SMART ROVER

by

APURV CHOUDHARY	20BEC1233
SALUGU MANOJ	20BEC1213
PRATYUSH RAJ	20BEC1273

A project report submitted to

Dr. BALA MURUGAN MS

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECE4003 – IoT SYSTEM DESIGN AND APPLICATIONS

in

B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

APRIL 2022

BONAFIDE CERTIFICATE

Certified that this project report entitled “**SMART ROVER”** is a bonafide work of **APURV CHOUDHARY-20BEC1233, SALUGU MANOJ-20BEC1213, PRATYUSH RAJ- 20BEC1273** who carried out the Project work under my supervision and guidance for **ECE4003 – IoT SYSTEM DESIGN AND APPLICATIONS.**

Dr. Bala Murugan MS

Associate Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

A Smart Rover is a type of mobile controlled rover that aggregates data (like Temperature, Pressure, Altitude, etc). In this Project, a Smart Rover is designed, using hardware. Implemented in hardware, through an Arduino based Rover that uses BMP 280 sensor to aggregate data and Microcontroller (Uno, Nano and ESP-8266) to process data and send back to ThingSpeak. Other microcontrollers of equal or greater capability can also be used.

The BMP280 sensor integrates atmospheric pressure, temperature and relative humidity sensors in a single device, with great precision, low energy consumption and an ultra compact format. Based on BOSCH piezo-resistive technology with great EMC robustness, high precision and linearity, as well as long-term stability. It connects directly to a microcontroller via I2C or SPI.

Depending on the input signal received, the micro-controller redirects the robot to move in an alternate direction by actuating the motors which are interfaced to it through a motor driver.

Data collection is one of the most important aspects, and in our project, we are using it as an extension of a normal rover, by collecting temperature, pressure and altitude of the place where a human can’t go.

In this project we use different simulation, modelling, programming and animation tools to realize the final aims of the project.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Bala Murugan MS,** Assistant Professor (Senior), School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Susan Elias,** Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr. Mohanaprasad K** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

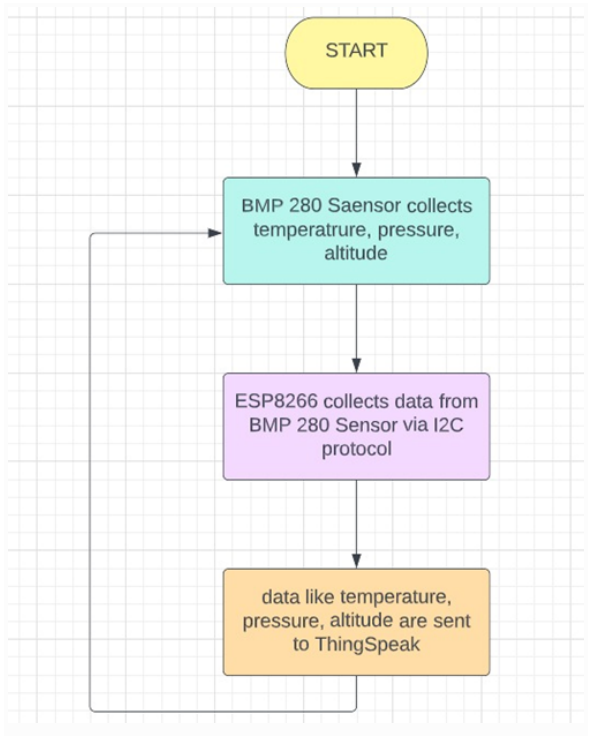
APURV CHOUDHARY **SALUGU MANOJ** **PRATYUSH RAJ**

TABLE OF CONTENTS

1. INTRODUCTION

Our smart rover system consists of a temperature sensor and pressure sensor for measuring temperature and pressure levels & an altitude sensor to measure altitude level.To get more information about a place or its environment where humans cannot reach easily due to various environmental limiting factors, a real-time monitoring system that continuously monitors the temperature , pressure and altitude of the area from which we need to gather the information about the environment has been build.

Proposed system block diagram


START
BMP 280 Saensor collects temperatrure, pressure, altitude
ESP8266 collects data from BMP 280 Sensor via I2C protocol
data like temperature, pressure, altitude are sent to ThingSpeak

LITERATURE REVIEW

There are very review review papers that give an account of the development of smart rovers from different aspects. Due to the rapid development of smart sensors and related analytical approaches, it is necessary to re-illustrate the trends and development frequently.

