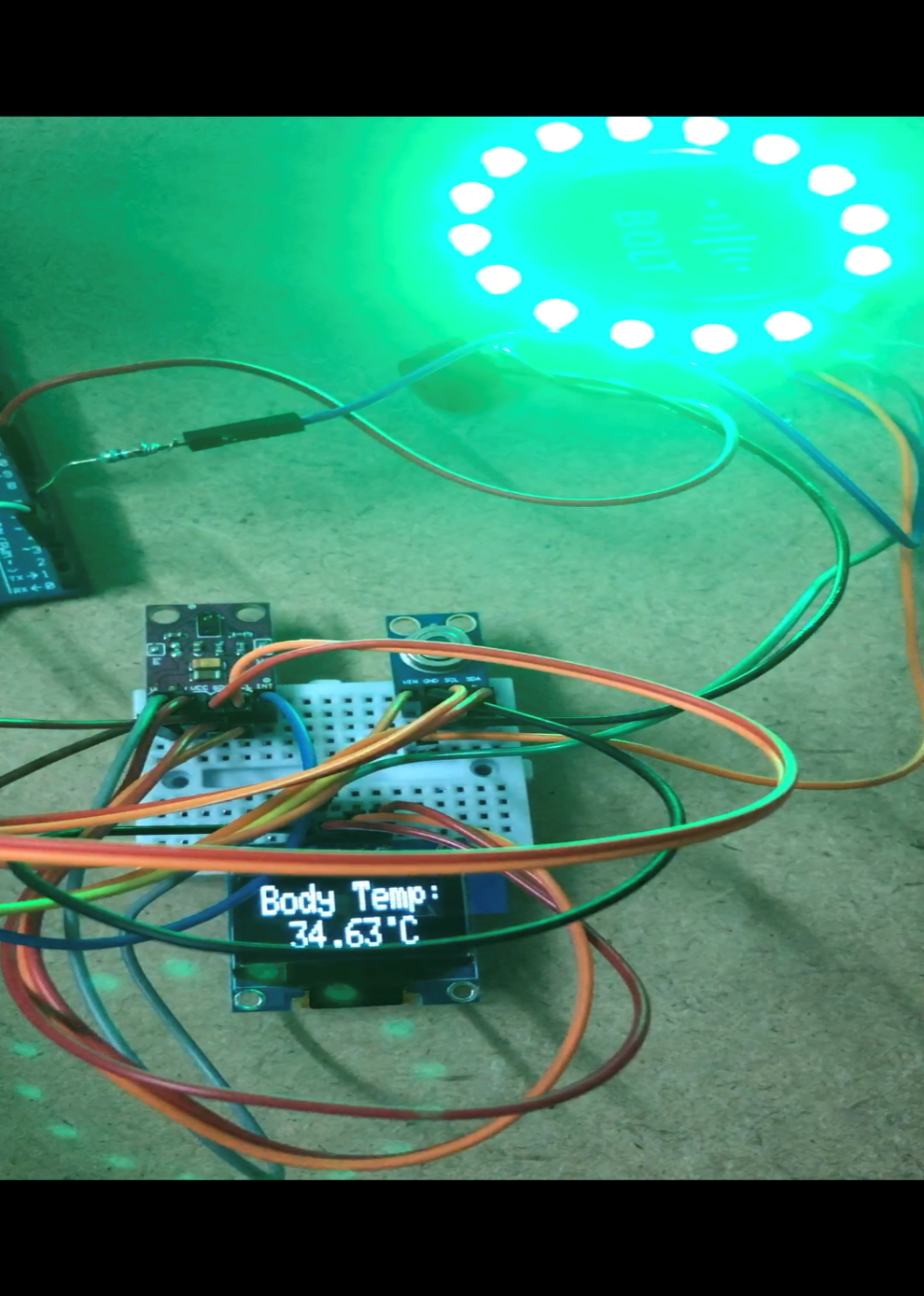
Now in heavily populated place like the supermarket or shopping mall where huge number of people come and go, having a person with higher than normal temperature can be harmful for the other customer too and to handle this type situation a bolt device have been made to notify the authorities from some person having abnormal body temp.

