REPORT

DAILY HEALTH

TEAM - AMPLITUDE

ABSTRACT

Daily Health is an Online Health Monitoring device controlled by IoT. This project can be used for measuring physiological parameters like Pulse Rate and Body Temperature. etc. This project is useful for observing senior citizens who are in need of frequent monitoring. This can also be applied to soldiers who are in remote regions without any medical assistance to monitor their own physiological factors. Arduino along with various sensors are used for measuring physiological parameters in this project. The Bolt Iot and the Bolt Cloud are used to send alert messages using Telegram(app). Graphs of pulse rate and temperature are plotted with the help of Arduino.

Two biosensors are used in this project for measuring pulse rate and temperature. The pulse rate is measured by using a Pulse Sensor and LM35, a precession Integrated circuit Temperature sensor, is used to measure the body temperature. The temperature sensor has a range between -55°C to 150°C. When the temperature goes beyond the threshold level, Bolt Cloud sends an alert message to the user via Telegram App.

CONTENT

INTRODUCTION

Daily Health is an Online Health Monitoring device controlled by IoT used for measuring physiological parameters like Pulse Rate and Body Temperature, etc. This is an effective way to monitor health without physical contact. This project is useful for observing senior citizens who are in need of frequent monitoring, and soldiers who are in remote regions without any medical assistance to monitor their own physiological factors. Arduino along with various sensors are used for measuring physiological parameters in this project. The Bolt Iot and the Bolt Cloud are used to send alert messages using Telegram (app) when the specified threshold value is exceeded. Graphs of pulse rate and temperature are plotted with the help of Arduino.

This is an humanitarian project aimed to provide efficient and adequate medical support when required. Handicapped citizens may use this project for their daily checkups. Rural areas lacking medical assistance facilities can make use of this project.

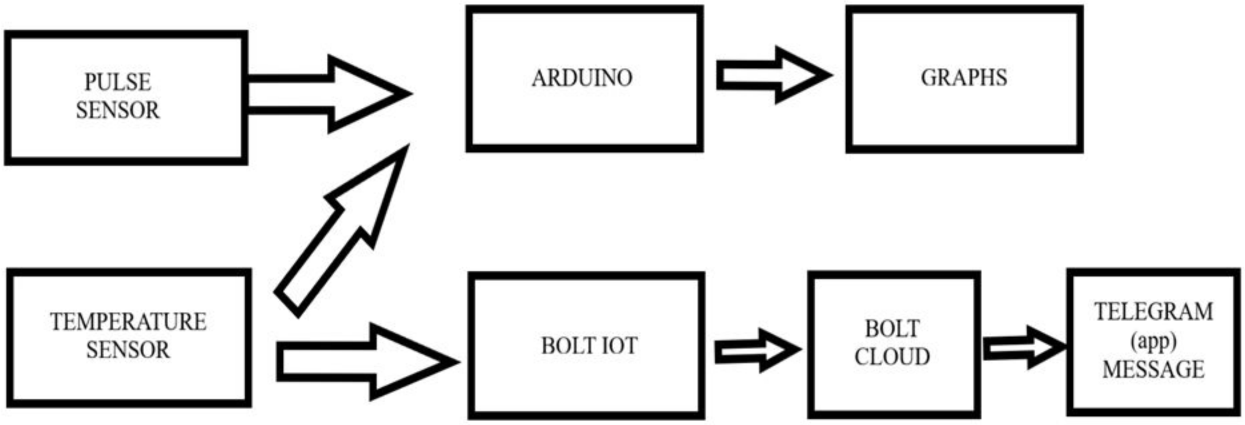
GOAL

Direct medical support is not available to everyone due to COVID-19 pandemic. This project is used for observing senior citizens who are in need of frequent monitoring and can not be physically present in the hospital. This can also be applied to soldiers who are in remote regions without any medical assistance to monitor their own physiological factors. This project will be an easy to use and an efficient medical assistance to bedridden patients to reduce the rate of hospital visits.