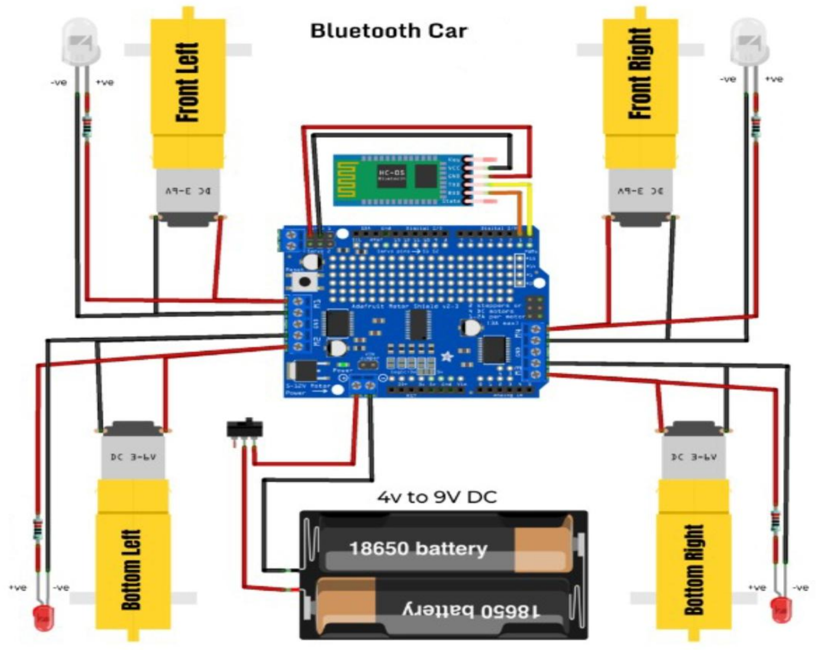
S No.	Title	Author	Findings	Publisher/Citations
1	Rocker Rover And Its Implementation In The Field Of Agriculture: A Review	Harshit Bisht	Advanced techniques to monitor plant growth	-
2	Design and Development of an Intelligent Rover for Mars Exploration(Updated)	Balakrishnan Shankar	Advanced techniques to study soil and atmosphere on mars	-
3	Mars Science Laboratory's Curiosity rover	Donald M	New Technology to extract data	-
4	Mars Exploration Rover (MER) mission.	Samayad Hayati	Cutting edge technology to explore mars	-
5	Materials representing interactions with aqueous environments	John P. Grotzinger	Techniques to find information about various material interaction	-

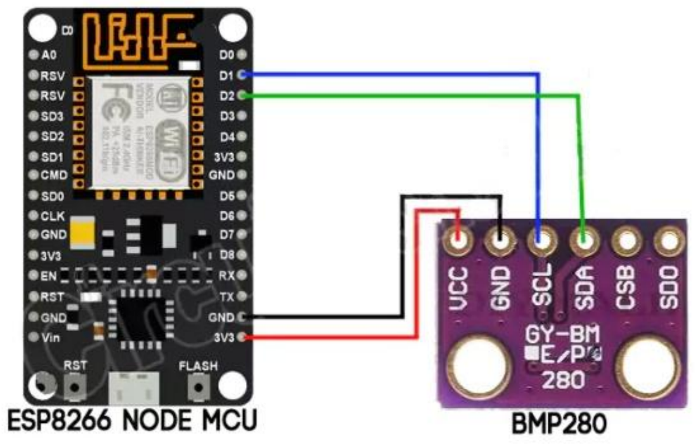
The survey papers mentioned above focus mostly on the different types of sensors that can be used for a rover and to gather temperature. To the best of our knowledge, There are no literature surveys that provide a holistic review of a amart rover system in terms of data acquisition, data analysis, data transport and storage, sensor networks, and Internet of Things (IoT) platforms, which are significant in the deployment of such systems.

1.1 OBJECTIVES AND GOALS

- Design and create a smart rover using an Arduino, that shows the temperature, altitude, pressure using ESP8266 & BMP.
- To make a hardware implementation of the same

Circuit diagram:





1.2 APPLICATIONS

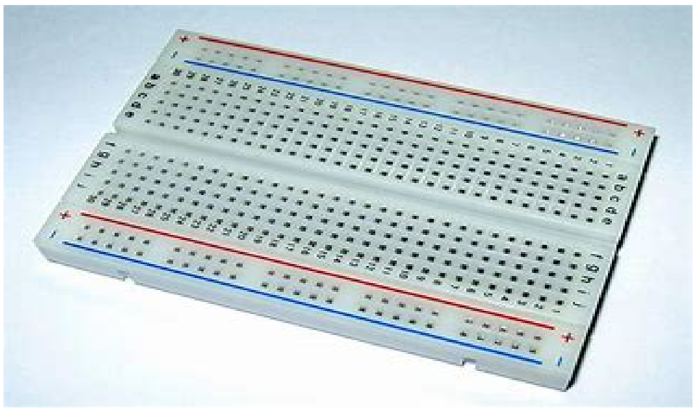
- Smart rovers are advancing, especially their design, battery capacity, and Research-related applications.

-Research-related applications include applications: Information extraction , Safety purposes etc.