1. DEMONSTRATION:-



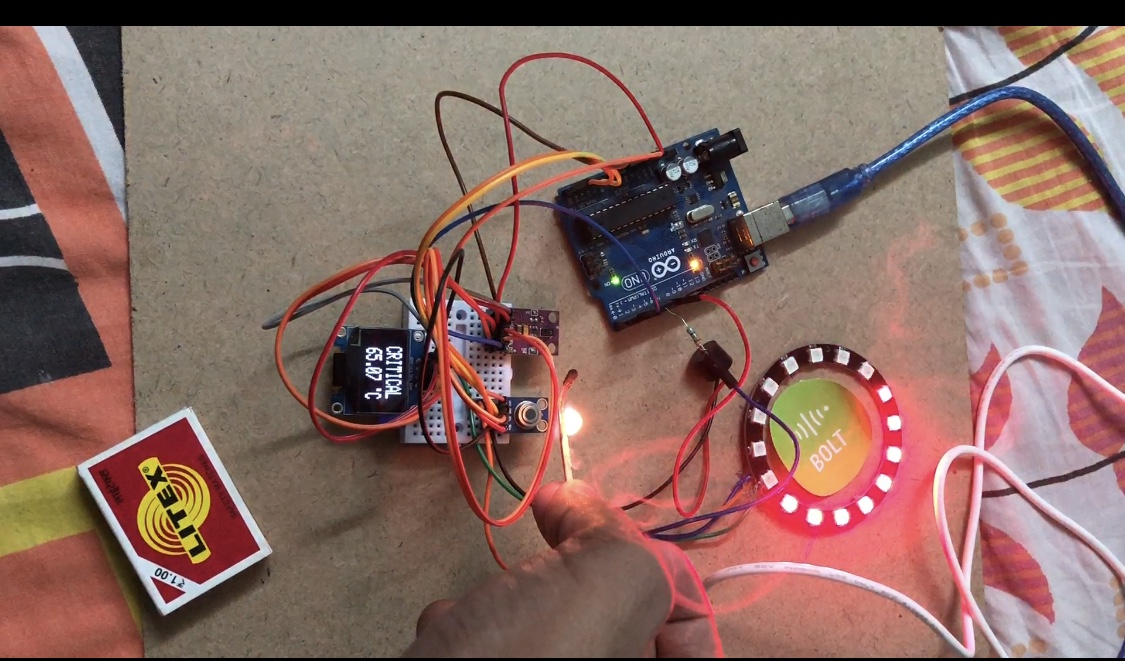
1. DETECTION OF TEMPERATURE:-

When the proximity sensor sense a value among the certain range it allows the IR temp. sensor to collect the temperature of the object and to calculate it’s value, if it is normal only the green led will turn on and the display will show the normal body temp.



1. ALERTING IN ABNORMAL TEMPERATURE

Again when the proximity sensor senses a value among the certain range it allows the IR temp. Sensor to collect the temperature of the object and to calculate its value, if it is not normal the led will do a wheel like pattern fastly and will also turn on the buzzer informing the security. Here I had use a matchstick to show the high temp. result.



1. SENDING SMS TO THE USER

Now, the sensor will also inform the authorities, to take necessary action and to forbid that particular person from entering for the shake and health of the other customers.



SHEMATICS



CODE FOR THE DEVICE:-

ARDIONO CODE:-

#include <Wire.h>

#include <SparkFun\_APDS9960.h>

#include <Adafruit\_MLX90614.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define OLED\_RESET 4

Adafruit\_SSD1306 display(OLED\_RESET);

#include <Adafruit\_NeoPixel.h>

#ifdef \_\_AVR\_\_

#include <avr/power.h>

#endif

#define PIN 6

#define NUMPIXELS 16

Adafruit\_NeoPixel pixels(NUMPIXELS, PIN, NEO\_GRB + NEO\_KHZ800);

#define DELAYVAL 10

SparkFun\_APDS9960 apds = SparkFun\_APDS9960();

uint8\_t proximity\_data = 0;

Adafruit\_MLX90614 mlx = Adafruit\_MLX90614();

void setup() {

#if defined(\_\_AVR\_ATtiny85\_\_) && (F\_CPU == 16000000)

clock\_prescale\_set(clock\_div\_1);

#endif

pinMode(3,OUTPUT);

pixels.begin();

mlx.begin();

apds.init();

apds.enableProximitySensor(false);

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C); // initialize with the I2C addr 0x3C (for the 128x32)

Serial.begin(9600);

}

void loop() {

pixels.clear();

String temperature = "";

/\*

Serial.print("Ambient ");

Serial.print(mlx.readAmbientTempC());

Serial.print(" C");

Serial.println();

Serial.print("Target ");

Serial.print(mlx.readObjectTempC());

Serial.print(" C");

Serial.println();

\*/

apds.readProximity(proximity\_data);

if (proximity\_data == 255 && mlx.readObjectTempF() < 100) {

temperature = String(mlx.readObjectTempC(), 1);