COMPONENTS

1. Arduino UNO
2. BOLT WiFi Module
3. Pulse Sensor
4. Temperature Sensor LM35
5. USB cable A/B for Arduino
6. Breadboard
7. Jumper wires
8. USB cable

WORKING



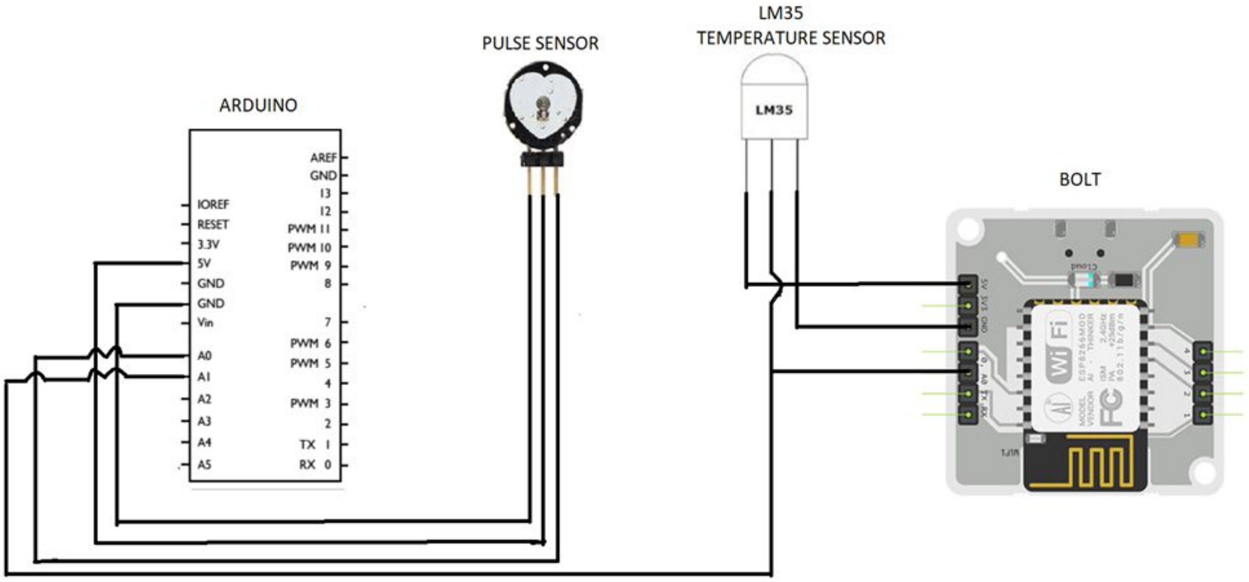
BLOCK DIAGRAM

Arduino UNO and BOLT IoT cloud are the main components of the project. The two Biosensors we use in this project are Pulse Sensor and Temperature sensor LM35. Pulse rate

sensor is used to detect heartbeats. It can be worn on the finger or earlobe and connected to Arduino via cables. It is an optical heart rate sensor integrated with amplifying circuit and noise cancellation circuit.