Components required:

- **ARDUINO UNO**
- **BATTERY**
- **ESP8266**
- **SERVO MOTOR**
- **BMP SENSOR**
- **JUMPER WIRE**
- **BREADBOARD**
- **CHASSIS**
- **L298N DRIVER MODULE**
- **BLUETOOTH MODULE HC05**

- **BREADBOARD**



A breadboard consists of a plastic block holding a matrix of electrical sockets of a size suitable for gripping thin connecting wire, component wires or the pins of transistors and integrated circuits (ICs). The sockets are connected inside the board, usually in rows of five sockets.

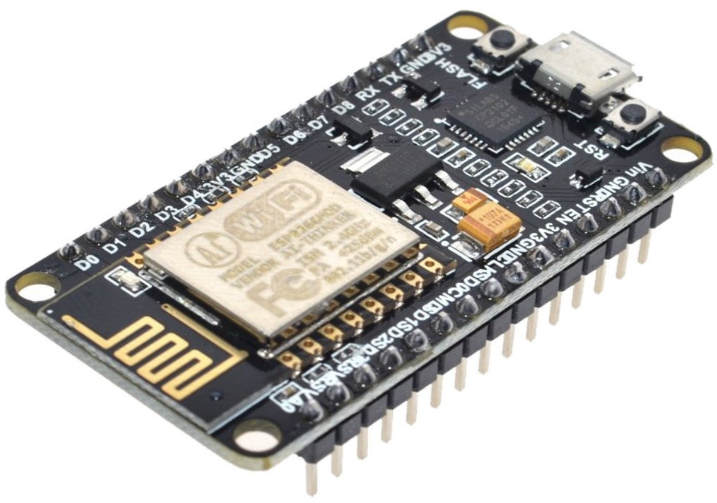
- **BATTERY**

Lithium-ion (Li-ion) batteries are used in many products such as electronics, toys, wireless headphones, handheld power tools, small and large appliances, electric vehicles and electrical energy storage systems.

- **<u>CHASSIS</u>**

Includes body of rover, wheels, etc

- **<u>ESP8266</u>**



The ESP8266 is a low-cost Wi-Fi microchip, with built-in TCP/IP networking software, and microcontroller capability

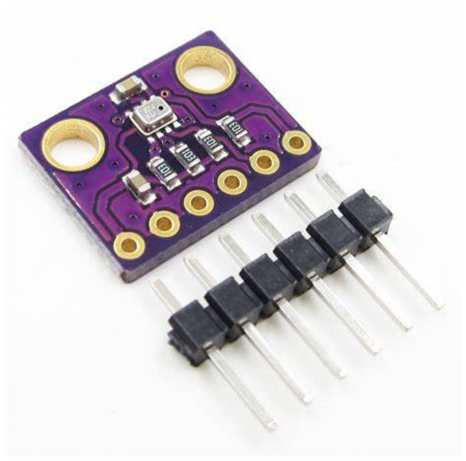
- **<u>JUMPER WIRES</u>**

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

- **ARDUINO UNO**

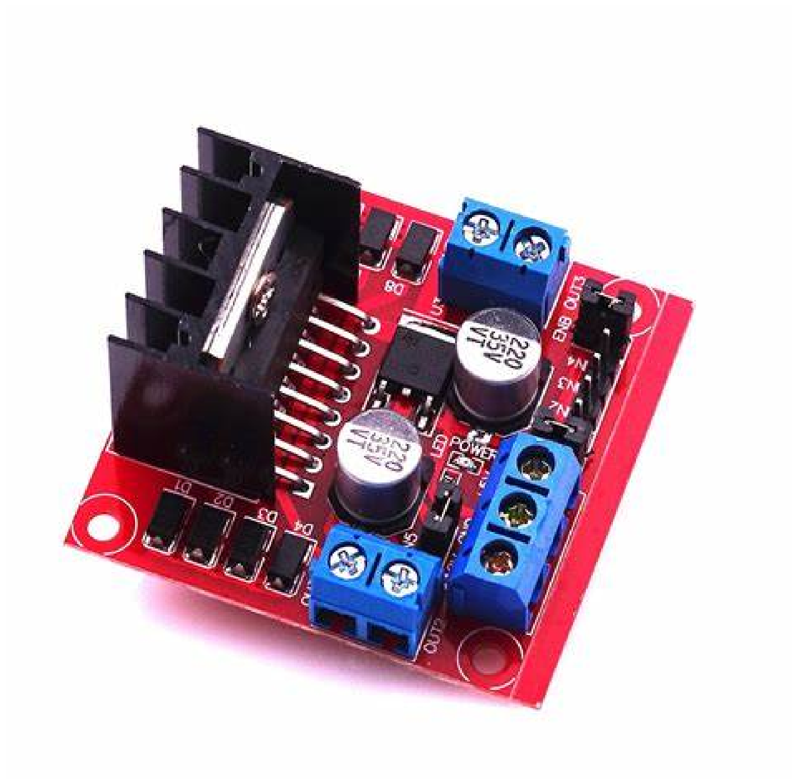
The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.