Serial.println(mlx.readObjectTempC());

display.clearDisplay();

display.invertDisplay(false);

display.setTextSize(2);

display.setTextColor(WHITE);

display.setCursor(8, 0);

display.clearDisplay();

display.println("Body Temp:");

display.setCursor(25, 18);

display.print(mlx.readObjectTempC());

display.setCursor(85, 8);

display.print(".");

display.setTextColor(WHITE);

display.setCursor(85, 18);

display.print(" C");

display.display();

for(int i=0; i<NUMPIXELS; i++)

{

pixels.setPixelColor(i, pixels.Color(0, 150, 0));

pixels.show();

delay(50);

}

delay (10);

for(int i=0; i<NUMPIXELS; i++)

{

pixels.setPixelColor(i, pixels.Color(0, 0, 0));

pixels.show();

delay(50);

}

delay(1000);

}

if (proximity\_data == 255) {

if (mlx.readObjectTempF() > 102) {

digitalWrite(3, HIGH);

noTone(5);

// play a note on pin 8 for 500 ms:

tone(8, 523, 500);

Serial.println(mlx.readObjectTempC());

display.clearDisplay();

display.invertDisplay(false);

display.setTextSize(2);

display.setTextColor(WHITE);

display.setCursor(20, 0);

display.clearDisplay();

display.println("CRITICAL");

display.invertDisplay(true);

display.setTextSize(2);

display.setTextColor(WHITE);

display.setCursor(20, 0);

display.clearDisplay();

display.println("CRITICAL");

display.invertDisplay(true);

display.setTextColor(WHITE);

display.setCursor(20, 0);

display.clearDisplay();

display.println("CRITICAL");

display.invertDisplay(false);

display.setTextSize(2);

display.setTextColor(WHITE);

display.setCursor(20, 0);

display.clearDisplay();

display.println("CRITICAL");

display.invertDisplay(true);

display.setTextSize(2);

display.setTextColor(WHITE);

display.setCursor(20, 0);

display.clearDisplay();

display.println("CRITICAL");

display.invertDisplay(true);

display.setTextColor(WHITE);

display.setCursor(20, 0);

display.clearDisplay();

display.println("CRITICAL");

display.invertDisplay(false);

display.setTextSize(2);

display.setTextColor(WHITE);

display.setCursor(20, 0);

display.clearDisplay();

display.println("CRITICAL");

display.setCursor(23, 18);

display.print(mlx.readObjectTempC());

display.setCursor(93, 8);

display.print(".");

display.setTextColor(WHITE);

display.setCursor(93, 18);

display.print(" C");

display.display();

for(int i=0; i<NUMPIXELS; i++)

{

pixels.setPixelColor(i, pixels.Color(150, 0, 0));

pixels.show();

delay(10);

}

delay (10);

for(int i=0; i<NUMPIXELS; i++)

{

pixels.setPixelColor(i, pixels.Color(0, 0, 0));

pixels.show();

delay(10);

}

delay(1000);

digitalWrite(3, LOW);

}

}

if (proximity\_data == 255) {

if (mlx.readObjectTempF() > 100) {

display.setCursor(93, 8);

display.print(".");

display.setTextColor(WHITE);

display.setCursor(93, 18);

display.print(" C");

display.display();

}

}

else if (proximity\_data <= 255) {

delay(1000);

display.clearDisplay();

display.invertDisplay(true);

display.setTextSize(2.8);

display.setTextColor(WHITE);

display.setCursor(45, 1);

display.clearDisplay();

display.println("NOT");

display.setCursor(17, 17);

display.print("DETECTED");

display.display();

digitalWrite(5, LOW);

}

}

PYTHON CODE:-

from boltiot import Bolt

import json,time

from boltiot import Sms

api\_key = "WRITE\_API\_KEY\_HERE"

device\_id = "WRITE\_DEVICE\_NAME\_HERE"

mybolt = Bolt(api\_key, device\_id)

from boltiot import Sms # Import SMS class from boltiot library

# Credentials required to send SMS

SID = 'You can find SID in your Twilio Dashboard'

AUTH\_TOKEN = 'You can find on your Twilio Dashboard'

FROM\_NUMBER = 'This is the no. generated by Twilio. You can find this on your Twilio Dashboard'

TO\_NUMBER = 'This is your number. Make sure you are adding +91 in beginning'

sms = Sms(SID, AUTH\_TOKEN, TO\_NUMBER, FROM\_NUMBER) # Create object to send SMS

a=mybolt.digitalRead('0')

data = json.loads(a)

b=int(data['success'])

sum=0

while(b == 1):

response =mybolt.digitalRead('0')

data1 = json.loads(response)

c=int(data1['value'])

if(c==0):

sum=0

if(c==1):

print(c)

sum=sum+1

if sum==1:

sms.send\_sms("Write your message here.") # Call function to send SMS!''