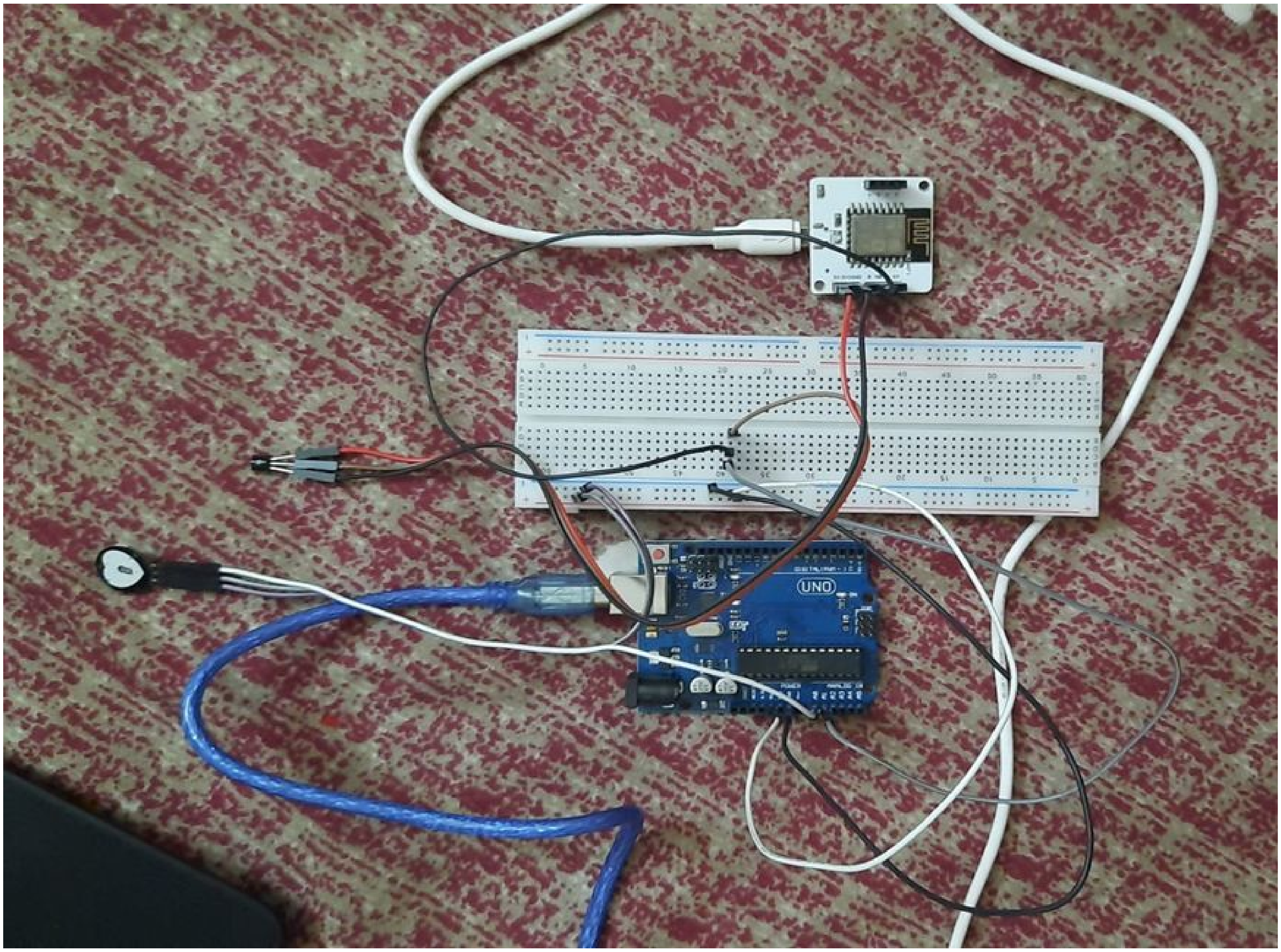
The Pulse Sensor Amped is a plug-and-play heart-rate sensor for Arduino. It essentially combines a simple optical heart rate sensor with amplification and noise cancellation circuitry making it fast and easy to get reliable pulse readings. The LED in the sensor starts emitting light when placed above the vein and hence detects the blood flow through the veins.