- **BMP SENSOR**



The BMP280 sensor integrates atmospheric pressure, temperature and relative humidity sensors in a single device, with great precision, low energy consumption and an ultra compact format. Based on BOSCH piezo-resistive technology with great EMC robustness, high precision and linearity, as well as long-term stability. It connects directly to a microcontroller via I2C or SPI.

- **L289 MOTOR DRIVER**



The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

- **DC GEAR MOTOR**

A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gear head to a motor reduces the speed while increasing the torque output.

- **Bluetooth Module HC05**

HC-05 Bluetooth Module is **an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup**. Its communication is via serial communication which makes an easy way to interface with a controller or PC.

<u>SOFTWARE COMPONENTS:</u>

- **Arduino IDE:**

ARDUINO

Arduino IDE or Arduino Integrated Development Environment is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards. This application works on all operating systems namely Windows, macOS, and Linux. Arduino IDE supplies a software library from the wiring project, which provides many common input and output procedures.

- **ThingSpeak**

ThingSpeak is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.

- **MATLAB:**

MATLAB is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

3. SOFTWARE –CODING AND ANALYSIS

FOR THE ROVER CONTROL

CODE:

```
//To include the GamePad module in the Arduino program, you have to

//include the following header:

#define CUSTOM_SETTINGS

#define INCLUDE_GAMEPAD_MODULE


//include Dabble app library

#include <Dabble.h>


#define motor1_en 10      // motor 1 enable pin

#define motor2_en 11      // motor 2 enable pin

#define motor1_dir1 4     // motor 1 input1 (InputA)

#define motor1_dir2 5     // motor 1 input2 (InputA)

#define motor2_dir1 6     // motor 2 input1 (InputB)

#define motor2_dir2 7    // motor 2 input2  (InputB)


void setup()

  {

    // put your setup code here, to run once:


    Serial.begin(9600);   //  start serial communication using 9600 baudrate

    Dabble.begin(38400); // Enter your bluetooth module baudrate

                    // NOTE : for arduino "UNO" use bluetooth module buadrate below 38400

    for(unsigned int i=4;i<8;i++)
```

```
        {
          pinMode(i,OUTPUT); // declaring input pins of motor1 and motor2 as a output pin
        }
      pinMode(motor1_en,OUTPUT);  // declaring enable pins of motor as a output
      pinMode(motor2_en,OUTPUT);




    }


void loop()
     {
       // put your main code here, to run repeatedly:

        Dabble.processInput(); //To refresh the data that the arduino UNO got from the mobile
app, you have to use the following line of code



       if (GamePad.isUpPressed()) // if UP is pressed in the gamepad then move robot forward
         {
           Serial.print("UP");
           forward();
         }

       else if (GamePad.isDownPressed()) // if DOWN is pressed in the gamepad then move
robot backward
         {
```

```
         Serial.print("DOWN");

         backward();

       }


     else  if (GamePad.isLeftPressed()) // if LEFT is pressed in the gamepad then move robot
LEFT

       {

         Serial.print("Left");

         left();

       }


     else  if (GamePad.isRightPressed()) // if RIGHT is pressed in the gamepad then move
robot RIGHT

       {

          Serial.print("Right");

          right();

       }


     else // stop the robot

       {

        Serial.println("strop");

        Stop();

       }




   }
```

```
void left()  // function for robot forward movement
{
  analogWrite(motor1_en,255);
  analogWrite(motor2_en,255);
  digitalWrite(motor1_dir1,HIGH);
  digitalWrite(motor1_dir2,LOW);
  digitalWrite(motor2_dir1,HIGH);
  digitalWrite(motor2_dir2,LOW);


}


void right() // function for robot backward movement
{
  analogWrite(motor1_en,255);
  analogWrite(motor2_en,255);
  digitalWrite(motor1_dir1,LOW);
  digitalWrite(motor1_dir2,HIGH);
  digitalWrite(motor2_dir1,LOW);
  digitalWrite(motor2_dir2,HIGH);


}
void backward() // function for robot left movement
{
  analogWrite(motor1_en,255);
  analogWrite(motor2_en,255);
  digitalWrite(motor1_dir1,LOW);
  digitalWrite(motor1_dir2,HIGH);

  digitalWrite(motor2_dir1,HIGH);

  digitalWrite(motor2_dir2,LOW);


}

void forward() // function for robot right movement

{

  analogWrite(motor1_en,255);

  analogWrite(motor2_en,255);

  digitalWrite(motor1_dir1,HIGH);

  digitalWrite(motor1_dir2,LOW);

  digitalWrite(motor2_dir1,LOW);

  digitalWrite(motor2_dir2,HIGH);


}

void Stop() // // function for no movement

{

  analogWrite(motor1_en,0);

  analogWrite(motor2_en,0);

  digitalWrite(motor1_dir1,LOW);

  digitalWrite(motor1_dir2,LOW);

  digitalWrite(motor2_dir1,LOW);

  digitalWrite(motor2_dir2,LOW);


}
```

For reading and sending data from BMP280 sensor to ThingSpeak using ESP8266.

<u>CODE:</u>

```
// ------style guard ----


#ifdef __cplusplus


extern "C"


{

#endif

uint8_t temprature_sens_read();

#ifdef __cplusplus

}


#endif


uint8_t temprature_sens_read();


// ------header files----


#include <ESP8266WiFi.h>


#include "ThingSpeak.h"


#include <Wire.h>
```

```
#include <Adafruit_BMP280.h>

#define BMP_SDA 2

#define BMP_SCL 1

Adafruit_BMP280 bmp280;

// -----network credentials

char* ssid = "Galaxy M312B5A"; // your network SSID (name)

const char* password = "00000000"; // your network password

WiFiClient client;

// -----ThingSpeak channel details

unsigned long myChannelNumber = 3;

const char * myWriteAPIKey = "UJL0LBCMD5AY744T";

// ----- Timer variables

unsigned long lastTime = 0;
unsigned long timerDelay = 1000;

void setup() {
  Serial.begin(115200); // Initialize serial

Serial.println("Initializing BMP280");
boolean status = bmp280.begin(0x76);
if (!status)
{
Serial.println("Not connected");
}

//Initialize Wi-Fi

WiFi.begin(ssid, password);

Serial.print("Connecting to Wi-Fi");

while (WiFi.status() != WL_CONNECTED)
{
Serial.print(".");
delay(100);
}

Serial.println();
Serial.print("Connected with IP: ");

Serial.println(WiFi.localIP());

Serial.println();


// Initialize ThingSpeak


ThingSpeak.begin(client);

}


void loop() {

  if ((millis() - lastTime) > timerDelay )

{

float temp = bmp280.readTemperature(); //temperature measurement


Serial.print("temperature: ");

Serial.print(temp);

Serial.println("*C");


float altitude = bmp280.readAltitude(1011.18); //altitude measurement


Serial.print("Altitude: ");

Serial.print(altitude);

Serial.println("m");


float pressure = (bmp280.readPressure()/100); //pressure measurement
Serial.print("Pressure: ");

Serial.print(pressure);

Serial.println("hPa");

Serial.println(" ");


ThingSpeak.setField(1, temp );

ThingSpeak.setField(2, altitude);

ThingSpeak.setField(3, pressure);


// Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8
different

// pieces of information in a channel. Here, we write to field 1.

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey );

if(x == 200)

{

Serial.println("Channel update successful." );

}

else

{

Serial.println("Problem updating channel. HTTP error code " + String(x) );

}

lastTime = millis();

}

}
```

MATLAB CODE (analysis):

```
clc;

clear;

close all;


readChId=1925613; % Channel ID

readKey="DID12AN6KRV5DXKE"; % Read API Key


[weatherData,timeStamp]=thingSpeakRead(readChId,'Fields',1,'Numpoints',3000,'Readkey
',readKey);


% Task-1 : plotting raw temp

figure;

plot(timeStamp,weatherData);

xlabel('Time');

ylabel('Temp (°C)');

legend('Temperature');

title('Raw Temp Data', 'fontsize',14,'fontweight','bold');

grid on;


%Task-2 - min & max of temp

fprintf('min temp is %d\n', min(weatherData));

fprintf('max temp is %d\n', max(weatherData));


%Task-3 - avg of temp

temp_sum = sum(weatherData);

avg_temp = temp_sum/length(weatherData);
```

```
fprintf('avg temp is %d\n', avg_temp);


% %Task-4 : histogram of temperature
figure;
histogram(weatherData);


% Task-5 : cleaning data (considering temperature which is less than 50 deg C) and plotting
cleanWeatherData = weatherData;
cleanTimeStamps = timeStamp;


l = length(weatherData);


for i = 1:l
   if (weatherData(i) > 50)
      cleanWeatherData(i) = sum(cleanWeatherData)/length(cleanWeatherData);
   end
end


% Task-5 : plotting cleaned data
display(cleanWeatherData, 'Cleaned data');
cleanWeatherData
figure;
histogram(cleanWeatherData);
figure;
plot(cleanTimeStamps,cleanWeatherData);
xlabel('Time');
```