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.



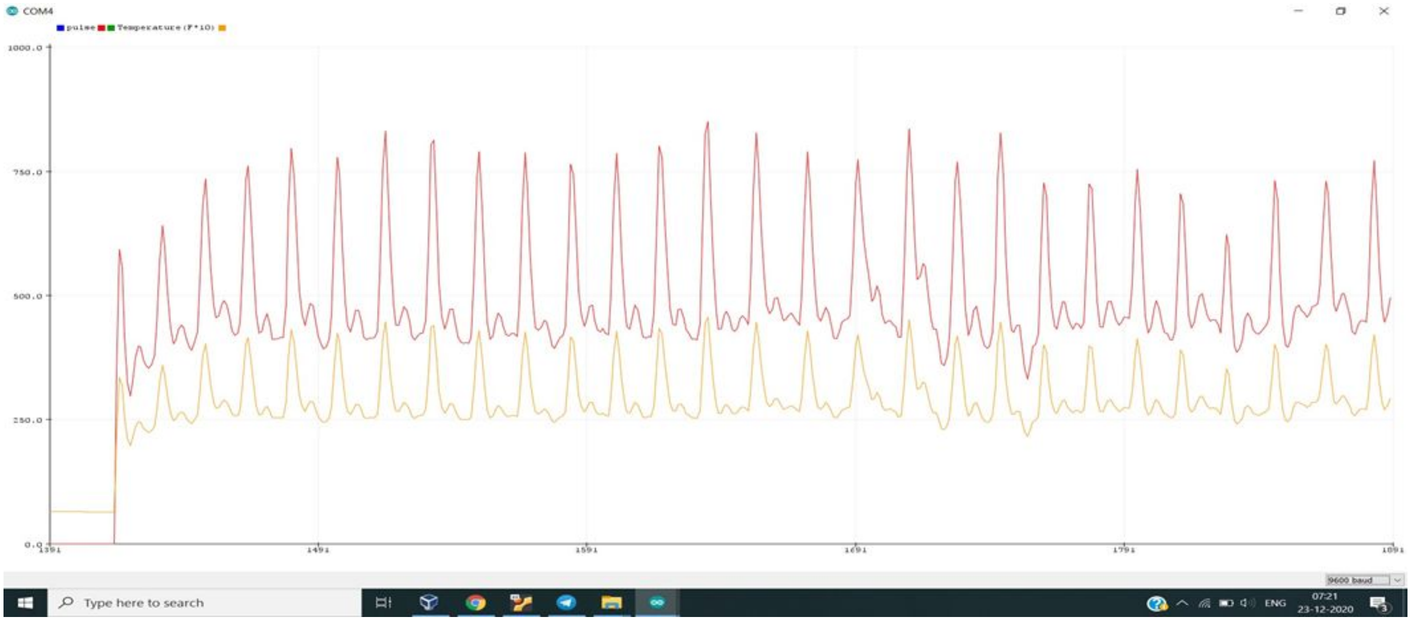
Arduino Board is used to transfer the temperature and pulse rate obtained to the local system and plot the graph of the pulse obtained. These graphs can be easily analysed by a person having little knowledge in this field and hence any abnormalities in pulse rate can be understood easily.