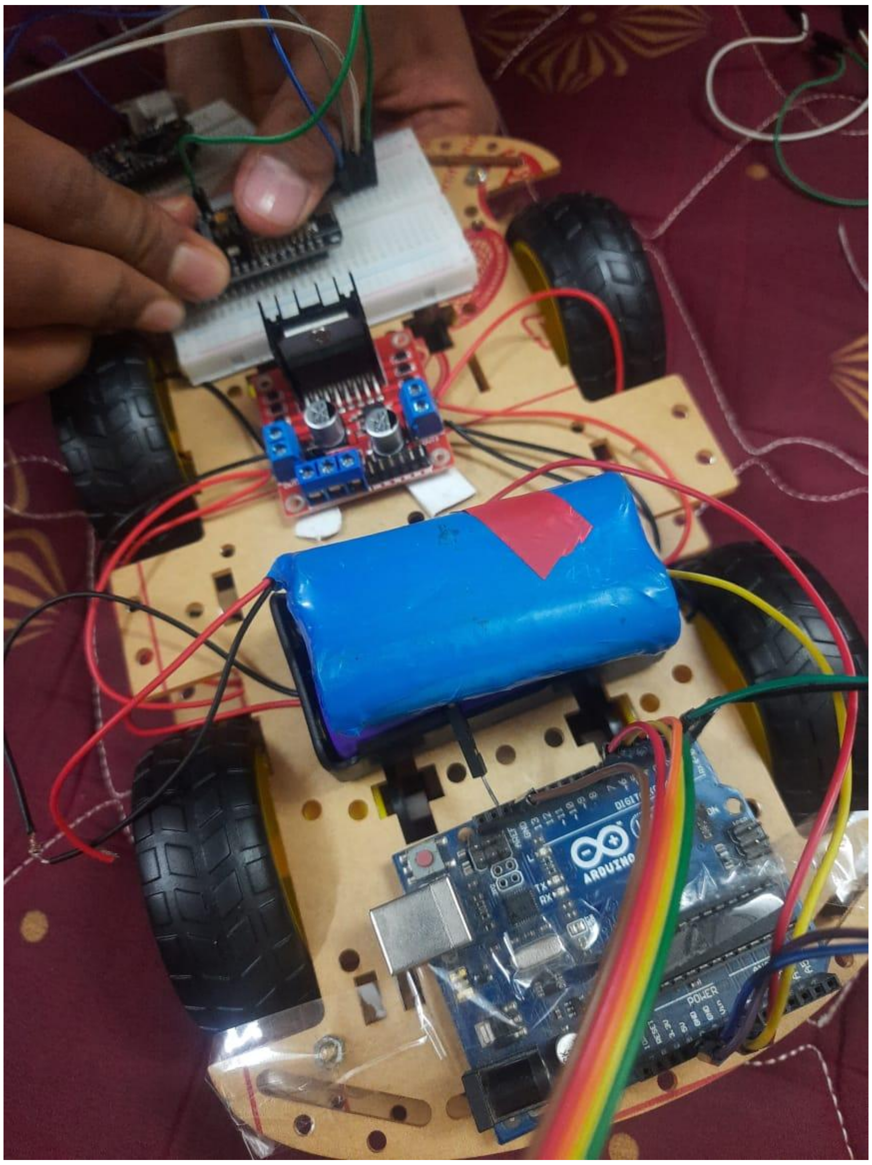
```
ylabel('Temp (°C)');

title('Cleaned Temp Data', 'fontsize',14,'fontweight','bold');

grid on;
```

4. CONCLUSION

A hardware design of the project, Smart rover, is made with all the components given, the connections in a bread board and the software code, and built a fully functioning smart rover.

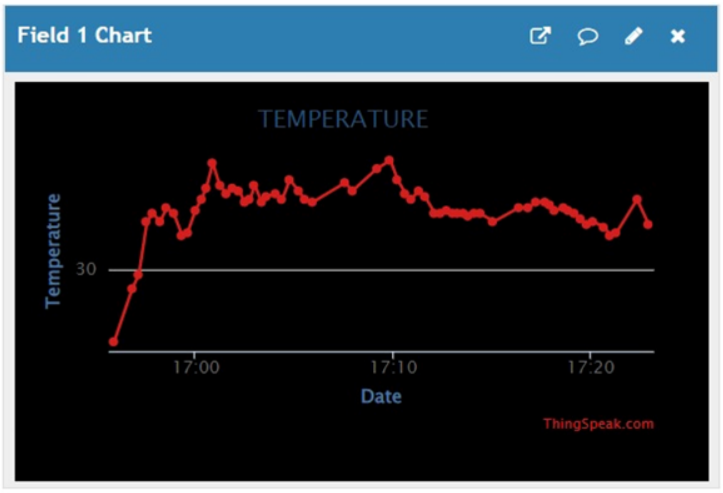
4.1 RESULT

HARDWARE:

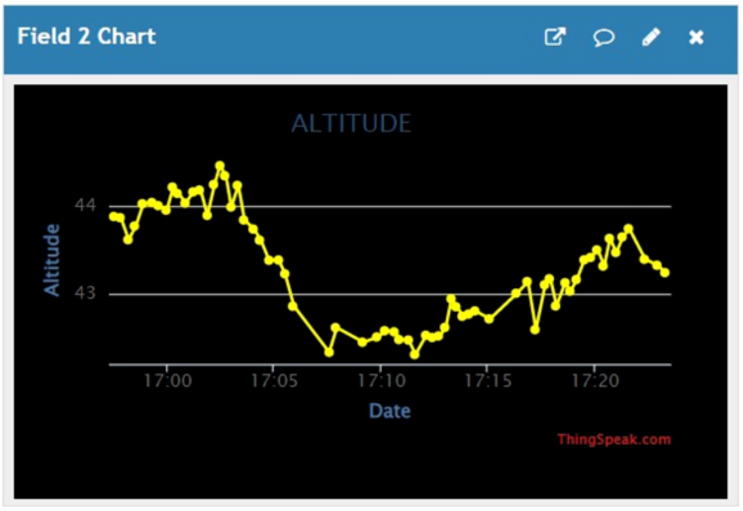


SMART ROVER

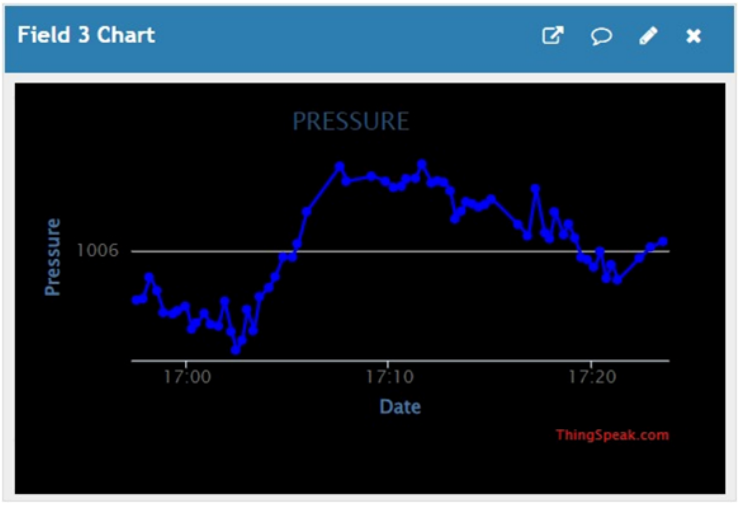
ThingSpeak:



Temp Reading (in deg Celcius, from BMP280)

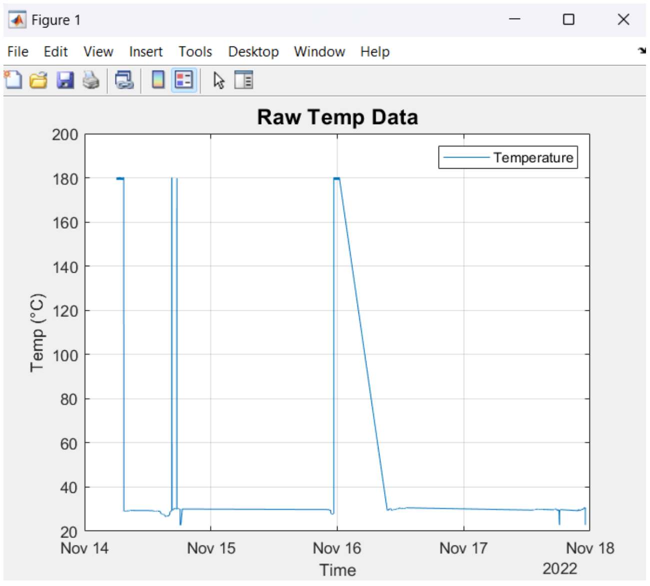


Altitude Reading (in meters, from BMP280)

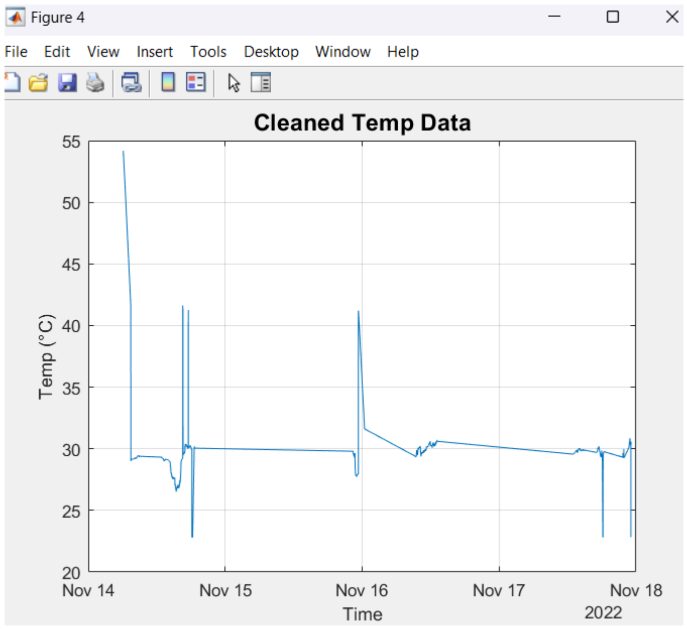


Pressure Reading (in hPa, from BMP280)