BOLT IoT module collects the information from the temperature sensor and pushes the same to the BOLT Cloud and plots the graph of temperature values obtained from the sensor, python code that runs on the Ubuntu Server helps us to send alert message to the user or the caretaker via telegram when the temperature value exceeds the threshold limit.

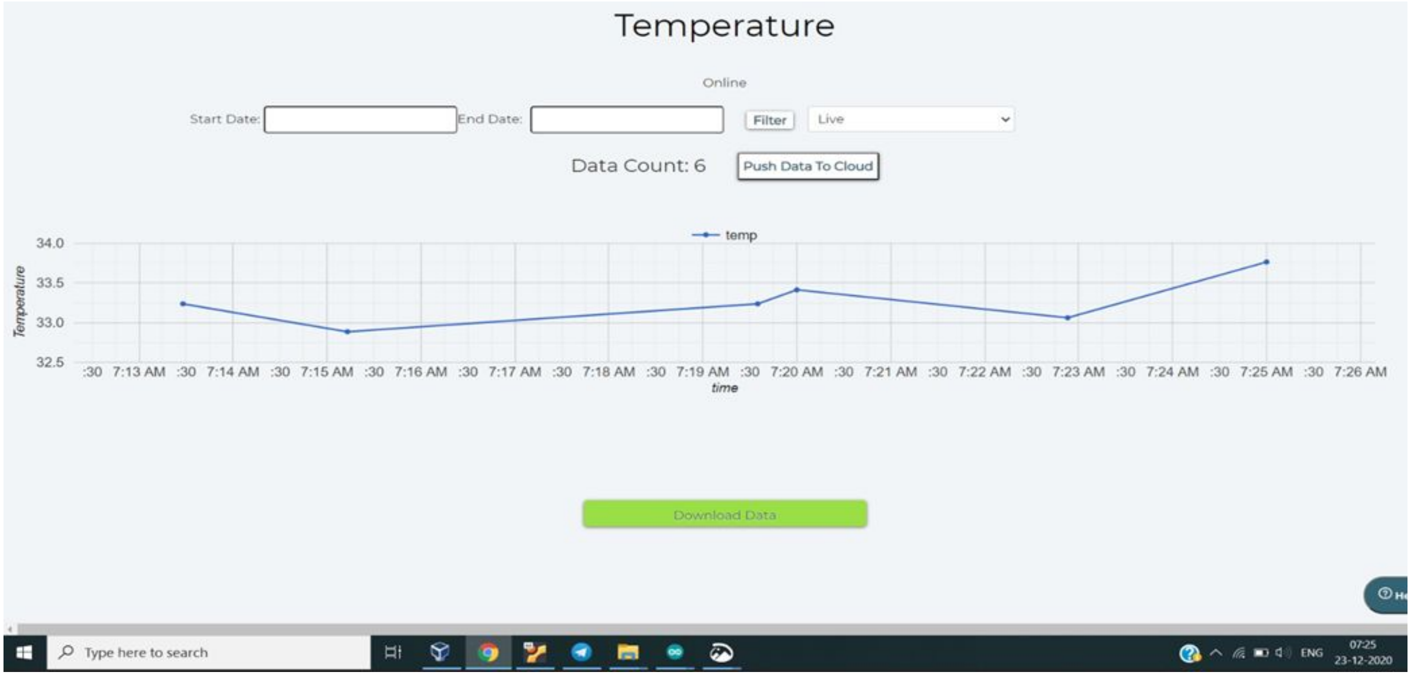


Circuit Connection

When the pulse sensor is placed on the fingertip it reads the value of pulse rate. This value is then given to the Arduino which plot graphs based on the input values.



The temperature sensor gives the value to the Bolt Cloud which plot graph based on the input values



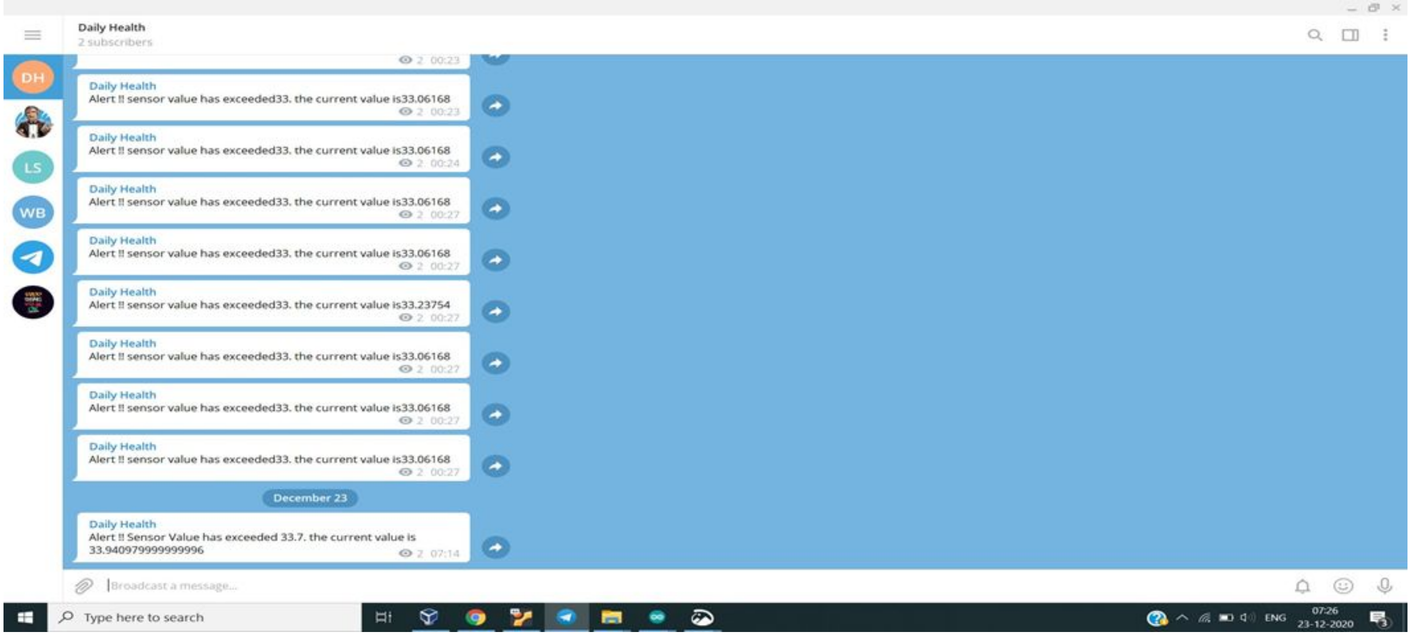
Temperature

Online

Start Date: End Date: Filter Live

Data Count: 6 Push Data To Cloud

When a value of temperature above threshold value is recorded, an alert is sent through Telegram(App).



Daily Health
2 subscribers

2 00:23

Daily Health
Alert !! sensor value has exceeded33. the current value is33.06168
2 00:23

Daily Health
Alert !! sensor value has exceeded33. the current value is33.06168
2 00:24

Daily Health
Alert !! sensor value has exceeded33. the current value is33.06168
2 00:27

Daily Health
Alert !! sensor value has exceeded33. the current value is33.06168
2 00:27

Daily Health
Alert !! sensor value has exceeded33. the current value is33.23754
2 00:27

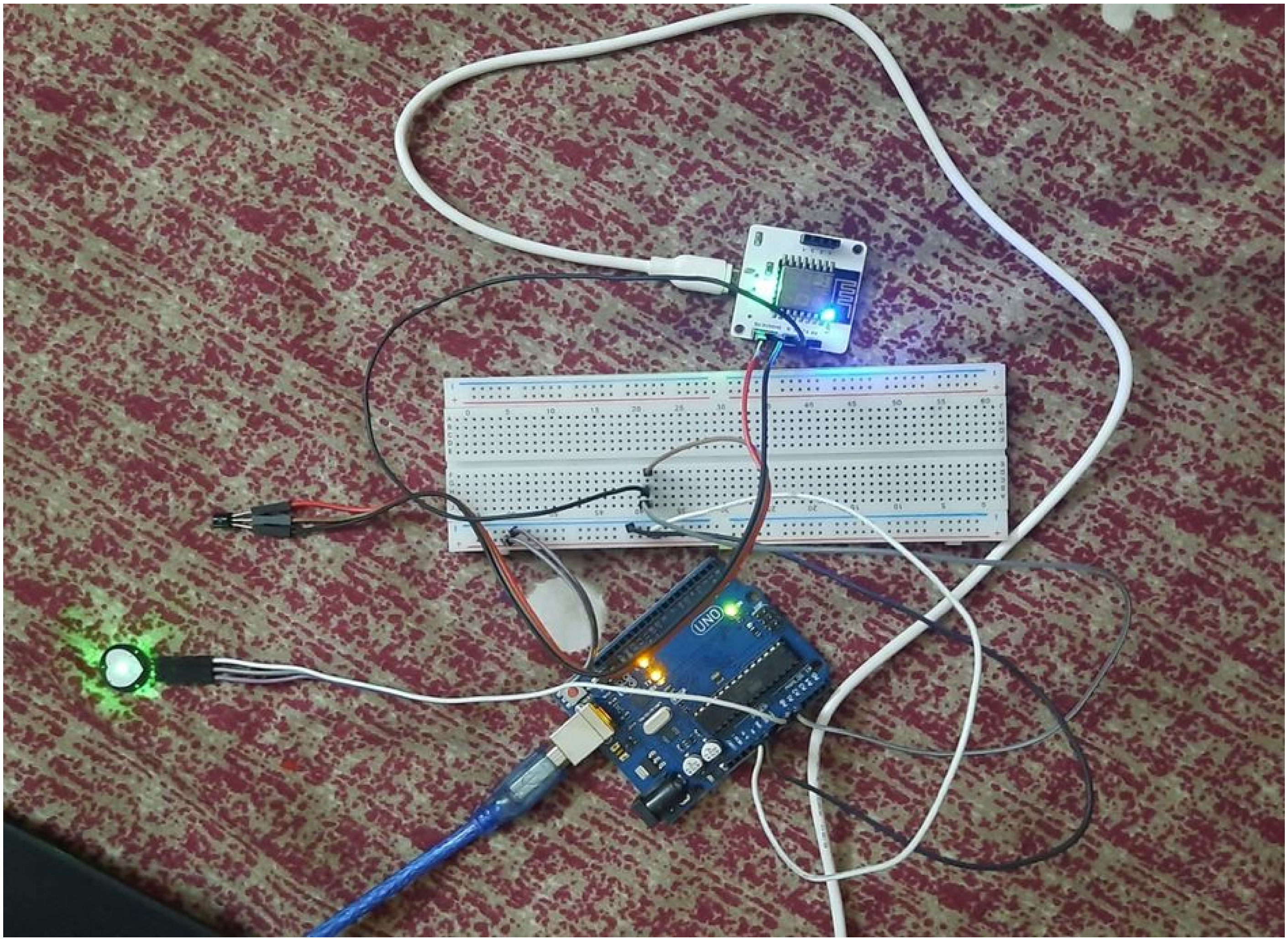
Daily Health
Alert !! sensor value has exceeded33. the current value is33.06168
2 00:27

Daily Health
Alert !! sensor value has exceeded33. the current value is33.06168
2 00:27

Daily Health
Alert !! sensor value has exceeded33. the current value is33.06168
2 00:27