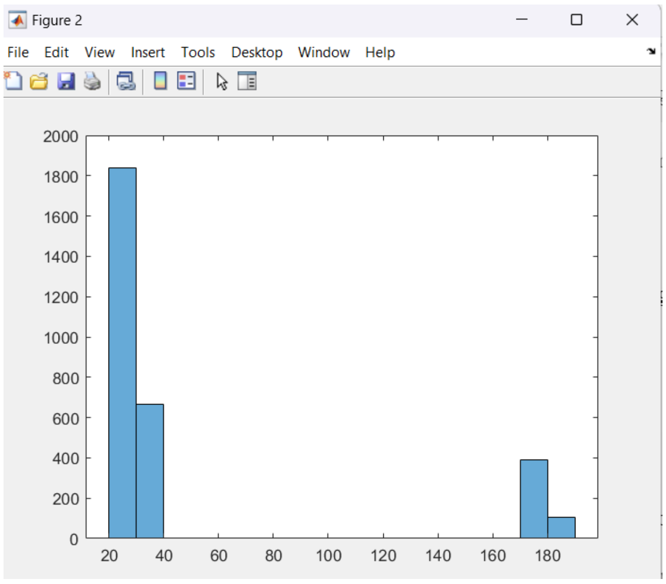
MATLAB:



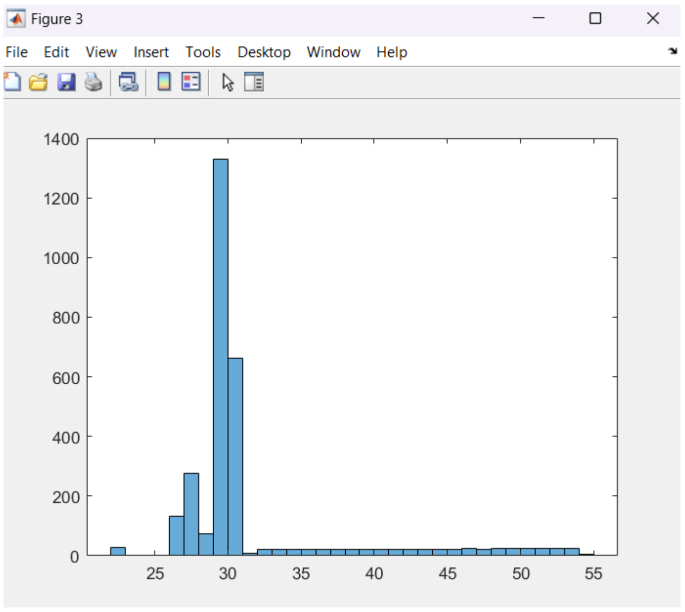
Raw Temp Data (from ThingSpeak)



Cleaned Tem Data (temp < 50 deg C)



Histogram (for Raw Temp Data)



Histogram (for cleaned Temp Data)

<u>Future work :</u>

With a better design of the body of the rover and minimising the space consumed,also we have planned to introduce a hand gesture controlling in the above model along with a proximity sensor that will enable a smooth control and automatic prevention from object collision this smart rover is ideal to be produced and be in the market with this product being ideal for research purposes such as people who test different kind of environments and also about the places which are inhabitable.

X.REFERNECES

- https://www.researchgate.net/publication/337388672_Rocker_Rover_And_Its_Implementation_In_The_Field_Of_Agriculture_A_Review?enrichId=rgreq-eb4e2b65d69019d4fb327ff45dff5fdeXXX&enrichSource=Y292ZXJQYWdlOzMzNzM4ODY3MjtBUzo4MjcyNDM5NDk0NjE1MDdAMTU3NDI0MTczNTkxNQ%3D%3D&el=1_x_2&_esc=publicationCoverPdf [PDF]

- Design and Development of an Intelligent Rover for Mars Exploration(Updated) (researchgate.net) Mars' Surface Radiation Environment Measured with the Mars Science Laboratory's Curiosity Rover | Science Analysis of Surface Materials by the Curiosity Mars Rover | Science

BIODATA

Name : Apurv Choudhary

Mobile Number : 9315224608

E-mail : apurvsinghchoudhary@gmail.com

Permanent Address: VIT Chennai

Name : Salugu Manoj

Mobile Number : 8074629945

E-mail : salugu.janakimanoj2020@vitstudent.ac.in

Permanent Address: VIT Chennai

Name : Pratyush Raj

Mobile Number : 7491899886

E-mail : pratyush.raj2020@vitstudent.ac.in

Permanent Address: VIT Chennai