December 23

Daily Health
Alert !! Sensor Value has exceeded 33.7. the current value is 33.940979999999996
2 07:14



Link to demonstration:
https://drive.google.com/file/d/1WWYhTOCYeyLDmtFRri899c-oQKfAEXvl/view?usp=sharing

CODE:

1. Arduino:

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  int pulse = analogRead(A0);
  int temp = analogRead(A2);
  Serial.print("pulse: ");
 Serial.print(pulse);
 Serial.print(" ");
  float mv = (temp / 2.048);
  float farh = (mv * 9) / 5;
  float tem = farh + 32;
  Serial.print("Temperature(F*10) :");
  Serial.println(tem);

}
```

2. Bolt code - JavaScript

```
setChartLibrary('google-chart');
setChartTitle('Temperature');
setAxisName('time','Temperature');
setChartType('lineGraph');
add(183);
mul(0.0977);
mul(1.8);
plotChart('time_stamp','temp');
```

3. Telegram - Python

1) Conf.py

```
bolt_api_key = "XXXX-XXXX-XX-XXX"          # This is your Bolt Cloud API Key
device_id = "BOLT2411545"              # This is the device ID
telegram_chat_id = "@XXXX"          # This is the channel ID of the created Telegram channel. Paste after @
telegram_bot_id = "botXXXX"         # This is the bot ID of the created Telegram Bot. Paste after bot
threshold = 33.7                    #threshold value of temperature
```

2) Telegram_alert.py

```
import requests             # for making HTTP requests
import json                # library for handling JSON data
import time                # module for sleep operation

from boltiot import Bolt       # importing Bolt from boltiot module
import conf                # config file
```

```
mybolt = Bolt(conf.bolt_api_key, conf.device_id)

def get_sensor_value_from_pin(pin):
    """Returns the sensor value. Returns -999 if request fails"""
    try:
        response = mybolt.analogRead(pin)
        data = json.loads(response)
        if data["success"] != 1:
            print("Request not successfull")
            print("This is the response->", data)
            return -999
        sensor_value = int(data["value"])
        return sensor_value
    except Exception as e:
                print("Something went wrong when returning the sensor value")
                print(e)
        return -999


def send_telegram_message(message):
    """Sends message via Telegram"""
    url = "https://api.telegram.org/" + conf.telegram_bot_id + "/sendMessage"
    data = {
        "chat_id": conf.telegram_chat_id,
        "text": message
    }
    try:
        response = requests.request(
            "POST",
            url,
            params=data
        )
        print("This is the Telegram URL")
        print(url)
        print("This is the Telegram response")
        print(response.text)
        telegram_data = json.loads(response.text)
        return telegram_data["ok"]
    except Exception as e:
        print("An error occurred in sending the alert message via Telegram")
        print(e)
        return False


while True:
    # Step 1
    sensor_value = get_sensor_value_from_pin("A0")
    print("The current sensor value is:", sensor_value)

    # Step 2
    if sensor_value == -999:
        print("Request was unsuccessfull. Skipping.")
        time.sleep(10)
        continue

    # Step 3
    if sensor_value >= conf.threshold:
        print("Sensor value has exceeded threshold")
        message = "Alert! Sensor value has exceeded " + str(conf.threshold) + \
                ". The current value is " + str(sensor_value)
        telegram_status = send_telegram_message(message)
        print("This is the Telegram status:", telegram_status)

    # Step 4
    time.sleep(20)
```

CONCLUSION

Daily Health is a project that conjoins the Medical and Engineering fields. This project is very useful for observing the physiological parameters of the senior citizens, patients, and soldiers. It can also be used in areas lacking an efficient medical assistance system. The setup is cost effective and easy to use without any prior knowledge.

RESULT

The project Daily Health is successfully implemented. The body temperature and pulse rate is measured successfully. The graph of temperature is plotted using Bolt Cloud and graph of pulse rate is plotted using arduino. Telegram alerts are also obtained at threshold value.

BIBLIOGRAPHY

- www.how2electronics.com
- https://trainings.boltiot.com
- https://forum.arduino.com
- https://www.instructables.com
- https://forum.boltiot.com/
- https://components101.com/
- https://en.wikipedia.org
- https://www.elprocus.com/
- blog.ouraring.com
- www.medicalnewstoday.com
- www.hackster.io